Credit Fraud Detection and the Introduction of the EMV Chip

Maxwell Wagner (Maxwell.V.Wagner@gmail.com)

CUNY School of Professional Studies – May 2017

1. Introduction

The United States recently implemented EMV chips in most common credit and debit cards, comparable to the standard set by European countries. These EMV chips, which get their namesake from Europay, Mastercard, and Visa, are said to be an improvement over the traditional strip and signature design that is in common use. However, there are mixed opinions about what a change in design can equate to in terms of amount and types of fraud.

To understand why an EMV chip may be an improvement, one must understand the old strip style credit cards and their inherent flaws. A typical American credit card has a few pieces of information. They have; a sixteen-digit card numbers, an expiration date, a name, and on the rear of the card, a security code. Under normal circumstances, obtaining information from the front and rear of the card is difficult, and wouldn't allow the thief to use the card in person, only in online scenarios. However, a magnetic strip on the back of a card contains an unchanging, hardcoded set of information about a customer's account which contains all the data from above (Kossman, 2016). The problem with a setup such as this, is that if copied, the information can be used repeatedly without being able to discern the copy from the original, both in person and online.

While currently available chip based cards still contain the same physical information as a strip-based card, the way the chip handles data storage is significantly different. At the point of sale, and chip card creates a unique code that is used once, and then made unavailable for future use. The advantage behind this design is that it allows customers to use their card in similar fashion to a strip card, but the data on the card itself cannot be copied. The difference at its core is the use of static versus dynamic data.

The intention and purpose of this research is primarily to identify the ways in which fraud is detected and provide understanding on how methods interact to provide a reliable model. A secondary goal is to identify the way fraud has, and will, change with widespread use of EMV chip cards.

2. Data Source and Methods

Obtaining a usable, consistent, and public set of data narrowed the possibilities to those of which that had been released by major companies into the hands of other researchers. A partnership between Wordline and the Machine Learning Group of ULB on data mining and fraud detection has made public a collection of 284,807 transactions, comprised of 31 variables each. Unfortunately, the variables have been PCA transformed (principal component analysis), which means the original values of each observation have been obfuscated. The names of variables have also been changed to protect the privacy of the original data owners, and what they are looking for in fraud detection. This is to both protect their own algorithm and prevent thieves from understanding exactly what kind of information they are looking for. The exception to this is \`time of transaction\` and \`amount of transaction.' The time of transaction was removed as it provided information that was obscuring results. The classification of each transaction remains valid, and is one of the best sources of fraud data available. To test certain aspects of models and algorithms, other smaller data samples were used.

The MLG ULB data source is binary unbalanced data, meaning in this case that the out of the total number of transactions, only a small (492 out of 284,807) number of cases are fraudulent. In cases where the minority class is being used for machine learning, algorithms struggle to find a pattern in the information, which leads to lower prediction ability (Japkowicz, 2002). A concept known as the accuracy paradox states that models with a high accuracy may have less predictive power than models with lower accuracy (Bruckhaus, 2007). In the case of the fraud data set, an accurate model could classify with 99.8% accuracy by predicating every single observation to be non-fraudulent. Even with 99.8% accuracy, the predictive power of the model would be nearly zero.

There are several ways to deal with sampling from unbalanced data. The three main strategies are undersampling, oversampling, and synthetic sampling. Undersampling eliminates observations from the majority class until it is equivalent to the minority class. Oversampling makes duplicates of the minority class until it is equal to the majority class. Synthetic sampling for this study will be done with the *Rose* package. *Rose* uses a nearest neighbor method with smoothed bootstrapping to generate new observation that are similar to, but not exactly the same as the pre-existing minority class. The majority class is then undersampled until the majority and minority class meet at a relative median between the two.

To understand what kind of models and sampling strategies best fit unbalanced fraud data, a number will be tested and compared to see their relative strengths. In specific, versions of linear, logistic, CART, ensemble, and various decision tree models will be tested alongside unbalanced, undersampled, oversampled, and synthetic data sets. To test a model, the following basic steps were taken. First, the raw data was run through the model, and a summary of the model was examined. Next, the insignificant values were removed from the model, followed by the removal of any variables that had a similar distribution (meaning they were correlated). An example of what a logit model looked like before trimming of variables looked like can be seen in Appendix B. In the later section of the paper the ways that EMV chips have affected fraud will be discussed.

