**TRABAJO DEL CURSO**

Inteligencia Empresarial

**AMAZON SALES**



Juan Miguel Moreno Valero

**ÍNDICE**

[1. Introducción 3](#_Toc135473733)

[2. Objetivo del proyecto 3](#_Toc135473734)

[3. Herramientas empleadas 3](#_Toc135473735)

[4. Análisis preliminar y representaciones visuales 4](#_Toc135473736)

[5. Sentiment Analysis 8](#_Toc135473737)

[1. Adaptación del dataset 8](#_Toc135473738)

[2. Comprensión y acercamiento a la realidad 9](#_Toc135473739)

[3. Construcción del modelo 9](#_Toc135473740)

[6. Conclusión 12](#_Toc135473741)

# **Introducción**

La motivación de este proyecto arranca desde la idea de poder llevar a cabo un análisis de los datos de una de las empresas más internacionales que existen hoy en día: Amazon. A partir de ahí, el proyecto abre la posibilidad de escoger un determinado dataset de más de mil productos y llevar a cabo un análisis del mismo, con el fin de obtener modelos que nos permitan descifrar la información que se esconde entre la nube de datos que comprende nuestro problema, una información valiosa que será aplicada a la mejora del rendimiento de alguno de sus aspectos.

# **Objetivo del proyecto**

El objetivo que ha perseguido este proyecto no ha sido otro que el de conseguir, a partir de los datos que disponíamos, extraer información en alguno de los campos que considerásemos que se podrían potenciar. Por tanto, el proyecto se basa en llevar a cabo la técnica de ‘sentiment analysis’, o análisis de sentimientos, de las reseñas que los diferentes usuarios han redactado. Para ello, tomaremos todas las reseñas publicadas de todos los usuarios a todos los productos, y estableciendo la calidad de valoración de cada producto, que viene dado por el rating que los propios usuarios han proporcionado a cada artículo, poder llevar a cabo una clasificación de la satisfacción que muestra el cliente con cada una de esas reseñas. Gracias a esto, podremos acercarnos a una comprensión de la opinión del cliente, lo cual nos puede aportar una ventaja estratégica para establecer una monitorización de la marca de algunos de los productos o para predecir tendencias que se estén produciendo en el mercado entre los individuos, que los sitúan más propensos a comprar ciertos tipos de artículos o a evitar el consumo de otros.

# **Herramientas empleadas**

Para alcanzar el objetivo que habíamos establecido, hemos hecho uso de las dos herramientas impartidas a lo largo del curso: la aplicación analista Power BI Desktop y el uso del lenguaje Python en su conjunto, concretamente de librerías como Pandas, NumPy, Matplotlib y Seaborn, aparte de la librería encargada de la creación de los modelos de clasificación y regresión, scikit-learn.

# **Análisis preliminar y representaciones visuales**

Antes de cualquier tipo de operación realizada sobre el conjunto de datos del cual disponíamos, es necesario llevar a cabo un acercamiento previo a nuestro dataset.

Inicialmente, y basando el esfuerzo en la herramienta Power BI Desktop, realizamos una limpieza y preprocesado de los datos para adaptar el dataset a nuestras necesidades, como el cambio de moneda y la creación y cambio de nomenclatura de algunas columnas para dotar de sentido al dataset. Además, implementamos nuevas tablas para poder hacer relaciones entre las columnas determinadas, cosa que nos facilita la posibilidad de construir asociaciones para una representación gráfica más adecuada.

Una vez hecho esto, llevamos a cabo una serie de representaciones gráficas que consideramos interesantes y útiles para la comprensión de los datos:

Texto

Descripción generada automáticamente con confianza media

Visualizamos los productos mejor valorados de nuestro dataset. Comprobamos que los dispositivos o complementos electrónicos son los que reciben las puntuaciones más altas.

Gráfico, Gráfico circular

Descripción generada automáticamente

Repetimos la operación para las categorías más valoradas, para intentar buscar una correlación entre productos y categorías.

Gráfico, Gráfico de proyección solar

Descripción generada automáticamente

Sin embargo, observamos que las principales categorías más valoradas son también las que más productos contienen y, por tanto, su ranting acumulado aumenta. Por tanto, no se puede sacar una relación precipitada entre productos con mayor rating y categorías más valoradas.

