**Customer Churn Analysis**

*By Kunal Chand*

**1. Problem Statement:**

Customer churn is when a company’s customers stop doing business with that company. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals.

Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can priorities focused marketing efforts on that subset of their customer base.

Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

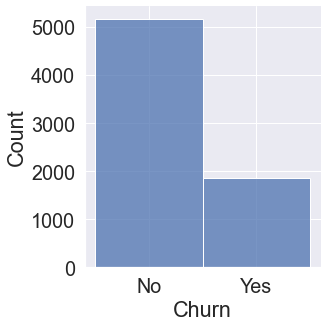
You will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

**2. Data Analysis**

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data, and other attributes. It is commonly conducted by data analysts using visual analytics tools, but it can also be done in more advanced statistical software, Python.

Before it can analyze data collected by multiple data sources and stored in data warehouses, an organization must know how many cases are in a data set, what variables are included, how many missing values there are, and what general hypotheses the data is likely to support. An initial exploration of the data set can help answer these questions by familiarizing analysts with the data with

Which they are working. We divided the data 8:2 for Training and Testing purposes respectively.



Data is highly imbalanced.

# download (50).png

# No null values are present.

# 3. EDA Concluding Remark

As for any basic model building, we have to understand the type of target variable; the data of the target variable is continued or classified.

Data Analysis is always the difficult part, for better understanding different kinds of bar plots, distribution plots are created with the target Column for finding the insights of the dataset we have.

Analytical Modeling always starts with the target variable we have, and in that case, our target variable is Churn, for that, we create some box plots with the target variable to understand which feature columns help to learn the model best and which feature columns reduce the accuracy of the model.

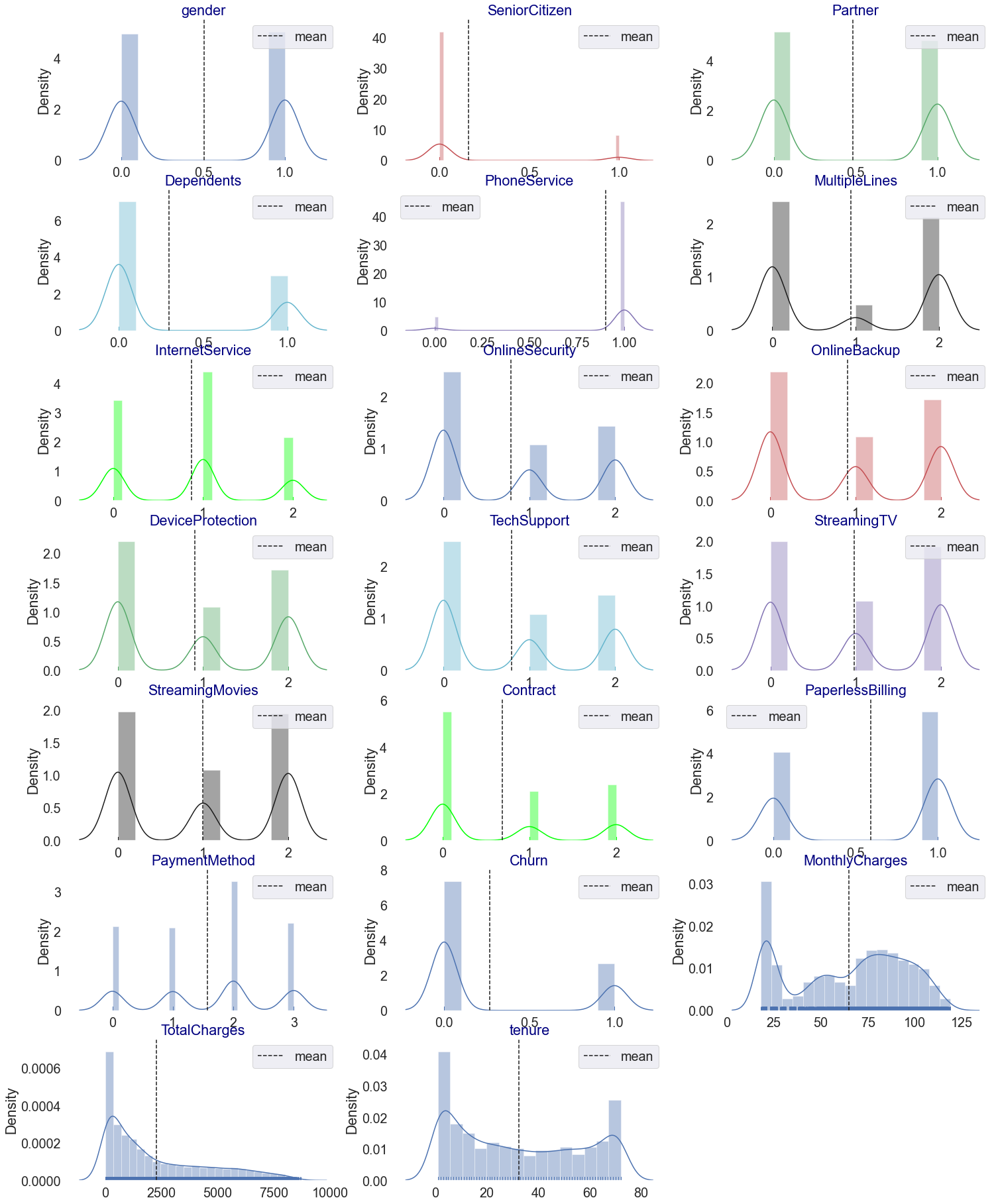
And after finding the relation and correlation with the target variable we choose either Regression Model or Classification Model. Here in this problem, our target feature column is categorized data so we build our Machine Learning model on Classification.

