Credit Card Approval Prediction

Introduction:

For a long time, the financial industry has used credit score cards to determine loan approval. Personal information and data submitted by credit card applicants are used by banks to assess the risk of applicants defaulting. This parameter, in turn, influences the future loan approval. The objective of this project is to determine if an applicant is a good client or a bad client based on multiple factors and enable a bank to decide whether or not to issue a credit card to an applicant.

The data set is taken from Kaggle: https://www.kaggle.com/rikdifos/credit-card-approval-prediction.

The data set consists of two files application_record that contains personal information of applicants and credit_record that consists of credit behavior of applicants. These two files are connected by ID.

Loading dataset

```
application_data <- read_csv("application_record.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   ID = col_double(),
##   CODE_GENDER = col_character(),
##   FLAG_OWN_CAR = col_character(),
##   FLAG_OWN_REALTY = col_character(),
##   CNT_CHILDREN = col_double(),
##   AMT_INCOME_TOTAL = col_double(),
##   NAME_INCOME_TYPE = col_character(),
##   NAME_EDUCATION_TYPE = col_character(),
##   NAME_FAMILY_STATUS = col_character(),
##   NAME_HOUSING_TYPE = col_character(),
##   DAYS_BIRTH = col_double(),
##   DAYS_EMPLOYED = col_double(),
##   FLAG_MOBIL = col_double(),
##   FLAG_WORK_PHONE = col_double(),
##   FLAG_PHONE = col_double(),
##   FLAG_EMAIL = col_double(),
##   OCCUPATION_TYPE = col_character(),
##   CNT_FAM_MEMBERS = col_double()
## )
```

```
credit_data <- read_csv("credit_record.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   ID = col_double(),
```

```
##   MONTHS_BALANCE = col_double(),
##   STATUS = col_character()
## )
```

```
head(application_data)
```

```
## # A tibble: 6 x 18
##       ID CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY CNT_CHILDREN AMT_INCOME_TOTAL
##    <dbl> <chr>       <chr>        <chr>                  <dbl>            <dbl>
## 1 5.01e6 M           Y            Y                          0           427500
## 2 5.01e6 M           Y            Y                          0           427500
## 3 5.01e6 M           Y            Y                          0           112500
## 4 5.01e6 F           N            Y                          0           270000
## 5 5.01e6 F           N            Y                          0           270000
## 6 5.01e6 F           N            Y                          0           270000
## # ... with 12 more variables: NAME_INCOME_TYPE <chr>,
## #   NAME_EDUCATION_TYPE <chr>, NAME_FAMILY_STATUS <chr>,
## #   NAME_HOUSING_TYPE <chr>, DAYS_BIRTH <dbl>, DAYS_EMPLOYED <dbl>,
## #   FLAG_MOBIL <dbl>, FLAG_WORK_PHONE <dbl>, FLAG_PHONE <dbl>,
## #   FLAG_EMAIL <dbl>, OCCUPATION_TYPE <chr>, CNT_FAM_MEMBERS <dbl>
```

The application data contains the personal information of applicants. This data contains:

- Binary variables like gender, own_car, own_realty, mobile, workphone, phone, and email. These variables takes values F(female) and M(male) for gender, Y(yes) and N(no) for own_car and own_realty, 1 and 0 for the remaining.
- Categorical variables like income type, occupation type,, house type, education, and marriage condition
- Continuous variables like number of children, annual income, age, experience, and family size.

```
head(credit_data)
```

```
## # A tibble: 6 x 3
##        ID MONTHS_BALANCE STATUS
##     <dbl>          <dbl> <chr>
## 1 5001711              0 X
## 2 5001711             -1 0
## 3 5001711             -2 0
## 4 5001711             -3 0
## 5 5001712              0 C
## 6 5001712             -1 C
```

The credit data contains monthly balance and status for each applicant. Here monthly balance 0 means current month, -1 is for previous month and so on. There are different status for each applicant. Here,C means balance is paid off that month, 0 means balance due for 0-29 days and so on,5 means overdue or balance due for $>$ 150 days, and X means no credit history

Intersection of application and credit data to get the list of IDs present in both the files.

```
application_credit_id <- intersect(application_data$ID, credit_data$ID)
```

Filtering the original data set to get the applicants who have personal information and credit history (applicants present in both the files).

```
data_application <- filter(application_data, ID %in% application_credit_id)
data_credit <- filter(credit_data, ID %in% application_credit_id)
```

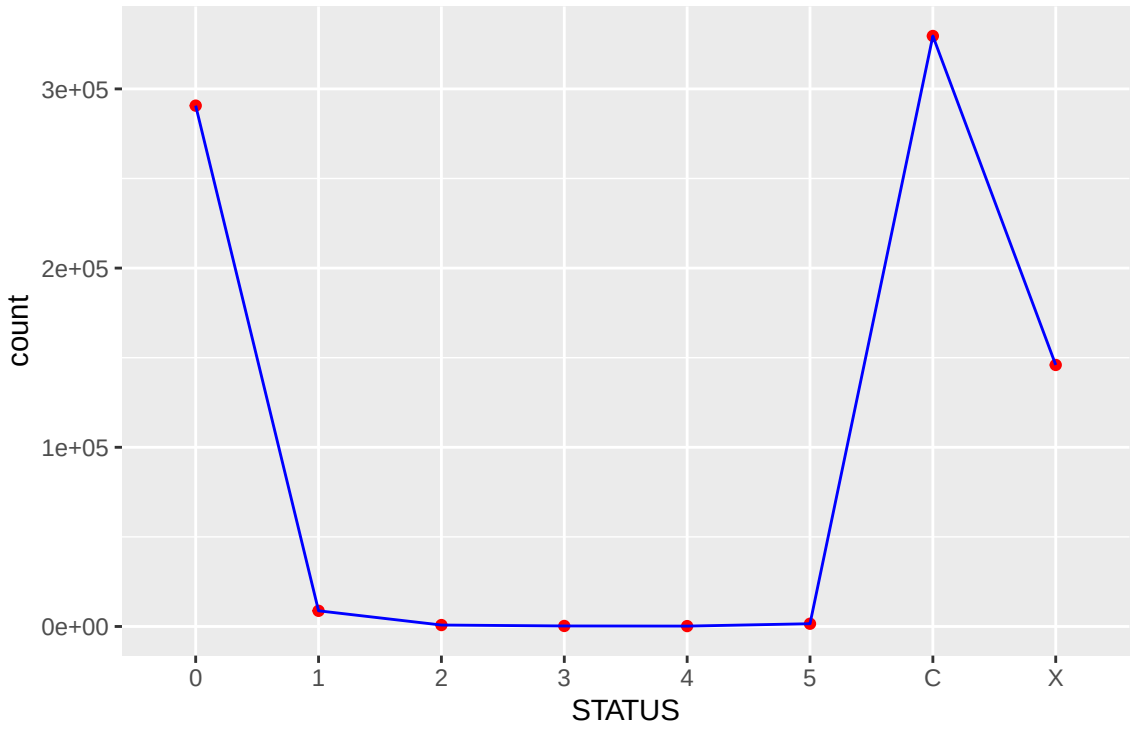
Data Transformation

There are applicants with credit status 0, 1, 2, 3, 4, 5, C, and X in the data set. Checking the number of applicants for each status.

```
check_status <- data_credit %>%
  select(STATUS) %>%
  group_by(STATUS)%>%mutate(count=n())
check_status <- unique(check_status)
check_status
```

```
## # A tibble: 8 x 2
## # Groups:   STATUS [8]
##   STATUS  count
##   <chr>   <int>
## 1 C      329536
## 2 1        8747
## 3 0      290654
## 4 X      145950
## 5 5        1527
## 6 4         214
## 7 3         286
## 8 2         801
```

```
ggplot(check_status,aes(x = STATUS)) +
  geom_point(aes(y = count), color = "red") +
  geom_line(aes(group = 1,y = count), color = "blue")
```



There are huge number of applicants with status 0 and C and very few applicants with the status 1 to 5. There are also good number of applicants who do not have a credit history

Converted the status to a numeric values to find the maximum credit status for each applicant.

```
credit_change <- data_credit

credit_change$STATUS[credit_change$STATUS == "5"] <- 7
credit_change$STATUS[credit_change$STATUS == "4"] <- 6
credit_change$STATUS[credit_change$STATUS == "3"] <- 5
credit_change$STATUS[credit_change$STATUS == "2"] <- 4
credit_change$STATUS[credit_change$STATUS == "1"] <- 3
credit_change$STATUS[credit_change$STATUS == "0"] <- 2
credit_change$STATUS[credit_change$STATUS == "C"] <- 1
credit_change$STATUS[credit_change$STATUS == "X"] <- 0
credit_change$STATUS <- as.numeric(credit_change$STATUS)

head(credit_change)
```

```
## # A tibble: 6 x 3
##        ID MONTHS_BALANCE STATUS
##     <dbl>          <dbl>  <dbl>
## 1 5008804              0      1
## 2 5008804             -1      1
## 3 5008804             -2      1
## 4 5008804             -3      1
## 5 5008804             -4      1
## 6 5008804             -5      1
```

Creating a new table for each applicant with the worst credit status by finding the maximum status they had in all months.

```r
worst_credit <- aggregate(credit_change$STATUS, by = list(credit_change$ID), max)
names(worst_credit)[1] <- "ID"
names(worst_credit)[2] <- "STATUS"

head(worst_credit)
```

```
##        ID STATUS
## 1 5008804      3
## 2 5008805      3
## 3 5008806      2
## 4 5008808      2
## 5 5008809      0
## 6 5008810      2
```

Here we have only one status for each applicant.

```r
credit <-  worst_credit[ !(worst_credit$STATUS %in% "0"), ]
head(credit)
```

```
##        ID STATUS
## 1 5008804      3
## 2 5008805      3
## 3 5008806      2
## 4 5008808      2
## 6 5008810      2
## 7 5008811      2
```

Here, applicants who does not have credit history have been removed.

Creating a new table with status for each applicant and determining if the applicant is a good client or a bad client.

```r
bad_balance <- credit
bad_balance <- bad_balance %>%
  mutate(Bad_client = ifelse(STATUS>=3, 1,0))
head(bad_balance)
```

```
##        ID STATUS Bad_client
## 1 5008804      3          1
## 2 5008805      3          1
## 3 5008806      2          0
## 4 5008808      2          0
## 5 5008810      2          0
## 6 5008811      2          0
```

Here, the applicants with status > 3 (original status - 1 to 5) are bad clients(1) , and applicants with status 1,2 (original status - C and 0) are good clients(0).

```r
bad_balance %>%
  count(Bad_client)
```

```
##   Bad_client     n
## 1          0 28819
## 2          1  4291
```

There are total 28819 number of good clients and 4291 number of bad clients in our data.

```
customer <- application_data %>%
  inner_join(bad_balance, by = "ID")
head(customer)
```

```
## # A tibble: 6 x 20
##       ID CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY CNT_CHILDREN AMT_INCOME_TOTAL
##    <dbl> <chr>       <chr>        <chr>                  <dbl>            <dbl>
## 1 5.01e6 M           Y            Y                          0           427500
## 2 5.01e6 M           Y            Y                          0           427500
## 3 5.01e6 M           Y            Y                          0           112500
## 4 5.01e6 F           N            Y                          0           270000
## 5 5.01e6 F           N            Y                          0           270000
## 6 5.01e6 F           N            Y                          0           270000
## # ... with 14 more variables: NAME_INCOME_TYPE <chr>,
## #   NAME_EDUCATION_TYPE <chr>, NAME_FAMILY_STATUS <chr>,
## #   NAME_HOUSING_TYPE <chr>, DAYS_BIRTH <dbl>, DAYS_EMPLOYED <dbl>,
## #   FLAG_MOBIL <dbl>, FLAG_WORK_PHONE <dbl>, FLAG_PHONE <dbl>,
## #   FLAG_EMAIL <dbl>, OCCUPATION_TYPE <chr>, CNT_FAM_MEMBERS <dbl>,
## #   STATUS <dbl>, Bad_client <dbl>
```

Here, we have combined both the files by ID using inner join.

```
customer <- unique(setDT(customer), by = c(2:20))
customer %>%
  count(Bad_client)
```

```
##    Bad_client    n
## 1:          0 9022
## 2:          1 2530
```

We got the unique values of the data set without the ID field here. Finally our dataset has 9022 good clients and 2530 bad clients.

