Machine Learning Engineer Nanodegree

Capstone Project

Noreen Abd ALLAH
October 23, 2018

I. Definition

Project Overview

Telecom industry is a highly competitive industry in today's world, there can be several service providers or companies that are highly competing to keep their customers and prevent them from churning to another company. Every Telecom company now have several varying services that are slightly different from other company's services making it easy for customers to go from one company to another.

with growing pressure from competition, it's becoming extremely important to predict customer behavior to improve customer's retention programs by proactively identifying customers that are about to churn in order to take preventive measures to retain these customers by making targeted retention strategies and offers.

churn rate is the measure of number of subscribers who leave a service provider during a given time period. In order for the Telecom company to maintain and increase it's growth rate; the number of it's new customers must exceed it's churn rate[1].

I have found the dataset on kaggle[2] datasets and it was initially obtained from IBM sample datasets. The dataset contains relevant customer data to predict their behavior and retain customers that have a tendency to churn from the company.

Problem Statement

The growth and revenue of Telecom companies are hugely affected by it's customers churn rate, when a customer leaves a company it causes a huge loss for that company as the costs for initially acquiring that customer to subscribe for the company's services may not have been recovered yet.

My solution for this problem is to build a machine learning model that can predict what class does the customer belong to; whether that customer will churn or not, given his/her relevant information.

I have tried many models like Random Forest, Ada Boost, Light GBM and, XG Boost and picked the best performing one which is Ada Boost and then further tuned it and eventually it has been evaluated using some evaluation metrics like type I Error and ROC curve which will be further discussed in the metrics section. some of these models need the dataset to be numeric or scaled so a preprocessing step with performed prior to modeling in order to prepare the data to be computed successfully with these models.

Metrics

I have chosen Type I error as my main evaluation metric that I chose the final model based on it. Type I error (False Negatives) is failing to identify a customer who has a high tendency to churn[3] so, business wise this is the least desirable error as failing to identify the customers who are going to churn will highly affect the revenue of the company. The model that I have chosen is the one with the least type I Error.

I have also used other metrics like recall, F score (sklearn report), accuracy and ROC/AUC curve for the benchmark and final model just to see how they are performing.

Sensitivity and Specificity

Recall is called sensitivity and it is the proportion of actual positives that are correctly identified while, Specificity is the proportion of actual negatives that are correctly identified. Sensitivity and Specificity measures can give a better insight in the performance of classification models[3].

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

In our case for telecom companies attrition misclassifying a churner as not churning is more costly than misclassifying a customer who is not going to churn as churner, for this reason sensitivity is more important than specificity in this problem. getting a higher sensitivity with a good specificity measures will be very good.

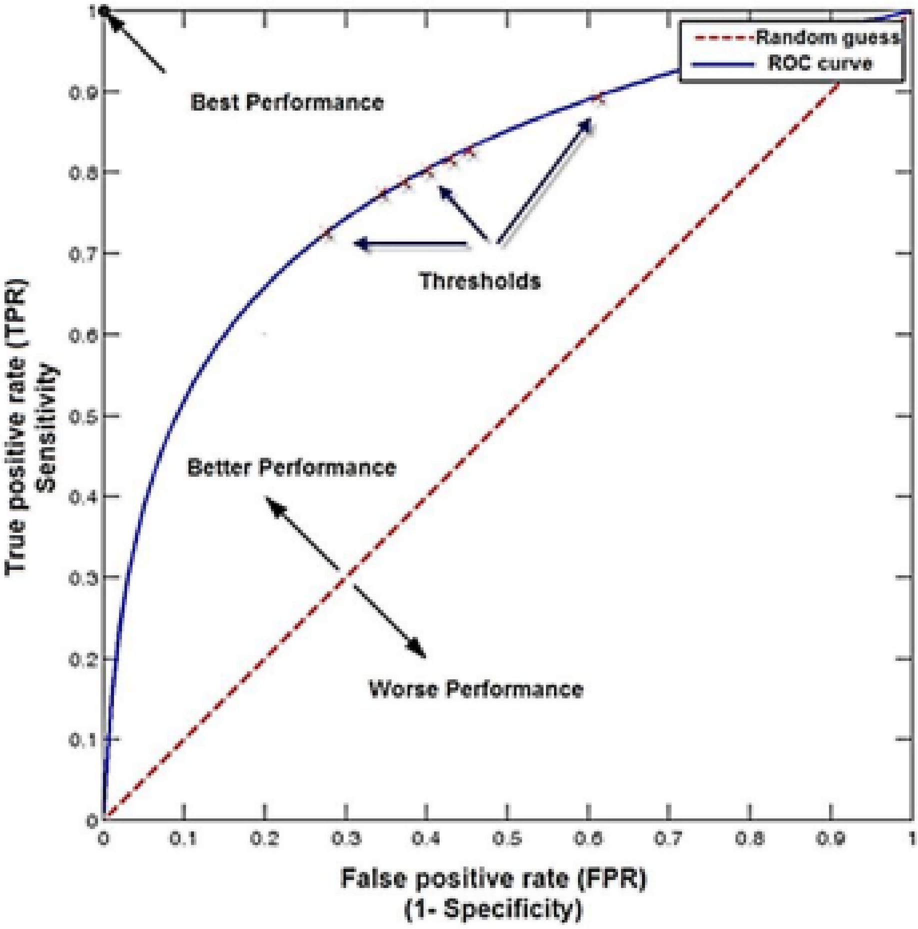
F-Score

The F-score (or F-measure) considers both the precision and the recall of the test to compute the score. Both evaluation metrics are required to adequately assess the performance of a prediction technique.

