Uso de Big Data para predecir riesgos en destinos turísticos

Alejandro Mejías Ríos
Grado en Ingeniería Informática

Tutor: Humberto Andrés Sanz

03/2015

FICHA DEL TRABAJO FINAL

Título del trabajo:	Uso de Big Data para predecir riesgos destinos turísticos
Nombre del autor:	Alejandro Mejías Ríos
Nombre del consultor:	Humberto Andrés Sanz
Fecha de entrega (mm/aaaa):	01/03/15
Área del Trabajo Final:	Big Data enfoque analítico
Titulación:	*Grado en Ingeniería Informática*

Resumen del Trabajo (máximo 250 palabras):

Recientemente se están desarrollando técnicas para poder ampliar el conocimiento que obtiene la empresa, a través de la información que genera, con nuevas fuentes de información externa y que son menos costosas de implantar para cualquier organización, por ejemplo, utilizando almacenamiento y procesamiento basado en Cloud o mediante plataformas de almacenamiento y de procesamiento de bajo coste. Al mismo tiempo estos datos se pueden integrar con la información que ya posee la organización a una velocidad cercana al tiempo real, por ejemplo utilizando tecnologías Big Data.

En el caso de los operadores turísticos se ve afectado periódicamente por pérdidas ocasionadas por las cancelaciones relacionadas con conflictos en las regiones de destino o por medidas tomadas por los gobiernos de las mismas (agencias de viaje y tour operadores prevén unas pérdidas de 350 millones de euros debido al bloqueo del tráfico aéreo), afectando en algunos casos a la propia integridad de los usuarios de estos servicios.

Es interesante disponer de alguna herramienta que pueda predecir con el suficiente tiempo de antelación esos problemas para que los turoperadores no ofrezcan servicios que vayan a ser cancelados o que proporcionen una mala experiencia a los clientes que provoque la pérdida de confianza y en consecuencia la pérdida del cliente.

Abstract (in English, 250 words or less):

New approaches had been developed recently to increase the knowledge that a enterprise get, through the information that it produce, with new sources of outer information at once they are less expensive when they are built up on any organization, ie., using data store and procesing on Cloud technologies or trough low-cost data and processing infrastructures. Also those data can be integrated, with the information systems that the organization already has, with a speed close to real time, ie. with Big Data technologies.

The particular scenary of the tour operator periodically had been hurt with loses debt to the trips cancellations because the conflicts that arise on the destiny areas or debt to changes related to changes promoted by the government of those zones, affecting in some cases, the own health of the clients of those services.

It is interesting to have tools that can predict, with the necessary anticipated time range, those problems to prevent the touroperators to offer services that will be cancelled or that provide a unpleasant experience to the clients that affect negatively the trust on the enterprise.

Palabras clave (entre 4 y 8):

analisis predictivo BigData turismo conflicto geopolitico

Índice

Lista de figuras

Índice de ilustraciones

Índice deTextos

1. Introducción

1.1. Contexto y justificación del Trabajo

Obtener un medio para poder realizar análisis predictivos sobre riesgos geopolíticos por regiones para reducir el riesgo financiero asumido por los turoperadores a la hora de ofertar destinos, así como para la mejora en la experiencia de los usuarios de estos servicios.
De momento no existe una herramienta o método para poder calcularlo.
Se quiere obtener una herramienta de fácil utilización y que dé resultados fiables para cumplir el objetivo previsto.

El desarrollo de nuevas arquitecturas y tecnologías para poder realizar análisis predictivos sobre grandes volúmenes de datos es un campo en auge en el sector de las TIC sobre todo por la demanda subyacente que existe en las organizaciones y en particular de las empresas que cuentan con un gran volumen de datos y que se están incrementando diariamente.

Un área donde se suelen aplicar este tipo de avances es en el área de prevención de riesgos financieros de organizaciones, desde hace años se vienen utilizando arquitecturas técnicas de BI mediante la integración de datawarehouses, y otras nuevas tecnologías con el sistema existente en las organizaciones, aplicando técnicas de minería de datos para hallar conocimiento a partir de los datos existentes.

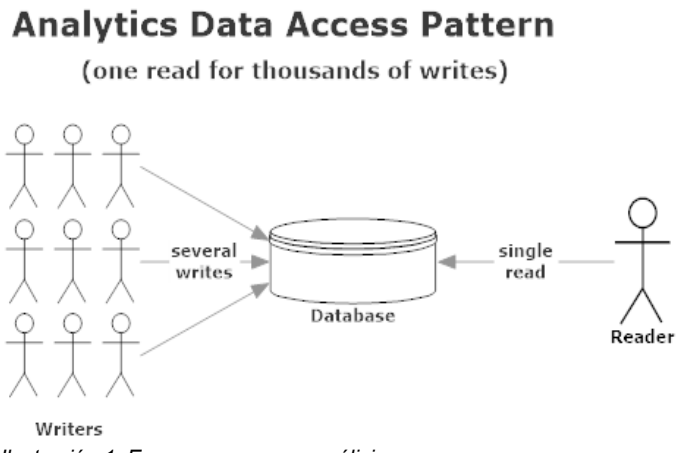
Sin embargo siempre han existido algunas deficiencias a la hora de extraer conocimiento y utilizar métricas adecuadas ya que las técnicas se aplicaban sobre información de la que ya disponía la organización, ignorando la información que existen en otras fuentes que no pertenecen a la organización o que no pueden gestionar directamente, por ejemplo repositorios de datos de noticias de medios periodísticos, Twitter, u otros.

Recientemente se están desarrollando técnicas para poder ampliar el conocimiento que obtiene la empresa, a través de la información que genera, con nuevas fuentes de información externa y que es menos costosa de implantar para cualquier organización, por ejemplo, utilizando almacenamiento y procesamiento basado en Cloud o mediante plataformas de almacenamiento y de procesamiento de bajo coste. Al mismo tiempo estos datos se pueden integrar con la información que ya posee la organización a una velocidad cercana al tiempo real, por ejemplo utilizando tecnologías Big Data.

Esto permite superar la limitación y reticencia existente en las organizaciones a la hora de extraer información desde repositorios de datos que hasta ahora no se podían utilizar debido al volumen de datos a tratar y el tiempo requerido para el procesamiento de los mismos y al prohibitivo coste que implicaba aumentar la infraestructura tecnológica que ya tienen.

En caso de que no tuviesen una infraestructura de inteligencia de negocios ya implantada este tipo de soluciones pueden ser una ventaja competitiva, ya que la integración de sistemas de información de la empresa se realiza a un menor coste que el necesario hasta hace unos años.

El análisis predictivo de riesgos es un área en la que cada vez más organizaciones están haciendo hincapié ya que evita pérdidas innecesarias que hasta ahora se producen debido a la falta de conocimiento que se puede obtener de la información que ya se posee o que está disponible.



(http://blog.markedup.com/2013/02/cassandra-hive-and-hadoop-how-we-picked-our-analytics-stack/)

Como caso particular de esta problemática se encuentran los operadores turísticos.
Las agencias de viaje y grandes [1] operadores turísticos son empresas que ofrecen producto o servicios turísticos, generalmente contratados por él, e integrados por más de uno de los siguientes items: transporte, alojamiento, traslados, excursiones, etc…

Este tipo de negocio ha ido incrementando su volumen de negocio desde mediados del siglo anterior y la tendencia, aunque atenuada continúa.

A pesar de ello este tipo de negocio se ve afectado periódicamente por pérdidas ocasionadas por las cancelaciones relacionadas con conflictos en las regiones de destino o por medidas tomadas por los gobiernos de las mismas [2], afectando en algunos casos a la propia integridad de los

usuarios de estos servicios [3] [7].

Este tipo de sucesos repercuten negativamente en los beneficios que obtienen las empresas de este sector, un sector que diariamente se están generando informaciones desde medios de información tradicional y online que aumentan sensiblemente el conocimiento que se tiene del mundo, tanto a nivel general como a nivel regional, estos datos se están empezando a utilizar para obtener información en tiempo real y para tener información histórica de sucesos recogidos en estos medios. A pesar de posible manipulación de noticias [4] que realicen algunos medios de información informan sobre hechos innegables que se producen y que se pueden utilizar para poder conocer la tendencia histórica de sucesos.

Continuamente están surgiendo nuevos proyectos de captación de información global realizar proyectos que capitalicen la información humana para poder extraer conocimiento desde datos ya existentes [J.E. Yonamine, 2011], en concreto el proyecto GDELT que utiliza la taxonomía CAMEO.

Esta base de datos se actualiza diariamente con lo que su volumen de datos se va incrementando de manera progresiva día a día, en consecuencia el volumen de datos que ofrece se está volviendo gigantesco que dependen en gran medida de las condiciones de los destinos para ofrecer servicios y destinos que cumplan las expectativas de sus clientes.

Durante los últimos años la cantidad de información generada, principalmente debido al auge de las tecnologías de computación ubicua, el e-commerce y las social media, se ha incrementado hasta el punto de que “en los últimos 10 años se ha creado más información que en toda la historia de la humanidad” Según el informe de la Online Business School “estudio Big Data en números 2014”[8].

Es interesante disponer de alguna herramienta que pueda predecir con el suficiente tiempo de antelación esos problemas que para que los turoperadores no ofrezcan servicios que vayan a ser cancelados o que proporcionen una mala experiencia a los clientes que provoque la pérdida de confianza y en consecuencia la pérdida del cliente.

Los algoritmos de minería de datos se clasifican en dos grandes categorías: supervisados o predictivos y no supervisados o de descubrimiento del conocimiento [Weiss e Indurkhya, 1998].

En caso de que los modelos predictivos no produzcan resultados esperados, es decir, no tiene el potencial necesario para una solución predictiva, en ese caso hay que recurrir a los métodos no supervisados La información obtenida es fácilmente traducible en beneficio para la organización.

Encontrar una solución que sea capaz de anticipar los conflictos o cambios políticos en materia de turismo que puedan afectar a los clientes o que evite que se contraten servicios a lugares que vayan a provocar la cancelación de los mismos será una herramienta cuyo retorno de inversión será muy alto, además de evitar las pérdidas derivadas de la cancelación.

Para ello es imprescindible anticiparse a las alteraciones políticas que se producen, y puesto que los cambios políticos se producen en un corto intervalo de tiempo lo más adecuado es realizar predicciones acertadas con los datos de los que se dispone. Es por ello una buena aproximación utilizar arquitecturas BI para realizar análisis predictivos sobre el estado político de destinos turísticos.

Para ello es necesario disponer de acceso a datos sobre noticias a nivel mundial y aplicar técnicas adecuadas para obtener modelos predictivos mediante técnicas de minería de datos, de inteligencia de negocio e incluso aprovechando la información disponible en los social media.

Puesto que el repositorio de datos GDELT se está actualizando diariamente el volumen actual y futuro de datos va a ser ingente y puesto que las peticiones que se vayan a realizar se han de poder resolver en un corto intervalo de tiempo (se puede predecir a largo plazo o consultar predicciones para destinos en la agencia de viaje), la solución puede utilizar no sólo un repositorio de datos, sino los datos disponibles en los social media para obtener un modelo predictivo más preciso, En este caso tenemos las tras Vs que recomienda utilizar las tecnologías Big Data, Volumen, de datos Variabilidad de datos y velocidad de procesamiento.

Se necesita pues definir un sistema de análisis predictivo para previsión de riesgos en las organizaciones pero aplicado a la previsión de riesgos geo-políticos aplicados a los turoperadores. Utilizando tecnologías de almacenamiento y procesamiento masivo de datos a bajo coste que permitan realizar consultas NoSQL, por ejemplo Hadoop, MongoDB o Cassandra y utilizando alguna suite para general inteligencia de negocios que permita realizar análisis predictivos, como por ejemplo Pentaho o Qlik

1.2. Objetivos del Trabajo

Realizar un análisis predictivo para detectar riesgos geopolíticos en distintas áreas.

Maximizar el retorno de inversión a la hora de implantar el sistema.

Auxiliar al cuadro estratégico de los operadores turísticos.

Diseñar la plataforma robusta y ajustada a las necesidades de la herramienta.

1.3. Enfoque y método seguido

No he encontrado un producto existente que aborde esta tarea, sin embargo se puede abogar por utilizar herramientas Opensource o bien utilizar herramientas propietarias.

He escogido utilizar herramientas OpenSource para reducir los costes de implantación y por resultados de pruebas de rendimiento [9], [10]. En este caso serán Hadoop y Weka 3.7 como parte de la inteligencia de la aplicación y Java como herramienta de la interfaz de trabajo.

Mediante la utilización de una tecnología de este tipo se puede proporcionar al área de planificación estratégica de la organización una herramienta que permita anticiparse a pérdidas o abrir nuevos nichos de negocio. Al ser una herramienta de análisis predictivo se puede utilizar en los departamentos de análisis de riesgos, de análisis financiero, de marketing

Como riesgo inicial es pensar que esta herramienta no va a requerir de supervisión de especialistas para evaluar los resultados, por ello es imprescindible la colaboración de departamentos y personas adecuadas para interpretar adecuadamente los resultados.

Aparte de lo anterior otro riesgo que se pueden identificar en el uso de este tipo de herramientas es la fuente de datos que proporcionan los medios de comunicación.

Sin embargo si se entiende esta herramienta como lo que es, una herramienta para ayudar a dar valor y prevenir costes en la organización el retorno de inversión al utilizarla se puede cuantificar en un corto periodo de tiempo. Además puesto que este tipo de tecnologías permiten aprender conforme se van acumulando datos conforme pase el tiempo el comportamiento de la misma será más preciso y producirá mejores resultados.

1.4. Planificación del Trabajo

Hito 1

- Determinación del problema a resolver.
- Determinación de objetivos.
- Estudio de herramientas similares en el mercado.
- Valoración de herramientas y plataformas a utilizar.
- Conocimiento de las herramientas a utilizar:
  - HADOOP (framework de procesamiento y almacenamiento de datos)
  - GDELT (Repositorio de datos)
  - CAMEO (Taxonomía utilizada por GDELT)
  - Weka 3.7.

    - Java 7.
    - Eclipse IDE for Java Developers Version: Kepler Service Release 2.
    - Oracle VM VirtualBox, diferentes versiones.
    - OpenOffice 4.1.0.

- Elaboración de calendario de hitos y actividades.
- Elaboración de PEC1
- Preparación del entorno de desarrollo y la plataforma de producción

Hito 2

- Comparar objetivos iniciales con los actuales.
- Realización del análisis de requisitos
- Diseño del almacén de datos.
- Fase de extracción y preparación de datos desde GDELT.
- Fase de análisis de resultados mediante minería de datos Mediante Weka 3.7.
- Conclusiones iniciales y retroalimentación de resultados.
- Elaboración de PEC2

Hito 3

- Diseño de las interfaces de acceso y consulta
- Creación del producto.
- Pruebas de rendimiento.
- Conclusiones finales.
- Elaboración de PEC3

Hito 4

- Redacción de memoria final.
- Producción de video-presentación del proyecto.
- Entrega de memoria final.

1.5. Breve sumario de productos obtenidos

- Aplicativo para obtener el porcentaje de riesgo de conflicto social en un país para un determinado día.
- Evaluación de herramientas adecuadas para realizar la tarea
- Estudio teórico de los resultados esperados.
- Estudio práctico de rendimiento al realizar el análisis predictivo de riesgos geo-políticos.

1.6. Breve descripción de los otros capítulos de la memoria

Estado del arte en BigData y análisis predictivo: Estudio resumido del desarrollo de esta tecnología.

Detectar las señales de conflictos sociales en publicaciones: Medios de detección de señales de conflictos en publicaciones online.

Arquitectura e infraestructura: Definición, justificación y explicación de la elección de la arquitectura del sistema así como de los productos utilizados para desarrollarlo.

Implementación: Justificación y explicación del diseño de la solución propuesta.

Trabajos relacionados: Trabajos que desarrollan estudios similares al del trabajo actual.

Estudio de rendimiento: Comparativa del uso de diferentes implementaciones de BigData (número de clusteres usados, etc...)

2. Estado del arte en BigData y análisis predictivo [36].

El análisis predictivo es la rama de minería de datos que tiene relación con la predicción de las probabilidades y tendencias futuras. Permite extraer conclusiones confiables sobre eventos futuros, a través de la aplicación de métodos estadísticos, matemáticos y de reconocimiento de patrones.

El elemento central del análisis predictivo es el *predictor*, una variable que puede ser medida para una entidad individual o de otro tipo para predecir el comportamiento futuro. Por ejemplo, en una compañía de seguros es probable que se tengan en cuenta los posibles predictores de conducción de seguridad, tales como la edad, el género y registro de conducir, al momento de la cotización de pólizas de seguros de automóviles.

Predictores múltiples se combinan en un modelo predictivo que, cuando se somete a análisis, se puede utilizar para predecir probabilidades futuras con un nivel aceptable de fiabilidad. En modelos de predicción, se recopilan los datos, se formula un modelo estadístico, se hacen las predicciones y el modelo se valida, con los datos adicionales que estén disponibles. El análisis predictivo se aplica a muchas áreas de investigación, incluyendo la meteorología, la seguridad, la genética, economía y marketing.

2.1. Pila de BigData

En el área de BigData se pueden tener una gran cantidad de atributos predictivos mediante una cantidad inmensa de observaciones.

Mientras que en el pasado se hubiese necesitado de gran cantidad de horas para ejecutar un modelo predictivo, con una gran cantidad de datos en tu dispositivo, ahora es posible ejecutar, de forma iterativa, estos modelos cientos de veces si se dispone de una infraestructura BigData.

La pila BigData es el diseño de las capas que componen una infraestructura BigData. Un diseño de Infraestructura BigData puede suponer un alto coste en tiempo de diseño e implementación pero llega a ahorrar mucho más tiempo en el desarrollo y evita futuras frustraciones si se hace con un punto de vista holístico.

La arquitectura debe sostener los requerimientos fundamentales en los que se apoya, que son capturar, integrar, organizar, analizar y actuar.

El ámbito del presente trabajo no tendrá en cuenta el diseño de infraestructuras BigData, sin embargo haré una somera explicación de las distintas capas que componen la pila de BigData.

En el nivel más bajo se encuentra una capa física redundante, en donde se debe proporcionar un sistema de alta disponibilidad tanto a nivel de almacenamiento y comunicaciones como a nivel de procesamiento y memoria, ya que el propio principio de BigData se basa en implementaciones distribuidas tanto a nivel lógico como físico.

Se ha de tener en cuenta el rendimiento para reducir la latencia, pero las infraestructuras que sean muy rápidas tienden a ser muy costosas.

Otro factor a tener en cuenta el la disponibilidad, para ello se debe alcanzar un compromiso entre la disponibilidad que requiere el sistema y el coste del mismo, como en el caso anterior a mayor tiempo de disponibilidad mayor suele ser el coste del soporte físico.

El tercer factor a tener en cuenta es la escalabilidad tanto a nivel horizontal como vertical, para ello también se debe llegar a un compromiso entre la capacidad de procesamiento que ha de proporcionar el sistema, teniendo en cuenta siempre que se ha de proporcionar un porcentaje de escalabilidad de más para las futuras demandas de procesamiento.

Otro factor crítico a tener en cuenta es la flexibilidad, que está muy relacionada con la escalabilidad, ya que los sistemas que vayan a proporcionar soporte BigData tienden a requerir cada vez más capacidad de almacenamiento y procesamiento al incrementarse continuamente la cantidad de información a gestionar y por supuesto, tal y como se ha ido comentando en cada uno de los factores anteriores el coste es un factor crítico a la hora de diseñar una infraestructura BigData.

El siguiente nivel de la pila será la seguridad de la infraestructura, donde se necesitan unos requerimientos similares a los de entornos de datos estándar, se ha de asegurar el acceso a datos, el acceso a nivel de aplicación, se ha de proteger la información mediante técnicas de encriptación y se ha de preparar el sistema para que se detectan las amenazas a la seguridad.

La siguiente capa de la pila son los motores de base de datos operacionales, los cuales ha de garantizar el que sean rápidos, escalables y robustos.

El siguiente nivel en las capas de la pila será el de la organización de servicios de datos y de herramientas para BigData que permitan validar y unir diferentes elementos BigData en colecciones relevantes contextualmente como por ejemplo utilizando técnicas de MapReduce para gestionar datos desde almacenamientos distribuidos y produciendo una colección única para ser tratada sin necesidad de gestionar cada fuente de datos individualmente.

La última capa son los almacenes de datos analíticos que permiten simplificar la generación de informes y también la visualización de datos ingentes. Estos almacenes se suelen crear casi desde cualquier arquitectura de almacenamiento, como bases de datos relacionales, bases de datos multi-dimensionales, ficheros planos y objetos de bases de datos.

2.2. Virtualización

Ha sido gracias a la virtualización el desarrollo de las tecnologías Cloud y BigData, ya que incrementan el porcentaje de utilización de recursos, la eficiencia y la escalabilidad reduciendo tanto coste de implementación como de administración, así como el coste de la infraestructura física, al menos potencialmente.

La virtualización permite aprovechar las infraestructuras IT para poder realizar análisis de grandes cantidades de datos ganando en eficiencia de procesamiento y gestión de grandes volúmenes de datos.

BigData implica acceder, gestionar y analizar datos estructurados y no estructurados en entornos distribuidos y en la práctica se ha comprobado que las tareas MapReduce funcionan mejor en entornos virtualizados.[29]

2.3. Hadoop como motor de búsqueda

Hadoop ha sido desarrollado debido a que representa el método más pragmático para permitir a las compañías gestionar ingentes cantidades de datos fácilmente. Lo que permite Hadoop es dividir problemas grandes en elementos más pequeños para que el análisis pueda ser realizado rápidamente y a bajo coste.

Hadoop es un sistema de almacenamiento seguro y de análisis. El almacenamiento lo proporciona el sistema de ficheros HDFS y el análisis lo proporcionan las tareas MapReduce, y esto es el núcleo del mismo [39].

Un framework útil que se utiliza sobre Hadoop es Hive, el cual fue creado para hacer posible el análisis cuando se tiene gran formación sobre SQL, pero pocas programando en Java para ejecutar consultas de datos almacenados sobre sistemas de ficheros HDFS y que utilizan muchas compañías como plataforma de procesamiento escalable de propósito general[39].

3. Detectar las señales de conflictos sociales en publicaciones

[32,33]El análisis de eventos que indican una manifestación o conflicto social es un concepto establecido en el desarrollo de ciencias sociales. El concepto de disturbio social trata de englobar las distintas formas en las que la gente expresa su protesta ante hechos que afectan a sus vidas y que relacionan con actuaciones realizadas por el gobierno, tanto local como nacional, o que este tiene algún tipo de responsabilidad en los mismos, como por ejemplo la subida de impuestos o la subida de los precios de los alimentos básicos.

