CHURN PREDICTION FOR A TELECOMMUNICATION COMPANY

For accessing the video of our demo, click here.

GROUP MEMBERS AND ROLES

Nimra Aslam (14815) – Understanding the problem, the dataset and determining the data type of each feature of the dataset.

Mohammad Ahsan Siddiqui (18076) – Feature engineering, data visualization, and use of statistical methods to find out relationships between set of features inside the data and extract any useful information from the existing features by using them to create new features.

Rohail Naushad (18039) – Label Encoding and Hot One Encoding, building of machine learning models, and recording of performance scores.

Problem Description: This includes the background of the company, the motivation of the study (challenges faced by the organization), how they are dealing it currently,

Problem: To predict whether a customer would change telecommunications provider, something known as “churning”.

Data Description: Size of the dataset and description of the columns, Quality of the data (missing values proportion, constant columns, etc.)

The training dataset contains 3333 samples. Each sample contains 20 features and 1 boolean variable "churn" which indicates the class of the sample. The 20 input features and 1 target variable are:

1. "**state**", string. 2-letter code of the US state of customer residence
2. "**account_length**", numerical. Number of months the customer has been with the current telco provider
3. "**area_code**", string="area_code_AAA" where AAA = 3 digit area code.
4. “**phone number**”, the phone number of the customer.
5. "**international_plan**", (yes/no). The customer has international plan.
6. "**voice_mail_plan**", (yes/no). The customer has voice mail plan.
7. "**number_vmail_messages**", numerical. Number of voice-mail messages.
8. "**total_day_minutes**", numerical. Total minutes of day calls.
9. "**total_day_calls**", numerical. Total minutes of day calls.
10. "**total_day_charge**", numerical. Total charge of day calls.
11. "**total_eve_minutes**", numerical. Total minutes of evening calls.
12. "**total_eve_calls**", numerical. Total number of evening calls.
13. "**total_eve_charge**", numerical. Total charge of evening calls.
14. "**total_night_minutes**", numerical. Total minutes of night calls.
15. "**total_night_calls**", numerical. Total number of night calls.
16. "**total_night_charge**", numerical. Total charge of night calls.
17. "**total_intl_minutes**", numerical. Total minutes of international calls.
18. "**total_intl_calls**", numerical. Total number of international calls.
19. "**total_intl_charge**", numerical. Total charge of international calls
20. "**number_customer_service_calls**", numerical. Number of calls to customer service
21. "**churn**", (yes/no). Customer churn - target variable.

Data Pre-Processing: What steps did you take to prepare the data?

We took the following steps to prepare our data:

1. Applied Hot One Encoding and Label Encoding to categorical features.
2. Applied statistical methods to determine which features do not help in determining the class label.
3. Removed features with low variance.

4. Engineered more features from the existing ones and applied statistical tests to find out if they help in determining the class label.

Hot One Encoding and Label Encoding

The following features were hot one encoding:

International plan (Yes = 1, No = 0)

Voice mail plan (Yes = 1, No = 0)

Label encoding was applied to the feature "**State**" because it had too many labels values and applying hot one encoding would have added too many columns to our dataset.

Statistical Methods

Our dataset comprised features that were either categorical or numeric. Our goal was to find out how significant each feature is in determining the class label. However, these statistical methods could also be used to confirm any likely relationship between any two features that a visual representation of data distribution might imply.

Feature Engineering

In this step, we created new features using the existing ones. For example, the values of total day minutes, total evening minutes, and total night minutes could be summed to create a new feature named "total minutes" for each customer.

Feature Selection – Removing Features With Low Variance

VarianceThreshold is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

Model Building + Evaluation: Which models did you try, which turned out to be the best (based on which metric), etc. How long did it take for a single evaluation? The description of the machine(s) on which you ran your experiments. Which software/libraries did you use, etc.?

So there were several models that we incorporated in our project. The different models used were Random Forest Classifier, Random Forest with entropy, Decision Tree Classifier, Multinomial NB Classifier and Gradient Boosted Classifier.

Initially, we started with the Random Forest Classifier. Since we were using cross validation method for our data, we started with different values for n_estimator. The initial value for cv was set at 5 and the scoring was done using 'f1'. The table for the values and different results is shown below.

N_ESTIMATORS	SCORE
1	0.7036341765856597
5	0.838549101383642
10	0.8550979964262542
30	0.8870577243682278

50	0.8848536544424611
100	0.8880692186211014
200	0.889346158412844

Then we started with the Random Forest Classifier but this time the value for cv was changed to 10. This was done so that the score is calculated from a large pool of values. Since we were using cross validation method for our data, we started with different values for n_estimator. The scoring was still done using ‘f1’. The table for the values and different results is shown below.

N_ESTIMATORS	SCORE
1	0.7210545320320055
5	0.8551932306792569
10	0.8715143926538229
30	0.8842646696028295
50	0.8892741453117408
100	0.8902444609524224
200	0.8889905424571245

Then we moved onto the Random Forest Classifier but this time we set the criterion as entropy. Since we were using cross validation method for our data, we started with different values for n_estimator. The initial value for cv was set at 5 and the scoring was done using ‘f1’. The table for the values and different results is shown below.

