**Entrega Proyecto Final – Clasificación Géneros de Películas**

Jaime Andrés Unriza Vargas

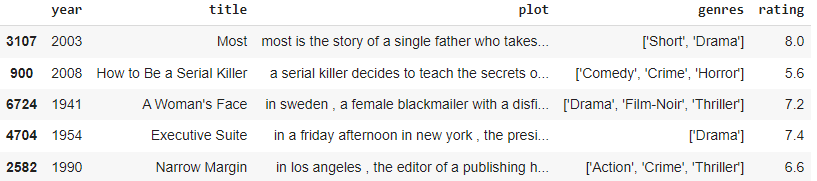
Roberto Bustamante

Iván Amarillo Lozada

En este proyecto, el objetivo principal era mediante el uso de una base de datos de películas establecer (haciendo uso de sus parámetros de calibración) un modelo de predicción que fuera capaz de asignar las películas, acorde con el género al cual pertenecen. Cada observación contaba con el título de la película, su año del lanzamiento, la sinopsis de la película y los géneros a los que esta película pertenece (cada película puede pertenecer a uno o más géneros).

Para poder llevar a cabo las etapas de entrenamiento, testing y validación sobre el algoritmo entrenado, contamos con dos bases de datos diferentes una de entrenamiento y una disponible para poder someter el algoritmo a prueba y así determinar su capacidad de clasificación sobre datos que no hubiese visto antes.

Para iniciar este ejercicio, fue necesario entender el contenido de cada una de las bases de datos disponibles, siendo la base de datos de entrenamiento aquella que contaba con los géneros y así poder indicar o entrenar al modelo de acuerdo al género al que realmente pertenecía cada película y la base de testing sin esta información acerca de los géneros sobre la cual debíamos someter el algoritmo para poder identificar su capacidad de clasificación. A continuación, mostramos la estructura de cada base empleada.



dataTraining.head()

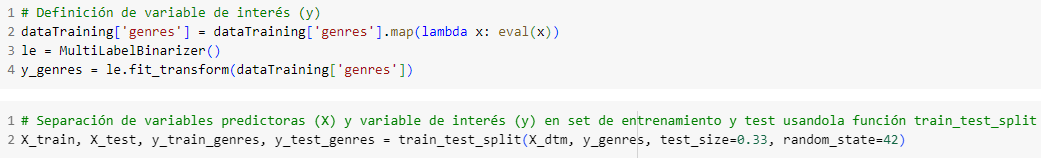
A screenshot of a computer

Description automatically generated

dataTesting.head()

