Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Escuela de Ingeniería y Ciencias

TC3006C.101

Inteligencia artificial avanzada para la ciencia de datos I (Gpo 101)

Alumno:

Eric Manuel Navarro Martínez - A01746219

Profesor:

Jorge Adolfo Ramírez Uresti

Fecha de entrega:

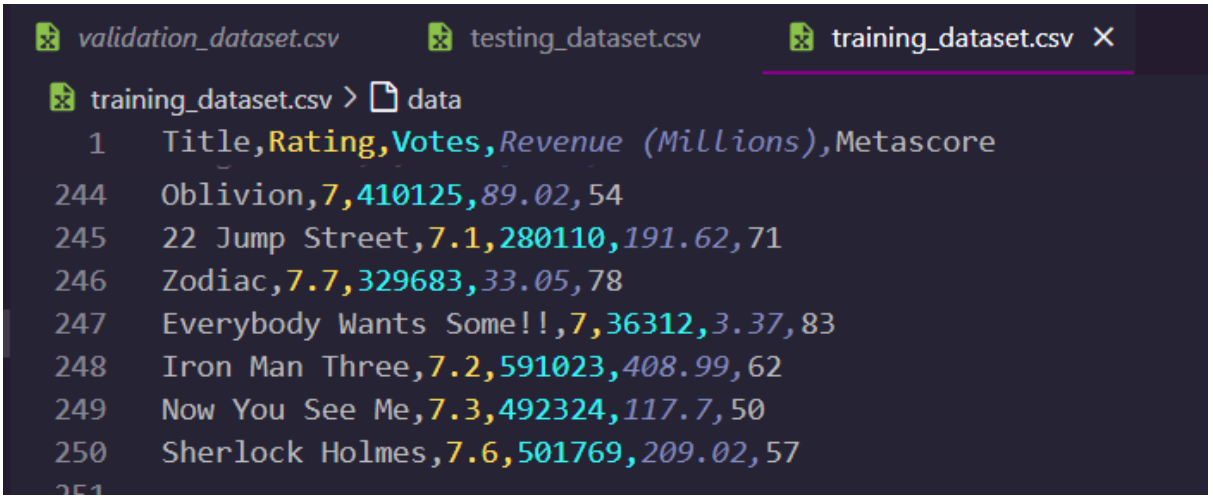
11/09/2024

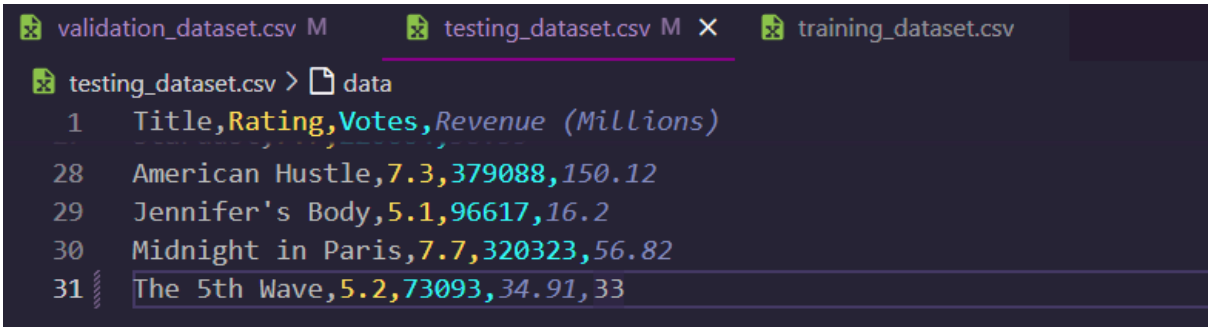
- Escoge una de las 2 implementaciones que tengas y genera un análisis sobre su desempeño en un set de datos. Este análisis lo deberás documentar en un reporte con indicadores claros y gráficas comparativas que respalden tu análisis.

