# Credit Card Default Prediction

LOGAN LIU

MATTHEW PENG

NEHA ANNA JOHN

QINPEI ZOU

VIKRANT VAIDYA

TEXAS McCombs
The University of Texas at Austin
McCombs School of Business

# Outline

- Introduction
- Exploratory Data Analysis
- Modeling & Insights
- Conclusion

# Introduction

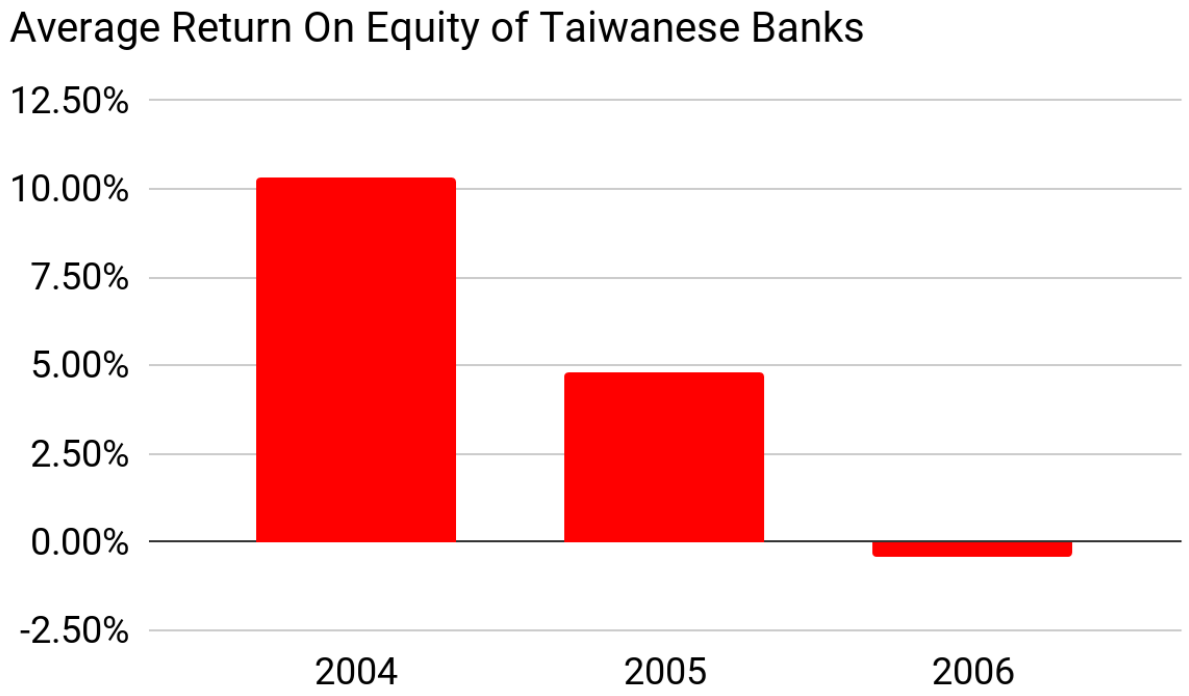
# Problem Statement

We are helping a Taiwanese bank to reduce loss caused by credit card payment defaults

Imagine you have

**USD 33,000 of credit card bills**

that dues next month



# Exploratory Data Analysis

# Key Variables

Predictor variables

- ID of each credit card client
- Demographic information: Sex, Education, Marriage, Age
- Repayment status on credit card bill for 6 months (Apr2005-Sep2005)
- Credit card bill statement for 6 months (Apr2005-Sep2005)
- Credit card repayment for 6 months (Apr2005-Sep2005)