3. How Fraud is Detected

Detecting fraudulent charges is considered an unbalanced binary classification problem. As explained above, the data set consists of 99.8% non-fraudulent and 0.2% fraudulent credit card charges. In binary classifiers, the first step to assessing the performance of a model is by creating a confusion matrix. A confusion matrix compares the prediction information with the actual values recorded in the data set.

	Prediction	
Actual	True Positive (TP)	False Negative (FN)
	False Positive (FP)	True Negative (TN)

Table 1: Confusion Matrix Diagram

In fraudulent data, a true positive refers to the number of cases of non-fraud, that is correctly classified as non-fraud. A false negative is a non-fraud being incorrectly classified as a fraud. A false positive is case of fraud that is incorrectly classified as a non-fraud, and a true negative is a fraud that is correctly classified as a fraud. The most dangerous and costly category in the confusion matrix is a false positive due to potentially misidentifying a fraudulent charge as a non-fraudulent charge. Because of this, each model should have the primary aim of minimizing the false positive category, even at the cost of a higher false negative value.

There are other metrics that can be derived from the confusion matrix which circumvent the problems the accuracy paradox stated earlier creates. Sensitivity is a measure of how many positive observations (non-frauds) are predicted positive while specificity is measure of how many negative observations (frauds) are predicted negative. Specificity is of high importance as it effectively measures the number of false positives versus the total number of fraudulent observations. Positive predictive value (PPV) is the measure of true positives in relation to the total number of positives, while negative predictive value does same, but for negative values. These values measure how many predicted frauds are frauds, and vice-versa. A final metric can be found in Cohen's kappa. Kappa takes the expected accuracy based on the sample, and compares it to the observed accuracy of the model (Landis and Koch, 1977). There are no set guidelines for kappa values, only opinions, so a comparison between models and techniques is more viable. There are many other ways to measure success but keeping a simple layout for comparison is more important in this setting. A visual guide to the various formulas can be seen below.

$$Sensitivity = \frac{\sum TP}{\sum Measured\ Positives} \qquad Specificity = \frac{\sum TN}{\sum Measured\ Negatives}$$

$$PPV = \frac{\sum TP}{\sum Predicted\ Positives} \qquad NPV = \frac{\sum TN}{\sum Predicted\ Negatives}$$

$$Cohen's\ Kappa = \frac{Observed\ Accuracy - Expected\ Accuracy}{1 - Expected\ Accuracy}$$

Equation 1: Equations for sensitivity, specificity, PPV, NPV, and Cohen's kappa

Comparing Sampling Methods

The expected and observed result when comparing sampling methods was that the synthetic method using *ROSE* would produce the best results. All testing for sampling methods were done with a CART decision tree. An explanation of numbers can be found below the table of raw results.

	Unbalanced	Undersampled	Oversampled	Synthetic
Accuracy	0.9987	0.9514	0.9679	0.9777
Sensitivity	0.9992	0.9515	0.968	0.9778
Specificity	0.6190	0.9048	0.9444	0.9184
PPV	0.9995	0.9999	0.9999	0.9999
NPV	0.5200	0.0255	0.0342	0.0633
Kappa	0.5646	0.0469	0.0639	0.1157

Table 2: Sampling Method Results

At first glance, it appears that the unbalanced data has reasonably high numbers, but the telling statistic is the low specificity. A 0.6190 specificity value means that the model only detects roughly 60% of fraud cases. In comparison, the three sampling methods obtain values in the 90% range for fraud cases. Each of the three sampling methods all have similar values for specificity, with oversampling having a slight edge by 2% detection. Synthetic, in comparison, maintains a 92% catch rate for fraud, with a higher catch rate for non-frauds than other methods. The advantage to this is lower number of false negatives for a credit card agency to deal with. In summation, oversampling and synthetic sampling are both viable with samples of smaller sizes, but when sample size increases, synthetic sampling scales better due to the nearest neighbor properties.

Model Evaluation

In this section, each model will be evaluated with synthetic sampling to determine how different types of models compare to each other in a fraud detection setting. Each model's inputs were manipulated to obtain the best possible result, which sometimes means arguments were slightly different from model to model.

