RISKLENS: CREDIT BORROWER RISK PREDICTION

Zeedan Shaikh   Instructor: Prof. Atanu Kumar Ghosh

Introduction

When assessing the risk of credit card default, financial institutions rely on predictive models to make informed decisions about lending. **Logistic Regression** is a traditional statistical method commonly used for binary classification problems, such as predicting whether a customer will default on a credit card payment. Its simplicity, interpretability, and relatively low computational cost make it a popular choice in the financial sector. Logistic Regression models the probability of default as a function of various customer characteristics, providing insights that are easy to understand and communicate.

Objective

- It is very important to observe we cannot really control **What Type of Customer Will Approach the Bank** so here regression aspect of logistic model is meaningless as we cannot change the predictors to get the suitable value of response.
- So we will use **Logistic Regression** for prediction Purpose and compute
  - Accuracy comparison
  - F Score comparison
  - ROC Curve visulization

Methodology

Data Collection

Dataset Link : Click Here

Variable Details

ID: A unique identifier for each customer.

Gender: The gender of the customer (e.g., Male, Female). Categorical variable.

Own_car: Indicates whether the customer owns a car (e.g., Yes, No). Binary categorical variable.

Own_property: Indicates whether the customer owns property (e.g., Yes, No). Binary categorical variable.

Work_phone: Indicates whether the customer has a work phone (e.g., Yes, No). Binary categorical variable.

Phone: Indicates whether the customer has a personal phone (e.g., Yes, No). Binary categorical variable.

Email: Indicates whether the customer has an email address (e.g., Yes, No). Binary categorical variable.

Unemployed: Indicates whether the customer is unemployed (e.g., Yes, No). Binary categorical variable.

Num_children: The number of children the customer has. Numeric variable.

Num_family: The number of family members the customer has. Numeric variable.

Account_length: The length of time the customer has held their account. Numeric variable (often in months or years).

Total_income: The total income of the customer. Numeric variable.

Age: The age of the customer. Numeric variable.

Years_employed: The number of years the customer has been employed. Numeric variable.

Income_type: The type of income the customer receives (e.g., Salary, Business, Pension). Categorical variable.

Education_type: The level of education the customer has attained (e.g., High School, Bachelor's, Master's). Categorical variable.

Family_status: The customer's family status (e.g., Single, Married, Divorced). Categorical variable.

Housing_type: The type of housing the customer lives in (e.g., Owned, Rented). Categorical variable.

Occupation_type: The customer's occupation (e.g., Professional, Clerical, Service). Categorical variable.

Target: The variable indicating the risk outcome (e.g., Defaulted, Not Defaulted). This is the dependent variable you are predicting.

Data preprocessing

Getting the required packages:

```
pacman::p_load(caret,ROCR,hnp,ggcorrplot)
```

Loading the data

```
data=read.csv("C:\\Users\\zeeda\\OneDrive\\Desktop\\dataset.csv")
data=data[,-1] #-- removing the ID column
sum(is.na(data)) #-- no missing values
```

```
[1] 0
```

Converting the categorical variables into factors

```
for (i in c(1:7,14:19)){
  data[,i]=as.factor(data[,i])
}
str(data)
```

```
'data.frame':   9709 obs. of  19 variables:
 $ Gender         : Factor w/ 2 levels "0","1": 2 2 1 1 2 2 1 1 1 2 ...
 $ Own_car        : Factor w/ 2 levels "0","1": 2 2 1 1 2 2 2 1 1 2 ...
 $ Own_property   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 1 2 2 2 ...
 $ Work_phone     : Factor w/ 2 levels "0","1": 2 1 1 1 2 1 1 1 1 1 ...
 $ Phone          : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 2 1 1 ...
 $ Email          : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
 $ Unemployed     : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ Num_children   : int  0 0 0 0 0 0 0 0 1 3 ...
 $ Num_family     : int  2 2 1 1 2 2 2 2 2 5 ...
 $ Account_length : int  15 29 4 20 5 17 25 31 44 24 ...
 $ Total_income   : num  427500 112500 270000 283500 270000 ...
 $ Age            : num  32.9 58.8 52.3 61.5 46.2 ...
 $ Years_employed : num  12.44 3.1 8.35 0 2.11 ...
 $ Income_type    : Factor w/ 5 levels "Commercial associate",..: 5 5 1 2 5 1 5 5 5 5 ...
 $ Education_type : Factor w/ 5 levels "Academic degree",..: 2 5 5 2 2 5 3 5 5 5 ...
 $ Family_status  : Factor w/ 5 levels "Civil marriage",..: 1 2 4 3 2 2 2 2 4 2 ...
 $ Housing_type   : Factor w/ 6 levels "Co-op apartment",..: 5 2 2 2 2 2 2 2 2 2 ...
 $ Occupation_type: Factor w/ 19 levels "Accountants",..: 13 18 16 13 1 9 1 9 13 9 ...
 $ Target         : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 2 1 1 ...
```

