Supervised Methods for Credit Card Fraud Detection

Tyler Scott
CSCI 4502-001B (Distance)
tysc7237@colorado.edu

Christina Nguyen
CSCI 5502-001B (Distance)
nguyencq@colorado.edu

Nicholas Clement
CSCI 4502-001B (Distance)
nicl7004@colorado.edu

Christopher Struckman
CSCI 4502-001B (Distance)
christopher.struckman@colorado.edu

1 INTRODUCTION

With the increase of online transactions, the opportunities for massive credit card fraud also increase. As techniques for stealing credit cards shift towards quick, massive transactions, the cost burden rises for card companies and consumers. Being able to detect fraud early on, and rapidly adapting to new fraudulent methods, can save millions of dollars per year. Since most existing research in this area comes before the surge in online shopping and the recent wave of machine learning, we believe that our project will help guide new work with respect to using technology to detect ever-changing approaches to credit card fraud.

The goal of this research is to apply supervised data mining techniques to the problem of credit card fraud, specifically behavioral fraud, which is defined as a fraudulent transaction where a credit card number was stolen. We aim to go beyond the existing work with binary classifiers and explore sophisticated deep learning models, such as deep feed-forward and recurrent neural networks, and also explore ensemble methods, such as bagging and boosting, for detecting fraud. To facilitate comparisons with previous work, we will also explore simpler binary classifiers such as logistic regression and naive Bayes.

2 LITERATURE REVIEW

Credit card fraud is not new, and the size of the issue in terms of costs and victims makes this an area previously explored. However, most studies took place before the rise in e-commerce and organized fraud crime rings [1]. We believe that these changes fundamentally alter the way credit card fraud is perpetrated. Thus, we aim to improve on credit card fraud detection by leveraging previous work in the field and recent advances in data mining. The prior literature we will use for reference and baseline model performance are:

- Data mining for credit card fraud: A comparative study, Bhattacharyya et al.
- Credit Card Fraud Detection Using Bayesian and Neural Networks, Maes et al.

Bhattacharyya et al. present three classification models: a logistic regression model, a support vector machine (SVM), and a random forest for automatically detecting credit card fraud. They note that almost all datasets for detecting fraudulent transactions are heavily skewed, leading to different training schemes and evaluation metrics. To adjust, they do random undersampling of the majority class (genuine transactions) [1]. They argue that this performs better than other approaches, such as oversampling the minority class or weighting predictions based on prior distributions for each of the two classes [1]. Table 1 below contains results from the research.

Maes et al. focused on using Bayesian networks and neural networks for detecting credit card fraud. Notably, they use several different feed-forward multi-layer perceptrons trained using the Backpropagation algorithm [2]. Table 2 below contains receiver operating characteristic (ROC) curve results from the networks. One point to note from this research is that their dataset contains confidential features because of customer privacy [2]. This is very similar to our dataset, presented in 3.1, where the features are anonymized and transformed through Principal Components Analysis (PCA).

A key claim in Maes et al. is that the loss from fraud outweighs the cost from investigating and stopping fraud. As such, we will adapt a similar stance and aim to minimize false negatives rather than false positives. To do this, a correlation analysis of attributes can help improve the results by indicating which highly correlated attributes might be unduly affecting results [2].

As these studies note, the training datasets for credit card fraud rely on correct categorization. There exists a non-zero probability in previous work that fraudulent transactions are misclassified as legitimate and legitimate transactions are misclassified as fraudulent [1, 2].

3 PRELIMINARY RESULTS AND PROPOSED WORK

3.1 Data

The data we are using for our research is the *Credit Card Fraud Detection* (CCFD) dataset. This dataset can be found at:

https://www.kaggle.com/dalpozz/creditcardfraud

The data contains 284, 807 examples, labeled into two classes: genuine transactions and fraudulent transactions, where the fraudulent transactions make up 0.172% of the total set. This results in 284, 315 genuine examples and 492 fraudulent examples. Since the data could potentially contain private information on the credit-card holders, the creators ran most of the features through the Principal Component Analysis (PCA) algorithm and renamed them to $V_1, V_2, \ldots, V_N$, where $N = 28$. The only features that are not renamed are the monetary amount of each transaction and the elapsed number of seconds since the first transaction. Thus, the proposed data mining research will refer to the features using their original notation, and lose context and interpretability of most dimensions of the data.