	Linear	Logistic	CART
Accuracy	0.9964	0.9892	0.9768
Sensitivity	0.9966	0.9894	0.9769
Specificity	0.8551	0.8841	0.8913
PPV	0.9998	0.9998	0.9998
NPV	0.2995	0.1224	0.0608
Kappa	0.4422	0.2127	0.1111

Table 3: Linear, Logistic, CART Statistics

	C4.5 Tree	Boosted C5.0	Random Forests
Accuracy	0.9977	0.9975	0.9983
Sensitivity	0.9980	0.9977	0.9986
Specificity	0.8261	0.8478	0.8406
PPV	0.9997	0.9997	0.9997
NPV	0.4101	0.3849	0.4957
Kappa	0.5471	0.5283	0.6229

Table 4: C4.5, C5.0, Random Forests Statistics

The first metric to look at when comparing the six models is the specificity, as it is directly comparing the rate at which the model correctly identifies fraud. The logistic and CART models have specificity values 4-5% better than any other of the other options. The downside to both is the relatively higher number of false negative errors, meaning that they are each more likely to classify a non-fraud as a fraud, than the linear, ensemble boosted C5.0, C4.5, or bagged tree based random forest models. In a realistic setting, multiple of the models would be combined in an ensemble to produce results that average out to a higher sensitivity and specificity. A solely CART based model produced an 89% catch rate for fraud, and a 98% chance to correctly classify non-fraud. Companies will likely need to choose whether to completely maximize specificity at the cost of sensitivity, or have a marginally lower specificity with increased sensitivity.

There is also a point of model complexity, especially when it comes to the tree based models. CART trees versus one of the C4.5 or C5.0 produce much different looking trees due to the way they place importance on different factors, and in some cases, have different thresholds for their decision making. Below is a CART decision tree, and then a tree produced by a C4.5 model. The C4.5 tree is unreadable, while the CART tree is simple and efficient to understand.

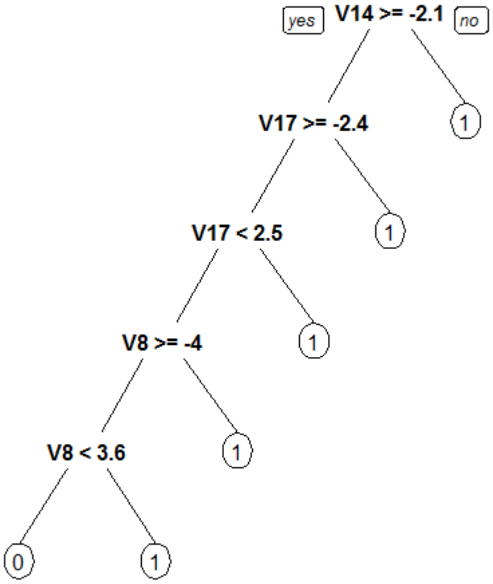

Figure 1: CART decision tree

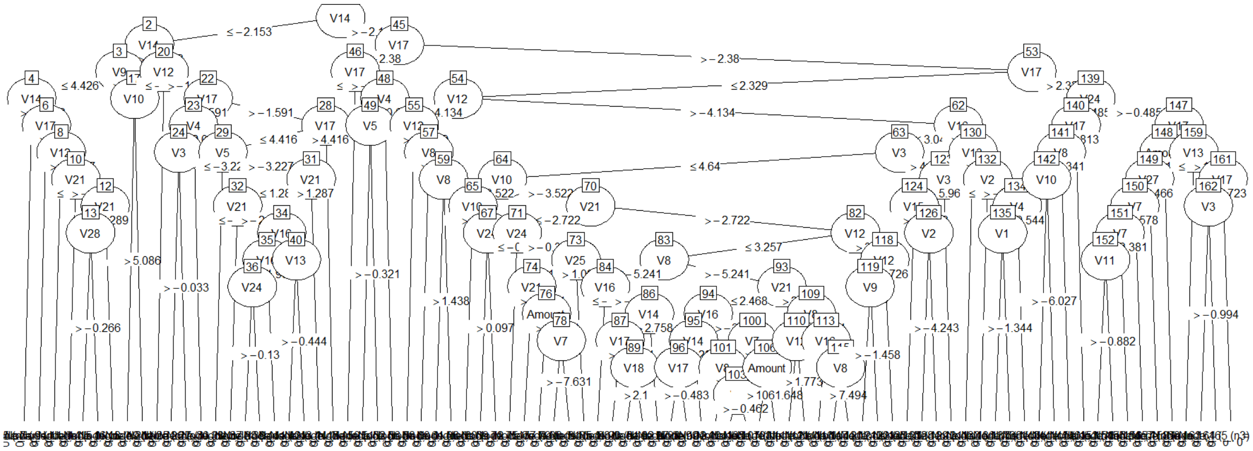

Figure 2: C4.5 Decision tree

In the plots below, two variables and their density compared to the classification can be seen. The first is variable 4, with a visible different between class 0 and class 1 observations. The second is variable 6, which shows very little difference between class 0 and class 1 data. The importance is that to a model, the density of a variable compared to class can be used as a decision point. The combination of the distributions of every variable's density distribution leads to the overall decision. In building models, individual plots can be used to determine whether to keep or discard a variable.

