

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
INGENIERÍA DE SOFTWARE
GESTIÓN DE LA INFORMACIÓN

PROYECTO FINAL

Análisis de sentimientos de videojuegos

Prof. Luis Mendoza

Integrantes:

Bryant, Yinela

Navarro, José

García, Donovan

Alvarado, Diego

Ortiz, Cristopher

Grupo: 1SF-141

04/07/2024

Contenido

Introducción	4
Contexto del proyecto.....	4
Objetivos	4
Metodología	5
Descripción de la metodología utilizada.....	5
Selección de los Datos.....	5
Preprocesamiento de los datos.....	6
Tokenización.....	6
Análisis de polaridad.....	7
Detalles técnicos (herramientas, tecnologías, etc.) para el desarrollo (Explicación)	8
Descripción del proceso de implementación	10
Problemas enfrentados y soluciones implementadas.....	12
Resultados.....	12
Presentación y gráficos representativos de los resultados obtenidos	12
Análisis de los resultados	18
Comparación de los resultados obtenidos con la métrica que aplique al caso.....	19
Conclusiones	21
Referencias bibliográficas	22

Tabla de ilustraciones

Ilustración 1. Preprocesamiento de datos	10
Ilustración 2. Tokenización del registro 10 de la columna de summary	11
Ilustración 3. Funcionamiento del análisis de sentimientos usando VADER.....	11
Ilustración 4. Funcionamiento del análisis de sentimientos del modelo entrenado	11
Ilustración 5. Gráfico de barra Distribución de Sentimientos usando VADER.....	12
Ilustración 6. Gráfico de caja Comparación de Sentimientos usando VADER.....	13
Ilustración 7. Gráfico de barras predicción de sentimiento de acuerdo al puntaje real usando VADER	13
Ilustración 8. Matriz de confusión usando VADER	14
Ilustración 9. Matriz de confusión usando el modelo personalizado	14
Ilustración 10. Gráfica de barra de Distribución de Sentimientos usando el modelo personalizado	15
Ilustración 11. Gráfica de caja de Comparación de Sentimientos usando el modelo personalizado	15
Ilustración 12. Gráfica de barra de predicción de sentimiento de acuerdo al puntaje real usando usando el modelo personalizado	16
Ilustración 13. Gráfica de barra de reseñas por año y sentimiento de Diablo 3 de PC (B00178630A).....	16
Ilustración 14. Gráfica de barra de reseñas por año y sentimiento de God of War 3 de PS3 (B000ZK9QCS).....	17
Ilustración 15. Gráfica de barra de reseñas por año y sentimiento de The Last Of Us de PS4 (B00JK00S0S)	17
Ilustración 16. Gráfica de barra de reseñas por año y sentimiento de Star Craft 2 de PC (B000ZKA0J6)	18

Introducción

Contexto del proyecto

El avance de la tecnología ha revolucionado numerosas áreas, entre ellas el comercio electrónico. Amazon, una de las plataformas de comercio electrónico más grandes del mundo, acumula diariamente una inmensa cantidad de datos valiosos provenientes de las reseñas de usuarios sobre diversos productos, incluidos los videojuegos. Este tipo de datos es crucial ya que refleja las opiniones y experiencias de los usuarios, ofreciendo información invaluable tanto para consumidores como para desarrolladores y vendedores de videojuegos.

El dataset utilizado en este proyecto contiene reseñas de usuarios de videojuegos en Amazon, abarcando casi medio millón de entradas. Este conjunto de datos incluye información detallada sobre cada reseña, como la identificación del revisor y del producto, el texto de la reseña, la calificación otorgada, y los votos de utilidad recibidos por la reseña, entre otros.

Las reseñas en Amazon no solo sirven como feedback para los vendedores y desarrolladores, sino que también ayudan a otros usuarios a tomar decisiones informadas sobre qué productos comprar. Sin embargo, debido al volumen masivo de datos, es insostenible procesar y analizar manualmente cada reseña. Aquí es donde entran en juego las técnicas automatizadas de procesamiento del lenguaje natural (PLN).

El análisis de sentimientos y la minería de opiniones son subcampos del PLN que se centran en la extracción y clasificación de opiniones y emociones expresadas en el texto. Estas técnicas son esenciales para transformar grandes volúmenes de texto no estructurado en información útil en la toma de decisiones.

La tarea fundamental del análisis de sentimientos es clasificar la orientación de un texto, es decir, determinar si la opinión expresada es positiva, negativa o neutra. Esta clasificación puede ser especialmente útil en el contexto de las reseñas de videojuegos, donde las opiniones de los usuarios pueden influir significativamente en las decisiones de compra de otros usuarios.

Objetivos

Objetivo principal: Implementar un análisis de sentimientos de reseñas de videojuegos en Amazon utilizando Python.

Objetivos específicos:

- **Recopilación de Datos:** Utilizar el dataset de reseñas de videojuegos en Amazon para obtener una amplia muestra de opiniones de los usuarios.
- **Preprocesamiento de Datos:** Limpiar y preparar los datos para el análisis, incluyendo tareas como la normalización del texto y la tokenización.
- **Aplicación de Algoritmos de Clasificación:** Emplear diferentes librerías y algoritmos de aprendizaje automático disponibles en Python, tales como Naive Bayes, Support

Vector Machines (SVM) y árboles de decisión, para clasificar las reseñas en positivas, negativas y neutras.

- Evaluación del Rendimiento: Evaluar el rendimiento de los modelos utilizando métricas como precisión, recall y F1-score, para determinar la eficacia de cada algoritmo en la clasificación de las reseñas.
- Interpretación de Resultados: Analizar los resultados obtenidos para identificar patrones y tendencias en las reseñas de videojuegos, proporcionando observaciones valiosas sobre la percepción del usuario y la calidad de los productos.

Metodología

Descripción de la metodología utilizada

Selección de los Datos

Para este proyecto, se seleccionaron datos de las reseñas de videojuegos en Amazon. Este conjunto de datos es una versión actualizada del conjunto de datos de reseñas de Amazon lanzado en 2014. La elección de este dataset se debe a su riqueza y diversidad, que proporcionan una amplia gama de información valiosa para el análisis de sentimientos. El dataset cuenta con aproximadamente 497,577 filas y consta de las siguientes columnas: `reviewerID` (ID del revisor, e.g., A2SUAM1J3GNN3B), `asin` (ID del producto, e.g., 0000013714), `reviewerName` (nombre del revisor), `vote` (votos útiles de la reseña), `style` (un diccionario de los metadatos del producto, e.g., "Format" es "Hardcover"), `reviewText` (texto de la reseña), `overall` (calificación del producto), `summary` (resumen de la reseña), `unixReviewTime` (hora de la reseña en tiempo UNIX), `reviewTime` (hora de la reseña en formato raw) e `image` (imágenes que los usuarios publican después de recibir el producto).

La selección de este conjunto de datos se basa en varios factores clave. En primer lugar, el volumen de datos, con casi medio millón de reseñas, ofrece una base sólida para la extracción de patrones y tendencias significativas. En segundo lugar, la diversidad de atributos disponibles ofrece varias vías de análisis, desde la evaluación de la calidad de las reseñas hasta el análisis de tendencias temporales. Además, el texto de las reseñas proporciona un recurso para el análisis de sentimientos, permitiendo la identificación de opiniones positivas y negativas, así como la extracción de conocimientos sobre la satisfacción del usuario.

