**ANEXO 1.**

Búsqueda de hiperparámetros de los modelos.

1. **Modelo líneal.**

Para este modelo se hizo la primera corrida teniendo en cuenta todo el dataset, y estos fueron los resultados. Después se hizo una exploración de los datos más representativos del dataset, que ayudaran a predecir la variable objetivo, por esta razón se tuvo en cuenta el mapa de correlaciones siguiente.

Chart

Description automatically generated with low confidence

En el anterior se puede ver claramente que la variable (aparte del “total amount”) es el de “trip distance”. Por esta razón se trató de ver qué tan importante era esta columna para inferir la variable objetivo.

|  |  |
| --- | --- |
| Con todas las columnas | Sólo con la columna Trip distance |
|  |  |

Conjunto de entrenamiento con todas las columnas

Chart, line chart

Description automatically generated

Conjunto de entrenamiento con sólo la columna “trip distance”

Chart, line chart

Description automatically generated

**Conclusión de variación de hiperparamentros con el modelo lineal:**

Como se pudo observar, el modelo arrojó los mismos valores ingresando todo el conjunto de variables como sólo con la columna “trip distance” por eso podemos decir que esta es la única variable que es representativa en todo el dataset para predecir la variable objetivo.

1. **Modelo polinomial.**

Para este modelo se hizo un modelo de grado 2, pues como se explica en el trabajo, con nuestra cantidad de datos 546.805 la RAM del colab no es suficiente para correr las representaciones las matrices de grado 3 o 4, entonces este hiperparametro del grado lo no fue posible variarlo.

Como se hizo con el modelo lineal, se dejó sólo la columna de “trip distance” para ver qué tan significativa era, ahora con el modelo polinomial

|  |  |
| --- | --- |
| Con todas las columnas | Con sólo 1 columna |
|  |  |

Con todas las columnas

Graphical user interface, chart, line chart

Description automatically generated

Con sólo 1 columna

Chart, line chart

Description automatically generated

En este caso dio el mismo resultado que en el modelo lineal, el resultado fue igual con todas las variables y con la variable más representativa.

**Conclusión de variación de hiperparamentros con el modelo polinomial:**

* No fue posible aumentar el grado del polinomio por la infraestructura
* El modelo polinomial, se comportó igual que el modelo lineal frente a el cambio de variables predictoras.
* No se intentó utilizar hiperparamentros de regularización, pues los modelos no presentan sobreajuste.

1. **Modelo MLP.**

Para este caso se presenta un modelo de percetron multicapa, un modelo secuencial con datos escalados de entrada.

La capa de entrada es de 13 neuronas, de acuerdo con el número de variables que tenemos

La salida es una capa de 1 neurona, pues estamos trabajando con un modelo de regresión

La función de pérdida es una MSE, Mean Squared Error para que el modelo corrija con el método de retropropagación del error.

Graphical user interface, text

Description automatically generated

A continuación se muestra los diferentes resultados con los diferentes hiperparamentros.

|  |  |
| --- | --- |
| * Batch size de 300 * 15 épocas * La función de pérdida es MSE * RMSprop 0.001 |  |
| * Batch size de 300 * 15 épocas * La función de pérdida es MSE * Adam 0.001 |  |
| Se observa un cambio en el error cuadrático, se sigue con el optimizador RMSprop     * Batch size de 300 * 15 épocas * La función de pérdida es MSE |  |
| Se observa que los errores disminuyen con menos capas ocultas     * Batch size de 300 * 15 épocas * La función de pérdida es MSE |  |
| Se observa que el número de capas ocultas óptimo es 2.    10 épocas |  |
|  |  |

En amarillo se presenta el modelo ganador que minimiza el MAE,

MLP

* Batch size de 300
* 15 épocas
* La función de pérdida es MSE
* Adam 0.001