Sistema de apoyo a la toma de decisiones basado en el análisis del sentimiento de titulares económicos

Trabajo de fin de máster

Alumno: José Agustín Vidal González
Universitat Oberta de Catalunya

Índice

1. Introducción.

1.1. Descripción:

Desde los orígenes del análisis bursátil, dos corrientes han prevalecido y dominado los estudios realizados, el análisis técnico y el análisis fundamental. El primero se centra en el estudio de gráficos y su comportamiento, aplicando diferentes métricas -- promedios móviles, osciladores -- directamente sobre el precio del producto con el propósito de inferir la tendencia futura de los precios. El segundo, por otro lado, pretende determinar el valor “fundamental” de un determinado producto únicamente en base a las noticias relacionadas, dejando en segundo plano tendencias o cualquier métrica aplicada directamente a los precios. Cada uno de estos estudios han tenido sus defensores y detractores, aunque el consenso general dice que ambos son complementarios si se desea hacer un análisis serio y riguroso.

Este proyecto es el intento de recopilar lo mejor de cada corriente y elaborar un estudio “híbrido” utilizando técnicas de aprendizaje supervisado que ofrezca una visión adicional a las que estamos acostumbrados. Así pues, en vez de analizar un gráfico con precios cómo haría el análisis técnico, este estudio va a tomar titulares provenientes de sitios web con reputación internacional, los va a clasificar en base a la “experiencia” adquirida del sistema, y en última instancia, los va a representar en un gráfico dependiendo de su sentimiento. Resumiendo, estamos reemplazando el gráfico de precios de un producto por las noticias clasificadas en las que se basa el análisis fundamental, permitiendo al mismo tiempo la utilización de métricas técnicas sobre estos datos “no numéricos”. La intención es que este estudio “híbrido” nos proporcione información sobre el posible comportamiento del mercado durante el anuncio de eventos globales como la llegada de un huracán o la quiebra de una empresa, de un modo casi inmediato. De modo que podamos actuar más rápidamente de lo que ofrecen las herramientas actuales y tomar las medidas adecuadas para obtener un mayor beneficio.

Desde el punto de vista técnico, la idea se basa en la creación de clasificadores que, entrenados con titulares económicos extraídos de importantes medios de comunicación internacionales (Bloomberg, Reuters, Financial Times y Wall Street Journal), sean capaces de determinar con confianza si los titulares de determinados productos -- en el caso concreto de este proyecto nos centraremos en el petróleo -- indican que, o bien los precios están en una tendencia ascendente, o descendente. Estos clasificadores utilizan una técnica denominada "Sentiment Analysis [1]", que a grandes rasgos intenta discernir el sesgo positivo, negativo o neutral de un

determinado texto.

Trabajando sobre esta idea inicial, podemos construir un sistema más complejo que nos permita visualizar con gráficos la tendencia de un producto financiero, y realizar en última instancia recomendaciones de compra o venta basados en los datos analizados. La intención última es la de proporcionar un sistema de apoyo a la toma de decisiones que analice los eventos que ocurren en el mundo a través de los titulares, y determine gracias a su experiencia si el precio va a subir o bajar.

Con el fin de tener a nuestra disposición un número considerable de titulares, se van a desarrollar herramientas de supervisión, extracción y “streaming” con el propósito tanto de construir un “corpus” fiable, como de "leer" titulares publicados en estas webs y almacenarlos en caso de que cumplan unos determinados requisitos. Estos requisitos van a consistir tanto en la inclusión de determinadas palabras en el texto, como en el análisis de la fecha de publicación y su comparación con la variación de precios de una materia prima determinada durante ese periodo de tiempo.

Por ejemplo, explicando una versión reducida del sistema, si quisiéramos entrenar a nuestro clasificador con titulares relacionados con el petróleo, podríamos buscar un rango de fechas donde el precio del petróleo haya subido, bajado o permanecido estable durante un periodo de tiempo prolongado e introducir la fecha de inicio y la fecha de fin en el extractor. Este, una vez haya sido ejecutado, parseará las noticias de estos medios dentro del mencionado rango de fechas, guardando los encabezados encontrados que contengan palabras claves como por ejemplo “Oil”o “Brent”. Más adelante, se supervisarán todos los titulares extraídos, y se les asignará una categoría definitiva de modo manual o automático. Posteriormente, una vez el corpus estuviera formado, podríamos entrenar a los diferentes clasificadores con él, y comprobar su precisión con nuevos titulares publicados en medios online.

1.2. Características principales:

- Análisis inteligente del sentimiento de un titular económico.
- Recomendaciones automáticas de compra y venta de productos financieros a través de la interpretación de eventos globales.
- Sistema basado en técnicas de aprendizaje supervisado (“Machine learning”) que mejora su precisión a través de la “experiencia”.
- Estudio de la evolución de los precios utilizando los puntos fuertes combinados tanto del análisis técnico como del análisis fundamental.
- Extracción automatizada de titulares provenientes de importantes medios de comunicación online.
- Posibilidad de combinarse fácilmente con APIs de terceros y de añadir nuevas métricas de análisis.
- Alta extensibilidad a la hora de ampliar la lista de productos financieros soportados.

1.3. Aplicaciones:

Las aplicaciones de este sistema son numerosas. Por ejemplo, este sistema podría disparar una orden de compra automática de un producto financiero cuando se lleven más de 3 días seguidos con noticias positivas en los diferentes medios de comunicación (*estrategias de entrada en inversiones*). Enviar un email con advertencias de riesgo a un grupo de personas cada vez que un cierto número de titulares "bajistas" aparezcan en un determinado medio (*estrategia de salida en inversiones*). O servir como "segunda opinión" a un broker antes de realizar una operación en el mercado, a través de los gráficos generados y las recomendaciones sugeridas (*apoyo a la toma de decisiones*).

A su vez, este sistema es especialmente idóneo para reaccionar rápidamente ante eventos globales que pueden afectar a la tendencia de precios de un producto, ya que va a analizar cientos de titulares en tiempo real y podrá detectar si una noticia afecta o no a la evolución de los precios del mismo.

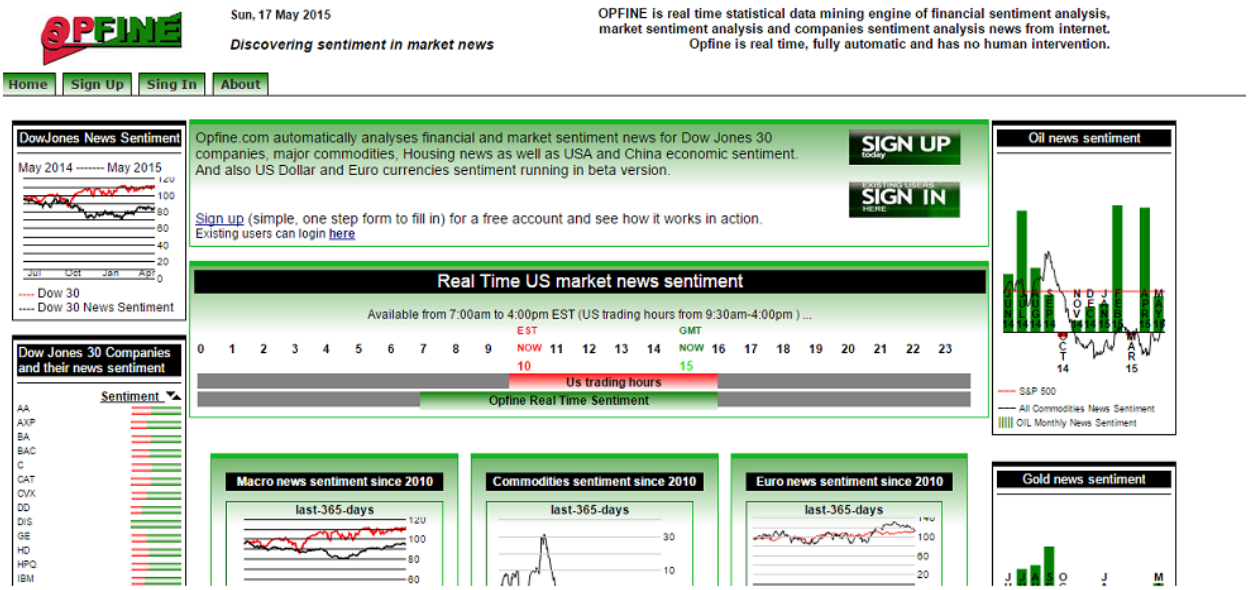
Al mismo tiempo, este sistema puede integrarse con APIs de otras entidades que proporcionen datos precisos de inventarios o historiales de operaciones, con el objetivo de generar órdenes de compra y venta automáticas dependiendo del comportamiento de los clientes (*toma de decisiones de automáticas en base a métricas*)

Por último, este sistema puede adaptarse a un rango de productos o aplicaciones mucho más amplio con un esfuerzo mínimo. Esto daría pie a sistemas más complejos en los que se pudieran observar la evolución de noticias relacionadas con dos productos al mismo tiempo, como por ejemplo, el oro y la plata, y realizar decisiones en base al comportamiento sumado de ambos elementos *(análisis combinado de productos financieros*).

1.4. Sistemas similares existentes en el mercado:

Actualmente existen muchos sistemas que clasifican en base al "sentimiento" de un determinado texto, y son utilizados en ámbitos tan diversos como los "search engines", "recommendation systems" o el análisis de feedbacks de usuarios, algunos ejemplos pueden ser SAS Sentiment Analysis [2] o Weotta [3].

Por otro lado, en lo referente al análisis bursátil, existen actualmente herramientas online para el análisis del sentimiento del mercado o de un producto en concreto como el petróleo. Opfine [4] es un ejemplo de lo que acabamos de mencionar.



