Actividad 11: Programando Regresión Logistica en Python

García Herrera Carlos Eduardo

March 2025

1 Introduction

La regresión lineal es un modelo estadístico que busca establecer una relación lineal entre una variable dependiente (o de salida) y una o más variables independientes (o explicativas). En términos simples, intenta encontrar la "mejor línea" que prediga el valor de la variable dependiente a partir de las variables independientes.

2 Metodología

2.1 Parte 1: Creacion del Ambiente Virtual

Las siguiente lineas de codigo, se encargan de crear un ambiente virtual llamado VirtualEnv con las librerias necesarias para ejecutar el script

```
#Automatic creation of a virtual environment to run the script and install the libraries
import subprocess
import os
import venv
import sys
script_dir = os.path.dirname(os.path.realpath(__file__))
env_name = os.path.join(script_dir, "VirtualEnv")
if os.path.exists(os.path.join(script_dir, "VirtualEnv")):
    #Checks if the VirtualEnv is activated
    if sys.prefix == sys.base_prefix:
        print("Activating the Virtual Environment...")
        python_exe = os.path.join(env_name, "Scripts", "python")
        subprocess.run([python_exe, __file__])
else:
    print("Installing the Required Libraries on a New Virtual Environment")
    venv.create(env_name, with_pip=True)

    libraries = ["scikit-learn", "matplotlib","seaborn","pandas","numpy"]
    for lib in libraries:
        subprocess.run([os.path.join(env_name, "Scripts", "pip"), "install", lib],
        check=True)

    python_exe = os.path.join(env_name, "Scripts", "python")
    subprocess.run([python_exe, __file__])
```

2.2 Parte 2: Analisis of the data

Se cargan los datos contenidos en articulos_ml.csv y se analizan sus distintas propiedades

```
#cargamos los datos de entrada
dataframe = pd.read_csv("./usuarios_win_mac_lin.csv")

#vemos los primeros Registros
print(dataframe.head())
```

```
#vemos las caracteristicas del dataframe
print(dataframe.describe())

#analizaremos cuantos resultados tenemos de cada tipo usando la función groupby y vemos
#que tenemos 86 usuarios \Clase 0", es decir Windows, 40 usuarios Mac y 44 de Linux.
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"]
plt.show()
```

2.3 Parte 3: Regresion Lineal

A continuacion se hace el calculo de la regresion Lineal:

```
#Creacion del Modelo de Regresion Logistica
X = np.array(dataframe.drop(['clase'],axis=1))
y = np.array(dataframe['clase'])
X.shape

model = linear_model.LogisticRegression()
model.fit(X,y)

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

print("Score:")
print(model.score(X,y))
```

2.4 Parte 4: Validacion del Modelo

A continuacion se hace la validacion del modelo:

```
#Validacion del Modelo
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, random_state=seed,shuffle=True)
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("Score")
print(accuracy_score(Y_validation, predictions))

print("Matriz Confusion")
print(confusion_matrix(Y_validation, predictions))
```

```
print("Reporte de Clasificacion")
print(classification_report(Y_validation, predictions))

##Clasificacion de Nuevos Valores
X_new = pd.DataFrame({'duracion': [10], 'paginas': [3],
'acciones': [5], 'valor': [9]})
new_predict=model.predict(X_new)
print("Nueva Prediccion:")
print(new_predict)

input("Press any key to Exit")
```

3 Resultados

Al ejecutar el script de pyhton la informacion obtenida es la siguiente:

3.1 Analisis de los Datos

La informacion de los datos es la siguiente:

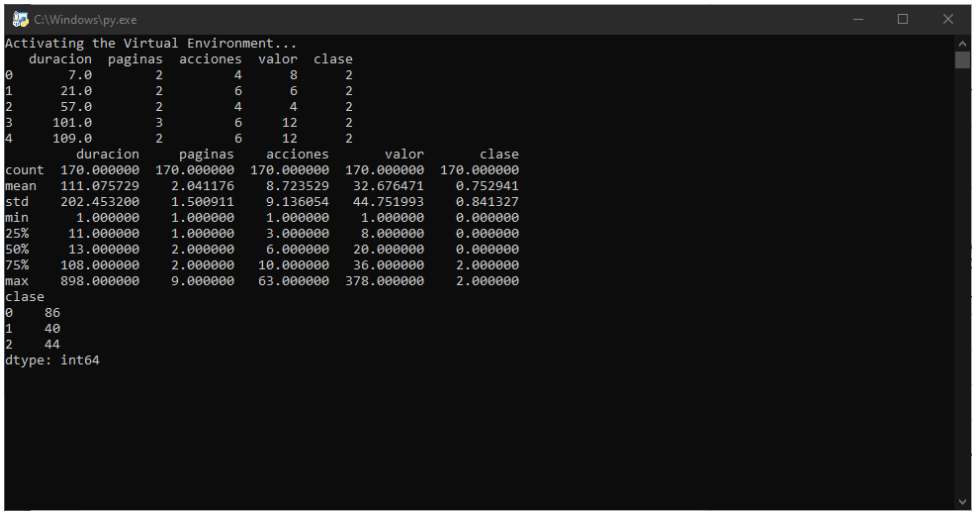


Figure 1: Informacion de los Datos

Teniendo en cuenta que:

- Clase 0: Windows
- Clase 1: Mac
- Clase 2: Linux

Es posible ver graficamente estos datos en las siguientes 2 figuras:

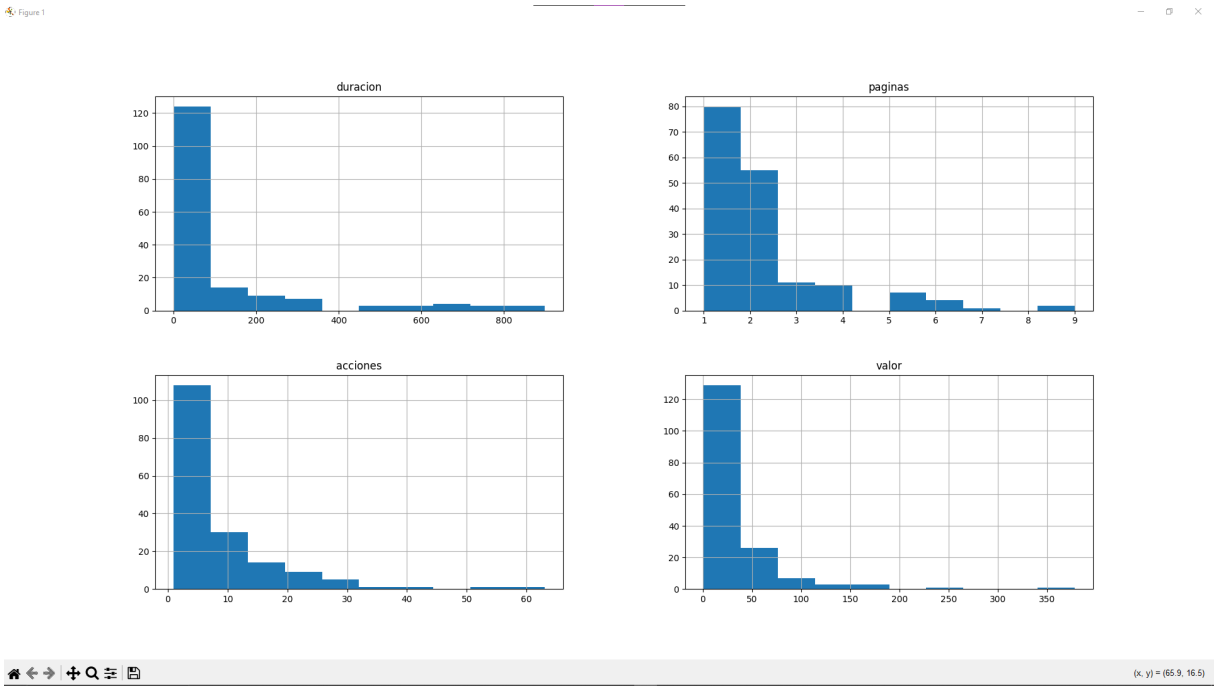


Figure 2: Informacion de las caracteristicas de los registros

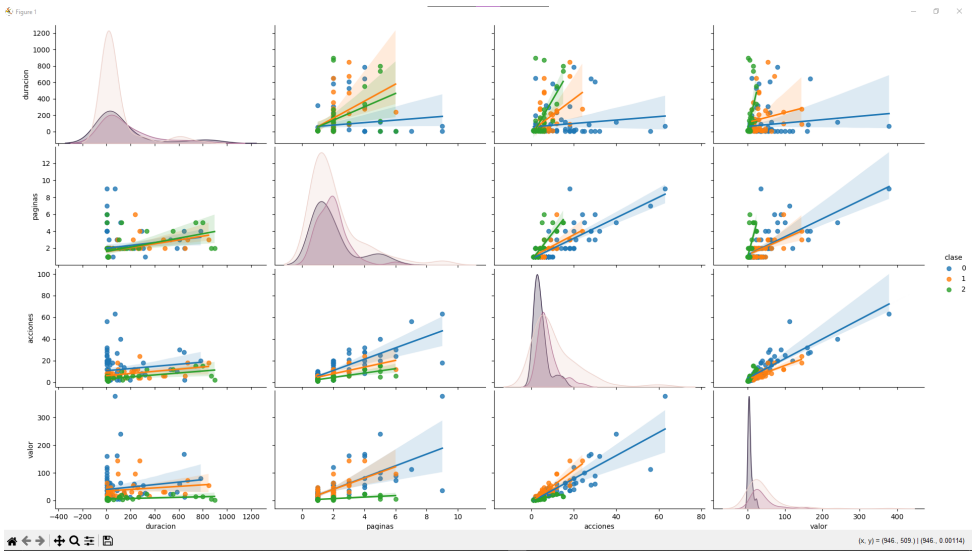
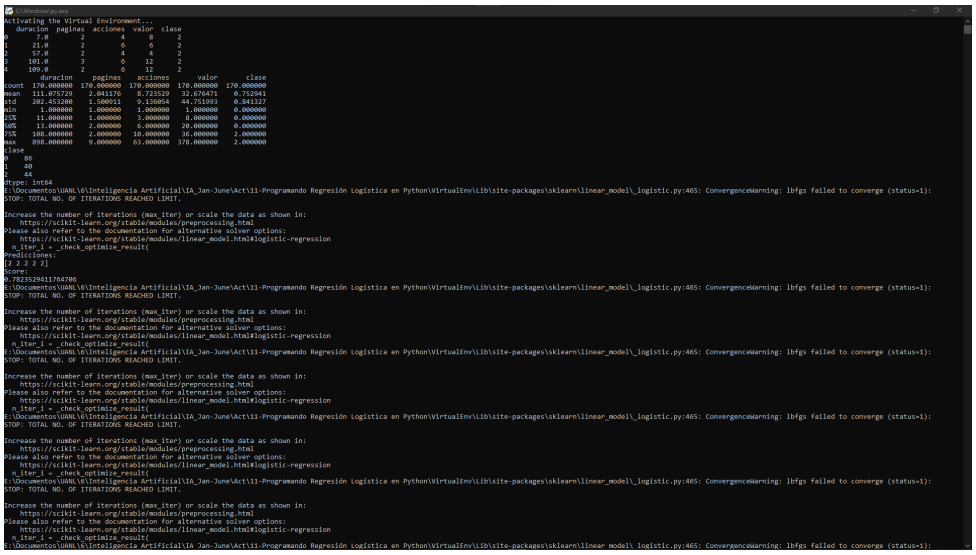


Figure 3: Informacion de las caracteristicas de los registros, clasificados por tipo de SO

3.2 Regresion Logistica

Los resultados de la regresion Logistica se pueden ver en la siguiente figura:



```
Activating the Virtual Environment...
   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
         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
clase
0    86
1    40
2    44
dtype: int64
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
Predicciones:
[2 2 2 2 2]
Score:
0.7823529411764706
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
```

Figure 4: Datos calculados de la regresion Logistica

Es posible apreciar que, despues de hacer el modelo logistico, se hacen las predicciones con los registros contenidos en el archivo csv, se muestran los primeros 5 resultados, siendo clase 2 todos ellos, coincidiendo con el valor que se almacena en la columna de Clase, por lo tanto, nuestro modelo es capaz de predicir el tipo de usuario de forma correcta