Summary of the Data

```r
summary(data)
```

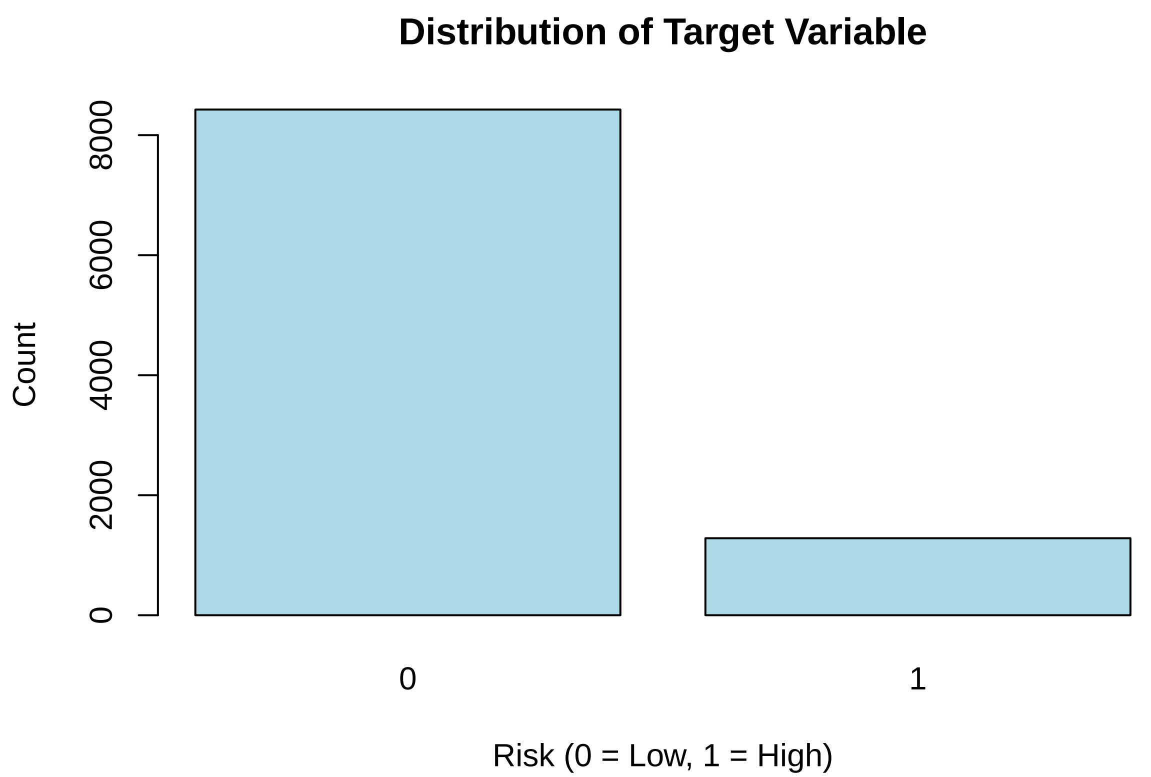
```
 Gender   Own_car  Own_property Work_phone Phone    Email    Unemployed
 0:6323   0:6139   0:3189       0:7598     0:6916   0:8859   0:8013
 1:3386   1:3570   1:6520       1:2111     1:2793   1: 850   1:1696





  Num_children       Num_family     Account_length   Total_income
 Min.   : 0.0000   Min.   : 1.000   Min.   : 0.00   Min.   :  27000
 1st Qu.: 0.0000   1st Qu.: 2.000   1st Qu.:13.00   1st Qu.: 112500
 Median : 0.0000   Median : 2.000   Median :26.00   Median : 157500
 Mean   : 0.4228   Mean   : 2.183   Mean   :27.27   Mean   : 181228
 3rd Qu.: 1.0000   3rd Qu.: 3.000   3rd Qu.:41.00   3rd Qu.: 225000
 Max.   :19.0000   Max.   :20.000   Max.   :60.00   Max.   :1575000

      Age        Years_employed                   Income_type
 Min.   :20.50   Min.   : 0.0000   Commercial associate:2312
 1st Qu.:34.06   1st Qu.: 0.9282   Pensioner           :1712
 Median :42.74   Median : 3.7619   State servant       : 722
 Mean   :43.78   Mean   : 5.6647   Student             :   3
 3rd Qu.:53.57   3rd Qu.: 8.2000   Working             :4960
 Max.   :68.86   Max.   :43.0207

                    Education_type          Family_status
 Academic degree             :   6   Civil marriage  : 836
 Higher education            :2457   Married         :6530
 Incomplete higher           : 371   Separated       : 574
 Lower secondary             : 114   Single_unmarried:1359
 Secondary_secondary special:6761   Widow           : 410


             Housing_type     Occupation_type Target
 Co-op apartment    :  34   Other      :2994   0:8426
 House_apartment    :8684   Laborers   :1724   1:1283
 Municipal apartment: 323   Sales staff: 959
 Office apartment   :  76   Core staff : 877
 Rented apartment   : 144   Managers   : 782
 With parents       : 448   Drivers    : 623
                            (Other)    :1750
```

Exploratory Data Analysis

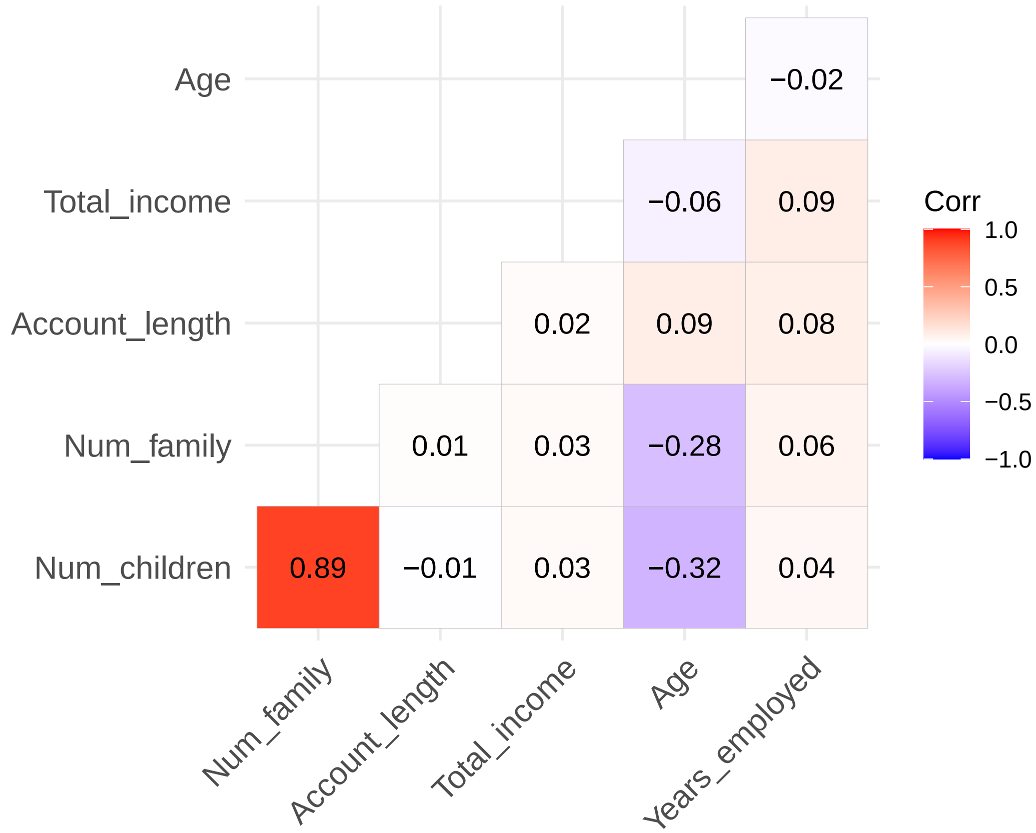
```r
categorical_cols <- c('Gender', 'Own_car', 'Own_property', 'Work_phone', 'Phone', 'Email', 'Unemployed'
                      'Income_type', 'Education_type', 'Family_status', 'Housing_type', 'Occupation_typ

numeric_cols <- c('Num_children', 'Num_family', 'Account_length', 'Total_income', 'Age', 'Years_employe
```

```r
# Distribution of the target variable (High risk vs Low risk)

barplot(table(data$Target),
```