N_ESTIMATORS	SCORE
1	0.6714075494113836
5	0.8253976728166738
10	0.8612125124977021
30	0.8762384109817034
50	0.8876044618932146
100	0.8865545373870087
200	0.8892956165325824

Then we started with the Random Forest Classifier but this time the value for cv was changed to 10 and the criterion was set to ‘entropy’. This was done so that the score is calculated from a large pool of values. Since we were using cross validation method for our data, we started with different values for n_estimator. The scoring was still done using ‘f1’. The table for the values and different results is shown below.

N_ESTIMATORS	SCORE
1	0.6573320060396121
5	0.8331451536997811
10	0.8395525253468655
30	0.876765538706423
50	0.8814691859258434
100	0.88755209377407

200	0.8878855340329033

Then we started with the Decision Tree Classifier. Since we were using cross validation method for our data, we started with different values for random_state. The initial value for cv was set at 5 and the scoring was done using ‘f1’. The table for the values and different results is shown below.

RANDOM_STATE	**SCORE**
1	0.7852595008087541
5	0.7993332812527125
10	0.7955213563516949
30	0.7935349795109639
50	0.7857746004853797
100	0.7916604922959655
200	0.7903554354416187

Then we started with the Decision Tree Classifier but this time the value for cv was changed to 10. This was done so that the score is calculated from a large pool of values. Since we were using cross validation method for our data, we started with different values for random_state. The scoring was still done using ‘f1’. The table for the values and different results is shown below.

RANDOM_STATE	**SCORE**
1	0.8128858838639419
5	0.8110170304082012
10	0.8146686103392948
30	0.8212910157714338
50	0.8016339369312933
100	0.8060080878827784
200	0.8174075572003623

Then we moved onto the Decision Tree Classifier but this time we set the criterion as entropy. Since we were using cross validation method for our data, we started with different values for random_state. The initial value for cv was set at 5 and the scoring was done using ‘f1’. The table for the values and different results is shown below.

RANDOM_STATE	**SCORE**
1	0.7954354841849816
5	0.8238836705553234
10	0.8214267933962704
30	0.8287840762248055
50	0.8288251304333807
100	0.8258702752765006
200	0.8237733837322848

Then we started with the Random Forest Classifier but this time the value for cv was changed to 10 and the criterion was set to ‘entropy’. This was done so that the score is calculated from a large pool of values. Since we were using cross validation method for our data, we started with different values for random_state. The scoring was still done using ‘f1’. The table for the values and different results is shown below.

RANDOM_STATE	**SCORE**
1	0.810505908177449
5	0.81524355278729731
10	0.819865697369308
30	0.8149984147940484
50	0.8155201011294084
100	0.8127735579429647
200	0.8197314833240774

After taking out values from the Random Forest and Decision Tree Classifier, we moved onto the method of LinearSVC since it is closely linked to feature engineering. The tolerance level was set constant at 1e-5. We started to take out values for the score at different values for random_state.

RANDOM_STATE	**SCORE**
1	0.21447049319143385
5	0.05886497460012756
10	0.08352288646919499
30	0.06864601377754502
50	0.1310732782363228
100	0.05780423280423279
200	0.027809985850010106

After this we moved onto our final model of Naïve Bayes. We started initially with no parameters and at cv = 5 got a value of 0.30896983169471615. We changed the value of cv to 10 and got the value of 0.30923415861522924.

After completing these, we moved onto our last model which was the Gradient Boosted Classifier. We Started with different values of n_estimators. The rest of the settings were set at default. The cv was set at 5. The table below represents the changes in the values.

N_ESTIMATORS	**SCORE**
1	0.0
5	0.7402129231209809
10	0.7534937744758866
30	0.882308716146091
50	0.8932833762970711
100	0.9008239589897193
200	0.898406092241606

Then we changed the value of cv to 10 and found out different values with different n_estimators. These values are represented below.

N_ESTIMATORS	**SCORE**
1	0.0
5	0.7269009169752786
10	0.7478849037419679
30	0.8872102376731243
50	0.886540362057524
100	0.884454003599618
200	0.8871340837921654

We then started to tweak with the different parameters of gradient boosted and now we set the learning_rate to 0.1 or 10%. The value for cv was set to 5. And the rest of the values were set to default. The value for n_estimators was changed to see the fluctuation in the value for scores. The results are shown below in the table.

N_ESTIMATORS	**SCORE**
1	0.0
5	0.7402129231209809
10	0.7534937744758866
30	0.882308716146091
50	0.891121214134909
100	0.8965206222743773
200	0.8951065132223771

Going on with the different parameters of gradient boosted and setting the learning_rate to 0.2 or 20%. The value for cv was set to 5 to get a more defined answer. And the rest of the values were set to default. The value for n_estimators was changed to see the fluctuation in the value for scores. The results are shown below in the table.

N_ESTIMATORS	**SCORE**
1	0.0
5	0.7520030912460729
10	0.8062954493887803
30	0.8937118955802422
50	0.8868623532729444
100	0.8888272414964037
200	0.8814481014180654

The last changes in parameters that we incorporated was we set the parameter for max_depth to 25. The value for cv was set at 5. The learning_rate was set at 0.1 The rest of the parameters were same. The value for n_estimators were changed to see the variation in the values. The table shows the results.

