Group 8 - Team Members	Contribution
Alex Haffner	Participated in Group discussion. Reviewed and analyzed Code file. Completed Overview of Data portion of report. Reviewed Presentation and contributed on modeling strategy slide.
Erica Winters	Participated in group discussion, reviewed and completed the insights and conclusions. Prepared the presentation and worked with group members to create the voiceover.
Chris Hargis	Project lead. Initiated the strategy for the project. Developed the foundation for the code file and the project report. Reviewed and helped develop the presentation slides.

Project Goal

The objective of our project is to create a model that can be used by ABC Wireless Inc. to predict the probability of churn per customer along with identifying the important variables related to churn. We decided to use a logistics regression model so that each customer can be assigned a probability of churn. ABC Wireless Inc will then be able to implement this model to identify specific customers who are likely to churn and provide incentives to improve customer retention along with other business decisions as needed.

Overview of Data

The churn data included 19 predicting variables such as account length, area codes, total daily minutes, calls, charges etc. We began our data exploration with two-way frequency tables to see if there were possible relationships among the categorical data. We found there was a relationship between our variables and the target variable "Churn."

```
{r}
table(df$churn)
```

```
  no  yes
2850  483
```

(Report continued below)

We also found that there were several pieces of data that were not available and so we removed those from the rest of our explorations and tested the relationship with another two-way frequency table. The data removed included the number of voicemails, account length, total evening minutes and total international calls.

```
{r}
summary(df)
```

```
    state           account_length     area_code          international_plan
 Length:3333        Min.   :-209.00   Length:3333        Length:3333
 Class :character   1st Qu.:  72.00   Class :character   Class :character
 Mode  :character   Median : 100.00   Mode  :character   Mode  :character
                    Mean   :  97.32
                    3rd Qu.: 127.00
                    Max.   : 243.00
                    NA's   :501
 voice_mail_plan    number_vmail_messages total_day_minutes total_day_calls
 Length:3333        Min.   :-10.000       Min.   :   0.0    Min.   :  0.0
 Class :character   1st Qu.:  0.000       1st Qu.: 149.3    1st Qu.: 87.0
 Mode  :character   Median :  0.000       Median : 190.5    Median :101.0
                    Mean   :  7.333       Mean   : 418.9    Mean   :100.3
                    3rd Qu.: 16.000       3rd Qu.: 237.8    3rd Qu.:114.0
                    Max.   : 51.000       Max.   :2185.1    Max.   :165.0
                    NA's   :200           NA's   :200       NA's   :200
 total_day_charge total_eve_minutes total_eve_calls total_eve_charge
 Min.   : 0.00    Min.   :   0.0    Min.   :  0.0   Min.   : 0.00
 1st Qu.:24.45    1st Qu.: 170.5    1st Qu.: 87.0   1st Qu.:14.14
 Median :30.65    Median : 209.9    Median :100.0   Median :17.09
 Mean   :30.63    Mean   : 324.3    Mean   :100.1   Mean   :17.08
 3rd Qu.:36.84    3rd Qu.: 257.6    3rd Qu.:114.0   3rd Qu.:20.00
 Max.   :59.64    Max.   :1244.2    Max.   :170.0   Max.   :30.91
 NA's   :200      NA's   :301       NA's   :200     NA's   :200
 total_night_minutes total_night_calls total_night_charge total_intl_minutes
 Min.   : 23.2       Min.   : 33.0     Min.   : 1.040     Min.   : 0.00
 1st Qu.:167.3       1st Qu.: 87.0     1st Qu.: 7.530     1st Qu.: 8.50
 Median :201.4       Median :100.0     Median : 9.060     Median :10.30
 Mean   :201.2       Mean   :100.1     Mean   : 9.054     Mean   :10.23
 3rd Qu.:235.3       3rd Qu.:113.0     3rd Qu.:10.590     3rd Qu.:12.10
 Max.   :395.0       Max.   :175.0     Max.   :17.770     Max.   :20.00
 NA's   :200                           NA's   :200        NA's   :200
 total_intl_calls total_intl_charge number_customer_service_calls
 Min.   : 0.00    Min.   :0.000     Min.   :0.000
 1st Qu.: 3.00    1st Qu.:2.300     1st Qu.:1.000
 Median : 4.00    Median :2.780     Median :1.000
 Mean   : 4.47    Mean   :2.762     Mean   :1.561
 3rd Qu.: 6.00    3rd Qu.:3.270     3rd Qu.:2.000
 Max.   :20.00    Max.   :5.400     Max.   :9.000
 NA's   :301      NA's   :200       NA's   :200
    churn
 Length:3333
 Class :character
 Mode  :character
```

```{r}
library(tidyr)

df = df[!is.na(df$number_vmail_messages),]
df = df[!is.na(df$account_length),]
df = df[!is.na(df$total_eve_minutes),]
df = df[!is.na(df$total_intl_calls),]

table(df$churn)
```

```
  no  yes
2250  380
```