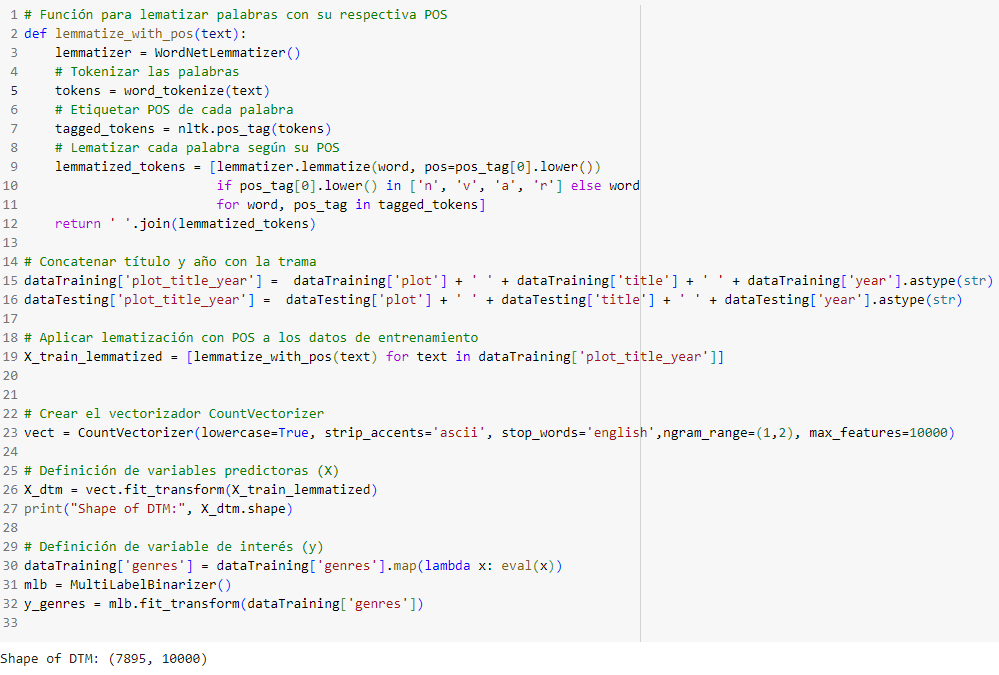
**Preprocesamiento de Datos:**

Para poder dar inicio al entrenamiento de los diferentes modelos y su comparación era necesario hacer un procesamiento de los datos, básicamente llevar los datos a un estado estandarizado que pudiera ser empleado en todos los modelos a entrenar por lo que utilizamos el fit\_transform() sobre los géneros, así como dividir la data en fragmentos para entrenamiento y otros para testing, esto con el fin de poder realizar entonces estos análisis y estos entrenamientos de forma estandarizada. El código empleado se muestra a continuación.



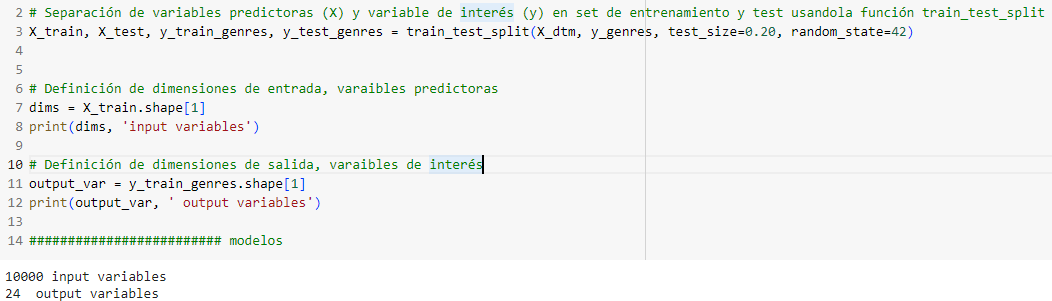
Estandarización y separación de la data en bases de entrenamiento y testing

Asimismo, como parte del ejercicio de preprocesamiento de los datos empleamos una función para lematizar las palabras con su respectiva posición y concatenar el título de año con la trama y poder cruzar así una matriz X lematizada durante el proceso de entrenamiento. El código se expone a continuación.



Función y proceso de lematización de la matriz X.

Este proceso nos permite ya tener la data lista para empezar a entrenar los modelos. La separación definitiva de la data se presenta en el siguiente código:



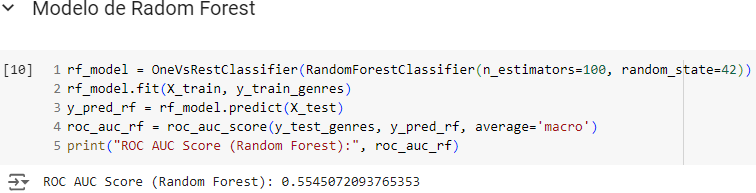
Separación en fragmentos de train y test lematizados.

**Modelos Entrenados, Comparación y Selección:**

Para poder identificar el mejor modelo se entrena son 3 diferentes con el fin de compararlos basándonos en el ROC AUC Score y así poder determinar aquel que tuviera el mejor nivel predictivo en torno al género de cada película en la base de datos.

***Modelo de Random Forest:***

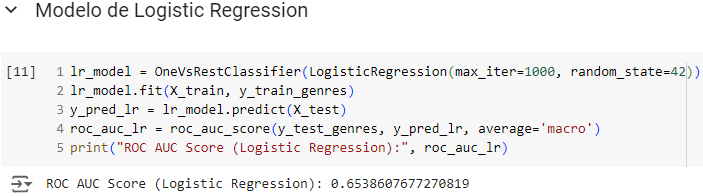
Nuestro primer modelo escogido es un modelo de random forest entrenado empleando la clase OneVsRestClassifier(), tomando 100 estimadores. Al entrenarlo encontramos un ROC AUC Score de 0.55 como se muestra en la imagen a continuación.



Modelo de Random Forest.

