Group 387: Finding the best classifier to predict credit card fraud detection

First Name	Last Name	Email address
Vikas	Sanil	vsanil1@hawk.iit.edu

Table of Contents

1. Introduction

Worldwide financial losses caused by credit card fraudulent activities are worth tens of billions of dollars. Financial Institutions keep upgrading their credit card fraud detection algorithm to early detect such incidents and reduce financial losses.

Analyzing credit card fraud detection datasets will provide a good opportunity to apply different algorithms learned in this course and understand their merits and demerits. This will also provide a platform to showcase my skills during my job search.

2. Data

Found data: From Kaggle datasets**:** https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
Source: Machine Learning Group – ULB
Data Type: Anonymized credit card transactions labeled as fraudulent or genuine
Content: The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly imbalanced, the positive class (frauds) accounts for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA(Principal Component Analysis) transformation. Unfortunately, due to confidentiality issues, the source cannot provide the original features and more background information about the data. There are 31 features. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. Feature 'Class' is the label and it takes value 1 in case of fraud and 0 otherwise.

3. Problems and Solutions

Problems:

- Use supervised learning algorithms such as Decision trees and SVC to come up with fraud prediction models.
- ~~Verify whether unsupervised learning can provide any insight into the data by clustering and then feeding those results to Supervised learning algorithms.~~
- Use Ensemble methods such as Random Forest and Gradient Tree Boosting to feed different model predictions and see whether that improves the performance.
- Validate each model using hold-out validation.
- Evaluate Neural Network classifier performance comparatively.
- Evaluate each model under AUC ROC and Brier loss.

Solutions: Will find the best ensemble method model to predict credit card fraud.

4. KDD

4.1. Data Processing

We found and removed 1081 duplicate transactions in the dataset.

Standardized the dataset using StandardScaler since we are using classifiers which need Standardized dataset.

As the dataset is imbalanced with positive class (label 1) being minority(473/283,726) we used **Synthetic Minority Oversampling Technique** (SMOTE) to over-sample the minority class in the training dataset. Also we have evaluated Random Under Sampler and SMOTETomek (Combine over- and under-sampling using SMOTE and Tomek links.) on the training dataset to compare the model performance.

4.2. Data Mining Methods and Processes

We are using following data mining classifiers to evaluate credit card fraud detection:

Decision Tree are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

We are using decision tree classification criteria *Gini and Entropy.* We are also finding best *ccp_alpha* value to use in decision trees by running '*cost_complexity_pruning_path'* method. The optimal ccp_alpha value identified was 5.5.

Logistic regression, despite its name, is a linear model for classification rather than regression. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

We are using solver 'saga' as it is faster for large dataset and penalty l2.

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression, and outliers' detection.

The advantages of support vector machines are:

- Effective in high-dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and the d regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

We are running *Linear kernel SVM* and *RBF kernel SVM*

RandomForest ensemble algorithm is based on randomized decision trees. RandomForest algorithm is perturb-and-combine techniques [B1998] specifically designed for trees. This means a diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers.

GradientBoostingClassifier or Gradient Boosted Decision Trees (GBDT) is a generalization of boosting to arbitrary differentiable loss functions. GBDT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems in a variety of areas including Web search ranking and ecology.

We are using both criterion '*friedman_mse'* and '*squared_error'* for our evaluation.

- *friedman_mse* – Mean squared error with improvement score by Friedman.
- *squared_error* – Mean squared error.

HistGradientBoostingClassifier is a histogram-based estimator, which can be orders of magnitude faster than GradientBoostingClassifier when the number of samples is larger than tens of thousands of samples.

We are evaluating this ensemble against GradientBoostingClassifier.

Multi-layer Perceptron (MLP) is a Neural Network models supervised learning algorithm that learns a function f(.) : $R^m \rightarrow R^o$ by training on a dataset, where '*m'* is the number of dimensions for input and '*o'* is the number of dimensions for output. Given a set of features X =x_1, x_2, …, x_m and a target '*y*', it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers.

We are using *solver* '*adam'* since it works well on relatively large datasets (with thousands of training samples or more) in terms of both training time and validation score with a *hidden layer* of ~~*10*~~ 1 and 2 with 50 neurons in 1 layer and 100, 200 neurons in 2 layers. We are also comparing solver 'lbfgs' and 'sgd' with 2 hidden layer.

KMeans algorithm clusters data by trying to separate samples in *n* groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters(K) to be specified. It scales well to large numbers of samples and has been used across a large range of application areas in many different fields.