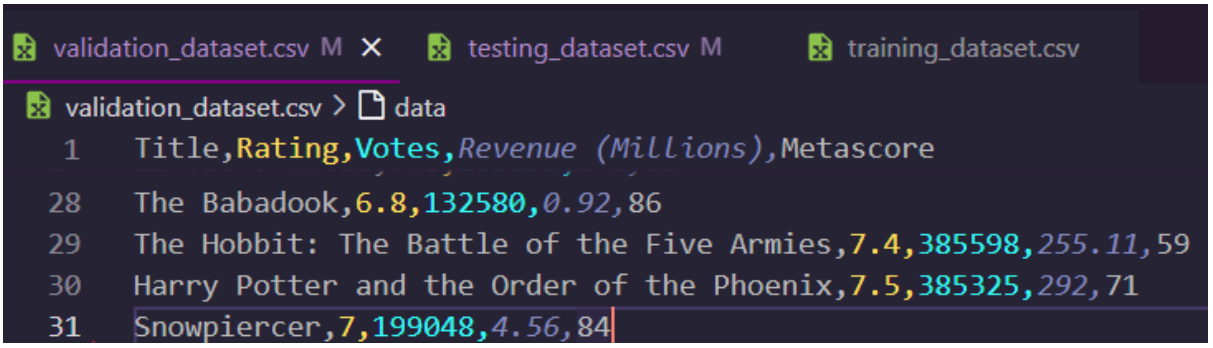
- El análisis debe de contener los siguientes elementos:

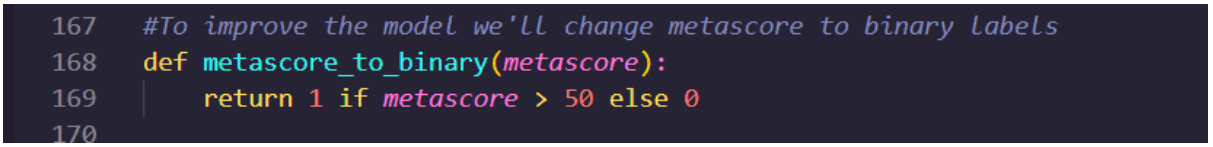
    - Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation).

  Comenzando por los datasets elegí uno que tuviera suficientes datos para hacer un buen entrenamiento del modelo; con esto en mente opté por usar un dataset que nos había provisto otro profesor el cual era de más de 1000 datos de películas de IMDB (Internet Movie Data Base).
  Seguí el pipeline de limpieza de manera manual, con el data wrangling, como había decidido desde un principio hacer regresión logística opte por no sobresaturar el modelo con información categórica; ya que en el dataset original estaban columnas con información de la trama, cast y directores.
  Una vez seleccioné los features con los que iba a entrenar empecé con el data cleansing, en este caso ignoré columnas que no tuvieran datos completos, una vez tuve mi dataset completo me aseguré que tuviera 250 datos para el training, 30 para testing y 30 para validation.
  Por último realicé un data preparation para transformar los datos del target feature en datos binarios, con una función simple siendo 1 si es mayor o igual a 51 y 0 en caso de no serlo, esto permite que la regresión logística entregue los datos que busco predecir, el metascore.