Table 1: Cross-validation performance of different techniques [1]

	Accuracy	Recall	Specificity	Precision	F	AUROC
LR	0.947	0.654	0.979	0.778	0.709	0.942
SVM	0.938	0.524	0.984	0.782	0.624	0.908
RF	0.962	0.727	0.987	0.86	0.787	0.953

Table 2: This table compares the results achieved with artificial neural networks for a false positive rate of respectively 10% and 15% [2]

Experiment	±10% false pos	±15% false pos
ANN-fig 2(a)	60% true pos	70% true pos
ANN-fig 2(b)	47% true pos	58% true pos
ANN-fig 2(c)	60% true pos	70% true pos

3.2 Initial Analysis

Before developing supervised classification models, we focused on analyzing and exploring the dataset from 3.1. The creators of the dataset stated that the features $V_1, V_2, \ldots, V_{28}$ resulted from PCA, so those features should be orthogonal. As a safety measure, we computed the dot product between pairs of feature vectors and ensured the resulting value was zero:

$$V_i \cdot V_j = 0 \quad \forall i = 1, \ldots, 28, j = 1, \ldots, 28, i \neq j$$

We found that this was indeed true. Furthermore, it wasn't explicitly stated whether dimensionality reduction had been conducted after finding an orthogonal basis of features from PCA, so we explored how well each feature separated genuine and fraudulent transactions. Figure 1 shows the distribution of each of the anonymous features, $V_1, \ldots, V_{28}$, color-coded by class label. This allowed us to reduce the dimensions of our dataset by removing the features that didn't distinguish the two classes. We removed $V_{13}, V_{15}, V_{20}, V_{22}, V_{23}, V_{24}, V_{25}, V_{26}, V_{28}$ as a result of this analysis. The removal of these features may not significantly improve supervised classification performance using discriminative models such as logistic regression and neural networks because those models can set weight values to zero for uninformative features, but for generative models such as naive Bayes, removing these features can help model the posterior probability distribution of the class label given the data.

A simple attribute correlation across the entire dataset showed no correlation between any attributes. This makes sense since most of the features are from PCA, so they are pairwise-orthogonal. However, by splitting the dataset based on class, we found a marginal increase in correlation between attributes within a given class. In particular, for transactions classified as fraudulent, both the pairs of V_{16} and V_{17} as well as V_{17} and V_{18} have correlation coefficients that are statistically significant with a significance level of 0.05 as seen in Figure 2. No attributes have strong negative correlations.

One of the key attributes of our dataset is that the transactions are temporally linear. Interestingly, the attributes that are highly correlated for fraud also have the highest correlation with the time attribute. However, their correlation with the time attribute is not high (between 0.25 and 0.3), however, it is the highest amongst all 28 attributes. Expanding the significance level to .1 revealed only a couple more highly correlated attributes, but going out to a significance level of .15 revealed a number more.

These correlation patterns do not hold for the genuine transactions, where there were no significant attribute-to-attribute correlations, even up to a significance level of .15. Unsurprisingly, the features that did not distinguish the two classes also had the lowest correlation coefficients with all other attributes when comparing attribute-to-attribute.

3.3 Visualization

We have completed preliminary visualization work, mostly related to the initial analysis in Section 3.2 and supervised classification in Section 3.4 (see Figure 1 and Figure 3), however, we would like to explore this area further.

Credit card fraud detection is a domain that lacks human intuition regarding the applicability of machine learning. This is most likely because humans can't visualize the many varying dimensions involved in a potential fraudulent transaction. As a result, we would like to explore methods for visualizing both classes of transactions while maintaining local proximity relationships between examples. Due to the large number of dimensions in the dataset, the t-distributed stochastic neighbor embedding (t-SNE) method will be utilized. By visualizing the dataset in a low-dimensional space, it can be used as a form of evaluation for the supervised classification models. We can compare the evaluation metrics from the models against the visualization. If the data points from each class are clearly separable, we should expect to see that, numerically, in the evaluation metrics. If we don't, it allows us to look for any bugs or choose evaluation metrics better suited for the domain.

