Classifier Models on Customer Churn Data

Main Objective

- In this report, we will analyze the customer churn dataset for a telecom company
- The objective of this report is to find out the best predictors regarding a customer continuing to stay with a telecom company or churning to find a new service provider.
- The primary purpose of this regression-based report will be based on the interpretation of the various features in the dataset.
- In doing so we will use a variety of classification models and figure out the one that is best suited to our needs.

Brief Description of the Dataset

This dataset has 20 features that describe the various aspects of a customer's relationship with a telecom company and one target column, the churn value of the customer, that shows if the customer has left the company. There are 7043 observations or data points that constitute the rows corresponding to each of the columns of our dataset.

The 20 features in the dataset are

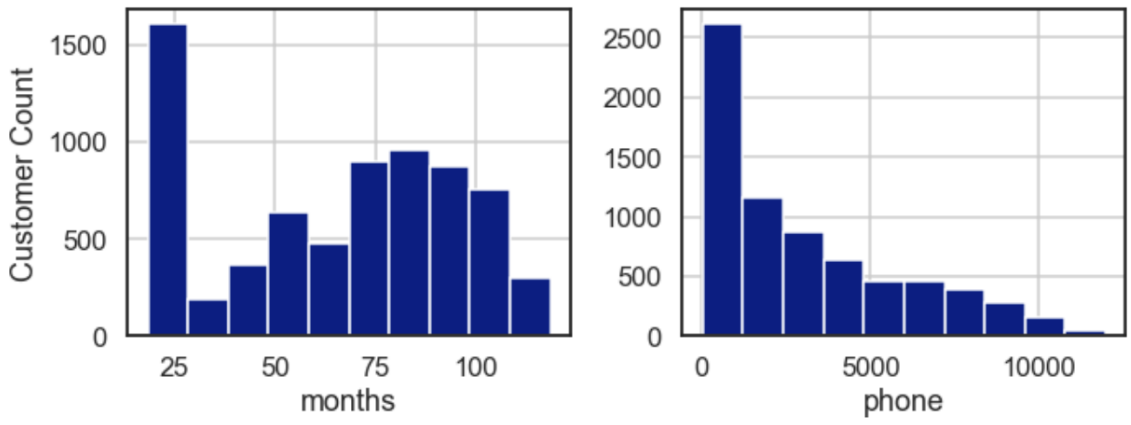
1. Id: A unique customer id of every customer(object).
2. Months: Duration in months for which the individual has been a customer(int).
3. Offer: 6 different options(string)
4. Phone: Yes/No(string)
5. Multiple: Yes/No(string)
6. Internet Type: 4 different options (string)
7. Data Usage per Month in GB : (int)
8. Security: Yes/No (string)
9. Backup: Yes/No (string)
10. Protection: Yes/No (string)
11. Support: Yes/No (string)
12. Unlimited: Yes/No (string)
13. Contract Duration: "Month-to-Month", "One Year" or "Two year" (string)
14. Paperless: Yes/No (string)
15. Payment: 3 different options (string)
16. Monthly charges : (float)

17. Total Revenue from a Customer: (float)
18. Customer Satisfaction: Number from 1 to 5 (int)
19. Customer Lifetime Value: (int)
20. Churn Score: Company's assigned score related to possibility of churning (int)
21. Churn Value: 1 if the customer has churned and 0 otherwise (int)

Overall, there are 13 categorical, 6 integer, and 2 float type columns.

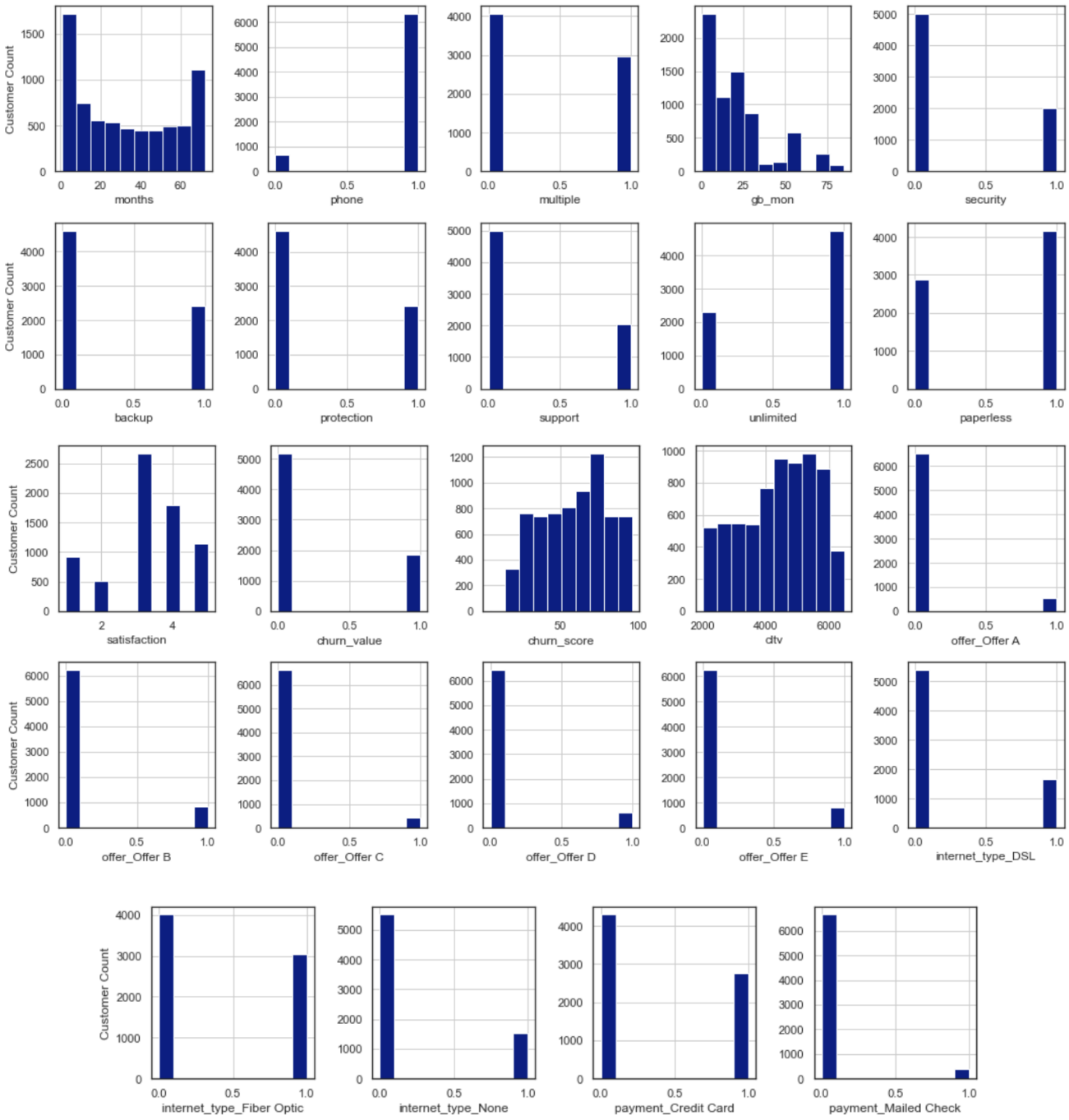
Data Cleaning and Feature Engineering

- There are no empty cells in our dataset, so we move straight to looking at the outliers.
- We drop the 'ID' column because it has unique values for every cell and is hence useless for any machine learning model.
- We now look at the outliers in the two float columns – 'Monthly Earnings' and 'Total Revenue':



- These two histograms show us that there are no significant outliers in the float features that need to be removed from our dataset.
- We now encode our categorical features.
- We start with the 8 features that are binary (Yes/No) – Phone, Multiple, Security, Backup, Protection, Support, Unlimited and Paperless. We use the labelBinarizer method for these features and convert them into zeros and ones.
- The 'Contract Duration' feature is ordinal. We convert 'Month-to-Month' to 0, 'One Year' to 1, and 'Two Year' to 2 for this feature.
- For the remaining three columns – 'Offer', 'Internet Type' and 'Payment' – we use one-hot encoding and drop one of the columns to ensure that our resulting columns are not highly correlated due to multi-collinearity.

