

# 1 Que es la regresión lineal múltiple

## 2 Metodología

Usamos las siguientes importaciones para el programa El código del programa es el siguiente:

```
1 import pandas as pd
2 import numpy as np
3 from sklearn import linear_model, model_selection
4 from sklearn.metrics import classification_report, confusion_matrix,
  accuracy_score
5 from sklearn.preprocessing import StandardScaler
```

```
1 dataframe = pd.read_csv("../usuarios_win_mac_lin.csv")
2 dataframe.drop(['class'], axis=1).hist()
3
4 X = np.array(dataframe.drop(['class'], axis=1))
5 y = np.array(dataframe['class'])
6
7 scaler = StandardScaler()
8 X = scaler.fit_transform(X)
9
10 model = linear_model.LogisticRegression(max_iter=500)
11 model.fit(X, y)
12
13 predictions = model.predict(X)
14
15 validation_size = 0.20
16 seed = 7
17
18 X_train, X_validation, Y_train, Y_validation =
  model_selection.train_test_split(
19     X, y, test_size=validation_size, random_state=seed
20 )
21
22 name = 'LogisticRegression'
23 kfold = model_selection.KFold(n_splits=10, random_state=seed, shuffle=True)
24
25 cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold,
  scoring='accuracy')
26
27 msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
28
29 predictions = model.predict(X_validation)
```

se modifiqué algunas partes del código ya que había elementos obsoletos que hacían que el programa no funcionara.

## 3 Resultados

Al ejecutar el programa obtenemos los siguientes valores Dimensiones de X: (170, 4) Primeras 0 predicciones: [2 2 2 2 0] Precisión en datos de entrenamiento: 0.7 LogisticRegression:

0.630769 (0.136055) Precisión en datos de validación: 0.7352941176470589 Matriz de confusión A =

$$\begin{bmatrix} 14 & 0 & 4 \\ 4 & 2 & 0 \\ 1 & 0 & 9 \end{bmatrix}$$

Reporte de clasificación:

Clase	Precisión	Recall	F1-Score	Soporte
0	0.74	0.78	0.76	18
1	1.00	0.33	0.50	6
2	0.69	0.90	0.78	10
<b>Exactitud</b>			0.74	34
<b>Promedio macro</b>	0.81	0.67	0.68	34
<b>Promedio ponderado</b>	0.77	0.74	0.72	34

Table 1: Reporte de clasificación

## 4 Conclusión

El modelo tiene un buen nivel de predicción (70%), aun que puede ser mejorable, ya tendrá buenas predicciones, podría mejorarse aumentando la cantidad de datos en la clase 1 ya que tiene un porcentaje de recall muy bajo, el modelo solo predice un clase 1 cuando este realmente seguro.