

Fraud Detection in Online Market Transactions

Dr. S. Sai Kumar¹, T. Swetha Chowdary², V. Akshara³, N. Akhil⁴,
Sd. Lukhman⁵, V. Kumar Parasuram⁶

Assistant Professor, Department of Information Technology¹

B. Tech Students, Department of Information Technology^{2,3,4,5,6}

Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

Abstract: *E-Commerce transaction process involves multiple entities at different stages such as market place, merchants, payment gateways, financial institutes. Each of them can act as a vulnerability or attack point for Malicious acts. This makes online marketing systems adapt to high-level security and data handling technology solutions like machine learning, deep learning and predictive analytics which are efficient enough to deal with highly sensitive data, predict frauds and unwanted behavioral patterns in this data. Predictive analytics with machine learning is good fraud detection system helps to identify the fraud transaction accurately and should make the detection possible in real time transactions. The techniques have been used to detect whether a transaction is fraudulent or not.*

Keywords: Machine Learning algorithms, Libraries, User Interface, Jupyter Notebook

I. INTRODUCTION

The increasing number of internet users has triggered market players to try opportunities to develop their business through internet media. One method used is to develop an E-Commerce business. Ever since the introduction of credit cards and online payments, many scammers have found ways to exploit people and steal their credit card information to use them for unauthorized purchases. This leads to a huge number of fraudulent purchases every day. E-Commerce websites are trying to identify these fraudulent transactions and stop them from happening again.

II. PROPOSED SYSTEM

The existing systems uses logistic regression and K-nearest neighbor algorithm. We proposed three different algorithms for this model namely Random Forest, Decision Tree and XGB classifier. These algorithms shows the accuracy more precisely than that of the existing models. This model uses pandas and numpy for data analysis. Sklearn library is used for data classification and regression. It uses seaborn and matplotlib libraries for data visualization. It uses imblearn, tkinter for balancing the imbalanced data and for creating user interface, where we give the values to predict whether the transaction is fraud or not, respectively

III. TECHNOLOGIES USED

3.1 Jupyter Notebook

It uses to compile all aspects of a project in one place. Instead of Google colab we used this because it helps to create a user interface, which is required to predict the output in this project. Users can create data visualizations through jupyter notebook. Data visualization is easy in jupyter notebook .

3.2 Python Libraries

Pandas[1] and NumPy[2] are used for data analysis. Seaborn and matplotlib libraries are used for data visualization. Imblearn is used to balance the imbalanced data in the database ,which helps to have an unbiased prediction for the output. Tkinter is used to create a user interface, where the user can give the values to get the output. Sklearn[3] library has been used for simple and effective tool for predictive analysis.

3.3 Dataset

Dataset used is an imbalanced dataset.



3.4 Modules

- Data collection: Data is collected in the form of Excel sheet.
- Data analysis: In this part the total information of the dataset will be known. Through this the relation between each instance, which is noted down as columns, will be known by finding the correlation. In data analysis module we assess the variability where most of the values in the dataset lie.
- Imbalanced data processing: imbalanced dataset is where the target class has uneven distribution of the values or observations. To overcome this we use the imblearn library, which helps to up sample the minorities and down sample the majorities in the imbalanced dataset.
- Prediction: After the data is trained, the data that is to be predicted is given as input X_test and the predicted value is stored in y_pred. If the prediction value is 0 the transaction value is noted as fraud, if it is 1 the transaction is noted as valid[3].

Accuracy: We find the accuracy for each model individually, even by including the existing algorithms. By using the following we find the accuracy which is shown in the output.

his helps to find the accuracy when we select model-0, which is Logistic regression[4].

```
lr_train_acc = accuracy_score (y_train, lr. predict(X_train))
```

```
lr_test_acc = accuracy_score (y_test, y_pred)
```

The following formula shows the accuracy when we select model-1, which is K-nearest neighbor[5]

```
knn_train_acc = accuracy_score (y_train, knn. predict(X_train))
```

```
knn_test_acc = accuracy_score (y_test, y_pred)
```

The above both are existing models. The following shows the accuracy of the proposed models

```
rand_clf_train_acc = accuracy_score (y_train, rand_clf. predict(X_train))
```

```
rand_clf_test_acc = accuracy_score (y_test, y_pred)
```

It shows the accuracy when model-2 is selected, which is Random Forest Classifier[7]

```
dtc_train_acc = accuracy_score (y_train, dtc. predict(X_train))
```

```
dtc_test_acc = accuracy_score (y_test, y_pred)
```