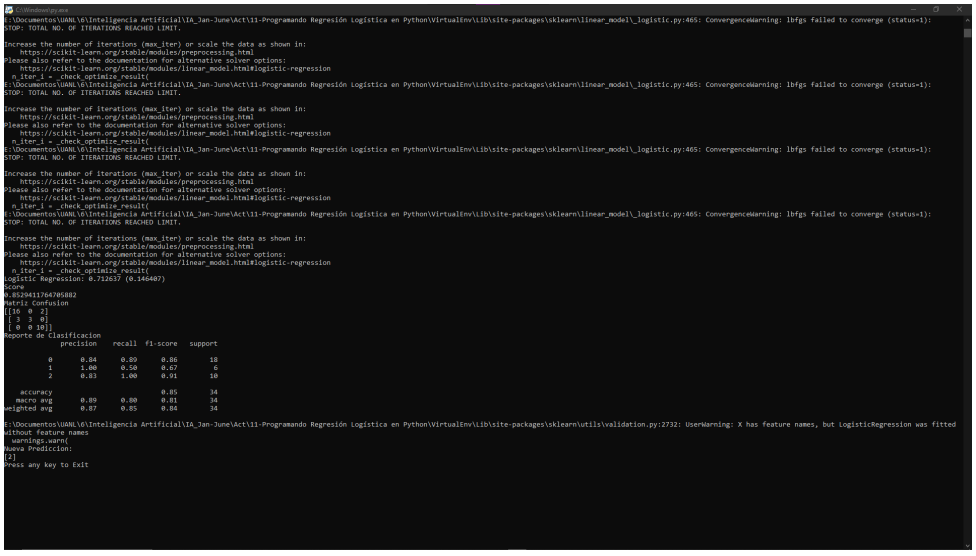
Tambien, se puede ver que el modelo alcanza un Score de 0.7823

3.3 Validacion del Modelo

Para hacer la validacion del modelo, se dividio el conjunto de datos en 0.80 para entrenamiento, y 0.20 para validacion, usando solamente el 0.80 de los datos para entrenamiento se obtiene:

- Score: 0.7126

Al hacer el las predicciones para todo el conjunto de datos con este nuevo modelo, se obtiene que el 85% de los registros fueron acertados correctamente



```
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\linear_model\_logistic.py:465: 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
  n_iter_i = _check_optimize_result(
Logistic Regression: 0.712637 (0.146407)
Score
0.8529411764705882
Matriz Confusion
[[16  0  2]
 [ 3  3  0]
 [ 0  0 10]]
Reporte de Clasificacion
              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

E:\Documentos\UANL\6\Inteligencia Artificial\IA_Jan-June\Act\11-Programando Regresión Logística en Python\VirtualEnv\Lib\site-packages\sklearn\utils\validation.py:2732: UserWarning: X has feature names, but LogisticRegression was fitted
without feature names
  warnings.warn(
Nueva Prediccion:
[2]
Press any key to Exit
```

Figure 5: Datos calculados de la regresion Logistica con Validacion

3.3.1 Matriz de Confusion

La matriz de confusion fue la siguiente:

$$C = \begin{bmatrix} 16 & 0 & 2 \\ 3 & 3 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

Podemos concluir lo siguiente:

- Windows: 16 aciertos
- Mac 1: 3 aciertos
- Linux 2: 10 aciertos

3.3.2 Reporte de Clasificacion

El reporte de los aciertos obtenidos por medio de este modelo es el siguiente:

	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

Table 1: Métricas de clasificación

3.4 Prediccion de un Nuevo Registro

Usando el modelo anterior, se trato de predecir el tipo de usario para los siguiente parametros:

- Duracion: 10
- Paginas: 3
- Acciones: 5
- Valor: 9

Como se puede apreciar en la figura 5, el modelo lo predijo como usuario de Linux (clase 2)

4 Conclusion

La regresión logística es una herramienta poderosa para abordar problemas de clasificación binaria y se ha aplicado con éxito en diversas áreas. Aunque es un modelo relativamente simple, su capacidad para predecir probabilidades y clasificar en función de características específicas la convierte en una técnica valiosa. Sin embargo, su rendimiento puede verse afectado por la calidad de los datos y la correcta selección de variables, por lo que es crucial realizar un análisis exhaustivo para obtener resultados confiables.