En caso de que la protesta se realice en contra de un agente privado suele haber una conexión entre la política o el comportamiento gubernamental, por ejemplo una huelga en una empresa que altera el ritmo diario del resto de la sociedad. No se considera disturbio social aquellos eventos que se producen por criminales que buscan su propio beneficio, aunque, como actualmente pasa en México con los carteles, pueden ser un factor a tener en cuenta.

Los disturbios se pueden preparar de forma organizada o como una respuesta espontánea a un hecho concreto. El problema de predicción que planteo utilizando indicadores y herramientas Opensource será pues una herramienta valiosa para aquellas compañías que ofrezcan servicios turísticos en gran variedad de destinos.

Utilizando como fuente de datos principal los datos que proporciona el proyecto GDELT a través de ficheros con eventos estructurados va a servir como base para poder realizar las labores de análisis y así poder obtener los resultados buscados por la herramienta.

Mediante la taxonomía CAMEO se clasifican los eventos en los siguientes tipos:

- MAKE PUBLIC STATEMENT
- APPEAL
- EXPRESS INTENT TO COOPERATE
- CONSULT
- ENGAGE IN DIPLOMATIC COOPERATION
- ENGAGE IN MATERIAL COOPERATION
- PROVIDE AID
- INVESTIGATE
- DEMAND
- DISAPPROVE
- REJECT

- THREATEN
- EXHIBIT MILITARY POSTURE
- REDUCE RELATIONS
- COERCE
- YIELD
- PROTEST
- ASSAULT
- FIGHT
- ENGAGE IN UNCONVENTIONAL MASS VIOLENCE

Puesto que lo que busco es predecir la posibilidad de conflicto consideraré los tipos de evento PROTEST, ASSAULT, FIGHT, NGAGE IN UNCONVENTIONAL MASS VIOLENCE.

Como disturbio provocado y todos los anteriores como eventos que conducen a un evento disturbio y los utilizaré como predictores.

4. Arquitectura e infraestructura

Diseñar una arquitectura y diseño de sistema que sea un sistema de procesamiento modular, el sistema deberá funcionar en un periodo 24x7 sin intervención humana.

El sistema utilizará datos altamente estructurados indicando cuando se va a producir una protesta, a que es debida la protesta y la probabilidad de que se produzca dicha protesta.

El sistema adoptará una aproximación con un único modelo desde una fuente de datos para de forma independiente generar predicciones y como esas predicciones se ajustan a las advertencias finales.

4.1. Arquitectura del sistema

Para el presente trabajo, debido a las restricciones de tiempo y de formación que se me presentan recurriré a una solución de plataformas virtualizadas con toda la infraestructura empaquetada.

4.1.1. Infraestructura

Evaluaré una plataforma Open Source (Hortonworks Sandbox) y una propietaria (Cloudera 5.3) para utilizarlos como soporte para el análisis predictivo que es la base del presente trabajo.

Para ello he recurrido a distribuciones de máquinas virtualizadas que ofrecen ambas compañías, a saber: Hortownworks HDP 2.2.4 Sandbox with ambari and Apache Sparks y Cloudera QuickStart VMs for CDH 5.3.0 cuyas características básicas se detallan en el anexo III y en el anexo IV respectivamente.

La plataforma Hortonworks Sandbox ofrece unas características similares a la ofrecida por Cloudera. Sin embargo la cantidad de información y tutoriales ofrecidos por Hortonworks, la formación y la simplicidad que ofrece para trabajar en el entorno es la que he escogido.

4.1.2. Herramientas de análisis

Utilizaré una herramienta que permita acceder a infraestructuras BigData, en concreto la versión 3.7 de Weka, ya permite la conexión a datos distribuidos de Hadoop a través del conector JDBC Hive de forma nativa, además permite utilizar otros conectores JDBC, que es lo que se ha hecho en esta implementación.

Mining Big Data usando Weka 3[35]

Hay una concepción generalizada respecto a que el software de aprendizaje computacional de Weka no se puede usar en grandes conjuntos de datos. El problema principal radica en el entrenamiento de modelos desde grandes grupos de datos, no en la predicción en sí misma.

Este software sí que tiene problemas a la hora de utilizar la interfaz gráfica con grandes conjuntos de datos ya que sí que carga todos los datos en memoria lo que conlleva una sobrecarga de memoria inaceptable, sin embargo sí que es cierto que al poder usarse mediante línea de comandos o integrándola en aplicativos desarrollados en Java produce los resultados buscados, siempre que se tenga en cuenta las limitaciones que posee a la hora de trabajar con los volúmenes de datos que proveen las tecnologías BigData.

4.1.3. Arquitectura

El sistema se dividirá en cuatro módulos principales a saber:

- Obtención de datos
- Enriquecimiento de datos
- Modelo de predicción
- Generación de la predicción

que se detallan a continuación

4.1.3.1. Obtención de datos:

Alimentación de datos:

El proyecto GDELT proporciona una fuente de datos estructurada sin procesar que identifica eventos que se producen en países diariamente. Mediante la ontología CAMEO, clasifica e identifica el tipo de evento que se produce en un día y país concreto. Cada mañana, durante los siete días de la semana se genera y se pone a disposición pública a las 6AM EST, un fichero que contiene los eventos recogidos en el anterior periodo de 24 horas.

Descarga de datos y carga de información en el sistema:

Creando una tarea programada se descarga el fichero, y se agregará al sistema de ficheros Hadoop para que sea utilizado en el análisis predictivo.

4.1.3.2. Identificación de predictores y enriquecimiento de datos

Puesto que la cantidad de eventos por día/país es muy grande y en aras de optimizar el tiempo de ejecución he aplicado técnicas de minería de datos para reducir el conjunto de datos, y para ello me apoyo tanto en la información aportada por GDELT como en los resultados empíricos y el sentido común, a saber:

Tal como he apuntado anteriormente, voy a basar la identificación de eventos que ya son protestas en sí. Usaré estos eventos, y su correspondiente atributo, para entrenar el modelo predictivo.

Lo que persigo es hallar la secuencia de eventos que no son de tipo protesta en sí mismo, pero que preceden a protestas definitivas, para poder predecir los eventos de conflicto, con la adecuada antelación, que es lo que pretendo conseguir al desarrollar la herramienta.

Extracción de identidades:

Se considerarán pertenecientes al mismo grupo de elementos aquellos que afecten al país a evaluar así como aquellos países que pertenezcan al mismo tipo que el país estudiado, con esto se incrementará la cantidad de muestras para entrenar el modelo predictivo.

Selección de atributos:

El conjunto completo de atributos que se obtienen de la fuente de datos de GDELT se detallan en el Anexo I:
GDELT establece las relaciones entre dos actores, el actor uno es el que realiza la acción identificada por el evento y el actor2 es el que recibe los efectos de la acción.

Reducción del conjunto de datos:

En el conjunto de datos no hay atributos duplicados, sólo especializaciones de algunos de ellos.

Existen atributos superfluos para el proyecto, como son: *actor1knowngroupcode, actor1Ethnitcode, actor1religion1code, actor1religion2code, actor2knowngroupcode, actor2Ethnitcode, actor2religion1code, actor2religion2code*, puesto que no están relacionados con la tarea de predicción puesto que interesa saber que se produjo un evento en cierto país, no quien lo provocó, no aportan información nueva.

- *Selección de atributos:* Mediante el análisis de grupos de atributos. Lo que busca es grupos de atributos que puedan ser sustituidos por un atributo único.

Atributos que son dependientes entre sí son los relacionados con el grupo de atributos que informan de ubicación física del país y datos informativos relacionados, a saber:
actor1geo_fullname, actor1geo_type, actor1geo_adm1code, actor1geo_featureid, actor1geo_Lat, actor1geo_Long, actor1geo_type, actor1geo_countrycode, actor1code y actor1name, escojo actor1geo_countrycode.

actor2geo_fullname, actor2geo_type, actor2geo_adm1code, actor2geo_featureid, actor2geo_Lat, actor2geo_Long, actor2geo_type, actor2geo_countrycode, actor2code y actor2name, escojo *actor2geo_countrycode.*

La escala goldstein asignada, *goldsteinscale*.

El tono medio del evento, *avgtone*.

El número de artículos generados por evento, *numarticles*, *numsources*, *nummentions*. De estos tomaré el atributo *numarticles*.

El indicador de si es un evento raíz, *isrootevent*.

El grupo de atributos relacionados con la clasificación CAMEO del evento, *eventcode, eventobasecode, rooteventocode*, Como busco el tipo de evento base que se genera en el conjunto de datos sólo escojo *eventorootcode*.

Seleccionaré atributos mediante una prueba de significación, o sea comprobar si un atributo es relevante para el modelo o no según sus valores, y para ello se compararan clases de un atributo dos a dos para ver si su media dentro del total de datos es la misma. Si es la misma la significación de ambas clases es la misma. Este método es útil si los atributos son independientes entre sí.

En este caso puesto que *actor2geo_countrycode* y *actor1geo_countrycode* son atributos independientes de los demás. La prueba de significación indica que ambas clases no son la misma, sin embargo no es necesario obtener información de *actor2geo_countrycode* puesto que nos interesa saber los eventos que se generaron en *actor1geo_countrycode* que es el país buscado.

Eventrootcode es un atributo independiente de los demás que nos proporciona una especialización de la información aportada por *rooteventocode*, con lo que se puede obviar
Siguiendo el trabajo realizado por N. Kallus [32] se simplificarán las señales precursoras, a los que llamaré señales de protesta, que se encuentran en los eventos recogidos respecto a los eventos que considero conflictos, que es el evento que se trata de discriminar.

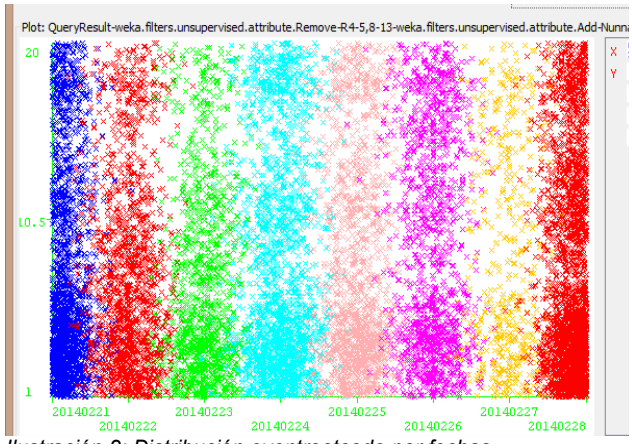


Imagen de la distribución del valor del atributo eventrootcode
a lo largo de los días 21/02/2014 hasta 28/02/2014 en Ucrania

Identificación de predictores:

Es importante predecir la cronología de eventos que conducen a un disturbio, con lo que me interesa poder conocer la fecha en la que se produce el evento y la fecha en la que se recogió el evento , que serán indicados como campo *sqldate*.

Un campo que es importante para poder realizar tareas predictivas es *IsRootEvent* ya que, tal como se comenta en GDELT, proyectos previos han encontrado que los eventos que aparecen en la cabeza del párrafo de un documento tienden a ser los más importantes y son menos propensos a contener errores.

Rooteventcode es un campo que indica el código raíz CAMEO del evento recogido en la observación, lo que es muy útil a la hora de reducir la complejidad del código a realizar.

GoldsteinScale. Es un campo importante ya que indica que efecto pude tener en la estabilidad del país el evento de la muestra.

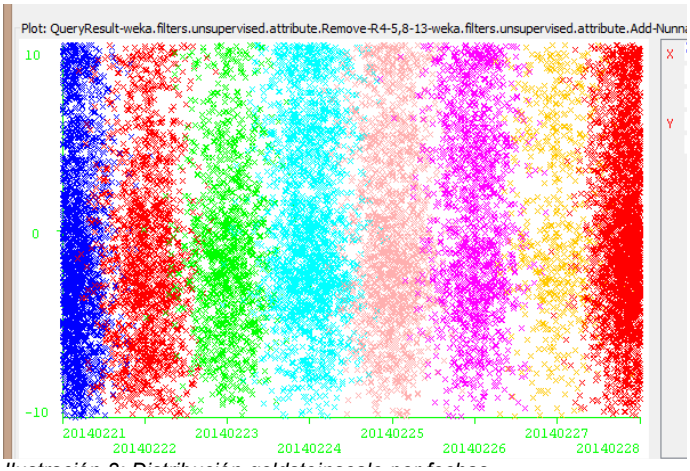


Imagen de la distribución del valor del atributo goldsteinscale
a lo largo de los días 21/02/2014 hasta 28/02/2014 en Ucrania

NumArticles. Otro campo que se puede utilizar para asignar importancia a un evento es la cantidad de artículos que están relacionados con el evento. Se recomienda normalizarlo a la media de otras medidas del universo de eventos que se recogen en el periodo de interés.

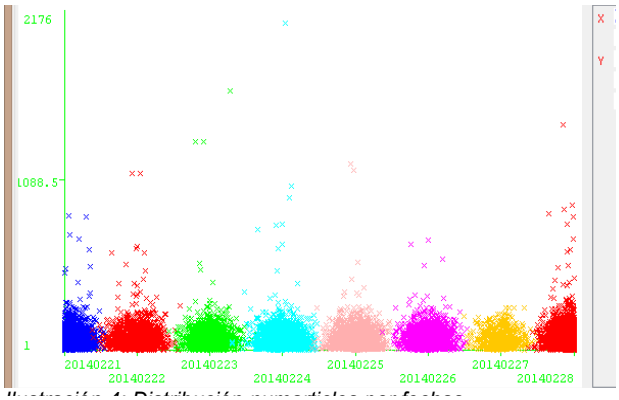


Imagen de la distribución del valor del atributo numarticles
a lo largo de los días 21/02/2014 hasta 28/02/2014 en Ucrania

AvgTone. Un campo que también es interesante es el tono que se asigna a este evento, GDELT proporciona una valoración del evento desde extremadamente negativa (-100) a extremadamente positiva (+100). se utiliza para filtrar el contexto de los eventos como una medida de la importancia de un evento. Así una protesta con una valoración positiva puede sugerir una incidencia menor que la que describe el contexto. Empíricamente he encontrado que el atributo *avgtone* sólo toma valor positivos en todos los casos, con lo que tendré en cuenta esta corrección en posteriores tratamientos.

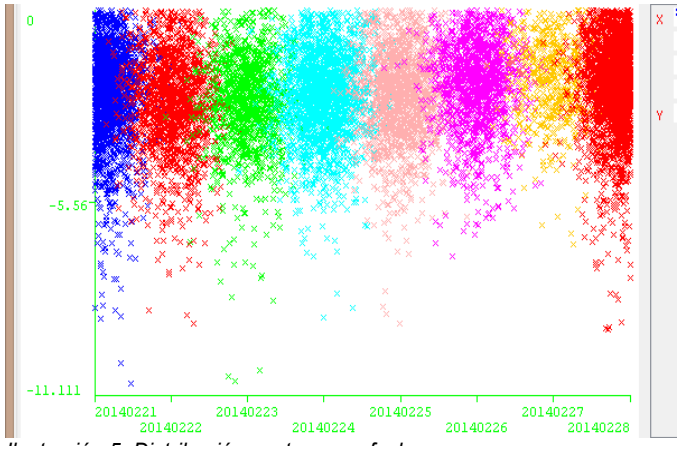


Imagen de la distribución del valor del atributo avgtone
a lo largo de los días 21/02/2014 hasta 28/02/2014 en Ucrania

Normalización de los datos:

Los atributos g*oldsteinscale* y *avgtone* tienen los valores continuos asociados con eventos considerados como positivos cuando alcanzan el valor mínimo, mientras que los atributos *numarticles* tiene valores continuos asociados con eventos positivos cuando alcanza valores máximos, así que transformaré los valores de los atributos *goldsteinscale* y *avgtone* a su valor inverso, para favorecer el tratamiento de datos.

En este caso no me interesa normalizar los datos puesto que aunque genero muchos atributos al final genero pocas instancias con mucho valor informativo.

Enriquecer la capacidad predictiva:

Con lo que explicado anteriormente, al tener tal cantidad de datos, muchas muestras no aportan información y sí que producen ruido en el conjunto de datos, de hecho cuando se reducen eventos que no generan gran cantidad de artículos y cuyo valor *goldsteinscale* y *avgtone* no son lo suficientemente grandes afectan negativamente a la capacidad predictiva.

Aparte de los anteriores atributos voy a recurrir al estudio realizado por N. Kallus[32] para incrementar la capacidad predictiva del conjunto de datos.

- *Escoger el nivel base de conflictos significativos:* Me centraré en la siguiente pregunta ¿Cuando los eventos serán suficientemente significativos?

Para poder cuantificar la significancia de un evento voy a utilizar la medida propuesta por N. Kallus[32] , Mcs(i, j) que será o sea número de eventos de los códigos indicados como señales de protestas y conflictos durante el día j en el país c respecto al evento conflicto que se produjo el día i.

Según N. Kallus, los eventos serán significativos si el número de fuentes y el número de artículos relacionados con este evento, así como el número de citaciones que se hace a este evento en las fuentes citadas son lo suficientemente altos.

Sin embargo, al trabajar con datos parcialmente procesados obtenidos de GDELT se observa que no sólo la cantidad de artículos por día son necesarios, tanto es así que tres atributos en concreto proporcionan más información que la cantidad de artículos por día. Esto atributos son *goldsteinscale*, *avgtone* e *isrootevent*, a parte por supuesto de la base para clasificar eventos que es *rooteventcode*, el código CAMEO que identifica el tipo de evento que se ha producido, clasificándolo según si es protesta o enfrentamientos, o no.

Me interesa, buscar aquellos conflictos que son lo suficientemente significativos, que estimaré que son aquellos que tienen un nivel de cobertura mayor que el habitual para ese tipo de eventos, los que tienen un tono medio bajo, cuya valor en la escala goldstein sea lo suficientemente negativo y que sean eventos que afecten al país, dato que obtengo de *isrootevent*.

O sea, y siguiendo la estimación que hace N. Kallus[32] ,habrá un conflicto significativo en el país *c* en el día i si el valor M_c (i; i) es mayor que el habitual.

Puesto que la cantidad de eventos se va incrementando a lo largo del tiempo este valor se ha de normalizar para que tenga el mismo peso independientemente del periodo en el que se produzca, para ello normalizaré, al igual que N. Kallus[32] , la media según la cantidad de menciones que se hacen sobre el evento a la media de los 60 días anteriores en todos los países, o sea para saber el número de menciones de protesta en el país *c* que tienen lugar en el día *i+k*, siendo *k* la cantidad de días anteriores al evento evaluado, que quedará:

$$M'_{cs}(i, i+k) = (M_{cs}(i, i+k))/((1/60)\Sigma_{j=i-60}^{i-1} M_{c's}(j,j+k))$$

He escogido tan solo 60 días debido a la gran cantidad de información que proporciona le fuente GDELT y a la falta de hardware que soporte más cantidad de procesamiento un tiempo razonable.

Ahora se definiré la media del conjunto de entrenamiento los eventos del mismo día de la fuente principal tomando el total de días a evaluar.

$$\mu'_c = (1/|dias|)\Sigma_{i \in dias} M'_{fuenteprincipal}(i,i)$$

Para suavizar los datos consideraré una media móvil de tres días, entonces por la definición se dirá que una protesta significativa en el país *c* se produce cuando el *rooteventcode* es 14,18,19 o 20 y además la media de *avgtone* y *goldsteinscale* es mayor de lo habitual. El límite se escoge para seleccionar sólo las protestas significativas.

Para cuantificar las señales predictivas que sirvan como base para hacer predicciones, utilizaré la escala Goldstein que proporciona la instancia para el día *i* en el país *c*, que denotaré por V_c.*y t*ambién el atributo *avgtone,* al que llamaré A_c.

Normalizaré las menciones a eventos para que no se vea afectado por el incremento de cantidad de eventos a través del tiempo utilizando los valores alcanzados los 60 días anteriores mediante V_c:

$$V'_c(i) = (V_c(i,))/((1/60)\Sigma_{j=i-60}^{i-1} V_c(j))$$

También utilizaré la característica $M_{cs}(i, i+k)$ con un $k \geq 1$, para facilitar el entrenamiento normalizaré esta característica con respecto a la que

queremos predecir, $M_{cfuenteprincipal(i,i)}$, y para ello utilizaré la media μ'_c por país como coeficiente constante. O sea será la característica:

$M_{cs}(i, i+k)/\mu'_c$

Por cada muestra incluiré también como características los diez días anteriores de reporte de protestas para el día evaluado y, también, normalizados con la media de los nueve días anteriores, hecho que utilizo para suavizar de forma más precisa los datos:

$M'_{cs}(i, i)/\mu'_c$
$M'_{cs}(i, i-1)/\mu'_c \ldots M'_{cs}(i, i-9)/\mu'_c$

También incluiré como características las escalas Goldstein normalizadas :

$V'_c(i) \ldots V'_c(i-9)$

Utilizaré también como característica el "tono medio" (*avgtone*) del evento asignado por GDELT para cuantificar la incidencia del evento, esto es debido a que el atributo *goldsetinscale* es significativo, pero cuanto menor es el valor *avgtone* más importancia se presenta en el evento evaluado, con esto no sería necesario calcular la incidencia de este evento en días posteriores puesto que ya estaría cuantificado. En este caso el atributo se corresponderá con el indicado como *avgtone,* al que denotaré por $A_c(i)$ y lo normalizaré de forma similar a $V'_c(i)$ y lo denotaré como $A'_c(i)$ donde:

$$A'_c(i) = (A_c(i,))/((1/60)\Sigma_{j=i-60}^{i-1} A_c(j))$$

corresponde con el valor de *avgtone* normalizado a los sesenta días anteriores para este atributo.

$A'_c(i) \ldots A'_c(i-9)$

La generación de los atributos A'_c, M'_c y V'_c la realizaré agrupando los eventos por día utilizando el filtro de Weka weka.filters.unsupervised.instance.Denormalize que es un método de tratamiento de datos no supervisado que agrupa instancias según un atributo, en este caso *sqldate*, y que permite normalizar los datos escogiendo un método de agregación que puede ser la suma de los atributos, la media de los atributos (*average*), el valor máximo o el valor mínimo.

Puesto que en todos los casos estoy calculando la suma de los valores de los atributos y hago el cociente con el número total de elementos y el método de agregación *average* hace exactamente lo mismo aprovecho esta capacidad para generar los valores de los atributos a añadir.

La codificación para el cálculo de los atributos de la instancia se referencia en el anexo V.