This shows the accuracy when model-3 is selected, which is Decision Tree Classifier[6]

```
xgb_train_acc = accuracy_score (y_train, dtc. predict(X_train))
```

```
xgb_test_acc = accuracy_score (y_test, y_pred)
```

This shows the accuracy when model-4 is selected, which is XGBClassifier[8]

IV. RESULTS

4.1 Loading Dataset

```
In [1]: import pandas as pd
df = pd.read_csv('C:/Users/SRIRAM/OneDrive/Desktop/soumya/Processed_data.csv')

print(df)
df.head(10)
```

Unnamed: 0	user_id	signup_time	purchase_time	purchase_value	\
0	22058	20150224225549	20150418024711	34	
1	333320	20150607203950	20150608013854	16	
2	1359	20150101185244	20150101185245	15	
3	150084	20150428211325	20150504135450	44	
4	221365	20150721070952	20150909184053	39	
...
151107	151107	345170	20150127030334	20150329003047	43
151108	151108	274471	20150515174329	20150526122439	35
151109	151109	368416	20150303230731	20150520070747	40
151110	151110	207709	20150709200607	20150907093446	46
151111	151111	138208	20150610070220	20150721020353	20

source	browser	sex	age	ip_address	class
0	0	0	39	7.327584e+08	0
1	1	0	53	3.503114e+08	0
2	0	1	53	2.621474e+09	1
3	0	2	41	3.840542e+09	0
4	1	2	45	4.155831e+08	0
...
151107	0	0	28	3.451155e+09	1
151108	0	2	32	2.439047e+09	0
151109	0	3	1	2.748471e+09	0
151110	0	0	37	3.601175e+09	0
151111	2	3	38	4.103825e+09	0

Fig.1.1: Loading Dataset



The above figure 1.1 shows the path setting, of where the dataset is located. The figure shows the dataset and it prints the top 10 elements[9].

4.2. Exploratory Data Analysis

```
Datatype of each column
Unnamed: 0      int64
user_id         int64
signup_time     int64
purchase_time   int64
purchase_value  int64
source          int64
browser         int64
sex             int64
age             int64
ip_address      float64
class           int64
dtype: object

displaying the 5 rows from top
Unnamed: 0  user_id  signup_time  purchase_time  purchase_value \
0           0    22058   2015022422549   20150418024711    34
1           1   333320   20150607203950   20150608013854    16
2           2    1359   20150101185244   20150101185245    15
3           3   150084   20150428211325   20150504135450    44
4           4    221365  20150721070952   20150909184053    39

      source  browser  sex  age  ip_address  class
0           0         0    0   39  7.327584e+08  0
1           1         0    1   53  3.503114e+08  0
2           0         1    0   53  2.621474e+09  1
3           0         2    0   41  3.840542e+09  0
4           1         2    0   45  4.155831e+08  0

displaying the 5 rows from bottom
Unnamed: 0  user_id  signup_time  purchase_time  purchase_value \
151107      151107   245170   20150137020324   2015033002047    43
```

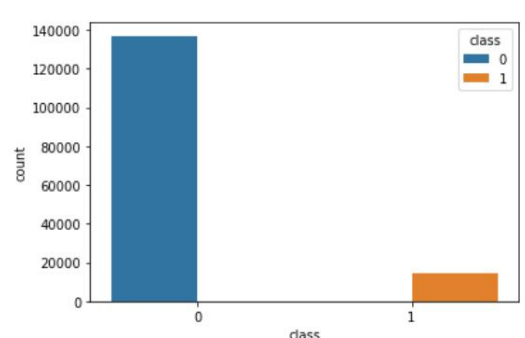
Figure 1.2: Exploratory Data Analysis

The above figure 1.2 shows a part of the output of the data analysis for the whole dataset, like data type of each column, displaying the top and bottom five values of the data set.

4.3 Data Visualization

```
In [9]: In import seaborn as sns
        sns.countplot(x='class', hue='class', data=df)

Out[9]: <AxesSubplot:xlabel='class', ylabel='count'>
```



```
In [10]: In from matplotlib import pyplot as plt
         plt.figure(figsize=(12,12))
         sns.heatmap(df.iloc[:, 7:].corr(), annot=True)
```

Figure 1.3: Data Visualization

The above figure 1.3 shows the graphical relation of the values in the instance class.



