IEEE-CIS Fraud Detection

EDA, Feature Engineering, and Predictive Modeling

By Dana DiVincenzo

Introduction to the Dataset

- Researchers from the IEEE Computational Intelligence Society (IEEE-CIS) partnered with Vesta Corporation, one of the world's leading payment service companies, to create a competition in which participants are given data that they can use to predict if an electronic purchase is Fraud or not
- Participants are encouraged to perform exploratory data analysis, feature engineering, modeling, and feature selection
- Participants are provided with a training and a testing dataset, which they can run predictive models on
    - Once the participant has finished their modeling, they can upload a csv file with their predictions to Kaggle, where they can find out how accurate their models were at predicting Fraud

Explaining the Data

- Along with the dependant variable “isFraud”, the datasets contain 433 different features which can be used for prediction
- The dataset is broken down into two categories
  - Transaction data: data about the transaction itself, such as the transaction payment amount
  - Identity data: such as network connection information, or the digital signature (UA/browser/os/version, etc) associated with transactions
- The meaning/ true values of many of the variables are masked or not explained to the participants

EDA and Feature Engineering

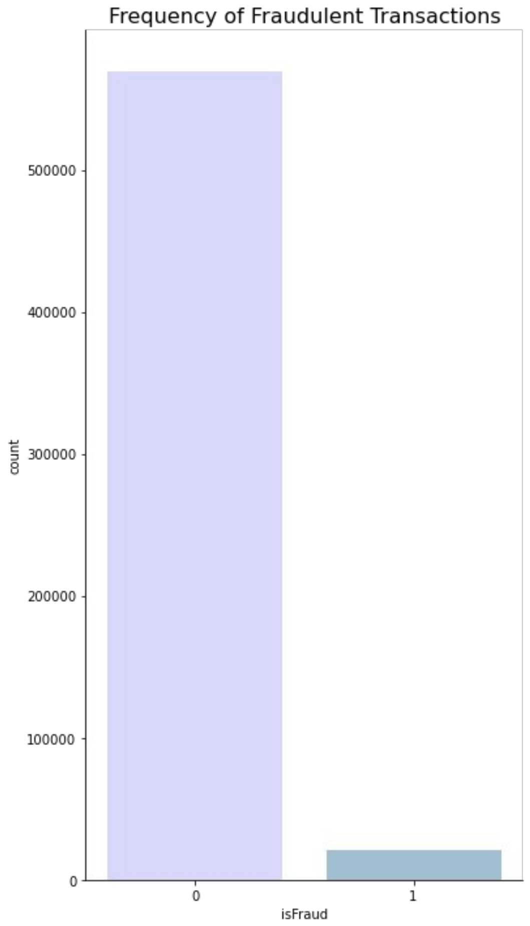
- EDA and feature engineering was performed to prep the training dataset for modeling
- The next few slides will briefly explain the main steps done here

Dropping Features with Excessive Null Values

- Any column that contained more than 50% null values were dropped
- This caused the dataset to drop from 434 columns to 220 columns

Investigating isFraud

- A quick graph was created to see the distribution of isFraud
- The graph shows us that the majority of the purchases described in the training dataset were not fraudulent

