Capstone Project Report

Credit Card Fraud Detection

Aditya Dewan

Contents:

Introduction

Credit Card fraud detection is a very important aspect of any financial organisation. Detecting and predicting Credit Card frauds is important in order to prevent major financial losses to customers and financial institutions. There are several ways to detect and predict Credit Card frauds, but the most popular ones include Machine Learning models. ML models are able to detect and predict Credit Card frauds by analysing past data when a transaction occurred and making predictions based on that data.

Analysing the already available data properly is important for any ML model. The transaction details when the fraud happened in the past can give us a robust mechanism to predict and prevent it from happening in the future.

For this particular project most of the data available to us was anonymised, presumably to hide confidential user information. This can result in certain problems such as difficulties in analysing results and understanding and interpreting results. However, many Machine Learning models are robust enough to work with anonymised data. In this project we look at four Machine Learning models and try and see which of them work best with our dataset.

Problem Statement

The primary problem statement is as follows: Correctly identifying a fraudulent transaction based on the available features. This is a classification problem wherein we need to decide whether or not a transaction is fraudulent or legitimate based on the transaction features provided to us.

Machine Learning Modelling pipeline

The basic pipeline that was followed in creating the different models was as follows:

- The first step of an ML pipeline is Exploratory Data Analysis. This involves understanding the structure of the data, and its different features, identifying patterns, detecting anomalies, and summarizing key characteristics. It also involves creating visualisations in order to get a better understanding of the data. Another important aspect of EDA involves mathematical/statistical evaluation of the data.
- Data Cleaning and Transformation: Data cleaning is the process of transforming data ensuring that your data is accurate, consistent, and ready for analysis or modelling. This involves handling any missing values in the data, duplicates and outliers.
- Modelling: After data has been cleaned and transformed, it is ready for creating the Machine Learning model. It is important to select an appropriate ML model based on the problem. The following ML models were implemented in the project:
    - Logistic Regression: It is a basic probabilistic model which was used as a starting model to understand how the data performs. This model however struggles

against imbalanced data. Some hyperparameters were tuned in order to get the best possible evaluation metrics, however there are better ML models with more enhanced performance.

- Support Vector Machines: The next model which was implemented was the Support Vector Machine which is a distance-based model. SVM is a distance-based algorithm, as its decision-making process relies on finding the optimal hyperplane that maximizes the margin (distance) between classes. In a 2D space, this hyperplane is a line; in a 3D space, it's a plane, and in higher dimensions, it's a hyperplane. SVM maximizes the margin, which is the distance between the hyperplane and the nearest data points from both classes. An important part of classification using SVM, is selecting an appropriate kernel which is a mathematical function used to transform the input data into a higher-dimensional space where a linear hyperplane can better separate the classes. Some common kernels used in SVM include linear, polynomial, radial basis function etc. For our project we use a linear and a polynomial kernel, specified by the kernel parameter in the SVC method. More complex kernels tend to perform better when a more complex relationship is involved. A regularization parameter (C) is also specified. A lower value of C is used which helps in lowering the chances of overfitting.
  However, SVM, like Logistic Regression, struggles against highly imbalanced data. This was evident in the results obtained.
- Random Forest: The next model implemented was Random Forest, which is an ensemble tree-based model. Tree-based models typically perform better than Logistic Regression and SVM. This is because its structure and inherent properties make it robust and effective for handling the challenges posed by class imbalance. Random Forest is an ensemble learning technique combining several individual Decision Trees. Random Forest aggregates predictions from all trees through majority voting (classification). Since each tree is trained on a slightly different subset, the model benefits from diverse perspectives, helping it generalize well even for the minority class.
- XGBoost: The final model implemented was XGBoost (eXtreme Gradient Boosting). XGBoost is again an ensemble tree-based model. XGBoost uses decision trees as its base models. XGBoost builds trees sequentially, where each tree attempts to correct the errors of the previous one. It minimizes a loss function by optimizing tree splits. Predictions from all trees are combined (summed) to make the final prediction. XGBoost builds trees iteratively by using the gradient of the loss function to improve predictions. The contribution of each tree is weighted based on its performance. XGBoost has several mechanisms that help address the issue of high imbalance in the data, which is one of the reasons it performs well in this case.

- After creating the models, evaluation metrics were computed for each model. The primary metric considered was F1-score. The reasoning for the same in the next section.

- After the most well performing model was found, it was saved as a pickle file for deployment.