# Multivariate Analysis

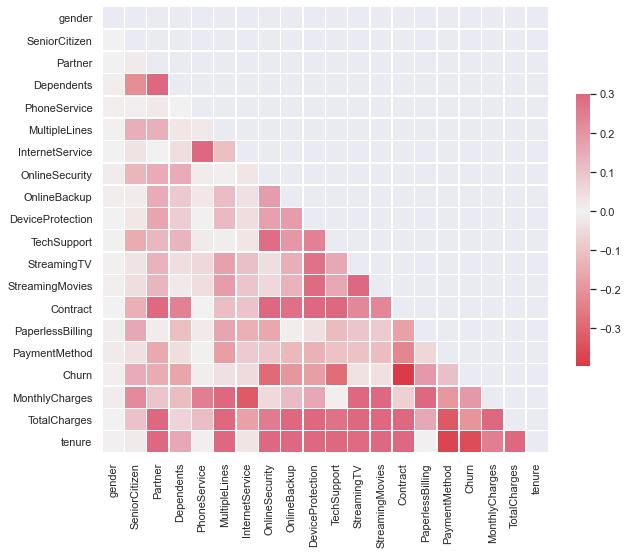
# download (33).png

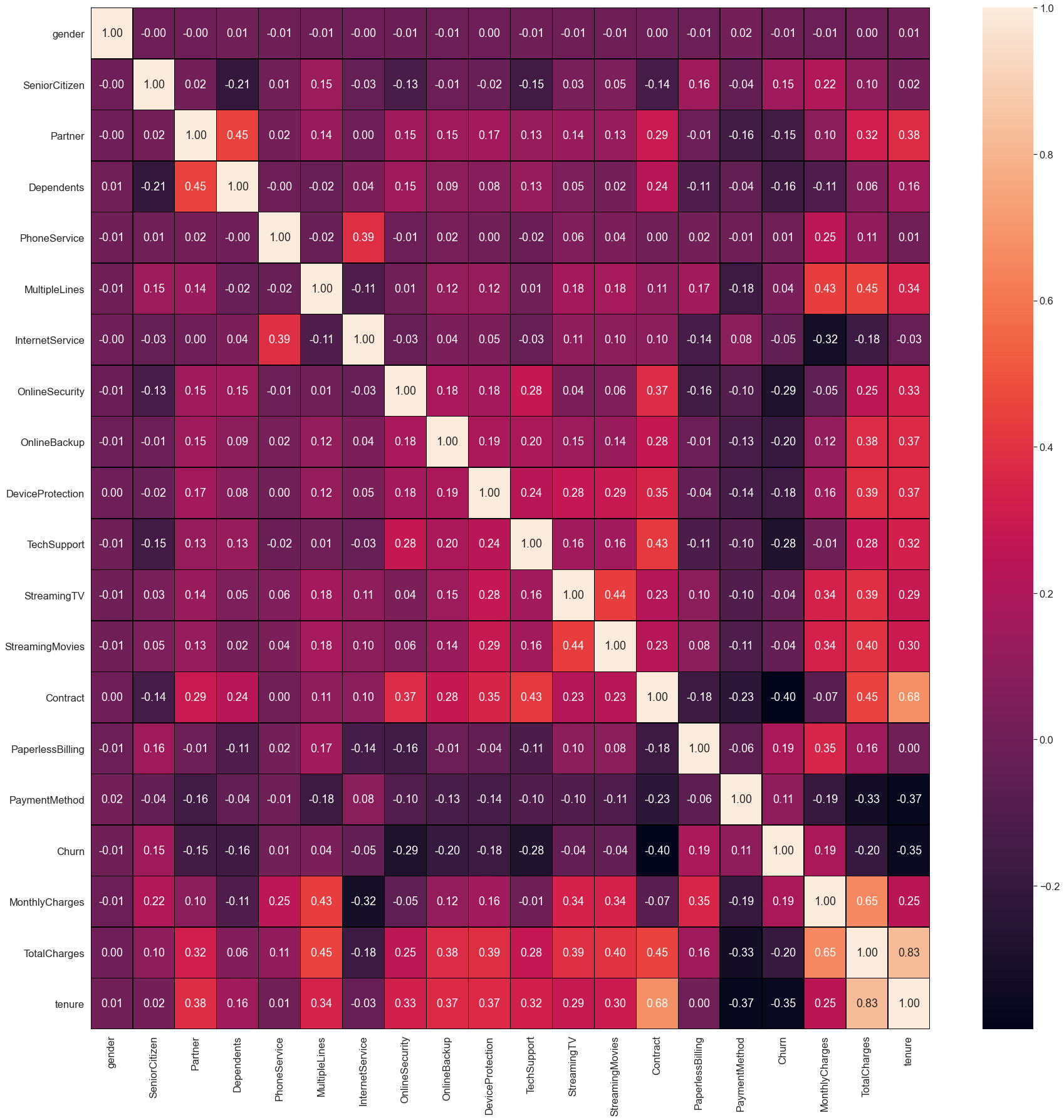
# All the above-plotted columns are self Exploratory.