4.1.3.3. Modelo de predicción:

Me interesa predecir, en cada día *i,* cuando ocurrirá una protesta significativa ocurrirá basado en los eventos que se han producido antes del día *i,* o sea que *rooteventcode* sea 14, 18,19 o 20*,* en cuyo caso su denotaré su clase como 1 y si no es así denotaré su clase como 0.

Siguiendo a N. Kallus cuantificaré el éxito de un mecanismo de predicción basado en su precisión balanceada. Supongo que la protesta es significativa en el país *c* para el día *i*, valor que denoto por T_{ci}. Estos valores serán positivos si la clase es 1.

Cuantificaré el éxito de un mecanismo de predicción basándome en su precisión balanceada.

El ratio de aciertos positivos (*RAP*) es la fracción de instancias positivas ($T_c = 1$) que han sido correctamente predichas, o sea *TP/(TP+FN)*, siendo TP los aciertos positivos y FN los falsos negativos y el ratio de aciertos negativos (*RAN*) son aquellas instancias negativas cuya predicción ha sido correcta, o sea *TN/(TN+FP)*, siendo *TN* los aciertos negativos y *FP* los falsos positivos. La precisión balanceada [38] será la media de los aciertos positivos y los aciertos negativos sin peso, o sea (*RAP*+*RAN*)/2. Uso este método porque, al contrario que la precisión marginal, esta no puede ser elevada artificialmente.

Con lo que a pesar de que no haya datos, la precisión balanceada para un acierto negativo será del 50% mientras que con la precisión marginal se obtendrá un valor de 94%. De hecho cualquier predicción sin datos relevantes siempre produce una precisión balanceada del 50% de media por la independencia estadística.

4.1.3.4. Generación de la predicción

Para generar el modelo predictivo utilizaré el algoritmo Random Forest entrenado en los datos desde septiembre del 2013 a marzo de 2014 para el país Ucrania , Turquía, Rusia, España, Siria, Túnez y México (he escogido esta periodo y estos países puesto que son muy significativos a la hora de producirse conflictos sociales, ya que coincide con el conocido como *Euromaidan*, una serie de protestas y conflictos que se produjeron en Ucrania y que condujeron a conflictos sociales y al conflicto armado con parte del país que se sentía más próximo a Rusia y las protestas en Turquía relacionadas con la transformación de la plaza *Taksim* propuesta por el gobierno) combinados con los datos para Ucrania para el periodo 01/09/2013 hasta 01/04/2014, que corresponde con el periodo de

protestas significativas relacionadas con la crisis conocida como *Euromaidan*.

Como atributo clasificador uso el atributo *class.* Como número de características sobre las que se aplicaremos el modelo clasificador será la raíz cuadrada del total de atributos.

La elección del *RandomForest* como algoritmo de regresión para generar el modelo predictivo se deriva principalmente de: [37]

- Los árboles que genera no necesitan poda,.
- Se genera automáticamente la precisión y la importancia de variables.
- No es un problema el sobre ajuste sobre los datos de entrenamiento y no es excesivamente sensible a los valores atípicos en los datos de entrenamiento.
- Facilidad de configuración de sus parámetros, donde sólo se indica la cantidad de árboles a generar y el número de características sobre las que agrupar.

Configuración de Weka para trabajar con Hadoop Hive:

En el cluster de Hadoop que he configurado para el tratamiento de datos he creado una base de datos en Hadoop Hive, la he llamado *TFG2015* que almacenará una tabla, a la que he llamado *gedeltdaily* con el conjunto de datos que voy obteniendo desde el repositorio GDELT.

Lo ideal sería crear una tarea mapreduce para generar el conjunto de entrenamiento automáticamente desde Hadoop Hive que seleccionara el conjunto de datos necesario para enriquecer dicho conjunto. Sin embargo, y debido a las restricciones en hardware que tengo, sólo hago la consulta en Hadoop Hive y utilizo weka y sus clases para tratar los datos y entrenar el modelo con ellos.

Utilizaré el driver genérico para el acceso a Hadoop Hive jdbcDriver=com.cloudera.hive.jdbc3.HS2Driver para acceder desde el entorno weka a la tabla que he creado en mi cluster hadoop. Para ello he utilizado el fichero de propiedades que almacena weka en su estructura de directorios denominado DatabaseUtils.props, donde se especifica el tipo de driver que se va a utilizar, el tipo de autentificación para acceder a la base de datos, y las características de los tipos básicos.

Utilizando las clases de weka weka.core.Instances y weka.experiment.InstanceQuery y el fichero de propiedades anterior no necesito especificar el método de acceso en el código y tan sólo tengo que hacer la llamada a la base de datos mediante el código:

```
import weka.core.Instances;
import weka.experiment.InstanceQuery;

InstanceQuery query = new InstanceQuery();
Instances istraining;


expre = "Select sqldate,eventrootcode, numarticles, goldsteinscale,
avgtone from GDELTDAILY where (actor1countrycode = "+ pais +")";
  expre = expre + " and (isrootevent = 1) and (sqldate > " + ffin + ")
and (sqldate < " + finic + ") order by sqldate";

  query = new InstanceQuery();
query.setUsername("");
query.setPassword("");
query.setQuery(expre);

try{
  ins = query.retrieveInstances();
}
```

Texto 1: Detalle de conexión a Base de datos Hadoop

Puesto que la cantidad de eventos por día/país es muy grande, y en aras de optimizar el tiempo de ejecución, he aplicado técnicas de minería de datos para reducir el conjunto de datos, y para ello me apoyo tanto en la información aportada por GDELT como en los resultados empíricos y el sentido común, a saber:

Una vez hecho eso trato los datos para generar las instancias de entrenamiento en la forma que he especificado en el apartado 4.1.3.2. y lo hago con el código que especifico en el anexo V mediante el método de clase *InstanciaDiasAntetriores(Instances , Instances , Instances , **int** , **int**)*.

Creo un método análogo para para crear la instancia de test a partir de los datos proporcionados por el usuario al que llamaré *prueba(Instances, **int**, **int**).*

Generación de resultados:

Una vez codificados los métodos para generar el conjunto de datos de entrenamiento y la instancia del día a evaluar he codificado la ejecución del proceso de entrenamiento.

Esto lo he hecho mediante el algoritmo weka.classfiers.RandomForest, con 500 árboles con el número de características a agrupar siendo la raíz del total de características del conjunto de datos de entrenamiento con validación cruzada para 4 folders.

```
      RandomForest rf;
        String[] options;
        StratifiedRemoveFolds filter;
        Instances test;
        Instances train;
        Classifier cModel;
        Evaluation eTest;

      // usar StratifiedRemoveFolds para separar aleatoriamente los
        filter = new StratifiedRemoveFolds();

        // Creo el conjunto de opciones
        options = new String[6];

        options[0] = "-N";                // Se indica que se requiere un número
de folders
        options[1] = Integer.toString(4);  // separa los datos en 4 folders
        options[2] = "-F";                // Se especifica un único folder para
reducir el tiempo de tto.
        options[3] = Integer.toString(1);  // Se seleccion el primer folder
        options[4] = "-S";                // Se indica que se vá a usar una
semilla aleatoria
        options[5] = Integer.toString(1);  // La semilla aleatoria va a ser 1

        filter.setOptions(options);
        filter.setInputFormat(fin);
        filter.setInvertSelection(false);

        // se aplica el filtro para el conjunto de test
        test = Filter.useFilter(fin, filter);

        //  Se cambia la selección para tener la inversa, o sea los tres folders
restantes
        filter.setInvertSelection(true);
        train = Filter.useFilter(fin, filter); //Se genera el conjunto de datos de
entrenamiento

        //Se aplica el entrenamiento
        rf = new RandomForest();
                rf.setNumExecutionSlots(2);
                rf.setNumTrees(500);
                rf.setPrintTrees(true);
                rf.setNumFeatures(10);

        cModel = (Classifier) rf;


        //cModel.buildClassifier(istraining);
        cModel.buildClassifier(train);

        eTest = new Evaluation(train);
        Random rand = new Random(1);
        eTest.crossValidateModel(cModel, fin, 3, rand);
```

Texto 2: Detalle de entrenamiento con Random Forest en Java

En esta implementación la clase java que realiza el trabajo produce una salida que será la probabilidad de que se produzca un conflicto para el día y país indicados, asumo que el día previo se calcula desde el instante en que se lanza la ejecución.

Aparte, como salida adicional, se produce un fichero con los datos estadísticos asociados al conjunto de entrenamiento y a la instancia a evaluar, así como el conjunto de datos que ha producido el resultado anterior.

Y procedo a generar datos con el código definido en el Anexo V en concreto el método que he denominado *testear*:

En concreto realizaré el entrenamiento con datos de los países comentados anteriormente para el periodo desde 01/01/2012 hasta el 31/05/2014

El conjunto de atributos quedará como explico en el Anexo VI.

Voy a detallar un poco la los resultados obtenidos en las pruebas realizadas con los conjuntos de datos obtenidos tras el entrenamiento.

Desde el conjunto de datos obtenido al ejecutar el código procedo a aplicar manualmente el entrenamiento para obtener el detalle de los datos generados para el algoritmo clasificatorio Random Forest que evalue 500 arboles con 10 características, use 1 como semilla aleatoria y lo evalúe mediante validación cruzada con 4 folders evaluando cada uno de ellos contra los tres restantes. El resultado se observa en la figura siguiente.

```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -I 500 -K 10 -S 1 -num-slots 2

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         354               84.8921 %
Incorrectly Classified Instances        63               15.1079 %
Kappa statistic                          0.5201
Mean absolute error                      0.1973
Root mean squared error                  0.3128
Relative absolute error                 59.5545 %
Root relative squared error             76.976  %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      78.0576 %
Total Number of Instances              417

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area
PRC Area  Class
               0,921    0,425    0,891      0,921    0,906      0,522    0,907     0,975     0
               0,575    0,079    0,658      0,575    0,613      0,522    0,907     0,704     1
Weighted Avg.  0,849    0,353    0,843      0,849    0,845      0,522    0,907     0,919

=== Confusion Matrix ===

   a   b   <-- classified as
 304  26 |   a = 0
  37  50 |   b = 1
```

Según lo comentado anteriormente el resultado de la predicción balanceada BAC será de un **84,9**%, comparado con el caso de no tener datos lo que supone un **41.10**% de reducción en el error balanceado respecto de no tener datos.

Estudio de rendimiento:

En este punto voy a realizar un estudio para mejorar el rendimiento del ciclo de vida completo desde la selección de datos, tratamiento y enriquecimiento de datos y generación de resultados, así como incrementar la precisión del modelo.

Rendimiento de la infraestructura:

Uno de los puntos fuertes de los sistemas basados en Big Data es utilizar múltiples servidores de bajo coste y capacidad media para procesar solicitudes de gran cantidad de datos en menos tiempo al repartir la carga de trabajo entre todos ellos.

No profundizaré en esta cuestión debido a la limitación en medios e infraestructura en la que me encuentro.

Rendimiento del modelo de predicción:

Comienzo con el test sobre los datos sin tratar, solamente escogiendo los atributos proporcionados por GDELT, en particular los que he indicado en el apartado 4.1.3.2, el procesamiento de tan solo un país para un sólo día consume al menos 2 horas para poder generarse, lo que es prohibitivo para cualquier solución que persiga el objetivo final.

Tan solo incrementando el número de artículos por muestra a un mínimo de 40, se reduce el tiempo de procesamiento a una cuarta parte del planteamiento anterior, y la precisión balanceada se ve afectada en un porcentaje menor de un 10%.

Para cada instancia (*país*, *día*, *eventrootcode*) obtenida del repositorio GDELT crearé un nuevo conjunto de atributos para permitir la clasificación y reducir el procesamiento total del modelo clasificatorio.

Haré la comparativa de precisión del modelo sobre los datos en crudo y a continuación lo compararé con los datos pre-tratados tal y como he explicado en los apartados anteriores:

Tomaré el periodo 01/01/2012 hasta el 31/05/2014 (es el rango de fechas de las que dispongo hasta el momento), para Ucrania como conjunto de entrenamiento.

Tras la selección obtengo 200812 instancias de eventos

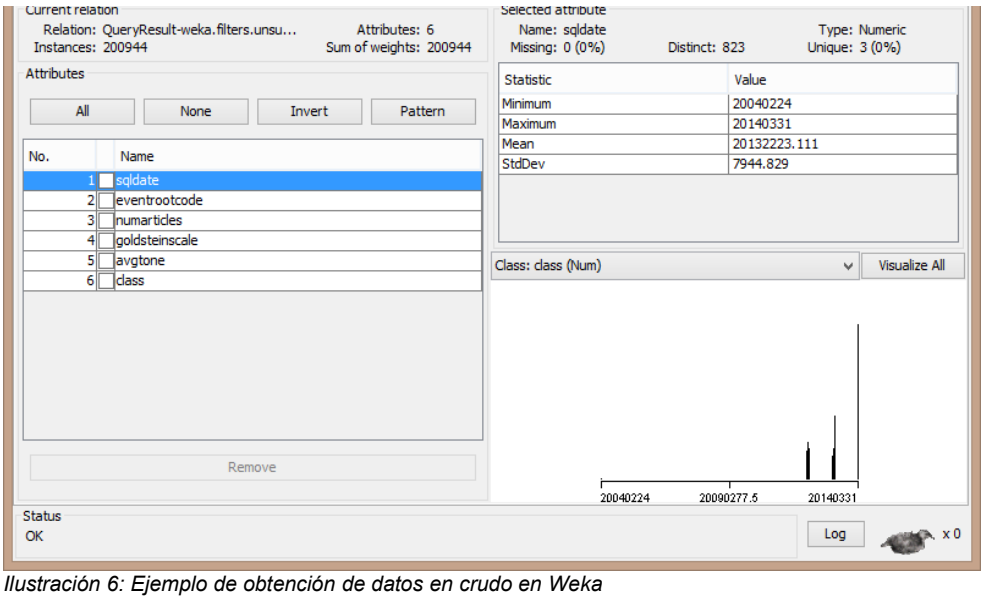


Ilustración 6: Ejemplo de obtención de datos en crudo en Weka

Imagen de los datos a pretratar en Weka para Ucrania en el rango de fechas 01/01/2012 hasta 31/05/2014

Si se toman los en crudo la capacidad predictiva se obtiene:

Time taken to build model: 113.67 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	200564	99.8765 %
Incorrectly Classified Instances	248	0.1235 %
Kappa statistic	0.9913	
Mean absolute error	0.0017	
Root mean squared error	0.0299	
Relative absolute error	1.2014 %	
Root relative squared error	11.2268 %	
Coverage of cases (0.95 level)	99.9841 %	
Mean rel. region size (0.95 level)	50.238 %	
Total Number of Instances	200812	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,999	0,007	0,999	0,999	0,999	0,991	1,000	1,000	0
	0,993	0,001	0,991	0,993	0,992	0,991	1,000	1,000	1
Weighted Avg.	0,999	0,006	0,999	0,999	0,999	0,991	1,000	1,000	

=== Confusion Matrix ===

a	b	<-- classified as
185239	141	a = 0
107	15325	b = 1

Texto 3: Detalle de clasificación con Random Forest en Weka 3.7

Si se reduce el conjunto de datos con eventos que afecten negativamente la situación del país, como es tomando solo los eventos cuya escala goldstein sea baja y el tono medio también sea bajo.

En concreto *goldsteinscale* > -2 y *avgtone* > -4 se reduce la cantidad de eventos y se reduce en consecuencia el tiempo de procesamiento de las solicitudes al tratar con un menor número de datos, en concreto **90696** muestras, una reducción de un **54**% en la cantidad de datos.

El resultado es el siguiente:

```
Time taken to build model: 36.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       90466               99.7464 %
Incorrectly Classified Instances       230                0.2536 %
Kappa statistic                          0.9902
Mean absolute error                      0.0035
Root mean squared error                  0.0428
Relative absolute error                  1.3745 %
Root relative squared error             11.9215 %
Coverage of cases (0.95 level)          99.9713 %
Mean rel. region size (0.95 level)      50.4813 %
Total Number of Instances            90696

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
               0,998    0,007    0,999      0,998    0,999      0,990    1,000     1,000    0
               0,993    0,002    0,991      0,993    0,992      0,990    1,000     0,999    1
Weighted Avg.  0,997    0,007    0,997      0,997    0,997      0,990    1,000     1,000

=== Confusion Matrix ===

    a     b   <-- classified as
 76803   128 |     a = 0
   102 13663 |     b = 1
```

Texto 4: Detalle de resultados de la clasificación en Weka

La precisión balanceada apenas se ve afectada, sin embargo el tiempo de tratamiento sigue siendo prohibitivo, sobre todo teniendo en cuenta que con el conjunto final utilizaré todos los datos de todos los países como conjunto de entrenamiento.

Si ahora aumento la cantidad mínima de artículos a 40 el conjunto de datos se reduce a tan sólo **4069** eventos y la precisión se ve afectada en poco:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4069               99.4866 %
Incorrectly Classified Instances        21                0.5134 %
Kappa statistic                          0.9799
Mean absolute error                      0.0072
Root mean squared error                  0.0568
Relative absolute error                  2.8349 %
Root relative squared error             15.921  %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      51.1002 %
Total Number of Instances             4090

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,997    0,015    0,997      0,997    0,997      0,980    1,000     1,000     0
               0,985    0,003    0,980      0,985    0,983      0,980    1,000     0,999     1
Weighted Avg.  0,995    0,013    0,995      0,995    0,995      0,980    1,000     1,000

=== Confusion Matrix ===

   a    b   <-- classified as
 3466   12 |    a = 0
   9  603 |    b = 1
```

Texto 5: Detalle de análisis clasificatorio en Weka 3.7

Aunque los resultados estadísticos son esperanzadores, la realidad es que la predicción que se obtiene al agrupar los datos por día no son nada precisos, de hecho se aproximan al caso de no tener datos.

Comparación de modelos de prediccion.

Voy a comparar distintos algoritmos de clasificación para comparar la fiabilidad de los resultados de predicción respecto al algoritmo que he escogido, el Random Forest y así evaluar si es mejorable por algún otro modelo de clasificación.

Es por ello que aplico la propuesta de N Kallus[32] para predecir con la generación de nuevas muestras según el sistema anteriormente descrito. En concreto son los resultados para predecir la probabilidad de que se produzca un evento positivo (es decir una protesta) el día 28/02/2014 en Ucrania con diez días de antelación.

El resultado que se obtiene es el siguiente:

=== Run information ===

Scheme: weka.classifiers.trees.RandomForest -I 500 -K 0 -S 1 -num-slots 2
Relation: Training dataset-weka.filters.unsupervised.attribute.NumericToNominal-R32
Instances: 477
Attributes: 32
avgt_media_fecha_previo_1
goldstein_media_fecha_previo_1
numarticles_media_fecha_previo_1
avgt_media_fecha_previo_2
goldstein_media_fecha_previo_2
numarticles_media_fecha_previo_2
avgt_media_fecha_previo_3
goldstein_media_fecha_previo_3
numarticles_media_fecha_previo_3
avgt_media_fecha_previo_4
goldstein_media_fecha_previo_4
numarticles_media_fecha_previo_4
avgt_media_fecha_previo_5
goldstein_media_fecha_previo_5
numarticles_media_fecha_previo_5
avgt_media_fecha_previo_6
goldstein_media_fecha_previo_6
numarticles_media_fecha_previo_6
avgt_media_fecha_previo_7
goldstein_media_fecha_previo_7
numarticles_media_fecha_previo_7
avgt_media_fecha_previo_8
goldstein_media_fecha_previo_8
numarticles_media_fecha_previo_8
avgt_media_fecha_previo_9
goldstein_media_fecha_previo_9
numarticles_media_fecha_previo_9
avgt_media_fecha_previo_10
goldstein_media_fecha_previo_10
numarticles_media_fecha_previo_10
rootcodeavg
class
Test mode: 4-fold cross-validation

=== Classifier model (full training set) ===

Random forest of 500 trees, each constructed while considering 5 random features.
Out of bag error: 0.0734

Time taken to build model: 0.64 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	441	92.4528 %
Incorrectly Classified Instances	36	7.5472 %
Kappa statistic	0.1636	
Mean absolute error	0.1109	
Root mean squared error	0.2276	
Relative absolute error	74.6574 %	
Root relative squared error	84.0392 %	
Coverage of cases (0.95 level)	99.3711 %	
Mean rel. region size (0.95 level)	67.9245 %	
Total Number of Instances	477	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,995	0,895	0,928	0,995	0,960	0,245	0,916	0,992	0
	0,105	0,005	0,667	0,105	0,182	0,245	0,916	0,486	1
Weighted Avg.	0,925	0,824	0,907	0,925	0,898	0,245	0,916	0,951	

=== Confusion Matrix ===

a	b	<-- classified as
437	2	a = 0
34	4	b = 1

En este caso la predicción balanceada de los resultados es muy buena, se tiene obtiene pues TNR=**92,5**% y el TPR= 10**,5**% de los que se obtiene una precisión balanceada BAC = **92,5**%, como ya he comentado en el apartado anterior.

Comparación con otros modelos de predicción:

Predicción a 10 días vista para el 28/02/2014 en Ucrania para los algoritmos:

Algoritmo ZeroR:

Sólo determina la clase más común, o la media si se clasifica mediante atributos numéricos,Testea como de bien se predice la clase sin considerar otros atributos.

Suele usarse como el peor caso en comparaciones de precisión.[44]

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Class
	1,000	1,000	0,920	0
	0,000	0,000	0,000	1
Weighted Avg.	0,920	0,920	0,847	

=== Confusion Matrix ===

a	b	<-- classified as
439	0	a = 0
38	0	b = 1

Texto 7: Resultados de precisión con algoritmo de clasificación ZeroR (sin prediccion)

Algoritmo Random forest:

La explicación a este algoritmo se ha explicado en apartados anteriores.

weka.classifiers.trees.RandomForest -- -I 100 -K 0 -S 1 -num-slots 1
=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Class
	0,995	0,895	0,928	0
	0,105	0,005	0,667	1
Weighted Avg.	0,925	0,824	0,907	

=== Confusion Matrix ===

a	b	<-- classified as
432	7	a = 0
30	8	b = 1

Algoritmo Naive Bayes:

Algortimo de clasificación basado en las reglas de Bayes, que considera que los atributos X_1 ...X_n are sin independientes condicionalmente de otro atributo dado Y[45].

