Fraud Transaction Detection

Detailed Project Report

By: Mukesh Kumar

Data scientist intern at ineuron.ai

Introduction

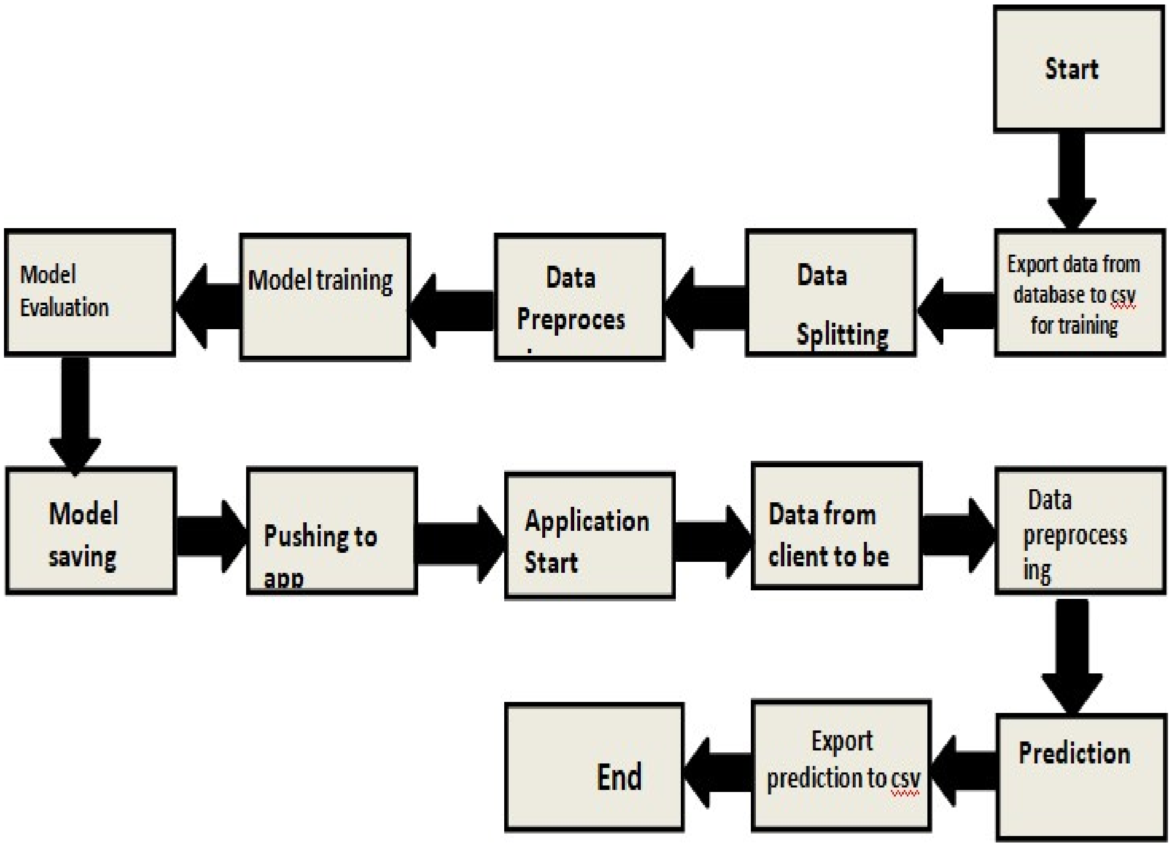
A fraudulent transaction is the unauthorized use of an individual’s accounts or payment information. Fraudulent transactions can result in the victim’s loss of funds, personal property, or personal information.

Fraudulent transaction is the one of the most serious threats to online security nowadays. Artificial Intelligence is vital for financial risk control in cloud environment. Many studies had attempted to explore methods for online transaction fraud detection; however, the existing methods are not sufficient to conduction detection with high precision. In this chapter, we propose a Deep-forest based approach for online transactions fraud detection, which integrates differentiation feature generation method and deep-forest based model. As a single-time transaction's information, which does not contain information such as the user's behavior, is not sufficient for detecting fraudulent transaction, we introduce a transaction time-based differentiation feature generation method into our scheme. *Individual Credibility Degree (ICD)* and *Group Anomaly Degree (GAD)*, which are based on transaction time, are derived to distinguish between legal and fraudulent transaction. Furthermore, to deal with the extreme imbalance of online transactions, we apply Deep-forest algorithm to detect fraudulent transactions. While raw deep-forest model could ignore the outlier transaction samples, we enhance the raw Deep Forest with detection mechanism for outliers, paying more attention on outliers to promote the precision of fraud detection model. Finally, we conduct test using one bank's transaction data. Compared with random Forest-detection model, our method improves precision rate by 15% and recall rate by 20%.

