**Trabajo final Integrador**

**Base de datos Masivas**

José Emanuel Rodriguez

1. **Resumen:**

En la actualidad la Play Store de Google es una fuente de millones de aplicaciones que los usuarios pueden descargar, tanto gratuitamente como pagando un monto especifico. Quien ingresa a Play Store se encuentra con una ponderación de las aplicaciones más populares de cada categoría. Así mismo, Google las ordena en base a las valoraciones de los usuarios y cantidad de descargas de las mismas.

El presente trabajo propone explorar cuales son los factores que llevaron a las aplicaciones más populares al éxito, y analizar las reviews de los usuarios sobre las mismas, realizando un análisis exploratorio de datos y aplicando técnicas de machine learning.

Este trabajo abarca desde un preprocesamiento de los datos disponibles de las aplicaciones de la Play Store, seguido por una visualización de los datos de exploración sobre las aplicaciones más populares y sobre las reviews realizadas por los usuarios sobre las mismas, y finalizando con la aplicación de técnicas de machine learning para predecir el rating de las aplicaciones (el desempeño del modelo se evalúa utilizando técnicas estándar de evaluación del desempeño), y para analizar las palabras claves de los comentarios positivos de las reviews de las aplicaciones.

Los resultados demuestran que hay factores que influyen en mayor medida sobre la popularidad de las aplicaciones. Por otro lado, el análisis de las palabras claves ha permitido encontrar palabras importantes que tienen relación con los comentarios positivos y negativos de las aplicaciones.

1. **Introducción:**

En la actualidad nos encontramos con una gran cantidad de aplicaciones para dispositivos Android, ya sea de grandes empresas, como de emprendedores que de forma independiente desarrollan su aplicación. Hay una serie de tiendas de aplicaciones para Android, y entre ellas se encuentra la más popular que es Google Play Store, la cual contiene millones de aplicaciones y usuarios activos. Las aplicaciones se encuentran agrupadas según una serie de categorías, como: juegos, educación, familia, entretenimiento, salud y belleza, entre otras. Dentro de ellas se pueden encontrar miles de apps identificadas con su nombre y valoración. Esta última es producto de un promedio de las opiniones de los usuarios que ya han descargado y utilizado la app anteriormente.

Entre estos millones de aplicaciones hay un pequeño porcentaje de las mismas que llega a tener éxito entre los usuarios, que son generalmente las que mejor valoración y mayor cantidad de descargas tienen. Es por esto que, se plantea hacer un análisis que nos permita ver cuáles son las causantes del éxito de estas aplicaciones, que hacen que se destaquen por sobre el resto (y también ver las causantes del fracaso).

Los trabajos sobre el tema generalmente abarcan factores específicos como es el caso de McIlroy, S., Ali, N. & Hassan [[1]](#footnote-1)(2016) que realizaron un análisis sobre la frecuencia de actualización de las aplicaciones de la Google App Store. También se encuentran trabajos como el de Haoyu Wang [[2]](#footnote-2)(2018) que analiza porque las aplicaciones son removidas de la Google play Store.

Sin embargo, es desafiante el hecho de analizar cuáles son los factores que influyen en el éxito de las aplicaciones y analizar los comentarios de los usuarios sobre las mismas para conocer qué es lo que las hace atractivas o desagradables para los usuarios.

Para poder realizar esto se dispone de 2 dataset provistos por el sitio web Kaggle, el primero [[3]](#footnote-3) contiene la información sobre 10.000 aplicaciones de la Play Store de Google, donde se detallan las características principales de las mismas. Este dataset contiene datos como: categoría, rating, tamaño, entre otras.

El segundo[[4]](#footnote-4) dataset contiene más de 64000 comentarios de los usuarios sobre más de 1000 aplicaciones distintas junto con un índice de subjetividad y polaridad para cada comentario.

El siguiente Trabajo práctico se estructura en una primera sección donde se detallan los Objetivos, siguiendo por una descripción de las Metodologías utilizadas y finalizando con los Resultados del análisis y la Conclusión.

1. **Objetivos:**

Para el caso del dataset “googleplaystore.csv” se busca determinar:

1. Cuáles son las aplicaciones con mayor cantidad de descargas.
2. Cuáles son las aplicaciones con más reviews.
3. Precio promedio de las aplicaciones de cada categoría.
4. Rating promedio de las aplicaciones.
5. Cantidad de descargas por categoría.
6. Porcentaje de apps pagas y gratuitas.
7. Cuáles son las apps pagas con más ganancia.
8. Ganancia promedio de apps pagas por cada categoría.
9. Se busca predecir el rating utilizando diferentes técnicas de machine learning, evaluando cual es el modelo que mejor se desempeña usando métricas estándar.

Para el dataset “googleplaystore\_user\_reviews.csv” se busca obtener:

1. Cantidad de reviews por cada tipo.
2. Análisis polaridad y subjetividad de los comentarios en sus diferentes tipos.
3. Cantidad promedio de palabras de los comentarios de cada tipo.
4. Frecuencia del largo de comentarios para cada tipo.
5. Nube de palabras claves para los diferentes tipos de comentarios.
6. Realizar un análisis Ngram sobre los comentarios, para obtener frases más comunes entre los comentarios.
7. Se busca realizar un modelado W2V para obtener las palabras que tienen relación con las palabras claves.
8. **Materiales y métodos:**
9. **Datos:**

En esta sección se brinda una descripción de los datos provistos, comenzando por el dataset “googleplaystore.csv”, este dataset contiene un total de 9660 aplicaciones (filas) y 13 columnas que son los atributos de estas aplicaciones. A continuación, la tabla 1.0 provee información importante de cada variable:

|  |  |  |
| --- | --- | --- |
| Variable | Tipo | Descripción |
| App | String | nombre de la aplicación |
| Category | String | categoría de la aplicación |
| Rating | Decimal | Calificación general del usuario de la aplicación |
| Reviews | Integer | numero de reviews de la app |
| Size | String | tamaño de la aplicación |
| Installs | String | numero de descargas de esa aplicación |
| Type | String | tipo de aplicación= paga o gratis |
| Price | String | precio de la aplicación |
| content rating | String | Grupo de edad al que se dirige la aplicación |
| Genres | String | genero al cual pertenece la aplicación |
| last updated | Date | fecha de última actualización de la app |
| current versión | String | versión actual de la aplicación |
| android versión | String | versión de Android requerida para correr la app |

Tabla 1.0- variables del dataset “googleplaystore.csv”

El dataset “googleplaystore\_user\_reviews.csv” contiene 64.300 registros donde se encuentran las reviews de 1074 apps distintas junto con su tipo(sentiment), subjetividad y polaridad. A continuación, la tabla 2.0 provee información importante de cada variable:

|  |  |  |
| --- | --- | --- |
| Variable | Tipo | Descripcion |
| App | String | Nombre de la aplicacion |
| Translated review | String | Comentario del usuario |
| Sentiment | String | Tipo de comentario, positivo, negativo o neutral |
| Sentiment polarity | String | Puntuación de polaridad |
| Sentiment subjetivity | String | Puntuación de subjetividad |

Tabla 2.0- variables del dataset “googleplaystore\_reviews.csv”

1. **Algoritmos y herramientas de software:**

En el presente trabajo se utilizaron los siguientes algoritmos y herramientas de software:

1. Para el preprocesamiento de los datos se utilizó el lenguaje Python, con la utilización de las librerías: “Pandas”, “re”, “string”, “stopwords”, “wordnetlemmatizer”, “ntlk”.
2. Para el proceso de extracción, transformación y loading (ETL) se utiliza el software PDI de la Siute Pentaho, se toman los datos procesados en R, y se modifican los datos para poder crear tablas en una bd relacional Postgresql. Para esto se hace uso de PgAdmin3 que es una herramienta de código abierto para la administración de bases de datos PostgreSQL.
3. Se utiliza la herramienta Mondrian Schema Workbench para la creación de los cubos OLAP, a partir de la base de datos creada como resultado del proceso etl.
4. Una vez armados los cubos OLAP, estos se usan como entrada para la herramienta Saiku de Pentaho, que nos va a permitir realizar consultas en base a este cubo. Esta herramienta también nos provee las consultas en formato MDX que se usa para crear gráficos utilizando CDE.
5. CDE (Community Dashboard Editor) es la herramienta de la Suite Pentaho para la creación y administración de Dashboards. Esta herramienta se utiliza para poder hacer un análisis grafico de los datos procesados.
6. Para la visualización de los datos exploratorios también se utilizó el lenguaje Python, con las siguientes librerías: “matplotlib”, “seaborn”, “plotly”, “wordcloud”.
7. Para la utilización de técnicas de machine learning también se utilizó el lenguaje Python, con las siguientes librerías: “sklearn”, “matplotlib”, “seaborn”, “keras”, “gensim”.
8. **Metodología:**

En este trabajo los pasos seguidos para realizar el análisis de los datos se muestran en el grafico 1.0:

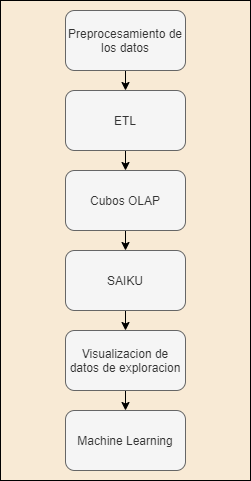


Grafico 1.0 – técnicas utilizadas

**c)1) Preprocesamiento de datos:**

Para una mejor comprensión se mostrará por separado el preprocesamiento de los dos datasets.

Preprocesamiento dataset “googleplaystore”:

Se comenzó por el tratamiento de los datos faltantes, para esto se analizó la cantidad de faltantes del dataset,

donde se encontraron 1474 datos faltantes en el campo “rating”, 8 en el campo “current\_ver”, 3 en el campo “Android\_ver”, y 1 en los campos “type” y “content rating”.

Se observa que el campo rating tiene una gran cantidad de faltantes, es por esto que se van a remplazar estos valores faltantes por la media de los demás valores de ese campo (imputación Hot Deck). El resto de los datos faltantes de los demás campos, al ser una cantidad mínima, fueron borrados.

También se hallaron registros duplicados, por lo que se eliminaron los mismos. Luego, se realizó una limpieza del campo “installs” donde se eliminaron los caracteres de tipo string para poder convertir ese campo a entero.

El campo “installs” y el campo “reviews” fueron convertidos a enteros. En el campo “float” se eliminaron los caracteres de tipo string y luego fue convertido a float.

El campo “size” tenía una gran cantidad de valores iguales a “varies with device”, estos datos fueron reemplazados por nan. El resto de los valores se estandarizan (debido a que había valores en kb y valores en mb) y se pasa el campo a tipo float.

Preprocesamiento dataset “googleplaytore\_reviews”:

En este dataset se comenzó con el tratamiento de datos faltantes, en este caso había 26868 registros faltantes que debieron ser eliminados ya que no se disponía de la opinión del usuario, que es el campo más importante del dataset para el análisis que se busca realizar.

Para poder trabajar sobre el campo “translated\_review” que contiene los comentarios de los usuarios sobre la aplicación se debió remover los caracteres de URLs presentes, los tags HTML, los emojis, las puntuaciones (‘n) que provienen del lenguaje inglés, y se pasaron todos los comentarios a minúsculas.

**c)2) ETL:**

Para la creación de la base de datos relacional se utilizará el programa “spoon”, se realiza el siguiente Job en Spoon (grafico 2.0):

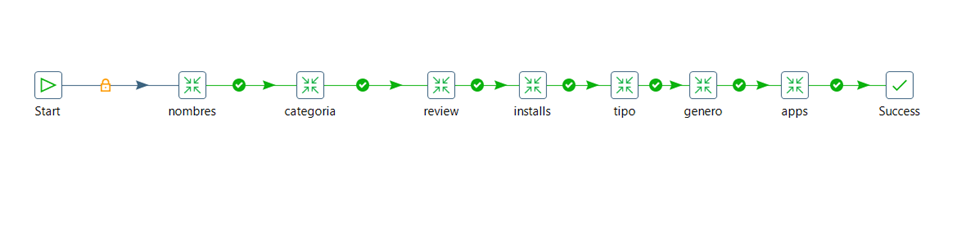


Grafico 2.0 -job transformation

Este job consta de una serie de transformaciones para crear las tablas: “nombre”, “categoría”, “review”, “installs”, “tipo” y “genero” que se realizan de la siguiente forma (grafico 3.0):

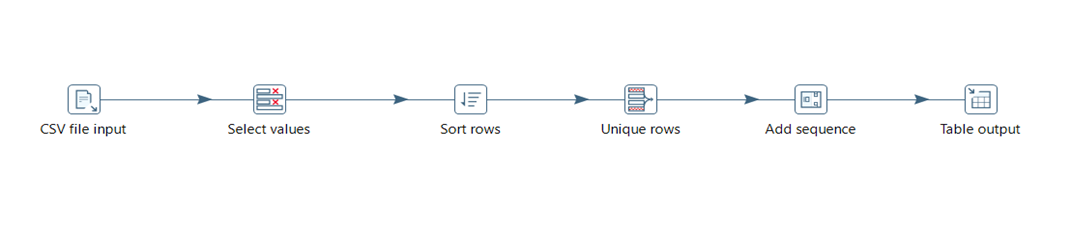


Grafico 3.0 - creacion de tablas

* Primero se carga el archivo csv generado luego de procesar los datos en R.
* Luego se selecciona la columna de la tabla que queremos armar, “nombre”, “categoría”, “review”, “installs”, “tipo” o “genero”.

Se ordena el campo seleccionado

* Nos quedamos con los valores únicos para esa variable
* Se agrega el número de secuencia(id) para cada valor de esa variable.
* Se genera la tabla en la bd Postgres.