```
Scheme:       weka.classifiers.bayes.NaiveBayes

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision   Class
                 0,888    0,184    0,982       0
                 0,816    0,112    0,388       1
Weighted Avg.    0,883    0,178    0,935

=== Confusion Matrix ===

  a   b   <-- classified as
390  49 |   a = 0
  7  31 |   b = 1
```

Texto 9: Resultados de precisión con algoritmo de clasificación Naive bayes

Algoritmo Bagging:

Es meta-algoritmo ideado para aprendizaje computacional diseñado para mejorar la estabilidad y la precisión de los algoritmos de aprendizaje automático utilizados en clasificación estadística y la regresión. También reduce la varianza y ayuda a evitar sobreajuste. A pesar de que se aplica generalmente a los métodos de árboles de decisión, que puede ser utilizado con cualquier tipo de método. Bagging es un caso especial del modelo de aproximación media.[46]

```
Scheme:       weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision   Class
                 0,904    0,184    0,983       0
                 0,816    0,096    0,425       1
Weighted Avg.    0,897    0,177    0,938

=== Confusion Matrix ===

  a   b   <-- classified as
338  10 |   a = 0
 64   5 |   b = 1
```

Texto 10: Resultados de precisión con algoritmo de clasificación Bagging

Algoritmo J48, una implementación del algoritmo C4.5:

C4.5 es una extensión del anterior algoritmo ID3 de Quinlan. Los árboles de decisión generados por C4.5 se pueden utilizar para la clasificación, y por esta razón, C4.5 se refiere a menudo como un clasificador estadístico.[47]

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision   Class
                   0,943    0,553    0,952     0
                   0,447    0,057    0,405           1
Weighted Avg.      0,904    0,513    0,908

=== Confusion Matrix ===

   a   b   <-- classified as
 338  10 |   a = 0
  64   5 |   b = 1
```

Como se puede observar de los anteriores datos la precisión balanceada que se obtiene del conjunto de datos pre-tratados para una fecha en concreta ejecutado con distintos algoritmos de clasificación, en concreto ZeroR, RandomForest, Bagging, Naive Bayes y el árbol J48 el que obtiene mejores resultados es el RandomForest con un BAC = **92,5 %** lo que mejora la predicción del resto de los algoritmos utilizados***.***

Si bien es cierto que el BAC es el mejor, en ningún caso el cuadrado del error relativo mejora el porcentaje del 84%, de hecho sólo se ha alcanzado este valor con Random Forest, el resto son bastante peores.

En versiones posteriores del desarrollo se debe revisar el cálculo de atributos enriquecidos para mejorar el modelo manteniendo la precisión pero reduciendo el porcentaje de error de clasificación.

- *Resultados obtenidos para fechas concretas:*

Para evaluar la precisión del modelo predictivo voy a escoger el día 28/02/2014 sobre Ucrania.

El detalle de los resultados se observa gráficamente en las siguientes figuras, por un lado hago un test de funcionamiento para predecir.

Con el modelo que he escogido se detecta una variación muy alta en la capacidad predictiva a lo largo del periodo de predictibilidad escogido.

Con el código desarrollado he generado los datos para los gráficos siguientes, información que proporciono en el Anexo VII y siguientes.

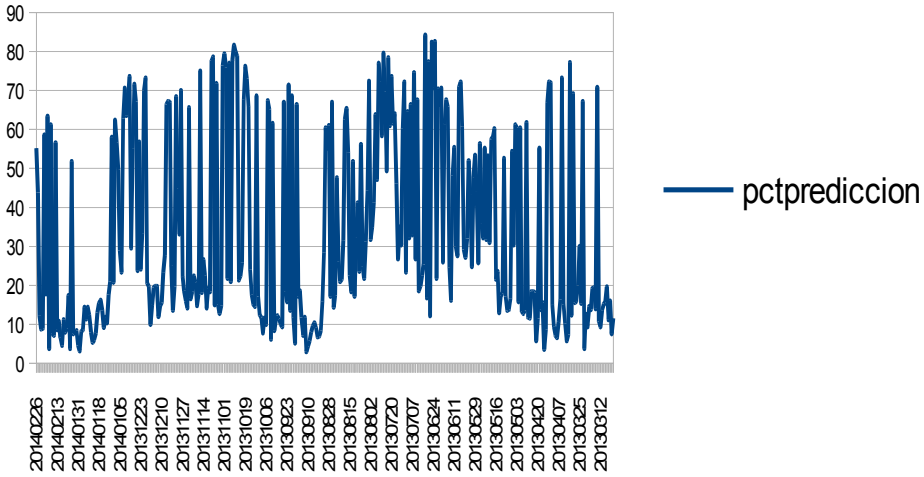


Ilustración 7: Porcentaje de probabilidad de disturbios en Ucrania para las

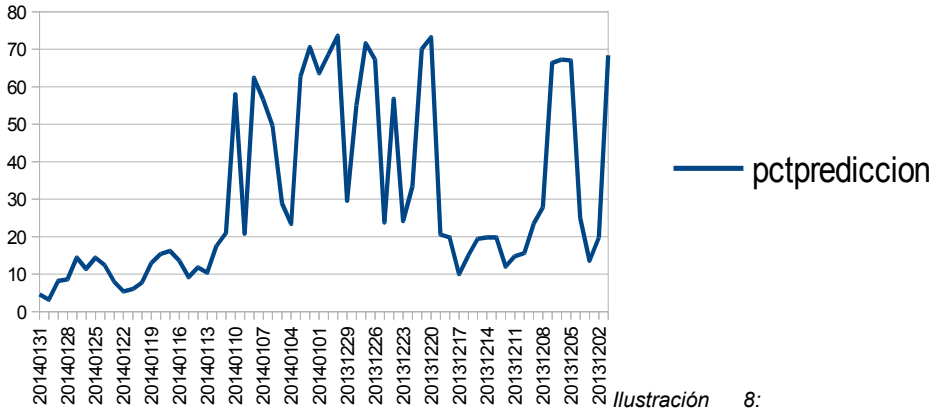


Ilustración 8: Detalle del porcentaje de probabilidad de que se produzca un conflicto social en Ucrania en durante el periodo 01/12/2013 hasta el 31/01/2014.Se observa que durante el periodo de Enero la probabilidad de que se produzcan protestas es muy alta, coincide pues con el periodo real de protestasen Ucrania durante esas fechas.

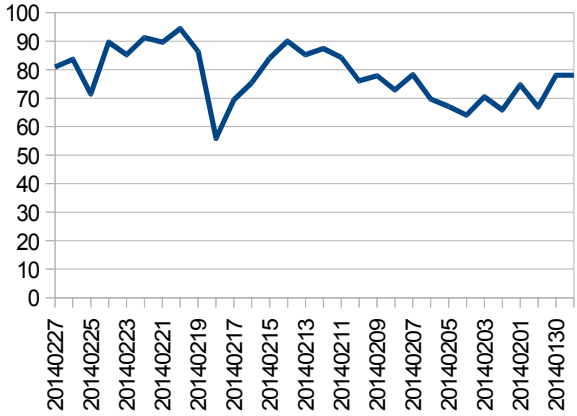


Ilustración 9: Porcentaje de probabilidad de que se produzca un conflicto social en Ucrania en el día 28/02/2014 calculado en base a los días previos, siendo estos desde un día hasta 31 días. Se observa que la precisiónse ve alterada según se aumenta el periodo previo.

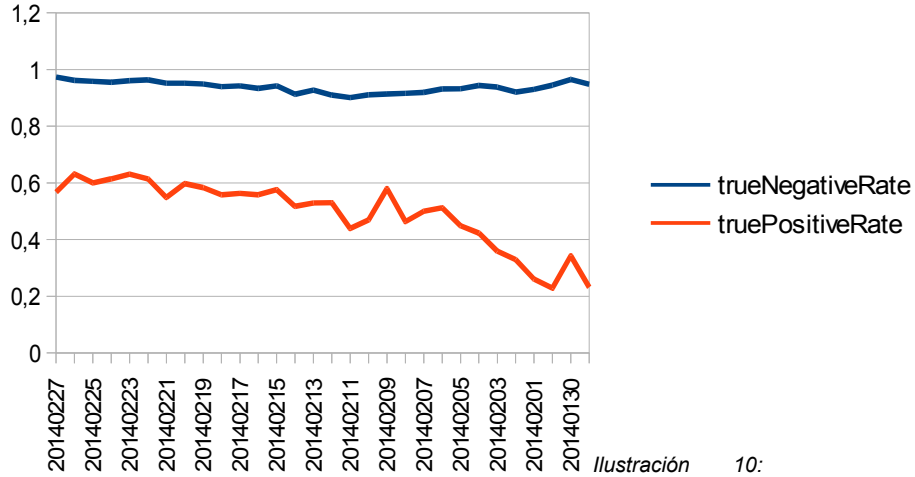


Ilustración 10: Ratio de aciertos negativos y aciertos positivos de la probabilidad de que se produzca un conflicto social en Ucrania en el día 28/02/2014 estimado con los datos de los 30 días anteriores

Como se puede observar en el gráfico anterior cuanto mayor es el desplazamiento en el tiempo requerido para obtener la predicción se introduce más ruido y los resultados se vuelven menos precisos.

5. Implementación

5.1 Resumen de la implementación:

Estos resultados se han plasmado en un aplicativo java que utiliza Weka y accede a un repositorio de datos Hadoop en la versión que ha proporcionado Hortonworks Sandbox, en concreto el repositorio se ha concretado en una base de datos que he llamado TFG2015 y forma una tabla que he llamado gdeltdaily.

5.2 Detalle de la implementación:

5.2.1 Justificación del diseño del aplicativo:

He diseñado una clase java con la posibilidad de ser llamado desde línea de comandos para facilitar su uso en ulteriores aplicaciones que hagan uso de él, ya sea mediante una interfaz web o mediante otros aplicativos que lo referencien.

5.2.2 Infraestructura de red:

He diseñado la infraestructura de red en una red local con acceso a internet, puesto que descargo un fichero al día desde el repositorio GDELT, ya que estos sólo proporcionan un fichero por día emitido a las 06:30 de la mañana, sólo se necesita un ancho de banda de conexión a internet de poca confianza,
La red local debe poder soportar las peticiones que se hagan desde el aplicativo Java.

5.2.3 Seguridad de la información y de la red:

Puesto que los datos de acceso son públicos, los servidores Hadoop están diseñados para trabajar en entornos de baja disponibilidad, ya que si cae algun servidor no afecta al sistema completo[39], no es necesario diseñar un sistema de seguridad perimetral sobre los servidores.

En caso de que en el entorno donde se sitúe el sistema de predicción exista un sistema de cortafuegos se ha de permitir la comunicación entre los servidores con la base de datos y el aplicativo a través del puerto TCP 10000 y se debe permitir el acceso al servidor Hadoop que obtiene los ficheros de eventos a través del puerto TCP 80.

5.2.4 Repositorio HDFS

Partiendo de la base de una imagen de máquina virtual proporcionada por Hortonworks Sandbox, en concreto la versión *HDP 2.2 : Ready for the enterprise,*disponible en la web de descargas de Hortonworks, he creado una nueba base de datos en Hive que contiene una tabla, a la que he llamado gdeltdaily, con los todos los campos que proporcionan los ficheros de descarga de GDELT, en concreto desde la ubicación "http://data.gdeltproject.org/events/" que es donde publica los ficheros con los eventos que utiliza el desarrollo.

En este servidor he programado una taréa que descarga los ficheros para el día actual, los descomprime y agrega los datos al sistema de ficheros HDFS y los inserta en la tabla Hive que he llamado gdeltdaily, tarea detallada en el anexo IX.

5.2.5 Instalación Weka y Java:

El aplicativo ha sido diseñado en Java y utiliza Weka como proveedor de clases, con lo que es necesario tener instalado Java 7[40] en el equipo que ejecute la aplicación.

Puesto que la inteligencia de la predicción la he basado en el uso de Weka y Java es necesario haber instalado la versión 3.7 de Weka [41] y se tienen que tener descargados los plugins de weka siguientes:

- denormalize Preprocessing
- distributedWekaBase Distributed
- distributedWekaHadoop Distributed

Estos no vienen instalados por defecto en la configuración básica.

Para la distribución de Hadoop proporcionada por Hortonworks Sandbox se ha de utilizar el driver de conexión JDBC Proporcionado por Cloudera[43], ya que el que proporciona Weka no es válido para la versión de Hortoworks Sandbox que he utilizado.

He configurado Weka 3.7 para poder utilizar el driver de Cloudera y para ello se ha de modificar el fichero DatabaseUtils.props para que pueda acceder a el. Se detalla la configuración en el Manual de usuario.

5.2.6 Entorno del aplicativo.

El aplicativo está basado en una clase Java llamada TFGPrediccion.

Esta clase permite guardar los datos intermedios o no mediante un atributo llamado *guardar*.

En caso de que se quieran guardar los ficheros intermedios, en el sistema de ficheros donde se ubique el aplicativo se generará una carpeta denominada d*ata* donde se guardarán los siguientes ficheros:

- tratado.arff: fichero con el conjunto de datos de entrenamiento que utilizará el aplicativo.

El fichero anterior contendrá los valores para las instancias y los campos descritos en el anexo VI.

- Out.txt: fichero con las estadísticas asociadas al resultado predictivo para el día y el país a evaluar.

Este último fichero contendrá los siguientes campos:

  - *pctprediccion*: porcentaje de probabilidad de que se produzca conflicto.
  - *truePositiveRate*: Es la proporción de muestras que fueron clasificados como clase x, entre todas las muestras que realmente tienen esa clase$_{[42]}$. Se obtiene de los datos que contiene el fichero tratado.arff, tras aplicar el entrenamiento para el algortimo clasificatorio RandomForest de Weka con los parámetros que he indicado en el apartado 4.1.3.4.
  - *trueNegativeRate*: Es la proporción de muestras que fueron clasificados como clase opuesta a x, entre todas las muestras que realmente tienen esa clase$_{[42]}$. Se obtiene de los datos que contiene el fichero tratado.arff tras aplicar el entrenamiento para el algortimo clasificatorio RandomForest de Weka con los parámetros que he indicado en el apartado 4.1.3.4.
  - *relativeAbsoluteError*: Error relativo absoluto de la clasificación
  - *rootRelativeSquaredError*: Cuadrado del error relativo raiz de la clasificación.
  - *fecha*: fecha del la instancia de eventos evaluada.
- Fichero con los datos obtenidos de la consulta a la base de datos Hadoop y que por defecto es *crudo.arff*.

6. Trabajos relacionados

Existen dos tipos de trabajos relacionados con el actual, aquellos que están desarrollados para la detección de eventos codificados a través de descripciones estructuradas de texto, por ejemplo informes de noticias, dentro de este grupo se encuentran los sistemas ICEWS[34] y GDELT[5], los cuales utilizan la taxonomía de codificación CAMEO.

El segundo grupo de sistemas lo representa el trabajo realizado por N. Kallus[32] donde propone un método para poder predecir disturbios sociales en un periodo corto de antelación y el sistema EMBERS[33] para la predicción de conflictos sociales, inicialmente restringido al área de latinoamérica, que utilizando fuentes de información variadas (twitter, facebook, etc...) realiza análisis del contenido de la información para predecir futuros conflictos sociales.

7. Conclusiones

He mostrado la posibilidad de mejorar la capadidad predictiva utilizando datos públicos masivos online gracias a las facilidades que posibilitan tanto el que agencias de información proporcionen datos relativos a artículos como el trabajo que ofrecen para al tratamiento de los mismos por organizaciones, como GEDLT, que hacen un trabajo previo de cualificación de la información que me ha sido muy util a la hora de reducir la complejidad del sistema.

Con el trabajo que he realizado se ofrece un método práctico y útil, tanto para organizaciones como para usuarios individuales, a la hora de poder estimar la posibilidad de que se produzcan protestas en paises, ofreciendo la posibilidad de conocer el riesgo que pueden sufrir aquellas personas que pretendan desplazarse a esos países, así como para poder evaluar los hechos futuros que se pueden deducir de la información aportada por los medios de comunicación online.

Hay mejoras obvias, como es la velocidad en la generación de resultados, pero esto sólo se puede solucionar con una infraestructura con la suficiente capacidad para gestionar toda la cantidad de datos en un tiempo razonable, de hecho los tests que he ido haciendo han consumido gran parte del tiempo dedicado a la realización del trabajo.

Otra mejora es reducir el porcentaje de error en la predicción, y para ello se debe revisar el proceso de enriquecimiento de atributos para generar el conjunto de datos de entrenamiento del modelo. Esta parte se deja para posteriores versiones

También se puede amplificar el grupo de fuentes de información para poder mejorar el modelo mediante el conjunto de datos utilizando datos desde proveedores públicos de información mediante la nube, como Amazon Web Services, Google Cloud o Gnip para obtener mas información aparte de la que proporcionan las agencias informativas oficiales.

8. Glosario

- **GDELT**: se ofrece como “la base de datos mayor, más comprensible y de mayor resolución de la sociedad humana alguna vez creada”. Es un proyecto 100% libre y abierto y en particular contiene los eventos que se encuentran en los medios de noticias mundiales y el formato de almacenamiento utiliza la taxonomía CAMEO. [5]
- **CAMEO**: (conflict and Mediation Event Observation) . Desarrollado por el “Computational Event Data Project” iniciado originalmente por la Universidad de Kansas y usado para la producción de conjunto de eventos generados por Reuters, Agence France Presse y el World Events Interaction Survey (WEIS) y para el desarrollo temprano de métodos de aviso ante cambios y conflictos políticos. [6]
- **Análisis predictivo**: El Análisis predictivo utiliza estadística junto con algoritmos de minería de datos. Se basan en el análisis de los datos actuales e históricos para hacer predicciones sobre futuros eventos. Dichas predicciones raramente suelen ser afirmaciones absolutas, pareciéndose más a eventos y su probabilidad de que suceda en el futuro. [17]
- **Operador Turístico**: Empresa que ofrece productos y servicios turísticos
- **Hadoop**: Es un framework que permite el procesamiento distribuido de grandes conjuntos de datos a través de clusters de computadores utilizando modelos de programación simple. Está diseñado para ser escalado desde un único servidor a miles de máquinas, cada una de ellas ofreciendo almacenamiento y computación local. [14]
- **Pentaho**: Plataform de Business Analytics, Integración y visualización de datos. [11]
- **Qlik**: Sofware de Business Intelligence y Visualización y Descubrimiento de datos.
- **Cassandra**: Es un proyecto soprtado por Apache para producir una base de datos altamente escalable y de gran disponibilidad sin comprometer el rendimiento. [12]
- **predictor,** una variable que puede ser medida para una entidad individual o de otro tipo para predecir el comportamiento futuro
- **APACHE**: O mejor conocido como Apache Tomcat, es un proyecto Opensource que proporciona una implementación de Java Servlet y paginas Java Server [13].
- **OPENSSL**: Conjunto de herramientas Opensource que implementan los protocolos SSL v2 y v3 y TLS así como una librería criptográfica de completa de propósito general. [16]
- **Datos semiestructurados**: Datos que no residen de bases de datos relacionales, y sin embargo poseen una organización interna que facilita su tratamiento, tales como documentos XML y

datos almacenados en bases de datos NoSQL.

- **Bases de datos NoSQL**: clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales (RDBMS) en aspectos importantes, el más destacado es que no usan SQL como el principal lenguaje de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no garantizan completamente atomicidad, consistencia, aislamiento y durabilidad, y habitualmente escalan bien horizontalmente.
- **Algoritmos de minería de datos supervisados**: Los algoritmos supervisados o predictivos predicen el valor de un atributo o etiqueta de un conjunto de datos, conocido el valor de otros atributos que lo clasifican. Buscan encontrar relaciones inducidas entre su etiqueta y otra serie de valores de atributos. Esas relaciones sirven para realizar la predicción en datos cuya etiqueta es desconocida. Es lo que se llama aprendizaje supervisado y se desarrolla en dos fases: Entrenamiento (se crea un modelo usando un subconjunto de datos clasificados con una etiqueta) y Test (comprueba que el modelo se verifica en el resto de los datos).
- **Algoritmos de minería de datos no supervisados**: O de descubrimiento del conocimiento que descubren patrones y tendencias en los datos actuales.
- **Hortonworks Sandbox:** Distribución integrada de Apache Hadoop.
- **Cloudera 5.3:** Distribución integrada de Apache Hadoop.
- **Weka**: Software OpenSource de análisis estadístico de conjuntos de datos desarrollado por la Universidad de Waikato, NZ.
- **Precisión balanceada:** Método estadístico utilizado como métrica para medir la precisión y similaridad predictiva de un conjunto de datos basada en hacer la media del ratio de aciertos positivos y el ratio de aciertos negativos [38].
- **Random Forest:** algoritmo de clasificación y predicción utilizado como algoritmo de minería de datos [37].
- **Oracle VM VirtualBox**: Paquete de software de virtualización X86 distribuido por Sun Microsystems.
- **Java**: Lenguaje de programación distribuido y soportado por Sun Microsystems.
- **Servidor Virtualizado**: Partición dentro de un servidor que habilita varias máquinas virtuales dentro de dicha máquina por medio de varias tecnologías.

