Data622 - Group2 - Homework3

Zachary Palmore, Kevin Potter, Amit Kapoor

10/2/2021

Contents

Overview

In this project, the dataset used, is for Loan approval where the prediction will be done for Loan approval status using Linear Discriminant Analysis (LDA), K-nearest neighbor (KNN), Decision Trees and Random Forest models.

R packages

We will use `r` for data modeling. All packages used for data exploration, visualization, preparation and modeling are listed in Code Appendix.

Data Exploration

Below is the description of the variables of interest in the data set.

VARIABLE NAME	DESCRIPTION
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents
Education	Applicant Education (Graduate/ Undergraduate)
Self_Employed	Self employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	Loan approved (Y/N)

Data summary

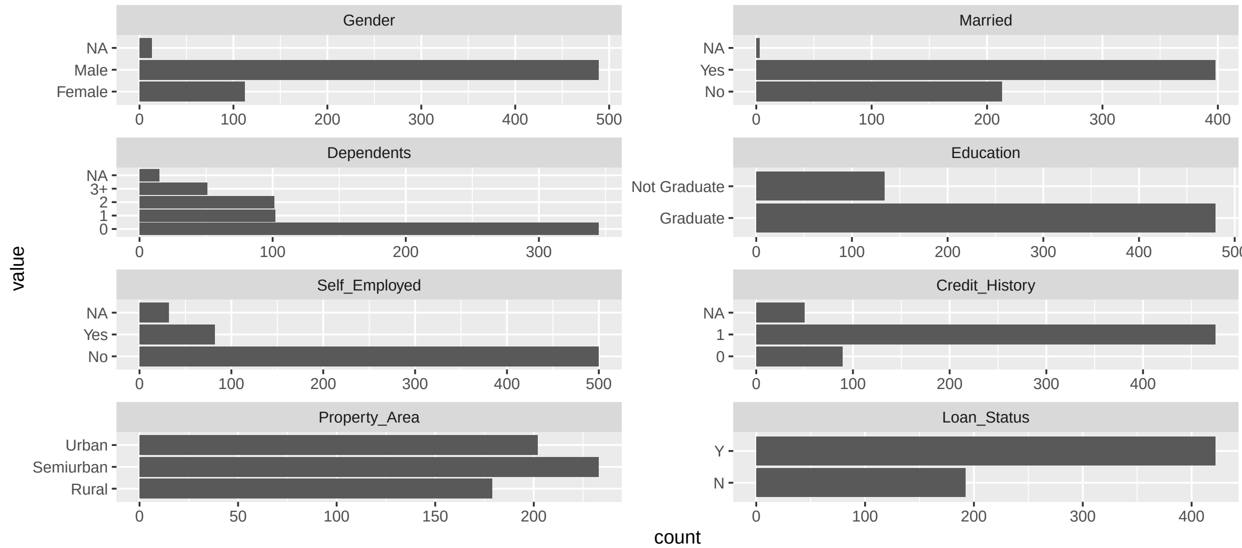
Below is summary of loan approval dataset.

```
## Data Frame Summary
## loan_data
## Dimensions: 614 x 12
## Duplicates: 0
##
## +----+-------------------+-----------------------------+--------------------+----------+---------+
## | No | Variable          | Stats / Values              | Freqs (% of Valid) | Valid    | Missing |
## +====+===================+=============================+====================+==========+=========+
## | 1  | Gender            | 1. Female                   | 112 (18.6%)        | 601      | 13      |
## |    | [factor]          | 2. Male                     | 489 (81.4%)        | (97.9%)  | (2.1%)  |
## +----+-------------------+-----------------------------+--------------------+----------+---------+
## | 2  | Married           | 1. No                       | 213 (34.9%)        | 611      | 3       |
## |    | [factor]          | 2. Yes                      | 398 (65.1%)        | (99.5%)  | (0.5%)  |
## +----+-------------------+-----------------------------+--------------------+----------+---------+
## | 3  | Dependents        | 1. 0                        | 345 (57.6%)        | 599      | 15      |
## |    | [factor]          | 2. 1                        | 102 (17.0%)        | (97.6%)  | (2.4%)  |
## |    |                   | 3. 2                        | 101 (16.9%)        |          |         |
## |    |                   | 4. 3+                       |  51 ( 8.5%)        |          |         |
## +----+-------------------+-----------------------------+--------------------+----------+---------+
## | 4  | Education         | 1. Graduate                 | 480 (78.2%)        | 614      | 0       |
## |    | [factor]          | 2. Not Graduate             | 134 (21.8%)        | (100.0%) | (0.0%)  |
## +----+-------------------+-----------------------------+--------------------+----------+---------+
```