En el proyecto utilizamos un enfoque híbrido que combina el modelo de detección basado en bolsas de palabras (BoW) con el procesamiento del lenguaje natural proporcionado por la biblioteca spaCy.

Construimos una Bolsa de Palabras (BoW) específica para nuestro dominio de interés. Esto implica seleccionar y definir un conjunto de palabras clave y términos relevantes que no están necesariamente cubiertos por los modelos de spaCy. Este diccionario de términos puede incluir jerga especializada, nombres de productos, términos técnicos o palabras claves específicas que son importantes para nuestro análisis.

Los resultados de la detección de entidades mediante spaCy y la Bolsa de Palabras se combinan para formar un conjunto unificado de entidades reconocidas.

Preprocesamiento de los datos

Para el preprocesamiento de datos, se utilizó el lenguaje de programación Python debido a su versatilidad y a la amplia disponibilidad de bibliotecas específicas para manipulación y análisis de datos.

Seguimos los siguientes pasos:

1. Lectura del Archivo
2. Conversión a DataFrame: Convertimos el diccionario de datos a un DataFrame de Pandas. Esto nos permite aprovechar las poderosas herramientas de manipulación de datos que ofrece Pandas.
3. Limpieza de Datos: Eliminamos columnas que no son relevantes para nuestro análisis. En este caso, las columnas `reviewerID` y `reviewerName` fueron consideradas innecesarias y se eliminaron del DataFrame.
4. Eliminación de registros nulos: en la columnas reviewText habían 192 registros nulos y en la columna de summary habían 120 registros nulos. Debido a que estas cantidades no representan un gran porcentaje de los datos, decidimos eliminarlas.

Tokenización

Utilizamos la biblioteca SpaCy para realizar la tokenización y visualización de las dependencias gramaticales de un texto. En primer lugar, cargamos el modelo pre entrenado de SpaCy para el idioma inglés. Luego, creamos una función que limpia y tokeniza el texto de una reseña. Esta función elimina comillas y otros caracteres no deseados del texto, lo procesa con SpaCy para obtener los tokens y devuelve una lista de los tokens resultantes. Posteriormente, definimos otra función que toma el texto limpio y tokenizado y lo visualiza en forma de árbol de dependencias gramaticales utilizando la función `displacy.render()` de SpaCy. Esta función toma como argumentos el texto tokenizado, el estilo de visualización ('dep' para visualizar las dependencias gramaticales), un parámetro para indicar que estamos ejecutando en Google Colab (jupyter=True) y opciones adicionales de visualización, como el espaciado entre las palabras. Finalmente, aplicamos estas funciones al texto de reseña en nuestro conjunto de datos.

Este sería un ejemplo del resultado de la función:

Tokens: ['My', 'boys', 'enjoys', 'these', 'strategy', 'games']

En la función donde tokeniza las reseñas, además lematiza. La lematización es el proceso de reducir palabras a su forma base o lema, lo que ayuda a manejar las variaciones morfológicas de una palabra. Por ejemplo, las formas "corriendo", "corrió" y "correr" se reducen a "correr". Este proceso facilita la detección de entidades al estandarizar las formas de las palabras, asegurando que las variaciones no afecten la precisión del análisis.

En la limpieza y estandarización de las reseñas también:

- Utilizando expresiones regulares, reemplazamos URLs y menciones de usuarios (ej. @usuario) con espacios en blanco.
- Convertimos todas las palabras a minúsculas para asegurar la uniformidad en el análisis, evitando problemas con el reconocimiento de entidades sensibles a mayúsculas.
- Limpiamos el texto eliminando caracteres especiales o puntuación no relevante que podría interferir con el proceso de tokenización.
- Para manejar contracciones comunes en el texto, utilizamos la biblioteca `contractions`. Esta biblioteca expande contracciones como "don't" a "do not", "you're" a "you are", y así sucesivamente. Esto ayuda a mantener la consistencia en el análisis y mejora la precisión en la detección de entidades.

Análisis de polaridad

VADER

Para la realización del análisis de polaridad utilizamos la biblioteca `VaderSentiment`, esta es una herramienta popular en Python para el análisis de sentimientos.

La función principal de esta biblioteca es evaluar el sentimiento de texto basado en una combinación de reglas y léxicos predefinidos. La clase `SentimentIntensityAnalyzer` es la más importante en esta biblioteca, y se utiliza para realizar el análisis de sentimientos en el texto. El funcionamiento de esta clase es evaluar el sentimiento de un texto dado y proporciona una puntuación de sentimiento que incluye valores de positividad, negatividad, neutralidad y una puntuación compuesta. Esta puntuación va desde -1 (muy negativo) hasta 1 (muy positivo). En nuestro caso, toma el texto de la columna `"cleaned_reviewText"` que contiene los tokens y utilizando su modelo predefinido se encarga de asignar la puntuación.

Modelo entrenado

Para la realización de nuestro modelo específico entrenado utilizamos varias librerías como:

- `pandas`: Para manejar y manipular datos.
- `train_test_split`: Para dividir los datos en conjuntos de entrenamiento y prueba.
- `CountVectorizer`: Para convertir el texto en vectores de conteo de palabras.
- `MultinomialNB`: Para el algoritmo de Naive Bayes Multinomial.
- `accuracy_score` y `classification_report`: Para evaluar el rendimiento del modelo.

El modelo utiliza técnicas de procesamiento de lenguaje natural y aprendizaje automático para realizar análisis de sentimientos en reseñas de texto. Primero, convierte las calificaciones numéricas en etiquetas de sentimiento ('Positivo', 'Negativo', 'Neutral') mediante la función `get_sentiment_label`. Luego, divide los datos en conjuntos de entrenamiento y prueba, utilizando `train_test_split` para asegurar que el modelo se evalúe con datos no vistos.

Posteriormente, las reseñas de texto se vectorizan con `CountVectorizer`, transformando el texto en matrices de conteo de palabras. El modelo de Naive Bayes Multinomial se entrena con estas matrices, ajustándose a los datos de entrenamiento. Una vez entrenado, el modelo predice los sentimientos de las reseñas en el conjunto de prueba.

Para evaluar el rendimiento del modelo, se calculan la precisión, el recall y el f1-score, mostrando que el modelo tiene una precisión general del 82%. Finalmente, el modelo se

aplica a todo el DataFrame original para predecir los sentimientos de todas las reseñas, y se muestran algunas filas del DataFrame con los sentimientos reales y predichos, ilustrando el rendimiento del modelo en ejemplos concretos.

Detalles técnicos (herramientas, tecnologías, etc.) para el desarrollo (Explicación)

Kaggle

Es una plataforma en línea que proporciona conjuntos de datos, herramientas de análisis y un entorno de codificación basado en la nube para científicos de datos y desarrolladores. En este proyecto, Kaggle fue utilizado para obtener el conjunto de datos de reseñas de videojuegos en Amazon.

