Real-Time behavioural stream analysis with Big Data stack technologies.

Pedro Puertas Ballesteros
Grau Enginyeria Informàtica

Humberto Andrés Sanz

14 de Juny de 2017

FITXA DEL TREBALL FINAL

Títol del treball:	*Real-Time behavioural stream analysis with Big Data stack technologies.*
Nom de l'autor:	*Pedro Puertas Ballesteros*
Nom del consultor:	*Humberto Andrés Sanz*
Data de lliurament (mm/aaaa):	*03/2017*
Àrea del Treball Final:	*Business Intelligence*
Titulació:	*Grau en Enginyeria Informàtica*

Resum del Treball (màxim 250 paraules):

En aquesta memòria es descriu la creació d'un projecte Big Data utilitzant les tecnologies de AWS Kinesis i Spark Streaming i de visualització de Looker. L'objectiu es poder valorar l'activitat del usuaris a través de les accions dintre de l'aplicació de Wallapop. Per tal de poder valorar aquesta es construeix una matriu de transicions on a través del model Hidden Markov es puntua aquestes. Per últim, es pre-bloquejen el usuaris per tal de valorar la seva activitat i prendre una decisió.

Abstract (in English, 250 words or less):

In this report shows the creation of a Big Data solution using the technologies of AWS Kinesis and Spark Streaming and for the visualization part was used Looker. The objective is to be able validate the activity of the users through the action inside the application of Wallapop. To be able to validate the user activity I made a matrix of transitions using the Hidden Markov Model, this model calculate the probability between the actions of the user. Finally, it is to

pre-block the user validating the activity and make a decision.
Paraules clau (entre 4 i 8):
big data, machine learning, business intelligence, deep learning, hadoop, spark, kinesis, wallapop

Índex

Agraïments

Aquest treball no sols representa el treball final de grau, sinó que es tracta d'una sèrie d'anys de dedicació i esforços enfocats en aprendre més del món tecnològic i en concret del business intelligence.

En primer lloc, donar les gracies als meus companys de feina de Wallapop, Jaume, Nico, German i Javi per ajudar-me en tot moment en tirar endavant el projecte internament amb els seus consells.

Donar les gracies al meu tutor de projecte, Humberto, per el seguiment fet durant l'avaluació continua.

A la meva parella Mónica, per ajudar-me en tot moment durant els moments més difícils dels estudis.

Per últim, però no menys importants, als meus pares. Gràcies per recolzar-me en totes les decisions que he pres a la vida.

Introducció

Avui en dia és molt comú sentir a parlar de Big Data. Aquest serveis permeten treballar amb grans volums de dades, varietat de informació i per últim velocitat d'accés a aquesta informació. Per aquest motiu, la finalitat d'aquest treball final de grau consistirà en recollir, analitzar i mostrar la informació per a poder prendre decisions de negoci que permeti identificar usuaris amb comportaments fraudulents.

Actualment treballo a Wallapop al departament de Business Intelligence, com a BI Developer, on degut al creixement d'usuaris és necessari disposar de processos "real-time" on es prenguin decisions que millorin l'experiència final de l'usuari. El projecte estarà inclòs dins l'ecosistema tecnològic de l'empresa que es troba al cloud d'Amazon (AWS[1]) i seguint els estàndards interns.

El projecte consistirà de tres parts diferenciades: una part d'adquisició en temps real de dades (ETL[2] real-time) on s'extrauran dades de la activitat dels usuaris en l'aplicació mòbil (Android i iOS), una part de processament de dades aplicant tècniques de "machine learning", on a partir d'una sèrie de regles es classificaran en diferents conjunts els usuaris segons el tipus d'activitat i, finalment, una part de visualització on es podrà veure i analitzar els resultats obtinguts.

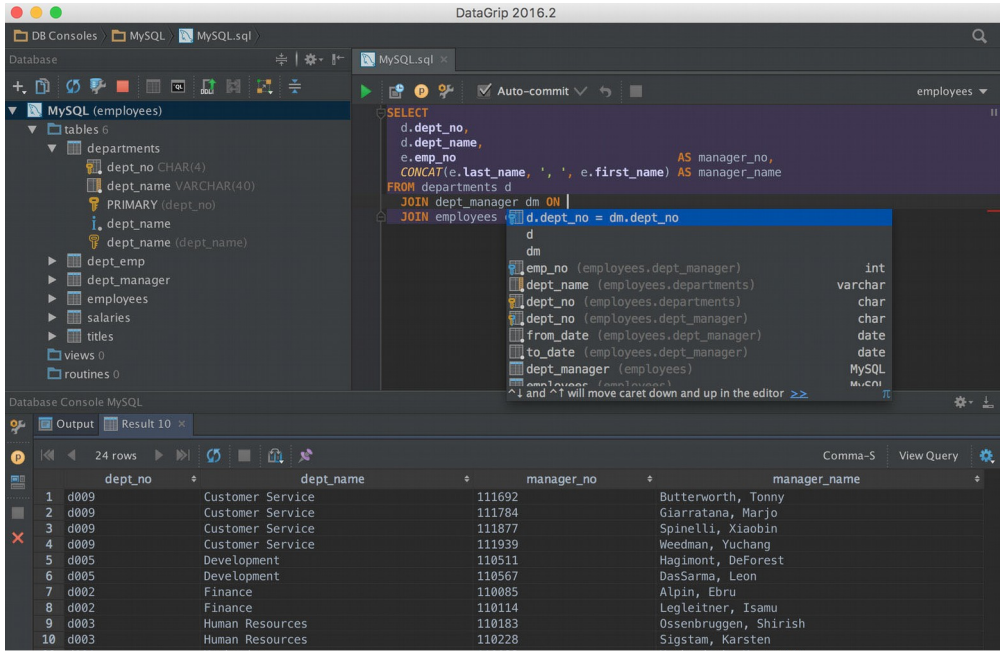
Existirà una font principal de dades anomenada "clickstreaming" on es reben les accions del usuari en la aplicació i una altre font provinent de la base de dades relacional (MySQL), a través de AWS Kinesis es processarà tota les dades en temps real.

Amb l'ús de l'aplicació obtenim una sèrie de patrons que ens permetrà classificar els usuaris. A partir d'aquesta informació es realitzaran una sèrie de comprovacions per tal de verificar el comportament utilitzant "machine learning" i utilitzant Apache Spark Streaming per a tractar les dades.

1 AWS: Amazon Web Services
2 ETL: Extract, Transform and Load (extracció, transformació i càrrega)

Per assolir els projecte, serà necessari utilitzar una sèrie de tecnologies que es descriuen a continuació, aquestes podran ser modificades durant la realització del projecte:

- DataGrip: aplicació multiplataforma que permetrà executar codi SQL[3] a les bases de dades relacionals.

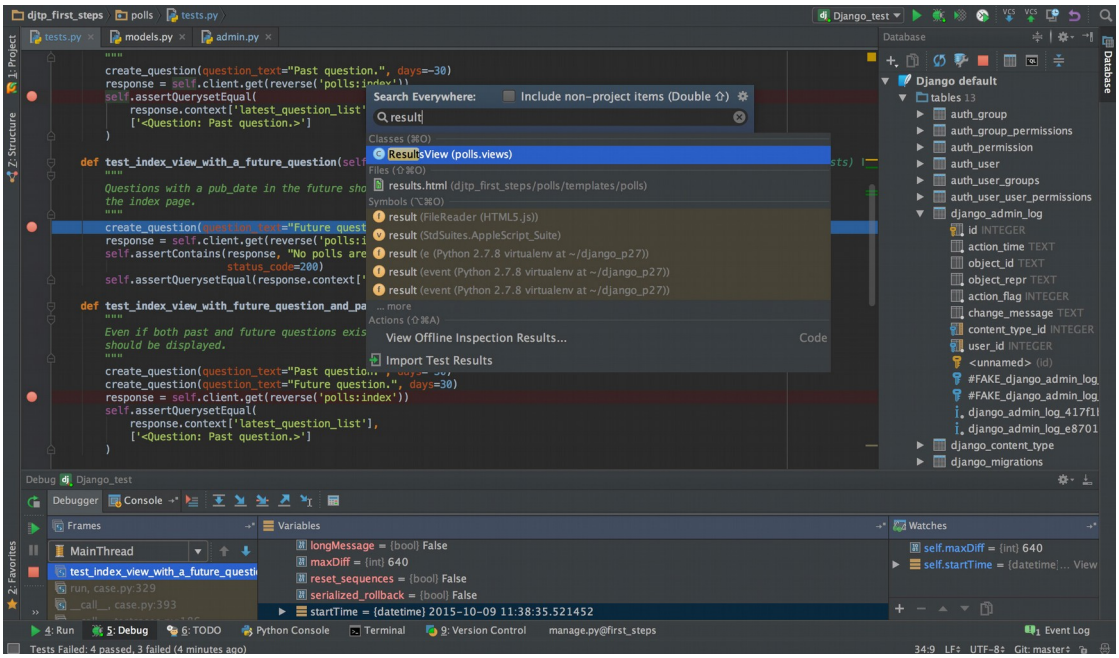


Il·lustració 1 DataGrip

- PyCharm: entorn de desenvolupament multiplataforma que permetrà programar gran part de l'aplicació amb llenguatge Python.

3 SQL: Structured Query Language

Il·lustració 2 PyCharm

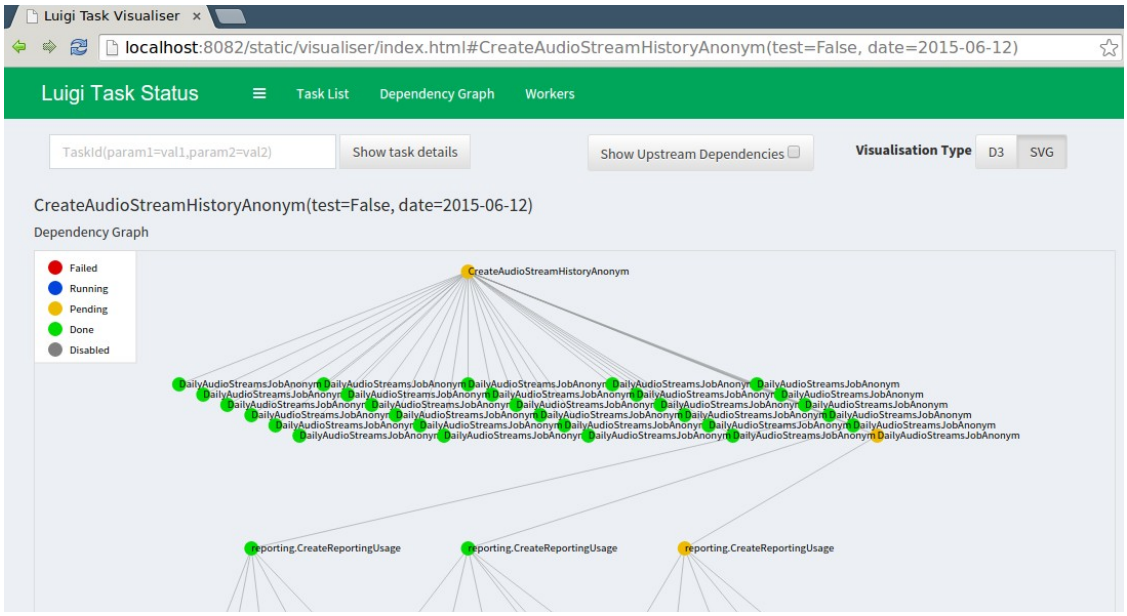


- AWS RDS MySQL: servei de Amazon de bases de dades relacionals en entorn cloud basada en MySQL.

- AWS Kinesis: es una plataforma que permet transmetre dades en “streaming” a AWS (Amazon Web Services).

- Apache Spark Streaming: sistema de computació distribuïda a través de clústers d’ordinadors que permet tractar grans volums de dades ràpidament.

- Luigi: es un *framework* que permet construir complexes data pipelines i permet gestionar el flux de treball, les dependències i la visualització de possibles errors entre d’altres.



Il·lustració 3 Luigi

- GitHub: plataforma “cloud” de emmagatzemament i versionat del codi a desenvolupar.

En funció del temps disponible, es podria incorporar altres funcionalitats “machine learning” addicionals per tal de detectar més casos d’usos fraudulents.

Per interactuar amb els resultats finals obtinguts serà necessari d’autentificar-se a l’eina de visualització corporativa Looker. On es podran generar “reports” i “dashboards” amb les dades.

Context i Justificació del Treball

La ràpida expansió de les aplicacions mòbils de cara al consum general han portat a una explosió de nous usuaris a les noves tecnologies on és molt important l’ús responsable d’aquestes.

