Credit Card Fraud

Project Overview

- Used sklearn (machine learning), pandas (library), and seaborn (data visualization)
- Aim:
  - Investigate what attributes differentiates legitimate and fraudulent credit card transactions
  - To create a model to help financial institutions predict credit card fraud based on several attributes
- Work with multivariate data on credit card transactions to create such model
- Used variables such as: transaction type, transaction amount, zip code, and more

Dataset

ADITYA KADIWAL · UPDATED 10 MONTHS AGO

10 New Notebook Download (2 MB)

Online Credit Card Transactions

Over 90k credit card transactions marked as Fraudulent or Legitimate

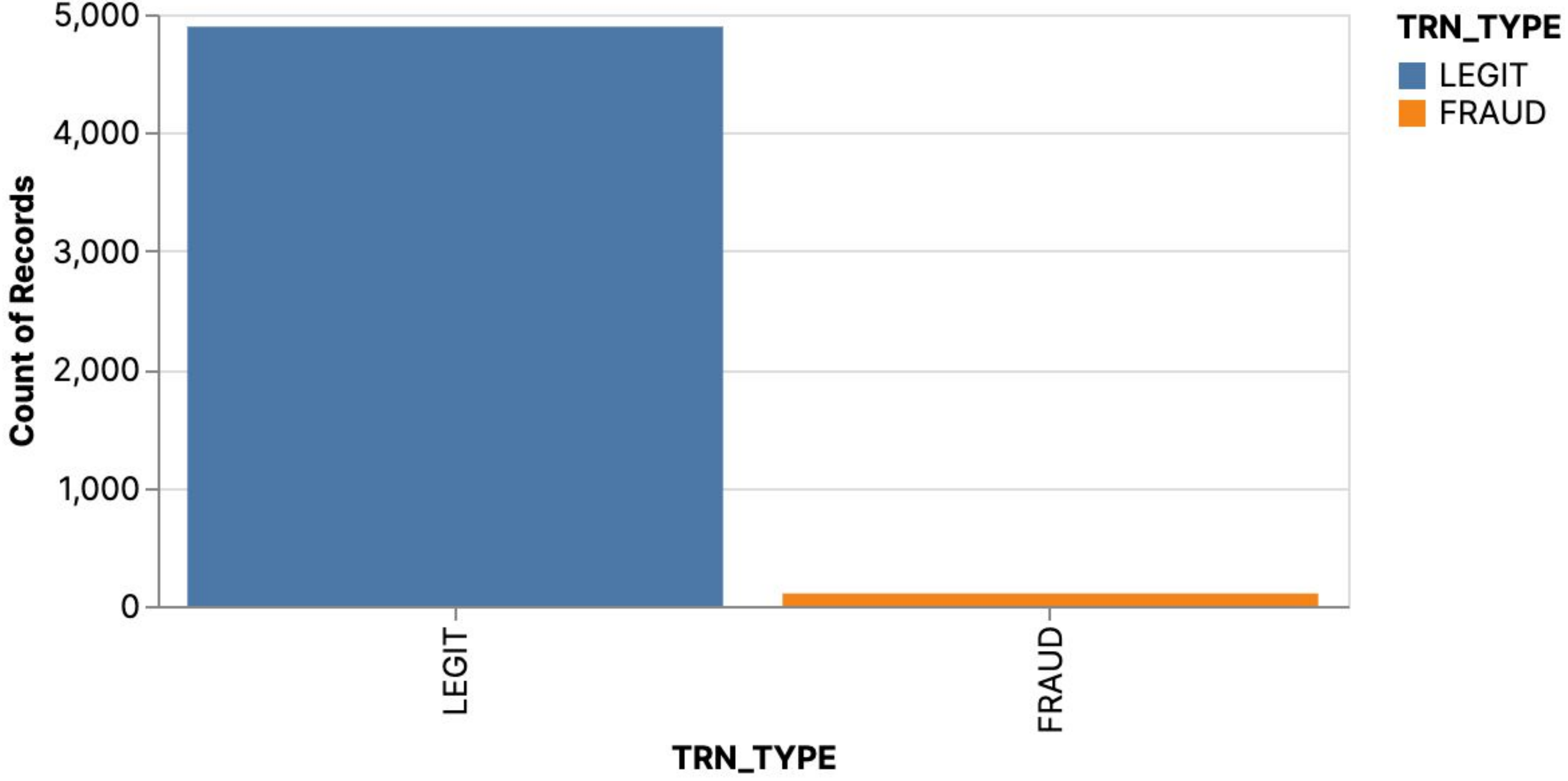
The file `CC_FRAUD.csv` contains details of 94682 transactions that are marked as fraudulent or legitimate transactions.