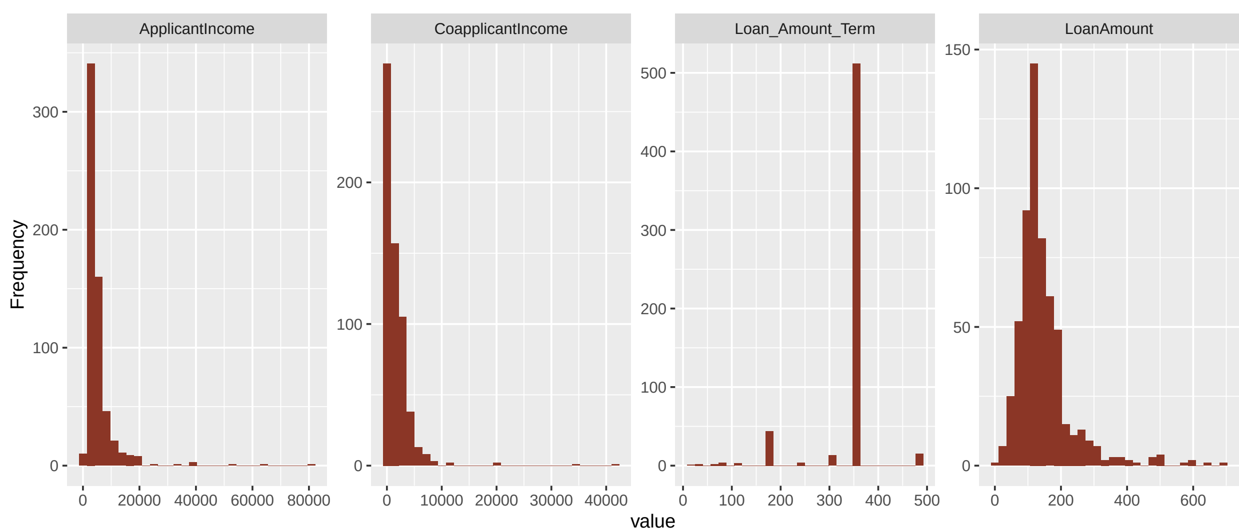
```
## | 5  | Self_Employed     | 1. No                        | 500 (85.9%)         | 582      | 32      |
## |    | [factor]          | 2. Yes                       |  82 (14.1%)         | (94.8%)  | (5.2%)  |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 6  | ApplicantIncome   | Mean (sd) : 5403.5 (6109)    | 505 distinct values | 614      | 0       |
## |    | [integer]         | min < med < max:             |                     | (100.0%) | (0.0%)  |
## |    |                   | 150 < 3812.5 < 81000         |                     |          |         |
## |    |                   | IQR (CV) : 2917.5 (1.1)      |                     |          |         |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 7  | CoapplicantIncome | Mean (sd) : 1621.2 (2926.2)  | 287 distinct values | 614      | 0       |
## |    | [numeric]         | min < med < max:             |                     | (100.0%) | (0.0%)  |
## |    |                   | 0 < 1188.5 < 41667           |                     |          |         |
## |    |                   | IQR (CV) : 2297.2 (1.8)      |                     |          |         |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 8  | LoanAmount        | Mean (sd) : 146.4 (85.6)     | 203 distinct values | 592      | 22      |
## |    | [integer]         | min < med < max:             |                     | (96.4%)  | (3.6%)  |
## |    |                   | 9 < 128 < 700                |                     |          |         |
## |    |                   | IQR (CV) : 68 (0.6)          |                     |          |         |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 9  | Loan_Amount_Term  | Mean (sd) : 342 (65.1)       | 12 :   1 ( 0.2%)    | 600      | 14      |
## |    | [integer]         | min < med < max:             | 36 :   2 ( 0.3%)    | (97.7%)  | (2.3%)  |
## |    |                   | 12 < 360 < 480               | 60 :   2 ( 0.3%)    |          |         |
## |    |                   | IQR (CV) : 0 (0.2)           | 84 :   4 ( 0.7%)    |          |         |
## |    |                   |                              | 120 :   3 ( 0.5%)   |          |         |
## |    |                   |                              | 180 :  44 ( 7.3%)   |          |         |
## |    |                   |                              | 240 :   4 ( 0.7%)   |          |         |
## |    |                   |                              | 300 :  13 ( 2.2%)   |          |         |
## |    |                   |                              | 360 : 512 (85.3%)   |          |         |
## |    |                   |                              | 480 :  15 ( 2.5%)   |          |         |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 10 | Credit_History    | 1. 0                         |  89 (15.8%)         | 564      | 50      |
## |    | [factor]          | 2. 1                         | 475 (84.2%)         | (91.9%)  | (8.1%)  |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 11 | Property_Area     | 1. Rural                     | 179 (29.2%)         | 614      | 0       |
## |    | [factor]          | 2. Semiurban                 | 233 (37.9%)         | (100.0%) | (0.0%)  |
## |    |                   | 3. Urban                     | 202 (32.9%)         |          |         |
## +----+-------------------+------------------------------+---------------------+----------+---------+
## | 12 | Loan_Status       | 1. N                         | 192 (31.3%)         | 614      | 0       |
## |    | [factor]          | 2. Y                         | 422 (68.7%)         | (100.0%) | (0.0%)  |
## +----+-------------------+------------------------------+---------------------+----------+---------+
```