Els usuaris estafadors s’han aprofitat d’aquesta oportunitat creant campanyes d’atacs ben organitzades utilitzant el creixement al seu favor, creant falsos exèrcits i comptes

comprometedores per tal de ocultar-se en l'ombra i dur a terme accions delictives en diverses webs i aplicacions mòbils. És un repte per als equips tecnològics mantenir la confiança i seguretat actualitzada per tal de poder combatre els atacants, aquestes armes de defensa necessiten un manteniment constant i en la majoria de casos s'actua després de rebre l'atac.

Per tant, és necessari poder complir una sèrie de requeriments per combatre els atacants com:

- Detecció prèvia abans de rebre l'atac.
- Preveure conjunts d'atacants.
- Utilitzar el mínim conjunt de dades per entrenar el model de prevenció d'atacs.
- Capacitat d'analitzar un gran nombre d'events.
- Verificar l'autenticitat de l'usuari.
- Donar de baixa el usuaris fraudulents.
- Establir noves polítiques de seguretat.

Objectius del Treball

L'objectiu d'aquest treball final de grau es basa principalment en la implementació d'una eina en el "cloud" de Amazon, que permeti identificar i prevenir atacs de frau a través de la identificació de patrons en l'activitat realitzada.

Per tal de prevenir els atacs serà necessari el poder disposar dels events en "real-time" i així de detectar-los el més aviat possible, integrant les diferents fonts de dades. A partir de tècniques de "machine learning" es classificaran el usuaris i s'identificaran els possibles usuaris fraudulents. Per últim, a equips de negoci i tècnics se'ls donarà una visió dels resultats obtinguts per tal de poder prendre decisions i millorar la confiança de l'usuari.

Enfocament i Mètode

El procés per a desenvolupar solucions tecnològiques és una estructura de passos organitzats per a obtenir un producte final. O el que es el mateix, unes guies que ens diuen que fer i quan poder obtenir un resultat. També es sol denominar a aquest grup de accions, cicle de vida del desenvolupament d'eines tecnològiques, ja que es defineix quan s'inicia el projecte, els passos a seguir, i com terminar el projecte.

A l'hora de planificar i desenvolupar el projecte ens centrarem en com es realitza actualment en les empreses tecnològiques. On a partir de diverses iteracions ha evolucionat la manera de gestionar la planificació i millorar el desenvolupament fins arriba a uns objectis més eficaços.

Mitjançant altres projectes ja realitzats i amb diverses lliçons apreses en l'empresa, i la identificació de les millors practiques i enfocaments de treball, es possible obtenir uns millors resultats que faran realitzar un projecte més productiu per a l'empresa.

Al nostre abast tenim diverses metodologies que poden ser utilitzades com a guia a l'hora d'administrar i executar els projectes de tipus tecnològics. Cal estructurar correctament les fites del treball a realitzar. A través d'aquestes metodologies, es possible determinar l'ordre i els passos necessaris a seguir per realitzar les fites de manera més eficaç complint els objectius.

En aquest sentit, existeixen múltiples tendències, filosofies, metodologies i eines que poden ser aplicables en funció del context del projecte, i de la organització sobre el que es vol desenvolupar.

Existeixen diversos cicles de vida que poden ser aplicats a la construcció de programari. No hi ha un mètode millor que un altre, sinó que cada un s'adapta a unes característiques particulars del producte a obtenir. A continuació, descriuré alguns models entre els més comuns a l'hora de realitzar projectes tecnològics.

Models en cascada

Aquest tipus de model està plantejat en blocs individuals que es van concatenant en el temps. Els grups d'activitats son els següents:

Requisits: es defineixen completament totes les necessitats del projecte. Poden ser tasques que s'han de realitzar o limitacions.

Disseny: en aquest pas es realitza una descripció del component necessaris, l'arquitectura, la neteja de les dades, etc.

Implementació: és on es codifica la solució amb les tecnologies necessàries.

Verificació: es comprova que tot funciona com es va definir en els requisits.

Manteniment: com es difícil detectar totes les funcionalitats siguin correctes, cal que es realitzi un seguiment de la solució per tal de solucionar futurs problemes o noves millores.

Il·lustració 4 Model en Cascada

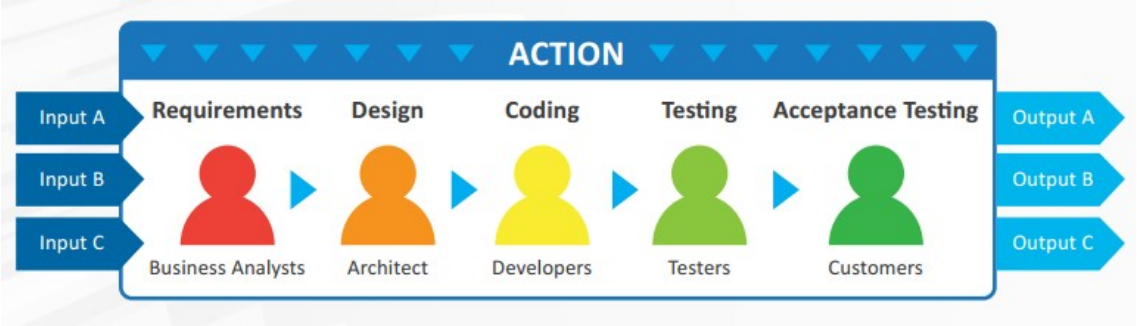
Es necessari definir tot el que es requereix des d'un inici en la fase inicial. Aquest punt es clau a l'hora d'obtenir el resultats definits.

Model iteratiu

Es una evolució del model en cascada per quan no estan clarament definits els requisits o volem dividir el projecte en diferents fases. L'estratègia es basa en seguir el model incremental, establint diverses entregues del producte, cada una amb major funcionalitat que la anterior. Per lo tant, al acabar el primer increment, es continua desenvolupant per el següent, i així successivament fins obtenir el producte finalitzat.

Aquest permet dividir el resultat en diversos paquets i poder veure l'evolució del resultat en cada entrega. La principal clau al utilitzar aquest model es definir prèviament les fites d'entrega, ja que si anem iterant sobre la solució, podem arribar a un punt on el desenvolupament s'allargui tant que no s'obtindrà el producte final.

Aquest detall permet que es pugui modificar requisits en cada entrega, tenint en compte que si es modifica algun requisit, es consumeix temps, tenint com obligació descartar d'altres requisits per a que el producte es realitzi en el termini definit.



Il·lustració 5 Model Iteratiu

Model per prototips

El model per prototips s'utilitza per tenir una visió del producte però sense funcionalitat. Amb aquest model es pot veure com quedarà la solució, per tal de poder fer les modificacions abans es desenvolupi.

L'estructura d'aquest model s'estableix com el desenvolupament d'una versió prèvia sense funcionalitat que s'entrega al client. Al revisar aquest es realitzen correccions,

canvis en el format, etc, però sense afegir cap nova funcionalitat. Quan s'arriba al prototip desenvolupat, s'afegeix totes les funcionalitat.

Una estratègia eficaç per aquest model es definir terminis per obtenir el prototip final, amb diversos prototips previs, i després d'establir els terminis per a desenvolupar la solució, que sol ser aplicant un model en cascada.

Metodologies àgils

Existeixen diversos models que destaquen dintre de les metodologies àgils a l'hora de desenvolupar un projecte tecnològic, com son *Scrum* i *Kanban*.

Scrum, es un model que defineix períodes dintre del desenvolupament, reunions i rols, així com una documentació pròpia. Es centra en la definició del treball per el que l'equip és capaç de fer.

Il·lustració 6 Metodologies Àgils

Kanban, es similar a l'anterior però es una estratègia de treball visual, es centra en seguir l'evolució de les tasques individualment fins que es finalitzant, passant per distints estats i sent assignades a membre de l'equip.

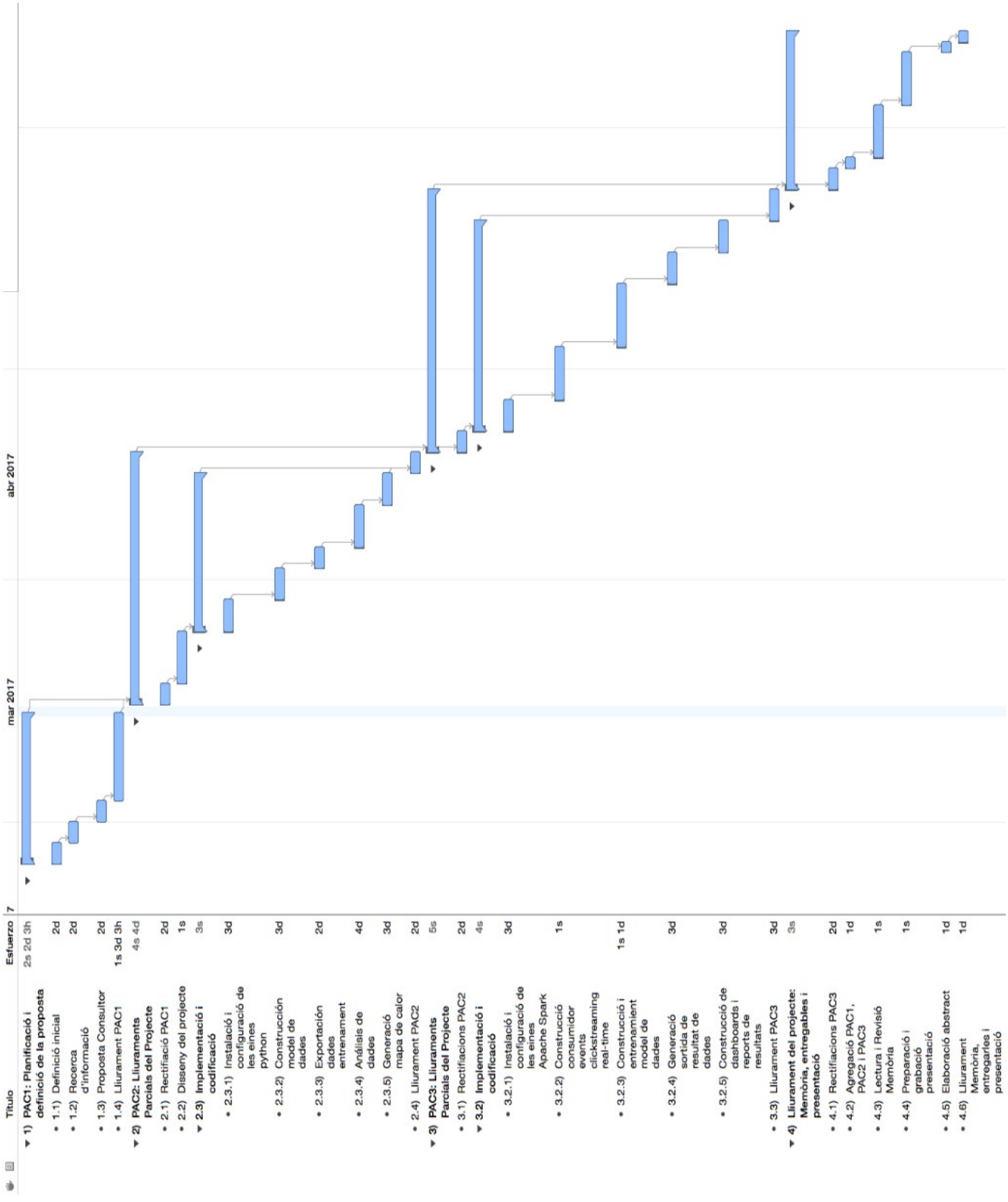
A l'empresa després d'analitzar el funcionament de les diverses tècniques i seguint la tendència del mercat, s'ha optat per seguir *Scrum*, on existeix un *Product Owner* per equip, aquest s'encarrega de gestionar el *backlog* dels membres de l'equip. Els responsables de planificar i executar les tasques son els desenvolupadors on se'ls atorga d'aquesta manera una sèrie de responsabilitats.

Lliurables

Els entregables principals es basaran en el model d'avaluació continua, on a les PAC2 i PAC3, s'inclourà en detall la implementació del projecte. A part es lliurarà un entregable final que constarà de la memòria final del Treball Final del Grau conjuntament de la presentació d'aquest.