N_ESTIMATORS	SCORE
1	0.0
5	0.8397393310944757
10	0.8325710478004545
30	0.8167749706468221
50	0.8206108783557244
100	0.8176176850305614
200	0.808477486263999

Most of the data that was calculated didn’t took much time, but the ones with values of n and random states going to 200, started to increase the time taken to get the values. The machine which was using 8 gb of RAM and was running on both ssd and hdd. SO, although the time taken was more, it was relatively short compared to other systems. These were all the models and the different changes that we incorporated in their parameter to get the best results.

Findings: This must include the insight you got from the data set, the limitations of your study, what advice you would like to give to the organization in terms of future data collection and processes, etc. If the organization plans to implement your solution, what are the key points related to deployment? When would the model expire, etc.?

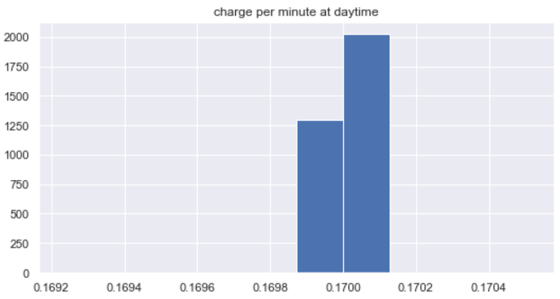
Insights

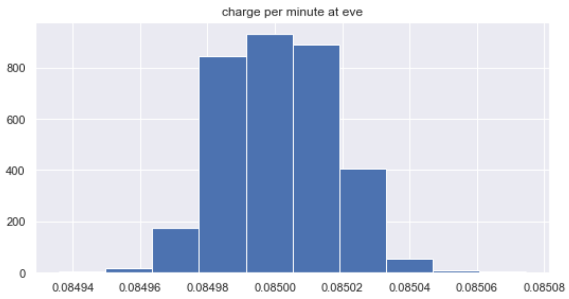
We employed two statistical methods, **Chi Square Test** and **t-test**, to find out if there exists a relationship between two features. Moreover, we used data visualization to visualize any noticeable patterns, any difference in the data distribution, and used boxplots to visualize any outliers.

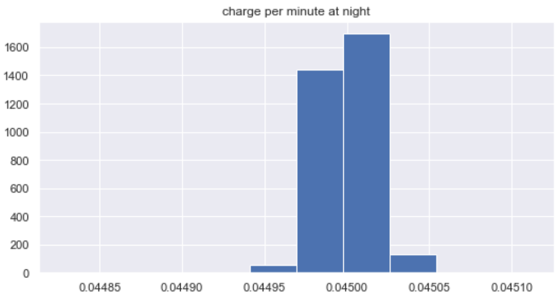
We also performed some computations on the features to **engineer** more features that could reveal some patterns. For example, we computed charge per minute for daytime calls by doing the following:

Charge per minute at daytime = [total day charge / total day minutes]

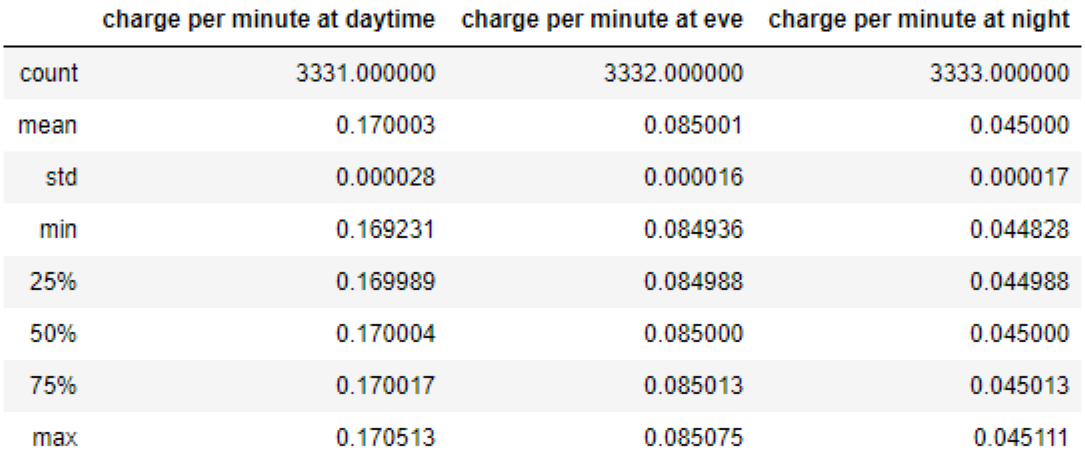
We did the same for the evening time and night time and then plotted histograms to visualize the distribution of values.







The table below summarizes the data of these three new columns:

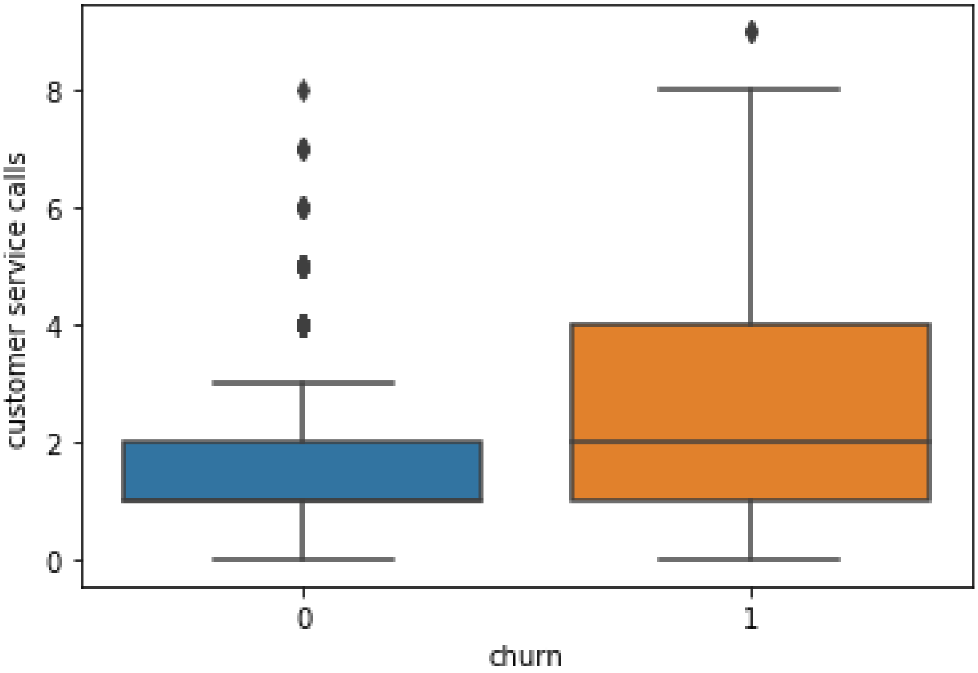


	charge per minute at daytime	charge per minute at eve	charge per minute at night
count	3331.000000	3332.000000	3333.000000
mean	0.170003	0.085001	0.045000
std	0.000028	0.000016	0.000017
min	0.169231	0.084936	0.044828
25%	0.169989	0.084988	0.044988
50%	0.170004	0.085000	0.045000
75%	0.170017	0.085013	0.045013
max	0.170513	0.085075	0.045111

We discover that the mean charge per minute at night is the least and the mean charge per minute at daytime is the highest. Later we will see

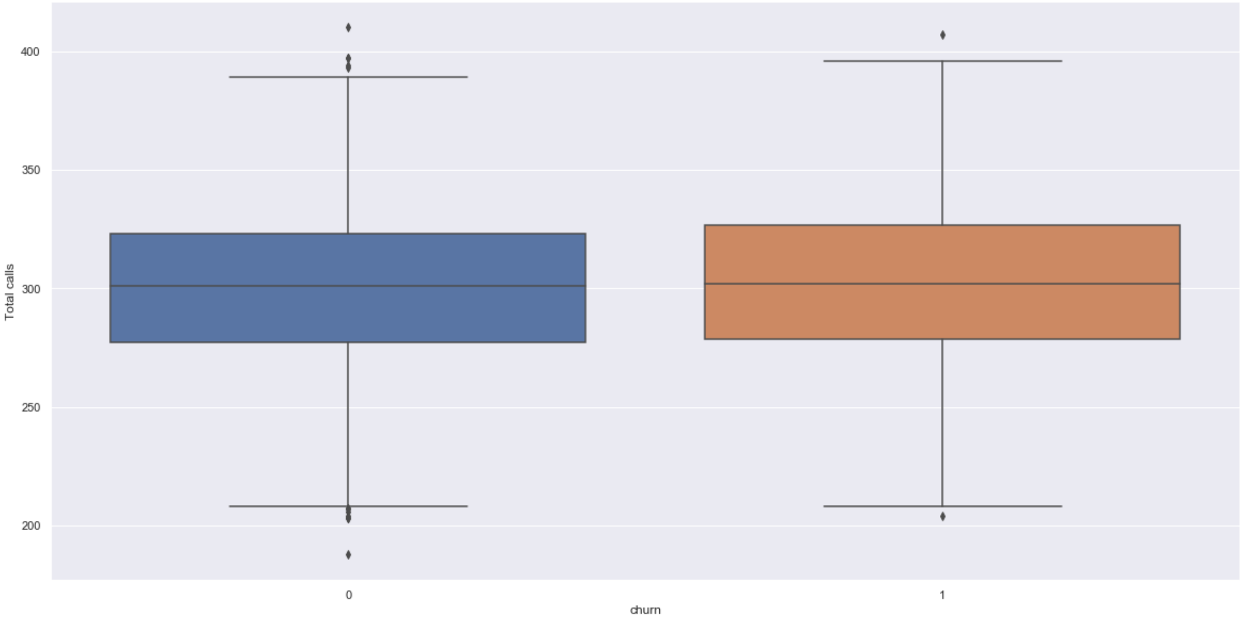
Difference in data distribution of numerical features in different classes (churn and active)

Did the customers who churned make more calls to customer service on average than those who remained active?