Renaming column names for better understanding.

```
customer <- customer %>%
  rename(
    Gender = CODE_GENDER,
    Own_Car = FLAG_OWN_CAR,
    Own_Realty = FLAG_OWN_REALTY,
    Children_Count = CNT_CHILDREN,
    Annual_Income = AMT_INCOME_TOTAL,
    Income_Type = NAME_INCOME_TYPE,
    Education_Type = NAME_EDUCATION_TYPE,
    Marital_Status = NAME_FAMILY_STATUS,
    Housing_Type = NAME_HOUSING_TYPE,
```

```
    Occupation_Type = OCCUPATION_TYPE,
    Family_members_count = CNT_FAM_MEMBERS,
    Client = Bad_client
    )
```

Checking missing values in the data set

```
sapply(customer,function(x)any(is.na(x)))
```

```
##                   ID               Gender              Own_Car
##                FALSE                FALSE                FALSE
##           Own_Realty       Children_Count        Annual_Income
##                FALSE                FALSE                FALSE
##          Income_Type       Education_Type       Marital_Status
##                FALSE                FALSE                FALSE
##         Housing_Type           DAYS_BIRTH        DAYS_EMPLOYED
##                FALSE                FALSE                FALSE
##           FLAG_MOBIL      FLAG_WORK_PHONE           FLAG_PHONE
##                FALSE                FALSE                FALSE
##           FLAG_EMAIL      Occupation_Type Family_members_count
##                FALSE                 TRUE                FALSE
##               STATUS               Client
##                FALSE                FALSE
```

Only the variable Occupation_Type has missing values.

Replacing missing values in occupation type as unknown

```
customer[is.na(customer)] <- "unknown"
head(customer)
```

```
##         ID Gender Own_Car Own_Realty Children_Count Annual_Income
## 1: 5008804      M       Y          Y              0        427500
## 2: 5008806      M       Y          Y              0        112500
## 3: 5008808      F       N          Y              0        270000
## 4: 5008812      F       N          Y              0        283500
## 5: 5008815      M       Y          Y              0        270000
## 6: 5008820      M       Y          Y              0        135000
##             Income_Type               Education_Type       Marital_Status
## 1:              Working              Higher education       Civil marriage
## 2:              Working Secondary / secondary special              Married
## 3: Commercial associate Secondary / secondary special Single / not married
## 4:            Pensioner              Higher education            Separated
## 5:              Working              Higher education              Married
## 6: Commercial associate Secondary / secondary special              Married
##         Housing_Type DAYS_BIRTH DAYS_EMPLOYED FLAG_MOBIL FLAG_WORK_PHONE
## 1:  Rented apartment     -12005         -4542          1               1
## 2: House / apartment     -21474         -1134          1               0
## 3: House / apartment     -19110         -3051          1               0
## 4: House / apartment     -22464        365243          1               0
## 5: House / apartment     -16872          -769          1               1
## 6: House / apartment     -17778         -1194          1               0
##    FLAG_PHONE FLAG_EMAIL Occupation_Type Family_members_count STATUS Client
```

```
## 1:          0          0         unknown                   2      3      1
## 2:          0          0  Security staff                   2      2      0
## 3:          1          1     Sales staff                   1      2      0
## 4:          0          0         unknown                   1      2      0
## 5:          1          1     Accountants                   2      2      0
## 6:          0          0        Laborers                   2      2      0
```

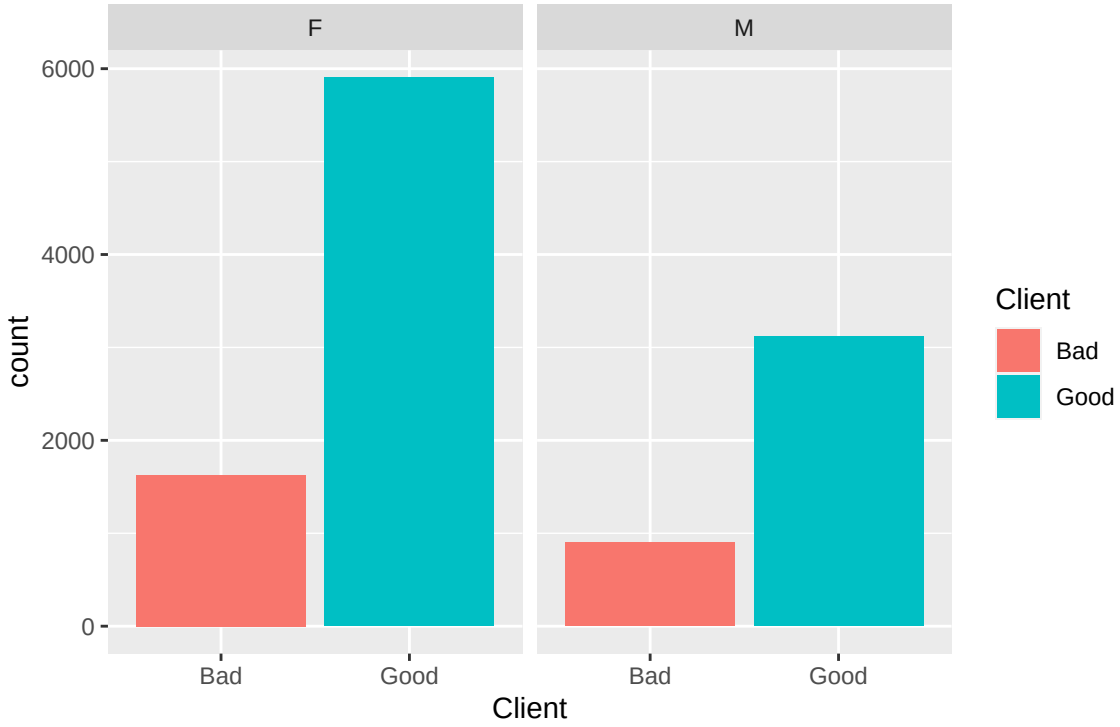
```
customer1 <- customer
customer1$Client[customer1$Client == 0] <- "Good"
customer1$Client[customer1$Client == 1] <- "Bad"
head(customer1)
```

```
##         ID Gender Own_Car Own_Realty Children_Count Annual_Income
## 1: 5008804      M       Y          Y              0        427500
## 2: 5008806      M       Y          Y              0        112500
## 3: 5008808      F       N          Y              0        270000
## 4: 5008812      F       N          Y              0        283500
## 5: 5008815      M       Y          Y              0        270000
## 6: 5008820      M       Y          Y              0        135000
##             Income_Type                Education_Type       Marital_Status
## 1:              Working               Higher education       Civil marriage
## 2:              Working Secondary / secondary special              Married
## 3: Commercial associate Secondary / secondary special Single / not married
## 4:            Pensioner               Higher education            Separated
## 5:              Working               Higher education              Married
## 6: Commercial associate Secondary / secondary special              Married
##         Housing_Type DAYS_BIRTH DAYS_EMPLOYED FLAG_MOBIL FLAG_WORK_PHONE
## 1:  Rented apartment     -12005         -4542          1               1
## 2: House / apartment     -21474         -1134          1               0
## 3: House / apartment     -19110         -3051          1               0
## 4: House / apartment     -22464        365243          1               0
## 5: House / apartment     -16872          -769          1               1
## 6: House / apartment     -17778         -1194          1               0
##    FLAG_PHONE FLAG_EMAIL Occupation_Type Family_members_count STATUS Client
## 1:          0          0         unknown                   2      3    Bad
## 2:          0          0  Security staff                   2      2   Good
## 3:          1          1     Sales staff                   1      2   Good
## 4:          0          0         unknown                   1      2   Good
## 5:          1          1     Accountants                   2      2   Good
## 6:          0          0        Laborers                   2      2   Good
```

Data Analysis

Analyzing data based on the gender

```
(ggplot(customer1, aes(Client, ..count..)) +
  geom_bar(aes(fill = Client), position = "stack") +
  facet_grid(~Gender))
```



There are double the number of female applicants in the data set than male applicants. Relatively it looks like there are more male bad clients than female bad clients.

Creating a subset data set with only binary variables

```
new_customer1 <- customer1 %>%
  select(Own_Car,Own_Realty,Work_Phone = FLAG_WORK_PHONE,Phone = FLAG_PHONE,Email =FLAG_EMAIL,Client, G
head(new_customer1)
```

```
##    Own_Car Own_Realty Work_Phone Phone Email Client Gender
## 1:       Y          Y          1     0     0    Bad      M
## 2:       Y          Y          0     0     0   Good      M
## 3:       N          Y          0     1     1   Good      F
## 4:       N          Y          0     0     0   Good      F
## 5:       Y          Y          1     1     1   Good      M
## 6:       Y          Y          0     0     0   Good      M
```

```
new_customer1$Own_Car[new_customer1$Own_Car == "Y"] <- 1
new_customer1$Own_Car[new_customer1$Own_Car == "N"] <- 0
new_customer1$Own_Realty[new_customer1$Own_Realty == "Y"] <- 1
new_customer1$Own_Realty[new_customer1$Own_Realty == "N"] <- 0
new_customer1$Own_Realty <- as.numeric(new_customer1$Own_Realty)
new_customer1$Own_Car <- as.numeric(new_customer1$Own_Car)
head(new_customer1)
```

```
##    Own_Car Own_Realty Work_Phone Phone Email Client Gender
## 1:       1          1          1     0     0    Bad      M
## 2:       1          1          0     0     0   Good      M
## 3:       0          1          0     1     1   Good      F
```

```
## 4:       0          1          0     0     0   Good      F
## 5:       1          1          1     1     1   Good      M
## 6:       1          1          0     0     0   Good      M
```

Here, the character variables are changed into numerical values by replacing them to 0 and 1.

```
new_customer1 <- new_customer1 %>%
  pivot_longer(c("Own_Car", "Own_Realty", "Work_Phone","Phone", "Email" ), names_to = "Flag", values_to
new_customer1
```

```
## # A tibble: 57,760 x 4
##    Client Gender Flag       value
##    <chr>  <chr>  <chr>      <dbl>
##  1 Bad    M      Own_Car        1
##  2 Bad    M      Own_Realty     1
##  3 Bad    M      Work_Phone     1
##  4 Bad    M      Phone          0
##  5 Bad    M      Email          0
##  6 Good   M      Own_Car        1
##  7 Good   M      Own_Realty     1
##  8 Good   M      Work_Phone     0
##  9 Good   M      Phone          0
## 10 Good   M      Email          0
## # ... with 57,750 more rows
```

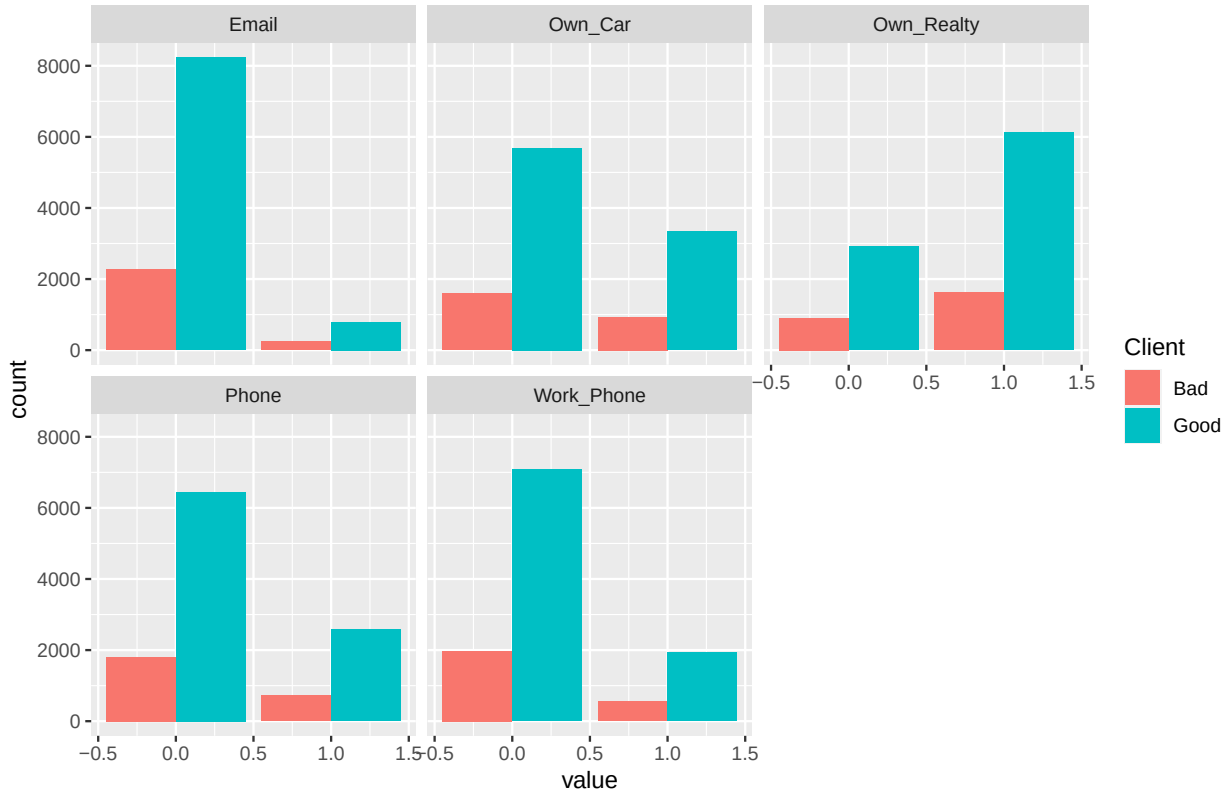
I have used pivot_longer to create a new column Flag that contains all binary flag variables mentioned earlier. The value takes 0 and 1.

```
new_customer1 %>%
  group_by(Flag,Client,value) %>%
count(Flag)
```

```
## # A tibble: 20 x 4
## # Groups:   Flag, Client, value [20]
##    Flag       Client value     n
##    <chr>      <chr>  <dbl> <int>
##  1 Email      Bad        0  2278
##  2 Email      Bad        1   252
##  3 Email      Good       0  8232
##  4 Email      Good       1   790
##  5 Own_Car    Bad        0  1597
##  6 Own_Car    Bad        1   933
##  7 Own_Car    Good       0  5691
##  8 Own_Car    Good       1  3331
##  9 Own_Realty Bad        0   908
## 10 Own_Realty Bad        1  1622
## 11 Own_Realty Good       0  2908
## 12 Own_Realty Good       1  6114
## 13 Phone      Bad        0  1806
## 14 Phone      Bad        1   724
## 15 Phone      Good       0  6450
## 16 Phone      Good       1  2572
```

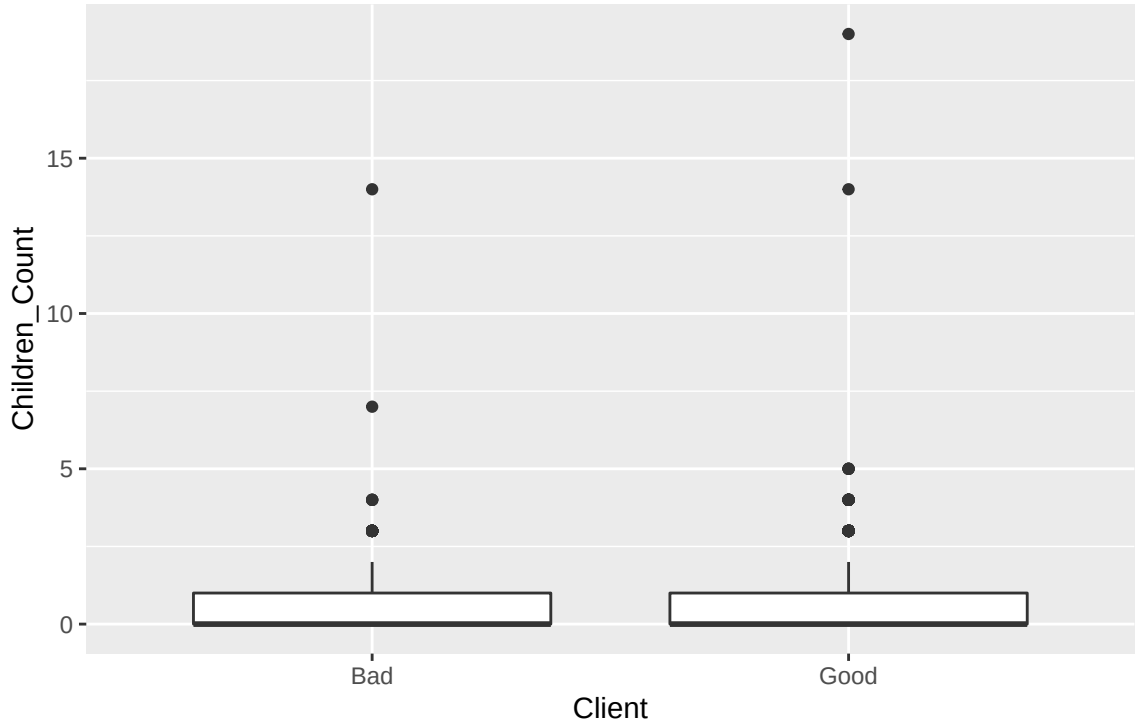
```
## 17 Work_Phone Bad         0  1970
## 18 Work_Phone Bad         1   560
## 19 Work_Phone Good        0  7087
## 20 Work_Phone Good        1  1935
```

```
ggplot(new_customer1, aes(value, ..count..)) +
  geom_bar(aes(fill = Client), position = "dodge") +
  facet_wrap(~Flag)
```



Plotting the graph for the feature children count

```
ggplot(customer1) + geom_boxplot(aes(x = Client, y = Children_Count))
```



Here we can observe that there are outliers where some applicants have 7, 14, and 19 children. Surprisingly some of them are good clients, where as most of the bad clients have either 0 or 1 children.

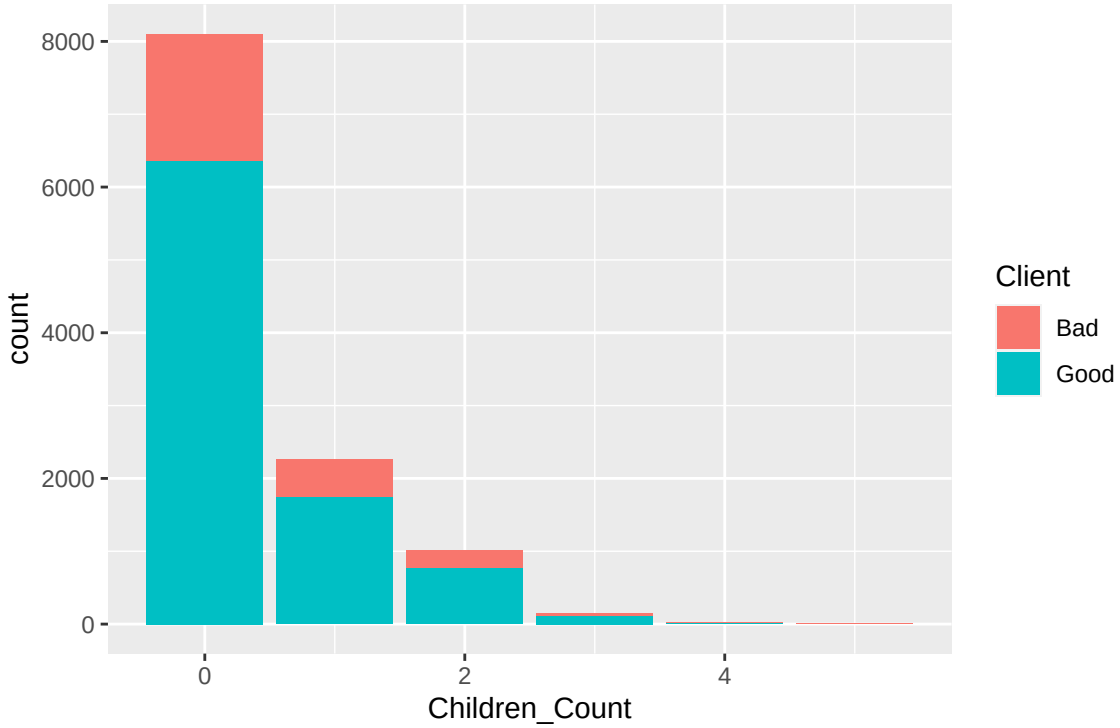
```
customer$Children_Count[customer$Children_Count >= 5] <- 5
customer1$Children_Count[customer1$Children_Count >= 5] <- 5
customer1 %>%
  count(Children_Count)
```

```
##    Children_Count    n
## 1:              0 8102
## 2:              1 2261
## 3:              2 1013
## 4:              3  145
## 5:              4   23
## 6:              5    8
```

I have changed the data set to modify number of children to 5 if the count is greater than 5.

