Predictive Analysis of Customer Credit Data

Yousuf Hasan Siddiqui
Orhan Ipek
Shikhar Srivastava

Statistical Programming Language
Humboldt–Universität zu Berlin

Outline

1. Introduction
2. Motivation
3. Data Set
4. Descriptive Statistics
5. Encoding Data
6. Predictive Modelling
7. Results

Introduction

- Constantly increasing outstanding consumer loans is one of the biggest problems in financial industry.

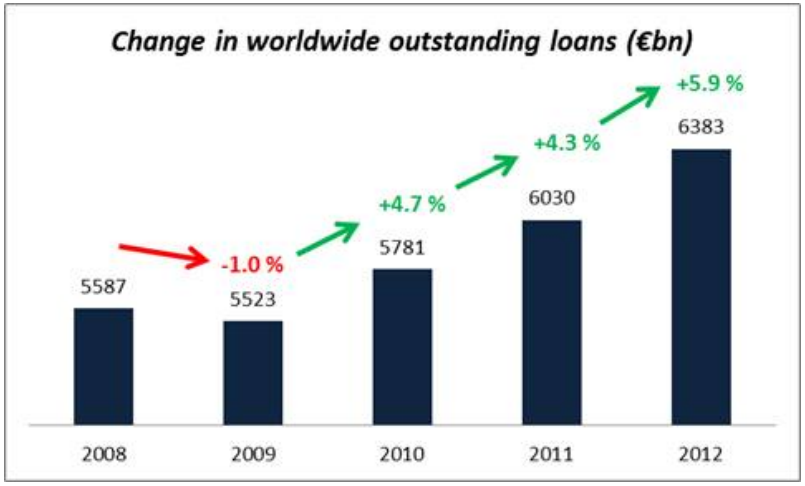


Figure 1: Sources: Central banks, Asterés, Crédit Agrole

- The credit approval decision involves the manual categorization of customers using various scoring and risks calculations.

Motivation

- Improvement in the computing machines and machine learning algorithms are swiftly adopted by the financial sector.
- Predictive analytics gives us the opportunity to use vast amounts of data to find trends and patterns and make future predictions.
- Efficient and rapid decision process
- Using machine learning helps as the algorithm updates itself without extra need of parameter calculation.