Temporalització de les tasques

S'ha planificat el projecte seguint les fites marcades en el calendari de l'avaluació continua i dividint aquestes en diferents tasques per aconseguir els diferents objectius. Per fer la planificació s'ha utilitzat el programa OmniPlan, a continuació es detalla el planning:

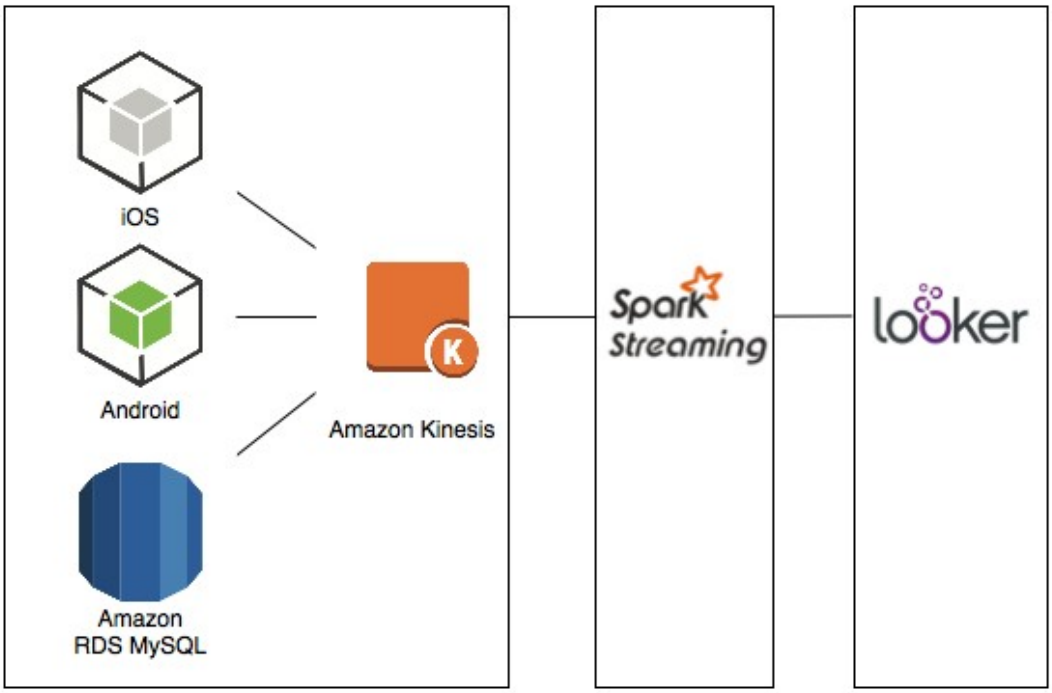

Título
Esfuerzo
7
mar 2017
abr 2017
1) PAC1: Planificació i definició de la proposta 2s 2d 3h
• 1.1) Definició inicial 2d
• 1.2) Recerca d'informació 2d
• 1.3) Proposta Consultor 2d
• 1.4) Lliurament PAC1 1s 3d 3h
2) PAC2: Lliuraments Parcials del Projecte 4s 4d
• 2.1) Rectificació PAC1 2d
• 2.2) Disseny del projecte 1s
2.3) Implementació i codificació 3s
• 2.3.1) Instalació i configuració de les eines python 3d
• 2.3.2) Construcción model de dades 3d
• 2.3.3) Exportación dades entrenament 2d
• 2.3.4) Análisis de dades 4d
• 2.3.5) Generació mapa de calor 3d
• 2.4) Lliurament PAC2 2d
3) PAC3: Lliuraments Parcials del Projecte 5s
• 3.1) Rectificacions PAC2 2d
3.2) Implementació i codificació 4s
• 3.2.1) Instalació i configuració de les eines Apache Spark 3d
• 3.2.2) Construcció consumidor events clickstreaming real-time 1s
• 3.2.3) Construcció i entrenament model de dades 1s 1d
• 3.2.4) Generació sortida de resultat de dades 3d
• 3.2.5) Construcció de dashboards i reports de resultats 3d
• 3.3) Lliurament PAC3 3d
4) Lliurament del projecte: Memòria, entregables i presentació 3s
• 4.1) Rectificacions PAC3 2d
• 4.2) Agregació PAC1, PAC2 i PAC3 1d
• 4.3) Lectura i Revisió Memòria 1s
• 4.4) Preparació i grabació presentació 1s
• 4.5) Elaboració abstract 1d
• 4.6) Lliurament Memòria, entregables i presentació 1d

Anàlisis i disseny

La arquitectura que es desenvoluparà al projecte es basa en un disseny de sistema modular, el qual haurà de funcionar en un període continu de 24x7 sense intervenció humana.

El sistema utilitzarà dades estructurades provinents del sistema de events, on es registra l'activitat de l'usuari en les accions que realitza en l'app. Aquest adoptarà una aproximació amb un únic model de dades per poder, de forma independent, generar decisions i mostrar-les.

A continuació es mostra un diagrama que il·lustra la arquitectura del projecte que s'implementarà:



Il·lustració 7 Arquitectura projecte

En el diagrama es pot observar l'estructura del projecte basada en tres parts diferents, cadascuna de les quals té una funció concreta:

- ETL Real-Time: S'observen les diferents fonts de dades com els events de les aplicacions mòbil i la base de dades relacional, que son recopilades per AWS Kinesis.

- Machine Learning: Es on es realitzaran els diversos processos per tal d'identificar i classificar l'activitat dels usuaris segons el seu perfil, en aquest cas ens centrarem en trobar patrons d'usos fraudulents.
- Visualització: Per últim, es crearan una sèrie de "reports" i "dashboards" per tal d'avaluar els resultats obtinguts.

Implementació

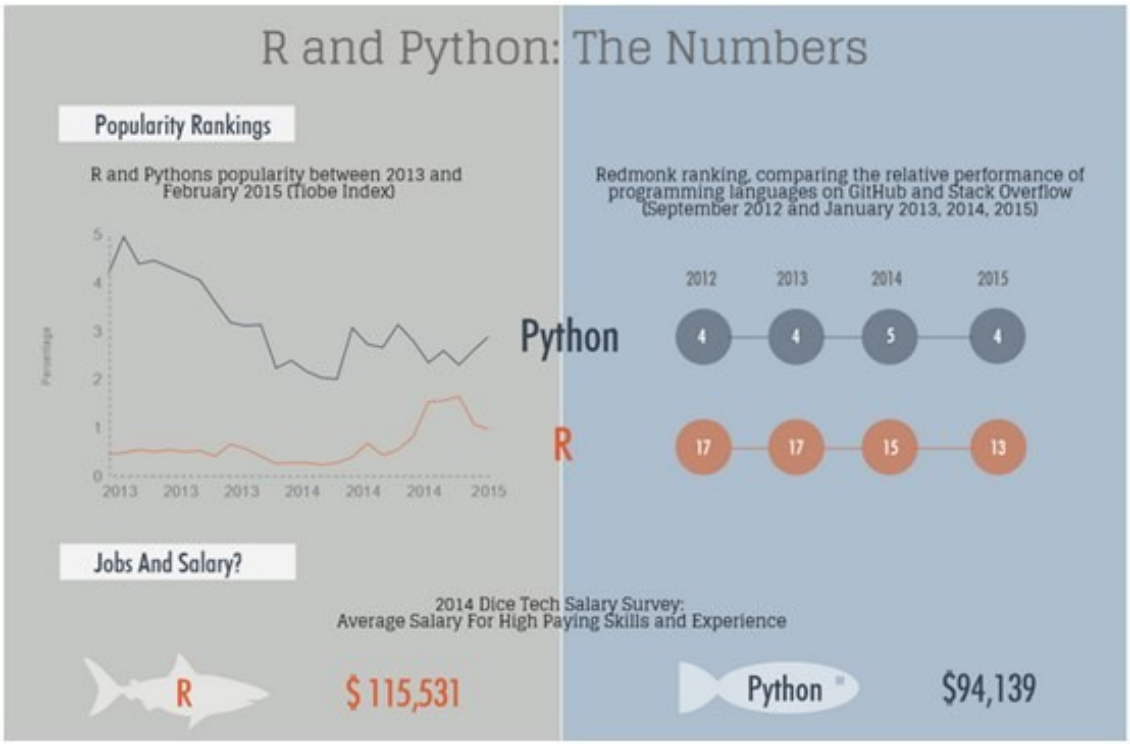
Elecció d'eines

Com s'ha comentat anteriorment, per motius d'integració amb les eines corporatives existeixen una sèrie d'eines que han de ser necessàriament utilitzades.

Llenguatge programació

En primer lloc, a l'empresa, s'opta, des d'un principi, per "Python" coma llenguatge de programació. La principal raó es el suport i comunitat que té al voltant si la comparem amb R.

A demés, quan necessitem fer tasques de "data analysis" que han de ser integrades en un entorn de producció, com es el nostre cas, on s'examinarà en temps real el tràfic de producció, no podem optar per R.

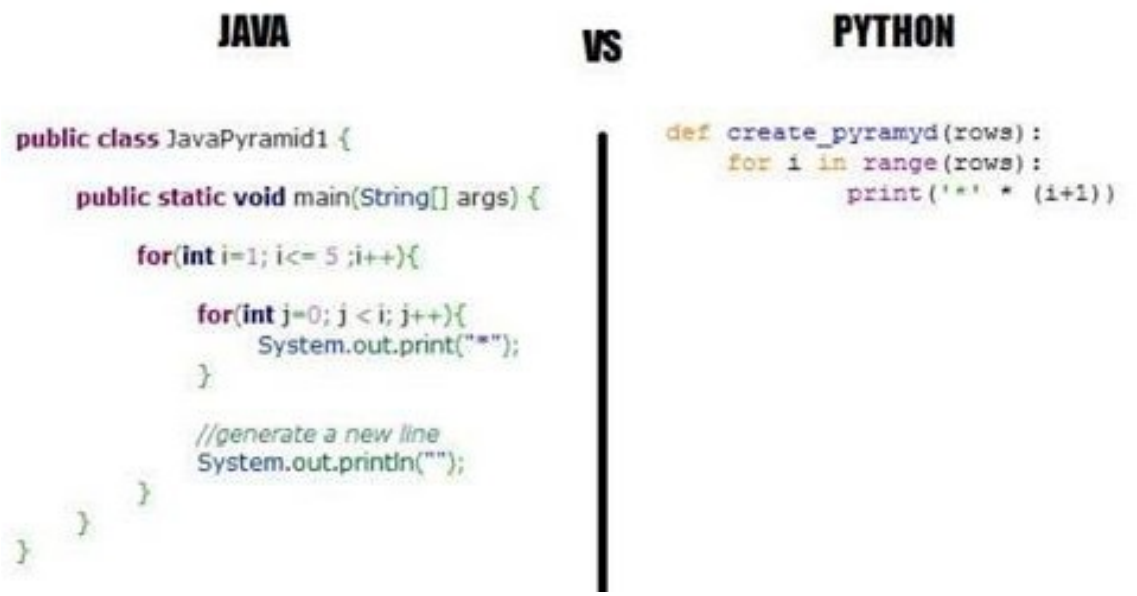


Il·lustració 8 Comparativa Python VS R(Font: KDNUGGETS)

També es van valorar altres llenguatges de programació com Java o Scala. En primer lloc Java cal dir que es un llenguatge més popular i potent però a l'hora de ser utilitzat en tasques de data *processing* Python aporta una sèrie d'eines i *frameworks* que la fan més atractiva.

	Java	*Python*
Execució	Més ràpid d'executar tasques	És més lent
Codificació	Estàtic	Dinàmic
Mètodes de bloqueig	Blocs tradicionals	Utilitza blocs de identació
Simple i compacte	Verbose	Simple i compacte
Llenguatge	Apropiat per implementacions de baix nivell	Python es més adequat com a llenguatge d'enllaç

Taula 1: Comparativa Java VS Python



Il·lustració 9 Java vs Python

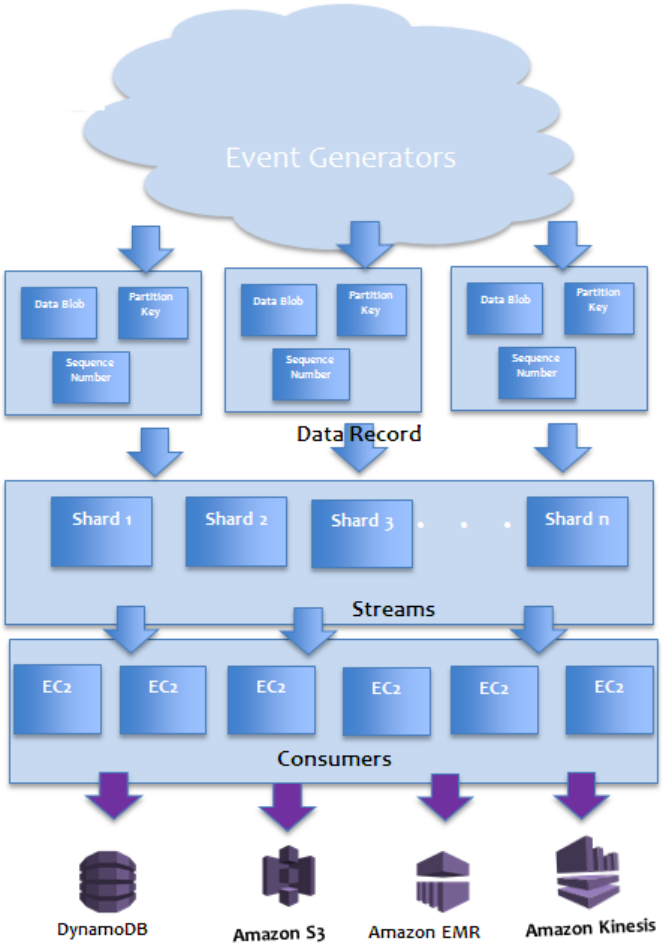
Per últim, tenim Scala, és un llenguatge de programació orientat a objectes purs, es tracta d'un llenguatge funcional on es necessari adaptar-se per tal de poder treballar amb ell. Si el comparem amb Python extraiem els següents punts:

- Python en general es més lent que Scala. Si es té coneixements de processament lògic definitivament Scala es el llenguatge a utilitzar però no existeixen moltes persones amb aquest coneixement.
- Scala es codificació estàtica.
- *Apache Spark* està construït en Scala.
- Les novetats de *Spark* son abans accessibles en *Scala* que en *Python*, ja que aquest està desenvolupat en *Scala*.