Como vemos es una herramienta con muchos gráficos y con información destacada en primera página sobre el petróleo o el oro. Por otro lado, y en concordancia con lo que acabamos de decir, el problema principal de esta herramienta

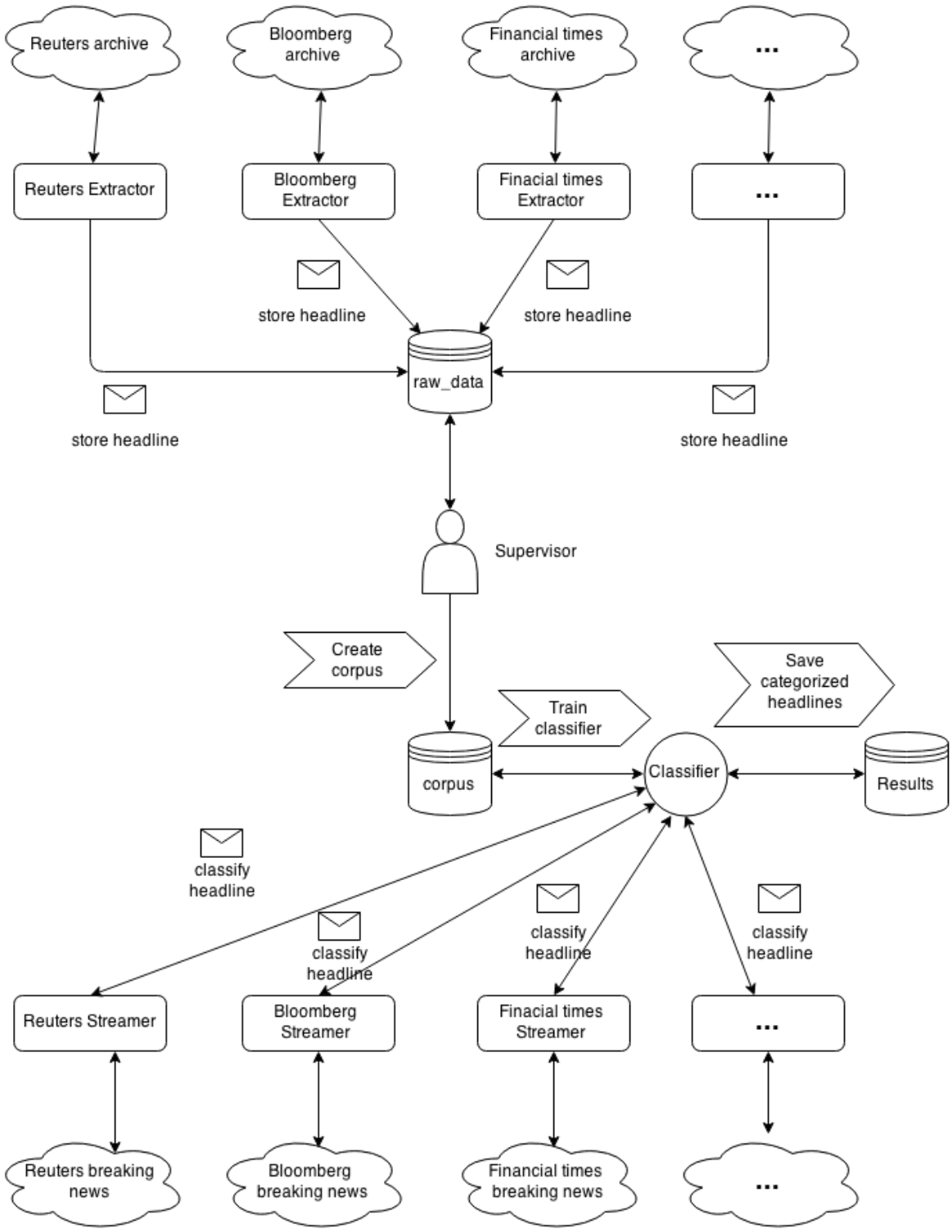
es su generalidad y falta de detalle. Proporciona al visitante un historial que empieza en 2010, lo cual puede ser un poco confuso. Al mismo tiempo, la información que se ofrece es poco específica, mostrando un dato por producto y mes y sin posibilidad de conocer ejemplos de noticias que han llevado al sistema a construir tales gráficos, ni de qué proveedores se han extraído. No sabemos si la información recopilada por la página es fiable ya que no tenemos acceso a ella.

Por último pero no menos importante, el sistema muestra únicamente datos en gráficos. No realiza ningún tipo de predicción o recomendación de compra que ayude a los usuarios a realizar sus inversiones.Tampoco muestra métricas técnicas aplicadas a los gráficos, ni es posible establecer alertas en base al comportamiento de los precios.

Como vemos, existen productos de características similares en el mercado, pero como se ha mostrado, carecen de la flexibilidad necesaria para poder ser utilizados como sistemas inteligentes de apoyo a la toma de decisiones.

2. Diseño del Sistema

2.1. Esquema general:



2.2. Proveedores de noticias:

Los proveedores de noticias son medios de comunicación online utilizados por los extractores y streamers con el fin de obtener noticias en inglés que podrán ser supervisadas para formar parte del conjunto de entrenamiento del clasificador, o clasificadas y analizadas posteriormente por éste para analizar la tendencia actual de un producto.

Actualmente estos son los proveedores de información integrados:

- **Reuters** (www.reuters.com): Es una importante agencia de noticias internacionales con sede en el Reino Unido que fue creada inicialmente en el año 1851. Con el paso de los años esta agencia ha construido su reputación en la mayor parte del mundo, operando actualmente en más de 200 países.
- **Bloomberg** (www.bloomberg.com): Es la mayor agencia de noticias económicas del mundo, y aquella con mayor prestigio internacional. Fue fundada por Michael Bloomberg en 1981 y actualmente aparte de elaborar noticias periodísticas también tiene gran parte de su negocio en la fabricación de terminales de inversiones en tiempo real para brokers.
- **Financial Times** (www.ft.com): Es una agencia de noticias británicas con especial énfasis en noticias económicas y de negocios. Fue fundada en 1888 por James Sheridan y Horatio Bottomley y ha crecido hasta tener una media de 2.2 millones de lectores diarios.
- **Wall Street Journal** (www.wsj.com): Es un periódico de noticias estadounidense con especial énfasis en noticias económicas y de negocios. Es el periódico con mayor tirada en los EEUU, con 2.4 millones, y actualmente tiene unos 900.000 suscriptores a su servicio online.
- **Yahoo** (www.yahoo.com): Yahoo es una multinacional americana con diversas ramas de negocio. La sección económica del buscador es conocida por contener un gran listado con las noticias más importantes publicadas por los diferentes brokers.
- **Peakoil** (www.peakoil.com): Peakoil es una página mucho menos conocida que las anteriores. Incluso podríamos denominarla como "amateur". A pesar de esto, incluye multitud de noticias de calidad, y una perspectiva diferente a los sitios anteriores.

2.3. Clasificadores utilizados:

En el proyecto se han utilizado 4 clasificadores diferentes:

- **Naive Bayes** [5]**:** Usa el teorema de Bayes para predecir la probabilidad que un “feature set” determinado pertenezca a una cierta clase. Se ha utilizado el clasificador de Naive Bayes implementado en la librería NTLK. La razón de su elección, es que es uno de los clasificadores más utilizados en la clasificación del lenguaje natural, uno de los que mejores resultados obtiene, y uno de los que más recursos disponibles tiene en internet.

- **Max Entropy** [6]**:** También conocido como *conditional exponential classifier*, convierte “feature sets” ya etiquetados en vectores utilizando codificación. Este vector es despues usado para calcular los pesos de cada “feature” que puede ser combinada, para determinar cuál será la clase más probable para un cierto “feature set”**.** Este clasificador es otro “clásico” en lo que respecta a la clasificación del lenguaje natural, por lo que he decidido incluirlo en el estudio**.** Se ha utilizado el clasificador implementado en la librería NLTK aunque lo he utilizado en conjunción con el algoritmo “Megan” [7], que mejora notablemente los resultados obtenidos .

- **Linear SVC** [8]: Los clasificadores de vectores de soporte, intentan construir un hiperplano que maximiza la distancia entre varias clases. Linear SVC, o Linear Support Vector Classification, es una implementación parecida al SVC pero utilizando un kernel lineal, es decir, intenta dividir las clases con una recta. He decidido incluir clasificadores de vectores de soporte ya que son bastante efectivos cuando existe un espacio dimensional elevado. Se ha utilizado el clasificador lineal implementado en la librería Scikit-learn.

- **Nu SVC** [9]**:** Es similar al clasificador SVC básico, pero usa un parámetro para controlar el número de vectores de soporte. Al igual que en el caso anterior, se ha utilizado el clasificador implementado en la librería Scikit-learn.

2.4. Estructura del sistema:

El sistema general se divide a su vez en cinco subsistemas especializados en determinadas áreas:

1. **Subsistema de extracción:** Este subsistema engloba todas las operaciones relacionadas con la extracción de noticias de los diferentes proveedores y su posterior almacenamiento. Estas operaciones están a cargo de los denominados "extractores". Existirá un extractor por cada proveedor, configurado especialmente para adaptarse a la estructura de su página de noticias antiguas. Este subsistema va a ser el primero en ser utilizado, ya que va a generar los primeros datos sin procesar que posteriormente serán supervisados en las etapas posteriores.

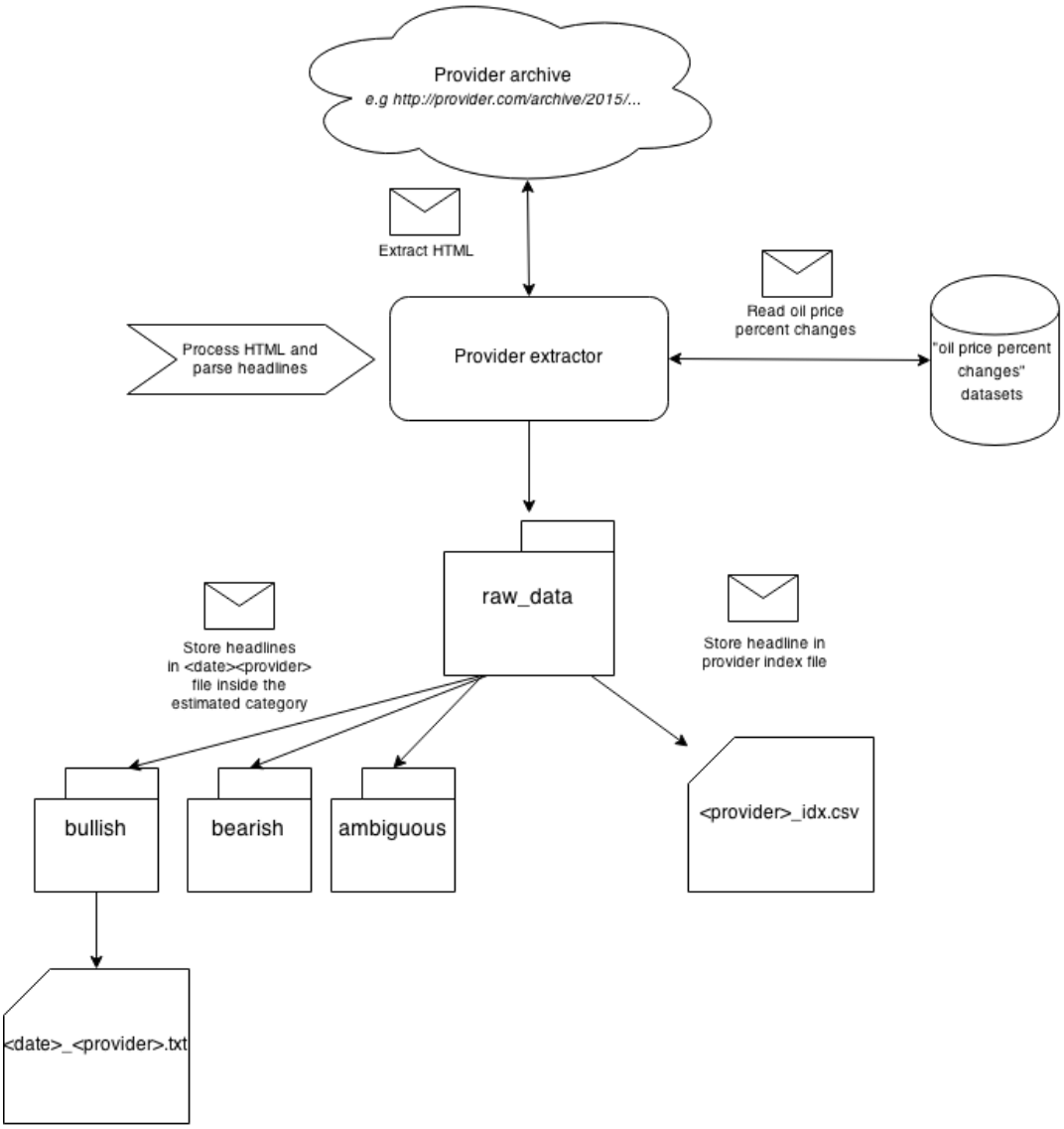
2. **Subsistema de supervisión:** Este componente se encarga de la supervisión de las noticias sin procesar. Una vez supervisadas, las noticias formarán parte del corpus y estarán listas para utilizarse en el entrenamiento de los clasificadores.