$$F_1 = \frac{2\times Precision\times Recall}{Precision+Recall}$$

ROC/AUC

- **ROC curve(Receiver Operating Characteristic curve):** The Roc curve shows the relations between the true positive rate and false positive rate so, in our case the ROC represents the relation between the churners ratio correctly predicted as churners, and non-churners ratio wrongly predicted as churners[4]. The ROC curve consists of points corresponding to prediction results. The best performance model is when the ROC curve passes through or close to (0,1) on the x and coordinates. The model sensitivity and specificity will then be 100% (i.e., no false negatives and no false positive respectively).



- **AUC:** Area Under the Curve (AUC) computes the area under the ROC-curve. The AUC value ranges from 0.0 to 1.0. Models perform better when having greater AUC, A random classification model has an AUC of 0.5 .

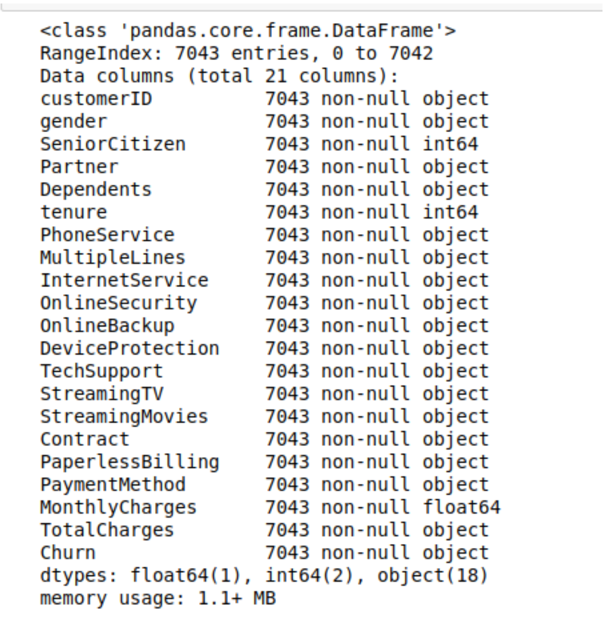
II. Analysis

Data Exploration

The dataset contains relevant customer data to predict their behavior and retain customers that have a tendency to churn from the company.

The data set includes information about:

- Customers who left within the last month – the column is called Churn .
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
- Customer account information – how long they have been a customer, contract, payment method, paperless billing, monthly charges, and total charges.
- Demographic info about customers – gender, age range, and if they have partners and dependents.

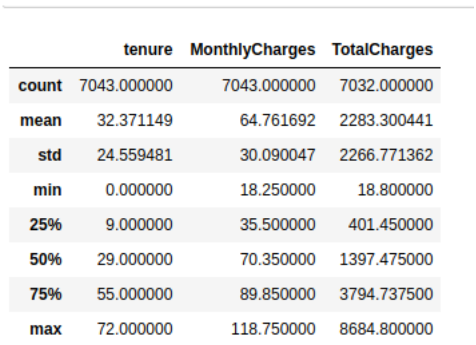


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

- The raw dataset contains 7043 rows (customers) and 21 columns (features).
- The “Churn” column is our target.

- There are no missing values to impute.
- The dataset contains 18 columns of the data type Object which indicated that it will need a lot of labeling and encoding during the preprocessing step.
- The "TotalCharges" feature is of type Object so it needs to be converted into float.

Statistical Insights

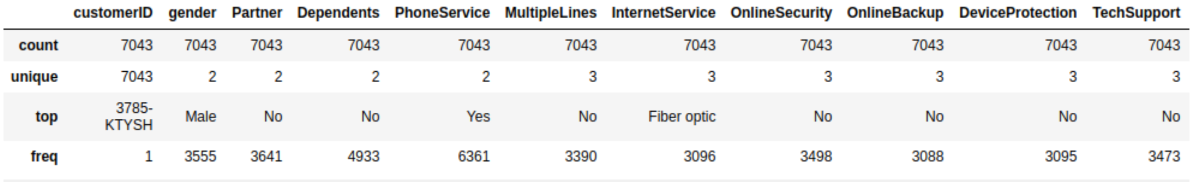


	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7032.000000
mean	32.371149	64.761692	2283.300441
std	24.559481	30.090047	2266.771362
min	0.000000	18.250000	18.800000
25%	9.000000	35.500000	401.450000
50%	29.000000	70.350000	1397.475000
75%	55.000000	89.850000	3794.737500
max	72.000000	118.750000	8684.800000

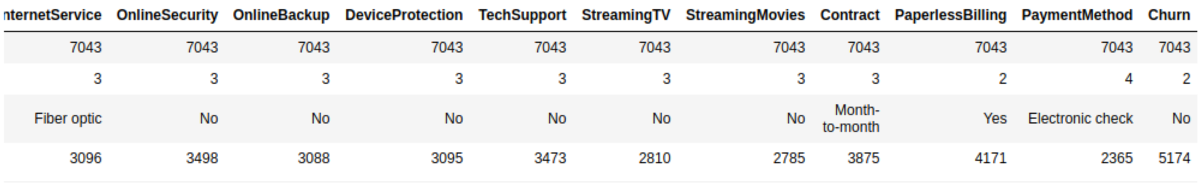
- customers spend an average of 2 years and 8 months subscribing to the company's services and 25% of the customers just churn after 9 months.
- 50% of the customers spend 70$ monthly charges and 3794 total charges.