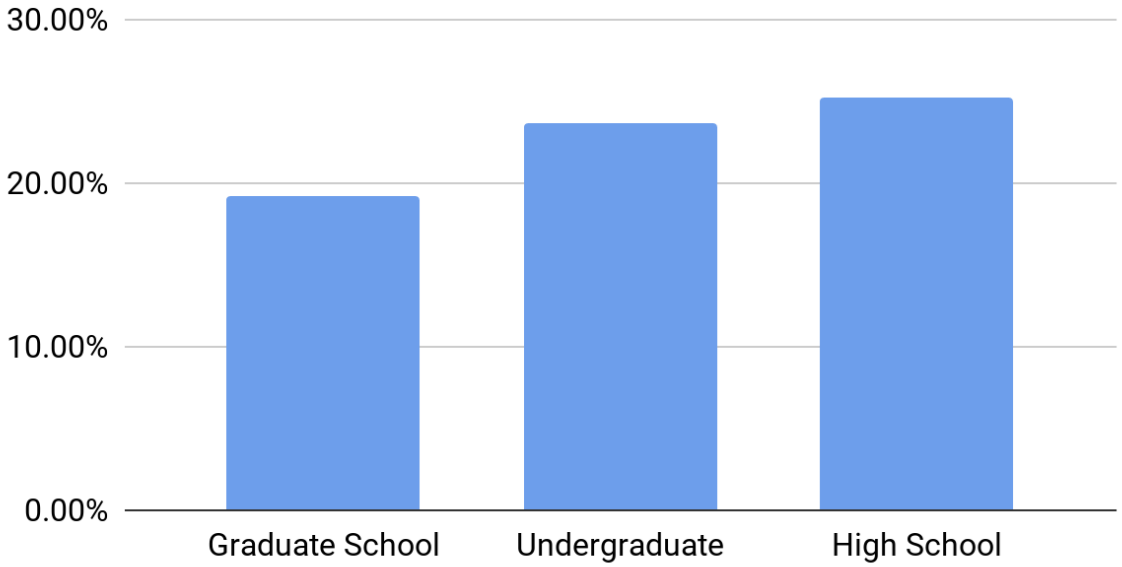
Output variables

Default payment flag for Oct 2005

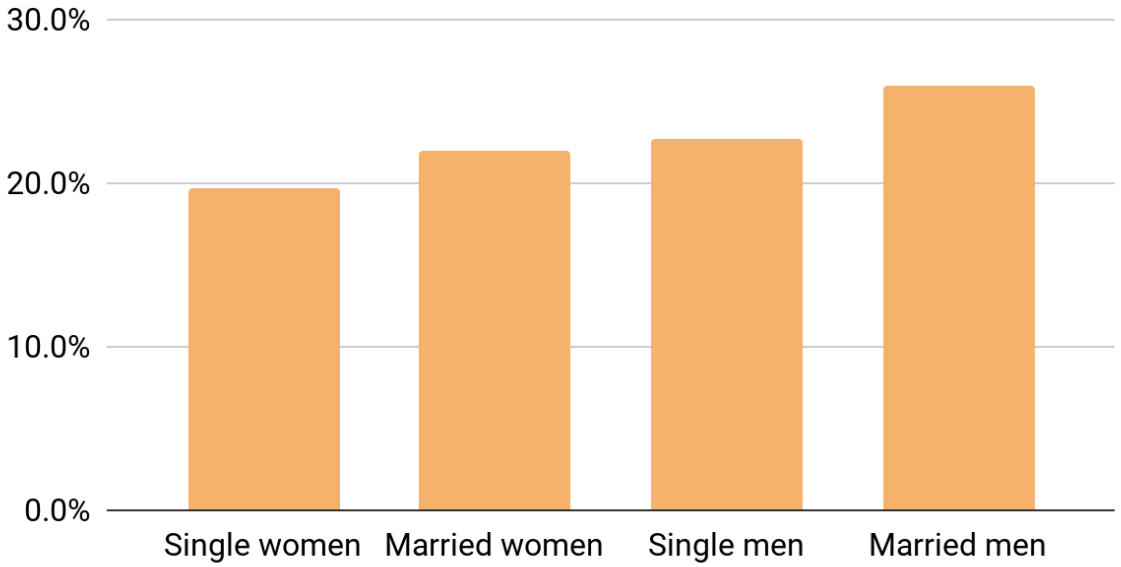
# Exploratory Data Analysis

Key findings:

Default Rate by Education




Default Rate by Marital Status/Gender



# Modeling & Insights

# Methodology

Implemented multiple models on training data:

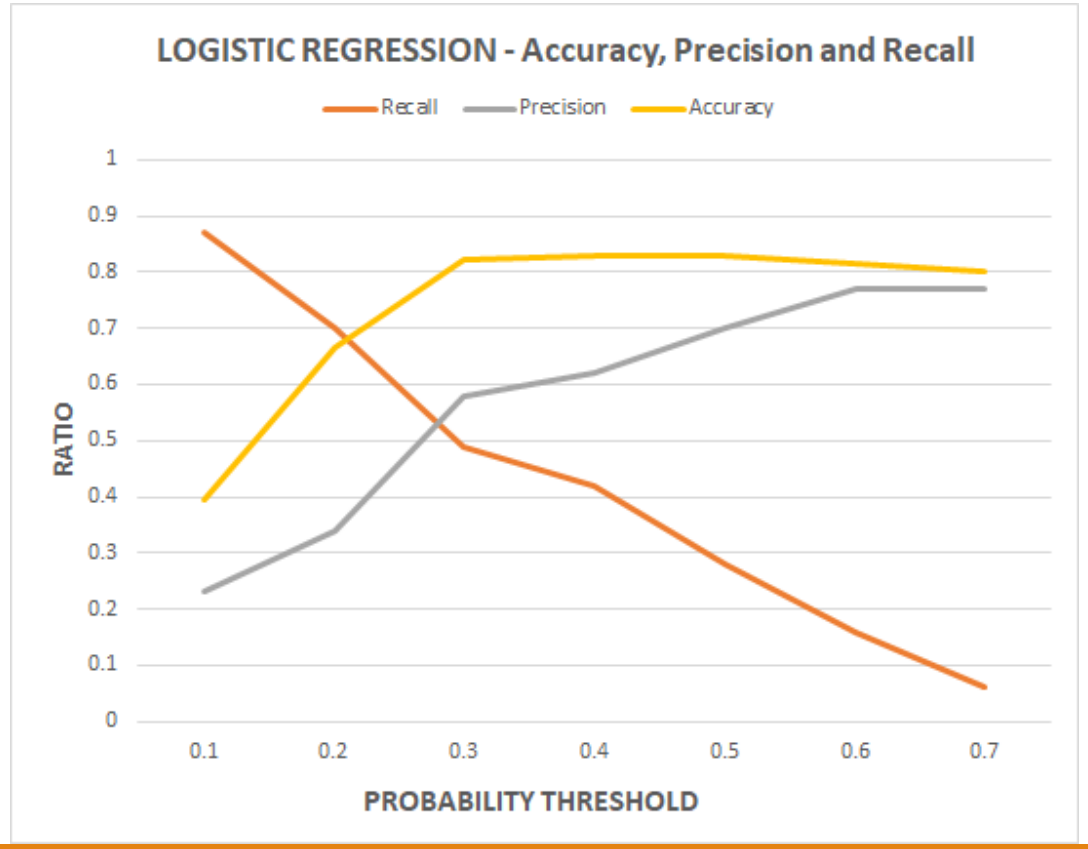
- Logistic Regression
- KNN Classifier
- Decision Tree, Random Forest and Gradient Boosting

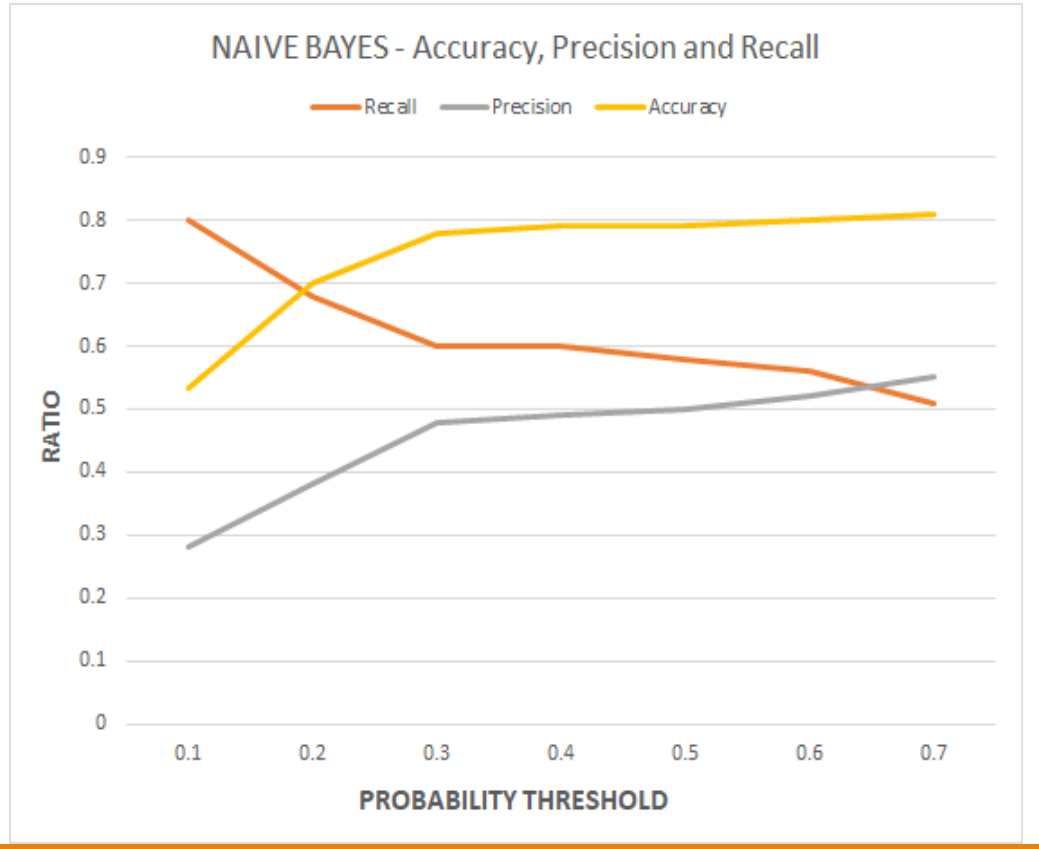
Evaluated key metrics for model performance on test data:

- Precision-Recall
- Misclassification rate
- Confusion Matrix

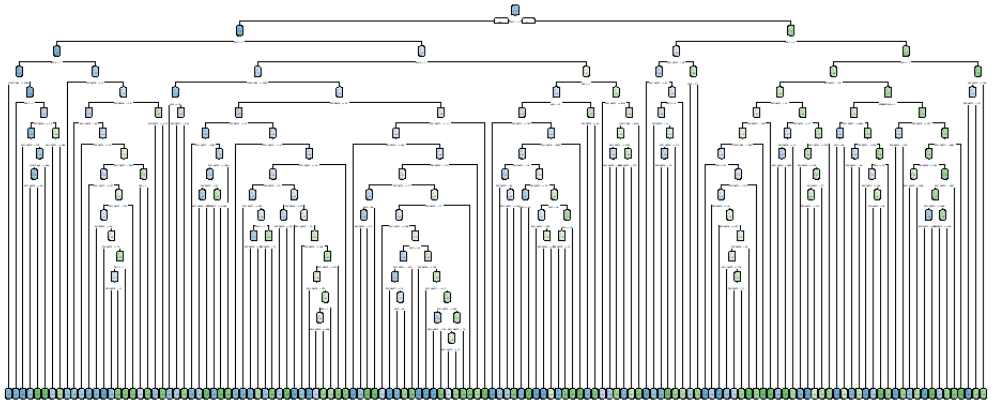
# Logistic Regression And Naive Bayes

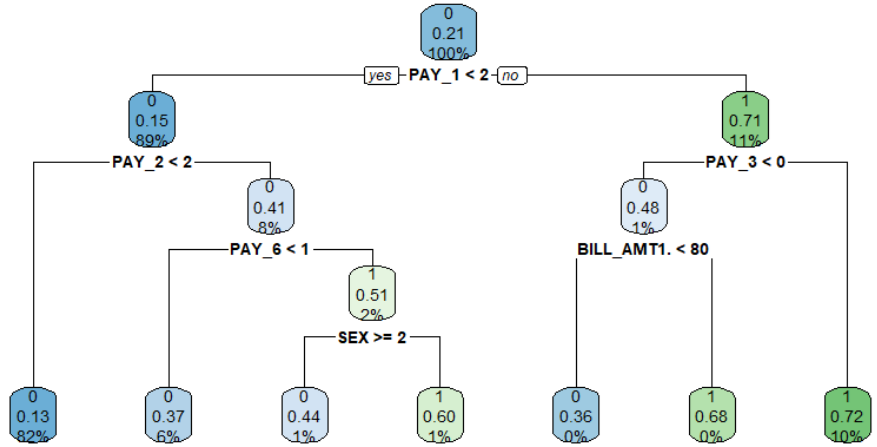
Used Lasso cross validation for variable selection - **LIMIT BAL, SEX, MARRIAGE, PAYMENT_FLAGS, BILL AMT** for last month and **PAYMENT AMTS** for last two months came out to be as the significant variables



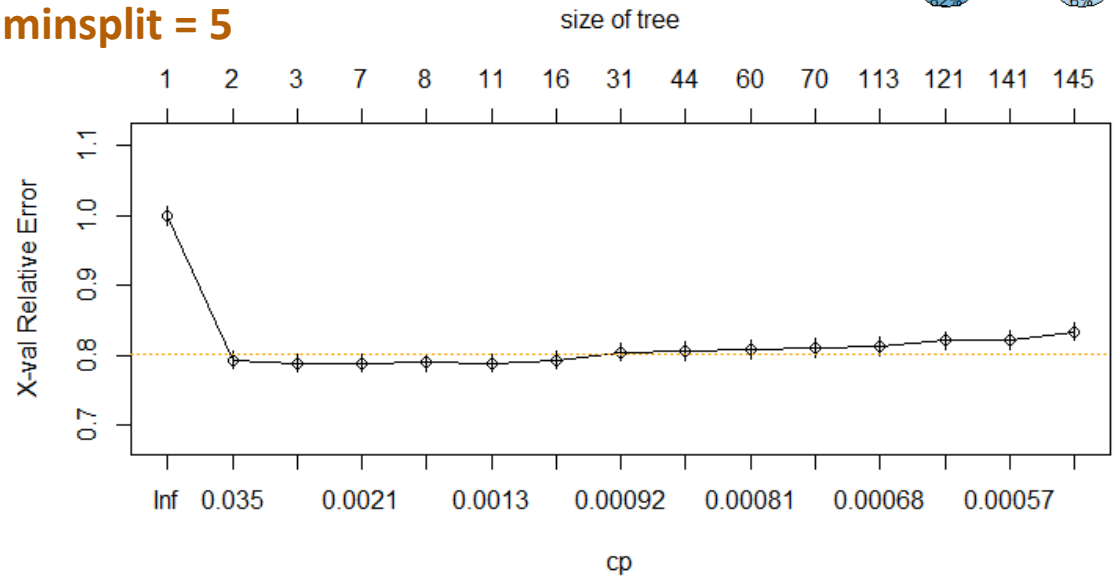



# Decision Tree: Analysis & Insights





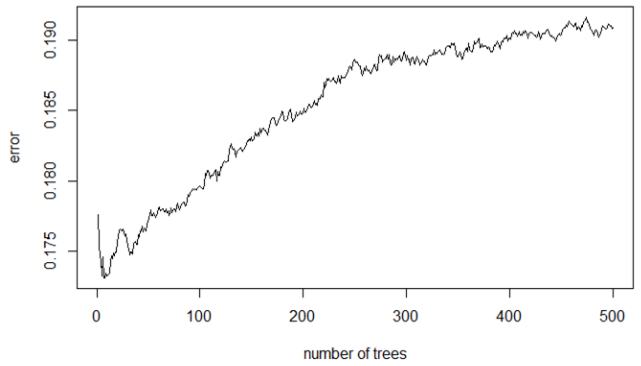

**minsplit = 5**



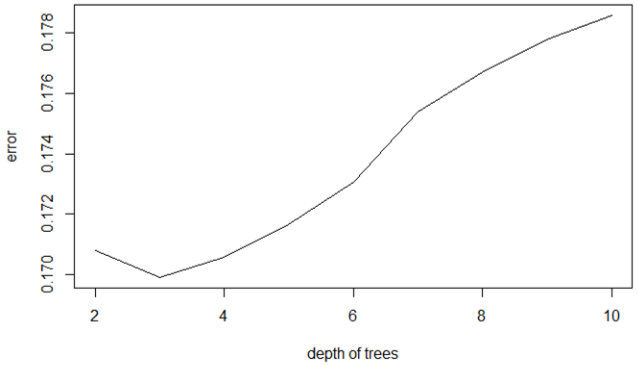

***Finding the best α (cost-complexity) = 0.002038 using cross validation***

# Gradient Boosting: Analysis & Insights

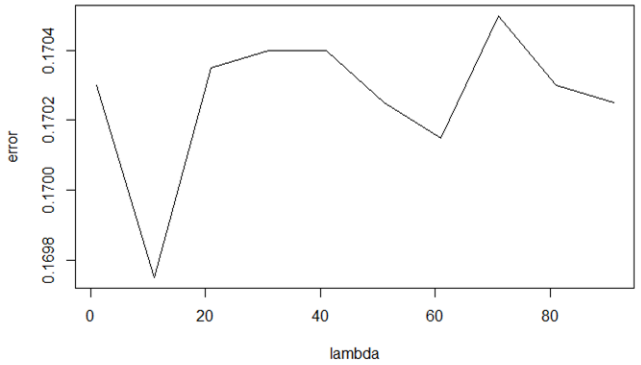
nround: 300 -> 10




depth of tree: 6(default) -> 3




lambda: 1(default) -> 10




Misclassification Rate:

0.180

↓

0.159

or…

eta: 0.3(default) ->0.05

other parameters remain the same

Misclassification Rate:

0.180

↓

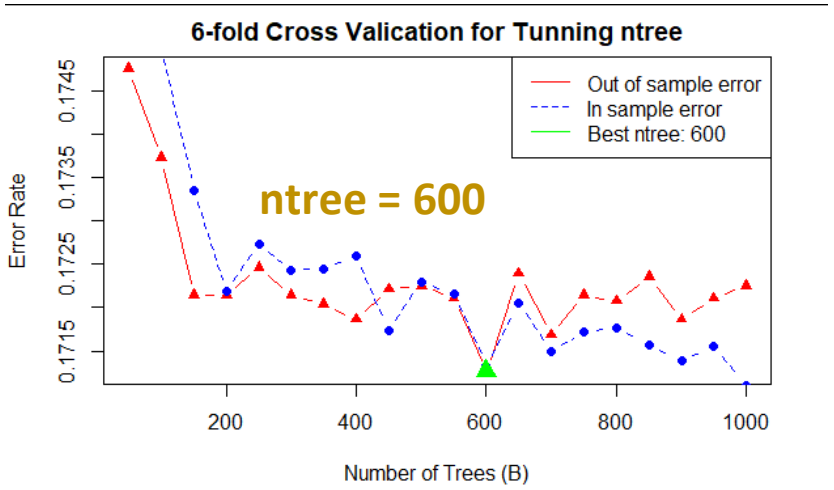
0.162

maybe next time we can start with eta?

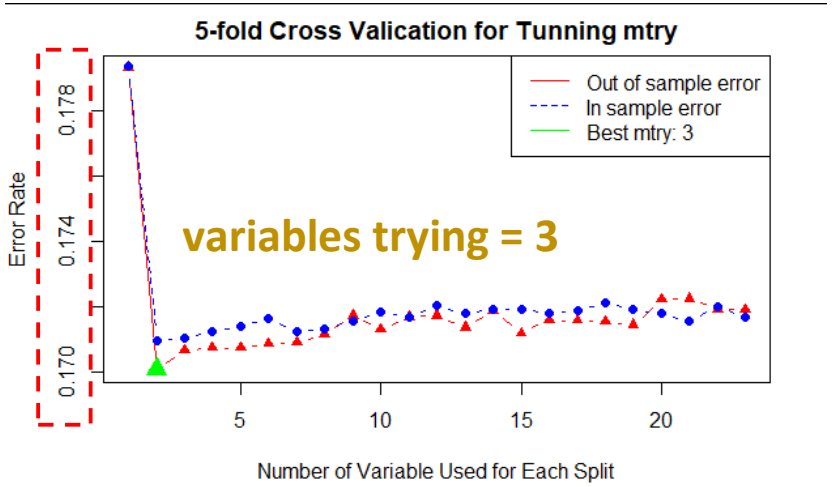
things to consider:

1. computing time
2. too larger -> miss the optimization
3. too small -> stuck in local optimization

# Random Forest: Analysis & Insights

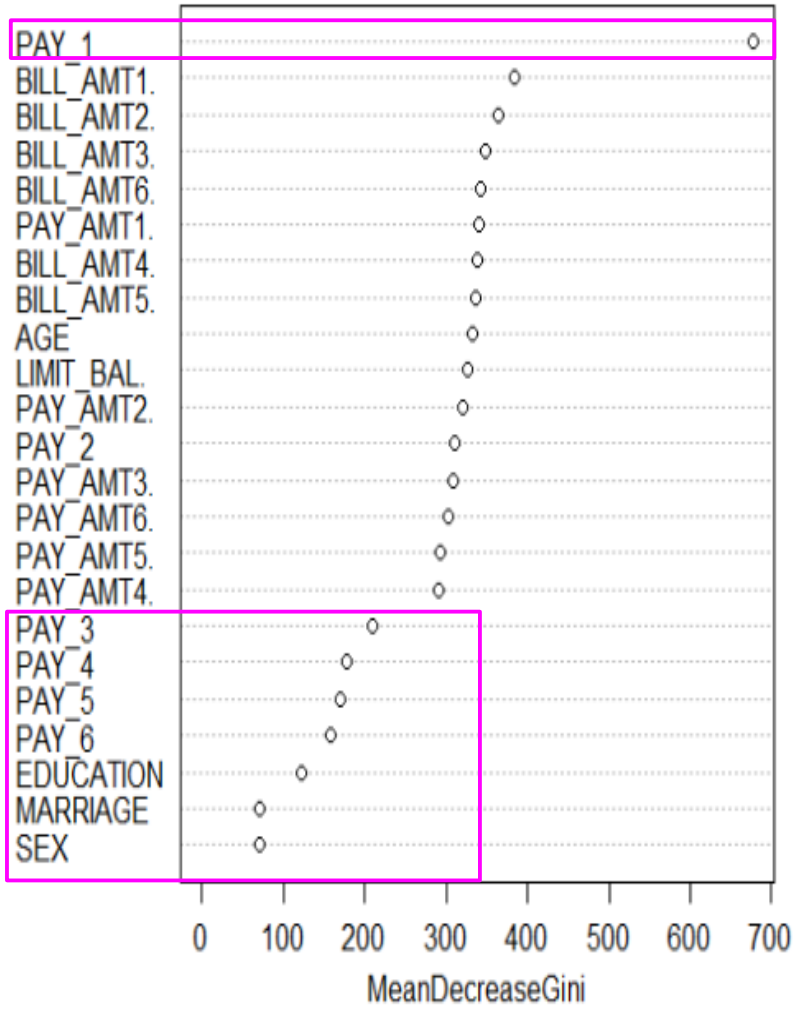



***One Predictor provides a lot of information***




| Pred / Actual | Non default(0) | Default(1) |
|---|---|---|
| Non-default(0) | 6301 | 1066 |
| Default(1) | 319 | 680 |

