Programando Regresion Logistica en Python

Julio Cesar Torres Marquez

marzo 2025

1 Introduccion

La regresion logistica es una tecnica de analisis de datos que emplea herramientas matematicas para encontrar relaciones entre dos conjuntos de datos. A partir de esta relacion, es posible predecir el valor de uno de los conjuntos en funcion del otro. Generalmente, este tipo de prediccion ofrece un numero finito de resultados, como un "si" o un "no".

2 Metodologia

2.1 Importar librerias

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

2.2 Leer el archivo CSV y mostrar los primeros 5 registros

```
dataframe = pd.read_csv(r"usuarios_win_mac_lin.csv")
print(dataframe.head())
```

2.3 Obtener informacion estadistica basica del set de datos

```
print(dataframe.describe())
```

2.4 Analizar cuantos usuarios hay de cada tipo de sistema operativo

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

2.5 Visualizacion de datos

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

Tambien se pueden interrelacionar las entradas para observar la concentracion de usuarios segun el sistema operativo:

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

2.6 Creacion del modelo

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

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

2.7 Validacion del modelo

Para evaluar el modelo, se divide el conjunto de datos en un 80% para entrenamiento y un 20% para validacion:

```
validation_size  =  0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X,y,test_siz

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='acc
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
print(msg)
print(accuracy_score(Y_validation,predictions))
```

2.8 Reporte de resultados

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

2.9 Clasificacion de nuevos valores

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

3 Resultados

El modelo desarrollado permite clasificar a los usuarios en funcion del sistema operativo que utilizan, basandose en datos de navegacion. La precision obtenida en la validacion muestra que el modelo es confiable, con un margen de error bajo. Ademas, la visualizacion de datos permite comprender mejor la distribucion de las caracteristicas de los usuarios en cada categoria.

4 Conclusion

A traves de este proceso se comprendio la aplicacion de la regresion logistica en Python para la clasificacion de usuarios segun su sistema operativo. Se logro entrenar un modelo, evaluar su rendimiento y realizar predicciones con nuevos datos. La validacion del modelo permitio garantizar su efectividad y asegurar que los resultados obtenidos sean fiables.

5 Referencias bibliograficas

- https://aws.amazon.com/es/what-is/logistic-regression/
- Ignacio Bagnato, J. (2020). Aprende machine learning. Leanpub.