Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

Memoria Final

Evaluación de Aplicaciones de Estadística de Android del PlayStore de Google

Autor: Miguel Moreno Mardones

Tutor(a): Juan Antonio Fernández del Pozo de Salamanca

Madrid, enero 2021

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática
Título: Evaluación de Aplicaciones de Estadística de Android del PlayStore de Google
Enero 2021

Autor: Miguel Moreno Mardones

Tutor: Juan Antonio Fernández del Pozo de Salamanca
Departamento Inteligencia Artificial (DIA)
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

El presente documento trata de establecer un estudio y evaluación de las aplicaciones que ofrece la plataforma de Google PlayStore relacionadas con el ámbito de la probabilidad y estadística.

Dicho estudio se centrará en conocer los aspectos principales del entorno en base a su contenido, funcionamiento o evaluación por parte de los usuarios, todo ello sobre un conjunto total de 250 aplicaciones. Sus principales características serán extraídas, y se procederá a la formación de un conjunto de datos que permita obtener una visión general tabulada de dichas aplicaciones.

La metodología, conceptos o teoría subyacente relacionada con la probabilidad y estadística corre a cargo del alumno, siendo este el encargado de proporcionar los métodos necesarios para lograr dicho análisis.

Por lo tanto, de manera autónoma y con la ayuda del entorno RStudio, se desarrollarán las medidas oportunas estadísticas, así como una implementación de algoritmos e interpretación de resultados. Todo ello con la previa adquisición de los conocimientos necesarios que permitan realizar dicho cometido.

Así mismo, se utilizará en menor medida la herramienta Open Refine, la cual es bien conocida por el alumno. Su objetivo no será otro que facilitar la manipulación, categorización, o establecimiento de las condiciones lógicas necesarias a la hora de extraer las medidas de las variables del conjunto de datos.

Abstract

This document aims to establish a study and evaluation related to probability and statistics applications offered by the Google PlayStore platform.

The study will focus on the main aspects of the environment based on its content, operation, or evaluation by users, all of this on a total set of 250 applications. Their main characteristics will be extracted, and a dataset will be created, providing a tabulated overview of these applications.

Methodology, concepts, or underlying theory related to probability and statistics will be administered by the student, being himself responsible for providing the necessary methods to achieve such analysis.

Therefore, in an autonomous manner and with the help of the RStudio environment, the appropriate statistical measures will be developed, as well as an implementation of algorithms and interpretation of results. All of this with the prior acquisition of the necessary knowledge to develop this task.

Furthermore, the Open Refine tool, which is well known to the student, will be used to a lesser extent. Its objective will be none other than to facilitate the manipulation, categorization, or establishment of the logical conditions necessary when extracting the measurements of the variables of the data set.

Tabla de contenidos

Lista de Figuras

Lista de Tablas

1 Introducción y objetivos

1.1 Introducción

El área de la probabilidad y la estadística es inmensamente amplia, utilizada en cualquier campo laboral. Se trata de un concepto indispensable hoy en día, tanto que, sin esta disciplina, el entendimiento, recolección y manejo de datos en aspectos como la economía o la salud no serían posibles. Aspectos del Big Data como la Minería de Datos o la creación de algoritmos de agrupación beben también de esta área, ya que requieren de la utilización de modelos de datos probabilísticos que definan las relaciones entre variables.

Es por eso por lo que puede llegar a convertirse en un verdadero quebradero de cabeza el hecho de querer realizar un estudio que abarque una gran cantidad de datos sin antes definir unos fundamentos previos que permitan seleccionar con más detalle cuales son necesarios. Definir el problema, establecer los objetivos, seleccionar las herramientas de análisis, concretar los resultados y elaborar la interpretación y las conclusiones son tareas que permiten trabajar con una metodología eficaz a la hora de realizar un estudio.

La intención del presente estudio pasa por recopilar información de aplicaciones Android relacionadas con la estadística y la probabilidad albergadas en la plataforma de Google Play Store. Posteriormente, con esta información realizaremos un análisis de tipo Clúster y de predicción con el objetivo de entender el entorno de estas aplicaciones en la plataforma de Google en base a aspectos como: el perfil de los usuarios de uso, el mercado de pago, las descargas, la relevancia que poseen más allá del ranking de puntuación....

Pero ¿que se pretende realizar mediante estos dos tipos de análisis? A través del algoritmo de Clúster podemos establecer la relación entre los datos recogidos de las aplicaciones, con el objetivo de poder crear conjuntos de grupos de datos que sean similares entre sí y que aparezcan en dichas aplicaciones. Por otra parte, el modelo predictivo puede indicarnos el comportamiento de un conjunto de datos como por ejemplo el número de descargas de una aplicación para así adelantarse en la toma de decisiones y poder medir su impacto en la plataforma.

1.2 Objetivos

La idea principal de este trabajo es desarrollar un estudio de las aplicaciones móviles relacionadas con la probabilidad y estadística albergadas en la plataforma de Google Play.Para llevar a cabo este objetivo, se pretende dividir el proceso en varios pasos los cuales ayudarán a conformar un análisis lo suficientemente válido para determinar el estado del ecosistema de este tipo de aplicaciones.

Por ello, se debe tomar en consideración aspectos como el contenido de las aplicaciones, su funcionamiento o evaluación, ya que conocer este conjunto de datos hará más sencillo el entendimiento actual del entorno.

De igual manera y a lo largo del proyecto, se plantearán posibles situaciones en las que será necesario aplicar ciertos conocimientos relacionados con la probabilidad y estadística. Uno de los objetivos por tanto es aprender dichas competencias necesarias para resolver estos problemas.

Finalmente, y como parte del proyecto, será necesario aplicar dichas técnicas definidas mediante la utilización del lenguaje de programación R, el cual nos ayudará a explorar gráficamente el conjunto de datos y comprender cuantitativamente las relaciones e hipótesis que nos sugiera la exploración.

1.3 Descripción de las tareas

De manera más extensa, podemos observar a continuación los objetivos específicos para conducir el desarrollo del trabajo.

- Determinar los fundamentos del problema planteado, así como establecer el conjunto de aplicaciones a analizar. Realizar una extracción de sus datos necesarios según el modelo de datos definido y crear un informe descriptivo de estas aplicaciones.

- Ofrecer una definición general de estos datos en base a aspectos como: El método utilizado para su adquisición, la necesidad e importancia de su extracción para el estudio, la utilidad que poseen dentro de la aplicación o sus características más importantes.

- Desarrollar, a través del lenguaje de programación R, un análisis estadístico clúster de estos datos y un modelo de predicción basado en regresión para obtener una idea más concisa de su comportamiento.

- Establecer una validación de los datos recogidos y redactar una memoria escrita donde se expongan todos los resultados del análisis realizado.

- Realizar finalmente una defensa y presentación oral del trabajo ante un tribunal.

2 Estudio del problema

Anteriormente ya planteamos una introducción más detallada del problema que se plantea en este estudio. Conocer los pasos a seguir que permitan abordarlo será la misión del estudio del problema, por lo tanto, en este apartado se definirá una introducción a los trabajos previos relacionados, la fuente de datos que se utilizará más adelante y una estructuración de las características de las aplicaciones mediante el modelo de datos.

2.1 Estado del arte

Conocer las técnicas, productos o publicaciones similares que han sido ya definidos para afrontar el problema planteado es lo que se conoce como estado del arte. Esto supone una gran ayuda a la hora de comenzar cualquier estudio, ya que nos permite conocer la situación previa del problema para establecer un punto de comienzo sobre el que realizar nuestro propio estudio y realizar mejoras sobre estudios que hayan sido desarrollados.

2.1.1 Minería y análisis de aplicaciones en Google Play

Se trata de un estudio publicado en el año 2013 en el cual se emplean técnicas estadísticas y probabilísticas con el objetivo de conocer las propiedades intrínsecas de las categorías de aplicaciones alojadas en la plataforma de Google Play [1].

¿Qué similitudes podemos encontrar reflejadas en este ejemplo? Este estudio pretende definir la situación de la tienda de aplicaciones, para ayudar tanto a empresas como desarrolladores a comprender la tendencia de las descargas de los usuarios y así poder adelantarse en la carrera de creación de aplicaciones que satisfagan el deseo de los clientes. En nuestro caso, la definición del ecosistema de las aplicaciones de probabilidad y estadística podrá ayudar de igual manera a posibles usuarios o desarrolladores a conocer el ámbito de esta categoría.

Otro concepto interesante planteado es la utilización de técnicas estadísticas que proponen en dicho estudio, el cual está dividido en dos partes. Mediante el análisis de correlación de las características de las aplicaciones, se estableció una fuerte relación negativa entre el número de descargas y el precio de las aplicaciones, así como la utilización de las aplicaciones respecto al precio. Observando una mayoría de descargas en las aplicaciones gratuitas y una mayor utilización si no eran de pago.

La segunda parte está asociada a la utilización de técnicas de análisis clustering y de modelización probabilística. Y es que empleando el algoritmo de K-means se definió que la categorización de las aplicaciones realizada por la propia Play Store no respeta adecuadamente la similitud de las aplicaciones. Ya que se incluía aplicaciones en categorías que no guardaban relación con su descripción.

Definen como conclusión la necesidad de una revisión por parte de Google Play de las aplicaciones de cada categoría, así como la implementación de mejoras en el área de las valoraciones u opiniones mediante el uso de reviews de repositorios comerciales, donde se estudia más a fondo dichas aplicaciones.

2.1.2 Evaluación de las aplicaciones de probabilidad y estadística para dispositivos móviles

Este estudio realizado en 2014 pretende examinar algunas de las aplicaciones móviles relacionadas con la probabilidad y estadística que permiten a los usuarios entender y aprender este campo, sin embargo, la principal diferencia que encontramos aquí es la utilización del App Store de Apple como fuente de datos [2].

Se procedió a clasificar un número de 37 aplicaciones desarrolladas tanto para iPhone como para iPad, ofreciendo una categorización de su contenido en base a la descripción que provee cada una de ellas. Aquí encontramos una gran similitud a nuestra idea propuesta, ya que para la creación del conjunto de datos deberemos establecer un contenido asociado a la aplicación que estamos estudiando.

Las categorías determinadas fueron las siguientes. Aplicaciones que permiten realizar operaciones con muestras de datos. Calculadoras de probabilidades, percentiles y distribuciones. Descriptivas, orientadas a la creación de gráficos. Paquetes estadísticos que permiten de mayor manera realizar análisis estadísticos sobre datos de entrada facilitados por el usuario como estimaciones, regresiones lineales o análisis correlativos. Paquetes educativos, como tutoriales, libros o aprendizaje. Y finalmente una última categoría, miscelánea, asociada a aplicaciones que no se albergan en ninguna de las categorías anteriores.

Y es que, tras realizar un informe descriptivo de cada una de las aplicaciones alojadas en estas categorías, el autor de este estudio concluye que la utilización de estas aplicaciones como herramientas para entender el área de la probabilidad y estadística es bastante negativa.

Por ejemplo, menos de la mitad de las aplicaciones hacen uso de las características dinámicas que ofrecen los smartphones. Tales como pellizcar con los dedos la pantalla para utilizar el zoom o realizar un barrido de las opciones mediante el deslizamiento a lo largo de la pantalla, únicamente seis de ellas hacen uso de este dinamismo. El autor define de igual manera que al menos un tercio de las aplicaciones contenían errores de programación o de diseño, y un cuarto de la muestra proveía ayuda para la comprensión de su metodología.

Así mismo, una gran cantidad de las aplicaciones albergadas en el App Store son versiones lite o de uso limitado, lo que disminuyen la adquisición de conceptos. No están categorizadas correctamente, al igual que el ejemplo anterior, hay algunos casos en los que determinadas aplicaciones no deberían ubicarse en esta área, ya que pueden producir confusión en los usuarios más inexpertos.

2.2 Fuente de datos

Una vez visto el planteamiento de las dos soluciones, es importante conocer de donde podemos extraer los datos necesarios de las aplicaciones. El estudio plantea la utilización de Google Play como fuente de datos, en cambio, la lógica nos dice que buscar en aquellas plataformas digitales donde haya más diversidad de contenido nos dará la clave para realizar un análisis más extenso. Y si, además de lo anterior, su accesibilidad a dicho contenido es sencilla y no depende de muchos patrones de búsqueda, tenemos la combinación perfecta para establecer nuestra fuente de datos para el estudio. Por eso en este apartado se analizará si la utilización de Google Play es correcta para nuestro estudio.

Una plataforma de distribución digital es un sistema mediante el cual podemos obtener contenido digital a través de internet sin tener que depender de la compra física. Se trata de un proceso rápido, sencillo y de completa disponibilidad, lo que favorece mucho su uso. Actualmente hay aproximadamente 13 tiendas digitales activas relacionadas con los sistemas operativos móviles, y 8 independientes [3]. La mayoría están basadas en sistemas operativos Android, algunas de las más conocidas son Google Play, Amazon Store y AppGallery de Huawei [3].

Cabe mencionar también a uno de los buques insignia, la App Store de Apple, que, aunque se desmarque de los dispositivos Android, está basada en los sistemas operativos iOS y iPadOS.

2.2.1 Google Play

Google Play fue lanzada al mercado de las plataformas de distribución digital el 22 de octubre de 2008 bajo el nombre de Android Market, hasta que en julio de 2013 tras su unión con la plataforma Google Music se renombró a Google Play, denominación por la que se le conoce actualmente.

Su lanzamiento y desarrollo tuvo un objetivo concreto, ofrecer contenido digital tanto gratuito como de pago a los usuarios que posean dispositivos móviles con sistema operativo Android. Posee un catálogo descargable muy variado, el cual puede obtenerse a través de la aplicación Play Store, ya incluida en estos dispositivos. Actualmente funciona también como tienda online para la compra de dispositivos móviles desarrollados por Google como smartphones, tablets u ordenadores [4].

Se trata del mayor portal de las plataformas de distribución digital, superando a su principal competidor la App Store tanto en número de descargas anuales, como en el catálogo de aplicaciones que ofrece. Por ejemplo, se calcula que el número de Apps disponibles para descargar en el Play Store durante el segundo cuarto del año 2020 asciende a los 2.870.000, mientras que por parte de la App Store el total alcanza 1.960.000 [5]. Véase la Figura 1.

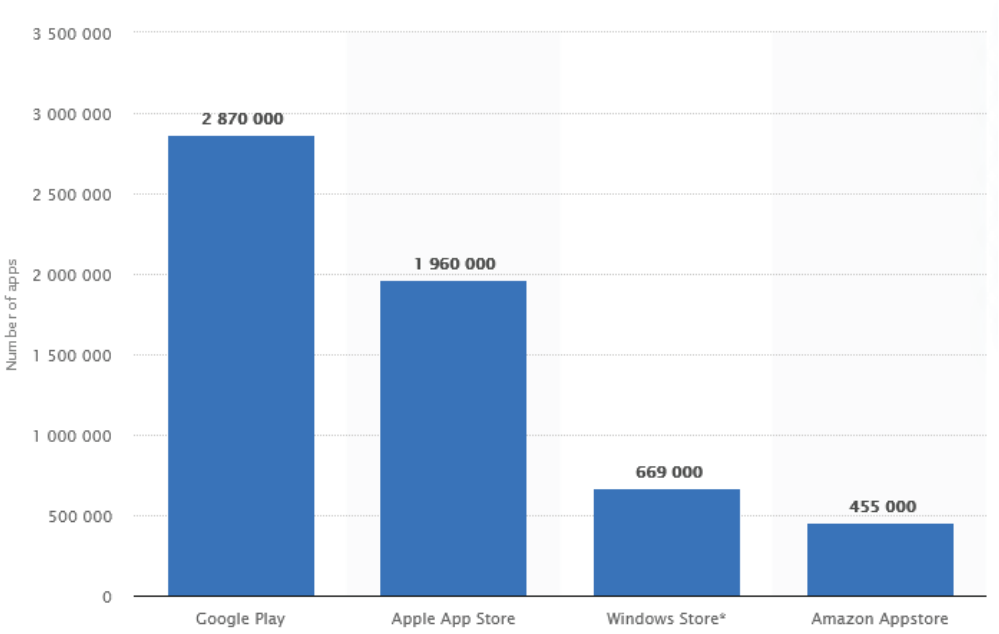


Figura 1: Número de aplicaciones en las principales tiendas de 2020.

Y es que, ofreciendo ese gran número de aplicaciones disponibles, cabe pensar en la gran cantidad de contenido que podemos llegar a encontrar en su interior. Su rango es muy variado, pero actualmente se encuentra dividido en varias secciones: Aplicaciones, Play Libros, Play Juegos, Play Música, Play Kiosco y Play Películas.

Abordamos la búsqueda de contenido relacionado con la probabilidad y estadística en la sección de Aplicaciones, y viendo que es la tienda digital que ofrece un mayor número catálogo, es posible que esta sea una de las razones para decantarse por ella.

Así mismo, otros aspectos aparecen a la hora de establecer Play Store como fuente de datos escogida. Posee un proceso de búsqueda, descarga e instalación de aplicaciones más sencillo que el que encontramos en App Store. Ya que principalmente, las aplicaciones a estudiar pueden ser descargadas en más tipos de dispositivos, el contrapunto del App Store es que únicamente estas aplicaciones pueden ser lanzadas en dispositivos Apple, lo que dificulta la extracción de los datos si no se posee un dispositivo de esta compañía.

Incluso a la hora de realizar una primera toma de contacto, podemos acceder al propio buscador de Google Play a través de internet, mientras que, para acceder al App Store, será necesario descargar previamente el software iTunes.

2.3 Modelo de Datos

Finalmente, el último paso, la preparación de los datos. Se debe conformar la estructura que van a poseer los elementos que serán medidos de las aplicaciones.

Preparar estos datos con la intención de operar con ellos es una condición necesaria dentro de la búsqueda de esta solución. Por lo tanto, se ha definido como estructura de datos a utilizar lo que se conoce como un dataset o conjunto de datos, técnica mayoritariamente usada en análisis estadísticos y totalmente válida para su utilización en un entorno de programación R. En cuyo caso dicha estructura de datos será conocida como data frame.

El objetivo de un dataset es establecer una matriz de datos única para realizar representaciones de datos de forma tabulada, de forma que las cuyas columnas de dicha matriz sean los posibles valores de las variables y las filas correspondan a cada una de dichas variables. En nuestro caso, las columnas serán representadas por las características a medir de las aplicaciones, y las filas cada una de las aplicaciones seleccionadas.

Podemos incluir como punto final, un breve diccionario provisional acerca de las variables que podría contener nuestro dataset, el cual será implementado más adelante una vez se haya avanzado en el proyecto. Una primera versión se expone a continuación:

- Nombre de la aplicación, donde se define su nombre por el que se la conoce en Google Play y su desarrollador.
- Contenido de la aplicación, donde se exponen las características que podemos encontrar en ella.
- El idioma en el que se encuentra la aplicación.
- El número de descargas actual de la aplicación.
- Coste o precio de la aplicación.
- El número de valoraciones ofrecidas por los usuarios y la media de sus puntuaciones.

3 Fundamentos para caracterizar las aplicaciones

3.1 Punto de partida

Existen múltiples razones y motivos por los que nos podemos guiar para estudiar nuestras aplicaciones. Ver que nos ofrece, que datos debemos tratar y cómo tratarlos pueden ser algunas de las preguntas que surgen cuando comenzamos con los primeros pasos, por eso siempre es conveniente tener una idea general de como caracterizar estas aplicaciones y definir una serie de criterios específicos a tener en cuenta.

Nos centraremos por lo tanto en características tecnológicas de la aplicación como la funcionalidad o la interfaz de diseño, así como en el grado de usabilidad y opinión que refleja el usuario que la ha utilizado.

3.2 Contenido

El primer criterio que se ha definido para caracterizar las aplicaciones ha sido el contenido. Esta característica es una parte indispensable en el proceso de creación, desarrollo y lanzamiento de una aplicación, ya que nos indica desde el objetivo que plantea, así como el público al que va a ser dirigida. Es gracias al contenido de una aplicación que podemos diferenciar en las plataformas de distribución a las aplicaciones móviles entre sí y decidir cual nos conviene más para su descarga.

El fundamento principal aquí es sencillo, debemos establecer un número de aplicaciones a estudiar relacionadas con los campos de la probabilidad y la estadística. Si realizamos una búsqueda superficial en Google Play, podemos encontrar distintos tipos de categorías, ya sea desde calculadoras de funciones estadísticas, aplicaciones educativas que ayudan a su entendimiento o juegos donde ponen a prueba tus conocimientos. En definitiva, se trata de generar etiquetas que permitan clasificar las Apps en grupos representativos de su diversidad.

No se pretende realizar una distinción exhaustiva entre ellas ya que cada aplicación tiene su propio objetivo, y si queremos realizar un estudio amplio, debemos abarcar el mayor número de posibilidades. Por lo tanto, uno de los fundamentos de la información que vamos a estudiar será el contenido que ofrecen, así como la categoría a la que pertenece en el Google Play Store.

3.3 Funcionamiento

Entrando en conceptos más técnicos, el funcionamiento de una aplicación nos indica la relación entre el contenido y la idea lógica de su uso. Ver cómo el usuario es capaz de explotar la aplicación es un punto importante que hemos de tener en cuenta en este estudio, ya que si bien es cierto que puede haber aplicaciones que ofrezcan un contenido similar, la manera de utilizarlas para lograr ese objetivo será distinta.