# Random Forest: Analysis & Insights



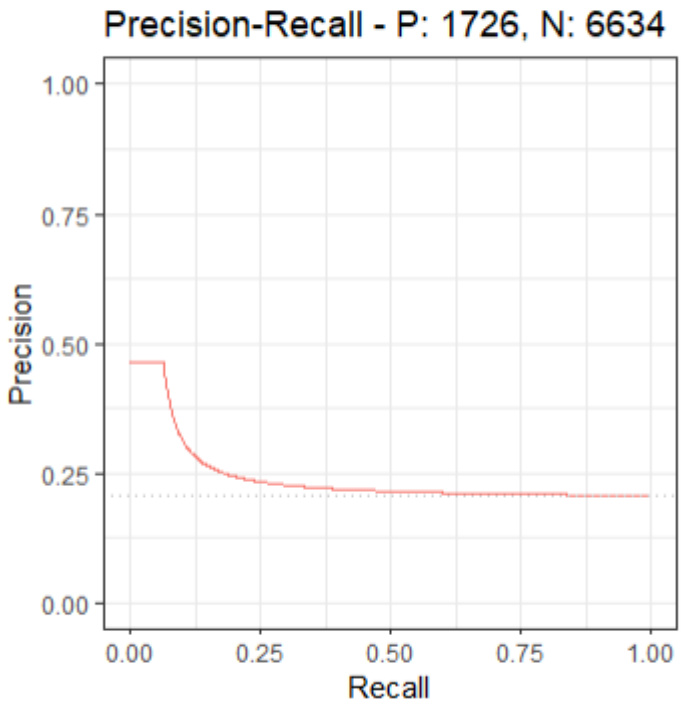

- Most of variables contribute similar weights to prediction of default
  - Error rate doesn't change much with number of variable trying
  - many predictors share similar mean decrease Gini
- One predictor strong influent prediction of default (Pay 1)
  - Pay ***1*** = ***5*** means the person ***1*** month ago delay the payment for ***5*** months
- Many variables are redundant information (Pay 3 - Pay6)
  - (Pay 2 = 4) ∈ (Pay 1 = 5)
- Confliction from EDA: Education doesn't contribute much information

| Pay.1 <int> | Default <dbl> |
|---|---|
| -2 | 0.08627608 |
| -1 | 0.16751453 |
| 0 | 0.12902997 |
| 1 | 0.28649802 |
| 2 | 0.69765625 |
| 3 | 0.76470588 |
| 4 | 0.70270270 |
| 5 | 0.54166667 |
| 6 | 0.60000000 |
| 7 | 0.77777778 |

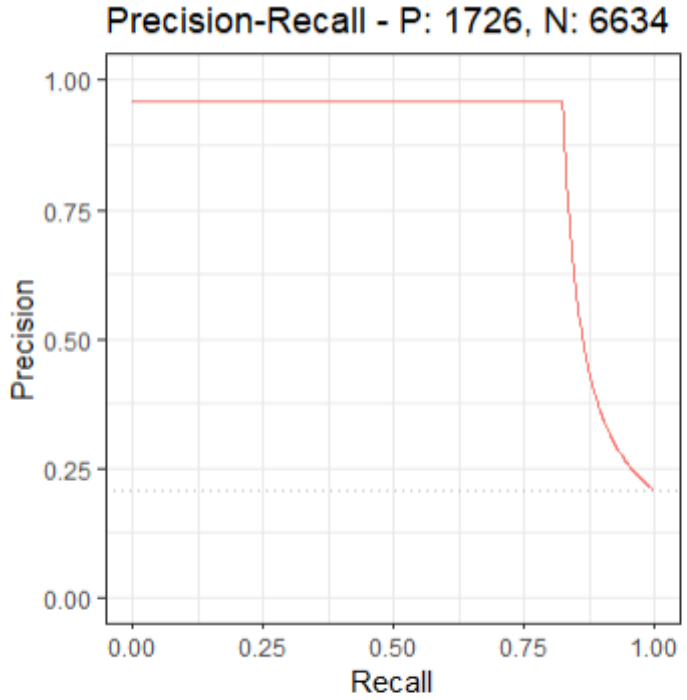
| Education <int> | Default <dbl> | |
|---|---|---|
| 1 | 0.1749217 | Grad School |
| 2 | 0.2288528 | College |
| 3 | 0.2428918 | Hign School |

# KNN Classifier: Analysis & Insights

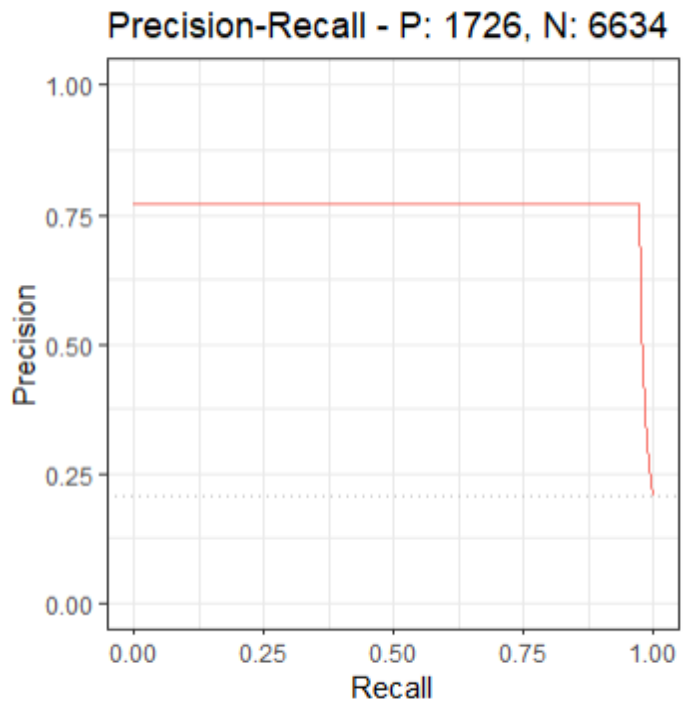
**KNN without normalization**




**KNN with normalization**




**KNN with normalization and SMOTE**



# KNN Classifier: Analysis & Insights

Normalization of continuous variables highly effective for KNN Classifier:

- Reduced misclassification rate by about 15%
- More than doubled precision and improved recall significantly

SMOTE technique:

- Reduced precision marginally
- Produced highest recall across all techniques

More important to correctly identify defaulters vs others-Recall:

- SMOTE delivers best on Recall

# Conclusion

# Output Summary

| Model Name | Precision | Recall | Misclassification Rate |
|---|---|---|---|
| Logistic Regression | 0.71 | 0.29 | 0.17 |
| Naive Bayes | 0.50 | 0.58 | 0.21 |
| KNN Classifier with Normalization | 0.95 | 0.82 | 0.043 |
| KNN Classifier with SMOTE | 0.77 | 0.97 | 0.066 |
| Decision Tree | 0.35 | 0.67 | 0.17 |
| Random Forests | 0.39 | 0.64 | 0.17 |
| Gradient Boosting without tuning | 0.594 | 0.407 | 0.180 |
| Gradient Boosting with tuning | 0.695 | 0.406 | 0.159 |