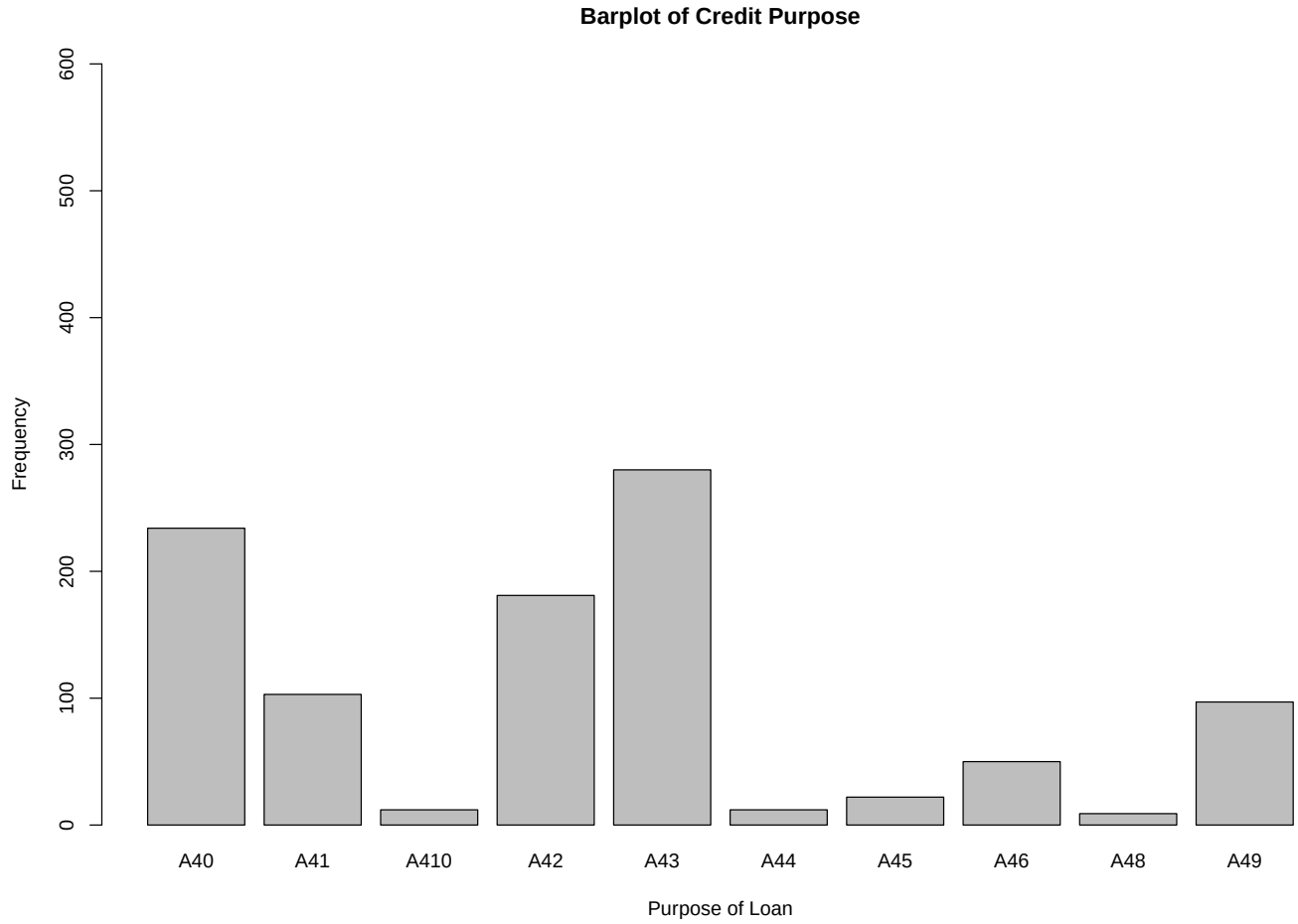
Data Set

- German Credit Data
- Mixture of Quantitative and Qualitative Attributes
- Observations: 1000 , Variables: 20
- Data Features:
  - No missing values
  - Outliers