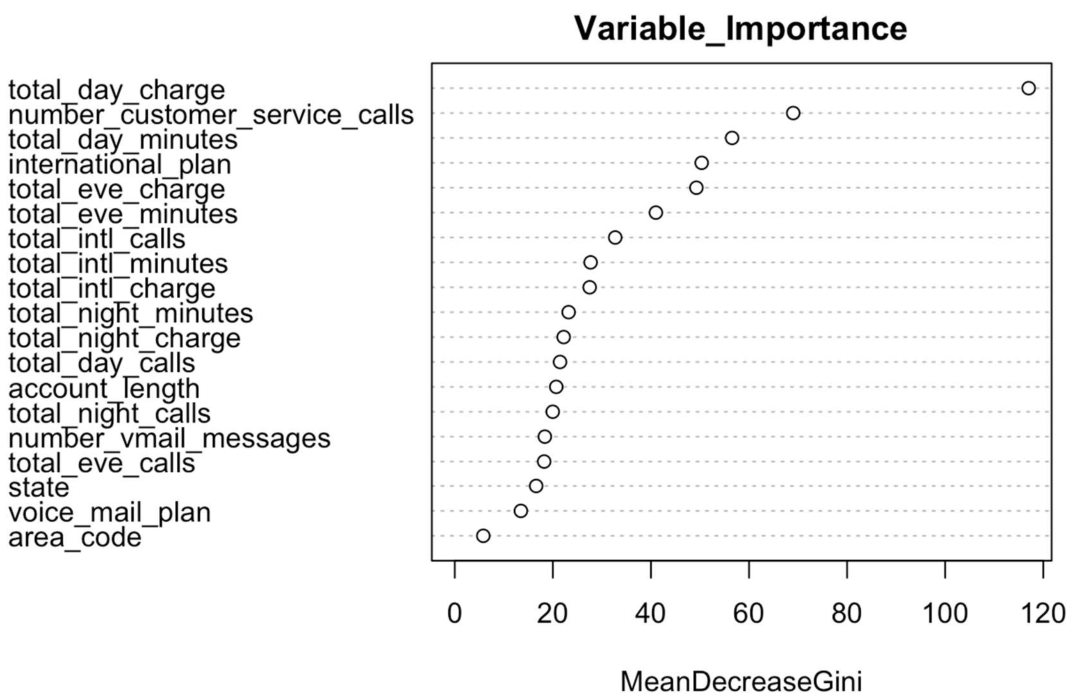
We utilized the RandomForest package to analyze the impact of variables on customer churn. RandomForest is easy to interpret, handles both categorical and continuous data efficiently and is not as sensitive to outliers. We concluded that the total daily charge ranked the highest in Gini score average with a score of 119. The next closest scores were the number of customer service calls and total daily minutes, with scores of 68 and 54.

```{r}
#install.packages("randomForest")
library(randomForest)
df$churn <- as.factor(df$churn)
variable_importance = randomForest(churn ~ 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_calls +
total_night_charge + total_intl_minutes + total_intl_calls + total_intl_charge +
number_customer_service_calls + international_plan + voice_mail_plan + state + area_code,
data = df)
randomForest::importance(variable_importance)
```

```
                              MeanDecreaseGini
account_length                       21.178655
number_vmail_messages                19.257073
total_day_minutes                    54.585464
total_day_calls                      20.810481
total_day_charge                    119.411494
total_eve_minutes                    40.886238
total_eve_calls                      18.068917
total_eve_charge                     48.759514
total_night_minutes                  22.115502
total_night_calls                    19.686517
total_night_charge                   22.038999
total_intl_minutes                   28.266959
total_intl_calls                     31.051902
total_intl_charge                    26.513217
number_customer_service_calls        68.566232
international_plan                   52.335430
voice_mail_plan                      13.149432
state                                16.521739
area_code                             5.642651
```

