Proyecto Grupal

Movie Genre Prediction

Alejandro Calderon Moreno 1[a,c] , Nicolas Andres Rey Ospina 2[a,c] , Ana Carolina Rubiano Manrique 3[a,c], Sebastian Zequeda Molina 4[a,c]

Sergio Alberto Mora Pardo[b,c]

[a]Estudiante de TAL(ES) PROGRAMA(S)
[b]Profesor, Departamento de Ingeniería Industrial
[c]Pontificia Universidad Javeriana, Bogotá, Colombia

1. Contexto

La Importancia De Identificar Los Gustos De Los Clientes En La Industria Cinematográfica

La industria cinematográfica ha experimentado una revolución en los últimos años, impulsada en gran parte por el avance tecnológico y la digitalización. Una de las claves para su éxito continuo ha sido la capacidad de adaptarse y satisfacer las crecientes demandas y expectativas de los consumidores. En este contexto, la identificación de las tendencias y preferencias de los clientes se ha vuelto más crucial que nunca. Una forma efectiva de hacerlo es a través de la descripción de películas que permita a los espectadores conocer su género.

El Poder De Las Descripciones De Películas

Las descripciones de películas son una herramienta fundamental en la industria cinematográfica. Sirven como puerta de entrada al mundo que una película ofrece a los espectadores. Estas descripciones no solo proporcionan información básica sobre la trama, sino que también transmiten la esencia del género al que pertenecen. Ya sea que una película sea una emocionante película de acción, una conmovedora historia de amor o un escalofriante thriller de terror, las descripciones deben capturar la esencia de lo que los espectadores pueden esperar. Por lo tanto, son una parte esencial para que los clientes elijan la película que mejor se adapte a sus preferencias.

2. Análisis Exploratorio de Datos

Se proporcionó una base de datos denominada "Training", que consta de 7,805 registros que incluyen las siguientes variables:

- **year** (Año de estreno de la película): Esta variable representa el año en que se estrenó cada película en el conjunto de datos.
- **Title** (Título de la película): Aquí se encuentra el título de cada película incluida en la base de datos.
- **plot** (Breve reseña de la película): Esta variable contiene una breve descripción o sinopsis de cada película.
- **genres** (Género de la película): Muestra el género al que pertenece cada película, lo que proporciona información sobre su categoría o estilo.
- **rating** (Calificación de la película): Esta variable indica la calificación o puntuación otorgada a cada película.

Esta base de datos servirá como punto de partida para el análisis y desarrollo de modelos relacionados con el género y la clasificación a partir de la breve reseña de la película.

En un primer análisis, se examinaron los valores nulos en la variable "plot", y no se encontró ninguno.

```
# Verificar si hay valores nulos en la columna 'plot'
nulos_en_plot = dataTraining['plot'].isnull().sum()

# Imprimir el número de valores nulos
print("Número de valores nulos en la columna 'plot':", nulos_en_plot)
```

```
Número de valores nulos en la columna 'plot': 0
```

Tabla 1: Evaluación valores nulos

Además, se llevó a cabo un recuento de los géneros cinematográficos, y se encontró que el género "drama" es el que tiene la mayor cantidad de películas, seguido por "comedia". Es importante destacar para el posterior análisis que una película puede estar catalogada en múltiples géneros.

```
conteo_genres = dataTraining['genres'].value_counts().head(10)

conteo_genres
```

```
['Drama']                          429
['Comedy']                         368
['Comedy', 'Drama', 'Romance']     306
['Comedy', 'Romance']              291
['Comedy', 'Drama']                287
['Drama', 'Romance']               282
['Documentary']                    154
['Crime', 'Drama', 'Thriller']     125
['Horror']                         115
['Drama', 'Thriller']              115
```

Tabla 2: Frecuencia de genero de películas

Se llevó a cabo un análisis de las palabras y símbolos que aparecen con mayor frecuencia en la breve reseña, y se determinó que es necesario aplicar la metodología de eliminación de stopwords. Además, se identificó la necesidad de eliminar algunas letras sueltas que quedan en el análisis.

```
# Muestra las 10 palabras más comunes
most_common_words = word_counts.most_common(10)
for word, count in most_common_words:
    print(f'{word}: {count}')
```

```
,: 57327
.: 46614
': 15571
-: 10466
``: 4776
n: 3832
one: 3010
life: 2721
new: 2255
(: 2072
```

Tabla 3: Palabras y caracteres más usados en las reseñas

Lo anterior permite identificar qué tipo de preprocesamiento puede ser relevantes para el modelo analítico.