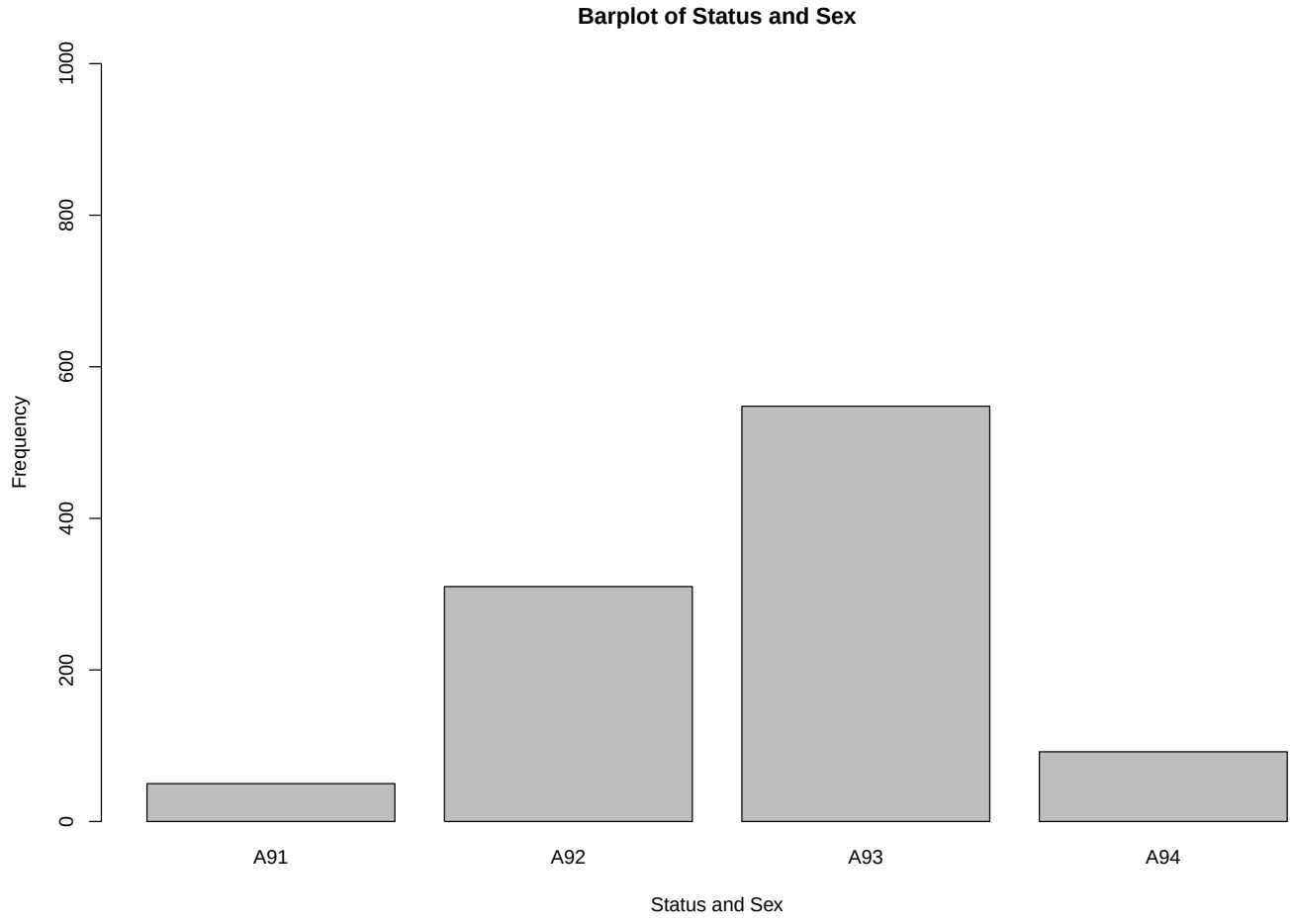
Overview

- Understand and describe data at an aggregate level
- Qualitative Variables
  - Bar Plots
  - Mosaic Plots
- Quantitave Variables
  - Histograms
  - Kernel Plots
  - Box Plots

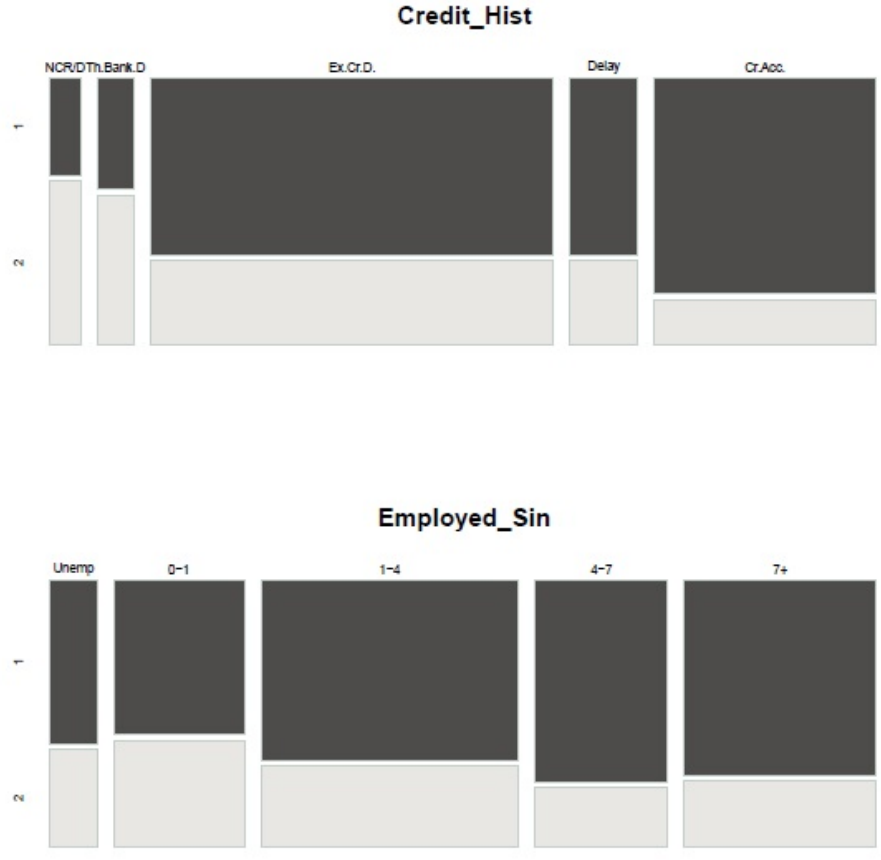
Bar Plots


Barplot of Credit Purpose
Frequency
0
100
200
300
400
500
600
A40
A41
A410
A42
A43
A44
A45
A46
A48
A49
Purpose of Loan

