Phase 3 Project

Project Overview

For this project, I used several regression models to model the data from SyriaTel and predict if their customer will churn the plan or not.

Business Problem

SyriaTel Customer Churn (Links to an external site.) Build a classifier to predict whether a customer will ("soon") stop doing business with SyriaTel, a telecommunications company. Note that this is a binary classification problem.

Most naturally, your audience here would be the telecom business itself, interested in losing money on customers who don't stick around very long. Are there any predictable patterns here?

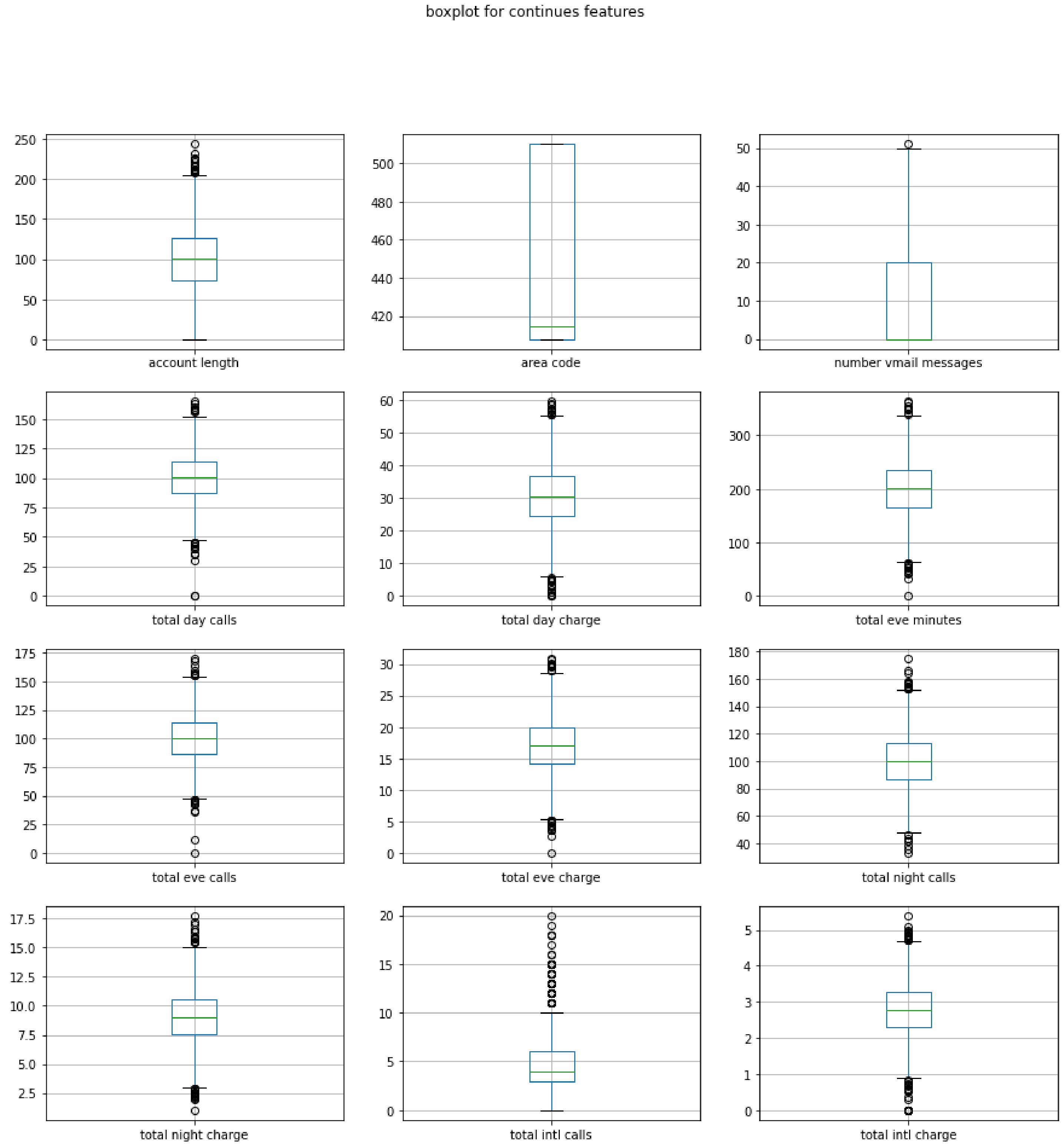
1. polish the data which have no meaning or is null to the price.
2. remove the features which do not contribute to the house price.
3. check if there are some high correlated features in which some of them can be removed.
4. build the linear regression model.
5. check how the features can contribute to the house change.

Plan

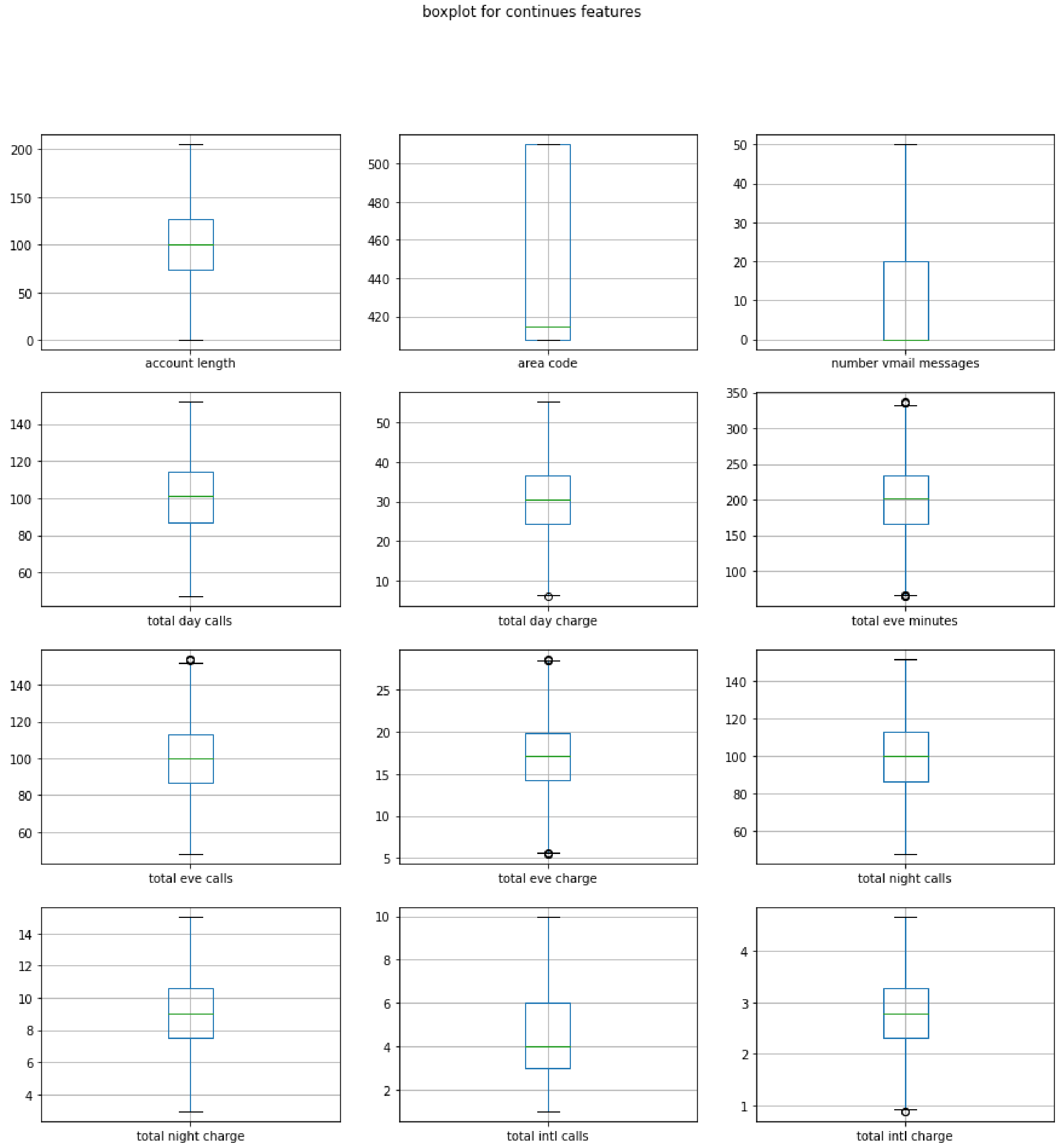
Since the SyriaTel Customer Churn is a binary classification problem problem, I will try to use several different algorithms to fit the data and select one of the best one. The algorithms I will try include Logistic Regression, k-Nearest Neighbors, Decision Trees, Random Forest, Support Vector Machine. The target of the data we need to fit is the column 'churn'. The features of the data is the other columns in dataframe. However, when I load the data file into dataframe, i found some of the columns are linear correlated with each other. I need to drop one of them. We need to polish the data first.

Looking at the dataframe, I need to steply polish some features and remove some of the columns:

1. The pairs of features inclued (total night minutes and total night charges), (total day minutes and total night charges), (total night minutes and total night charges), (total intl charge and total intl minutes) are high correlated with each other. I need to remove one in each columns.
2. All the phone numbers are unique and act as id. So it should not related to the target. I will remove this feature.
3. The object columns will be catalized.



The above figures show that there are multipal columns contain some outlier data. I then collected all the columns and remove the outlier by 1.5 x IQR



The data looks much better now with very few of outlier numbers.

Now the data was ready and we need to prepare and modeling the data with varies models.

Plan

1. Perform a Train-Test Split

For a complete end-to-end ML process, we need to create a holdout set that we will use at the very end to evaluate our final model's performance.

2. Build and Evaluate several Model including Logistic Regression, k-Nearest Neighbors, Decision Trees, Randdom forest, Support Vector Machine.

For each of the model, we need several steps

```
1. Build and Evaluate a base model
2. Build and Evaluate Additional Logistic Regression Models
3. Choose and Evaluate a Final Model
```

3. Compare all the models and find the best model

1. Prepare the Data for Modeling

The target is Cover_Type. In the cell below, split df into X and y, then perform a train-test split with random_state=42 and stratify=y to create variables with the standard X_train, X_test, y_train, y_test names.

Since the X features are in different scales, we need to make them to same scale. Now instantiate a StandardScaler, fit it on X_train, and create new variables X_train_scaled and X_test_scaled containing values transformed with the scaler.

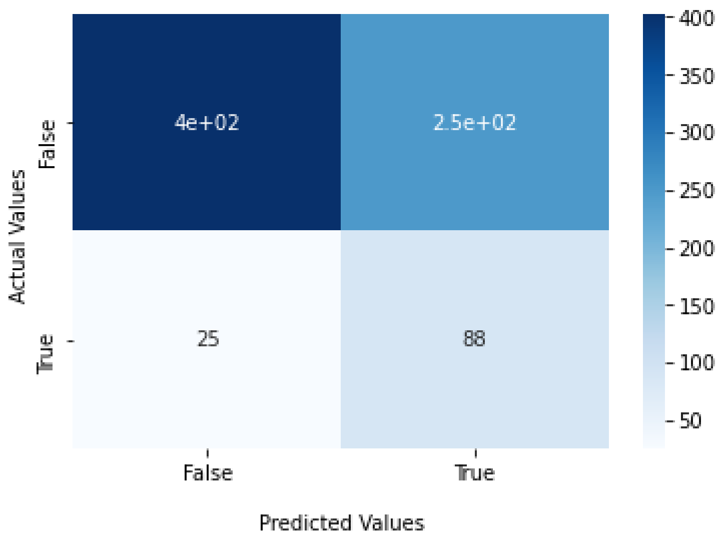
2. Build and Evaluate several Model

I. Build the model with Logistic Regression

Basescore for training and testing data are: 0.62997

0.64267

I then plot the confusion matrix for this model



Seaborn Confusion Matrix with labels

The score for LogisticRegression is not very high. It is just above the random guessing. The false positive and false negtive rate are very high.

II. Build the model with k-Nearest Neighbors

Basescore for training and testing data are: 0.90782

0.88874

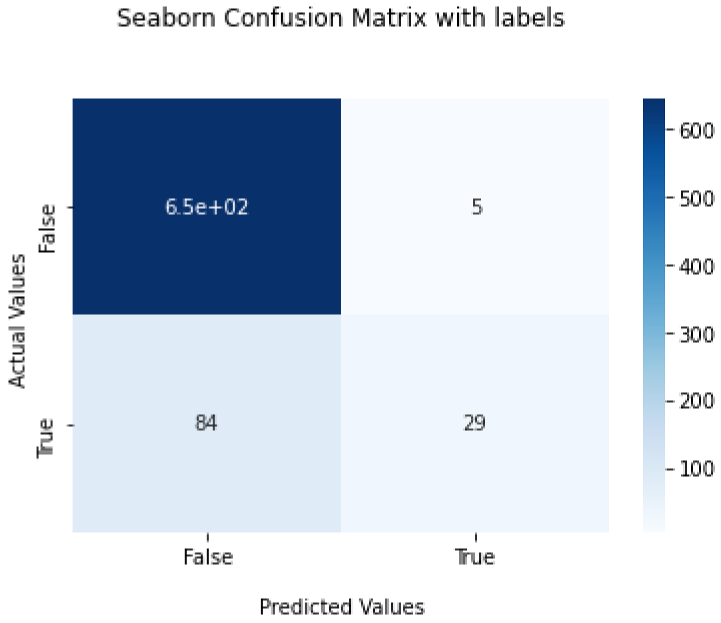
The scores for KNeighborsClassifier are pretty high. But the score for traing is higher than testing data. We will try to use other parameter to find the best number of neighbor used for fitting.