3.4 Supervised Methods

Since the CCFD dataset is labeled, supervised binary classification methods can be utilized to learn from the training examples and predict unseen, novel examples. As stated in Section 2, prior research on credit card fraud detection has focused on simple, binary classifiers such as logistic regression, random forests, and feed-forward

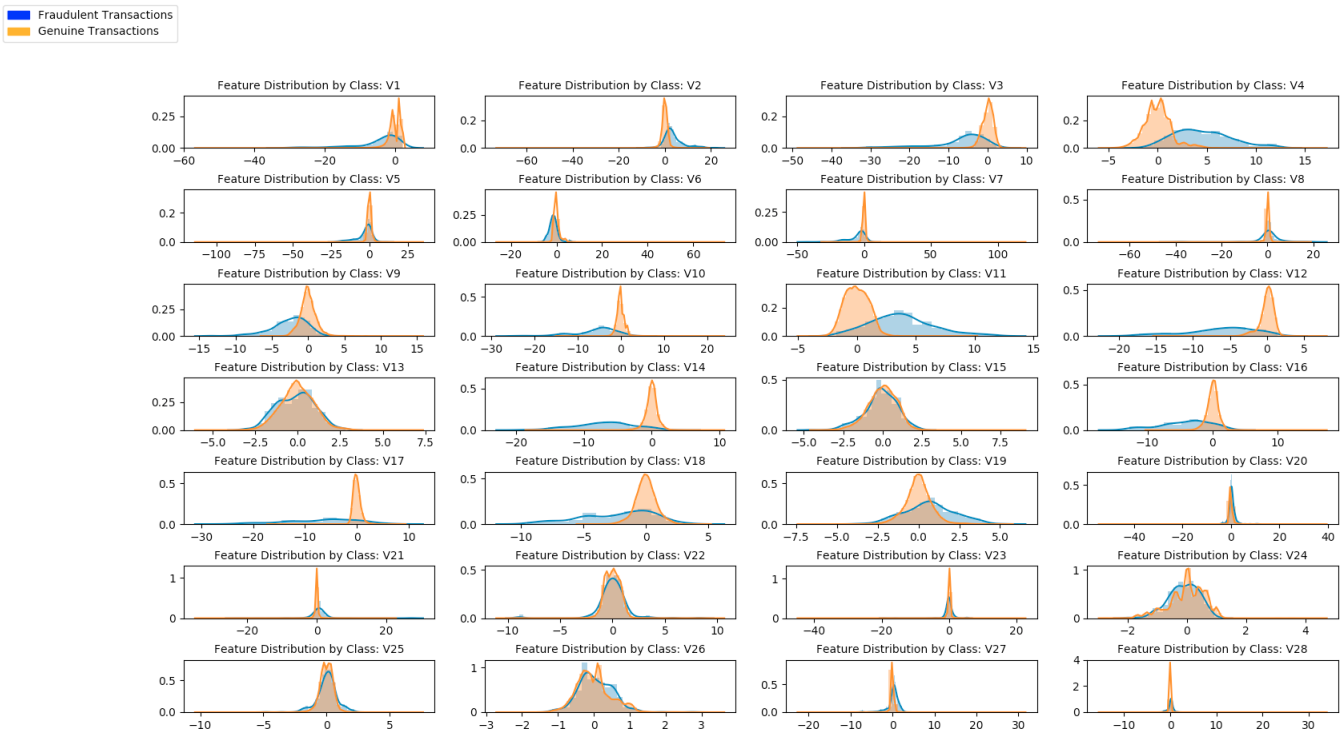


Figure 1: Distribution histograms for each of the anonymous features $V_1, \ldots, V_{28}$, color-coded by class label.