- There are 7 columns having missing values.
- The proportion of values for few columns shows significant differences i.e. Gender (more males), Married(more married), Credit_History (more having credit history).

Below graphs shows the distribution of all categorical variables.

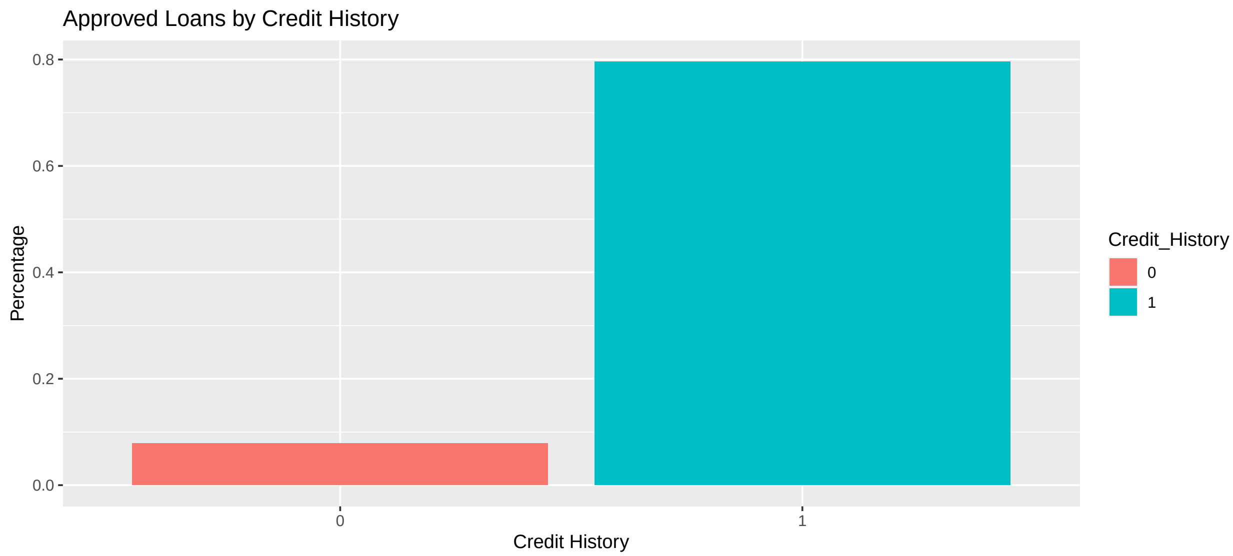


Below graph shows the distribution of numeric predictors.