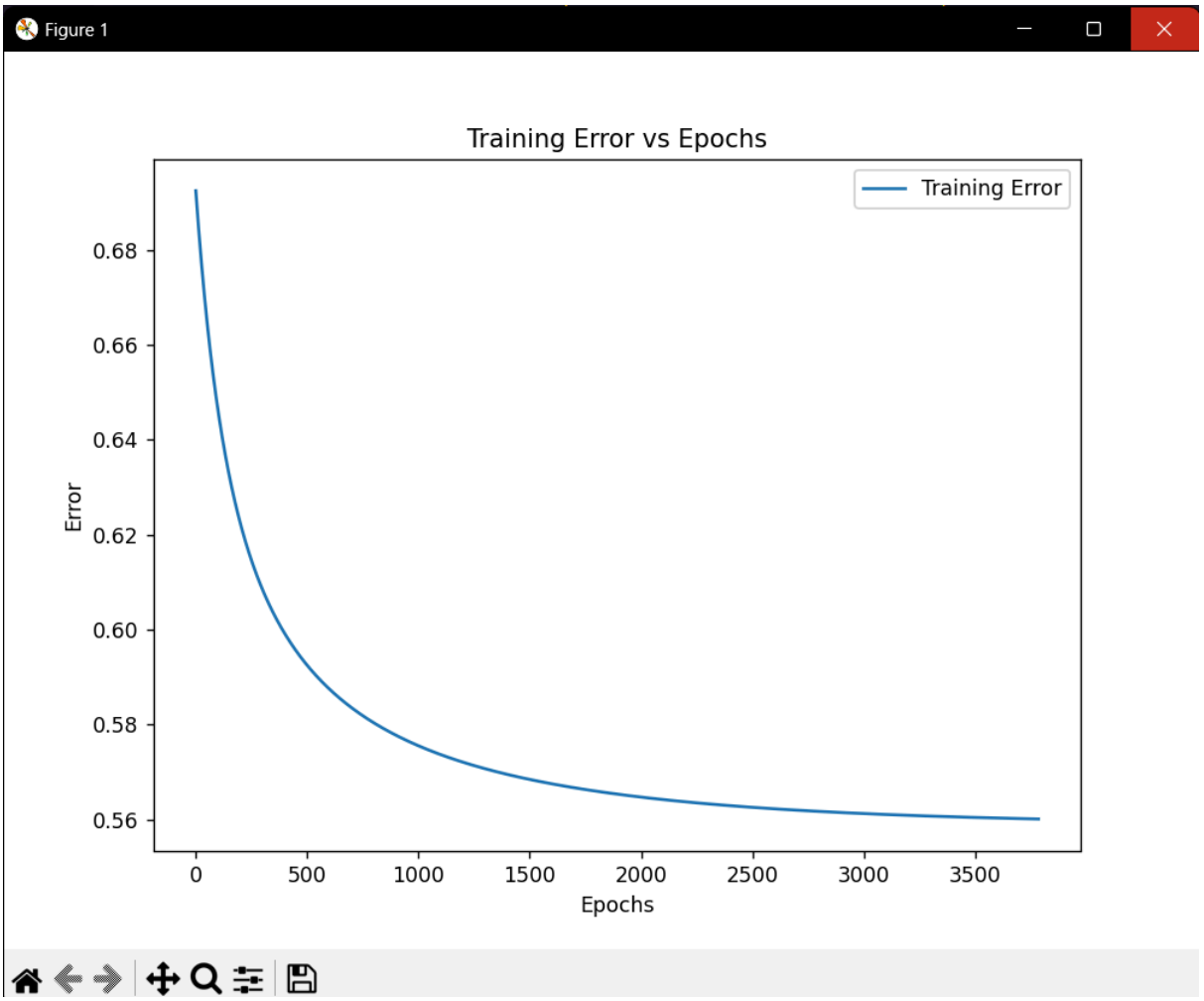


Para la realización de los diagnósticos del código opté por calcular la matriz de errores al final del entrenamiento, con esto obtengo los valores para saber distinta métricas de precisión, exactitud, especificidad, sensibilidad y F1 score.
La matriz de confusión calcula lo siguiente:
R^2: 0.1545 esto fue en un principio preocupante ya que asume un posible underfitting.
True Positives: El modelo predijo correctamente 15 películas como buenas.
True Negatives: El modelo predijo correctamente 10 películas como malas.
False Positives: El modelo solo erró una película mala como buena.
False Negatives: El modelo predijo incorrectamente 5 películas buenas como malas.