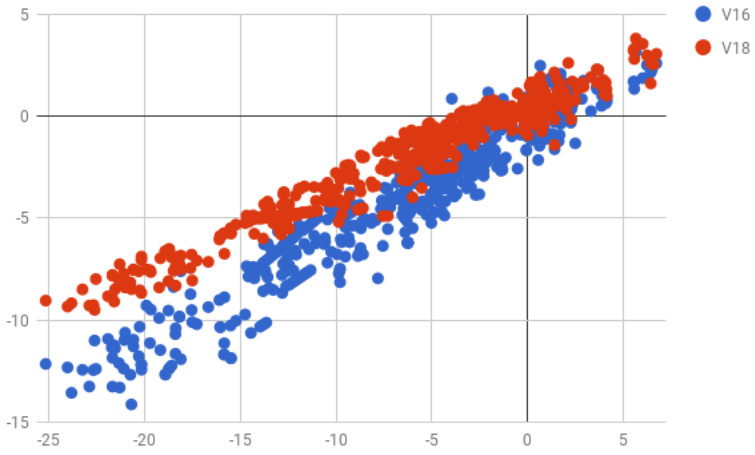


Figure 2: Correlation scatter plots with V_{17} as the x-axis, and V_{16} and V_{18} as points. This is for fraudulent transactions only.

multi-layer Perceptrons. However, recent advances in classification may show to perform better than previous methods and improve on state-of-the-art results. We have implemented several classification models including naive Bayes, logistic regression, and a deep feed-forward neural network. For these models, we also removed uninformative features, as determined in Section 3.2. In addition, we propose using support vector machines and ensemble methods such as bagging and boosting to see if further improvement can be made. We hypothesize that ensemble methods may perform much better than traditional classifiers because we can intelligently sample the data for each ensembled classifier such that all fraudulent transactions are drawn, while only a fraction of genuine transactions are drawn. This helps re-balance the dataset with respect to each ensembled classifier, but also use most training examples in the

dataset, which is not a natural extension to single classifiers such as deep neural networks. Thus, the ensemble methods will hopefully weigh fraudulent transactions equivalently to genuine transactions. Furthermore, since the data has temporal structure, recurrent neural networks may significantly outperform other methods, as they can leverage sequential information in the data. The goal of implementing several different classification models is to compare them using a variety of evaluation metrics, as well as compare them to research conducted in the past. We think it would be a useful contribution to explore several supervised models, highlighting metrics such as time and space complexity, accuracy, ability for online learning after training, and others. For more information on specific evaluation metrics, see Section 4.

3.4.1 Naive Bayes. As a baseline model, we implemented naive Bayes. The goal of the model was simply to provide a realistic measure of performance that we could compare to. Since naive Bayes is a generative model that relies heavily on the prior probability of each class, we expected the classifier to guess that each transaction was genuine. Also, the model doesn't rely on a loss function, like logistic regression and neural networks, so we couldn't weigh the fraudulent examples more heavily like in Sections 3.4.2 and 3.4.3. However, a simple extension that we would like to explore is manually setting the prior probabilities of each class such that the fraudulent transactions have a larger weight compared to genuine transactions. The performance of the naive Bayes model can be found in Table 3.

3.4.2 Logistic Regression. We also implemented a logistic regression model to detect credit card fraud. Unlike naive Bayes, the logistic regression model uses a log-loss cost function, which allowed us to weight different examples. Thus, we empirically determined that weighing the fraudulent example five times more than genuine transactions seemed to have the best performance. We also experimented with extreme cases, such as not using a weighing scheme and apply a weighing scheme that balanced the importance of each class (weigh fraudulent examples approximately 2000 times more). What we found is that when a weighting scheme is not applied, the classifier just guesses all transactions as genuine, however, when each class is weighted equally, the precision metric and F-score metric (which depends on precision) drop significantly. This intuitively makes sense because precision is defined as:

$$precision = \frac{TP}{TP+FP}$$

and since there are so many genuine transactions, the false-positive value is large relative to the true-positive value, driving precision to zero. Thus, using a value in the middle was desired, and a value of five was determined to perform best. We found that the weight value of five performed significantly better than other values, but we plan to conduct a form of validation, such as k-fold cross validation to search the hyperparameter space for the optimal weight value. The performance of the logistic regression model was evaluated using several metrics, shown in Table 3.