We are using Elbow Method for optimal K value for KMeans. Once the dataset is clustered, we are passing the dataset again through all the above-mentioned classifiers to evaluate result improvement.

5. Evaluations and Results

5.1. Evaluation Methods

The following validation and scores are used to evaluate the classifiers for better credit card fraud detection.

We are using **hold-out** validation since it is a common practice when performing a (supervised) machine learning experiment.

Brier score loss is a proper score function that measures the accuracy of probabilistic predictions. It is applicable to tasks in which predictions must assign probabilities to a set of mutually exclusive discrete outcomes. The Brier score loss is between 0 to 1 and the lower the value (the mean square difference is smaller), the more accurate the prediction is.

Accuracy is the proportion of correctly classified test instances. The best value is 1 and the worst value is 0.

Precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0.

Recall is intuitively the ability of the classifier to find all the positive samples. The best value is 1 and the worst value is 0.

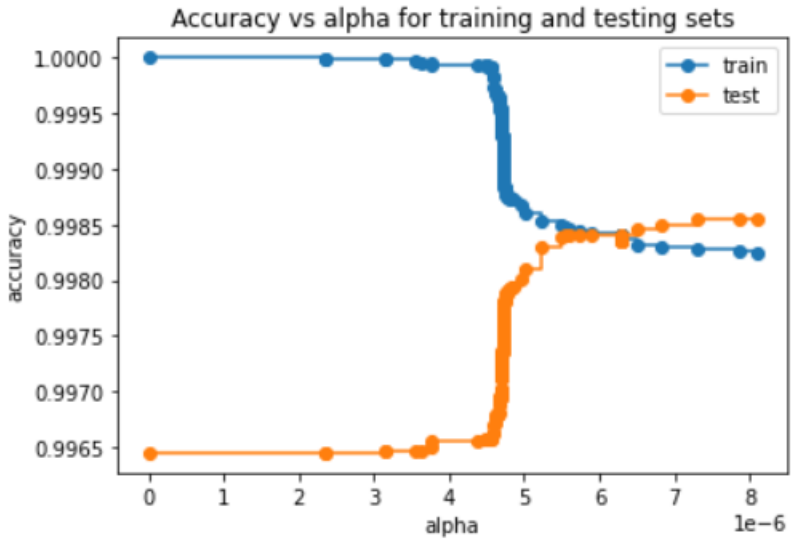
F1 score can be interpreted as a harmonic mean of precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.

ROC AUC score, the area under the ROC curve gives the ranking accuracy, an estimate of the probability that a random positive is ranked before a random negative.

The performance of classifiers can be summarized by means of a table known as a contingency table or **confusion matrix**. In this table, each row refers to actual classes as recorded in the test set, and each column to classes as predicted by the classifier.

5.2. Results and Findings

When we ran cost complexity pruning path method on credit card fraud detection dataset and found 5.5 is best ccp_alpha value to use in Decision Tree.



Below are the results when classifiers ran on an **imbalanced dataset with few positive cases**.

Classifier	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Decision Tree	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Logistic Regression	0.001671	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Linear SVM	0.001672	0.993773	0.002577	0.007042	0.003774	0.501235	0.843750
RBF SVM	0.001672	0.929843	0.001203	0.049296	0.002348	0.490308	1.781250
Random Forest	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	0.234375
GradientBoosting Squared Error	0.003691	0.996309	0.000000	0.000000	0.000000	0.498990	0.031250
HistGradientBoostingClassifier	0.003265	0.996733	0.000000	0.000000	0.000000	0.499203	0.093750
MLPClassifier Adam 100, 200	0.001970	0.997913	0.027027	0.007042	0.011173	0.503308	7.359375
MLPClassifier LBFGS 100, 200	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	1.250000
MLPClassifier SGD 100, 200	0.001683	0.998325	0.000000	0.000000	0.000000	0.500000	0.906250

From above table we can observe **MLPClassifier** has better Brier loss, F1 and Area under ROC score compared to other classifiers considered in this evaluation. But as we can also observe most of our classifier lack on precision and recall due to imbalanced data we will be performing over sampling and under sampling of training data.

Below are the results when classifiers ran after **over sampling minority class using SMOTE** on credit card fraud detection training dataset.