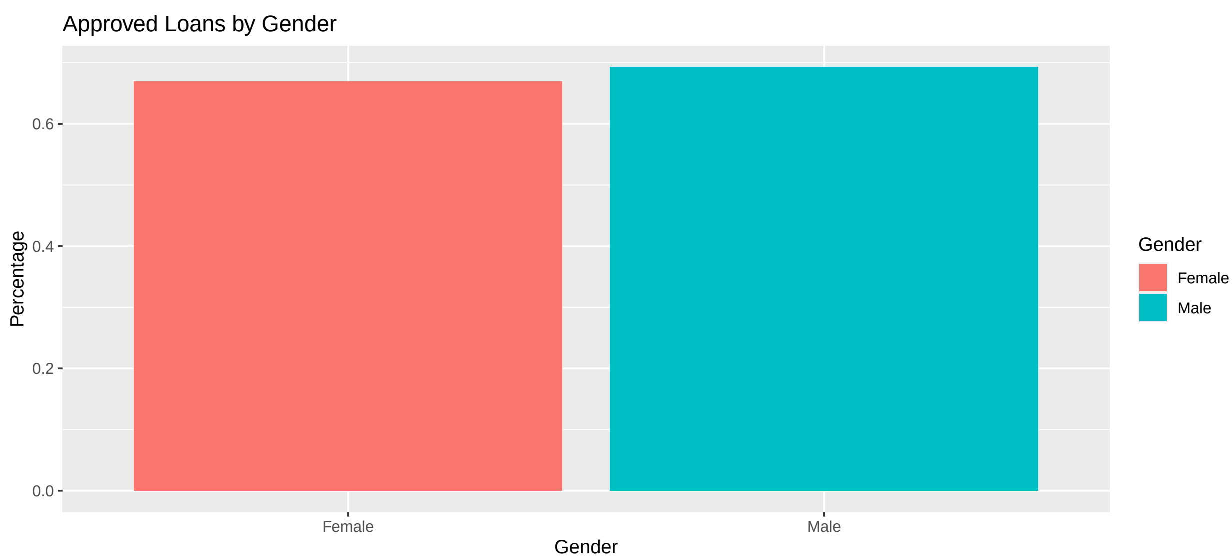


Next we will cover impact of categorical variables on loan approval.

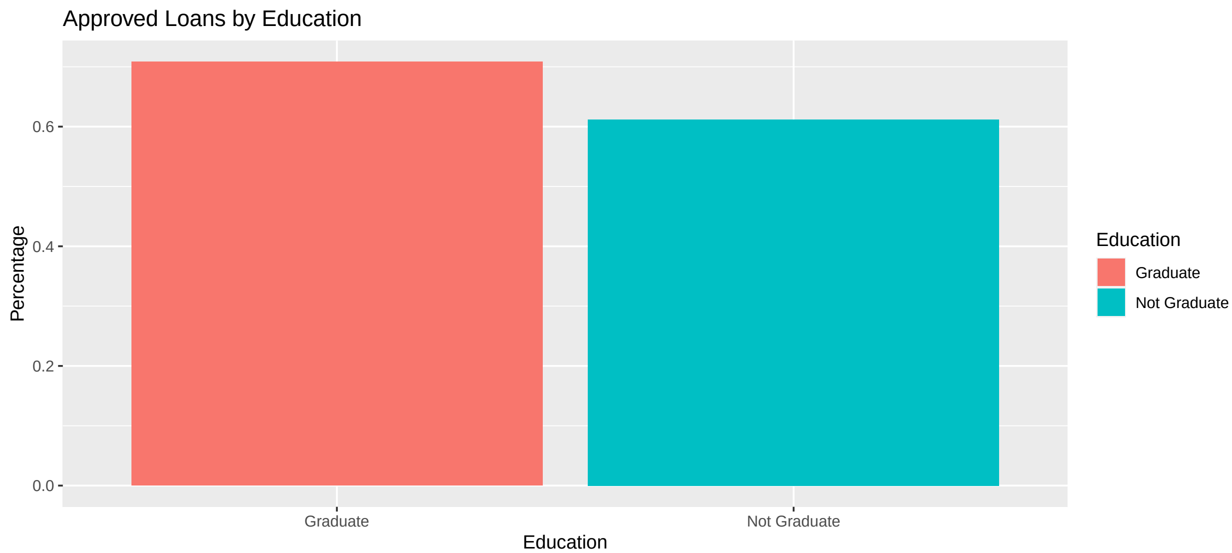
```
##   Credit_History Loan_Status       Freq
## 1              0           Y 0.07865169
## 2              1           Y 0.79578947
```



