Background

Vesta Corporation, the world's leading payment service company, is seeking the best solutions for fraud prevention. They build fraud prevention systems which save consumers millions of dollars per year. IEEE Computational Intelligence Society (IEEE-CIS) works across a variety of AI and machine learning areas, including deep neural networks, fuzzy systems, evolutionary computation, and swarm intelligence. Today they're partnering with Vesta, seeking the best solutions for the fraud detection industry. Researchers from the IEEE-CIS want to improve fraud detection accuracy and improve customer experiences. If successful, the model will improve the efficiency of fraudulent transaction alerts for millions of people around the world, helping hundreds of thousands of businesses reduce their fraud loss and increase their revenues.

Data Sources

The data came from Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features. The datasets are available in Kaggle.

Exploratory Data Analysis and Visualization

3.24% transactions in the training set were identified as fraud and 96.76% were identified as non-fraud (Figure 1). 1.83% of transactions in the training set were outliers, specifically, 3.31% of fraud transactions were outliers and 1.80% of non-fraud transactions were outliers. When comparing the average transaction amount of fraud and non-fraud data, I noticed that there was no significant difference between the means of fraud and non-fraud transaction amount.

On the product side, 99% of products had Product Code of W or C (Figure 2). Considering the purchaser email domain, I observed that the top three companies were Google (46.78%), Yahoo (21.76%), and Microsoft (12.26%) (Figure 3).