En conclusió, tenim que les dues millors opcions actualment són *Scala* i *Python*, el primer es més ràpid però per una altre banda tenim *Pyhton* que té més suport de la comunitat i es

Agregador de dades

Pel que fa al origen de dades utilitzarem “Amazon Kinesis”, aquest sistema es troba actualment en producció i es tracta d’una implementació basada en “Apache Kafka”[4]. Aquest aporta una sèrie d’avantatges a l’hora de ser implementat per simplificar la tasca com a seguretat de la integritat de la informació.



Il·lustració 10 Arquitectura Amazon Kinesis

L’arquitectura de Amazon Kinesis es troba dividida en tres parts:

4 Apache Kafka: es projecte “open source” desenvolupat per la “Apache Software Foundation”, el projecte té com a objectiu proporcionar una eina d’alt rendiment i de baixa latència que permeti manipular fonts de dades en temps real.

- Data Records, contenen la informació de cada event, cadascun d'aquest conté un nombre de seqüencia , la clau de partició i un *blob* amb el contingut de les dades.
    - Seqüència, és creada a Kinesis, consisteix en nombre incremental ordenat de manera que entren en el consumidor de dades.
    - Clau de partició, es un identificador en format *hash*, que determina a quina partició pertanyen les dades.
    - Dades, aquestes es troben contingudes en un sol camp en format blob, aquets no tenen un format particular i poden ser més grans de 50KB.

- Streams, son el nucli del servei de Amazon Kinesis. Les dades son escrites en aquest punt per els events *producers* i son consumits per el consumidor de events. Aquest està compost per un o més particions, les dades son emmagatzemades un màxim de 24 hores.
- Particions, són els objectes on les dades son particionades, cadascun obté la informació a partir d'un rang de *hashes*. Aquest identificador es troba en format 128-bit. Cada partició suporta 5 lectures per segon, amb un màxim de 2MB de dades de per partició. Pel que fa a la escriptura suporta 1000 per segon i un màxim de 1MB per segon.

- Consumidors, son aplicacions que corren sobre AWS EC2 i consisteixen en unes llibreries client que llegeixen les dades de les diferents particions.

- Operacions, a través de la API es poden fer tres accions, afegir dades, llegir dades i per últim redistribuir les particions.

Consumidor de dades

A l'hora d'escollir el "consumer" de les dades, no existeix cap restricció d'eina a ser utilitzada en l'empresa. Per tal d'escollir l'eina més adequada, fem una comparació basada en les característiques de les eines més utilitzades per processar dades en "real-time"

Les eines mes comuns actualment son tres; Spark Streaming, Apache Flink i Apache Storm.

La primera eina que descartem es "Apache Storm" aquesta eina es troba actualment discontinuada i amb poc suport a més de ser molt difícil de controlar l'estat dels processos. Per evitar problemes i futurs inconvenients en l'arquitectura optem per utilitzar altres eines no obsoletes la descartem.

A l'hora d'escollir entre Spark Streaming i Apache Flink va ser una difícil elecció, a continuació es mostra una taula de les principals característiques de cadascuna:

	Spark Streaming	*Apache Flink*
Iteració	Batches	Streaming Architecture
Temps de processament	Pipe Line Execution	Real execution
Model computacional	Micro-batches	Continious Flow
Streaming	Batch processing	Stream processing
Flux de dades	Graf cíclic	Dependència graf cíclic
Latència	Depèn dels batches	Depèn del streaming
Gestió de la memòria	Gestió de la memòria explicita	Gestió de la memòria explicita
Procés iteratiu	Cada iteració ha de ser programada per separat	Només pot tindre una instrucció d'un procés

Taula 2: Comparativa Spark Streaming VS Apache Flink (Font: VILANCE SOLUTIONS)

Una vegada analitzats tots els punts de vista, ens fem la pregunta si necessitem 100% real-time o amb un sistema de micro-baches es suficient. Ja que ambdues eines son

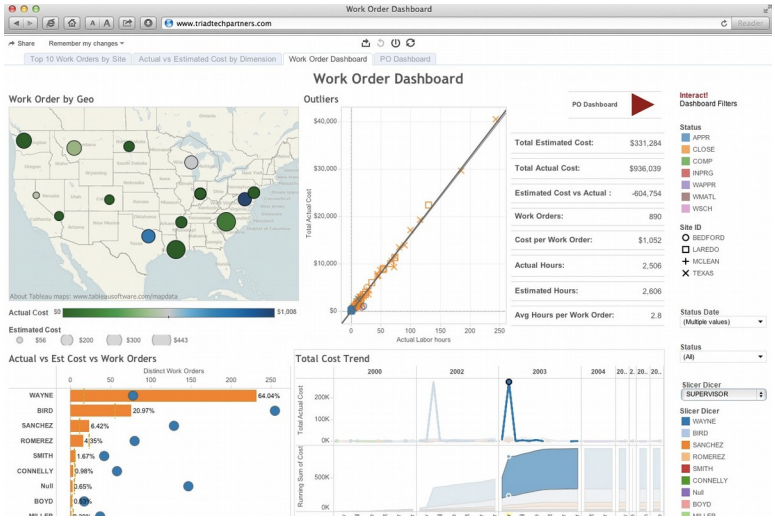
molt similars i es molt difícil decantar-se per alguna de les dues, tenint en compte que Spark té una major comunitat al darrera es va optar per aquesta.

Visualitzador de dades

Looker es la eina corporativa per a la visualització de dades, aquesta utilitza un llenguatge propi anomenat "LookML" basant en "Yaml"[5]. El seu codi es fàcil de gestionar i desenvolupar al tractar-se d'un llenguatge d'estructura de dades.

A qualsevol companyia es difícil prendre la decisió de quines eines utilitzar, en aquest cas es van valorar en el seu moment diferents eines de visualització.

En un principi es va optar per utilitzar "Tableau", aquesta eina permetia poder realitzar individualment a cada usuari els seus dashboards contra una base de dades relacional, però comportava una sèrie de problemes com un alt cost de llicencies i la difícil compartició de les dades. Cal dir que actualment existeix una versió "cloud" per a poder treballar en equip.

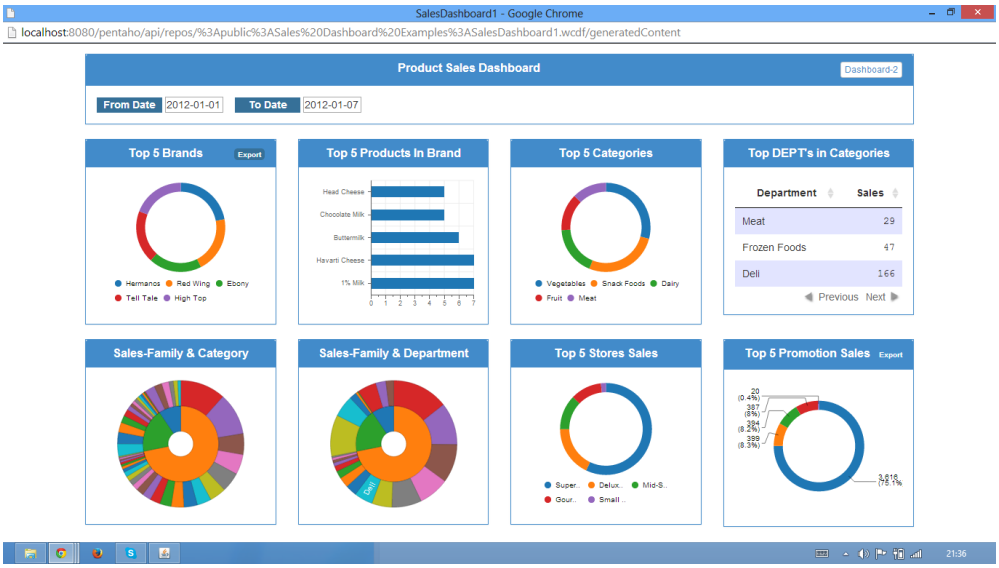


Il·lustració 11 Tableau

5 Yaml: Es un llenguatge de programació creat amb la premissa de poder representar les dades adequadament com una combinació de llistes, "hashes" i valors simples.

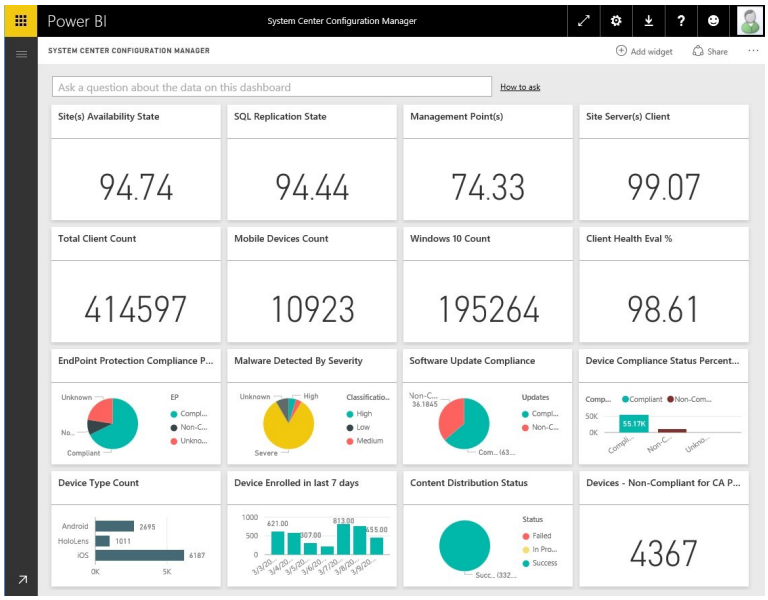
Quan es va valorar quines eines podrien substituir “Tableau” es va pensar en dos tipus, una transversal que servis des de “ETL” fins a visualització o únicament una eina de visualització.

Les eines transversals que es van valorar van ser “Pentaho” i “Microsoft BI”, la primera disposa d’una eina “ETL” senzilla i simple, però a l’hora d’utilitzar la part de visualització, no és bona gràficament, és molt difícil d’utilitzar i per últim no és massa accessible per a treballar en equip.



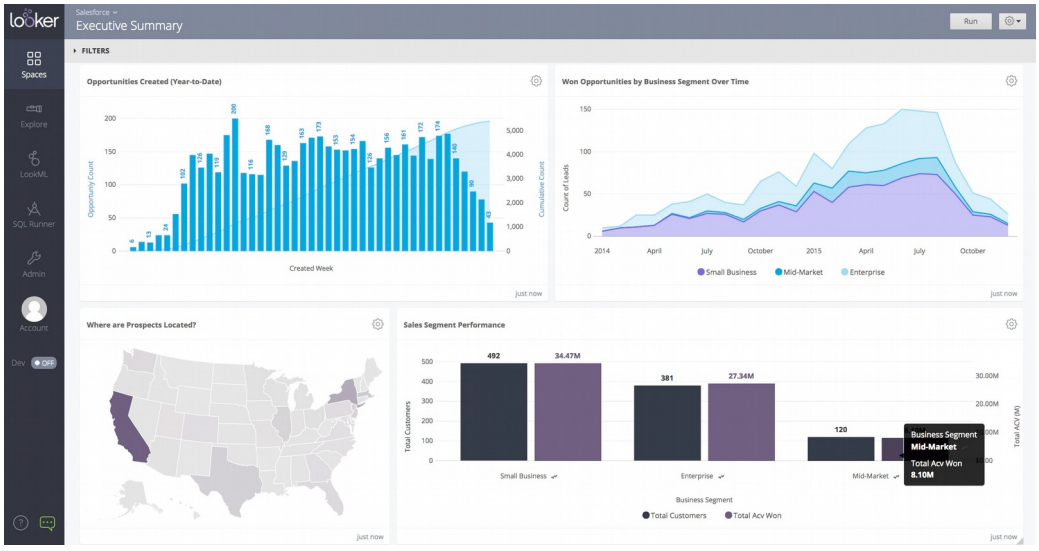
Il·lustració 12 Pentaho

En quant a “Microsoft BI” totes les eines son senzilles d’utilitzar i el fet de poder treballar en una versió modificada de “Visual Studio” fa que sigui molt fàcil la seva adaptació. Per contra, el fet d’haver d’utilitzar com a “datawarehouse” “SQL Server” es convertia en un problema de cost i rendiment. Per tant, es van descartar ambdues opcions.