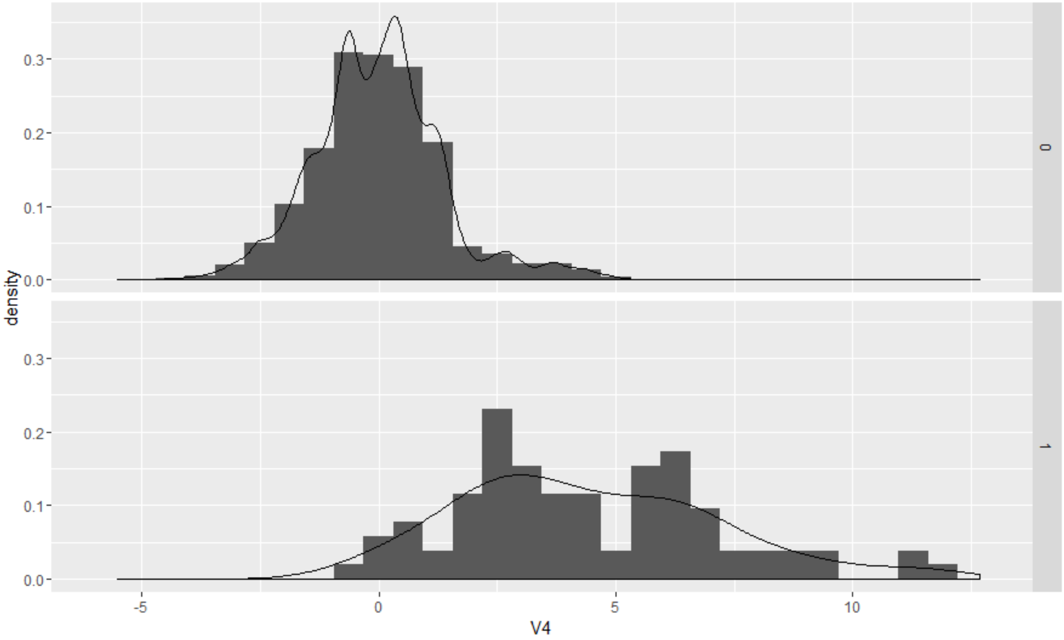

Figure 3: Variable 4 density by class

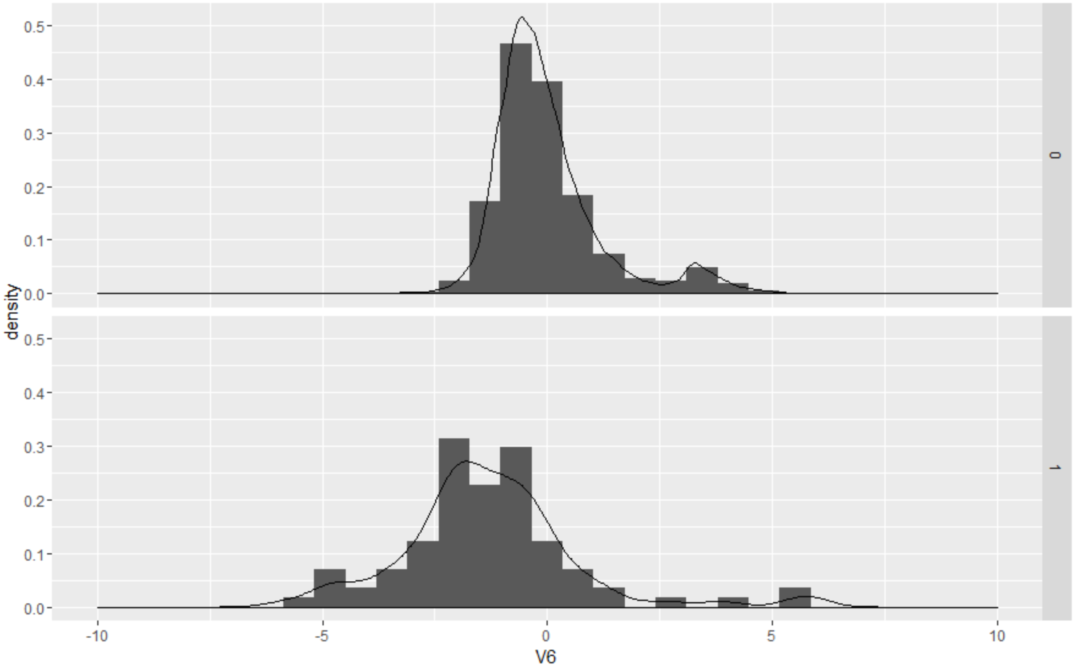

Figure 4: Variable 6 density by class

4. How EMV Chips Change the Nature of Fraud

In old strip based credit cards, one of the easiest ways to steal someone's information was by copying the magnetic strip on the back of the card. With the new EMV cards, copying that information is supposedly impossible. Mastercard, along with other companies have announced that with the introduction of EMV chips card-present fraud has had a reduction of 60% (Kitchener, 2016). This reduction means that counterfeiting and replicating credit cards in the US has decreased significantly. While the reduction is a relief for merchants, what does this decrease mean for customers?

As in with any economic model, when one measurement falls, another usually rises. In the case of fraud, this rising metric is card-not-present fraud, commonly called online transactions. The main difficulty of classifying online transaction is the loss of many variables that most detection models are based on. There are alternative methods in online transactions that many merchants do not take advantage of to improve classification such as reading security or text verification. Balancing ease of use and security is a constant concern according to experts from Fiserv.

In addition to online transactions, there are other concerns, such as out-of-country counterfeiting, identity theft, and check fraud. In an interview with Fiserv advisors, they mention that skimming credit card information is still possible in foreign countries. That information can then be used in the United States. Furthermore, instead of stealing a credit card, thieves will go directly to the source and steal an identity. Both issues have the underlying issue of US based companies having no influence or ability to detect these cases. In the same interview with Fiserv, it is mentioned that while the overall use of checks has decreased, the rate of check fraud has remained roughly the same. However, it is expected that with the decrease in card-present fraud, check fraud will see higher rates in the coming years.

