Group Project XL Files

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) install.packages("devtools") library(devtools) install.packages("fpc") library(fpc) install.packages("ggcorrplot") library(ggcorrplot) library(fastDummies) install.packages("VIM") library(VIM) library(pROC) library(ggcorrplot) library(gmodels) library(rpart) library(corrplot) library(ROCR) library(ISLR) library(dplyr) library(caret) library(VIM) library(tidyr) library(pROC) library(ggcorrplot) library(ggplot2)

Churn_Train <- read_csv("Churn_Train.csv") churnTrain <- read_csv("Churn_Train.csv")

Examining the dataset

head(churnTrain) summary(churnTrain) glimpse(churnTrain)

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

#character variables to factors. churnTrain <- churnTrain%>% mutate_if(is.character, as.factor)

Checking NULL values in the dataset at column level.

colSums(is.na(churnTrain))

#Review fields with NAs #Create a dataframe where we are seeing any record that has a null value churnTrain_NA <- churnTrain[!complete.cases(churnTrain),]

#Working to populate those values that are null using K-Nearest Neighbor to see how it affects the data. All of the #fields in the vector are those with records that have NA's. Churn_Train_NA_Updated <- kNN(churnTrain, variable = c("account_length","number_vmail_messages","total_day_minutes","total_day_calls","total_
k=7)

#removing the fields created by the K-Nearest Neighbor Method Churn_Train_NA_Updated_Final <- Churn_Train_NA_Updated %>% select(-one_of("total_intl_calls_imp","total_intl_minutes_imp","total_eve_minutes_

#Replaced fields that had negative values with the absolute value of those records Churn_Train_NA_Updated_Final$account_$
<- abs(Churn_Train_NA_Updated_Final$account_length[Churn_Train_NA_Updated_Final$account_length<0])

Churn_Train_NA_Updated_Final$number_vmail_messages[Churn_Train_NA_Updated_Final$number_vmail_messages<0]
<- abs(Churn_Train_NA_Updated_Final$number_vmail_messages[Churn_Train_NA_Updated_Final$number_vmail_messages<

#Process for selecting only numeric variables and removing any non-complete (any null values in any field) records #and using corrplot to visualize any possible correlations of numerical values churn_numerical <- Churn_Train_NA_Updated_Final %>% select(-one_of("state","area_code","international_plan","voice_mail_plan","chu corrplot(cor(churn_numerical),method="square", col=colorRampPalette(c("purple","orange"))(200))

#Also wanted to look to see what data have outliers or have a significant variation in value. Created boxplot for all of #the numerical categories.

Churn_Train_NA_Updated_Final%>% 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")

#creation of Churn Proportion Chart ggplot(Churn_Train_NA_Updated_Final, 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"))

#Reviewing the frequency tables for categorical variables table(Churn_Train_NA_Updated_Final$churn)table(Churn_{T}rain_{}$
Churn_Train_NA_Updated_Final$state)table(Churn_{T}rain_{N}A_{U}pdated_{F}inal$churn, Churn_Train_NA_Updated_Final$inte$
Churn_Train_NA_Updated_Final$voice_{m}ail_{p}lan)table(Churn_{T}rain_{N}A_{U}pdated_{F}inal$churn, Churn_Train_NA_Updated_

Changing any categorical variables (outside of state and churn to binary - 0 or 1).

Churn_Data <- Churn_Train_NA_Updated_Final%>% select(-state, -churn) %>% fastDummies::dummy_cols(.) %>% mutate(state = Churn_Train_NA_Updated_Final$state, churn = Churn_{T}rain_{N}A_{U}pdated_{F}ina$

#Removing fields that will not be used for modeling purposes. Churn_Data <- Churn_Data %>% select(-one_of("area_code","international_plan","state","voice_mail_plan","area_code_area_code_510","international_plan_no

Pre-Processing of data for model

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

set.seed(123) index <- createDataPartition(Churn_Data$churn, p=0.8, list=FALSE) Churn_Data_train_df <- Churn_Data[index,] Churn_Data_test_df <- Churn_Data[-index,]

Model Construction

Model_1 <- glm(churn ~ ., data = Churn_Data_train_df , family= "binomial") summary(Model_1)

Predicting values using based on Model_1.

pred_probs <- predict(object = Model_1,Churn_Data_test_df , 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_df $churn)

#True positive Rate vs. False Positive Rate pred <- prediction(pred_probs, Churn_Data_test_df$churn) roc.perf = performance(pred, measure = "tpr", x.measure = "fpr") plot(roc.perf) abline(a=0, b= 1)

AUC of the churn model

roc(Churn_Data_test_df$churn, pred_probs) auc.perf = performance(pred, measure = "auc") auc.perf@y.values

accuracy vs. cutoff value

acc.perf = performance(pred, measure = "acc") plot(acc.perf)

Prediction's File

Applying the model to the Customers to Predict data file

Load the data file

the below address is specific to me and where I housed the 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 — using the Model_1 file created earlier 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

table(Customers_To_Predict$churn)

View(Customers_To_Predict)