3. **Subsistema de entrenamiento:** Este subsistema engloba el entrenamiento de los clasificadores a través del corpus creado en el componente anterior. La ejecución de este componente devolverá un conjunto de clasificadores entrenados y listos para ser utilizados.

4. **Subsistema de clasificación:** Este subsistema se encarga de la lectura de nuevas noticias de los diferentes brokers y su posterior clasificación utilizando los clasificadores. Estas operaciones están a cargo de los denominados "streamers". Existirá un streamer por cada proveedor, configurado para extraer las últimas noticias de su página. La ejecución de este componente devolverá como resultado archivos con las nuevas noticias clasificadas.

5. **Subsistema de visualización:** Este componente se encarga de visualizar las noticias clasificadas dentro de un periodo de tiempo determinado realizando gráficos comparativos con estos datos. Al mismo tiempo, se realizarán recomendaciones de compra y venta basándonos en métricas simples y comparaciones de resultados.

2.4.1 Subsistema de extracción:



Como podemos apreciar, la tarea principal de este subsistema es simple a primera vista, extraer el HTML, procesarlo y parsear las noticias a partir de él, y posteriormente guardarlas tanto en el index del proveedor, como en un archivo .csv del proveedor que guarda las noticias relevantes de ese día.

Por otro lado, conviene explicar algunos detalles que darán una visión más precisa y completa del procedimiento, como por ejemplo cómo se seleccionan las noticias, qué fechas se tienen en cuenta, o cómo se calcula la categoría inicial de las mismas.

Así pues, el extractor para un proveedor determinado se ejecuta proporcionando tanto la fecha de inicio como la fecha final para el que el proveedor buscará noticias. Por ejemplo, siendo el primer argumento la fecha de inicio, y el segundo la fecha final:

./proveedor_extractor.py "2014-05-01" "2014-07-01"

Extraerá en primer lugar una página HTML del proveedor entre las dos fechas proporcionadas, y calculará automáticamente la tendencia de los precios del petróleo entre dos rangos de fechas diferente. Para obtener estos datos, vamos a utilizar dos datasets "*brent-1987-2015-percent-change.json*" y "*wti-1986-2015-percent-change.json*" descargados de Quandl [10], que como su nombre indica van proporcionar los porcentajes de cambios de precios para cada día. El cálculo de tendencia se hace en base al porcentaje positivo o negativo de cambio del precio del petróleo de lunes a viernes, dependiendo del rango de fechas introducido. Si este cambio es positivo, la tendencia de precios para ese rango de fechas se definirá como "bullish", si es negativo, como "bearish", y si no está bien definido, como "ambiguous".

Así pues, el algoritmo introducido establece que únicamente se cogerán noticias del proveedor para ese día si al menos la tendencia de precios para los dos rangos de fechas introducidos es la misma. Es decir, si la tendencia de precios para el primer rango es "bullish", y para el segundo es "bearish", ese día no será tenido en cuenta y se pasará al siguiente. La razón de este comportamiento es que se intenta por encima de todo encontrar noticias "relevantes" que puedan indicar un cambio brusco de los precios marcando una tendencia. Las noticias que provienen de un rango de precios indefinido tienden a ser "menos claras" por lo que he decidido descartarlas utilizando el método anterior.

Una vez la tendencia esté definida, se procesará el HTML en busca de noticias relevantes. Una noticia relevante por otro lado, será aquella que contenga ciertas palabras de "interés", en nuestro caso ["fracking", "hurricane", "opec", "oil", "surplus", "glut"].

Por último, una vez se hayan seleccionado las noticias, están serán guardadas junto a la categoría de los precios calculada anteriormente, la fecha, y su url, en un archivo "<provider>_idx.csv" dentro de la carpeta "raw_data".

Al mismo tiempo, se generará otro archivo con la fecha de publicación de la noticia en el encabezado ("<date>_<provider>.txt"), donde se publicarán todas las

noticias seleccionadas del día para ese proveedor, y que será utilizado cuando posteriormente se supervisen las noticias.

En lo que respecta a esta categoría, no está de más recordar que esta categoría no va a ser la definitiva, durante la fase de supervisión la categoría será confirmada o modificada.

Así pues, como hemos dicho, de la ejecución de este subsistema, se generan dos tipos de archivos:

- **<provider>_idx.csv:** Es un archivo índice con todas las noticias extraídas de ese proveedor. En caso de que el archivo ya exista, las nuevas noticias se añadirán al mismo. Tiene como cometido facilitar la búsqueda de noticias anteriormente almacenadas cuando se añada alguna noticia nueva, para evitar repeticiones. Va a tener la siguiente estructura interna, [“category”, “date”, “url”, “headline”]. Un ejemplo puede ser:

  bullish,2008-12-11,http://www.businessweek.com/news/2008-12-11/opec-cut-may-push-oil-to-80-a-barrel-lukoil-says-update1,"OPEC Cut May Push Oil to $80 a Barrel, Lukoil Says (Update1)"

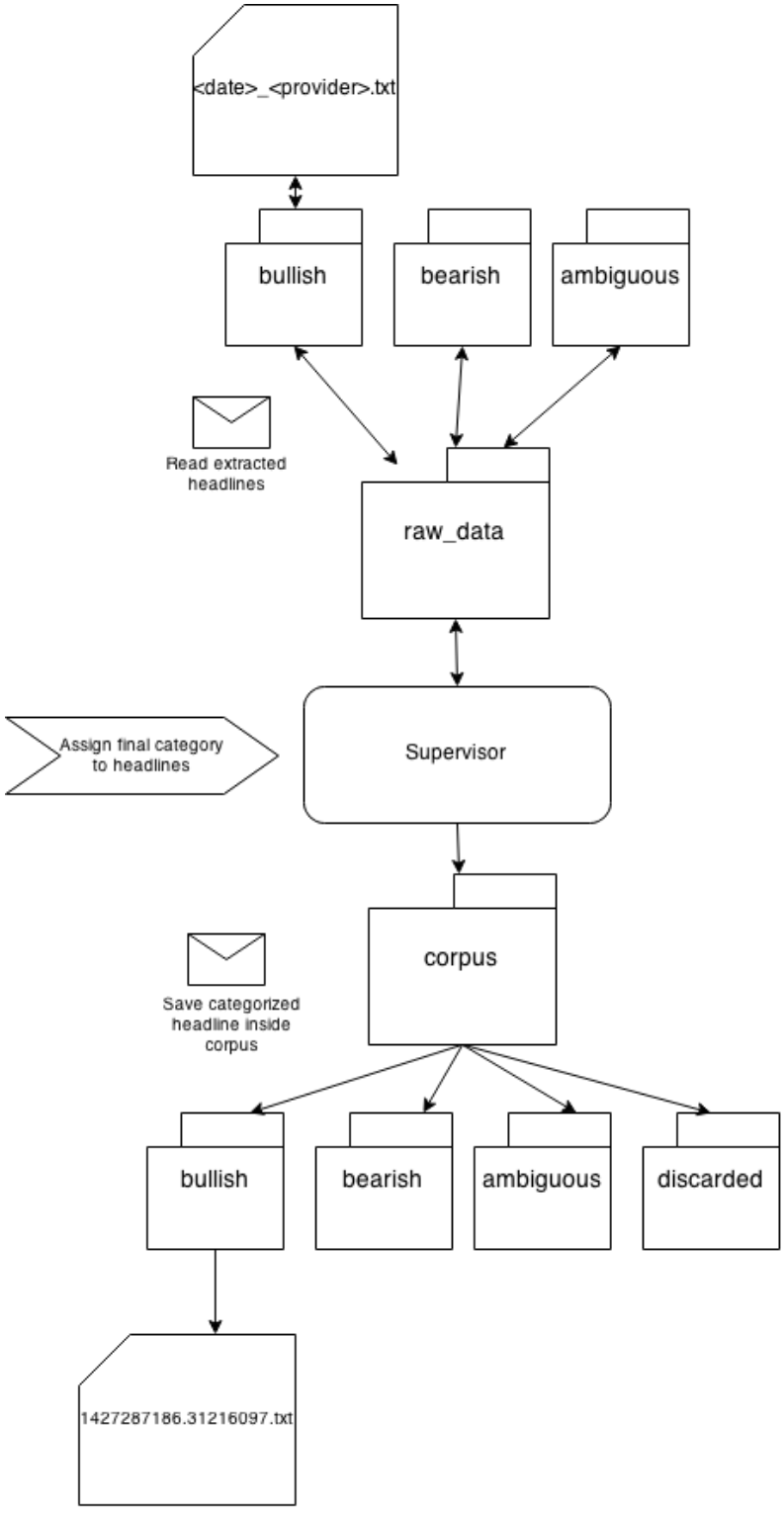
- **<date>_provider.txt:** Estos datos van a contener los titulares resultantes de ejecutar un extractor de un broker determinado. Van a contener la fecha y el broker en su nombre (e.g 2015-03-23_bloomberg.txt). Este archivo será creado dentro una carpeta con el nombre de la categoría calculada para esas noticias, es decir, “bullish”, “bearish” o “ambiguous”.

  En su interior estos archivos van a contener una o múltiples líneas dependiendo del número de titulares seleccionados en ese día y ese broker. Por ejemplo:

  Russia's Daily Crude Oil Shipments to Rise 2 Percent Next Month
  Mboweni Says Rising Oil Prices Are `Major Concern'

  Estos ficheros van a ser utilizados y leídos por el “supervisor” para ser correctamente categorizados y utilizados posteriormente como parte del corpus.

2.4.2 Subsistema de supervisión:



La misión de este subsistema, como dijimos anteriormente, es la de supervisar, es decir, categorizar de un modo definitivo las noticias extraídas y almacenadas por los extractores en la carpeta "raw_data". Si recordamos correctamente lo dicho en el subsistema anterior, en esta carpeta se encuentran las noticias directamente parseadas y categorizadas automáticamente en base a la tendencia de los precios. Esta tendencia, como dijimos, no es definitiva, así pues durante la supervisión deberemos evaluar si la categoría asignada anteriormente es correcta o es necesario modificarla.