Objective

The main goal is to predict which transaction is a fraud transaction & which Is not. The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing.

Architecture



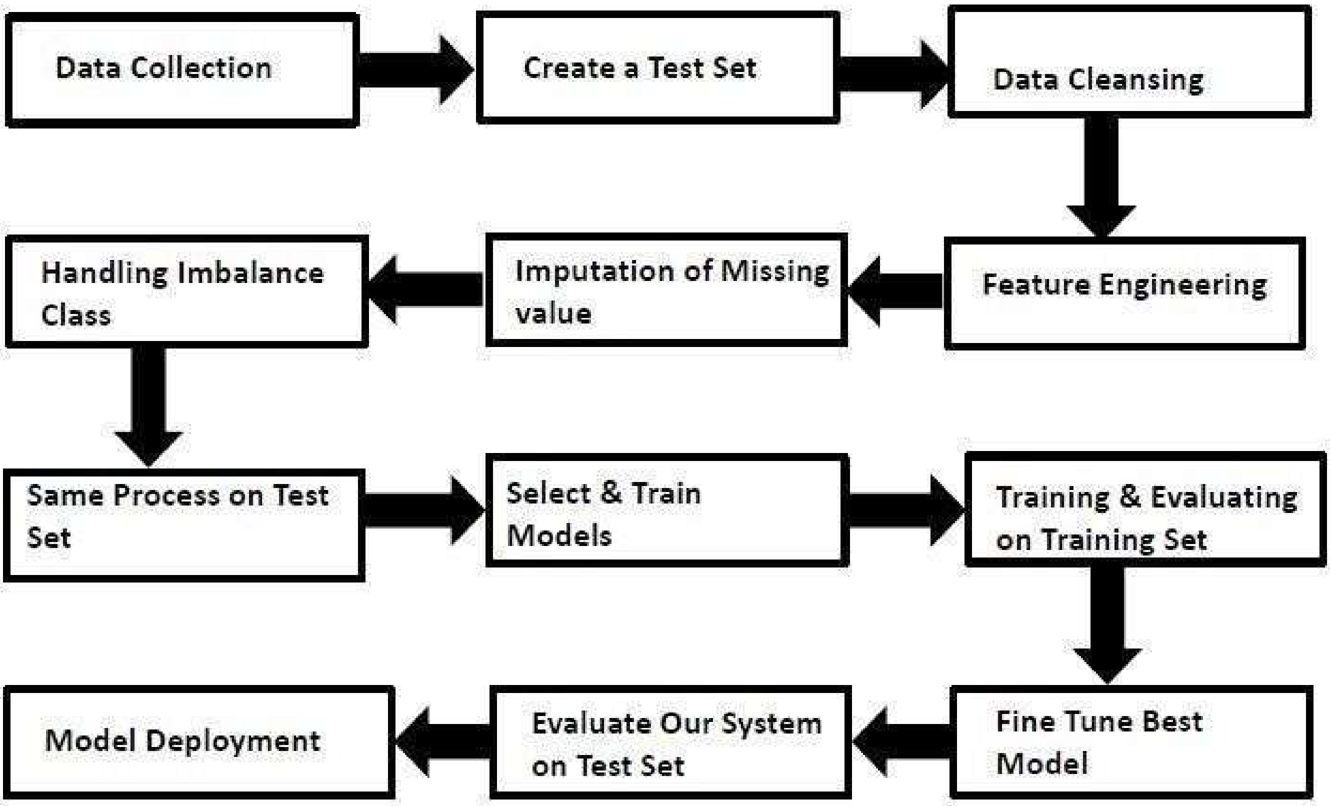
DATASET

- The dataset contains transactions made by credit cards in September 2013 by European cardholders.
- This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

- It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

- Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

Model architecture and workflow



Data Collection

- Fraud transaction Data Set from Kaggle Repository
- For Data Set: Credit Card Fraud Detection | Kaggle

Model training and evaluation

Classification algorithm - SVC is tested since it given better result and was chosen for model trainingand testing.
Model performance evaluated based on accuracy, classification report.