3.4.3 Deep Feed-Forward Neural Network. A basic deep feed-forward neural network was implemented on the credit card fraud detection task. Several architectures, loss functions, and activation functions were used and evaluated using a held-out validation set. We tested both shallow and deep models with cross-entropy loss, logistic loss, and squared-error loss, as well as activation functions such as sigmoid, hyperbolic tangent, and rectified linear unit (ReLU). The network with best performance was the following:

- Input layer of 20 units (after removing features based on 3.2)
- 2 hidden layers of 64 units with hyperbolic tangent activation
- Output layer of 1 unit with sigmoid activation
- Weights and biases initialized using a random normal distribution with mean 0 and standard deviation of 1
- Weighted cross-entropy loss function where fraudulent transactions are heavily weighted (5 times more than genuine transactions)
- Adam optimizer trained for 500 epochs, learning rate of 0.01, and 256 examples per batch

The performance of the neural network was evaluated using several metrics, shown in Table 3.

3.5 Tools

For other researchers and peers who would like to explore credit card fraud detection, below is a description of the tools we used for our analysis. The goal is that others can build on our results and get started quickly by leveraging useful data mining tools. For many of the machine learning tasks, Scikit-learn was the main library for development of the models, data preprocessing, validation, and associated evaluation. Tensorflow was used in conjunction with Scikit-learn for development of the deep learning models. For simpler tasks such as data storage and transformation, Numpy and Pandas were used. Finally, for data analysis and visualization tasks, Matplotlib and Seaborn were used.

3.6 Changes from Project Proposal

We initially wanted to explore unsupervised and semi-supervised techniques for detecting credit card fraud. We proposed applying clustering techniques such as k-Nearest-Neighbors and Gaussian mixture models to see if a distribution of the fraudulent and genuine transactions could be learned, as well as outlier detection methods where we solely model genuine transactions and use statistical methods to determine outliers, which we assumed would be fraudulent transactions. However, due to time constraints and the interests of the team, we decided to focus on supervised classification. As mentioned in Section 3.4, we would like to explore ensemble methods, support vector machines, and recurrent neural networks for detecting credit card fraud. Although, if time permits, it would be interesting and potentially valuable to explore clustering methods and outlier-detection methods when applied to credit card fraud.

4 EVALUATION

As with many classification tasks, simple evaluation metrics such as accuracy are used to compare performance between models. This is most likely because accuracy is a simple measure to implement, it generalizes to more than just binary labels, and it is model-agnostic. However, one major assumption with accuracy is that it is assumed that there is an equal representation of examples from each class. In cases where the number of examples per class is skewed, accuracy is a poor and misleading measure. For example, if we consider a dataset with 99 examples from class A and 1 example from class B, then a classifier that simply guesses class A, regardless of the input

Table 3: Performance of different supervised classification models on the credit card fraud detection task. The dataset used in [1] is different, so relative comparisons should be used as compared to absolute comparisons.

	Accuracy	Recall	Precision	F	AUROC	Average Precision (AP)
LR [1]	0.947	0.654	0.778	0.709	0.942	N/A
SVM [1]	0.938	0.524	0.782	0.624	0.908	N/A
RF [1]	0.962	0.727	0.86	0.787	0.953	N/A
Naive Bayes (Ours)	0.981	0.889	0.063	0.156	0.933	0.13
LR (Ours)	0.9993	0.808	0.782	0.795	0.904	0.74
Feed-forward Neural Net (Ours)	0.9994	0.8532	0.7812	0.8105	0.946	0.786

features, will have an accuracy of 99%. It seems like the classifier is doing extremely well, when in fact, the classifier is learning nothing from the data. This is exactly the case we have with the CCFD dataset. For this dataset, 99.828% of the data is labeled as genuine transactions (see Section 3.1). To combat the misleading measure of accuracy, we instead will focus on the following metrics:

- Recall
- Precision
- F-Score
- Area under the ROC curve (AUROC)
- Average precision (AP)

