


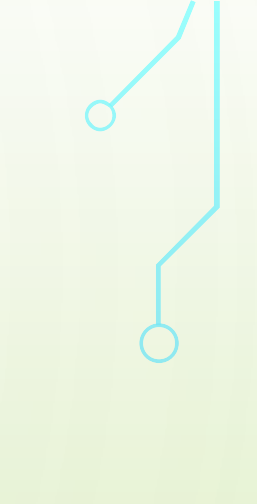
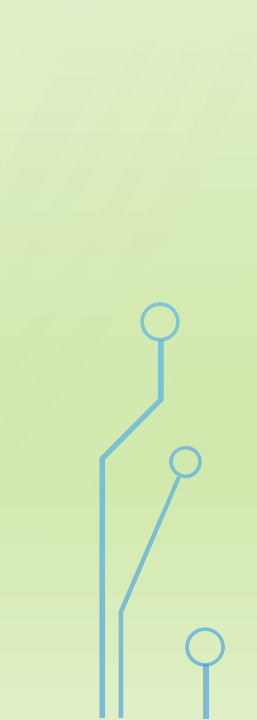
PREDICTING CUSTOMER CHURN

Thanks to Mentor.
Mr. A J Sanchez

By,
Sheema Murugesh Babu


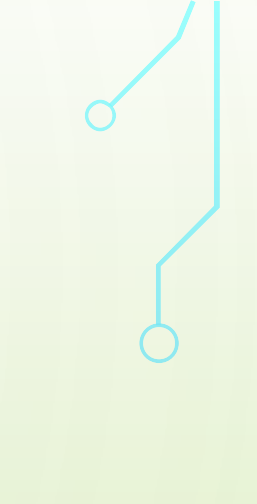
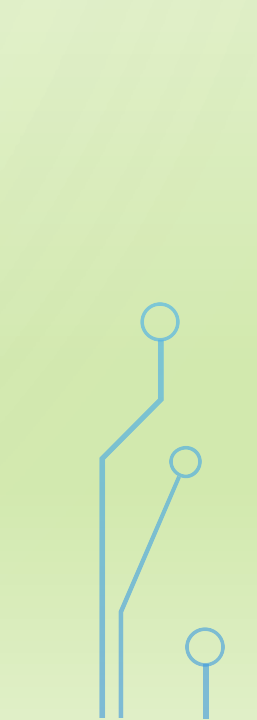


INTRODUCTION

- Churn quantifies the number of customers who have unsubscribed or canceled their service contract. Customers turning their back to a service or product are no fun for any business. It is very expensive to win them back once lost.
 - Keeping the right customers can be quite valuable for a company. Not only because customer acquisition is much more expensive, but as more and more business models are shifting towards subscription plans, a customer can be worth thousands of dollars in future.
 - Reducing churn ultimately leads to a sustainable growing business.
- 
- 
- 



PROBLEM STATEMENT

- The challenge here is to build a model that identifies customers with the intention to leave a service in the near future.
 - The data contains Demographic information like gender, age range, and whether they have partners and dependents, contains customer account information, services that each customer has signed up for and lastly customers who left within the last month – the column is called Churn.
 - My solution looks into building machine learning models to predict customer churn. Given the dataset, the model could estimate whether a customer may or may not be unsubscribed to a service.
- 
- 
- 

DATASET INFORMATION

- Dataset was available in Kaggle's website and was saved into local as 'BlackFriday.csv'
- The data available in the data set has below columns.

Variable	Definition
customerID	Customer ID
gender	Sex of User
SeniorCitizen	Senior Citizen
Partner	Partner
Dependents	Dependents
tenure	Tenure
PhoneService	Phone Service
MultipleLines	Multiple Lines
InternetService	Internet Service
OnlineSecurity	Online Security
OnlineBackup	Online Backup
DeviceProtection	Device Protection
TechSupport	Tech Support
StreamingTV	Streaming TV
StreamingMovies	Streaming Movies
Contract	Contract
PaperlessBilling	Paperless Billing
PaymentMethod	Payment Method
MonthlyCharges	Monthly Charges
TotalCharges	Total Charges
Churn	Churn

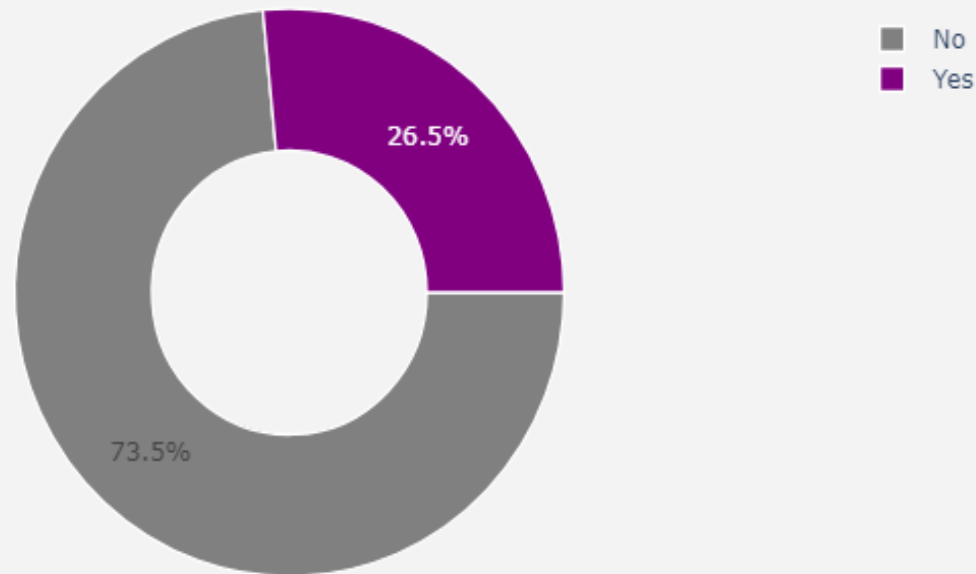
DATA WRANGLING STEPS

- Used pandas read_csv module to import the data set.
- Upon executing the read_csv function using the info() method, the csv file has 7043 entries and 21 columns with different formats of data. There were no missing values.
- Changing the data type of TotalCharges column resulted in error as there were blank spaces in the column. Rectified these errors by replacing them with the mean values of the column.
- Changed the data in few other columns like 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'MultipleLines' and 'SeniorCitizen'. Also added an extra column called TenureGroup to group the data from Tenure column.

DATA STORYTELLING

- After I wrangled and cleaned the dataset, I started to explore the data in detail. To put great visualizations, I used 'Plotly' library.

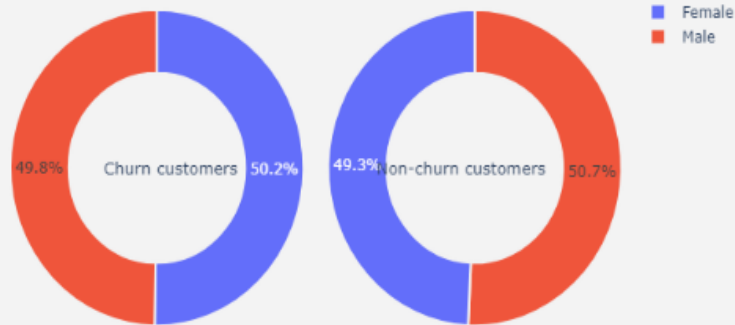
Customer churn in data



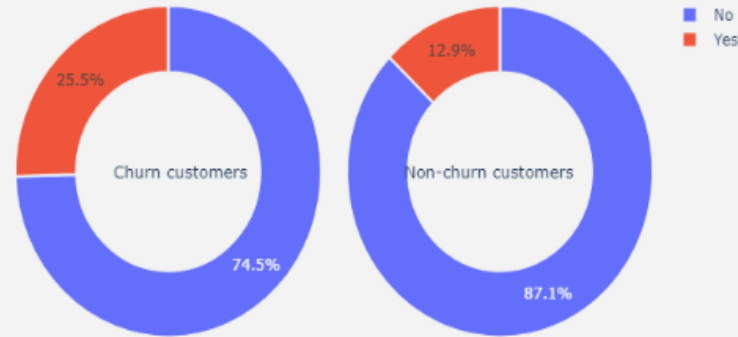
- Only 26.6% of the data represents churn customers and the majority are the non-churn customers.
- We might be dealing with a class imbalance problem as there are more non-churned customers than the churned ones.