Actualmente se han definido 3 categorías diferentes: **Bullish, Bearish** y **Ambiguous**. Dentro del argot del mundo financiero, las dos primeras categorías indican que un cierto producto está en tendencia alcista y bajista respectivamente. La tercera categoría, como su nombre da a entender, indica que la tendencia no se puede identificar con claridad. Así pues, relacionando estos tres conceptos con el proyecto que nos ocupa, una noticia clasificada como "bullish" sugerirá que un determinado producto está en tendencia alcista actualmente o va a estarlo en el corto plazo, una noticia clasificada como "bearish" sugerirá que está en tendencia bajista actualmente o va a estarlo en el corto plazo, y por último, una noticia clasificada como "ambiguous" será demasiado ambigua como para sugerir cualquier tendencia a corto plazo.

La supervisión se puede hacer de un modo "automático" o "manual". El script encargado de realizar ambos modos se denomina "**supervise_raw_data.py**" y en base a los argumentos proporcionados ejecutará una modalidad u otra. Por ejemplo, si quisiéramos ejecutar la supervisión de un modo automático, utilizaríamos como primer argumento “auto_mode=True”. La supervisión automática asume que las categorías asignadas en un primer momento por el extractor son correctas, y por lo tanto las confirma nuevamente durante la supervisión. Para realizar la supervisión manual tendríamos que ejecutar el script sin ningún argumento. De hacerlo, obtendríamos una pregunta por cada noticia a supervisar, como por ejemplo:

[V] Manual mode
[I] Please classify: (bu : bullish, a: ambiguous, be: bearish, d: discard)
** Default: bullish*
---> GAO report on peak oil to be released

Dependiendo de la opción que elijamos, la noticia se almacenará en una categoría u otra del corpus. Una excepción es la categoría “discarded”, que como su nombre indica contendrá o bién noticias duplicadas o irrelevantes, y por lo tanto no será utilizada en posteriores operaciones.

Aparte de lo anteriormente mencionado, el script de supervisión es capaz de realizar operaciones secundarias como detectar duplicidades o revisar el corpus en base a las palabras más significativas. Todas estas operaciones extras están descritas con detalle en el README file que acompaña al código.

En lo que respecta a las noticias que actualmente componen el corpus, durante su supervisión manual se han seleccionado como alcistas noticias que indican claramente que el precio ha subido y como bajistas noticias que indican claramente que el precio ha bajado. Por otro lado, se han tomado algunas decisiones iniciales en la clasificación con respecto a las noticias "dudosas" o de complicada categorización al tratarse de noticias no directamente relacionadas -- con el petróleo en el caso que nos ocupa--, pero que pueden afectar a su tendencia directamente debido a su importancia. A continuación se especifica qué tipo de noticias alternativas han sido clasificadas en cada categoría:

Categoría "Bullish"

- Daños en infraestructuras ocasionados por un desastre natural.
- Caída en la exportación de petróleo de un importante país exportador.
- Reducción del inventario de petróleo de un país con gran consumo de petróleo.
- Situación de guerra o embargo en un país estratégico.
- Miedo a las restricciones de distribución de petróleo.
- Prohibición a las técnicas de exploración alternativas (e.g fracking)
- Demanda de petróleo más grande que la esperada de un país con gran consumo de petróleo.

Categoría "Bearish"

- Situación de excedente (surplus) en la extracción/consumo de petróleo.
- Incremento en la exportación de petróleo en un importante país exportador.
- Incremento del inventario de petróleo de un país con gran consumo de petróleo.
- Descubrimiento de importantes nuevas reservas de petróleo.
- Firma de tratados de paz entre naciones exportadoras.
- Retirada de sanciones a la exportación de petróleo concernientes a grandes países exportadores.
- Demanda de petróleo más débil que la esperada de un país con gran consumo de petróleo.

Categoría "Ambiguous"

- Opiniones individuales de periodistas.
- Declaraciones aisladas de países.
- Preguntas realizadas por columnistas.
- Noticias no relacionadas con el petróleo.

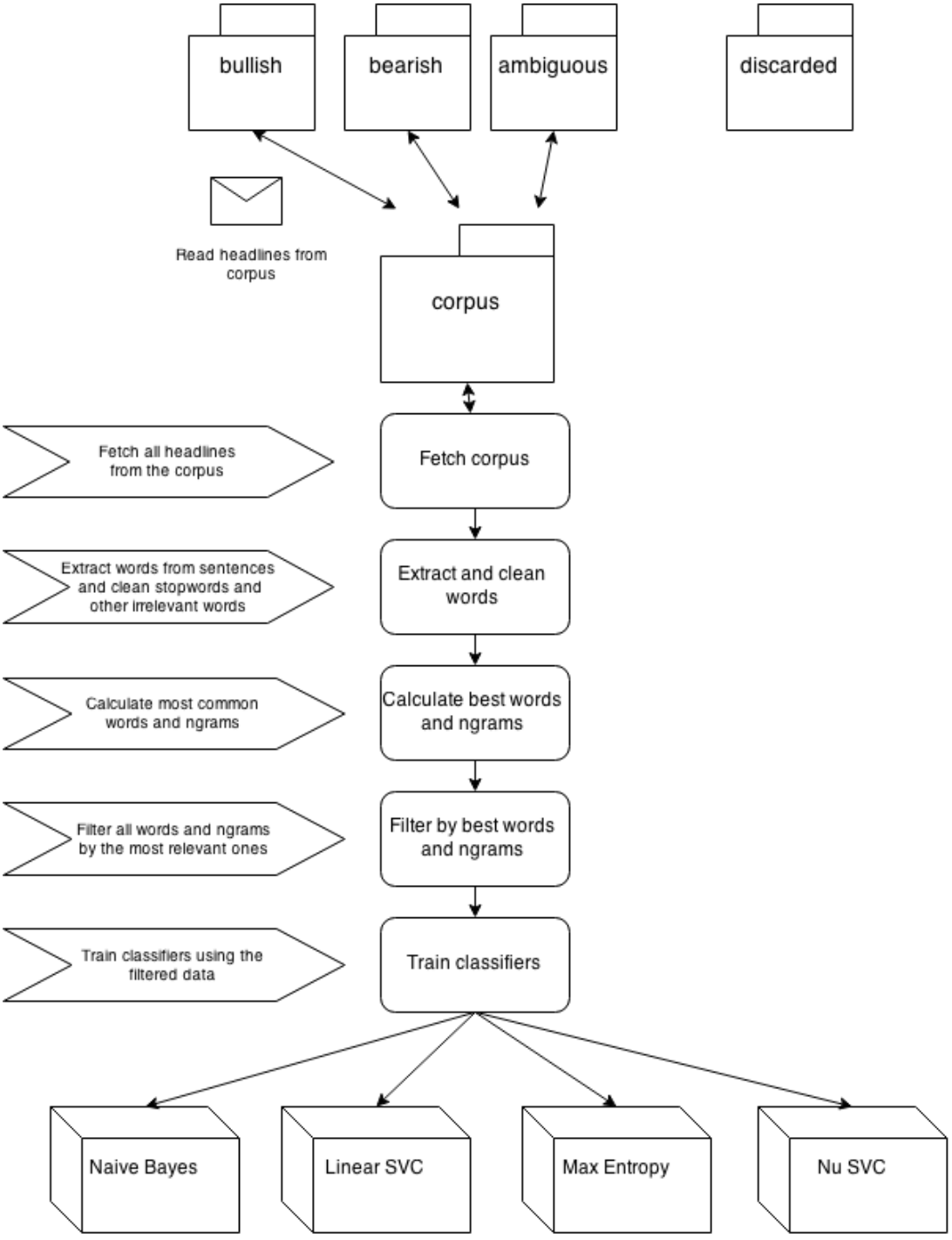
De la ejecución de este subsistema, se genera un tipo de archivo:

- **<timestamp>.txt:** Este archivo, a diferencia de los anteriores, va a utilizar únicamente un “timestamp” como nombre. Este “timestamp” va a corresponder al tiempo en el que fueron supervisados. Por ejemplo:

  1427287186.31216097.txt

  Otra diferencia es que sólo va a haber un titular en cada archivo, por lo que es más granular que en el caso anterior. Estos ficheros van a ser utilizados directamente por el clasificador en su entrenamiento.

2.4.3 Subsistema de entrenamiento:



Este subsistema se va a centrar en el entrenamiento de clasificadores en base a las noticias ya categorizadas que componen el corpus. Para ello se van a preparar y limpiar los datos de modo que el entrenamiento sea lo más efectivo posible. El script que realiza todo el proceso se denomina "**train_classifiers.py**". A continuación se van a detallar cada uno de las operaciones realizadas en el pre-procesamiento de los datos y el posterior entrenamiento de clasificadores:

1. Se extraen todos los nombres de archivos de noticias "txt" que componen cada categoría del corpus, hasta que se alcance el número máximo de elementos del corpus por categoría, que actualmente está establecido en 1050.

2. Se utiliza la función "random" para mezclar todos estos nombres de archivos por categoría y conseguir una representación lo más uniforme posible.

3. Se establecen los números máximos de elementos que conformarán el grupo de las palabras y ngrams más informativos (“best words and ngrams”). Actualmente, y después de multitud de pruebas para conseguir la configuración con mejores resultados, este número se ha establecido en 2000 para tanto las “palabras simples”, “bigrams” y “trigrams”, y 666 para “fourgrams”.

4. Se seleccionan las palabras, bigrams, trigrams y fourgrams más informativos en base a las noticias seleccionadas en el punto 1. El proceso se basa en el cálculo de la frecuencia de cada elemento dentro de una categoría determinada, y su comparación con el resto de categorías. Es decir, una palabra que aparezca únicamente en noticias pertenecientes a una categoría determinada, será muy informativa, ya que nos va a indicar una posible correlación. Por otro lado, una palabra que aparezca en todas las categorías a igual frecuencia no va a ser informativa, ya que no nos da ningún tipo de “pista” en la clasificación.

5. Utilizando los grupos de mejores palabras y ngrams que acabamos de calcular, se van a filtrar las palabras que contiene cada noticia. Durante la extracción se van a filtrar palabras denominadas “stopwords”, es decir, aquellas sin un significado semántico útil (e.g “to”, “from”, ...), dígitos (e.g 1995), palabras con longitud igual a 1 (exceptuando “?”) y por último, palabras dentro de una lista negra, que actualmente contiene únicamente

["..."]. Por último, y en lo referente a las frases, se van a "cortar" el principio o final de las mismas en caso que contengan ":" o "-". La razón es que por norma general el contenido que viene antes o después de estos caracteres, en caso de que se sitúen al principio o al final, son nombres de personas o nombres de compañías sin valor semántico.

6. A las palabras filtradas, se las va a convertir en minúscula, y aplicar posteriormente el Stemmer de Porter para eliminar las terminaciones y dejar únicamente la raíz de las palabras.

7. Una vez las palabras hayan sido seleccionadas, se va a utilizar la técnica "bag of words" para almacenar los features sets. E.g ("surg", True) o ((“oil”, “plung”), True). Cada una de estas tuplas va a estar contenidas en un diccionario cuya clave va a ser la categoría de las noticias que contenían tales palabras (“bullish”, “bearish” o “ambiguous”).

8. Posteriormente, y una vez tengamos los feature sets, se realizarán cinco iteraciones de todas las operaciones que vienen a continuación, de modo que los resultados sean suficientemente representativos.

9. Se seleccionan los grupos de entrenamiento y de test. El grupo de entrenamiento va a ser 3/4 partes, y el grupo de test, 1/4 parte del total.