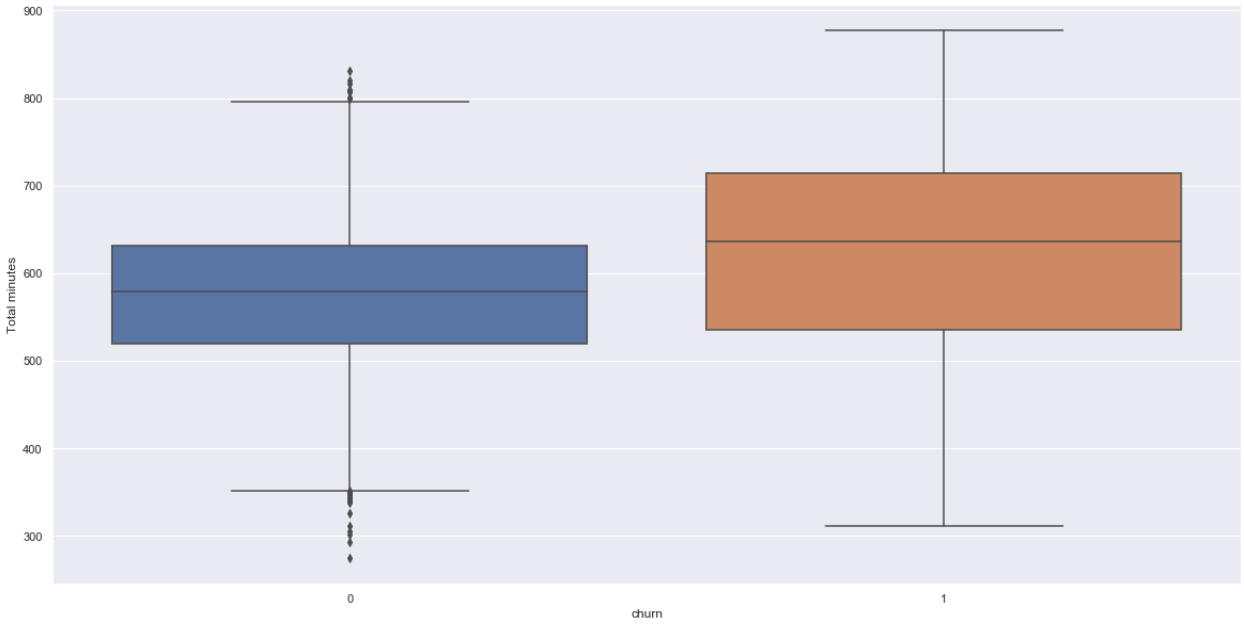
On applying t-test, we discover that distribution of data in each class is significantly different from the distribution in the other. Customers who churned made more calls to customer service on average.

Did the number of total calls (an engineered feature) made by a customer determined whether they churned or not?



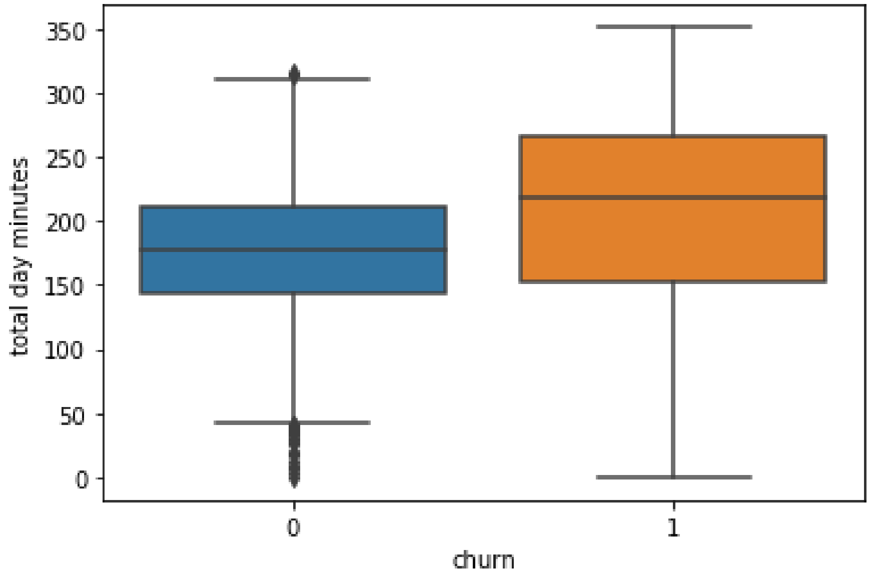
The box plot implies that the distribution of total calls' values is same in both classes. We confirm this by applying a t-test and get a p-value of 0.257 which is greater than 0.05.

Did the total minutes (an engineered feature) of a customer helped in determining whether they churned or not?



Distributions of data appear to be different in the two classes. We confirm this by applying t-test and get a p-value of 0.000.

Total day minutes of a customer determined whether they churned or not.