3.Metodología Analítica

3.1 Preprocesamiento

De acuerdo con lo observado en el análisis exploratorio se definen funciones para poder realizar el preprocesamiento de nuestra base de datos, específicamente del plot de la película.

Primero, de la librería 'nltk' se importan los stopwords que se encuentran por default para inglés. Adicional a esto se crea una lista donde se pueden añadir palabras adicionales que se consideren necesarias añadir a la lista, como lo encontrado en ese análisis de exploración inicial.

```
#Añadir stopwords adicionales que se consideren necesarios
stopwords_adicionales = ["is", "the","huw","a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"]
stopwords_en.extend(stopwords_adicionales)

#Lista de los stopwords en ingles
print(list(stopwords_en))
```

Seguido de esto, se define la función "preprocesamiento" donde se realizan los siguientes procesos:

- En primera instancia, se convierte todo a lowercase.
- Luego, se eliminan todos los stopwords definidos previamente.
- Seguido de esto se eliminan todos los caracteres especiales y los números de los "plot" de las películas. Esto se realiza debido a que se llegó a la conclusión que estos caracteres no serían de gran importancia al momento de terminar el género, netamente los caracteres alfabéticos serían lo primordial para la predicción.
- Por último, se opta por lematizar los plots. (En este caso se escogió la lematización sobre stemming debido a que en los escenarios analizados, al lematizar las palabras teníamos una mejor predicción

```python
#Creacion de una función para poder realizar un preprocesamiento del plot.
def preprocesamiento(text):
  #Convertir todo a lowercase
  text = text.str.lower()
  # Eliminar stopwords
  text = text.apply(lambda x: ' '.join([word for word in x.split() if word not in stopwords_en]))
  # Quitar caracteres que no sean letras (Se eliminan caracteres especiales y numeros)
  text = text.apply(lambda x: ' '.join([re.sub(r'[^a-zA-Z]', '', word) for word in x.split()]))
  #Quitarle espacios en blanco
  text = text.apply(lambda x: ' '.join(x.split()))
  #Lematizacion
  text = text.apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for word in x.split()]))
  #Stemming Porter
  #text = text.apply(lambda x: ' '.join([stemmer.stem(word) for word in x.split()]))
  #Stemming Snowball
  #text = text.apply(lambda x: ' '.join([stemmersnow.stem(word) for word in x.split()]))

  #text = text.apply(quitar_nombres)


  return text
```

Por último, tuvimos una idea adicional donde se decidió que los nombres de personas no serían lo suficientemente significativos al momento de predecir, sin embargo, se evidenció que su impacto era negativo y por eso se decidió no optar por esta opción. A continuación, se deja la función definida (Esta fue creada con Spacy y su modelo en ingles mas pequeño)

```python
def quitar_nombres(text):
    # Tokenizacion texto con spacy
    doc = nlp(text)

    # Filtrar palabras que son nombres propios, personas, numeros ordinales y cardinales
    filtro = [token.text for token in doc if not token.ent_type_ in ('PERSON')]

    # join the las palabras filtradas
    text = ' '.join(filtro)

    return text
```

3.2 Feature Engineering

En este apartado se realiza la representación numérica en vectores del vocabulario del corpus, donde a cada palabra se asigna un identificador binario. En el desarrollo de este proyecto fueron utilizadas vectorización básica y TF - IDF.

La forma de vectorización básica convierte un conjunto de documentos de texto en una matriz de conteo de términos. Es decir, cuenta cuántas veces aparece cada palabra en cada documento y crea una matriz numérica a partir de esta información.

Por otro lado, TFIDF (Term Frequency-Inverse Document Frequency Vectorizer), asigna un valor numérico a cada palabra basado en dos componentes:

- **TF (Term Frequency)**: Mide la frecuencia de una palabra en un documento específico. Cuanto más a menudo aparece una palabra en un documento, mayor será su valor TF para ese documento.
- **IDF (Inverse Document Frequency)**: Mide la importancia de una palabra en todo el conjunto de documentos. Las palabras que son comunes en muchos documentos tendrán un IDF bajo, mientras que las palabras que son raras o específicas de un documento tendrán un IDF alto.