Il·lustració 13 Microsoft BI

A l'hora de valorar eines de visualització es van agafar dues eines conegudes i amb referencies per altres companyies; "Looker" i "Zoomdata". Ambdues eines complien perfectament les necessitats requerides pel departament de negoci però Looker destacava per una millor presentació de les dades.



Il·lustració 14 Looker

Algorismes

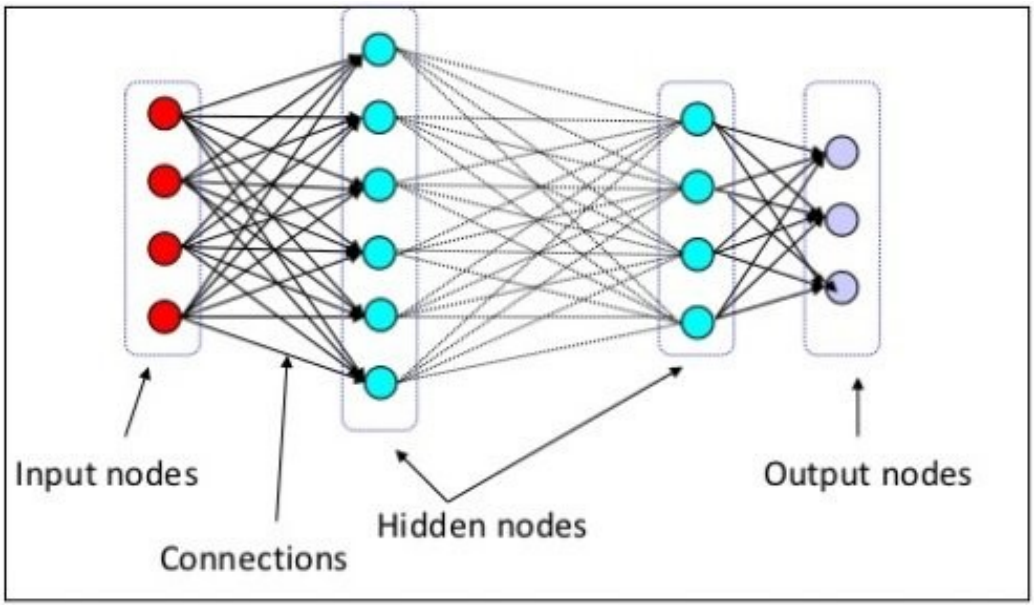
L'algorisme que s'han valorat a l'hora de desenvolupar el projecte es tracta de "Hidden Markov Model" i "Deep Learning".

Deep Learning

En "Deep Learning" s'utilitzen estructures lògiques que s'aproximen en major mesura a l'organització del sistema nerviós, tenint capes d'unitats de procés com a neurones artificials que s'especialitzen en detectar determinades característiques existents en els objectes percebuts.

La visió artificial és una de les àrees on el "Deep Learning" proporciona una millora considerable en comparació amb els algorismes tradicionals. Existeixen diversos entorns i biblioteques on s'executa principalment orientat a servidors amb potents targetes gràfiques o "GPUs".

No deixa de ser una representació apropant-se al sistema nerviós del ser humà permetent dintre d'un sistema global que existeixin xarxes d'unitats de procés que s'especialitzin en la detecció de determinades característiques ocultes en les dades.



Il·lustració 15 Deep Learning

Degut a l'alt cost d'utilitzar maquines dedicades a "GPUs" es descarta utilitzar aquest tipus d'algorismes.

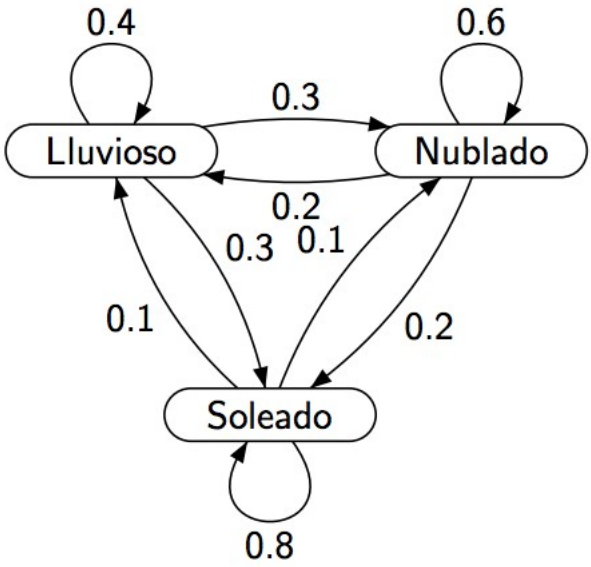
Hidden Markov Model

Finalment es va decidir realitzar el projecte amb "Hidden Markov Model" (HMM) per la seva simplicitat, ja que, es tracta d'una funció de probabilitat d'un model de Markov. El seu primer us va ser en lingüística, modelant les seqüencies de lletres en la literatura russa, on a partir d'aquest ús es va dissenyar com a mètode general.

Actualment, es un dels sistemes estadístics que més s'aplica en els sistemes moderns de reconeixement de la veu, i a dia d'avui segueix sent una de les tècniques més utilitzades per a diverses tasques.

Un procés de Markov es un procés estocàstic que serveix per a representar seqüencies de variables aleatòries no independents entre sí. Es a dir, on la probabilitat del següent estat sobre una seqüencia completa depèn dels estats previs a l'estat actual.

Un exemple per entendre el HMM és la transició entre els canvis del temps. Tenim tres estats diferents; solejat, plujós i ennuvolat. A partir d'un conjunt de dades d'un període construïm una matriu i calculem la probabilitat entre el total de canvis que existeixen en el període.



$$\mathbf{A} = \begin{pmatrix} 0{,}4 & 0{,}3 & 0{,}3 \\ 0{,}2 & 0{,}6 & 0{,}2 \\ 0{,}1 & 0{,}1 & 0{,}8 \end{pmatrix}$$

$$\mathbf{\Pi} = \begin{pmatrix} 0{,}25 & 0{,}25 & 0{,}5 \end{pmatrix}$$

Il·lustració 16 Exemple HMM (Font: PUJ)

Per entendre el nostre model tractaria de les transicions entre accions, per exemple tenim un usuari que puja producte, la primera acció es obrir la app, la segona buscar productes similars al que vol vendre i per últim publicar el producte.

Es tractaria de valorar les diferents probabilitats entre accions i veure comportaments anormal, un exemple de comportament anormal seria publicar un producte sense obrir la app primerament, en aquest cas es tractaria d'algun ordinador simulant activitat i pujant productes possiblement falsos.

Configuració entorn de treball

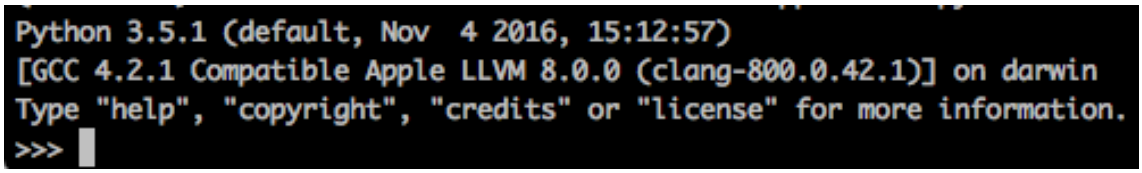
Per a dur a terme el projecte ha sigut necessari instal·lar python 3.5 amb homebrew executant la següent comanda:

```
brew install python3
```

Per comprovar que la instal·lació ha sigut correcte executem la següent comanda:

```
python3
```

En la nostra terminal apareixerà el següent contingut:



Instal·lem a continuació les dues eines que utilitzarem per programar, la primera es *PyCharm* per treballar amb el codi python, la segona es *DataGrip* per executar "querys" en llenguatge "SQL".

D'ambdós programes existeix una versió gratuïta per a estudiants.[6]

Una vegada instal·lat el programari creem el repositori de codi, en el meu cas seguint la estructura interna de la empresa es crearà a Github amb el nom de "FraudDetection". Una vegada creat executem la següent comanda que crearà el repositori a la nostra maquina introduint la credencial personal.

```
git clone https://github.com/Wallapop/FraudDetection.git
```

Quan es desenvolupa "software" amb python, es molt comú utilitzar diferents versions d'un mateix paquet. En un punt tenim una versió instal·lada que no pot ser eliminada per una versió nova, necessitem disposar d'accés a ambdues llibreries per tal de poder desenvolupar dos projectes de forma simultània, la solució consisteix en crear entorns virtual. Un entorn virtual es un espai completament independent d'altres entorn virtuals i d'altres paquets instal·lats globalment al sistema.

```
/usr/local/bin/pythin3 -m venv fraud_env
```

Una vegada creada l'activem per tal de poder instal·lar els paquets necessaris i executar el codi en aquest entorn virtual.

```
source fraud_env/bin/activate
```

6 https://www.jetbrains.com/student/

Construcció model de dades i exportació

Les dades d'entrada al projecte per entrenar el model es divideix en dos fitxers:

- Conjunt d'usuaris bons, a partir de la informació que conté la base de dades s'extrau un conjunt d'usuaris que actualment son actius amb una activitat correcta.

- Conjunt d'usuaris dolents, dades d'usuaris que per diverses raons han sigut bloquejats de poder fer ús de l'aplicació.

Per motius de confidencialitat i privadesa, el conjunt de dades d'entrada per realitzar l'entrenament no s'inclou al projecte. Aquestes dades tenen la següent estructura:

event_ai	Identificador del event que realitza l'usuari
event_created	Data de creació de l'acció del usuari
dvce_access_token_id	Identificador únic del usuari en un dispositiu

Taula 2: Descripció camps de les dades

Ens disposem a extraure les dades per analitzar-les i utilitzar-les en l'entrenament del model. Obrim el *Datagrip* i executem les dues "querys" que apareixen en l'annex I i II i exportem les dades.

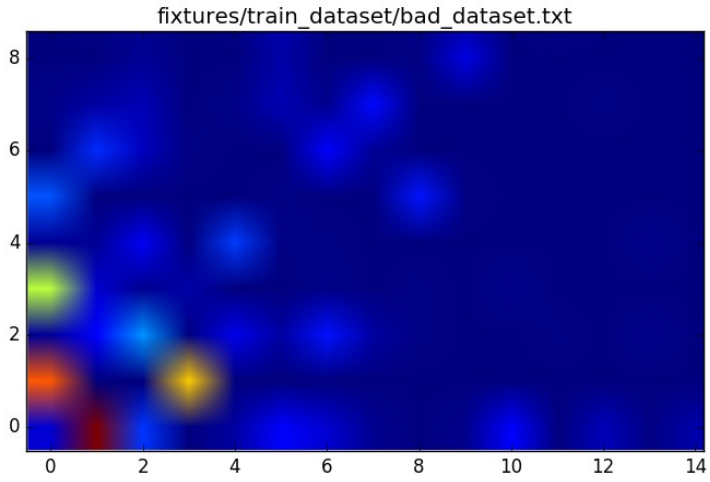
A continuació es mostra un "preview" de les dades extretes:

```
2,2016-11-08 14:36:43.667000,39970
0,2016-11-08 14:37:50.798000,39970
6,2016-11-08 14:38:50.816000,39970
6,2016-11-08 14:39:57.990000,39970
6,2016-11-08 14:45:33.470000,39970
3,2016-11-08 14:46:35.650000,39970
1,2016-11-08 14:46:35.650000,39970
0,2016-11-08 14:46:36.526000,39970
1,2016-11-08 14:47:20.099000,39970
3,2016-11-08 14:47:20.100000,39970
```

Anàlisis de dades

Tornem al *Pycharm* i programem el codi per tal de llegir les dades i representar-les, com es veu a l'annex III. Per tal de realitzar les tasques necessàries instal·lem tres paquets, "pickle" que ens permetrà emmagatzemar les dades que obtindrem en aplicar la anàlisis, "numpy" per a generar la matriu de transicions i "matplotlib" per representar la informació.