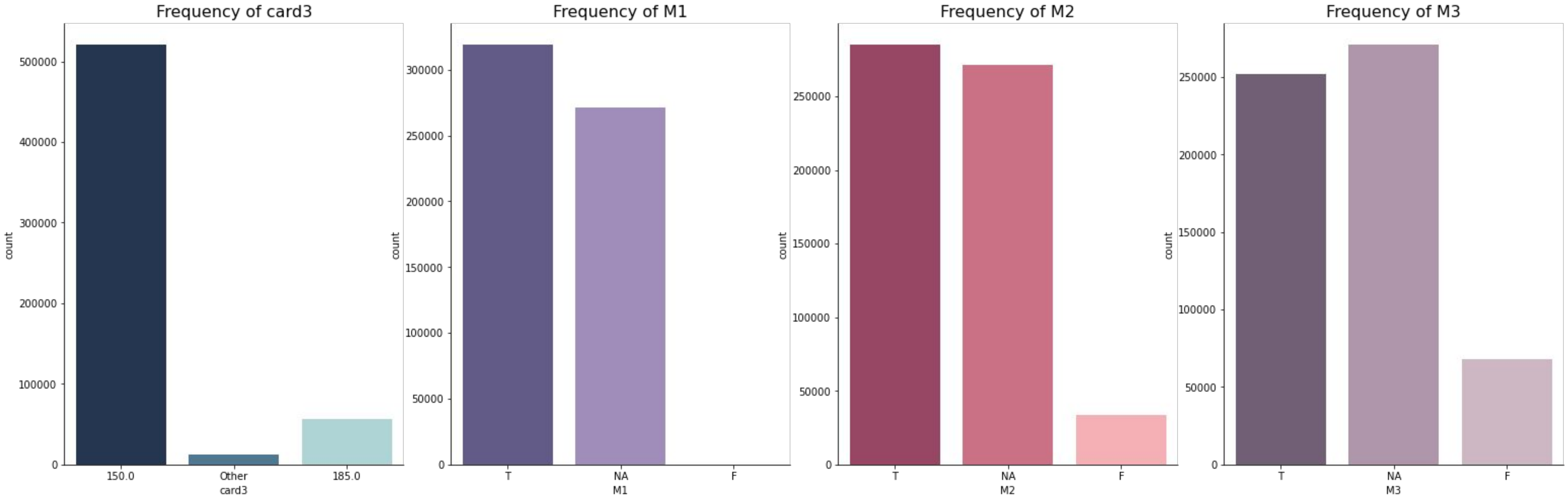


Categorical Variables

- The dataset had a small number of categorical variables
- After dropping the columns with over 50% null values, the categorical variables left were:
  - ProductCD'
  - 'card1', 'card2', 'card3', 'card4', 'card5', 'card6'
  - 'addr1', 'addr2'
  - 'P_emaildomain'
  - 'M1', 'M2', M3', 'M4', 'M6'
- Each categorical variable was investigated with a histogram
  - Those with low variance were dropped from the dataset
  - Those with too many factors were collapsed into fewer levels
  - card1, card2, and addr1 were dropped due to having too many infrequent levels
  - Some were left alone
- Some graphs, along with any edits or necessary description, are on the following slides