9. Bibliografía

1.- Operador Turístico, http://es.wikipedia.org/wiki/Operador_tur%C3%ADstico; 10/03/2015
2.- http://cincodias.com/cincodias/2010/12/07/empresas/1291892334_850215.html, 10/03/2015
3.- http://www.20minutos.es/noticia/944583/0/turistas/dejan/egipto/, revisado 10/03/2015
4.- Manipulación mediática, http://es.wikipedia.org/wiki/Manipulaci%C3%B3n_medi%C3%A1tica_seg%C3%BAn_Noam_Chomsky, 10/03/2015
5.- GDELT, http://gdeltproject.org/data.html#intro, 10/03/2015
6.- CAMEO, http://eventdata.parusanalytics.com/data.dir/cameo.html, 10/03/015
7.- http://news.ugo.co.ug/kenya-lose-billions-travel-agents-cancel-flights/, 10/03/2015
8.-http://www.obs-edu.com/noticias/informe/el-volumen-de-datos-generado-por-smartphones-crecera-un-63-los-proximos-cuatro-anos/., 10/03/2015
9.- http://blog.markedup.com/2013/02/cassandra-hive-and-hadoop-how-we-picked-our-analytics-stack/, 10/03/2015
10.- Qlik, http://community.qlik.com/thread/105759, 10/03/2015
11.- Pentaho, http://community.pentaho.com/ 10/03/2015
12.- Cassandra, http://cassandra.apache.org/ 10/03/2015
13.- Tomcat, http://tomcat.apache.org/, 10/03/2015
14.- Hadoop, http://hadoop.apache.org/, 10/03/2015
15.- http://es.wikipedia.org/wiki/AJAX, 10/03/2015
16.- https://www.openssl.org/, 10/03/2015
17.- Analisis predictivo, http://fundacionbigdata.org/analisis-predictivo/, 10/03/2015
18.- [Weiss y Indurkhya, 1998], Weiss, S.M. y Indurkhya, N. "*Predictive Data Mining, A Practical Guide*", Morgan Kaufmann Publishers, San Francisco, 1998.
19.- [JE Yonamine, 2011], J.E. Yonamine, "*Working with event data: A guide to aggregation choices*", Department of Political Science , Pennsylvania State university, 2011.
20.- [J. Duran, J.Conesa, R. Clarisó, 2014], J. Duran, J.Conesa, R. Clarisó, "*Representación del conocimiento*", Material docente UOC, 2014.
21.- *[E. Mor, R. Sangüesa, L.C. Molina, 2015]* E. Mor, R. Sangüesa, L.C. Molina, "*Data Mining*", Material docente UOC, 2015
22.- [J. Garcia, X. Perramont, 2014], J. Garcia, X. Perramont, 2014 "*Seguridad en redes de computadores*", Material docente UOC, 2014.
23.- [J. Domingo, J Herrera, H. Rifá, 2014], J. Domingo, J Herrera, H. Rifá, "*Criptografía*", Material docente UOC, 2014
24.- [G. Coulouris et al., 2012], G. Coulouris, J. Dollimore, T. Kindberg, G. Blair, "*Distributed Systems, Concepts and Design*", 5ª ed., Addison-Wesley/Pearson, Essex, 2012
25.- [K. Sandoe et al., 2001], K. Sandoe, G. Corbitt, R. Boykin, "*Enterprise Integration*", Ed. Wiley, 2001.
26.- [J. R. Rodriguez, J. García, I. Lamarca, 2014], "*Gestion de proyectos informáticos: métodos, herramientas y casos*", 6ª ed., Ed. UOC, Barcelona, 2014.
27.- [J.C. Díaz, 2010], J.C Díaz, "*Introducción al Business Intelligence*" (Vol. 163), Editorial UOC, 2010.
28.- [J. L. Cano, 2007], J.L. Cano, "*Business Intelligence: Competir con información*", Banesto, Fundación Cultur, 2007.
29.- [J. Hurwitz et al., 2013], J. Hurwitz, A. Nugent, F. Halper, M. Kaufman, "*Big Data for Dummies*" , John Wiley & sons lmtd., Hoboken, 2013
30.- [P.Russom,], P. Russom, "*Big Data Analytics Q4*", TDWI Best Practices Report, TDWI, 2011.
31.- [J. Manyika et al., 2011], J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. H. Byers, "*Big data: The next frontier for innovation, competition, and productivity*", McKynsey Global Institute, 2011.

32.- [N. Kallus] N.Kallus, "Predicting Crowd Behavior with Big Public Data", Massachusets Institute of technology, 2014.
33.- [N. Ramakrishnam et al.], N. Ramakrishnan, P. Butler, S. Muthiah, Nathan Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, C. Kuhlman, A. Marathe, L. Zhao, T. Hua, F. Chen, C. Lu, B. Huang, A. Srinivasan, K. Trinh, L. Getoor,Graham Katz, A. Doyle, C. Ackermann, I. Zavorin, J. Ford, K. Summers, Y. Fayed, J. Arredondo,D. Gupta,D. Mares, "Beating the News' with EMBERS: Forecasting Civil Unrest using Open Source Indicators", *27 Feb 2014 (v1), last revised 28 Feb 2014, http://arxiv.org/abs/1402.7035*
34.- ICEWS, http://en.wikipedia.org/wiki/Integrated_Conflict_Early_Warning_System.
35.- Weka, http://www.cs.waikato.ac.nz/ml/weka/bigdata.html, revisado 03/2015.
36.- http://www.evaluandosoftware.com/nota-3059-Que-es-el-analisis-predictivo.html, *revisado 03/2015.*
37.- Random Forest, http://en.wikipedia.org/wiki/Random_forest, revisado 05/2015.
38.- Precisión y fiabilidad, *http://en.wikipedia.org/wiki/Accuracy_and_precision, revisado el 05/2015.*
39.- [T. White, 2012] ,T. White, Hadoop the definitive Guide 3ª ed., O'Reilly, 27/01/2012.
40.- **http://docs.oracle.com/javase/7/docs/webnotes/install/windows/jre-installation-windows.html**, revisado el 06/2014.
41.- http://www.gtbit.org/downloads/dwdmsem6/dwdmsem6lman.pdf, revisado el 06/2014.
42.- https://weka.wikispaces.com/Primer, revisado el 06/2014.
43.- http://www.cloudera.com/content/cloudera/en/downloads/connectors/hive/jdbc/hive-jdbc-v2-5-4.html, revisado 06/2014
44.- https://www.cs.auckland.ac.nz/courses/compsci367s1c/tutorials/IntroductionToWeka.pdf, revisado el 06/2015
45.- [T. Mitchel, 2015], Tom M. Mitchell. Machine Learning. McCraw Hill, All rights reserved. Copyright 2015, Revisión Jenero 31, 2015
46.- Bagging, https://en.wikipedia.org/wiki/Bootstrap_aggregating, revisado 3l 06/2015
47.- J48, https://translate.google.es/#en/es/C4.5%20is%20an%20extension%20of%20Quinlan's%20earlier%20ID3%20algorithm.%20The%20decision%20trees%20generated%20by%20C4.5%20can%20be%20used%20for%20classification%2C%20and%20for%20this%20reason%2C%20C4.5%20is%20often%20referred%20to%20as%20a%20statistical%20classifier, revisado 06/2015

10. Anexos

ANEXO I

Atributos
(THE GDELT EVENT DATABASE DATA FORMAT CODEBOOK V2.0 2/19/2015)
This codebook provides a quick overview of the fields in the GDELT data file format and their descriptions. GDELT event records are in the dyadic CAMEO format, capturing two actors and the **action performed by Actor1 upon Actor2.** A wide array of variables break out the raw CAMEO actor codes into their respective fields to make it easier to interact with the data, the Action codes are broken out into their hierarchy, the Goldstein ranking score is provided, an average tone score is provided for all coverage of the event, several indicators of importance are provided, and a special array of georeferencing fields offer estimated landmark-centroid-level geographic positioning of both actors and the location of the action

• **SqlDate**. Date the event took place in YYYYMMDD format.
• **GlobalEventID**. Globally unique identifier assigned to each event record that uniquely identifies it in the master dataset. NOTE: While these will often be sequential with date, this is NOT always the case and this field should NOT be used to sort events by date: the date fields should be used for this.
• **MonthYear**. Alternative formatting of the event date, in YYYYMM format.
• **Year**. Alternative formatting of the event date, in YYYY format.
• **FractionDate**. Alternative formatting of the event date, computed as YYYY.FFFF, where FFFF is the percentage of the year completed by that day. This collapses the month and day into a fractional range from 0 to 0.9999, capturing the 365 days of the year. The fractional component (FFFF) is computed as (MONTH * 30 + DAY) / 365. This is an approximation and does not correctly take into account the differing numbers of days in each month or leap years, but offers a simple single-number sorting mechanism for applications that wish to estimate the rough temporal distance between dates.
•**Actor1Code**. The complete raw CAMEO code for Actor1 (includes geographic, class, ethnic, religious, and type classes). May be blank if the system was unable to identify an Actor1.
• **Actor1Name**. The actual name of the Actor 1. In the case of a political leader or organization , this will be the leaders formal name (GEORGE W BUSH, UNITED NATIONS), for a geographic match it will be either the country or capital/major city name (UNITED STATES / PARIS), and for ethnic, religious, and type matches it will reflect the root match class (KURD, CATHOLIC, POLICE OFFICER, etc). May be blank if the system was unable to identify an Actor1.
• **Actor2Code**. The complete raw CAMEO code for Actor2 (includes geographic, class, ethnic, religious, and type classes). May be blank if the system was unable to identify an Actor2.
• **Actor2Name**. The actual name of the Actor 2. In the case of a political leader or organization , this will be the leaders formal name (GEORGE W BUSH, UNITED NATIONS), for a geographic match it will be either the country or

capital/major city name (UNITED STATES / PARIS), and for ethnic, religious, and type matches it will reflect the root match class (KURD, CATHOLIC, POLICE OFFICER, etc). May be blank if the system was unable to identify an Actor2.

• **Actor1CountryCode**. The 3-digit CAMEO code for the country affiliation of Actor1.

• **Actor1CountryLabel**. The human-readable name of the country affiliation of Actor1.

• **Actor1KnownGroupCode**. If Actor1 is a known IGO/NGO/rebel organization (alQaeda, United Nations, World Bank, etc) with its own CAMEO code, this field will contain that code.

• **Actor1KnownGroupLabel**. The human-readable formal name for Actor1KnownGroupCode.

• **Actor1EthnicCode**. If the source document specifies the ethnic affiliation of Actor1 and that ethnic group has a CAMEO entry, the CAMEO code is entered here.

NOTE: a few special groups like ARAB may also have entries in the type column due to legacy CAMEO behavior.

• **Actor1EthnicLabel**. The human-readable formal name for Actor1EthnicCode.

• **Actor1Religion1Code**. If the source document specifies the religious affiliation of Actor1 and that religious group has a CAMEO entry, the CAMEO code is entered here.

NOTE: a few special groups like JEW may also have entries in the geographic or type columns due to legacy CAMEO behavior.

•**Actor1Religion1Label**. The human-readable formal name for Actor1Religion1Code.

• **Actor1Religion2Code**. If multiple religious codes are specified for Actor1, this contains the secondary code. Some religion entries automatically use two codes, such as Catholic, which invokes Christianity as Code1 and Catholicism as Code2.

•**Actor1Religion2Label**. The human-readable formal name for Actor1Religion2Code.

• **Actor1Type1Code**. The 3-digit CAMEO code of the CAMEO type or role of Actor1, if specified. This can be a specific role such as Police Forces, Government, Military, Political Opposition, Rebels, etc, a broad role class such as Education, Elites, Media, Refugees, or organizational classes like Non-Governmental Movement. Special codes such as Moderate and Radical may refer to the operational strategy of a group.

• **Actor1Type1Label**. The human-readable formal name for Actor1Type1Code.

• **Actor1Type2Code**. If multiple type/role codes are specified for Actor1, this returns the second code.

• **Actor1Type2Label**. The human-readable formal name for Actor1Type2Code.

• **Actor1Type3Code**. If multiple type/role codes are specified for Actor1, this returns the third code.

• **Actor1Type3Label**. The human-readable formal name for Actor1Type3Code.

These codes are repeated for Actor2, using the prefix Actor2 instead of Actor1. As with Actor1, if no Actor2 could be extracted, these fields will be blank. Only in extremely rare circumstances will both Actor1 and Actor2 be blank.

These fields break out various attributes of the event action and offer several mechanisms for assessing the importance or immediate-term impact of an event.

• **IsRootEvent**. The system codes every event across an entire document, using an array of techniques to deference and link information together. A number of previous projects such as the ICEWS initiative have found that events occurring in the lead paragraph of a document tend to be the most important and are the least likely to have any errors. Thus, this flag can be used as a proxy for the rough importance of an event to create subsets of the event stream.

• **EventCode**. This is the raw CAMEO action code describing the action that Actor1 performed upon Actor2.

• **EventDesc**. This is the human-readable formal label for the given CAMEO action code.

• **EventBaseCode**. CAMEO event codes are defined in a three-level taxonomy. For events at level three in the taxonomy, this yields its level two leaf root node. For example, code 0251 (Appeal for easing of administrative sanctions) would yield an EventBaseCode of 025 (Appeal to yield). This makes it possible to aggregate events at various resolutions of specificity. For events at levels two or one, this field will be set to EventCode.

• **EventBaseDesc**. This is the human-readable formal label for EventBaseCode.

• **EventRootCode**. Similar to EventBaseCode, this defines the root-level category the event code falls under. For example, code 0251 (Appeal for easing of administrative sanctions) has a root code of 02 (Appeal). This makes it possible to aggregate events at various resolutions of specificity. For events at levels two or one, this field will be set to EventCode.

• **EventRootDesc**. This is the human-readable formal label for EventBaseCode.

• **QuadClass**. The entire CAMEO event taxonomy is ultimately broken into four primary classifications: Verbal Cooperation, Material Cooperation, Verbal Conflict, and Material Conflict. This field specifies this primary classification for the event type, allowing analysis at the highest level of aggregation.

• **GoldsteinScale**. Each CAMEO event code is assigned a numeric score from -10 to +10, capturing the likely impact that type of event will have on the stability of a country. This is known as the Goldstein Scale. This field specifies the Goldstein score for each event type. NOTE that this score is based on the type of event, not the specifics of the actual event record being recorded thus two riots, one with 10 people and one with 10,000, will both receive the same Goldstein score. This can be aggregated to various levels of time resolution to yield an approximation of the stability of a geography over time.

• **NumMentions**. This is the total number of mentions of this event across all source documents. Multiple references to an event within a single document also contribute to this count. This can be used as a method of assessing the importance of an event: the more discussion of that event, the more likely it is to be significant. The total universe of source documents and the density of events within them vary over time, so it is recommended that this field be normalized by the average or other measure of the universe of events during the time period of interest.

• **NumSources**. This is the total number of information sources containing one or more mentions of this event. This can be used as a method of assessing the

importance of an event: the more discussion of that event, the more likely it is to be significant. The total universe of sources varies over time, so it is recommended that this field be normalized by the average or other measure of the universe of events during the time period of interest.
• **NumArticles**. This is the total number of source documents containing one or more mentions of this event. This can be used as a method of assessing the importance of an event: the more discussion of that event, the more likely it is to be significant. The 45 total universe of source documents varies over time, so it is recommended that this field be normalized by the average or other measure of the universe of events during the time period of interest.
• **AvgTone**. This is the average tone of all documents containing one or more mentions of this event. The score ranges from -100 (extremely negative) to +100 (extremely positive). Common values range between -10 and +10, with 0 indicating neutral. This can be used as a method of filtering the context of events as a subtle measure of the importance of an event and as a proxy for the impact of that event. For example, a riot event with a slightly negative average tone is likely to have been a minor occurrence, whereas if it had an extremely negative average tone, it suggests a far more serious occurrence. A riot with a positive score likely suggests a very minor occurrence described in the context of a more positive narrative (such as a report of an attack occurring in a discussion of improving conditions on the ground in a country and how the number of attacks per day has been greatly reduced).

• **Actor1GeoType**. This field specifies the geographic resolution of the match type and holds one of the following values: COUNTRY (the match was at the country level), USSTATE (the match was to a US state), USLOC (the match was to a US city or landmark), WORLDLOC (the match was to a city or landmark outside the US). This can be used to filter events by geographic specificity, for example, extracting only those events with a landmark-level geographic resolution for mapping. Note that both 46 COUNTRY and USSTATE matches will still provide a latitude/longitude pair, which will be the centroid of that country or state. Matches to foreign Administrative Division 1s (ADM1s) (the rough equivalent of a US state) will be coded as a WORLDLOC location.
• **Actor1GeoFullname**. This is the full human-readable name of the matched location. In the case of a country it is simply the country name. For US states it is in the format of State, United States, while for all other matches it is in the format of Landmark, State/ADM1, Country. This can be used to label locations when placing events on a map.
• **Actor1GeoCountryCode**. This is the 2-character FIPS10-4 country code for the location.
• **Actor1GeoADM1Code**. This is the 2-character FIPS10-4 administrative division 1 (ADM1) code for the administrative division housing the landmark. In the case of the United States, this is the 2-character shortform of the states name (such as TX for Texas).
• **Actor1Geo** Lat. This is the centroid latitude of the landmark for mapping.
• **Actor1Geo**Long. This is the centroid longitude of the landmark for mapping. These codes are repeated for Actor2 and Action, using those prefixes.

ANEXO II

Códigos CAMEO

Descripción del código	**Identificador del código**
MAKE PUBLIC STATEMENT	010-019
APPEAL	020-029
EXPRESS INTENT TO COOPERATE	030-039
CONSULT	040-046
ENGAGE IN DIPLOMATIC COOPERATION	050-059
ENGAGE IN MATERIAL COOPERATION	060-064
PROVIDE AID	070-075
INVESTIGATE	090-094
DEMAND	100-108
DISAPPROVE	110-116
REJECT	120-129
THREATEN	130-139
EXHIBIT MILITARY POSTURE	150-155
REDUCE RELATIONS	160-1663
COERCE	170-176
YIELD	080-0874
PROTEST	140-1454
ASSAULT	180-186

FIGHT	190-196
ENGAGE IN UNCONVENTIONAL MASS VIOLENCE	200-2042

ANEXO III

Hortownworks HDP 2.2.4 Sandbox with ambari and apache sparks

características:
Ambari:
postgres Hive Metastore :
Mysql Ranger:
Mysql Oozie:
derby (embedded)
Apache Spark
Apache Ambari
Apache Hadoop (HDFS, YARN, Mapreduce)
Apache Falcon
Apache Hive
Apache Hbase
Apache Flume
Apache Hue
Apache Kafka
Apache Knox
Apache Oozie
Apache Pig
Apache Ranger
Apache Solr
Apache Slider
Apache Sqoop
Apache Storm
Apache Tez
Apache Zookeeper

ANEXO IV

Cloudera QuickStart VMs for CDH 5.3.x

Apache Hadoop 2.5.0-cdh5.3.0
Apache Hadoop Mrv1 2.5.0-mr1-cdh5.3.0
Apache Hive 0.13.1-cdh5.3.0
Apache Hbase 0.98.6-cdh5.3.0
Apache ZooKeeper 3.4.5-cdh5.3.0
Apache Sqoop 1.4.5-cdh5.3.0
Apache Pig 0.12.0-cdh5.3.0
Apache Flume 1.x 1.5.0-cdh5.3.0
Apache Oozie 4.0.0-cdh5.3.0
Apache Mahout 0.9-cdh5.3.0
Apache Whirr 0.9.0-cdh5.3.0
Linkedin DataFu 1.1.0-cdh5.3.0
Apache Sqoop2 1.99.4-cdh5.3.0
Apache Sentry 1.4.0-cdh5.3.0
Twitter Parquet 1.5.0-cdh5.3.0
Cloudera Llama 1.0.0-cdh5.3.0
Apache Spark 1.2.0-cdh5.3.0
Apache Crunch 0.11.0-cdh5.3.0
Apache Avro 1.7.6-cdh5.3.0
Kitesdk Kite 0.15.0-cdh5.3.0
Apache Solr 4.4.0-cdh5.3.0
Cloudera Search 1.0.0-cdh5.3.0
Ngdata HBase Indexer 1.5-cdh5.3.0

