Problem 4: Predictive Analytics
Sarah Lazio-Maimone

Vesta Corporation is hosting a crowd sourced analytical competition via Kaggle.com. The company is seeking an analytical predictive model using machine learning that will help detect fraudulent transactions. The participants of this competition are given both a training set of data and a testing set of data to gauge the performance of each model. If this analytical task is completed successfully, Vesta will save millions for the world's consumers.

We begin analyzing our data set by performing exploratory data analysis. Figure 1 below shows the probability distribution by transaction amount. It appears that the probability is significantly skewed right, meaning that most of the transactions have a higher probability of being low. The right image in Figure 1 zooms into transaction amounts less than $1,000 for a better view of this probability distribution.

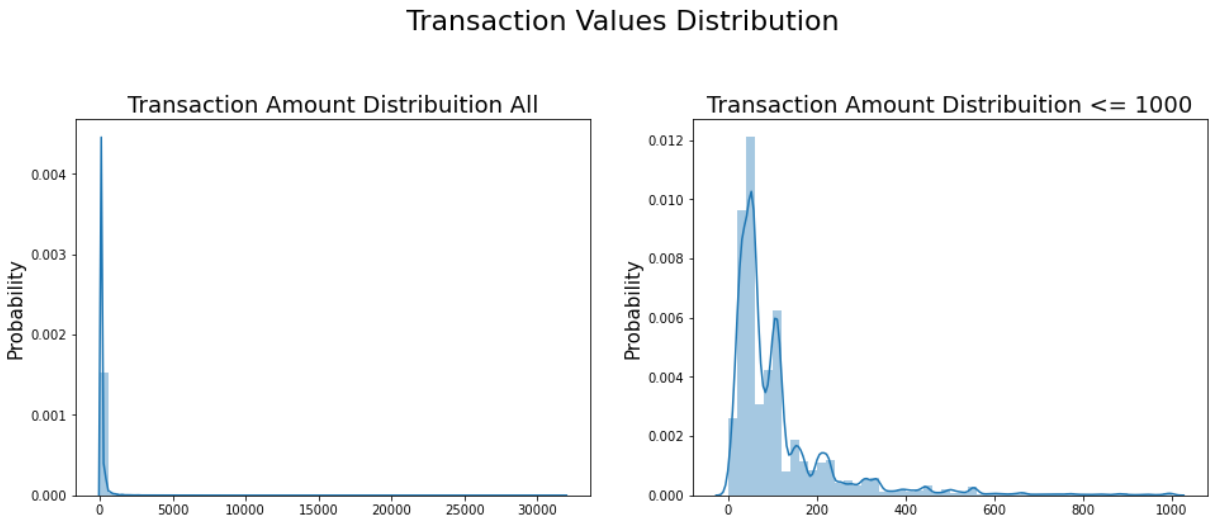

Figure 1

In Figure 2, we see that a over three quarters of transactions in this Vesta data set are for Product W. The next highest Product C makes up 11.6% of transactions. Next, Product R with 6.38% of transactions. Finally, Product H had 5.59% and Product S had 1.97% of transactions in the given data set.

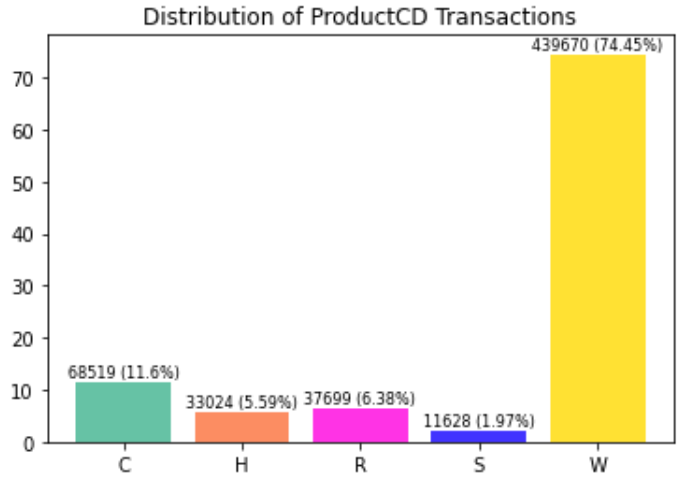

Figure 2

Problem 4: Predictive Analytics
Sarah Lazio-Maimone

Figure 3 below displays the distribution of credit card companies used for each transaction in the main training data set. Visa cards are used predominately at 65.33%. Mastercard is second most used at 32.13%. Both American Express and Discover were used in less than 2% of transactions.