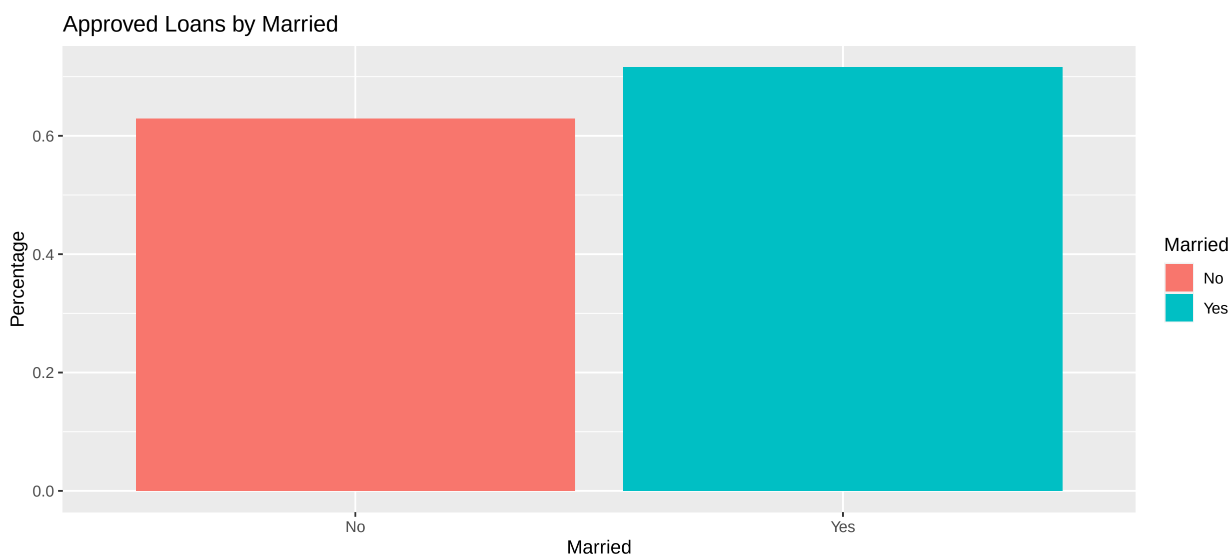

```
##   Gender Loan_Status      Freq
## 1 Female           Y 0.6696429
## 2   Male           Y 0.6932515
```



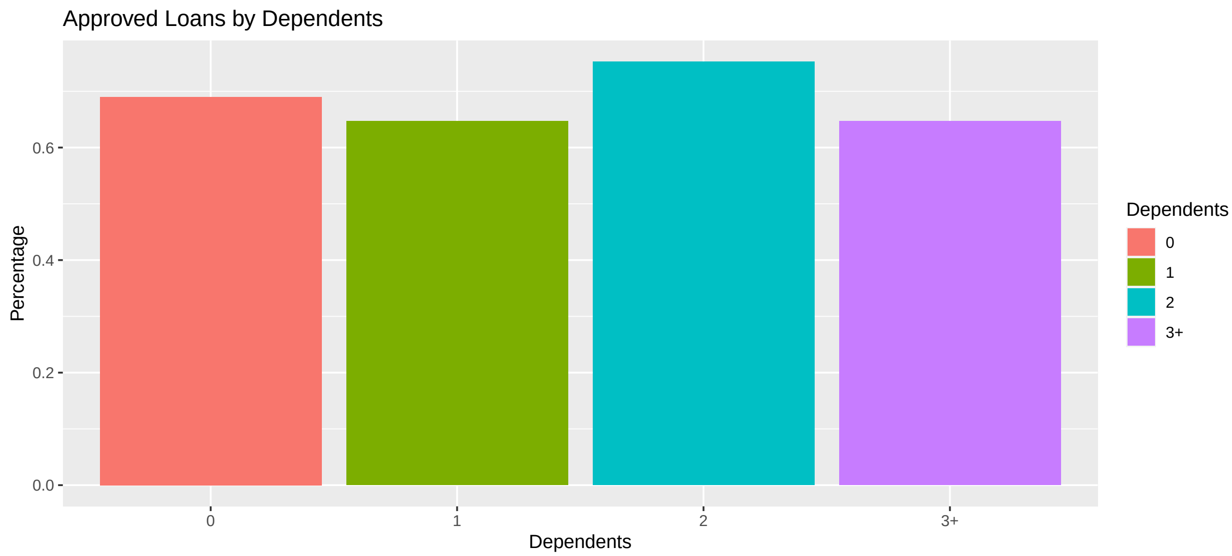

```
##      Education Loan_Status      Freq
## 1     Graduate           Y 0.7083333
## 2 Not Graduate           Y 0.6119403
```




```
##   Married Loan_Status      Freq
## 1      No           Y 0.6291080
## 2     Yes           Y 0.7160804
```