Due to the imbalance in the dataset, we chose to focus on metrics related to receiver operating characteristic (ROC) curves and precision-recall curves. These metrics are better indicators of model performance on skewed data, where higher values for all of the metrics are better. Furthermore, our main metric is average precision (AP), which is a better-calibrated numerical representation of area under the precision-recall curve, which is analogous to area under the ROC curve (AUROC). Precision-recall curves are very similar to ROC curves, however, precision-recall curves focus more on correct classification of the positive class (fraudulent class), whereas ROC curves focus on distinguishing between the two classes. In other words, ROC curves focus on both true positives and true negatives, however, we want to focus more on true positives, false negatives, and false positives, which aligns with precision-recall curves. For credit card fraud detection, fraudulent transactions are much more important to classify than genuine transactions, which naturally leads to using AP as the main metric.

Table 3 shows a comparison of several supervised classification models from [1] and our team. For reference, LR refers to logistic regression, SVM refers to support vector machine, and RF refers to random forest. Also, [1] does not compute average precision as a metric, so that can only be used to compare our models. Also, [1] used a different credit card fraud detection dataset, with a different class imbalance, so the comparison between the two groups of models shouldn't be absolute, but rather used as a relative comparison of our models to state-of-the-art performance on supervised fraud detection. We used naive Bayes as a baseline model to gauge performance on the task. In terms of accuracy and AUROC, the model appears to perform well, however, when using metrics more suitable for imbalanced data such as precision, F-score, and AP, we see that naive Bayes had a challenging time classifying fraudulent transactions. Since the prior probability on genuine transactions was so high, the classifier most likely guessed that each transaction was genuine, which accounts for the low precision value. An interesting result is that logistic regression and the deep feed-forward neural network both perform very similarly. The neural network outperforms logistic regression slightly, but the improvements are modest. This is most likely an indication that logistic regression, which can be thought of as a neural network with no hidden layers and a sigmoid output activation, has enough capacity to distinguish genuine and fraudulent transactions. Thus, the 4-layer neural network didn't need the extra layers to do well on the task. When compared to naive Bayes, we can see that logistic regression and the neural network were able to classify fraudulent transactions and didn't guess genuine for all of the training examples. The two reasons for this are that the logistic regression and neural network classifiers were more flexible and could better distinguish the two classes, and we could apply a weighting scheme to these classifiers such that fraudulent training examples were five times more important to classify correctly. In other words, if the classifier incorrectly labeled a fraudulent transaction (false negative), the loss function was penalized five times more than incorrectly labeling a genuine transaction (false positive). This is the exact intuition that we had when conducting preliminary research on credit card fraud. It is much more important to classify fraudulent transactions correctly, at the expense of misclassifying genuine transactions, than misclassifying fraudulent transactions (false negatives are more important than false positives). We can also compare our models to those implemented in [1]. Since the datasets are different, it may not be correct to do an absolute comparison of the two, but we can see that our models have very similar metric values, if not better, compared to [1]. This is an indication that our models compare to state-of-the-art research on credit card fraud detection.

Figure 3 shows ROC curves and precision-recall curves for each of our three classifiers—naive Bayes, logistic regression, and the neural network. As shown in the figure, the AUROC for each classifier is approximately the same, with all reaching an area of greater than 0.90. This is support from our claim above that ROC curves are not necessarily the best indicator of classifier performance on imbalanced data, mainly because they consider true negatives when calculating the false positive rate. Precision-recall curves are much better indicators of overall performance on skewed datasets because they only consider true positives and disregard true negatives, specifically for our task, the transactions that are genuine. As seen in the plots, naive Bayes has a very low average precision

score, whereas logistic regression and the neural network have much higher scores. This is because the naive Bayes model simply guessed that almost every transaction was genuine, forcing the precision value to be near zero. The logistic regression and neural network models were able to classify many of the fraudulent transactions correctly, which helped to balance precision and recall.