Classifier	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Decision Tree	0.250000	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Logistic Regression	0.223241	0.673778	0.002639	0.514085	0.005250	0.594065	0.000000
Linear SVM	0.250000	0.987841	0.002240	0.014085	0.003865	0.501780	0.953125
RBF SVM	0.010106	0.989009	0.000000	0.000000	0.000000	0.495334	1.828125
Random Forest	0.249970	0.998325	0.000000	0.000000	0.000000	0.500000	0.187500
GradientBoosting Squared Error	0.012332	0.987287	0.003185	0.021127	0.005535	0.505017	0.062500
HistGradientBoostingClassifier	0.193199	0.733675	0.002351	0.373239	0.004672	0.553760	0.234375
MLPClassifier Adam 100, 200	0.016176	0.978631	0.023945	0.295775	0.044304	0.637775	4.796875
MLPClassifier LBFGS 100, 200	0.017251	0.978136	0.004630	0.056338	0.008556	0.518010	0.625000
MLPClassifier SGD 100, 200	0.077173	0.891350	0.003068	0.197183	0.006042	0.544849	1.078125

We can observe The Precision and Recall score has bettered compared imbalance data for most of the classifiers. MLPClassifier is still better performing classifier among the classifiers.

When MLPClassifer is excluded then Gradient Boosting has better Brier loss and F1 score but Logistic Regression has better Area under ROC.

Below are the results when classifiers ran after **under sampling performed using RandomUnderSampler** on credit card fraud detection training dataset.

Classifier	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Decision Tree	0.250000	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Logistic Regression	0.237233	0.663695	0.002594	0.521127	0.005163	0.592530	0.015625
Linear SVM	0.221464	0.003632	0.001666	0.992958	0.003327	0.497465	0.781250
RBF SVM	0.266428	0.353087	0.001603	0.619718	0.003198	0.486179	1.515625
Random Forest	0.250105	0.001675	0.001675	1.000000	0.003344	0.500000	0.265625
GradientBoosting Squared Error	0.454323	0.535586	0.002056	0.570423	0.004097	0.552975	0.015625
HistGradientBoostingClassifier	0.422429	0.553169	0.001926	0.514085	0.003839	0.533660	1.078125
MLPClassifier Adam 100, 200	0.013766	0.982629	0.016000	0.154930	0.029005	0.569473	4.859375
MLPClassifier LBFGS 100, 200	0.138794	0.847998	0.002886	0.260563	0.005709	0.554773	0.671875
MLPClassifier SGD 100, 200	0.053220	0.927012	0.003449	0.147887	0.006740	0.538103	0.765625

With under sampling, we observe that Precision and Recall score has bettered for few classifiers (excluding MLPClassifier) and Accuracy has reduced. This may be due to the controlled over-fitting performed.

Still MLPClassifier has the best brier loss and F1 score but Logistic Regression has better Area under ROC.

When MLPClassifier excluded then Logistic Regression classifier has better scores in brier, F1 and Area Under ROC.

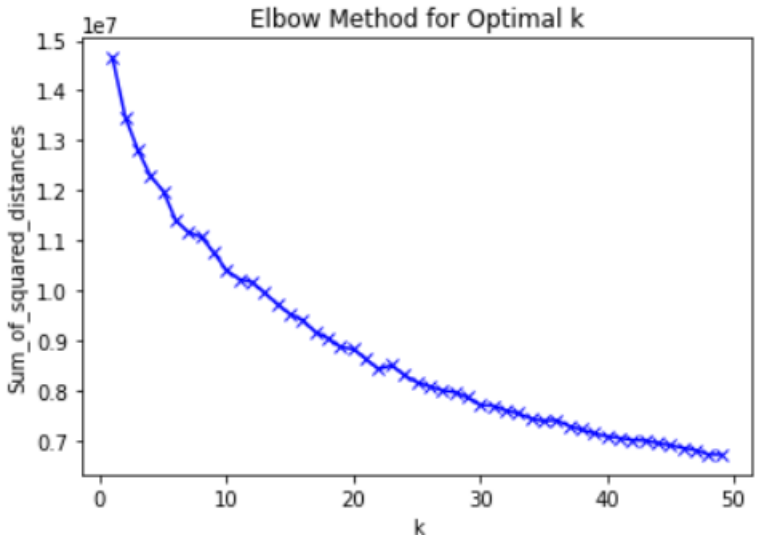
Below are the results when classifiers ran after **combine under sampling and oversampling algorithm SMOTETomek** used on credit card fraud detection training dataset.