Categorical features Description

a view of the number of values, number of unique values, top value and it's frequency.



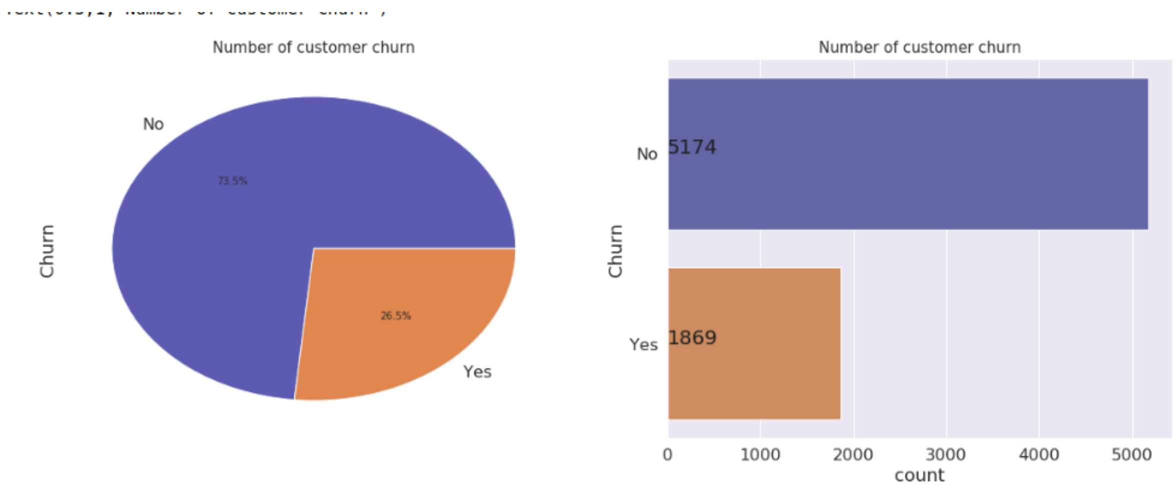
	customerID	gender	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport
count	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043
unique	7043	2	2	2	2	3	3	3	3	3	3
top	3785-KTYSH	Male	No	No	Yes	No	Fiber optic	No	No	No	No
freq	1	3555	3641	4933	6361	3390	3096	3498	3088	3095	3473



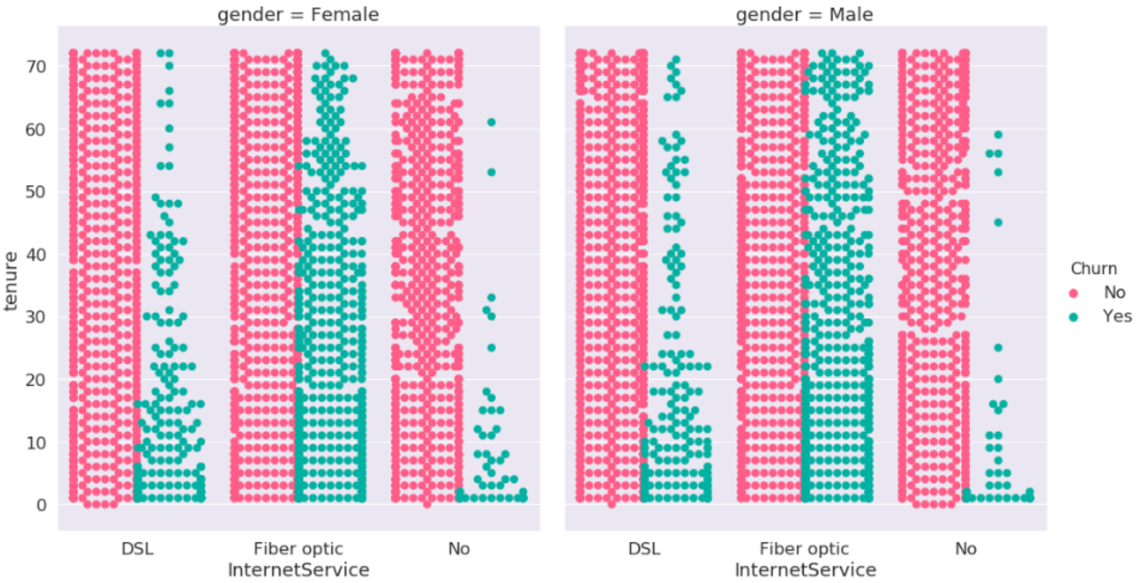
nternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	Churn
7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043
3	3	3	3	3	3	3	3	2	4	2
Fiber optic	No	No	No	No	No	No	Month-to-month	Yes	Electronic check	No
3096	3498	3088	3095	3473	2810	2785	3875	4171	2365	5174

Exploratory Visualization

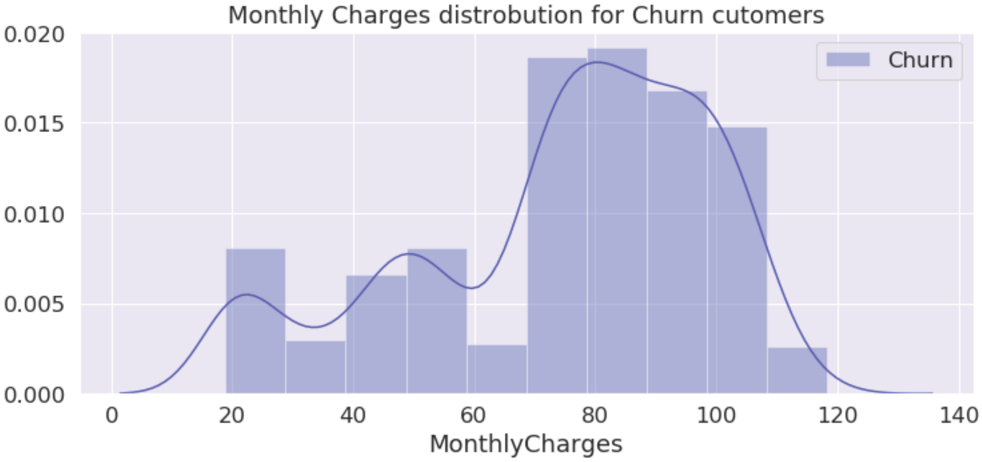
- The dataset contains 26.5% of customers who had churned and 73.5% of customers who haven't churned which indicates that the dataset is partially imbalanced and the model may be a little bit biased towards not churn.



- Customers who have Fiber Optic internet service are more likely to churn which is kind of weird because fiber optics mean that the quality and speed of internet is very high so I am assuming that customers who have Fiber optic internet service have high payments or monthly charges and customers who have high total charges are more likely to churn so, I will investigate this in the following graphs.

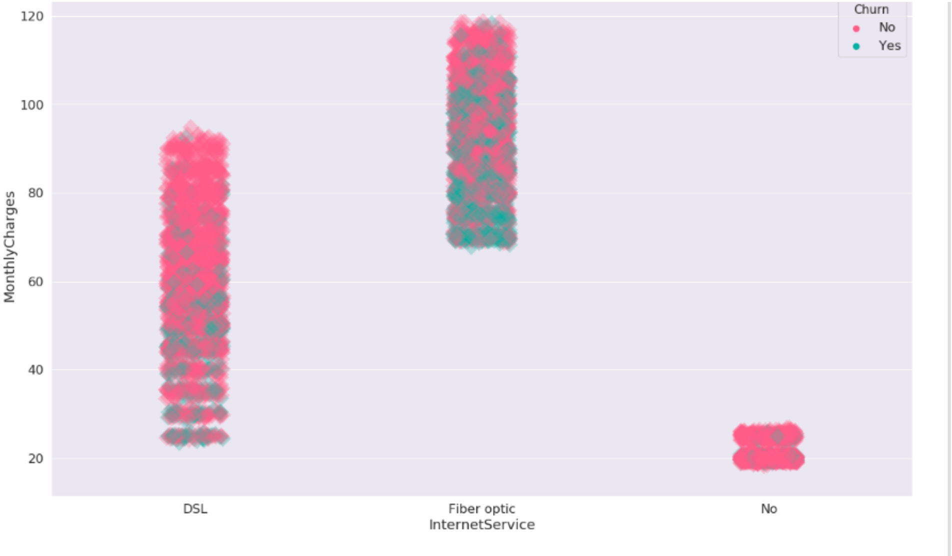


- Customers have higher probability to churn when monthly charges are high which agrees with my hypothesis, now the last part of my hypothesis is that to show that Fiber Optic customers pay high monthly charges compared to other internet services which eventually leads to their churn.

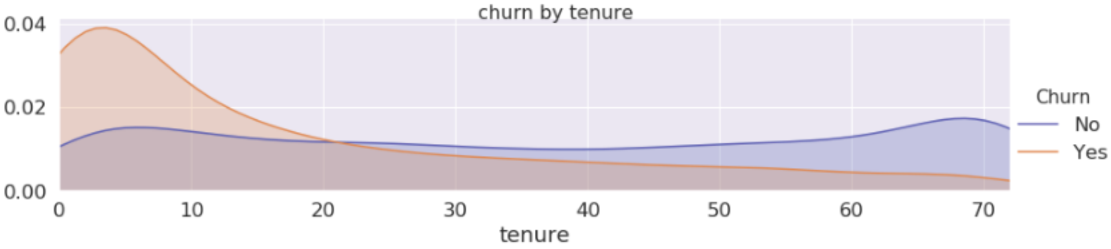


- Now this confirms my hypothesis, as you see the Fiber optic internet service is the most expensive one and customers who have high monthly charges have higher tendency to leave the company, this is why customers who subscribe to the Fiber optic service are more likely to leave the company although they have high quality and internet speed service.