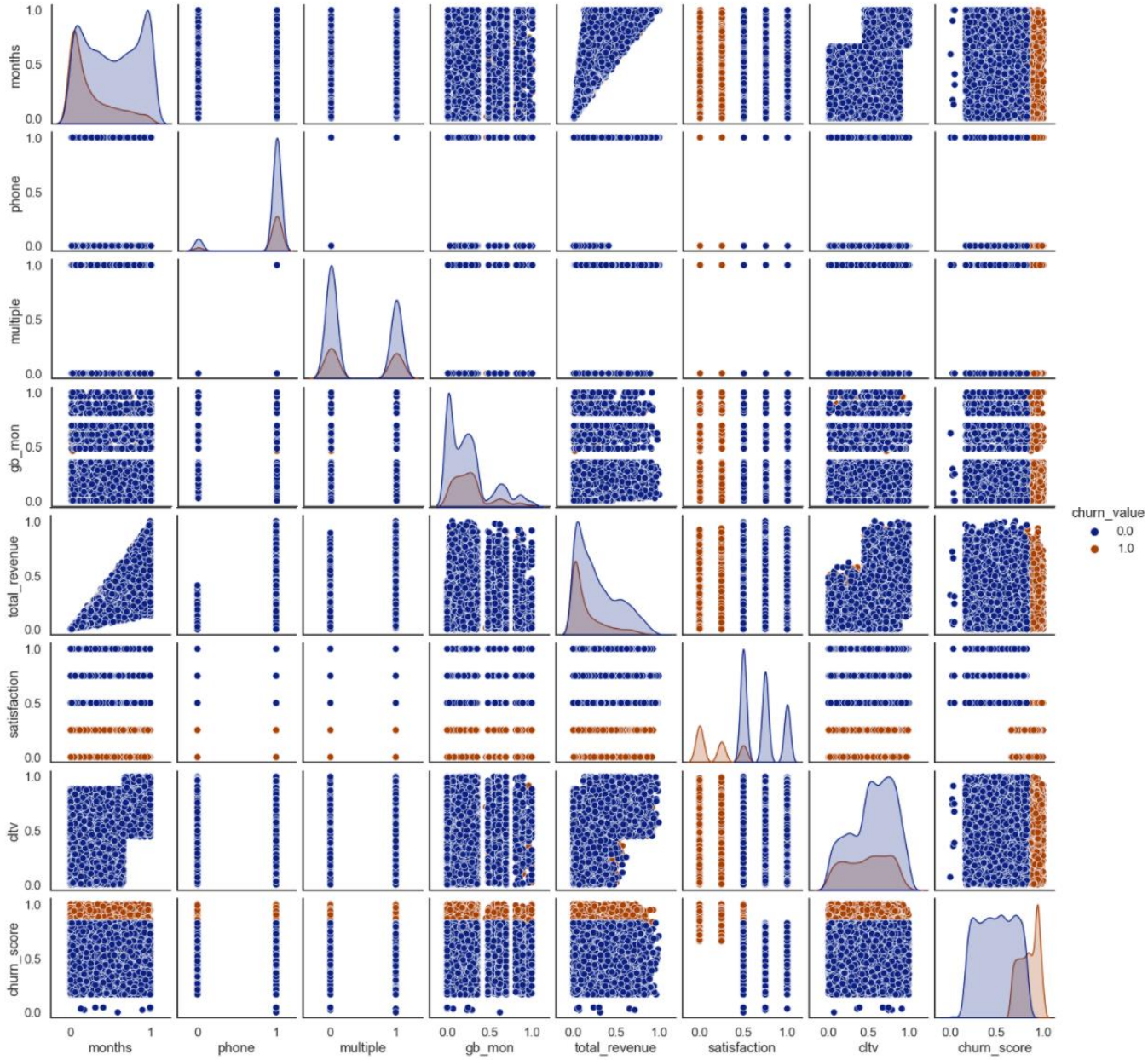
- After encoding all the categorical features, we have 24 more numerical features. We now look at their histograms to see if there are any outliers:



- Again there are no serious outliers that need to be removed from our dataset.

- We now use the MinMaxScaler to scale all the columns to numbers between 0 and 1.

- We then look at the pair plots of some of the features by separating them for the two possibilities of our target column.

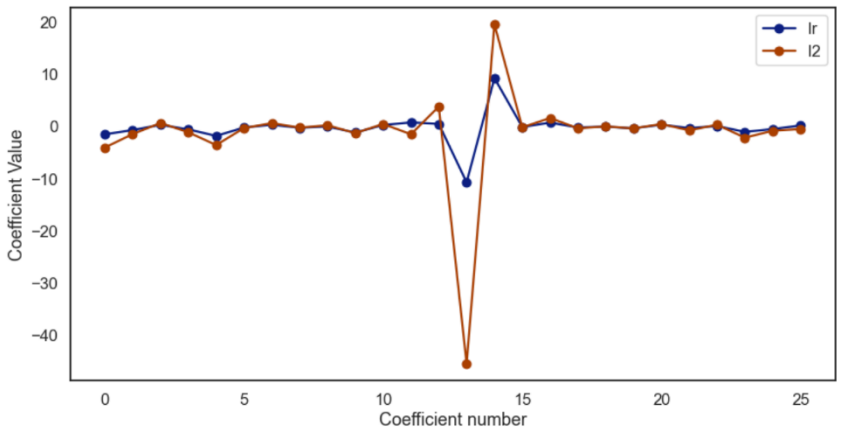


- From the pairplot we can easily see that the following features of a customer are correlated to not churning –longer duration, larger revenue, high satisfaction score, high customer lifetime value and a low churn score. Our classification models will confirm these observations will help us in establishing the relationship more rigorously.

Classification Models

In this section, we will use () models to first train our model on the dataset and then test it on another part of the dataset. In all the models, we will use the Stratified Shuffle Split method with a test size of 0.3, the number of splits set to 1 and the random state set to 17 to ensure we compare all the models on the same data. The Stratified Shuffle Split method ensures that our training and testing data have an equal number of ones and zero in our target column - Churn Value.

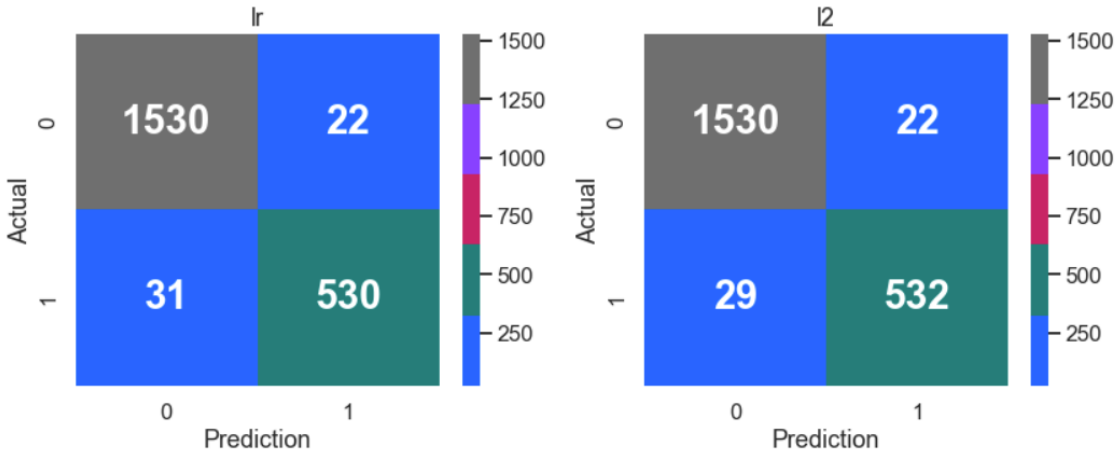
1. We start with the baseline **Logistic Regression** model with the solver set to liblinear. We implement this model simultaneously with another Logistic Regression model that has an *l2* penalty as a regularizer. The coefficients for these two models are comparable for most features but have quite different values for two features – Satisfaction and Churn Score. The coefficients are shown in the plot below:



We get the following metrics for the two Logistic Regression models.

	lr	l2
precision	0.974832	0.975794
recall	0.974917	0.975864
fscore	0.974852	0.975815
accuracy	0.974917	0.975864

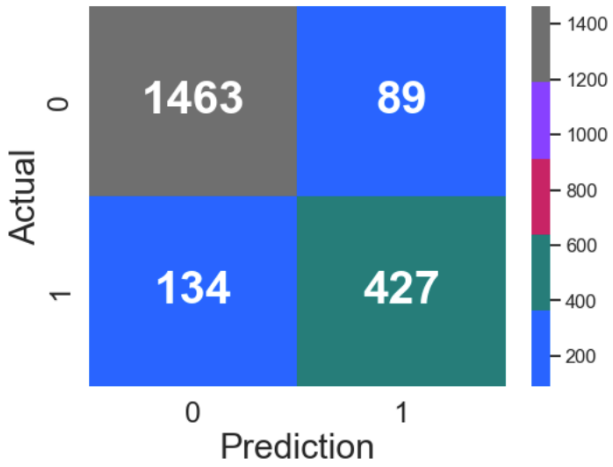
The confusion matrices for the two models are shown below



The difference between the two models is quite minimal for all the four metrics that are shown in the table.

2. We then use the **K nearest neighbors** method with the number of neighbors set to 3.
   The metrics for this model are shown in the table along with the confusion matrix

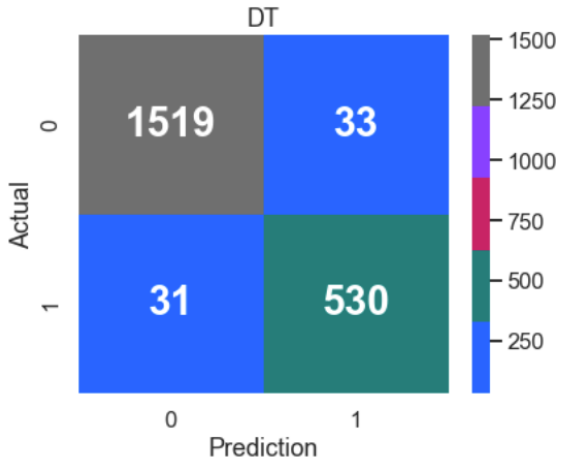
	KNN
precision	0.892577
recall	0.894463
fscore	0.893012
accuracy	0.894463



The metrics for this model are significantly worse than those for the two Logistic Regression models we tried. Thus, this is clearly not the best model suited for our needs.

3. Next, we try a **Decision Tree Classifier**. The metrics and the confusion metrics in this case are

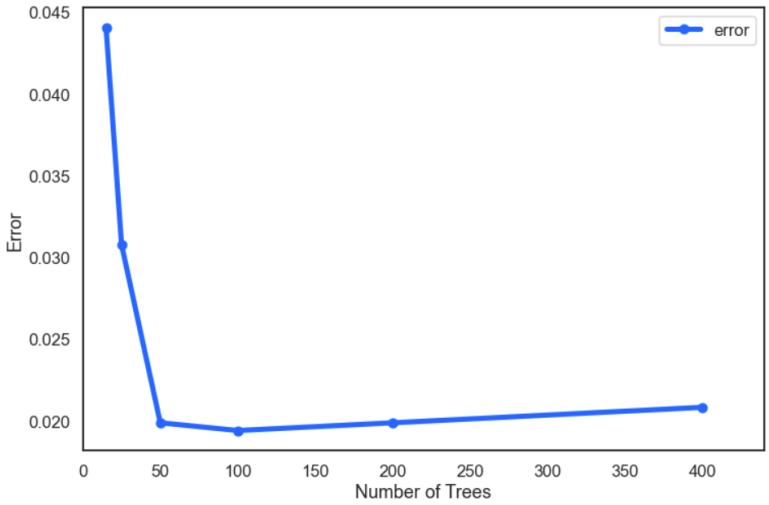
	DT
precision	0.971272
recall	0.971131
fscore	0.971188
accuracy	0.971131



The decision tree has 161 nodes, and its depth is 18.