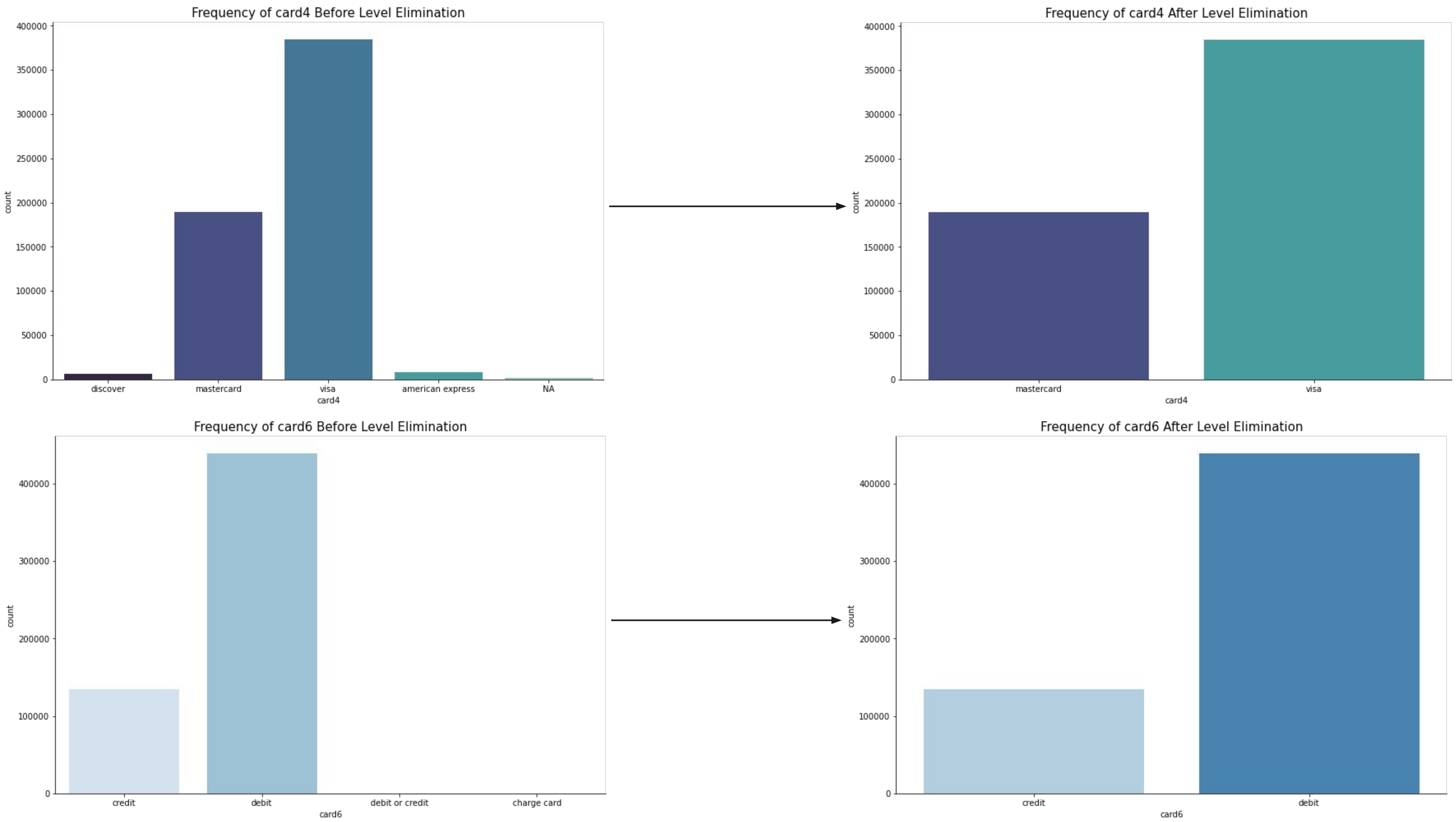
Features with Low Variance

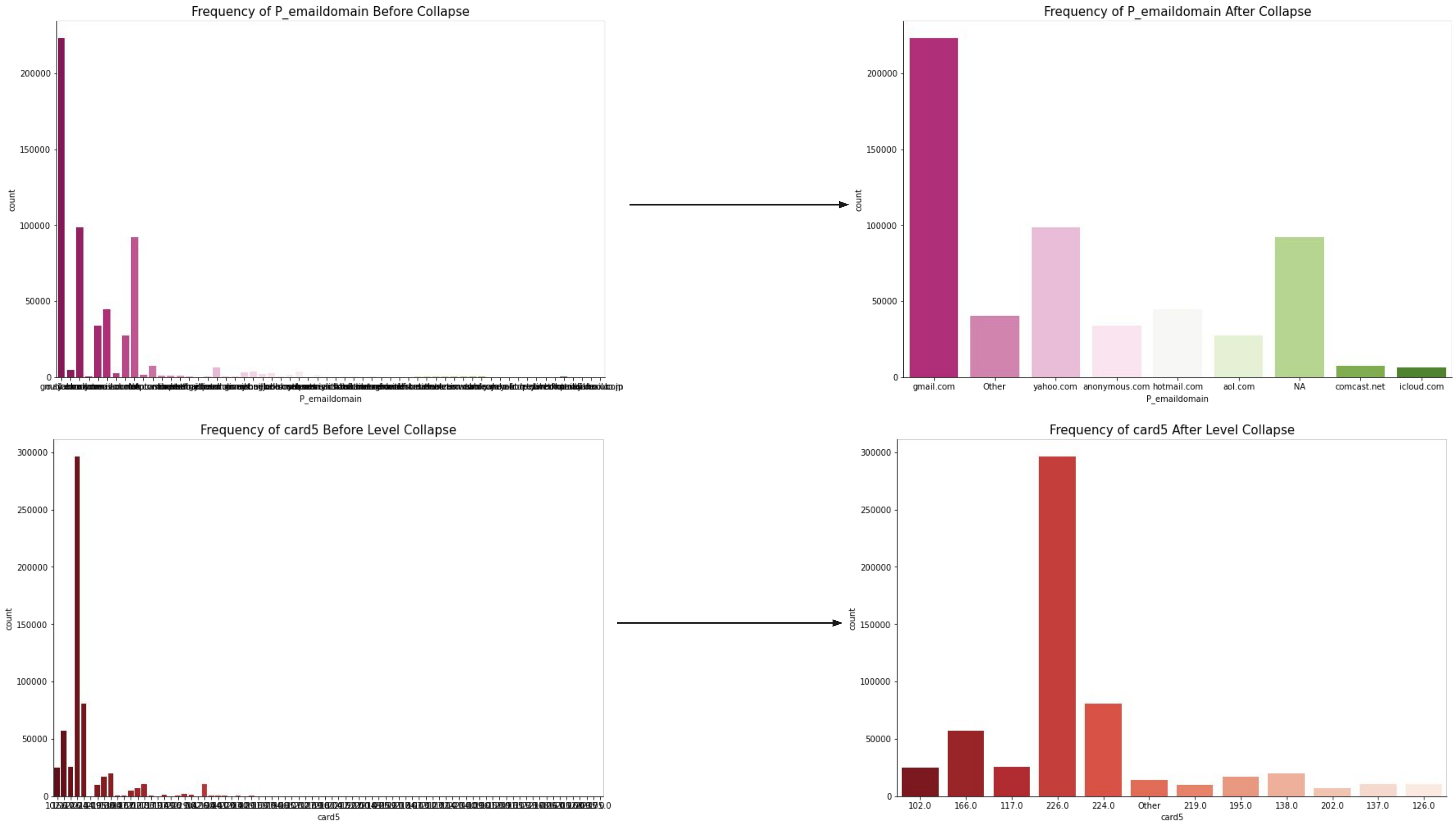
- Card3, M1, M2, and M3 were all dropped due to low variance