Para tener en cuenta los datos relacionados con el funcionamiento de las aplicaciones será necesario establecer algunos puntos clave que permitan analizar más en detalle estos aspectos técnicos. Para ello, podemos establecer una división del funcionamiento en tres conceptos más concretos como son la estructura de la aplicación, su usabilidad y la calidad de interfaz visual.

3.3.1 Estructura

Cuando pensamos en la estructura de una aplicación, podríamos considerarla como la manera en la se establece visualmente el menú de la aplicación en nuestra pantalla. Sin embargo, se pretende diferenciar el concepto de estructura con aquellos relacionados con el diseño visual, por eso se ha definido como la posibilidad de movimientos o patrones que posee la aplicación para llegar a un objetivo o mismamente para utilizarla.

El hecho de poder ver las distintas opciones del menú ya podría considerarse como un dato a extraer de las aplicaciones, otro punto fuerte sería medir la posibilidad de desplegables o transiciones entre pantallas para escoger la opción deseada, su lógica de uso o incluso la posibilidad de introducción de datos en la aplicación para operar con ellos.

3.3.2 Usabilidad

A través de esta cualidad se es capaz de medir la interacción entre el usuario y la aplicación. Se trata de uno de los puntos que caracterizan la calidad de una aplicación y que a su vez más cuesta conseguir exitosamente como desarrollador software. El número de patrones mínimos necesarios para su utilización, el modo de navegación que se establece en la aplicación o su sencillez son factores a tener en cuenta en las mediciones estadísticas que deseamos hacer [6].

Podremos agrupar bajo este concepto y con ayuda de la estructura a muchas de las aplicaciones relacionadas con la probabilidad y estadística. Por ejemplo, la mayoría de las calculadoras estadísticas poseen un patrón de uso y estructura similar, introducir los datos necesarios en celdas y aplicar la fórmula para obtener el resultado deseado.

3.3.3 Interfaz visual

Como último concepto relacionado con el funcionamiento, se ha seleccionado como objeto de estudio la interfaz visual de las aplicaciones.

Se trata de una medida más sencilla de contabilizar y estudiar que las anteriores, ya que de ella podemos extraer varios factores simplemente visuales que observamos en toda aplicación.

Conceptos como los colores de la estructura de la aplicación, el estilo que se ha empleado para el diseño, la tipografía y ortografía utilizada, ver si posee animaciones en la pantalla, iconos o el lenguaje con el que se describe la información son detalles que considerar en la extracción de datos de la interfaz de una aplicación [7].

3.4 Evaluación

Una vez hemos recogido los datos más técnicos de la aplicación, entramos quizás en la parte más subjetiva del proceso de desarrollo de las aplicaciones y que puede dar a una gran diversidad de datos a estudiar. La evaluación por parte del usuario una vez se ha lanzado la aplicación puede marcar la diferencia entre el éxito o el fracaso. Es por eso por lo que, a través de la plataforma de Android Google Play Store, podemos ver el feedback proporcionado, ya sea de manera individual como por ejemplo en base a los comentarios o de manera general, en base a la puntuación reflejada de las opiniones de la aplicación.

Se debe tener en cuenta que, en gran parte, es gracias a esta característica donde podemos ver las aplicaciones recomendadas a descargar en la plataforma de distribución. Por eso podemos considerar que la evaluación por parte de terceras personas es un elemento clave e indispensable de nuestro estudio de las aplicaciones de probabilidad y estadística.

Trabajar con números es la manera idónea de utilizar herramientas estadísticas, y que mejor manera de aprovechar esta capacidad con datos en los que evaluaremos el número de descargas, la cantidad de valoraciones recibidas, la media de dichas valoraciones o el precio de estas.

En términos generales, las variables extraídas de la evaluación pueden ser realmente útiles para definir el estudio. Utilizar el número de descargas o valoraciones para el análisis predictivo quizás posea más utilidad que conocer una predicción acerca de las posibles tipografías. Esto no indica que plantear predicciones sobre la interfaz visual de las aplicaciones sea erróneo, pero para este estudio, cobra más sentido utilizar variables basadas en la evaluación, ya que forman parte de un análisis más utilizado en la vida real para realizar predicciones o agrupaciones de características sobre aplicaciones móviles.

3.5 Creación del conjunto de datos

Tras desarrollar los fundamentos sobre los que se basa la recolección de variables y caracterización de las aplicaciones, se debe definir qué tipo de estructura final conformará el conjunto de datos. En el apartado del modelo de datos, se estableció un diccionario provisional en el que se definía distintas variables a extraer de las aplicaciones.

Finalmente, el conjunto final de estas ha sido seleccionado en base a la utilidad que respectan para los análisis y en base a ofrecer la mejor descripción de cada una de las aplicaciones. Esto es: características del contenido de la aplicación, funcionamiento y evaluación de esta. Además, distintos conjuntos de datos y estudios relacionados con la caracterización de aplicaciones han servido como ejemplo para determinar y contabilizar ciertas variables que en un primer momento no habían sido tenidas en cuenta [8]-[9].

1. *Nombre*: Nombre de la aplicación que posee en la plataforma Google Play.
2. *Desarrollador/es*: Estudio/s o autor/es encargado del desarrollo de la aplicación.
3. *Contenido*: Variable cualitativa nominal que indica el contenido o tipo de aplicación de probabilidad y estadística.
4. *Categoría*: Variable cualitativa nominal que indica la ubicación de la aplicación en la plataforma Google Play.
5. *Idioma*: Variable cualitativa nominal que indica el idioma de la aplicación.
6. *Elementos desplegables*: Variable cualitativa dicotómica asociada a la estructura que indica si la aplicación posee elementos desplegables.
7. *Barra de acción*: Variable cualitativa dicotómica asociada a la estructura que indica si la aplicación posee barra de acción.
8. *Sistema de navegación*: Variable cualitativa nominal asociada a la estructura que indica el modo de navegación de la aplicación.
9. *Inserción de datos*: Variable cualitativa asociada a la estructura que indica la posibilidad de inserción de datos.
10. *Transiciones entre pantallas*: Variable cualitativa dicotómica asociada a la estructura que indica si la aplicación ofrece transiciones entre sus pantallas.
11. *Panel lateral*: Variable cualitativa dicotómica asociada a la estructura que indica si la aplicación ofrece panel lateral.
12. *Usabilidad*: Variable cualitativa ordinal que indica el nivel de usabilidad a la hora de utilizar la aplicación.
13. *Calidad* de la interfaz visual: Variable cualitativa ordinal que indica la calidad de interfaz empleada en el desarrollo.
14. *Precio*: Variable cuantitativa continua que indica el *precio* de la aplicación.
15. *Descargas*: Variable cuantitativa que indica el número de *descargas* realizadas a través de la plataforma Google Play.
16. *Tamaño*: Variable cuantitativa continua que indica el *tamaño* que ocupa la aplicación.
17. *Valoraciones*: Variable cuantitativa continua que indica el total de *valoraciones* realizadas por los usuarios en la plataforma Google Play.
18. *Media de las valoraciones*: Variable cuantitativa continua que indica la puntuación *media de las valoraciones* en la plataforma Google Play.

En cuanto a la variable *contenido*, se ha decidido utilizar 6 posibles valores como dominio, los cuales resumen el contenido de cada una de las aplicaciones que albergan este estudio.

- Autoaprendizaje: Tutoriales, explicaciones o cursos de corta duración interactivos orientados al aprendizaje.
- Divulgación: Libros divulgativos o contenido explicativo acerca del área de la probabilidad y estadística.
- Recordatorios: Hoja de referencia de funciones, conceptos o fórmulas más utilizadas del área de la probabilidad y estadística.
- Informativas: Utilización de análisis estadísticos para ofrecer información o contenido acerca de un área concreta. p. ej. Aplicaciones financieras o aplicaciones probabilísticas deportivas.
- Calculadoras: Cálculo de funciones, hipótesis, operaciones o gráficos de estadística y probabilidad.
- Minijuegos: Aprendizaje de conceptos del área de la probabilidad y estadística en base a puzles, cuestionarios o juegos interactivos.

Seguidamente, se ha dividido la variable *idioma* estableciendo como dominio un conjunto de idiomas fundamentales. Los valores que puede tomar esta variable son los siguientes: Alemán, Árabe, Español, Francés, Hindi, Indonesio, Inglés, Italiano, Portugués o Multilingüe, el cual corresponde a aplicaciones que poseen más de uno de los idiomas nombrados.

Para referirse a la estructura de las aplicaciones no era recomendable establecer una única variable descriptiva, ya que imposibilitaba la creación de análisis estadísticos que tuvieran en consideración esta variable. Por esta razón, se ha agrupado este concepto en 6 opciones distintas, donde se recogen los aspectos más importantes relacionados con la estructura de las aplicaciones.

- Elementos desplegables: Un elemento despegable es aquel que ofrece cualquier tipo de información mediante un desplegable, tanto de manera horizontal o vertical. Los valores que puede tomar esta variable son: Si o No.

- Barra de acción: Ubicado normalmente en la parte superior de las aplicaciones, ofrece las principales funciones relacionadas con la lógica de uso de la aplicación como ajustes, información de la aplicación o búsqueda de contenido entre otras. Los valores que puede tomar esta variable son: Si o No.

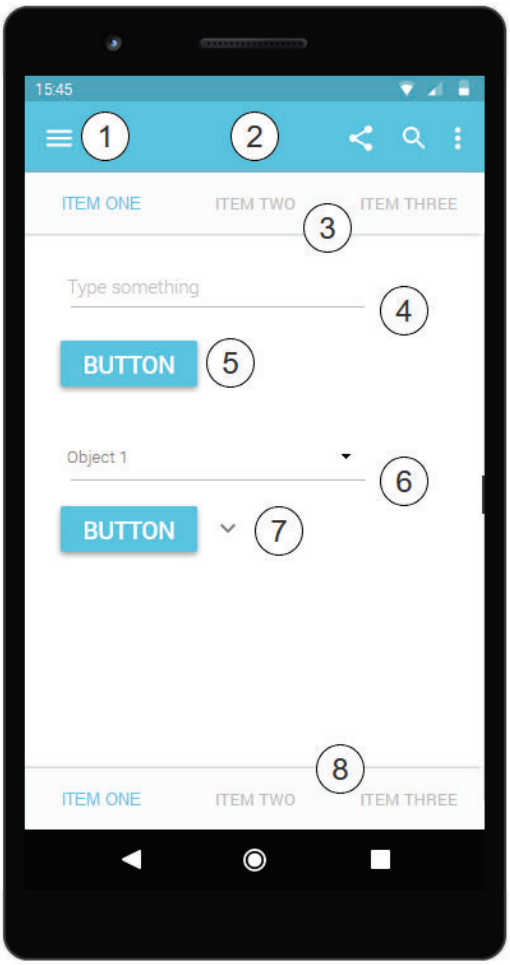
- Sistema de navegación: Ubicado en la pantalla principal, se trata del elemento esencial que permite movernos a través del contenido de la aplicación. Los posibles valores que puede tomar esta variable son: Accesos directos, Pestañas (superior o inferior) o Ambos.

- Inserción de datos: Permite conocer el modo que ofrece la aplicación para insertar cualquier tipo de dato o si por lo contrario carece de espacios habilitados para esta función. Los posibles valores que puede tomar esta variable son: Manual (a través del teclado), Seleccionables, Ambos o Ninguno.

- Transiciones entre pantallas: Indica si la información o el contenido de la aplicación es representada en una única pantalla o si, por el contrario, existen transiciones a través de distintas pantallas que la muestran. Los posibles valores que puede tomar esta variable son: Si o No.

- Panel lateral: Ubicado normalmente en la parte izquierda de la aplicación ofrece la posibilidad de seleccionar y navegar entre el contenido o las categorías de la aplicación. Los posibles valores que puede tomar esta variable son: Si o No.

A continuación, se añade una imagen ilustrativa para identificar cada uno de los elementos que conforman las variables relacionadas con la estructura, véase la Figura 2.



1. Panel lateral.
2. Barra de acción.
3. Sistema de navegación: Pestaña superior.
4. Inserción de datos de manera manual.
5. Sistema de navegación: Accesos directos.
6. Inserción de datos mediante: seleccionables.
7. Elementos desplegables.
8. Sistema de navegación: Pestaña inferior.

Figura 2: Elementos definidos para la estructura de una aplicación.

En lo que respecta a las secciones de *usabilidad* o interfaz visual, y de igual manera que en los casos anteriores, se ha establecido un conjunto de posibilidades que harán más sencillo la interpretación de este concepto.

Se proponen como dominio los valores de alta, media y baja, valores que la mayoría de análisis utilizan para referenciar valores ordinales [6].

Usabilidad:

- Alta: La aplicación se caracteriza por poseer un método de navegación intuitivo para el usuario y fácil de interpretar. Es efectiva a la hora de realizar las operaciones deseadas. El aprendizaje de su uso es sencillo. Todos los apartados de la aplicación funcionan a la perfección. El usuario refleja agrado mediante la sección de *valoraciones*.

- Media: El modo de navegación es agradable pero sus patrones no logran ser del todo intuitivos. El grado de efectividad es correcto. Pueden aparecer ciertos errores los cuales no interrumpen el uso de la aplicación.

- Baja: Posee ciertos errores de programación, visuales o de contenidos que dificultan la navegación. Existen apartados o enlaces que no funcionan correctamente. La aplicación no es intuitiva lo que dificulta su entendimiento. Las *valoraciones* ofrecidas por el usuario son negativas.

Calidad de la interfaz visual:

- Alta: La interfaz visual es agradable, la gama de colores elegida permite observar cada detalle perfectamente y no se solapa con ningún otro elemento. No hay sobrecarga de elementos visuales en la pantalla. El diseño de los iconos encaja perfectamente con el estilo y guarda relación con las funcionalidades que ofrece la aplicación. La aplicación posee animaciones visuales fluidas. La representación de la información ofrece una tipografía legible y acorde al diseño.

- Media: Posee una interfaz visual correcta, su gama de colores permite apreciar a grandes rasgos su funcionalidad. La tipografía utilizada es legible pero quizás existe algún fallo de ortografía. El diseño de los iconos es representativo con cada función. Las animaciones utilizadas en la aplicación son correctas. Puede existir algún fallo en dicha interfaz visual.

- Baja: El estilo de la interfaz entorpece el entendimiento de la aplicación. La gama de colores dificulta la visualización de ciertos elementos. Existe desplazamiento, fallos o solapamiento entre elementos estéticos que entorpecen la experiencia visual del usuario. La tipografía utilizada para representar la información no es legible o posee fallos ortográficos graves. No existe ningún tipo de animación visual o poseen errores.

4 Estudio de algunas soluciones

Definimos el estudio de algunas soluciones como una sección del documento en la que se describen más a fondo los tipos de análisis planteados en el inicio del estudio y el software requerido, así como una selección de las distintas técnicas que se podrían usarse para tratar las variables recolectadas del conjunto de datos.

4.1 Análisis Clúster

También conocido como algoritmo de agrupamiento o algoritmo de conglomerados, se trata de una técnica que permite agrupar objetos bajo un mismo criterio: la búsqueda de la mayor homogeneidad para cada grupo y la máxima diferenciación entre ellos [16]. Cada uno de los grupos resultantes tras la aplicación del análisis es lo que se conoce como clúster.

La realización de este tipo de algoritmo permitirá diferenciar conjuntos de aplicaciones dentro de la *categoría* de probabilidad y estadística. Los objetos *M* a dividir serán las aplicaciones seleccionadas. Así mismo cada una de estas poseerá *P* características, las cuales serán extraídas del dataset realizado.

El objetivo podría ser encontrar una división de las *M* aplicaciones en *C* grupos en base a sus características, de manera que cada una de estas *M* aplicaciones pertenezca únicamente a un grupo *C* [16].

4.1.1 Técnicas del análisis clúster

El hecho de poder establecer una categorización sobre este tipo de algoritmos pasa por definir con exactitud el concepto de "clúster". Por ello, no se concibe como una simple técnica capaz de ofrecer una única solución, más bien se define como un proceso iterativo basado en el ensayo y error, el cual depende normalmente del analista. Este tendrá que realizar una posterior validación, que indicará si se ha seleccionado el correcto algoritmo para formar los clústers.

El estudio ofrecido al comienzo de este documento menciona el uso del análisis clúster *K-means* sobre el cual se pretende generar un conjunto de grupos para clasificar las aplicaciones de Google Play [1].

Esta técnica podría llegar a servir como solución a nuestro estudio, sin embargo, conocer qué división queremos establecer y cómo debemos realizarla son pasos previos que deben ser definidos y que nos ayudarán a decidir el algoritmo a utilizar.

La lista de todas las técnicas puede ascender a más de 100 tipos distintos de algoritmos [17]. En el artículo "*Survey of Clustering Data Mining Techniques*" se presenta una descripción más detallada el conjunto de ellos [18].

Según este estudio, los métodos descritos a continuación agrupan a la mayoría de los algoritmos.

- Métodos de datos jerárquicos.
- Métodos de partición o división.
- Métodos de agrupación en redes.
- Métodos basados en la coocurrencia de datos categóricos.
- Métodos basados en restricciones.
- Métodos de agrupación usados en Machine Learning.
- Métodos de algoritmos escalables.
- Métodos para conjunto de datos de grandes dimensiones.

4.2 Análisis Predictivo

El propósito de este tipo de análisis recae en conocer el rendimiento de una variable, teniendo como objetivo principal predecir su comportamiento, patrón o tendencia. Por esta razón, el análisis predictivo conforma una serie de técnicas estadísticas que suelen aplicarse sobre cualquier evento desconocido para conocer los riesgos y oportunidades que este ofrece [10].

Comúnmente, este tipo de técnicas se emplean en el estudio de datos de marketing, financieros, sanitarios, o sociológicos. Sin embargo, un estudio realizado en el año 2014 consiguió demostrar que era posible predecir la cantidad de ventas que realizaría la empresa Apple del iPhone midiendo el grado de satisfacción de sus usuarios a través de la red social Twitter [11]. Con la ayuda de la regresión lineal, se pudo establecer un modelo que convertía dichos tweets en una predicción acerca de cómo se desarrollaría el mercado del iPhone en el próximo semestre. Así mismo, la correlación entre el número de ventas de los años pasados y el crecimiento de la cantidad de estas interacciones era altamente positiva cuando los usuarios expresaban su agrado en los tweets. Finalmente, los resultados ofrecieron una gran predicción con muy poco margen de error, y estableció la importancia que poseen las redes sociales a la hora de medir la satisfacción de los usuarios con un producto.

¿Pero cómo puede entenderse el modelo predictivo como una solución válida para nuestro problema planteado? Si tomamos en cuenta datos cuantitativos como el número de *descargas* de las aplicaciones, podremos llegar a observar el grado de fidelización que poseen los usuarios con las aplicaciones de ese entorno. De esta manera se podría llegar a interpretar si las aplicaciones relacionadas con la probabilidad y estadística forman parte de una *categoría* activa de la plataforma de Play Store u observar cuales son los factores que afectan a dichas *descargas*.

4.2.1 Técnicas del modelo predictivo

El conjunto de técnicas del modelo predictivo puede clasificarse de manera generalizada bajo dos grandes grupos, las técnicas de regresión y las técnicas de clasificación [10].

4.2.1.1 Técnicas de regresión

Se trata de una agrupación de técnicas cuyo objetivo es representar la interacción entre las distintas muestras de variables mediante la utilización de fórmulas matemáticas [10]–[12]. Estas tratan de ofrecer un resultado explicativo bajo las hipótesis estadísticas de error que ofrezca el modelo. Algunos de los ejemplos más comunes que tratan de explicar un objetivo numérico mediante un conjunto de variables predictoras son: el modelo de regresión logístico, el modelo de regresión lineal o los Árboles de clasificación y regresión [12].

4.2.1.2 Técnicas de clasificación

Comúnmente conocido como aprendizaje computacional, se trata de una clasificación de algoritmos basados en técnicas de regresión cuyo funcionamiento es mucho más complejo. El origen de estas técnicas pretendía establecer patrones de aprendizaje sobre máquinas, las cuales eran capaces de proveer resultados numéricos fiables mediante el análisis predictivo [10].

Sin embargo, dichos resultados carecían de fundamentos que permitieran componer relaciones entre las variables, por eso en la actualidad se trabaja para lograr que este tipo de técnicas sean capaces de emular el pensamiento humano a la hora de predecir posibles eventos. Aquí podemos encontrar algoritmos como: Los SVM o máquinas de vectores de soporte, las redes neuronales, las redes Bayesianas y los árboles de clasificación [12].

4.2.2 Modelo de regresión lineal

Perteneciente al género de los algoritmos de regresión OLS o mínimo cuadrados ordinarios en castellano, pretende analizar la relación existente entre una variable dependiente y una o más variables independientes del estudio [10].

La hipótesis de este modelo contempla que la predicción de la variable dependiente Y será el objeto del estudio, en cuyo caso estará asociada a la media de los valores que obtiene dicha variable. Para nuestro caso, un buen ejemplo sería el número de *descargas* de las aplicaciones de probabilidad y estadística [13].

Por otra parte, las variables independientes o predictoras Xi (donde $i>0$) serán las causas que afectan a la variable dependiente. Estas pueden ser por ejemplo el *precio*, la media de puntuación, o las *valoraciones de las aplicaciones*, ya que así se observará como estas variables afectan a dicho ecosistema en función de las *descargas* [13]. Así mismo, se debe tener en cuenta los valores de los coeficientes de regresión, que serán llamados como βp (donde $p>=0$) y que miden el grado de influencia de las variables independientes sobre la variable dependiente [14].