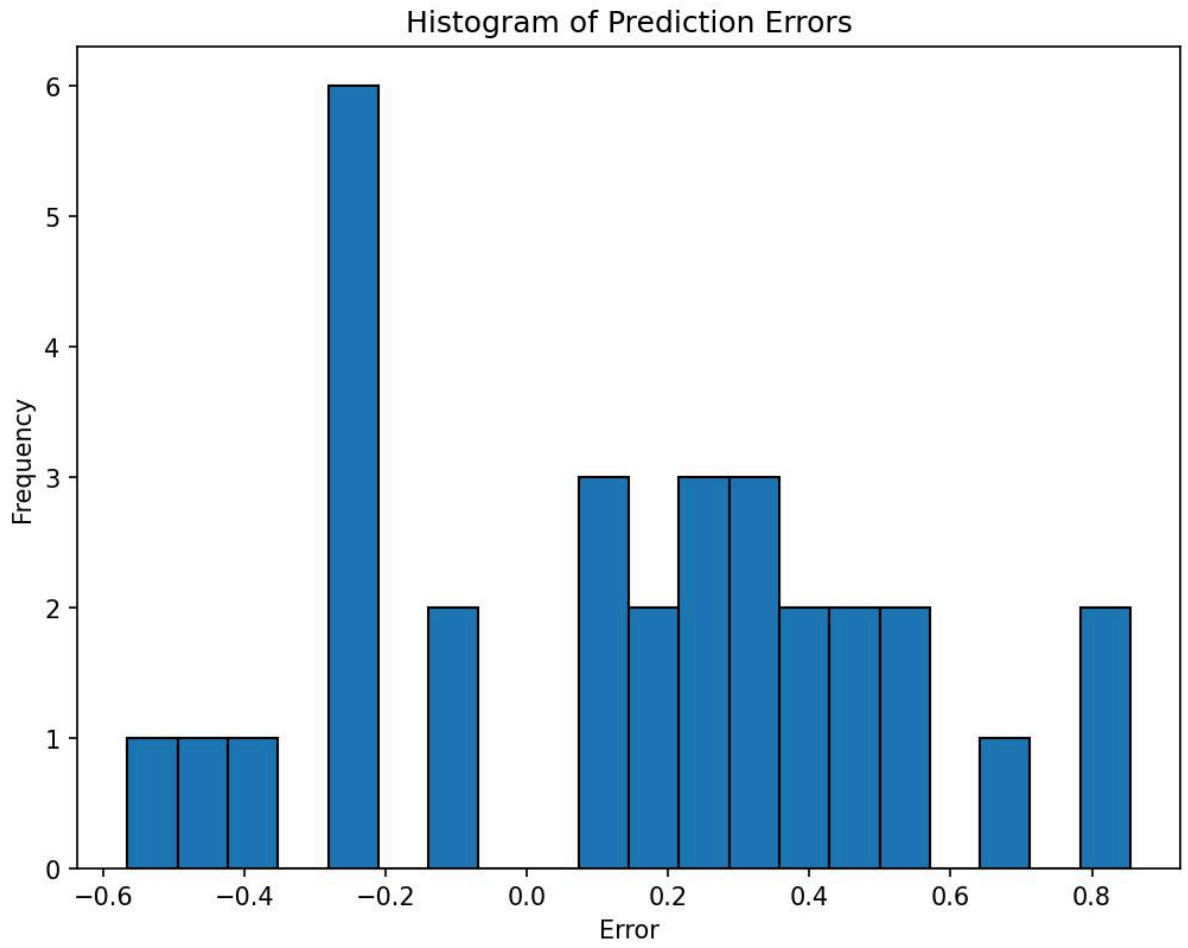
Las métricas implican que la r^2 que previamente asumimos como un valor negativo, puede que simplemente refleje información distinta dentro del dataset como lo es baja varianza, siguiendo con esto se calcularon las siguientes métricas

Accuracy: La precisión global del modelo es del 80.6%, prediciendo correctamente de forma constante.
Specificity: La capacidad para predecir correctamente las películas malas es del 90.9%.
Precisión: La precisión del modelo es del 93.75%, haciendo que clasifique películas buenas de manera correcta constantemente.
Recall: El modelo predice correctamente el 75% de las películas buenas.
F1 Score: Fue de 0.833 mostrando buen equilibrio entre precisión y especificidad.

Para complementar los valores de la matriz de errores hice posteriormente graficos que pudieran mostrar de manera más visual el desempeño del modelo



Con esta gráfica podemos ver como el error se minimiza conforme aumentan las épocas, siendo alrededor de la 3784 que se detiene el entrenamiento al considerarse lo suficientemente libre de error.
Igualmente graficamos un histograma con los errores.