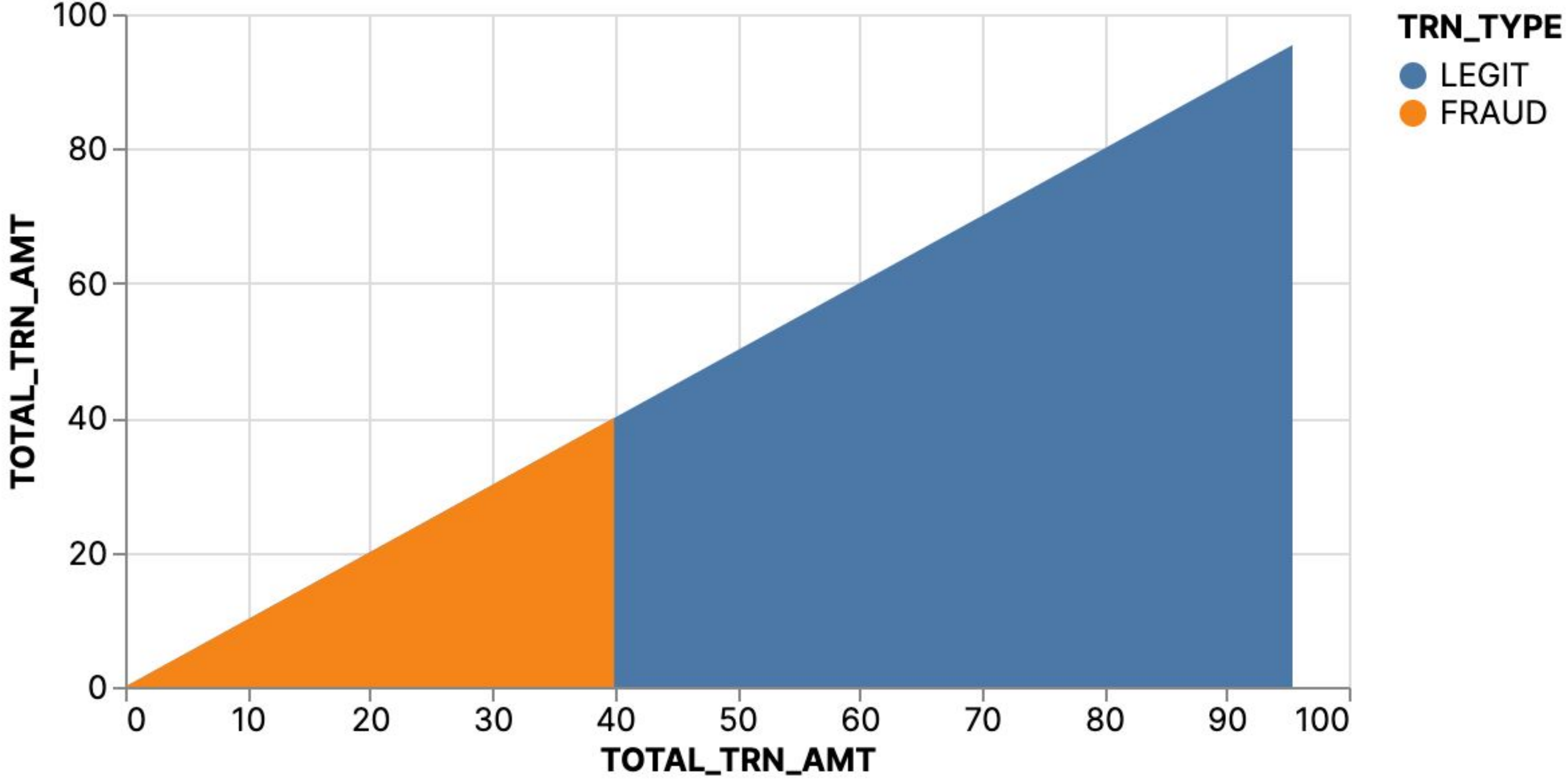
Column description:

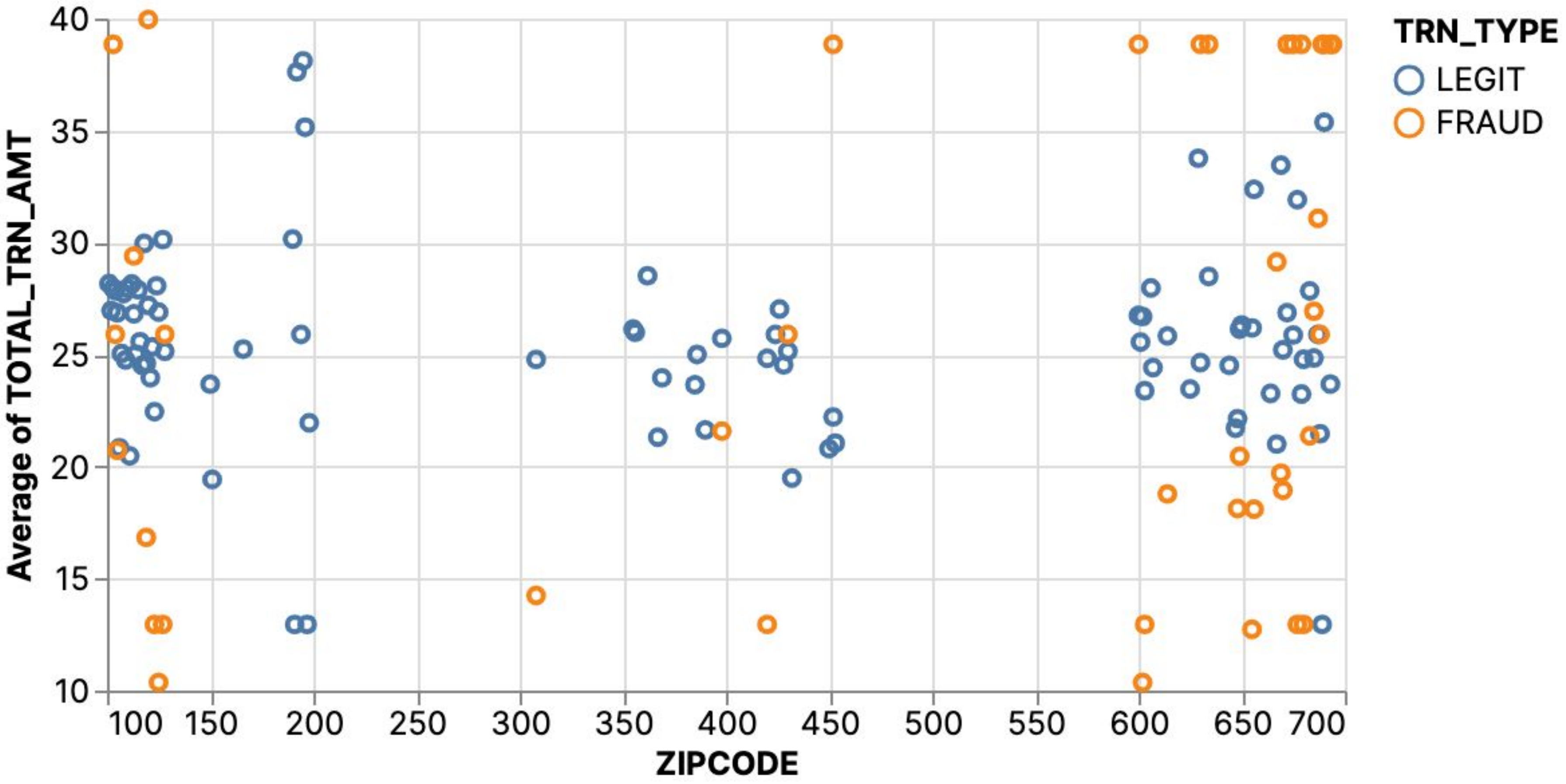
```
1. DOMAIN: The domain name of the customer's email address that was used for the transaction
(Masked)
2. STATE: The state code of the customer's location.
3. ZIPCODE: The zip code of the customer's location.
4. TIME1: Hour feature #1 of the transaction.
5. TIME2: Hour feature #2 of the transaction.
6. VIS1: Anonymized feature #1 for feature VIS.
7. VIS2: Anonymized feature #2 for feature VIS.
8. XRN1: Anonymized feature #1 for feature XRN.
9. XRN2: Anonymized feature #2 for feature XRN.
10. XRN3: Anonymized feature #3 for feature XRN.
11. XRN4: Anonymized feature #4 for feature XRN.
12. XRN5: Anonymized feature #5 for feature XRN.
13. VAR1: Anonymized feature #1 for feature VAR.
14. VAR2: Anonymized feature #2 for feature VAR.
15. VAR3: Anonymized feature #3 for feature VAR.
16. VAR4: Anonymized feature #4 for feature VAR.
17. VAR5: Anonymized feature #5 for feature VAR.
18. TRN_AMT: The transaction amount.
19. TOTAL_TRN_AMT: The total transaction amount.
20. TRN_TYPE: The type of transaction whether FRAUD or LEGIT.
```