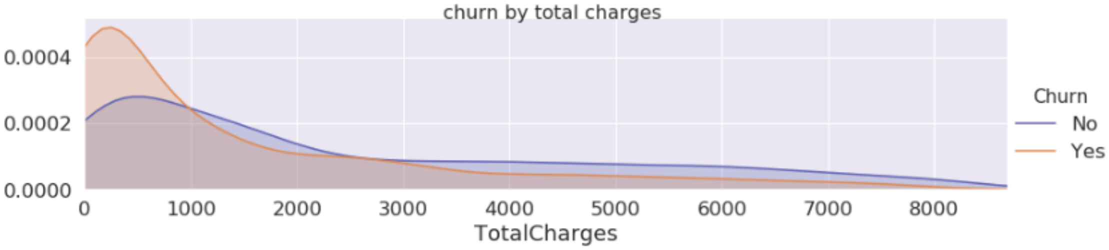
  - This observation is very important for the company as they might target this customer segment and try to make offers to decrease the monthly charges on the fiber optics service or suggest them to use DSL for example if they can't afford the fiber optic service for so long.¶



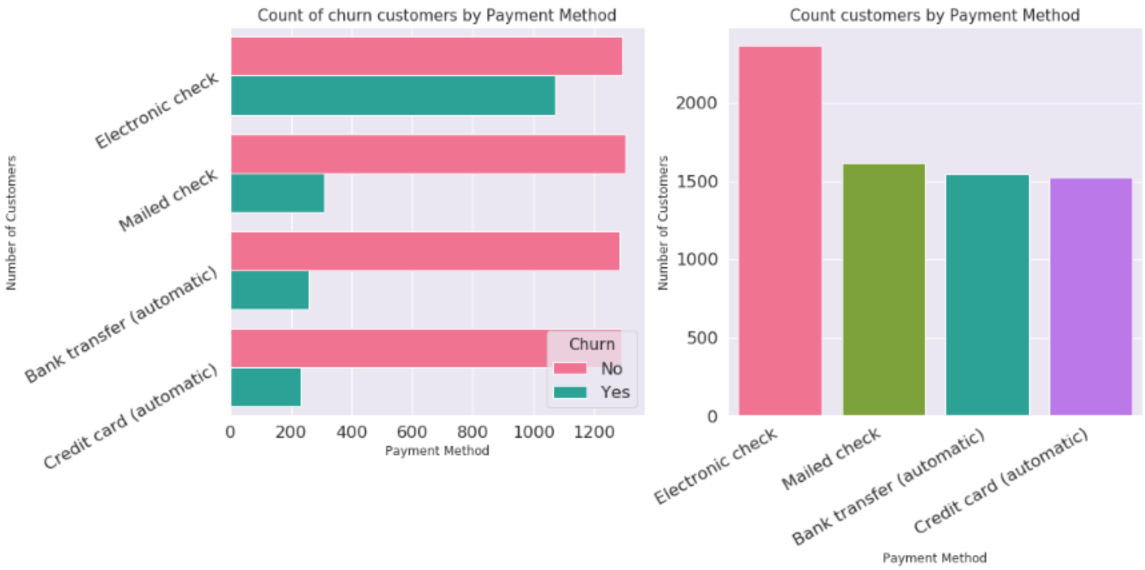
- customers tend to churn after the first few months of their subscription so, the company may need to target new customers with marketing campaign and offers.



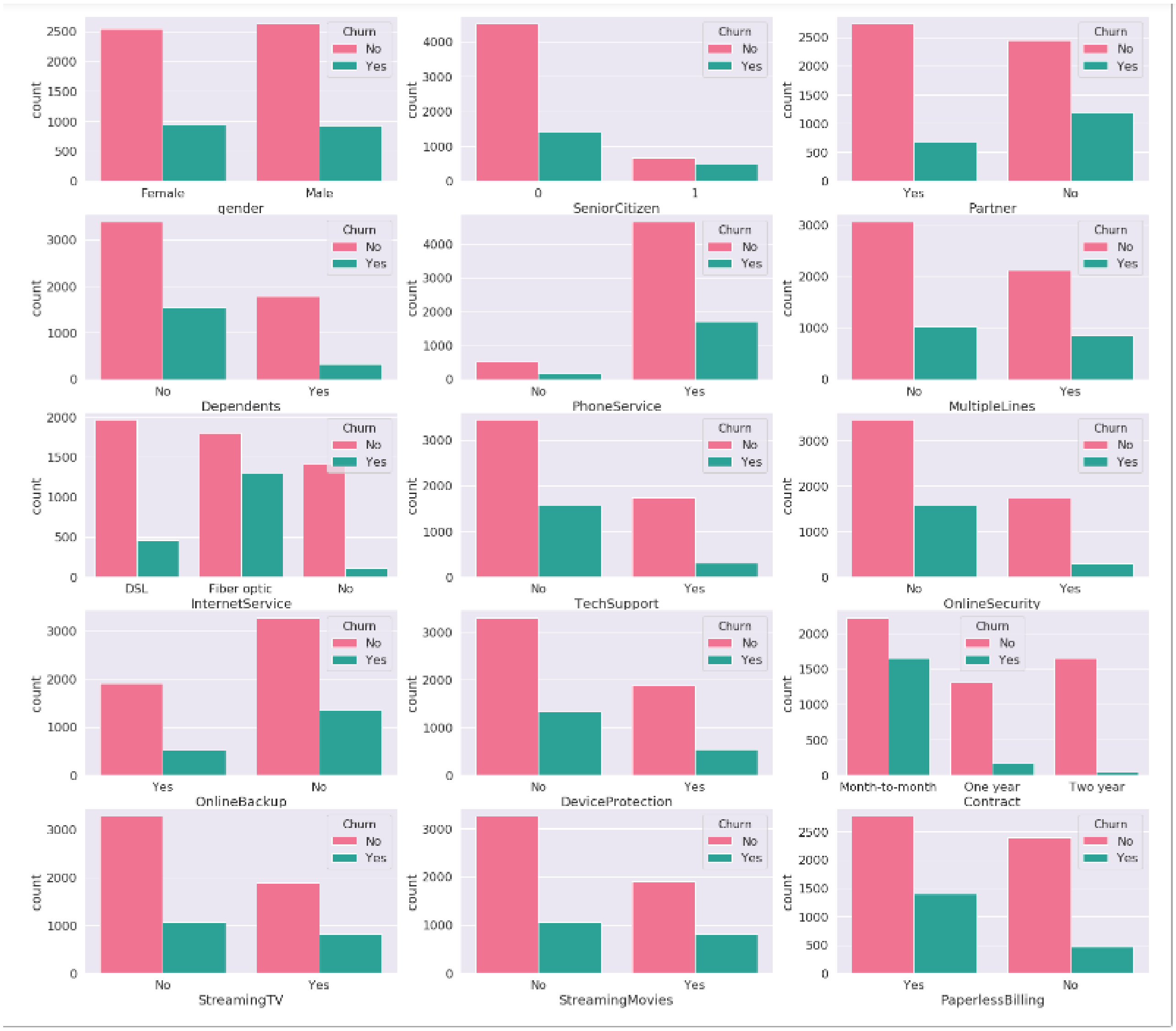
- customers with low Total charges are more likely to churn.