The best parameter is KNeighborsClassifier(n_neighbors=7)

Score for best parameter 0.90083

0.88351

Plot the confusion matrix



Compare to the baseline model, even though the training score decreased, the testing score increased. However, the confusion matrix showed there are a lot of false negtive.

III. Build the model with Decision Trees

Basescore for training and testing data are: 1.0

0.90314

The scores for DecisionTreeClassifier are very high even 100% for trainning data. However, the score for testing is only 90% which suggest the DT_baseline is overfitting.

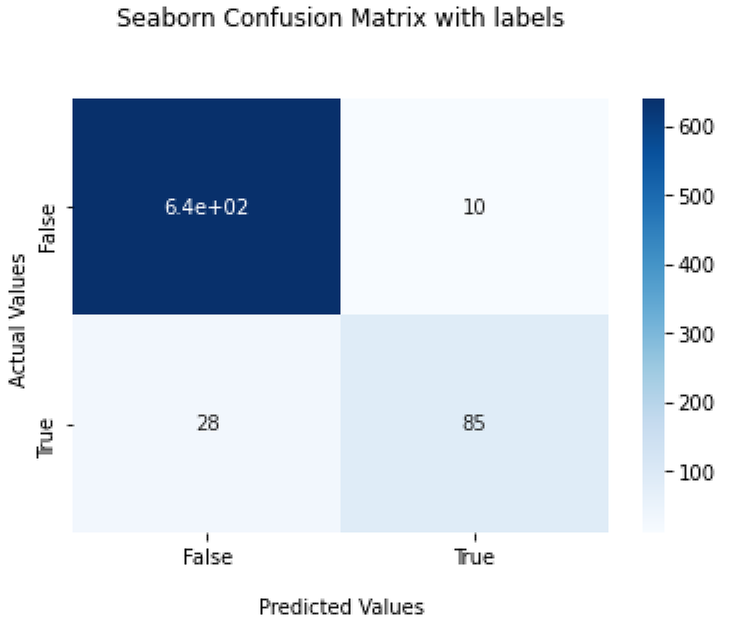
find the best parameter with dt_grid_search

{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 6, 'min_samples_split': 2}

Best score: 0.96461

0.95026

Plot confusion matrix



Compare to the DT baseline model, even though the training score decreased, the testing score increased. Now the two scores are close to each other and both of them are very high. The confusion matrix is also pretty resonable compare to other models.

IV. Build the model with Support Vector Machine

Basescore for training and testing data are: 0.93578

0.90707

find the best parameter with dt_grid_search

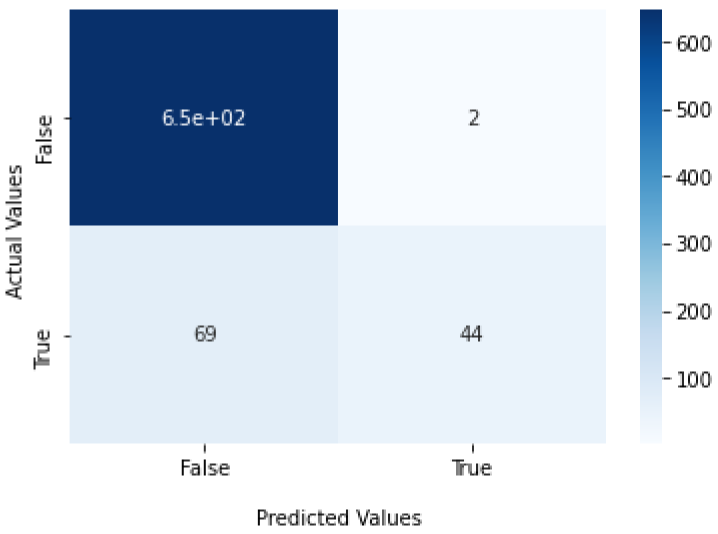
{'C': 5, 'gamma': 'auto', 'kernel': 'rbf'}

Bestscore are:

0.93578

0.90707

Seaborn Confusion Matrix with labels



Compare to the SVC baseline model, the training score decreased, the testing score is not changing. They are pretty high but still less than DT model. The False negtive rate for this model is also very high.