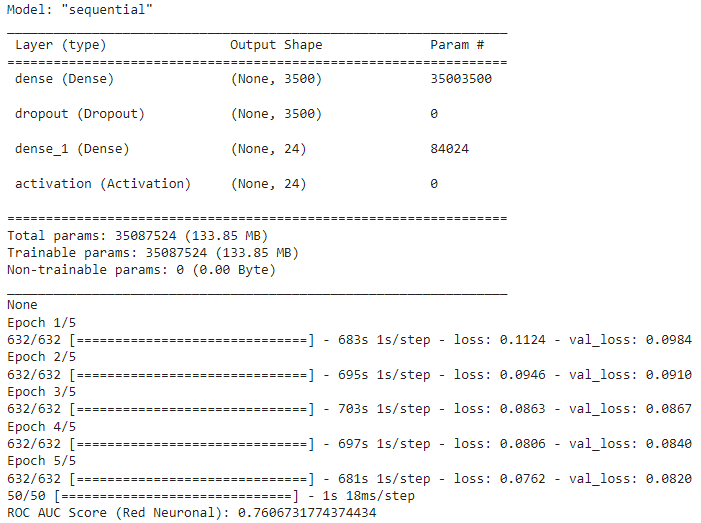
***Modelo de Logistic Regression:***

Posteriormente, entrenamos un modelo de regresión logística tomando como un máximo de 1.000 iteraciones y resultando en un ROC AUC Score de 0.65, resultando en un mejor Score que el previamente evidenciado mediante el entrenamiento y testing del modelo de Random Forest

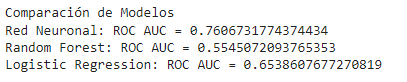


***Modelo de Red Neuronal:***

Por último, entrenamos un modelo de red neuronal secuencial resultando en un ROC AUC Score de 0.76, este resultado acompañado de los parámetros de densidad y dropout se exponen a continuación.



La comparación de los resultado arroja un mejor modelo, siendo este la red neuronal, razón por la cual escogimos el mismo para poder someterlo a competencia en la plataforma de Kaggle.



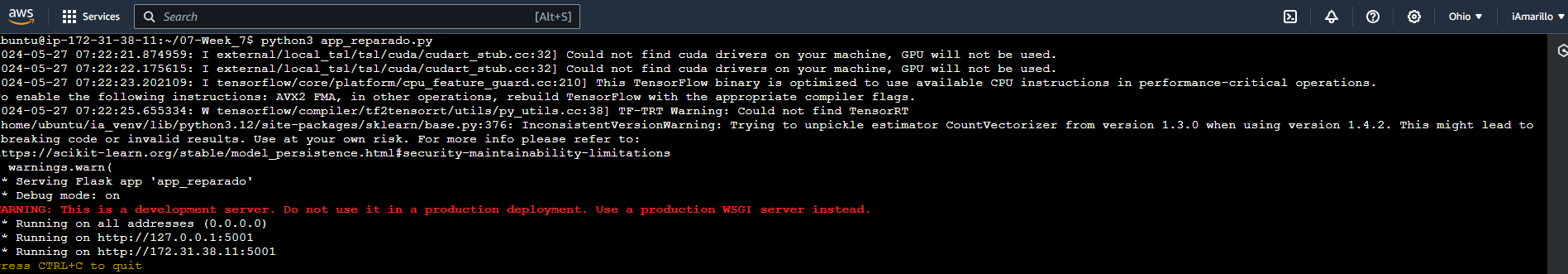
A screenshot of a computer

Description automatically generated

Fragmento en Jupyter del modelo escogido y justificación.

**Disponibilización del modelo en API:**

Al intentar desplegar el microservicio, dado el tamaño del modelo seleccionado, el cual se carga empleando la lectura de ‘model.h5’, no logra cargar la página (se pasa el tiempo de espera y retorna un error por tiempo de ejecución). Sin embargo a continuación, exponemos el código y el runtime en AWS para garantizar que está funcionando, al parecer Chrome retorna error por tiempo y por ello no se logra desplegar en producción.



**A screenshot of a computer program

Description automatically generated**

**Conclusiones:**

A manera de conclusión y a pesar de seleccionar el modelo con el mejor indicador dentro de los tres modelos entrenados, no logramos una posición alta en la competencia, sugiriendo que nuestro modelo en relación a los otros grupos no predice de forma adecuada. Se necesita calibrar parámetros y probablemente incluir en el análisis otros algoritmos más avanzados al igual que probar con otras aproximaciones de clasificación para conseguir mejores resultados.

Desde el trabajo en grupo hemos identificado que debemos reforzar el proceso de calibración de los modelos, que escoger modelos complejos como una red neuronal por si sola, no garantiza un alto poder predictivo y de clasificación. Trabajaremos en ello para robustecer nuestro criterio de calibración a futuros modelos a desarrollar.

A la fecha de elaboración del presente documento, el top 10 de grupos dentro de la competencia superan scores sobre 0.89.