Plots for some of the categorical columns

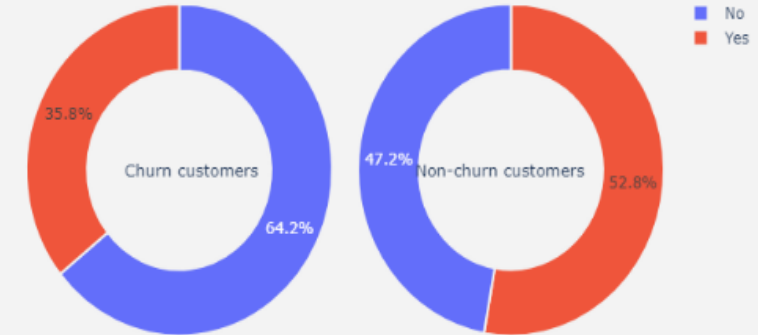
"gender" distribution in Customer Churn



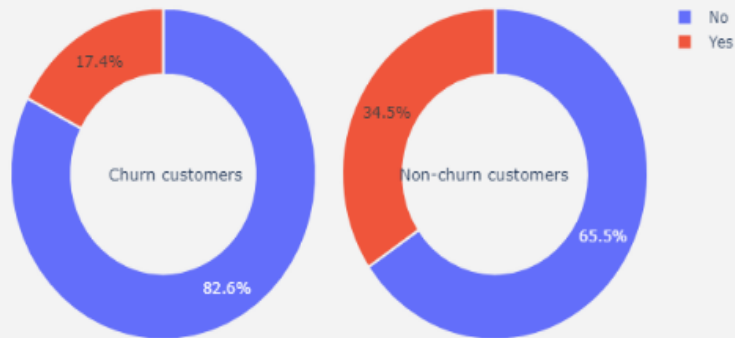
"SeniorCitizen" distribution in Customer Churn



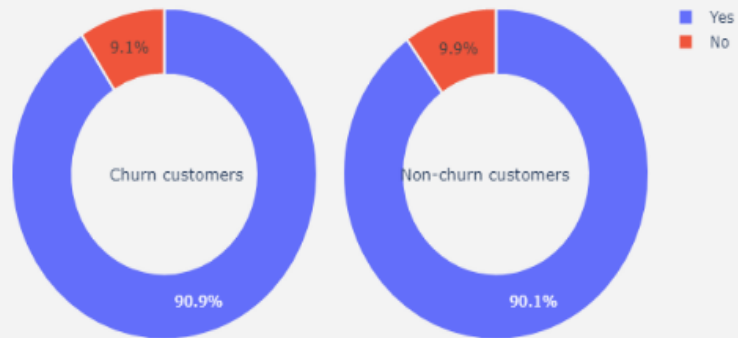
"Partner" distribution in Customer Churn



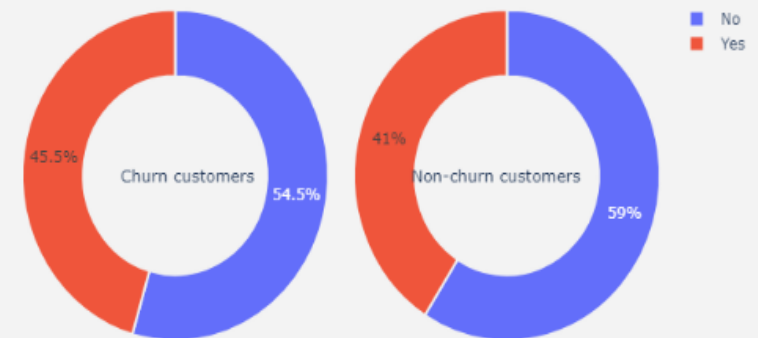
"Dependents" distribution in Customer Churn



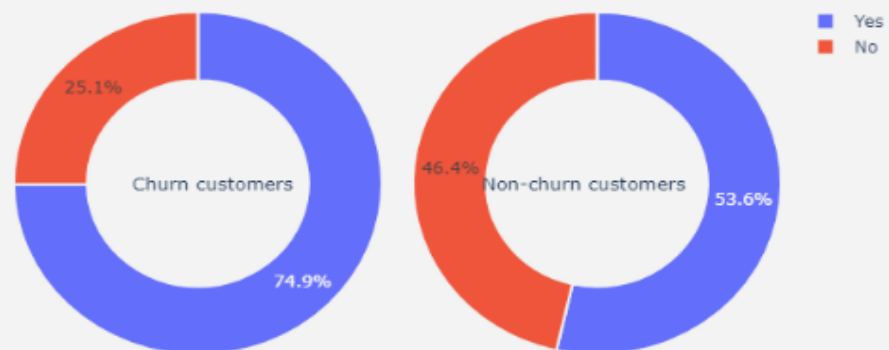
"PhoneService" distribution in Customer Churn



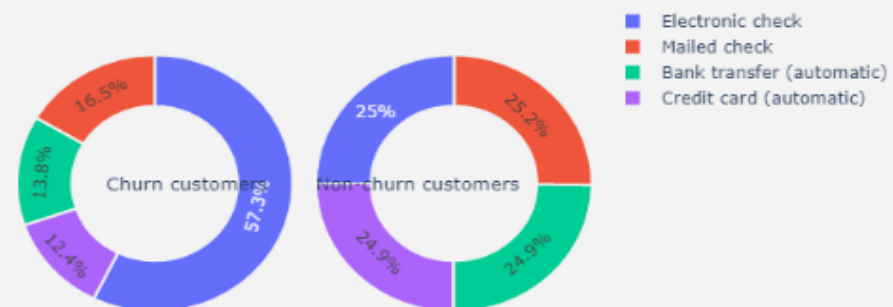
"MultipleLines" distribution in Customer Churn



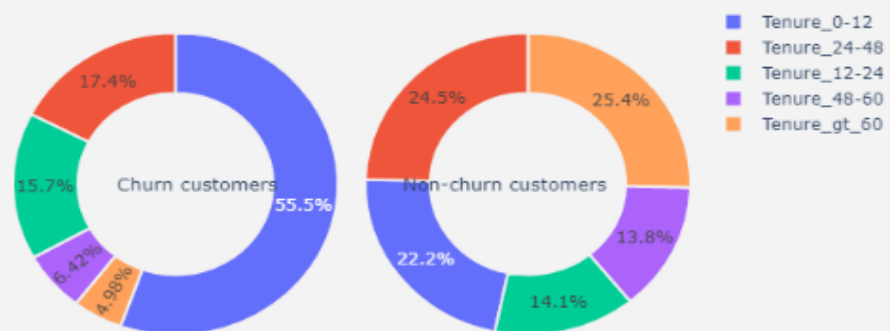
"PaperlessBilling" distribution in Customer Churn



