

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
# upload json cert
# from google.colab import files
# files.upload()
# pending, bash change permissions
```

```
!pip install kaggle
```

Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.6.14)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.6.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.4)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.7)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.7)

```
!mkdir ~/.kaggle/
!cp /content/drive/MyDrive/kaggle.json ~/.kaggle/
```

mkdir: cannot create directory '/root/.kaggle/': File exists

```
!chmod 600 ~/.kaggle/kaggle.json
```

```
import kaggle
!kaggle datasets download -d mlg-ulb/creditcardfraud
#Downloading creditcardfraud.zip to /content
```

Dataset URL: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
License(s): DbCL-1.0
Downloading creditcardfraud.zip to /content
62% 41.0M/66.0M [00:00<00:00, 116MB/s]
100% 66.0M/66.0M [00:00<00:00, 146MB/s]

```
!unzip /content/drive/MyDrive/XAI/Projects/Data/creditcardfraud.zip -d /content/drive/MyDrive/XAI/Projects/Data/CreditCardFraudDbKaggle
```

Archive: /content/drive/MyDrive/XAI/Projects/Data/creditcardfraud.zip
replace /content/drive/MyDrive/XAI/Projects/Data/CreditCardFraudDbKaggle/creditcard.csv? [y]es, [n]o, [A]ll, [N]one, [r]ename:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
```

```
import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/XAI/Projects/Data/CreditCardFraudDbKaggle/'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

/content/drive/MyDrive/XAI/Projects/Data/CreditCardFraudDbKaggle/creditcard.csv
/content/drive/MyDrive/XAI/Projects/Data/CreditCardFraudDbKaggle/creditcardNew (1).csv

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
pd.set_option('display.max_columns', None)

#MODEL SELECTIONS
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
#Thresholds
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve

data=pd.read_csv('/content/drive/MyDrive/XAI/Projects/Data/CreditCardFraudDbKaggle/creditcard.csv')
data.head()

```

	Time	V1	V2	V3	V4	V5	V6	V7	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270

```
data.columns
```

```

Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
      'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
      'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
      'Class'],
      dtype='object')

```

```

!pip install shap
import xgboost
import shap

```

```

#To start with : fit this model to xboost
#model = xgboost.XGBRegressor().fit(data)

```

```

Requirement already satisfied: shap in /usr/local/lib/python3.10/dist-packages (0.45.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from shap) (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from shap) (1.11.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from shap) (1.2.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from shap) (2.0.3)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-packages (from shap) (4.66.4)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-packages (from shap) (24.0)
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.10/dist-packages (from shap) (0.0.8)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from shap) (0.58.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba->shap) (0.41.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2024.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->shap)

```

Double-click (or enter) to edit

```

x_dummy=data.drop(columns='Class', axis=1)
y=data['Class']

```

```

scaler=StandardScaler()
x=scaler.fit_transform(x_dummy)

```

data.shape

➡ (284807, 31)

```
x_train, x_test, y_train, y_test=train_test_split(x,y, test_size=0.11, random_state=123)
```

```
def logic_regression(x_train, y_train, x_test):
    lr=LogisticRegression()
    lr.fit(x_train, y_train)
    y_train_pred=lr.predict(x_train)
    y_train_cl_report=classification_report(y_train, y_train_pred, target_names = ['No Fraud', 'Fraud'])
    print("_"*100)
    print("TRAIN MODEL CLASSIFICATION REPORT")
    print("_"*100)
    print(y_train_cl_report)
    y_test_pred=lr.predict(x_test)
    y_test_cl_report=classification_report(y_test, y_test_pred, target_names = ['No Fraud', 'Fraud'])
    print("_"*100)
    print("TEST MODEL CLASSIFICATION REPORT")
    print("_"*100)
    print(y_test_cl_report)
    print("_"*100)
    return y_test_pred, lr
```

x_test.shape

➡ (31329, 30)

```
model = xgboost.XGBRegressor().fit(x_test, y_test,)
```

```
explainer = shap.Explainer(model)
```

```
shap_values = explainer(x_test)
```

```
# visualize the first prediction's explanation
shap.plots.waterfall(shap_values[11])
```

➡