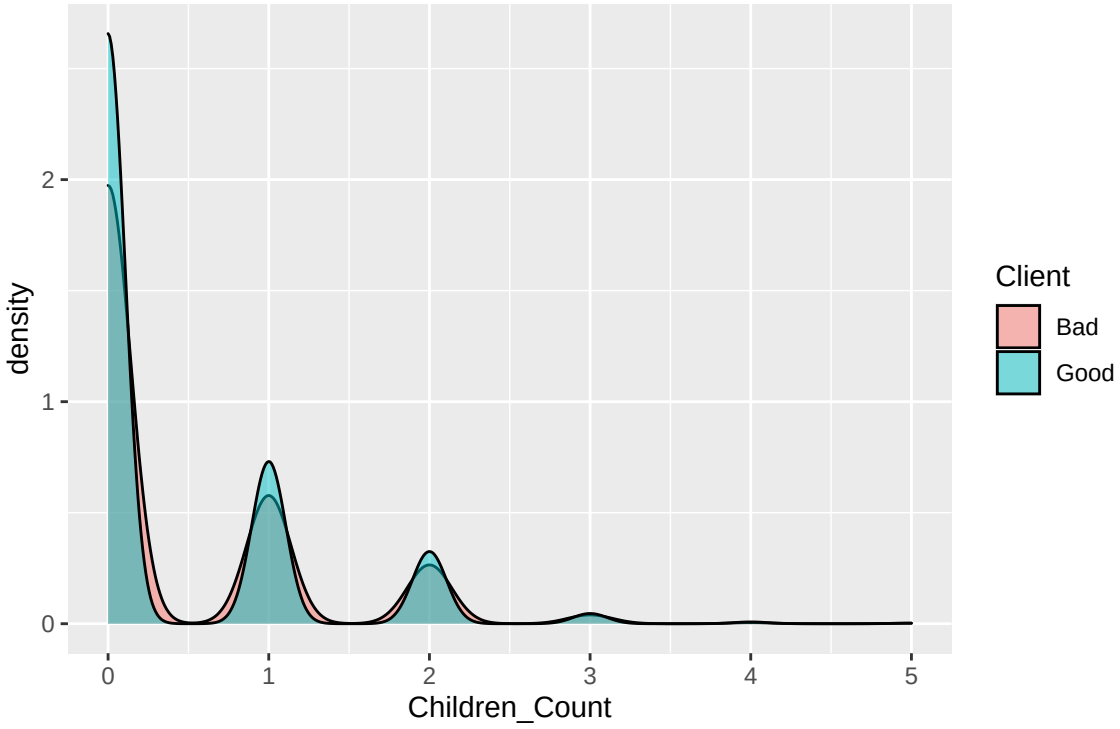
```
child <- customer1 %>%
  select(Client,Children_Count) %>%
  group_by(Children_Count, Client) %>%
  mutate(count = n())
child <- unique(child)

ggplot(child) +
  geom_bar(mapping = aes(x = Children_Count, y = count, fill = Client), stat = "identity")
```



This plot shows that there are more number of applicants with 0 children, but we cannot identify which applicant are more bad clients. Lets take a look at the below density plot.

```
ggplot(customer1,aes(Children_Count, fill = Client))+
  geom_density(alpha = 0.5)
```

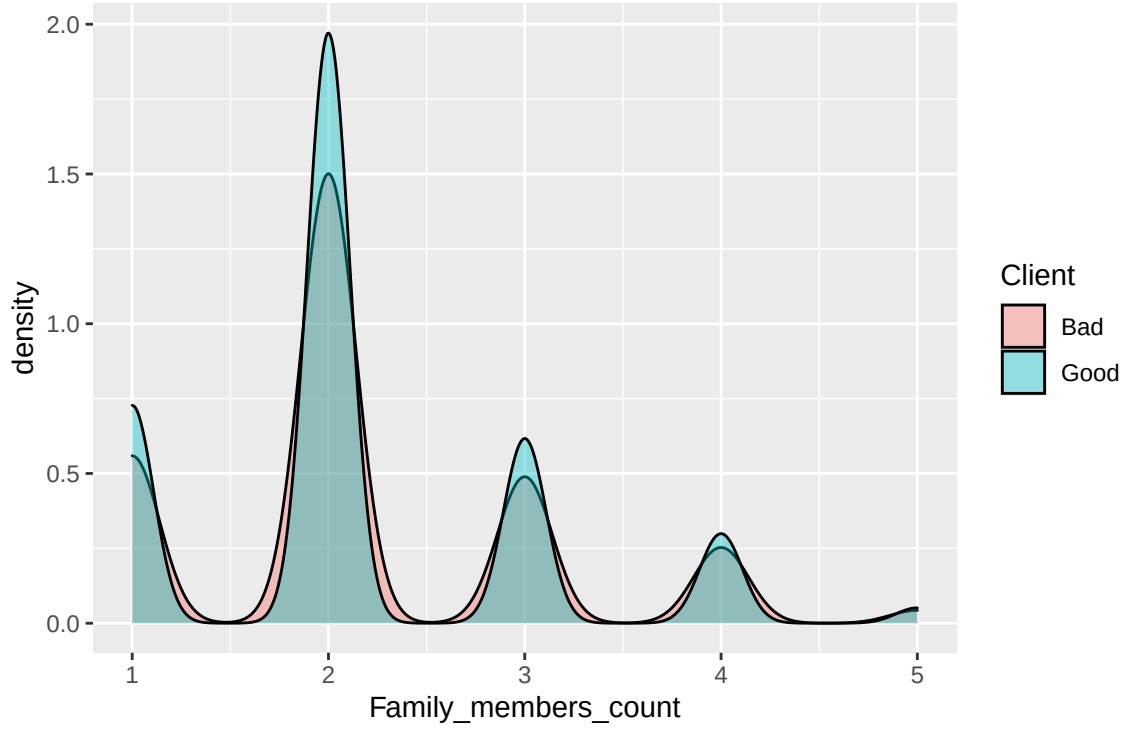


From this plot we can say that number of children is directly affecting the applicant's credit status. Looks like the applicants with 3 children are mostly bad clients than others.

```
customer$Family_members_count[customer$Family_members_count >= 5] <- 5
customer1$Family_members_count[customer1$Family_members_count >= 5] <- 5
```

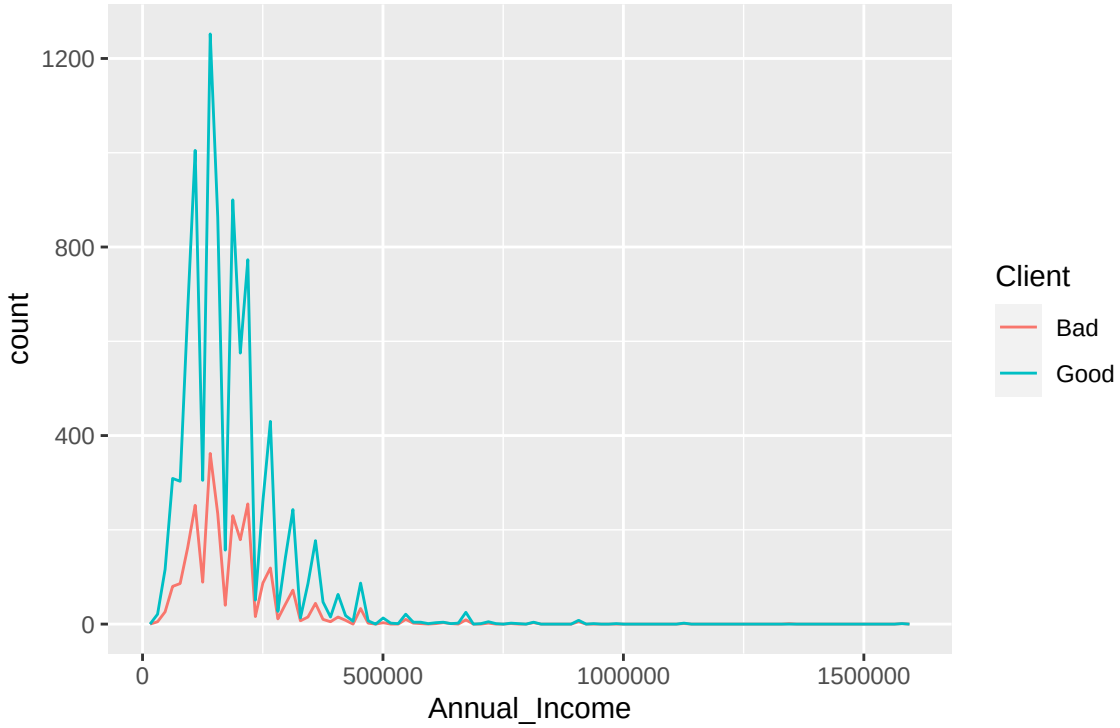
I have convereted the family members count more than 5 as 5 for better data analysis.

```
ggplot(customer1,aes(Family_members_count, fill = Client))+
  geom_density(alpha = 0.4)
```



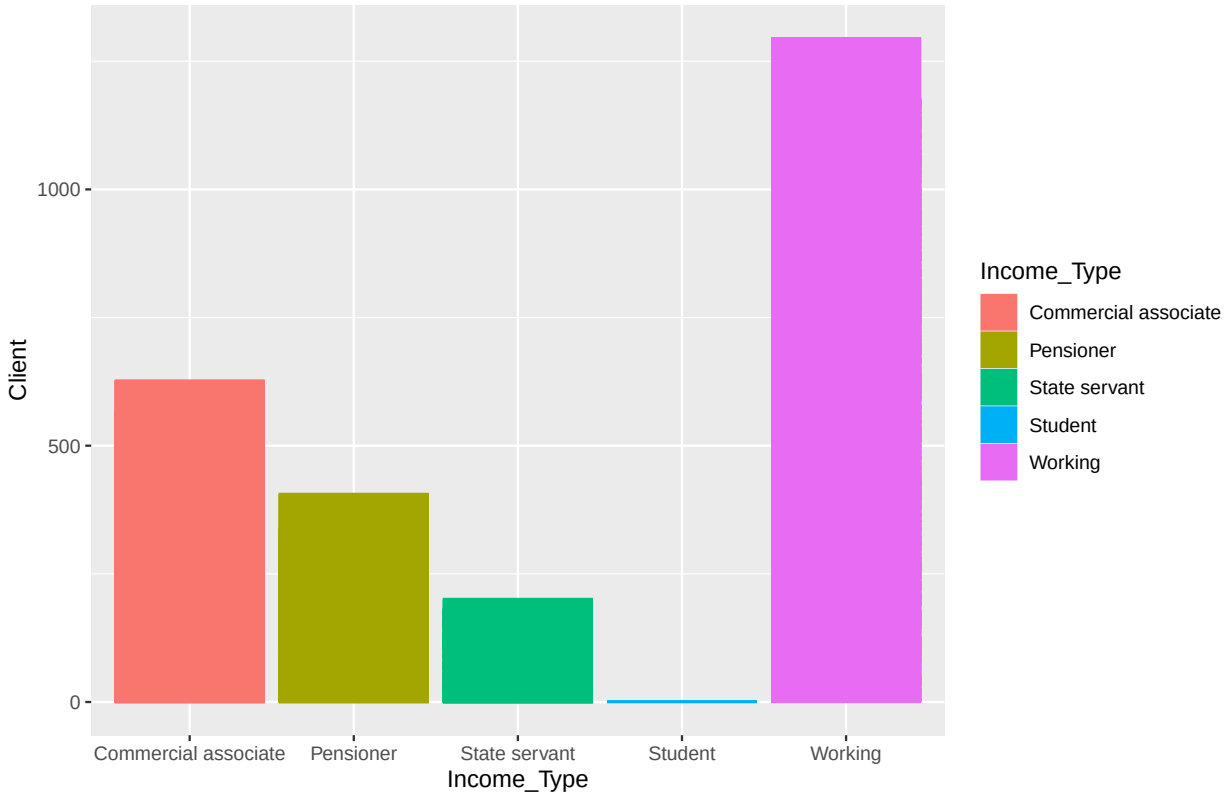
From the plot we can say that there are more applicants with family count two. May be they are the applicants who have 0 children. We can say from the graph that applicants with family count more than 4 are relatively bad clients and this feature is directly affecting the credit status of an applicant.

```
ggplot(customer1, mapping = aes(x = Annual_Income,fill = Client, color = Client)) +
  geom_freqpoly(bins = 100)
```



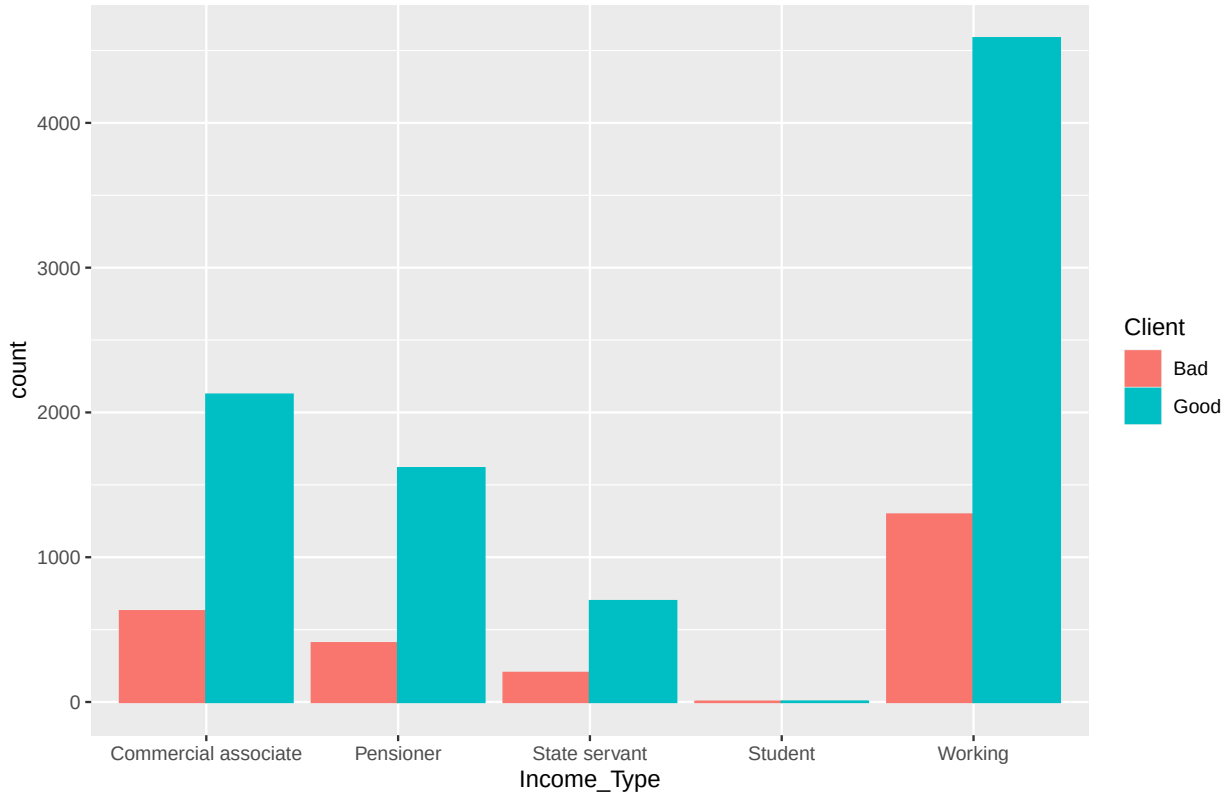
There are very few applicants whose annual income is more than 500k, where as most of the applicants have annual income between 50k to 300k. From the graph, we can say that most of the bad clients have their income between 80k to 150k.

```
ggplot(customer) +
  geom_bar(mapping = aes(x = Income_Type, y = Client, fill = Income_Type,color = Income_Type), stat = "
```

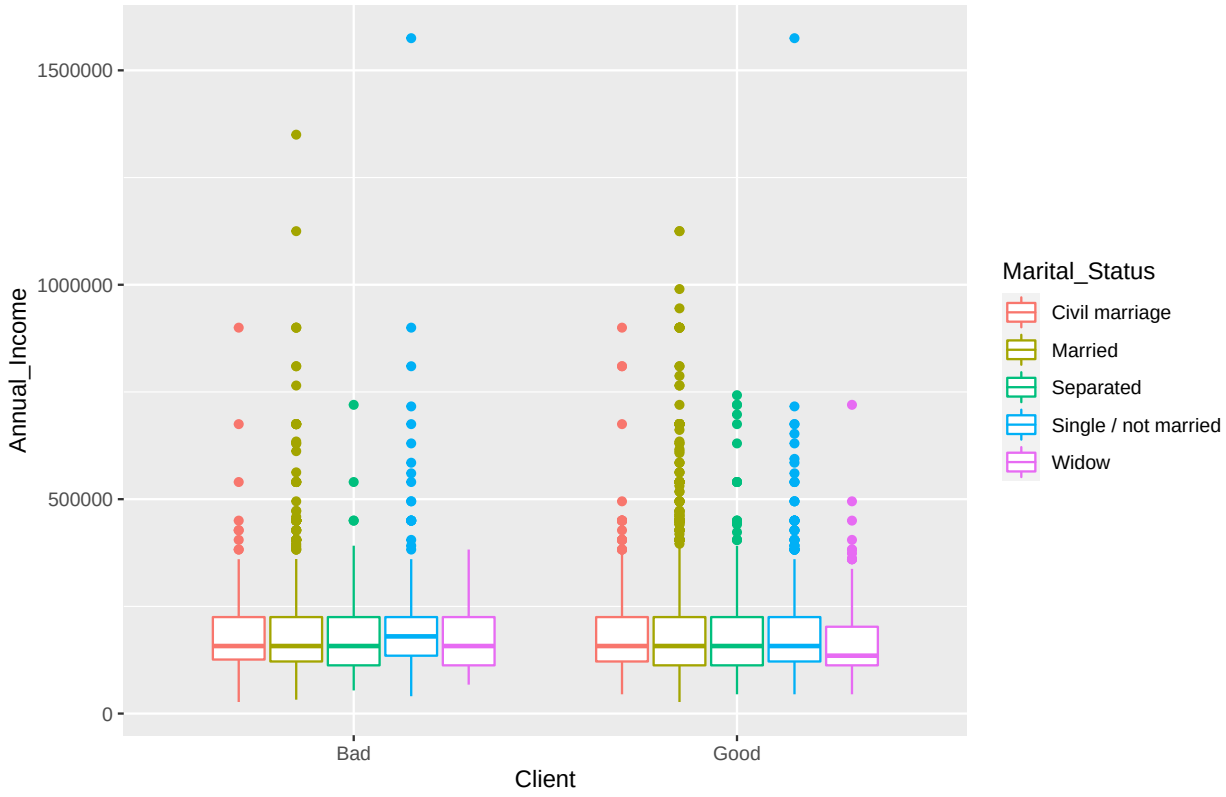


There are more bad clients whose income type is working compared to others. From the below graph, we can also say that there are equal number of bad clients as good clients who are students.

```
ggplot(customer1, aes(Income_Type, ..count..)) + geom_bar(aes(fill = Client, color = Client), position =
```

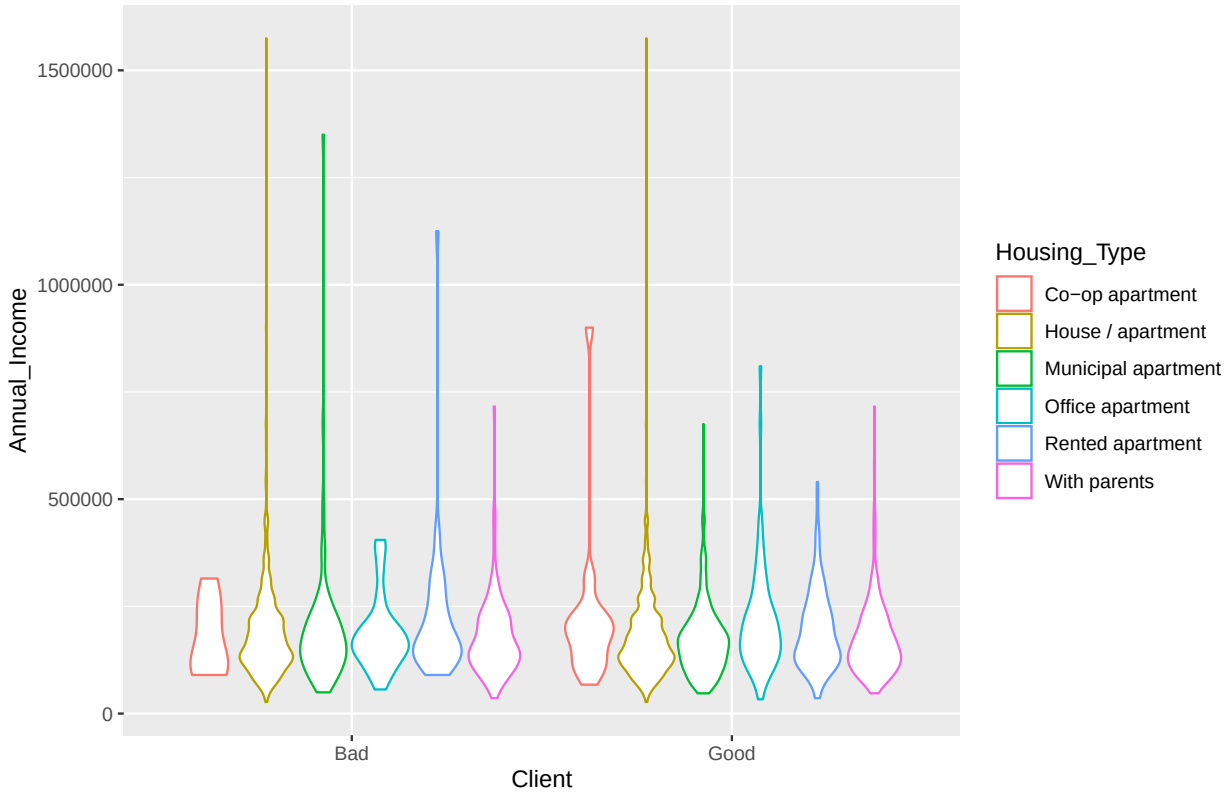



```
ggplot(customer1,aes(x = Client, y = Annual_Income, color = Marital_Status)) +
  geom_boxplot()
```



This graph shows the relation between applicants marital status, their income and credit status. We can see that applicants with marital status as single/not married and widow are bad clients compared to others. The married applicants are getting relatively more salaries.

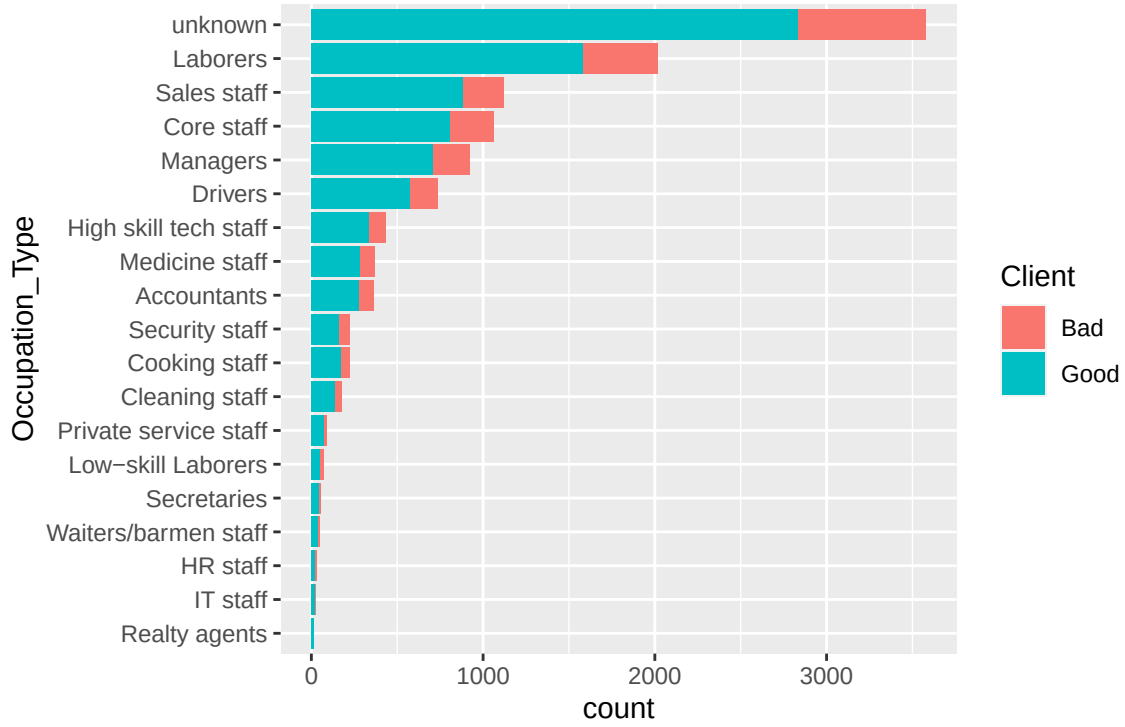
```
ggplot(customer1,aes(x = Client, y = Annual_Income, color = Housing_Type)) +
  geom_violin()
```



From the graph we can say that people staying in municipal apartment and office apartment are relatively bad clients where as people staying in co-op apartments are less bad clients. Applicants whose housing type is house/apartment are having higher annual income.

```
occupation <- customer1 %>%
  select(Occupation_Type,Client)
occupation <- occupation%>%group_by(Occupation_Type,Client)%>%mutate(count=n())
occupation <- unique(occupation)

myColors <- brewer.pal(6, "Set1")
df <- transform(occupation, Occupation_Type = reorder(Occupation_Type, count))
ggplot(df, aes(x=Occupation_Type, y=count, fill=Client)) + geom_bar(stat="identity") + scale_colour_manu
```



This plot shows various occupations the applicants have and how occupation type is affecting their credit status. Low-skill laborers are relatively bad clients than other applicants with different occupation.