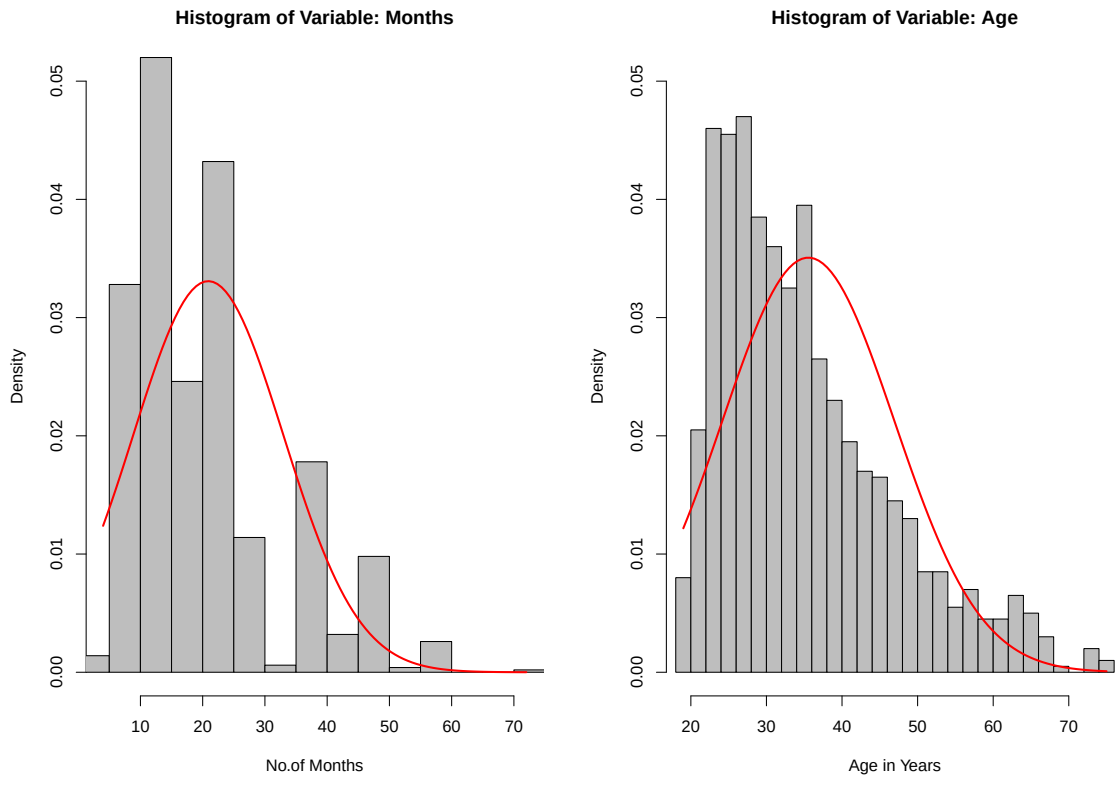
Bar Plots


Barplot of Status and Sex
Frequency
0
200
400
600
800
1000
A91
A92
A93
A94
Status and Sex

Mosaic Plots

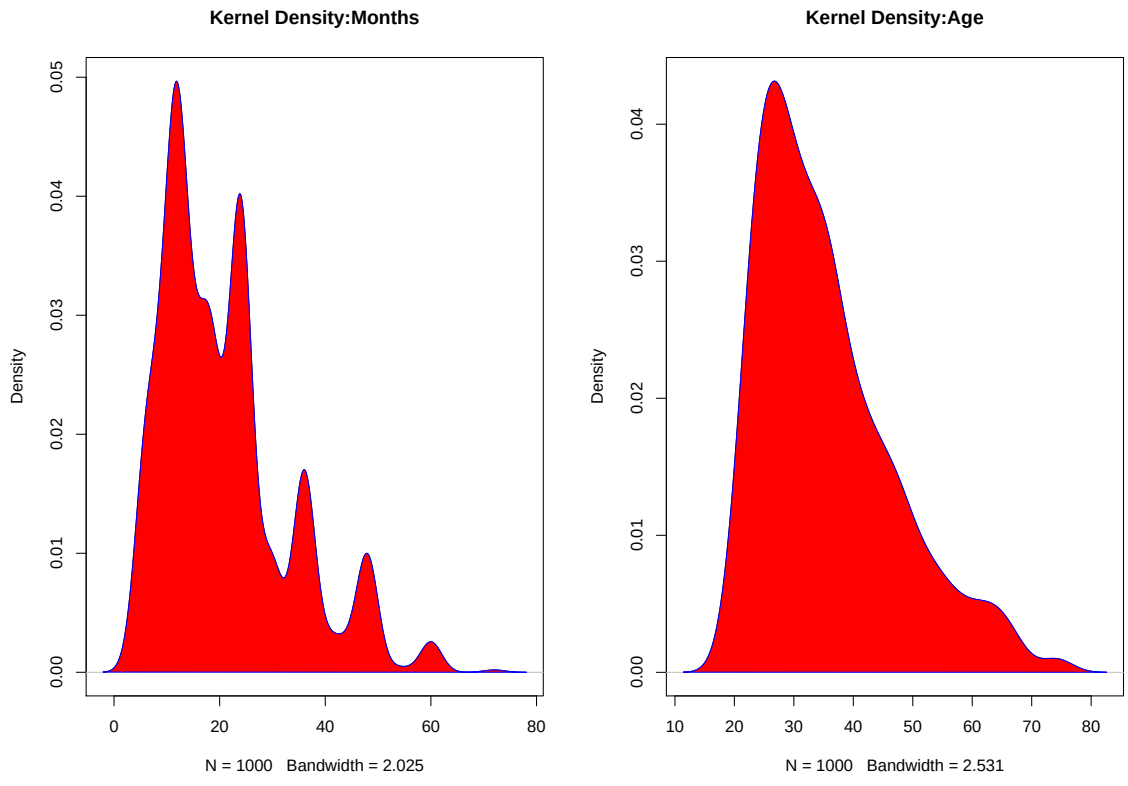


Histogram



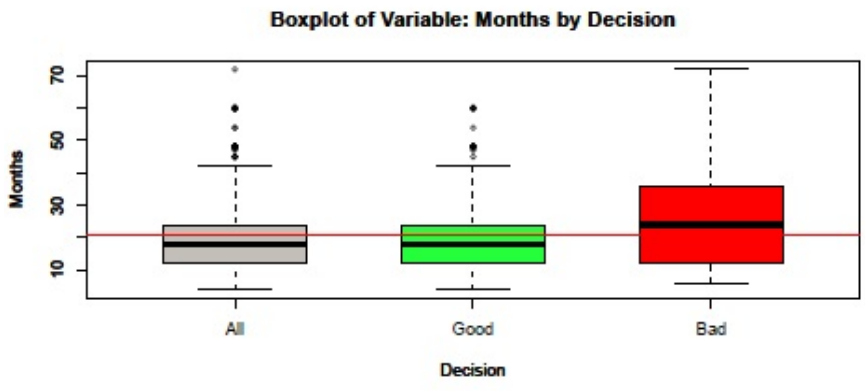
- Superimposed with Normal Distribution

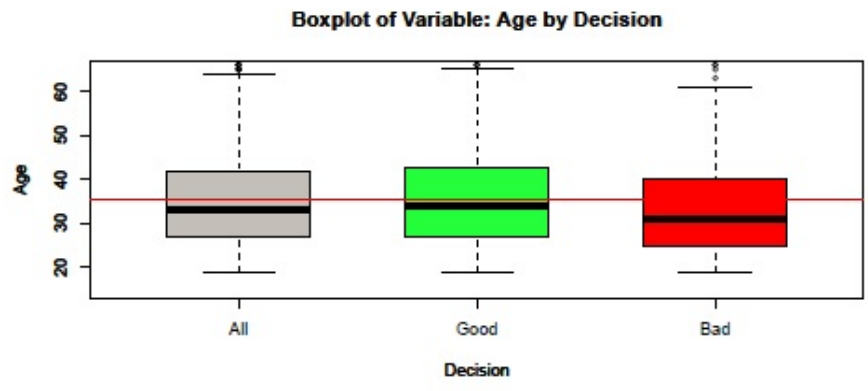
Kernel Density



- Accurate measure of probability density

Box Plots





- Outlier detection

Encoding Data

- Real world data-set usually contains two types of predictors
  - Categorical
  - Continuous
- We performed industry-wide used transformations on these predictors

Encoding Categorical Data

- Dummy encoding transformation was applied to transform them into numerical forms.

```
dummies = dummyVars( ~ ., data = dat_cat[,1:(
  ncol(dat_cat)-2)])
category_data_dummies=as.data.frame(predict(
  dummies,newdata=dat_cat[,1:(ncol(dat_cat)-2)])
  )
```

Encoding Continuous Data

- We performed outlier detection and imputation by replacing values with z value higher than 3 with predictor's mean value
- Outlier Function:

```
remove_outliers = function(x, na.rm = TRUE, ...)
VarMean = mean(x, na.rm = na.rm, ...)
VarSD = sd(x,na.rm = na.rm, ...)
y = x
y[abs((x - VarMean)/VarSD) >3] = NA
y
```

- We applied this function on dataset using sapply
- Imputed the NA replacements with mean value.