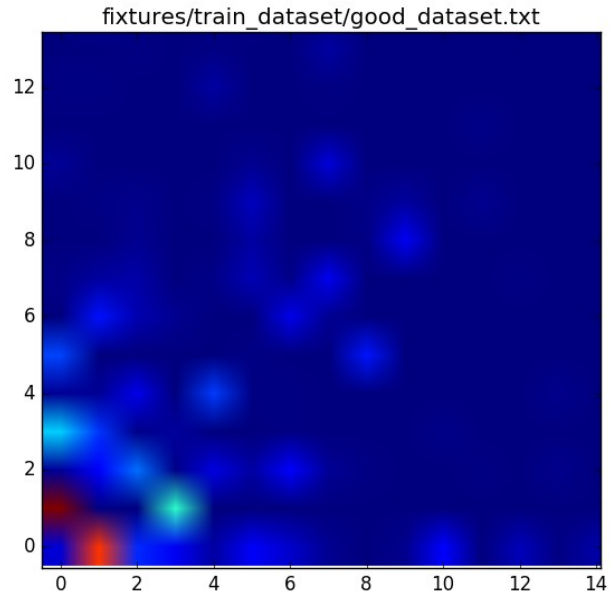
Per tal de poder centrar l'activitat de l'usuari per volum d'events, els identifiquem per orde de major a menor activitat, com es vot veure en ambdues gràfiques el conjunt de dades es més visible en valors propers a 0.



Il·lustració 17 Matriu usuaris bons

Com es veu l'activitat queda centrada en els 20 primers events de la matriu, en quan a la matriu d'usuaris fraudulents es pot veure que es molt similar a la dels usuaris bons a simple vista:

Ens enfoquem en el conjunt més comú, com es pot veure en les dues pròximes imatges son molt similars però amb petites diferencies, aquestes diferencies ens permetran identificar activitats anormals i comportaments no comuns. A més de obviar els patrons d'activitat més comuns entre els conjunts per quedar-nos amb les dades que permetin identificar correctament els usuaris.



Il·lustració 18
Ampliació matriu usuaris bons

Instal·lació i Configuració de Spark Streaming

Per a realitzar el projecte ha sigut necessari instal·lar Spark Streaming per tal de poder consumir les dades que envien els dispositius mòbils a AWS Kinesis.

Primerament descarreguem Spark del següent enllaç:

```
http://spark.apache.org/downloads.html
```

Escollim la versió de “Spark 2.0.0” i el tipus de paquet “Pre-built for Hadoop 2.7 and later”

Descarreguem el fitxer “spark-2.0.0-bin-hadoop2.7.tgz” i el descomprimim.

Una vegada tenim el directori, el movem de les descarregues “~/Downloads/spark-2.0.0-bin-hadoop2.7/” a el següent directori “/usr/local/spark-2.0.0/”

Editem el fitxer “~/.bash_profile” com a administrador i introduïm la següent línia:

```
export SPARK_HOME=/usr/local/spark-2.0.0
```

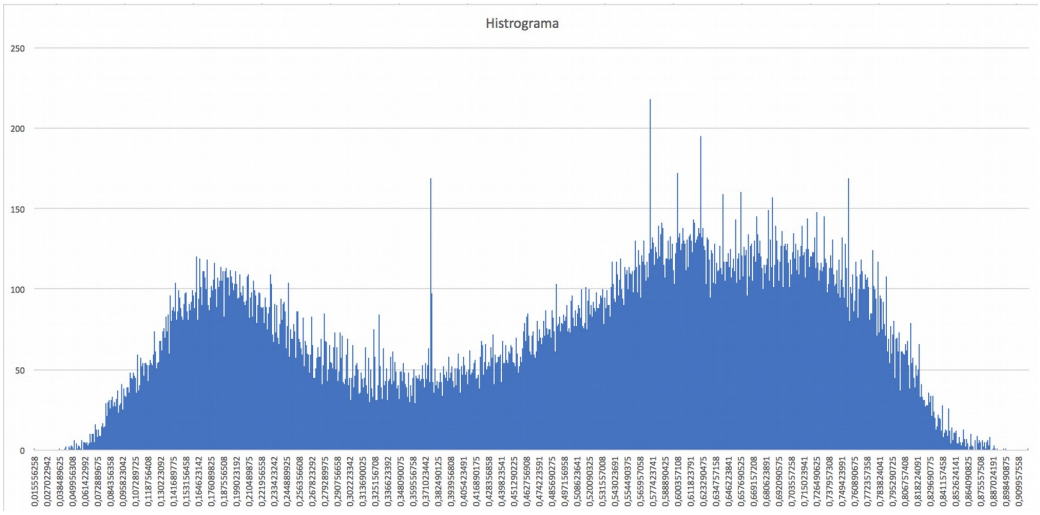
Per tal de poder executar el codi necessitem descarregar el fitxer “jar” per tal de poder integrar “Spark Streaming” amb “AWS Kinesis”.

```
https://mvnrepository.com/artifact/org.apache.spark/spark-streaming-kinesis-asl_2.10/2.0.0
```

Entrenament de dades

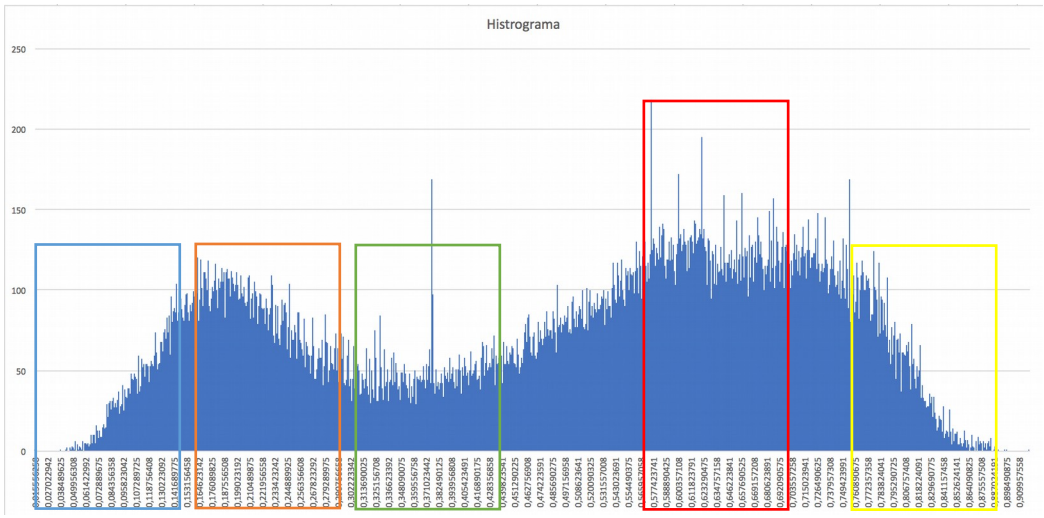
A l’hora de començar a entrenar les dades es quan es veu clarament la qualitat de la informació extreta dels events del usuaris. Es en aquest punt quan podem identificar els patrons més comuns i distingir el diferents conjunts.

El primer pas que realitzem es unir les dades d’usuaris bons i dolents i realitzem un histograma per tal de veure gausianament el contingut de l’activitat.



Il·lustració 19 Resultat matriu en representació gaussiana

Com es pot veure a la imatge hi ha dues campanes, cal investigar manualment quin tipus d’activitat hi ha a cada punt, per tant ens centrem en cinc punts diferents.



Il·lustració 20 Resultat matriu en representació gaussiana dividida en parts

El primer (blau) conjunt destaca per activitat fraudulenta manual, poques accions però sense sentit en les transicions. El segon (taronja) i el tercer (verd) conjunt es tracta d'usuaris puntuals de tot tipus com poden ser compradors, venedors o ambdues. El quart (vernell) conjunt es tracta d'usuaris molt actius que realitzen activitat molt continuadament amb moltes accions. Per últim, el cinquè (groc) conjunt es tracta d'usuaris fraudulents amb molta activitat, per exemple enviadors massius de missatges, scrappers de catàleg, etc.

Per últim com a sortida del fitxer es genera la matriu de transicions amb els pesos corresponents per tal de poder executar "Spark Streaming" amb les dades, el codi realitzat es troba en l'annex IV.

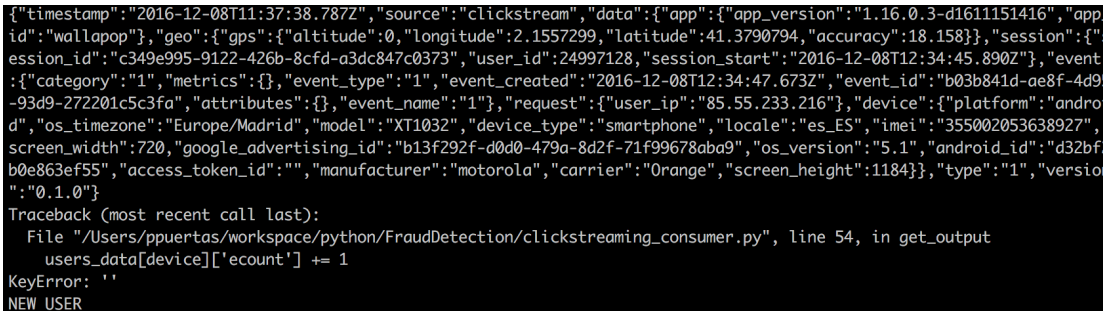
Execució procés

Per últim, necessitem provar el correcte funcionament executant la següent comanda fent referencia al fitxer "jar" descarregat anteriorment i al codi que es troba en l'annex V:

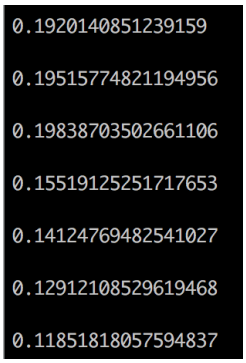
```
./bin/spark-submit --jars
/Users/ppuertas/workspace/python/FraudDetection/jars/spark-streaming-
kinesis-asl-assembly_2.10-2.0.0.jar
```

```
/Users/ppuertas/workspace/python/FraudDetection/clickstreaming_consumer.py
```

Una vegada executem la següent comanda veiem per pantalla l'activitat que va entrant d'usuaris:



Per a poder valorar l'activitat d'un usuari decidim que a partir de tres events es comença a verificar la seva activitat, de moment considerem un valor a partir del qual ja comença a ser una sessió més complexa. A continuació es mostra l'activitat d'un usuari i els valors de transició de la matriu:



Visualització de dades

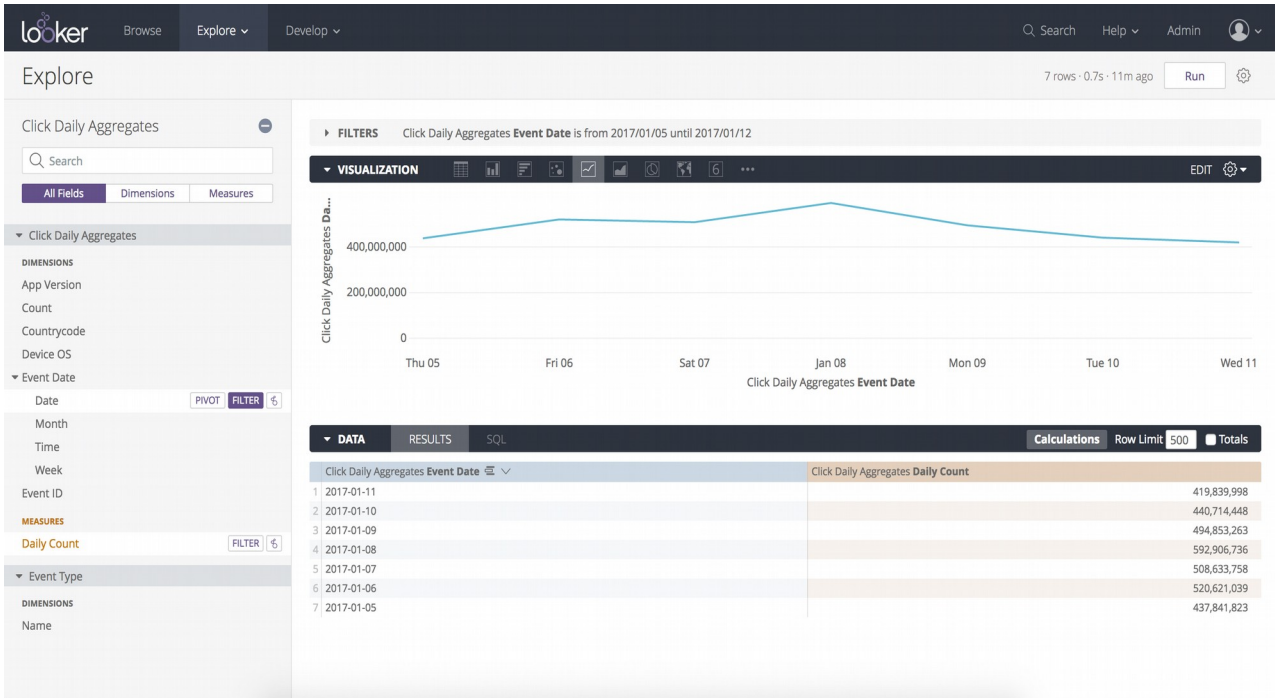
Una vegada posat el projecte a producció es necessari veure com està funcionant, per tal de visualitzar la informació es crearan tres reports, el primer tractarà del events consumits per l'eina, el segon tractarà de com està marcant els usuaris on després existeix una verificació manual i classificació per tal d'entendre el comportament i per últim el grups on s'estan classificant.