5 REMAINING TASKS

As mentioned in Section 3.4, our team wants to focus on implementing more sophisticated supervised classification models for detecting credit card fraud. Specifically, we would like to focus on support vector machines, ensemble methods, and recurrent neural networks. Up to this point, there hasn't been an emphasis on the sequential structure of the data, so using a model designed to exploit this information, such as a recurrent neural network, may end up outperforming all other classifiers. Also, we would like to improve our existing classifiers by using validation data to tune hyperparameters. Our results are based on values that seemed to perform well, but this wasn't a result of rigorous analysis. By searching the hyperparameter space for the classifiers with more rigor, we may be able to improve upon existing results. Another interesting area to research is visualization. Since credit card fraud is very grounded in terms of real-world applicability, being able to visualize the data may be invaluable. Due to the "black-box" nature of machine learning, visualizing genuine transactions and fraudulent transactions in a 2-dimensional or 3-dimensional space may open up the models and give insight to customers. For example, if a bank determines that a transaction is fraudulent, they could send the customer a graphic showing their usual (genuine) transactions, and where the new (fraudulent) transaction sits in this space. That will allow society, in general, to put more trust in machine learning algorithms related to credit card fraud. Finally, if time permits, the team would like to explore unsupervised methods for detecting credit card fraud such as clustering analysis and outlier analysis. It isn't always the case that data is readily available, and being able to detect fraud, statistically, without class labels may be much more applicable to the real-world.

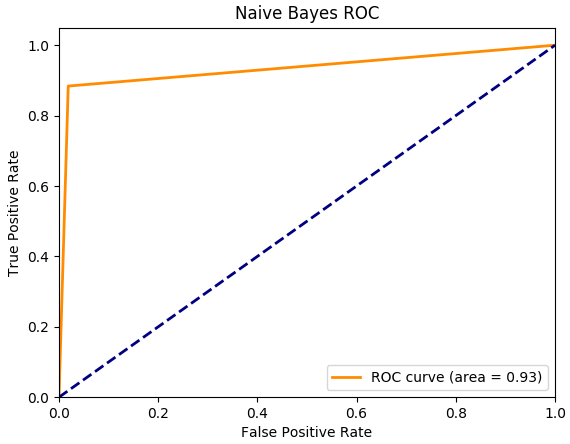
6 PROJECT DISCUSSION

Thus far, we feel like our results have been very positive. We have successfully applied machine learning and data mining techniques to a real-world problem, credit card fraud detection, and our results compare to those in previous state-of-the-art research. Development of the models and analysis has been relatively smooth and we currently don't foresee challenges preventing us from moving forward. However, the course staff provided feedback to explore the domain deeper, outside of just machine learning and the associated metrics. Since our project is so related to everyday people, it is interesting to formulate the problem as if we were working for a bank. That is, if we were performing this analysis on behalf of a bank, we would be exploring metrics much more general than average precision, F-score, etc. One metric that is much more important is revenue loss. When a fraudulent transaction occurs, the bank has to refund the customer with the amount of the transaction. Thus, this is a direct loss to the bank's revenue. As a result, banks are most likely much more concerned with classifying fraudulent transactions correctly. Also, by incorporating a revenue-loss-per-fraud-transaction value, such as $25 per transaction, for example, it would allow the bank to quantify the number of false positives and false negatives that are acceptable. Since we don't have this data, it is much harder to perform this analysis, but this is a further direction the project could go, if we were paired with a bank and had access to their data. This projects the task into a real-world domain, where not only is the classifier's performance important, but also how the performance is optimized for the real-world and the associated measures of performance such as revenue loss.

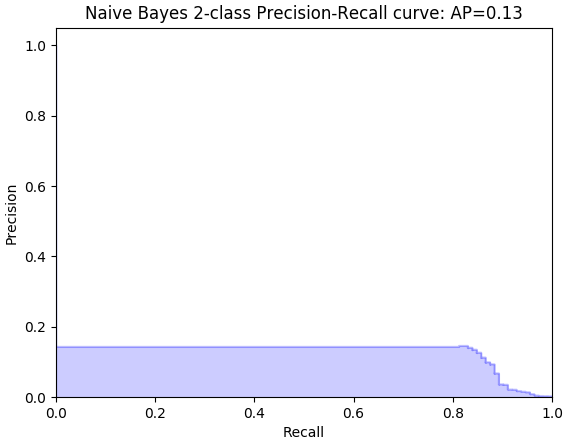
REFERENCES

[1] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J. Christopher Westland. 2011. Data Mining for Credit Card Fraud: A Comparative Study. *Decis. Support Syst.* 50, 3 (Feb. 2011), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