Python

Es un lenguaje de programación de alto nivel ampliamente utilizado en el campo de la ciencia de datos y el aprendizaje automático. En este proyecto, Python se utilizó para llevar a cabo varias tareas esenciales, incluyendo el preprocesamiento de datos, la limpieza de datos, y la conversión de los datos al formato adecuado para su análisis. Algunas de las bibliotecas clave de Python utilizadas en este proyecto son: Pandas y NumPy.

Procesamiento de Lenguaje Natural (NLP, por sus siglas en Inglés):

Es un campo de la inteligencia artificial y lingüística computacional que se enfoca en la interacción entre las computadoras y los humanos a través del lenguaje natural. Python es uno de los lenguajes de programación más populares para el desarrollo de aplicaciones de NLP debido a sus bibliotecas robustas y fáciles de usar.

Scikit-learn (sklearn):

Es una librería de python de código abierto mayormente utilizada para el aprendizaje automático y el análisis de datos, esta es eficiente y muy fácil de usar. Cuenta con clasificación, regresión y agrupación, algoritmos vectoriales, bosques aleatorios, etc.

Características:

- **Modelos de Clasificación:** K-Nearest Neighbors, Support Vector Machines, Decision Trees, Random Forests, Logistic Regression, Naive Bayes, entre otros.
- **Modelos de Regresión:** Linear Regression, Ridge Regression, Lasso, Elastic Net, y más.
- **Agrupamiento (Clustering):** K-Means, DBSCAN, Mean Shift, Agglomerative Clustering, entre otros.
- **Reducción de dimensionalidad:** PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), t-SNE, etc.

NLTK (Natural Language Toolkit)

NLTK es una de las bibliotecas más completas y utilizadas para el procesamiento de lenguaje natural en Python. Ofrece herramientas para trabajar con textos, etiquetado de partes del discurso, análisis sintáctico, análisis semántico y más.

Características:

- **Tokenización:** División de textos en palabras o frases.
- **Etiquetado de Partes del Discurso** (POS tagging): Asignación de etiquetas gramaticales a cada palabra en un texto.
- **Lematización y Sistematización:** Reducción de palabras a su forma base o raíz.
- **Análisis de Sentimiento:** Evaluación del tono emocional del texto.
- **Corpora y Modelos Pre-Entrenados:** Acceso a una variedad de corpus y modelos para tareas comunes en NLP.

Algunas bibliotecas de NLP en Python:

- **spaCy:** Biblioteca de NLP de alta eficiencia y rapidez diseñada para tareas de procesamiento a nivel industrial. Es conocida por su rapidez y por ofrecer modelos pre-entrenados de alta calidad.
- **TextBlob:** Biblioteca sencilla y fácil de usar para el procesamiento de textos en Python. Es adecuada para proyectos pequeños y aplicaciones donde la simplicidad y la rapidez de desarrollo son prioritarias.

Pandas:

- Librería de Python especializada en el manejo de análisis de estructuras de datos.
- Ofrece a los usuarios una amplia biblioteca de datos para explorar y es un recurso común para científicos y análisis de datos.
- Es utilizado para explorar, limpiar, transformar, visualizar y analizar datos de forma rápida y eficiente.

Algunas características de Pandas:

- **Estructuras de datos.** Ofrece una variedad de estructuras de datos de Python, incluidas Series y Data Frames, para facilitar el trabajo con datos.
- **Indexación.** Pandas te permite indexar datos de forma rápida y sencilla, permitiéndote acceder a elementos específicos dentro de un DataFrame.
- **Limpieza de datos.** Pandas proporciona varios métodos para limpiar e imputar datos, lo que le facilita trabajar con conjuntos de datos desordenados.
- **Manipulación de datos.** Pandas proporciona un conjunto de funciones integradas para manipular datos, incluida la clasificación, el filtrado y la agregación.
- **Visualización de datos.** Pandas también ofrece bibliotecas de trazado integradas, lo que facilita la visualización de datos de forma rápida y sencilla.

NumPy:

- Librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.

- Esta incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación.
- Esta librería permite que el procesamiento de los arrays se realice mucho más rápido que las listas, lo cual lo hace ideal para el procesamiento de vectores y matrices de grandes dimensiones.

Algunas características de NumPy:

- **Arrays multidimensionales:** Los arrays de NumPy son estructuras de datos altamente eficientes que pueden almacenar elementos de un mismo tipo en múltiples dimensiones, desde matrices unidimensionales hasta matrices multidimensionales.
- **Operaciones vectorizadas:** NumPy permite realizar operaciones aritméticas y matemáticas de manera vectorizada, lo que significa que las operaciones se aplican a todo el array sin necesidad de escribir bucles explícitos. Esto resulta en un código más limpio y eficiente.
- **Funciones matemáticas y estadísticas:** NumPy proporciona una amplia gama de funciones matemáticas y estadísticas, como operaciones trigonométricas, logarítmicas, exponenciales, agregaciones (suma, media, mediana, desviación estándar), entre otras.
- **Broadcasting:** El broadcasting es una técnica que permite realizar operaciones aritméticas en arrays de diferentes formas de manera eficiente, sin necesidad de duplicar los datos.

Descripción del proceso de implementación



Ilustración 1. Preprocesamiento de datos

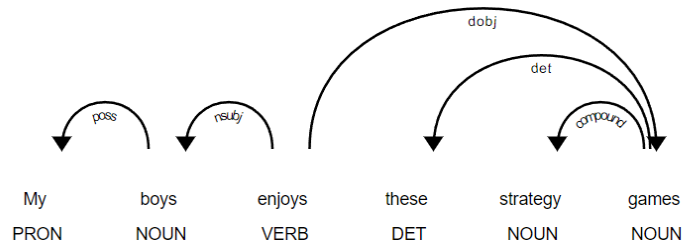


Ilustración 2. Tokenización del registro 10 de la columna de summary

Aunque pensamos utilizar la columna de “reviewText” para el análisis de sentimientos, tokenizamos un registro de la columna “summary” por fines de visualización.

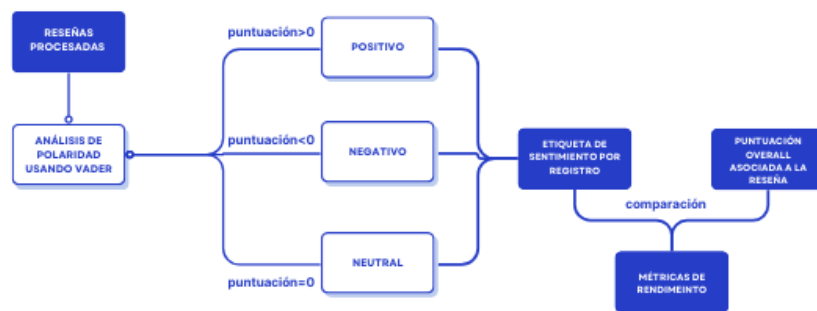


Ilustración 3. Funcionamiento del análisis de sentimientos usando VADER

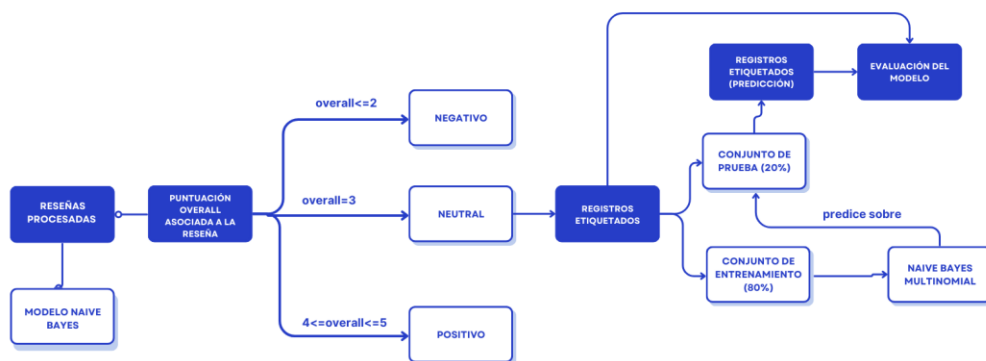


Ilustración 4. Funcionamiento del análisis de sentimientos del modelo entrenado

Problemas enfrentados y soluciones implementadas.