- The initial dataset had this many columns
- Decided to disregard some columns because no relevance
- Only ended up using:
  - Domain
  - State
  - Zip Code
  - Transaction amount
  - Total transaction amount
  - Transaction type

	DOMAIN object TMA.COM 17.4% XOSOP.COM 16.7% 9807 others 65.9%	STATE object	ZIPCODE int64	TRN_AMT float64	TOTAL_TRN_AMT f.	TRN_TYPE object
0	CDRZLKAJIJVQHCN.COM	AO	675	12.95	12.95	LEGIT
1	NEKSXUK.NET	KK	680	38.85	38.85	LEGIT
2	XOSOP.COM	UO	432	38.85	38.85	LEGIT
3	TMA.COM	KR	119	11.01	11.01	LEGIT
4	VUHZRNB.COM	PO	614	12.95	12.95	LEGIT
5	CIWEVXGWRG.ORG	ROI	386	49.95	49.95	LEGIT
6	KZOGEIFBAVSI.NET	LM	127	12.95	12.95	LEGIT
7	TMA.COM	AR	649	10.36	10.36	LEGIT
8	VUHZRNB.COM	BO	308	38.85	38.85	LEGIT
9	EAYROLLTBU.COM	PO	614	10.36	10.36	LEGIT


5,000
4,000
3,000
2,000
1,000
0
Count of Records
LEGIT
FRAUD
TRN_TYPE
TRN_TYPE
LEGIT
FRAUD


TRN_TYPE
LEGIT
FRAUD
TOTAL_TRN_AMT
TOTAL_TRN_AMT
0
20
40
60
80
100
0
10
20
30
40
50
60
70
80
90
100



Average of TOTAL_TRN_AMT
40
35
30
25
20
15
10
100
150
200
250
300
350
400
450
500
550
600
650
700
ZIPCODE
TRN_TYPE
LEGIT
FRAUD

Sklearn using Logistic Regression Model: Splitting Data

- Predicts based on inputted attributes to predict whether a transaction is legitimate or fraudulent
- 70% training data, 30% test data

```
training_data, test_data = train_test_split(cc_fraud_tbl, test_size=0.3, random_state=2)
```

```
training_data, test_data = train_test_split(cc_fraud_tbl, test_size=0.3)
# Here we are splitting the data into training and test data
# We decided to split 70% trainings and 30% test!
```

[18]

```
#Below is the training data (70%)
training_data.head()
```

Visualize

	DOMAIN object	STATE object	ZIPCODE int64	TRN_AMT float64	TOTAL_TRN_AMT f...	TRN_TYPE object
62239	WKHWUSK.NET	ROB	452	49.95	49.95	LEGIT
20754	VGTPKPNI.COM	AO	675	38.85	38.85	LEGIT
72564	QAXPGCXPTV.COM	KO	650	12.95	12.95	LEGIT
88594	TMA.COM	AR	649	38.85	38.85	LEGIT
20337	TMA.COM	ROK	655	12.95	12.95	LEGIT

5 rows, showing 10 per page   Page 1 of 1

[19]

```
#Below  is the test data (30%)
test_data.head()
```

Visualize

	DOMAIN object	STATE object	ZIPCODE int64	TRN_AMT float64	TOTAL_TRN_AMT f...	TRN_TYPE object
92009	CYYUTQPG.COM	ROB	453	38.85	38.85	LEGIT
31395	TMA.COM	KR	101	38.85	38.85	LEGIT
53475	NEKSXUK.NET	AR	649	12.95	12.95	LEGIT
85320	CIYORAAVNRXQEY.NET	AR	649	31.08	31.08	LEGIT
42021	TMA.COM	PO	614	38.85	38.85	LEGIT

5 rows, showing 10 per page   Page 1 of 1

Sklearn using Logistic Regression Model: Training the Data

```
X_train = training_data.drop(columns='TRN_TYPE')[['ZIPCODE','TRN_AMT','TOTAL_TRN_AMT']]
y_train = training_data['TRN_TYPE']

X_test = test_data.drop(columns='TRN_TYPE')[['ZIPCODE','TRN_AMT','TOTAL_TRN_AMT']]
y_test = test_data['TRN_TYPE']
```

```
logistic_model = LogisticRegression()
logistic_model.fit(X_train, y_train)
```

```
LogisticRegression()
```