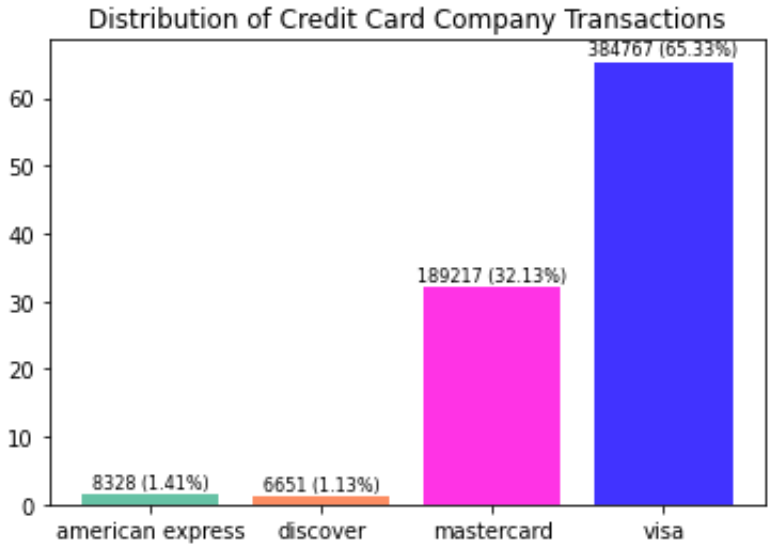

Figure 3

When we break out the frequency of transactions in the Vesta data set, we see that a majority of them are completed using a debit type credit card. Only 25.3% of the transactions used a credit card, verses 74.7% that were debit (Seen in Figure 4).

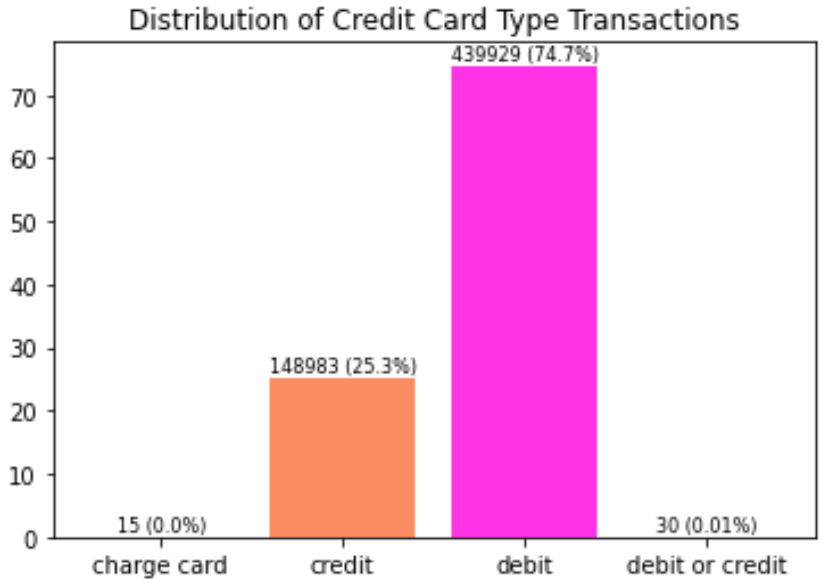

Figure 4

Problem 4: Predictive Analytics
Sarah Lazio-Maimone

For the purpose of this analysis, we want to analyze and understand indicators of fraudulent charges. The following graphs below will break out fraud charges as orange, while blue bars will represent legit transactions. In Figure 5, we see that only 3.5% of transactions in the training dataset are fraudulent. These fake transactions represent about 3.87% of the total dollar value of all transactions.

