Applying Machine Learning Algorithms to Predicting Loan Default Risk

DATS 6202
Final Project Group 5
Kimberly Kreiss, Bijiao Shen & Zhoudan Xie

Introduction

- Home Credit
- Wants to use provided data to predict whether or not a client will default
- Operates in Eastern European countries and Asia as well as the US
    - Makes loans to people with little or no credit

Data Description

Source: Kaggle “Home Credit Default Risk” competition

Format: CSV

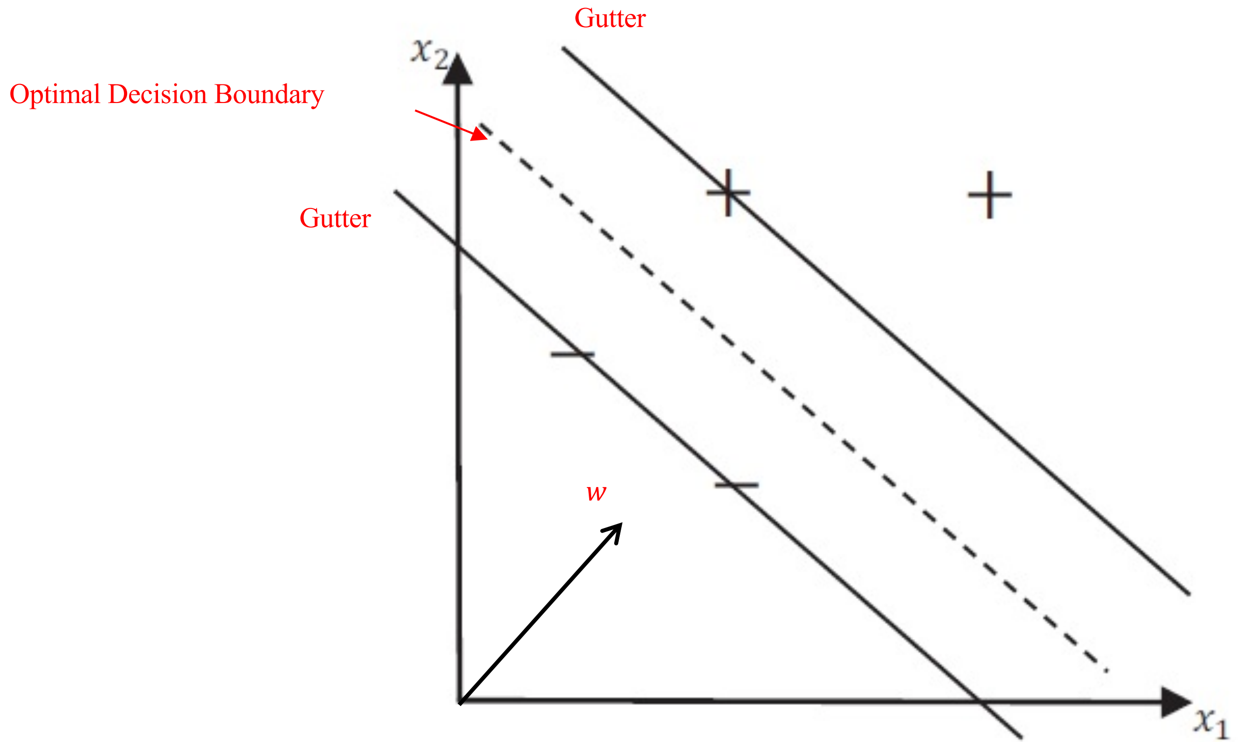
Objective: Predict whether or not borrowers repay the loan

Target: Target - Categorical (0 or 1)