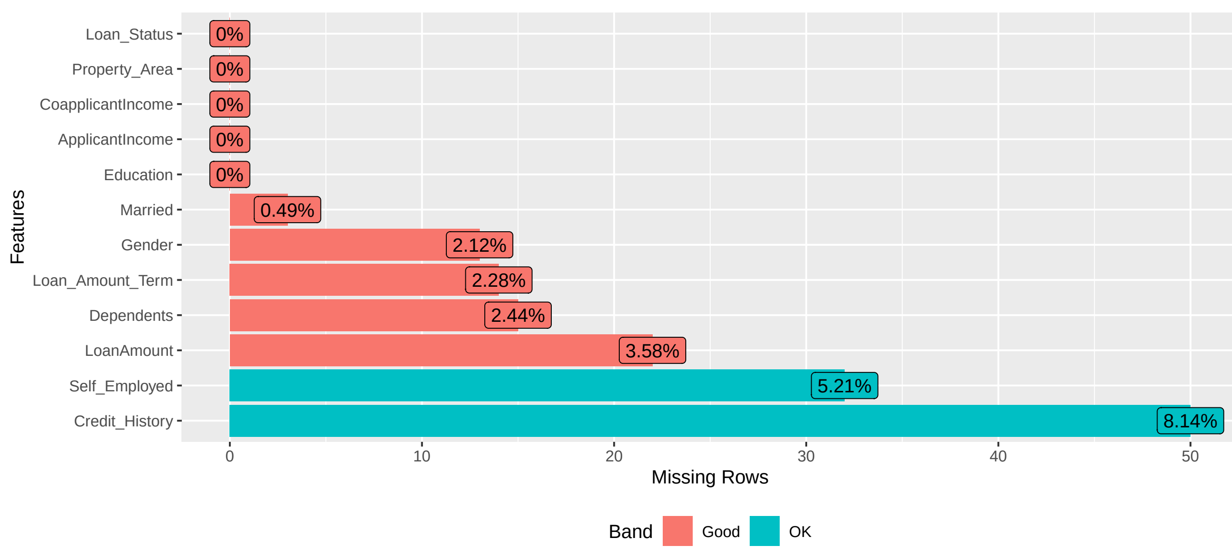



```
##   Dependents Loan_Status      Freq
## 1          0           Y 0.6898551
## 2          1           Y 0.6470588
## 3          2           Y 0.7524752
## 4         3+           Y 0.6470588
```



Data Preparation

Handling missing values



We can see above credit_history contributes to 8% of missing data alongwith self_employed that accounts for more than 5% of missing data. All records having missing categorical predictors will be removed. Next we will impute numeric values using MICE (Multivariate Imputation by Chained Equations).

```
## [1] 511  12
```

Finally our clean dataset contains 511 rows and 12 columns.

Preprocess using transformation

We have seen above that numeric features are right skewed so in this step we will use caret `preprocess` method using box cox, center and scale transformation.

Training and Test Partition

In this step for data preparation we will partition the training dataset in training and validation sets using `createDataPartition` method from `caret` package. We will reserve 75% for training and rest 25% for validation purpose.

Build Models

Linear Discriminant Analysis (LDA)

```
## Call:
## lda(Loan_Status ~ ., data = loan_data)
##
## Prior probabilities of groups:
##         N         Y
## 0.3209393 0.6790607
##
## Group means:
##   GenderMale MarriedYes Dependents1 Dependents2 Dependents3+
## N  0.7926829  0.5792683   0.1829268   0.1341463   0.09756098
## Y  0.8357349  0.6801153   0.1585014   0.1902017   0.08069164
##   EducationNot Graduate Self_EmployedYes ApplicantIncome CoapplicantIncome
## N             0.2682927        0.1463415     0.003576320         0.0571435
## Y             0.1902017        0.1325648    -0.001690249        -0.0270073
##    LoanAmount Loan_Amount_Term Credit_History1 Property_AreaSemiurban
## N  0.07966414      0.016992352       0.5548780              0.2682927
## Y -0.03765106     -0.008030968       0.9798271              0.4409222
##   Property_AreaUrban
## N          0.3719512
## Y          0.2997118
##
## Coefficients of linear discriminants:
##                                LD1
## GenderMale              0.185159211
## MarriedYes              0.375755462
## Dependents1            -0.209004726
## Dependents2             0.137509542
## Dependents3+            0.007142953
## EducationNot Graduate  -0.294391997
## Self_EmployedYes       -0.025905262
## ApplicantIncome        -0.012085555
## CoapplicantIncome      -0.106529320
## LoanAmount             -0.099136040
## Loan_Amount_Term       -0.049820158
## Credit_History1         3.073804026
## Property_AreaSemiurban  0.616732100
## Property_AreaUrban      0.066231320
```