4.4 Heatmap Graph

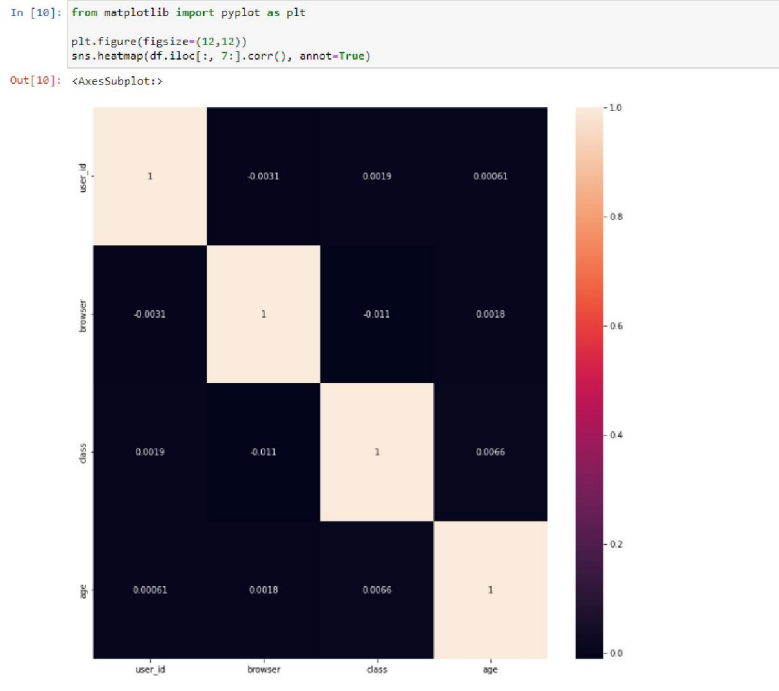


Figure 1.4: Heatmap graph

The above heatmap graph shows the correlation of one instance with another.

4.5 Comparison of Scores Of Different Algorithms



Figure 1.5: Comparisons of scores of different algorithms

The above graphical representation (figure 1.5) shows the comparison of scores of the algorithms used.



4.6 User Interface

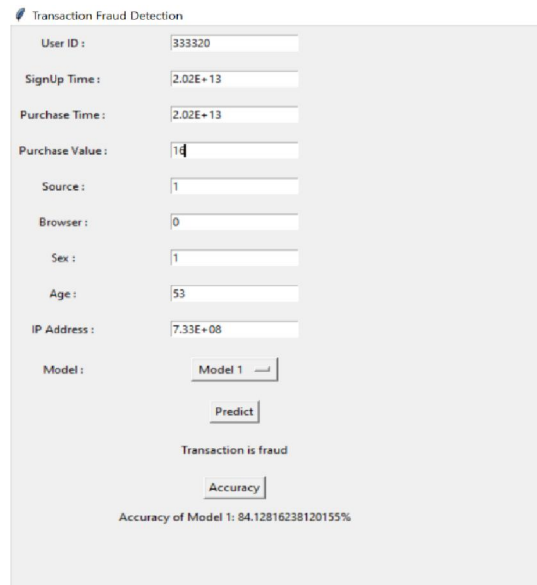


Figure 1.6: User Interface

The above shows the user interface, where the users can enter the values to know whether the transaction is fraud or not and to know its accuracy[10].

V. SCOPE OF FUTURE USE

We have predicted whether the transaction is fraud or not in our project. This project can have further scope in order to mark that website as fraudulent website if it exceeds a certain accuracy percentage so that there are no further transactions carried out in that website. Since we are predicting it based on the IP address whatever transactions are carrying on that particular IP can be halted and safety of the transaction is restored.

VI. CONCLUSION

Therefore, we have predicted if the transaction is fraud or valid and also compared the accuracies of models which we have used thereby finding out the percentage of accuracy on whether the transaction is valid or transaction is fraud.

REFERENCES

- [1]. <https://www.w3schools.com/python/pandas/default.asp>
- [2]. https://www.w3schools.com/python/numpy/numpy_intro.asp
- [3]. <https://www.javatpoint.com/what-is-sklearn-in-python>
- [4]. https://www.w3schools.com/python/python_ml_logistic_regression.asp
- [5]. <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
- [6]. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
- [7]. <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
- [8]. <https://www.geeksforgeeks.org/xgboost/>
- [9]. <http://localhost:8888/notebooks/Mini%20Project.ipynb>
- [10]. <https://www.geeksforgeeks.org/python-gui-tkinter/>