Finalmente, se debe añadir al final de la ecuación matemática el valor conocido como épsilon ε, el cual es aleatorio y se distribuye normalmente.

Este dato conforma el conjunto de valores no controlables que afectan a la ecuación y que convierte al modelo en no determinista [13]. La ecuación resultante del modelo sería similar a:

[15] $$Y_t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

Un ejemplo de los resultados puede verse a continuación, donde se utiliza la ecuación obtenida en forma de recta para situar los distintos valores que obtiene la variable dependiente en base a las variables independientes. Véase la Figura número 3.

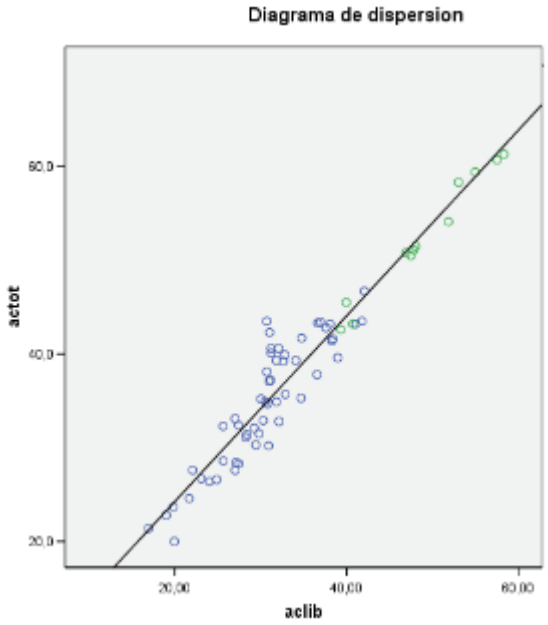


Figura 3: Diagrama de dispersión de un modelo de regresión lineal.

4.3 RStudio

RStudio es un entorno de desarrollo de libre uso orientado a la computación estadística y creación de gráficos mediante el lenguaje de programación R [19]. R fue creado como una reimplementación del lenguaje de programación S, el cual proporciona una amplia serie de técnicas estadísticas que lo convierten en el instrumento perfecto para su uso en minería de datos, finanzas o investigación científica entre otras.

Gracias a que se trata de un proyecto colaborativo, los usuarios pueden utilizar una gran variedad de paquetes y librerías las cuales extienden enormemente la funcionalidad de R [20]. Convirtiéndolo en un software actualizado, sencillo y capaz de emplear las herramientas estadísticas más modernas. Por lo tanto, se utilizará RStudio para la generación tanto del análisis predictivo como del algoritmo de agrupamiento, y de igual manera, para la obtención de los gráficos que permitan observar de una manera más sencilla los resultados de este estudio.

5 Desarrollo

Una vez definidas las variables a medir y la fuente datos utilizada, el proceso de desarrollo se dividió en tres grandes fases, las cuales concuerdan normalmente con cualquier tipo de estudio relacionado con el análisis de datos. Por lo tanto, durante la primera fase del desarrollo, estableceremos un proceso de adquisición de datos en crudo y su posterior limpieza, seguidamente aplicaremos la segunda fase, relacionada con el análisis en sí. Finalmente, la tercera y última fase estará relacionada con la validación, interpretación y documentación de estos, donde se muestran los resultados e ideas obtenidas.

5.1 Adquisición y limpieza de los datos

Para obtener los valores de las variables del conjunto de datos, debíamos establecer unos parámetros de búsqueda en la plataforma de distribución Play Store. Esto se realizó mediante la introducción de las palabras clave "*probability*" y "*statistics*" en la sección de Apps, para así maximizar el rango de búsqueda, acotando como primer idioma el inglés.

Un gran resultado de aplicaciones fue dispuesto, sin embargo, con el objetivo de facilitar el estudio, fueron escogidas aquellas cuya utilidad realmente tenía que ver con conceptos relacionados con la probabilidad y estadística. No se realizaron restricciones respecto al idioma, y fueron anotadas en su correspondiente casilla todas las posibilidades que ofrecía. De igual manera fue observada con buenos ojos la inserción de aplicaciones que carecían de evaluación por los usuarios, ya que nos darían una visión más amplia del estado del entorno.

Mediante la descripción que ofrecía el desarrollador de cada una de ellas, así como su utilización en un dispositivo Android, se recopiló toda esta información en una hoja de Microsoft Excel. En una primera versión se seleccionó un total de 100 aplicaciones, sin embargo, se consideró aumentar dicha cantidad debido a que el número de aplicaciones no funcionales era elevado. Tras varias modificaciones, la cifra final sería un total de 250. Una vez obtenida la versión del conjunto de datos comenzaba la etapa de limpieza de contenido.

Para esta segunda fase, se decidió en última instancia utilizar la herramienta OpenRefine, la cual había sido previamente utilizada en asignaturas como "*Semantic Web, Linked Data and Knowledge Graphs*" para el refinamiento, reconciliación y limpieza de datasets. Se importó la hoja de Excel y se procedió a modificar ciertos valores del conjunto de datos, así como eliminar el contenido no necesario mediante las funciones de tratamiento de columnas y celdas.

Algunos de los cambios más importantes realizados en esta fase tuvieron relación con la simplificación de algunas variables, un ejemplo concreto de ello ocurrió en el apartado del idioma.

Se diseñó una serie de idiomas fundamentales y se estableció la categoría de multilingüe, la cual acogía a aquellas aplicaciones que poseían varios valores. Este cambio era necesario ya que, si bien es cierto que la mayoría poseían un rango de entre uno y cinco idiomas, había casos en los cuales este número ascendía a 60 posibilidades distintas, esto aumentaba la cantidad de datos no tan necesarios que debíamos analizar.

Otro punto importante fue el reemplazamiento de la variable descriptiva "Estructura" por 6 categorías distintas cualitativas con el fin de ofrecer un posterior mejor análisis. Esta variable carecía de validez para nuestro estudio al no poseer una faceta o patrón uniforme para cada aplicación, por lo que, tras otro encuentro con el tutor, se acordó que lo mejor era establecer dicho cambio.

Así mismo, en las primeras versiones que conformaban el conjunto de datos existía una variable que medía la posibilidad de ofrecer contenido adicional de pago, esta fue cambiada por la variable *precio*, ya que era más representativa para el estudio.

Como punto final del proceso de limpieza de datos, se estableció como variable numérica las categorías asociadas a la evaluación, *precio* y *tamaño* de la aplicación, las cuales estaban construidas como variables de texto. Se insertó como unidad de medida el MegaByte para la columna de *tamaño*. Estas modificaciones ofrecían un mejor proceso de tratamiento de datos a la hora de realizar la interpretación de los gráficos necesarios, la creación del análisis de regresión o el de agrupamiento al ser únicamente cuantitativas.

5.2 Análisis de datos

El análisis de datos establecido para este estudio conforma tres grandes puntos: la aplicación de un modelo predictivo basado en regresión, un algoritmo de clústering que realice un agrupamiento del conjunto de datos y la creación de medidas de estadística descriptivas básicas y gráficos para cada una de las variables del conjunto. Todo ello mediante la utilización de la plataforma RStudio, ya que nos permite observar las cualidades de las aplicaciones.

5.2.1 Análisis general de las variables

Tomamos como primera variable a estudiar la *categoría* de las aplicaciones. Y como era de esperar, se observa una clara mayoría en el apartado de educación con un total de 134 aplicaciones, lo que equivale al 53.6% del conjunto. Esto es debido a la estrecha relación que guarda esta categoría con las aplicaciones cuyo *contenido* es el autoaprendizaje, las calculadoras, la divulgación y recordatorios de fórmulas en general, ya que están orientadas a ofrecer enseñanza al usuario.

La segunda *categoría* más representada es la de herramientas con 35 aplicaciones, en la cual prevalece la interacción aplicación-usuario para obtener resultados. Las calculadoras e informativas son aquellas que conformarán mayoritariamente esta sección.

Como dato curioso, con 27 aplicaciones aparece la *categoría* de deportes, relacionada con aplicaciones que ofrecen estadística deportiva y probabilidad de apuestas, únicamente informativas. Y con un total de 17 aplicaciones, la *categoría* de libros y obras de consulta es la cuarta con un valor más alto, la cual representa a grandes rasgos el *contenido* divulgativo.

Debido a la gran cantidad de variables, se ha decidido representar la relación entre las *categorías* de las aplicaciones con la cantidad de *valoraciones* y la media del conjunto *valoraciones* recibidas por *categoría* mediante la siguiente Tabla. La *categoría* más valorada de la muestra es educación, con un total de 86.702 *valoraciones*, mientras que la que posee una media más alta es arcade y social, con 5 respectivamente. Los valores representados con un guion en la *media de las valoraciones* indican que no ha sido posible extraer dicho valor.

Categorías	Número de aplicaciones	Valoraciones	Media de las valoraciones
Educación	134	86.702	4.1
Empresa	2	108	4.7
Herramientas	35	14.886	4.0
Productividad	9	536	4.1
Libros y obras de consulta	17	14.090	3.9
Estilo de vida	1	14	4.9
Deportes	27	3.275	4.1
Medicina	3	165	4.2
Comunicación	1	13	4.0
Educativos	1	7	4.0
Entretenimiento	4	387	4.3
Música y audio	1	41	3.6
Noticias y revistas	2	0	-
Mapas y navegación	1	0	-
Personalización	2	0	-
Puzles	2	6.620	4.8
Casual	1	0	-
Cartas	1	6.508	4.5
Arcade	1	7	5.0
Social	1	5	5.0
Total (21 categorías)	250	133.423	-

Tabla 1: Categoría de las aplicaciones.

Respecto al *contenido* de las aplicaciones, se ha desarrollado un diagrama de barras el cual se representa en la Figura 4. Este refleja la relación entre el *contenido* de cada una de las 250 aplicaciones de probabilidad y estadística y la cantidad de apariciones.

Las calculadoras, con un valor de 86 aplicaciones, ocupan el 34.4% de los datos recogidos, siendo estas el área mayoritaria del conjunto de datos. Tras esto, 61 aplicaciones están relacionadas con aplicaciones informativas, el cual equivale al 24.4% del total.

Les siguen con el 14.8% y un 14.4%, las aplicaciones de autoaprendizaje y divulgación. Como porcentajes finales tenemos con un 6% las aplicaciones que conforman el *contenido* de minijuegos y recordatorios, siendo estas dos áreas las menos representadas.

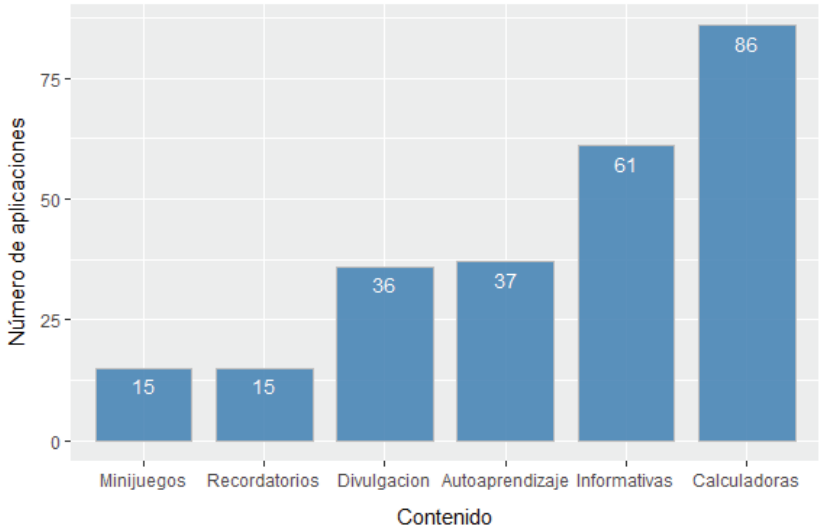


Figura 4: Diagrama de barras del contenido de las aplicaciones.

La Figura 5 representa en un diagrama de cajas, la relación entre el *contenido* de las aplicaciones respecto a su *media de sus valoraciones*. Como puntos rojos se representan los valores atípicos de cada contenido, estos valores son de gran importancia, ya que podrían influir en la distribución normal del posterior análisis de regresión. A través del *tamaño* de la cajas podemos conocer la distribución de las puntuaciones de cada contenido, siendo las más dispersas en calculadoras y minijuegos. Respecto al contenido de aplicaciones que posee una mejor media de *valoraciones*, aparecen los minijuegos y recordatorios, mientras que el peor valorado es la divulgación, con un 3.8 sobre 5. Un análisis completo de este diagrama se encuentra representado en la Tabla 2.

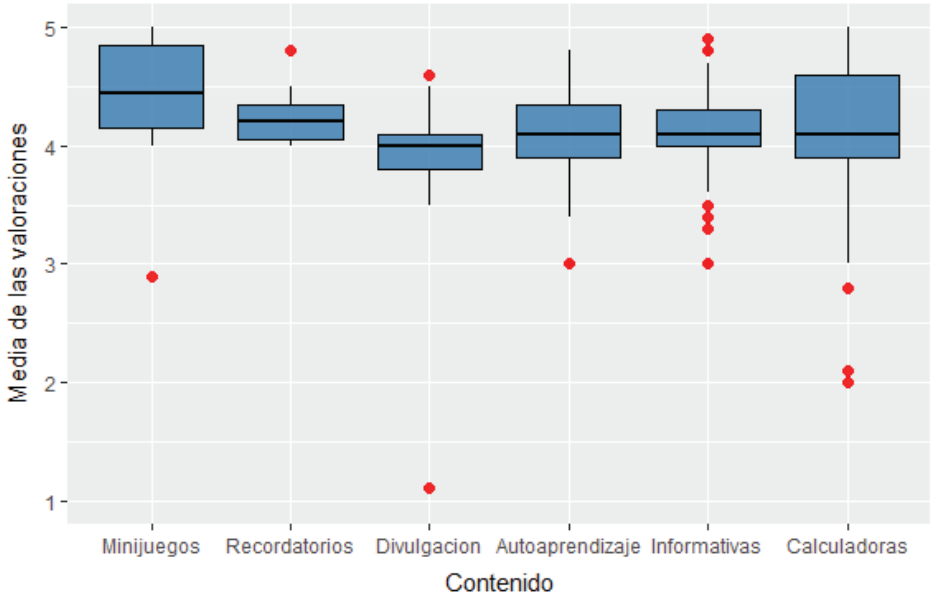


Figura 5: Diagrama de cajas del contenido y media de valoraciones.

	Autoaprendizaje	Calculadoras	Recordatorios	Divulgación	Informativas	Minijuegos
Q1	3.9	3.9	4.0	3.8	4.0	4.1
Mediana	4.1	4.1	4.2	4.0	4.1	4.4
Q3	4.3	4.6	4.3	4.1	4.3	4.8
Media	4.0	4.1	4.2	3.8	4.1	4.3
Máximo	4.8	5.0	4.8	4.6	4.9	5.0
Mínimo	3.0	2.0	4.0	1.1	3.0	2.9

Tabla 2: Estadísticos descriptivos del diagrama de cajas del contenido.

En cuanto a la estructura de las aplicaciones, no se ha podido establecer un posible gráfico de barras apiladas debido a la gran cantidad de variables, por ello se ha decidido crear otra Tabla que muestre la frecuencia de aparición de las aplicaciones que poseen dicho elemento. Véase la Tabla 3.

	Número de aplicaciones	Porcentajes
Elementos desplegables		
Si	158	63.2%
No	92	36.8%
Barra de Acción		
Si	143	57.2%
No	107	42.8%
Sistema de navegación		
Accesos directos	158	63.2%
Pestañas	55	22%
Ambos	37	14.8%
Inserción de datos		
Manual	74	29.6%
Seleccionables	29	11.6%
Ninguno	112	44.8%
Transiciones entre pantallas		
Si	174	69.6%
No	76	30.4%
Panel lateral		
Si	83	33.2%
No	167	66.8%
Total (6 categorías)	250	100%

Tabla 3: Tabla de la estructura de las aplicaciones.

- Elementos desplegables: Como se puede observar, el 63.2% sí poseen esta característica, esto es debido a que se trata de un elemento de representación de la información óptimo y sencillo para el usuario. Sin embargo, aquellas que no los poseen (36.8%) suelen concentrarse en las aplicaciones de tipo calculadora o divulgación, ya que su estructura normalmente está pensada en ofrecer un único contenido sin necesidad de establecer criterios o filtros que modifiquen dicho resultado.

- Barra de acción: Este elemento permite ubicarnos entre sus categorías y acceder a las funcionalidades más elementales de la aplicación. Por ello, aquellas aplicaciones que carecen de transiciones entre pantallas a la hora de mostrar los resultados suelen escasear de este complemento, véase las calculadoras.

- Sistema de navegación: No se observa una clara relación entre el sistema de navegación de las aplicaciones con el contenido de estas. Pero a grandes rasgos podemos decir que los accesos directos predominan en este entorno, siendo las calculadoras el contenido que más aprovecha este acceso a la información.

- Inserción de datos: Es una función representativa en aplicaciones que requieren normalmente un conocimiento previo e interacción superior por parte del usuario. Por ello, encontramos esta característica en las aplicaciones calculadoras, informativas y minijuegos. Esto es debido a que su inserción manual (43.6%), es necesaria para proveer números, letras o símbolos que permitan realizar los cálculos o gráficos apropiados.

  Sin embargo, esta característica puede repercutir en una posible disminución de la usabilidad, ya que, al ofrecer un método manual, el usuario podría llegar a introducir valores incorrectos, entorpeciendo así la experiencia.

  Por esta razón, un 25.6% de estas incluyen la inserción seleccionable. Aquí las aplicaciones informativas poseen un gran papel, ya que mediante seleccionables el usuario es capaz de decidir qué valor estadístico desea observar, calcular o representar. Esto ofrece un proceso más sencillo de toma de resultados, ya que al a ver un número finito de opciones seleccionables, nunca se introducirán valores erróneos.

- Transiciones entre pantallas: El estudio refleja que un 69.6% de aplicaciones poseen dicha función. La mayoría se encuentran asociadas a aquellas con múltiples categorías que requieren de un mayor espacio para representar su contenido, lo que repercute en una división de este a lo largo de varias transiciones.

  El autoaprendizaje debido a las distintas transiciones para avanzar a lo largo del tutorial, la divulgación para navegar entre los capítulos del libro o los resultados ofrecidos por las calculadoras o informativas que requieren una posterior transición a la ejecución de las operaciones son ejemplos de contenido que no puede ser mostrado en una única pantalla principal.

- Panel lateral: Debido a la representación de información a través de accesos directos que ofrecen este tipo de aplicaciones, el panel lateral no es un elemento estructural tan necesario, ya que el usuario podrá desplazarse entre las categorías únicamente desde la pantalla principal. Esto puede observarse en sus porcentajes, ya que el 66.8% de las aplicaciones carecen de este elemento.

Seguidamente, el análisis descriptivo de la categorías de *usabilidad* muestra una evaluación positiva de la *usabilidad* por parte de los usuarios, la cual se ve reflejada así mismo en la media de *valoraciones* y la caja de comentarios.

Si observamos la Figura 6, obtenemos que 215 aplicaciones poseen un grado de *usabilidad* alto o medio, lo que equivale al 86% del conjunto de aplicaciones medidas.

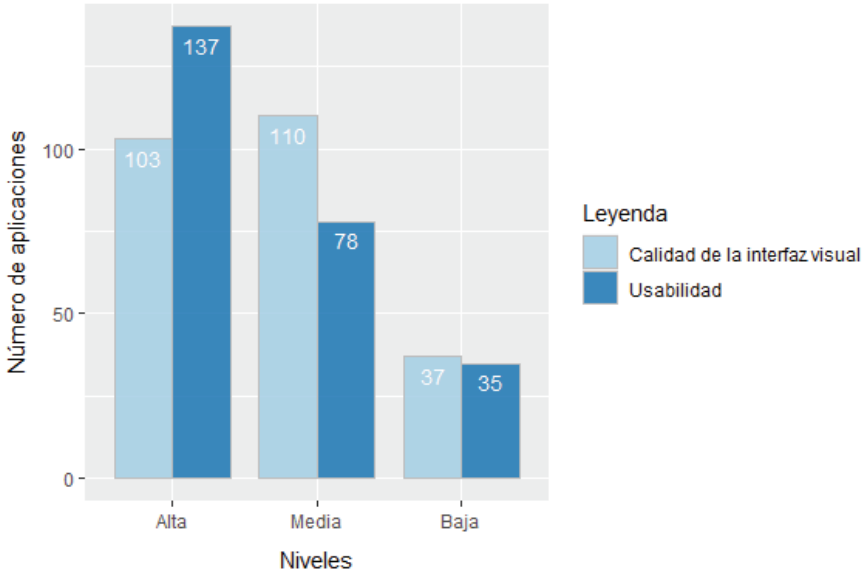


Figura 6: Diagrama de barras de la usabilidad y calidad de la interfaz.

Por esta razón, es comprensible pensar que aquellas aplicaciones que tienen un alto grado de *usabilidad* posean la mayoría de elementos definidos para la estructura, ya que una buena *usabilidad* está altamente ligada a una estructura de aplicación óptima.

Y efectivamente observamos esta relación. De estas 137 aplicaciones, el 66.4% sí poseen elementos *desplegables*, así mismo, un 60% contienen *barra de acción* y finalmente, el 73.7% también ofrece *transiciones* a la hora de representar contenido entre pantallas.