Classifier	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Decision Tree	0.250000	0.998325	0.000000	0.000000	0.000000	0.500000	0.015625
Logistic Regression	0.223241	0.673778	0.002639	0.514085	0.005250	0.594065	0.000000
Linear SVM	0.292967	0.987841	0.002240	0.014085	0.003865	0.501780	0.656250
RBF SVM	0.019574	0.989009	0.000000	0.000000	0.000000	0.495334	1.781250
Random Forest	0.249970	0.998325	0.000000	0.000000	0.000000	0.500000	0.328125
GradientBoosting Squared Error	0.012332	0.987287	0.003185	0.021127	0.005535	0.505017	0.093750
HistGradientBoostingClassifier	0.201764	0.742957	0.002115	0.323944	0.004203	0.533802	0.234375
MLPClassifier Adam 100, 200	0.011455	0.985294	0.016623	0.133803	0.029572	0.560263	7.171875
MLPClassifier LBFGS 100, 200	0.008811	0.989091	0.003802	0.021127	0.006445	0.505921	0.796875
MLPClassifier SGD 100, 200	0.033275	0.954408	0.003731	0.098592	0.007191	0.527217	1.234375

We can observe MLPClassifer has better brier loss and F1 score and second-best Area Under ROC. Logistic Regression still has a better Area under ROC.

When MLPClassifier excluded then Gradient Boosting has better Brier loss and F1 score.

~~We ran Elbow method to find optimal K value for KMeans clustering and found sum of squared distance elbows between 20 and 30. We have chosen optimal K value as 25.~~



~~Below are the results when classifiers ran after **clustering (using KMeans)** minority class over sampled credit card fraud detection dataset.~~

Classifier	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Decision Tree	0.250000	0.499997	0.000000	0.000000	0.000000	0.500000	0.015625
Logistic Regression	0.226057	0.621068	0.637198	0.562295	0.597408	0.621069	0.031250
Linear SVM	0.249944	0.502791	0.501399	0.999279	0.667748	0.502794	7.156250
RBF SVM	0.348008	0.504787	0.512555	0.195317	0.282850	0.504785	32.671875
Random Forest	0.250000	0.499997	0.499997	1.000000	0.666664	0.500000	4.718750
GradientBoosting Squared Error	0.010620	0.989032	0.983531	0.994720	0.989094	0.989032	0.671875
GradientBoosting Friedman MSE	0.010221	0.989386	0.984617	0.994306	0.989438	0.989386	0.953125
HistGradientBoostingClassifier	0.037361	0.954112	0.935979	0.974909	0.955048	0.954112	1.906250

6. Conclusions and Future Work

6.1. Conclusions

Following are the conclusions:

- Imbalanced dataset will not yield good prediction model.

- Multi-layer Perceptron has better prediction model as we increase the hidden layer and neuron value.

- For imbalanced data combine method of over-sampling and under-sampling has better performance result compared to individual methods.

- Gradient Boosted Decision Trees has better performing prediction model among Decision Tree, Logistic Regression, Support Vector Machine, Random Forest and Histogram-based Gradient Boosted Decision Trees(excluding Multi-layer Perceptron) for Credit Card Fraud Detection dataset as it has more balanced accuracy, precision and recall score compared to others.

-

6.2. Limitations

We didn't have the dataset in entirety hence were not able to run normalization and feature selection to find the right dimension of the credit card fraud detection dataset for our prediction models.

6.3. Potential Improvements or Future Work

As we can observe from the results, the Area under ROC is less than 0.7, which means the prediction is below the acceptable model. With further fine-tuning of the classifier parameters, we must improve the model prediction.

References:

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler

https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

https://scikit-learn.org/stable/modules/svm.html#svm-classification

[B1998]

12. Breiman, "Arcing Classifiers", Annals of Statistics 1998.

- P. Geurts, D. Ernst., and L. Wehenkel, "Extremely randomized trees", Machine Learning, 63(1), 3-42, 2006.

https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html#sphx-glr-auto-examples-classification-plot-classifier-comparison-py

https://scikit-learn.org/stable/modules/ensemble.html#forest

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

https://scikit-learn.org/stable/modules/clustering.html#k-means

https://scikit-learn.org/stable/modules/cross_validation.html

https://scikit-learn.org/stable/modules/tree.html

https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC

https://arxiv.org/pdf/1106.1813.pdf

https://imbalanced-learn.org/stable/zzz_references.html#id4

https://imbalanced-learn.org/stable/references/generated/imblearn.combine.SMOTETomek.html#imblearn.combine.SMOTETomek

https://imbalanced-learn.org/stable/combine.html#combine

https://imbalanced-learn.org/stable/under_sampling.html

https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html#imblearn.under_sampling.RandomUnderSampler