"PaymentMethod" distribution in Customer Churn

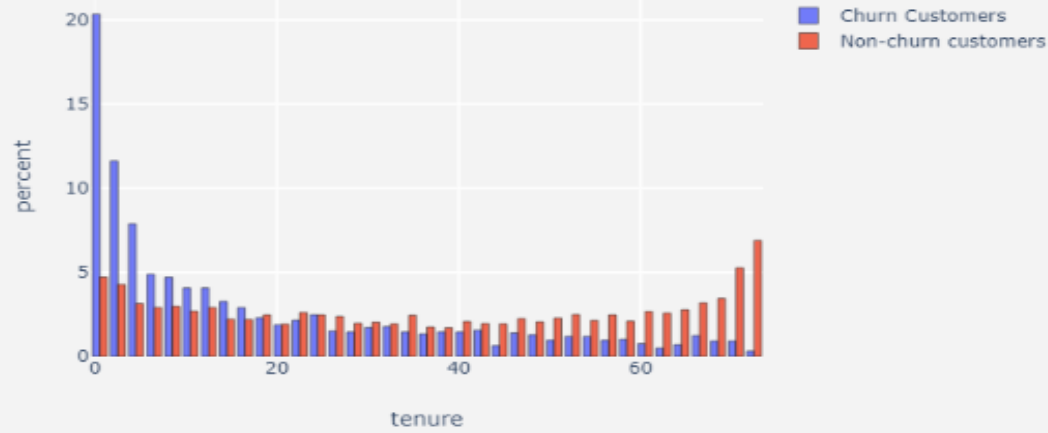


"Tenure_group" distribution in Customer Churn

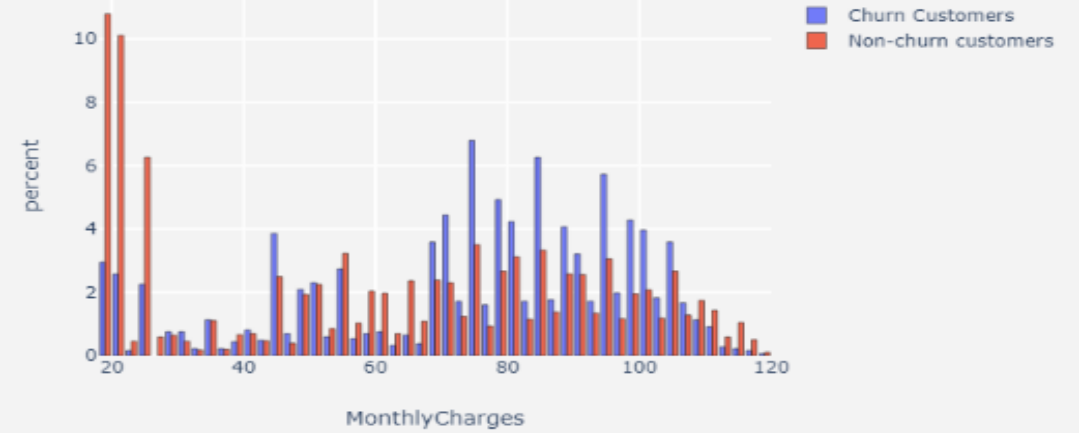


Histograms for all the numerical columns

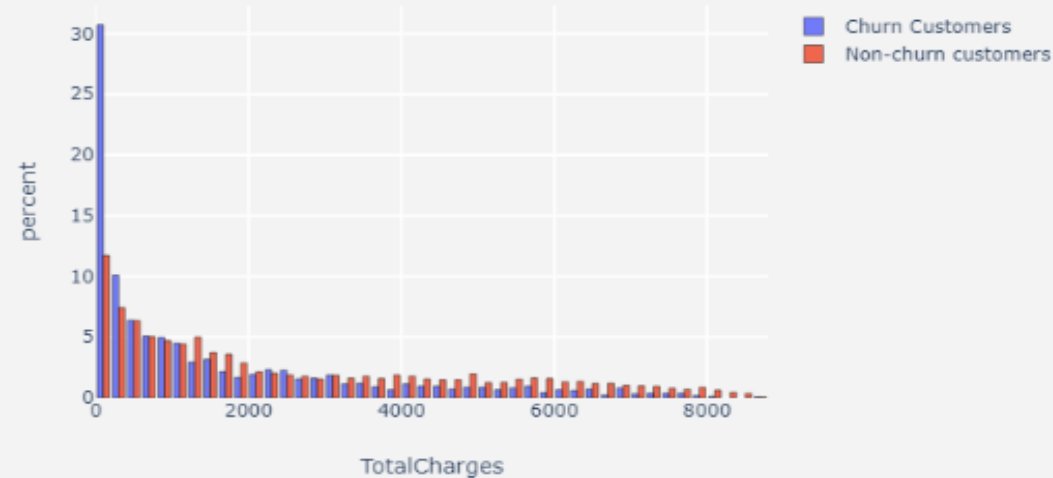
"tenure" distribution in Customer Churn



"MonthlyCharges" distribution in Customer Churn

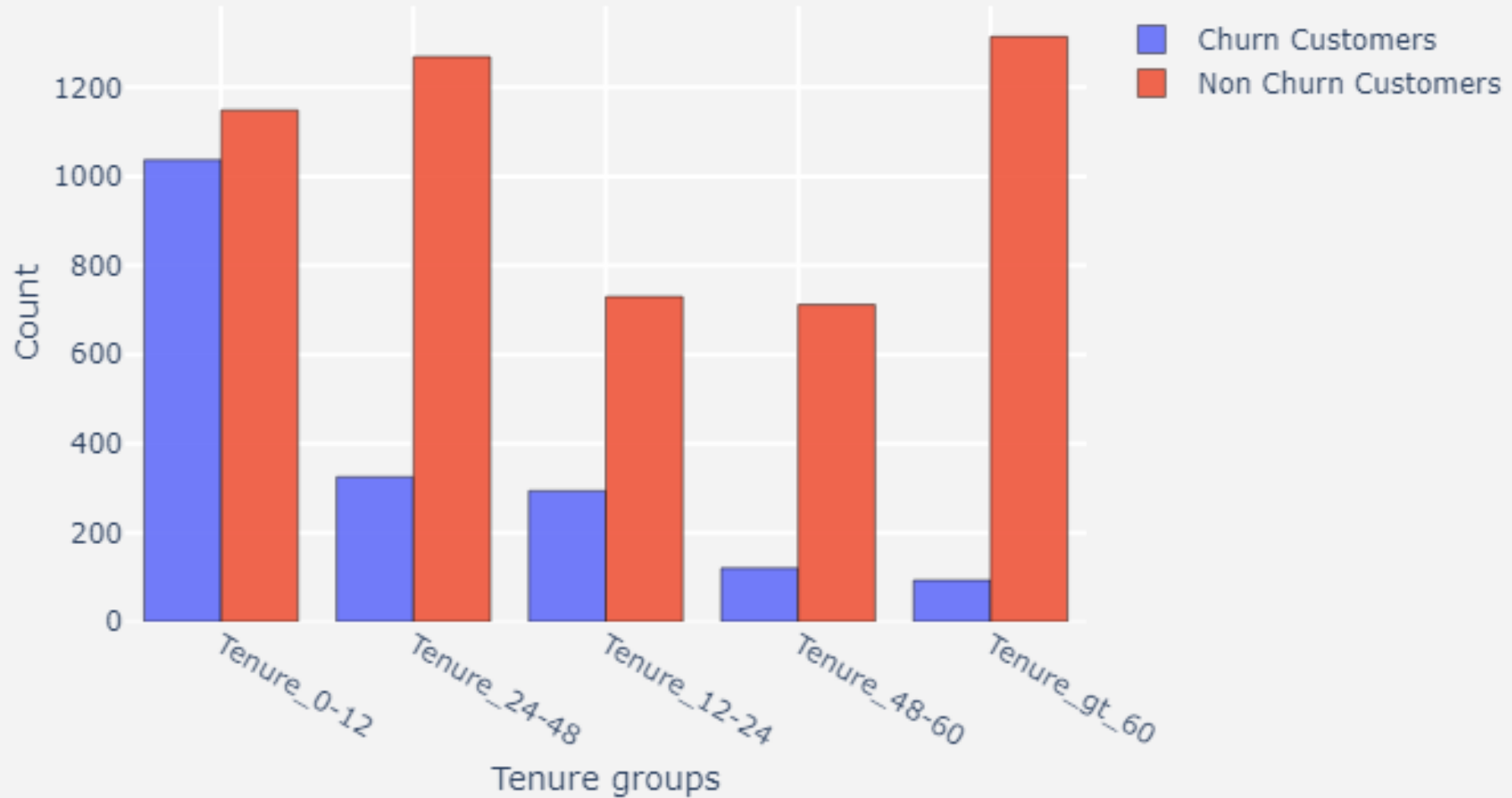


"TotalCharges" distribution in Customer Churn



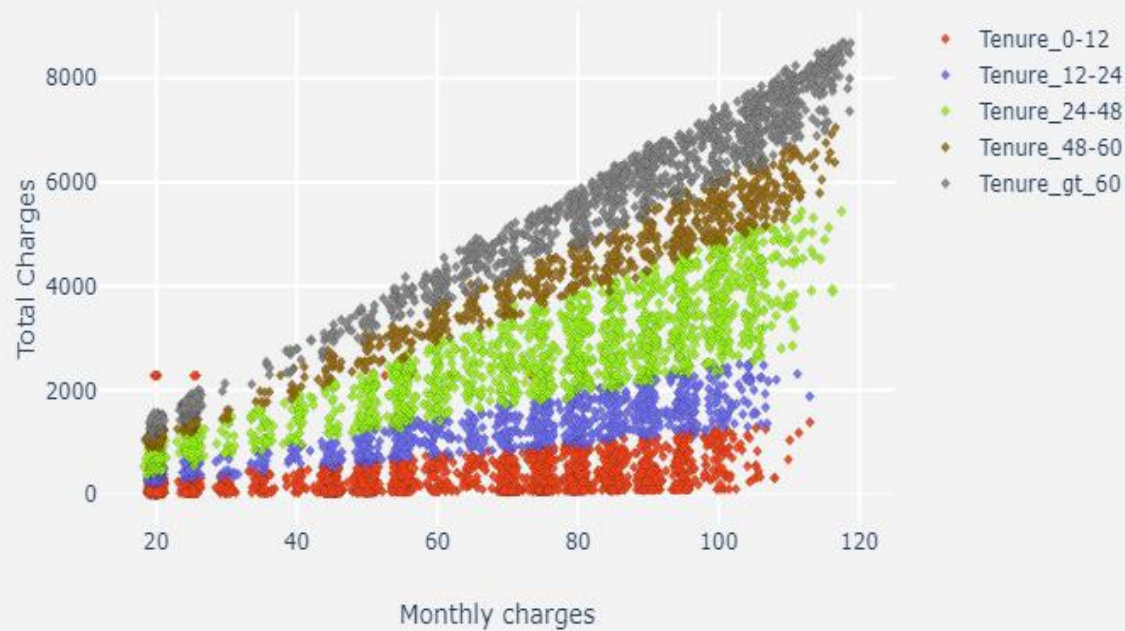
Customer Churn in Tenure groups

Customer churn in tenure groups

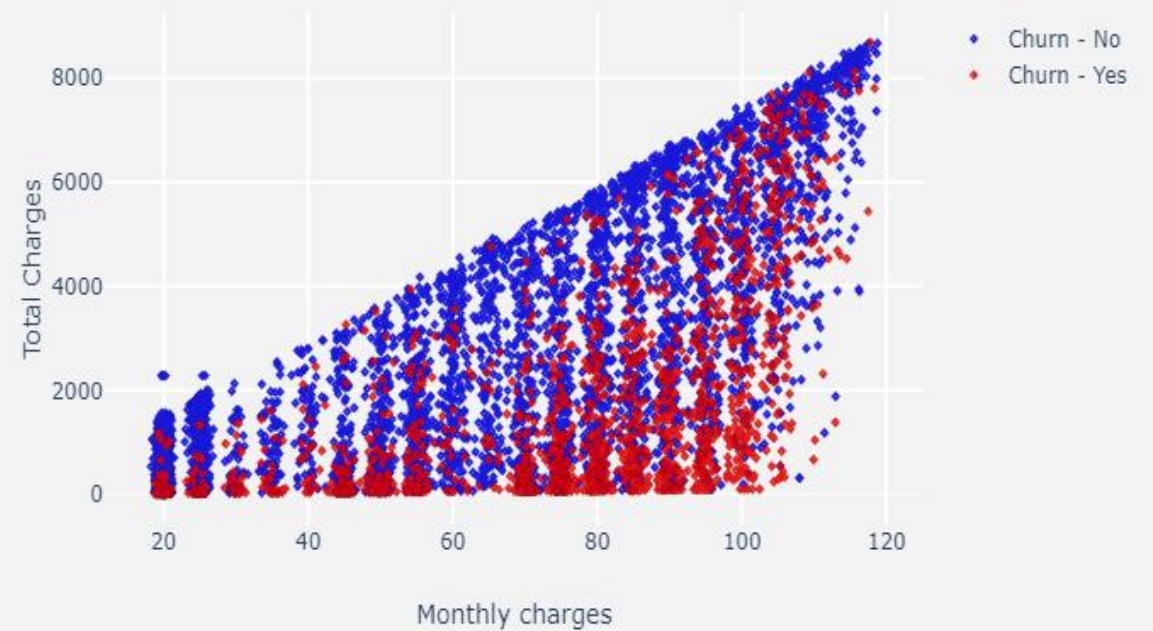


Monthly Charges and Total Charges by Tenure group and Churn group

Monthly Charges & Total Charges by Tenure group

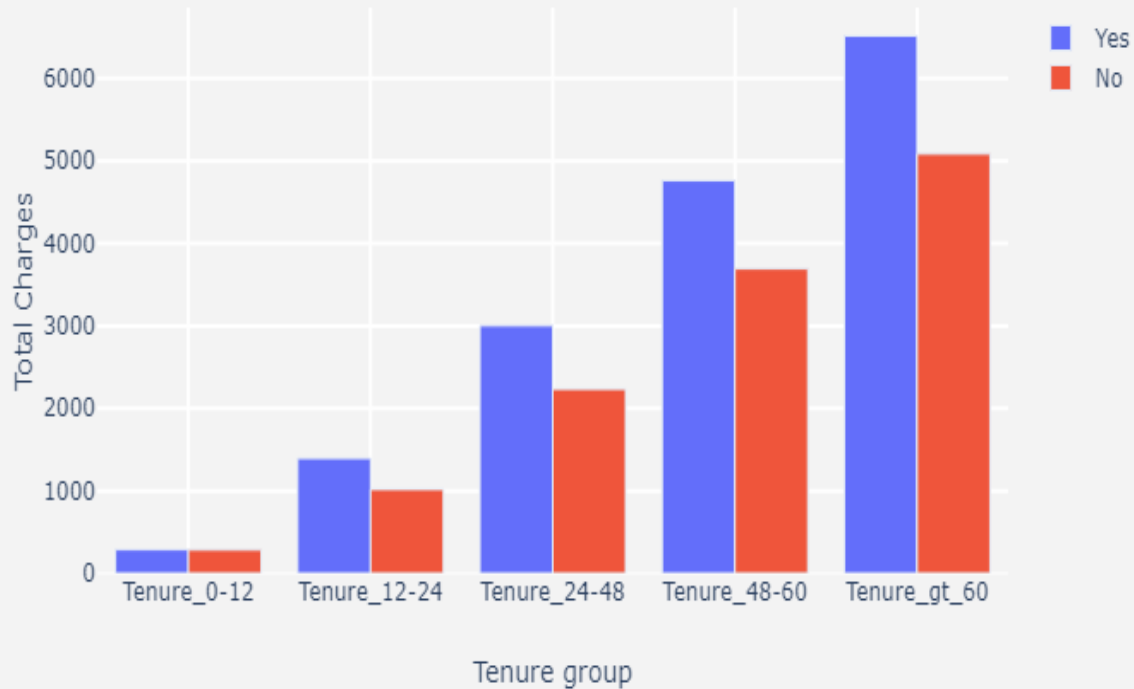


Monthly Charges & Total Charges by Churn group

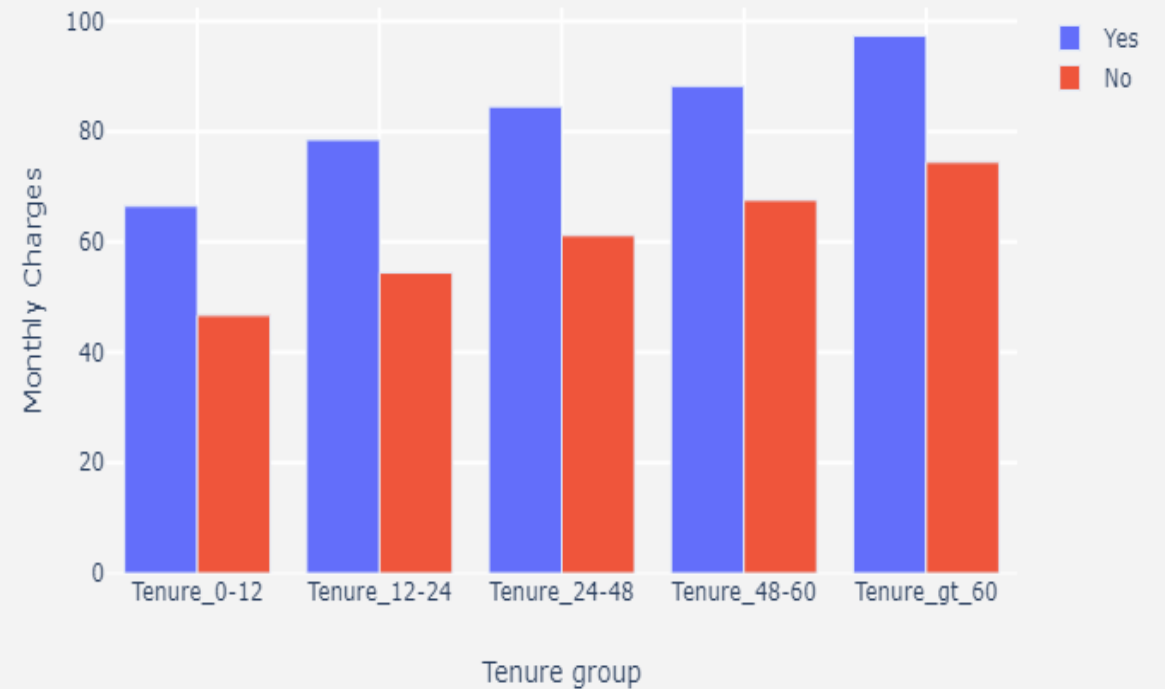


Average charges by Tenure Groups

Average Total Charges by Tenure groups



Average Monthly Charges by Tenure groups



HYPOTHESIS TESTING - CHI-SQUARE TEST