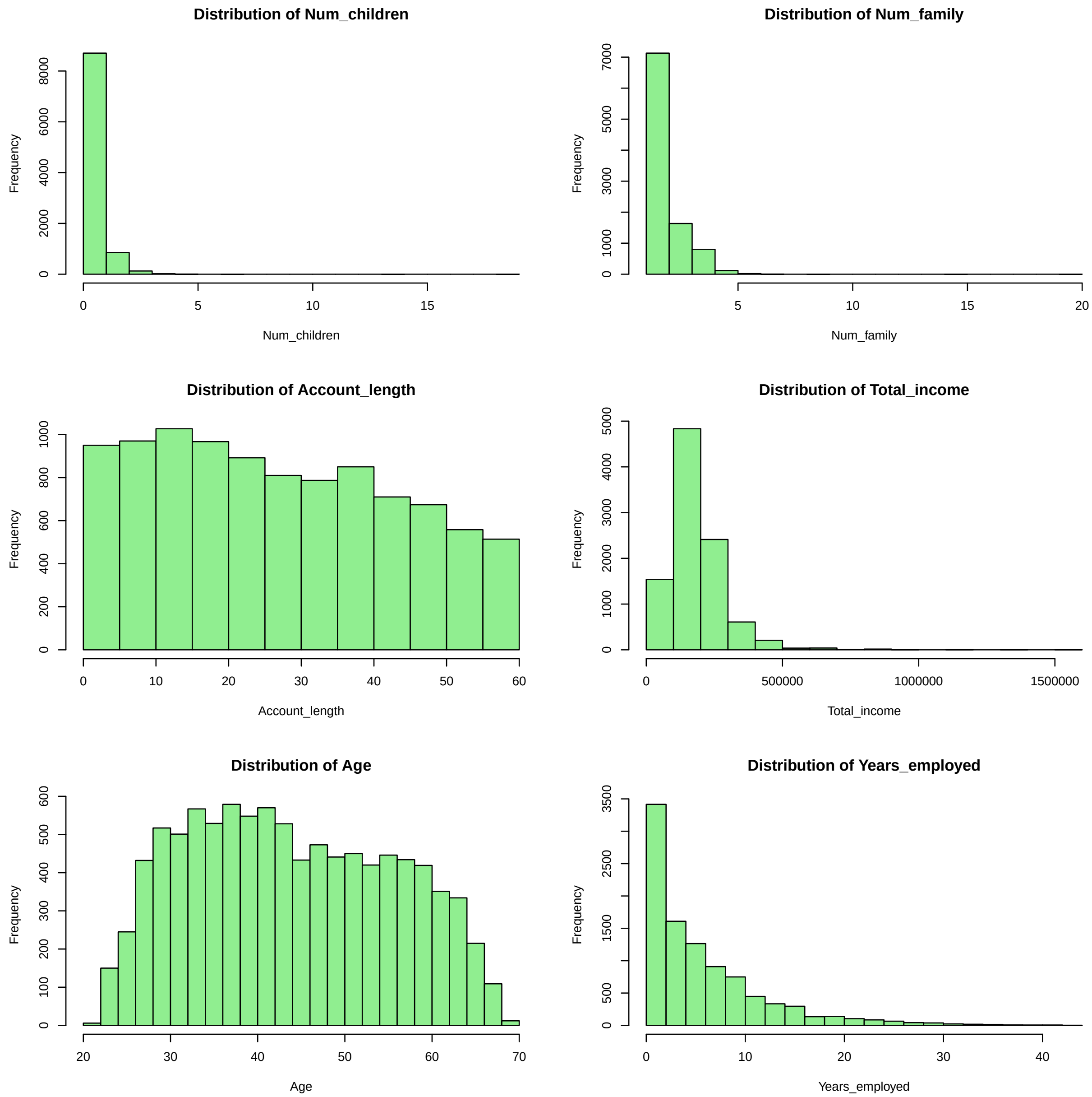
```
        main = "Distribution of Target Variable",
        xlab = "Risk (0 = Low, 1 = High)",
        ylab = "Count",
        col = "lightblue")
```