The usage of EMV chips brings in a shift of liability to both merchants and customers. The overall change is to shift the blame to whichever party is the least EMV compliant. For instance, if a customer is using an EMV card and a business does not have the capability to read it, the liability falls to the business. If a customer is using a strip card, and the business can read chip cards, the liability shifts to the customer. There is an exception to this; if a customer's card supplier does not offer chip cards, the card supplier it liable. A 2015 liability shift, along with a

2017 shift to gas station and other standalone style transactions complicates who is responsible for a fraudulent transaction (EMV Migration, 2015). One criticism of this shift, is that it can potentially pin blame on consumers who are either unaware of changes, or were simply never informed of the changes.

In the earlier model section, the detection rate was for card-present fraud. With the rise of card-not-present fraud, new methods and models will need to be developed. One current problem with analyzing and producing new models for an independent researcher is the lack of public data sets dealing with online fraud prediction. Companies such as Visa and Mastercard wish to keep their information private. As told in the methods section, a key component in keeping fraud detection possible is that thieves are unaware of exactly what a detection model is searching for. By keeping information private, this improves the chances the model has to be successful, even if it limits other researchers from replicating the success.

5. Conclusions

As seen in the earlier section of the paper, even simple algorithms can detect card-present fraud to reasonable levels. More complex ensemble methods may achieve detection in the high 90% range. The larger issue for both consumers and merchants is how a lower rate of card-present fraud will affect other aspects of their life. As a consumer, if asked whether an EMV card is an improvement over a strip card, a simple answer is available. EMV chips may not be a complete fix to fraud, but It plugs one of the many holes that thieves can exploit. Even with 600 million chip cards in circulation in March of 2017, only 39% of merchants offer EMV chip processing (Kossman, 2016). According to both Mastercard and Visa, EMV leads to a reduction of card-present by 60%. For this reason alone, it would be advantageous for both customers and businesses to be EMV ready. If for liability reasons alone, EMV chips are a step in the right direction.