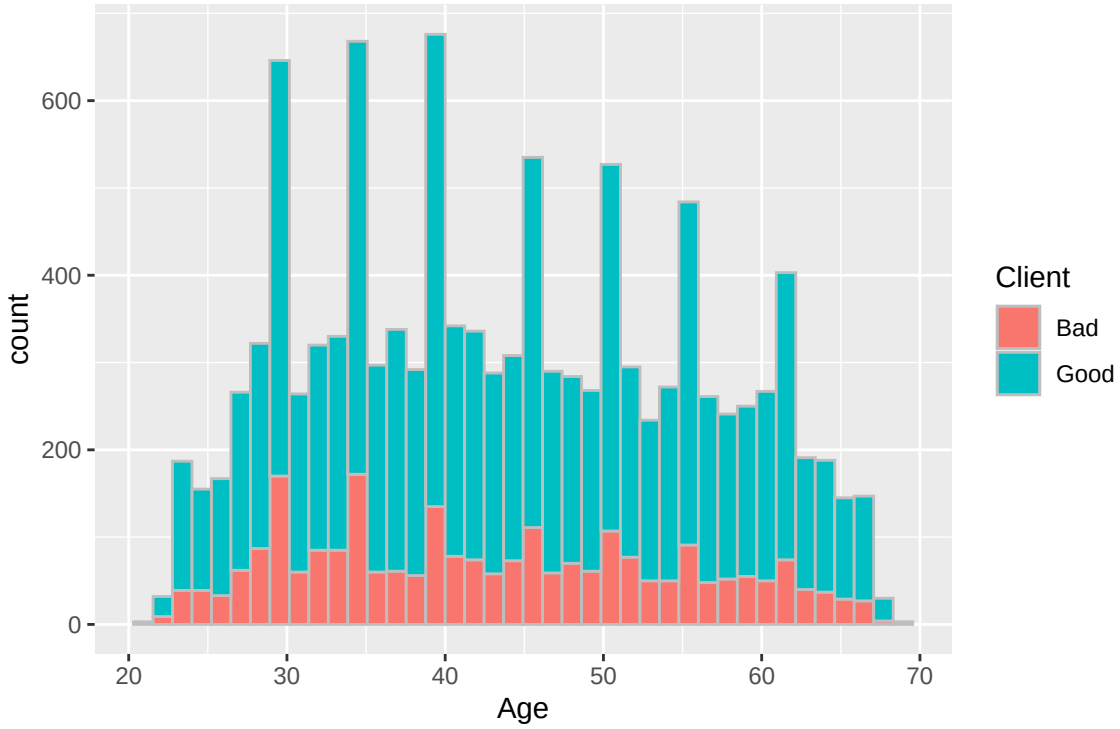
```
customer <- customer %>%
  mutate(Age = round(abs(customer$DAYS_BIRTH/365), digits = 0))
customer1 <- customer1 %>%
  mutate(Age = round(abs(customer1$DAYS_BIRTH/365), digits = 0))
head(customer)
```

```
##         ID Gender Own_Car Own_Realty Children_Count Annual_Income
## 1: 5008804      M       Y           Y              0        427500
## 2: 5008806      M       Y           Y              0        112500
## 3: 5008808      F       N           Y              0        270000
## 4: 5008812      F       N           Y              0        283500
## 5: 5008815      M       Y           Y              0        270000
## 6: 5008820      M       Y           Y              0        135000
##              Income_Type                Education_Type       Marital_Status
## 1:               Working                Higher education       Civil marriage
## 2:               Working Secondary / secondary special              Married
## 3: Commercial associate Secondary / secondary special Single / not married
## 4:             Pensioner                Higher education            Separated
## 5:               Working                Higher education              Married
## 6: Commercial associate Secondary / secondary special              Married
##          Housing_Type DAYS_BIRTH DAYS_EMPLOYED FLAG_MOBIL FLAG_WORK_PHONE
## 1:  Rented apartment     -12005         -4542          1               1
## 2: House / apartment     -21474         -1134          1               0
## 3: House / apartment     -19110         -3051          1               0
## 4: House / apartment     -22464        365243          1               0
## 5: House / apartment     -16872          -769          1               1
## 6: House / apartment     -17778         -1194          1               0
```

```
##    FLAG_PHONE FLAG_EMAIL Occupation_Type Family_members_count STATUS Client Age
## 1:          0          0         unknown                     2      3      1  33
## 2:          0          0  Security staff                     2      2      0  59
## 3:          1          1     Sales staff                     1      2      0  52
## 4:          0          0         unknown                     1      2      0  62
## 5:          1          1     Accountants                     2      2      0  46
## 6:          0          0        Laborers                     2      2      0  49
```

Here, age of applicants is calculated based on the column DAYS_BIRTH by dividing it with number of days in a year and getting teh absolute value.

```
ggplot(customer1, mapping = aes(x = Age,fill = Client)) +
  geom_histogram(bins = 40,color='grey',position=position_stack(reverse = TRUE))
```



The average number of applicants who are good clients are more than bad clients.

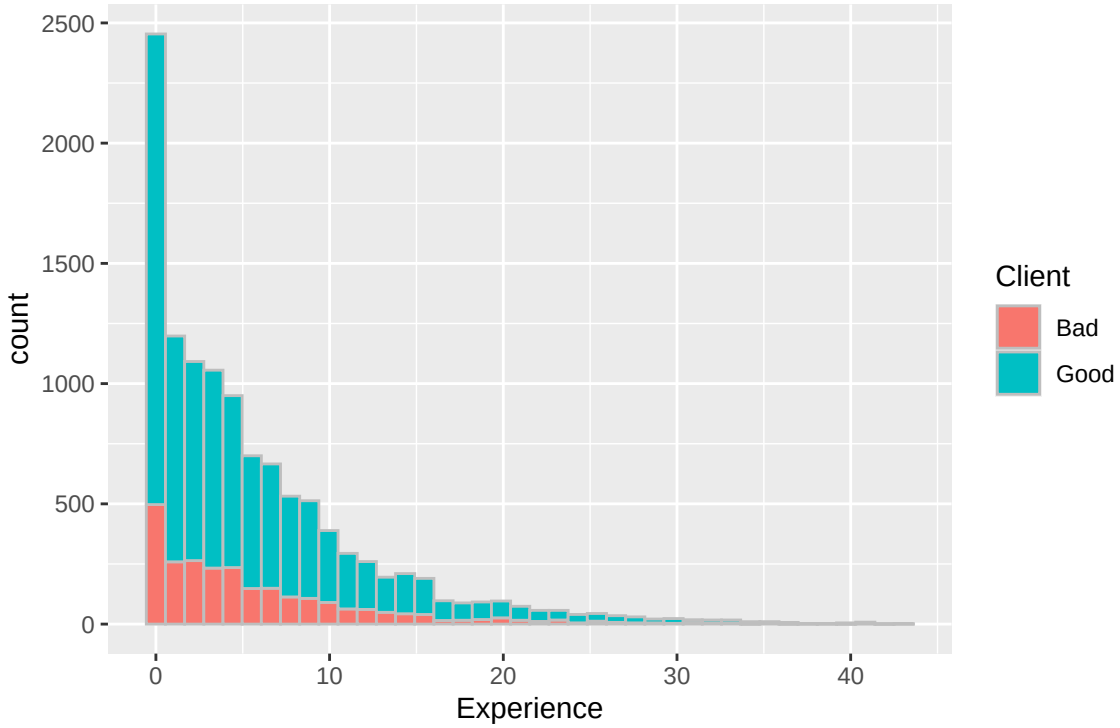
```
customer <- customer %>%
  mutate(Experience = ifelse(DAYS_EMPLOYED <= 0, abs(DAYS_EMPLOYED/365), 0 ))
customer1 <- customer1 %>%
  mutate(Experience = ifelse(DAYS_EMPLOYED <= 0, abs(DAYS_EMPLOYED/365), 0 ))
head(customer1)
```

```
##         ID Gender Own_Car Own_Realty Children_Count Annual_Income
## 1: 5008804      M       Y          Y              0        427500
## 2: 5008806      M       Y          Y              0        112500
## 3: 5008808      F       N          Y              0        270000
## 4: 5008812      F       N          Y              0        283500
## 5: 5008815      M       Y          Y              0        270000
## 6: 5008820      M       Y          Y              0        135000
```

```
##              Income_Type                Education_Type       Marital_Status
## 1:               Working              Higher education       Civil marriage
## 2:               Working Secondary / secondary special              Married
## 3: Commercial associate Secondary / secondary special Single / not married
## 4:             Pensioner              Higher education            Separated
## 5:               Working              Higher education              Married
## 6: Commercial associate Secondary / secondary special              Married
##          Housing_Type DAYS_BIRTH DAYS_EMPLOYED FLAG_MOBIL FLAG_WORK_PHONE
## 1:  Rented apartment     -12005         -4542          1               1
## 2: House / apartment     -21474         -1134          1               0
## 3: House / apartment     -19110         -3051          1               0
## 4: House / apartment     -22464        365243          1               0
## 5: House / apartment     -16872          -769          1               1
## 6: House / apartment     -17778         -1194          1               0
##    FLAG_PHONE FLAG_EMAIL Occupation_Type Family_members_count STATUS Client Age
## 1:          0          0         unknown                    2      3    Bad  33
## 2:          0          0  Security staff                    2      2   Good  59
## 3:          1          1     Sales staff                    1      2   Good  52
## 4:          0          0         unknown                    1      2   Good  62
## 5:          1          1     Accountants                    2      2   Good  46
## 6:          0          0        Laborers                    2      2   Good  49
##    Experience
## 1:  12.443836
## 2:   3.106849
## 3:   8.358904
## 4:   0.000000
## 5:   2.106849
## 6:   3.271233
```

A new column is created with experience by dividing the number if days employed with 365 days in a year. Applicants whose experience is positive takes the value 0 as they are unemployed.

```
ggplot(customer1, mapping = aes(x = Experience,fill = Client)) +
  geom_histogram(bins = 40,color='grey',position=position_stack(reverse = TRUE))
```



We can see that the applicants with less experience are more bad clients. We can observe that as the experience increases, the bad client bars went down. Surprisingly there are more bad clients with experience 5 years.

