Credit Card Default Fraud

Introduction

In this project, we will be implementing classification algorithms like Support Vector Machines, Decision Trees and Boosting Algorithm on Credit card payment default dataset. We will use k fold cross validation and parameter tuning to identify the best classification algorithm.

The dataset source of Credit card default dataset:
https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset

About Credit Card Default Dataset

Credit card fraud causes a huge amount of bad debt for banks. Detecting fraud such as credit card payment default in advance can save financial institutions from significant losses. Predictive analytics is extensively used by financial institutions to identify fraud instances and take action to avoid or mitigate such situation. I've chosen this dataset as it would help me explore how machine learning algorithms can be used for fraud prevention.

The dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.It has 30,000 observations and 25 attributes. The target variable is whether credit card user will default or not.

Attributes:

- ID: ID of each client
- LIMIT_BAL: Amount of given credit in NT dollars
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
- PAY_2: Repayment status in August, 2005 (scale same as above)
- PAY_3: Repayment status in July, 2005 (scale same as above)
- PAY_4: Repayment status in June, 2005 (scale same as above)
- PAY_5: Repayment status in May, 2005 (scale same as above)
- PAY_6: Repayment status in April, 2005 (scale same as above)
- BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
- BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
- BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
- BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
- BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
- BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
- PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
- PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)

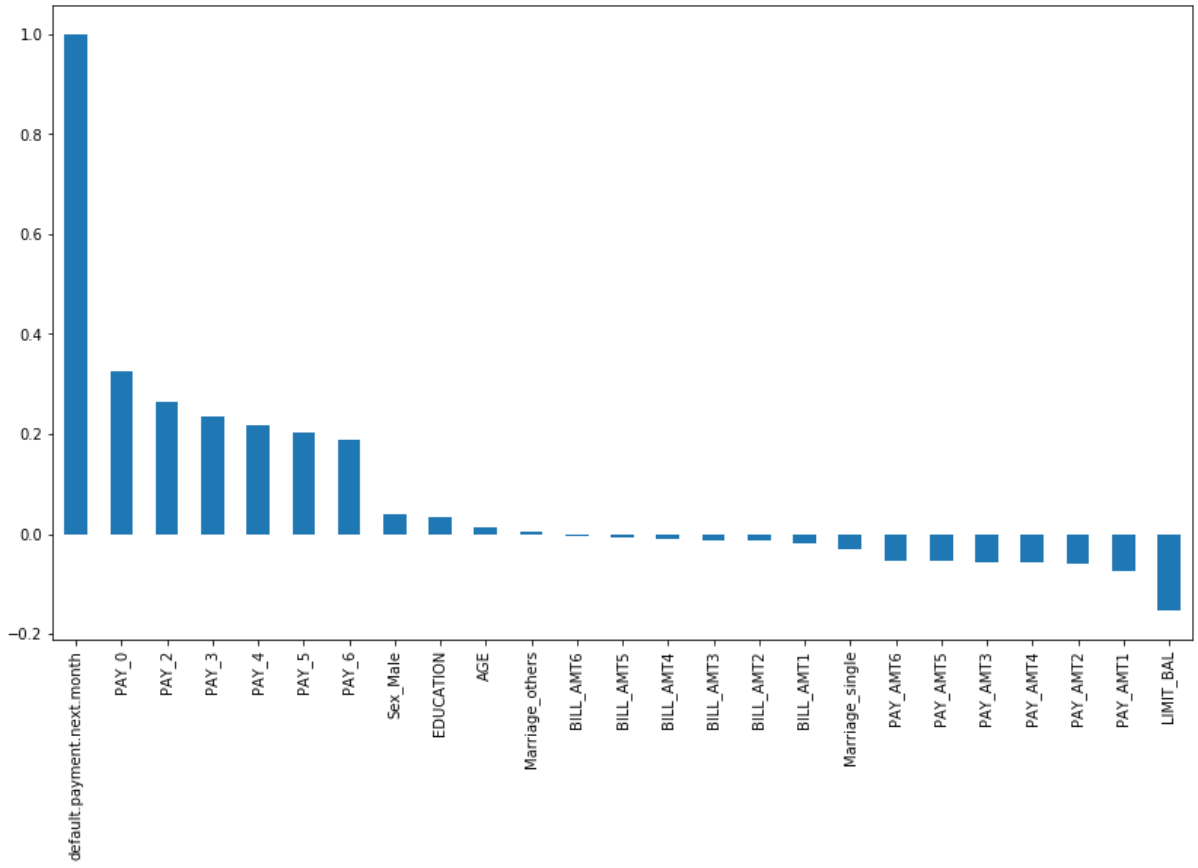
- PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
- PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
- PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
- PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
- default. payment. next. month: Default payment (1=yes, 0=no

Data Pre-Processing

1. Check for Null/Missing values: There are no missing values in the dataset.

2. Feature engineering: The categories of Education 4: others,5: unknown,6: unknown is grouped together under category 4. Marriage consists of category 0 which is grouped with category 3: others. Dummy variables for Marriage and Sex have been created. Education has been considered as ordinal variable and will be standardized with other numerical variables.

3.Correlation: No feature is highly correlated with the target variable as seen below.



4. Feature scaling: Numerical variables are standardized using formula: $z = \frac{x-u}{\sigma}$ for SVM as it is a distance-based algorithm.

Finally, the dataset is split into training set and test set in 80:20 ratio.

Algorithms and Analysis

Implemented 3 classification algorithms which are Support Vector Machine, Decision Trees and Boosting using Adaboost algorithm. Performed 10-fold cross validation to find the best model with lowest bias and variance that generalizes well with unseen data.

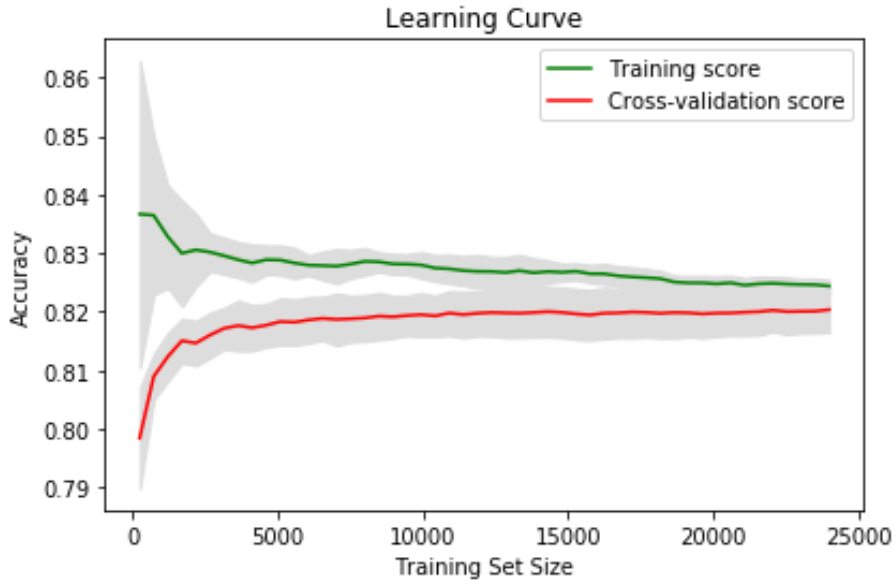
Support Vector Machines (SVM)

Applying SVM to classify the target variable. Experimenting with 3 different kernels to find the best function to classify data. The kernels used are:

- Linear kernel: The kernel tries to classify the target variable using linear decision boundary.
- Polynomial Kernel: The kernel tries to classify the target variable using 3rd degree polynomial decision boundary.
- Radial Basis Function Kernel: The kernel tries to classify the target variable using radial decision boundary.

SVM Kernel	Credit card default Validation Accuracy
Linear Kernel	80.6%
Polynomial Kernel	80.4%
RBF kernel	81.7%
	Credit card default Test Accuracy
Linear Kernel	81%
Polynomial Kernel	81%
RBF kernel	82%

The best SVM model is model with RBF kernel with highest test accuracy.
Learning curve for RBF SVM model of Credit card default dataset:



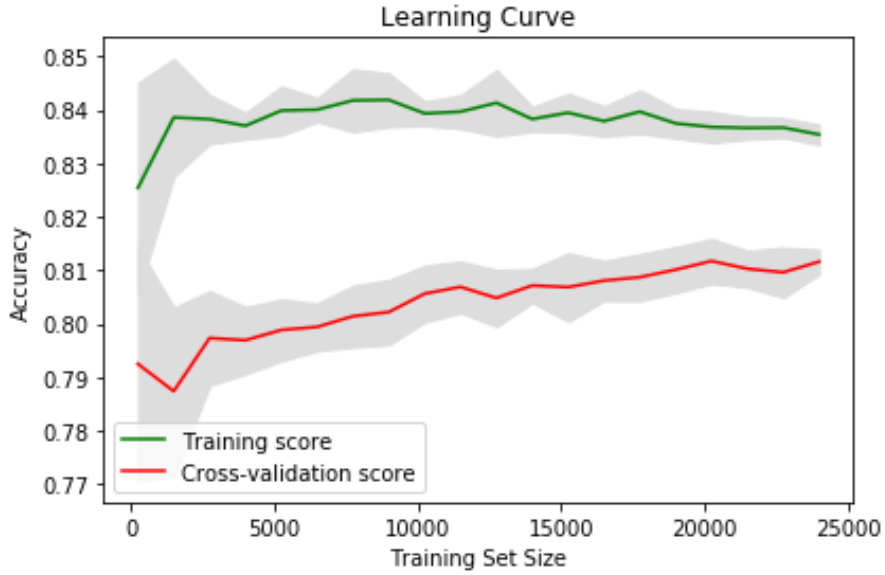
From learning curves, we can see that as training accuracy reduces with increasing sample size, validation accuracy increases. This indicates that our model is good and will generalize well.

Decision Trees

Applying Decision Trees to classify the target variable. Experimenting with 2 different methods of node splitting which are Gini and Entropy. Applying pruning to the best tree obtained by removing features which do not contribute to classification. Applied grid search with 10-fold cross validation to find the best depth of the pruned tree and avoid overfitting.

Decision tree	Credit card default Validation Accuracy
Entropy tree	72%
Gini tree	72%
Pruned tree	80%
Decision tree	Credit card default Test Accuracy
Entropy tree	73%
Gini tree	73%
Pruned tree	81%

The best parameters for the pruned tree are:
Best Parameters: {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 20}
Pruning has improved the test accuracy and model fit. But from learning curve we observe pruned tree has high variance. Hence the model will not generalize well.

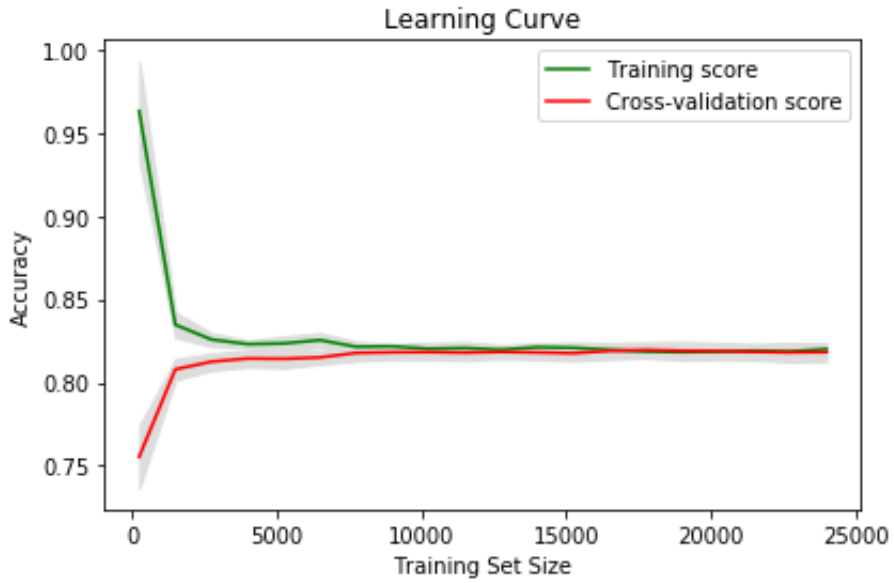


Boosting

Boosting is the algorithm which is an iterative process to train weak learners to generate the prediction rule. I have used Adaboost algorithm that is a successful boosting algorithm developed for binary classification. I also performed pruning on the Adaboost algorithm using grid search technique to find the best parameters and 10-fold cross validation to avoid overfitting.

Model	Credit card default Validation Accuracy
Adaboost tree	81%
	Credit card default Test Accuracy
Adaboost tree	82%

Learning curve for boosted model:



From learning curves, we can see that as training accuracy reduces with increasing sample size, validation accuracy increases. This indicates that our model is good and will generalize well.

Summary Table

Model	**Credit Card Default Test Accuracy**
SVM (RBF Kernel)	**82%**
Decision Tree	**73%**
Adaboost	**82%**

Conclusion

After performing all the three classification algorithms on both the datasets, we can conclude that Adaboost tree provides best accuracy (~82%) with lowest standard deviation of accuracies from cross validation, so is the best model to classify the dataset.