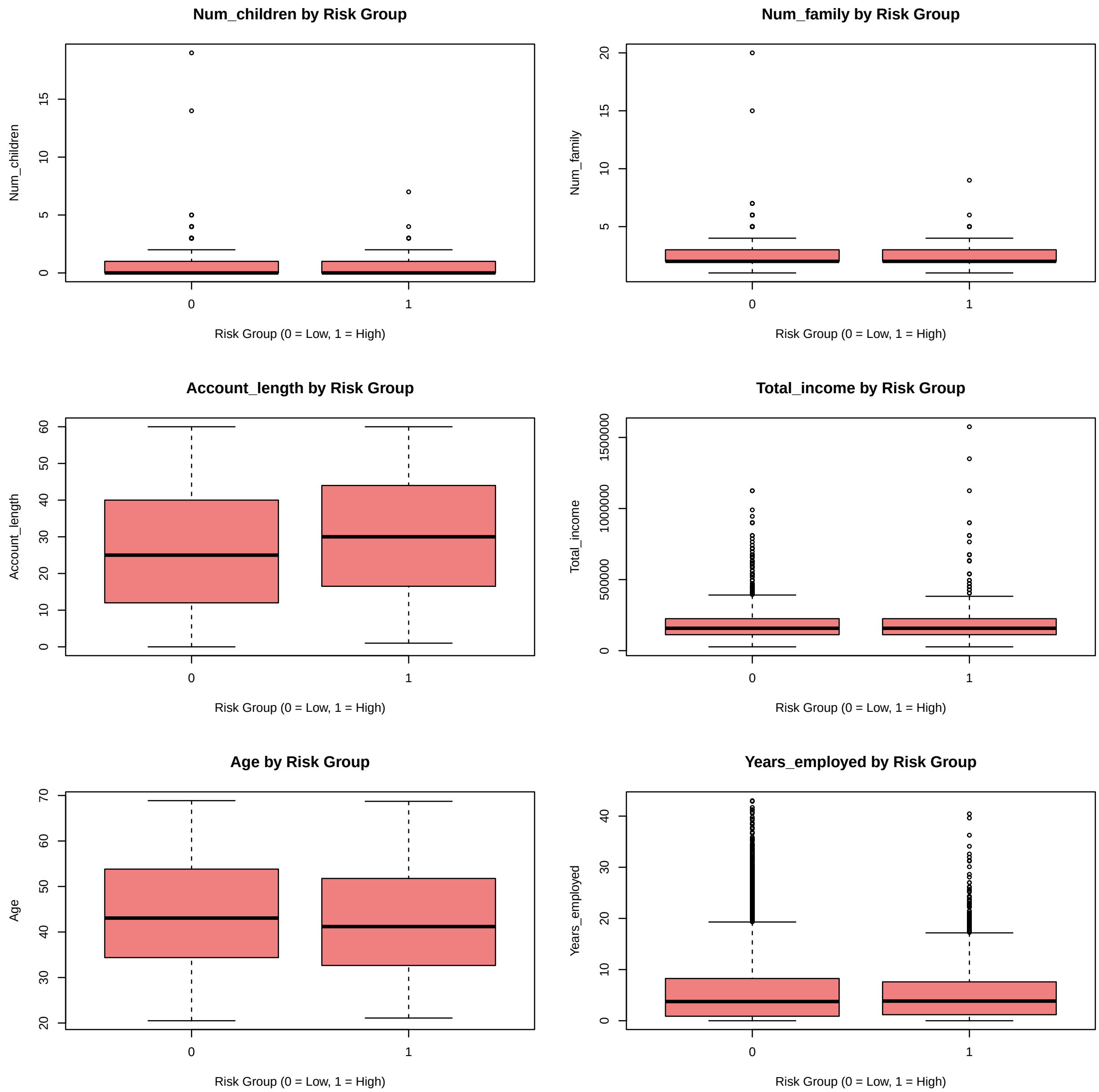

```
# Correlation between numeric variables
numeric_data <- data[, numeric_cols]
corr_matrix <- cor(numeric_data, use = "complete.obs")

# Visualize the correlation matrix
ggcorrplot(corr_matrix,method="square",type="lower",lab=TRUE)
```



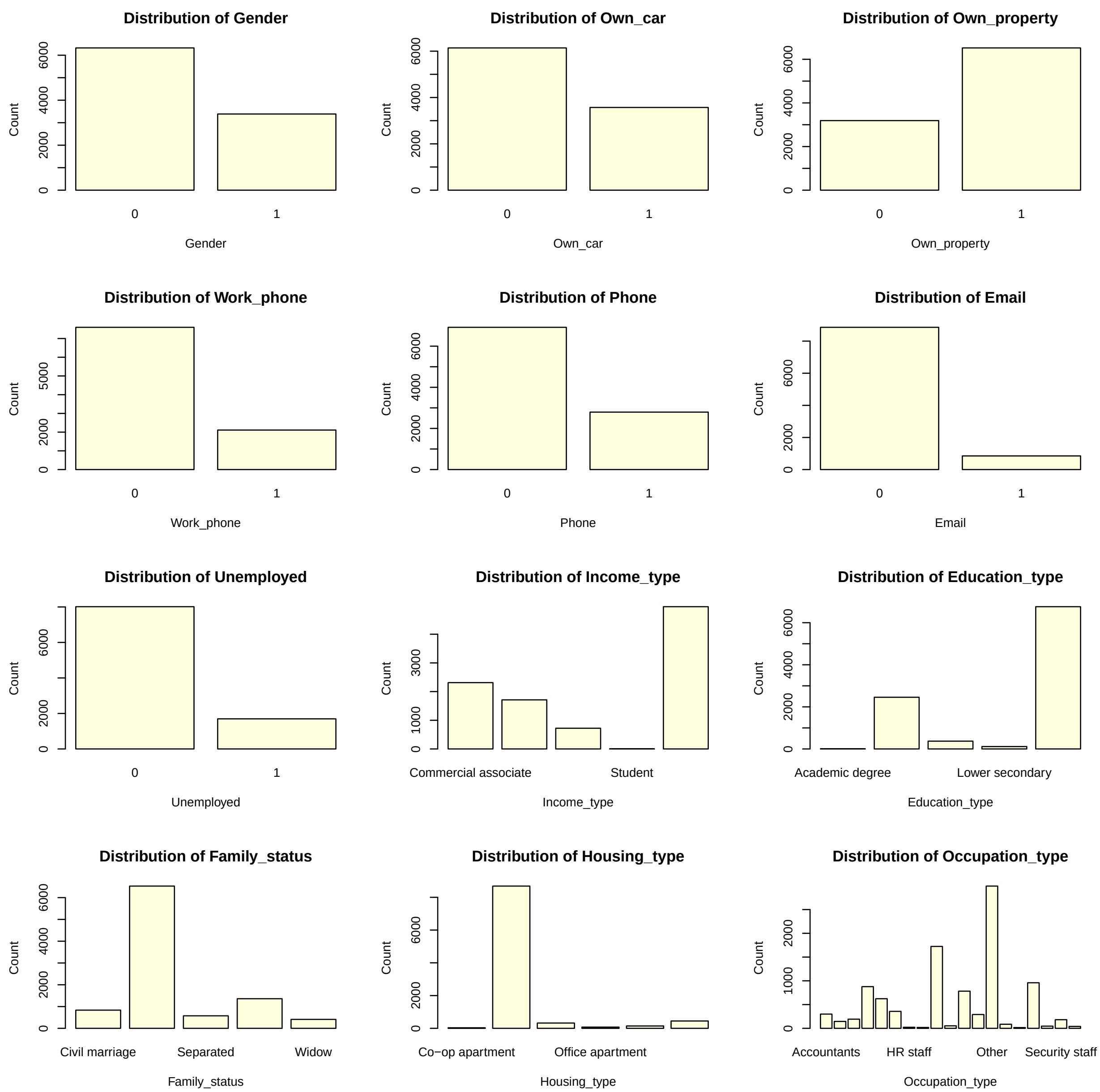

```r
# Plotting numeric variables using histograms
par(mfrow=c(3,2))
for (var in numeric_cols) {
  hist(data[[var]],
       main = paste("Distribution of", var),
       xlab = var,
       col = "lightgreen",
       breaks = 20)
}
```