- customers who pay by Electronic check have a high tendency to check



- gender: Females have a very small tendency to churn other than men, I guess there is no much impact for the gender on churn.
- SeniorCitizen: Seniors are more likely to churn.
- Partner: customers who have no partners are more likely to churn.
- Dependents: having dependents makes customers less likely to churn.
- MultipleLines: having multiple lines makes customers more likely to churn.
- Contract: Month_to_Month contracts are more likely to churn.
- PaperlessBilling: A lot of customers who have paperless billing have churned so, the company need to investigate it's system searching for the problem.



Algorithms and Techniques

The dataset is partially imbalanced, it's of high dimension and it consists of 7043 data point so, according to these characteristics i chose to use ensemble classifiers as normal classifiers actually suffer from high dimensionality and the enormous size of the data to be classified, although the this dataset is not that big but, we need to generalize will for future events for the telecom company.

Researchers who has focused on the problems specific to telecom churn prediction have found that ensemble classifiers are considered as better performers compared to individual classifiers. In the light of this I will compare four models (Random Forest, Ada Boost, Light GBM and, XGBoost) and choose the one with best performance that would be the one with the least Type I Error.

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model.

Random Forest

Random Forest is a supervised learning algorithm. They create a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

This works well because a single decision tree may be prone to a noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results.

Random forests does not overfits it uses the bootstrap aggregation or bagging which reduces variance.

AdaBoost

AdaBoost is a supervised ensemble classifier that's usually used for binary classification. AdaBoost uses the boosting ensemble method, in fact it was the first proposed method to use boosting. Boosting is a general ensemble method that creates a strong classifier from a number of weak classifiers[5]. It focuses on classification problems and aims to convert a set of weak classifiers into a strong one. This is done by building a model from the training data, then creating a second model that attempts to correct the errors from the first model. Models are added until the training set is predicted perfectly or a maximum number of models are added.

Light GBM

Light GBM is a gradient boosting framework that uses tree based learning algorithm.

Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms[6]. The disadvantage of light GBM is that it can be prone to overfitting as it grows trees leaf-wise. There is also something to consider when using Light GBM is that it needs large datasets in order to come up with a good performance which may be a trouble for the telecom company if it needs these predictions to be on a daily basis.

XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance[7]. when working with actual data in the telecom sector there may be other features like transactions and time that make a large dataset that needs to be processed in a meaningful time, XGBoost will be very suitable for this as it makes use of parallelization of tree construction using all of the CPU cores during training. By talking about the performance XGBOOST uses the gradient boosting approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.it is also the go-to algorithm for competition winners on the Kaggle and Analytics Vidhya.

Benchmark

My Benchmark model is a plain Logistic regression model with it's default parameters value, it can be used as a binary classification model and it is the most commonly reported data science method used at work for all industries according to kaggle's The State of Data Science & Machine Learning survey so I thought it would be interesting to outperform it and beat it's score.

III. Methodology

Data Preprocessing

- Removing features that doesn't contribute to the prediction of target variable like the customer ID.
- Converting categorical object features into numeric labels.
  - most of the categorical objects in this dataset are put as a string labels e.g(Yes, NO) but, there are models used that cannot operate on label data directly. They require all input variables and output variables to be numeric e.g(1,0). and in order to achieve that we can get numeric labels using sci-kit Learn's label encoder.
- Getting dummy variables for categorical columns of more than two values.
  - For categorical variables where no such ordinal relationship exists, the label encoding is not enough it may assume a natural ordering between categories which may result in poor performance or unexpected results. To avoid this problem we use one hot encoding or pandas get dummies which which transforms each categorical feature with n_categories possible values into n_categories binary features, with one of them 1, and all others 0 i.e the integer encoded variable is removed and a new binary variable is added for each unique integer value.
- feature scaling.
  - Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without scaling which is a method used to standardize the range of independent variables or features of data. in order to scale the dataset using standard scaling that standardize features by removing the mean and scaling to unit variance.

  $$z = \frac{x-m}{s}$$

  where m is the mean (average) and s is the standard deviation from the mean, z is the standard scores (also called z scores).
- splitting dataset into training and testing sets.