Scaling Data

- A (-1 to 1) scaling transformation was applied on the data to make them more comprehensible for algorithms
- Formula:

```
## MinMaxScaling (-1 to 1)
min_max_scaling=function(col)((col-min(col))/(
  max(col)-min(col))*2-1)
##check
min_max_scaling(c(1,2,3,4))
[1] -1.0000000 -0.3333333  0.3333333  1.0000000
```

- The formula was applied on dataset using sapply

Sampling Data

- The dataset was randomly divided into Train(60%), Validation(20%) and Test samples(20%) with control over target variable (same distribution of target variable in each sample).
- We used "createFolds" function of caret package which has similar functionality of dividing data.
- We removed variables with 0 variances using caret's "nearZeroVar" function and saved the samples sets.

Sampling Data

```
1  library(caret)
2  set.seed(3456)
3  ##dividing rows in 5 folds with control over target
     variable
4
5  devIndex = as.data.frame(createFolds(data_final$dv,
     k=5, list = TRUE))
6
7  data_div_1= data_final[devIndex$Fold1,c("id","dv")]
8  data_div_2= data_final[devIndex$Fold2,c("id","dv")]
9  data_div_3= data_final[devIndex$Fold3,c("id","dv")]
10 data_div_4= data_final[devIndex$Fold4,c("id","dv")]
11 data_div_5= data_final[devIndex$Fold5,c("id","dv")]
```

Methodology

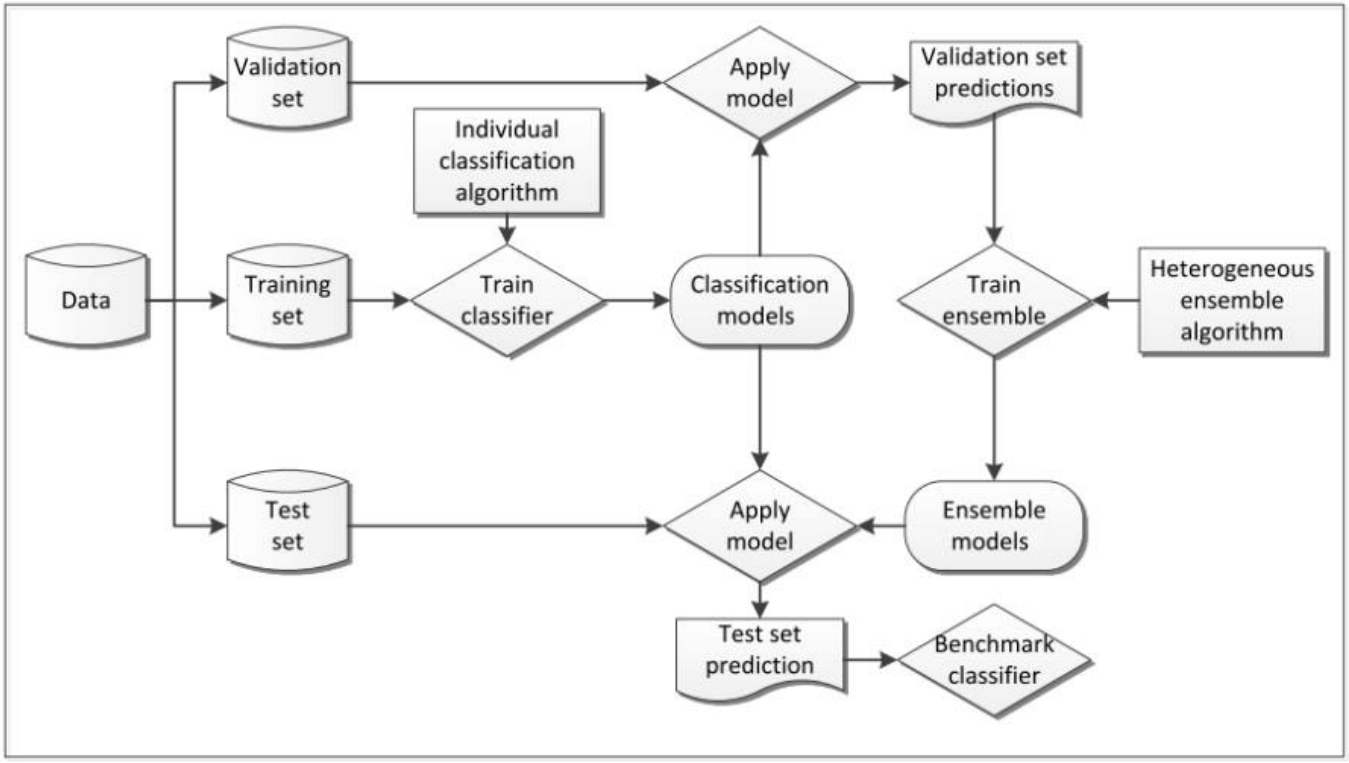
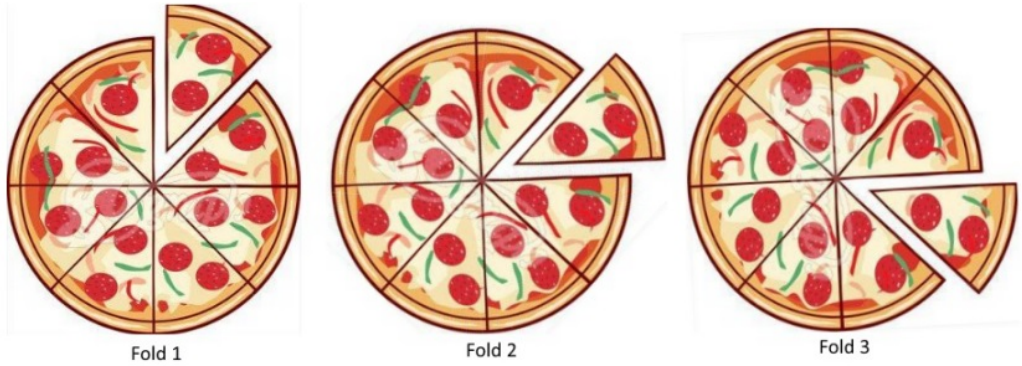