- The Pearson's Chi-Squared test, or just Chi-Squared test is a statistical hypothesis test that assumes (the null hypothesis) that the observed frequencies for a categorical variable match the expected frequencies for the categorical variable. The test calculates a statistic that has a chi-squared distribution, named for the Greek capital letter Chi (χ) pronounced "ki" as in kite.
- The Chi-Squared test uses something called a contingency table, by first calculating the expected frequencies for the groups, then determining whether the division of the groups, called the observed frequencies, matches the expected frequencies. The result of the test is a test statistic that has a chi-squared distribution and can be interpreted to reject or fail to reject the assumption or null hypothesis that the observed and expected frequencies are the same.

Inferential Statistics

- Imported the 'chi2' and the 'chi2_contingency' from the scipy.stats library and wrote a function for Contingency table for all the categorical columns and the numerical columns.
- The function returned a Contingency table for each categorical/numerical column against the target variable i.e. Churn, degrees of freedom, expected values, the test statistics such as Probability, Critical values, Chi-square statistic, significance and the p-value.
- If the p-value < 0.05 , it would mean there is a relationship between the 2 categorical variables.

Table shows the significance of all the Categorical and Numerical Columns.

Categorical Columns		Numerical Columns	
Column Name	Significance	Column Name	Significance
SeniorCitizen	YES	tenure	YES
Partner	YES	MonthlyCharges	YES
Dependents	YES	TotalCharges	NO
MultipleLines	YES		
InternetService	YES		
OnlineSecurity	YES		
OnlineBackup	YES		
DeviceProtection	YES		
TechSupport	YES		
StreamingTV	YES		
StreamingMovies	YES		
Contract	YES		
PaperlessBilling	YES		
PaymentMethod	YES		
Tenure_group	YES		
Gender	NO		
PhoneService	NO		

