Part3_Metric

November 20, 2021

1 Part 3 - Metric

1.1 Getting to know the dataset

[1]:
```
import pandas as pd
df = pd.read_csv("creditcard.csv")[:80_000]
#??pd.read_csv # calls function string
# Input indices can be written with tousender seperator "_"
df.shape
```

[1]:
```
(80000, 31)
```

[9]:
```
df.head(3)
```

[9]:
```
   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
1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803
2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461

         V8        V9  …       V21       V22       V23       V24       V25  \
0  0.098698  0.363787  … -0.018307  0.277838 -0.110474  0.066928  0.128539
1  0.085102 -0.255425  … -0.225775 -0.638672  0.101288 -0.339846  0.167170
2  0.247676 -1.514654  …  0.247998  0.771679  0.909412 -0.689281 -0.327642

        V26       V27       V28  Amount  Class
0 -0.189115  0.133558 -0.021053  149.62      0
1  0.125895 -0.008983  0.014724    2.69      0
2 -0.139097 -0.055353 -0.059752  378.66      0

[3 rows x 31 columns]
```

The are 28 columns with numeric features, one class (label) column where 1 stands for fraud and 0 for not fraud. Column amount shows how much the fraud amount was.

[19]:
```
# Import
X = df.drop(columns=["Time", "Amount", "Class"]).values # numpy array
y = df["Class"].values
f"Shape of X{X.shape}, Shape of y{y.shape}, Fraud cases = {y.sum()} "
```

[19]: 'Shape of X(80000, 28), Shape of y(80000,), Fraud cases = 196 '

1.2 Preparing the Test and Training Data

1.3 Evaluating a Models Performance

1.3.1 Regression Analysis

For regression models the most common methods for evaluating a models performance are - Coefficient of Determination

$$R^2 = 1 - \frac{\sigma_{Residuals}}{\sigma_{total}}$$

- Root Mean Square Error (RMSE)

$$RMSD = \sqrt{MSE(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)}$$

- Mean Absolute Error

$$MAE = \frac{\Sigma|y_i - x_i|}{n}$$

1.3.2 Classification Analysis

For a classification problem there are differnt metrics to understand if the model predictes the desiered out.

- Accuracy

  $$\frac{\text{Correct Prediction}}{\text{Total Prediction}}$$

  It is calculated as a proportion of prediction in the test set that were predited correctly, divided by all predictions that were made in the test set. Conversley, the error rate can be calculated as the total number of wrong predicition on the test sets, divided by all prediction made on the test set.

  $$ErrorRate = \frac{\text{Incorrent Predictions}}{\text{Total Predictions}}$$

- Precision
- Recall
- F1 Score

[2]:
```
from sklearn.model_selection import train_test_split
```

1.3.3 Simple logistic regression

The model optimization algroithm is not converting, therefore the max iteratonsteps must be specify.

[27]:
```
from sklearn.linear_model import LogisticRegression

mod = LogisticRegression(max_iter=200)
# Fit and Transform
pred_Cases = mod.fit(X,y).predict(X).sum()
```

```
f"Actually Fraud cases {y.sum()} and predicted cases {pred_Cases}"
```

[27]: 'Actually Fraud cases 196 and predicted cases 151'