Esto nos indica que esta mayoría cumple con las expectativas que habían sido definidas para el área de la *usabilidad*, ofreciendo un funcionamiento completo de todos sus apartados, un contenido intuitivo y sencillo, así como patrones ágiles gracias a los *desplegables* y *transiciones*.

La calidad de interfaz visual sin embargo refleja una necesidad de mejoría en cuanto al diseño nos concierne. Si bien es cierto que existen 103 aplicaciones con un alto grado de calidad, la mayoría poseen una *interfaz visual de calidad* media o baja, lo cual se resume en un 58% del total de la población estudiada.

Se ha obtenido este valor debido a un gran número de aplicaciones que poseían elementos visuales en la pantalla los cuales entorpecían ocasionalmente la visión del contenido, una elección de gama de colores que contrastaba negativamente con sus elementos y con el fondo de pantalla, así como una utilización de tipografía que no acompañaba al diseño de la aplicación.

Como última sección de variables a analizar del conjunto de datos, tenemos aquellas relacionadas con la evaluación del usuario y el *tamaño* de la aplicación. Estas variables poseen valores únicamente cuantitativos, ya sea de carácter discreto o continuo. Representamos mediante un diagrama de barras la variable discreta acumulativa del número de *descargas* de las 250 aplicaciones contabilizado por Google Play. Véase en la Figura 7.

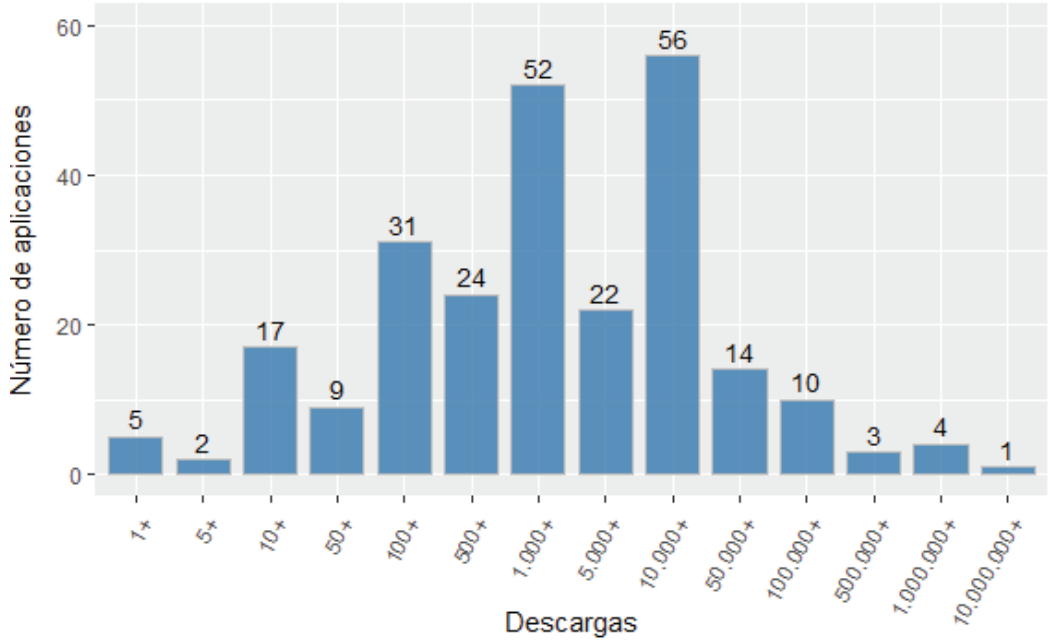


Figura 7: Diagrama de barras del número de descargas de las aplicaciones.

Para este tipo de variable hay que tener en cuenta su factor acumulativo, ya que puede llegar a inducir dudas respecto a su medición. Por lo que hay que discernir entre el conjunto de estas que superan una cantidad de *descargas*, o aquellas que están situadas en un rango concreto.

Un 20.8% está relacionado con las aplicaciones (52) de más de 1.000 *descargas*, este dominio aplica a aquellas cuyo rango oscila entre las 1.001-5.000, ambos valores incluidos. De igual manera, un 22.4% del total representa a las aplicaciones (56) que alcanzan más de 10.000 *descargas*, este rango oscila entre las 10.001-50.000, ambos igualmente incluidos.

Si la observamos de manera global, y deseamos conocer el número de *descargas* de cada tipo de aplicaciones, basta con realizar otro diagrama de barras estableciendo el número de *descargas* como una variable numérica no acumulativa.

Por lo tanto, la Figura 8 deberá interpretarse como una aproximación mínima del número de *descargas*, ya que como hemos repetido antes, los valores que ofrece Google Play para esta variable se encuentran dentro de un rango, no siendo cantidades exactas.

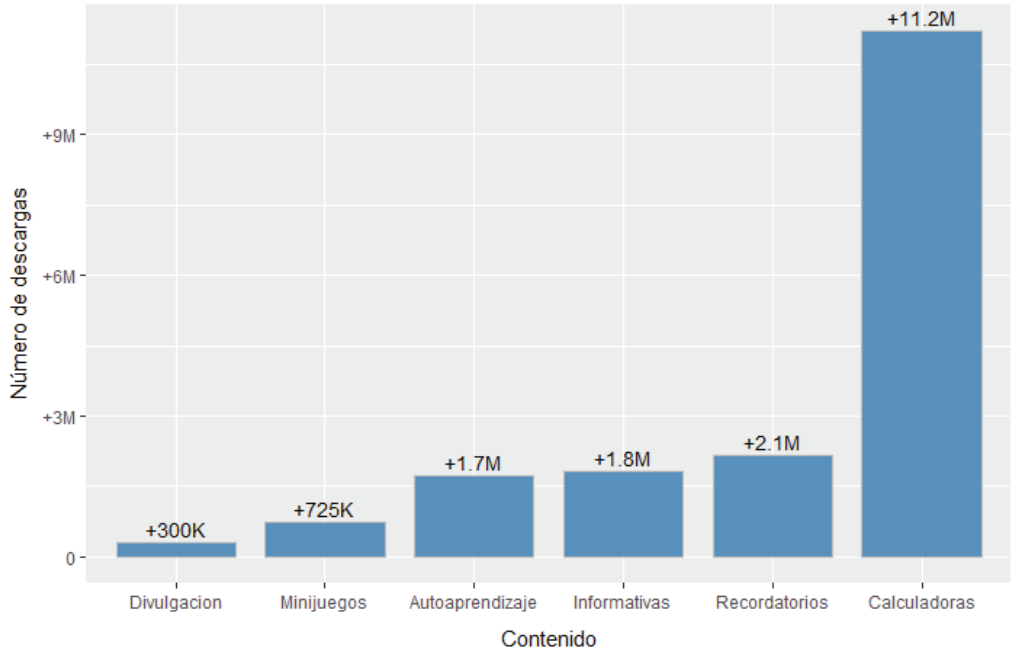


Figura 8: Histograma de descargas del contenido de las aplicaciones.

Las calculadoras se convierten así, en el *contenido* de aplicaciones más descargado del conjunto de datos, con más de 11 millones de *descargas*. Todo esto debido a la aplicación "GeoGebra Calculadora Gráfica", cuyo valor extremo asciende a más de 10 millones de *descargas*.

Una vez acabado el análisis de las variables discretas y nominales, saltamos al conjunto de variables continuas conformadas por el *tamaño* de aplicación, el *número de valoraciones* realizadas y la *media de las valoraciones*. Se propone la utilización de comandos como *summary()* en RStudio que permiten obtener medidas adicionales para este tipo de variables.

Por otra parte, el empleo de histogramas es la perfecta herramienta para representarlas gráficamente, ya que se observa la distribución de frecuencias de cada una de las variables. Con el fin de ofrecer la mejor interpretación visual en cada histograma, se ha aplicado previamente la regla de "Freedman–Diaconis", la cual permite conocer el ancho ideal de cada caja del histograma a través del IQR [21].

El rango de *tamaño*s de las aplicaciones medidas en el análisis es muy variado, desde los 0.086MB de *tamaño* de aplicación asociados a una calculadora, hasta los 81MB de espacio de almacenamiento que requiere una aplicación de *contenido* informativo. Debido a este contraste de valores, la media y mediana del *tamaño* del conjunto de aplicaciones poseen valores dispares, 8.9MB y 4.6MB respectivamente.

Todas estas estadísticas reflejan una distribución asimétrica de la frecuencia de aparición, la cual puede ser apreciada en el histograma de la Figura 9.

Tras esto, el primer cuartil está establecido en 2.9MB, lo que significa que el 25% de las aplicaciones posee un *tamaño* menor o igual a este valor, mientras que el tercer cuartil en 10MB, lo que equivale a que el 75% de estas posee un *tamaño* menor o igual que dicha cantidad.

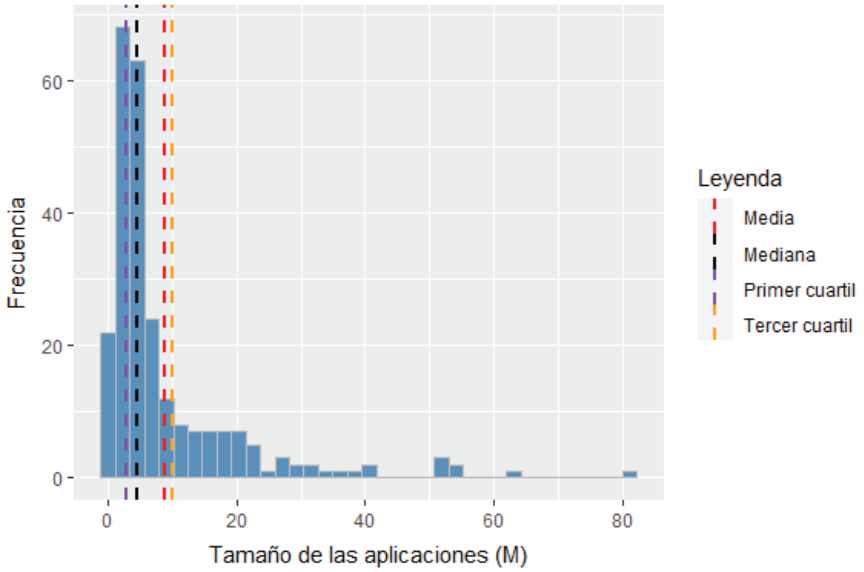


Figura 9: Histograma de frecuencias absolutas del tamaño (M).

Con el fin de obtener una mejor representación del histograma anterior, se realiza un cambio a escala logarítmica del valor del *tamaño* de las aplicaciones. De esta manera se puede observar que el histograma acoge una distribución normal de dicha variable a lo largo del eje x, con valores de media y mediana más próximos respecto al histograma anterior. Véase la Figura 10.

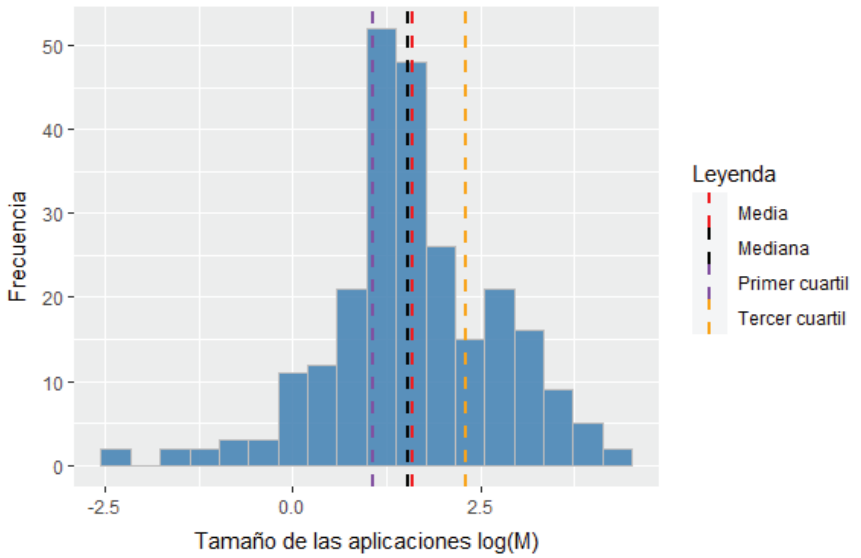


Figura 10: Histograma de frecuencias absolutas del tamaño log(M).

Respecto a la cantidad de *valoraciones*, existían aplicaciones las cuales no habían sido evaluadas. Por ello, para este subconjunto, se ha asociado la cantidad de 0 evaluaciones, con el fin de obtener el mínimo de celdas con valores vacíos. Obteniendo de esta manera, 195 aplicaciones que sí han sido evaluadas por parte de los usuarios.

De esta muestra, existe una aplicación cuyo *contenido* está relacionado con las calculadoras la cual posee una única valoración por parte de los usuarios, mientras que, del mismo contenido y con 39.927 *valoraciones*, la aplicación "GeoGebra Calculadora Gráfica" se convierte en la más valorada. El número total de *valoraciones* realizadas en el entorno por parte de los usuarios asciende a un total de 133.423.

Para finalizar, nos centraremos en la variable más significativa relacionada con la evaluación de las aplicaciones de probabilidad y estadística, la *media de las valoraciones* ofrecidas por los usuarios. Al igual que el *tamaño* de las aplicaciones se trata de una variable continua, por lo que será interesante analizar sus medidas descriptivas básicas de estadística.

El valor más alto corresponde a cinco aplicaciones cuyo *contenido* está relacionado con las calculadoras y minijuegos, donde se registra una puntuación de 5 sobre 5, el máximo para cualquier aplicación. Por el contrario, el valor mínimo es un 1.1 sobre 5, el cual posee la aplicación divulgativa "SPSS Data & Analysis".

La media y mediana total de *valoraciones* corresponden a un mismo valor, 4.1 sobre 5, lo que podría llegar a inducir en una posible distribución uniforme de la variable. Por otra parte, el primer cuartil se sitúa en un 3.9, por lo que un 25% de las aplicaciones poseen una media de *valoraciones* menor o igual a este valor, mientras que el tercer cuartil está ubicado en 4.4 de media.

De manera global, observamos que 55 aplicaciones no ofrecen esta información, sin embargo, las 195 restantes poseen una buena media de *valoraciones*, con un 73.8% de *valoraciones* superiores o iguales a 4.0. Se ven representados de manera gráfica todos estos conceptos en el histograma de la Figura 11.

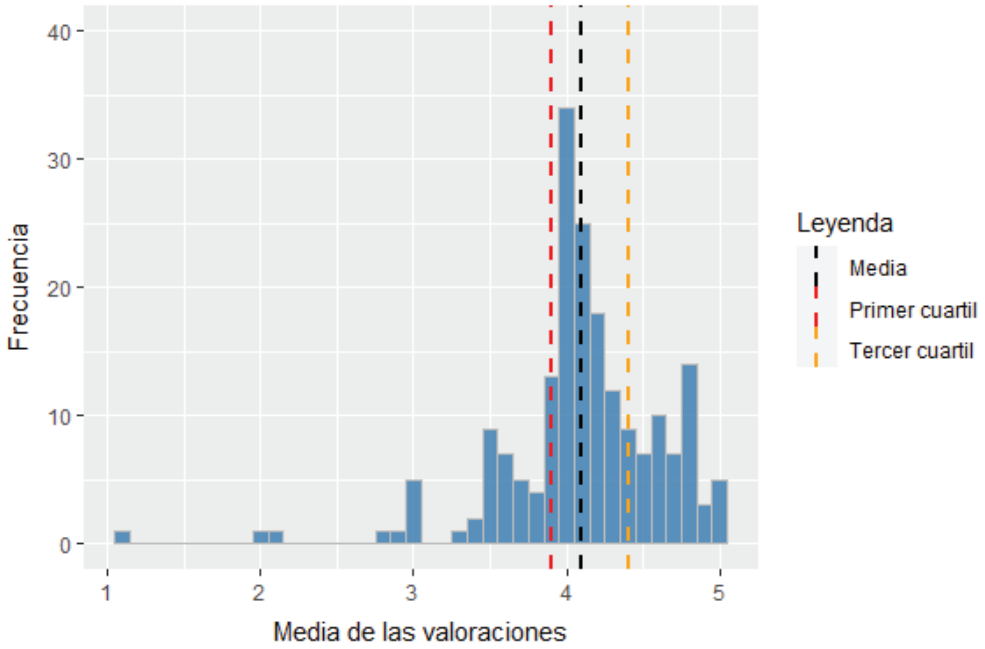


Figura 11: Histograma de frecuencias absolutas de la media de valoraciones.

5.3 Regresión lineal simple

La necesidad de predecir la evolución del número de *descargas* de las aplicaciones de probabilidad y estadística en la plataforma Google Play es el objetivo que propusimos al inicio de este estudio. Por ello se estableció de entre todos los tipos de análisis predictivos, el uso del algoritmo de regresión simple.

Este fue descrito de manera general en la sección 4.1, conociendo así un poco más su funcionamiento y características más relevantes. En este apartado mostraremos su proceso de desarrollo paso por paso, mientras que una documentación de resultados y validación será descrita más adelante.

Empezamos seleccionando entre las posibles variables independientes que ayuden a predecir y establecer la fórmula de comportamiento de la variable dependiente del número de *descargas*, la cual ha sido extraída a modo de vector del conjunto de datos, y transformada a escala logarítmica con el fin de obtener resultados más representativos.

5.3.1 Correlaciones lineales

El primer paso es obtener mediante los análisis de correlación del coeficiente Pearson y Tau de Kendall las correlaciones entre las distintas variables del estudio.

Pearson suele mostrar resultados fiables frente a variables cuya distribución suele ser normal, sin embargo, el coeficiente de Tau de Kendall está orientado a aquellas cuyos valores pueden agruparse en rangos, están acumulados en una región, y quizás no presentan una distribución normal, razones por las que se ha decidido así mismo utilizarlo [22].

Con un intervalo de confianza del 95%, se ha encontrado correlación entre la variable dependiente número de *descargas* respecto a las variables predictoras del número de *valoraciones* y *precio* de las aplicaciones. Estos valores de correlación muestran una significancia $\alpha < 0.05$, lo que equivale a que dichas medidas si son significativas, permitiéndonos rechazar la hipótesis nula H_0 y continuar con el desarrollo [22].

Los resultados mostrados en la Tabla 4 determinan un nivel medio de correlación negativa entre el *precio* de las aplicaciones y el número de *descargas*, mientras que el número de *valoraciones* y el número de *descargas* muestran un nivel medio de correlación positiva. No se ha obtenido correlación significativa entre la media de *valoraciones* (-0.006007714) o el *tamaño* de las aplicaciones (-0.01152177) en relación al número de *descargas*.

	Número de descargas	
	Nivel de correlación (r)	Significancia (p-value)
Pearson		
Número de valoraciones	0.3902799	1.1602e-10
Precio de las aplicaciones	-0.3537317	8.819e-09
Kendall		
Número de valoraciones	0.6840943	2.2e-16
Precio de las aplicaciones	-0.4042288	1.255e-14

Tabla 4: Tabla de correlaciones entre variables.

5.3.2 Generación de los modelos

Mediante la función *lm()* de RStudio, se han generado dos modelos de regresión [23]. Estos poseen el número de *descargas* como variable dependiente, siendo el *precio* de las aplicaciones y el número de *valoraciones* recibidas por aplicación las variables independientes o predictoras respectivamente. A continuación, se muestran los diagramas de dispersión, donde se representan los valores de las variables mediante puntos y las rectas de regresión obtenidas para cada modelo. Véase las Figuras 12 y 13.

En concreto, la Figura número 12 muestra un alto rango de *descargas* asociado a las aplicaciones gratuitas (cercanas a 0). Esta relación quizás podría explicarse de mejor manera mediante, un modelo que acoja un rango de aplicaciones que posean un *precio* significativo y otro modelo relacionado únicamente con aquellas que son gratuitas. Podemos ver en la recta dibujada de color negro la correlación negativa antes medida.

De igual manera sucede con la Figura número 13, cuya representación muestra una concentración del número de *descargas* en un rango de *valoraciones* pequeño, por lo que el modelo lineal presentado podría dividirse en dos distintos, estableciendo cada uno de ellos un límite o umbral de *valoraciones*. Se observa de igual manera en la recta dibujada la correlación positiva entre ambas variables.

Finalmente, podemos comprobar en sendas Figuras que no existe una distribución uniforme de los valores de las variables, así como una relación no lineal entre la variable independiente y dependiente, lo que dificultaría la validación del modelo lineal escogido, dando a entender definitivamente que otro tipo de modelo podría explicar esta relación. Sin embargo, al obtener valores de correlación significativos se decidió continuar.

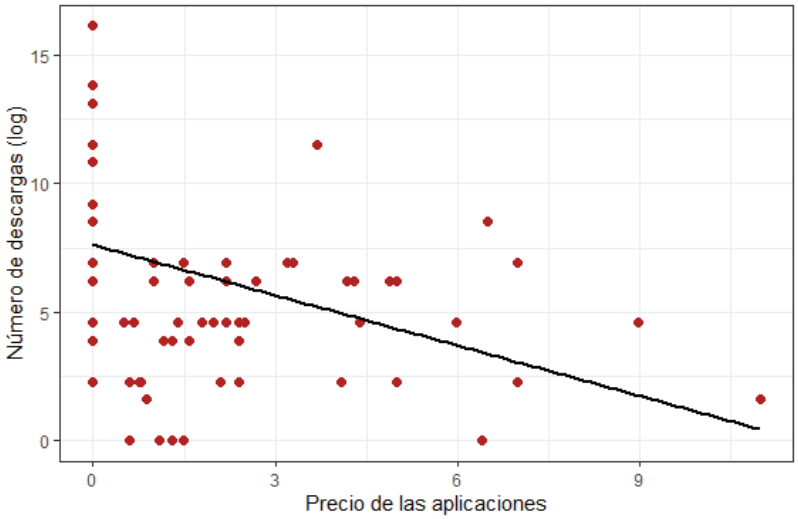


Figura 12: Diagrama de dispersión Descargas ~ Precio.

