

Inteligencia Artificial

Act 10: Programando Regresión Lineal Múltiple en Python

Docente: Luis Ángel Gutiérrez Rodríguez

Alumno: Jhoana Esmeralda Escobar Barron. 1950748.

Gpo:031

1 Introducción

La regresión lineal múltiple es una técnica estadística utilizada para modelar la relación entre una variable dependiente y varias variables independientes. A diferencia de la regresión lineal simple, que emplea solo un predictor, esta extensión permite capturar relaciones más complejas dentro de los datos.

Este modelo es ampliamente utilizado en diversas áreas, como economía, biología, ingeniería y ciencia de datos, ya que permite analizar y predecir tendencias basadas en varios factores al mismo tiempo. La capacidad de estimar el impacto de cada variable independiente sobre la variable de interés lo convierte en una herramienta poderosa para la toma de decisiones informadas.

Matemáticamente, la ecuación de la regresión lineal múltiple se expresa como:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon \quad (1)$$

donde Y representa la variable dependiente, X_1, X_2, \dots, X_n son las variables independientes, β_0 es el término de intercepción, β_1, \dots, β_n son los coeficientes de regresión y ϵ es el error residual.

En este informe, se presenta un análisis de regresión logística, un tipo de regresión lineal múltiple aplicada a problemas de clasificación. Se explican los pasos seguidos, los resultados obtenidos y las conclusiones derivadas del estudio.

2 Metodología

Para el desarrollo de esta actividad se siguieron los siguientes pasos:

2.1 Carga de Datos

Se utilizó la biblioteca `pandas` para cargar los datos desde un archivo CSV:

```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sb
```

```
dataframe = pd.read_csv("usuarios_inmactin.csv")
dataframe.head()
```

2.2 Exploración de Datos

Se examinaron las primeras filas del conjunto de datos y se realizó un análisis de distribución:

```
print(dataframe.groupby('clase').size())
dataframe.drop(['clase'], axis=1).hist()
plt.show()
sb.pairplot(dataframe.dropna(), hue='clase', size=4, vars=["duracion", "paginas", "acciones", "valor"], kind='reg')
```

2.3 Preparación de los Datos

Se separaron las variables independientes de la variable dependiente:

```
X = np.array(dataframe.drop(['clase'], axis=1))
y = np.array(dataframe['clase'])
X.shape
```

2.4 Entrenamiento del Modelo

Se utilizó la regresión logística para entrenar el modelo con los datos preparados:

```
model = linear_model.LogisticRegression()
model.fit(X, y)
predictions = model.predict(X)
print(predictions[0:5])
```

3 Resultados

Tras entrenar el modelo, se evaluo su desempeño obteniendo su puntuacion:
`model.score(X, y)`

El modelo logro predecir correctamente varias observaciones, aunque es importante evaluar su precision con metricas adicionales.

4 Conclusion

La regresion lineal multiple es una herramienta util para modelar relaciones complejas entre variables. En este ejercicio, se entreno un modelo de regresion logistica para clasificar datos segun diversas características, obteniendo predicciones razonables.

Para mejorar el modelo, podrian explorarse otras tecnicas de preprocesamiento y validacion. Entre ellas se incluyen la normalizacion de datos, la seleccion de variables mas relevantes y la optimizacion de hiperparametros. Ademas, el uso de validacion cruzada ayudaria a obtener una mejor estimacion del desempeño real del modelo.

Finalmente, aunque la regresion logistica es una solucion efectiva para problemas de clasificacion binaria, en futuros estudios podrian explorarse modelos mas complejos, como redes neuronales o arboles de decision, para mejorar la precision y robustez del analisis.