Features with Collapsed Levels

- P_emaildomain, card4, card6, and card5 were all manipulated to have a smaller number of levels
  - card4 and card6 had low frequency factors removed
  - P_emaildomain and card5 had the lowest frequency factors combined into a level called “Other”
- Any changes in each level will be shown graphically on the next slides
- Any categorical variables not shown were determined to be acceptable for the model without further manipulation


Frequency of card4 Before Level Elimination
count
card4
discover
mastercard
visa
american express
NA
Frequency of card4 After Level Elimination
count
card4
mastercard
visa
Frequency of card6 Before Level Elimination
count
card6
credit
debit
debit or credit
charge card
Frequency of card6 After Level Elimination
count
card6
credit
debit


Frequency of P_emaildomain Before Collapse
count
0
50000
100000
150000
200000
P_emaildomain
Frequency of P_emaildomain After Collapse
count
0
50000
100000
150000
200000
gmail.com
Other
yahoo.com
anonymous.com
hotmail.com
aol.com
NA
comcast.net
icloud.com
P_emaildomain
Frequency of card5 Before Level Collapse
count
0
50000
100000
150000
200000
250000
300000
card5
Frequency of card5 After Level Collapse
count
0
50000
100000
150000
200000
250000
300000
102.0
166.0
117.0
226.0
224.0
Other
219.0
195.0
138.0
202.0
137.0
126.0
card5