Predictors: Income and age - Numerical

Loan Type, Residual Condition - Categorical

Support Vector Machine



Support Vector Machine

$$Width\ of\ the\ street = \frac{2}{||w||}$$

Gutter equation: $y_i(w \cdot x_i + b) - 1 = 0$

$$\Downarrow$$

$$w = \sum \alpha_i y_i x_i$$

$$\sum \alpha_i y_i = 0$$

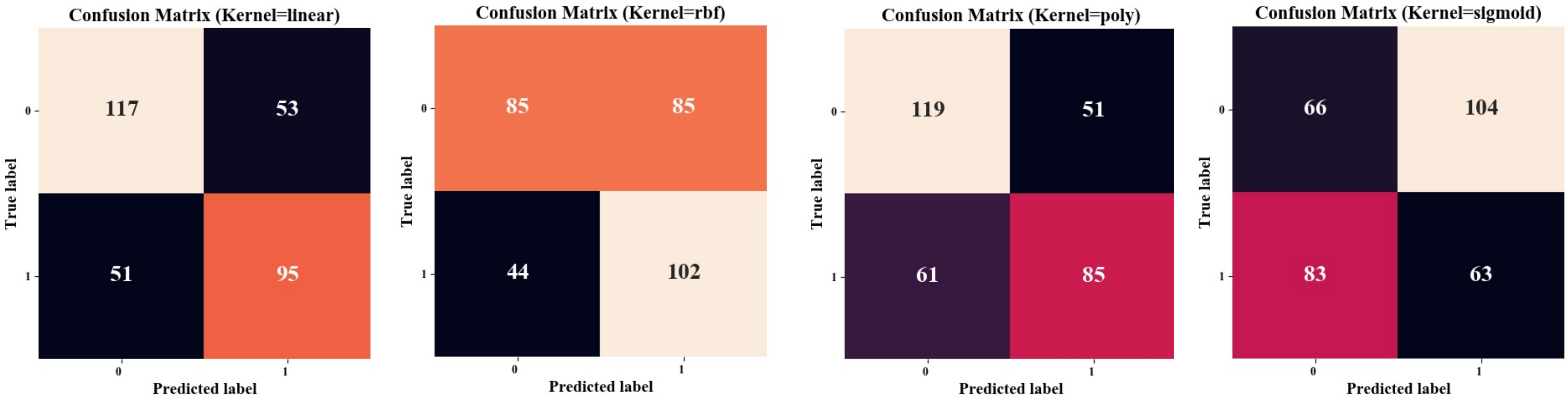
Experimental Setup for SVM

- Resample the data
- Select important features
- Split the data
- Train the data by using various kernels
    - Linear, RBF, polynomial, sigmoid
- Test the performance
    - Accuracy score, confusion matrix, ROC curve

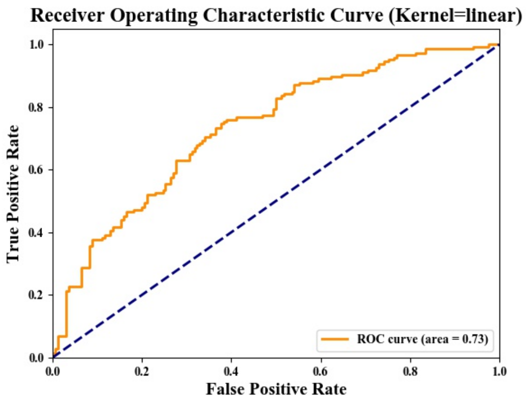
SVM Results: Classification Reports

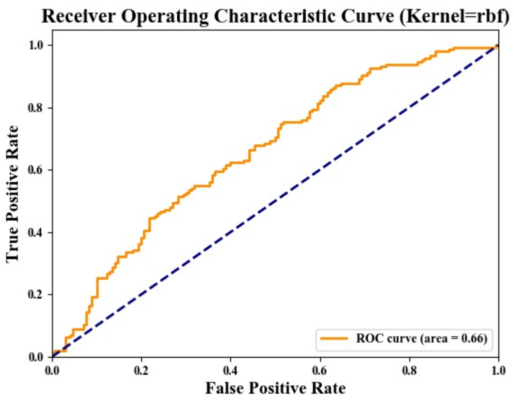
	Precision	Recall	F1-score	Support
Kernel=linear				
0	0.70	0.69	0.69	170
1	0.64	0.65	0.65	146
Avg/ total	0.67	0.67	0.67	316
Kernel=rbf				
0	0.66	0.50	0.57	170
1	0.55	0.70	0.61	146
Avg/ total	0.61	0.59	0.59	316
Kernel=poly				
0	0.66	0.70	0.68	170
1	0.62	0.58	0.60	146
Avg/ total	0.64	0.65	0.64	316
Kernel=sigmoid				
0	0.44	0.39	0.41	170
1	0.38	0.43	0.40	146
Avg/ total	0.41	0.41	0.41	316