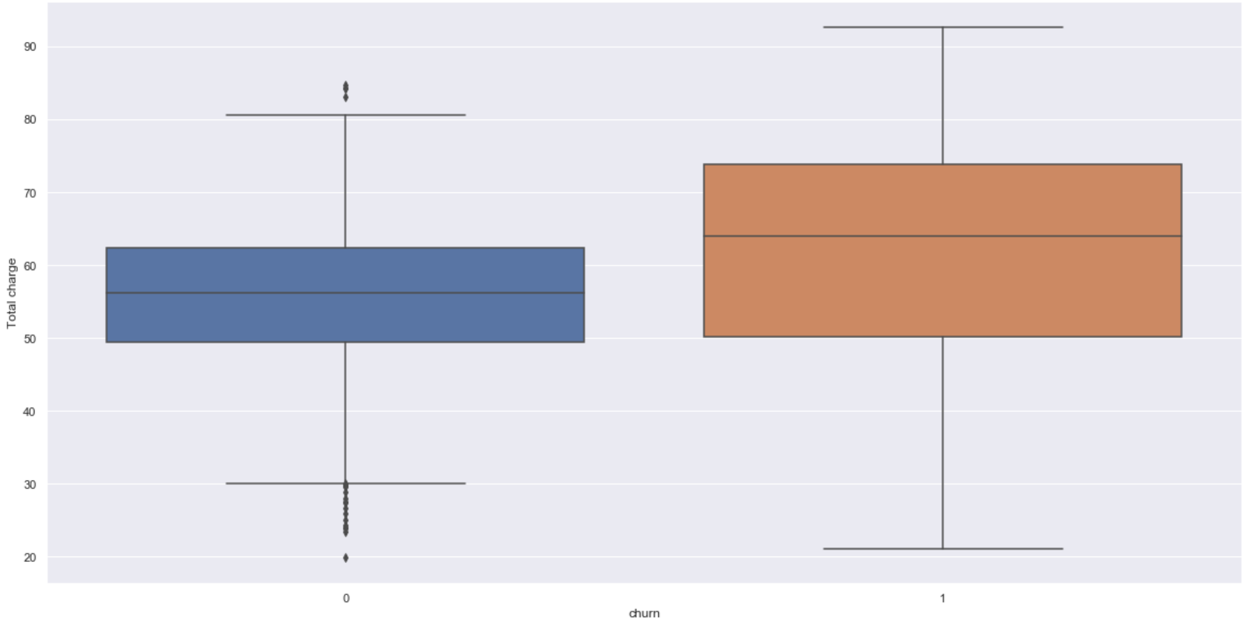
Not only the visual representation enough shows the significant difference in the data distributions in the two classes, but the t-test also confirms this.

Total charge to a customer and their decision to churn

A new feature named “Total charge” was engineered using the following formula:

Total charge = total day charge + total evening charge + total night charge

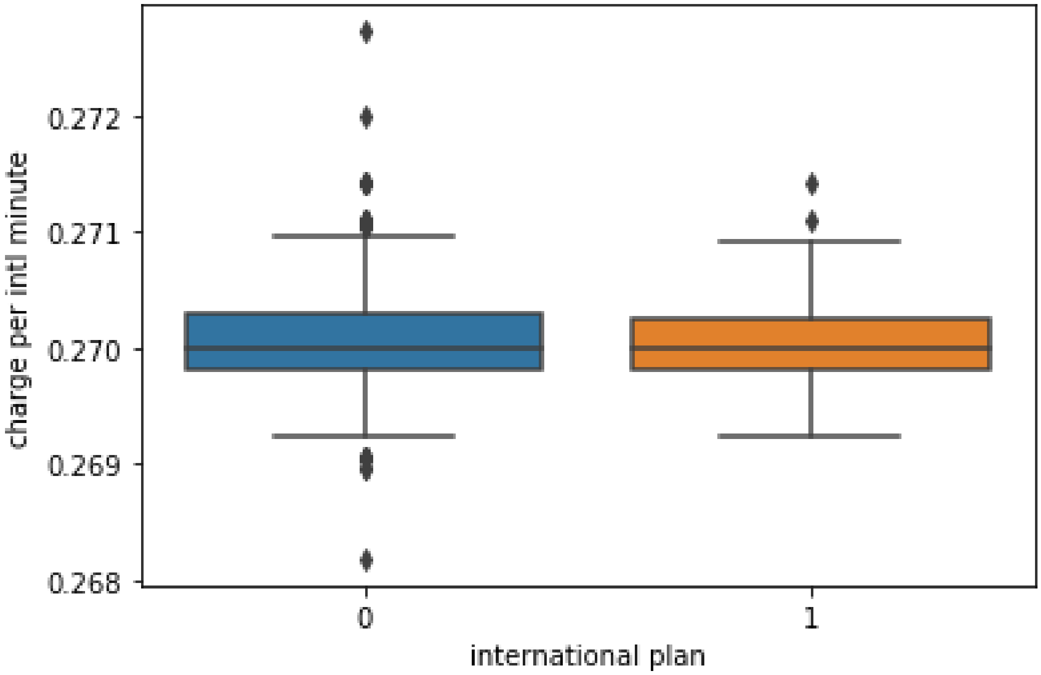
This new feature turned out to be important in determining a customer’s decision to churn.



Both the visual representation and the t-test confirm the difference in the data distributions in each class.

Relationships Between Non-target Variables

Were the customers who had an international plan charged different for call per minute from those who did have an international plan?



The distribution of the data above appears to be same for the customers who had an international plan and those who did not. We confirm this by applying t-test and getting a p-value of 0.293 which is greater than 0.05. *Could this be a reason for the existence of a relationship between a customer having an international plan and their likelihood to churn?*

We know that there exists a relationship between a customer having an international plan and the likelihood that they would churn. We confirm this with a Chi Square Test by getting a p-value of 2.493e-50.