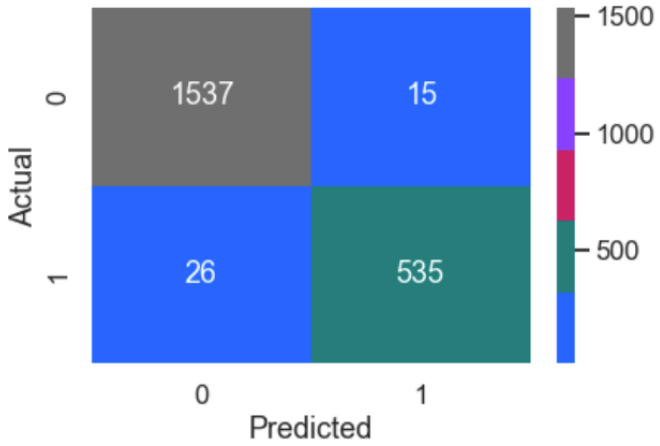
4. We now try a boosting classifier. Specifically, we use the **Gradient Boosting Classifier**. We try a variety of values for the number of trees, which is our hyperparameter for this method, and set the maximum number of features to 5. The error score versus the number of trees used is given by

n_trees	error
15.0	0.044013
25.0	0.030762
50.0	0.019877
100.0	0.019404
200.0	0.019877
400.0	0.020823



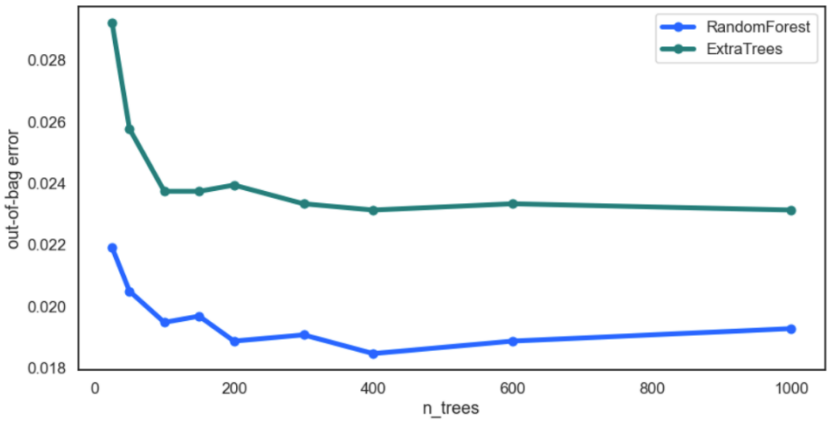
From this plot we find that the error is lowest for n = 100. Thus, we use this as our best estimator and work out all the metrics and the confusion matrix

	GBC
precision	0.980541
recall	0.980596
fscore	0.980535
accuracy	0.980596



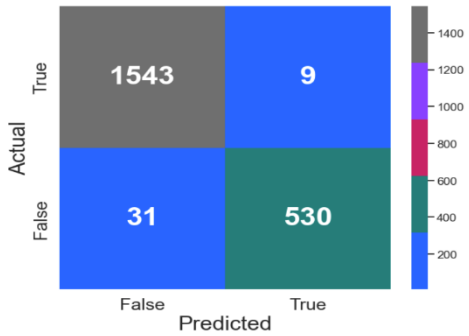
5. Finally, we try two bagging methods. Specifically, we use a **Random Forest Classifier** and an **Extra Trees Classifier**. Once again we use a variety of values for the number of trees for both the methods and try to find the best one by comparing the out-of-bag error for each of them. The errors are shown in the tables below:

n_trees	RandomForest	ExtraTrees
25.0	0.021907	0.029209
50.0	0.020487	0.025761
100.0	0.019473	0.023732
150.0	0.019675	0.023732
200.0	0.018864	0.023935
300.0	0.019067	0.023327
400.0	0.018458	0.023124
600.0	0.018864	0.023327
1000.0	0.019270	0.023124