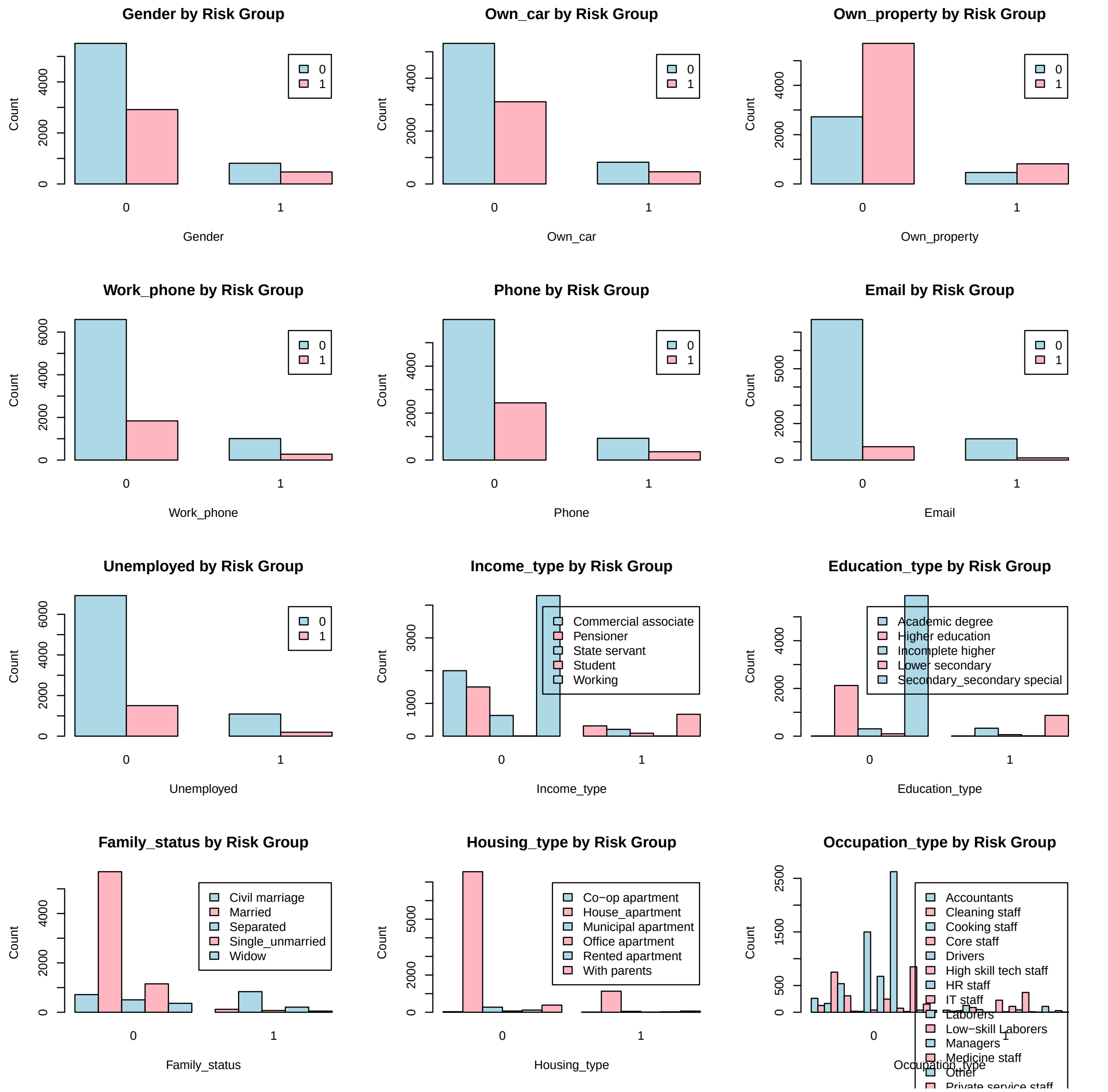

```r
par(mfrow=c(3,2))

# Boxplots to check distribution of numeric variables by Target
for (var in numeric_cols) {
  boxplot(data[[var]] ~ data$Target,
          main = paste(var, "by Risk Group"),
          xlab = "Risk Group (0 = Low, 1 = High)",
          ylab = var,
          col = "lightcoral")
}
```




```r
par(mfrow=c(4,3))
# Plotting categorical variables using barplots
for (var in categorical_cols) {
  barplot(table(data[[var]]),
          main = paste("Distribution of", var),
          xlab = var,
          ylab = "Count",
          col = "lightyellow")
}
```




```r
par(mfrow=c(4,3))
# Cross-tabulation of categorical variables with the Target variable
for (var in categorical_cols) {
  cat_table <- table(data[[var]], data$Target)
  barplot(cat_table,
          beside = TRUE,
          main = paste(var, "by Risk Group"),
          col = c("lightblue", "lightpink"),
          legend = rownames(cat_table),
          xlab = var, ylab = "Count")
}
```



Building Logistic Regression Model

Before constructing the model we will check whether the resposne vraible **Target** is balanced or not

```
table(data$Target)
```

```
   0    1 
8426 1283 
```

- Clearly it's an unbalanced problem so we will under sample the majority class and reconstruct the data again

```
index_0=which(data$Target==0)
index=sample(index_0,(8426-1283),F)
data=data[-index,]
```

```r
table(data$Target)
```

```
   0    1
1283 1283
```

- Now we will perform train-test split

```r
set.seed(42)
size=floor(nrow(data)*0.8)
split_index=sample(1:nrow(data),size,F)
train_data=data[split_index,]
test_data=data[-split_index,]
```