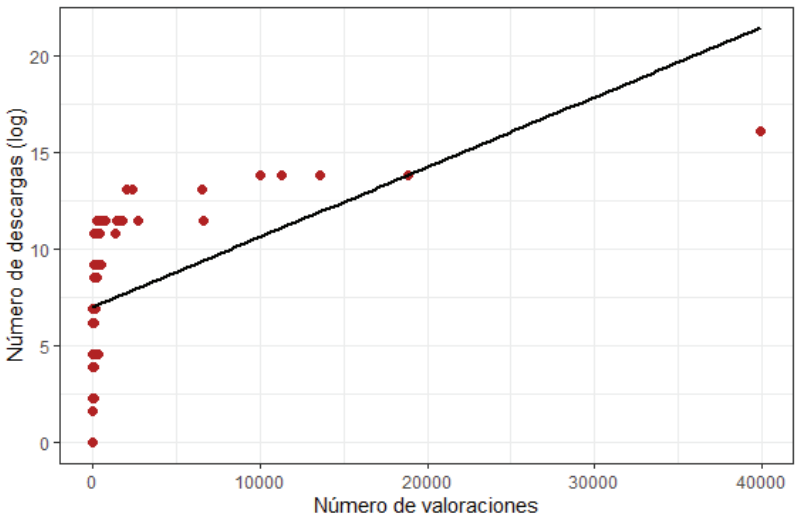


Figura 13: Diagrama de dispersión Descargas ~ Número de valoraciones.

Como último apunte acerca de las Figuras generadas, se pretendió establecer de igual manera las variables *precio* y número de *valoraciones* a escala logarítmica. Pero tras realizar dichos cambios, se observó que los diagramas generados de ambos modelos eran muy poco representativos, y de igual manera, se obtuvo valores infinitos asociados a las aplicaciones con *precio* y *valoraciones* igual a cero, por lo que dicho cambio acabo desestimándose.

5.3.3 Resultados e interpretación

De igual manera que para la obtención de los datos descriptivos, se utiliza la operación *summary()* de RStudio con el fin de obtener un resumen de cada uno de los modelos generados, los cuales deben ser interpretados [22].

Fórmula: Descargas ~ Precio

Residuals:	Min	1Q	Median	3Q	Max
	-7.2301	-1.4001	-0.1047	1.5957	8.5034
Coefficients:	Estimate	Std. Error	t-value	Pr(>|t|)	-
(Intercept)	7.6147	0.1818	41.887	< 2e-16	-
Precio	-0.6519	0.1095	-5.956	8.82e-09	-
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Residual standard error: 2.694 on 248 degrees of freedom					
Multiple R-squared: 0.125 Adjusted R-squared: 0.1216					
F-statistic: 35.47 on 1 and 248 DF, p-value: 8.819e-09					

Tabla 5: Resultados del modelo de regresión Descargas ~ Precio.

Fórmula: Descargas ~ Valoraciones

Residuals:	Min	1Q	Median	3Q	Max
	-7.0442	-2.4390	-0.1389	2.1407	5.3441
Coefficients:	Estimate	Std. Error	t-value	Pr(>|t|)	-
(Intercept)	7.044e+00	1.702e-01	41.399	< 2e-16	-
Valoraciones	3.609e-04	5.406e-05	6.676	1.6e-10	-
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Residual standard error: 2.651 on 248 degrees of freedom					
Multiple R-squared: 0.152 Adjusted R-squared: 0.1489					
F-statistic: 44.56 on 1 and 248 DF, p-value: 1.602e-10					

Tabla 6: Resultados del modelo de regresión Descargas ~ Valoraciones.

La sección superior *Residuals* relaciona la diferencia entre los valores observados de la variable dependiente *descargas*, y los estimados por ambos modelos para dicha variable. En ambos casos, se ha observado una distribución no aleatoria sobre el valor cero del eje x (no representada en el documento), indicando nuevamente la no linealidad de ambos modelos.

La columna *Estimate* representa el valor aproximado que obtendrían los coeficientes del modelo lineal (β_0 y β_1) equivalentes a la ordenada y la pendiente en la ecuación generada. Y un poco más a la derecha, observamos el valor del *p-value*, representado mediante Pr(>|t|) en cada uno de los coeficientes.

Se pretende determinar si los efectos de la constante y de cada variable independiente son significativos para explicar la variable predictora o si, por el contrario, pueden considerarse nulos. Para estos modelos generados estos valores Pr(>|t|) son menores que 0.05 y significativamente distintos a cero.

Los valores de R^2, *Multiple R-squared y Adjusted R-squared,* cuyos rangos están situados en [0-1], indican la bondad del ajuste realizado en el modelo respecto a los datos, siendo valores cercanos a 1 un ajuste del modelo correcto. En nuestros modelos los valores de R^2 son respectivamente 0.125 y 0.152, lo que representa que ambos modelos son capaces de explicar el 12.5% y 15.2% de la variabilidad del número de *descargas* frente al *precio* y número de *valoraciones*, resultados realmente poco relevantes para explicar la variable predictora.

Finalmente, el resumen obtenido del estadístico F (*F-statistic)* muestra de igual manera un valor de p significativamente distinto a cero, el cual había sido medido anteriormente en el análisis de correlación de Pearson. Por lo tanto, las ecuaciones generadas para cada modelo podrían ser las siguientes:

*Número de Descargas (log) = 7,6147 – 0,6519 * Precio de las aplicaciones*

*Número de Descargas (log) = 7,044 + 0,3609 * Número de valoraciones*

5.3.4 Validación

Como último paso de la generación del modelo, se deben verificar las condiciones necesarias para validar el modelo de regresión lineal simple utilizado, o, por el contrario, utilizar otro modelo de análisis predictivo con el que obtener mejores resultados. Para ello, nos centraremos en los residuos generados después de crear ambos modelos lineales, los cuales serán analizados y dispuestos de manera gráfica [22].

Respecto a los modelos generados podemos concluir que su la aplicación de un modelo de regresión no la mejor opción. Esto es debido a que las características necesarias que permiten establecerlos tales como la normalidad de residuos, linealidad y homocedasticidad no se cumplen al completo.

La Figura 14 muestra los resultados gráficos de los residuos del primer modelo. En su parte izquierda se puede comprender que los datos no se distribuyen de forma lineal, ya que no están dispuestos de forma aleatoria a lo largo del cero, si no que ocupan una gran mayoría la parte derecha del gráfico.

El gráfico Normal Q-Q nos habla acerca de la distribución de los residuos, la cual como vemos, no adquiere su condición de normalidad. Esto puede comprobarse de igual manera mediante la creación del Test de Saphiro sobre los residuos. El test arroja un valor de p = 8.11e-05, y al ser menor que 0.05 debe rechazarse la normalidad.

La medida de la homocedasticidad puede comprobarse de igual manera mediante gráficos, sin embargo, se ha preferido utilizar el Test de Breusch-

Pagan sobre el modelo generado. Aplicándolo en RStudio mediante la función *bptest()*, obtenemos un valor de p = 0.5648, siendo este mayor que α = 0.05 y sin poder rechazar la hipótesis nula H_0 se consigue la homocedasticidad de los datos.

De manera consecutiva, analizamos los residuos del segundo modelo relacionado con el número de *descargas* y la cantidad de *valoraciones* recibidas por los usuarios. En la izquierda de la Figura 15 se observa de igual manera que los residuos no están distribuidos a lo largo del cero.

La representación del Normal Q-Q también ofrece una distribución no normal como en el primer caso. El Test de Saphiro muestra en este caso un valor de p = 6.979e-08 menor que 0.05. En este punto, podíamos invalidar el segundo modelo, ya que no cumplimos con estas dos propiedades. El valor del Test de Breusch-Pagan en este caso otorga un valor de p = 0.04302, el cual al ser menor que 0.05 rechaza la hipótesis nula H_0 y considera que no se cumple la homogeneidad de varianzas de este segundo modelo.

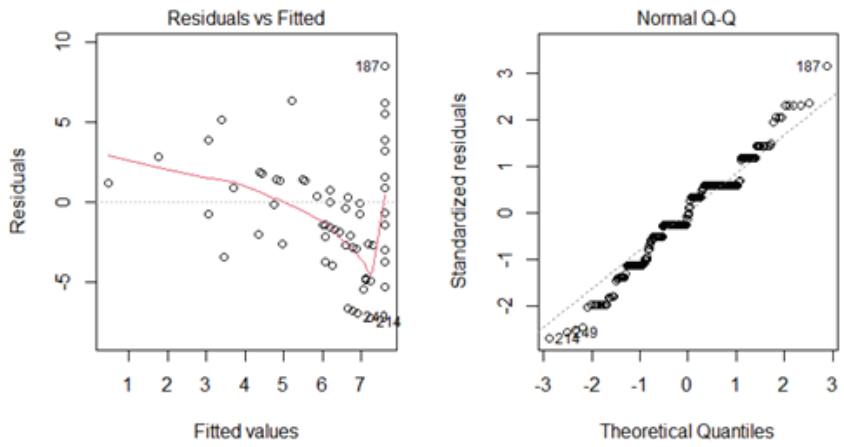


Figura 14: Resultados de los residuos del modelo Descargas ~ Precio.

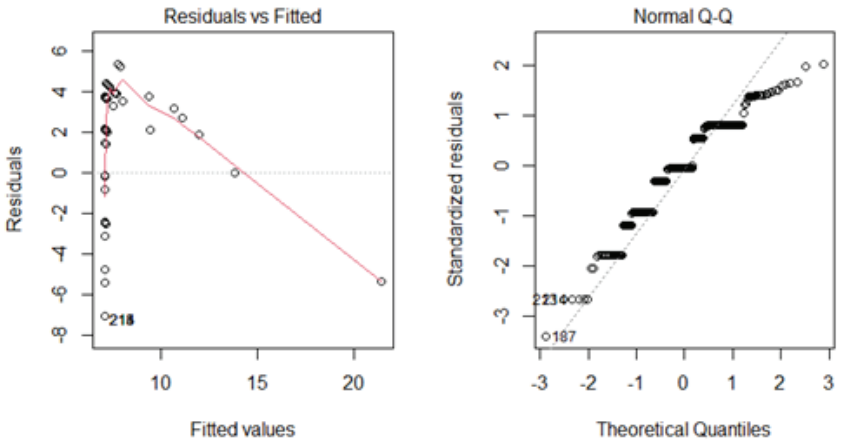


Figura 15: Resultados de los residuos del modelo Descargas ~ Valoraciones.

5.4 Regresión logística simple

La no validación de los modelos de regresión realizados implica un proceso de búsqueda de otras soluciones que puedan representar la relación planteada entre el número de *descargas* y las demás variables obtenidas del conjunto de datos. Realizar un modelo de regresión lineal múltiple parecía ser la opción acertada, ya que podríamos modelizarlo en base a los resultados obtenidos de los ejemplos anteriores. Sin embargo, su creación y evaluación suponían un nivel de dedicación superior, por ello, tras una reunión con el tutor, se decidió no contemplar dicha opción debido a su dificultad y grado de entendimiento.

Por esta razón, se ha planteado reformular dichos modelos utilizando el método de regresión logística simple, el cual es válido para variables cuya condición de distribución normal es prescindible. En este apartado de igual manera mostraremos la realización de dicho algoritmo y su correspondiente validación.

5.4.1 Generación de los modelos

Comenzamos con la creación de los modelos, transformando previamente la variable dependiente numérica del modelo de regresión lineal a variable dicotómica, en nuestro caso el número de *descargas*. Para ello se estableció 5.000 como cantidad de *descargas* mínima a superar, convirtiendo las aplicaciones que no alcanzaban dicha cantidad en 0 y 1 para las que poseían valores superiores.

Tras esta modificación, el modelo de regresión logístico intentaría calcular la probabilidad de que una aplicación superase este número de *descargas* en base a la cantidad de *valoraciones* recibidas y el *precio* de la aplicación [24].

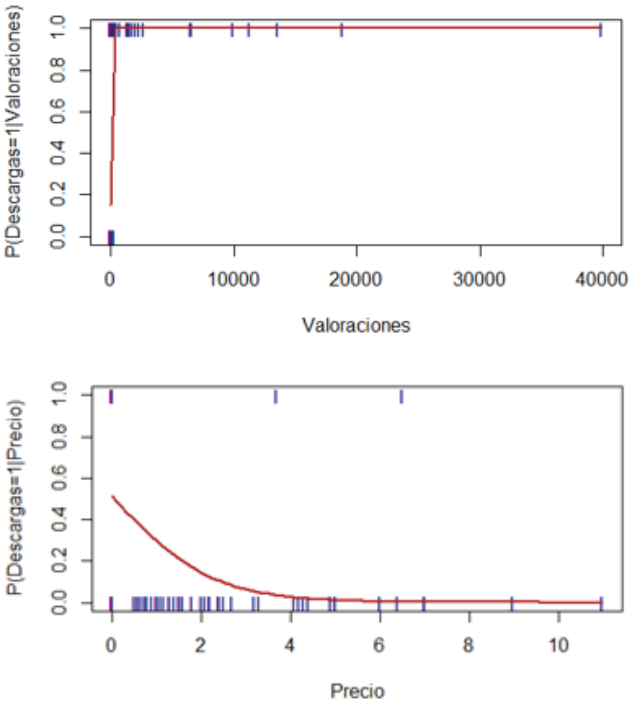


Figura 16: Representación gráfica de los modelos de regresión logística.

La Figura 16 muestra de manera gráfica las probabilidades que tomarían las predicciones en los modelos generados. Cabe recalcar, que en la imagen superior de la Figura 16, la curva definida para las probabilidades toma ese aspecto debido a la diferencia de unidades de medida tan dispares entre el ejes de *valoraciones* y probabilidad de *descargas*.

Aun así, se ve claramente la tendencia al alza en el modelo gráfico de las *valoraciones*, donde vimos con anterioridad como ambas variables poseían una relación positiva de aumento. Lo contrario sucede en su parte inferior, donde parece ser que la probabilidad de *descargas* disminuye a medida que las aplicaciones comienzan a ser de pago.

5.4.2 Resultados e interpretación

Establecemos mediante la función de RStudio *glm()* los modelos correspondientes, y seguidamente con el comando *summary()* se obtiene un resumen de las características más significativas (Tablas 7-8) [24].

Fórmula: Descargas ~ Valoraciones

Deviance Residuals:	Min	1Q	Median	3Q	Max
	-4.2007	-0.6501	-0.5750	0.5826	1.8212
Coefficients:	Estimate	Std. Error	z value	Pr(>|z|)	-
(Intercept)	-1.716269	0.231324	-7.419	1.18e-13	-
Valoraciones	0.038464	0.006034	6.375	1.83e-10	-
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Null deviance: 342.96 on 249 degrees of freedom Residual deviance: 218.55 on 248 degrees of freedom AIC: 222.55					
Number of Fisher Scoring iterations: 11					

Tabla 7: Resultados del modelo logístico Descargas ~ Valoraciones.

Fórmula: Descargas ~ Precio

Deviance Residuals:	Min	1Q	Median	3Q	Max
	-1.1969	-1.1969	-0.3751	1.1580	3.4247
Coefficients:	Estimate	Std. Error	z value	Pr(>|z|)	-
(Intercept)	0.04574	0.13937	0.328	0.742759	-
Precio	-0.91021	0.26215	-3.472	0.000516	-
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Null deviance: 342.96 on 249 degrees of freedom Residual deviance: 312.97 on 248 degrees of freedom AIC: 316.97					
Number of Fisher Scoring iterations: 6					

Tabla 8: Resultados del modelo logístico Descargas ~ Precio.

En ambos modelos aparece de nuevo la columna *Estimate,* que representa el valor que obtendrían los coeficientes para la intersección representados por el logaritmo de *odds* de que una aplicación pase el umbral de las 5.000 *descargas*. Este se calcula elevando el número *e* dicha cantidad.

Más abajo surge el coeficiente de regresión, donde se relacionará este logaritmo de manera positiva o negativa según el resultado de *precio* o *valoraciones*. Esto indica que, por cada unidad incrementada de la variable *valoraciones* o *precio*, se espera que dicho logaritmo de la variable *descargas* incremente o disminuya su valor en promedio de 0.038464 y -0.91021 unidades.

Finalmente, en la columna de Pr(>|z|) de ambos modelos, los valores de los predictores contribuyen al modelo al ser menores que la significancia establecida en 0.05. Siendo en este caso p (*valoraciones*) = 1.83e-10 y p (*precio*) = 0.000516.

5.4.3 Validación

La validación de cualquier modelo de regresión logística simple debe realizarse analizando el conjunto en sí como el valor de los predictores obtenidos de la ecuación. Por ello, si observamos cierta mejoría en el modelo predicho respecto a su modelo sin predictores, se podrá concluir que el modelo generado es significativo. El análisis de varianzas, o *ANOVA* nos ayudará a comprender dicha validación mediante la utilización del *Wald chi-test*, el cual puede emplearse en RStudio mediante la función *anova()* en ambos modelos [24].

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
Null			249	342.96	
Valoraciones	1	121.41	248	218.55	< 2.2e-16
Signif. codes:	0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Tabla 9: Test ANOVA del modelo logístico Descargas ~ Valoraciones.

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
Null			249	342.96	
Precio	1	29.997	248	312.97	4.326e-08
Signif. codes:	0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Tabla 10: Test ANOVA del modelo logístico Descargas ~ Precio.

Se debe observar si existe mejoría en cuanto a la diferencia de residuos en ambos modelos. Esta se calcula mediante la diferencia entre el modelo primario (null) y el predicho, valores que son tomados de la columna *Resid. Dev*, y que establecen una diferencia de residuos de 124.41 para el modelo de *descargas* y *valoraciones*, así como 29.99 para *descargas* y *precio*, por lo que sí existiría una mejoría y nos acercaría a una validación correcta.

La última columna establece el valor del *p-value* del *Likelihood ratio*. Como en otras ocasiones, se deberá asegurar que este obtiene valores menores a 0.05. Si nos fijamos en la columna Pr(>Chi), obtenemos valores de < 2.2e-16 y 4.326e-08, por lo que los coeficientes establecidos para ambos modelos sí son significativos, lo que equivaldría en última instancia a la validación correcta de ambos modelos.

Como paso final, debemos comparar dichas predicciones en base a las observaciones establecidas, pero para ello establecemos como límite el valor de 0.5. Este límite medirá la probabilidad de obtener un numero de *descargas* inferior o superior a 5.000 dependiendo del valor de probabilidad obtenido, si la probabilidad es menor a 0.5, establecerá un 0, el cual indica que no ha superado las 5.000 *descargas*, en el caso contrario se establecerá un 1, indicando que sí.

La matriz de predicciones obtenida mediante el límite 0.5 establece que, el modelo logístico de *descargas* y *valoraciones* permite clasificar correctamente el 129+76/34+11+76+129 = 0.82, 82% de las observaciones realizadas, mientras que el modelo de *descargas* y *precio* clasifica el 48+108/2+48+92+108 = 0.624, 62.4%. Cabe recalcar, que para la interpretación de nuevas medidas de muestreo habría que tener en cuenta un factor de error.

Definimos pues que ambos modelos no son directamente comparables en rendimiento. Respecto al *precio*, el modelo ofrece valores inferiores a Beta0 (interceptor) = 7.6, pues su pendiente es negativa. En el caso del modelo de las *valoraciones*, este da valores superiores a Beta0 = 7.0, debido a su pendiente positiva. Por lo que en esta predicción ambos errores son mayores que el respectivo obtenido de la regresión logística. Véanse la Figuras 12 y 13.

Las fórmulas probabilísticas generadas para cada modelo son las siguientes:

$$\text{P (Descarga} > 5.000) = \frac{e^{-1.716269 + 0.038464 * Valoraciones}}{1+e^{-1.716269 + 0.038464 * Valoraciones}}$$

$$\text{P (Descarga} > 5.000) = \frac{e^{0.04574 - 0.91021 * Precio}}{1+e^{0.04574 - 0.91021 * Precio}}$$

Por ejemplo, si proponemos una aplicación gratuita que ha obtenido 70 *valoraciones*, la probabilidad de que supere las 5.000 *descargas* será del 72.6%. Sin embargo, si dicha aplicación tuviera 30 *valoraciones*, la probabilidad descendería al 36.3% debido a la relación positiva que poseen estas dos variables.

Respecto al *precio*, se verifica la relación probabilística negativa que tiene respecto al número de *descargas*. Una aplicación simplemente por el hecho de ser gratuita tendría una probabilidad del 51.1% de alcanzar las 5.000 *descargas*. Con un *precio* 8.99€ su probabilidad sería tan solo del 0.03% y con un *precio* económico de 0.99€ ascendería al 29.8%.

