Trabajo Práctico 2 - Críticas Cinematográficas

Integrantes:  
Ignacio Latorre - Padrón: 101305

Nicolas Ronchese - Padrón: 108169

Gastón Avila - Padrón: 104482

**Introducción**

El trabajo consistió en procesar una colección de críticas cinematográficas y entrenar modelos de aprendizaje con estos datos para determinar si una crítica es positiva o negativa.

**Desarrollo**

Bayes Naïve

Resultado de kaggle: bn3.csv -> Score: **0.73018**

Modelo: bayes\_naive - el otro modelo de bayes es utilizado en una notebook-

Para la implementación de Bayes Naïve primero identificamos que vectorizador y modelo dentro de sklearn nos daba mejores resultados que fue el TfidfVectorizer con BernoulliNB.

Luego realizamos aplicamos GridSearchCV para obtener los hiperparametros que mejor se ajustaran al modelo.Las métricas se encuentran dentro de la notebook.

Random Forest

Resultado de Kaggle: rf4.csv -> Score: **0,69877**

Modelo: RF\_model

En el modelo de Random Forest se probó procesar el texto con el CountVectorizer y también con el TfifdVectorizer. Se ejecutó el mismo RandomSearch para ambos, variando los hiperparámetros lo más posible.

El modelo que mejores resultados dio fue el procesado con el CountVectorizer. Los hiperparametros obtenidos fueron: {bootstrap=False, max\_features='auto',

min\_samples\_split=5, n\_estimators=72}

Tanto la precisión, el recall, el accuracy y el f1 score dieron **0,84** en el test interno.

XGBoost

Resultado de Kaggle: xgboost.csv -> Score: **0.70556**

Modelo: XGB\_model

En el modelo de XGBoost se procesó el texto con CountVectorizer. Se utilizó RandomSearch para buscar hiperparámetros. Sin embargo, los mejores resultados se obtenían utilizando los hiperparámetros por default, por lo que se focalizó el RandomSearch en la búsqueda de la cantidad de estimadores.

El modelo que mejores resultados dió fue con n\_estimators=80 con un recall, precisión, accuracy y f1 score de **0,84** en el test interno.

Redes

Resultado de kaggle: redes\_predictions7.csv -> Score: 0.71428

Modelo: redes\_tp2

Para la implementación de redes decidimos crear archivos a partir de roBERTuito, en estos se guarda la predicción que hace dicho sistema sobre el porcentaje de positividad, neutralidad y negatividad en un cierto largo de los textos. Creamos diversos archivos variando el largo que tomamos pero al final decidimos quedarnos con 500 y 350 porque nos dieron los mejores resultados.

Con estos archivos creamos un nuevo dataset con 9 columnas:

* Las cuatro primeras contienen los valores del train/test 500
* len y len% contiene el largo de los textos a analizar y el porcentaje que ya ha sido analizado
* y POS/NEU/MEG evolución simbolizan la evolución de las predicciones desde las 350 hasta las 500 líneas

En cuanto a la arquitectura nuestro modelo utiliza dropout para evitar el overfitting y mientras que el restos de la red fue construida a partir de los intentos que mejor resultado nos dieron en el tp1. Las métricas se encuentran dentro de la notebook.

Aclaración:

En el mismo archivo donde se crearon los archivos se había usado para crear predicciones sus resultado son los primeros 4 redes\_prediction.csv sin número esto es porque me olvide de agregarle un número cuando pase al siguiente modelo. Luego de este error cada modelo tendrá su redes\_prediction.csv representando un cambio en el modelo.

Ensamble

Resultado de Kaggle: voting\_soft\_tfifd.csv -> Score: **0,76526**

Modelo: Voting\_model.pickle

A la hora de realizar el ensamble se optó por realizar un modelo de Voting. En este caso, se utilizaron los modelos de Bayes Naive y de Random Forest generados para este TP sumados a un modelo de Regresión Logística, al cual se le encontró los hiperparametros mediante un GridSearch y se obtuvo un puntaje de **0,89** en test interno.

Se probó procesando tanto con el TfifdVectorizer y con el CountVectorizer, y también cambiando la modalidad entre ‘soft’ y ‘hard’.

El mejor resultado se obtuvo con el TfifdVectorizer y utilizando la modalidad ‘soft’, los puntajes de precisión, recall, accuracy y f1 score en test interno son **0,87.**

**Conclusión**

El mejor resultado, se obtuvo con ensamble de modelos de Voting, con un puntaje de 0.76526 en Kaggle. Este enfoque combinó los modelos de Bayes Naïve, Random Forest y Regresión Logística, utilizando el TfifdVectorizer y la modalidad 'soft'.