Building the model

```r
model=glm(Target~.,data=train_data,family="binomial")
```

summary of the model

```r
summary(model)
```

```
##
## Call:
## glm(formula = Target ~ ., family = "binomial", data = train_data)
##
## Coefficients:
##                                         Estimate Std. Error z value
## (Intercept)                           -1.661e-01  1.687e+00  -0.098
## Gender1                               -8.618e-02  1.189e-01  -0.725
## Own_car1                              -3.700e-02  1.065e-01  -0.347
## Own_property1                         -1.055e-01  1.010e-01  -1.045
## Work_phone1                           -2.279e-02  1.232e-01  -0.185
## Phone1                                -2.047e-01  1.042e-01  -1.964
## Email1                                -9.594e-02  1.679e-01  -0.572
## Unemployed1                           -1.450e+01  2.602e+02  -0.056
## Num_children                          -5.652e-01  3.495e-01  -1.617
## Num_family                             5.526e-01  3.432e-01   1.610
## Account_length                         1.537e-02  2.810e-03   5.469
## Total_income                           2.248e-08  5.172e-07   0.043
## Age                                   -1.459e-02  5.788e-03  -2.521
## Years_employed                        -8.371e-03  8.852e-03  -0.946
## Income_typePensioner                   1.452e+01  2.602e+02   0.056
## Income_typeState servant               8.940e-02  2.065e-01   0.433
## Income_typeStudent                     1.422e+01  8.827e+02   0.016
## Income_typeWorking                     2.721e-03  1.154e-01   0.024
## Education_typeHigher education        -8.390e-01  1.238e+00  -0.678
## Education_typeIncomplete higher       -6.253e-01  1.258e+00  -0.497
## Education_typeLower secondary         -1.165e+00  1.302e+00  -0.895
## Education_typeSecondary_secondary special -9.317e-01  1.236e+00  -0.754
## Family_statusMarried                  -1.891e-01  1.635e-01  -1.157
## Family_statusSeparated                 3.436e-01  4.214e-01   0.815
## Family_statusSingle_unmarried          4.719e-01  3.733e-01   1.264
## Family_statusWidow                     1.873e-01  4.365e-01   0.429
## Housing_typeHouse_apartment            3.055e-01  8.388e-01   0.364
## Housing_typeMunicipal apartment        4.421e-01  8.777e-01   0.504
```

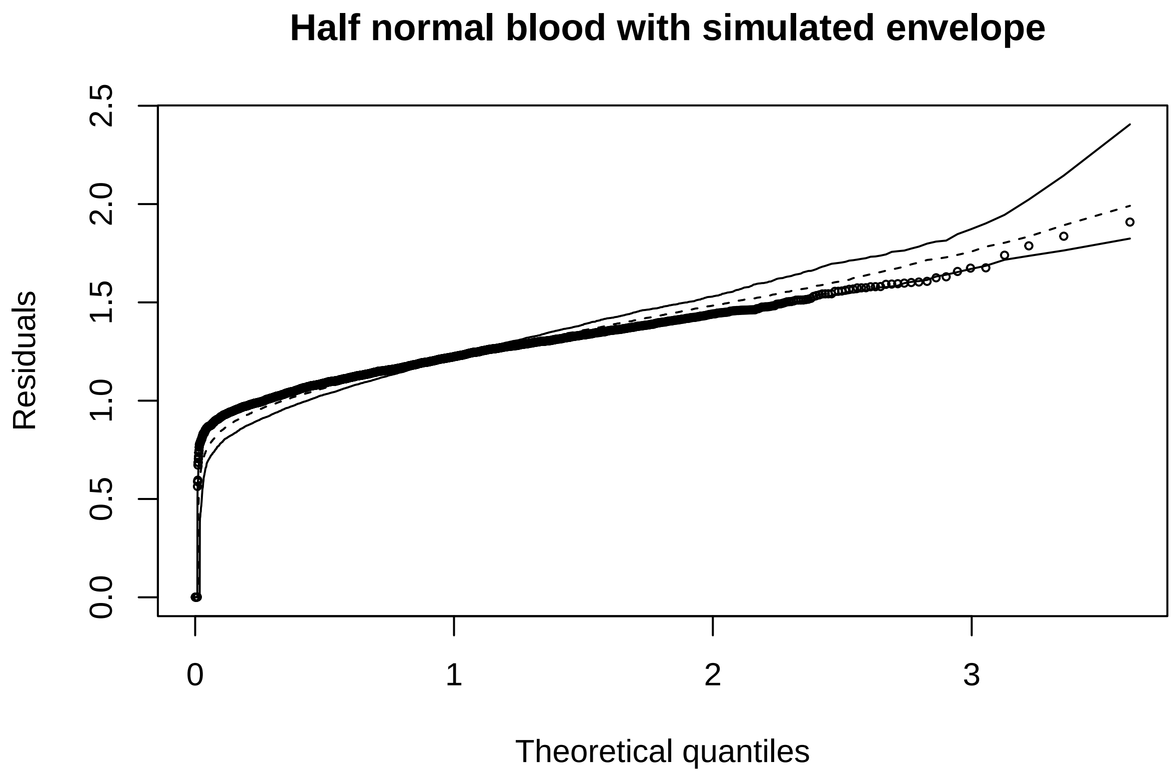
```
## Housing_typeOffice apartment              -6.260e-01  1.003e+00  -0.624
## Housing_typeRented apartment               4.900e-01  9.085e-01   0.539
## Housing_typeWith parents                   4.536e-01  8.687e-01   0.522
## Occupation_typeCleaning staff              5.002e-01  4.588e-01   1.090
## Occupation_typeCooking staff               5.967e-01  4.167e-01   1.432
## Occupation_typeCore staff                  4.600e-01  2.960e-01   1.554
## Occupation_typeDrivers                     4.437e-01  3.298e-01   1.345
## Occupation_typeHigh skill tech staff       2.563e-01  3.439e-01   0.745
## Occupation_typeHR staff                    1.182e+00  1.197e+00   0.988
## Occupation_typeIT staff                   -4.104e-01  9.698e-01  -0.423
## Occupation_typeLaborers                    1.043e-01  2.826e-01   0.369
## Occupation_typeLow-skill Laborers          5.154e-01  6.650e-01   0.775
## Occupation_typeManagers                    2.736e-01  2.991e-01   0.915
## Occupation_typeMedicine staff              3.400e-01  3.618e-01   0.940
## Occupation_typeOther                       2.104e-01  2.805e-01   0.750
## Occupation_typePrivate service staff       3.714e-01  5.586e-01   0.665
## Occupation_typeRealty agents               1.490e+01  6.240e+02   0.024
## Occupation_typeSales staff                -1.454e-01  2.921e-01  -0.498
## Occupation_typeSecretaries                 1.421e+00  1.154e+00   1.232
## Occupation_typeSecurity staff              5.364e-01  4.411e-01   1.216
## Occupation_typeWaiters_barmen staff        5.927e-01  7.334e-01   0.808
##                                           Pr(>|z|)
## (Intercept)                                 0.9215
## Gender1                                     0.4684
## Own_car1                                    0.7284
## Own_property1                               0.2961
## Work_phone1                                 0.8533
## Phone1                                      0.0495 *
## Email1                                      0.5676
## Unemployed1                                 0.9556
## Num_children                                0.1058
## Num_family                                  0.1074
## Account_length                            4.54e-08 ***
## Total_income                                0.9653
## Age                                         0.0117 *
## Years_employed                              0.3443
## Income_typePensioner                        0.9555
## Income_typeState servant                    0.6651
## Income_typeStudent                          0.9871
## Income_typeWorking                          0.9812
## Education_typeHigher education              0.4978
## Education_typeIncomplete higher             0.6191
## Education_typeLower secondary               0.3708
## Education_typeSecondary_secondary special   0.4509
## Family_statusMarried                        0.2474
## Family_statusSeparated                      0.4148
## Family_statusSingle_unmarried               0.2062
## Family_statusWidow                          0.6678
## Housing_typeHouse_apartment                 0.7158
## Housing_typeMunicipal apartment             0.6145
## Housing_typeOffice apartment                0.5327
## Housing_typeRented apartment                0.5897
## Housing_typeWith parents                    0.6016
## Occupation_typeCleaning staff               0.2756
```

```
## Occupation_typeCooking staff                0.1522
## Occupation_typeCore staff                   0.1202
## Occupation_typeDrivers                      0.1785
## Occupation_typeHigh skill tech staff        0.4561
## Occupation_typeHR staff                     0.3233
## Occupation_typeIT staff                     0.6722
## Occupation_typeLaborers                     0.7122
## Occupation_typeLow-skill Laborers           0.4383
## Occupation_typeManagers                     0.3604
## Occupation_typeMedicine staff               0.3472
## Occupation_typeOther                        0.4533
## Occupation_typePrivate service staff        0.5061
## Occupation_typeRealty agents                0.9810
## Occupation_typeSales staff                  0.6187
## Occupation_typeSecretaries                  0.2180
## Occupation_typeSecurity staff               0.2240
## Occupation_typeWaiters_barmen staff         0.4190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2844.7  on 2051  degrees of freedom
## Residual deviance: 2743.4  on 2003  degrees of freedom
## AIC: 2841.4
##
## Number of Fisher Scoring iterations: 13
```

