

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

Group member:

Vikas Sanil – [vsanil1@hawk.iit.edu](mailto:vsanil1@hawk.iit.edu)

# Data:

**Found data:** From Kaggle datasets: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

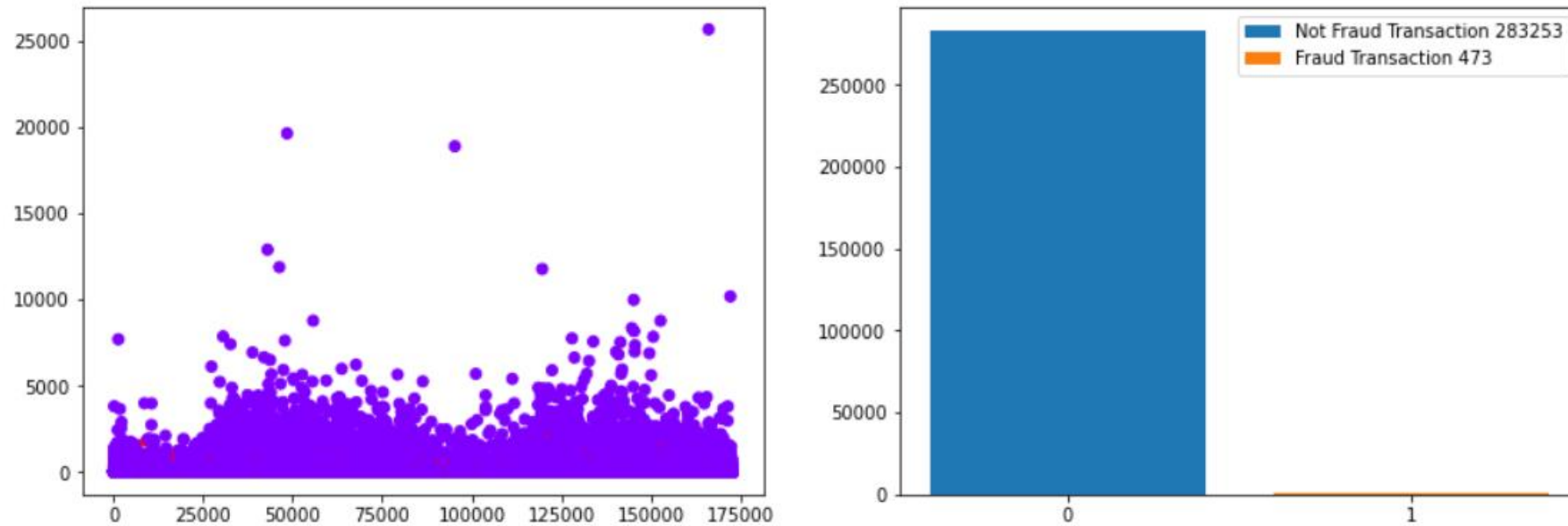
**Source:** Machine Learning Group – ULB