10. Se utiliza la función "random" nuevamente para mezclar todos los features sets de cada grupo.

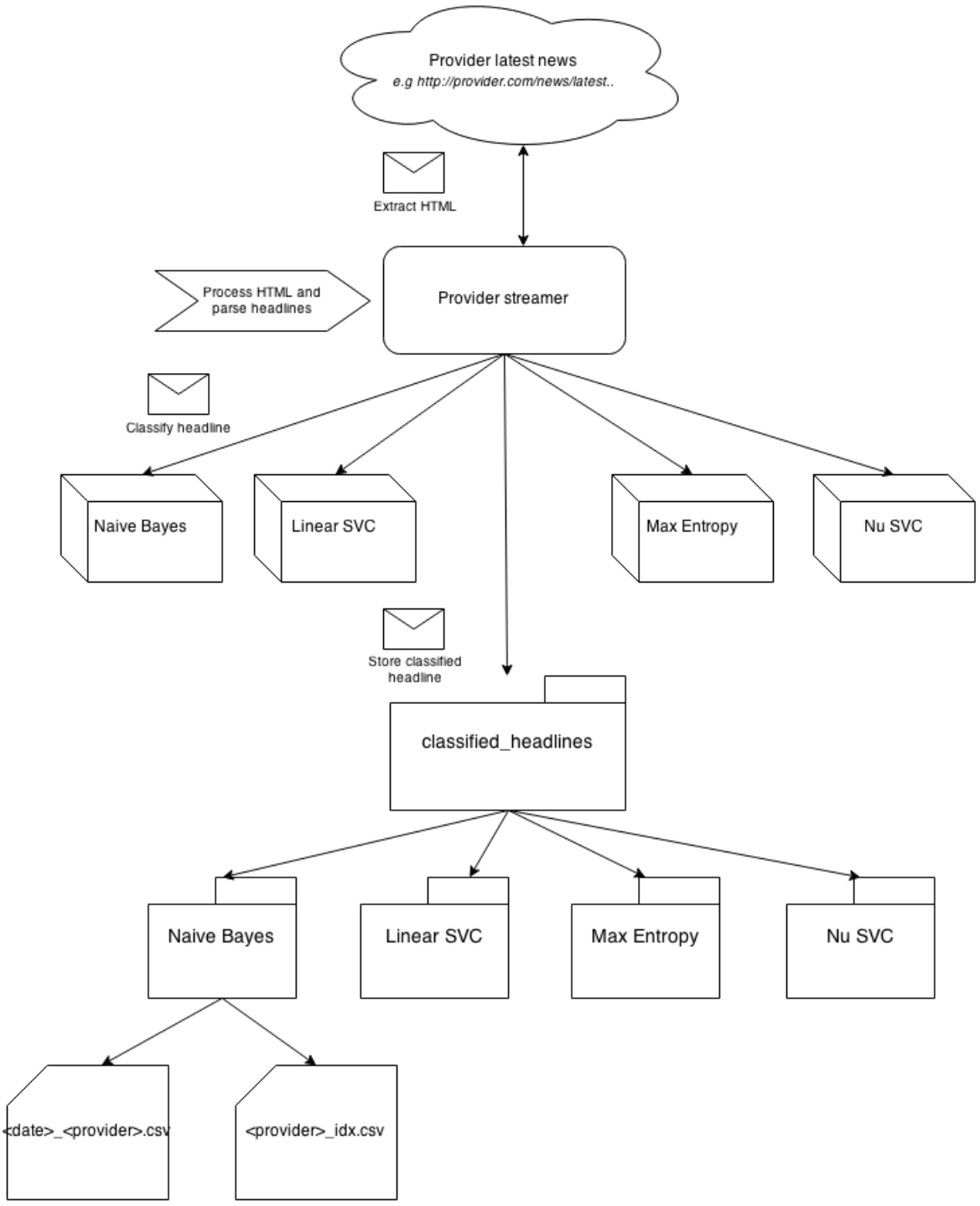
11. Se entrena el clasificador correspondiente (“Naive Bayes”, “Linear SVC”, “Max Entropy” o “Nu SVC”) con el grupo de entrenamiento.

12. Una vez entrenado, se hacen tests de precisión y recall para cada categoría utilizando el conjunto test, además del intervalo de confianza del 95% para la medición general.

13. Cuando las 5 iteraciones hayan concluido, se realizará la media de cada una de las mediciones anteriores y se presentarán los resultados obtenidos.

14. Los clasificadores entrenados se guardarán en la carpeta “classifiers” para futuras clasificaciones.

2.4.4 Subsistema de clasificación:



Una vez tengamos los clasificadores entrenados, en siguiente paso será utilizarlos para clasificar las últimas noticias de los proveedores directamente extraídas de internet. Ese será el principal cometido del subsistema de clasificación.

Así pues, el principal componente de este subsistema será el “Streamer”, que será el encargado de extraer el HTML de las últimas noticias publicadas en cada web, parsearlas, y clasificarlas utilizando alguno de los clasificadores entrenados anteriormente. Durante la clasificación, se van a utilizar todos los clasificadores existentes, de modo que se puedan comparar los resultados posteriormente, y ver qué clasificador arroja mejores resultados.

A diferencia de los “extractores”, los “streamers” no buscan en el archivo de noticias antiguas de los medios, sino en las últimas noticias publicadas. La idea es que los streamers se ejecuten periódicamente (e.g cada hora, cada 3 horas, etc...), de modo que podamos conocer lo antes posible los cambios de opiniones relacionados con el petróleo, y así brindar una información más valiosa al usuario.

Esto último es muy importante, y es lo que convierte a este subsistema en uno de los más críticos. La importancia de tener unos clasificadores bien entrenados, se iguala con la necesidad de disponer de información de última hora de calidad que poder clasificar y transmitir a los usuarios.

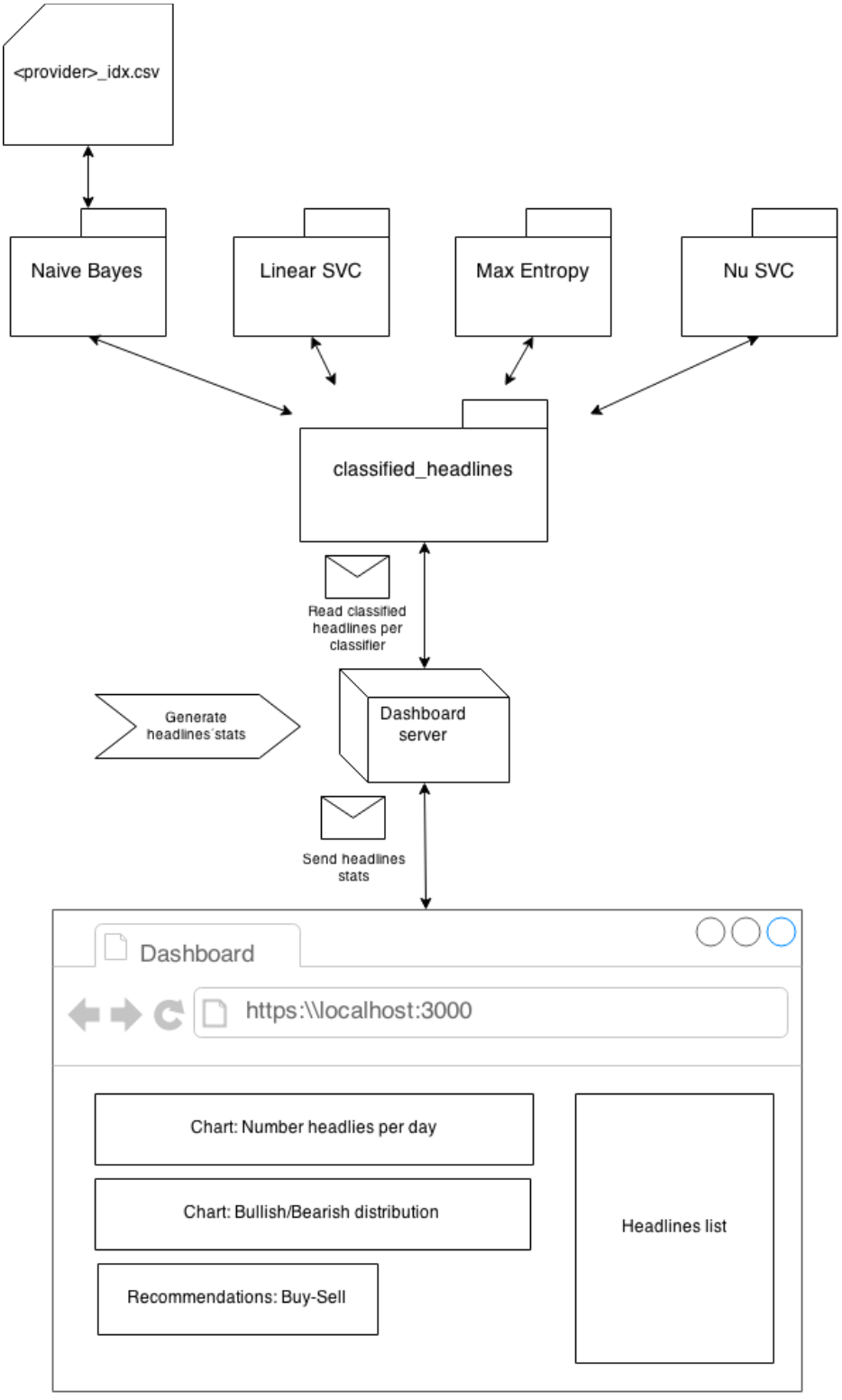
De la ejecución de este subsistema, se generan dos tipos de archivo por clasificador:

- **<provider>_idx.csv:** Al igual que en el caso de los extractores, se va a generar un archivo índice con todas las noticias clasificadas por ese clasificador. En caso de que el archivo ya exista, las nuevas noticias se añadirán al mismo. Este archivo va a ser utilizado más adelante por el dashboard para extraer las noticias del mismo y así mostrar los diferentes gráficos. Va a tener la siguiente estructura interna, [“category”, “date”, “url”, “headline”]. Un ejemplo puede ser:

  bullish,2008-12-11,http://www.businessweek.com/news/2008-12-11/opec-cut-may-push-oil-to-80-a-barrel-lukoil-says-update1,"OPEC Cut May Push Oil to $80 a Barrel, Lukoil Says (Update1)"

- **<date>_provider.csv:** Estos datos van a contener los titulares “del día” clasificados, resultantes de ejecutar un streamer de un broker determinado. Van a contener la fecha y el broker en su nombre (e.g 2015-03-23_bloomberg.csv). La estructura interna del contenido de estos archivos es idéntica a la de los archivos csv “idv”: [“category”, “date”, “url”, “headline”]. La diferencia con los anteriores radica en que en este caso habrá un archivo por día, mientras que el archivo índice contendrá todas las noticias independientemente del día.

2.4.5 Subsistema de visualización:



El subsistema de visualización va a encargarse de generar estadísticas y mostrar visualmente información útil sobre las noticias clasificadas por los clasificadores. El componente principal de este subsistema va a ser el servidor "Dashboard". Este servidor va a encargarse de leer las noticias almacenadas en los archivos índices ".csv" de las diferentes carpetas de los clasificadores, y parsearlos para extraer información de los mismos.