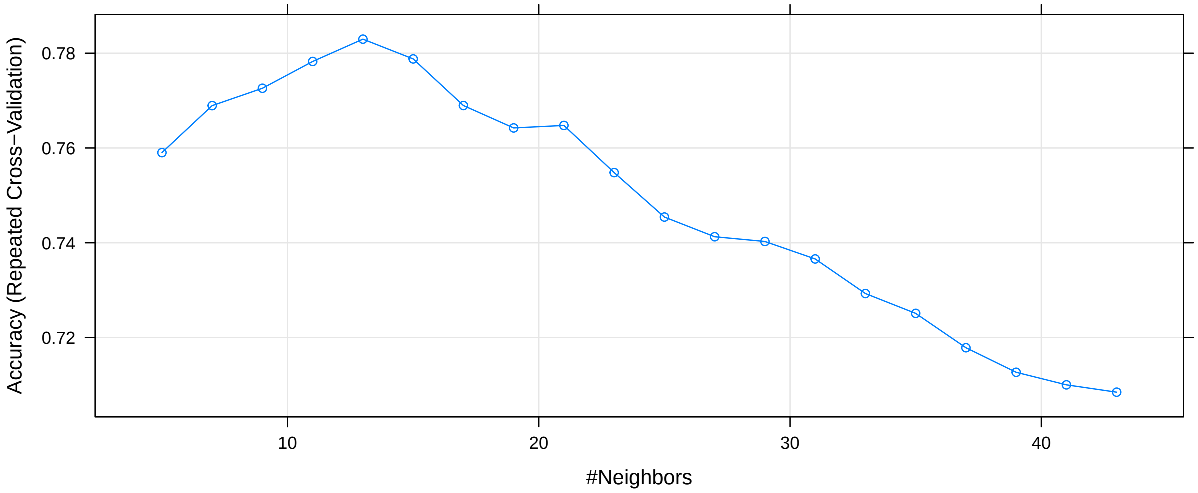
```
## [1] 0.8110236
```

LDA model accuracy comes out as ~81%

K-nearest neighbor (KNN)

```
## Created from 384 samples and 12 variables
##
## Pre-processing:
##   - centered (4)
##   - ignored (8)
##   - scaled (4)
```

```
## k-Nearest Neighbors
##
## 384 samples
##  11 predictor
##   2 classes: 'N', 'Y'
##
## Pre-processing: centered (14), scaled (14)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 346, 346, 346, 345, 346, 345, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7590209  0.3575145
##    7  0.7689406  0.3751117
##    9  0.7725985  0.3746036
##   11  0.7782524  0.3884713
##   13  0.7829615  0.3994494
##   15  0.7787908  0.3849571
##   17  0.7689528  0.3503238
##   19  0.7642294  0.3309567
##   21  0.7647551  0.3299468
##   23  0.7548097  0.2947001
##   25  0.7454298  0.2607665
##   27  0.7412733  0.2448000
##   29  0.7402746  0.2423939
##   31  0.7365911  0.2319391
##   33  0.7292901  0.2050164
##   35  0.7251066  0.1885755
##   37  0.7178596  0.1615955
##   39  0.7126768  0.1383143
##   41  0.7100317  0.1282468
##   43  0.7084798  0.1212857
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 13.
```




```
## [1] 0.7952756
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N  Y
##          N 17  2
##          Y 24 84
##
##                Accuracy : 0.7953
##                  95% CI : (0.7146, 0.8617)
##     No Information Rate : 0.6772
##     P-Value [Acc > NIR] : 0.002202
##
##                   Kappa : 0.4553
##
##  Mcnemar's Test P-Value : 3.814e-05
##
##             Sensitivity : 0.4146
##             Specificity : 0.9767
##          Pos Pred Value : 0.8947
##          Neg Pred Value : 0.7778
##              Prevalence : 0.3228
##          Detection Rate : 0.1339
##    Detection Prevalence : 0.1496
##       Balanced Accuracy : 0.6957
##
##        'Positive' Class : N
##
```