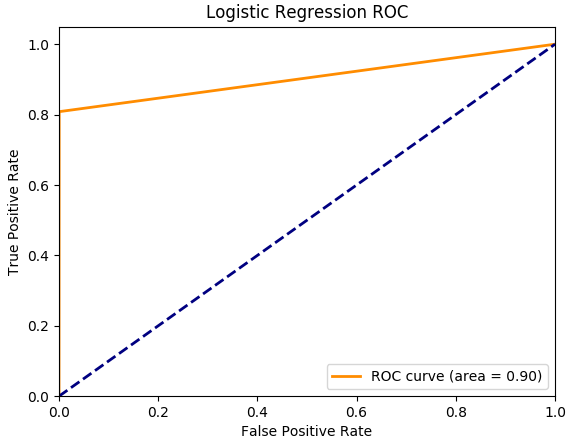
[2] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. 1993. Credit Card Fraud Detection Using Bayesian and Neural Networks. In *In: Maciunas RJ, editor. Interactive image-guided neurosurgery. American Association Neurological Surgeons.* 261–270.



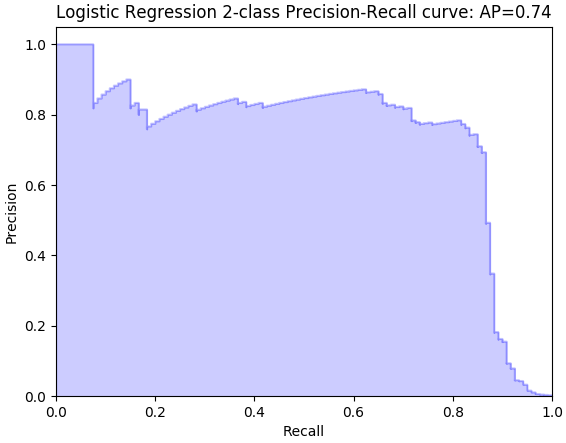
(a) Naive Bayes ROC Curve



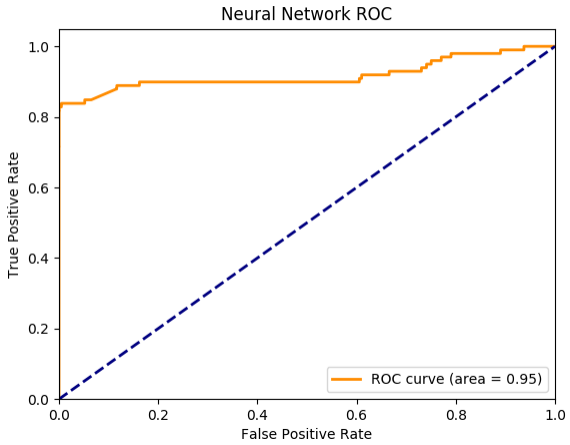
(b) Naive Bayes Precision-Recall Curve



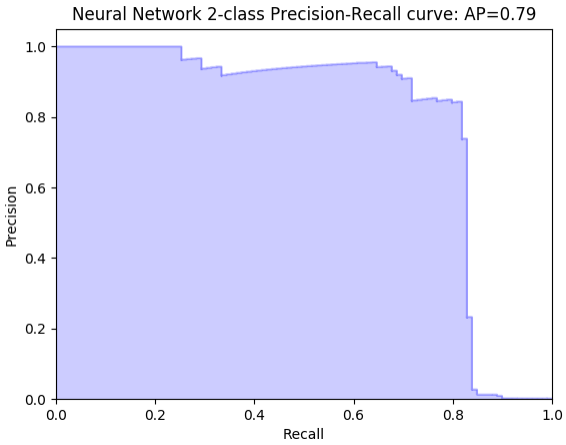
(c) Logistic Regression ROC Curve



(d) Logistic Regression Precision-Recall Curve



(e) Neural Network ROC Curve



(f) Neural Network Precision-Recall Curve

Figure 3: ROC curves and Precision-Recall curves for our different models.