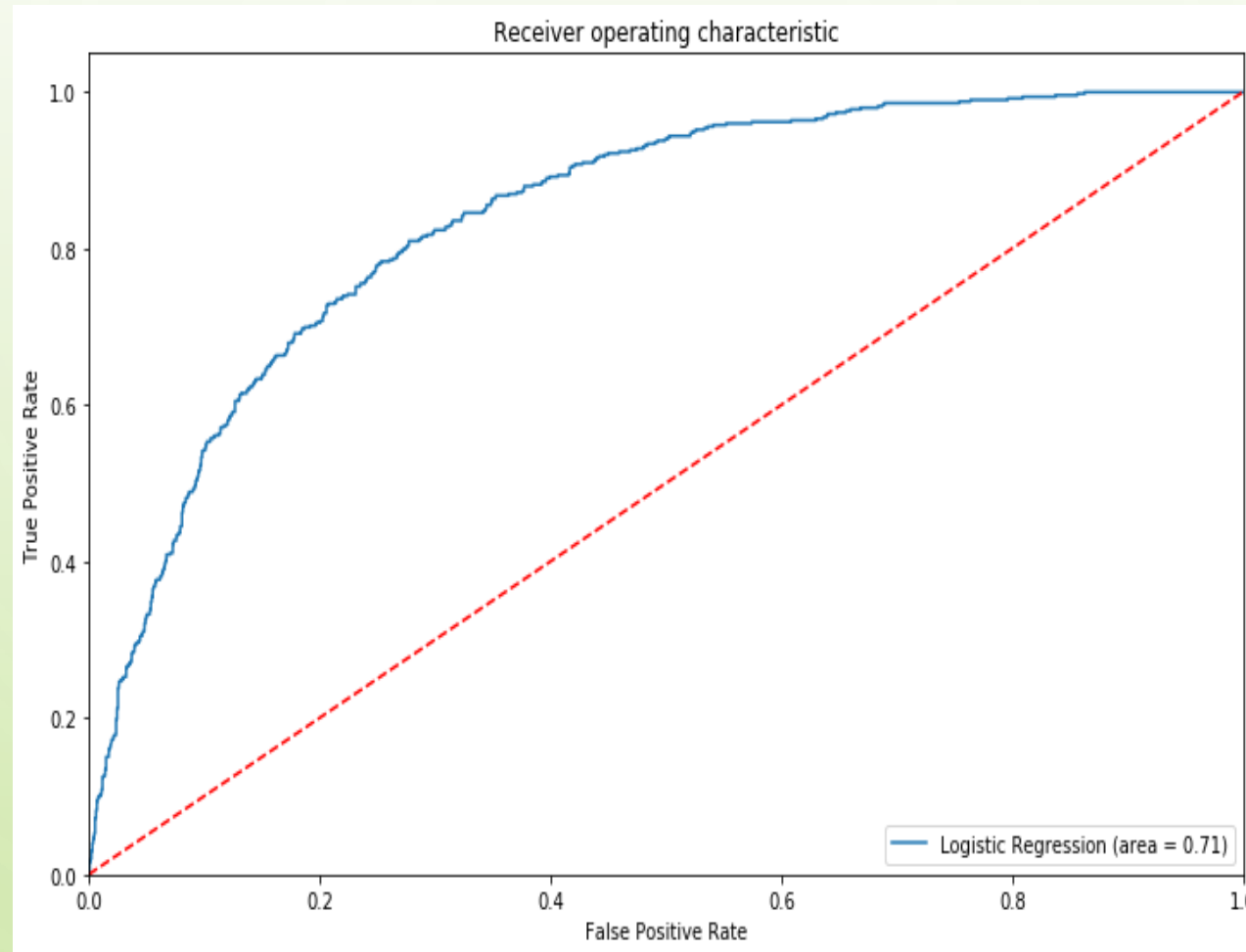
BASELINE MODELLING

- Used Label encoding for all the binary columns and created dummy variables for them using the `get_dummies` method of pandas to create dummy features for all the categorical data.
- Used Train Test Split method to split my data set into “X”, “y”.
- The variable ‘X’ contains all columns except the target variable and the variable ‘y’ contains only the target variable.
- The training set is 75% of our total data and training set is 25% of the data.

BASELINE MODELLING – LOGISTIC REGRESSION

- Instantiated a Logistic Regression object, fitted on the Training set, predicted on the test set and calculated different scoring metrics to measure the performance of the model.
- Classification Report Summary for Logistic Regression (LR):
 - The class of interest (Churn class) for the training set has only 67% precision and 53% recall score and the test set has 68% precision and 51% recall score.
 - Our model for the churn classes has not performed great. But the overall global accuracy scores were much higher than the individual scores for the churn class.

ROC curve for Logistic Regression

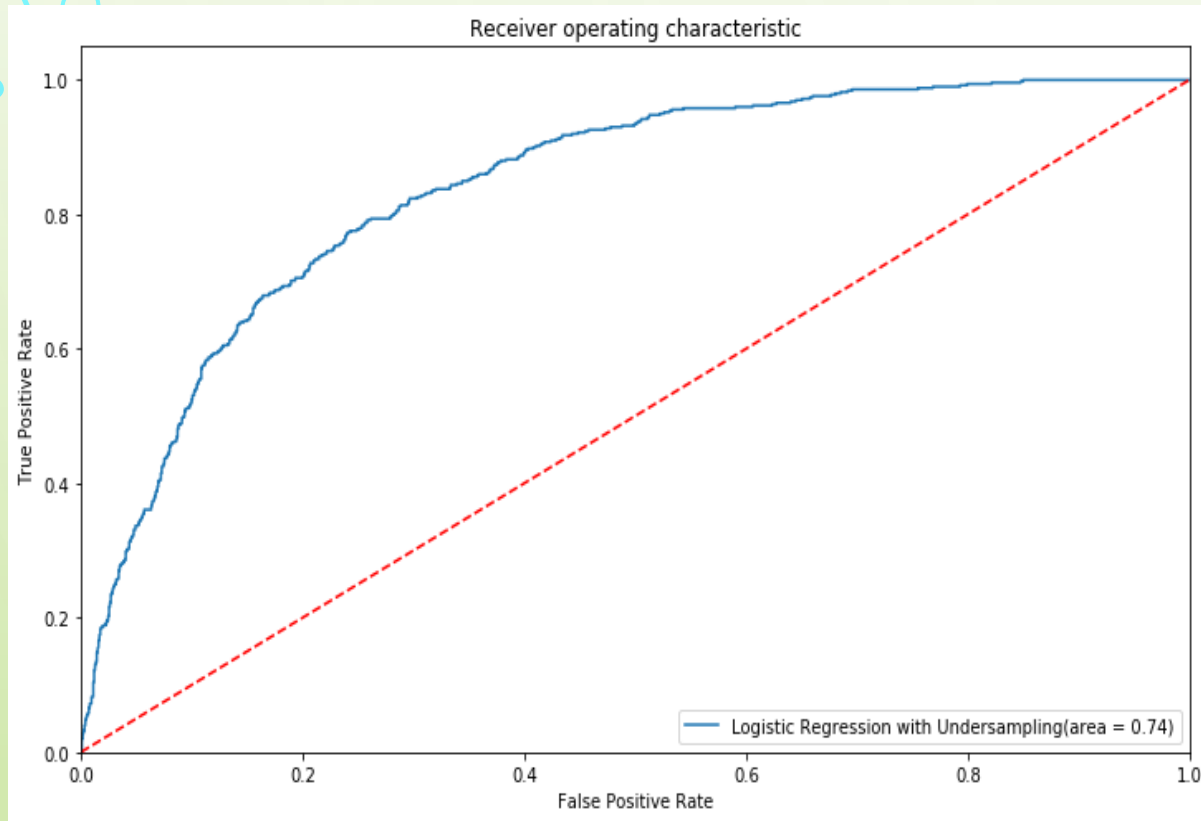


The Area Under Curve for the Logistic Regression model is 0.706359628925045.