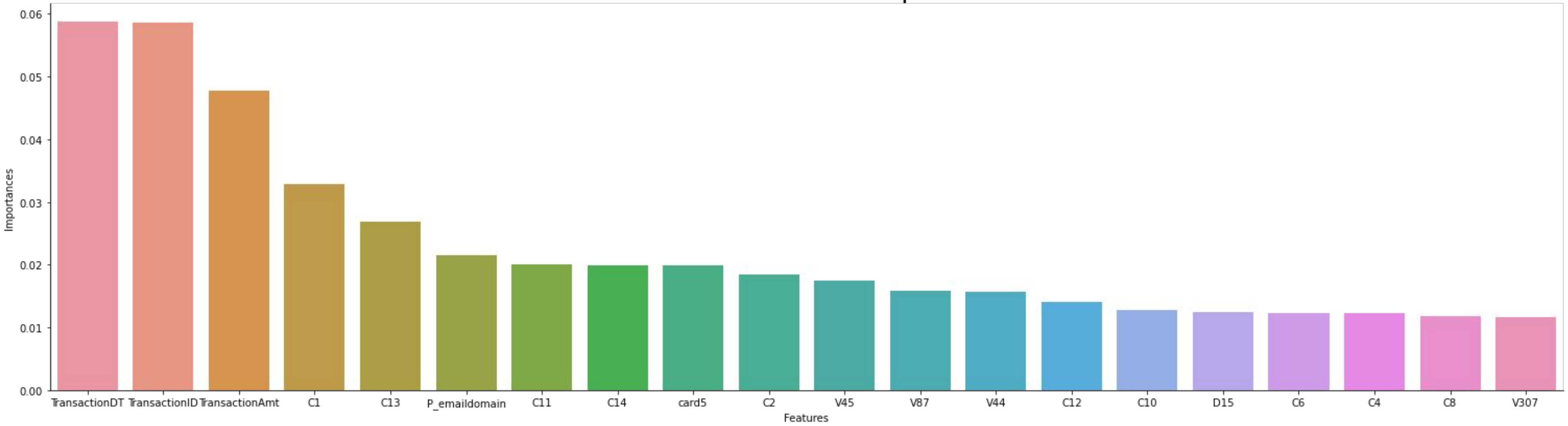
Initial Modeling

- A random 10 percent of the training data was taken to create some initial predictive and classification models
  - This was done in order to keep processing time low, as the dataset was very large
- Of this data, a random 90 percent was used as training data, and the last 10 percent was used as testing data
- A regression model was made with this data and had an accuracy of 96.28%
  - This accuracy is very high
  - It is possible that this is because of overfitting and will not translate to the full test dataset

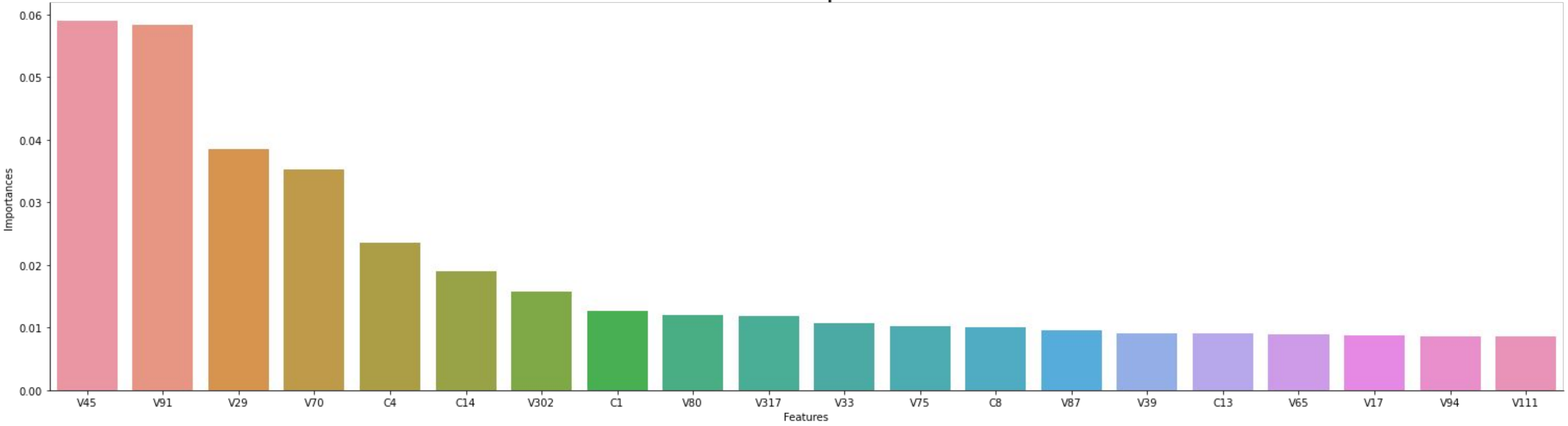
Modeling Continued

- An XGBClassifier and a Random Forest Classifier were used to find the top 20 most important features, given the slice of the training dataset
  - These are the features that are determined to have the most power in predicting isFraud
- The graphs showing the top 20 features are shown on the next slide
  - These are the features that Vesta Corporation would likely be interested in focusing on when trying to reduce fraud
  - However, because most of the features were masked or not explained, it is difficult to talk about what they mean or their significance within any context
- The accuracy of these models were similar to each other
  - Random Forest Classifier score = 97.28%
  - XGBClassifier score = 97.35%
  - Again, this seems strangely high, and I predict that the test dataset will perform differently