V. Build the model with RandomForestClassifier

Basescore for training and testing data are: 1.0

0.94895

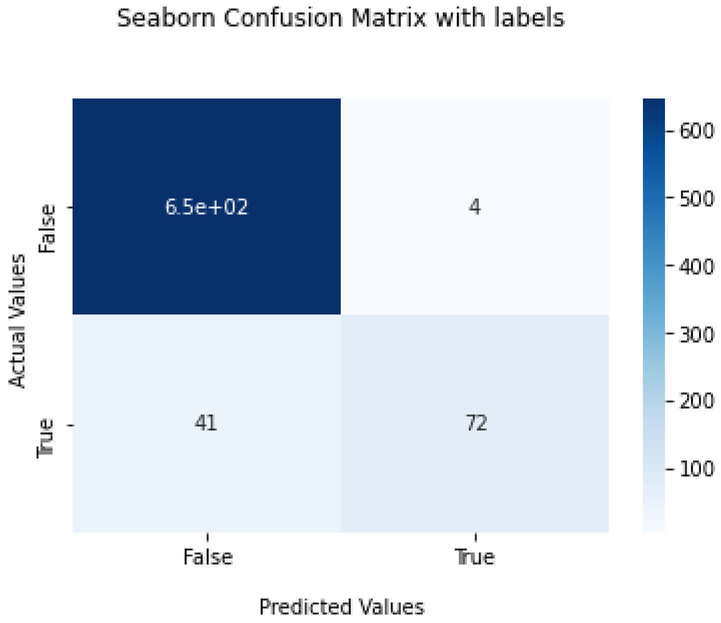
find the best parameter with dt_grid_search

Optimal Parameters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 10, 'n_estimators': 30}

Best score are:

0.96461

0.92539



Compare all the models and find the best model, then evaluate it.

When comparing the final score for training and testing data, the decision tree model give us best results.

I make this model to the final one.

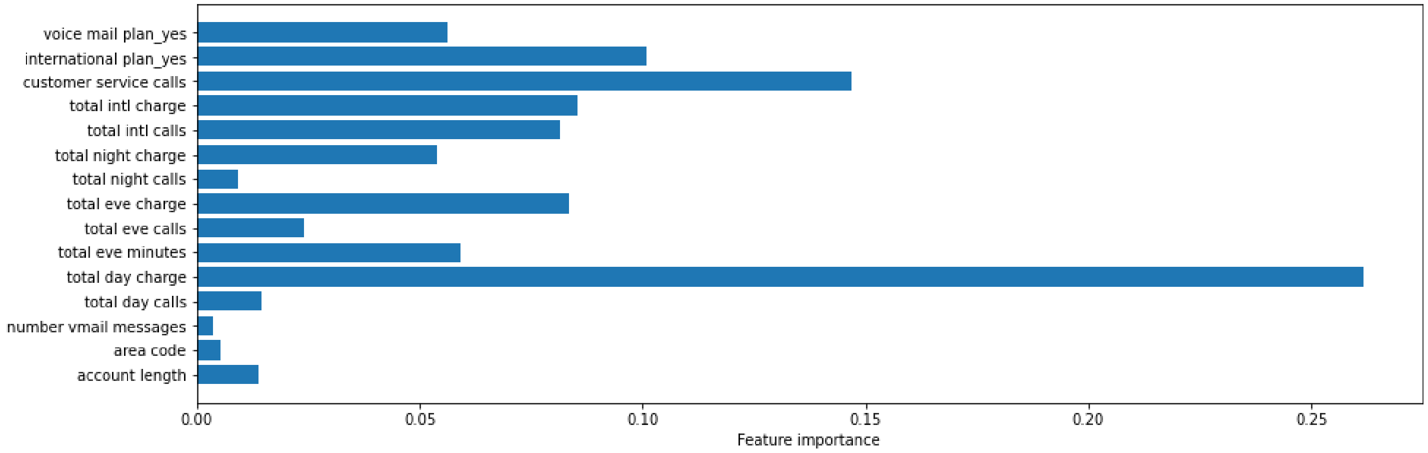
0.96461

0.95026

Replot the confusion matrix.

The final score for training and testing data are very high and close to each other which suggest there is no overfit or downfit to the trainning data. Now let find out the weight of each features to the target results.

