|  |
| --- |
|  |
| Credit card fraud app |
| By Sambit Nath |
|  |
|  |
| **11/7/2019** |

|  |
| --- |
| The intention of this project is to build a simple API that is in the form of a web application which will predict whether a particular credit card transaction is fraud or non fraud. The UI takes various parameters from the user such as time of transaction, amount and various other unnamed parameters which are not disclosed because of security reasons. There is an additional feature of this application which sends an automated mail to the concerned user if the transaction was a fraud or non fraud one. |

**Prerequisites (python libraries and frameworks):**

Pandas (for Machine Leraning Model), Scikit Learn, Flask (for API), Imblearn (for generating synthetic datapoints)

**Project files involved:**

**credit\_card\_fraud.py** : This file contains the code which fits the input data in a ML algorithm and predicts if the transaction was a fraud or not. The prediction is in the form of ‘0’ or ‘1’ depending on the outcome. The pickle function converts the code into a serialized object.

**flask3.py :** This file contains the code which first imports the serialized object (converted through pickle previously). The input values taken from the user by requests.form and sends the predicted outcome to the user interface using the render\_template function. The code also simultaneously sends an alert mail to the customer if the transaction was a fraud or non fraud.

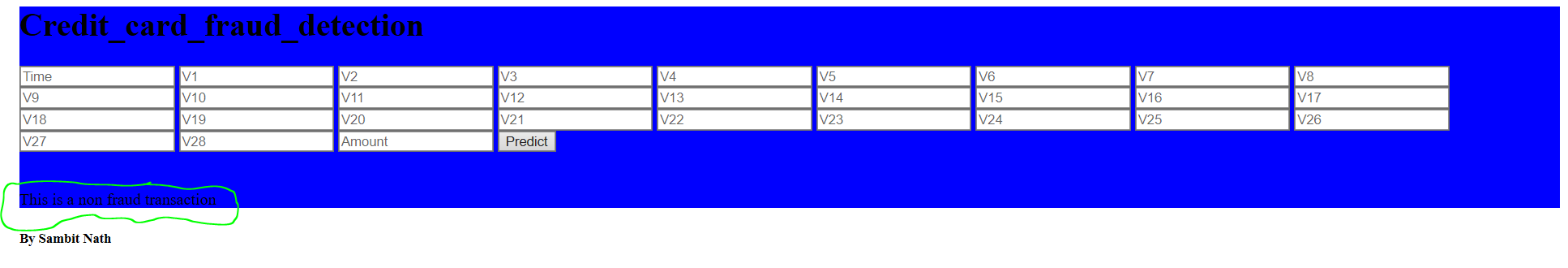
**UI.html:** This file is the frontend html page which acts as an interface to the user where the input parameters are taken and the predicted result is displayed.

**How does the API work?**

1. The login page of the API where the user inputs the details of the employee. The screenshot of the page is as shown below:



1. The inputs are written and the user clicks on the submit button. The algorithm present in the credit\_card\_fraud.py file predicts the output and presents to the user. The screenshot is shown below:

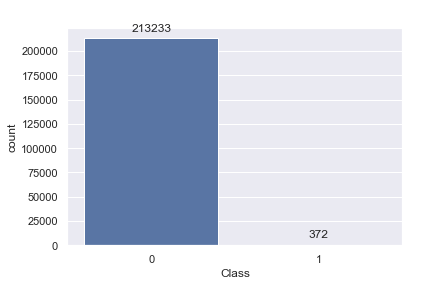


**Note**: The input values need to be standardized or normalized before entering into the application. This is because the model was trained on a set of feature scaled values.

1. The flask3.py file also sends a mail to the customer if the transaction was fraud or non fraud. We need to first mention the mail server, the port number and the email credentials in the main code present in the file. The mail would be sent when the code runs.

**Data preprocessing, visualization and model creation**

1. The data is comprised of a csv file of which consists of the transactional information of 284807 cases containing 30 attributes. Out of these attributes only 2 of them are known and the rest are not disclosed because of security reasons.
2. We import the dataset and check for any null values and outliers. None are present in this case.
3. We check if the dataset is balanced or not. It was found that the data was highly imbalanced with the minority class belonging to the fraud cases. We fixed it by generating synthetic data points using SMOTE and framed new train and test sets accordingly.



1. Now we apply feature scaling to the train and test dataframes.
2. We fit the data into three classification algorithms viz. Naïve Bayes algorithm, Logistic regression and Decision tree classifiers and find their accuracies from the confusion matrix and also their cross validation scores.
3. We find the Decision tree to be the most perfect model with the cross validated accuracy mean to be 99.77 %. If it was found to be low then we would have tuned the hyparameters using techniques such as grid search.

**Working structure**

The ‘credit\_card\_fraud.py’ file is first run. In this file the data is preprocessed and the model is trained and finally the accuracy is checked and model validation is done. The file is then converted into a serialized object by using the ‘pickle’ function.

The ‘flask3.py’ file consists of the end point URL where the API is being hosted. This is done by using the flask micro framework of python. This file also performs the function of sending a mail to the concerned user informing if the transaction was a fraud or non fraud. The sender’s and the reciever’s email address, the email server and port need to be mentioned.

Now when we need to run the application we first run the file ‘credit\_card\_fraud.py’ and go to the URL [http://localhost:5000](http://localhost:5000/) which runs the application on port 5000.

flask3.py (takes in data from the user through requests.form and sends the predicted case of transaction to the user.

credit\_card\_fraud.py (sends serialized object by using pickle function)

UI.html (the user interface of the API)