Future Work

Models need to be consistently updated and changed to keep with the changing nature of fraud. A great example of how models can be improved can be taken from the outcome of the Netflix challenge, in particular the use of big chaos and ensemble (Toscher, 2009). The idea is to use hundreds of different models, and train each predictor individually within each model. The resulting ensemble is then balanced and weighted to obtain optimum results. There are obvious challenges to overcome with this method. The primary is the massive computing power and time needed to individually train hundreds of thousands of predictors, and then weight them all in an ensemble. The time factor is also a problem, as it requires fine tuning by hand to perform optimally. The final barrier is the technology and cost required to use the method. This goes hand in hand with the first problem, but the sheer cost of the method makes it unobtainable without major corporate or industry backing.

6. Citations

Bruckhaus, T. "The Business Impact of Predictive Analytics." Knowledge Discovery and Data Mining: Challenges and Realities. IGI Global, 2007. 114-138.

Chawla NV, Bowyer KW, Hall L, and Kegelmeyer W. Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research (JAIR), 16:321–357, 2002.

"Confusion Matrix – Another Single Value Metric – Kappa Statistic." Software Journal. N.p., n.d. Web.

Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intelligent data analysis, 6(5):429–449, 2002

Kitchener, Beth. "Eighty Percent of Mastercard U.S. Consumer Credit Cards Have Chips."MasterCard Social Newsroom. Mastercard, 3 Aug. 2016. Web. 28 May 2017.

Kossman, Sienna. "8 FAQs about EMV Credit Cards." CreditCards.com. Creditcards.com, 02 Feb. 2016.

Landis JR, Koch GG. “The Measurement of Observer Agreement for Categorical Data.” Biometrics, vol. 33, no. 1, 1977, pp. 159–174. JSTOR, www.jstor.org/stable/2529310.

Phua C, Gayler R, Vincent Lee, Kate Smith-Miles. A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review. 2005

Pozzolo A, Caelen O, Johnson R, and Bontempi, G. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015

Toscher, Andreas, Michael Jahrer, and Robert M. Bell. The BigChaos Solution to the Netflix Grand Prize (n.d.): n. pag. AT&T Labs, 5 Sept. 2009. Web.

"Understanding the 2015 U.S. Fraud Liability Shifts." EMV Connection. N.p., May 2015. Web. 28 May 2017.

7. Appendix A: Code

Splitting for sampling and under/over/both/rose/smote balancing

```
#split data for testing models
trainLength <- floor(.7*nrow(cc_simp))
testLength <- nrow(cc_simp) - trainLength
train_model <- cc_simp[1:trainLength,]
train_eval <- cc_simp[(trainLength + 1):nrow(cc_simp),]

#undersamp
fraud <- nrow(train_model[train_model$Class == 1,])
notFraud <- nrow(train_model) - fraud
paste("fraud: ", fraud, "|| not fraud: ", notFraud)
train_model <- ovun.sample(Class ~ ., data = train_model, method = "under",N = fraud*2, seed = 1)$data

#oversamp
fraud <- nrow(train_model[train_model$Class == 1,])
notFraud <- nrow(train_model) - fraud
paste("fraud: ", fraud, "|| not fraud: ", notFraud)
train_model <- ovun.sample(Class ~ ., data = train_model, method = "over",N = notFraud*2, seed = 1)$data

#over and undersample, to meet in the middle
fraud <- nrow(train_model[train_model$Class == 1,])
notFraud <- nrow(train_model) - fraud
paste("fraud: ", fraud, "|| not fraud: ", notFraud)
train_model <- ovun.sample(Class ~ ., data = train_model, method = "both", p = 0.5, N = fraud+notFraud, seed =
1)$data

#use rose synthetic
fraud <- nrow(train_model[train_model$Class == 1,])
notFraud <- nrow(train_model) - fraud
paste("fraud: ", fraud, "|| not fraud: ", notFraud)
train_model <- ROSE(Class ~ ., data = train_model, seed = 1)$data

#SMOTE synthetic
train_model$Class <- as.factor(train_model$Class)
train_model <- SMOTE(Class ~ ., data = train_model, perc.over = 100, perc.under = 200)
```