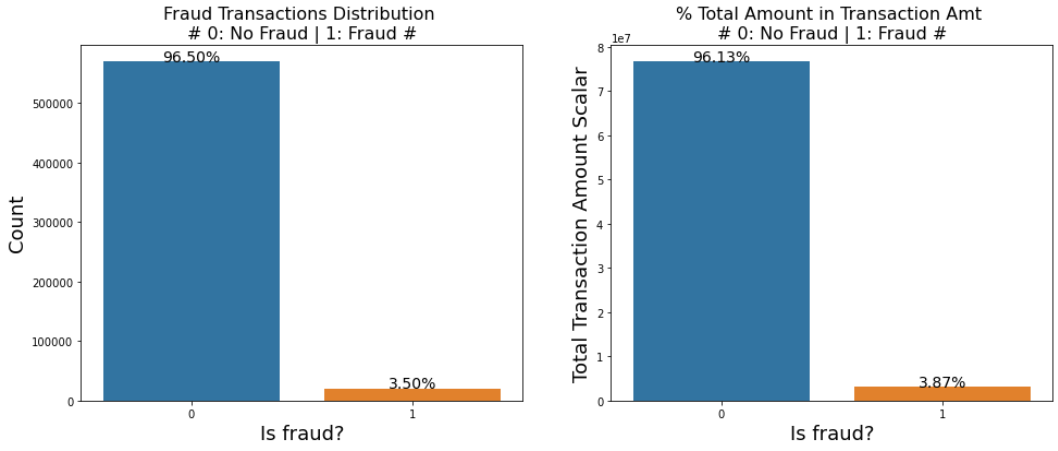

Figure 5

An ECDF, or Empirical Cumulative Distribution Function graph, allows us to visualize the dollar amount distributed in order from least to greatest across the data set. We see again below in Figure 6, when we pull out the Fraud charges in orange, it doesn't represent a large portion of total transaction amount, compared to the blue real transactions.

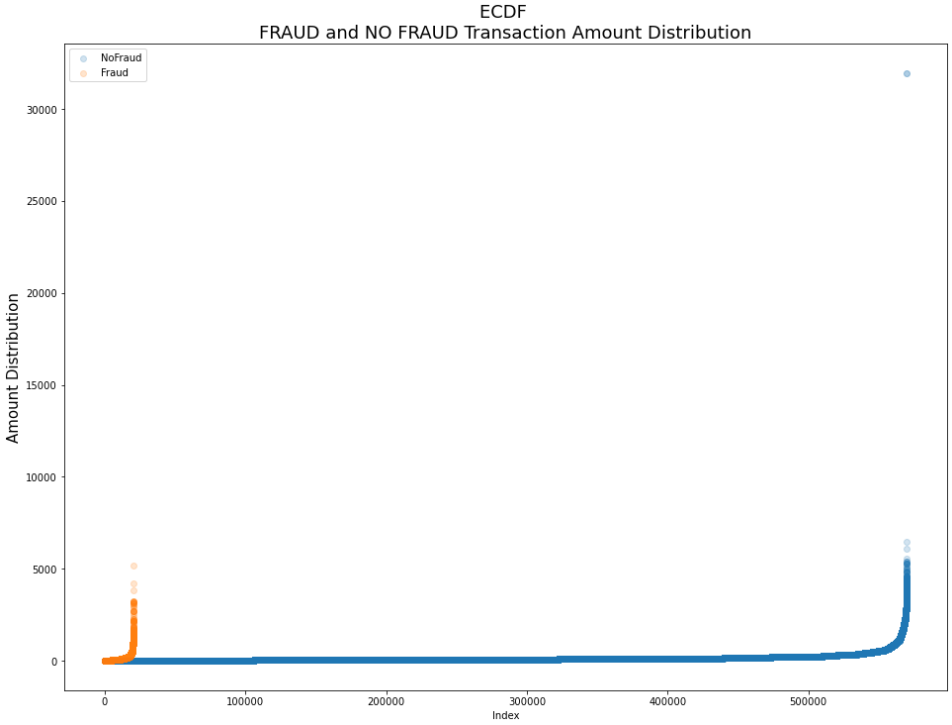

Figure 6

We break out these ECDF plots below in Figure 7, notice the range of values for 'Fraud' verses 'No Fraud'. The fraudulent transaction amounts are between $0 and $20,000, while the real transactions range from $0 and >$500,000. This could be an indicator that fraud transactions tend to be lower, as to not raise a flag among the Vesta Corporation stakeholders.

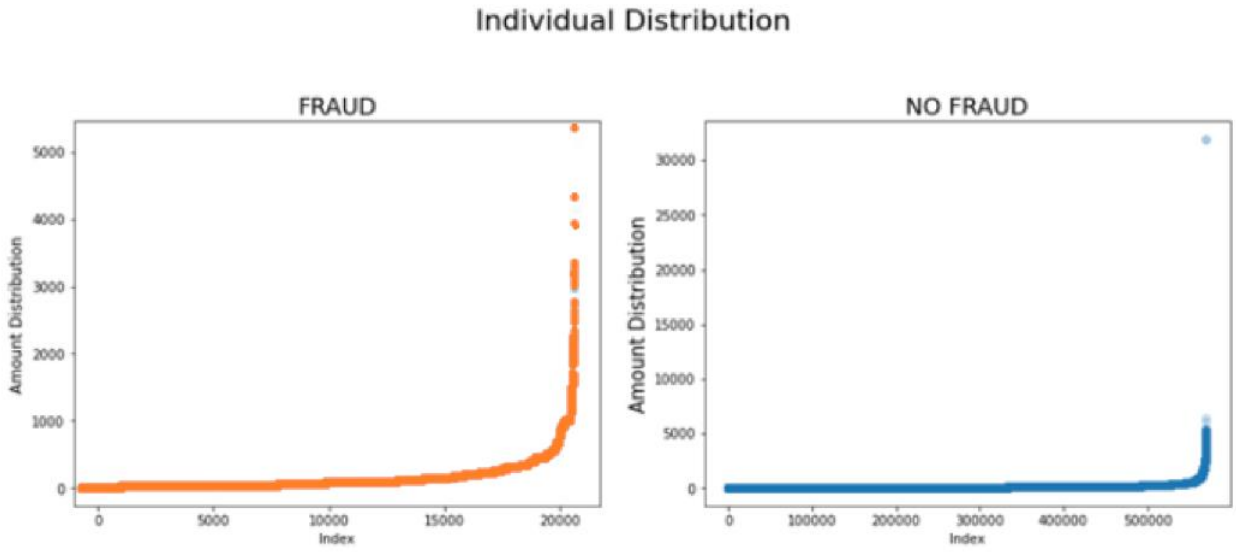

Figure 7

As we saw before, Product W was the most purchased item in this training data set. When we break out Fraud in Figure 8 below, we see that W is still the most prevalent in the fraudulent charges. There is also an interesting indicator with Product C. Even though that product isn't transacted as much in our combined data set, Product C is relatively present among the Fraud transactions.

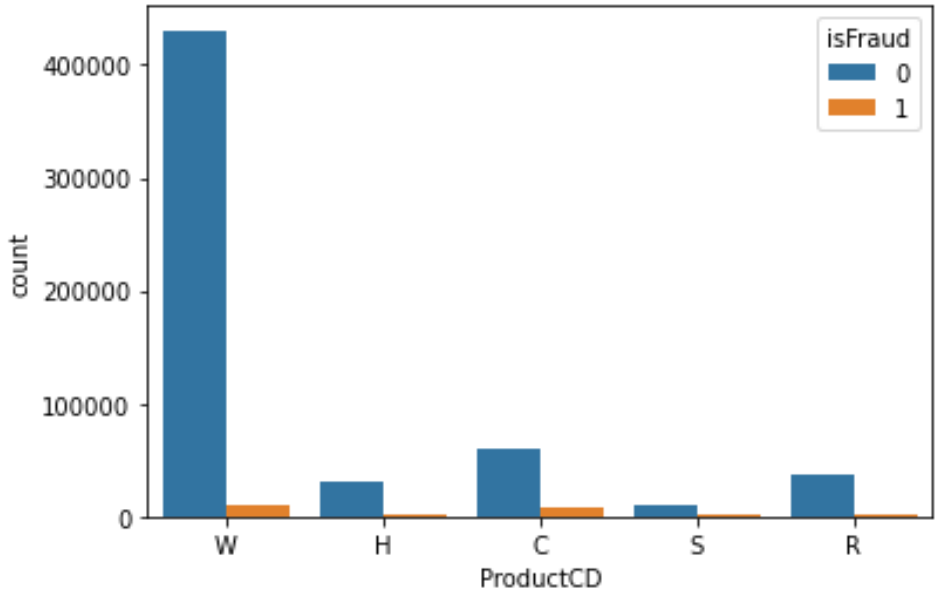

Figure 8

Problem 4: Predictive Analytics
Sarah Lazio-Maimone

When we split the credit card type by fraudulent charges, we see that distribution of fraud transactions align with the full data set. Visa having the most fraud charges and Mastercard having the second most fraudulent charges (seen in Figure 9).

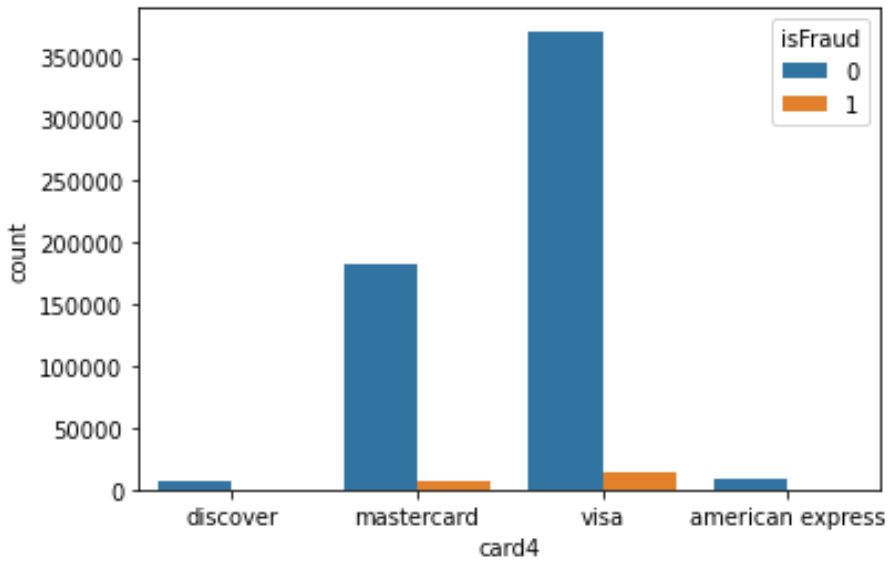

Figure 9

Figure 10 shows fraudulent transactions broken out by credit card type. Below, we see an equal divide of fraud transactions among both credit and debit cards. Since credit card transactions are less occurrent than debit in the full dataset, we can say the fraudulent charges are more present in the credit card charges overall.

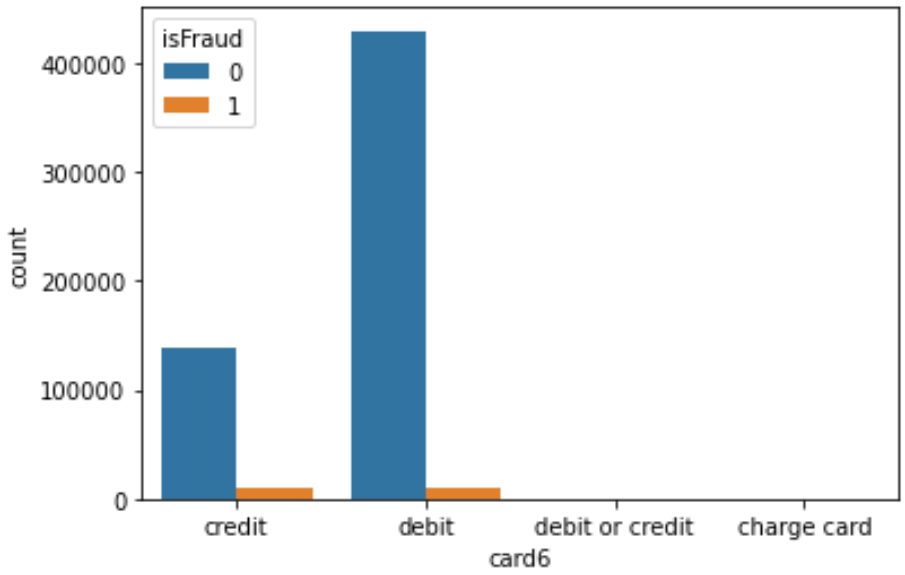

Figure 10

Finally, in Figure 11 below, we again split each device type by fraud and not fraud transactions. Similarly, to the card type graph, there is an equal divide of fraud among both mobile and desktop transactions. Because mobile transactions don't represent the most frequent type, fraudulent transactions are more pronounced in that device type.

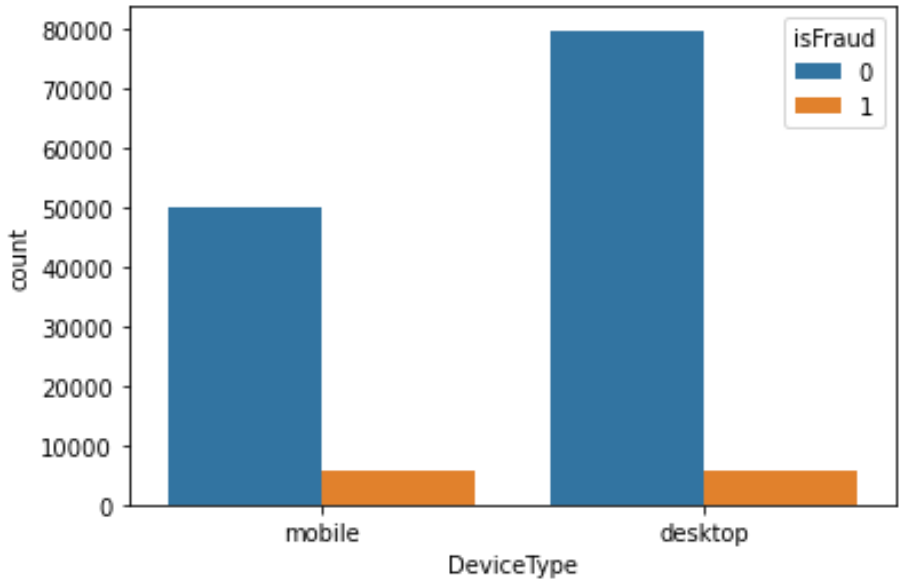

Figure 11

We now begin our machine learning modeling by applying a Random Forest Regression to the given training data set. The Random Forest classifier is a hybrid decision tree that implements many different prediction algorithms at once, then takes the populous decision among them. In Figure 12 below, we have completed the Random Forest and are gauging how well this model performs. This model has a score of 51.6% of variability in fraudulent transactions that can be determined by our prediction variables.

```
clf.score(X,y) #how well did we fit/train
#51.6% of variabilty in fraudulent transactions can be determined by variables
```

```
[Parallel(n_jobs=4)]: Using backend ThreadingBackend with 4 concurrent workers.
[Parallel(n_jobs=4)]: Done  42 tasks      | elapsed:    2.7s
[Parallel(n_jobs=4)]: Done 192 tasks      | elapsed:   10.8s
[Parallel(n_jobs=4)]: Done 400 out of 400 | elapsed:   21.7s finished
0.5169231021374173
```

Figure 12

Problem 4: Predictive Analytics
Sarah Lazio-Maimone

In Figure 13 below, we calculate the R-squared and mean squared error for our Random Forest model. The R-squared in this case is the same as previous score at 51.6% of variance in fraudulent transactions are predicted by our variables. The mean squared error, or average squared difference between our estimated fraudulent charges and the actual fraudulent charges is 1.62%. These numbers signify that our model is decent at predicting fraudulent transactions, given the training data set.

```
# Evaluate model pipeline on test data
pred = clf.predict(X)
print (r2_score(y, pred))
print (mean_squared_error(y, pred))
```

```
[Parallel(n_jobs=4)]: Using backend ThreadingBackend with 4 concurrent workers.
[Parallel(n_jobs=4)]: Done  42 tasks      | elapsed:    2.3s
[Parallel(n_jobs=4)]: Done 192 tasks      | elapsed:   10.4s
0.5169231021374173
0.01622014130910015
[Parallel(n_jobs=4)]: Done 400 out of 400 | elapsed:   21.6s finished
```