Una vez parseados los archivos, la información generada será procesada para crear una estructura de datos que contenga todas las noticias de un clasificador por cada día. Por ejemplo:

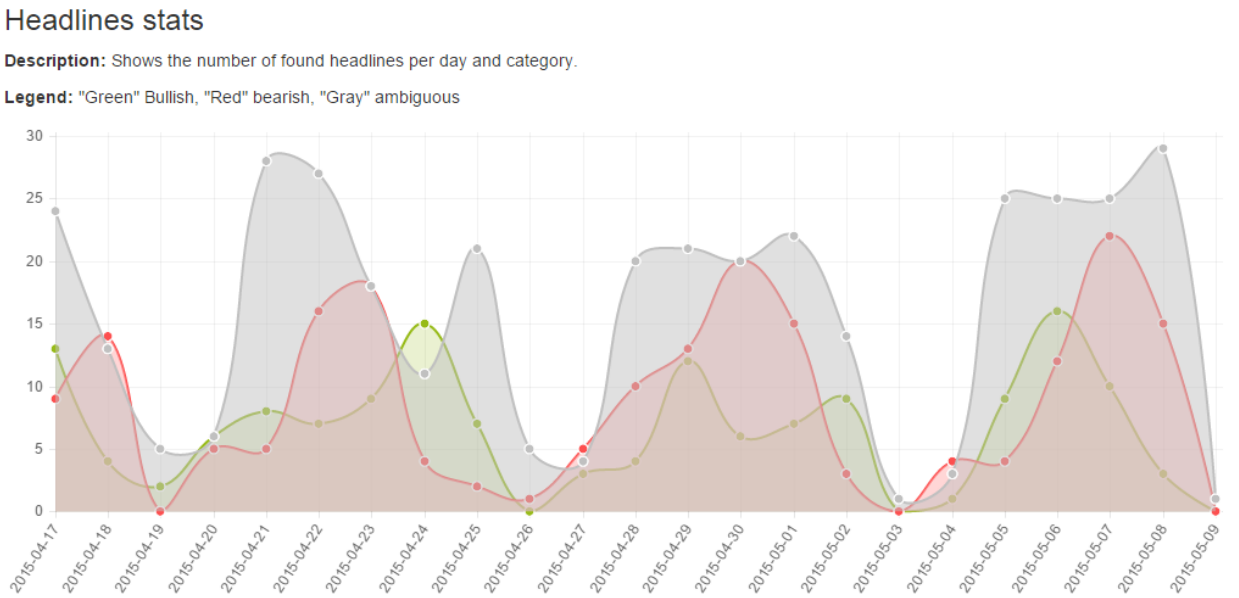
headlinesPerDate[‘2015-05-07’][‘bullish’] = [“Saudi Arabia holds oil prices for Asia steady on good demand”, “Wall St lower as weak trade data offsets rally in oil prices”, ...]

A su vez, también se creará una estructura que contenga el número de elementos por categoría. Estando en la primera posición la categoría “bullish”, en la segunda la categoría “bearish”, y en la tercera la categoría “ambiguous”. Por ejemplo:

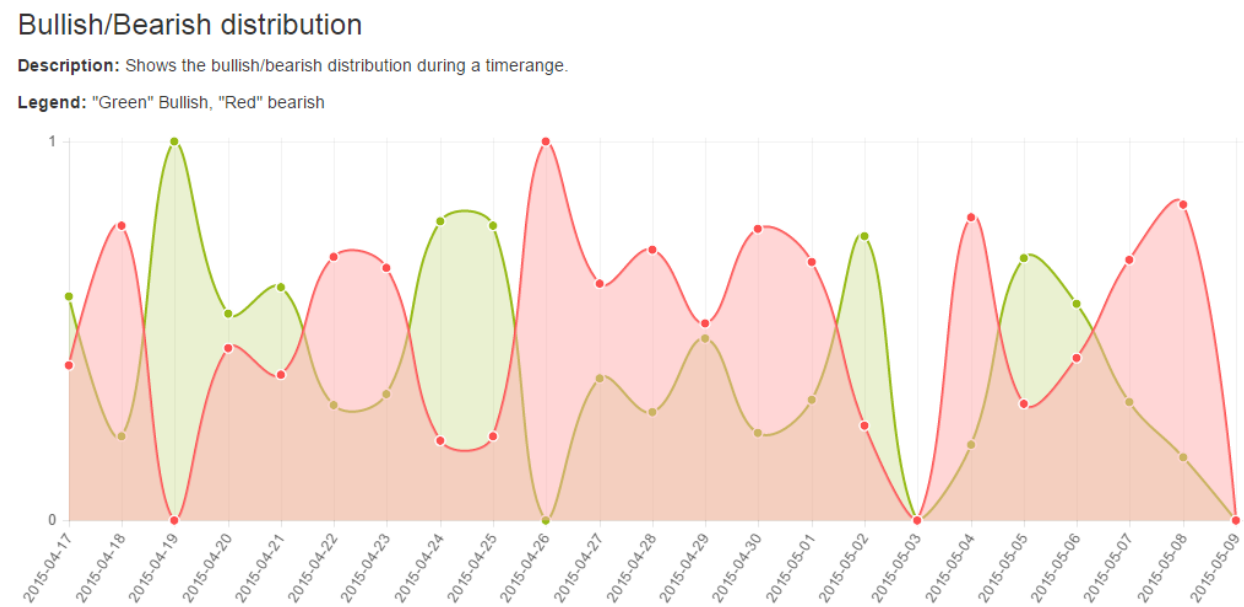
categoryPerDate[‘2015-05-07’] = [5, 2, 9]

Significará, por lo tanto, que existen 5 noticias bullish, 2 noticias bearish y 9 noticias ambiguous.

Una vez completadas, estas dos estructuras serán enviadas al navegador, donde se filtrarán los datos y se crearán dos gráficos. El primero mostrará simplemente el número de titulares para cada día y categoría:



El segundo, la distribución “bullish/bearish”, con el fin de analizar qué porcentaje de titulares de cada una de estas dos categorías tenemos cada día, siendo 1 la totalidad de titulares, y 0 ninguno (omitiendo la categoría "ambiguous"):



Como se puede observar, este gráfico va a ser prácticamente simétrico, ya que al ser una distribución de estas dos categorías, la suma de valores va a ser siempre 1, excepto cuando no tenemos ningún valor ese día. Por ejemplo:

Bullish[´2015-04-17´] = 0.6
Bearish[´2015-04-17´] = 0.4

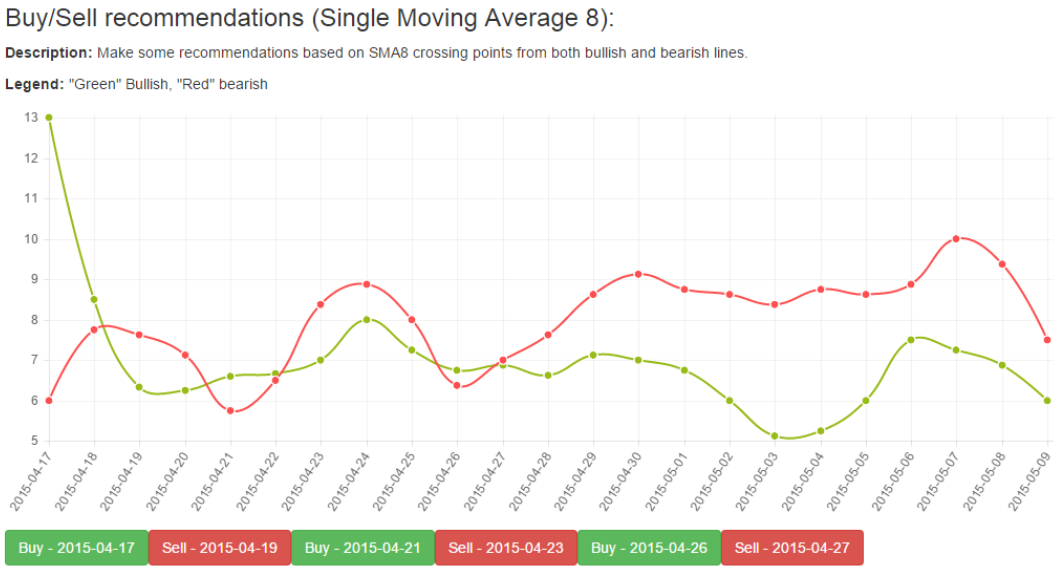
Bullish[´2015-04-17´] + Bearish[´2015-04-17´] = 1

Las recomendaciones de compra y venta merecen una explicación adicional debido a lo genérico de su título. Con recomendaciones, nos referimos a sugerencias basadas en ciertas métricas sencillas. Actualmente esta métrica es la comparación del **Simple Moving Average** [10] con periodo 8 para la distribución anterior bullish/bearish.

La métrica “Simple Moving Average” es muy utilizada a la hora de analizar la tendencia de los valores de un gráfico, ya que dependiendo del periodo, es capaz de minimizar el ruido de los datos y reaccionar únicamente cuando la tendencia sea “real”, es decir, estable. Básicamente, el funcionamiento de esta métrica es extremadamente sencillo, ya que indica simplemente el valor medio durante el periodo seleccionado. Es decir, si el periodo es 8 como en nuestro caso, SMA calculará el valor medio de los últimos 8 días.

El valor del periodo es bastante importante. Si quisiéramos reaccionar a los cambios de un modo más rápido, deberíamos utilizar un periodo menor, ya que los valores proporcionados por el SMA se parecerían más a los reales. Por otro lado, esto crearía muchos más falsos positivos al existir más ruido. Por el contrario, si utilizáramos un periodo mayor, se reduciría aún más el ruido, pero también se perderían muchas más oportunidades de compra o venta ya que el SMA tardaría mucho más en reflejar datos actuales. Así pues, se ha optado por elegir un periodo igual a 8, pues es un término medio.

Veamos un ejemplo de la utilización de esta métrica para la realización de recomendaciones, utilizando un gráfico sacado directamente del “dashboard”.



Como podemos apreciar se han creado recomendaciones de compra y venta cada vez que las dos líneas se cruzan. Si la línea verde ("bullish") cruza a la línea roja ("bearish") de un modo ascendente, indicará que la tendencia es alcista, es decir, que el sentimiento de las noticias empieza a ser más positivo, y por lo tanto, se emitirá una recomendación de compra.

Por otro lado, si la línea roja ("bearish") cruza a la línea verde ("bullish") de un modo descendente, indicará que la tendencia es bajista, es decir, que el sentimiento de las noticias empieza a ser negativo, y por lo tanto, se emitirá una recomendación de venta.

2.5 Herramientas utilizadas:

En el diseño del código del servidor se ha utilizado el lenguaje de programación **Python [11]**. La principal razón de su uso es su soporte excepcional de librerías científicas de ámbito general y numéricas, como Numpy y scipy, y más concretamente, su soporte de librerías de Machine learning, lo que lo convierten en un referente en este ámbito, y uno de los lenguajes más utilizados en proyectos de investigación.

