

Ejercicio Regresión Lineal - OS

July 15, 2023

```
[ ]: # Ejercicio para predecir qué sistema operativo está utilizando quien visita mi
    ↳sito web
    # Usando regresión lineal
```

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

```
[2]: #carga del csv
dataframe = pd.read_csv("usuarios_win_mac_lin.csv")
dataframe.head()
```

```
[2]:
```

	duracion	paginas	acciones	valor	clase
0	7.0	2	4	8	2
1	21.0	2	6	6	2
2	57.0	2	4	4	2
3	101.0	3	6	12	2
4	109.0	2	6	12	2

```
[3]: #Estadística descriptiva
dataframe.describe()
```

```
[3]:
```

	duracion	paginas	acciones	valor	clase
count	170.000000	170.000000	170.000000	170.000000	170.000000
mean	111.075729	2.041176	8.723529	32.676471	0.752941
std	202.453200	1.500911	9.136054	44.751993	0.841327
min	1.000000	1.000000	1.000000	1.000000	0.000000
25%	11.000000	1.000000	3.000000	8.000000	0.000000
50%	13.000000	2.000000	6.000000	20.000000	0.000000
75%	108.000000	2.000000	10.000000	36.000000	2.000000
max	898.000000	9.000000	63.000000	378.000000	2.000000

```
[4]: #Cuantos renglones tengo de cada clase  
dataframe.groupby('clase').size()
```

```
[4]: clase  
0    86  
1    40  
2    44  
dtype: int64
```

```
[5]: dataframe.shape
```

```
[5]: (170, 5)
```

```
[10]: #Separar mi dataset en matriz de variables y vector de respuestas correctas  
X = np.array(dataframe.drop(columns=["clase"]))  
y = np.array(dataframe["clase"])
```

```
[11]: X.shape
```

```
[11]: (170, 4)
```

```
[15]: #Crear modelo  
  
model = linear_model.LogisticRegression(max_iter = 350)
```

```
[16]: #Entrenar modelo  
  
model.fit(X,y)
```

```
[16]: LogisticRegression(max_iter=350)
```

```
[17]: #Crear predicciones  
  
predictions = model.predict(X)  
predictions[:5]
```

```
[17]: array([2, 2, 2, 2, 2])
```

```
[18]: model.score(X,y)
```

```
[18]: 0.7764705882352941
```

*** MANERA CORRECTA DE HACER REGRESION LOGISTICA ***

Dividir mi dataset en 80 y 20 de manera aleatoria

```
[20]: validation_size = 0.20
```

```
X_train, X_validation, y_train, y_validation = model_selection.  
↳train_test_split(X, y, test_size = validation_size, random_state = 7)
```

```
[24]: kfold = model_selection.KFold(n_splits=10)  
cv_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold,↳  
↳scoring='accuracy')  
msg = "Logistic Regression: " + str(cv_results.mean()), str(cv_results.std())
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-  
packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed  
to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,  
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-  
packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed  
to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,  
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-  
packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed  
to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,  
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-  
packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed  
to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

[25]: msg

[25]: ('Logistic Regression: 0.7285714285714284', '0.09418550477897196')

[30]: *#Predicciones de mi set de prueba*
predictions = model.predict(X_validation)
accuracy_score(Y_validation, predictions)

[30]: 0.8529411764705882

[31]: *#Imprimir la matriz de confusión*
confusion_matrix(Y_validation, predictions)

[31]: array([[16, 0, 2],
[3, 3, 0],
[0, 0, 10]])

[32]: *#Reporte de clasificación*
print(classification_report(Y_validation, predictions))

	precision	recall	f1-score	support
0	0.84	0.89	0.86	18
1	1.00	0.50	0.67	6
2	0.83	1.00	0.91	10
accuracy			0.85	34
macro avg	0.89	0.80	0.81	34
weighted avg	0.87	0.85	0.84	34

[33]: *#Crear una nueva predicción con datos de entrada*
X_new = pd.DataFrame({'duracion': [10], 'paginas':[3], 'acciones':[5], 'valor':
↪ [9]})

[34]: model.predict(X_new)

/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/sklearn/base.py:439: UserWarning: X has feature names, but LogisticRegression was fitted without feature names
f"X has feature names, but {self.__class__.__name__} was fitted without"

[34]: array([2])

[]: