Actividad #11: Programando Regresión Logística en Python

Nombre: Dayla Marely Carrizales Ortega

Matricula: 1952471

1. Introducción

La Regresión Logística es una técnica estadística multivariante que nos permite estimar la relación existente entre una variable dependiente no métrica, en particular dicotómica y un conjunto de variables independientes métricas o no métricas.

El Análisis de Regresión Logística tiene la misma estrategia que el Análisis de Regresión Lineal Múltiple, el cual se diferencia esencialmente del Análisis de Regresión Logística por que la variable dependiente es métrica; en la práctica el uso de ambas técnicas tiene mucha semejanza, aunque sus enfoques matemáticos son diferentes.

El objetivo primordial de esta técnica es el de modelar como influyen las variables regresoras en la probabilidad de ocurrencia de un suceso particular. Mientras que El objetivo general de la Regresión Logística es predecir la probabilidad de un evento de interés en una investigación, así como identificar las variables predictoras útiles para tal predicción.

2. Metodología

Para llevar a cabo esta actividad, primero cargamos el archivo CSV llamado **usuarios_win_mac_lin.csv**. Luego, utilizamos métodos estadísticos que conocemos para describir el conjunto de datos y observar la distribución de cada variable numérica. También agrupamos los datos por la variable clase para conocer la cantidad de registros pertenecientes a cada categoría.

Después se realiza la visualización de los datos usando gráficos de histograma para identificar relaciones entre las variables y posibles patrones en los datos. Una vez analizada la estructura de los datos, preparamos las matrices de características (X) y etiquetas (y) tomando las columnas

necesarias y descartando la variable objetivo. Después, implementamos el modelo de Regresión Logística utilizando la biblioteca sklearn. Ajustamos el modelo a los datos completos y verificamos las predicciones iniciales.

Se realizo una validación cruzada con 10 pliegues para calcular la precisión media y la desviación estándar del modelo, lo que nos permitió estimar su rendimiento con mayor precisión.

Finalmente, realizamos predicciones sobre el conjunto de validación y evaluamos la precisión del modelo usando métricas como la matriz y el informe de clasificación. También probamos el modelo con un nuevo conjunto de datos para predecir el valor de una instancia específica.

2.1 Código Regresión Logistica

```
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(r"usuarios_win_mac_lin.csv")
dataframe.head()
dataframe.describe()

print(dataframe.groupby('clase').size())

dataframe.drop(['clase'], axis=1).hist()
plt.show()

sb.pairplot(dataframe.dropna(), hue='clase', height=4, vars=["duracion",
"paginas", "acciones", "valor"], kind='reg')

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

model = linear_model.LogisticRegression(max_iter=1000)
model.fit(X, y)
```

```
predictions = model.predict(X)
print(predictions[0:5])

model.score(X,y)

validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation =
model_selection.train_test_split(X, y, test_size=validation_size,
random_state=seed)

name='Logistic Regression'
kfold = model_selection.KFold(n_splits=10, shuffle=True, random_state=seed)
cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold,
scoring='accuracy')
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
print(msg)

predictions = model.predict(X_validation)
print(accuracy_score(Y_validation, predictions))

print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

X_new = pd.DataFrame({'duracion': [10], 'paginas': [3], 'acciones': [5], 'valor':
[9]})
model.predict(X_new)
```

3. Resultados

Los resultados obtenidos muestran la precisión del modelo tanto en el entrenamiento como en el de validación. El informe de clasificación muestra la precisión, el recall y el F1-score. El modelo fue capaz de realizar una predicción para un conjunto de valores específicos.

4. Conclusión

La regresión logística en este tipo de problemas permite clasificar de manera efectiva los datos, logrando una precisión notoria en la validación. Los resultados obtenidos eran los que se esperaban y demuestran que el modelo puede ser útil en situaciones similares de clasificación de datos.