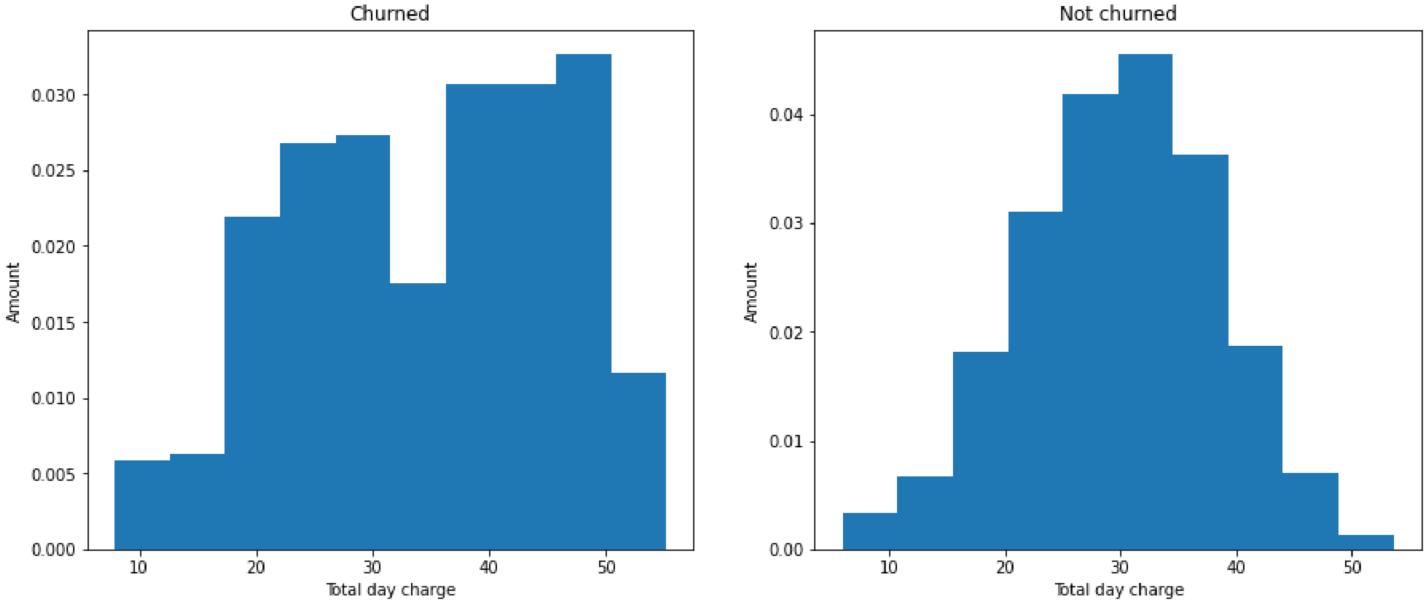
Feature: account length , Score: 0.01396 Feature: area code , Score: 0.00536 Feature: number vmail messages , Score: 0.00381 Feature: total day calls , Score: 0.01465 Feature: total day charge , Score: 0.26170 Feature: total eve minutes , Score: 0.05911 Feature: total eve calls , Score: 0.02410 Feature: total eve charge , Score: 0.08361 Feature: total night calls , Score: 0.00912 Feature: total night charge , Score: 0.05405 Feature: total intl calls , Score: 0.08132 Feature: total intl charge , Score: 0.08538 Feature: customer service calls , Score: 0.14691 Feature: international plan_yes , Score: 0.10091 Feature: voice mail plan_yes , Score: 0.05603



Find the top 5 important features.