5.5 Algoritmo de agrupamiento

Hemos descubierto gracias a los algoritmos de regresión realizados que existe relación entre el número de *descargas* que posee una aplicación con su número de *valoraciones* y *precio* que posee en la Play Store. Sin embargo, quedan otras cualidades numéricas que no han sido aprovechadas debido a la no existencia de correlación entre ellas, estamos hablando de la puntuación de las *valoraciones* que los usuarios otorgaban a cada una de las aplicaciones.

Y es que anteriormente mencionamos que la *media de las valoraciones* recibidas es una cualidad muy importante a la hora de entender el funcionamiento de nuestro entorno, por esta razón trataremos de implementar un algoritmo de clúster que permita obtener una visión más extensa del panorama teniendo en cuenta el *precio* de las aplicaciones, y las variables del dataset asociadas al apartado de evaluación, todas ellas de carácter numérico.

5.5.1 Selección del algoritmo

Para esta realización se propusieron a lo largo de las reuniones varias opciones, siendo el algoritmo de clúster *k-means* la opción primordial. Sin embargo, tras obtener las medidas estadísticas descriptivas principales asociadas a cada una de estas variables numéricas, se comprobó la existencia de valores atípicos o extremos que entorpecían la realización de este tipo de algoritmo.

Aquí es donde aparece el algoritmo *k-medoids clustering,* cuyo funcionamiento es bastante similar a *k-means,* ya que de igual manera es el propio analista el que selecciona el número de k grupos a crear, pero se diferencia en su robustez frente a valores atípicos y su medida para representar los agrupamientos. El algoritmo más utilizado para este método es *PAM* (*Partitioning Around Medoids*) [25].

Por un lado, *k-means* ofrece como medida principal el centroide, el cual corresponde al promedio de todas las observaciones de la muestra, mientras que *k-medoids* se distingue por el uso del *medoid*, el cual es un elemento correspondiente a dicho clúster cuya distancia de él respecto a sus vecinos es la menor posible.

El primer paso consistía en establecer el número de grupos k para nuestra muestra de las variables numéricas *precio*, *valoraciones*, *media de las valoraciones* y *descargas*. Se decidió no incluir el *tamaño* de las aplicaciones en el algoritmo de agrupamiento ya que se consideraba una variable poco representativa para el agrupamiento y podía alterar el funcionamiento de este. De igual manera se transformó el número de *descargas* a escala logarítmica para una mejor interpretación visual debido a los altos valores que posee en comparación con las otras variables.

Por ello, tras realizar un escalamiento del conjunto de variables, la función *fviz_nbclust()* del método *elbow* con la distancia *manhattan* (debido a la presencia de valores atípicos) permitió conocer el número de k donde la curva comenzaba a estabilizarse [26]. Véase la Figura 17. Observamos finalmente que k = 5 era la medida correcta a establecer.

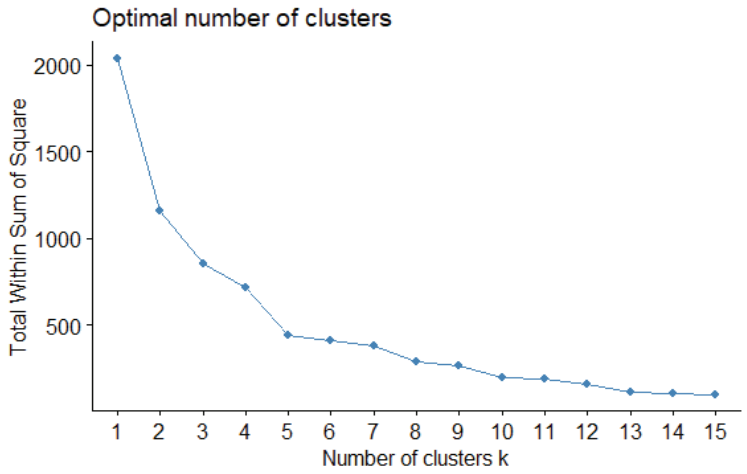


Figura 17: Representación del número de k clústers para el modelo PAM.

Una vez obtenido dicho valor. Se procedió a realizar una comparativa del ancho medio de las siluetas generadas (una por cada grupo) para *K-Means* y PAM [25]. Esta comparativa muestra la medida de la proximidad de cada punto de un clúster respecto a los puntos de los clústeres vecinos y, por tanto, repercute en una forma de evaluar visualmente lo bien que se ha clasificado cada grupo. Los valores de las siluetas cercanos a 1 muestran una agrupación correcta entre puntos de un mismo clúster y negativa entre clústeres vecinos.

El conjunto de siluetas para cada algoritmo poseía una media de anchura = 0.5 (Figura 18). Por lo que, al poseer una media de proximidad de distancias similares, se determinó escoger PAM debido a la robustez frente a los extremos que ofrece.

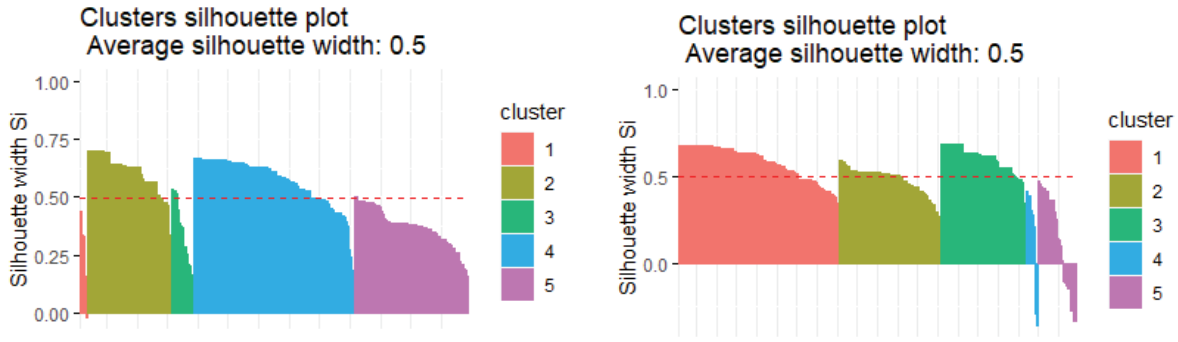


Figura 18: Representación del ancho medio de siluetas K-Means - PAM.

5.5.2 PAM

A continuación, y tras realizar un escalado de las variables, se procedió a generar la representación del algoritmo de clúster. Estableciendo de igual manera la distancia entre puntos de *manhattan* e incluyendo el número de k obtenidos, no obstante, se representó dicho algoritmo sin poder mostrar los *medoids* asociados a cada clúster.

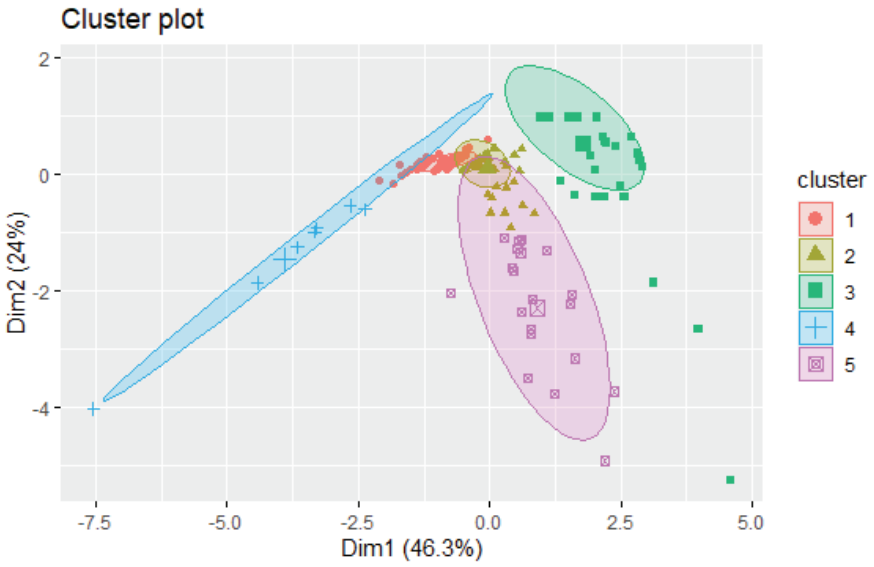


Figura 19: Representación del algoritmo de clústering PAM.

Los valores *Dim1* y *Dim2* en los ejes de la Figura 18 representan las nuevas variables generadas tras el proceso denominado *PCA* o análisis de componentes principales, donde se realiza una reducción de la dimensionalidad de, en nuestro caso 4 variables (*precio*, *valoraciones*, media de *valoraciones* y *descargas*) a únicamente 2 componentes principales. Los porcentajes explican la variabilidad que poseen los datos originalmente, por lo que *Dim1* explicaría el 46.3% de la variabilidad de los datos, mientras que *Dim2* el 24% [27].

Continuando con el análisis, en la Figura 19 se aprecia que el algoritmo PAM no consiguió recrear las elipsis perfectas para la demarcación de cada clúster, quizás debido a la aparición de múltiples valores atípicos como podemos observar en la parte baja del gráfico. Sin embargo, sí que consiguió diferenciar de manera significativa varios grupos del conjunto de variables, en concreto los asociados a los números 3,4,5.

Estos grupos a su vez son los que más abarcan en el gráfico, esto sucede debido a que la distancia entre los puntos que los conforman, y el *medoid* del clúster asociado es realmente alta, cosa que no sucede como podemos observar en los grupos 1 y 2, donde sus puntos están acumulados en zonas muy específicas del gráfico. Por lo que será en los clústers 3,4,5 donde obtendremos probablemente valores del conjunto de datos más dispares entre sí. Este razonamiento está basado en las distancias obtenidas del algoritmo PAM generado, las cuales son dispuestas en las Tablas siguientes.

	size	max_diss	av_diss	diameter	separation
[1,]	101	2.450148	0.4489316	3.709600	0.5600087
[2,]	65	2.292464	0.5839814	3.874054	0.5600087
[3,]	57	8.089427	1.1400959	8.890619	1.4153489
[4,]	7	10.017394	2.6697951	12.658899	1.3130804
[5,]	20	3.751735	1.7561100	6.313957	0.6712904

Tabla 11: Medidas de distancia de los clústers obtenidos.

	Precio	Valoraciones	M. Valoraciones	Descargaslog
[1,]	-0.3717305	-0.1533809	0.5101089	0.6867050
[2,]	-0.3717305	-0.1691472	0.5101089	-0.1144863
[3,]	-0.3717305	-0.1717213	-1.8106264	-0.9156777
[4,]	-0.3717305	3.4590309	0.5667123	2.2890878
[5,]	2.3149987	-0.1630338	0.5101089	-0.3556690

Tabla 12: Medoids de los clústers obtenidos.

Si nos fijamos en la Tabla 11, encontraremos las propiedades de cada uno de los clústers generados. Tales como el número de aplicaciones asociado a cada clúster (*size*), donde podemos ver que el grupo 1 es el de mayor *tamaño*, la máxima distancia entre puntos (*max_diss*), la distancia media entre puntos (*av_diss*) y la separación de estos (*separation*) donde podremos extraer de nuevo como los grupos 3,4 y 5 tienen valores muy dispares entre sí al tener los valores más altos asociados a estas columnas. La Tabla 12 por el contrario muestra el valor de los *medoids* asociados a cada uno de los clúster creados por PAM.

5.5.2.1 Resultados e interpretación

Asignada cada aplicación a un único clúster, se deberá realizar un pequeño análisis e interpretación de los grupos generados por PAM, el cual nos dará una visión generalizada del entorno de las aplicaciones. Para ello se creó un vector que conformaba el número de clúster asociado a cada fila (aplicación), el cual fue exportado al dataset original, conformando así una nueva columna numérica de la variable clúster.

De nuevo se utilizarían las funciones de RStudio que permitían obtener las medidas estadísticas descriptivas de cada grupo, así como la herramienta Open Refine para discernir entre las distintas variables que pertenecían a cada clúster.

Los cinco grupos formados nos muestran unos resultados muy interesantes a la hora de interpretar la situación de las aplicaciones de probabilidad y estadística de Google Play Store, una visión general de estas se ofrece en la siguiente Tabla.

	Precio (€)	Descargas	Valoraciones	Media de las valoraciones
Grupo 1				
Mínimo	0.00	5.000	7	2.1
Máximo	0.00	500.000	2.715	4.8
Media	0.00	31.337	228	4.0
Mediana	0.00	10.000	64	4.1
Grupo 2				
Mínimo	0.00	10	1	2.9
Máximo	2.19	1.000	77	5.0
Media	0.19	720	14	4.1
Mediana	0.00	1.000	9	4.1
Grupo 3				
Mínimo	0.00	1	0	1.1
Máximo	10.99	1.000	6	2.0
Media	0.80	250	0	1.5
Mediana	0.00	100	0	1.5
Grupo 4				
Mínimo	0.00	100.000	6.508	4.2
Máximo	0.00	10.000.000	39.927	4.8
Media	0.00	2.085.714	15.243	4.4
Mediana	0.00	1.000.000	11.284	4.5
Grupo 5				
Mínimo	2.19	10	5	3.0
Máximo	8.99	100.000	1.444	5.0
Media	4.33	5.582	136	4.3
Mediana	4.14	500	33	4.2

Tabla 13: Estadísticos descriptivos de los grupos generados (PAM).

- Grupo número 1: Aplicaciones gratuitas de carácter general

Es el clúster de mayor *tamaño*, lo que contendrá la mayor cantidad de datos distintos, pero no atípicos. Estas aplicaciones superan el umbral de *descargas* establecido en 5.000, por lo que podemos decir que poseen un alto grado de *descargas* respecto al conjunto.

En cuanto a la cantidad de *valoraciones* recibidas y su media, los usuarios ofrecen opiniones muy diversas, sin embargo, se observa un claro feedback, lo que las convierte en aplicaciones que probablemente tengan una vida útil dentro del entorno debido a esta interacción entre el desarrollador y el usuario.

Respecto a la media de *valoraciones*, estas aplicaciones poseen generalmente una valoración mayor o igual a 4 sobre 5, lo que las convierte en aplicaciones agradables de cara al público.

Por ello, su análisis promete niveles de *usabilidad* y *calidad de interfaz visual* de todo tipo, incluyendo a aquellas cuyo funcionamiento es impecable, hasta aquellas con un diseño ineficaz, no intuitivas o con categorías no funcionales. Este clúster por lo tanto cumple en líneas generales con las expectativas mínimas requeridas por el usuario, ofreciendo un nivel de satisfacción medio.

- Grupo número 2: Aplicaciones con escasa cantidad de evaluaciones

De manera similar al grupo número uno, este set conforma a aquellas aplicaciones que poseen una valoración positiva por parte de los usuarios y unos niveles de *usabilidad* y *calidad de interfaz visual* similares al grupo número 1, pero que, debido primordialmente a su carácter de pago, su *contenido* o la *categoría* de Google Play, se ven reducidos drásticamente el número de *descargas* o las *valoraciones* recibidas.

Esto es debido a que el algoritmo de Play Store sitúa en las casillas superiores a las *categorías* más descargadas, como son la educación o las herramientas, posicionando en la zona inferior a estas aplicaciones de pago y a aquellas cuya *categoría* no es tan popular, como son el Arcade, Empresa, Estilo de vida etc., dificultando que los usuarios puedan acceder a ellas.

- Grupo número 3: Aplicaciones no evaluadas

Aquí encontramos al grupo de aplicaciones que, debido al *precio* elevado de algunas de ellas y, con tan solo dos aplicaciones valoradas, poseen la menor media de *descargas* con un 1.5 sobre 5 y una cantidad de *valoraciones* ínfima con tan solo dos aplicaciones valoradas.

El desagrado de los usuarios a la hora de no poder evaluar estas aplicaciones se ve reflejado en la cantidad de *descargas* de estas, ya que se intuye que aquellas aplicaciones que no ofrecen la posibilidad de interacción de ningún tipo hacen sentir al usuario inseguro respecto a la fiabilidad del *contenido* y funcionamiento de la aplicación. Esto sumado a su condición de aplicaciones no gratuitas, agrava esta situación.

Como último punto a recalcar y por las razones explicadas anteriormente, se trata del único grupo que no posee ninguna aplicación que converge o cuya elipsis es colindante con alguno de los clústers restantes, siendo así el grupo más aislado del conjunto de datos y con la mayor cantidad de valores atípicos.

- Grupo número 4: Aplicaciones gratuitas con evaluación positiva

Las mejores aplicaciones en términos de evaluación, funcionamiento o diseño se encuentran en este grupo. Con unos niveles de *usabilidad* y *calidad de interfaz visual* únicamente altos, encontramos en estas aplicaciones la media más alta del dataset (4.4) y la mayor cantidad de evaluaciones registradas por el público, con una media de 15.243 por aplicación.

Esta medida se ve aun así afectada por el valor atípico de 39.927 *valoraciones* y las 10.000.000 de *descargas* de la aplicación ya mencionada "GeoGebra Calculadora Gráfica", estas cantidades son las responsables de la forma que toma la elipsis en dicho grupo, no obstante, el grupo número 4 ofrece el mayor contento y feedback por parte del público.

Como se observó en las correlaciones de los modelos de regresión obtenidos, el *precio* y la cantidad de *descargas* están relacionados de manera negativa, este grupo es por lo tanto el claro ejemplo que resume a grandes rasgos el pensamiento de los usuarios frente a las aplicaciones de probabilidad y estadística. Ya que ofrece un *contenido* general educativo muy recomendable en aplicaciones cuya utilización no requiere de un pago previo, de ahí el gran éxito de este conjunto de aplicaciones.

- Grupo número 5: Aplicaciones de pago con evaluación positiva

El último grupo generado por PAM está asociado a aquellas aplicaciones que ofrecen un contenido adicional o de mejor calidad respecto a la mayoría del conjunto de datos a costa de un pago previo por su descarga.

El hecho de permitir la interacción del usuario-desarrollador a la hora de su evaluación, hace que su número de *descargas* (aunque sea bajo respecto a las gratuitas) aumente considerablemente respecto a los otros grupos de pago, ya que como pudimos comprobar las *valoraciones* sí poseen una relación positiva respecto a las *descargas*.

Su media de *precio* de descarga de 4.33€, resulta sorprendentemente en una sensación agradable por parte del usuario respecto a la relación calidad-*precio* de la aplicación. El público aparentemente sabe que está pagando una cantidad razonable por una aplicación cuyo contenido o funcionamiento va más allá de lo ofrecido por las aplicaciones gratuitas.

Además, las mediciones realizadas acerca de la *calidad de interfaz visual* y *usabilidad* corroboran este pensamiento, ya que obtenemos valores mayoritariamente altos en estas dos variables, siendo el grupo número 5, el segundo clúster con mejor media de *valoraciones*, con un 4.3 sobre 5.

5.6 Otras soluciones: DBSCAN

La comparación de distintos tipos de algoritmos de clúster y su posterior elección forma parte de cualquier estudio donde se pretende realizar un agrupamiento de un conjunto de datos.

De igual manera que lo hicimos anteriormente en el modelo de regresión, donde se comparaba que tanto era capaz de explicar cada modelo, deberemos buscar otro tipo de algoritmo de agrupamiento con el que poder comparar los resultados frente a PAM.

La elección previa por parte del analista del número de clústers (k) que generará nuestro algoritmo puede variar completamente la estructura del conjunto formado. Por esta razón, se propone como posible solución alternativa, la utilización del algoritmo DBSCAN, el cual no requiere de una identificación previa del número de clústers a generar, y está basado en la identificación de grupos en base a la distribución de densidad de sus puntos. Ofreciendo así representaciones quizás más realistas del agrupamiento realizado al no depender de estructuras con forma de elipsis

Su generación, sin embargo, requiere definir previamente dos tipos de variables, la variable épsilon *(ε)*, relacionada con el radio de cada región generada, y el mínimo número de puntos dentro de dicha región épsilon que se identificará como *minPts*.

5.6.1 Generación del algoritmo

Empleamos como variables a utilizar, las mismas con las que realizamos el algoritmo de PAM, es decir: el *precio* de las aplicaciones, su cantidad de *valoraciones*, la *media de las valoraciones* y su número *descargas*, también transformada a escala logarítmica, todas ellas escaladas de manera previa.

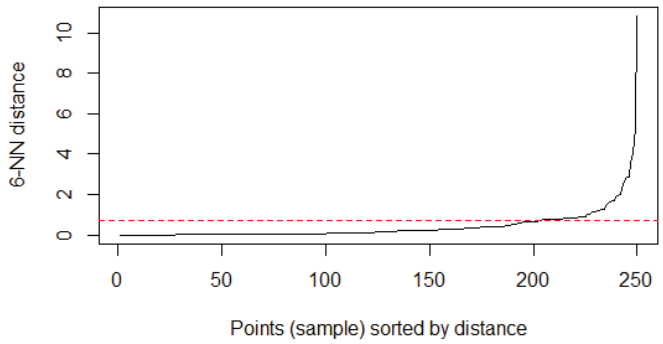


Figura 20: Representación de la curva de k-distancias.

Para conjuntos con pocos datos será mejor un valor pequeño de puntos mínimos, por ello tras varias pruebas, seleccionamos como *minPts* el valor de 6. Tras esto, se establece mediante la función *kNNdistplot()* de RStudio la curva de k-distancias para obtener una idea aproximada del valor de épsilon, el cual vendrá determinado por un valor próximo al punto de inflexión que toma la curva. En nuestro caso el valor óptimo equivale a $\varepsilon = 0.72$ [26]. Véase la Figura 20.