# Recommendations

- Leverage Recall as the primary metric for evaluating model performance
- KNN Classifier with SMOTE provides best performance and can help predict the maximum number of probable credit defaulters
- Next Steps:
  - Build a stronger model leveraging data from last quarter/shorter period rather than half year

# Appendix

# KNN Classifier

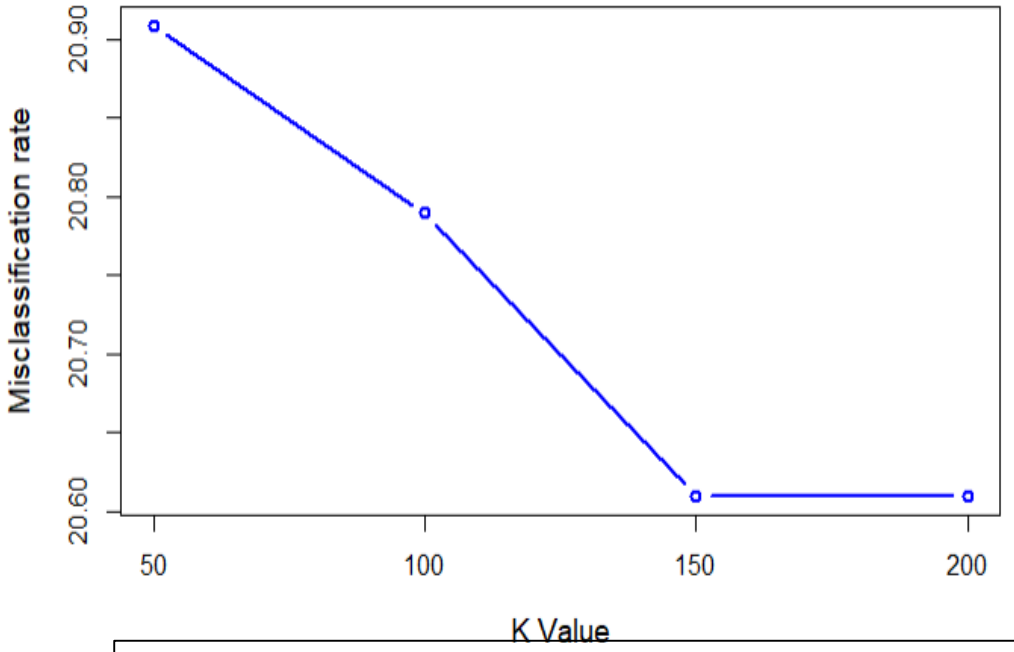
# Findings: K=20,K=50

| Metric | Without normalization | With normalization | With SMOTE |
|---|---|---|---|
| Precision | 0.47 | 0.96 | 0.84 |
| Recall | 0.09 | 0.79 | 0.98 |
| Misclassification rate | 21.4 | 3.09 | 4.24 |

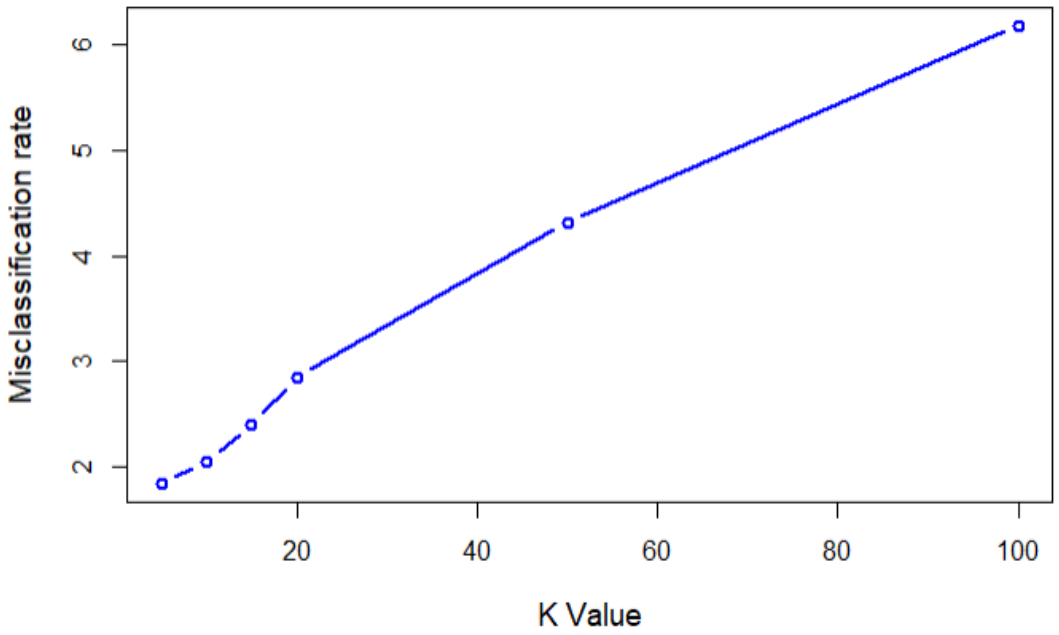
| Metric | Without normalization | With normalization | With SMOTE |
|---|---|---|---|
| Precision | 0.46 | 0.95 | 0.77 |
| Recall | 0.06 | 0.82 | 0.97 |
| Misclassification rate | 20.86 | 4.35 | 6.58 |