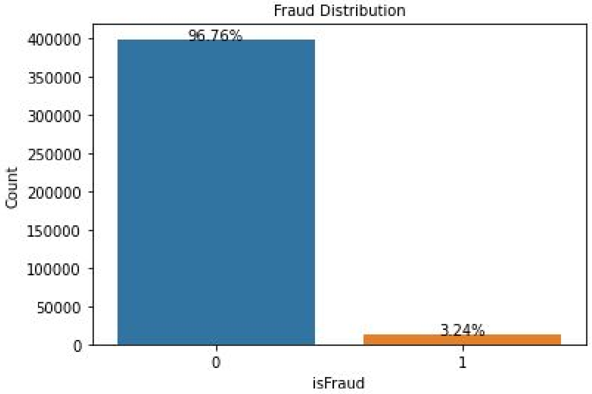


Figure 1: Fraud Distribution

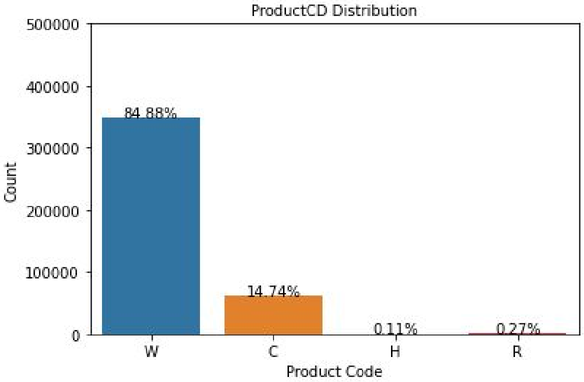


Figure 2: ProductCD Distribution

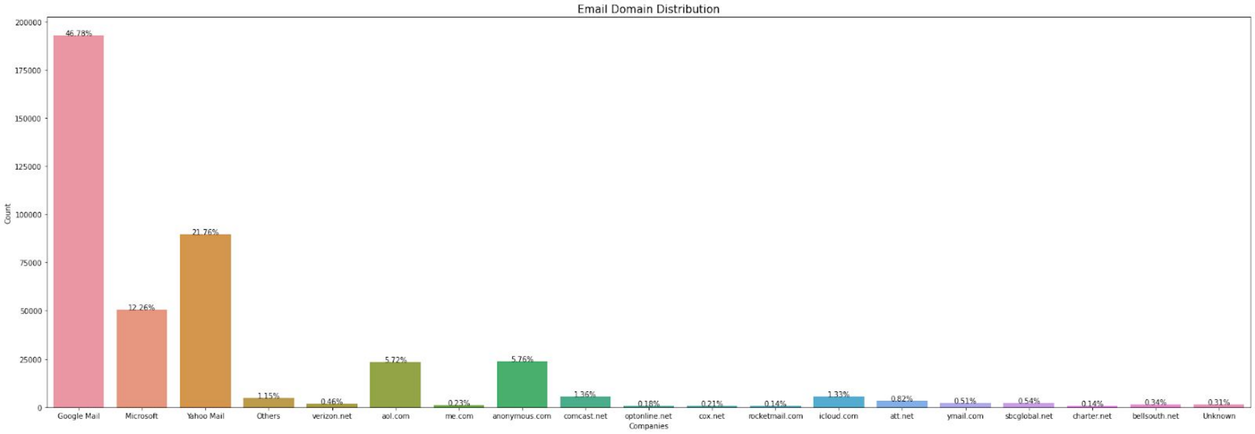


Figure 3: Purchaser Email Domain Distribution

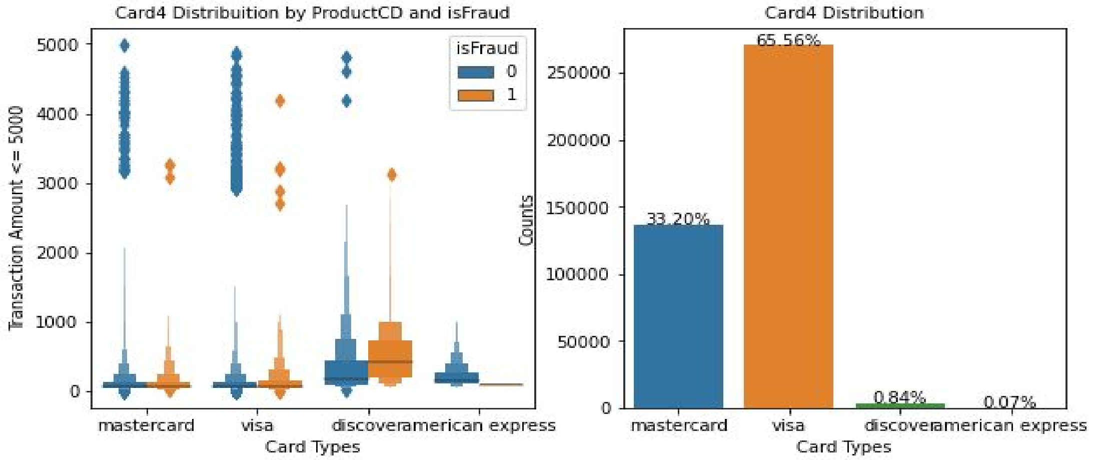


Figure 4: Card Types Distribution (1)

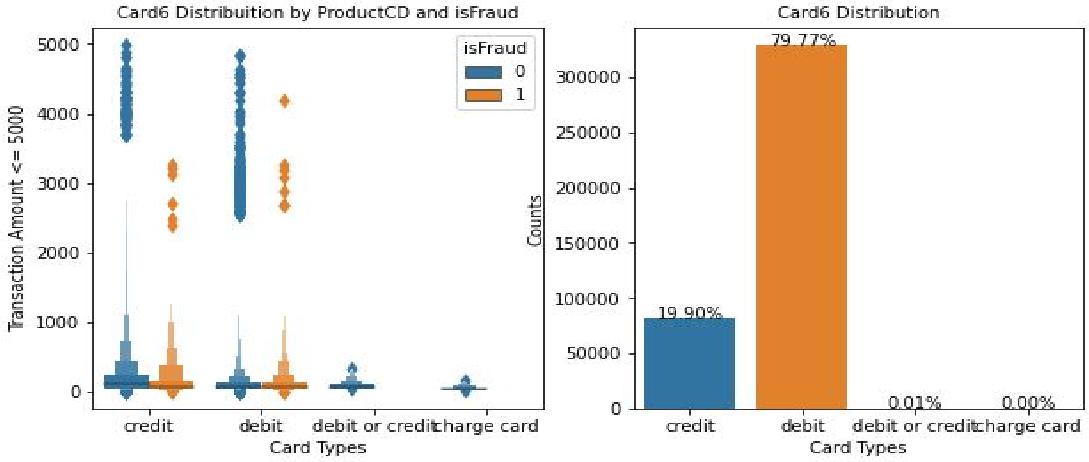
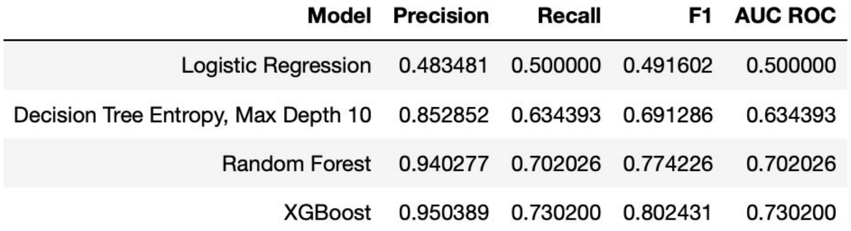


Figure 5: Card Types Distribution (2)

Considering the payment method, I found out that 33.20% of customers used Mastercard and 65.56% of customers used Visa. For a purchase amount less than $5,000, Discover cards accounted for only 0.90% of all transactions, but it had the highest number of fraudulent transactions (Figure 4). All transactions in the training set were paid by either debit or credit cards. I observed that 79.77% of people used debit cards while 19.90% of people used credit cards. However, there were more fraudulent transactions made by credit card than by debit card (Figure 5).