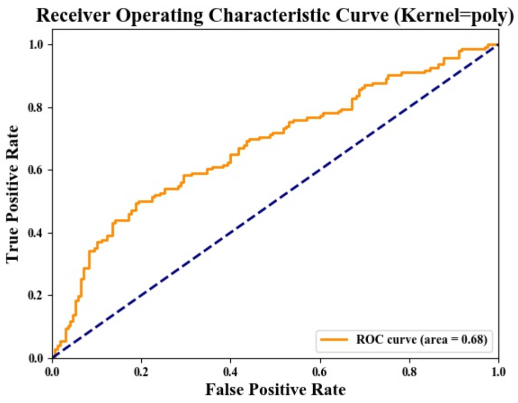
SVM Results: Confusion Matrix

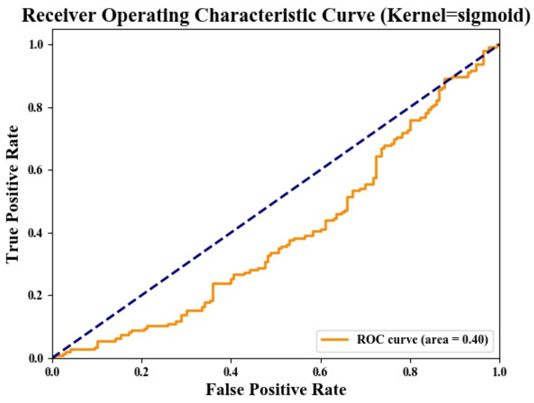


SVM Results: Confusion Matrix









Naïve Bayes

Classifier Overview:

1. A simple “pure statistical” technique relies on conditional probability.
2. Fast and accurate given its “Naïve Assumptions” of independence.
3. Naïve Bayes equation:

$$\underset{Y}{\operatorname{argmax}} P(Y|X) = \underset{Y}{\operatorname{argmax}} P(x_1|Y)P(x_2|Y)\cdots P(x_n|Y)$$

4. Bayes Rule: A mathematical relation between prior probability and posterior probability.

Naïve Bayes

Classifier Application of Credit Risk:

- Classifier Specification: Gaussian Naïve Bayes
- Steps:

  Data Processing and Train-Test Split

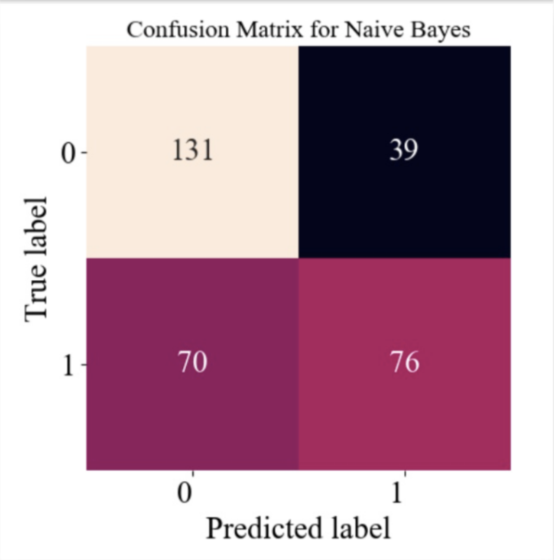
  Fit Model Using GaussianNB

  Model Performance Assessment

- Model Performance Assessments: Acceptable Classification Results

Naïve Bayes

Model Performance - Confusion Matrix



We have two classes in the analysis: 1 represents client with payment difficulties while 0 represents all other cases. The Confusion Matrix shows that the we correctly classify 131 loans and 76 loans about its repayment performance. We classify 39 actual performing loans as loans with repayment difficulties and classify 70 actual loans which have repayment difficulties as performing loans.