EXTENDED MODELLING

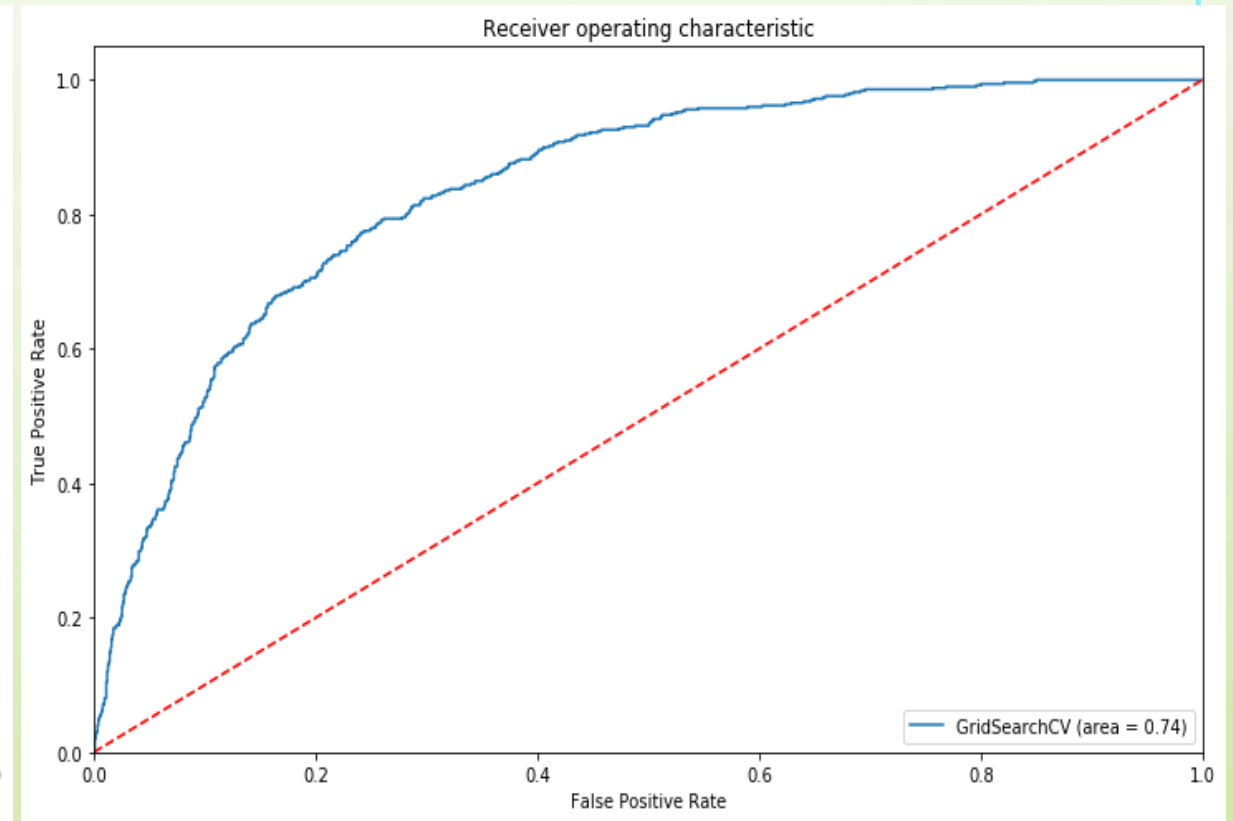
- Performed below listed modelling
 - Logistic Regression with Undersampling.
 - The class of interest for the train set now has 69% precision and 64% recall score and that of the test set has 64% precision and 63% recall score.
 - Logistic Regression with GridSearchCV.
 - There wasn't much of a difference in the recall scores for the two. Hence, we could say there is no overfitting.
 - XGBoost with SMOTE.
 - The class of interest for the train set for XGBoost model has 88% precision and 87% recall score and that of the test set has 64% precision and 58% recall score.
 - XGBoost with GridSearchCV.
 - The class of interest for the train set for XGBoost with GridSearchCV has 92% precision and 88% recall score and that of the test set has 65% precision and 55% recall score.

ROC curve for Logistic Regression with Undersampling



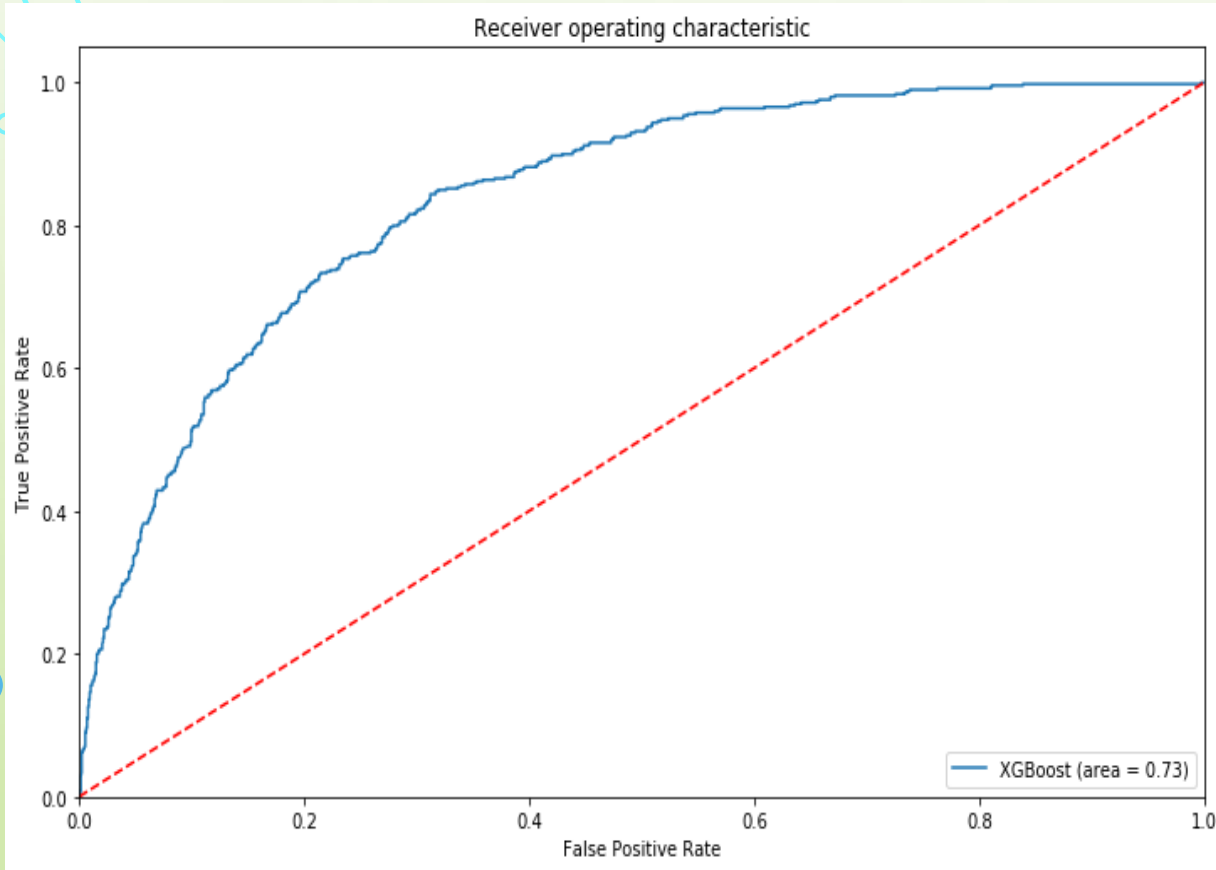
Area Under Curve for the Logistic Regression with undersampling is 0.7424984676166914.

ROC curve for Logistic Regression with GridSearchCV



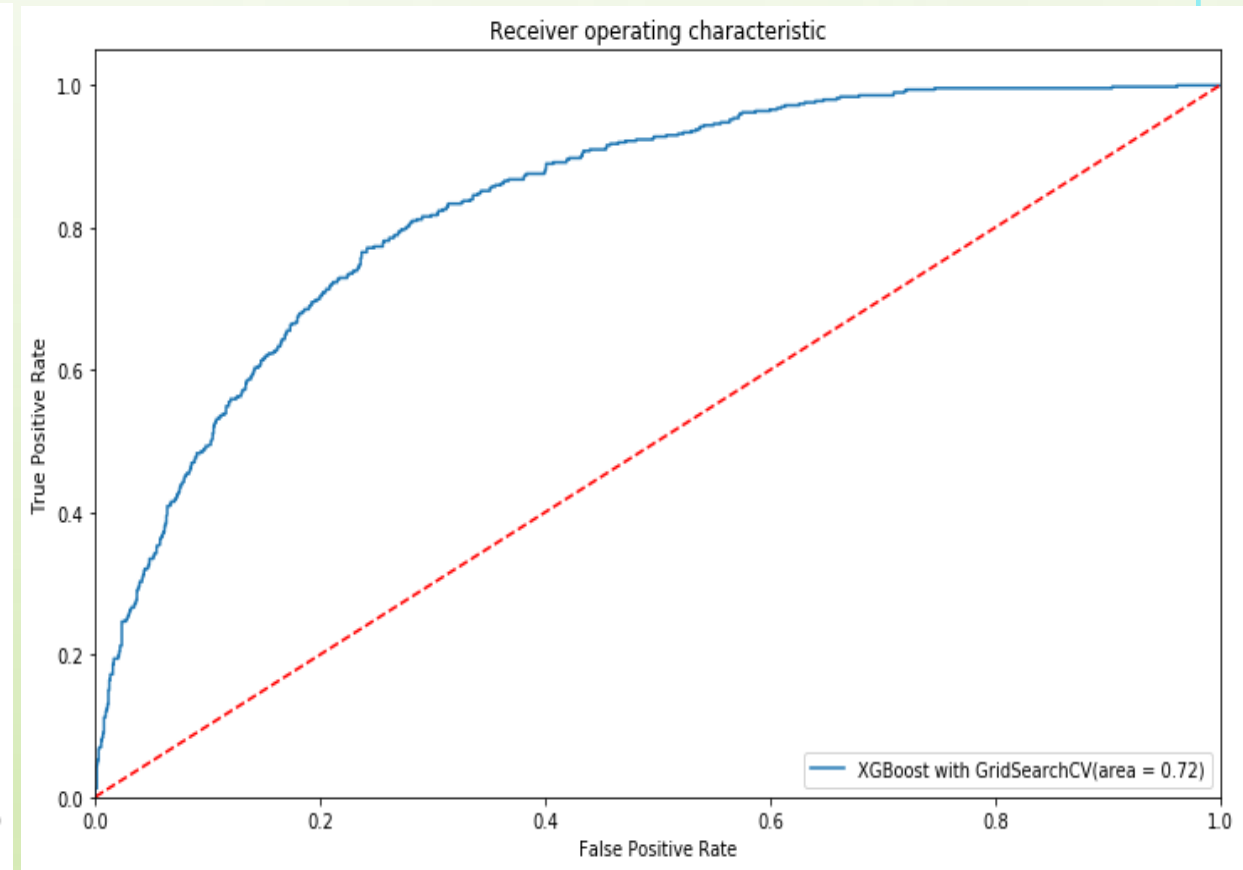
Area Under Curve for the GridSearchCV model is 0.7414964636086754.