Aparte de Python, se ha utilizado la librería **BeautifulSoup3 [12]**, que se especializa en el parseo y procesado de HTML, en la extracción y streaming de noticias de los diferentes medios online. Se ha elegido esta librería debido a su popularidad, y a la cantidad de recursos sobre ella disponibles en internet, lo que es una ayuda fundamental en caso de problemas con la misma.

Con respecto al uso de clasificadores y a su entrenamiento, se ha utilizado la librería **Scikit-learn [13]**, la cual es una de las librerías de Machine learning más conocidas en Python, y que ofrece soporte para multitud de técnicas y clasificadores. La razón de su uso, radica en la simplicidad que ofrece a la hora de realizar técnicas complejas, como por ejemplo el entrenamiento de clasificadores y posterior clasificación de elementos, y a las posibilidades que brinda debido a la cantidad de métodos implementados en la misma, que minimiza la necesidad de implementar funciones propias.

En lo referente a la creación del corpus y la extracción de frases, palabras, ngrams, y su filtrado posterior, se ha utilizado la librería **Nltk [14]**. Esta librería es actualmente una de las más conocidas a la hora de procesar lenguaje natural en Python, y ofrece multitud de herramientas que simplifican tareas como la “tokenización” de frases, o la eliminación de terminaciones de palabras gracias a la utilización de “stemmers”. Así pues, su uso en este proyecto ha sido fundamental, y ha reducido la complejidad de los algoritmos de post-procesado y filtrado.

Con respecto al código del "Dashboard" de la aplicación, se ha utilizado el lenguaje de programación **Javascript [15]** con algunas librerías para ciertos propósitos específicos. La utilización de este lenguaje se debe a la naturaleza “web” del “Dashboard”, y a los requerimientos de que éste debe de poder visualizarse en un navegador. A su vez, este lenguaje permite también su ejecución en el servidor con la ayuda de ciertos frameworks, lo que lo convierten en bastante polivalente, y ha sido una de las principales razones para su uso en este proyecto.

Con respecto a su uso en el servidor, se ha utilizado el framework **Express.js [16]** que precisamente permite la utilización de código escrito en Javascript en el servidor, lo que en nuestro caso nos permite acceder al disco y leer los archivos de noticias almacenados que posteriormente formarán los gráficos. En lo referente a esto último, es decir, al parseo de los archivos de noticias almacenados en forma de ficheros ".csv", se ha utilizado la librería **Papaparse [17]**, que permite la lectura sencilla y rápida de ficheros con esta extensión y estructura.

Una vez se ha extraído la información de estos archivos, hemos utilizado la librería **Chart.js [18]** para generar los gráficos a partir de las noticias parseadas. Esta librería utiliza el elemento "canvas" de HTML5 para crear gráficos elegantes e interactivos con facilidad. La razón de su uso radica nuevamente en su simplicidad, ya que con muy poca configuración es capaz de mostrar la información precisa visualmente.

En lo referente a la distribución, instalación y ejecución de la aplicación, se han utilizado dos herramientas que últimamente están revolucionado este ámbito debido a las ventajas que ofrecen, **Vagrant [19]** y **Docker [20]**.

La primera herramienta, Vagrant, permite la distribución sencilla de entornos de desarrollo, es decir, máquinas virtuales, con el código ya "preinstalado" en la misma. Por lo que para ejecutar el código, únicamente se necesitaría inicial la máquina virtual y ejecutar el script.

La segunda herramienta, Docker, introduce el concepto de "contenedores". Un contenedor es un sistema aislado que permite ejecutar comando en él. Este contenedor es más liviano que una máquina virtual, por lo que es mucho menos costoso en términos de utilización de recursos.

Así pues, en el proyecto que nos ocupa, se ha utilizado Vagrant para instalar automáticamente todas las dependencias necesarias de todas las aplicaciones que componen este proyecto (extractores, streamers, dashboard, etc...) dentro de una máquina virtual, de modo que el ordenador principal no tenga que instalar nada por sí mismo, y por lo tanto, evitamos que haya conflictos de librerías o versiones.

Por otro lado, en lo referente a Docker, se ha utilizado siempre dentro de esta máquina virtual, para la ejecución de todos los scripts necesarios, como por ejemplo la ejecución del entrenamiento de clasificadores, la ejecución de streamers, extractores o para la inicialización del dashboard.

Esto va a tener varias ventajas, en primer lugar al ejecutarse cada comando dentro de su propio “contenedor”, no van a existir conflictos de ningún tipo si dos comandos están ejecutándose al mismo tiempo. En segundo lugar, esto nos da la posibilidad de ejecutar en paralelo multitud de comandos con diferentes variables de entorno, imaginemos por ejemplo que una variable de entorno selecciona el petróleo para un contenedor, y el “oro” en otro contenedor, sin que haya conflictos de ningún tipo. En tercer y último lugar, en caso de errores la inspección “post-mortem” de un contenedor es trivial gracias a la interfaz proporcionada por Docker, por lo que es mucho más sencillo conocer los logs del contenedor y por qué se ha producido un error en el mismo.

3. Resultados.

Los resultados obtenidos son muy prometedores. En lo relativo a la clasificación, utilizando los siguientes clasificadores: *NaiveBayes, Maxent, LinearSVC y NuSVC*, se ha conseguido una precisión en la clasificación cercana al 80%, siendo el clasificador NaiveBayes el que arroja por el momento mejores resultados, seguido muy de cerca por NuSVC.

Tal como se ha explicado anteriormente en el subsistema de entrenamiento, para “feature set” se han seleccionando palabras sueltas, bigrams y trigrams dentro de las 2000 ocurrencias (palabras y grupos) más informativas del corpus. Al mismo tiempo, también se han seleccionado fourgrams, pero en este caso dentro de los 666 (2000/3) conjuntos y palabras más informativas del corpus. Por último se ha aplicado el Stemmer de Porter a todas las palabras para almacenar únicamente las raíces de las palabras y lograr una mayor uniformidad. Esta configuración ha sido la que mejores números ha obtenido durante todas las combinaciones probadas.

El corpus utilizado tiene unas 1050 noticias por categoría, conteniendo un total de 3150 noticias entre las 3 categorías (bullish, bearish y ambiguous). Existe una cuarta categoría "discarded" que como su nombre indica está formada por noticias demasiado similares o no relacionadas con el petróleo. Esta categoría **no va a ser utilizada en ningún momento** y únicamente nos va a servir para almacenar noticias descartadas.

A continuación se van a detallar los resultados obtenidos a grandes rasgos con cada clasificador, utilizando como resultados la media después de 5 iteraciones, un 75% del corpus como conjunto de entrenamiento y el restante 25% como conjunto de test. Por último, se han filtrado todas las stopwords detectadas, al igual que todas las palabras con menos de 2 letras:

- **Naive Bayes:** 0.798 (Accuracy)
- **Max Entropy (utilizando el algoritmo Megam)**: 0.78 (Accuracy)
- **Linear SVC:** 0.782 (Accuracy)
- **NuSVC:** 0.798 (Accuracy)

Como se puede apreciar los dos clasificadores con mejor precisión son **Naive Bayes** y **NuSVC**. Dependiendo de la ejecución estos resultados pueden variar unos **+-0.2** aproximadamente de media.

Poniendo en perspectiva estos valores, estudios y artículos como [21] y [22] indican que la precisión propia de un ser humano a la hora de clasificar el sentimiento de un determinado texto ronda entre el 75% y el 80%. La razón es que la clasificación de textos en muchas ocasiones es subjetiva ya que incluye gran cantidad de ambigüedades. Para una persona, una noticia puede ser positiva, para otra persona, negativa, y para una tercera, ambigua.

Un ejemplo que me encontrado en multitud de ocasiones durante la realización de este proyecto, han sido las preguntas que no indican absolutamente ningún sentimiento, pero que incluyen adjetivos positivos y negativos en ellas. Por ejemplo:

Is plunging oil good for the Economic Growth?

Es fácil de comprender, por lo tanto, que un clasificador tendría bastantes problemas en clasificar correctamente la pregunta anterior. Incluso iría más allá y diría que muchas personas identificarían algún sentimiento inexistente en la pregunta dependiendo del modo en el que se les entrevistara.

Otro ejemplo que he visto en algunas ocasiones, son los titulares con varias frases en ellos. Por ejemplo:

Oil slumps; Oil might go up next week.

En este caso tenemos dos frases diferentes dentro de un mismo titular. Teóricamente el clasificador debería ser capaz de identificar qué adjetivo es el más importante, y cuales podría descartar. En la práctica, no es tan sencillo ya que ambos están relacionados con el petróleo, aunque sólo uno indica la tendencia actual.

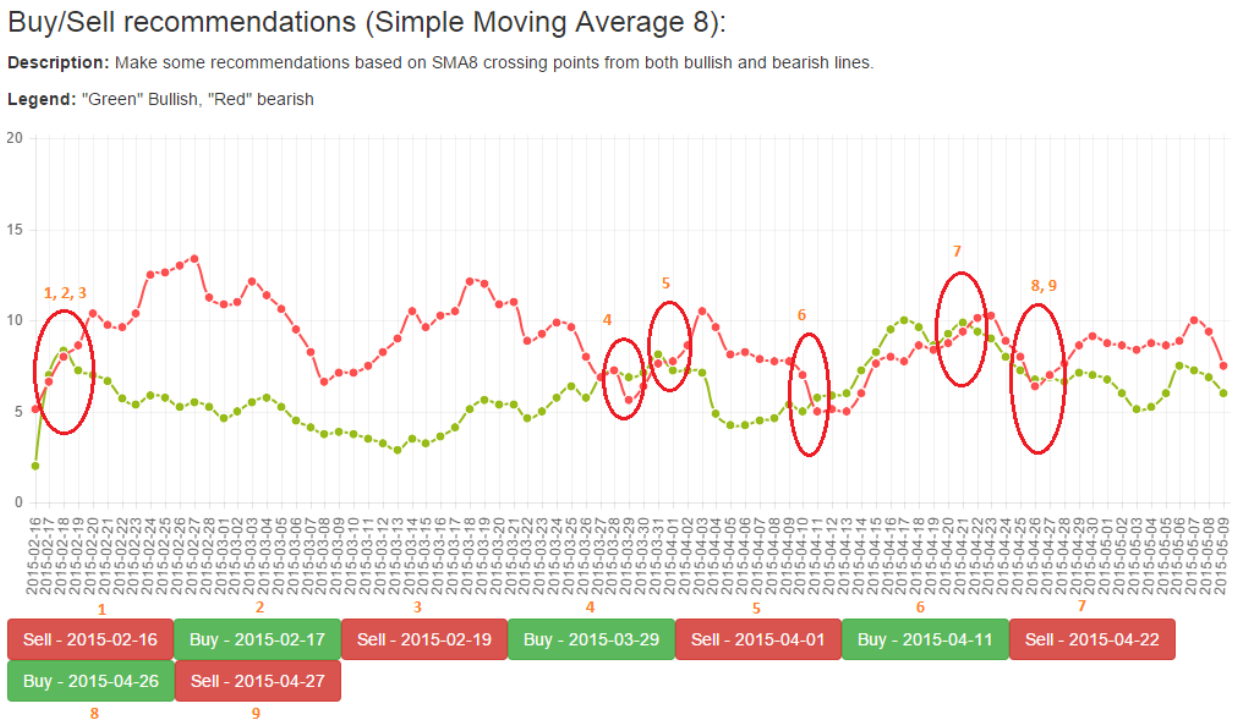
Así pues, la clasificación de textos en lenguaje natural es una tarea difícil y en muchos casos inexacta debido al contexto de cada frase. Por otro lado, es posible sacar información útil si se ponen los resultados en perspectiva, y se aceptan estas limitaciones.