ANEXO V

```
/**
 *
 */
package Prediccion;

/**
 * @author Alejandro Mejías
 *
 */

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Random;
import java.util.Date;
import java.util.Calendar;
import java.text.SimpleDateFormat;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.OutputStream;
import java.io.PrintStream;

import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.core.Instances;
import weka.experiment.InstanceQuery;
import weka.classifiers.trees.RandomForest;
import weka.core.Attribute;
import weka.core.Instance;
import weka.core.DenseInstance;
import weka.filters.Filter;
import weka.filters.unsupervised.instance.SubsetByExpression;
import weka.core.converters.ArffSaver;
import weka.core.converters.ArffLoader.ArffReader;
import weka.filters.unsupervised.attribute.NumericToNominal;
import weka.filters.unsupervised.instance.Denormalize;
import weka.filters.supervised.instance.StratifiedRemoveFolds;
import weka.filters.unsupervised.attribute.MathExpression;


public class TFGPrediccion {


	private static String[] codigopais = {
		"AND","ARE","AFG","ATG","AIA","ALB","ARM","ANT","AGO","ATA","ARG","ASM","AUT","AUS","ABW","ALA","AZE",
		"BIH","BRB","BGD","BEL","BFA","BGR","BHR","BDI","BEN","BLM","BMU","BRN","BOL","BRA","BHS","BTN","BVT",
		"BWA","BLR","BLZ","CAN","CCK","CAF","COG","CHE","CIV","COK","CHL","CMR","CHN","COL","CRI","CUB","CPV",
		"CXR","CYP","CZE","DEU","DJI","DNK","DMA","DOM","DZA","ECU","EST","EGY","ESH","ERI","ESP","ETH","FIN",
		"FJI","KLK","FSM","FRO","FRA","GAB","GBR","GRD","GEO","GUF","GGY","GHA","GIB","GRL","GMB","GIN","GLP",
		"GNQ","GRC","SGS","GTM","GUM","GNB","GUY","HKG","HMD","HND","HRV","HTI","HUN","IDN","IRL","ISR","IMN",
		"IND","IOT","IRQ","IRN","ISL","ITA","JEY","JAM","JOR","JPN","KEN","KGZ","KHM","KIR","COM","KNA","PRK",
		"KOR","KWT","CYM","KAZ","LAO","LBN","LCA","LIE","LKA","LBR","LSO","LTU","LUX","LVA","LBY","MAR","MCO",
		"MDA","MNE","MDG","MHL","MKD","MLI","MMR","MNG","MAC","MTQ","MRT","MSR","MLT","MUS","MDV","MWI","MEX",
		"MYS","MOZ","NAM","NCL","NER","NFK","NGA","NIC","NLD","NOR","NPL","NRU","NIU","NZL","OMN","PAN","PER",
		"PYF","PNG","PHL","PAK","POL","SPM","PCN","PRI","PSE","PRT","PLW","PRY","QAT","REU","ROU","SRB","RUS",
		"RWA","SAU","SLB","SYC","SDN","SWE","SGP","SHN","SVN","SJM","SVK","SLE","SMR","SEN","SOM","SUR","STP",
		"SLV","SYR","SWZ","TCA","TCD","ATF","TGO","THA","TZA","TJK","TKL","TLS","TKM","TUN","TON","TUR","TTO",
		"TUV","TWN","UKR","UGA","USA","URY","UZB","VAT","VCT","VEN","VGB","VIR","VNM","VUT","WLF","WSM","YEM",
		"MYT","ZAF"
		};

	public static List<String> countries = Arrays.asList(codigopais);

	public static String baseDatos;
	public static String tabla;
	public static String ficheroOut;
	public static String ficheroMuestras;
	public static String ficheroCrudo;
	public static double prediccion;
	public static boolean guardar;

	/**
	 * @return
	 * Constructor básico de la clase
	 * Se inicializa con los valores:
	 * baseDatos = "TFG2015",
	 * tabla = "gdeltdaily",
	 * ficheroOut = "./info.txt";
	 * ficheroCrudo = "./crudo.arff",
	 * ficheroMuestras = "./tratado.arff",
	 * prediccion = 0,
	 * guardar = true
	 */
	public TFGPrediccion(){
		baseDatos = "TFG2015";
		tabla = "gdeltdaily";
		ficheroOut = "./info.txt";
		ficheroCrudo = "./crudo.arff";
		ficheroMuestras = "./tratado.arff";
		prediccion = 0;
		guardar = true;
	}
```

```
public TFGPrediccion(String bd, String tb, String fout, String fc, String fm, boolean g){
            baseDatos = bd;
            tabla = tb;
            ficheroOut = fout;
            ficheroCrudo = fc;
            ficheroMuestras = fm;
            prediccion = 0;
            guardar = g;
}

/**
 * @return
 * Cambia el nombre del fichero
 * de datos estadísticos finales
 */
public void setFicheroOut(String file){ficheroOut=file;}
/**
 * @return
 * Devuelve el nombre del fichero
 * de datos estadísticos finales
 */
public String getFicheroOut(){ return ficheroOut;}
/**
 * @return
 * Cambia el nombre del fichero
 * de muestras para realizar el test
 */
public void setFicheroMuestras(String file){ficheroMuestras=file;}
/**
 * @return
 * Devuelve el nombre del fichero
 * con las muestras obtenidas de la tabla
 */
public String getFicheroMuestras(){ return ficheroMuestras;}
/**
 * @return
 * Cambia el nombre del fichero
 * con las muestras obtenidas de la tabla
 */
public void setFicheroCrudo(String file){ficheroCrudo=file;}
/**
 * @return
 * Devuelve el nombre del fichero
 * con las muestras obtenidas de la tabla
 */
public String getFicheroCrudo(){ return ficheroCrudo;}
/**
 * @return
 * Indica si se quiere guardar la información en ficheros (true)
 * o no (false)
 */
public void setGuardar(boolean b){guardar = b;}
/**
 * @return
 * double:
 * Devuelve El valor del atributo guardar
 * true = Si se guarda
 * false = no se guarda
 */
public boolean getGuardar(){return guardar;}
/**
 * @return
 * Cambia el nombre de la
 * BAse de datos a la que acceder
 */
public void setbaseDatos(String bd){
            baseDatos = bd;
}
/**
 * @return
 * double:
 * Devuelve el porcentaje de predicción del elemento
 */
public double getPrediccion(){return prediccion;}

/**
 * @return
 * Cambia el nombre de la
 * tabla a la que acceder
 */
public void settabla(String tb){tabla = tb;}
/**
 * @return
 * Devuelve String con el nombre de la
 * base de datos a la que acceder
 */
public String getbaseDatos(){return baseDatos;}
/**
 * @return
 * Devuelve el nombre de la
 * tabla a la que acceder
 */
public String gettabla(){return tabla;}

private static Instances crearIns(int i) throws Exception{

            Attribute attr1 = new Attribute("avgt_media_fecha_previo_1");
    Attribute attr2 = new Attribute("goldstein_media_fecha_previo_1");
    Attribute attr3 = new Attribute("numarticles_media_fecha_previo_1");
```

```
    Attribute attr4 = new Attribute("avgt_media_fecha_previo_2");
    Attribute attr5 = new Attribute("goldstein_media_fecha_previo_2");
    Attribute attr6 = new Attribute("numarticles_media_fecha_previo_2");
    Attribute attr7 = new Attribute("avgt_media_fecha_previo_3");
    Attribute attr8 = new Attribute("goldstein_media_fecha_previo_3");
    Attribute attr9 = new Attribute("numarticles_media_fecha_previo_3");
    Attribute attr10 = new Attribute("avgt_media_fecha_previo_4");
    Attribute attr11 = new Attribute("goldstein_media_fecha_previo_4");
    Attribute attr12 = new Attribute("numarticles_media_fecha_previo_4");
    Attribute attr13 = new Attribute("avgt_media_fecha_previo_5");
    Attribute attr14 = new Attribute("goldstein_media_fecha_previo_5");
    Attribute attr15 = new Attribute("numarticles_media_fecha_previo_5");
    Attribute attr16 = new Attribute("avgt_media_fecha_previo_6");
    Attribute attr17 = new Attribute("goldstein_media_fecha_previo_6");
    Attribute attr18 = new Attribute("numarticles_media_fecha_previo_6");
    Attribute attr19 = new Attribute("avgt_media_fecha_previo_7");
    Attribute attr20 = new Attribute("goldstein_media_fecha_previo_7");
    Attribute attr21 = new Attribute("numarticles_media_fecha_previo_7");
    Attribute attr22 = new Attribute("avgt_media_fecha_previo_8");
    Attribute attr23 = new Attribute("goldstein_media_fecha_previo_8");
    Attribute attr24 = new Attribute("numarticles_media_fecha_previo_8");
    Attribute attr25 = new Attribute("avgt_media_fecha_previo_9");
    Attribute attr26 = new Attribute("goldstein_media_fecha_previo_9");
    Attribute attr27 = new Attribute("numarticles_media_fecha_previo_9");
    Attribute attr28 = new Attribute("avgt_media_fecha_previo_10");
    Attribute attr29 = new Attribute("goldstein_media_fecha_previo_10");
    Attribute attr30 = new Attribute("numarticles_media_fecha_previo_10");
    //Attribute attr31 = new Attribute("clase");
    //Attribute attr32 = new Attribute("rootcodeavg");
    Attribute attr31 = new Attribute("rootcodeavg");
    Attribute attr32 = new Attribute("class");


    ArrayList<Attribute> attributes = new ArrayList<Attribute>();

    attributes.add(attr1);
    attributes.add(attr2);
    attributes.add(attr3);
    attributes.add(attr4);
    attributes.add(attr5);
    attributes.add(attr6);
    attributes.add(attr7);
    attributes.add(attr8);
    attributes.add(attr9);
    attributes.add(attr10);
    attributes.add(attr11);
    attributes.add(attr12);
    attributes.add(attr13);
    attributes.add(attr14);
    attributes.add(attr15);
    attributes.add(attr16);
    attributes.add(attr17);
    attributes.add(attr18);
    attributes.add(attr19);
    attributes.add(attr20);
    attributes.add(attr21);
    attributes.add(attr22);
    attributes.add(attr23);
    attributes.add(attr24);
    attributes.add(attr25);
    attributes.add(attr26);
    attributes.add(attr27);
    attributes.add(attr28);
    attributes.add(attr29);
    attributes.add(attr30);
    attributes.add(attr31);
    attributes.add(attr32);

    return new Instances("Training dataset", attributes, 0);

}

private static Instances formarDataset(Instances ins, String pais, int dia, int previo) throws Exception{

          Instances istraining, fin;
          NumericToNominal nt;
          String expre;
          Double fec;
          int fe;
          Instances retraining;

          istraining = ins;

          fin = crearIns(1);

          expre = "(ATT4 > -2) and (ATT5 > -4) and (ATT3 > 40)";
          SubsetByExpression su = new SubsetByExpression();
          su.setExpression(expre);
          su.setInputFormat(istraining);

          istraining=Filter.useFilter(istraining, su);

          Denormalize de = new Denormalize();

          String[] opciones= new String[5];

    de = new Denormalize();

    opciones[0]="-G";
```

```
        opciones[1]="first";
        opciones[2]="-A";
        opciones[3]="Maximum";
        opciones[4]="-S";

        PrintStream stderr = System.err; // Save stderr stream.

        System.setErr(new PrintStream(new OutputStream() {
    public void write(int b) {
        //DO NOTHING
    }
}));

de.setOptions(opciones);
de.setInputFormat(istraining);

retraining = Filter.useFilter(istraining, de);

            System.setErr(stderr);

            //System.out.println("Agregamos "+retraining.numInstances()+" Instancias");

            for(int i=0; i < retraining.numInstances(); i++){

                        fec = new Double(retraining.instance(i).value(0));
                        fe = fec.intValue();
                        fin = InstanciaDiasAntetriores(retraining, istraining, fin, fe, previo);
            }

            if (istraining != null){

                        nt = new NumericToNominal();
                        nt.setAttributeIndices("32");
                        nt.setInputFormat(fin);

                        fin = Filter.useFilter(fin, nt);

                        fin.setClass(fin.attribute("class"));

    ArffSaver saver = new ArffSaver();
    saver.setInstances(fin);
    //saver.
    saver.setFile(new File(ficheroMuestras));
    saver.writeBatch();

    //System.out.println("salvado dataset");

    return fin;


}else{
   //System.out.println("No hay instancia de entrenamiento salgo");
   return null;
}



   }
   /*
    * método para generar el conjunto de entrenamiento
    * tomará 7 archivos con los dataset obtenidos desde el servidor Hadoop
    * para los paises: Ucrania, Turquia, Mexico, Rusia, España, Siria y Tunez.
    *
    * El objetivo de este método es generar el conjunto de datos de entrenamiento y
    * la instancia de test para realizar la predicción
    * parámetros:
    * Input:
    * int dia: dia sobre el que realizar la prediccion
    * Output:
    * Instances: conjunto de
    */
   private static Instances crear(int diasVista) throws Exception{

            Instances istraining, fin;
            String fichero;
            NumericToNominal nt;
            String expre;
            Double fec;
            int fe;
            Instances retraining;

            fin = crearIns(1);

            for(int index=1;index<2;index++){

                        System.out.println();
                        fichero = "./data/" + Integer.toString(index) + "invtest.arff";
                        BufferedReader reader = new BufferedReader(new FileReader(fichero));
                        ArffReader arff = new ArffReader(reader);
                        istraining = arff.getData();

                        expre = "(ATT4 > -2) and (ATT5 > -4) and (ATT3 > 40)";

                        SubsetByExpression su = new SubsetByExpression();
                        su.setExpression(expre);
                        su.setInputFormat(istraining);

                        Denormalize de = new Denormalize();
```

```
                        String[] opciones= new String[5];

                    de = new Denormalize();

                    opciones[0]="-G";
                    opciones[1]="first";
                    opciones[2]="-A";
                    opciones[3]="Maximum";
                    opciones[4]="-S";

                    PrintStream stderr = System.err; // Save stderr stream.

                    System.setErr(new PrintStream(new OutputStream() {
                  public void write(int b) {
                     //DO NOTHING
                  }
               }));

               de.setOptions(opciones);
               de.setInputFormat(istraining);

               retraining = Filter.useFilter(istraining, de);

                        System.setErr(stderr);

                        System.out.println("Agregamos "+retraining.numInstances()+" Instancias");

                        for(int i=60; i < retraining.numInstances(); i++){

                                 fec = new Double(retraining.instance(i).value(0));
                                 fe = fec.intValue();
                                 fin = InstanciaDiasAntetriores(retraining, istraining, fin, fe, diasVista);
                        }
                  }
                  if (fin != null){

                        nt = new NumericToNominal();
                        nt.setAttributeIndices("32");
                        nt.setInputFormat(fin);

                        fin = Filter.useFilter(fin, nt);

                        fin.setClass(fin.attribute("class"));

                        if (guardar){
                                 ArffSaver saver = new ArffSaver();
                                 saver.setInstances(fin);
          //saver.
                                 saver.setFile(new File("./data/fini1class.arff"));
                                 saver.writeBatch();
                        }

          System.out.println("salvado dataset");

          return fin;


       }else{
          System.out.println("No hay instancia de entrenamiento salgo");
          return null;
       }



          }
/*
 * método para generar los datos predictivos desde el dataset
 *
 * parámetros:
 * Input:
 * Instances fin: Dataset con el conjunto completo de datos para generar la instancia de test
 * Instances testeo: Dataset con el conjunto de muestras para entrenar el modelo
 * Int dia: día sobre el que generar la predicción
 * Int previo: cantidad de dias previos sobre los que hacer la predicción
 * Output:
 * genera un fichero con los resultados estadísticos obtenidos.
 */
          private static double testear(Instances fin, Instances testeo, int dia, int previo) throws Exception
          {

                    RandomForest rf;
                    String[] options;
                    StratifiedRemoveFolds filter;
                    Instances test;
                    Instances train;
                    Classifier cModel;
                    Evaluation eTest;
                    String strSummary;
                    String registro;
                    Date date;
                    Calendar cal;
                    SimpleDateFormat originalFormat;
                    PrintStream log;
                    PrintStream stdout;

       // usar StratifiedRemoveFolds para separar aleatoriamente los
       filter = new StratifiedRemoveFolds();

       // Creo el conjunto de opciones
```

```
        options = new String[6];

        options[0] = "-N";                // Se indica que se requiere un número de folders
        options[1] = Integer.toString(4);  // separa los datos en 4 folders
        options[2] = "-F";                // Se especifica un único folder para reducir el tiempo de tto.
        options[3] = Integer.toString(1);  // Se seleccion el primer folder
        options[4] = "-S";                // Se indica que se vá a usar una semilla aleatoria
        options[5] = Integer.toString(1);  // La semilla aleatoria va a ser 1

        filter.setOptions(options);
        filter.setInputFormat(fin);
        filter.setInvertSelection(false);

        // se aplica el filtro para el conjunto de test
        test = Filter.useFilter(fin, filter);

        //  Se cambia la selección para tener la inversa, o sea los tres folders restantes
        filter.setInvertSelection(true);
        train = Filter.useFilter(fin, filter); //Se genera el conjunto de datos de entrenamiento

        //Se aplica el entrenamiento
        rf = new RandomForest();
                    rf.setNumExecutionSlots(2);
                    rf.setNumTrees(500);
                    rf.setPrintTrees(true);
                    rf.setNumFeatures(10);

        cModel = (Classifier) rf;


        cModel.buildClassifier(train);

        eTest = new Evaluation(train);
        Random rand = new Random(1);
        eTest.crossValidateModel(cModel, fin, 3, rand);

        strSummary = eTest.toSummaryString();

        stdout = System.out;
        /*
         * System.out.close();


        System.out.println(strSummary);
        System.out.println("Verdaderos positivos en 1: "+eTest.truePositiveRate(1));
        System.out.println("Verdaderos negativos en 1: "+eTest.trueNegativeRate(1));
        System.out.println("Verdaderos positivos en 0: "+eTest.truePositiveRate(0));
        System.out.println("Verdaderos negativos en 0: "+eTest.trueNegativeRate(0));
        System.out.println("BAC: "+ (eTest.truePositiveRate(1)+eTest.trueNegativeRate(1))/2);

        System.setOut(stdout);
        */

        // Get the confusion matrix
        //double[][] cmMatrix = eTest.confusionMatrix();

        //
        // fDistribution[0] probabilidad de ser “positivo”
        // fDistribution[1] probabilidad de ser “negativo”

        Instance insta;
        insta = prueba(testeo, dia, previo);
        test.setClassIndex(test.numAttributes() - 1);

        insta.setDataset(test);

        double[] fDistribution = cModel.distributionForInstance(insta);


        //System.out.println(fDistribution.length);
        //System.out.println("Para el dia "+ Integer.toString(dia)+ " existe una probabilidad del " + fDistribution[0]*100 + "%
de pertenecer a la clase " + insta.value(31));
        //System.out.println("Y una probabilidad de " + fDistribution[1]*100 + "de no pertenecer a esa clase a " + previo + "
dias vista"  );

        if(guardar){
           stdout = System.out;
           log = new PrintStream(new FileOutputStream("./data/out.txt",true));

           System.setOut(log);


                    originalFormat = new SimpleDateFormat("yyyyMMdd");
                                date = originalFormat.parse(Integer.toString(dia));
                                cal = Calendar.getInstance();
                                cal.setTime(date);
                                cal.add(Calendar.DATE, (-1)*(previo));

                    registro = previo + ";"+ fDistribution[0]*100 + ";" + eTest.truePositiveRate(1) + ";" +
eTest.trueNegativeRate(1) + ";" + eTest.relativeAbsoluteError() + ";" + eTest.rootRelativeSquaredError()
+";"+originalFormat.format(cal.getTime());
                    System.out.println(registro);

                    System.setOut(stdout);
        }
        prediccion = fDistribution[1]*100;
        return prediccion;

           }
```

```
   /*
    * método para crear la instancia con el día a evaluar
    * parámetros:
    * Input:
    * Instances ins: Instancia con el conjunto de datos obtenidos
    * de GDELT (no es necesario se puede hacer una consulta a la hora de llamar el procedimiento)
    * Int dia: fecha a ser evaluada (obligatorio)
    * int distanciaDias: previsión de días (no es necesario se puede calcular en base a la fecha
    * actual)
    * Output: Instancia para evaluar
    */
   private static Instance prueba(Instances ins, int dia, int distanciaDias) throws Exception{

              Date fecha;
              Calendar cal;
              SimpleDateFormat originalFormat;
              double[] instancia;
              SubsetByExpression su;
              String expre;
              Instances parcial;
              double avgNumArt,avgAvgtone,avgGoldstein, avgRootcode;
              Instance insta;
              String finic,ffin;
              String[] opciones;


       instancia = generarInstanciaVacia();

       for (int i=0;i < instancia.length-2; i++){
   instancia[i++]=3; //avgtone
   instancia[i++]=2; //goldstein
   instancia[i]=0;
}
instancia[30]=0;
instancia[31]=0;

String fa;
fa = Integer.toString(dia);
       originalFormat = new SimpleDateFormat("yyyyMMdd");
              fecha = originalFormat.parse(fa);
              cal = Calendar.getInstance();
              cal.setTime(fecha);


              /*
               * se calculan las medias para los 60 dias anteriores a la distancia de dias, distanciaDias, proporcionada
               */
              cal.add(Calendar.DATE, (-1)*(distanciaDias));
              finic = originalFormat.format(cal.getTime());

              cal.add(Calendar.DATE, (-1)*60);
              ffin = originalFormat.format(cal.getTime());

              su = new SubsetByExpression();
              expre = "(ATT1 >" + ffin+ ") and (ATT1 < " + finic + ")";
       su.setExpression(expre);
       su.setInputFormat(ins);

       parcial = Filter.useFilter(ins, su);


       avgNumArt=parcial.attributeStats(2).numericStats.mean;
       avgGoldstein = parcial.attributeStats(3).numericStats.mean;
       avgAvgtone = parcial.attributeStats(4).numericStats.mean;
       avgRootcode = parcial.attributeStats(1).numericStats.mean;

       avgAvgtone = 1;
       avgGoldstein = 1;

       /*
        * ahora sólo escogemos los diez dias anteriores a dia-distanciaDias
        */

       cal.add(Calendar.DATE, 60);

              int fini=-10;
              do{
                        fini--;
                        cal.add(Calendar.DATE, fini);
                   expre = "(ATT1 >" + ffin + ") and (ATT1 <" + originalFormat.format(cal.getTime()) + ")";
                   su.setExpression(expre);
                   su.setInputFormat(ins);
                   parcial = Filter.useFilter(ins, su);
                   if (fini < -50) break;
              }while (parcial.numInstances() < 11);


       /*
        * Procedo a agrupar los eventos por día para obtener los valores buscados de
        * M'c(i,i-k)/media, V'c(i,i-k)/media, A'c(i, i-k)/media con k = {1,...,9}
        * ello se consigue con la herramienta Weka.filter.unsupervised.Denormalize
        * utilizando las opciones de agrupar por sqldate (el primer atributo)
        * Y escogiendo el método de agrupación mediante la media de los valores
        * numarticles, avgtone y goldsteinscale de esos 13 días
        */
       Denormalize de;
```

```
            PrintStream stderr = System.err; // Salvar stderr stream.

            System.setErr(new PrintStream(new OutputStream() {
            public void write(int b) {
                //DO NOTHING
            }
        }));

            opciones= new String[5];

            de = new Denormalize();

            opciones[0]="-G";
            opciones[1]="first";
            opciones[2]="-A";
            opciones[3]="Average";
            opciones[4]="-S";

        de.setOptions(opciones);
        de.setInputFormat(parcial);

        parcial = Filter.useFilter(parcial, de);

        System.setErr(stderr);

    try{

                if (parcial.instance(0) != null){
                          instancia[0] = parcial.instance(0).value(4)/avgAvgtone;
                          instancia[1] = parcial.instance(0).value(3)/avgGoldstein;
                          instancia[2] = parcial.instance(0).value(2)/avgNumArt;

                          instancia[30] += parcial.instance(0).value(1);
                }
                else{
                   instancia[0] = 0;
                   instancia[1] = 0;
                   instancia[2] = 0;
                }

                if (parcial.instance(1) != null){
                          instancia[3] = parcial.instance(1).value(4)/avgAvgtone;
                          instancia[4] = parcial.instance(1).value(3)/avgGoldstein;
                          instancia[5] = parcial.instance(1).value(2)/avgNumArt;

                          instancia[30] += parcial.instance(1).value(1);
                }else{
                   instancia[3] = 0;
                   instancia[4] = 0;
                   instancia[5] = 0;
                }

                if (parcial.instance(2) != null){
                          instancia[6] = parcial.instance(2).value(4)/avgAvgtone;
                          instancia[7] = parcial.instance(2).value(3)/avgGoldstein;
                          instancia[8] = parcial.instance(2).value(2)/avgNumArt;

                          instancia[30] += parcial.instance(2).value(1);
                }else{
                   instancia[6] = 0;
                   instancia[7] = 0;
                   instancia[8] = 0;
                }

                if (parcial.instance(3) != null){
                          instancia[9] = parcial.instance(3).value(4)/avgAvgtone;
                          instancia[10] = parcial.instance(3).value(3)/avgGoldstein;
                          instancia[11] = parcial.instance(3).value(2)/avgNumArt;

                          instancia[30] += parcial.instance(3).value(1);
                }else{
                   instancia[9] = 0;
                   instancia[10] = 0;
                   instancia[11] = 0;
                }

                if (parcial.instance(4) != null){
                          instancia[12] = parcial.instance(4).value(4)/avgAvgtone;
                          instancia[13] = parcial.instance(4).value(3)/avgGoldstein;
                          instancia[14] = parcial.instance(4).value(2)/avgNumArt;

                          instancia[30] += parcial.instance(4).value(1);
                }else{
                   instancia[12] = 0;
                   instancia[13] = 0;
                   instancia[14] = 0;
                }

                if (parcial.instance(5) != null){
                          instancia[15] = parcial.instance(5).value(4)/avgAvgtone;
                          instancia[16] = parcial.instance(5).value(3)/avgGoldstein;
                          instancia[17] = parcial.instance(5).value(2)/avgNumArt;

                          instancia[30] += parcial.instance(5).value(1);
                             }else{
                   instancia[15] = 0;
                   instancia[16] = 0;
                   instancia[17] = 0;
                }
```

```
        if (parcial.instance(6) != null){
                  instancia[18] = parcial.instance(6).value(4)/avgAvgtone;
                  instancia[19] = parcial.instance(6).value(3)/avgGoldstein;
                  instancia[20] = parcial.instance(6).value(2)/avgNumArt;

                  instancia[30] += parcial.instance(6).value(1);
                     }else{
                     instancia[18] = 0;
                     instancia[19] = 0;
                     instancia[20] = 0;
               }

        if (parcial.instance(7) != null){
                  instancia[21] = parcial.instance(7).value(4)/avgAvgtone;
                  instancia[22] = parcial.instance(7).value(3)/avgGoldstein;
                  instancia[23] = parcial.instance(7).value(2)/avgNumArt;

                  instancia[30] += parcial.instance(7).value(1);
                     }else{
                              instancia[21] = 0;
                              instancia[22] = 0;
                              instancia[23] = 0;
                     }

        if (parcial.instance(8) != null){
                  instancia[24] = parcial.instance(8).value(4)/avgAvgtone;
                  instancia[25] = parcial.instance(8).value(3)/avgGoldstein;
                  instancia[26] = parcial.instance(8).value(2)/avgNumArt;

                  instancia[30] += parcial.instance(8).value(1);
               }else{
                     instancia[24] = 0;
                     instancia[25] = 0;
                     instancia[26] = 0;
               }

        if (parcial.instance(9) != null){
                  instancia[27] = parcial.instance(9).value(4)/avgAvgtone;
                  instancia[28] = parcial.instance(9).value(3)/avgGoldstein;
                  instancia[29] = parcial.instance(9).value(2)/avgNumArt;

                  instancia[30] += parcial.instance(9).value(1);
               }else{
                     instancia[27] = 0;
                     instancia[28] = 0;
                     instancia[29] = 0;
               }


        instancia[30] = instancia[30]/avgRootcode;
        /*
         * como lo que interesa es saber la probabilidad de conflicto
         * la clase a estimar va a ser 1
         */
        instancia[31] = 1;

        insta = new DenseInstance(1.0, instancia);
        return insta;
           }
           catch (Exception e){

                     insta = null;
                     return insta;

           }

}

private static double[] generarInstanciaVacia(){

           ArrayList<Attribute> attributes;
           double[] instancia;

           Attribute attr1 = new Attribute("avgt_media_fecha_previo_1");
    Attribute attr2 = new Attribute("goldstein_media_fecha_previo_1");
    Attribute attr3 = new Attribute("numarticles_media_fecha_previo_1");
    Attribute attr4 = new Attribute("avgt_media_fecha_previo_2");
    Attribute attr5 = new Attribute("goldstein_media_fecha_previo_2");
    Attribute attr6 = new Attribute("numarticles_media_fecha_previo_2");
    Attribute attr7 = new Attribute("avgt_media_fecha_previo_3");
    Attribute attr8 = new Attribute("goldstein_media_fecha_previo_3");
    Attribute attr9 = new Attribute("numarticles_media_fecha_previo_3");
    Attribute attr10 = new Attribute("avgt_media_fecha_previo_4");
    Attribute attr11 = new Attribute("goldstein_media_fecha_previo_4");
    Attribute attr12 = new Attribute("numarticles_media_fecha_previo_4");
    Attribute attr13 = new Attribute("avgt_media_fecha_previo_5");
    Attribute attr14 = new Attribute("goldstein_media_fecha_previo_5");
    Attribute attr15 = new Attribute("numarticles_media_fecha_previo_5");
    Attribute attr16 = new Attribute("avgt_media_fecha_previo_6");
    Attribute attr17 = new Attribute("goldstein_media_fecha_previo_6");
    Attribute attr18 = new Attribute("numarticles_media_fecha_previo_6");
    Attribute attr19 = new Attribute("avgt_media_fecha_previo_7");
    Attribute attr20 = new Attribute("goldstein_media_fecha_previo_7");
    Attribute attr21 = new Attribute("numarticles_media_fecha_previo_7");
    Attribute attr22 = new Attribute("avgt_media_fecha_previo_8");
    Attribute attr23 = new Attribute("goldstein_media_fecha_previo_8");
    Attribute attr24 = new Attribute("numarticles_media_fecha_previo_8");
    Attribute attr25 = new Attribute("avgt_media_fecha_previo_9");
```

```
                Attribute attr26 = new Attribute("goldstein_media_fecha_previo_9");
                Attribute attr27 = new Attribute("numarticles_media_fecha_previo_9");
                Attribute attr28 = new Attribute("avgt_media_fecha_previo_10");
                Attribute attr29 = new Attribute("goldstein_media_fecha_previo_10");
                Attribute attr30 = new Attribute("numarticles_media_fecha_previo_10");
                Attribute attr31 = new Attribute("rootcodeavg");
                Attribute attr32 = new Attribute("class");


                attributes = new ArrayList<Attribute>();

                attributes.add(attr1);
                attributes.add(attr2);
                attributes.add(attr3);
                attributes.add(attr4);
                attributes.add(attr5);
                attributes.add(attr6);
                attributes.add(attr7);
                attributes.add(attr8);
                attributes.add(attr9);
                attributes.add(attr10);
                attributes.add(attr11);
                attributes.add(attr12);
                attributes.add(attr13);
                attributes.add(attr14);
                attributes.add(attr15);
                attributes.add(attr16);
                attributes.add(attr17);
                attributes.add(attr18);
                attributes.add(attr19);
                attributes.add(attr20);
                attributes.add(attr21);
                attributes.add(attr22);
                attributes.add(attr23);
                attributes.add(attr24);
                attributes.add(attr25);
                attributes.add(attr26);
                attributes.add(attr27);
                attributes.add(attr28);
                attributes.add(attr29);
                attributes.add(attr30);
                attributes.add(attr31);
                attributes.add(attr32);

                instancia = new double[32];

                        return instancia;
            }

            /*
             * Metodo para crear cada una de las isntancias de dias anteriores desde
             * el dataset original obtenido de la tabla gdeltdaily
             * en este método se generan
             * V'cs(i,i+k) la llamaré goldstein_media
             * A'cs(i,i+k) la llamaré avgt_media
             * M'cs(i,i+k) la llamaré numart_media
             * i el día
             * k el desplazamiento en días, (i+k) se calcula dentro del bucle
             * ins es el conjuno de datos que será los 60 días anteriores de eventos, necesarios para calcula la media de los 30
anteriores
             * parámetros
             * Input:
             * Instances ins: dataset original
             * Instances reins: dataset agrupado por el atributo sqldate
             * Instances fin: dataset resultante
             * int dia: dia sobre el que crear la instancia
             * int previsto: cantidad de dias previos sobre los que calcular los atributos de la instancia
             * Output:
             * Instances fin: Instancia resultante
             */

            private static Instances InstanciaDiasAntetriores(Instances ins, Instances reins, Instances fin, int dia, int
previsto) throws Exception{

                        double[] instancia;
                        Date date;
                        Calendar cal;
                SimpleDateFormat originalFormat;
                SubsetByExpression su;
                        String expre;
                        Instances parcial;
                double avgNumArt,avgAvgtone,avgGoldstein, avgRootCode;
                Denormalize de;
                        String[] opciones;

                instancia = generarInstanciaVacia();
                /*
                 * inicialización de la instancia
                 */

          for (int i=0;i < instancia.length-2; i++){
             instancia[i++]=3; //avgtone
             instancia[i++]=2; //goldstein
             instancia[i]=0;
          }
          instancia[30]=0;
          instancia[31]=0;

                originalFormat = new SimpleDateFormat("yyyyMMdd");
```

```
                date = originalFormat.parse(Integer.toString(dia));
                cal = Calendar.getInstance();
                cal.setTime(date);


                /*
                 * calculo medias para los 60 dias anteriores
                 */
                cal.add(Calendar.DATE, -60);
                su = new SubsetByExpression();
                expre = "(ATT1 <" + Integer.toString(dia)+ ") and (ATT1 > " + originalFormat.format(cal.getTime()) + ")";
            su.setExpression(expre);
            su.setInputFormat(reins);

            parcial = Filter.useFilter(reins, su);

            avgNumArt = 0;
            avgAvgtone = 0;
            avgGoldstein = 0;
            avgRootCode = 0;

            /*
             * calculo la media de los 60 días anteriores utilizando las estadísticas del dataset
             */
            avgNumArt=parcial.attributeStats(2).numericStats.mean;
            avgAvgtone = 1;
            avgGoldstein = 1;
            avgRootCode = parcial.attributeStats(1).numericStats.mean;

            avgAvgtone = 1;
            avgGoldstein = 1;

            parcial.clear();

            //tomo el actual y los 10 dias anteriores con datos
            date = originalFormat.parse(Integer.toString(dia));
                cal = Calendar.getInstance();
                cal.setTime(date);


                /*
                 * Procedo a filtrar los datos para los 13 días anteriores
                 * Debido a que tras el filtrado inicial muchos eventos suprefluos han sido eliminados
                 * y se roducen bastantes faltas de diez días anteriores
                 * Este valor se ha hallado de forma puramente empírica
                */
                int finic=-10;
                do{
                        finic--;
                        cal.add(Calendar.DATE, finic);
                    expre = "(ATT1 <" + Integer.toString(dia)+ ") and (ATT1 >" + originalFormat.format(cal.getTime()) +
")";
                    su.setExpression(expre);
                    su.setInputFormat(reins);
                    parcial = Filter.useFilter(reins, su);
                    if (finic < -50) break;
                }while (parcial.numInstances() < 11);


            /*
             * Procedo a agrupar los eventos por día para obtener los valores buscados de
             * M'c(i,i-k)/media, V'c(i,i-k)/media, A'c(i, i-k)/media con k = {1,...,9}
             * ello se consigue con la herramienta Weka.filter.unsupervised.Denormalize
             * utilizando las opciones de agrupar por sqldate (el primer atributo)
             * Y escogiendo el método de agrupación mediante la media de los valores
             * numarticles, avgtone y goldsteinscale de esos 13 días
             */
            opciones= new String[5];
            de = new Denormalize();

            opciones[0]="-G";
            opciones[1]="first";
            opciones[2]="-A";
            opciones[3]="Average";
            opciones[4]="-S";

            PrintStream stderr = System.err; // Save stderr stream.
            System.setErr(new PrintStream(new OutputStream() {
        public void write(int b) {
            //DO NOTHING
        }
    }));

    de.setOptions(opciones);
    de.setInputFormat(parcial);

    parcial = Filter.useFilter(parcial, de);

    System.setErr(stderr);

    /*
     * añado los valores para el evento de este día con los eventos de los 10 días anteriores
     * suavizados con la media para los 60 días anteriores
     */

    try{

       if (parcial.instance(0) != null){
                        instancia[0] = parcial.instance(0).value(4)/avgAvgtone;
```

```
          instancia[1] = parcial.instance(0).value(3)/avgGoldstein;
          instancia[2] = parcial.instance(0).value(2)/avgNumArt;

          instancia[30] += parcial.instance(0).value(1);
}
else{
   instancia[0] = 0;
   instancia[1] = 0;
   instancia[2] = 0;
}

if (parcial.instance(1) != null){
          instancia[3] = parcial.instance(1).value(4)/avgAvgtone;
          instancia[4] = parcial.instance(1).value(3)/avgGoldstein;
          instancia[5] = parcial.instance(1).value(2)/avgNumArt;

          instancia[30] += parcial.instance(1).value(1);
}else{
   instancia[3] = 0;
   instancia[4] = 0;
   instancia[5] = 0;
}

if (parcial.instance(2) != null){
          instancia[6] = parcial.instance(2).value(4)/avgAvgtone;
          instancia[7] = parcial.instance(2).value(3)/avgGoldstein;
          instancia[8] = parcial.instance(2).value(2)/avgNumArt;

          instancia[30] += parcial.instance(2).value(1);
}else{
   instancia[6] = 0;
   instancia[7] = 0;
   instancia[8] = 0;
}

if (parcial.instance(3) != null){
          instancia[9] = parcial.instance(3).value(4)/avgAvgtone;
          instancia[10] = parcial.instance(3).value(3)/avgGoldstein;
          instancia[11] = parcial.instance(3).value(2)/avgNumArt;

          instancia[30] += parcial.instance(3).value(1);
}else{
   instancia[9] = 0;
   instancia[10] = 0;
   instancia[11] = 0;
}

if (parcial.instance(4) != null){
          instancia[12] = parcial.instance(4).value(4)/avgAvgtone;
          instancia[13] = parcial.instance(4).value(3)/avgGoldstein;
          instancia[14] = parcial.instance(4).value(2)/avgNumArt;

          instancia[30] += parcial.instance(4).value(1);
}else{
   instancia[12] = 0;
   instancia[13] = 0;
   instancia[14] = 0;
}

if (parcial.instance(5) != null){
          instancia[15] = parcial.instance(5).value(4)/avgAvgtone;
          instancia[16] = parcial.instance(5).value(3)/avgGoldstein;
          instancia[17] = parcial.instance(5).value(2)/avgNumArt;

          instancia[30] += parcial.instance(5).value(1);
            }else{
   instancia[15] = 0;
   instancia[16] = 0;
   instancia[17] = 0;
}

if (parcial.instance(6) != null){
          instancia[18] = parcial.instance(6).value(4)/avgAvgtone;
          instancia[19] = parcial.instance(6).value(3)/avgGoldstein;
          instancia[20] = parcial.instance(6).value(2)/avgNumArt;

          instancia[30] += parcial.instance(6).value(1);
            }else{
            instancia[18] = 0;
            instancia[19] = 0;
            instancia[20] = 0;
      }

if (parcial.instance(7) != null){
          instancia[21] = parcial.instance(7).value(4)/avgAvgtone;
          instancia[22] = parcial.instance(7).value(3)/avgGoldstein;
          instancia[23] = parcial.instance(7).value(2)/avgNumArt;

          instancia[30] += parcial.instance(7).value(1);
            }else{
                     instancia[21] = 0;
                     instancia[22] = 0;
                     instancia[23] = 0;
            }

if (parcial.instance(8) != null){
          instancia[24] = parcial.instance(8).value(4)/avgAvgtone;
          instancia[25] = parcial.instance(8).value(3)/avgGoldstein;
          instancia[26] = parcial.instance(8).value(2)/avgNumArt;
```

```
                        instancia[30] += parcial.instance(8).value(1);
                    }else{
                           instancia[24] = 0;
                           instancia[25] = 0;
                           instancia[26] = 0;
                    }

                if (parcial.instance(9) != null){
                           instancia[27] = parcial.instance(9).value(4)/avgAvgtone;
                           instancia[28] = parcial.instance(9).value(3)/avgGoldstein;
                           instancia[29] = parcial.instance(9).value(2)/avgNumArt;

                           instancia[30] += parcial.instance(9).value(1);
                    }else{
                           instancia[27] = 0;
                           instancia[28] = 0;
                           instancia[29] = 0;
                    }

                //inicializo la clase del evento
                instancia[31] = 0;
                //Suavizo la media de la média de códigos de evento para los 60 días anteriores
                instancia[30] = instancia[30];///avgRootCode;
                //Clasifico el evento para el día actual
                if (clase(reins, dia, previsto)){
                   instancia[31] = 1;
                }

                //Añado la instancia al conjunto de entrenamiento
                fin.add(new DenseInstance(1.0, instancia));
                //Limpio las variables
                parcial.clear();

                      return fin;

                   }
                   catch (Exception e){

                           instancia[30] = instancia[30]/avgRootCode;

                           fin.add(new DenseInstance(1.0, instancia));

                           return fin;
                   }

        }


        /*
         * calcula la clase del evento que se produce el día dia+previsto, que es la fecha evaluada.
         * Parametros:
         * Input:
         * Instances ins: conjunto de datos donde buscar el evento
         * int dia: dia del evento a clasificar
         * int previsto: día donde se va a comprobar si se produjo conflicto o no
         * Output:
         * boolean: true si se produjeron conflictos en el día dia+previsto
         *                              false si no se produjo.
         */

        private static boolean clase(Instances ins, int dia, int previsto) throws Exception{

                   //Si hay algun resultado tras aplicar el filtro es porque
                   //hay eventos peligrosos
                   SubsetByExpression su;
                   String expre;
                   Instances fin;
                   Date date;
                   Calendar cal;
                   SimpleDateFormat originalFormat;

               originalFormat = new SimpleDateFormat("yyyyMMdd");
                      date = originalFormat.parse(Integer.toString(dia));
                      cal = Calendar.getInstance();
                      cal.setTime(date);
                      cal.add(Calendar.DATE, previsto);


                   //El evento evaluado pertenecerá a la clase 1 (conflicto) o 0 (sin conflicto) si el
                   //valor de rootEventCode es o 14 o 18 o 19 o 20

                   expre = "(ATT1 = "+ originalFormat.format(cal.getTime()) +") and (ATT5 > -1) and (ATT4 > 6) and ((ATT2 =
14) or (ATT2 = 18) or (ATT2 = 19) or (ATT2 = 20))";


                   //System.out.print(".");

                   su = new SubsetByExpression();
                   su.setExpression(expre);
                   su.setInputFormat(ins);

                   fin = Filter.useFilter(ins, su);

                   if (!fin.isEmpty()){
                              if (fin.numInstances() == 0){return false;}
                              else{return true;}
                   }
                   else{return false;}
```

```
            }

            /*
             * método para obtener los datos desde el repositorio Hadoop
             * base de datos TFG2015 y tabla gdletdaily
             * parámetros:
             * Input:
             * String pais: pais sobre el que obtener datos
             * int dia:  día base sobre el que establecer la predicción
             * int previo: cantidad de días previos sobre los que obtener las muestras
             * Output:
             * Instances: Dataset con los datos requeridos
             */
            private static Instances instanciaPaisDia(String pais, int dia, int previo) throws Exception{

                        String expre;
                        InstanceQuery query;
                        Instances ins;
                        MathExpression ma;
                        String[] opciones;

                        /*
                         * El conjunto de datos del pais a buscar comienza a partir de la fecha a estimar menos los dias previos
                         * que se han de considerar hasta los 60 dias anteriores para calcular las estadísticas
                         * Subo los dias a 65 para asegurar que existen datos, este valor es puramente empirico.
                         */


                        expre = "Select sqldate,eventrootcode, numarticles, goldsteinscale, avgtone from "+ baseDatos + "." + tabla
+" where (actor1countrycode = \'"+ pais +"\')";
                        expre = expre + " and (isrootevent = 1) order by sqldate";

                        query = new InstanceQuery();
                query.setUsername("");
                query.setPassword("");
                query.setQuery(expre);

                try{


                   PrintStream stderr = System.err; // Save stderr stream.
                       System.setErr(new PrintStream(new OutputStream() {
                        public void write(int b) {
                            //DO NOTHING
                        }
                    }));


                   ins = query.retrieveInstances();


                   System.setErr(stderr);
                   ma = new MathExpression();
                   opciones= new String[5];
                       opciones[0]="-E";
                       opciones[1]="A*(-1)";
                       opciones[2]="-V";
                       opciones[3]="-R";
                       opciones[4]="4,5";
                       ma.setOptions(opciones);
                       ma.setInputFormat(ins);


                                ins = Filter.useFilter(ins, ma);


                                if (guardar){


                                            ArffSaver saver = new ArffSaver();
                                                      saver.setInstances(ins);
                    //saver.
                                                      saver.setFile(new File(ficheroCrudo));
                                                      saver.writeBatch();


                                }


                }
                catch (Exception e){
                   System.err.println(e.toString());
                   ins = null;
                }
                return ins;
                   }

                   /**
                    * @return
                    * Método para poder realizar la predicción sobre un pais y un día en concreto
                    * parámetros:
                    * Input:
                    * String pais: país sobre el que hacer la predicción
                    * Int dia: día en formato YYYYMMDD valor entero, sobre el que hacer la predicción
                    */
                   /*
                    * Método para poder realizar la predicción sobre un pais y un día en concreto
                    * parámetros:
                    * Input:
```

```
         * String pais: país sobre el que hacer la predicción
         * Int dia: día en formato YYYYMMDD valor entero, sobre el que hacer la predicción
         */
        public double prediccion(String pais, int dia) throws Exception{

                  Instances ins, test;
                  Date ahora, date;
                  SimpleDateFormat originalFormat;

                  date = new Date();
                  originalFormat = new SimpleDateFormat("yyyyMMdd");

                  try{
                  date = originalFormat.parse(Integer.toString(dia));


        }catch(Exception e){
                  System.err.println("Formato de fecha incorrecto: " + dia);
                  System.err.println("Uso: "+ TFGPrediccion.class.getName()+" YYYYMMDD pais");
                  System.exit(1);
        }

                  if (!countries.contains((String) pais) ){
                  System.err.println("codigo de pais incorrecto: " + pais);
                  System.err.println("Uso: "+ TFGPrediccion.class.getName()+" YYYYMMDD pais");
                  for(int i=0;i<countries.size();i++){
                      System.err.println(countries.get(i));
                  }
                  System.exit(1);
        }

                  ahora = new Date();
        int diff = (int) (date.getTime() - ahora.getTime())/(1000 * 60 * 60 * 24);


        if (diff > 365){
                  System.err.println("diferencia en días para hacer predicción es excesiva");
                  System.err.println("introduzca una fecha futura < 365 días");
                  System.exit(1);
        }


        //test
        //diff = 10;
        //Logica de la aplicación
        //Obtengo los datos en crudo
                  ins = instanciaPaisDia(pais,dia,diff);
                  if (ins != null){
                            //Formo el conjunto de datos de entrenamiento
                            test = formarDataset(ins,pais,dia,diff);
                                if ((test != null) && (ins != null)){
                                      testear(test, ins, dia, diff);
                                      return prediccion;
                                }else{return -1000;}
                  }
                  else{
                            return -1000;
                  }

        }

public static void main(String[] args) throws Exception{


        String fecha;
        int dia;
        SimpleDateFormat originalFormat;
        Date date;
        double res;


         if (args.length != 2){


                  System.out.println("número de argumentos incorrecto.");
                  System.out.println("Uso: "+ TFGPrediccion.class.getName()+" YYYYMMDD pais");
                  System.exit(1);
        }


        fecha = args[0];
        dia = Integer.parseInt(fecha);
        date = new Date();
        originalFormat = new SimpleDateFormat("yyyyMMdd");

        try{
                  date = originalFormat.parse(Integer.toString(dia));


        }catch(Exception e){
                  System.out.println("Formato de fecha incorrecto: " + fecha);
                  System.out.println("Uso: "+ TFGPrediccion.class.getName()+" YYYYMMDD pais");
                  System.exit(1);
        }


        res=0;
        TFGPrediccion r = new TFGPrediccion();
        r.prediccion(args[1], dia);
```

```
            res = r.getPrediccion();


            if (res == 0){
                        System.out.println(res);
            }else{
                                    System.out.println("no hay resultado");
            }
          System.exit(0);

    }

}
```