Naïve Bayes

Model Performance - ROC and Classification Report

```
Classification Report:
             precision    recall  f1-score   support

        0.0       0.65      0.77      0.71       170
        1.0       0.66      0.52      0.58       146

avg / total       0.66      0.66      0.65       316


Accuracy :  65.50632911392405


ROC_AUC :  71.55519742143433
```

We have a 65.51% accuracy;
we have 0.65 precision for estimating class 0 which is clients without payment difficulties and 0.66 precision for classifying class 1 which is clients with payment difficulties.

Random Forest Model

- Ensemble Method
- Averages across several decision trees that each use random pulls of the dataset
- This allows to correct for overfitting that often results from decision trees

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

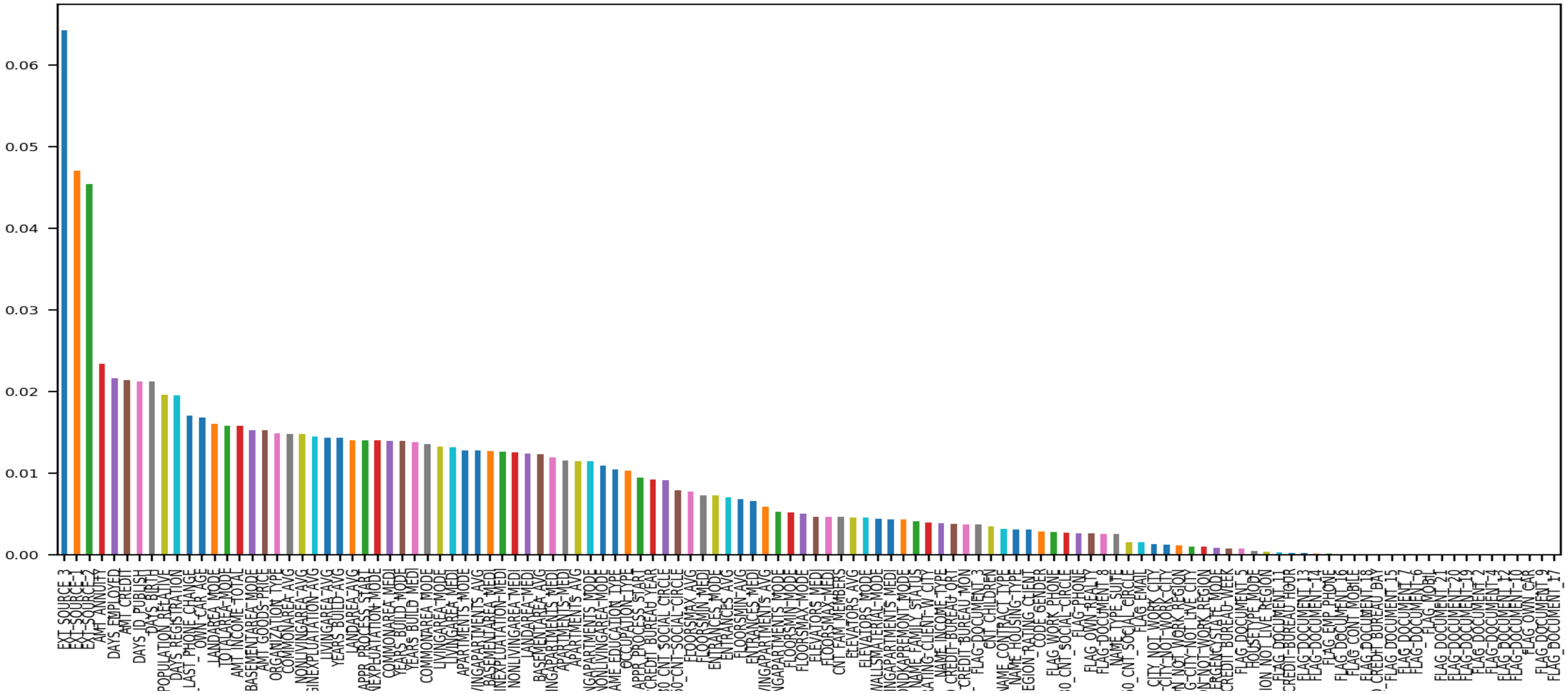
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