Random Forest Classifier: Top 20 Features



XGBClassifier: Top 20 Features

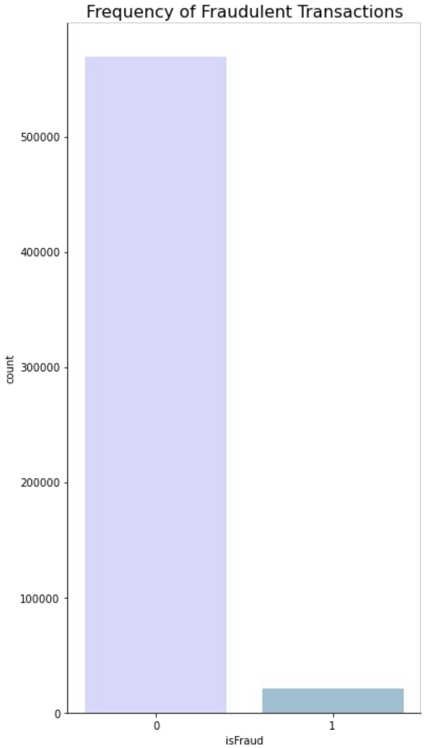


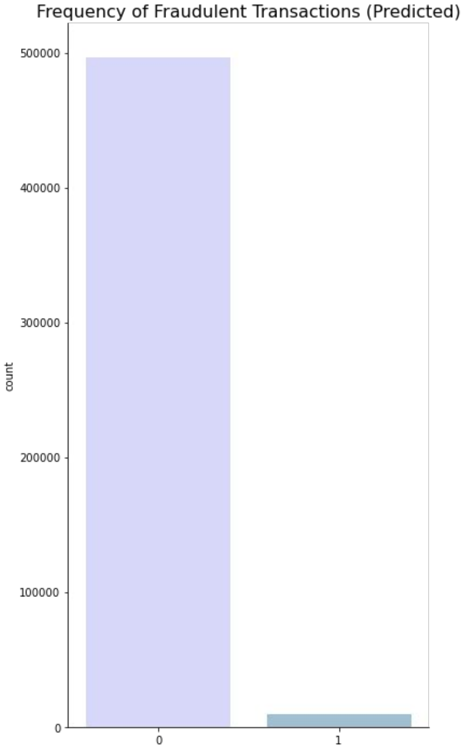
Comparing the Top 20 Features

- Between the two graphs, 7 features were shared
  - C1, C13, C14, V45, V87, C4, C8
  - This is a fairly high number, if we consider the fact that there were 212 features at this point for the model to choose from
- The scales for the graph are also the same
  - This means that the models identified the features as having similar predictive weights
  - If one graph had a larger scale, the features there would be more important
  - The top value on both graphs is .06, which means that even the most important features are not that strong of predictors
- The Random Forest Classifier identified multiple categorical variables as being important, but the XGBClassifier only had numeric variables within its top 20 features

Running the Final Model

- The regression model trained on the training data was applied to the testing data
    - The model predicted whether each transaction in the testing data was fraud or not
- Below is a histogram of the predicted isFraud variable
    - This is compared with isFraud in the testing dataset
    - We can see that the ratio of yes to no is similar between the two graphs, but there is a higher percentage of frauds in the testing dataset





Submission and Results

- When the prediction results were uploaded to kaggle, the accuracy of the predictions were shown to be 51.95%
    - This is significantly lower that how the regression/classification models performed on the testing dataset
    - However, this seems more in line with what I would expect given the complicated nature of the assignment
- Other Kagglers had results of up to 96% accuracy
    - However, many of these top performers used more sophisticated techniques such as:
        - Scaling the data
        - Using other predictive models, or using them on larger training datasets
        - Performing much more intensive EDA and feature engineering to limit the number of features within the model