La tabla apps se forma a partir de las demás tablas, para esto se realiza el siguiente proceso (grafico 4.0):

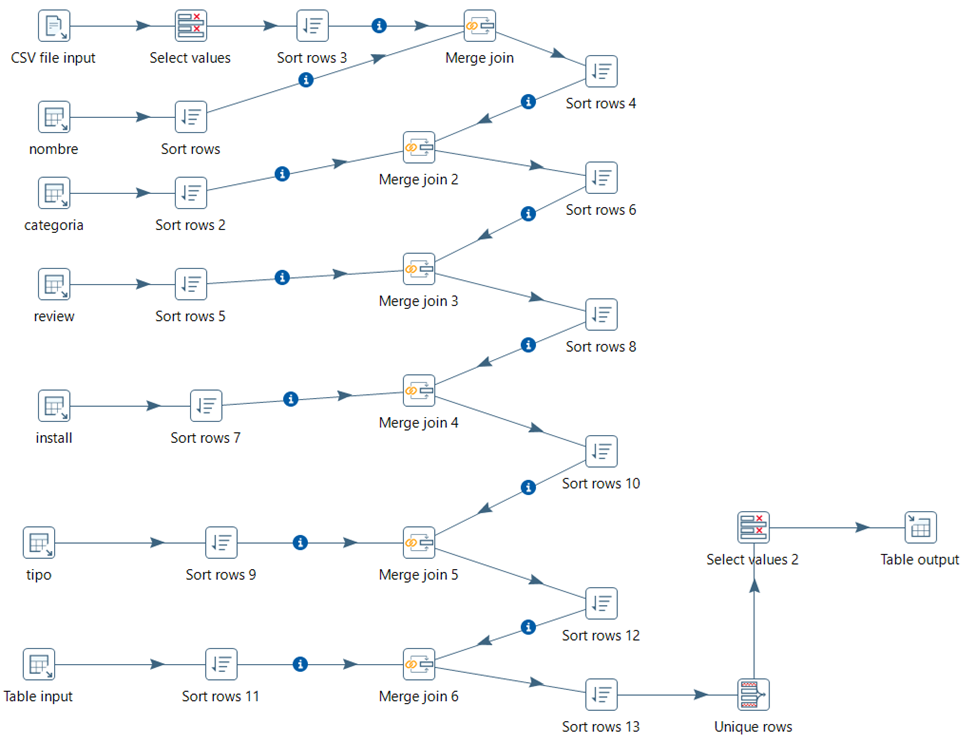


Grafico 4.0 - creacion de la tabla app

* Se carga el archivo csv generado en R, y se hace un select de todos los campos de cada una de las tablas creadas anteriormente.
* Para hacer los joins es necesario ordenar los datos según el campo que se va a realizar el join.
* Una vez ordenados los campos se lleva a cabo un merge join entre estos.
* Una vez hecho todos los joins se procede a seleccionar los campos “id\_nombre”, “id\_categoría”, “id\_review”, “id\_installs”, “id\_tipo” y “id\_genero” y con estos campos se crea la tabla app en la base de datos.

**c)3) Armado de cubos:**

Una vez creada la base de datos se procede con la generación de los cubos que se desarrollan con la herramienta Mondrian de la Suite Pentaho. Se utiliza como tabla de hechos a la tabla Apps, y se define una serie de medidas para poder hacer las consultas necesarias en Saiku.

Una vez realizado el cubo, se publica el esquema en el servidor de Pentaho (bi-server), luego, con los cubos ya publicados se pueden realizar las consultas con Saiku Analytics.

**c)4) Saiku:**

Una vez creado el cubo se realizan las consultas en los datos a través de Saiku utilizando las medidas definidas en el cubo y las dimensiones del mismo. Se busca generar las consultas para cumplir con los objetivos definidos anteriormente.

Primero se debe crear una conexión con la base de datos a partir de la cual se creó el cubo, una vez hecho esto se procede a realizar las consultas.

Se realizan consultas correspondientes en Saiku, para lograr generar las consultas en formato MDX, y estas se utilizan para crear los gráficos que nos permiten hacer un análisis exploratorio de los datos.

**c)5) Visualización de los resultados de exploración:**

Se crean graficas para mostrar los resultados de las consultas sobre los cubos de una forma más simple e intuitiva. Se parte de las consultas MDX generadas con Saiku para poder crear gráficos de barra y gráficos de torta, donde se analizan las relaciones entre las distintas variables del dataset. Mediante este análisis visual se busca lograr una mejor comprensión del problema que se está tratando y se busca explicar los comportamientos de las variables del dataset.

Además, se generaron gráficos utilizando el programa Python mediante las librerías: “matplot”, “seaborn”, “plotly” y “pylab”. Gracias a estas librerías se pudieron generar gráficos que aportan al análisis exploratorio de los datos.

Los gráficos generados a través de esta metodología se pueden ver en la sección de resultados con sus respectivos análisis de los mismos.

**c)6) Machine Learning:**

Para el dataset “googleplaytore” se aplicaron diferentes modelos de regresión para intentar predecir el rating de una aplicación, para esto se aplicaron 3 métodos que son: regresión líneal, SVM, y random forest.

La performance de cada uno de los diferentes modelos será evaluada usando diferentes metricas de evaluación, para cada uno de los métodos de regresión aplicados se aplicarán las métricas: MSE, RMSE, Y MAE.

Para visualizar cada modelo de regresión se genera un gráfico “regplot” para comparar los resultados obtenidos por cada modelo, estos gráficos sirven para visualizar una relación lineal determinada mediante regresión.

Por otro lado, para el dataset “googleplaytore\_user\_review” se aplicó el método W2V para que, una vez obtenidas las palabras claves de las reviews de los usuarios, poder hallar que palabras se relacionan con esas palabras claves, word2vec aprende embeddings de las palabras clave intentando predecir un contexto dada esas palabras clave. Word2Vec fue el método elegido ya que brinda sinónimos con alta precisión, diferenciando incluso entre representaciones de la misma palabra, como por ejemplo “Good” o “gud”, que es de mucha ayuda para el caso de los comentarios de los usuarios.

**5) Resultados:**

**a) Datos faltantes:**

El dataset “googleplaystore” presentaba 1474 datos faltantes en el campo “Rating”, que corresponde a que un 13% de las filas totales que posee el dataset no tienen valores para este campo. Dada la gran cantidad de datos faltantes se optó por utilizar una imputación “Hot Deck”, reemplazando estos valores faltantes por la media de los demás valores de dicho campo.

Ademas había 8 datos faltantes en el campo “current\_ver”, 3 en el campo “Android\_ver”, y 1 en los campos “type” y “content rating”. Para este caso, al ser una cantidad mínima de datos se opto por eliminar los registros con faltantes.

Para el caso del dataset “googleplaystore\_user\_review” el dataset presentaba 26868 datos faltantes, un 41% de los registros del dataset, pero en este caso eran registros que solo tenían el nombre de la aplicación, y no tenían ningún dato sobre las reviews, por lo que se opto por eliminar estos registros ya que para este trabajo los datos que nos son útiles son justamente las reviews.

**b) ETL:**

El proceso de integración de datos incorpora los resultados de los pasos previos de preprocesamiento en un esquema de tipo estrella. Luego de haber generado un archivo .csv con los datos procesados en R, estos datos son extraídos, transformados y utilizados para crear una base de datos necesaria para poder realizar los pasos posteriores. Para hacer esto se hacen los procesos explicados en la sección de metodologías en el programa Spoon.