**Data Type:** Anonymized credit card transactions labeled as fraudulent or genuine

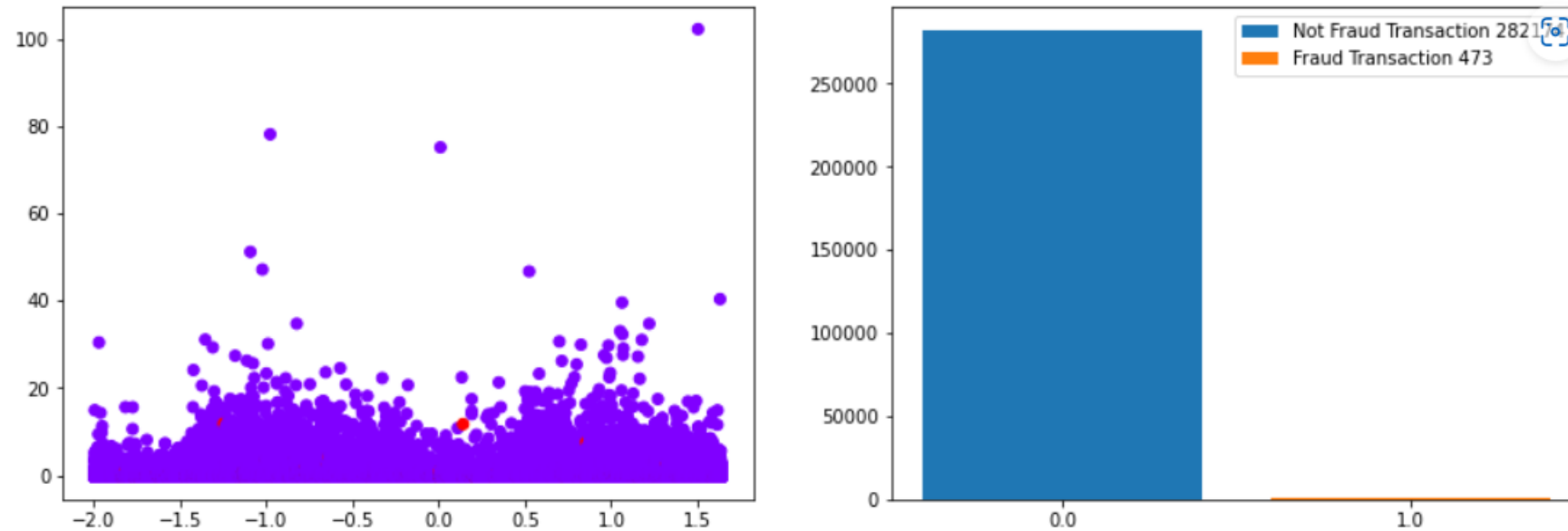
**Content:**

- 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 unbalanced, the positive class (frauds) accounts for 0.172% of all transactions.
- It contains only numerical input variables which are the result of a PCA 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.

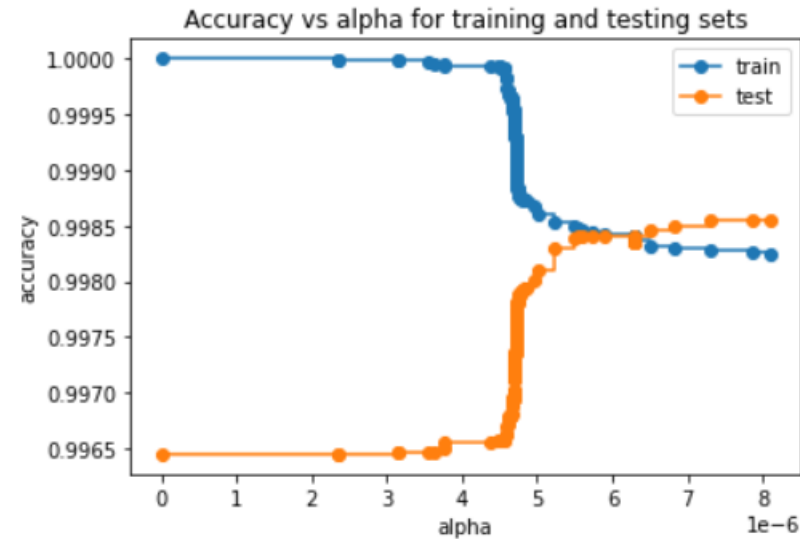
# Dataset before data preprocessing:



# Data after duplicate removal and standardization:



# Cost Complexity Pruning Path method result used to find optimal ccp\_alpha for Decision Tree:



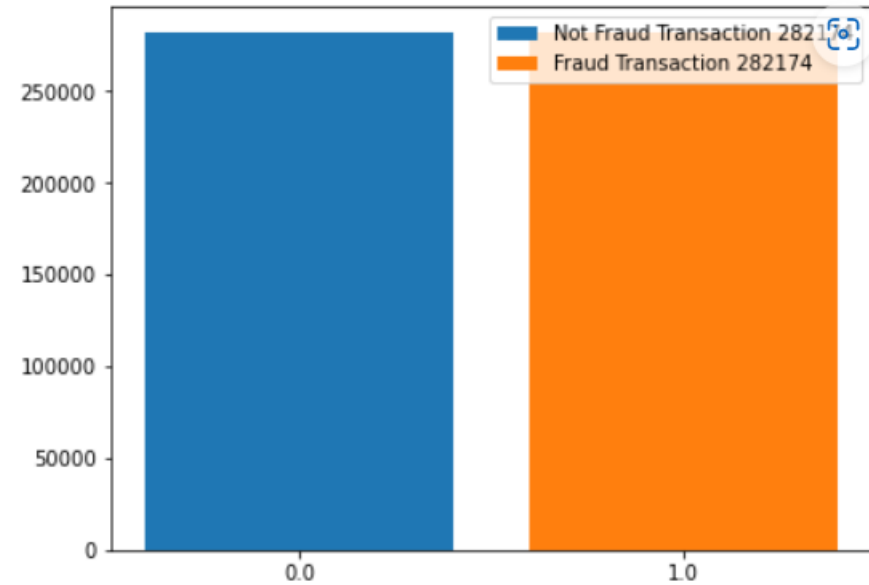
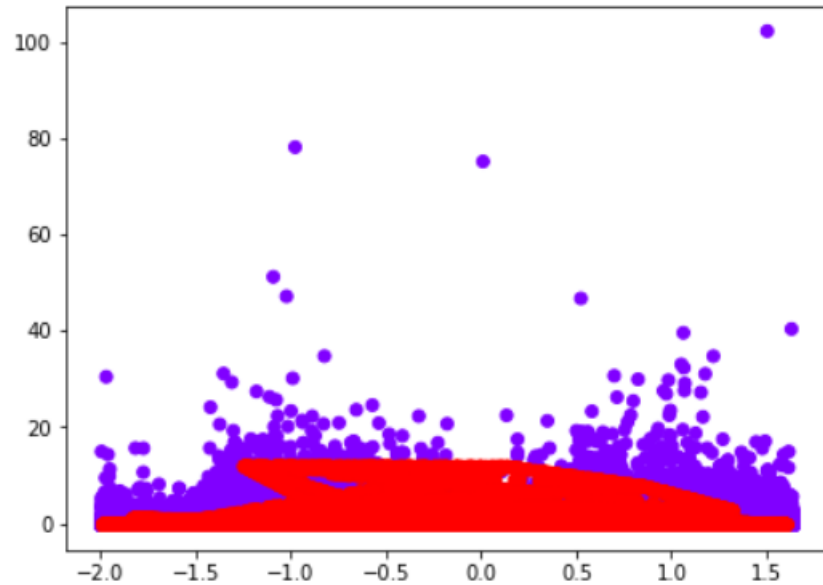
# Classifiers Used:

- `classifiers = [`
- `DecisionTreeClassifier(criterion='gini', max_depth=1000, ccp_alpha = 5.5),`
- `LogisticRegression(penalty='l2', solver='saga', max_iter=1000),`
- `SVC(kernel="linear", C=0.025, max_iter=1000, probability=True),`
- `SVC(gamma='auto', C=1, max_iter=1000, probability=True),`
- `RandomForestClassifier(criterion='gini', max_depth=1000, ccp_alpha = 5.5, n_estimators=1000, max_samples=0.8, random_state=42),`
- `GradientBoostingClassifier(n_estimators=1000, random_state=42, learning_rate=1.0, criterion='squared_error', max_depth=1000),`
- `GradientBoostingClassifier(n_estimators=1000, random_state=42, learning_rate=1.0, criterion='friedman_mse', max_depth=1000),`
- `HistGradientBoostingClassifier(loss='log_loss', learning_rate=1.0, max_iter=1000, max_depth=1000),`
- `MLPClassifier(hidden_layer_sizes=(10,), solver='adam', random_state=1, max_iter=1000, warm_start=True)`
- 
- `]`

# Results on imbalanced dataset with minority positive cases:

	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Classifier							
Decision Tree	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Logistic Regression	0.001670	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Linear SVM	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	1.265625
RBF SVM	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	8.781250
Random Forest	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	2.343750
GradientBossting Squared Error	0.003868	0.996132	0.005319	0.007042	0.006061	0.502417	0.171875
GradientBossting Friedman MSE	0.003550	0.996450	0.012270	0.014085	0.013115	0.506091	0.125000
HistGradientBoostingClassifier	0.002154	0.997854	0.000000	0.000000	0.000000	0.499764	0.171875
MLPClassifier	0.001677	0.998325	0.000000	0.000000	0.000000	0.500000	0.046875

# Data after over sampling minority cases using SMOTE:

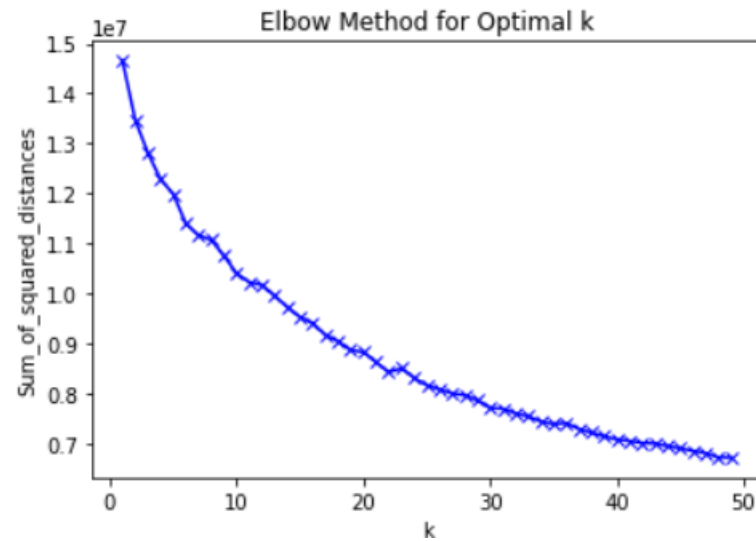




# Results on minority cases oversampled dataset:

	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Classifier							
Decision Tree	0.250000	0.499997	0.000000	0.000000	0.000000	0.500000	0.015625
Logistic Regression	0.226456	0.617247	0.634334	0.553660	0.591258	0.617247	0.015625
Linear SVM	0.250000	0.502720	0.501368	0.997874	0.667407	0.502717	5.203125
RBF SVM	0.288719	0.539529	0.537067	0.572785	0.554351	0.539529	25.703125
Random Forest	0.250000	0.500003	0.500003	1.000000	0.666669	0.500000	4.562500
GradientBossting Squared Error	0.009857	0.989764	0.985378	0.994283	0.989810	0.989764	0.718750
GradientBossting Friedman MSE	0.010129	0.989416	0.984097	0.994909	0.989473	0.989416	0.656250
HistGradientBoostingClassifier	0.037925	0.953055	0.934574	0.974319	0.954033	0.953055	1.156250
MLPClassifier	0.205163	0.668143	0.686995	0.617729	0.650523	0.668143	0.062500

# Finding optimal K value for KMeans clustering



# Results on clustered minority cases over sampled dataset:

	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Classifier							
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

# Comparison of Results:

	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Classifier							
Decision Tree	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Logistic Regression	0.001670	0.998325	0.000000	0.000000	0.000000	0.500000	0.000000
Linear SVM	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	1.265625
RBF SVM	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	8.781250
Random Forest	0.001672	0.998325	0.000000	0.000000	0.000000	0.500000	2.343750
GradientBossting Squared Error	0.003868	0.996132	0.005319	0.007042	0.006061	0.502417	0.171875
GradientBossting Friedman MSE	0.003550	0.996450	0.012270	0.014085	0.013115	0.506091	0.125000
HistGradientBoostingClassifier	0.002154	0.997854	0.000000	0.000000	0.000000	0.499764	0.171875
MLPClassifier	0.001677	0.998325	0.000000	0.000000	0.000000	0.500000	0.046875

	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Classifier							
Decision Tree	0.250000	0.499997	0.000000	0.000000	0.000000	0.500000	0.015625
Logistic Regression	0.226456	0.617247	0.634334	0.553660	0.591258	0.617247	0.015625
Linear SVM	0.250000	0.502720	0.501368	0.997874	0.667407	0.502717	5.203125
RBF SVM	0.288719	0.539529	0.537067	0.572785	0.554351	0.539529	25.703125
Random Forest	0.250000	0.500003	0.500003	1.000000	0.666669	0.500000	4.562500
GradientBossting Squared Error	0.009857	0.989764	0.985378	0.994283	0.989810	0.989764	0.718750
GradientBossting Friedman MSE	0.010129	0.989416	0.984097	0.994909	0.989473	0.989416	0.656250
HistGradientBoostingClassifier	0.037925	0.953055	0.934574	0.974319	0.954033	0.953055	1.156250
MLPClassifier	0.205163	0.668143	0.686995	0.617729	0.650523	0.668143	0.062500

	Brier loss	Accuracy	Precision	Recall	F1	Roc auc	Model prediction time
Classifier							
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

# Conclusion:

Following are the conclusions:

- Imbalanced dataset will not yield a good prediction model.
- Gradient Boosted Decision Trees has better performing prediction model among Decision Tree, Logistic Regression, Support Vector Machine, Random Forest, Histogram-based Gradient Boosted Decision Trees, and Multi-layer Perceptron for Credit Card Fraud dataset.