1. Fue problemático la gran cantidad de reseñas (casi medio millón) que tomaban más de cuatro horas en procesar, lo solucionamos creando el doc de NLP solo una vez y guardando el dataframe nuevo en formato .csv, los siguientes pasos se realizaron con el .csv nuevo.
2. El análisis de sentimientos del modelo pre-entrenado para el idioma inglés VADER, no se ajustaba bien a las necesidades de nuestros dataset, lo solucionamos creando un modelo más personalizado con la librería sklearn de Python.

Resultados

Presentación y gráficos representativos de los resultados obtenidos

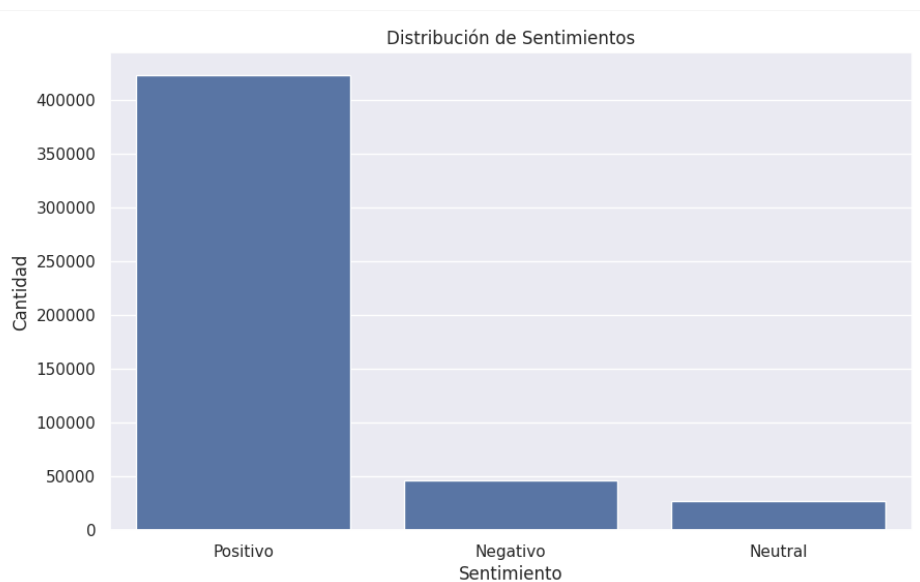


Ilustración 5. Gráfico de barra Distribución de Sentimientos usando VADER

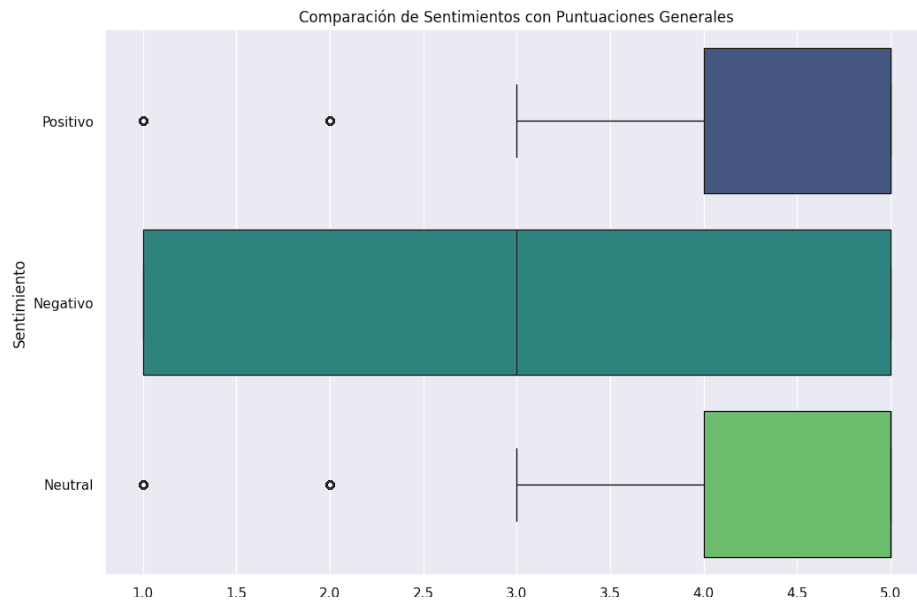


Ilustración 6. Gráfico de caja Comparación de Sentimientos usando VADER

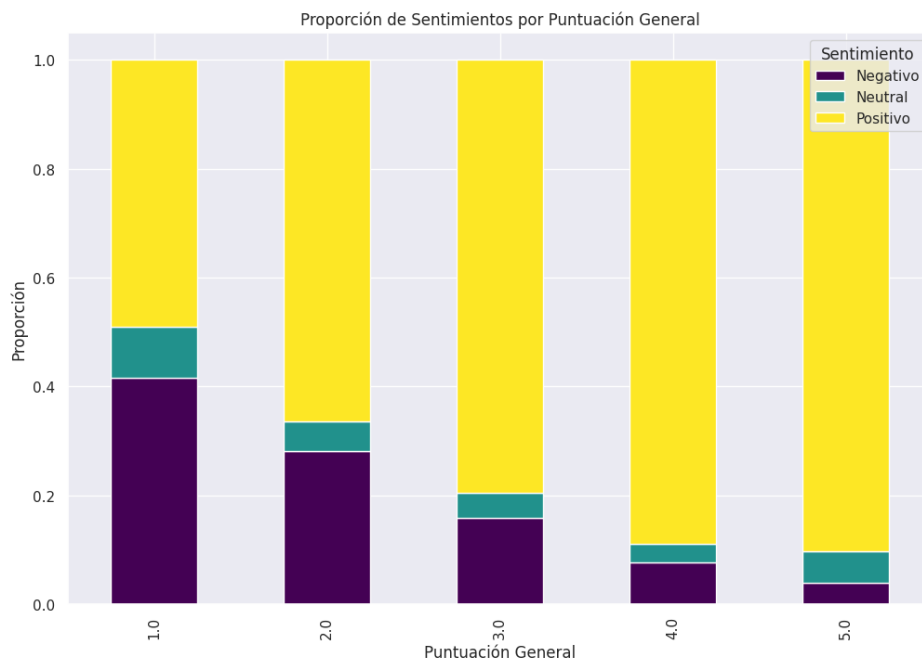


Ilustración 7. Gráfico de barras predicción de sentimiento de acuerdo al puntaje real usando VADER