Totes aquestes dades a més d'existir els corresponents reports també seran agrupades en un dashbord general per tal de poder comprovar el estat general de totes les mesures i dades obtingudes.

Per utilitzar l'eina de Looker no fa falta instal·lar ni configurar cap màquina, es un servei web el qual es connecta al nostre *datawarehouse* via SQL. L'Access es via web i es compatible amb qualsevol navegador web comú.

Report Events

En aquest primer report, es pot consultar el nombre d'events que està processant l'eina per dia, a més de les dades que s'envia a la matriu de transicions s'ha afegit informació corresponent al dispositiu com versió de l'aplicació, sistema operatiu, país i el nom de l'event.

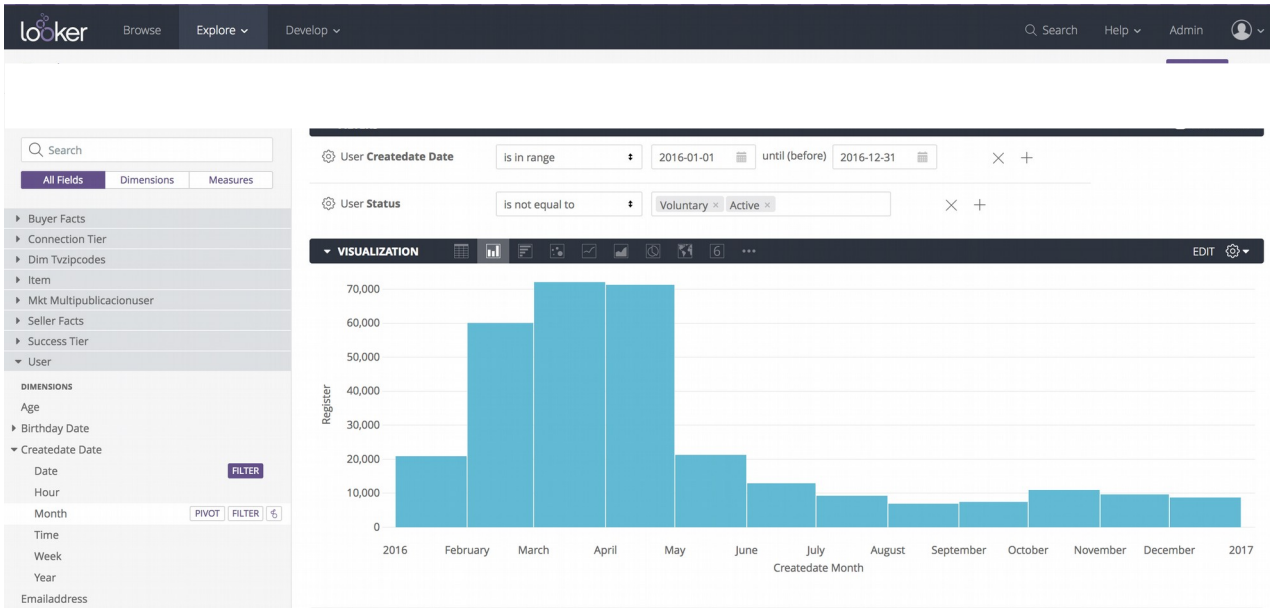


Il·lustració 21 Report Events

Report Usuaris

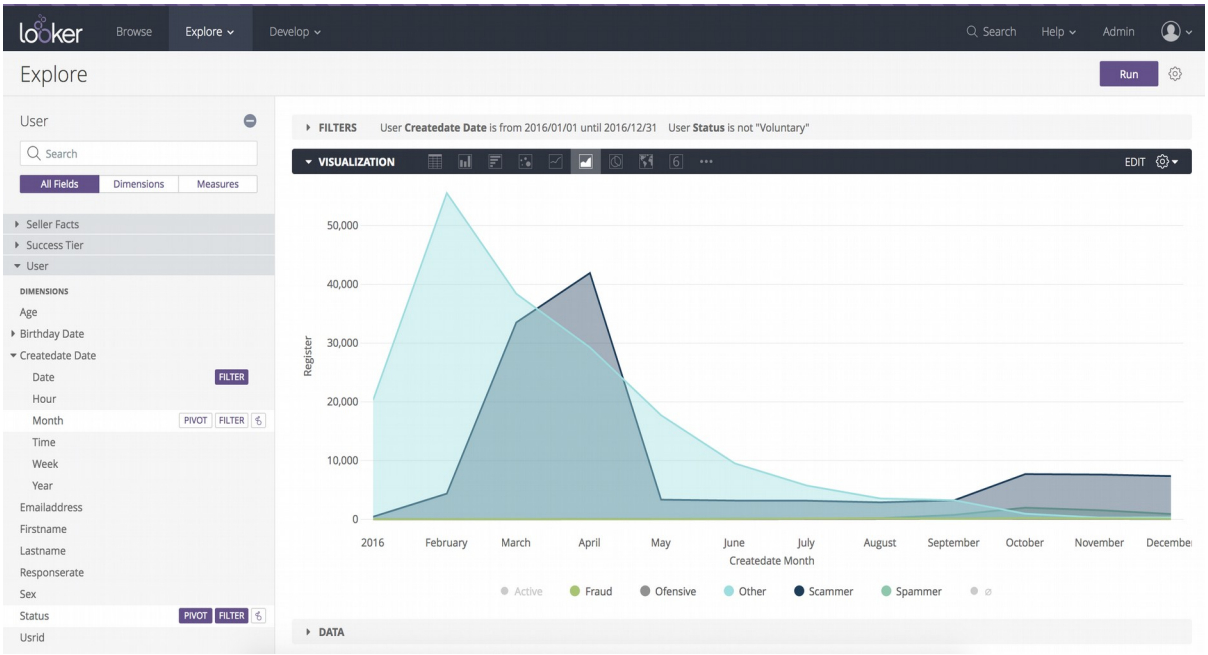
L'objectiu d'aquest report es poder veure el nombre d'usuaris que s'està bloquejant en l'aplicació, així controlar les accions que s'estan duent a terme.

Il·lustració 22 Rerport Usuaris



Report Grups

Aquest report mostrarà per data de registre els conjunts d'usuaris per tipologia de bloqueig. A partir de les dades recollides amb la matriu de transicions es prebloqueja l'usuari on existeix una revisió manual, a partir d'aquesta decisió manual veiem els diferents tipus d'activitat. Com es pot veure en la gràfica antigament no existia un tipus de bloqueig, i amb la informació obtinguda del procés permet visualitzar i poder classificar el usuaris.



Il·lustració 23 Report Grups

Dashboard

Per últim, centralitzarem tota la informació generada per tal de veure-la conjuntament, per una banda tenim el report d'events que mostra tota la informació recollida per activitat, en segon lloc els usuaris i el nombre d'usuaris bloquejats i per últim els diferents conjunt d'usuaris bloquejats, classificats per el tipus d'activitat fraudulenta realitzada.


looker
Browse
Explore
Develop
Search
Help
Admin
Pedro Puertas
[TFG] Dashboard
3m ago
Edit
[TFG] Events
Click Daily Aggregates Daily Count
400,000,000
200,000,000
0
Thu 05
Fri 06
Sat 07
Jan 08
Mon 09
Tue 10
Wed 11
Click Daily Aggregates Event Date
[TFG] Grups
Register
50,000
40,000
30,000
20,000
10,000
0
2016
February
March
April
May
June
July
August
September
October
November
December
Createdate Month
Active
Fraud
Offensive
Other
Scammer
Spammer
[TFG] Usuaris
Register
60,000
40,000
20,000
0
2016
February
March
April
May
June
July
August
September
October
November
December
2017
Createdate Month

Conclusions

Després de la finalització d'aquest projecte, valoro molt positivament l'experiència de poder crear una solució tècnica completa. No es la primera vegada que haig d'afrontar un projecte d'aquestes característiques i dimensions, però sempre es gratificant treballar amb eines noves i potents.

Per altra banda, el temps disponible en el semestre per la implementació ha estat reduït i, en conseqüència, hi ha hagut una sèrie d'enrederaments en les entregues. Tot i això, el programari obtingut s'ha finalitzat correctament i es troba en producció actualment.

Es pot assegurar que els objectius que es plantejaven en un principi del projecte han estat assolits. En primer lloc, s'ha obtingut una aplicació que funciona perfectament i que serveix per poder millor l'experiència del usuari en l'aplicació.

En segon lloc, s'ha demostrar els coneixements adquirits al llarg de la meva carrera professional, més concretament on m'ha donat la possibilitat de treballar amb diverses eines i clients.

I per últim en tercer lloc, poder posar en pràctica alguns dels coneixements adquirits durant els anys que porto aprenent coses a la UOC.

En un futur em semblaria correcte que la UOC introduís més temari o més assignatures referents al Business Intelligence, Big Data i Machine Learning, ja que donen molta sortida professional i són molt valorats els coneixements en aquestes temàtiques.

Podríem dir que el Big Data es el futur a totes les empreses i calen persones formades per a ser capaces de solucionar tots aquets reptes, els volums de dades no paren de créixer i es necessari saber gestionar aquesta informació i utilitzar-la correctament.

En general, ha sigut una experiència molt positiva i espero que totes les persones que llegeixen aquest document els hi hegui agradat.

Confidencialitat

Per temes de confidencialitat hi ha parts que no s'ha pogut mostrar en aquest document, pel que fa al codi generat es troba en la seva totalitat. Únicament la part no s'ha pogut mostrar son les dades que s'han analitzat i el resultat de les matrius de transicions ja que es tracta d'informació personal i privada dels usuaris. Tampoc es podrà accedir a l'eina de visualització de dades ja que comporta també veure informació personal dels usuaris.

Bibliografia

https://aws.amazon.com/es/kinesis/

https://aws.amazon.com/es/rds/mysql/

https://github.com/spotify/luigi

https://github.com/

https://www.jetbrains.com/pycharm/

https://www.jetbrains.com/datagrip/

https://spark.apache.org/streaming/

https://www.microsoft.com/en-us/research/wp-content/uploads/2013/11/aisec10-leontjeva.pdf

https://www.datavisor.com/blog/

http://www.kdnuggets.com/2015/05/r-vs-python-data-science.html

http://valiancesolutions.com/2016/04/26/apache-flink-vs-apache-spark/

https://es.wikipedia.org/wiki/YAML

https://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov

https://es.wikipedia.org/wiki/Aprendizaje_profundo

http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:bfc_sesion3_2008-2.pdf

http://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial

http://www.northware.mx/desarrollo-en-cascada-waterfall-vs-desarrollo-agile-scrum/

http://blog.nubelo.com/proceso-del-desarrollo-software/

http://www.rhpware.com/2014/02/amazon-kinesis-analisis-de-datos.html

https://www.linkedin.com/pulse/java-python-r-scala-many-languages-big-data-speaks-debajani

https://es.wikipedia.org/wiki/Scala_(lenguaje_de_programaci%C3%B3n)

http://cloudacademy.com/blog/amazon-kinesis-real-time-event-processing/

http://www.matematicasdigitales.com/la-herramienta-que-todos-quieren-la-campana-de-gauss/

Annex I

Extracció de dades per l'entrenament amb usuaris fraudulents.

```
SELECT b.event_ai
, a.event_created
, a.dvce_access_token_id
FROM clickstreaming.events a
LEFT JOIN
( SELECT (ROW_NUMBER() OVER ( ORDER BY sumevents DESC)-1) AS event_ai
, event_name
FROM ( SELECT event_name, count(*) AS sumevents
      FROM clickstreaming.events
      WHERE event_name NOT IN (208,85)
      GROUP BY 1)
) b ON a.event_name = b.event_name
WHERE a.dvce_access_token_id IN (SELECT DISTINCT(c.accesstokenid)
                        FROM ods.accesshistory c
                        WHERE c.usrid NOT IN (SELECT usrid
                                    FROM ods.usr_usr
                                    WHERE status IN (1,2,3,4,5))
                         )
 AND a.event_name NOT IN (208,85)
ORDER BY 3,2;
```