Checking the model adequacy using hnp plot and residual vs predicted value plot

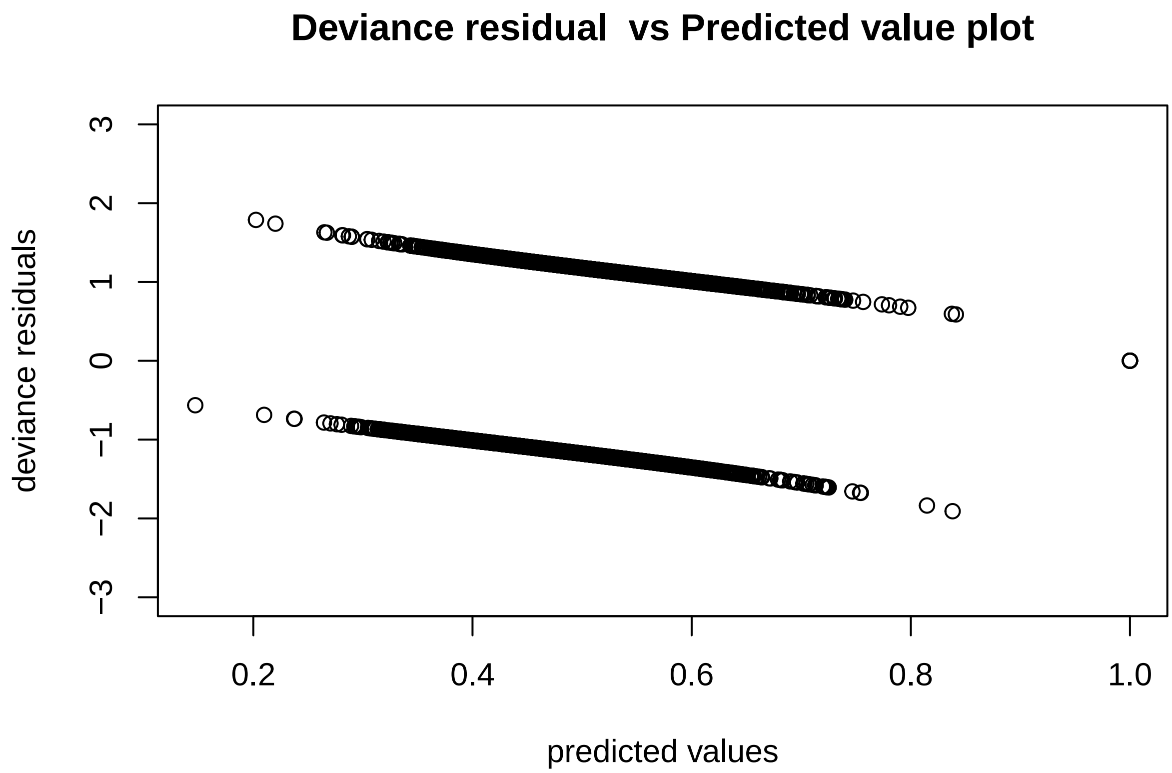
```r
set.seed(42)
hnp(model,main=" Half normal blood with simulated envelope")
```

```
Binomial model
```



- So many residuals lie outside the simulated envelop of half normal plot so clear this cannot fall under that 5 percent chance. Deviance residual vs Predicted value plot

```
dev_res=resid(model,type="deviance")
pred= fitted(model)
plot(pred,dev_res,ylim=c(-3,3),xlab="predicted values",ylab="deviance residuals",main= "Deviance residu
```



