Churn\_Train\_NA\_Updated <- kNN(churnTrain, variable = c("account\_length","number\_vmail\_messages","total\_day\_minutes","total\_day\_calls","total\_day\_charge","total\_eve\_minutes","total\_eve\_calls","total\_eve\_charge","total\_night\_minutes","total\_night\_charge","total\_intl\_minutes","total\_intl\_calls","total\_intl\_charge","number\_customer\_service\_calls"), k=7)

library(readr)

library(tidyverse)

library(caret)

library(pROC)

library(ggcorrplot)

library(gmodels)

library(rpart)

# Loading the churn dataset

Churn\_Data <- read\_csv("Churn\_Train.csv")

# Inspecting data

head(Churn\_Data)

# Examining the dataset

glimpse(Churn\_Data)

# Summary statistics of dataset

summary(Churn\_Data)

# From glimpse we can see that, Some of the character variables can be converted into factors, So Converting character variables to factors.

Churn\_Data <- Churn\_Data %>% mutate\_if(is.character, as.factor)

# Checking NULL values in the dataset at column level.

colSums(is.na(Churn\_Data))

# imputation of missing values - median imputation technique

imputation\_model <- preProcess(Churn\_Data %>% select\_if(is.numeric),method = "medianImpute")

data <- predict(imputation\_model, Churn\_Data %>% select\_if(is.numeric))

Churn\_Data <- Churn\_Data %>% select(setdiff(names(Churn\_Data), names(data))) %>% cbind(data)

# Box plot - to detect the outliers

Churn\_Data %>% select\_if(is.numeric) %>% mutate\_all(scale) %>% gather("features","values") %>% na.omit() %>%

ggplot(aes(x = features, y = values)) +

geom\_boxplot(show.legend = FALSE) +

stat\_summary(fun = mean, geom = "point", pch = 1) + # Add average to the boxplot

scale\_y\_continuous(name = "Variable values", minor\_breaks = NULL) +

scale\_fill\_brewer(palette = "Set1") +

coord\_flip() +

theme\_minimal() +

labs(x = "Variable names") +

ggtitle(label = "Distribution of numeric variables in Churn dataset")

# Visualizing distribution of Churn categorical variable.

ggplot(Churn\_Data, aes(x=churn, y=..prop..,group = 1)) +

geom\_bar(fill="light blue") +

theme\_classic() +

geom\_text(aes(label=round(..prop..,2)),stat = "count",

position = position\_stack(vjust=0.5)) +

labs(y = 'Proportion', title = "Proportion of churn") +

scale\_x\_discrete(labels = c("No","Yes"))

# finding correlation between variables

Churn\_Data\_cor <- round(cor(Churn\_Data %>% select\_if(is.numeric)), 1)

ggcorrplot(Churn\_Data\_cor, title = "Correlation", type = "lower") +

theme(plot.title = element\_text(hjust = 0.5),

axis.text.x = element\_text(angle = 90))

# Total minutes and total charge for the day, evening, night, and international are strongly linked, we can deduce.

Churn\_Data <- Churn\_Data %>% select(-state, -churn) %>%

fastDummies::dummy\_cols(., remove\_selected\_columns = TRUE) %>% mutate(state = Churn\_Data$state, churn = Churn\_Data$churn)

# Pre-Processing of data

# Splitting dataset into training (80%) and validation (20%) sets

set.seed(12)

index <- createDataPartition(Churn\_Data$churn, p=0.8, list=FALSE)

Churn\_Data\_train\_df <- Churn\_Data[index,]

Churn\_Data\_test\_df <- Churn\_Data[-index,]

# scaling the data

scaling <- preProcess(Churn\_Data\_train\_df %>% select\_if(is.numeric), method = c("center", "scale"))

Churn\_Data\_train\_norm <- predict(scaling, Churn\_Data\_train\_df %>% select\_if(is.numeric))

Churn\_Data\_test\_norm <- predict(scaling, Churn\_Data\_test\_df %>% select\_if(is.numeric))

Churn\_Data\_train\_norm$churn <- Churn\_Data\_train\_df$churn

Churn\_Data\_test\_norm$churn <- Churn\_Data\_test\_df$churn

# Model Construction

Model\_1 <- glm(churn ~ ., data = Churn\_Data\_train\_norm , family= "binomial")

summary(Model\_1)

# Predict values using based on Model\_1.

pred\_probs <- predict(object = Model\_1,Churn\_Data\_test\_norm, type = "response")

# Assigning labels based on probability prediction

Model\_Pre\_lables <- as.factor(ifelse(pred\_probs>0.6 ,"yes","no"))

# Performance Metrics

# Confusion matrix for significant variable model.

confusionMatrix(Model\_Pre\_lables,Churn\_Data\_test\_norm$churn)

# AUC of the churn model

roc(Churn\_Data\_test\_df$churn, pred\_probs)

plot.roc(roc(Churn\_Data\_test\_df$churn, pred\_probs))