Decision Trees

Random Forests

Model performance

Conclusion

References

https://www.r-bloggers.com/2018/07/prop-table/

Code Appendix

```
knitr::opts_chunk$set(echo=FALSE, error=FALSE, warning=FALSE, message=FALSE, fig.align="center", fig.wi
# Libraries

library(summarytools)
library(tidyverse)
library(DataExplorer)
library(reshape2)
library(mice)
library(caret)
library(MASS)
library(e1071)
library(caret)

set.seed(622)
# read data, change blank to NA and and remove loan_id
loan_data <- read.csv('https://raw.githubusercontent.com/amit-kapoor/Data622Group2/main/Loan_approval.c
  na_if("") %>%
  dplyr::select(-1)

# categorical columns as factors
loan_data <- loan_data %>%
  mutate(Gender=as.factor(Gender),
         Married=as.factor(Married),
         Dependents=as.factor(Dependents),
         Education=as.factor(Education),
         Self_Employed=as.factor(Self_Employed),
         Property_Area=as.factor(Property_Area),
         Credit_History=as.factor(Credit_History),
         Loan_Status=as.factor(Loan_Status))

dfSummary(loan_data, style = 'grid', graph.col = FALSE)

# select categorical columns
cat_cols = c()
j <- 1
```

```
for (i in 1:ncol(loan_data)) {
  if (class((loan_data[,i])) == 'factor') {
      cat_cols[j]=names(loan_data[i])
      j <- j+1
  }
}

loan_fact <-  loan_data[cat_cols]
# long format
loan_factm <- melt(loan_fact, measure.vars = cat_cols, variable.name = 'metric', value.name = 'value')

# plot categorical columns
ggplot(loan_factm, aes(x = value)) +
  geom_bar() +
  scale_fill_brewer(palette = "Set1") +
  facet_wrap( ~ metric, nrow = 5L, scales = 'free') + coord_flip()
plot_histogram(loan_data, geom_histogram_args = list("fill" = "tomato4"))
loan_ch <- with(loan_data, table(Credit_History, Loan_Status)) %>%
  prop.table(margin = 1) %>% as.data.frame() %>% filter(Loan_Status == 'Y')

loan_ch
ggplot(loan_ch, aes(x=Credit_History, y=Freq, fill=Credit_History)) + geom_bar(stat='identity') + labs(t
loan_gen <- with(loan_data, table(Gender, Loan_Status)) %>%
  prop.table(margin = 1) %>% as.data.frame() %>% filter(Loan_Status == 'Y')

loan_gen
ggplot(loan_gen, aes(x=Gender, y=Freq, fill=Gender)) + geom_bar(stat='identity') + labs(title = 'Approve
loan_ed <- with(loan_data, table(Education, Loan_Status)) %>%
  prop.table(margin = 1) %>% as.data.frame() %>% filter(Loan_Status == 'Y')

loan_ed
ggplot(loan_ed, aes(x=Education, y=Freq, fill=Education)) + geom_bar(stat='identity') + labs(title = 'Ap
loan_mar <- with(loan_data, table(Married, Loan_Status)) %>%
  prop.table(margin = 1) %>% as.data.frame() %>% filter(Loan_Status == 'Y')

loan_mar
ggplot(loan_mar, aes(x=Married, y=Freq, fill=Married)) + geom_bar(stat='identity') + labs(title = 'Appro
loan_dep <- with(loan_data, table(Dependents, Loan_Status)) %>%
  prop.table(margin = 1) %>% as.data.frame() %>% filter(Loan_Status == 'Y')

loan_dep
ggplot(loan_dep, aes(x=Dependents, y=Freq, fill=Dependents)) + geom_bar(stat='identity') + labs(title =
# plot missing values
plot_missing(loan_data)
# Filter out the data which has missing categorical predictors
loan_data <- loan_data %>% filter(!is.na(Credit_History) &
                                  !is.na(Self_Employed) &
                                  !is.na(Dependents) &
                                  !is.na(Gender) &
                                  !is.na(Married))
# impute numeric predictors using mice
loan_data <- complete(mice(data=loan_data, method="pmm", print=FALSE))
dim(loan_data)
```

```r
# library(e1071) - where this was used
set.seed(622)
loan_data <- loan_data %>%
  dplyr::select(c("ApplicantIncome", "CoapplicantIncome", "LoanAmount", "Loan_Amount_Term")) %>%
  preProcess(method = c("BoxCox","center","scale")) %>%
  predict(loan_data)
set.seed(622)
partition <- createDataPartition(loan_data$Loan_Status, p=0.75, list = FALSE)

training <- loan_data[partition,]
testing <- loan_data[-partition,]

# training/validation partition for independent variables
#X.train <- ld.clean[partition, ] %>% dplyr::select(-Loan_Status)
#X.test <- ld.clean[-partition, ] %>% dplyr::select(-Loan_Status)

# training/validation partition for dependent variable Loan_Status
#y.train <- ld.clean$Loan_Status[partition]
#y.test <- ld.clean$Loan_Status[-partition]
# LDA model
lda_model <- lda(Loan_Status~., data = loan_data)
lda_model
# prediction from lda model
lda_predict <- lda_model %>%
  predict(testing)
# accuracy
mean(lda_predict$class==testing$Loan_Status)
# KNN model
set.seed(622)
train.knn <- training[, names(training) != "Direction"]
prep <- preProcess(x = train.knn, method = c("center", "scale"))
prep
cl <- trainControl(method="repeatedcv", repeats = 5)
knn_model <- train(Loan_Status ~ ., data = training,
                method = "knn",
                trControl = cl,
                preProcess = c("center","scale"),
                tuneLength = 20)
knn_model
# prediction from knn model
plot(knn_model)
knn_predict <- predict(knn_model,newdata = testing)
mean(knn_predict == testing$Loan_Status) # accuracy
confusionMatrix(knn_predict, testing$Loan_Status)
```