Estos errores son en números sumamente pequeños cercanos al 0, lo cual nos ayuda a entender en cuantos casos y de qué forma los casos fueron ya sea subestimados o sobreestimados.
Esto es debido a que este gráfico explica la cantidad de porcentaje de error que hubo y como podemos apreciar el modelo tiene mayor problema para identificar valores positivos que negativos, los positivos llegando a más de 0.8 de margen de error lo cual aunque siga siendo un número pequeño es algo a notar para saber donde están las áreas de mejora en el modelo

Con los gráficos podemos llegar a las siguientes conclusiones:
En cuanto el bias o sesgo, podemos observar que hay un sesgo alto, ya que las puntuaciones de r^2 son muy bajas aunque tenga un accuracy relativamente bueno, de esto hablaremos más adelante.
Siguiente la varianza es baja, ya que como vimos anteriormente en el histograma hay un bajo rango de error y con las métricas de la matriz de confusión no vemos evidencia de algún overfitting.

Por último este modelo aunque es la base del mismo de la primer entrega le realicé algunas modificaciones debido a que en la segunda vista noté donde podía mejorar,

ya que hacía un overfitting al realizar un segundo gradient descent con los datos de validación. Y con las nuevas gráficas que muestran mi desempeño del modelo he concluido que aunque los parámetros son buenos, puede que el data wrangling que haya hecho en un principio no haya sido el más efectivo y sería muy bueno agregar más features para ayudar al modelo a mejorar su underfitting ya que tiene un bias aún muy alto.

Para realizar mejoras opté primero por decrementar el alfa, para que realice más épocas en entrenamiento, lo cual resulta en un desempeño más tardado a la hora de entrenar, pero al momento de hacer la validación podemos observar una r^2 más alta, al igual que mejores valores para specificity y precisión aunque los números siguen sin ser lo suficientemente adecuados ya que pueden representar un overfitting.

```
-----------------------------------
R^2 test and prediction: 0.4363636363636362
Confusion Matrix: {'True Positives: ': 16, 'True Negatives: ': 11, 'False Positives:': 0, 'False Negatives: ': 4, 'Accuracy': 0.8709677419354839, '
Specifity': 1.0, 'Precision': 1.0, 'F1 Score': 0.888888888888889, 'Recall: ': 0.8}
.................................
 Finished comparing
```

alfa = 0.001

```
 ---------------------------------------
 R^2 test and prediction: 0.15454545454545432
 Confusion Matrix: {'True Positives: ': 15, 'True Negatives: ': 10, 'False Positives:': 1, 'False Negatives: ': 5, '
 Accuracy': 0.8064516129032258, 'Specifity': 0.9090909090909091, 'Precision': 0.9375, 'F1 Score': 0.8333333333333334
 , 'Recall: ': 0.75}
 .....................................
  Finished comparing
```

alfa = 0.01

Posteriormente cambié el tipo de normalización para usar una con min max en lugar de un z-score en conjunto con el aumento de alfa, que empieza a tener el efecto opuesto en nuestro comparativo, teniendo peor desempeño que con Z-score, el modelo incluso pareciera que opta por marcar todo como una buena película.

```
-----------------------------------
R^2 test and prediction: -0.5500000000000005
Confusion Matrix: {'True Positives: ': 20, 'True Negatives: ': 0, 'False Positives:': 11, 'False Negatives: ': 0, 'Accuracy': 0.6451612903225806, '
.................................
 Finished comparing
```

Con esto en mente para poder mejorar el desempeño del modelo use nuevamente el normalización por medio de z-score y usar data augmentation para agregarle una ligera cantidad de ruido al modelo, obteniendo mejoras en r^2 y accuracy a comparación del primer método que sólo decrementó alfa

```
-----------------------------------
R^2 test and prediction: 0.4363636363636362
Confusion Matrix: {'True Positives: ': 16, 'True Negatives: ': 11, 'False Positives:': 0, 'False Negatives: ': 4, '
Accuracy': 0.8709677419354839, 'Specifity': 1.0, 'Precision': 1.0, 'F1 Score': 0.888888888888889, 'Recall: ': 0.8}
.................................
 Finished comparing
```