Some of the model information, manipulation of variables was done in a separate file and depends on each model.

```
#Model1 - linear
mlr1 <- glm(Class~., data = train_model)
BIC(mlr1)
predict_mlr1 <- predict(mlr1, train_eval, type = 'response')
confusionMatrix(round(predict_mlr1), train_eval$Class)
mysd(predict_mlr1, train_eval$Class)
myse(predict_mlr1, train_eval$Class)
auc_mlr1 <- roc(train_eval$Class, predict_mlr1)
plot(auc_mlr1)

#bagged tree random forest (kind of broke with raw data)
```

```
rforest <- randomForest(Class ~ ., data = train_model)
ptm <- proc.time();rforest <- randomForest(Class ~ ., data = train_model, ntree = 100);proc.time() - ptm
predict_rforest <- predict(rforest, train_eval)
confusionMatrix(round(predict_rforest), train_eval$Class)
auc_rforest <- roc(train_eval$Class, predict_rforest)
plot(auc_rforest)

#c4.5
c45 <- J48(as.factor(Class) ~., data = train_model)
predict_c45 <- predict(c45, train_eval)
confusionMatrix(predict_c45, train_eval$Class)
plot(c45)

#boosted c50
ptm <- proc.time();c50 <- C5.0(as.factor(Class) ~., data = train_model);proc.time() - ptm
predict_c50 <- predict(c50, train_eval)
confusionMatrix(predict_c50, train_eval$Class)
plot(c50)

# CART
ptm <- proc.time();decision <- rpart(Class ~ ., data = train_model, method = "class");proc.time() - ptm
prp(decision)
predict_decision <- predict(decision, train_eval, type = "class")
confusionMatrix(predict_decision, train_eval$Class)
```

Example code to produce a feature plot comparison with *ggplot*, and the simple way to make a decision tree graph with *party*.

```
ggplot(cc_simp, aes(x = V6, y = ..density..)) + geom_histogram() + geom_density() + facet_grid(Class ~ . ) + xlim(-
10,10)

require("party")
require("partykit")
plot(c50)
```

7. Appendix B: Logit Output

Below is output from the summary of the logit model. It shows the significance of each variable. The insignificant variables for each model were removed until only significant ones remained.

```
Call:
glm(formula = Class ~ ., family = binomial(link = "logit"), data = train_model)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.4662  -0.2755  -0.0420   0.0140   4.5316

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.227e+00  4.146e-02 -77.842  < 2e-16 ***
V1          -7.338e-02  7.283e-03 -10.076  < 2e-16 ***
V2           1.079e-01  1.005e-02  10.729  < 2e-16 ***
V3          -1.450e-01  8.035e-03 -18.051  < 2e-16 ***
```

```
V4          6.456e-01  1.371e-02  47.083  < 2e-16 ***
V5          1.076e-01  9.596e-03  11.217  < 2e-16 ***
V6         -2.789e-01  1.640e-02 -17.002  < 2e-16 ***
V7         -9.748e-03  7.472e-03  -1.305 0.192036
V8         -6.525e-02  6.864e-03  -9.506  < 2e-16 ***
V9         -3.036e-01  1.729e-02 -17.558  < 2e-16 ***
V10        -1.635e-01  1.089e-02 -15.010  < 2e-16 ***
V11         3.218e-01  1.518e-02  21.202  < 2e-16 ***
V12        -2.998e-01  1.221e-02 -24.556  < 2e-16 ***
V13         7.285e-03  2.039e-02   0.357 0.720868
V14        -5.137e-01  1.256e-02 -40.893  < 2e-16 ***
V15        -1.590e-01  2.214e-02  -7.180 6.98e-13 ***
V16        -1.671e-01  1.378e-02 -12.126  < 2e-16 ***
V17        -2.452e-02  7.828e-03  -3.132 0.001736 **
V18         2.610e-03  1.682e-02   0.155 0.876638
V19        -2.062e-01  2.185e-02  -9.438  < 2e-16 ***
V20        -8.795e-02  2.485e-02  -3.539 0.000402 ***
V21         1.086e-01  9.762e-03  11.126  < 2e-16 ***
V22        -7.733e-02  2.225e-02  -3.476 0.000510 ***
V23         5.088e-02  2.849e-02   1.786 0.074102 .
V24        -3.904e-01  3.686e-02 -10.591  < 2e-16 ***
V25         3.802e-02  3.746e-02   1.015 0.310137
V26        -3.464e-01  4.633e-02  -7.477 7.59e-14 ***
V27         8.237e-02  3.206e-02   2.570 0.010182 *
V28         4.733e-02  5.148e-02   0.919 0.357887
Amount      6.934e-04  9.389e-05   7.386 1.52e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 48520  on 34999  degrees of freedom
Residual deviance: 10521  on 34970  degrees of freedom
AIC: 10581
```