Annex II

Extracció de dades per l'entrenament amb usuaris bons.

```
SELECT b.event_ai
, a.event_created
, a.dvce_access_token_id
FROM clickstreaming.events a
LEFT JOIN
( SELECT (ROW_NUMBER() OVER ( ORDER BY sumevents DESC)-1) AS event_ai
, event_name
FROM ( SELECT event_name, count(*) AS sumevents
      FROM clickstreaming.events
      WHERE event_name NOT IN (208,85)
      GROUP BY 1)
) b ON a.event_name = b.event_name
WHERE a.dvce_access_token_id IN (SELECT DISTINCT(c.accesstokenid)
                        FROM ods.accesshistory c
                        WHERE c.usrid NOT IN (SELECT usrid
                                    FROM ods.usr_usr
                                    WHERE status NOT IN (1,2,3,4,5))
                         )
 AND a.event_name NOT IN (208,85)
ORDER BY 3,2;
```

Annex III

```
# -*- coding: utf-8; -*-
"""
  Contains the analysis of the data for the train of the hidden markov model

  @author: pedro.puertas@wallapop.com (Pedro Puertas)
  @copyright: Wallapop (c) 2017
"""

import pickle
import numpy as np
import matplotlib.pyplot as plt


__author__ = 'pedro.puertas@wallapop.com (Pedro Puertas)'

bad_examples_file = 'fixtures/train_dataset/bad_dataset.txt'
good_examples_file = 'fixtures/train_dataset/good_dataset.txt'
pickle_file_bad = "fixtures/train_dataset/analysis/bad_dataset.pickle"
pickle_file_good = "fixtures/train_dataset/analysis/good_dataset.pickle"
dict_example_files = {bad_examples_file: pickle_file_bad, good_examples_file: pickle_file_good}

for example_file in dict_example_files.keys():

    from_event = None
    to_event = None
    current_device_token = None
    num_transitions = 0
    max_event = 0
    pickle_file = dict_example_files[example_file]
    use_pickle = True

    # Pickle is the file where is save it the data to avoid load in every execution
    if use_pickle:
        transition_matrix = pickle.load(open(pickle_file, 'rb'))
    else:
        with open(example_file, 'r') as fp:
            lines = fp.readlines()
            for line in lines:
                to_event, timestamp, device_token = line[:-1].split(',')
                if int(to_event) > max_event:
                    max_event = int(to_event)

            # Create matrix
            max_event += 1
            matrix = np.zeros((max_event, max_event), dtype=float)
            transition_matrix = np.zeros((max_event, max_event), dtype=float)

            # Read the file
            for line in lines:
                to_event, timestamp, device_token = line[:-1].split(',')

                if not (from_event is None or current_device_token != device_token):

                    #   Fill transition matrix
                    matrix[from_event, to_event] += 1
                    num_transitions += 1
```

```
            from_event = to_event
            current_device_token = device_token

    transition_matrix = matrix / num_transitions

if not use_pickle:
    pickle.dump(transition_matrix, open(pickle_file, 'wb'))

if np.sum(transition_matrix).astype(np.float32) != 1.0:
    print('Examples Matrix Error')
else:

    # Draw the content of the matrix
    plt.imshow(transition_matrix, origin='lower')
    plt.title(example_file)
    plt.show()
```

Annex IV

```
# -*- coding: utf-8; -*-
"""
  Contains detect train

  @author: pedro.puertas@wallapop.com (Pedro Puertas)
  @copyright: Wallapop (c) 2017
"""

import inspect
import math
import os
import pickle

__author__ = 'pedro.puertas@wallapop.com (Pedro Puertas)'

max_event = 336
event_master_table = 'fixtures/train_dataset/event_mastertable.csv'
training_data = 'fixtures/train_dataset/data.csv'


def get_package_path():
    file_path = os.path.dirname(os.path.abspath(inspect.getfile(inspect.currentframe())))
    file_path = file_path.split('/')
    source_path = '/'.join(file_path) + '/'
    return source_path


def event_pairs(list_event):
    for start in range(0, len(list_event) - 1):
        yield list_event[start], list_event[start + 1]


def train():

    transition_matrix = [[10 for i in range(max_event)] for i in range(max_event)]

    with open(get_package_path() + training_data, 'r') as fp:

        current_user_events = []
        current_user = None

        for line in fp.readlines():

            to_event, new_user = line[:-1].strip().split(',')

            if not current_user or current_user == new_user:
                current_user_events.append(int(to_event))
            else:
                finalize_user(current_user_events, transition_matrix)
                current_user_events = [int(to_event)]

            current_user = new_user

    for i, row in enumerate(transition_matrix):
```

```
        s = float(sum(row))
        for j in range(len(row)):
            row[j] = math.log(row[j] / s)

    pickle.dump({'mat': transition_matrix}, open('/tmp/gib_model.pki', 'wb'))

    print("DONE TRAINING")
    histogram = open('/tmp/histogram_goods.csv', 'w+')

    with open(get_package_path() + training_data, 'r') as fp:

        current_user_events = []
        current_user = None

        for line in fp.readlines():

            to_event, new_user = line[:-1].strip().split(',')

            if not current_user or current_user == new_user:
                current_user_events.append(int(to_event))
            else:
                if len(current_user_events) > 10:
                    histogram.write(
                        "{}\n".format(avg_transition_prob(current_user_events, transition_matrix)).replace('.',
',',))
                current_user_events = [int(to_event)]

            current_user = new_user

    histogram.close()


def finalize_user(event_list, transition_matrix):
    try:
        for a, b in event_pairs(event_list):
            transition_matrix[int(a)][int(b)] += 1
    except Exception as e:
        raise e


def avg_transition_prob(list_events, log_prob_mat):
    log_prob = 0.0
    transition_ct = 0
    for a, b in event_pairs(list_events):
        log_prob += log_prob_mat[int(a)][int(b)]
        transition_ct += 1
    return math.exp(log_prob / (transition_ct or 1))


if __name__ == '__main__':
    train()
```

Annex V

```
# -*- coding: utf-8; -*-
"""
  Contains Spark Streaming consumer from AWS Kinesis (Clickstreaming)
```

```
  @author: pedro.puertas@wallapop.com (Pedro Puertas)
  @copyright: Wallapop (c) 2017
"""

from __future__ import absolute_import, print_function

import inspect
import os
import pickle
import json
import math
import traceback

import sys
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark.streaming.kinesis import KinesisUtils, InitialPositionInStream

__author__ = 'pedro.puertas@wallapop.com (Pedro Puertas)'

aws_access_key_id = ''
aws_secret_access_key = ''
pickle_file = '/tmp/gib_model.pki'
event_master_table = 'fixtures/train_dataset/event_mastertable.csv'

transition_matrix = pickle.load(open(pickle_file, 'rb'))['mat']
MIN_EVENTS = 3
users_data = {}


def get_package_path():
    file_path = os.path.dirname(os.path.abspath(inspect.getfile(inspect.currentframe())))
    file_path = file_path.split('/')
    source_path = '/'.join(file_path) + '/'
    return source_path


def get_output(_, rdd):
    for data in rdd.collect():
        cs_data = json.loads(data)

        try:

            device = cs_data["data"]["device"]["access_token_id"]

            try:
                event = int(cs_data["data"]["event"]["event_type"])
            except:
                event = int(cs_data["data"]["event"]["event_name"])
            users_data[device]['ecount'] += 1
            users_data[device]['accum_prob'] += transition_matrix[int(users_data[device]['last'])][event]
            transition_matrix[int(users_data[device]['last'])][event]

            if users_data[device]['ecount'] > MIN_EVENTS:
                print(device)
                print(math.exp(users_data[device]['accum_prob'] / users_data[device]['ecount']))

            users_data[device]['last'] = event
```

```
        except KeyError as e:
            print(data)
            traceback.print_exc(file=sys.stdout)
            print("NEW USER")
            users_data[device] = {}
            users_data[device]['ecount'] = 1
            users_data[device]['accum_prob'] = 0
            users_data[device]['last'] = event
        except OverflowError as e:
            print("ERROR MATH OVERFLOW")
            print(users_data[device]['ecount'])
            print(users_data[device]['accum_prob'])
        except Exception:
            print("OTHER EXCEPTION")
            traceback.print_exc(file=sys.stdout)
            print(data)
            print(users_data[device]['last'])
            print(event)
            print(len(transition_matrix))
            print(len(transition_matrix[0]))


def main():
    # Create a local StreamingContext with two working thread and batch interval of 10 seconds
    sc = SparkContext('local[2]', 'Stream Consumer App')
    ssc = StreamingContext(sc, 10)

    try:
        stream = KinesisUtils.createStream(ssc, 'StreamConsumer', 'Clickstream',
                                           'kinesis.eu-west-1.amazonaws.com', 'eu-west-1',
                                           InitialPositionInStream.LATEST, 10,
                                           awsAccessKeyId=aws_access_key_id,
                                           awsSecretKey=aws_secret_access_key)

        stream.foreachRDD(get_output)
        ssc.start()  # Start the computation
        ssc.awaitTermination()  # Wait for the computation to terminate
    except Exception as e:
        print('Exception ==================================> %s' % (str(e)))


if __name__ == '__main__':
    main()
```

Annex VI

```
# Wallapop Looker View
# @author: pedro.puertas@wallapop.com (Pedro Puertas)
# @copyright: Wallapop (c) 2017
# Events


- view: click_daily_aggregates
  sql_table_name: clickstreaming.events
  fields:

  - dimension: count
    type: number
    sql: ${TABLE}.count

  - dimension_group: event
    type: time
    timeframes: [time, date, week, month]
    sql: ${TABLE}.event_date

  - dimension: event_id
    type: number
    # hidden: true
    sql: ${TABLE}.event_id

  - dimension: device_os
    # hidden: true
    sql: ${TABLE}.deviceos

  - dimension: app_version
    # hidden: true
    sql: ${TABLE}.appversion

  - dimension: countrycode
    # hidden: true
    sql: ${TABLE}.countrycode

### MEASURES ###

  - measure: daily_count
    type: sum
    sql: ${TABLE}.count
```

Annex VII

```
# Wallapop Looker View
# @author: pedro.puertas@wallapop.com (Pedro Puertas)
# @copyright: Wallapop (c) 2017
# Event Name Dimension

- view: event_type
  sql_table_name: clickstreaming.event_name

  fields:

  - dimension: id
    primary_key: true
    hidden: true
    sql: ${TABLE}.id

  - dimension: name
    sql: ${TABLE}.name
```

Annex VIII

```
# Wallapop Looker View
# @author: pedro.puertas@wallapop.com (Pedro Puertas)
# @copyright: Wallapop (c) 2017
# User

- view: user
  sql_table_name: dlv.user

  fields:

  - dimension_group: birthday
    type: time
    timeframes: [time, date, week, month, year]
    sql: ${TABLE}.birthday

  - dimension: age
    type: number
    sql:  CASE
            WHEN ${TABLE}.birthday::date='1930-01-01' THEN 0
            WHEN ${TABLE}.birthday::date='1950-01-01' THEN 0
            WHEN ${TABLE}.birthday::date='1970-01-01' THEN 0
            WHEN ${TABLE}.birthday::date<'1915-01-01' THEN 0
            WHEN ${TABLE}.birthday::date>GETDATE() THEN 0
          ELSE DATEDIFF(year,${TABLE}.birthday,GETDATE())
          END

  - dimension_group: createdate
    type: time
    timeframes: [time, hour, date, week, month, year]
    sql: ${TABLE}.createdate

  - dimension: sexid
    hidden: true
    type: number
    sql: ${TABLE}.sexid

  - dimension: sex
    sql: DECODE(${sexid},1,'M',2,'F','U')

  - dimension: status
    type: string
    sql: CASE ${TABLE}.status
         WHEN 0 THEN 'Active'
         WHEN 1 THEN 'Scammer'
         WHEN 2 THEN 'Spammer'
```

```
        WHEN 3 THEN 'Fraud'
        WHEN 4 THEN 'Ofensive'
        WHEN 5 THEN 'Other'
        WHEN 6 THEN 'Voluntary'
      END


  - dimension: firstname
    type: string
    sql: ${TABLE}.firstname

  - dimension: lastname
    type: string
    sql: ${TABLE}.lastname

  - dimension: usrid
    primary_key: true
    type: number
    sql: ${TABLE}.usrid

  - dimension: emailaddress
    sql: ${TABLE}.emailaddress

  - dimension: responserate
    sql: ${TABLE}.responserate


### MEASURES ###

  - measure: register
    type: count_distinct
    sql: ${TABLE}.usrid

  - measure: acum
    type: running_total
    sql: ${register}
```