Figure 13

Area Under the Curve, or AUC explains how much the model is capable of distinguishing between a fraudulent verses non-fraudulent classification. 0 would be the lowest AUC and would mean that your model has a hard time determining differences between fraud and real transactions, 1 is the highest. In Figure 14, our AUC score is .986. This tells us that our Random Forest is actually very good at classifying key differences in fraudulent charges.

```
#Area under curve = 98.58% pretty good vclose to 1
roc_auc_score(y, pred)
```

```
0.9858276779625101
```

Figure 14

Problem 4: Predictive Analytics
Sarah Lazio-Maimone

It might help us understand our model by pulling out the most important, or influential variables in our dataset. These variables have a larger weight to the model because they are considered strong indicators in deciding whether the transaction is real or fake. In Figure 15, we see below that column V258, C1, V257, and V201 have the biggest influence on our Random Forest classifying the transaction. The Vesta Corporation unfortunately did not explain exactly what each of those variables represent.

	Features	Importances
259	V258	0.075137
15	C1	0.050376
258	V257	0.046441
202	V201	0.041427
26	C13	0.036339
27	C14	0.027511
201	V200	0.022598
25	C12	0.021358
190	V189	0.021067
4	card1	0.020644

Figure 15

Figure 16 visualizes those same variable or feature importances in the bar chart below.

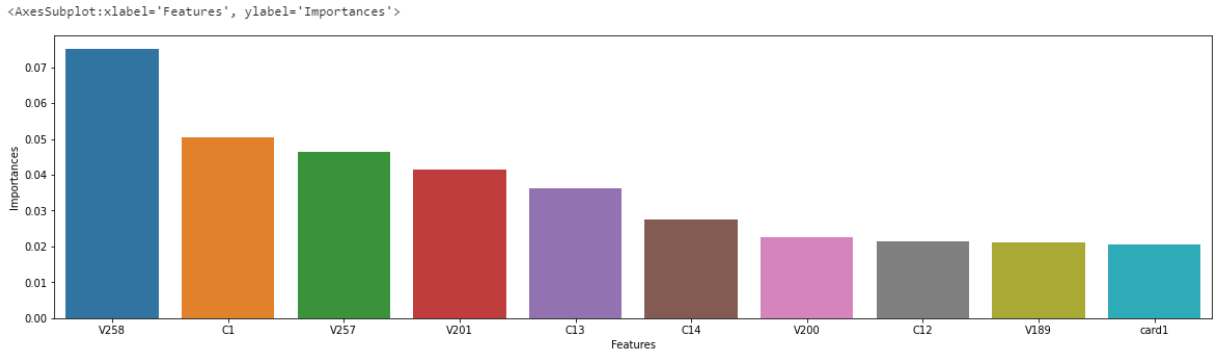

Figure 16

Problem 4: Predictive Analytics
Sarah Lazio-Maimone

We apply a different machine learning technic below to diversify our modeling. XGBoost is an algorithm that is also a decision tree type technique, but it puts most of its effort in attempting to minimize the error of the model. Boosting usually tends to give more importance to the specifics of your training set rather than trying to find a real life, all-encompassing model. In Figure 17, the XGBoost had a higher score of 98.3% of variability in fraudulent transactions can be determined by our predictive variables. We're not surprised it fits our training data better than Random Forest, but we move on to see how well it predicts.

```
xgbc.score(X, y) #how well did we fit/train
#98.3% of variabilty in fraudulent transactions can be determined by variables
```

```
/opt/conda/lib/python3.7/site-packages/xgboost/data.py:114: UserWarning: Use subset (sliced dat
ay is not recommended because it will generate extra copies and increase memory consumption
  "because it will generate extra copies and increase " +
0.9835383847096745
```