Implementation

some of the algorithms that i will experiment needs all the data to be numeric, this was handled during the preprocessing step and it was mentioned in the above section.

preprocessing is the actual first and most important step that needs to be handled carefully and with patience as it takes 80% of the work time on the project, preprocessing the data well leads to a better results in the prediction of any model. After analyzing the data to explore trends or it's characteristics, I have preprocessed the data to be ready and suitable for modeling and prediction. The final output of the preprocessing phase is a clean dataset that's labeled numerically and encoded without any unnecessary feature that doesn't contribute to the prediction of the target variable.

The dataset is then divided into 75% for training and 25% testing data. The features has then been.

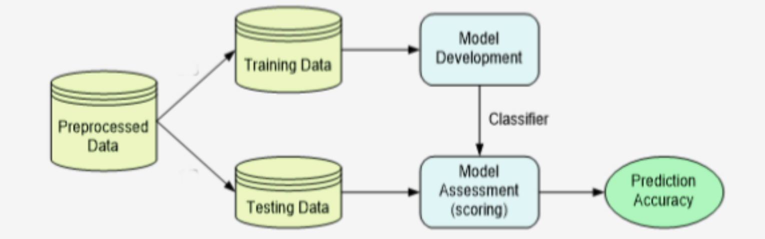
I have then built the benchmark model which is a logistic regression model, it has been trained on all of the training set and it got a **type I error of 232** that is even without cross validation.

the main metric that I evaluate the model on, is the **type I error,** the model that will be deployed in the company needs to have the least number with type I error. After that I have implemented these four algorithms (Random Forest, Ada Boost, Light GBM, XG Boost) without any modification in the defaults parameters. The last step was to choose the model with least type I error and further optimize using grid search cross validation(this will be discussed in detail in the following section). Here is an overview of the implementation process:

complications

- Grid search cross validation takes a lot of time to be done.

- It is hard to cross validate many parameters without a GPU.



Refinement

After training and evaluating different algorithms with their default parameters the model with the least Type I Error was chosen for further tuning and optimization.

Model	Type I Error
Ada Boost	224
Light GBM	228
XG Boost	238
Random Forest	275

As you can see the Ada Boost got the least Type I Error with value 224 so It's time to apply Grid Search cross validation on it in order to optimize it and get the best performing parameters.

The parameters that I was performing Grid search on were:

- n_estimators:
  - 50
  - 100
  - 300
  - 500
- Learning rate:
  - 1
  - 0.0001
  - 0.5

I have made sure to include the default parameters as well in case that they are the model is performing optimally with them, and surprisingly the result were that the default parameters were actually the best ones among the rest of the values.

So the best parameters were n_estimators=50, and learning rate of 1, and the result of the accuracy and Type I Error doesn't change.

It got an accuracy of 80.12%, Type I Error of 224 and ROC score of 71.70%

Of course better results could have been achieved but that needs a lot more time to train and try all different combinations of parameters.

what is also good about Grid Search it that it also perform a cross validation which solves the variance problem where the accuracy obtained on one test is very different to accuracy obtained on another test set using the same algorithm[9]. I've used cross validation of 10 folds.

IV. Results

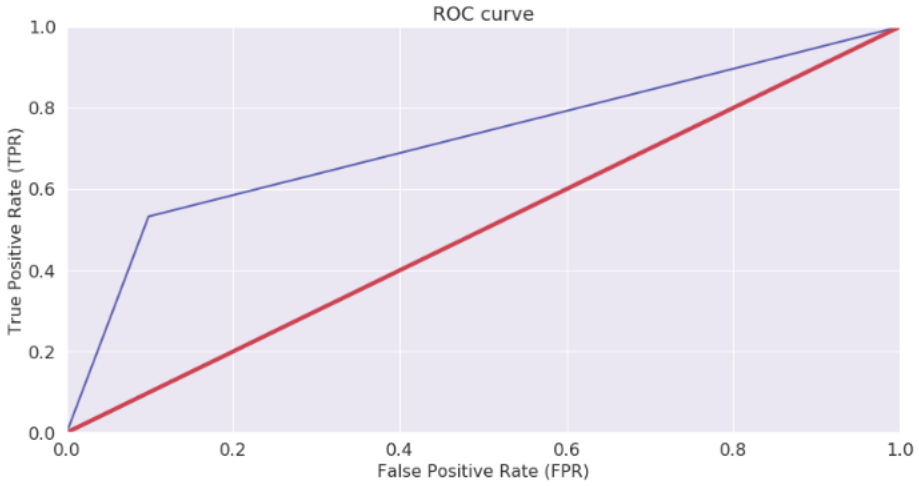
Model Evaluation and Validation

All of the models have been evaluated and validated using the accuracy and type I error.

The best performing model in terms of the type I error is Ada Boost. It was surprising since I thought that XG Boost or Light GBM would perform better. The robustness of the model was measured with ROC curve. Here are the final results.