En lo relativo a las recomendaciones de compra y venta mostradas en el dashboard, los resultados obtenidos son positivos por norma general, emitiendo recomendaciones acertadas en la mayoría de los casos. Existe, no obstante, un margen de mejora claro en lo relativo a la agregación de datos, ya que en algunas ocasiones, sobre todo en momentos de alta volatilidad, los valores son tan cambiantes que si se muestran los datos diarios el recomendador tiende a ser demasiado "lento" y

confundirse con facilidad, dando sugerencias cuando la tendencia ya ha concluído. Debido a esto, actualmente se recomienda el uso del recomendador para operaciones a medio y largo plazo (utilizando el rango de 3 - 6 meses), pero no para operaciones de varios días de duración.

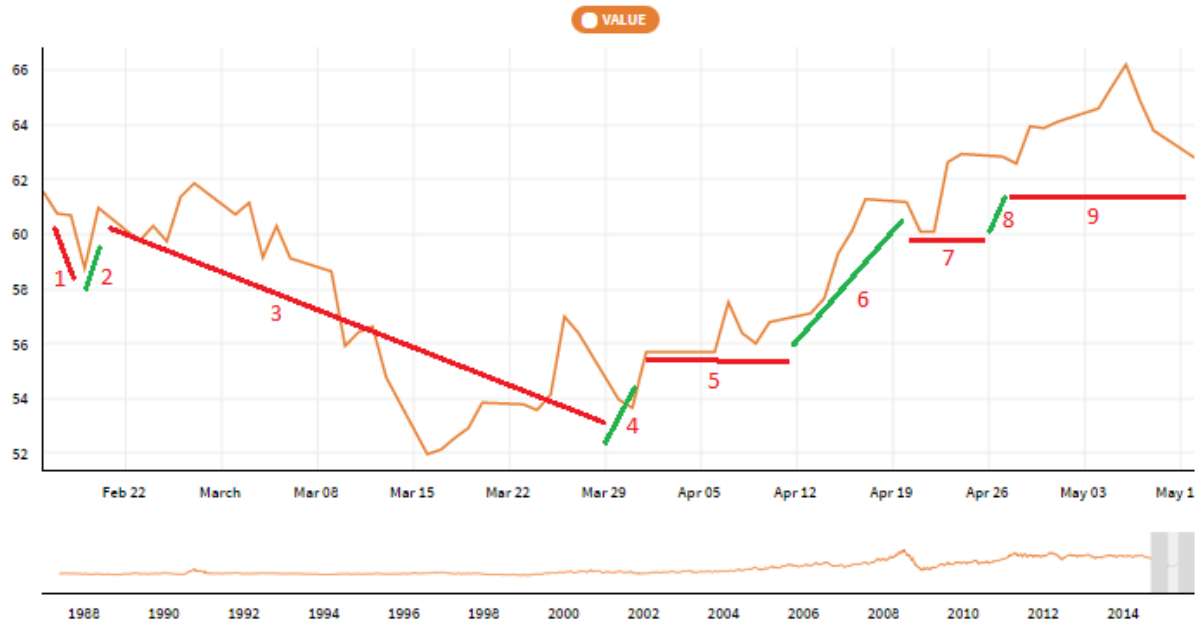
Para solucionar parcialmente estos problemas, o bien se agregan los datos semanalmente para rangos de tiempo mayores a tres meses, o se reduce el periodo del SMA. Otro cambio que podría mejorar la precisión sería incrementar el número de proveedores disponibles, incluyendo un mayor número de noticias por día, lo que haría el porcentaje más representativo.

A continuación, y para analizar los resultados con detalle, vamos a observar un ejemplo con recomendaciones de compra y venta emitidas por el sistema durante el periodo entre el **16 de Febrero de 2015, al 9 de Mayo del mismo año** para noticias relativas al petróleo. Las noticias utilizadas han sido extraídas previamente por los extractores y clasificadas posteriormente por los clasificadores ya entrenados.



Cada círculo rojo representa una zona donde se han emitido recomendaciones de compra o venta. Como se puede apreciar, cada recomendación ha sido numerada, y su numeración su muestra al lado del círculo al que pertenece con el fin de mejorar su localización.

Veamos a continuación el reflejo de estas recomendaciones en un gráfico del precio del barril de Brent durante el periodo antes mencionado:



Como se puede ver, las recomendaciones son en su mayoría acertadas, identificando la tendencia general de precio correctamente. Como errores destacables, podemos apreciar como las recomendaciones 5, 7 y 9, aún habiendo sido identificadas como tendencias bajistas, no se corresponden como tal en el gráfico superior, mostrando un comportamiento estacionario o incluso ligeramente alcista. Achaco estos errores al número de titulares diarios extraídos y utilizados, que si bien es un número aceptable en la mayoría de las ocasiones, quizá no sea suficiente en un ambiente de tanta volatilidad.

4. Conclusiones.

Las conclusiones generales son positivas. Se han conseguido clasificar titulares reales de medios como Bloomberg, Financial Times y Wall Street Journal en base al sentimiento, de un modo bastante certero. Se ha conseguido utilizar los titulares clasificados para realizar recomendaciones de compra y venta de un producto financiero con un porcentaje de acierto importante.

Por lo tanto, se ha demostrado que el análisis y predicción de tendencias basado en titulares publicados puede ser una importante herramienta a utilizar por personas cuyo trabajo dependa de conocer o intuir qué tendencia tiene o va a tomar un determinado producto, con el fin de realizar operaciones de mercado.

Este análisis tiene propiedades objetivas, que no dependen de la persona que visualiza los resultados ni de su estado emocional, por lo que es un complemento perfecto a utilizar en entornos de alta presión donde los sentimientos del evaluador puedan distorsionar el resultado final. Es un análisis con unas posibilidades de configuración casi ilimitadas, donde podemos utilizar multitud de métricas, algoritmos y variables externas provenientes de diferentes brokers. Es un análisis que puede utilizar las ventajas que ofrece la nube y el Big Data para detectar patrones de comportamiento y así desarrollar mejores estrategias predictivas.

Por último, otra de sus mayores virtudes es su generalidad, ya que es posible utilizar este enfoque con la mayoría de los productos financieros -- petróleo, oro, plata, gas, etc.. -- o con acciones de empresas dentro de mercados bursátiles. Únicamente necesitaríamos fuentes de noticias que hablen de estos productos regularmente o los evalúen de algún modo.

Así pues, creo firmemente que la utilización de esta aplicación, o de un modo más general, el enfoque de analizar el sentimiento de noticias y realizar recomendaciones, puede ayudar a profesionales del sector, o simplemente a personas interesadas en el tema, a mejorar las planificaciones de sus inversiones y la gestión del riesgo, proporcionándoles una potente herramienta para el soporte de sus decisiones en lo relativo a la compra y venta de productos financieros.

5. Bibliografía.

[1] Wikipedia. "Sentiment Analysis." <http://en.wikipedia.org/wiki/Sentiment_analysis>

[Consulta: 27 febrero. 2015].

[2] SAS. "SAS Sentiment Analysis tool"
<http://www.sas.com/en_us/software/analytics/sentiment-analysis.html>

[Consulta: 6 mayo. 2015]

[3] Weotta. "Weotta" <http://www.weotta.com/>

[Consulta: 6 mayo. 2015]

[4] Opfine. "Opfine" <http://www.opfine.com/>

[Consulta: 6 mayo. 2015]

[5] Nltk. "Naive Bayes." <http://www.nltk.org/_modules/nltk/classify/naivebayes.html>

[Consulta: 27 febrero. 2015].

[6] Nltk. "Max Entropy" <http://www.nltk.org/_modules/nltk/classify/maxent.htmlr>

[Consulta: 27 febrero. 2015].

[7] MEGA Model Optimization Package. "Megam."
<http://www.umiacs.umd.edu/~hal/megam/version0_3/>

[Consulta: 8 abril. 2015].

[8] Scikit-learn. "Linear SVC."
<http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>

[Consulta: 27 febrero. 2015].

[9] Scikit-learn. "Nu SVC."
<http://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVC.html>

[Consulta: 27 febrero. 2015].

[10] Wikipedia. "Simple moving average"
<http://en.wikipedia.org/wiki/Moving_average#Simple_moving_average>

[Consulta: 6 mayo. 2015].

[11] Python Software Foundation. "Python" <https://www.python.org/>

[Consulta: 27 febrero. 2015].

[12] Crummy. "Beautiful Soup 3"
<http://www.crummy.com/software/BeautifulSoup/bs3/>

[Consulta: 27 febrero. 2015].

[13] Scikit Learn. "Scikit-learn" <http://scikit-learn.org/stable/>

[Consulta: 27 febrero. 2015].

[14] Natural Language Toolkit. "NLTK" <http://www.nltk.org/>
[Consulta: 27 febrero. 2015].
[15] Wikipedia. "Javascript" <http://en.wikipedia.org/wiki/JavaScript>
[Consulta: 27 febrero. 2015].
[16] Express. "Express.js" <http://expressjs.com/>
[Consulta: 8 abril. 2015].
[17] Papaparse. "Papaparse.js" <http://papaparse.com/>
[Consulta: 8 abril. 2015].
[18] Chart.js. "Chart.js" <http://www.chartjs.org/>
[Consulta: 8 abril. 2015].
[19] Vagrant. "Vagrant" <https://www.vagrantup.com/>
[Consulta: 8 abril. 2015].
[20] Docker. "Docker" <https://www.docker.com/>
[Consulta: 8 abril. 2015].
[21] Brnrd. "Brnrd.me" <http://brnrd.me/sentiment-analysis-never-accurate/>
[Consulta: 6 mayo. 2015].
[22] Grimes, Seth. "Is Sentiment Analysis a 80% solution?" <http://www.informationweek.com/software/information-management/expert-analysis-is-sentiment-analysis-an-80--solution/d/d-id/1087919?>
[Consulta: 6 mayo. 2015].
[23] Perkins, Jakob. "Text classification for sentiment Analysis, Naive Bayes classifier" <http://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/>
[Consulta: 27 febrero. 2015].
[24] Perkins, Jakob. "Text classification for sentiment Analysis, Stopwords and collocations" <http://streamhacker.com/2010/05/24/text-classification-sentiment-analysis-stopwords-collocations/>
[Consulta: 27 febrero. 2015].
[25] Perkins, Jakob. "Text classification for sentiment Analysis, Eliminate low information features" <http://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/>
[Consulta: 27 febrero. 2015].
[26] Perkins, Jakob. "Python 3 Text Processing with NLTK 3 Cookbook". Paperback – August 26, 2014