Data Modeling

Converting all the binary variables to 0 and 1 for modeling.

```
new_data <- customer %>%
  select(Client,Gender, Own_Car, Own_Realty,Children_Count,Annual_Income,FLAG_WORK_PHONE, FLAG_PHONE, FL
new_data$Gender[new_data$Gender == "F"] <- 0
new_data$Gender[new_data$Gender == "M"] <- 1
new_data$Gender <- as.numeric(new_data$Gender)
new_data$Own_Car[new_data$Own_Car == "Y"] <- 1
new_data$Own_Car[new_data$Own_Car == "N"] <- 0
new_data$Own_Car <- as.numeric(new_data$Own_Car)
new_data$Own_Realty[new_data$Own_Realty == "Y"] <- 1
new_data$Own_Realty[new_data$Own_Realty == "N"] <- 0
new_data$Own_Realty <- as.numeric(new_data$Own_Realty)
head(new_data)
```

```
##    Client Gender Own_Car Own_Realty Children_Count Annual_Income
## 1:      1      1       1          1              0        427500
## 2:      0      1       1          1              0        112500
## 3:      0      0       0          1              0        270000
## 4:      0      0       0          1              0        283500
## 5:      0      1       1          1              0        270000
## 6:      0      1       1          1              0        135000
##    FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL Age Experience          Income_Type
```

```
## 1:               1          0          0  33  12.443836              Working
## 2:               0          0          0  59   3.106849              Working
## 3:               0          1          1  52   8.358904 Commercial associate
## 4:               0          0          0  62   0.000000            Pensioner
## 5:               1          1          1  46   2.106849              Working
## 6:               0          0          0  49   3.271233 Commercial associate
##    Family_members_count                Education_Type       Marital_Status
## 1:                    2              Higher education       Civil marriage
## 2:                    2 Secondary / secondary special              Married
## 3:                    1 Secondary / secondary special Single / not married
## 4:                    1              Higher education            Separated
## 5:                    2              Higher education              Married
## 6:                    2 Secondary / secondary special              Married
```

Converting categorical variables to binary variables.

```
student = ifelse(new_data$Income_Type=="Student",1,0)
CommercialAssociate = ifelse(new_data$Income_Type=="Commercial associate",1,0)
Pensioner = ifelse(new_data$Income_Type=="Pensioner",1,0)
StateServant = ifelse(new_data$Income_Type=="State servant",1,0)
Working = ifelse(new_data$Income_Type=="Working",1,0)
HigherEducation = ifelse(new_data$Education_Type=="Higher education",1,0)
IncompleteEducation = ifelse(new_data$Education_Type=="Incomplete higher",1,0)
SecondaryEducation = ifelse(new_data$Education_Type=="Secondary / secondary special",1,0)
widow = ifelse(new_data$Marital_Status=="Widow",1,0)
model_data = cbind(new_data,student,CommercialAssociate,Pensioner,StateServant,Working,HigherEducation,

model_data = subset(model_data, select = -c(12,14,15) )
head(model_data)
```

```
##    Client Gender Own_Car Own_Realty Children_Count Annual_Income
## 1:      1      1       1          1              0        427500
## 2:      0      1       1          1              0        112500
## 3:      0      0       0          1              0        270000
## 4:      0      0       0          1              0        283500
## 5:      0      1       1          1              0        270000
## 6:      0      1       1          1              0        135000
##    FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL Age Experience Family_members_count
## 1:               1          0          0  33  12.443836                    2
## 2:               0          0          0  59   3.106849                    2
## 3:               0          1          1  52   8.358904                    1
## 4:               0          0          0  62   0.000000                    1
## 5:               1          1          1  46   2.106849                    2
## 6:               0          0          0  49   3.271233                    2
##    student CommercialAssociate Pensioner StateServant Working HigherEducation
## 1:       0                   0         0            0       1               1
## 2:       0                   0         0            0       1               0
## 3:       0                   1         0            0       0               0
## 4:       0                   0         1            0       0               1
## 5:       0                   0         0            0       1               1
## 6:       0                   1         0            0       0               0
##    IncompleteEducation SecondaryEducation widow
## 1:                   0                 0     0
```

```
## 2:                    0                 1     0
## 3:                    0                 1     0
## 4:                    0                 0     0
## 5:                    0                 0     0
## 6:                    0                 1     0
```

Here, categorical variables are changed to binary. If an applicant has education type as higher education, the colum HigherEducatuion takes the value 1 and other related columns IncompleteEducation, SecondaryEducation takes 0 value.

```
correlation_res <- cor(model_data)
round(correlation_res, 2)
```

```
##                       Client Gender Own_Car Own_Realty Children_Count
## Client                  1.00   0.01    0.00      -0.03           0.01
## Gender                  0.01   1.00    0.36      -0.06           0.07
## Own_Car                 0.00   0.36    1.00      -0.01           0.10
## Own_Realty             -0.03  -0.06   -0.01       1.00          -0.01
## Children_Count          0.01   0.07    0.10      -0.01           1.00
## Annual_Income           0.03   0.20    0.22       0.03           0.04
## FLAG_WORK_PHONE         0.01   0.06    0.03      -0.19           0.05
## FLAG_PHONE              0.00  -0.01    0.01      -0.06          -0.03
## FLAG_EMAIL              0.02   0.00    0.02       0.06           0.01
## Age                    -0.04  -0.18   -0.13       0.13          -0.34
## Experience              0.00  -0.04    0.02      -0.02           0.04
## Family_members_count    0.01   0.09    0.15       0.00           0.88
## student                 0.00   0.00   -0.01       0.00           0.00
## CommercialAssociate     0.01   0.04    0.06      -0.01           0.04
## Pensioner              -0.02  -0.17   -0.15       0.08          -0.23
## StateServant            0.00  -0.05   -0.02      -0.01           0.04
## Working                 0.00   0.12    0.07      -0.05           0.12
## HigherEducation         0.01  -0.02    0.09      -0.01           0.03
## IncompleteEducation     0.02   0.03    0.02      -0.02           0.00
## SecondaryEducation     -0.01   0.00   -0.09       0.01          -0.03
## widow                   0.00  -0.14   -0.10       0.02          -0.10
##                       Annual_Income FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL   Age
## Client                         0.03            0.01       0.00       0.02 -0.04
## Gender                         0.20            0.06      -0.01       0.00 -0.18
## Own_Car                        0.22            0.03       0.01       0.02 -0.13
## Own_Realty                     0.03           -0.19      -0.06       0.06  0.13
## Children_Count                 0.04            0.05      -0.03       0.01 -0.34
## Annual_Income                  1.00           -0.03       0.02       0.09 -0.07
## FLAG_WORK_PHONE               -0.03            1.00       0.29      -0.05 -0.18
## FLAG_PHONE                     0.02            0.29       1.00       0.01  0.03
## FLAG_EMAIL                     0.09           -0.05       0.01       1.00 -0.11
## Age                           -0.07           -0.18       0.03      -0.11  1.00
## Experience                     0.09            0.11       0.04      -0.01 -0.01
## Family_members_count           0.03            0.06      -0.02       0.00 -0.29
## student                       -0.01           -0.01      -0.01       0.03 -0.01
## CommercialAssociate            0.17            0.01       0.01       0.07 -0.17
## Pensioner                     -0.16           -0.24      -0.01      -0.08  0.61
## StateServant                   0.04            0.01      -0.01       0.00 -0.06
## Working                       -0.04            0.17       0.00       0.00 -0.29
```

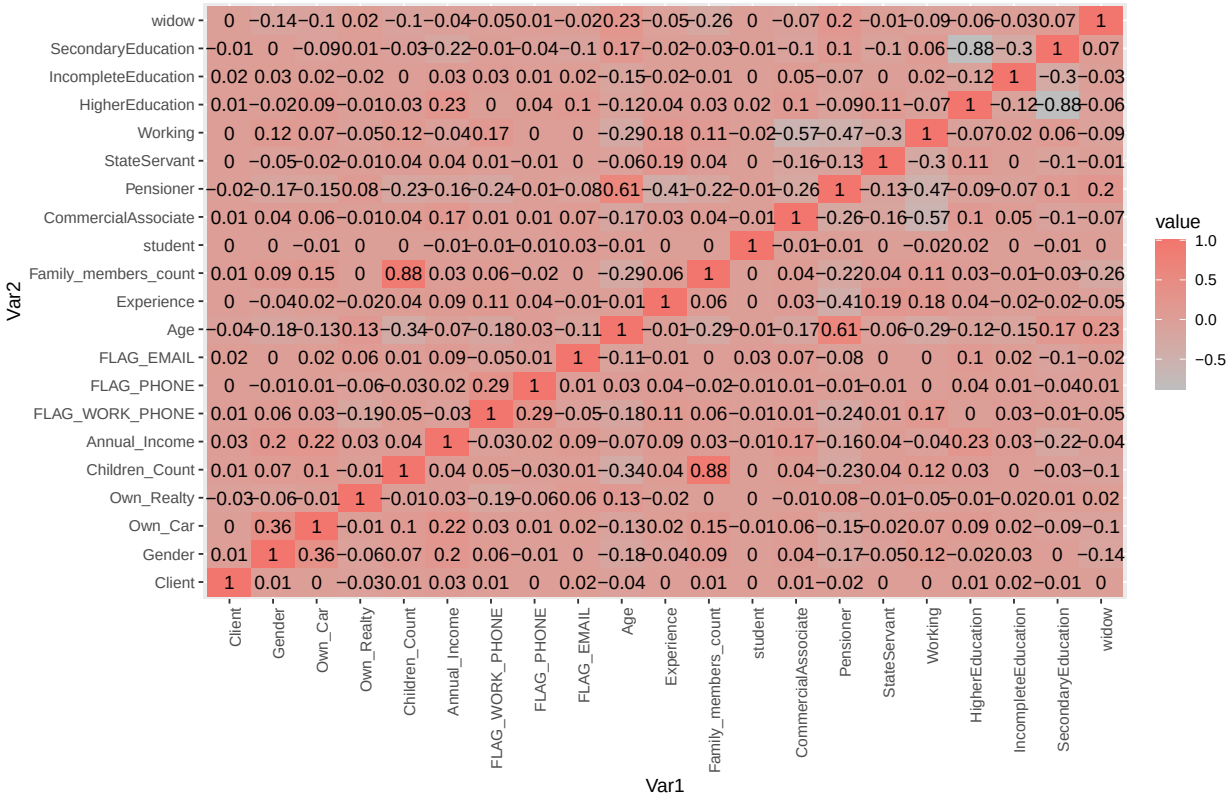
```
## HigherEducation              0.23            0.00       0.04       0.10 -0.12
## IncompleteEducation          0.03            0.03       0.01       0.02 -0.15
## SecondaryEducation          -0.22           -0.01      -0.04      -0.10  0.17
## widow                       -0.04           -0.05       0.01      -0.02  0.23
##                       Experience Family_members_count student
## Client                      0.00                 0.01    0.00
## Gender                     -0.04                 0.09    0.00
## Own_Car                     0.02                 0.15   -0.01
## Own_Realty                 -0.02                 0.00    0.00
## Children_Count              0.04                 0.88    0.00
## Annual_Income               0.09                 0.03   -0.01
## FLAG_WORK_PHONE             0.11                 0.06   -0.01
## FLAG_PHONE                  0.04                -0.02   -0.01
## FLAG_EMAIL                 -0.01                 0.00    0.03
## Age                        -0.01                -0.29   -0.01
## Experience                  1.00                 0.06    0.00
## Family_members_count        0.06                 1.00    0.00
## student                     0.00                 0.00    1.00
## CommercialAssociate         0.03                 0.04   -0.01
## Pensioner                  -0.41                -0.22   -0.01
## StateServant                0.19                 0.04    0.00
## Working                     0.18                 0.11   -0.02
## HigherEducation             0.04                 0.03    0.02
## IncompleteEducation        -0.02                -0.01    0.00
## SecondaryEducation         -0.02                -0.03   -0.01
## widow                      -0.05                -0.26    0.00
##                       CommercialAssociate Pensioner StateServant Working
## Client                               0.01     -0.02        0.00    0.00
## Gender                               0.04     -0.17       -0.05    0.12
## Own_Car                              0.06     -0.15       -0.02    0.07
## Own_Realty                          -0.01      0.08       -0.01   -0.05
## Children_Count                       0.04     -0.23        0.04    0.12
## Annual_Income                        0.17     -0.16        0.04   -0.04
## FLAG_WORK_PHONE                      0.01     -0.24        0.01    0.17
## FLAG_PHONE                           0.01     -0.01       -0.01    0.00
## FLAG_EMAIL                           0.07     -0.08        0.00    0.00
## Age                                 -0.17      0.61       -0.06   -0.29
## Experience                           0.03     -0.41        0.19    0.18
## Family_members_count                 0.04     -0.22        0.04    0.11
## student                             -0.01     -0.01        0.00   -0.02
## CommercialAssociate                  1.00     -0.26       -0.16   -0.57
## Pensioner                           -0.26      1.00       -0.13   -0.47
## StateServant                        -0.16     -0.13        1.00   -0.30
## Working                             -0.57     -0.47       -0.30    1.00
## HigherEducation                      0.10     -0.09        0.11   -0.07
## IncompleteEducation                  0.05     -0.07        0.00    0.02
## SecondaryEducation                  -0.10      0.10       -0.10    0.06
## widow                               -0.07      0.20       -0.01   -0.09
##                       HigherEducation IncompleteEducation SecondaryEducation
## Client                           0.01               0.02             -0.01
## Gender                          -0.02               0.03              0.00
## Own_Car                          0.09               0.02             -0.09
## Own_Realty                      -0.01              -0.02              0.01
## Children_Count                   0.03               0.00             -0.03
```

```
## Annual_Income                  0.23              0.03             -0.22
## FLAG_WORK_PHONE                0.00              0.03             -0.01
## FLAG_PHONE                     0.04              0.01             -0.04
## FLAG_EMAIL                     0.10              0.02             -0.10
## Age                           -0.12             -0.15              0.17
## Experience                     0.04             -0.02             -0.02
## Family_members_count           0.03             -0.01             -0.03
## student                        0.02              0.00             -0.01
## CommercialAssociate            0.10              0.05             -0.10
## Pensioner                     -0.09             -0.07              0.10
## StateServant                   0.11              0.00             -0.10
## Working                       -0.07              0.02              0.06
## HigherEducation                1.00             -0.12             -0.88
## IncompleteEducation           -0.12              1.00             -0.30
## SecondaryEducation            -0.88             -0.30              1.00
## widow                         -0.06             -0.03              0.07
##                          widow
## Client                    0.00
## Gender                   -0.14
## Own_Car                  -0.10
## Own_Realty                0.02
## Children_Count           -0.10
## Annual_Income            -0.04
## FLAG_WORK_PHONE          -0.05
## FLAG_PHONE                0.01
## FLAG_EMAIL               -0.02
## Age                       0.23
## Experience               -0.05
## Family_members_count     -0.26
## student                   0.00
## CommercialAssociate      -0.07
## Pensioner                 0.20
## StateServant             -0.01
## Working                  -0.09
## HigherEducation          -0.06
## IncompleteEducation      -0.03
## SecondaryEducation        0.07
## widow                     1.00
```

