Proyecto 3 - Predicción del Género de la Película

Rincon Hortua, Juan Sebastian
Codigo 201214767

Mora Pardo, Sergio Alberto
Codigo 201920547

Orjuela Viracacha, Jaime
Codigo 201924252

Aldana Martinez, Nelson
Codigo 201924128

Julio 20, 2020

1. Preparación de los datos

La base de datos plot, contiene la información de los resúmenes de las películas, el año en que se estrenó, el rating y el género o géneros a los que pertenece. Lo que se busca en este proyecto es calcular la probabilidad de que cada película pertenezca a uno u otro(s) género(s), dependiento del resumen de la misma.

De esta manera lo primero que se procedió a hacer es la preparación de los datos. Por lo que se eliminaron todos los signos de puntuación. Luego de esto se creo la variable dependiente y, como los géneros a los que pertenece cada una de las películas. De los que se obtienen la siguiente información:

Action	Horror
Adventure	Music
Animation	Musical
Biography	Mistery
Comedy	News
Crime	Romance
Documentary	Sci-Fi
Drama	Short
Family	Sport
Fantasy	Thriller
Film-Noir	War
History	Western

Cuadro 1: Géneros

Los datos de entrenamiento son 7895 entradas de películas. Una misma entrada puede tener hasta 9 géneros. Por lo que se procedió a crear con una

matriz de 7895 filas con 9 columnas, en donde cada fila contiene un género identificado en cada entrada para de esta manera saber si esa película está o no calsificada dentro de este.

A continuación se procede a crear la variable x, que contiene los *'plots'* de las películas que servirá de entrada para la predicción. Para esto se procedió a hacer la preparación de los datos, en primer lugar se procedió a dejar todas las entradas en minúscula, luego se eliminaron las *'stopwords'*, finalmente se hizo un proceso de *'stemming'*[1].

Luego de desarrollado este proceso, se creó el vocabulario con el que se hará la preddición de la probabilidad de que una película pertenezca a uno u otro(s) género(s). Teniendo en cuenta que el percentil 97,5 % de las palabras en cada una de los *'plots'* , es de 1202,65, se decide trabajar con 1202 como máxima longitud de cada entrada. Finalmente se procede a crear el vocabulario con las primeras 1202 entradas de cada una de las observaciones y a crear el *'pad'*.

2. Modelos usados

Para abordar el problema se probaron varios modelos, entre los que se encuentran: XGboost, Redes Neuronales, Regresión Logística y un ensamble de modelos que se realizó posteriormente. En esta sección se hará un barrido rápido de las principales características de cada modelo.

2.1. XGBoost [1]

El metódo de *'gradient boosting'*, es una generalización del *'AdaBoosting'*, mejorando el desempeño del modelo, mediante la incorporación de ideas de *'boostrap'*, como el muestreo aleatorio para la calibración del modelo. Incluye así, tres elementos: una función de pérdida a ser optimizada, un aprendiz débil para hacer predicciones y un modelo aditivo para agregar aprendices débiles para minimizar la función de pérdida. [2]

El XBoost es una forma optimizada de *'gradient boosting'*, en donde lo que se busca es con cada árbol de decisión estimado buscar los parámetros débiles y desecharlos, para de esta manera obtener cada vez mejores resultados.

[1] El proceso por el cual se remueven los afijos de las palabras.

2.2. Redes Neuronales

Las redes neuronales son un sistema de aprendizaje computacional que usa para transformar datos de entrada en una salida de predicción para el desarrollo de un modelo. Se basa en el concepto biológico de neurona, en donde aprende dependiendo de las interconexiones que tiene entre sí. Funciona dependiendo de los datos que se le hayan suministrado para el aprendizaje, es de esta manera que la red luego de procesar muchos datos, puede empezar a predecir el siguiente. [3]

2.2.1. Redes Simples con Neuronas Sigmoides

Las función básica de activación con la que trabaja una red neuronal es la lineal, sin embargo, se pueden usar funciones no lineales para darle felixibildiad al modelo. Es por esto que se usan funciones como la tangencial hiperbólica y la sigmoide para la activación de la misma. En este caso particular se usará una red neuronal con una función sigmoide de activación. Pero ¿qué ventajas tiene hacer uso de esta?