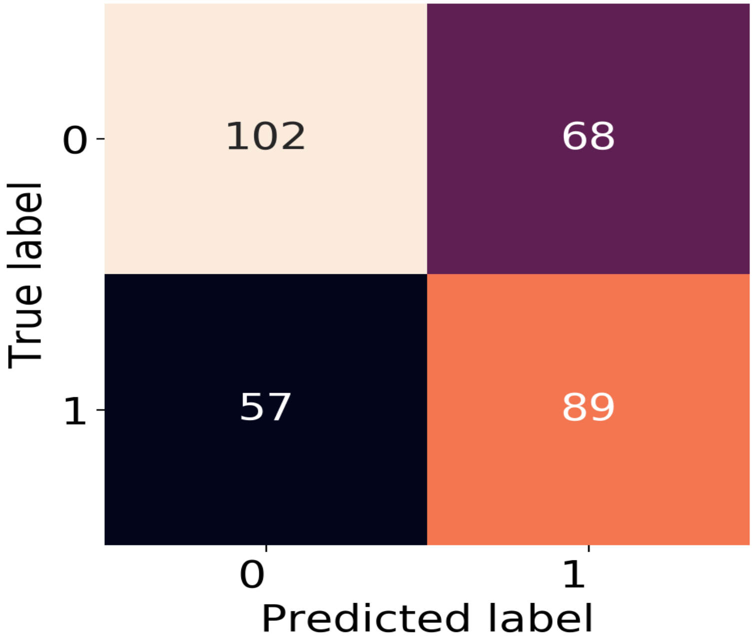
Feature importance



Classification report (all features)

```
Results Using All Features:

Classification Report:
             precision    recall  f1-score   support

          0       0.64      0.60      0.62       170
          1       0.57      0.61      0.59       146

avg / total       0.61      0.60      0.60       316



Accuracy :  60.44303797468354


ROC_AUC :  65.03827558420629
```

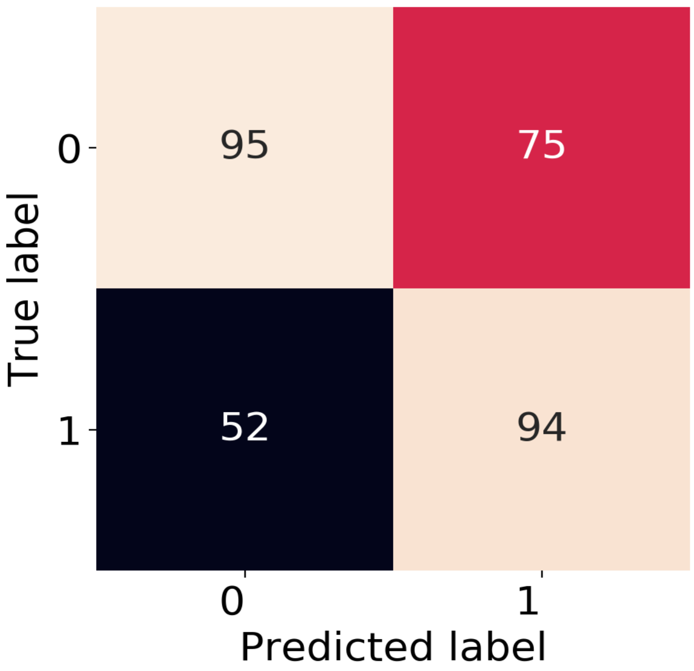
Confusion matrix (all features)



Classification report (top 25 features)

```
Results Using K features:

Classification Report:
             precision    recall  f1-score   support

          0       0.65      0.56      0.60       170
          1       0.56      0.64      0.60       146

avg / total       0.60      0.60      0.60       316



Accuracy :  59.81012658227848


ROC_AUC :  65.75543916196615
```

Confusion Matrix (top 25 features)



Final results

- These models are not great
- Could be improved by using more features
- Could be improved by testing for correlation in the k-features model and choosing features that are not correlated