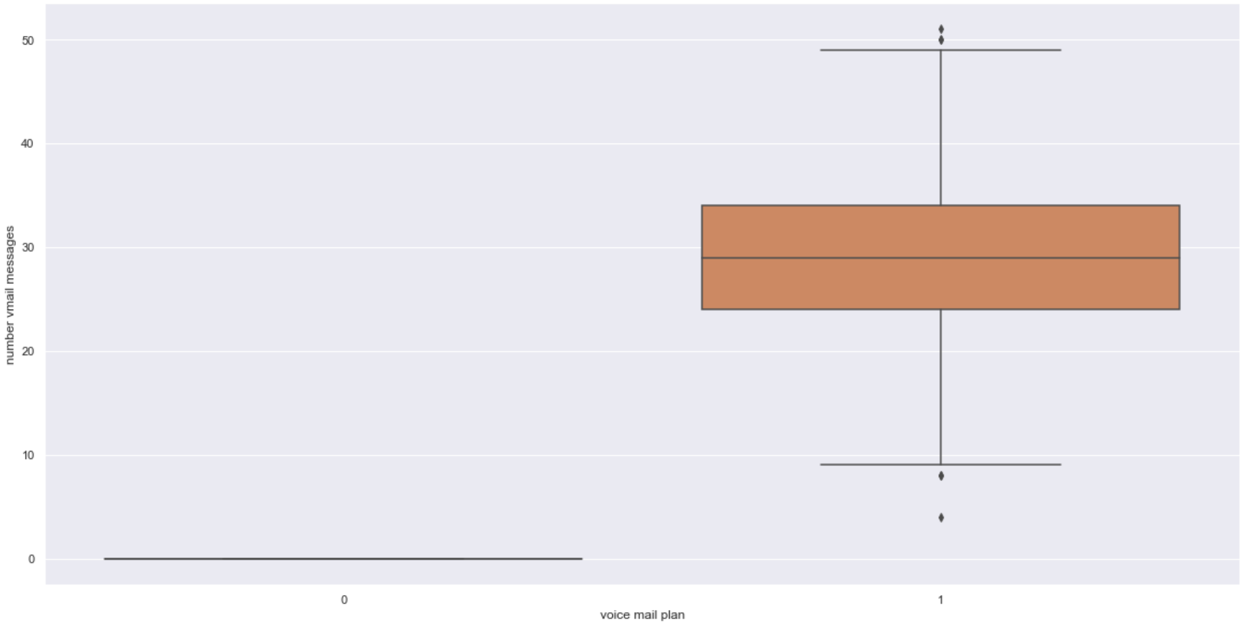
Our findings related to the international plan:

Customers who had the international plan were not charged lesser per minute of an intl call than those without the international plan.

Customers with international plan made almost the same number of calls as those without the international plan.

However, customers with the international plan made longer international calls than those without the plan, as their international calls' minutes were more than those without than of those without the plan.

Customers with a voice mail plan sent more voice mail messages on average than those without the plan

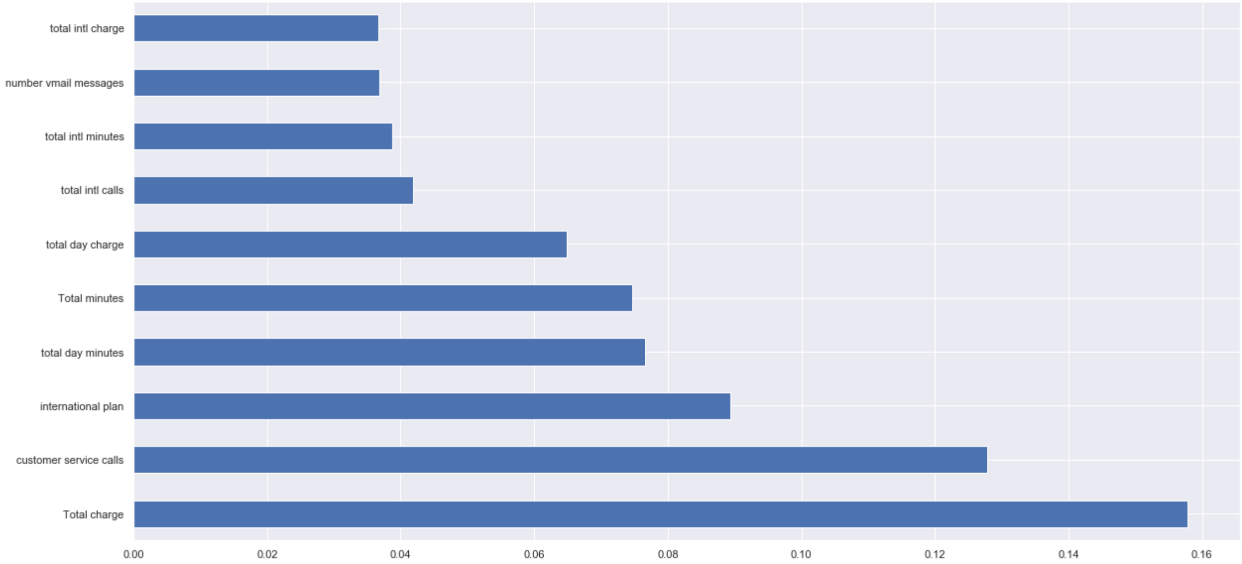


The figure above clearly indicates the difference in data distributions of number of voice mail messages sent by customers in the two classes of “voice mail plan” feature (Yes/No). We confirm the relationship using a t-test that gives a p-value below 0.05.