Interfaz de usuario gráfica

Descripción generada automáticamente

Llevamos a cabo un análisis de los 15 productos más caros y los 15 más baratos, para descubrir si existe alguna relación entre el precio que un usuario paga por un producto y la valoración que establece para este, ya sea por comprender una mejor calidad, o porque la percepción humana para votar ante un producto caro nos induce a pensar que es mejor de lo que puede realmente ser.

Efectivamente, todos los 15 productos más caros tienen una valoración de al menos 4 estrellas, mientras que los 15 más baratos comprenden valores algo más bajos. La conclusión de esta premisa, como ya he comentado anteriormente, puede ser interpretable.

Para buscar detalles diferentes en nuestro análisis exploratorio, usaremos las herramientas y librerías de Python para descubrir qué se esconde en nuestro dataset. Mediante métodos de Pandas, desglosamos nuestros datos como un conjunto compuesto por 1464 productos (uno de ellos es eliminado por un error en la columna ‘rating’) y un total de 16 columnas que son las siguientes: [‘product\_id’, ‘product\_name’, ‘category’, ‘discounted\_price’, ‘actual\_price’, ‘discounted\_percentage’, ‘rating’, ‘rating\_count’, ‘about\_product’, ‘user\_id’, ‘user\_name’, ‘review\_id’, ‘review\_title’, ‘review\_content’, ‘img\_link’, ‘product\_link’].

Además, tras una serie de transformaciones, llevamos a cabo representaciones realizadas gracias a la ayuda de las herramientas Matplotlib y Seaborn:

Gráfico, Histograma

Descripción generada automáticamente

Vemos la distribución de rating que se obtiene del total de productos, siendo 4’1 la valoración más otorgada por los usuarios a un producto, con un total de 244 veces aplicado.

Gráfico

Descripción generada automáticamente

Además, llevamos a cabo un proceso de búsqueda de los nombres de aquellos usuarios que más puntúan los productos, siendo en su mayoría usuarios anónimos bajo el nombre de ‘Amazon Customer’ o ‘Placeholder’.

# **Sentiment Analysis**

## **Adaptación del dataset**

Existen varias formas de realizar un sentiment analysis, estableciendo una crítica binaria (0 negativa, 1 positiva) o repartiéndolas en categorías (muy buena, buena, estándar, mala, muy mala), entre otras; y se puede llevar su modelado mediante técnicas clásicas o mediante técnicas más novedosas, como su aplicación mediante redes neuronales recurrentes. Por ser la primera vez que enfrentamos este tipo de desafío, se ha optado por utilizar una clasificación binaria mediante técnicas clásicas, pero abordar diferentes combinaciones de estos modelos para asegurarnos de encontrar el máximo rendimiento de nuestro modelo.

Por tanto, nuestro dataset estaba compuesto por productos, los cuales tienen un rating individual otorgado, que es la media de todas las puntuaciones dadas a ese artículo, y de una serie de reseñas, de número desigual por producto. En consecuencia, el inconveniente era transformar nuestro conjunto de datos en uno en el que pudiésemos relacionar las críticas de cada producto de forma independiente. Tras realizar una serie de transformaciones, conseguimos adaptar nuestro dataset en otro nuevo formado por tres columnas. La primera sería el nombre del producto, que se repetirá tantas veces como número de reseñas tenga dicho producto. El segundo campo comprenderá cada una de esas reseñas atribuidas a cada uno de esos ‘product\_name’, y la última columna vendría dada por el rating relacionado con cada producto, repetido tantas veces como repetido esté el producto. Con este dataset, conseguimos simular que cada producto tiene una reseña y un rating diferente, lo que utilizamos como clave para poder aplicar nuestro sentiment analysis a cada reseña de forma independiente.

Esto sería una imagen del dataset adaptado a nuestras necesidades.

Tabla

Descripción generada automáticamente

Además, la doble intención de este proceso es que pasamos de un dataset de 1464 productos, quizás insuficiente para obtener un rendimiento adecuado tras el entrenamiento, a un modelo que cuenta con 9157 filas, y por tanto reseñas, que enriquece enormemente nuestro volumen de datos.

## **Comprensión y acercamiento a la realidad**

Enfocamos la división de reseña positiva o reseña negativa basándonos en el rating medio obtenido por ese producto. La separación se establece cogiendo los valores del rating y marcando un valor divisorio que provocaría que, por debajo de ese umbral, las críticas fueran negativas (0), mientras que, si son superiores, pasarían a ser positivas (1), eliminándose los valores iguales al umbral, considerándose neutros.

