Submission HW1

ISYE 6501 | Fall 22

- Ashish Dhiman | ashish.dhiman@gatech.edu
- Abhinav Arun | aarun60@gatech.edu
- Anshit Verma | averma373@gatech.edu

***Analysis Notes are marked with Red header: #Analysis**

Question 2.1

Example 1: Credit Risk Evaluation

Classification models find use cases across a spectrum of Credit Risk functions. A typical example being classifying a particular transaction as risky or otherwise basis which it is either declined or authorized. Here risk implies that the individual might not be able to make the required payments in the future, and therefore transactions made are riskier.

Objective Function (Y): To classify individual transactions as risky (decline) or not.
Predictor Variables (X_i): Some of the key predictor variables given below:

- ***Past Delinquency***: Delinquency implies that the individual was unable to keep up on his monthly payments previously and missed on his obligated payments. This is a key marker for credit risk, and past delinquent behavior hints towards potential future risk.
- ***Credit Utilization***: Credit Utilization is defined as the ratio of current balance to overall credit limit accorded to the individual. Suppose a individual has a credit line of $10,000 and he already has utilized $ 9k of it. This individual is generally prone to more risk as compared to the individual who only has utilized say $ 2k of his $ 10k line.
- ***Current Debt to Income ratio***: This is a measure of Income to Debt Capacity of the individual. In other words, it stacks up the overall debt obligations of the individual across mortgage, auto loan, credit cards etc., against his total income. If a larger part of an individual's income is directed towards his debt payment, than he/she again might be more susceptible to miss payments in the future and hence is riskier.
- ***Amount of Transaction***: This is the dollar amount of the transaction. Intuitively a transaction of $20k is riskier than $5, since even if the individual misses his payment, the hit taken by the firm is restricted to only $5.

Question 2.2

Exploratory Data Analysis (EDA)

```r
#imports
library(cowplot)
library(ggplot2)
library(reshape2)
```

```r
org_cc_data <- read.table(file = './data 2.2/credit_card_data-headers.txt', sep = "\t", header = TRUE)
dim(org_cc_data)
```

Read Data and Summary

```
## [1] 654  11
```

```r
head(org_cc_data)
```

```
##   A1    A2    A3   A8 A9 A10 A11 A12 A14 A15 R1
## 1  1 30.83 0.000 1.25  1   0   1   1 202   0  1
## 2  0 58.67 4.460 3.04  1   0   6   1  43 560  1
## 3  0 24.50 0.500 1.50  1   1   0   1 280 824  1
## 4  1 27.83 1.540 3.75  1   0   5   0 100   3  1
## 5  1 20.17 5.625 1.71  1   1   0   1 120   0  1
## 6  1 32.08 4.000 2.50  1   1   0   0 360   0  1
```

```r
summary(org_cc_data)
```

```
##        A1               A2              A3               A8
##  Min.   :0.0000   Min.   :13.75   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:0.0000   1st Qu.:22.58   1st Qu.: 1.040   1st Qu.: 0.165
##  Median :1.0000   Median :28.46   Median : 2.855   Median : 1.000
##  Mean   :0.6896   Mean   :31.58   Mean   : 4.831   Mean   : 2.242
##  3rd Qu.:1.0000   3rd Qu.:38.25   3rd Qu.: 7.438   3rd Qu.: 2.615
##  Max.   :1.0000   Max.   :80.25   Max.   :28.000   Max.   :28.500
##        A9              A10              A11              A12
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median : 0.000   Median :1.0000
##  Mean   :0.5352   Mean   :0.5612   Mean   : 2.498   Mean   :0.5382
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 3.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :67.000   Max.   :1.0000
##       A14               A15               R1
##  Min.   :   0.00   Min.   :     0   Min.   :0.0000
##  1st Qu.:  70.75   1st Qu.:     0   1st Qu.:0.0000
##  Median : 160.00   Median :     5   Median :0.0000
##  Mean   : 180.08   Mean   :  1013   Mean   :0.4526
##  3rd Qu.: 271.00   3rd Qu.:   399   3rd Qu.:1.0000
##  Max.   :2000.00   Max.   :100000   Max.   :1.0000
```

#Analysis:

A1,A9,A10,A12 are binary basis min/max values, rest are continuous

For target variable mean is ~45% ==> variable is not grossly imbalanced

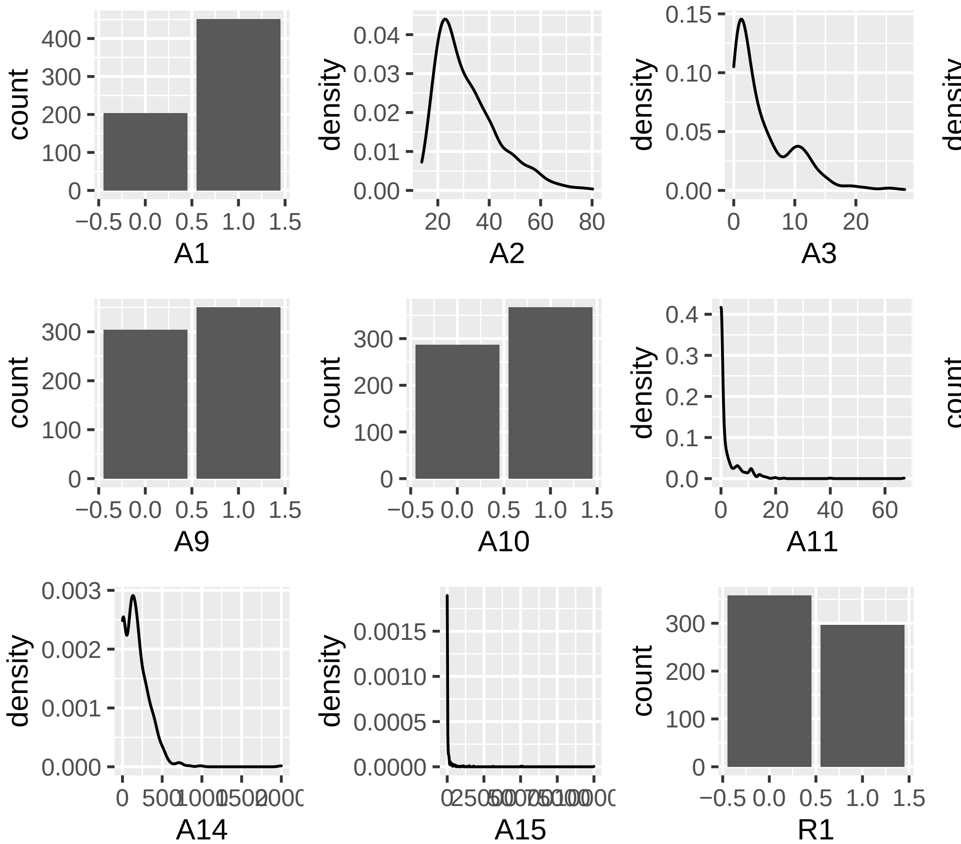
```
### Data Distribution
my_plots <- lapply(names(org_cc_data), function(var_x){
  p <-
    ggplot(org_cc_data) +
    aes_string(var_x)

  if(var_x %in% list("A1","A9","A10","A12","R1")) {
    p <- p + geom_bar()

  } else {
    p <- p + geom_density()
  }

})

plot_grid(plotlist = my_plots)
```

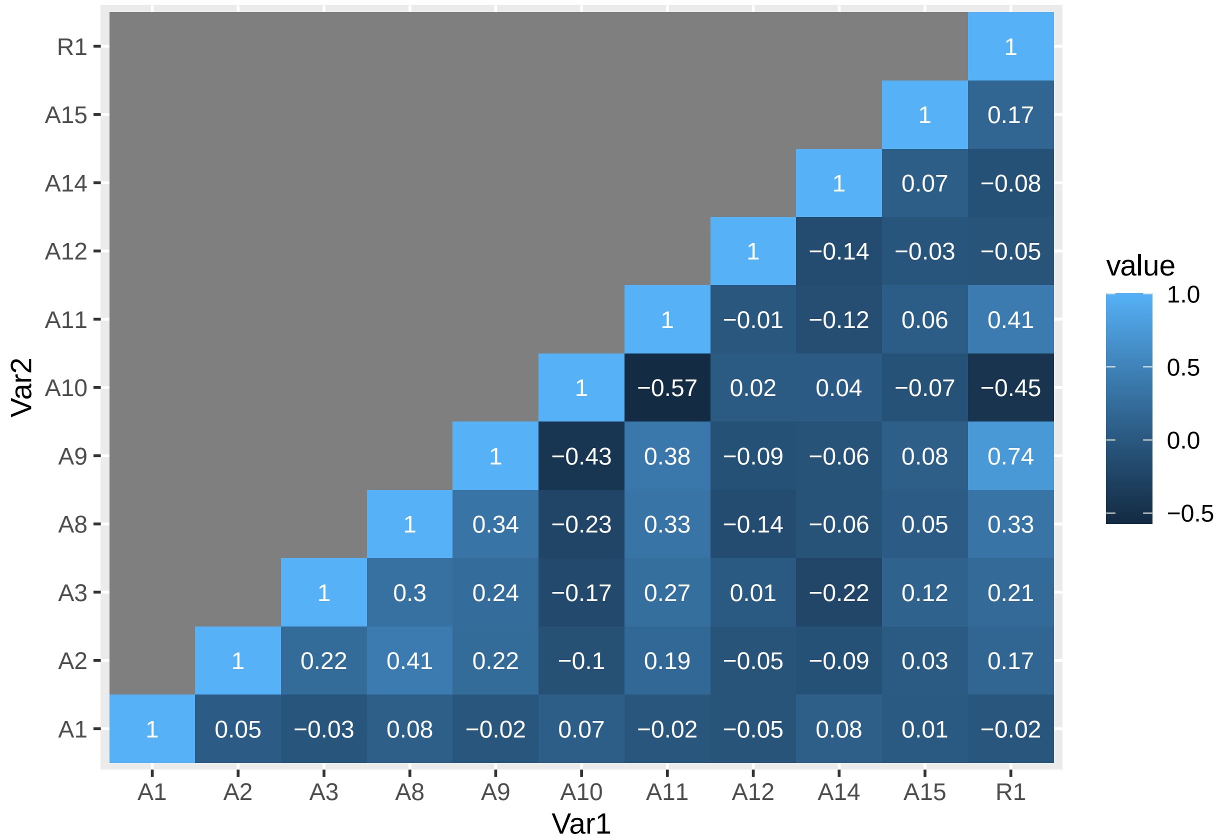


Data Distribution and Co relation

```
### Corealtion
cormat <- round(cor(org_cc_data),2)
cormat[upper.tri(cormat)] <- NA
melted_cormat <- melt(cormat)
# plotting the correlation heatmap
library(ggplot2)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2,
```

```r
                                     fill=value)) +
geom_tile() +
geom_text(aes(Var1, Var2, label = value),
          color = "white", size = 3)
```

```
## Warning: Removed 55 rows containing missing values (geom_text).
```



#Analysis:

If |cor| > 0.3 ==> Significant, then following pairs show high correlation:

- R1: A8,A9,A10,A11 (We expect these to show up with high weights in SVM eqn.)
- A11: A8,A9,A10
- A10: A9
- A9: A8
- A8: A2

Question 2.2 (part 1: kSVM with Linear kernel)

```r
#imports
library(kernlab)
```

```
##
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(ggplot2)
```

```
dim(org_cc_data)
```

```
## [1] 654  11
```

```
names(org_cc_data)
```

```
##  [1] "A1"  "A2"  "A3"  "A8"  "A9"  "A10" "A11" "A12" "A14" "A15" "R1"
```

v0: Vanilla Model and Accuracy function

```
model_v0 = ksvm(x=as.matrix(org_cc_data[,1:10]), y=org_cc_data[,11], scaled =TRUE, type = "C-svc",kerne
```

```
##  Setting default kernel parameters
```

```
### Accuracy function: Overall and in each class
acc_func <- function(model) {
   pred_all <- predict(model,org_cc_data[,1:10])
   print (paste("Overall Acc:", round(sum(pred_all == org_cc_data[,11]) * 100 / nrow(org_cc_data),4)))

   pred_1 <- predict(model,org_cc_data[org_cc_data$R1 == 1,1:10])
   print (paste("Acc in 1's:", round(sum(pred_1 == org_cc_data[org_cc_data$R1 == 1,11]) * 100 / nrow(or

   pred_0 <- predict(model,org_cc_data[org_cc_data$R1 == 0,1:10])
   print (paste("Acc in 0's:", round(sum(pred_0 == org_cc_data[org_cc_data$R1 == 0,11]) * 100 / nrow(or
}

acc_func(model_v0)
```

```
## [1] "Overall Acc: 86.3914"
## [1] "Acc in 1's: 94.2568"
## [1] "Acc in 0's: 79.8883"
```

```
print(paste("#Support Vectors",model_v0@nSV))
```

```
## [1] "#Support Vectors 190"
```

#Analysis: **There are 190 support vectors, or roughly 30% of the total data points.

v1: Optimise C

```
C_values <- c(0.0001,0.001,0.0015,0.002,0.005,0.01,0.03) #Range identified with hit and trial

for (C_i in C_values) {
  print (paste("For C = ",C_i))
  acc_func(ksvm(x=as.matrix(org_cc_data[,1:10]), y=org_cc_data[,11], scaled =TRUE, type = "C-svc",kernel
  print("")
}
```

```
## [1] "For C =  1e-04"
##  Setting default kernel parameters
## [1] "Overall Acc: 54.7401"
## [1] "Acc in 1's: 0"
## [1] "Acc in 0's: 100"
## [1] ""
## [1] "For C =  0.001"
##  Setting default kernel parameters
## [1] "Overall Acc: 83.792"
## [1] "Acc in 1's: 73.6486"
## [1] "Acc in 0's: 92.1788"
## [1] ""
## [1] "For C =  0.0015"
##  Setting default kernel parameters
## [1] "Overall Acc: 86.3914"
## [1] "Acc in 1's: 86.4865"
## [1] "Acc in 0's: 86.3128"
## [1] ""
## [1] "For C =  0.002"
##  Setting default kernel parameters
## [1] "Overall Acc: 86.3914"
## [1] "Acc in 1's: 94.2568"
## [1] "Acc in 0's: 79.8883"
## [1] ""
## [1] "For C =  0.005"
##  Setting default kernel parameters
## [1] "Overall Acc: 86.3914"
## [1] "Acc in 1's: 94.2568"
## [1] "Acc in 0's: 79.8883"
## [1] ""
## [1] "For C =  0.01"
##  Setting default kernel parameters
## [1] "Overall Acc: 86.3914"
## [1] "Acc in 1's: 94.2568"
## [1] "Acc in 0's: 79.8883"
## [1] ""
## [1] "For C =  0.03"
##  Setting default kernel parameters
## [1] "Overall Acc: 86.3914"
## [1] "Acc in 1's: 94.2568"
## [1] "Acc in 0's: 79.8883"
## [1] ""
```

#Analysis: With increasing C, we weight the cost function more towards misclassification (realtive to margin). Therefore intuitively inceasing C, should increase Accuracy albeit at

the cost of margin. And in the above data points too, similar effect is apparent

C=0.015 seems the best option, because even with almost equal accuracy, the accuracy in the negative class is higher, where for other C values the accuracy amongst classes is more lopsided

v1: Equation of SVM

$W = \sum_i(\alpha Y_i X_{support\_vector_i})$\$