This is the correlation matrix for the data that shows how much each pair of variables are correlated.

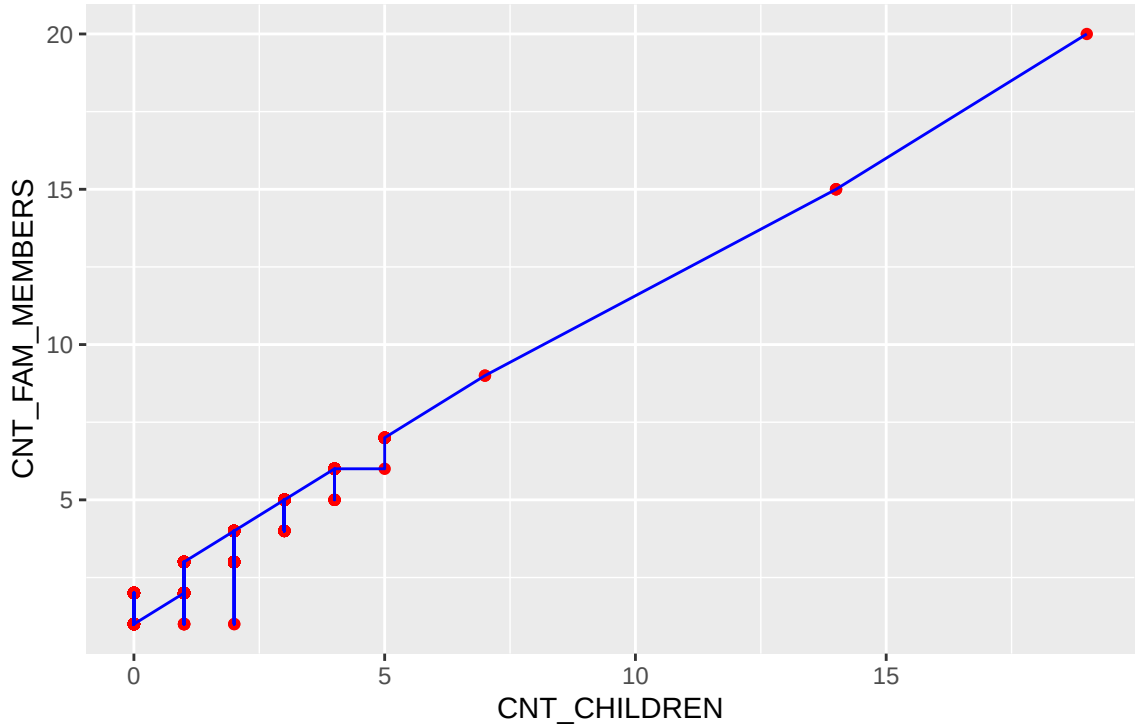
```
cormat <- reshape2::melt(correlation_res)
```

```
ggplot(cormat, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
    geom_text(aes(label = round(value, 2))) +
    scale_fill_gradient(low = "grey", high = "#f2766d")
```



This plot shows correlation matrix. Here we can observe, children_count and family_members_count are highly correlated with value 0.88

```
ggplot(data_application,aes(x = CNT_CHILDREN, y = CNT_FAM_MEMBERS)) +
  geom_point(colour = "red") +
  geom_line(colour = "blue")
```



Lets plot the graph for children count and family members count. It shows a linear relationship. It says that children count is a subset of family members count.

```
index_train = sample(1:nrow(model_data),0.70 * nrow(model_data))
# Create training set
train_data <- model_data[index_train, ]
# Create test set
test_data <- model_data[-index_train, ]
```

Here, I have split the dataset into training and testing. Training data set is 70% of the whole data and testing data set is 30% of the whole data.

```
outputforest <- randomForest(as.factor(Client) ~ .,data = train_data,importance=TRUE, ntree = 200)
print(outputforest)
```

```
##
## Call:
##  randomForest(formula = as.factor(Client) ~ ., data = train_data,      importance = TRUE, ntree = 20
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 25.28%
## Confusion matrix:
##      0   1 class.error
## 0 6011 297  0.04708307
## 1 1747  31  0.98256468
```

Here, I have run the random forest model with trees as 200 on the training data.

```
y_predicted =predict(outputforest,newdata = test_data)
```

Predicting the model for testing data

```
table(predicted = y_predicted, actual = test_data$Client)
```

```
##           actual
## predicted    0    1
##         0 2583  737
##         1  131   15
```

The confusion matrix is shown here. The model is accurate for good clients but it is inaccurate for bad clients.

```
test_data %>%
  mutate(g_pred = y_predicted) %>%
  group_by(Client) %>%
  summarize(accuracy = mean(g_pred == Client))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   Client accuracy
##    <dbl>    <dbl>
## 1      0   0.952
## 2      1   0.0199
```

The accuracy for good clients and the accuracy for bad clients is calculated.

```
con_mat <- confusionMatrix(data = y_predicted, reference = as.factor(test_data$Client),
                      positive = "0")
con_mat$overall["Accuracy"]
```

```
##  Accuracy
## 0.7495672
```

```
con_mat$byClass[c("Sensitivity", "Specificity", "Prevalence")]
```

```
## Sensitivity Specificity  Prevalence
##  0.95173176  0.01994681  0.78303520
```

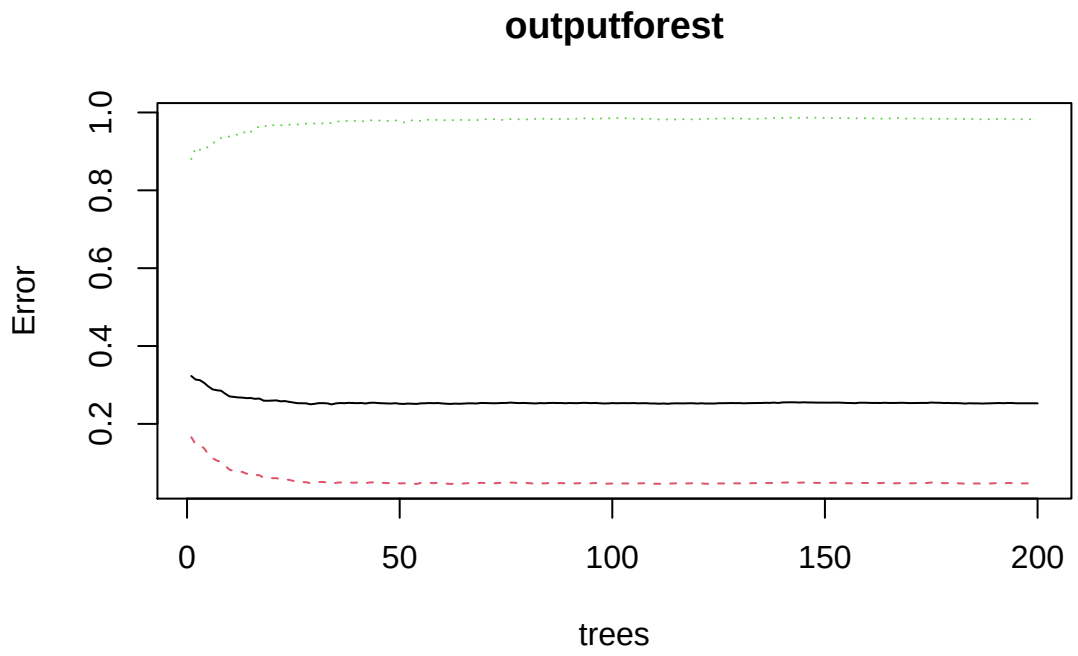
The overall accuracy, sensitivity, and specificity are calculated for our model.

```
F_meas(data = y_predicted, reference = as.factor(test_data$Client))
```

```
## [1] 0.8561485
```

The F1 score or balanced accuracy is calculated for our model

```
plot(outputforest)
```



This plot shows error for 200 trees. The error looks same throughout the graph after 20 trees.

Conclusion

Data transformation and feature selection plays an important role in modeling.To understand the distribution of the data, data analysis is most important. This model for predicting an applicant is helpful to banks but it is difficult to provide the reason for rejecting an applicant.