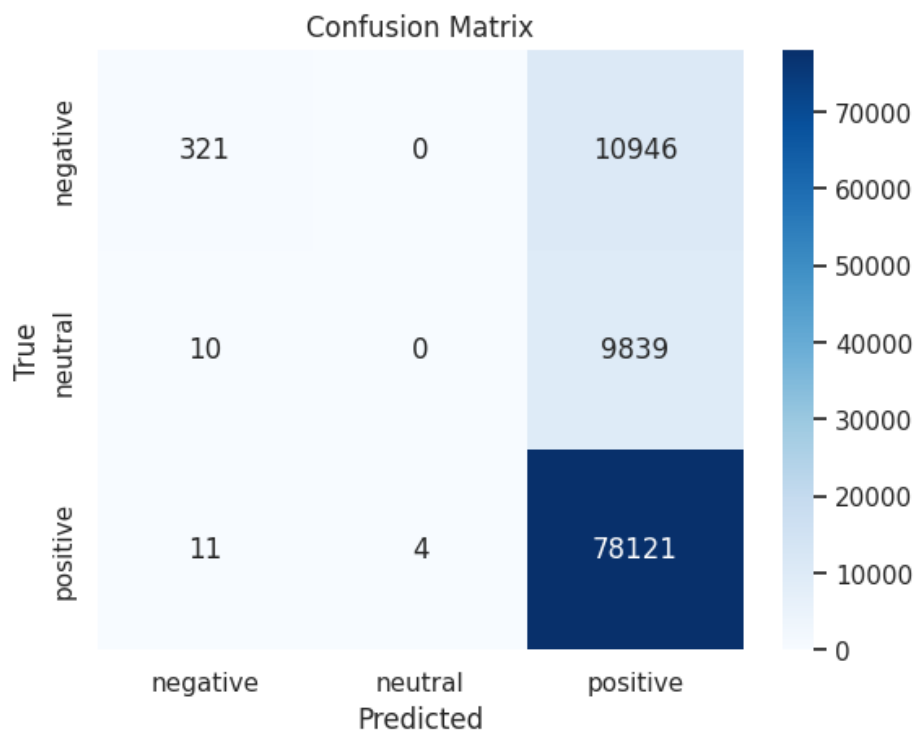


Ilustración 8. Matriz de confusión usando VADER

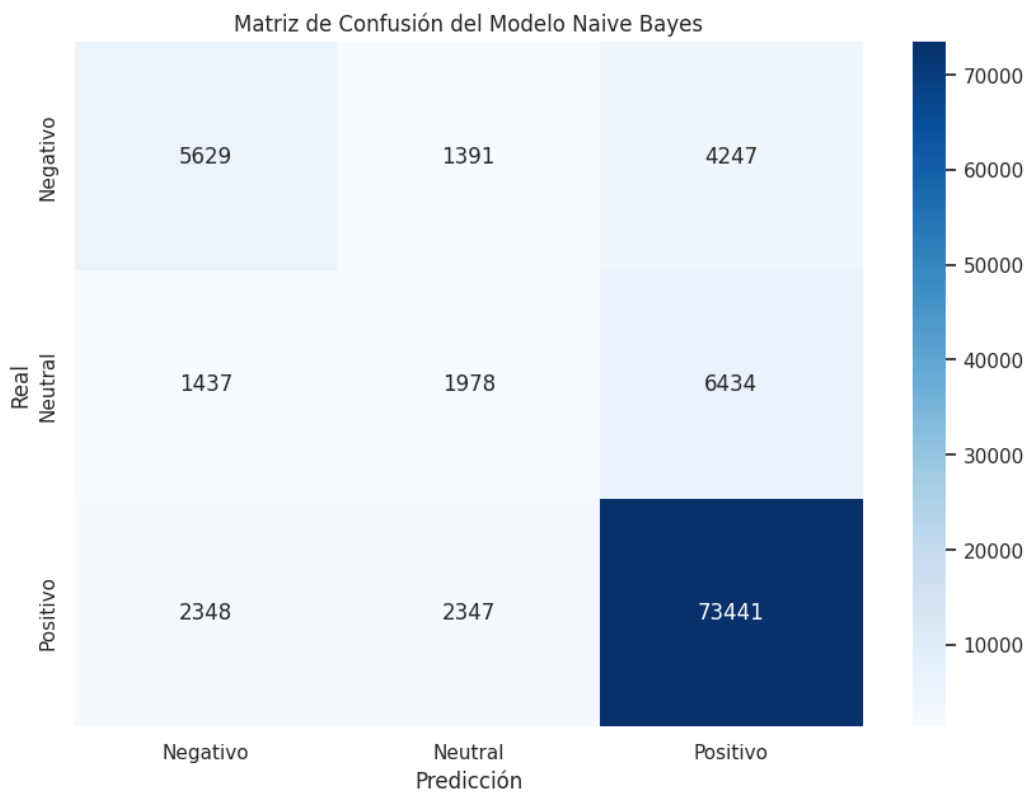


Ilustración 9. Matriz de confusión usando el modelo personalizado

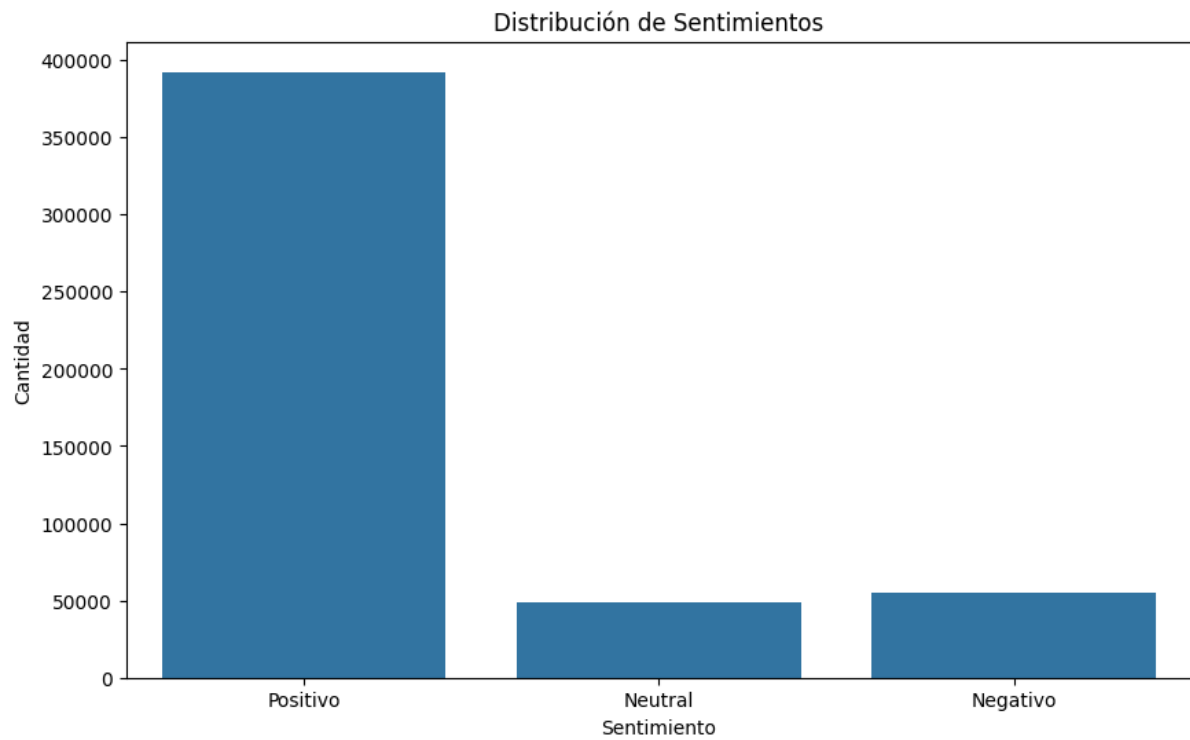


Ilustración 10. Gráfica de barra de Distribución de Sentimientos usando el modelo personalizado