Our results from this phase were visualized in the Variable Importance Plot.

```
{r}
varImpPlot(Variable_Importance)
```



<u>Modeling Strategy</u>

Our strategy is to use a logistic regression model that is easy to build and maintain. The purpose of this model will be to predict and classify customers that will churn or will not churn. This model can use a single, several or all the variables from a given data set to return a binary output. This strategy will work well because we can determine the important criteria for customer churn for ABC Wireless Inc. and can provide us insight on which areas of the business to focus attention.

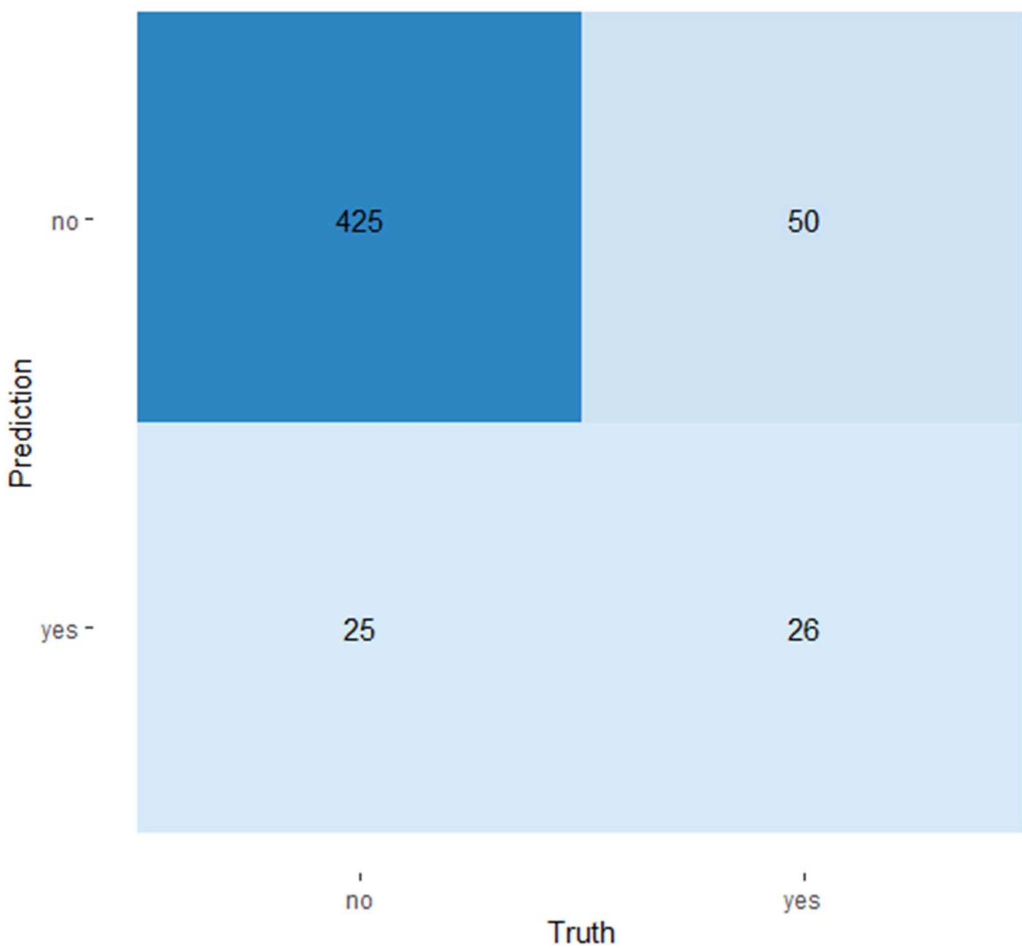
Here we built the model using only the total_day_charge variable as this would yield the highest accuracy scores and was rated the highest importance from the random forest chart above. We tested the model using a few of the top variables but after analyzing the results only using total_day_charge resulted in the best scores.

```
```{r}
model = glm(churn ~ total_day_charge, data = training_data, family = "binomial")
```
```