En principio, se hizo la media para los valores de rating para determinar cuál sería el valor de umbral a escoger, siendo este el valor 3. Sin embargo, comprobamos que había un claro desbalance ya que eran muy pocos los productos (6 productos únicos sin repetición de los 1464 iniciales) que se catalogaban con menos de 3 estrellas, mientras que los restantes tomaban valores por encima de este límite.

Por tanto, se reestructuró el criterio y, adaptando la lógica al problema, se consideró que realmente un usuario tiene una percepción de que un producto es malo si su valor de rating es por debajo del 4, ya que un cliente dudará si hacerse con un artículo si su valoración es menor que dicha cifra. Tras establecer este nuevo umbral, el dataset queda más balanceado, aunque siguen prevaleciendo las reseñas positivas frente a las negativas ya que, en este tipo de contexto, una crítica mala afecta en gran medida a la predisposición del usuario para comprar el producto, por lo que las valoraciones asociadas a los diferentes productos cobran vital importancia.

## **Construcción del modelo**

Para la construcción del modelo, llevaremos a cabo dos estrategias: aplicación de modelos de clasificación y de regresión.

* 1. **Modelos de clasificación**

Emplearemos dos tipos de vectorizadores para transformar las secuencias de palabras en formato de tipo numérico, CountVectorizer y TfidfVectorizer, añadiendo el parámetro ‘stop\_words’, para eliminar palabras que no aportan información, y alternaremos su uso con los modelos de clasificación Random Forest y SVM. Los resultados que obteníamos del entrenamiento de dichas combinaciones se muestran a continuación:

* **CountVectorizer + Random Forest**

Tabla

Descripción generada automáticamente con confianza media

* **CountVectorizer + SVM**

Calendario

Descripción generada automáticamente con confianza baja

* **TfidfVectorizer + Random Forest**

**Tabla

Descripción generada automáticamente con confianza media**

* **TfidfVectorizer + SVM**

**Imagen que contiene Tabla

Descripción generada automáticamente**

* 1. **Modelo de regresión**

Al realizar el modelo de regresión utilizando Linear Regression para su entrenamiento, nos hemos dado cuenta de que los resultados de entrenamiento producen valores óptimos, entrenados mediante regresión básica y aplicándole regularización de segundo tipo (L2), pero que, por el contrario, los valores resultantes tras las pruebas del test salen números negativos. Probando diferentes maneras de optimizarlo, incluso añadiendo un escalado de los datos con StandardScaler, no hemos podido mejorar la precisión en el conjunto de test. Desconocemos el por qué.

* 1. **Modelo de regresión erróneo (Regresión Logística es clasificación)**

Construimos a su vez un modelo de clasificación gracias a la librería LogisticRegression utilizando nuevamente ambos vectorizadores, para obtener resultados más completos, mostrados debajo:

* **Utilizando CountVectorizer**

**Tabla

Descripción generada automáticamente con confianza baja**

* **Utilizando TfidfVectorizer**

**Tabla

Descripción generada automáticamente con confianza media**

# **Conclusión**

Como podemos apreciar, todos los modelos de entrenamiento devuelven una precisión bastante similar, en torno al 75%, la cual consideramos óptima ya que nuestro fin era hacer una distinción entre las reseñas consideradas positivas y aquellas consideradas como negativas. El modelo que mejores resultados nos ha dado ha sido el que lleva a cabo la combinación del vectorizador TfidfVectorizer con el modelo de clasificación Random Forest, llegando a una precisión de hasta el 77%.

A partir de ahí, y dependiendo de lo que pudiéramos pretender buscar en nuestro problema, podríamos seleccionar aquella combinación que otorga mejores resultados a las características que buscamos. Por ejemplo, si damos mayor prioridad a la precisión de aquellos resultados positivos, que verdaderamente son positivos (medida ‘precision’), la mejor combinación sería aquella que une CountVectorizer junto con el clasificador SVM, llegando a una precisión media de hasta el 77%.

Como conclusión, podemos afirmar que la realización de un sentiment analysis de las opiniones de sus clientes por parte de las empresas cobra un valor fundamental. Tener el conocimiento de lo que se esconde tras las críticas o reseñas de los usuarios puede ayudar a la compañía a anticiparse a las tendencias futuras en las que su público pueda estar a punto de entrar, así como detectar qué productos están consiguiendo una mayor aceptación y, por tanto, mayor rendimiento económico, identificando aquellos que los usuarios determinan que no cumplen sus expectativas y que serán eliminados del mercado.