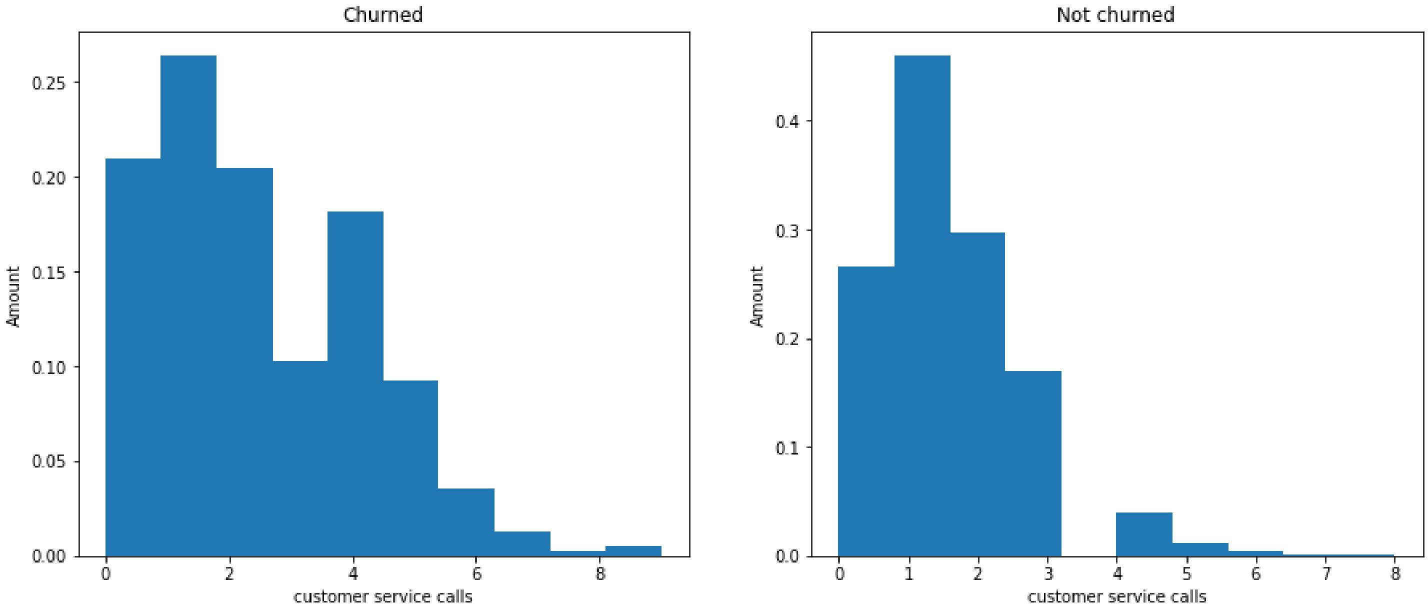
('total day charge', 0.26169763518771966), ('customer service calls', 0.1469102061483546), ('international plan_yes', 0.1009100150705904), ('total intl charge', 0.08537600988682308), ('total eve charge', 0.08360637737007212)

Check if there is special patten for the top five important features



The histograms for customers who churned and not churned show that the total day chare have a lot of overlap with each other.The customers who had total day charg more than 40 have more chance to churn the plan.

Plot the histogram for 'customer service calls' of customers who churned and not churned with similar code.



The histogram are similar to each other. However, the customer who had 4 international calls had higher chance to churn the plan.

Since the column 'international plan_yes' contains only 0 and 1. I plot the value counts for bot churned and not churned.