# Findings: Best K value

***Finding the best K value for KNN with and without normalization***

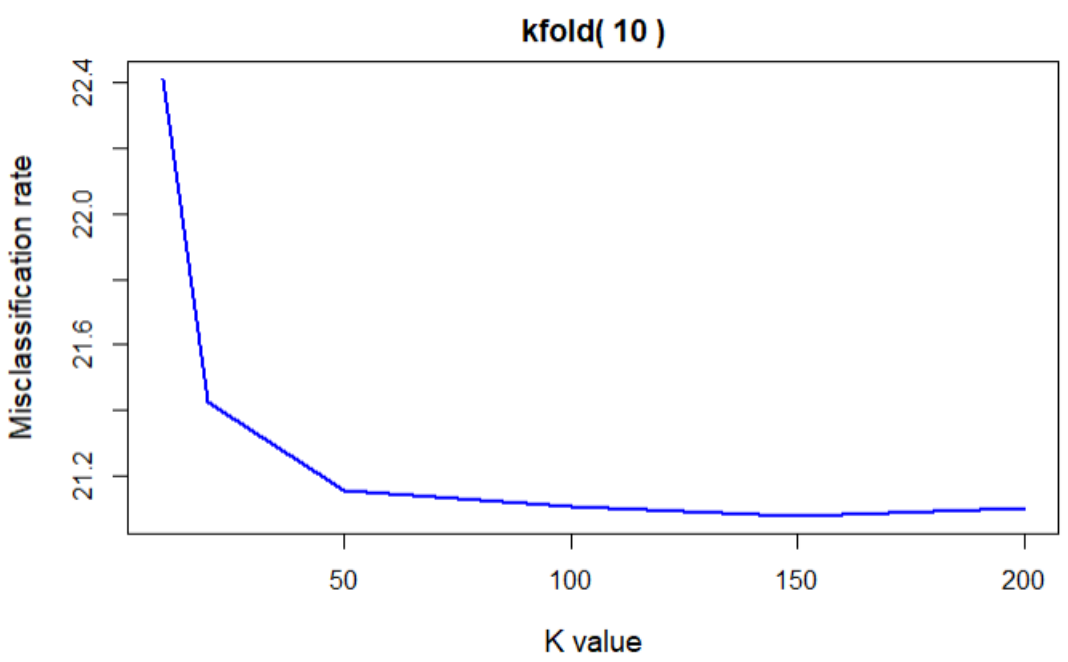


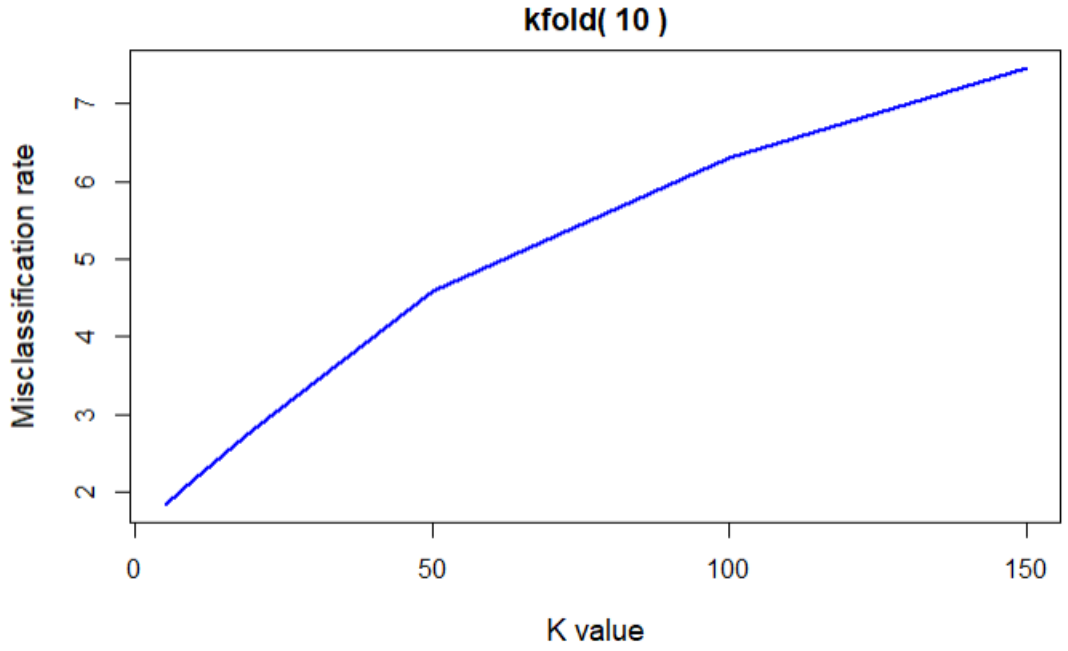

***K value 200 with misclassification rate 20.61***




***K value 5 with misclassification rate 1.84***

# Findings: KCV=10




***K fold Cross Validation=10 for data with and without normalization***

# Gradient Boosting

# Gradient Boosting: Importance

| Feature <chr> | Gain <dbl> | Cover <dbl> | Frequency <dbl> |
|---|---|---|---|
| PAY_1 | 0.7634134859 | 0.3333333306 | 0.14285714 |
| PAY_2 | 0.1660897340 | 0.2980205198 | 0.14285714 |
| PAY_4 | 0.0181468821 | 0.1097148970 | 0.05714286 |
| LIMIT_BAL. | 0.0163464585 | 0.1074236377 | 0.05714286 |
| PAY_3 | 0.0136124543 | 0.0522952944 | 0.14285714 |
| PAY_6 | 0.0095069087 | 0.0372762791 | 0.18571429 |
| PAY_5 | 0.0050551374 | 0.0374183922 | 0.05714286 |
| BILL_AMT1. | 0.0025109365 | 0.0104823072 | 0.04285714 |
| PAY_AMT4. | 0.0020528418 | 0.0008665749 | 0.05714286 |
| BILL_AMT5. | 0.0015292959 | 0.0006500527 | 0.04285714 |