Group_Project_XL

```
install.packages(“caret”) library(caret) library(dplyr) install.packages(“tidyverse”) library(tidyverse) install.packages(“cluster”) library(cluster) install.packages(“factoextra”) library(factoextra) install.packages(“cowplot”) library(cowplot) library(ggplot2) install.packages(“tidyr”) library(tidyr) library(dplyr) install.packages(“tidyverse”) library(tidyverse) install.packages(“cluster”) library(cluster) library(readr) library(tidyr) install.packages(“devtools”) library(devtools) library(cluster) install.packages(“fpc”) library(fpc) library(readr) library(dplyr) library(ggplot2) install.packages(“ggcorrplot”) library(ggcorrplot) library(tidyr) library(fastDummies) library(caret) install.packages(“VIM”) library(VIM) library(readr) library(tidyverse) library(caret) library(pROC) library(ggcorrplot) library(gmodels) library(rpart)
```

```
Churn_Train <- read_csv(“Churn_Train.csv”) 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))

Applying the model to the Customers to Predict data file

Load the data file

load(“C:/Users/xlamo/Desktop/XanLamoreux/Group Project/Customers_To_Predict.RData”)

creating a copy to work with

```
customer_predict <- Customers_To_Predict
```

removing the state column as it is not necessary

```
customer_predict <- customer_predict %>% select(-state) %>% fastDummies::dummy_cols(., remove_selected_columns = TRUE)
```

Transformation for scaling the data (Z score transformation)

```
customer_predict <- as.data.frame(scale(customer_predict))

#predicting the model with the test data

predict_labels <- predict(object=Model_1,customer_predict,type=“response”)
```

applies the probability ratio if under 60% customer will not churn

```
Model_Pre_lables_2 <- as.factor(ifelse(predict_labels>0.6 ,“yes”,“no”))
```

adding chrun column and attaching the predictor from the model

```
Customers_To_Predict <- Customers_To_Predict %>% mutate(churn=Model_Pre_lables_2)
```

visual of the results which shows that 267 will churn

```
table(Customers_To_Predict$churn)

View(Customers_To_Predict)
```

The Customers_To_Predict file can be exported as the final results of our model