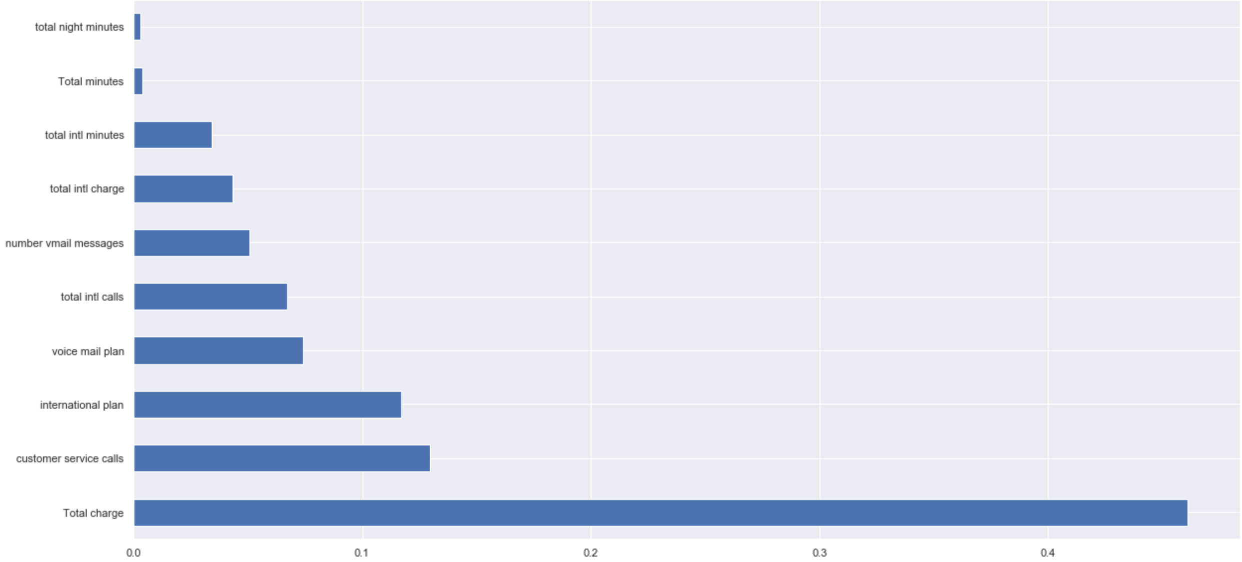
The "Feature Importance" Attribute Of Some Classifiers

The feature importance (variable importance) describes which features are relevant. It can help with better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection.

Below are the top 10 most important features in determining the class label when we fit a Random Forest Classifier with 50 estimators and entropy as criterion.



And below are the top 10 most important features in determining the class label when we fit a Gradient Boosting Classifier with 100 estimators. This gave us the highest F1 score.



We conclude that the "Total charge" has been the most important feature in determining whether a customer would churn or not. This feature was not originally present in our dataset and we engineered it from three existing features.

Furthermore, the set of 10 most important features for both classifiers are not much different. It means that are models are well generalized and are safe from overfitting. The cross validation scores also do not vary greatly, and this is another indication of our models' good generalization.

Feature Selection

We summarize the feature selection methods used and the scores they gave with best classifier in the table below.

Feature Selection Method	F1 score with best classifier (GB)
Removing features with low variance	0.8998349416819164
Select K-Best	0.8985077715168481
L1-based feature selection	0.7494902438078578
Tree-based feature selection	0.89885418438831

Challenges and limitations

Thinking of extracting more information from the existing features was difficult. And there were so many variables that appeared related in the dataset and so using the most appropriate of these for data visualization and statistical tests was a challenge. Often only one of the two related variables was related significantly with a third variable.

What we could not find out in our study was: what was the exact purpose of international plan and how it impacted a customer's decision to churn?". Our analysis of the international plan part revealed some information but it could not answer how the customer's decision depends on it.

Visualizing the impact of a categorical variable on the class label was not possible when a categorical variable, such as the variable "State", had too much categories.

For instance, you can go back above and see the boxplot of total charge against churn status. When one feature is numeric and another is categorical with just two categories, we can visual the distribution and see what values of numeric variable increases the likelihood of a customer to churn.

However, when there are too many categories in a categorical variable, it becomes difficult to find out which value of the categorical variable increases the likelihood of which value for class label.

When would the model expire?

The model would require retraining if the company makes changes to any of its policy. For example, if the company lowers the charge per international call for those customers having an international plan, then it might totally change how the value of international plan's column determines the class label. Currently, as per our research there seem to be no difference in charge per minute of an intl call for customers with and without the international plan.

If the company decides to change the value they charge per minute during the day, evening, and/or night, then this would also have its impact on a customer's decision to churn and hence would require you to train your model with a new dataset.