Ada	Boost
Metric	Measure
Type I Error	224
Accuracy	80.12%
Roc Score	71.70%
Precision	0.79
Recall	0.80
f1-score	0.79
support	1761

The ROC/AUC indicates by how much the model performs better than a random guess. the red line of AUC = 50% represents a random guessing model. as we go higher our model becomes more robust and trusted. The Ada Boost model has an AUC of 71.70%.



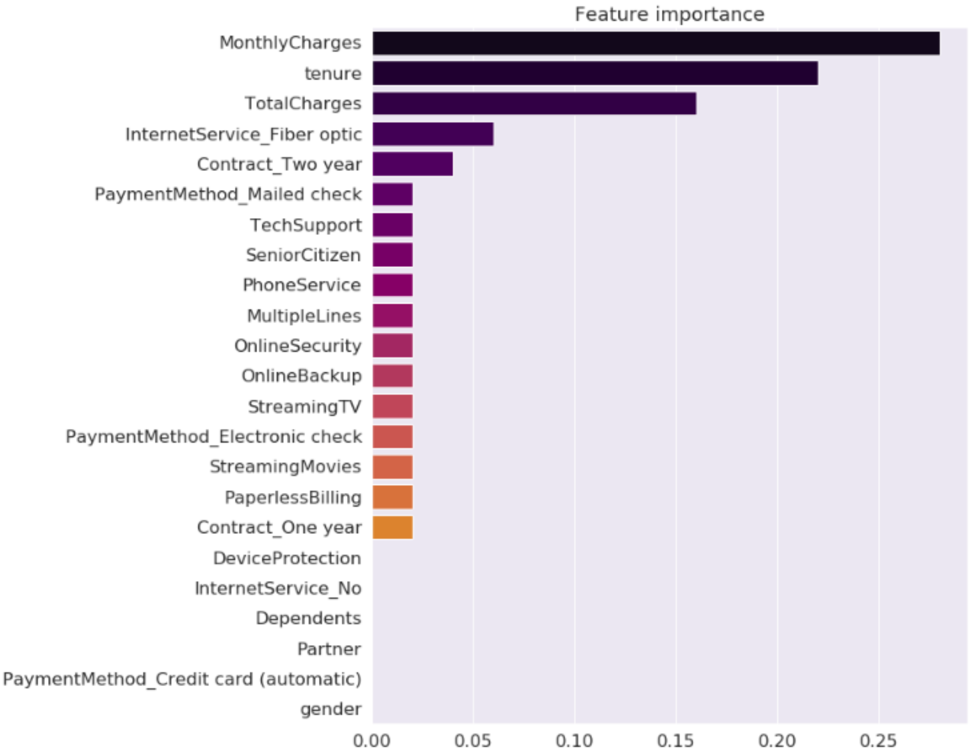
Justification

My Benchmark model is a plain logistic regression model that has a type I error of 232 and accuracy of 80.58%, while it has an accuracy higher than the proposed Ada Boost model by only 0.46% the Ada Boost model on the other have beaten it's type I error and got 224, also a difference of 0.46% is not a significant difference and it can be tolerated.

V. Conclusion

Free-Form Visualization

Plotting Ada Boost feature importance.



For Ada Boost the most important features are MonthlyCharges, tenure, TotalCharges, and InternetService_Fiber optic which confirms my analysis.

Reflection

The process used for this project can be summarized using the following steps:

- Getting the dataset.
  - in this project the dataset was already collected and put in suitable format on IBM or kaggle. In real life we would need to do data acquisition from the telecom company's data warehouses.
- Exploratory data analysis.

- Visual data exploration and analysis to discover trends and patterns in the behavior of customers.
- Data preprocessing to prepare it for modeling.
- building benchmark model and evaluate it.
- building proposed model and evaluate it.
- model validation.
- model optimization and hyper-parameter tuning.

At the end we will end up with labels labeling customers for whether they will churn or not for the company to set their retention strategies targeted exactly to the customer that have a propensity for churning.

During the analysis phase I found the fiber optics results very interesting being the internet service with the highest churn rate that made me very curious to investigate more for why would a good service be related with churners which I have found out that it's because of the high monthly charges, this would be so valuable for the telecom company to consider when putting out payment plans. Another thing that really surprised me was that Ada Boost has beaten XG Boost in the type I Error rate which was quite interesting an unexpected, I know XG Boost can possibly beat Ada Boost with some parameter tuning but, I also expected that it will also perform better with it's default parameters value.

Improvement

Improvements that will try do in future works:

- Doing feature reduction with PCA and compare results with and without feature reduction.
- Grid Search with more values on a GPU accelerated machine.
- Ada Boost doesn't have many parameters to tune so I think although it has performed better than XG Boost and Light GBM with default parameters, I think they can beat Ada Boost using Grid Search as they have so many parameters to tune which can be a pro and a con at the same time but it's worth experimenting.
- combining all 4 models into a custom ensemble model using model stacking.

References

[1]Efficient ways for Customer Churn Analysis in Telecom Sector

[2]Kaggle Telco Customer Churn dataset

[3]Customer Churn in Mobile Markets: A Comparison of Techniques

[4]Benchmarking analytical techniques for churn modelling in a B2B context

[5]Building Predictive Models for Customer Churn in Telecom

[6] Boosting and AdaBoost for Machine Learning

[7]Which algorithm takes the crown: Light GBM vs XGBOOST?

[8]A Gentle Introduction to XGBoost for Applied Machine Learning

[9]Cross Validation and Grid Search for Model Selection in Python