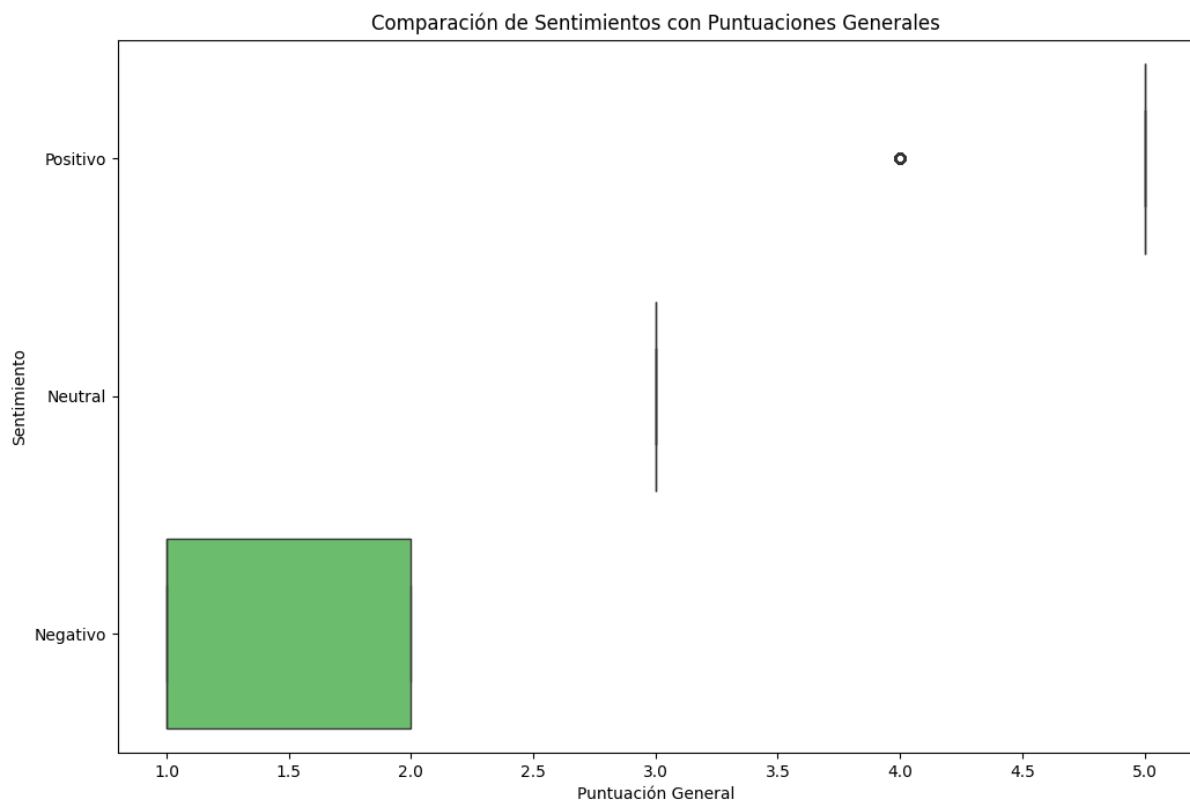


Ilustración 11. Gráfica de caja de Comparación de Sentimientos usando el modelo personalizado

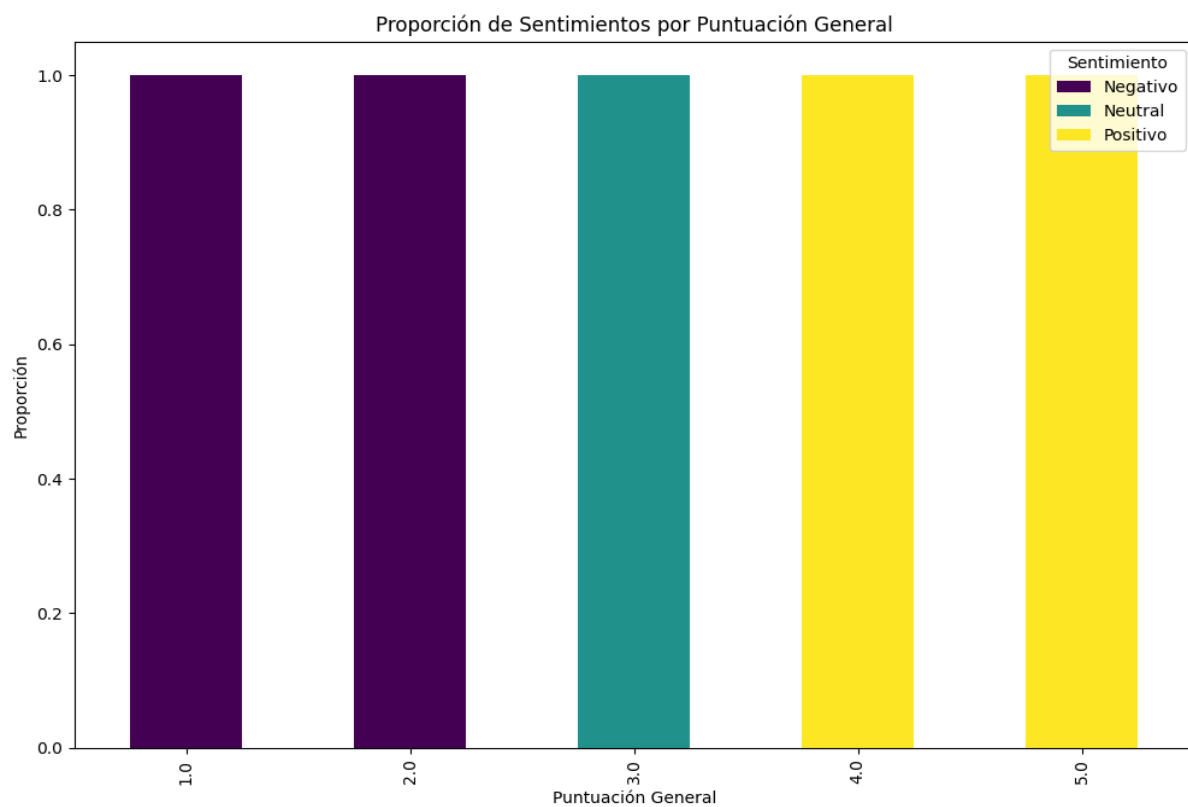


Ilustración 12. Gráfica de barra de predicción de sentimiento de acuerdo al puntaje real usando usando el modelo personalizado