not churned

0 2446

1 173

Name: international plan_yes, dtype: int64

churned

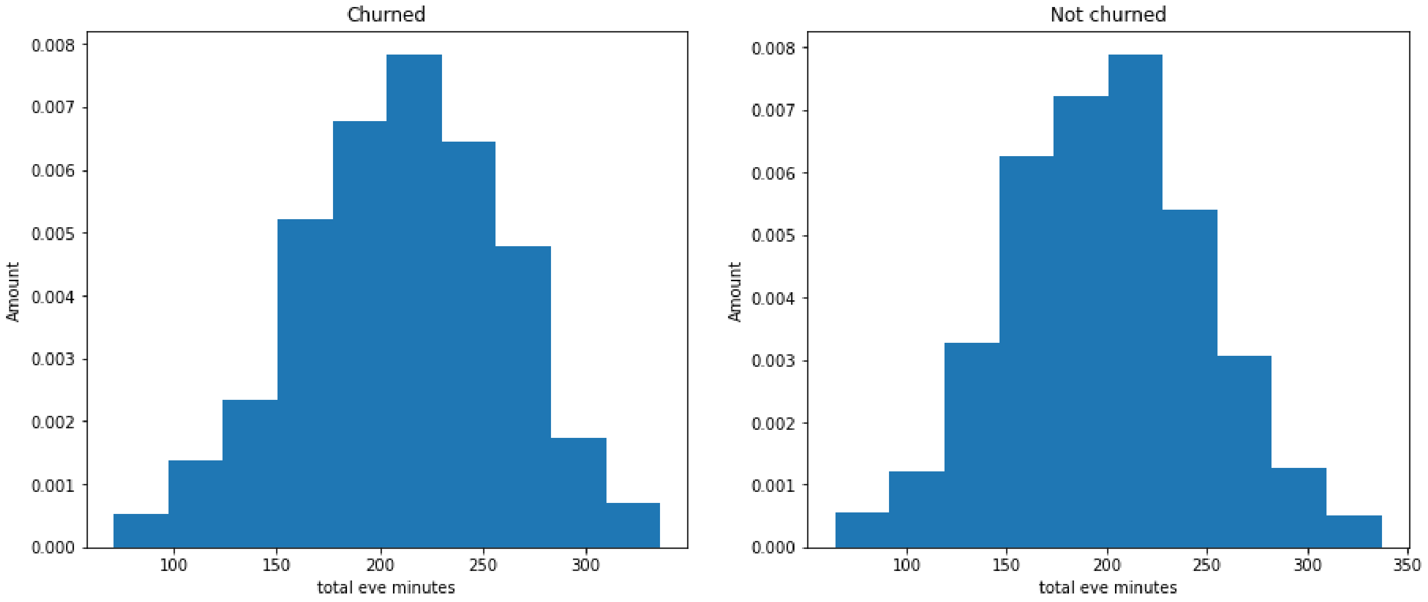
0 313

1 121

Name: international plan_yes, dtype: int64

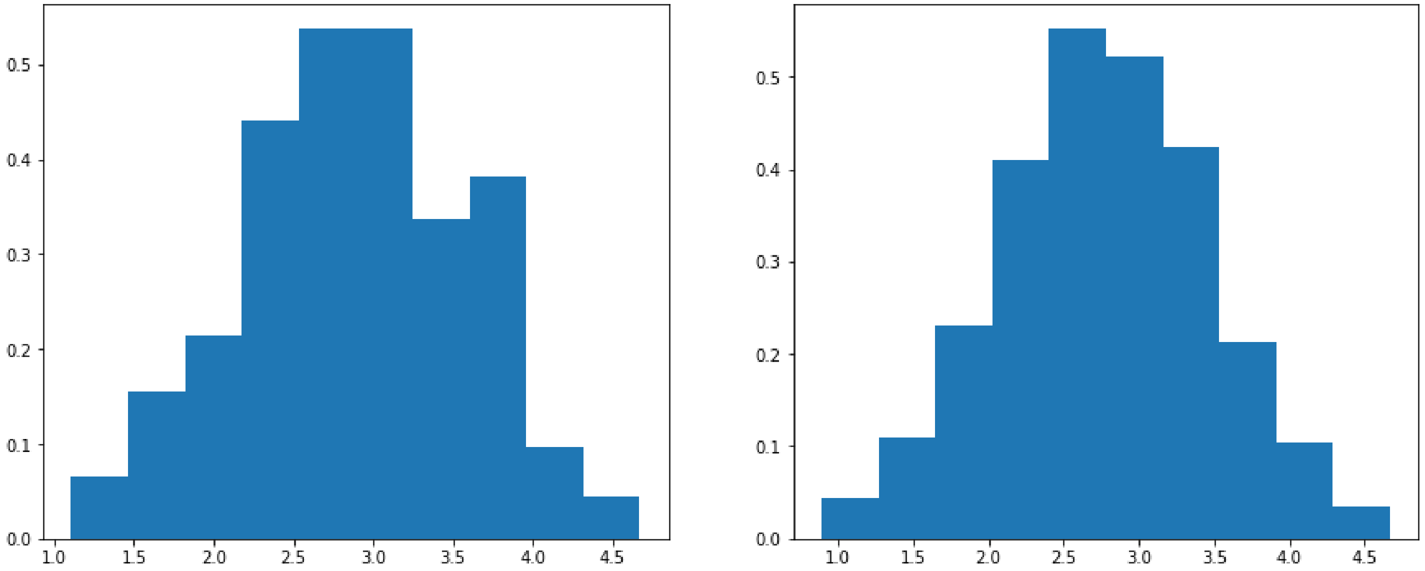
This data show that the customer who had international plan have much higher chance to churn the plan. .

Plot the histogram for 'total eve minutes' of customers who churned and not churned.



There is no clear relationship between total eve minutes and churn or not.

Plot the histogram for 'total intl charge' of customers who churned and not churned.



There is no clear relationship between total intl charge and churn or not.

Conclusion

We polished our orignal data by removing the outlier and catlize the necessary columns. We then tested several of models to fit out data and selected the best one which is desicion tree. The final score of predicting is 0.94 which is very high. By dig out the relation ship between the top 5 weighted features and target column (churn), we found that people who had day charge more than 40 or had customer service calls 4 and more, or had international plan had higher chance to churn the plan. So the company might focus on these customers and make some special promotions on these plan to attract more customer on them.