We see that the Random Forest Classifier for n = 400 provides the smallest error. We, thus, look at the error metrics and the confusion matrix for this case

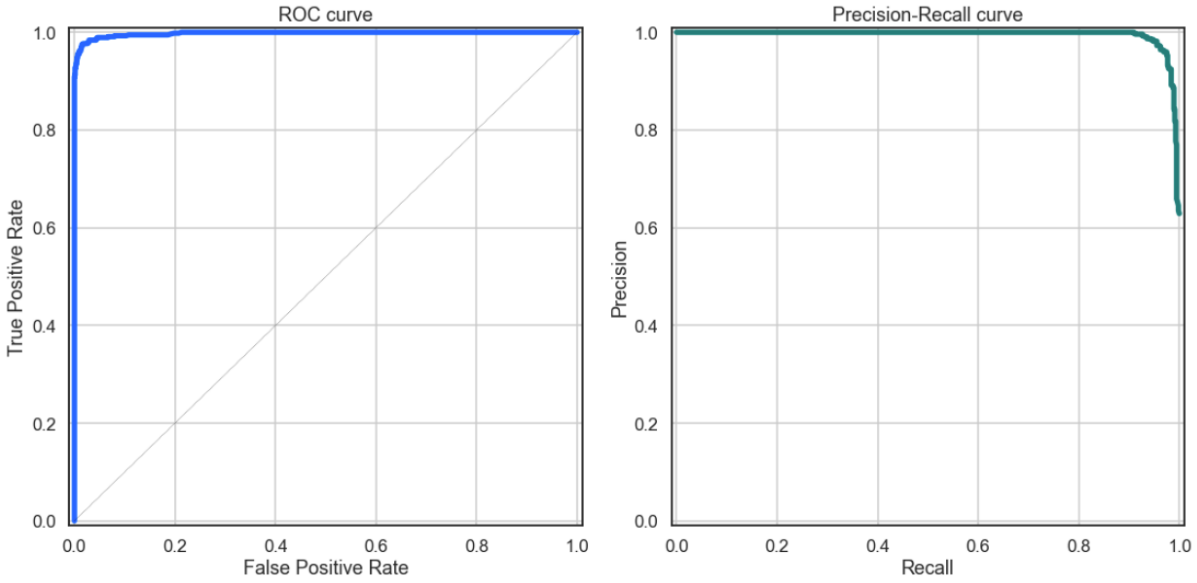
	Random Forest
precision	0.981101
recall	0.981070
fscore	0.980947
accuracy	0.981070