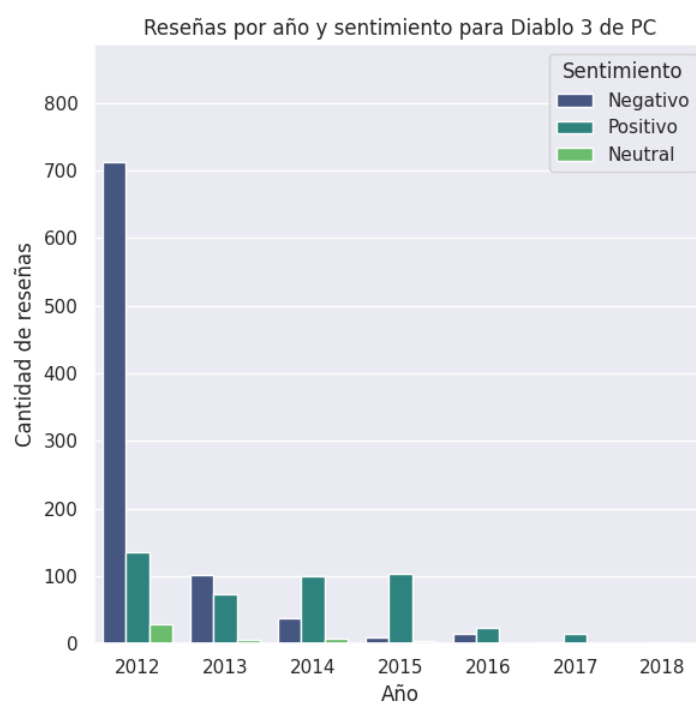


Ilustración 13. Gráfica de barra de reseñas por año y sentimiento de Diablo 3 de PC (B00178630A)

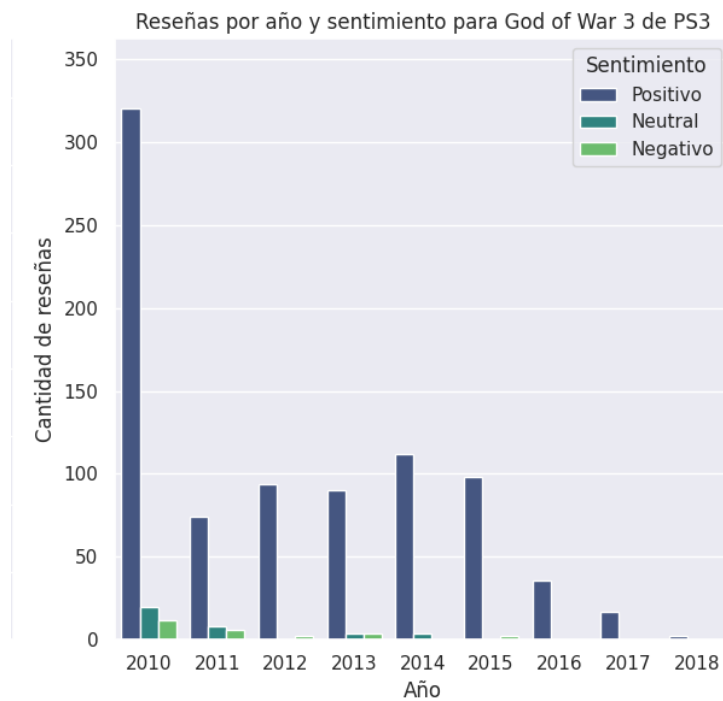


Ilustración 14. Gráfica de barra de reseñas por año y sentimiento de God of War 3 de PS3 (B000ZK9QCS)

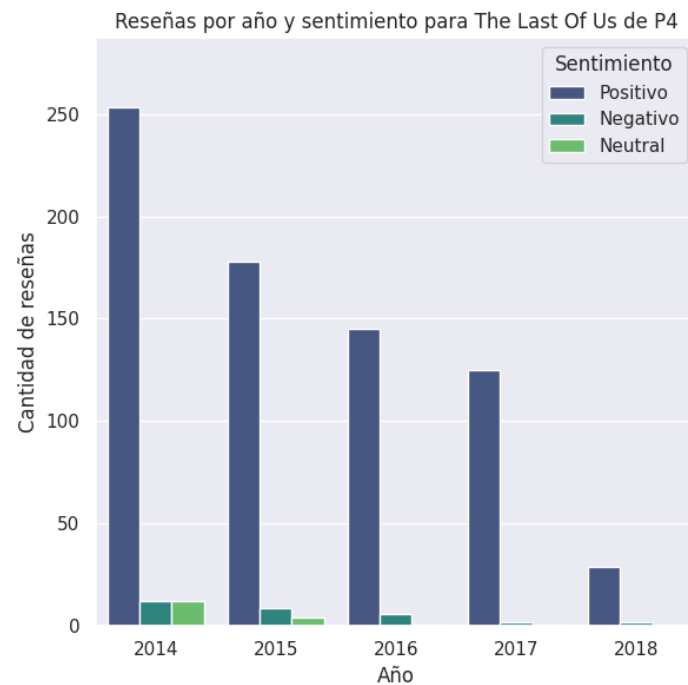


Ilustración 15. Gráfica de barra de reseñas por año y sentimiento de The Last Of Us de PS4 (B00JK00S0S)

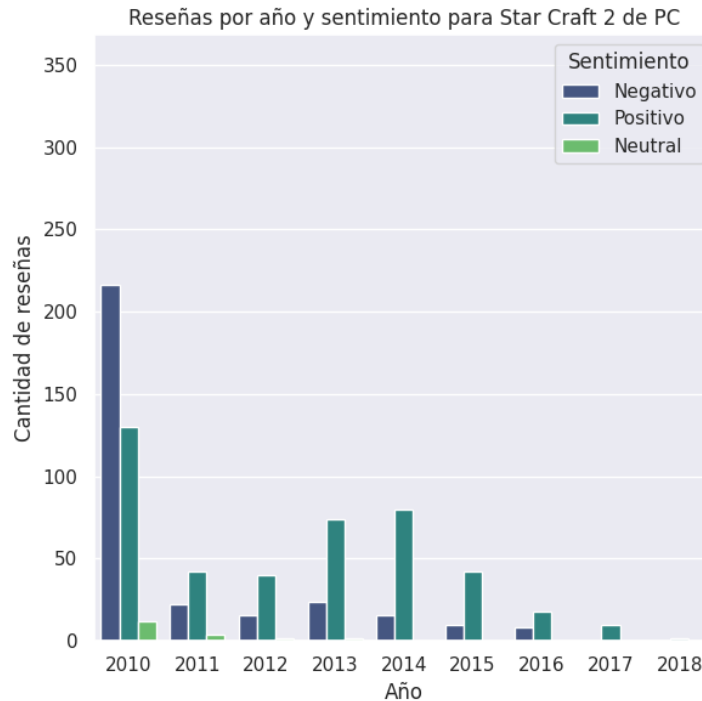


Ilustración 16. Gráfica de barra de reseñas por año y sentimiento de Star Craft 2 de PC (B000ZKA0J6)

Análisis de los resultados

VADER

En la imagen 5, observamos que la gran mayoría de resultados se ven concentrados en reseñas positivas, esto puede decirnos que la información de nuestro dataset contiene gran información de reseñas positivas que el modelo VADER identificó.