* Clearly there is a pattern here and the residuals are not randomly spread out * **We can conclude that the model is not a very good fit**

```
test_pred=predict(model,newdata = test_data[,-19])
test_pred_bin=ifelse(test_pred>0.5,1,0)
table(test_pred_bin)
```

Predicting the risk for the test set

```
test_pred_bin
  0   1
437  77
```

Using different accuracy metrices

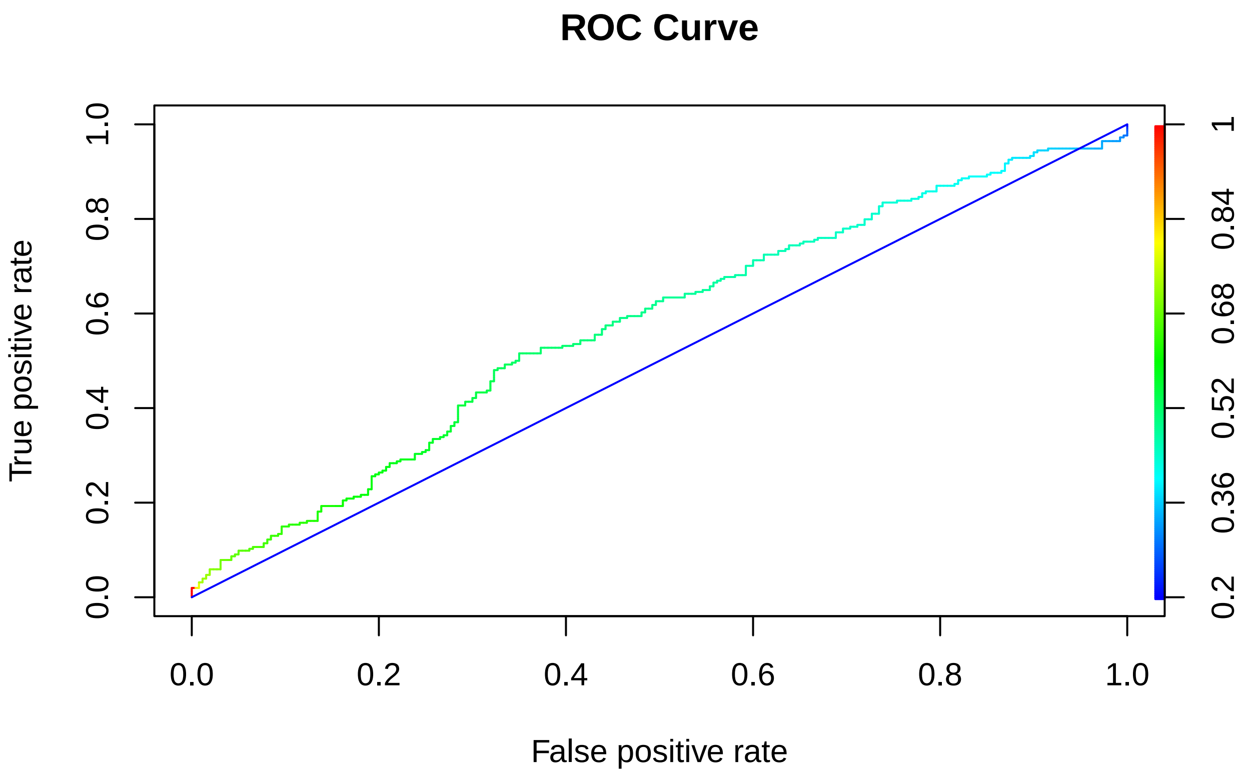
- Confusion Matrix

```
set.seed(42)
confusionMatrix(as.factor(test_pred_bin),test_data$Target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 225 212
         1  35  42
```

```
               Accuracy : 0.5195
                 95% CI : (0.4753, 0.5634)
    No Information Rate : 0.5058
    P-Value [Acc > NIR] : 0.2832

                  Kappa : 0.031

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8654
            Specificity : 0.1654
         Pos Pred Value : 0.5149
         Neg Pred Value : 0.5455
             Prevalence : 0.5058
         Detection Rate : 0.4377
   Detection Prevalence : 0.8502
      Balanced Accuracy : 0.5154

       'Positive' Class : 0
```

```r
test_pred2=predict(model,newdata = test_data[,-19],type="response")
test_pred3=prediction(test_pred2,test_data$Target)
perf=performance(test_pred3,"tpr","fpr")
plot(perf,colorize=T,main="ROC Curve")
curve(1*x,add=T,col="blue")
```



ROC Curve
* AUC ROC score

```
auc=performance(test_pred3,"auc")@y.values[[1]]
auc
```

```
## [1] 0.575
```

Summary

The project aimed to predict the risk of credit card borrower default using logistic regression. The dataset included customer demographic, financial, and employment-related features. The data was preprocessed to handle categorical and numerical variables appropriately, and an exploratory data analysis (EDA) was conducted to understand the distribution and correlations among the features. The logistic regression model was trained and evaluated to predict whether a customer is a high or low credit risk.

Key metrics such as accuracy, F1 score, and ROC-AUC were calculated to assess model performance. Additionally, model residuals were analyzed to check the adequacy of the logistic regression fit.

Conclusion

The logistic regression model showed moderate performance with an accuracy of approximately 55%. The ROC-AUC score of 0.5 suggests that the model's predictive power was slightly better than random guessing but not highly robust. Furthermore, the deviance residual analysis indicated that the model was not an ideal fit, as patterns were observed in the residuals, highlighting the need for a more complex model or additional feature engineering.

Despite the limitations, the model demonstrated an ability to identify some key predictors, such as `Account_length`, `Age`, and `Years_employed`, which significantly impacted the risk of default.

Discussion

The analysis revealed that logistic regression, while easy to interpret, struggled with the imbalanced nature of the dataset and the complexity of relationships between features. Key features like `Email`, `Age`, and `Years_employed` were found to be significant predictors, but the high residuals and low specificity suggested that the model did not capture the full complexity of the problem.

For future work, alternative models such as random forests or gradient boosting could be explored to improve predictive accuracy. Additionally, addressing class imbalance with techniques like SMOTE (Synthetic Minority Over-sampling Technique) may lead to better model performance.

The project successfully provided insights into the features influencing credit risk and set the foundation for further refinement in predictive modeling.