Figure 2: Schematic View of Predictive Modelling Process

Sub-sampling using cross validation

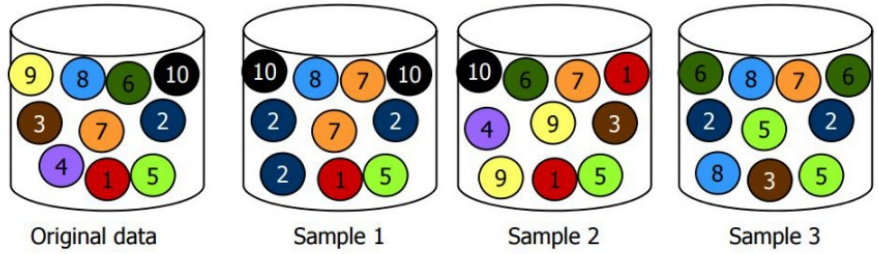
- To increase diversity for ensemble modelling, we applied cross validation method, except the validation part.
- Created 5 folds out of Training dataset with control over target variable.

k-fold cross-validation:

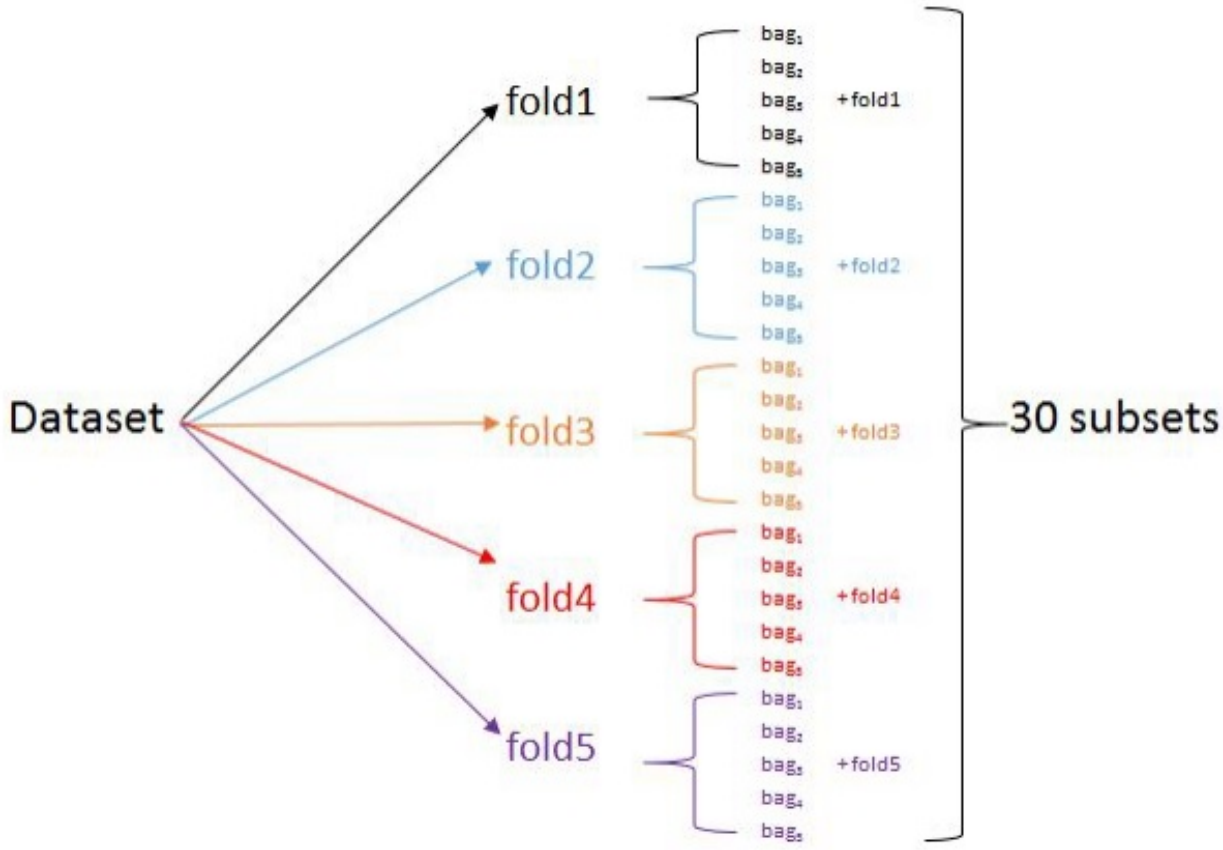


Further Sampling using Bootstrap Aggregation (Bagging)

- Bagging is random selection from original dataset with replacements.
- We controlled the random selection to 67-70 percent being the original data.



Sub Sampling Overview



Models

- Models Used:
  - Logistic Regression
  - Random Forest
  - Gradient Boosting
- Ensemble learning thrives on data diversity, it was in common interest to used maximal number of parametric combinations.
- Helped in saving time over finding optimal parameter values which otherwise would have been manual labor.

Models

- Grid Diagram

```
> lr_parameters=expand.grid(lambda = 2^seq(-19,6,2), cp = c("aic","bic"))
> lr_parameters
         lambda  cp
1  1.907349e-06 aic
2  7.629395e-06 aic
3  3.051758e-05 aic
4  1.220703e-04 aic
5  4.882812e-04 aic
6  1.953125e-03 aic
7  7.812500e-03 aic
8  3.125000e-02 aic
9  1.250000e-01 aic
10 5.000000e-01 aic
11 2.000000e+00 aic
12 8.000000e+00 aic
13 3.200000e+01 aic
14 1.907349e-06 bic
15 7.629395e-06 bic
16 3.051758e-05 bic
17 1.220703e-04 bic
18 4.882812e-04 bic
19 1.953125e-03 bic