Al realizar un diagrama de comparación de sentimientos con la puntuación dada (imagen 6), se presenta una peculiaridad y es que en las reseñas negativas hay valores de 3, 4 y 5 estrellas que los usuarios le colocaron. Esto puede significar:

- Que los usuarios expresaron su descontento con algún aspecto específico del juego, pero valoraron el juego positivamente ya que entienden que es una gran entrega.
- Que el modelo no está reconociendo bien el sentimiento negativo.

Probablemente sea la segunda opción, ya que el gráfico de proporción de sentimientos (imagen 7), presenta otra gran anomalía pues en puntuación de 1 estrella contiene mayoría de reseñas etiquetadas como positivas, lo que resulta muy poco probable. Esto nos indica que nuestro modelo está careciendo de identificar y clasificar algunos comentarios negativos y está teniendo parcialidad por los positivos.

Utilizando la matriz de confusión podemos hacer salvedad de que el modelo está presentando algunos problemas más que todo tratando de clasificar los comentarios negativos, ya que con esta matriz nos percatamos de que hay un número exagerado de falsos positivos.

Para solucionar este problema de identificación y clasificación de los sentimientos de las reseñas de videojuegos hemos decidido utilizar un modelo personalizado en este caso optamos por el MODELO NAIVE BAYES, ya que puede aprender patrones complejos a partir de datos etiquetados, capturando una gama más amplia de sentimientos y permitiendo una interpretación más rica de la relación entre la puntuación y el sentimiento.

Luego de haberle aplicado el modelo Naive Bayes podemos observar que los datos han cambiado y se han distribuido de mejor manera en cuanto a las reseñas(imagen 10) y la puntuación dada por el usuario, esto debido a que el modelo a sido entrenado para identificar e interpretar de mejor el tipo de sentimiento (positivo, neutro, negativo) de las reseñas presentadas por los usuarios , esto también se ve reflejado es la disminución de los falsos positivos (imagen 9).

En cuanto al sentimiento expresado por los usuarios podemos resaltar dos casos importantes, el primer caso (imagen 13) en cuanto al juego Diablo III el cual tuvo una enorme carga de reseñas negativas que se ven reflejada en la gráfica a la hora de su lanzamiento, y esto se debió a unos errores en los servidores del videojuego que impedía disfrutar de la entrega y debido a esto hubo tantas reseñas negativas.

El otro caso (imagen 14) podemos hablar de God of War III, las reseñas de este juego tuvieron una tendencia positiva, debido a que esta entrega es la más galardonada de la saga God of War, su jugabilidad, historia y gráficos hicieron que este juego se mantuviera en el top de los usuarios y así se vio reflejado en la parte estadística.

Comparación de los resultados obtenidos con la métrica que aplique al caso

Métricas	VADER			Modelo Naive Bayes		
Precisión	Negativo 0.08	0.42 Positivo	Neutral 0.83	Negativo 0.60	Neutral 0.35	Positivo 0.87
Recall	Negativo Neutral Positivo	0.36 0.05 0.90		Negativo Neutral Positivo	0.50 0.20 0.94	
F1-Score	Negativo Neutral Positivo	0.39 0.06 0.90		Negativo Neutral Positivo	0.54 0.25 0.91	

Precisión

Una alta precisión indica que el modelo tiende a predecir correctamente la clase específica cuando hace una predicción positiva para esa clase.

VADER, aunque tiene una presión alta para la clase positiva, es muy deficiente para predecir la clase negativa y neutral. Con el modelo Naive Bayes mejoran las métricas un poco, aunque igualmente tiene mayor precisión para la clase positiva.

Recall (Sensibilidad)

Un alto recall indica que el modelo es capaz de identificar la mayoría de las instancias que realmente pertenecen a esa clase específica.

VADER tiene un recall muy alto para la clase Positivo (0.90) pero muy bajo para Neutral (0.05). Naive Bayes también muestra un alto recall para Positivo (0.94) y un mejor desempeño para Negativo (0.50) y Neutral (0.20) en comparación con VADER.

F1-Score

El F1-Score es una medida combinada que proporciona un balance entre precisión y recall.

Un valor alto indica un buen equilibrio en la capacidad del modelo para identificar correctamente instancias positivas.

Naive Bayes tiene un F1-Score más equilibrado con un alto desempeño en Positivo (0.91), pero también una mejora significativa en Negativo (0.54) y Neutral (0.25) en comparación con VADER.

Conclusiones

El análisis de sentimientos mediante inteligencia artificial es una herramienta crucial para entender opiniones y emociones expresadas en texto, permitiendo a organizaciones tomar decisiones informadas, mejorar productos y servicios, monitorear la marca y realizar investigaciones de mercado.

En este proyecto pudimos simular lo que haría una empresa como Amazon, analizando los sentimientos de aproximadamente 500,000 reseñas de videojuegos. Pudimos comparar dos enfoques: Naive Bayes y VADER. Naive Bayes demostró ser una opción equilibrada, capturando con precisión sentimientos negativos y neutrales, lo cual es ideal para análisis complejos que requieren un entendimiento matizado de las emociones.

Por otro lado, VADER mostró una fuerte capacidad para identificar sentimientos positivos, siendo útil cuando el objetivo es resaltar comentarios positivos, aunque su bajo rendimiento en clasificaciones negativas y neutrales puede limitar su aplicabilidad en algunos contextos. La elección del modelo depende de los objetivos específicos del análisis, donde Naive Bayes es mejor para un análisis equilibrado y VADER es ventajoso para identificar sentimientos positivos.

Referencias bibliográficas

[1] N. author found, "Sentiment analysis for spanish reviews on Google Play Store using BERT." [Online]. Available: https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052021000300557&lng=en&nrm=iso&tlng=en

[2] A. S. Alberca, "La librería Numpy | Aprende con Alf." [Online]. Available: <https://aprendeconalf.es/docencia/python/manual/numpy/>

[3] A. S. Alberca, "La librería Pandas | Aprende con Alf." [Online]. Available: <https://aprendeconalf.es/docencia/python/manual/pandas/>

[4] N. author found, "NLP in Python - HBS Grid Documentation." [Online]. Available: https://hbs-rcs.github.io/hbsgrid-docs/tutorials/nlp_with_python/

[5] N. author found, "NLTK :: Natural Language Toolkit." [Online]. Available: <https://www.nltk.org/>

[6] N. author found, "transformers · PyPI." [Online]. Available: <https://pypi.org/project/transformers/>

[7] U. Kant, "Tokenization — A complete guide - Utkarsh Kant - Medium," Medium, Mar. 19, 2022. [Online]. Available: <https://medium.com/@utkarsh.kant/tokenization-a-complete-guide-3f2dd56c0682>