ANEXO VI

```
@attribute avgt_media_fecha_previo_1 numeric
@attribute goldstein_media_fecha_previo_1 numeric
@attribute numarticles_media_fecha_previo_1 numeric
@attribute avgt_media_fecha_previo_2 numeric
@attribute goldstein_media_fecha_previo_2 numeric
@attribute numarticles_media_fecha_previo_2 numeric
@attribute avgt_media_fecha_previo_3 numeric
@attribute goldstein_media_fecha_previo_3 numeric
@attribute numarticles_media_fecha_previo_3 numeric
@attribute avgt_media_fecha_previo_4 numeric
@attribute goldstein_media_fecha_previo_4 numeric
@attribute numarticles_media_fecha_previo_4 numeric
@attribute avgt_media_fecha_previo_5 numeric
@attribute goldstein_media_fecha_previo_5 numeric
@attribute numarticles_media_fecha_previo_5 numeric
@attribute avgt_media_fecha_previo_6 numeric
@attribute goldstein_media_fecha_previo_6 numeric
@attribute numarticles_media_fecha_previo_6 numeric
@attribute avgt_media_fecha_previo_7 numeric
@attribute goldstein_media_fecha_previo_7 numeric
@attribute numarticles_media_fecha_previo_7 numeric
@attribute avgt_media_fecha_previo_8 numeric
@attribute goldstein_media_fecha_previo_8 numeric
@attribute numarticles_media_fecha_previo_8 numeric
@attribute avgt_media_fecha_previo_9 numeric
@attribute goldstein_media_fecha_previo_9 numeric
@attribute numarticles_media_fecha_previo_9 numeric
@attribute avgt_media_fecha_previo_10 numeric
@attribute goldstein_media_fecha_previo_10 numeric
@attribute numarticles_media_fecha_previo_10 numeric
@attribute rootcodeavg numeric
@attribute class {1,0}
```