We used a cutoff of value of 0.27. Through testing several different cutoff values this would return a higher confusion matrix efficiency. Given the ratio of yes's to no's in our dataset we started with 0.17 and expanded from there analyzing the different results.

```
```{r}
cutoff_test = c(.17, .20, .23, .25, .27, .3, .33)
for (i in cutoff_test){
  training_data$Predict = as.factor(ifelse(model$fitted.values > i, "yes", "no"))
  table1 = table(training_data$churn, training_data$Predict)
  efficiency = round(sum(diag(table1))/sum(table1), 2)
  print(paste0("Training CM efficiency with the cutoff as ", i, " is equal to: ", efficiency))
}
```
```

Model's Performance Analysis

The confusion matrix (CM) shows how well our model performs to the data given. It's not just about maximizing the efficiency of the CM. The value in the CM is to see how many of the yes's (churns) we can capture without having the model predict yes of customers that wouldn't have left. This would result in wasted incentives. With our accurate model ABC Wireless Inc can accurately incentivize customers that are likely to churn.

Here we wanted to analyze how the model performed with different training/test split percentages. We determined that 90% was a good split after viewing the confusion matrix for each split.

````
```{r}
training_percentages = c(.65, .70, .75, .80, .85, .90)
for (i in training_percentages){
  set.seed(9)
  training_data_1 = sample(1:nrow(no_churn_df), i*nrow(no_churn_df))
  training_data_2 = sample(1:nrow(yes_churn_df), i*nrow(yes_churn_df))
  training_1 = no_churn_df[training_data_1, ]
  training_2 = yes_churn_df[training_data_2, ]
  training_data = rbind(training_1, training_2)
  test_1 = no_churn_df[-training_data_1, ]
  test_2 = yes_churn_df[-training_data_2, ]
  test_data = rbind(test_1, test_2)
  model = glm(churn ~ total_day_charge, data = test_data, family = "binomial")
  test_data$Predicted = as.factor(ifelse(model$fitted.values > 0.27, "yes", "no"))
  table2 = table(test_data$churn, test_data$Predicted)
  efficiency = round(sum(diag(table2))/sum(table2), 2)
  print(paste0("Test CM efficiency for a training percent of: ", i , "% is: " , efficiency))
}
```
````

(Report continued below)

The confusion matrix for the test data shows us how well our model performs on unseen data. Here we wanted to maximize correct predictions (upper left & lower right) and minimize the incorrect predictions (bottom left & upper right). When altering the model's parameters it is important to continuously view this confusion matrix as it will show you how your model is performing.

We noticed there can be a trade off with this model. We can improve the model to capture more of the churning customers however it will then predict more yes's in the future resulting in wasted incentives. On the other end we can improve overall accuracy resulting in more "No" predictions. This can result in the company not capturing as many of the churning customers as they would like. After fine tuning the parameters we decided that the below confusion matrix could be a happy medium.



Insights and Conclusion

The model that we created to be used to predict the probability of customer churn. We have ensured that our model is not only the most accurate it can be but also that it works efficiently and effectively. The goal here was to provide the most amount of accurate predictions to help minimize wasted incentives and maximize customer retention. When using our model, there is an 86% chance the model will return a correct prediction. Most of the results are a correct no, which means incentives will not be wasted on those customers. There is a greater chance that when a yes is predicted, that it will be a correct prediction. Overall the implementation of this model will result in an increase in the bottom line due to the model's ability to accurately predict customer churn along with indirectly providing insights on the most important factors leading up to churn.