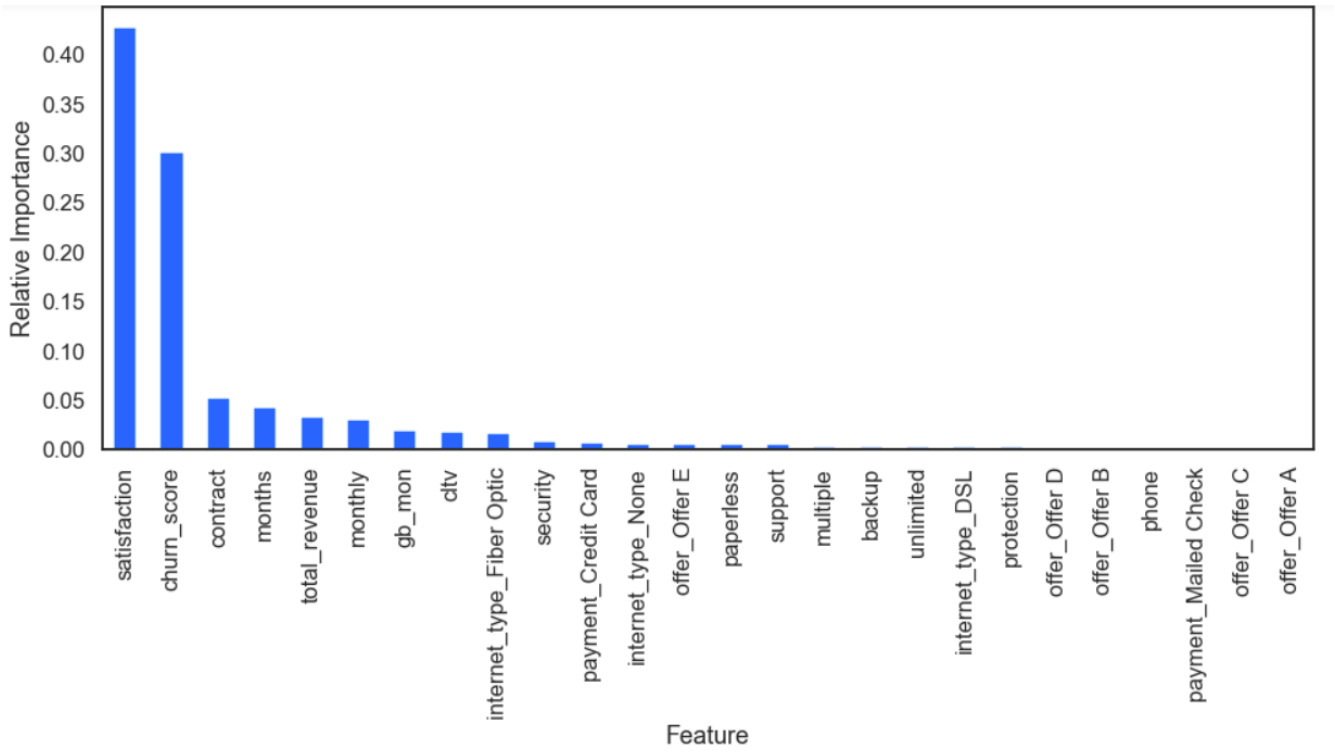
We find that the Random Forest Classifier for number of trees equal to 400 has the best error metrics of all the classifier methods we used. This is not very surprising because the Random Forest Classifier lowers the error metrics by using ensemble methods. We will thus use this method to extract more details about our dataset and to figure out the most important features that can help the company rightly predict the churn value of a customer.

Key Findings

The AUC value for the Random Forest Classifier for n = 400 is given by 0.97. This is a high value and we confirm this by looking at the ROC and the Precision-Recall curve.



This suggests the model provides a very accurate description of the actual churn value of the customers of the company. We thus look at the predictions of the model by specifically looking at the relative importance of the coefficients in the histogram below



We see that the most important feature in predicting the Churn Value of a customer is the satisfaction score. This was expected from the pair plot we saw early on. A high satisfaction score makes it extremely unlikely that the customer will churn while a low satisfaction score makes the churning seem very likely.

The second most important feature is the Churn Score as assigned by the company to various customers. This is the internal assessment of the company, and we see that it is actually less important than the satisfaction score as given by the customer.

The other features are significantly less important in deciding the customer Churn Value. While some features like the type of offer the customer has, the method of payment, multiple connections, security, backup, and phone are almost irrelevant in deciding the Churn Value.

Thus, this dataset gives us a lot of valuable insights into the customer base and helps us better understand the features based on which we can predict their Churn Value which can in turn help us improve our business so as to ensure we retain a higher fraction of our customers.

Next Steps

While we tried a variety of classifiers on our dataset to find the most accurate one, there are multiple other ways in which we could have used our classifiers. One such possibility is to use a Support Vector Machine Classifier to find the decision boundary between various features that correspond to the two Churn Values. Specifically, we could work out the decision boundary between Satisfaction and Churn Score, since these are the two most important features. Similarly, we can try more such features to narrow in on specific features and understand exactly what impact various features have in much more detail.

More data about when a given customer churns could be valuable for our models. This could help us identify if there are certain features through which we can predict if a customer will churn in the short term or in the long term. Additionally, data about the geographical location of the customers could be useful in identifying any systematic reasons for why certain customers might be churning. For example, certain customers might have a low satisfaction score due to a poor network connection in their neighborhood because of which people in those neighborhoods churn more often than others. Similarly, there might be a variety of other reasons that might be driving a low or a high satisfaction score and other features which would help us explain the satisfaction score will be valuable additions to our model.