3.3 Modelos Planteados

Durante el desarrollo del proyecto implementamos cinco modelos de aprendizaje automático para encontrar el mejor AUC:

- **SVM (Support Vector Machine):** Algoritmo de aprendizaje supervisado que encuentra un hiperplano óptimo para separar datos en clases diferentes. Puede manejar tanto problemas de clasificación como de regresión.
- **XGBoost:** Algoritmo de aprendizaje automático que pertenece a la familia de Gradient Boosting. Se utiliza para problemas de clasificación y regresión y es conocido por su eficacia y capacidad de manejar conjuntos de datos complejos.
- **Red Neuronal:** Modelo de aprendizaje profundo inspirado en la estructura y funcionamiento del cerebro humano. Consiste en capas de nodos interconectados que procesan información para tareas de clasificación, regresión y reconocimiento de patrones, entre otros.
- **Regresión Logística:** Algoritmo de aprendizaje supervisado utilizado para problemas de clasificación binaria y multiclase. Modela la relación entre una variable dependiente categórica y una o más variables independientes mediante la función logística.

3.2 Métrica

Área Bajo la Curva ROC (AUC): Métrica de evaluación comúnmente utilizada en problemas de clasificación. Es una representación gráfica de la tasa de verdaderos positivos (Sensibilidad) frente a la tasa de falsos positivos (1 - Especificidad) para diferentes umbrales de decisión. Un valor de AUC cercano a 1 indica un modelo de clasificación muy efectivo, mientras que un valor cercano a 0.5 sugiere un rendimiento similar al azar. El AUC mide la capacidad de un modelo para distinguir entre clases positivas y negativas.

4.Solución Propuesta

Para evaluar el rendimiento de los modelos mencionados en la sección 3.3, seguimos un enfoque estándar de división de datos. Inicialmente, utilizamos el 67% de la base de entrenamiento para

entrenar los modelos, y luego reservamos el 33% restante para su evaluación. A continuación, presentamos los resultados de estas evaluaciones en las Tablas 1 y 2.

En la Tabla 1 se muestran las métricas de evaluación obtenidas al aplicar los modelos a los datos de entrenamiento. Estas métricas nos brindan una comprensión del rendimiento de los modelos en los datos con los que se entrenaron.

MODELO	AUC
REGRESIÓN LOGÍSTICA	99,7%
RED NEURONAL	87,7%
SVM	82,6%
XGBOOST	98,8%

Tabla 4: Métricas de Evaluación en Train

La Tabla 2 presenta los resultados de la evaluación de los modelos en el conjunto de prueba (test), que representa datos independientes que los modelos no habían visto durante el entrenamiento. Estas métricas son cruciales para medir la capacidad de generalización de los modelos a nuevos datos.

MODELO	AUC
REGRESIÓN LOGÍSTICA	**89%**
RED NEURONAL	58%
SVM	53%
XGBOOST	85%

Tabla 5: Métricas de Evaluación en Test.

Negrilla mejor resultado.

Todos los modelos superan el umbral del AUC del 50%, lo que indica un rendimiento prometedor en la tarea de evaluación. Sin embargo, es importante destacar que la Regresión Logística se destaca con el mejor desempeño en ambas tablas. Estos resultados resaltan la efectividad de la Regresión Logística en comparación con los otros modelos evaluados.

Los resultados de estas evaluaciones proporcionan información esencial sobre el rendimiento y la robustez de los modelos propuestos. Con base en estos resultados, se tomarán decisiones y se realizarán ajustes necesarios para lograr los objetivos establecidos en este estudio de analítica de datos

Todos los modelos superan el AUC de 50%, sin embargo la Regresión Logística presenta el mejor desempeño.

5.Conclusiones

A lo largo de este estudio de procesamiento de lenguaje natural, hemos implementado cuatro modelos de aprendizaje automático en Python, a saber: Regresión Logística, XGBoost, Redes Neuronales y Máquinas de Vectores de Soporte (SVM). Tras exhaustivas pruebas y evaluaciones, observamos que la Regresión Logística se destacó como el modelo más eficaz en nuestra tarea de predicción de género de películas. Este modelo obtuvo un impresionante valor de AUC del 89%, lo que confirma su excelente desempeño.

El AUC del 89% indica que la Regresión Logística es altamente competente en la capacidad de distinguir entre diferentes categorías de género de películas. Este resultado respalda la utilidad y la eficacia de este enfoque para la clasificación de género cinematográfico en el contexto de nuestro proyecto.

Es importante destacar que si bien la Regresión Logística superó a los otros modelos en esta tarea específica, la elección del modelo adecuado puede depender en gran medida del problema y los datos específicos. En consecuencia, esta investigación proporciona una base sólida para futuros desarrollos y mejoras en la clasificación de género de películas utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático.