Credit Card Fraud Detection

Date No.

→ ML with Python

→ Credit card data → Data Pre Processing (Imbalance) → Data Analysis → Train Test Split → Logistic Regression Model → Evaluation

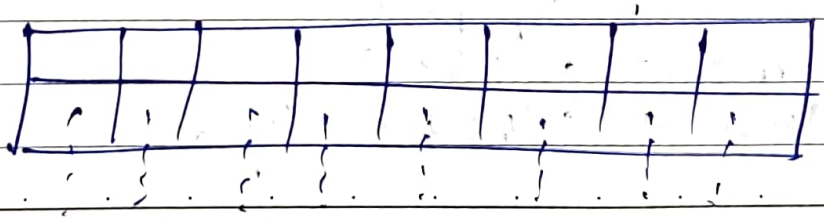
① Dependies → Libraries
· Shift + Enter = Run
Import the dependencies

1. import numpy as np → Making arrays
2. import pandas as pd → Pointers data frames
3. from sklearn.model_selection import train_test_split → Split data into training test data
4. from sklearn.linear_model import LogisticRegression
5. from sklearn.metrics import accuracy_score

② Load dataset to Pandas dataframe

1. credit_card_data = pd.read_csv('')
⇒ Copy path of credit_data.csv → paste

③ First 5 rows of dataset
credit_card_data.head()



Label 0
Label 1
↳ fraudulent

→ credit_card_data.tail()
After 2 days of transaction | Last 5 rows

(4) Dataset information
credit_card_data.info()

(5) Checking no. of missing values in each column
credit_card_data.isnull().sum()

(6) Distribution of legit transaction & fraudulent transaction
credit_card_data['Class'].value_counts()
⇓
This dataset is highly unbalance.
0 – Normal transaction
1 – fraudulent transaction

(7) Separating the data for analysis
legit = credit_card_data[credit_card_data.Class == 0]
fraud = credit_card_data[credit_card_data.Class == 1]
print(legit.shape)
print(fraud.shape)

(8) Statistical measures of the data
① legit.Amount.describe()
count → No. of points
② fraud.Amount.describe()
Mean → More than legit.

Date No.

(9) Compare values for both transaction
Credit_card_data.groupby('Class').mean()
0 - Normal
1 - Fraudulent] - Show
wille diff. of both of mean.

(10) Undersampling
⇒ Build a sample dataset containing similar distribution of normal transaction and fraudulent transaction
⇒ No. of fraudulent transaction = 492

Ex: Legit (284315, 31)
Fraud (492, 31)
we will take randomly 492 data from legit and join to fraud.
↓
Now 492 + 492
(Fraud) (legit)
Uniform dataset

legit_sample = legit.sample (n = 492)
Concatenate 2 datasets/frames.
new_dataset = pd.concat([legit_sample, fraud], axis = 0)
↓
(Added (by 1)
↓
Add rows wise.

axis = 1 → Add column wise.

MATRIKAS

Date No.

⇒ (1) New_dataset.head()
first 5 rows.
(2) New_dataset.tail()
last 5 rows.

(3) New_dataset['class'].value_counts()
1 492
0 492] shows

(4) new_dataset.groupby('class').mean()

↓

(Nature of dataset not changed).

↓

Split the data into features & target

store features [X = new_dataset.drop(columns='class', axis=1)
Y = new_dataset['class']

↓

Don't have class column

print(Y)

↓

Splits data into Training & Testing data

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

stratify=Y → distribution of 0, 1

random_state → How to split your data

test_size=0.2 → 20% of data to test data

Date No.

print (X.shape, X_train.shape, X_test.shape)

(_,_) (_,_) (_,_)

Reference: Intellipaat

Date No.

Machine Learning → Based on Predictions

Ex: For instance 100 Records

Types → ① Supervised (Predicted Models) ↓

② Unsupervised ID → doesn't pay head.

Why? → Reasons → (Bunch of reasons)

① Take Dataset

Customer C1 = 100K 20K (given) 80K (left)

Find → Avg. age, financial situations ... etc

then decide whether he/she is fraud or not.

② Make a characteristic to determine fraud

Ex: [Age: 30, salary < 50K, G = Male]

Bank issue credit card or not.

Not Paying amount

Invalid transactions.

Dataset

C1	C2	C3	C4	C5	CN
Customer ID	Amt. loaned to Bank	Age	salary		whether paying or not on time (✓)

variables → (Dependent variable) that you will predict

(CN-1) → independent variable

Dependent Variable → ① Categorical Problem

Have fixed no. of values → Fraud, ontime, profit/loss → ML Models ↓

① Logistic reg, Decision tree, Naive Bay Algorithm, Random forest, KNN, SVM, LDA. → Classification Models.

MATRIKAS

Date No.

(2) Numerical → Sales, Profit, Salary? → ML Models
Regression Models

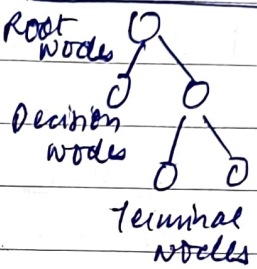
→ Linear Regression, Decision Tree, RF → Random Forest, KNN, SVM

Random Forest

Tree-Based Model
(Flow Chart)

Ways to optimize tree
Ex: 5 trees - Test all trees and then count the prediction from each tree.

Decision Tree



(Tree)

Initially 1 tree prediction then Forest of trees

GOOD MODEL CHARACTERISTICS

(1) Accuracy
(2) Stability

Dataset

Time	V_1 ——— V_{28}	Amount	Class
	. . . . .		0 → Legit
	. . . . .		1 - fraud.