20 7.812500e-03 bic
21 3.125000e-02 bic
22 1.250000e-01 bic
23 5.000000e-01 bic
24 2.000000e+00 bic
25 8.000000e+00 bic
26 3.200000e+01 bic
```

- Created models for each sample over all aforementioned parametric values.

Model

- To speed up the process we used "doSNOW" and "foreach" packages which allows parallel processing.
- Instead of 1 core, we could now engage 3 cores simultaneously for this purpose.
- For each dataset, a for loop was required to parse through parametric grid.

Models Overview

Classifier	No.of Models	No.of Candidates
Logistic Regression	26	780
ANN	72	2160
Random Forest	54	1620
Total	152	4560

Table 1: Overview of base models

Combining and Cleaning

- Combining
  - Since the predictions of algorithms were saved in different CSVs, we had to combine them for final ensemble learning.
  - We used "import" and "do.call" function of plyr package which called "read.csv" and "cbind" on all the CSVs in a single line code.
- Correlation
  - Removed candidates with more than 0.85 correlation value with other candidates

Ensemble Learning and Scoring Results

- Ensemble Creation
  - We used logistic regression and random forest as generalizer to predict the target variable using candidates.
- Results
  - We calculated AUC (Area under the score) value for judging model performances.

Algorithm	AUC value
Stacking with RF	0.812
Stacking with LogR	0.70
ANN	0.770
RF	0.791
LogR	0.761

Table 2: Summary of Performance scores

Thank You