Sklearn using Logistic Regression Model: Predictions

```
pd.concat([pd.Series(y_predicted_logistic).reset_index(), pd.Series(y_test).reset_index()], axis = 1)[[
```

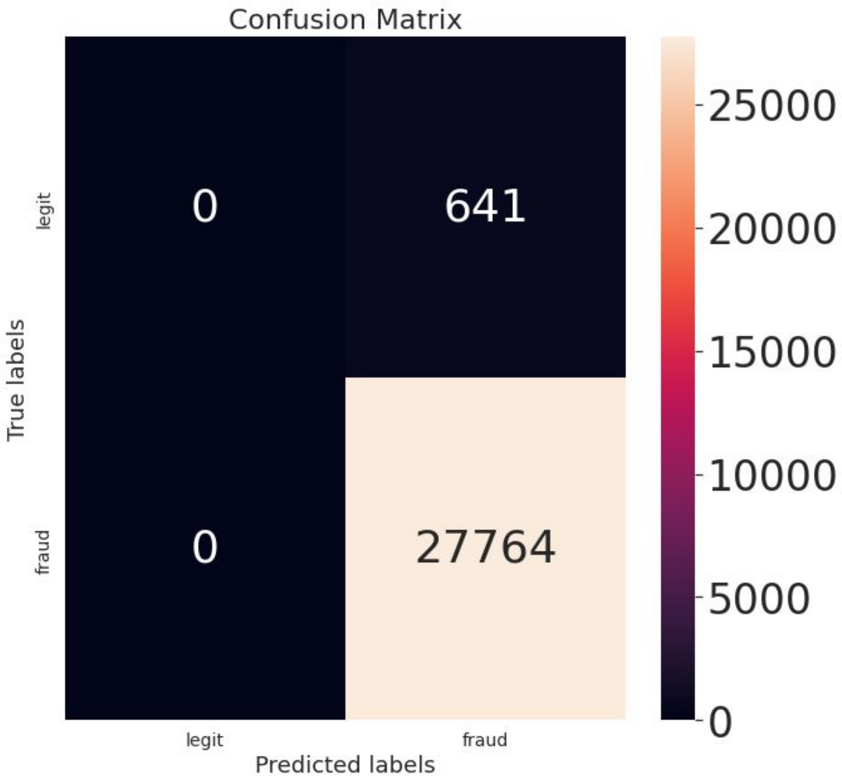
	predicted object	actual object
	LEGIT 100%	LEGIT 97.7% FRAUD 2.3%
0	LEGIT	LEGIT
1	LEGIT	LEGIT
2	LEGIT	LEGIT
3	LEGIT	LEGIT
4	LEGIT	LEGIT
5	LEGIT	LEGIT
6	LEGIT	LEGIT
7	LEGIT	LEGIT
8	LEGIT	LEGIT
9	LEGIT	LEGIT

28405 rows, showing 10 per page — Page 1 of 2841

Confusion Matrix

```
from sklearn import metrics
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted_logistic)
cm
```

```
array([[    0,   641],
       [    0, 27764]])
```



Model Evaluation

```
Accuracy = (cm.item(0) + cm.item(3)) / (sum(cm).item(1))
Precision = cm.item(3) / (cm.item(2) + cm.item(1))
Recall = cm.item(3) / (cm.item(3) + cm.item(2))
f1 = 2 * (Precision * Recall) / (Precision + Recall)
```

```
Accuracy:    0.9774335504312621
Precision:   43.31357254290172
Recall:      1.0
F1:          1.9548671008625242
```

Conclusion

- The logistic regression model seem to predict 100% legit transactions and did not seem to predict any fraudulent transactions
- Out of 28,405 tests, we only misclassify 641
- The accuracy of our prediction is pretty good (97.74%)
- Precision is 43.31%, which is not the worst
- Recall value is 1, which is perfect recall value but we are rather suspicious
- The F1 value is not too bad either

Obstacles We Encountered

- Dataset finding
- Importing dataset
- Learning new libraries
- Topic
- Up-to-date data

Improvements

- Predict using more attributes (e.g. not just zip code and transaction amount)
- Predict using attributes that are logical
- Balanced distribution between two classes
  - Probably would not result in all legit predictions
  - Could perhaps increase precision

Thank You!