La base de datos resultante tiene la siguiente estructura (tabla 3.0):

|  |  |  |
| --- | --- | --- |
| Tabla | Campos | Tipo |
| Apps | id\_nombre | Integer |
|  | id\_categoria | Integer |
|  | id\_review | Integer |
|  | id\_installs | Integer |
|  | id\_tipo | Integer |
|  | id\_genero | Integer |
| Nombre | id\_nombre | Integer |
|  | Nombre | Varchar |
| Categoría | id\_categoria | Integer |
|  | Categoría | Varchar |
| Review | id\_review | Integer |
|  | Review | Integer |
| Installs | id\_installs | Integer |
|  | Installs | Varchar |
| Tipo | id\_tipo | Integer |
|  | Tipo | Varchar |
| Genero | id\_genero | Integer |
|  | Genero | Varchar |

Tabla 3.0- bd resultante

**c)Cubos OLAP:**

A continuación, se muestra el cubo definido con sus dimensiones y sus métricas, en este caso, la tabla apps será la tabla de hechos (grafico 5.0):

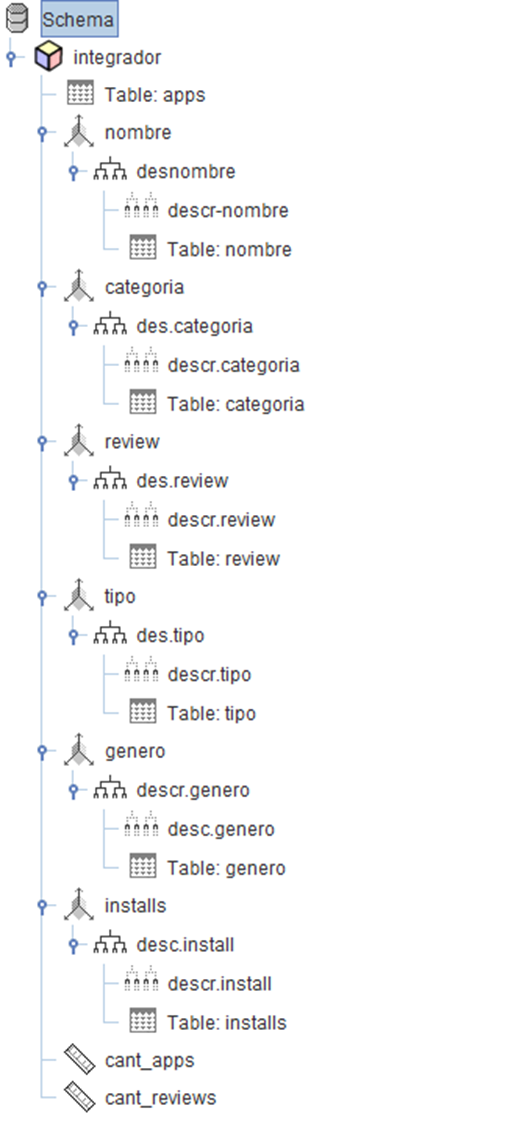


Grafico 5.0 – culo olap

Una vez finalizado este cubo, se debe publicar el esquema en el servidor de Pentaho, con esto realizado ya se puede pasar a hacer las consultas necesarias en Saiku, que es el paso siguiente.

1. **Saiku:**

En Saiku se proceden a hacer las consultas necesarias para cada uno de los objetivos descriptos al principio del trabajo, a partir de estas consultas se obtiene la consulta MDX, que se utiliza para crear gráficos que nos permitan observar cada consulta de una forma gráfica e intuitiva.

1. **Visualización de los datos de exploración:**

Se comienza con la visualización de datos de exploración sobre el dataset “googleplaystore”.

1. Apps con más cantidad de descargas:

|  |  |  |
| --- | --- | --- |
| App | Category | Installs |
| Google Play Books | books and reference | 1 billon |
| Google News | news and magazines | 1 billon |
| Google | tools | 1 billon |
| Subway Surfers | game | 1 billon |
| Hangouts | communication | 1 billon |
| Google Drive | productivity | 1 billon |
| Google Chrome | communication | 1 billon |
| Google Maps | travel and local | 1 billon |
| Whatsapp Messanger | communication | 1 billon |
| Google Play Games | family | 1 billon |
| Gmail | communication | 1 billon |
| Facebook | social | 1 billon |
| Google play Movies & TV | video players | 1 billon |
| Google Photos | photography | 1 billon |
| Youtube | video players | 1 billon |
| Messanger | communication | 1 billon |
| Google+ | social | 1 billon |
| Instagram | social | 1 billon |
| Skype | communication | 1 billon |
| Google Street View | travel and local | 1 billon |

Tabla 4.0

Como se puede observar en la tabla 4.0, la lista de apps que tienen más de 1 billón de descargas muestra a las siguientes aplicaciones:

* Gmail, Google Street View, Google Play Movies & TV, Google maps, YouTube (Apps pertenecientes a Google)
* Facebook, WhatsApp Messenger, Instagram, Messenger – Text and Video Chat for Free (Apps pertenecientes a Facebook)
* Skype (pertenece a Microsoft)
* Subway Surfers (pertenece a SYBO GAMES)

La razón por la que las aplicaciones de Google tienen más descargas que el resto es porque vienen preinstaladas en todos los sistemas Android. Las aplicaciones pertenecientes a Facebook son las redes sociales y la aplicación de comunicación mas utilizada por los usuarios. Skype es la aplicación de videollamada mas utilizada por los usuarios y Subway Surfers es el juego mas descargado de la Google play Store.

1. Apps con mayor cantidad de reviews:

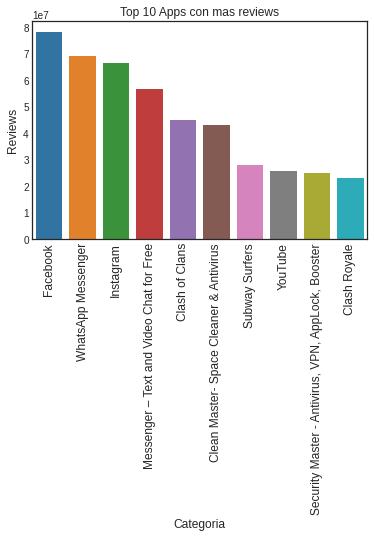


Grafico 6.0 - apps con mas reviews

En el grafico 6.0 se observan las 10 aplicación con mayor cantidad de reviews, entre estas se encuentran gran cantidad de las aplicaciones con mas descargas mencionadas anteriormente, algo que tiene sentido si se piensa en que son las que mayor cantidad de usuarios tienen. También se destaca que 3 de las 10 aplicaciones son juegos (crash royale, Clash of clans, subway surfers).

1. Precio promedio de las aplicaciones por categoría:

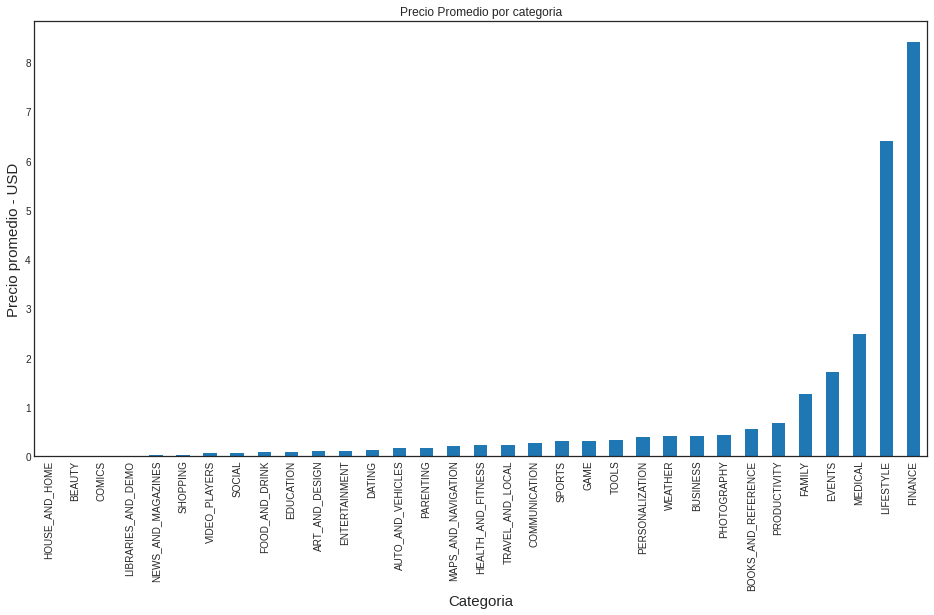


Grafico 7.0- precio por categoría

En el grafico 7.0 se observan las categorías de aplicaciones pagas que tienen(en promedio) mayor valor en dólares, se observa que la categoría finanzas es la que posee aplicaciones mas caras, seguidas de “lifestyle” y las aplicaciones médicas.