Tras conocer el valor de épsilon, mostramos gráficamente la resolución que nos devuelve DBSCAN para el conjunto de variables. Como indica la Figura 21, el número de clústers obtenido por dicho algoritmo indica un agrupamiento del conjunto de datos k = 3. Los puntos negros representan el ruido generado por los valores atípicos reconocidos por el algoritmo, los cuales se dispersan alrededor del gráfico.

Recalcamos que, tras varias pruebas realizadas (jugando con los valores de *ε y minPts)*, este agrupamiento obtenido por DBSCAN ha sido el óptimo y parejo a PAM en cuanto a cercanía a número de clústers concierne y máxima división de los grupos, ya que anteriormente se obtuvo una división del conjunto en k = 4, la cual fue rechazada por la influencia del ruido y escasos patrones de relación entre aplicaciones del mismo grupo.

Este valor de k se ve por lo tanto se ve menguado en comparación con las aproximaciones realizadas en PAM, por lo que a grandes rasgos parece ser que la medida de la distancia *medoids* entre observaciones funciona de mejor manera que la densidad de observaciones que ofrece DBSCAN.

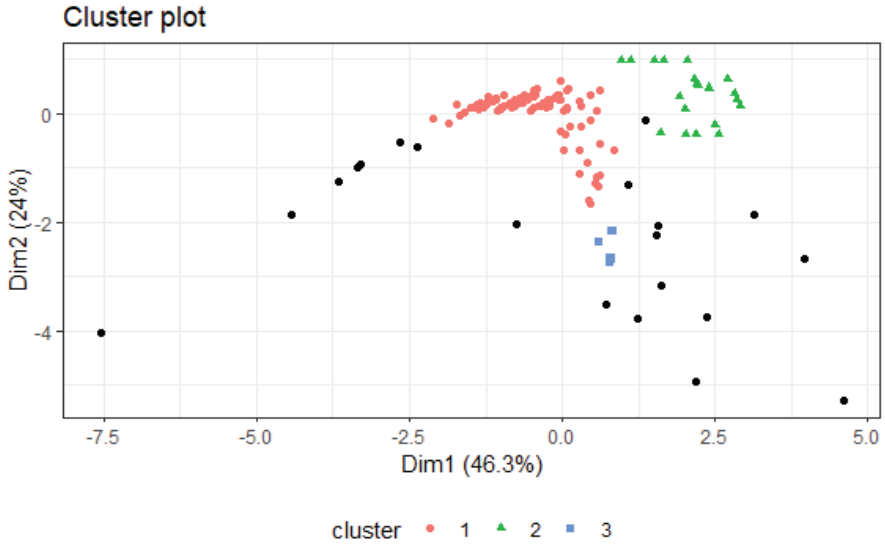


Figura 21: Representación del algoritmo de clústering DBSCAN.

Igual que en la representación del algoritmo PAM, *Dim1* y *Dim2* representan las nuevas variables generadas tras el proceso de reducción de la dimensionalidad. Siendo los porcentajes (46.3% y 24%) asociados a cada una de ellas el grado de variabilidad del conjunto de datos originario medidos por estas nuevas variables [27].

El patrón generado por los k = 3 clústers de DBSCAN corresponde con la cantidad de *valoraciones* recibidas en las aplicaciones. Estableciendo una clara diferenciación de aplicaciones que sí han sido valoradas respecto a las que no. Véase la Tabla 14. La cual ofrece las estadísticas descriptivas asociadas a las aplicaciones que conforman cada clúster.

	Precio (€)	Descargas	Valoraciones	Media de las valoraciones
Grupo 1				
Mínimo	0.00	10	1	2.1
Máximo	3.29	500.000	2.715	5.0
Media	0.00	18.585	142	4.1
Mediana	0.17	5.000	31	4.1
Grupo 2				
Mínimo	0.00	1	0	1.1
Máximo	2.39	1.000	5	1.1
Media	0.40	267	0	1.1
Mediana	0.00	100	0	1.1
Grupo 3				
Mínimo	4.19	500	18	4.1
Máximo	4.99	500	52	4.9
Media	4.59	500	32	4.6
Mediana	4.59	500	33	4.6

Tabla 14: Estadísticos descriptivos de los grupos generados (DBSCAN).

Se observa que una alta tasa de aplicaciones se ha quedado fuera del agrupamiento y han sido consideradas como ruido. Estas estaban relacionadas con aquellas que poseían los números más altos de *descargas* y cantidad de *valoraciones*, así como una media de *valoraciones* realmente positiva y un *precio* atípico respecto al resto, aplicaciones muy similares a las que conformaban los clústers 4 y 5 de PAM.

Concluimos que los grupos generados por DBSCAN no ofrecen resultados significativos para realizar un análisis detallado de cada uno de ellos. Sin embargo, podemos sacar algunas características sobre ellos:

- Grupo 1: Este grupo es el más grande formado, albergando en el a un total de 173 aplicaciones, por lo que contendrá la mayor variabilidad del *contenido*, funcionamiento y evaluación de los agrupamientos realizados. Con un alto nivel de interacción recibido por el público, una media de *valoraciones* superior a 4 sobre 5, y un buen nivel de *descargas* en comparación con los demás grupos, podemos decir que este clúster guarda mucha similitud con el clúster número 1 generado en PAM, siendo el grueso de las aplicaciones medidas del entorno.

- Grupo número 2: En este grupo aparecen las aplicaciones que no han recibido *valoraciones*, las cuales ascienden a un total de 53. Se obtiene como media de *valoraciones* el valor de 1.1 sobre 5, ya que solo existe una aplicación alojada en este grupo que ha ido evaluada por el usuario. Por lo que de igual manera que en PAM, y como podemos observar, las aplicaciones que no poseen dicha característica influyen de manera negativa en el número de *descargas*.

- Grupo número 3: Se trata del grupo más escaso, pero mejor valorado generado por DBSCAN en cuanto a media de *valoraciones* se refiere. Este agrupa a las aplicaciones cuyo *precio* no es condicionante a la hora de su descarga, lo cual unido a las *valoraciones* recibidas, influye de manera positiva en el agrado del público. Notamos mediante la Tabla 14 el patrón designado para la creación de este grupo, ya que las aplicaciones que lo conforman únicamente poseen un número de 500 *descargas*.

5.6.2 PAM vs DBSCAN

Los resultados obtenidos muestran una clara diferencia de qué algoritmo ha realizado una mayor diferenciación de grupos sobre las variables numéricas. Y es que, aunque DBSCAN no requiera de una definición previa de k, sus resultados no son óptimos para establecer una concreta definición de las aplicaciones del entorno de la probabilidad y estadística. Ya que se observa una gran influencia del ruido sobre las distribuciones de densidad tomadas a la hora de formar los grupos.

La reimplementación de ambos algoritmos de manera reiterada realizando los cambios oportunos recalcarían en una serie de resultados más concluyentes. Sin embargo, a grandes rasgos la elección de los grupos establecidos por PAM ofrece una mejor visión del estado de las aplicaciones de probabilidad y estadística de la plataforma Google Play Store, por ello en la documentación final del análisis del estudio se tomará como referencia dicho algoritmo.

6 Documentación de los resultados del análisis y conclusiones

Una vez llegado al final del estudio, regresamos a la pregunta que se formuló al comienzo de este trabajo, la cual pretendía definir la situación actual del entorno de las aplicaciones móviles de probabilidad y estadística de la plataforma Google Play. Para ello, intentemos sacar una conclusión en base a los resultados estadísticos obtenidos en este trabajo.

Las 250 aplicaciones medidas del conjunto de datos establece un patrón claro, aquellas aplicaciones alojadas en la *categoría* de educación tienen mayor probabilidad de ser descargadas frente a las demás. Esto es debido a que el usuario cuando introduce las palabras *probability* y *statistics,* lo hace con intención de obtener algún tipo de ayuda, herramienta o conocimiento en este campo, por eso se observa que las aplicaciones cuyo *contenido* son las calculadoras, el autoaprendizaje y la divulgación son latentes en este ámbito.

Pero ¿Qué ocurre con las aplicaciones informativas?. Se trata del segundo *contenido* de aplicaciones con más apariciones en el dataset, sin embargo, la mayoría de ellas se aleja bastante de la idea de aprendizaje de este terreno. Aplicaciones que ofrecen probabilidades deportivas o de juegos de azar abundan en este campo, y con ellas, *categorías* como casino, deportes o entretenimiento.

Este tipo de resultados no concuerdan con el tono referido al introducir sendas palabras, por lo que quizás haya una inconsistencia a la hora de determinar los resultados que ofrece el algoritmo de Play Store, el cual debería discernir de mejor manera las palabras introducidas respecto a los resultados ofrecidos.

La situación sería diferente si estas aplicaciones únicamente aparecieran en el final de los resultados, ya que sería el propio usuario el que podría distinguir de mejor manera la tónica del contenido que está buscando. Pero, como se mencionó a lo largo de este estudio, las *categorías* mostradas tienen relación con el *número descargas* y *categoría*, por lo que es común encontrarse con este tipo de aplicaciones a lo largo de nuestra búsqueda.

Una posible solución que podría implementarse para solventar este problema recae en una mejora de los filtros que proporciona dicha plataforma a la hora de introducir las palabras clave, o quizás añadir un filtro previo una vez introducidas dichas palabras que permita definir los criterios de aplicaciones que buscamos, ya que actualmente solo existen tres tipos de filtros.

En cuanto al apartado del funcionamiento de las aplicaciones, el conjunto medido posee una estructura de elementos consecuente con el tipo de utilización que requieren, esto se vio reflejado a la hora de medir cada una de las facetas de su estructura.

Sin embargo, la falta de explicaciones de su contenido o funcionamiento en la descripción del desarrollador tiene mucho que ver con las opiniones reflejadas en la caja de comentarios.

Esto conlleva a que probablemente haya muchos usuarios que, al desconocer el ámbito de la probabilidad y estadística, descarguen estas aplicaciones y no sepan utilizarlas o se sientan confusos, aunque están ofrezcan incluso un alto grado de funcionamiento. Esto ocurre frecuentemente en varias de las aplicaciones medidas, habiendo algunas cuyos niveles de *usabilidad* y *calidad de interfaz visual* son excelentes, pero que, debido a su complejidad, obtienen comentarios o *valoraciones* negativas, haciendo que estas aplicaciones sean cada vez menos visibles de cara al público.

Es por esto por lo que, si desea obtener mejores resultados, será responsabilidad del desarrollador realizar una mejora de las aplicaciones o una actualización más frecuente de estas en base al feedback recibido por el usuario.

Si, por el contrario, el desarrollador considera que su aplicación es óptima, una buena práctica sería asesorar a los usuarios, ya sea en la propia aplicación o en la descripción de esta. Indicar el nivel de complejidad de las aplicaciones en la descripción o establecer tutoriales acerca de su funcionamiento son soluciones que se proponen en este trabajo.

Como último apunte relacionado con el funcionamiento de las aplicaciones, en la estadística descriptiva de los datos, se obtuvo que aquellas cuyo contenido es divulgativo, poseen la menor media de *valoraciones*, unido a unas facetas de *calidad visual* y *usabilidad* media o bajas.

Esto quizás está relacionado con la presentación en forma de aplicación de libros, apuntes o contenido científico de probabilidad y estadística que quizás debería haberse valorado de otra manera.

La implementación de este contenido en aplicaciones de smartphones quizás pueda llegar a dificultar su lectura o interpretación, de ahí los valores negativos respecto a la *calidad visual*. Por esta razón si hubiéramos tenido en cuenta la *categoría* de libros de Google Play, seguramente habríamos encontrado infinidad de libros electrónicos con temática de probabilidad y estadística fascinantes, elevando sus puntuaciones respecto a los niveles de evaluación, funcionamiento y contenido.

Finalmente, llegamos al último fundamento con el que caracterizaremos al conjunto, la evaluación de las aplicaciones. Es aquí donde la implementación de los algoritmos empleados a lo largo del proyecto nos ha dado una visión más significativa de dicho entorno al tratar con variables exclusivamente numéricas.

La utilización de RStudio ha sido clave para formalizar este gran segmento del estudio, ya que, sin él, las medidas descriptivas, los algoritmos de regresión o los de agrupamientos habrían sido mucho más complicados de generar.

Al querer obtener una visión de estas en base a su número de *descargas*, se procedió a realizar el algoritmo de regresión predictivo, el cual, al no proporcionar las condiciones necesarias para su validación, acabó rechazándose. Es por ello por lo que no se ha utilizado ninguna de las ecuaciones obtenidas, ya que carecían de valor representativo.

Sin embargo, mediante su implementación, se comenzó a observar como la cantidad de *descargas* de nuestra muestra poseía cierta relación con algunas de las variables numéricas obtenidas. No fue hasta la medida las correlaciones de *Pearson* y *Kendall* del modelo de regresión lineal simple, donde verificamos dichas relaciones, positiva para el número de aplicaciones, y negativa respecto al *precio* de estas.

De igual manera, estas relaciones ya fueron intuidas a medida que realizábamos las Tablas de los datos descriptivos, como las *descargas* asociadas al *contenido*, donde destacaban las aplicaciones calculadoras que poseían un alto nivel de interacción del público. El número de aplicaciones de cada *categoría*, o la *media de valoraciones* de cada uno de los contenidos incitaban a que estas aplicaciones funcionaban mejor si poseían mayor cantidad de *valoraciones*.

Posteriormente, tras la creación del algoritmo de regresión logística, se consiguió crear una serie de fórmulas que permitían discernir del límite de las 5.000 *descargas* de una manera bastante acertada. Se decidió escoger dicha cantidad ya que se necesitaba excluir al grueso de aplicaciones cuyos valores estaban situados en cantidades inferiores.

Por esta razón, aunque el valor de 5.000 *descargas* no representaba a la media ni mediana del conjunto, era suficientemente significativo para comprender a los dos extremos de las aplicaciones, ya que el entorno de las aplicaciones de probabilidad y estadística no es uno de los más representativos que aloja la plataforma de Google Play, siendo su número de *descargas* inmensamente menor en comparación con las aplicaciones de entretenimiento, comunicación o aplicaciones de juegos.

Por lo que el hecho de haber aumentado o disminuido esta cantidad habría resultado en una visión no tan representativa, debemos recordar que el modelo de regresión logística explica en un 82% el comportamiento de las *descargas* frente a las *valoraciones* de la población, y con un 62.4% el *precio* de estas.

Por lo que finalmente habríamos conseguido generado dos fórmulas capaces de explicar en base a probabilidades las relaciones observadas durante el estudio.

Y de manera concluyente, nuestras predicciones acerca de la implicación de las *descargas* respecto al número de *valoraciones* y *precio* de las aplicaciones fueron resueltas mediante el algoritmo de PAM. La generación de los distintos grupos supuso comprender como realmente el usuario de este entorno prioriza ante todo las aplicaciones que sí han sido evaluadas y las de contenido gratuito, dejando a un lado variables que a comienzos del estudio se pensó que tendrían relación, como la *media de las valoraciones*.

Pero ¿cómo era posible que la media de *valoraciones* de las aplicaciones no estuviera tan relacionada?, si es el primer aspecto en el que cualquier usuario se fija a la hora de descargar contenido digital.

En primer lugar, el análisis de correlaciones ofrecía un nivel de correlación no significativo de esta variable respecto a las *descargas* o la cantidad de *valoraciones*.

Y, en segundo lugar, en los grupos 4 y 5 del algoritmo PAM observaríamos como una cantidad de *valoraciones* razonables sumado a un grupo de aplicaciones que no eran del todo económicas, repercutía en un alto número de *descargas*, así como un agrado del usuario que se veía reflejado en la caja de comentarios y, consecuentemente en la media de *valoraciones*. De igual manera, el análisis establecido en el grupo número 3, mostraba la gran repercusión que tienen estas dos variables para nuestro entorno.

Se concluye finalmente que, los objetivos y tareas fijadas al inicio del proyecto han sido exitosamente completados, ya que hemos obtenido una visión del entorno de las aplicaciones en base a tres aspectos fundamentales:

- El *contenido* de las aplicaciones, en cuyo caso pese a la gran diversidad de resultados obtenidos, el principal objetivo pasa por ayudar al usuario a interpretar los principales conceptos del área de probabilidad y estadística. Ofreciendo garantías de que dichas aplicaciones poseen un tipo de contenido no anticuado, sencillo de interpretar, y que en muchos casos reciben actualizaciones periódicas gratuitas o permiten la adquisición de contenido adicional gracias a sus versiones de pago.

- El funcionamiento basado en su estructura, *usabilidad* y diseño no muestra signos de aplicaciones de un entorno obsoleto o engañoso, ya que la mayoría de ellas son completamente funcionales e intuitivas para su uso, cumpliendo las necesidades básicas de un usuario estándar que desee adentrarse en este campo.

- Gracias a los análisis estadísticos realizados, su evaluación nos asegura que dicho entorno es totalmente válido a la hora de obtener cualquier recurso necesitado por el usuario. Una media global de 4.1 sobre 5, un feedback mayoritariamente positivo por parte de los usuarios, y un valor mínimo de *descargas* que supera los 17 millones son solo algunos de los aspectos que lo definen como un entorno seguro y agradable para el público.

6.1 Líneas futuras

Este estudio ha supuesto un reto absoluto a la hora de aprender e interpretar conceptos totalmente novedosos como los algoritmos de clúster, o la regresión logística, así como el entorno de programación RStudio que nunca había sido utilizado con anterioridad a lo largo del grado.

Pero este tipo de análisis realizado es tan solo un ápice de todas las posibilidades que nos ofrece el mundo de los algoritmos de agrupamiento y regresión. Sin embargo, debido a la complejidad y falta de tiempo, se propone una serie de aspectos que no han podido ser desarrollados en este trabajo, pero que pueden servir de gran utilidad para continuar con las líneas futuras del mismo.

Numerosos estudios de las aplicaciones de cualquier plataforma de distribución hacen hincapié en el análisis de comentarios y opiniones expresadas por los usuarios en la caja de comentarios. Estas características pueden ser analizadas para extraer información acerca de su experiencia tras la descarga de dichas aplicaciones. Por lo que la implantación de análisis de sentimientos, así como la utilización de técnicas de minado de texto podrían utilizarse en un futuro para esclarecer el apartado de evaluación de nuestro estudio.

Así mismo, respecto a la evaluación de las aplicaciones, un ensamble de modelos con el objetivo de predecir de mejor manera la respuesta, o la aplicación de estos con su variante de regresión múltiple, habrían mejorado las predicciones obtenidas en cuanto a la relación de *descargas* y número de *valoraciones* o *precio*, permitiéndonos realizar suposiciones más explicativas sobre el rendimiento de dicha variable.

Otro ejemplo de ello pasa por haber aplicado el algoritmo de regresión lineal o logística a los k-grupos obtenidos por PAM. Quizás así al disponer de conjuntos más uniformes entre sí, habríamos obtenido probablemente una distribución normal para cada uno de ellos, revirtiendo así la situación con la que nos encontramos cuando se realizó su validación durante el desarrollo.

El empleo del algoritmo de agrupamiento jerárquico o la generación de reglas lógicas mediante un árbol de decisión, nos habría permitido no tan solo operar con las variables numéricas del conjunto, si no establecer series jerárquicas de grupos de aplicaciones o condiciones que se dan en el conjunto de datos en base a las variables de funcionamiento, contenido y evaluación.

Al tener en cuenta la mayoría de variables que conforman nuestro dataset, esta implementación habría dado al lector una visión mucho más amplia de los grupos establecidos y, por ende, del entorno medido.

Frente a la agrupación ofrecida por DSCAN, la cual no se acogió a los resultados esperados, se podría utilizar en su lugar el algoritmo de agrupamiento OPTICS. Este soluciona varios aspectos de DBSCAN relacionados con la detección de clústers significativos en conjuntos de datos con densidades variables. Por lo que su aplicación quizás mostraría una mejora respecto al segundo agrupamiento realizado en el estudio, donde apenas surgieron grupos significativos.

El trabajo no termina aquí, nuestras 250 aplicaciones medidas son un porcentaje de la cantidad que alberga la plataforma Google Play, por lo que este estudio está definido sobre una muestra de la población total. El incluir periódicamente en el dataset nuevas aplicaciones o modificar las variables ya definidas, reforzará cualquier tipo de estudio que planee seguir los pasos de este, ofreciendo resultados mucho más acordes a la realidad.