Figure 17

Figure 18 below calculates the R-squared and mean squared of predicting fraudulent charges. Here we see a large difference in the previous score and R-squared. 50.9% of the variability in determining fraud transactions in our data set can be explained by the predictor variables. This is a bit concerning because they're so different, there's a chance this new boosting model overfit our training data significantly. The mean squared is similar to the Random Forest model at 1.65% difference between our estimated fraudulent charges and the actual fraudulent charges.

```
# Evaluate model pipeline on test data
predxgbc = xgbc.predict(X)
print (r2_score(y, predxgbc))
print (mean_squared_error(y, predxgbc))
```

```
/opt/conda/lib/python3.7/site-packages/xgboost/data.
  "because it will generate extra copies and increas
0.5097313952624944
0.01646161529032551
```

Figure 18

Our AUC score for the XGBoost model is 0.775, seen in Figure 19. This is again much lower than the Random Forest, since we want this score to be as close to 1 as possible. This provides more evidence of an overfit model, because the XGB fit the training set extremely well, but did not translate as well on the actual testing data set. Our boosting model is not as good at classifying fraudulent transactions as the Random Forest was.

```
#Area under curve = 77.47% interestingly farther from 1 than rf
roc_auc_score(y, predxgbc)
```

```
0.7747153470524334
```

Figure 19

Finally, in Figure 20, we proceed to look at the most important variables in determining the XGBoost model. Columns V258, V244, V201, and V70 are weighted the heaviest and considered the most influential in deciding whether or not a transaction is fraudulent.

	Features	Importances
259	V258	0.136555
245	V244	0.072559
202	V201	0.065517
108	V70	0.064229
125	V91	0.037706
188	V187	0.028291
345	id_35	0.026218
190	V189	0.025781
157	V156	0.011800
290	V294	0.011631

Figure 20

We visualize the feature our variable importance again in the Figure 21 bar chart found below.

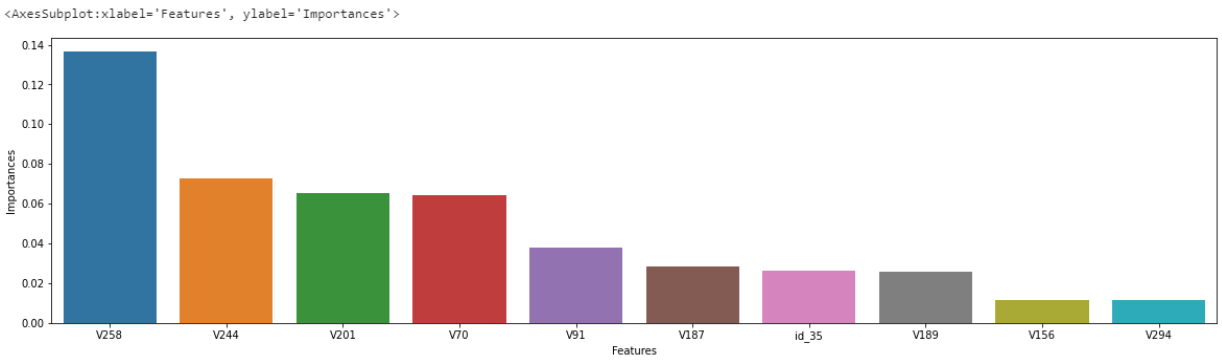

Figure 21

Through our data exploration we discovered that fraudulent transactions tend to have a lower transaction range amount than non-fraud transactions. We guess that might be because a person committing fraud doesn't want a lot of attention to what they're doing. There was an odd prevalence of Product C in transactions that were fraud that did not align with the overall distribution of product transactions. Both Mastercard and Visa were used to perform fraud, but they aligned well with the frequency of non-fraudulent charges. Credit card type and mobile device both had a higher percentage of fraud than debit card type or desktop devices. Once we applied machine learning modeling to the training data and began predicting fraudulent transactions, we found that the Random Forest Classifier performed best. The XGBoost machine learning technique most likely overfit our training data which did not perform as well once given the new testing data. Both models did agree that the most important feature in determining fraud was V258, and saw similarities with V257 and V201 being in the top 3 and 4 most influential.