Facultad de
Ciencias Exactas,
Ingeniería y Agrimensura

Trabajo Practico Final

Aprendizaje Automático 1

Alumno:

- Garcia, Timoteo

Objetivo del trabajo

Poner a prueba conocimientos adquiridos a lo largo de la materia en términos de Análisis, limpieza y preprocesamiento de datos, para luego implementar modelos de Machine Learning con las librerias Scikit-Learn y Tensorflow (Redes Neuronales), con su posterior análisis de métricas y evaluación de dichos modelos.

Resolución:

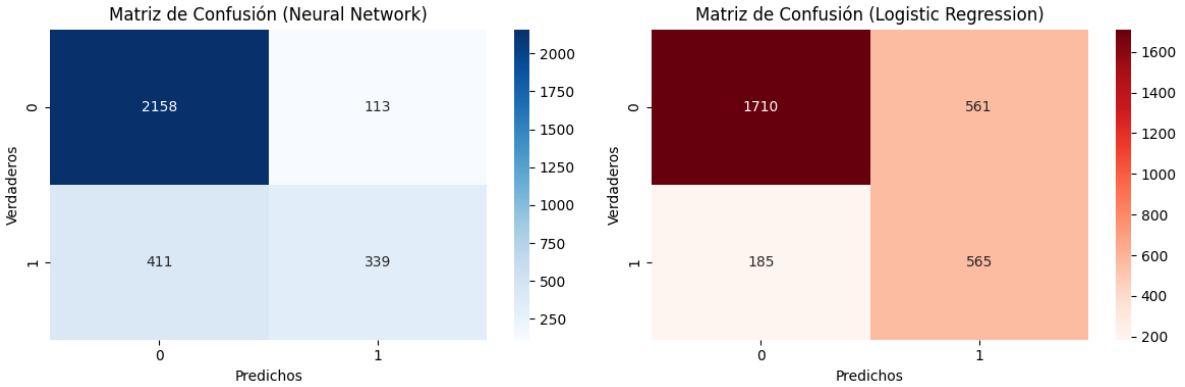
Para la resolución de este trabajo lo separe en 3, el Análisis exploratorio de datos, la predicción de la variable “RainfallTomorrow” mediante regresión Lineal, y la predicción de la variable “RainTomorrow” mediante clasificación. Cada una de las 3 las resolví en notebooks diferentes, donde analice por separado los modelos, elegí el ‘mejor’ para mi, analice su explicabilidad con SHAP, y lo exporte mediante JobLib para poder usarlo en MLOps en Streamlit.

- **EDA (Análisis Exploratorio de Datos):**
    - Para el EDA utilice librerías de python como: Pandas, Numpy, Seaborn y Matplotlib.
    - Realice un análisis general entre variables, detección de valores nulos, para su posterior imputación tratando de entrenar los modelos con la mayor cantidad

de datos posible.

- Para la inputación separe en train y test para no caer en la *Fuga de Datos.*
- Análisis de correlación entre variables.
- Visualización de datos.

- **Regresión Lineal:**
  - En esta etapa, entrene y evalue 5 modelos. Regresión Lineal, con regularización Lasso, con regularización Ridge, Elastic Net, una Red Neuronal con Tensorflow con 7 capas.
  - En todas las instancias, a excepción de la Red Neuronal (NN), empleé Pipelines para la construcción del modelo. En el primer paso de estos Pipelines, los datos fueron sometidos a un proceso de preprocesamiento a través de un Column Transformer. Este componente se encargó de realizar la transformación de las variables categóricas mediante la creación de variables dummy, mientras que las variables numéricas fueron estandarizadas mediante el uso de StandardScaler.
  - Para la NN, lo que hice fue lo mismo pero no dentro de un Pipeline.
  - Termine eligiendo como mejor modelo a la red Neuronal, ya que generalizaba mejor, teniendo un R2 mas alto, y tenia un error cuadratico medio (MSE) mas bajo que los demas modelos.
- **Clasificación:**
  - En este caso probé con 2 modelos, una regresión Logística, y una red neuronal de 5 capas.
  - Similar a las regresiones, arme un Pipeline para la regresión Logistica con un Preprocesador para las columnas, y simil para la Red Neuronal, pero no en un Pipeline
  - Termine eligiendo la Red Neuronal tambien para este caso, ya que me clasificaba mejor si iba o no a llover mañana. Adjunto las matrices de confusión.



Tome esta decisión ya que quiero que mi algoritmo me ayude a predecir ‘no va llover’ mañana, para asi poder tomar una decision de por ejemplo en el Agro si largarme a sembrar o no. (De todas maneras miraria el pronostico y no este programa que no funciona muy bien jeje)

```
Classification Report - Neural Network:
              precision    recall  f1-score   support

           0       0.84      0.95      0.89      2271
           1       0.75      0.45      0.56       750

    accuracy                           0.83      3021
   macro avg       0.80      0.70      0.73      3021
weighted avg       0.82      0.83      0.81      3021
```

Ya que como podemos ver en el reporte de clasificación, para la clase 0 (”No llueve”) tiene un Recall del 0.95, lo que significa que el 95% de las instancias reales de "no llueve" fueron identificadas correctamente por el modelo.

Comentarios del Trabajo Práctico:

- Primero que nada creo que el dataset era feo para resolverlo de esta manera, creo que es un problema para resolverlo quizás con series temporales, pero no dimos.
- El modelo me bajo mucho el rendimiento, pero me dio mas real, cuando deje de tener en cuenta para predecir la variable target del otro algoritmo, es decir, en un principio para la regresión lineal habia entrenado con la variable “RainTomorrow”, y en Streamlit el usuario debia ingresar como input esa variable, y medio que no

tenia sentido, por que básicamente vos queres predecir si va llover y cuanto, no sabes si efectivamente va a llover o no. Lo mismo en clasificación, no tuve en cuenta la variable “RainfallTomorrow” , que en un principio la tuve en cuenta, y el modelo funcionaba perfecto, literal el AOC era 1, pero estaba re sesgado, para mi detectaba si RainfallTomorrow > 0, ya clasificaba que iba a llover, la correlación era re alta, pero no tenia sentido.

- No es excusa, pero el jueves pasada mi compañera dejo la materia, y lo encare solo la recta final, y probablemente me quedaron algunas cosas por hacer o hice mal, como SHAP u CV con Optuna. Esperare la corrección y lo mejorare!
- Y como ultimo, me intereso mucho la parte de MLOps y MLFlow, ya que fue lo mas real a un entorno real que vimos (creo), no me ha tocado laburar en Data Science aun, espero que en breve si, pero me re intereso!

Muy buena la materia!

Muchas gracias!

Atte.: Timoteo Garcia