Model Building and Selection

I chose to work with Python library scikit-learn for training models. I split the data into 60% training, 20% validation, and 20% testing set. I built four models to examine how various features impact fraud detection and predicted the probability that an online transaction is fraudulent. For each model, we did parameter tuning to figure out the best-fit parameters when possible.



Model	Precision	Recall	F1	AUC ROC
Logistic Regression	0.483481	0.500000	0.491602	0.500000
Decision Tree Entropy, Max Depth 10	0.852852	0.634393	0.691286	0.634393
Random Forest	0.940277	0.702026	0.774226	0.702026
XGBoost	0.950389	0.730200	0.802431	0.730200

Table 1: Model Performance Comparison

I then compared the models and selected the optimal one based on the F1 score which is the weighted average of recall and precision. Recall is the percentage of results that are correctly classified. It can be interpreted as the probability of making the right prediction when people pick a random fraud transaction. Precision is the percentage of relevant results. It can be interpreted as the probability that a transaction labelled as fraud is indeed a fraud. Here, F1 scores worked better than accuracy scores because of the existence of imbalanced class distribution. Based on the F1 score, the best model is the XGBoost Algorithm with a score of 0.8024 (Table 1). It's also clear that, in this case, the model with the highest F1 score also scored the highest in precision, recall, and roc attributes. It should be noted that the most accurate model was also the most computationally expensive, taking about 20 minutes to train.

Key Findings

The top 5 most important features based on the XGBoost algorithm were card3, addr2, V47, V55, and V62. However, to protect client confidentiality, the actual meaning of features were masked.

If we applied the XGBoost algorithm to the testing set, we would detect 1038 fraudulent transactions out of 506691 testing transactions. The actual values for fraud transactions were not available, so we were unable to tell whether the prediction was accurate or not.

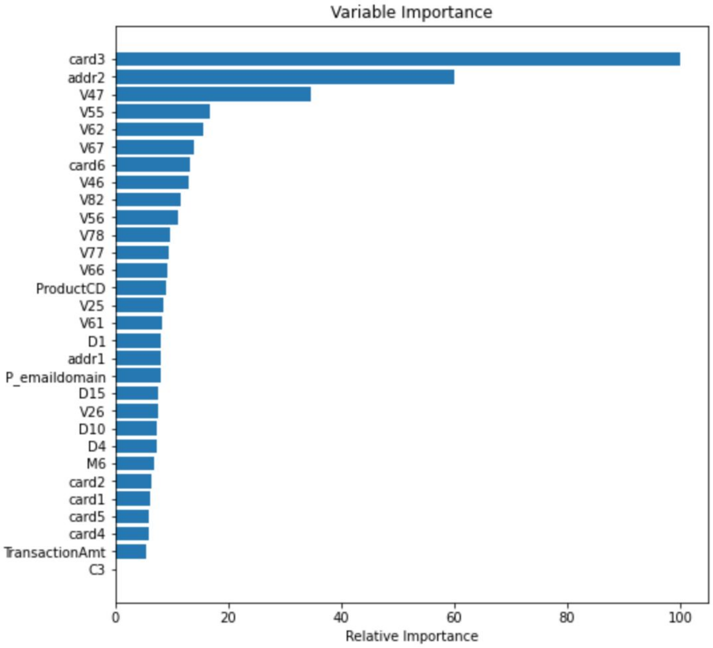


Figure 6: XGBoost Model Feature Importance

Future Improvements

First, due to RAM constraints, the tuning took longer than expected and I was not able to train several algorithms such as KNN and SVM. Even if the tuning were successful, I would not be expecting a huge jump on the model performance.

Second, I used the SMOTE technique to deal with heavily imbalanced datasets. It did not seem to boost the model performance. Other resampling techniques could be used to balance fraud and non-fraud data. Or it would be more helpful but certainly difficult if we could gather more fraud transaction data.

Third, although the model worked well in the training set, it might not perform so well in the testing set. It became a question whether this model could be deployed into production. More advanced data handling and modelling techniques would be encouraged to obtain a more accurate and consistent performance.