ROC curve for XGBoost with SMOTE



Area Under Curve for the XGBoost model is 0.7252087376019867.

ROC curve for XGBoost with GridSearchCV




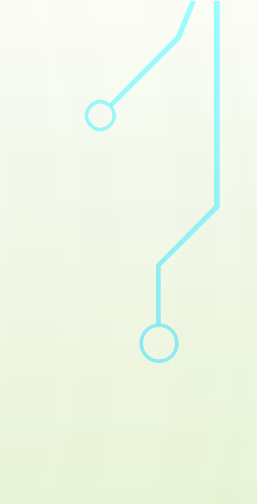
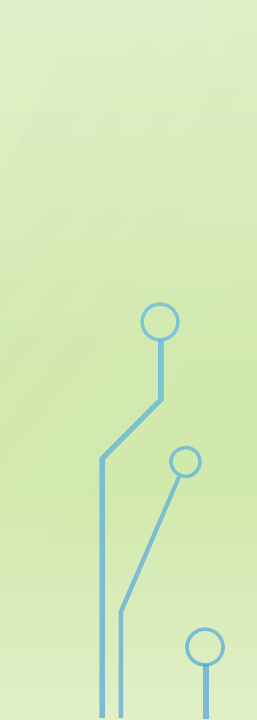
Area Under Curve for the XGBoost model with GridSearch is 0.7161216251838066.

FINDINGS

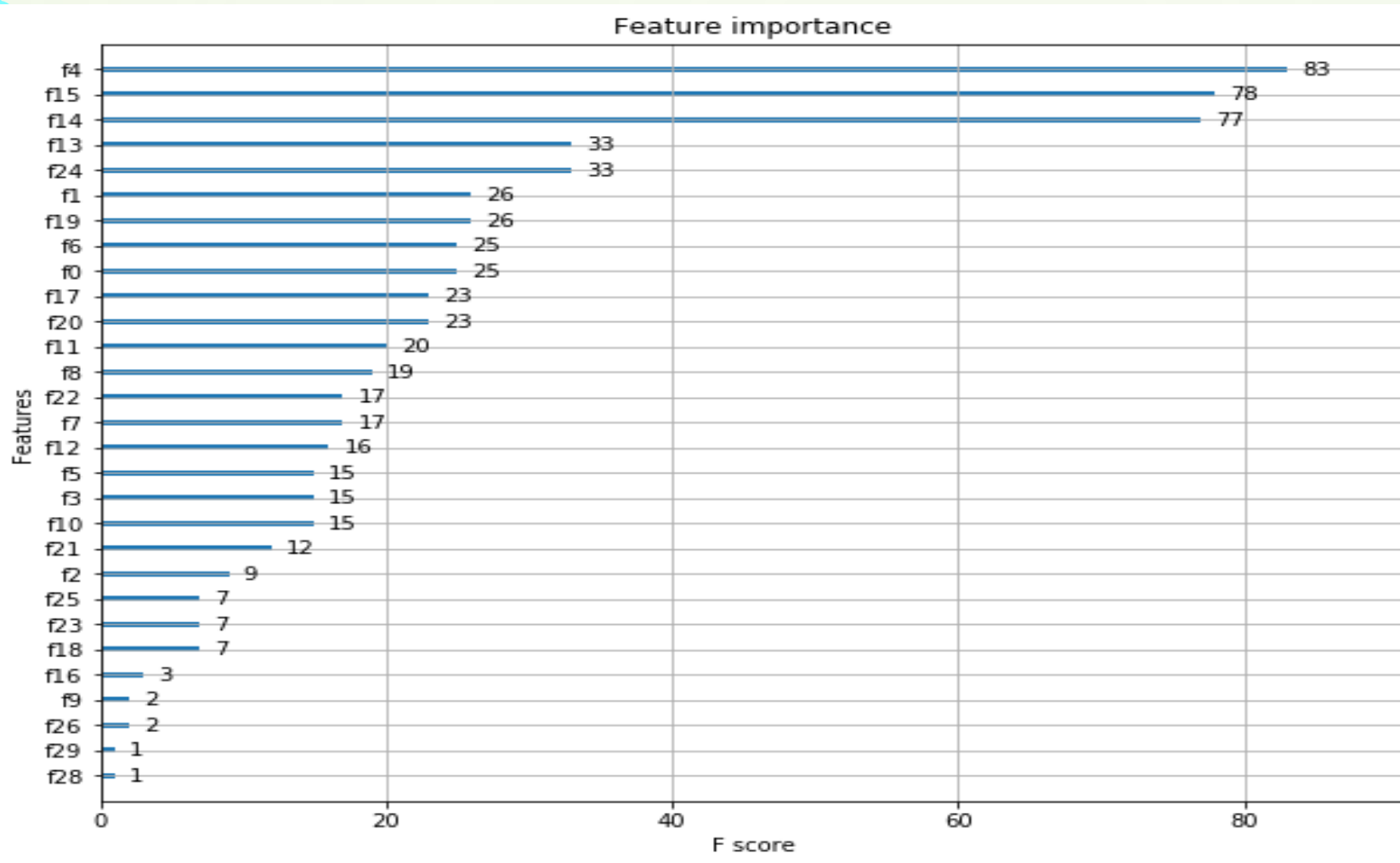
Algorithms	Precision	Recall
Logistic Regression (LR)	68%	51%
LR with Undersampling	64%	63%
LR with GridSearchCV	64%	62%
XGBoost with SMOTE	64%	58%
XGBoost with GridSearchCV	65%	55%



CONCLUSIONS AND FUTURE WORK

- Previously, we jotted down the metrics for all the machine learning models.
 - We can conclude that Logistic Regression with undersampling performed better for the recall score which is our metric of interest.
 - Going forward I would like to improve the performance of the XGBoost model by using more parameters and also by trying out different algorithms.
- 
- 
- 

RECOMMENDATION FOR THE CLIENT



Feature Important Plot for XGBoost Model

The top 5 features are f4, f15, f14, f13, f24 and so on. Future work involves finding out the exact feature names for these numbers.