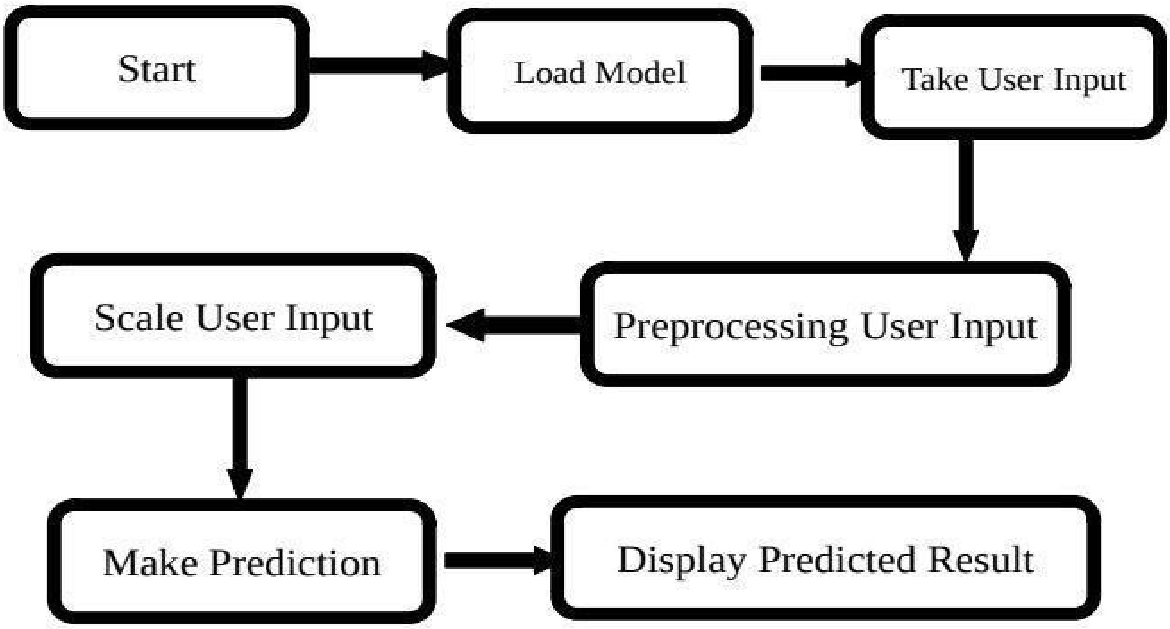
Work Flow

- **Data Description**: We will be using Fraud transaction Data Set present in Kaggle Repository. This Data set is satisfyingour data requirement. Total 284808 instances present in different batches of data.
- **Export data from database to CSV for training**: Here we will be exporting all batches of data from database into one csv file for training.
- **Data Splitting**: We split the data here for our train and test data for further uses.
- **Data Preprocessing**: We will be exploring our data set here and perform data preprocessing depending on the data set. We first explore our data set in Jupyter Notebook and decide what pre-processing and validation we must convert all those to numerical values by label encoding and then we must write separate modules according to our analysis, so that we can implement that for training as well as prediction data.
- **Model Training**: We trained various model in our notebook and SVC was good on it. We trained with our processed data.

- **Model Saving:** We will save our models so that we can use them for prediction purpose.

- **Push to app:** Here we will do cloud setup for model deployment. We also create our streamlit app and user interfaceand integrate our model with streamlit app and UI.
- **Data from client side to prediction purpose**: Now our application on cloud is ready for doing prediction. The prediction data which we receive fromclient side.
- **Data Preprocessing**: Client data will also go along the same process Data pre-processing and according to that we willpredict those data.
- **Export prediction to CSV**: Finally, when we get all the prediction for client data, then our final task is to export prediction to csv file and handover it to client.

Model Deployment



- Finally, this model is deployed on Heroku using Streamlit framework.