La principal razón para usar una función sigmoide es que esta existe entre (0 , 1), es por esto que por ejemplo en este caso es de utilidad debido a que lo que se necesita es predecir probabilidades. Además con este tipo de funciones de activación pequeños cambios en la data de entrada no genera grandes cambios en la data de salida. [4]

2.2.2. Redes Convolucionales

Las redes convolucionales tienen como objetivo mirar ¿cómo dos funciones se sobreponen la una a la otra? De esta manera aunque su principal aplicación es al procesamiento de imágenes, también se pueden usar para el procesamiento de lenguaje. Esto se debe a que se los datos de entrada en este caso son matrices que representan oraciones y palabras. Sin embargo, no es recomendable para hacer esto debido a que está pensado en tres dimensiones. Por lo que es mejor idea usar redes neuronales recurrentes. [5]

2.2.3. Redes Recurrentes

Son redes neuronales que se alimentan de ellas mismas una cantidad N de veces, es decir cada vez que se pasa el algoritmo, se vuelve a pasar por allí de manera que reaprende de sí misma. Además permite conexiones arbitrarias entre las neuronas, que crea ciclos de temporalidad.

2.3. Regresión Logística

La regresión logística simple fue desarrollada por David Cox en 1958, y permite mediante el uso de datos cuantitativos predecir datos cualitativos o etiquetas como en este caso con los géneros de las películas. Es de esta manera que este modelo predice la probabilidad de que se pertenezca a una u otra clase. Por lo que dejará la probabilidad de que basado en la matriz de entrada, pertenezca a cada una de las clases.

2.4. Modelo Ensamblado

No obstante todos los modelos descritos anteriormente, la mejor solución puede llegar a hacer una combinación de estos, por lo que se ha decidido, ha hacer un ensamble de tres modelos, regresión logística, *'XGBoost'* y un *'random forest'*.

Una vez descritos los modelos que se tuvieron en cuenta para el desarrollo de la predicción de la probabilidad, se procedió a hacerlos en *Python*, mediante el uso de la herramienta *Collab* de *Google*.

3. Evaluación de los modelos

Las bases de datos de entrenamiento y prueba con las que se desarrolló este proyecto se encuentran disponibles en el GitHub de la clase en las siguientes rutas:

- Entrenamiento: shorturl.at/gAHMO
- Prueba: shorturl.at/btRTZ

Para la preparación de los datos descrita en la primera parte de este documento se utilizaron las siguientes librerías y funciones:

1. Sklearn:
    - TFidfVectorizer
    - MultiLaberBinarizer
    - CountVectorizer
2. NLKT
    - StopWords

- word tokenize
- SnowballStemmer
- WordNetLemmatizer

Para el desarrollo de los modelos se utilizaron las siguientes librerías y funciones:

1. Sklearn
    - LinearSCV
    - LogisticRegression
    - LogisticRegressionCV
    - Pipeline
    - OneVsRestClassifier
    - RandomForestClassifier
2. Keras
    - BatchNormalization
    - Sequential
    - Bidirectional
3. XGboost
    - XGBClassifier

Una vez corridos los modelos se obtuvieron los siguientes resultados:
Este resultado se obtuvo con un algoritmo de OneVsRestClassifier de SkLearn.

Modelo	Acurracy
XGBoost	0.6910
NeuralNet	0.8797
LogisticRegression V1	0.9006
LogisticRegression V2	0.9007
LogisticRegression V3	0.8972
LogisticRegression Pipeline	0.8878
LogisticRegression CV	0.8809
Ensamble	0.8947

Cuadro 2: Acurracy

Que es una técnica ampliamente usada, por su eficacia computacional. Permite escoger el método que se le pondrá para hacer el trabajo de manera

que en este caso se uso esta funcionalidad de SkLearn, combinada con una regresión logística, dando de esta manera, el mejor modelo a usar.

Para obtener mayor detalle del modelo se deja el link de GitHub del proyecto:
https://github.com/sergiomora03/NLP-Natural-Language-Processing/tree/master/Project

Además si se quiere hacer uso del modelo aquí propuesto se pone a disposición la siguiente API de AWS:
http://ec2-18-218-105-72.us-east-2.compute.amazonaws.com:5000

4. Conclusiones

Luego de hacer un recorrido por los diferentes modelos que se propusieron para este proyecto, se puede ver que la mejor opción fue una regresión logística del paquete SkLearn, en donde se hace una validación con la técnica de OneVsRest.

Esto fenómeno se observó gracias a que por definición la regresión logística es una metodología para la estimación de probabilidades, que es el objetivo primordial del proyecto, estimar la probabilidad que de acuerdo al resumen una película pertenezca a uno u otro género.

Referencias

[1] xgboost developers, *XGBoost Parameters*, 2020.

[2] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent in function space," 06 1999.

[3] DeepAI, "Neural network."

[4] N. Kumar, "Sigmoid neuron building block of deep neural networks," 2019.

[5] D. Britz, "Understanding convolutional neural networks for nlp," 2015.