1. Rating promedio de las aplicaciones:

Vamos a observar la distribución de la variable rating a través de un histograma:

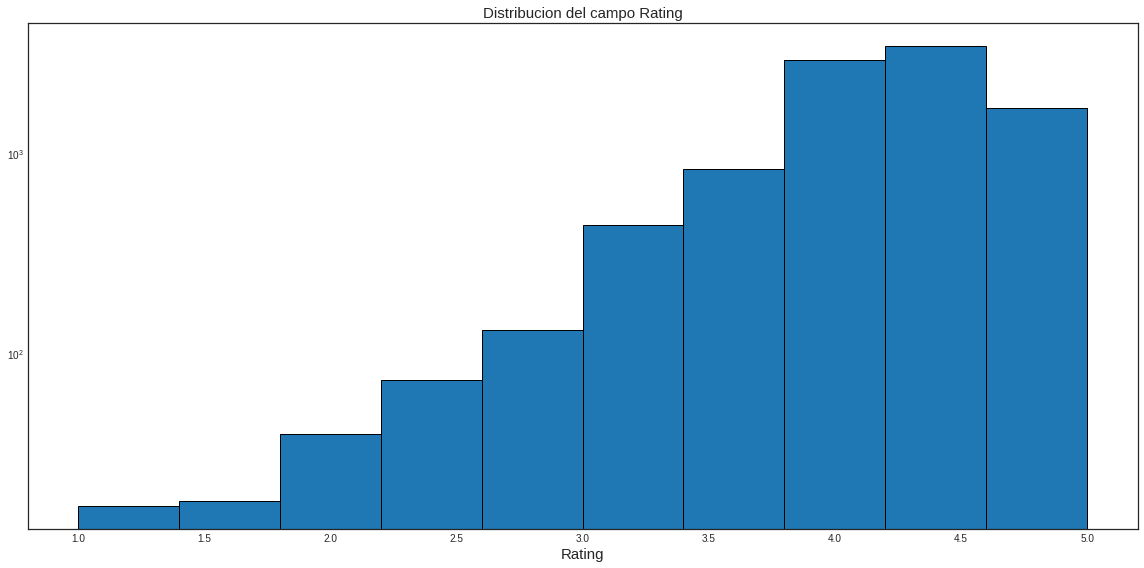


Grafico 8.0 - distribucion variable rating

El campo rating tiene una media de 4,19 puntos de rating por aplicación, una media bastante elevada si consideramos que los puntos de rating van de 0-5. En el grafico 8.0 podemos observar que hay una distribución sesgada a la derecha, la distribución tiene una curtosis alta, por lo que muchas de las aplicaciones tienen una calificación de entre 4,1 y 4,5.

1. Cantidad de descargas de aplicaciones por categoría:

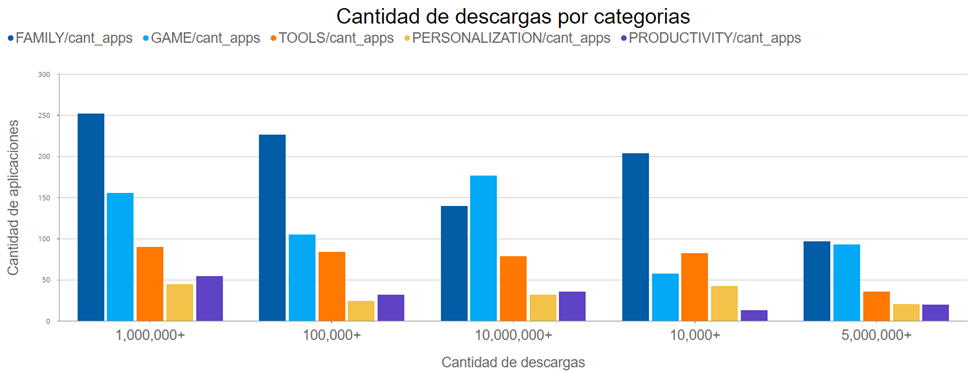


Grafico 9.0

En el gráfico 9.0 se muestran las principales categorías, junto con las descargas de cada una. Podemos ver que la categoría familia es la mas descargada, seguida por la categoría juegos, y por detrás se encuentran las categorías tolos, personalización y productividad.

1. Aplicaciones pagas y gratuitas:

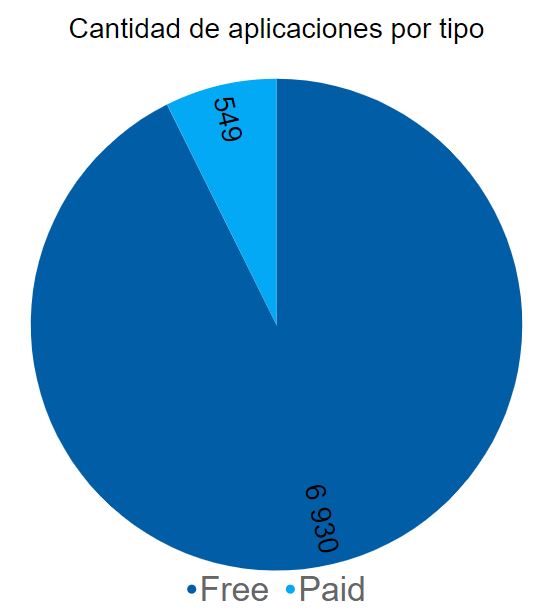


Grafico 10.0 - cantidad de aplicaciones pagas y gratuitas

Para el análisis de la cantidad de aplicaciones clasificadas por tipo se utiliza el grafico de torta que se muestra en el grafico 10.0, en este caso hay 2 tipos de aplicaciones, las pagas y las gratis. Se observa que el mercado está dominado por aplicaciones gratis con un 93% del total, por lo que se puede deducir que estas son las más buscadas por los usuarios.

1. Apps pagas con mayor ganancia:

Para poder conocer cuales son las apps pagas con mayor ganancia se creó una columna “ganancia” que proviene del resultado de multiplicar el precio de la aplicación por la cantidad de descargas de la misma.

De esto se obtuvo que la la aplicación paga con mayor ganancia es Minecraft, el videojuego de la empresa “Mojang” tiene mas de 10 millones de descargas y su precio es de 7 dólares, lo que da un total de 69900000.0 usd.

1. Categorías con más ganacia:

Utilizando la columna ganancia creada para el punto anterior se realiza un gráfico (11.0) que puede verse a continuación:

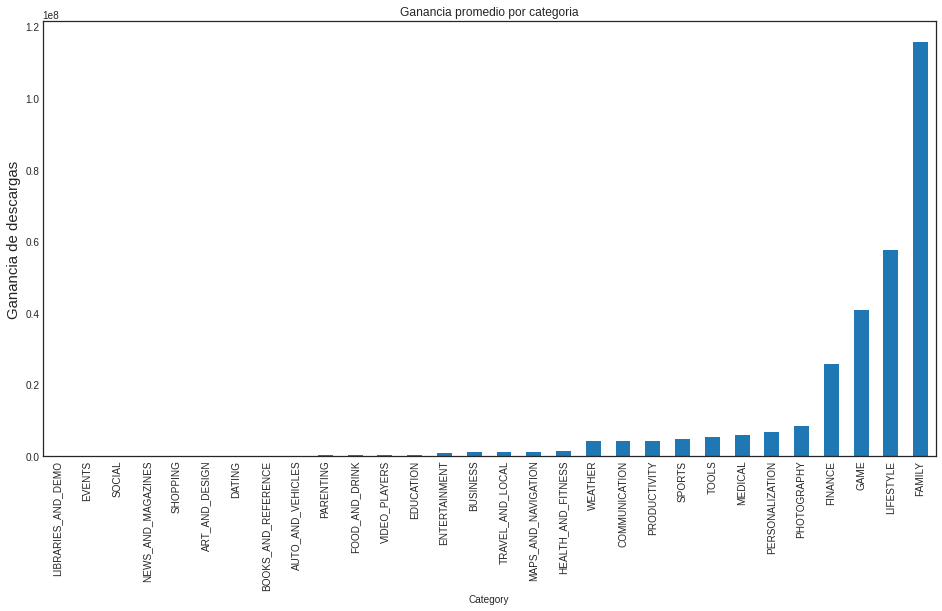


Grafico 11.0 - ganancia promedio por categoria.

Como se observa en el gráfico, las categorías family, lifestyle, game y finance son las que mayor ganancia tienen, vale recordar que las categorías finance y lifestyle tenían aplicaciones cuyo valor en dólares era elevado como se vio anteriormente en el grafico 7.0.

Hasta aquí hemos visto los resultados de la exploración de datos para el dataset “googleplaystore”, a continuación, se ven los resultados para “googleplaystore\_user\_review”:

1. Cantidad de reviews por tipo:

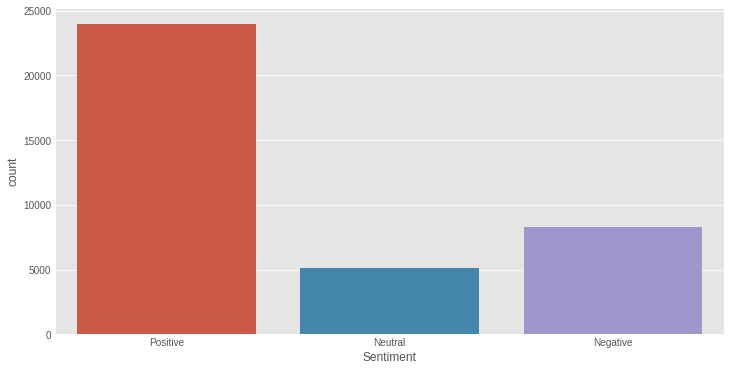


Grafico 12.0

Se puede observar que la mayor cantidad de comentarios de los usuarios sobre las aplicaciones son comentarios positivos, con mas de 24000, seguido por los negativos con mas de 8000 y por último están los comentarios neutrales con poco más de 5000.

1. Análisis de la subjetividad y polaridad de los comentarios en cada tipo:

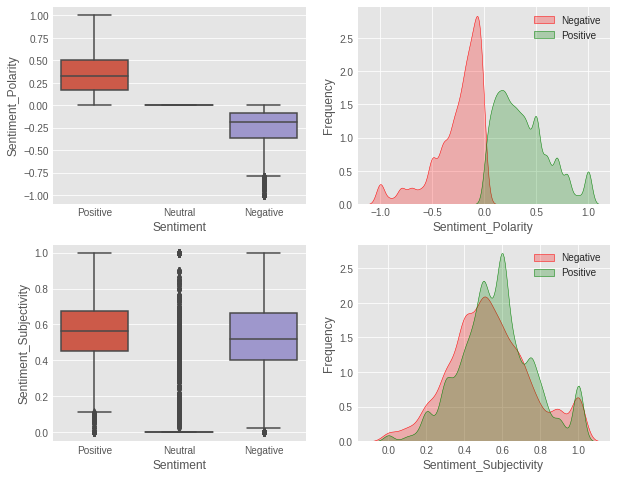


Grafico 13.0

Gracias al grafico 14.0 podemos hacer el siguiente análisis de cada variable:

Polaridad:

* Positivos = la polaridad de las reviews es superior, la mediana es 0,37
* Negativos = la polaridad de las reviews es inferior, la mediana es -0.12
* Neutros = la polaridad de las reviews es 0.

Subjetividad:

* Positivos = la Subjetividad de las reviews es superior, la mediana es 0,58
* Negativos = la Subjetividad de las reviews es inferior a los positivos, la mediana es 0.55
* Neutros = se observan gran cantidad de outliers.

1. Cantidad de palabras promedio en los comentarios de cada tipo:

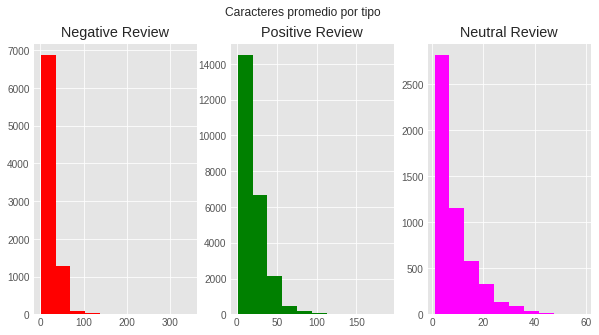


Grafico 14.0

Podemos observar que los comentarios negativos tienen entre 0-30 palabras promedio en mas de 6900 reviews, los comentarios positivos tienen en promedio de 0-20 palabras por review en mas de 14200 reviews y los comentarios neutrales tienen entre 0-6 palabras promedio en más de 2700 reviews.

1. Frecuencia de largo de comentarios

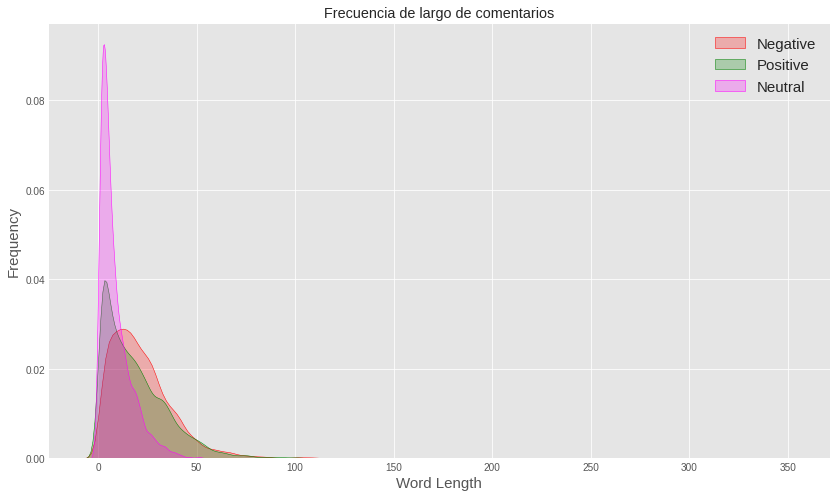


Grafico 15.0

En el grafico 15.0 podemos observar que la densidad es mucho mayor en los comentarios neutrales que en los otros dos tipos, estos tienden a ser mas cortos en la mayoría de los casos, a diferencia de los otros dos tipos donde tienden a ser un poco más largos.

1. Nubes de palabras:

Comentarios negativos:



Grafico 16.0 - wordcloud de comentarios negativos

Se puede obsevar que entre los comentarios negativos se repiten mucho palabras como waste time, need (to improve something), update (error en algun update), frustrate, cant, bad, please fix, hate, ads(muy invasivas), terrible, problem.

Comentarios positivos:



Grafico 17.0 - wordcloud de comentarios negativos

Se puede observar que entre los comentarios positivos se repiten con frecuencia palabras como love, good, great, better, use, work, thank, best.

Comentarios neutrales:

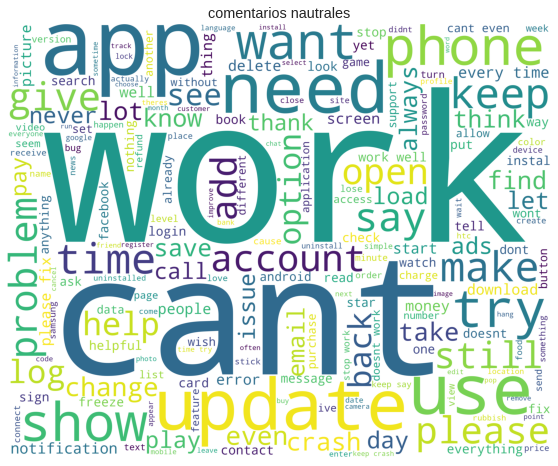


Grafico 18.0 - wordcloud de comentarios neutrales

Entre los comentarios neutrales observamos que se repiten palabras como cant, work, update, make, keep, please, problem, waste time, want, account, crash.

1. Frases mas populares en los comentarios:

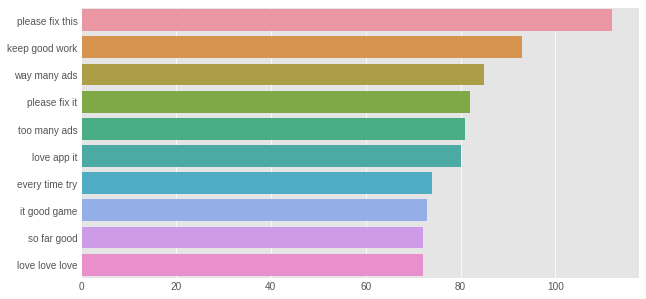
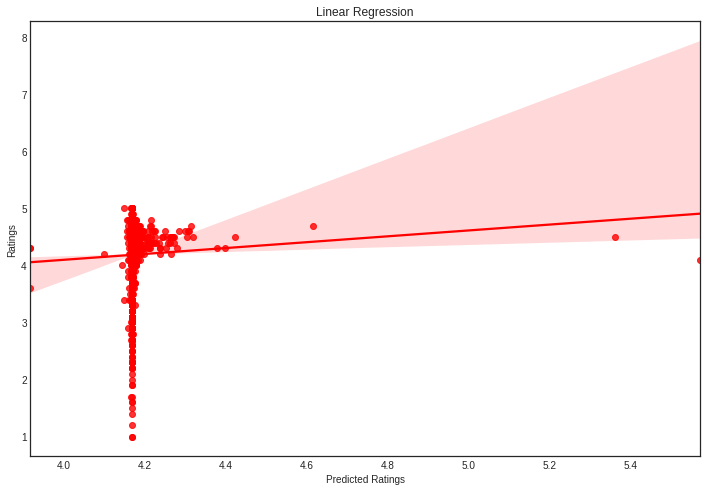


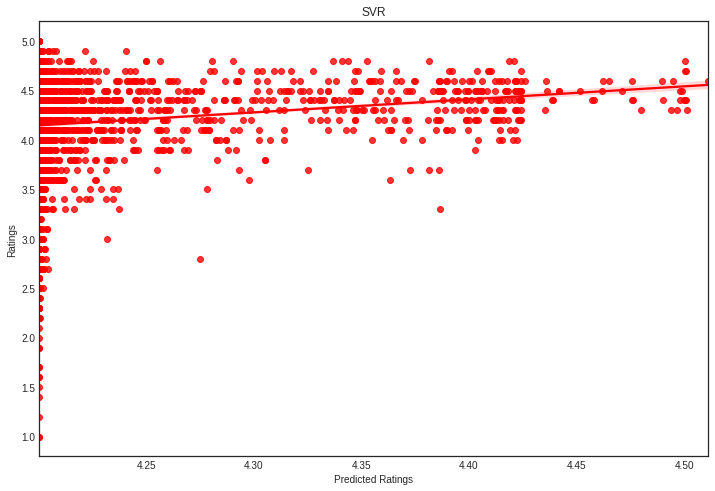
Grafico 19 - frases mas populares

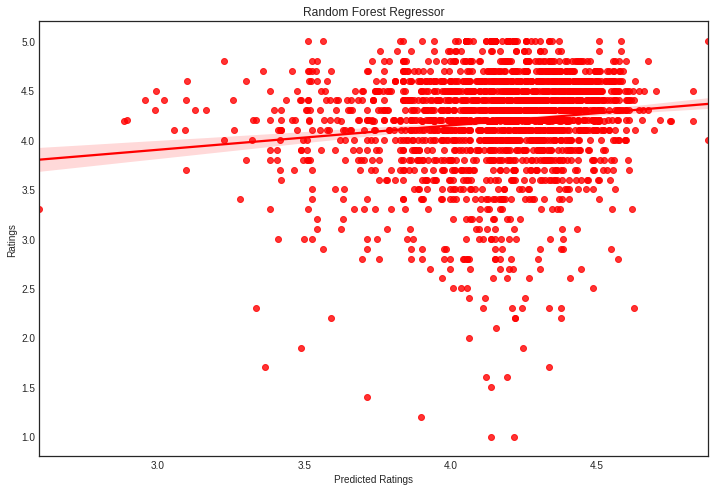
Entre las frases podemos destacar frases positivas como: “keep Good work”, “love app it”, “it Good game”, “so far good”, “love love”. Entre las frases negativas encontramos: “please fix this/it” (haciendo referencia a errores que afectan a la usabilidad por parte del usuario), “way many ads” (normal entre las aplicaciones gratuitas que utilizan promociones para obtener ganancias).

1. **Machine learning:**

Sobre el dataset “googleplaystore” se ejecutaron los algoritmos de regresión lineal, random forest y SVM para poder predecir el rating de las aplicaciones, y se realizó una comparación entre estos, los resultados fueron los siguientes:







Sobre los diferentes métodos se aplicaron métricas estándar de evaluación de performance, los resultados fueron los siguientes (tabla 5.0):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metrica | Regresion Linear | SVM | Random Forest | Voting regressor |
| MSE | 0.2378 | 0.2329 | 0.2738 | 0.2547 |
| MAE | 0.3293 | 0.3117 | 0.3609 | 0.3457 |
| MSLE | 0.0113 | 0.0112 | 0.0127 | 0.0131 |

Tabla 5.0

Por último se utilizó la librería “sklearn.ensamble.VotingRegressor” que permite realizar una predicción en base al promedio de todos los modelos contribuyentes. Los resultados fueron los siguientes (Grafico 20.0):

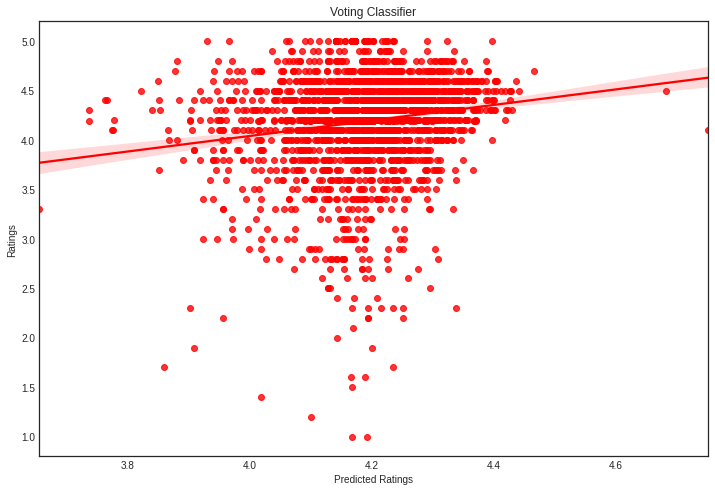


Grafico 20.0

De todo lo visto en esta sección se puede concluir en que, entre los 3 modelos propuestos, el Random Forest fue el que mejor performance tuvo, por encima de la regresión linear y el SVM, mientras que el voting Regressor tuvo una buena performance si bien la diferencia con respecto al resto no es demasiada. Se pudo predecir un rating de 4.2 puntos sobre 5.

Para el dataset “googleplaystore\_user\_reviews” se aplico el modelo Word2vec, al cual se le cargo un conjunto de frases (llamado corpus) para que el modelo puede analizarlo y predecir palabras vecinas que estén relacionadas. Para mostrar los resultados del mismo se opto por utilizar palabras claves de comentarios positivos que se pueden observar en los gráficos 16.0 y 17.0, los resultados fueron los siguientes:

Reviews positivos:

* Palabra “great”: amanzing, perfect, Good, wonderful, awesome, cool, nice, well, awesome.
* Palabra “love”: enjoy, awesome, excite, absolutely, appreciate, liked, favorite, cool, amazing.

Reviews negativos:

* Palabra “cant”: wont, couldnt, trough, unable, able, none, anymore, everytime.
* Palabra “update”: mess, reléase, latest, longer, unusable, july, june, marshmallow, break.

**6) Conclusión:**

En este trabajo se aplicaron técnicas de preprocesamiento, análisis y visualización de los datos provenientes de las aplicaciones de Google Play Store y de las reviews de los usuarios de dichas aplicaciones. Fueron analizadas diferentes variables que caracterizan a las aplicaciones y a los comentarios que realizaron los usuarios de los mismos, y fueron analizados los resultados obtenidos.

Los resultados mostraron que, si bien las aplicaciones gratis son las que mayor cantidad de descargas logran, sobre todo las redes sociales y las aplicaciones de comunicación, las aplicaciones pagas logran grandes ganancias incluso con menor cantidad de descargas, como es el caso del juego “Minecraft” que logro recaudar mas de 70 millones de dólares. Por el lado de las aplicaciones gratis las ganancias se basan en la aparición de publicidad, algo que es bastante molesto para los usuarios, como se pudo observar en el análisis de los comentarios negativos que los mismos realizan, donde se vieron que frases como “too many ads” se repetían con frecuencia. Vale aclarar que, dentro de los datos analizados, solo un 7% del total de las aplicaciones son aplicaciones pagas y el 93% restante son aplicaciones gratuitas.

También se realizó un análisis de las palabras claves presentes en los comentarios de los usuarios sobre las aplicaciones, donde se pudo observar que hay palabras que guardan fuerte relación con los comentarios positivos, como “love this app”, “good work”, y ciertas palabras que guardan relación con los comentarios negativos, como “fix this”, “many ads”. El análisis aplicado sobre los comentarios puede ser aplicado de forma estándar a otros datasets que contengan comentarios u opiniones sobre un producto o servicio.

También se pudo observar que hay una mayor cantidad de comentarios positivos que negativos, un 64% del total de los comentarios son positivos y un 21% son comentarios negativos, el resto son comentarios neutrales. Se pudo observar que los comentarios negativos tienen, en promedio, una mayor longitud que los positivos, teniendo en promedio de 0-30 palabras, mientras que los positivos de 0-20.

Por último, se aplicaron diferentes técnicas para predecir el rating de una aplicación, donde se encontró que el rating seria de 4,2 puntos sobre 5. Con respecto a las metodologías usadas, si bien la regresión linear es la metodología mas utilizada en estos casos, se aplicaron los algoritmos de SVM y de Random forest para poder comparar los resultados obtenidos, y sorprendentemente el Random forest fue el modelo que mejores resultados brindo, si bien la diferencia con respecto al resto no fue amplia.

**7) Referencias:**

References

Mahmood, A. (2020). Identifying the influence of various factor of apps on google play apps ratings. *Journal of Data, Information and Management*, *2*(1), 15-23.

Malavolta, I., Ruberto, S., Soru, T., & Terragni, V. (2015, May). Hybrid mobile apps in the google play store: An exploratory investigation. In *2015 2nd ACM international conference on mobile software engineering and systems* (pp. 56-59). IEEE.

<https://play.google.com/store/apps/>

Islam, M. R. (2014, April). Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews. In *2014 International Conference on Electrical Engineering and Information & Communication Technology* (pp. 1-4). IEEE.

Malavolta, I., Ruberto, S., Soru, T., & Terragni, V. (2015, May). Hybrid mobile apps in the google play store: An exploratory investigation. In *2015 2nd ACM international conference on mobile software engineering and systems* (pp. 56-59). IEEE.

Martin, W., Sarro, F., Jia, Y., Zhang, Y., & Harman, M. (2016). A survey of app store analysis for software engineering. *IEEE transactions on software engineering*, *43*(9), 817-847.

McIlroy, S., Ali, N., & Hassan, A. E. (2016). Fresh apps: an empirical study of frequently-updated mobile apps in the Google play store. *Empirical Software Engineering*, *21*(3), 1346-1370.

H. Wang, H. Li, L. Li, Y. Guo and G. Xu, "Why are Android Apps Removed From Google Play? A Large-Scale Empirical Study," 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR), Gothenburg, Sweden, 2018, pp. 231-242.

McIlroy, S., Ali, N. & Hassan, A.E. Fresh apps: an empirical study of frequently-updated mobile apps in the Google play store. *Empir Software Eng* **21,**1346–1370 (2016).

<https://gruizdevilla.medium.com/introducci%C3%B3n-a-word2vec-skip-gram-model-4800f72c871f>

<https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines>

<https://programminghistorian.org/en/lessons/keywords-in-context-using-n-grams>

1. McIlroy, S., Ali, N. & Hassan, A.E. Fresh apps: an empirical study of frequently-updated mobile apps in the Google play store. Empir Software Eng 21, 1346–1370 (2016). https://doi.org/10.1007/s10664-015-9388-2 [↑](#footnote-ref-1)
2. H. Wang, H. Li, L. Li, Y. Guo and G. Xu, "Why are Android Apps Removed From Google Play? A Large-Scale Empirical Study," 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR), Gothenburg, Sweden, 2018, pp. 231-242. [↑](#footnote-ref-2)
3. El dataset lo Podemos encontrar en: https://www.kaggle.com/lava18/google-play-store-apps?select=googleplaystore.csv [↑](#footnote-ref-3)
4. El dataset puede encontrarse en https://www.kaggle.com/lava18/google-play-store-apps?select=googleplaystore\_user\_reviews.csv [↑](#footnote-ref-4)