Evaluation Metrics

There are several methods to evaluate the performance of a model. Some of these include accuracy, precision, recall, etc. However, it is important to choose the correct evaluation metric for the model. Choosing the evaluation metric depends on a number of factors, including the type and nature of the problem, and the threshold tolerance. For a problem like credit card fraud prediction, it is important to correctly identify frauds and catch them. Missing frauds or misclassifying them can be a big financial problem. Hence selecting an appropriate evaluation metric is essential to correctly estimate and evaluate the performance of the model. The following are the most common evaluation metrics and their meaning in context to credit card fraud prediction.

- Accuracy: Accuracy denotes the models' ability to correctly predict an observation (fraud or non-fraud). It is not such a good metric in the context of fraud prediction as it also gives a major weightage to the non-fraud records, which might skew the results to a high value due to the inherent imbalance in the data, and this might not be necessarily indicative of a good classification model and can be misleading.
- Precision: Precision tells us the ratio of the correctly predicted frauds, to the total predicted frauds. This is a good metric in the context of fraud prediction because it quantifies the rate of correct prediction.
- Recall: Recall is the ratio of the correctly predicted frauds to the actual frauds. It is again a good metric for this particular context.
- Confusion Matrix: It is a 2x2 which indicates the following values: True Positive (non-fraud detected as non-fraud), False Positive (non-fraud detected as fraud - type-1 error), False Negative (fraud detected as non-fraud - type-2 error), True Negative (fraud detected as fraud). The matrix is represented as follows:

  $$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

  The Confusion Matrix gives a good rounded numerical visualisation of how the model is performing and how many data points it is able to correctly classify.
- F1-score: F1-score is the Harmonic mean of the precision and recall. It combines both precision and recall and gives us a well-rounded evaluation metric for fraud-prediction. **F1-score is the metric which would be used to compare the different models in this project, though the previously mentioned metrics would also be computed.**

Results

The F1-score of the different models implemented in the project are given below:

S.No.	Model	F1-score
1.	Logistic Regression	0.77
2.(a)	Support Vector Machines: linear kernel	0.25
2.(b)	Support Vector Machines: polynomial kernel	0.65
3.(a)	Random Forest	0.82
3.(b)	Random Forest (hyperparameter tuning)	0.83
4.(a)	XGBoost	0.81
4.(b)	XGBoost (after hyperparameter tuning)	0.88

Conclusions:

This concludes the credit-card fraud prediction project. Some of the important conclusions are as follows:

- For a dataset which is as imbalanced as the one used in this project, it is important to transform the data when using simpler Machine Learning models to ensure that they can perform well.
- Stratifying on the minority class is important when splitting the data to ensure that both the train and test datasets have a representation of the minority class.
- Selecting the appropriate evaluation metric is also essential. A metric like accuracy might yield very high values on the train data; however, this can be very misleading due to the highly imbalanced nature of the data which leads to the majority class dominating the accuracy.
- As a result, metrics like precision, recall and F1-score are more appropriate for such a problem. In this problem, F1-score was employed as the metric of choice as it takes into account both precision and recall and gives a balanced evaluation quantity.
- The model of choice is an important decision. 4 models were used in this problem: Logistic Regression, Support Vector Machines, Random Forest and XGBoost.
- Logistic Regression is a probabilistic model, which is not always apt for such classification problems, especially when the dataset is highly imbalanced, like this case. It can be too simplistic and does not catch the minority class accurately, because it assumes a linear decision boundary.
- Support Vector Machine model is not the best choice for fraud prediction. This was evident from the low precision, recall and F1-score obtained using this model. SVM can also be computationally intensive when the dataset is large or a complex kernel is used.

- Random Forest is a good model for fraud detection because it can handle imbalanced data well. Random Forest, being an ensemble of Decision Trees is able to handle imbalance better than Logistic Regression and SVM.
- XGBoost is another tree-based ensemble model which was used in this project. XGBoost is amongst the best choices for such problems because it can inherently handle imbalance very well. Another advantage of this model is that it does not need too much data transformation and can handle unprocessed data very well through its parameters. However, it can be slightly slower and more computationally intensive compared to other models.
- The choice of model used in the end depends on a number of factors like the specific requirements of the project; whether a high precision/F1-score is more important or interpretability and model simplicity has a higher weightage. If it is the first, XGBoost would be a better choice for this problem, if it is the second, Logistic Regression could be a good choice without compromising too much on the evaluation metrics.