Donde:

```
avgt_media_fecha_previo_1 numeric
avgt_media_fecha_previo_2 numeric
avgt_media_fecha_previo_3 numeric
avgt_media_fecha_previo_4 numeric
avgt_media_fecha_previo_5 numeric
avgt_media_fecha_previo_6 numeric
avgt_media_fecha_previo_7 numeric
avgt_media_fecha_previo_8 numeric
avgt_media_fecha_previo_9 numeric
avgt_media_fecha_previo_10 numeric
```

corresponden a las caracteristicas $A'_c(i)/\mu'_c$ . . . $A'_c(i-9)/\mu'_c$ respectívamente

goldstein_media_fecha_previo_1 numeric
goldstein_media_fecha_previo_2 numeric
goldstein_media_fecha_previo_3 numeric
goldstein_media_fecha_previo_4 numeric
goldstein_media_fecha_previo_5 numeric
goldstein_media_fecha_previo_6 numeric
goldstein_media_fecha_previo_8 numeric
goldstein_media_fecha_previo_9 numeric
goldstein_media_fecha_previo_10 numeric

corresponden a las características $V'_c(i)/\mu'_c$ . . . $V'_c(i-9)/\mu'_c$ respectívamente

numarticles_media_fecha_previo_1 numeric
numarticles_media_fecha_previo_2 numeric
numarticles_media_fecha_previo_3 numeric
numarticles_media_fecha_previo_4 numeric
numarticles_media_fecha_previo_5 numeric
numarticles_media_fecha_previo_6 numeric
numarticles_media_fecha_previo_7 numeric
numarticles_media_fecha_previo_8 numeric
numarticles_media_fecha_previo_9 numeric
numarticles_media_fecha_previo_10 numeric

corresponden a las características $M'_c(i)/\mu'_c$ . . . $M'_c(i-9)/\mu'_c$ respectívamente

class {1,0} será el atributo de evento positivo (1) o negativo (0)

ANEXO VII

Datos predictivos para un día previos para las fechas comprendidas entre el 04/03/2013 y el 26/02/2014 en Ucrania.

```
diasprevios;Porcentaje de
prediccion;truePositiveRate;trueNegativeRate;relativeAbsoluteError;rootRelativeSquaredError;fecha
1;55,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140226
1;43,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140225
1;12,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140224
1;8,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140223
1;9;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140222
1;58,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140221
1;17,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140220
1;63,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140219
1;3,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140218
1;61,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140217
1;7,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140216
1;7,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140215
1;56,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140214
1;8,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140213
1;10,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140212
1;6,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140211
1;4,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140210
1;11,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140209
1;8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140208
1;9;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140207
1;17,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140206
1;3,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140205
1;51,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140204
1;7,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140203
1;8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140202
1;8,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140201
1;4,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140131
1;3,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140130
1;8,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140129
1;8,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140128
1;14,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140127
1;11,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140126
1;14,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140125
1;12,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140124
1;8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140123
1;5,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140122
1;6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140121
1;7,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140120
1;13;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140119
1;15,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140118
1;16,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140117
1;13,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140116
1;9,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140115
1;11,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140114
1;10,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140113
1;17,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140112
1;21;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140111
1;58;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140110
1;20,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140109
1;62,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140108
1;56,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140107
1;49,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140106
1;29;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140105
1;23,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140104
1;62,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140103
1;70,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140102
1;63,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20140101
1;68,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131231
1;73,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131230
1;29,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131229
1;55;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131228
1;71,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131227
1;67,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131226
1;23,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131225
1;56,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131224
1;24,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131223
1;33,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131222
1;70;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131221
1;73,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131220
1;20,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131219
1;19,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131218
1;10;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131217
1;15;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131216
1;19,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131215
1;19,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131214
1;19,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131213
1;12;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131212
```

1;14,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131211
1;15,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131210
1;23,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131209
1;27,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131208
1;66,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131207
1;67,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131206
1;67;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131205
1;25;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131204
1;13,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131203
1;19,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131202
1;68,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131201
1;44,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131130
1;33,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131129
1;70;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131128
1;22,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131127
1;18,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131126
1;16;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131125
1;14,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131124
1;65,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131123
1;16,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131122
1;18,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131121
1;22,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131120
1;20,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131119
1;14,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131118
1;17,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131117
1;75;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131116
1;18,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131115
1;26,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131114
1;22,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131113
1;14,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131112
1;19,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131111
1;18,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131110
1;77,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131109
1;78,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131108
1;15;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131107
1;71,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131106
1;15,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131105
1;12,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131104
1;14,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131103
1;76,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131102
1;79,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131101
1;76;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131031
1;21,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131030
1;77;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131029
1;21;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131028
1;77,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131027
1;81,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131026
1;80;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131025
1;78,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131024
1;21,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131023
1;22,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131022
1;25,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131021
1;67,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131020
1;76,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131019
1;73;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131018
1;65,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131017
1;24;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131016
1;17,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131015
1;15,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131014
1;14,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131013
1;68,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131012
1;17,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131011
1;12,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131010
1;11,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131009
1;7,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131008
1;11,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131007
1;10;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131006
1;67,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131005
1;65,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131004
1;6,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131003
1;61,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131002
1;8,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20131001
1;11,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130930
1;12,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130929
1;11,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130928
1;10;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130927
1;9,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130926
1;67;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130925
1;18,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130924
1;15,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130923
1;71,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130922
1;13,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130921
1;68,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130920
1;11,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130919
1;5,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130918
1;66,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130917
1;17,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130916
1;18,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130915

1;11,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130914
1;7,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130913
1;11,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130912
1;3;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130911
1;4,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130910
1;5,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130909
1;8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130908
1;9,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130907
1;10,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130906
1;9,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130905
1;6,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130904
1;7;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130903
1;8,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130902
1;15,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130901
1;28,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130831
1;60,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130830
1;55,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130829
1;61;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130828
1;17,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130827
1;67;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130826
1;14,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130825
1;17,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130824
1;47,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130823
1;26,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130822
1;21;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130821
1;21,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130820
1;31,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130819
1;62,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130818
1;65,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130817
1;54;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130816
1;22,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130815
1;18,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130814
1;51,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130813
1;17,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130812
1;32,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130811
1;41,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130810
1;23,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130809
1;56,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130808
1;24,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130807
1;21,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130806
1;31,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130805
1;44,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130804
1;72,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130803
1;31,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130802
1;35;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130801
1;41,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130731
1;63,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130730
1;47,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130729
1;77;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130728
1;72,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130727
1;58,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130726
1;79,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130725
1;69,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130724
1;49,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130723
1;78,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130722
1;61;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130721
1;73,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130720
1;61,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130719
1;64;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130718
1;46,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130717
1;26,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130716
1;35;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130715
1;30,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130714
1;60,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130713
1;72,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130712
1;23,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130711
1;64,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130710
1;32,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130709
1;66,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130708
1;33;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130707
1;74,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130706
1;26,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130705
1;67,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130704
1;18,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130703
1;19,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130702
1;22,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130701
1;25,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130630
1;84,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130629
1;16,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130628
1;77,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130627
1;12,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130626
1;82,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130625
1;70,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130624
1;82,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130623
1;21,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130622
1;70,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130621
1;66,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130620
1;70,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130619

1;26;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130618
1;61,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130617
1;67,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130616
1;65,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130615
1;24,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130614
1;16,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130613
1;48,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130612
1;55,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130611
1;30;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130610
1;27,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130609
1;71;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130608
1;72,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130607
1;60,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130606
1;29,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130605
1;27,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130604
1;32;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130603
1;52;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130602
1;43,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130601
1;24,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130531
1;47,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130530
1;53,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130529
1;44,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130528
1;25,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130527
1;56,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130526
1;35,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130525
1;32,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130524
1;55,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130523
1;31,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130522
1;53,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130521
1;31;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130520
1;57,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130519
1;58;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130518
1;60,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130517
1;21,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130516
1;23,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130515
1;13;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130514
1;17,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130513
1;18;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130512
1;52,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130511
1;17,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130510
1;13,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130509
1;13,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130508
1;16,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130507
1;54,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130506
1;30,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130505
1;61,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130504
1;60,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130503
1;15,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130502
1;60,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130501
1;13,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130430
1;12,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130429
1;13,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130428
1;61,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130427
1;11,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130426
1;11,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130425
1;18,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130424
1;18,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130423
1;18,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130422
1;5,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130421
1;11,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130420
1;55,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130419
1;13,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130418
1;15,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130417
1;3,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130416
1;8,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130415
1;66,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130414
1;72,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130413
1;72;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130412
1;15;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130411
1;9,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130410
1;7,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130409
1;6,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130408
1;10,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130407
1;16,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130406
1;73,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130405
1;15,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130404
1;10,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130403
1;5,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130402
1;7,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130401
1;77,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130331
1;12,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130330
1;69,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130329
1;15,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130328
1;16;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130327
1;20;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130326
1;30;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130325
1;15,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130324
1;67,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130323

1;3,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130322
1;12,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130321
1;9,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130320
1;14,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130319
1;13,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130318
1;19,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130317
1;19,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130316
1;14;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130315
1;70,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130314
1;10,8;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130313
1;9,4;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130312
1;13,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130311
1;15,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130310
1;15,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130309
1;19,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130308
1;11,2;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130307
1;16;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130306
1;7,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130305
1;11,6;0,7602739726;0,5784615385;81,3498903746;90,6733571027;20130304

ANEXO VIII

Información obtenida para la predicción sobre el día 28/02/2014 en Ucrania tomadas desde 1 día de antelación hasta 30 días de antelación.

```
diasprevios;pctprediccion;truePositiveRate;trueNegativeRate;relativeAbsoluteError;rootRelativeSquared
Error;fecha
1;80,8532374101;0,5675675676;0,9737609329;58,2465949601;73,0839678017;20140227
2;83,6412470024;0,6315789474;0,9618768328;58,854553163;73,9606512601;20140226
3;71,4633093525;0,6;0,9584569733;58,86896227;75,1712159753;20140225
4;89,6278177458;0,6144578313;0,9550898204;59,5645046468;74,7214843635;20140224
5;85,2302158273;0,630952381;0,960960961;59,2427525802;73,4532492799;20140223
6;91,2167865707;0,6144578313;0,9640718563;58,4555428555;74,8017585884;20140222
7;89,6268585132;0,5487804878;0,952238806;58,2539617867;73,1141785681;20140221
8;94,4129496403;0,5975609756;0,952238806;59,707649359;75,6127123784;20140220
9;86,4287769784;0,5833333333;0,9489489489;59,1821281023;74,8161451378;20140219
10;55,9122302158;0,5581395349;0,9395770393;59,6612336274;76,4787132616;20140218
11;69,4666666667;0,5632183908;0,9424242424;58,5982836957;75,4016149983;20140217
12;75,4561151079;0,5581395349;0,9335347432;60,7950719246;77,5183308204;20140216
13;84,0350119904;0,5764705882;0,9427710843;59,5084140232;75,3114128309;20140215
14;90,0230215827;0,5176470588;0,9126506024;64,3809944868;80,2687437714;20140214
15;85,2412470024;0,5294117647;0,9277108434;64,2406570788;80,5428880146;20140213
16;87,4263788969;0,5301204819;0,9101796407;64,2454884652;79,9456102608;20140212
17;84,2369304556;0,4390243902;0,9014925373;67,4689150178;83,0721086105;20140211
18;76,0618705036;0,4691358025;0,9107142857;65,4741292377;81,2804039631;20140210
19;77,8537170264;0,5802469136;0,9136904762;64,7317937395;78,6880820878;20140209
20;72,8709832134;0,4634146341;0,9164179104;65,090468427;80,2022337255;20140208
21;78,2470023981;0,5;0,9194029851;64,9357066216;79,7398015702;20140207
22;69,6824940048;0,5125;0,9317507418;65,0257511027;79,3786127257;20140206
23;67,0757793765;0,4487179487;0,9321533923;65,9434096319;79,5166611525;20140205
24;64,0805755396;0,4230769231;0,9439528024;65,9942874394;78,8205076538;20140204
25;70,4652278177;0,358974359;0,9380530973;69,4313515633;81,8954213747;20140203
26;65,8738609113;0,3289473684;0,9208211144;71,836999848;85,7281982473;20140202
27;74,6647482014;0,2602739726;0,9302325581;72,0141537812;84,8594523461;20140201
28;66,8834532374;0,2285714286;0,9452449568;70,5836937035;83,3089409746;20140131
29;78,0508393285;0,3428571429;0,9654178674;70,5241186475;83,1170096785;20140130
30;78,0527577938;0,231884058;0,9482758621;72,9525762796;84,5074097089;20140129
```

ANEXO IX

La tarea ejecuta lo siguiente:

```
#!/bin/bash
cd /root/downloads
fecha=$(date +%Y%m%d)
wget http://data.gdeltproject.org/events/$fecha.export.CSV.zip
sleep 120
cd /root
for file in /root/downloads/*
do
        a="${file}"
        b=$(echo $a | cut -d'/' -f4 | cut -d. -f1)
        echo $c
        unzip -p $a >> $c
        sleep 2
        hadoop fs -put $c /user/hue
        sleep 15
        rm -f $c
        /usr/bin/hive -e "LOAD DATA INPATH '/user/hue/$c' INTO TABLE
TFG2015.gdeltdaily"
        sleep 15
done
```

11. Plan de trabajo del proyecto

	Nombre	Inicio	Terminado	Predecesores	Nombres del Recurso
1	**Hito I**	**27/02/15 8:00**	**13/04/15 17:00**		
2	**Definicion del proyecto**	**27/02/15 8:00**	**4/03/15 9:00**		
3	**Definicion de necesidades**	**27/02/15 8:00**	**2/03/15 9:00**		
4	Determinación del problema a resolver	27/02/15 8:00	2/03/15 9:00		
5	Determinación de objetivos	27/02/15 8:00	2/03/15 9:00		
6	**Analisis de riesgos**	**3/03/15 8:00**	**4/03/15 9:00**		
7	Estudio de herramientas similares en el mercado	3/03/15 8:00	4/03/15 9:00		
8	Valoración de herramientas y plataformas a ut...	3/03/15 8:00	4/03/15 9:00		
9	**Elaboracion del plan de proyecto**	**9/03/15 8:00**	**6/04/15 17:00**		
10	Definición de hitos	9/03/15 8:00	10/03/15 9:00		
11	**Definición de actividades y tareas**	**9/03/15 8:00**	**6/04/15 17:00**		
12	**Conocimiento de herramientas a utilizar**	**9/03/15 8:00**	**6/04/15 17:00**		
13	Hortonworks Sandbox 2.2	9/03/15 8:00	6/04/15 17:00		
14	Cloudera 5.3	9/03/15 8:00	6/04/15 17:00		
15	Openssl seguridad de acceso	9/03/15 8:00	10/03/15 9:00		
16	GDELT repositorio de datos	9/03/15 8:00	10/03/15 9:00		
17	CAMEO tasonomia de acceso	9/03/15 8:00	10/03/15 9:00		
18	Distribución de trabajo y recursos necesarios	9/03/15 8:00	10/03/15 9:00		
19	Elaboración de calendario de hitos y actividades	10/03/15 8:00	11/03/15 9:00		
20	Realización de PEC1	9/03/15 8:00	16/03/15 17:00		
21	**Preparación del entorno de desarrollo**	**1/04/15 8:00**	**13/04/15 17:00**		
22	Instalación y configuración de infraestructura	1/04/15 8:00	6/04/15 17:00		
23	Instalación y formación de Hadoop	1/04/15 8:00	6/04/15 17:00		
24	Formación GDELT y CAMEO	1/04/15 8:00	13/04/15 17:00		
25	**Hito 2**	**13/04/15 8:00**	**27/04/15 17:00**		
26	Comparar objetivos iniciales con los actuales	13/04/15 8:00	14/04/15 9:00		
27	Realización de análisis de requisitos	13/04/15 8:00	14/04/15 9:00		
28	Diseño del almacén de datos	13/04/15 8:00	14/04/15 9:00		
29	Preparación de datos para minería de datos	13/04/15 8:00	17/04/15 17:00		
30	Análisis de resultados mediante minería de datos	20/04/15 8:00	27/04/15 17:00		
31	Conclusiones iniciales y retroalimentación	20/04/15 8:00	27/04/15 17:00		
32	Realización de PEC2	16/04/15 7:00	20/04/15 17:00		
33	**Hito 3**	**21/05/15 7:00**	**12/06/15 17:00**		
34	Realización de PEC3	21/05/15 7:00	25/05/15 17:00		
35	**Análisis del modelo predictivo**	**25/05/15 8:00**	**12/06/15 17:00**		
36	Evaluación de características de las muestras	25/05/15 8:00	12/06/15 17:00		
37	Evaluación de la precisión del modelo	25/05/15 8:00	12/06/15 17:00		
38	Desarrollo de la aplicación	25/05/15 8:00	12/06/15 17:00		
39	Conclusiones	25/05/15 8:00	12/06/15 17:00		
40	**Hito 4**	**26/05/15 8:00**	**15/06/15 17:00**		
41	Realización de memoria final	26/05/15 8:00	5/06/15 17:00		
42	Realización de memoria de competencias	5/06/15 8:00	5/06/15 17:00		
43	Producción de video-presentación del proyecto	7/06/15 8:00	15/06/15 17:00		

TFG_BIGDATA_ENFOQUE_ANALITICO

Ilustración 11: Planning de proyecto

12. Diagrama de Gantt del proyecto

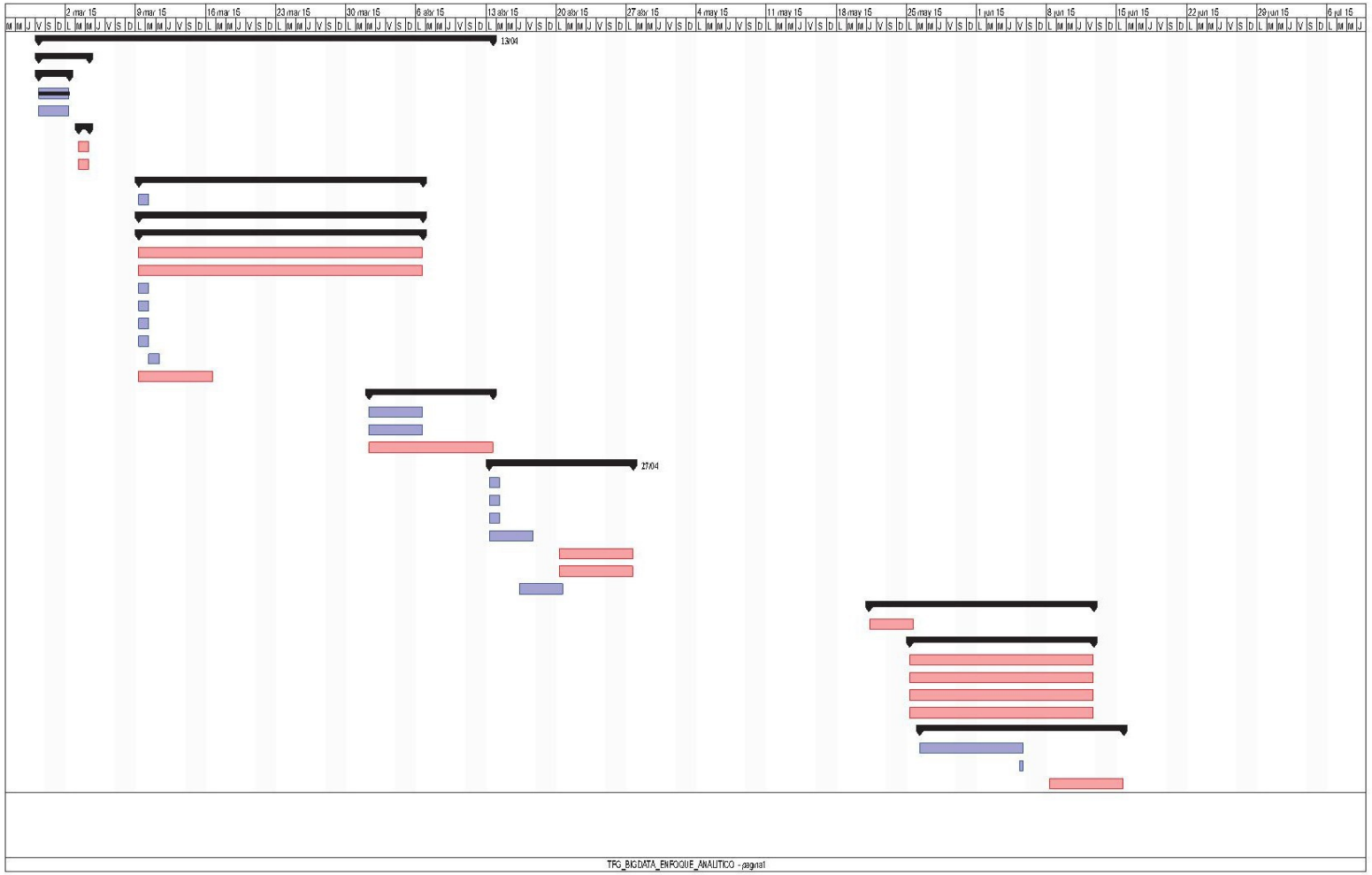


Ilustración 12: Diagrama de Gantt de proyecto