7 Código empleado

7.1 Generación de las medidas descriptivas

```
#Importacion de librerias y dataset
library(readxl)
library(dplyr)
Dataset <- read_excel("C:/Users/../Desktop/Dataset.xlsx")

#Estadisticas y porcentajes basicos de las variables
#Medimos cantidad de valores NA, apariciones, porcentajes y medidas
descriptivas
basicStatistics <- function(arg1) {
sum(is.na(arg1))
sum(arg1[!is.na(arg1)])
summary(arg1[!is.na(arg1)])
}
basicPercentages <- function(arg1){
tableArg1 <- table(arg1)
percentages <- prop.table(tableArg1)*100
}

#Extraccion de datos de una columna mediante condiciones
dataExtract <- function(arg1,arg2,arg3){
vector <- Dataset[Dataset$arg1=="arg2","arg3"]
sum(vector[!is.na(vector)])
summary(vector)
}
```

7.2 Generación de los gráficos

```
#Importacion de librerias y dataset
library(ggplot2)
library(modeest)
library(ggpubr)
Dataset <- read_excel("C:/Users/../Desktop/Dataset.xlsx")

#Diagrama de barras del contenido de las aplicaciones
ContenidoApps <- Dataset$Contenido
CountsContenido <- table(ContenidoApps)
```

```
df <- data.frame(CountsContenido)
BarplotContenido <- ggplot(data=df,aes(x=ContenidoApps, y=Freq))+
geom_bar(stat="identity",color="grey",fill="steelblue", width = 0.8,
alpha=0.87)+
geom_text(aes(label=Freq), vjust=1.6, color="white", size=3.7)+
xlab("Contenido")+
ylab("Número de aplicaciones")+
theme(
   axis.title.x = element_text(vjust = -2),
   axis.title.y = element_text(vjust = 3)
)

#Diagrama de cajas del contenido de las valoraciones
gv <- Dataset$`Media de las valoraciones`
boxPlotMValoraciones <- ggplot(data = Dataset, aes(x = factor(ContenidoApps),
y = gv))+
geom_boxplot(outlier.colour="red",fill='steelblue', color="black",
alpha=0.87, outlier.size=2)+
scale_y_continuous(limits=c(1,5))+
xlab("Contenido")+
ylab("Media de las valoraciones")+
theme(
   axis.title.x = element_text(vjust = -1),
   axis.title.y = element_text(vjust = 3)
)

#Grafico de datos agrupados Calidad/usabilidad
calidad <- Dataset$`Calidad de la interfaz visual`
calidad <- table(calidad)
usabilidad <- Dataset$Usabilidad
usabilidad <- table(usabilidad)
data3 <- data.frame(Leyenda = c("Usabilidad", "Calidad de la interfaz
visual"),x1=rep(c("Alta","Media","Baja"), each=2),
y1=c(137,103,78,110,35,37))

usabIntPlot <- ggplot(data=data3, aes(x=x1, y=y1, fill=Leyenda)) +
geom_bar(stat="identity", color="grey", position=position_dodge(), width =
0.8, alpha=0.87)+
geom_text(aes(label=y1), vjust=1.6, color="white", size=3.7, position =
position_dodge(0.8))+
xlab("Niveles") +
ylab("Número de aplicaciones")+
```

```
scale_fill_brewer(palette="Paired")+
scale_x_discrete(limits=c("Alta", "Media", "Baja"))+
theme(
   axis.title.x = element_text(vjust = -2),
   axis.title.y = element_text(vjust = 3)
)

#Diagrama de barras del numero de descargas de las aplicaciones
DescargasApps <- Dataset$Descargas
CountsDescargas <- table(DescargasApps)
df4 <- data.frame(CountsDescargas)
BarplotDescargas <- ggplot(data=df4,aes(x=DescargasApps, y=Freq))+
geom_bar(stat="identity",color="grey",fill="steelblue", width = 0.8,
alpha=0.87)+
geom_text(aes(label=Freq), vjust=-0.4, color="black", size=3.7)+
xlab("Descargas")+
ylab("Número de aplicaciones")+
scale_y_continuous(limits=c(0,60))+
scale_x_discrete(labels=c("1+","5+","10+","50+","100+","500+","1.000+","5.000
+","10.000+","50.000+","100.000+","500.000+","1.000.000+","10.000.000+"))+
theme(
   axis.text.x = element_text(angle = 60,hjust=1,vjust=0.95),
   axis.title.x = element_text(vjust = 6),
   axis.title.y = element_text(vjust = 4)
)

#Diagrama de barras del numero de descargas por contenido
y <- sort(unique(Dataset$Contenido))
descargasLabel = c("+1.7M","+11.2M","+2.1M","+300K","+1.8M","+725K")
data4 <- data.frame(contenido = y, descargas =
c(1722025,11205864,2161750,306360,1816656,725080))
bardescargas <- ggplot(data=data4, aes(x=contenido, y=descargas))+
geom_bar(stat="identity",color="grey",fill="steelblue", width = 0.8,
alpha=0.87)+
geom_text(aes(label=descargasLabel), vjust=-0.4, color="black", size=3.7)+
xlab("Contenido") +
ylab("Número de descargas")+
scale_y_continuous(labels=c("0","+3M","+6M","+9M","+12M"))+
theme(
   axis.title.x = element_text(vjust = -2),
    axis.title.y = element_text(vjust = 3)
```

```
)

#Histograma del tamaño de las aplicaciones
VectorTamano = Dataset$Tamaño
bw <- 2 * IQR(VectorTamano) / length(VectorTamano)^(1/3)
df10 <- data.frame(VectorTamano)
HistTamano<-ggplot(df10, aes(x=VectorTamano))+
geom_histogram(binwidth = bw, color="grey", fill="steelblue", alpha=0.87)+
geom_vline(aes(xintercept=mean(VectorTamano),color="Media"),linetype="dashed"
, size=1)+
geom_vline(aes(xintercept=median(VectorTamano),color="Mediana"),linetype="das
hed", size=1)+
geom_vline(aes(xintercept=quantile(VectorTamano, 0.25),color="Primer
cuartil"),linetype="dashed", size=1)+
geom_vline(aes(xintercept=quantile(VectorTamano, 0.75),color="Tercer
cuartil"),linetype="dashed", size=1)+
xlab("Tamaño de las aplicaciones (M)")+
ylab("Frecuencia")+
scale_color_manual(name = "Leyenda", values = c(Media = "red", "Mediana" =
"black", "Primer cuartil" = "purple", "Tercer cuartil" = "orange"))+
theme(
   axis.title.x = element_text(vjust = -1),
   axis.title.y = element_text(vjust = 3)
)

#Histograma de la media de valoraciones
VectorMedia = Dataset$`Media de las valoraciones`
VectorMedia <- VectorMedia[!is.na(VectorMedia)]
df3 <- data.frame(VectorMedia)
HistValo<-ggplot(df3, aes(x=VectorMedia))+
geom_histogram(bins = 40, color="grey", fill="steelblue", alpha=0.87)+
geom_vline(aes(xintercept=mean(VectorMedia),color="Media"),linetype="dashed",
size=1)+
geom_vline(aes(xintercept=quantile(VectorMedia, 0.25),color="Primer
cuartil"),linetype="dashed", size=1)+
geom_vline(aes(xintercept=quantile(VectorMedia, 0.75),color="Tercer
cuartil"),linetype="dashed", size=1)+
xlab("Media de las valoraciones")+
ylab("Frecuencia")+
scale_y_continuous(limits=c(0,40))+
scale_color_manual(name = "Leyenda", values = c("Media" = "black", "Primer
cuartil" = "red", "Tercer cuartil" = "orange"))+
```

```
theme(axis.title.x = element_text(vjust = -1),
axis.title.y = element_text(vjust = 3))
```

7.3 Generación del modelo de regresión lineal

```
#Importacion de librerias
library(effects)

#Creacion de vectores
VectorValoraciones2 <- Dataset$Valoraciones
DescargasApps2 <- Dataset$Descargas
DescargasAppslog <- log(DescargasApps2)

#Generacion de correlaciones
correlaciones <- function(arg1,arg2){
cor.test(arg1,arg2,method="pearson")
cor.test(arg1,arg2,method="spearman")
cor.test(arg1,arg2,method="kendall")
}

#Modelo de regresion lineal simple para descargas-precio
linealr <- lm(DescargasAppslog~VectorPrecio2)
plot(VectorPrecio2,DescargasAppslog, xlab = "Precio de las aplicaciones",
ylab = "Número de descargas (log)")
abline(linealr,col = "red")
ggplot(data = Dataset, mapping = aes(x = VectorPrecio2, y =
DescargasAppslog)) + geom_point(color = "firebrick", size = 2) +
labs(x = "Precio de las aplicaciones",y='Número de descargas (log)') +
geom_smooth(method = "lm", se = FALSE, color = "black") +
theme_bw() + theme(plot.title = element_text(hjust = 0.5))

#Modelo de regresion lineal simple para descargas-valoraciones
linealr2 <- lm(DescargasAppslog~VectorValoraciones2)
plot(VectorValoraciones2,DescargasAppslog, xlab = "Número de valoraciones",
ylab = "Número de descargas (log)")
abline(linealr2, col = "blue")
ggplot(data = Dataset, mapping = aes(x = VectorValoraciones2, y =
DescargasAppslog)) +geom_point(color = "firebrick", size = 2) +
labs(x = "Número de valoraciones",y='Número de descargas (log)') +
geom_smooth(method = "lm", se = FALSE, color = "black") + theme_bw() +
theme(plot.title = element_text(hjust = 0.5))
```

```
#Extraccion de datos de los modelos
summary(linealr)
summary(linealr2)

#Analisis de los residuos del modelo descargas-precio
linealrResid <- linealr$residuals
linealrPred <- linealr$fitted.values

#Grafico linealidad de residuos
ggplot(data = Dataset, aes(x = linealrPred, y = linealrResid)) +
geom_point(aes(color = linealrResid)) +
scale_color_gradient2(low = "blue3", mid = "grey", high = "red") +
geom_hline(yintercept = 0) + geom_segment(aes(xend = linealrPred, yend = 0),
alpha = 0.2) +
labs( x = "Predicción",y = "Residuos") +
theme_bw() +theme(plot.title = element_text(hjust = 0.5), legend.position =
"none")+
theme(
   axis.title.x = element_text(vjust = -1),
   axis.title.y = element_text(vjust = 3)
)

#Representacion y calculo de la normalidad residuos
qqnorm(linealrResid)
qqline(linealrResid)
shapiro.test(linealrResid)

#Homocedasticidad de residuos
library(lmtest)
bptest(linealr)
par(mfrow=c(1,2))
plot(linealr)


#Analisis de los residuos del modelo descargas-valoraciones
linealr2Resid <- linealr2$residuals
linealr2Pred <- linealr2$fitted.values
```

```
#Grafico linealidad de residuos
ggplot(data = Dataset, aes(x = linealr2Pred, y = linealr2Resid)) +
geom_point(aes(color = linealr2Resid)) +
scale_color_gradient2(low = "blue3", mid = "grey", high = "red") +
geom_hline(yintercept = 0) +
geom_segment(aes(xend = linealr2Pred, yend = 0), alpha = 0.2) +
labs( x = "Predicción",y = "Residuos") +
theme_bw() +
theme(plot.title = element_text(hjust = 0.5), legend.position = "none")+
theme(
   axis.title.x = element_text(vjust = -1),
   axis.title.y = element_text(vjust = 3)
)

#Representacion y calculo de la normalidad residuos
qqnorm(linealr2Resid)
qqline(linealr2Resid)
shapiro.test(linealr2Resid)

#Homocedasticidad de residuos
library(lmtest)
bptest(linealr2)
par(mfrow=c(1,2))
plot(linealr2)
```

7.4 Generación del modelo de regresión logístico

```
#Importacion de librerias
library(vcd)
library(MASS)

#Importamos el dataset modificado
Dataset2 <- read_excel("C:/Users/.../Desktop/Dataset2.xlsx")

#Extraemos los vectores
Descargas <- as.factor(Dataset2$Descargas)
Valoraciones2 <- Dataset2$Valoraciones
Precio <- Dataset2$Precio
MValoraciones <- Dataset2$`Media de las valoraciones`
```

```r
#Conjuntos de data frames
datos <- data.frame(Descargas, Valoraciones2)
datos2 <- data.frame(Descargas, Precio)

#Generacion del modelo
modelo <- glm(Descargas ~ Valoraciones2, data = datos, family = "binomial")
summary(modelo)

#Representacion del modelo
datos$Descargas <- as.character(datos$Descargas)
datos$Descargas <- as.numeric(datos$Descargas)
plot(Descargas ~ Valoraciones2, datos, col = "darkblue",
ylab = "P(Descargas=1|Valoraciones)",
xlab = "Valoraciones", pch = "I")
curve(predict(modelo, data.frame(Valoraciones2 = x), type = "response"),
col = "firebrick", lwd = 2.5, add = TRUE)

#Evaluacion del modelo
anova(modelo, test = "Chisq")
predicciones <- ifelse(test = modelo$fitted.values > 0.5, yes = 1, no = 0)
matriz_confusion <- table(modelo$model$Descargas, predicciones,
dnn = c("observaciones", "predicciones"))
matriz_confusion

#Generacion del segundo modelo
modelo <- glm(Descargas ~ Precio, data = datos2, family = "binomial")
summary(modelo)
confint(object = modelo,level = 0.95)

#Representacion del segundo modelo
datos2$Descargas <- as.character(datos2$Descargas)
datos2$Descargas <- as.numeric(datos2$Descargas)
plot(Descargas ~ Precio, datos2, col = "darkblue",
ylab = "P(Descargas=1|Precio)",
xlab = "Precio", pch = "I")
curve(predict(modelo, data.frame(Precio = x), type = "response"),
col = "firebrick", lwd = 2.5, add = TRUE)
```

```
#Evaluacion del segundo modelo
anova(modelo, test = "Chisq")

predicciones <- ifelse(test = modelo$fitted.values > 0.5, yes = 1, no = 0)
matriz_confusion <- table(modelo$model$Descargas, predicciones,
dnn = c("Observaciones", "Predicciones"))
matriz_confusion
```

7.5 Generación del algoritmo de agrupamiento PAM

```
#Importacion de librerias
library(cluster)
library(factoextra)
library(clustertend)
library(cclust)

#Importamos el dataset de variables únicamente numericas
DatasetNumerico <- read_excel("C:/Users/.../Desktop/DatasetNumerico.xlsx")

#Creacion del subconjunto de datos
grupo <- select(DatasetNumerico, Precio, Valoraciones, 'Media de las
valoraciones', Descargaslog)

#Extraemos un subgrupo de variables para generar el algoritmo
str(grupo)
datos <- scale(grupo)

#Medida del numero de k grupos con distancia manhattan
fviz_nbclust(x = datos, FUNcluster = pam, method = "wss", k.max = 15,
diss = dist(datos, method = "manhattan"))
fviz_nbclust(x = datos, FUNcluster = kmeans, method = "wss", k.max = 15,
diss = dist(datos, method = "manhattan"))

#Comparativa de modelo K-means y PAM para k = 5
km <- eclust(datos,FUNcluster="kmeans", k=5 ,hc_metric = "manhattan")
km.sil<-silhouette(km$cluster, dist(datos))
fviz_silhouette(km.sil)
```

```
pm <- eclust(datos,FUNcluster="pam", k=5,hc_metric = "manhattan")
pm.sil<-silhouette(pm$cluster, dist(datos))
fviz_silhouette(pm.sil)

#Generacion de PAM para k = 5 grupos
set.seed(123)
pam_clusters <- pam(x = datos, k = 5, metric = "manhattan")

#Asociamos cada aplicacion del dataset a su cluster
grupo$cluster = pam_clusters$cluster
head(grupo)

#Representacion del algoritmo
fviz_cluster(object = pam_clusters, data = datos, geom = "point",ellipse.type
= "t", repel = TRUE, show.clust.cent = TRUE)
```

7.6 Generación del algoritmo de agrupamiento DBSCAN

```
#Importacion de librerias
library(fpc)
library(DBSCAN)
library(factoextra)
library(xlsx)
library(writexl)

#Seleccionamos el numero de epsilon
DBSCAN::kNNdistplot(datos, k = 6)

#Representacion de la curva de k-distancias
abline(h = 0.72, lty = 2, col="red")

#Generacion de DBSCAN con valores de epsilon = 0.72 y minPts = 6
set.seed(321)
DBSCAN_cluster <- fpc::DBSCAN(data = datos, eps = 0.72, MinPts = 6)
fviz_cluster(object = DBSCAN_cluster, data = datos, stand = FALSE,
geom = "point", ellipse = FALSE, show.clust.cent = FALSE,
pallete = "jco") +
theme_bw() + theme(legend.position = "bottom")
```

```
#Exportamos el nuevo dataset creado
Dataset3<-cbind(Dataset2,DBSCAN_cluster$cluster)
write_xlsx(x = Dataset3, path = "Dataset3.xlsx", col_names = TRUE)

#Importamos el dataset con el vector cluster asociado a cada aplicacion
Dataset3 <- read_excel("C:/Users/.../Desktop/Dataset3.xlsx")
```

7.7 Comparación de ambos algoritmos de clúster

```
#Comparamos los resultados de ambos algoritmos
comparacionClusters <- function(arg1, arg2) {
z <- Dataset7[Dataset3$'Cluster DBSCAN==arg1, arg2]
summary(z)
z2 <- Dataset7[Dataset3$Cluster PAM=="arg1","arg2]
summary(z2)
}
```

8 Bibliografía

[1] S. Mokarizadeh, M. Tafiqur, and M. Matskin, “Mining and Analysis of Apps in Google Play”, M.S thesis., ICT School., KTH Univ., Estocolmo, 2013.

[2] H. Edwards, “A review of probability and statistics apps for mobile devices”, M.S thesis., Massey Univ., Nueva Zelanda, 2014.

[3] Wikipedia. (2020, Noviembre). *List of mobile app distribution platforms*. Disponible: https://en.wikipedia.org/wiki/List_of_mobile_app_distribution_platforms

[4] Wikipedia. (2020, Noviembre). *Google Play*. Disponible: https://en.wikipedia.org/wiki/Google_Play

[5] J. Clement, “Number of apps available in leading app stores as of 2nd quarter 2020”, Statista., Hamburgo, Tech. Rep., Octubre, 2020.

[6] F. Hujainah, H. Dahlan, B. Al-haimi, “Usability Guidelines of Mobile Learning Application”, M.S. Tesis, Technology Malaysia Univ., Skudai, Johor Bahru, Malasia, 2013.

[7] J. Cuello and J. Vittone, “Diseño visual” en *Diseñando apps para móviles*, 2013, ch. 8, pp. 110-150.

[8] L. Gupta (2020). *Google Play Store Apps* (*Versión 6*). Disponible: https://www.kaggle.com/lava18/google-play-store-apps?select=googleplaystore.csv

[9] G. Federico Martínez, F. Mir, L. García Romano, “Caracterización de aplicaciones móviles para la enseñanza y el aprendizaje de la anatomía humana”, in *X Congr. Internacional sobre investigación en didáctica de las ciencias.*, Sevilla., 2017.

[10] C. Espino, “Análisis predictivo: técnicas y modelos utilizados y aplicaciones del mismo - herramientas Open Source que permiten su uso”, M.S tesis., Dept Buss Inteligence., Univ UOC., 2017.

[11] N. Buus, R. Madsen, y R. Vatrapu, “Predicting iPhone Sales from iPhone Tweets”, Ph.D. dissertation., Dept of ITM. and Mobile. Tech. Lab., Business S., Copenhague, ITC Univ., Oslo, 2014.

[12] V. Kumar y M.L., “Predictive Analytics: A Review of Trends and Techniques,” *International Journal of C.A*, vol. 182, no. 1, pp. 32-36, Jul. 2018.

[13] S. Pértega y S. Pita Fernández, “Técnicas de regresión: Regresión Lineal Múltiple”, U. de Epidemiología Clínica y Bioestadística, C. H. Universitario de A Coruña., A Coruña, 2001.

[14] M. Carmen Carollo, “Regresión lineal simple”, Dept. de Estadística e Inv. Operativa., USC., Santiago de Compostela., 2011-2012.

[15] UTN, F. Regional de Buenos Aires, "Fórmulas" en *Probabilidad y Estadística*, Cs. Básicas. U.D.B. Matemática, Ed. CEIT-FRBA.

[16] S. de la Fuente, “Análisis Conglomerados”, M.S. thesis., Fac. CC., Univ Autónoma., Madrid, 2011.

[17] Wikipedia. (2020, Noviembre). *Cluster analysis, Algorithms*. Disponible: https://en.wikipedia.org/wiki/Cluster_analysis

[18] P. Berkhin, “Survey of Clustering Data Mining Techniques”, Accrue Software., Nueva York., NY, 2002.

[19] R-project. (2020), *What is R?, Introduction,* «r-project.org,» 2020. Disponible: https://www.r-project.org/about.html

[20] R-project. (2020), *What is R?, The R enviroment,* «r-project.org,» 2020. Disponible: https://www.r-project.org/about.html

[21] Wikipedia. (2020, Diciembre). *Freedman–Diaconis rule*, *General equation*. Disponible: https://en.wikipedia.org/wiki/Freedman%E2%80%93Diaconis_rule

[22] J. Amat Rodrigo, (2016, Junio). *Ejemplo práctico de regresión lineal simple, múltiple, polinomial e interacción entre predictores*. Disponible: https://www.cienciadedatos.net/documentos/24_correlacion_y_regresion_lineal#Condiciones_para_la_regresi%C3%B3n_lineal

[23] R. I. Kabacoff, “Data analysis and graphics with R” en R In Action, Shelter Island, NY, Manning Publications, 2011, ch. 8, sec 8.2.

[24] J. Amat Rodrigo, (2016, Agosto). *Regresión logística simple y múltiple.* Disponible: https://rpubs.com/Joaquin_AR/229736

[25] K. Kryńska, (2018, Diciembre). *Using K-means and PAM clustering for Customer Segmentation*. Disponible: https://rpubs.com/kkrynska/USL_k-means

[26] J. Amat Rodrigo, (2017, Septiembre). *Clustering y heatmaps: aprendizaje no supervisado*. Disponible: https://rpubs.com/Joaquin_AR/310338

[27] C. Gil Martínez, (2018, Junio). *Análisis de componentes principales (PCA)*. Disponible: https://rpubs.com/Cristina_Gil/PCA

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Wed Jan 27 23:04:57 CET 2021
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)