# Use subplot and distribution plot to check data are normalized or not.



**CORRELATION BETWEEN THE COLUMNS:**

****

****

# Description of the dataset:

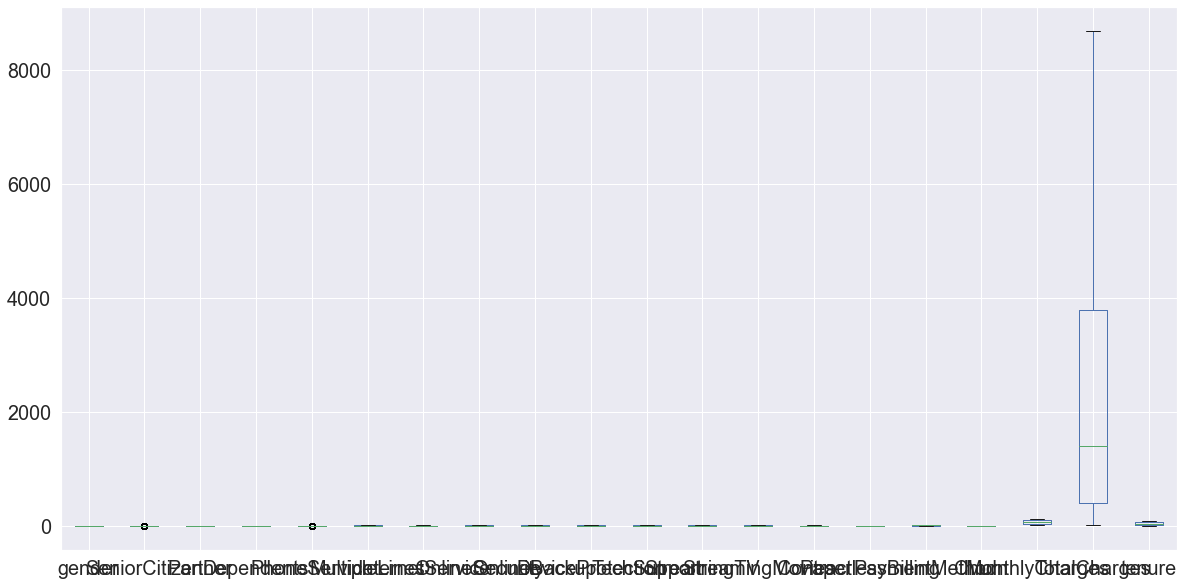
1. This helps us to understand the mean, mode, and deviation in the dataset.

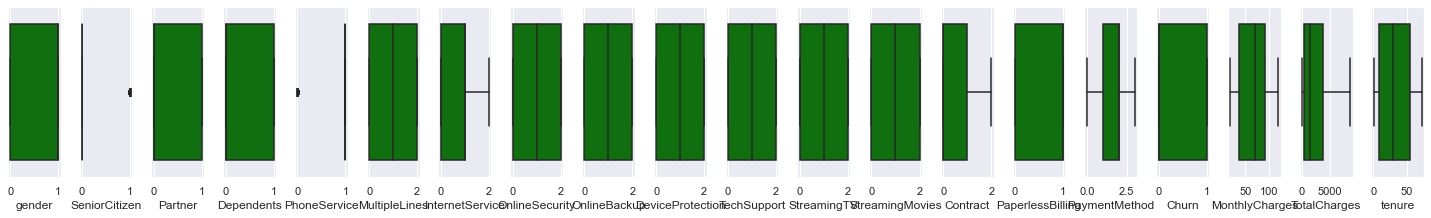
# PLOTTING THE DESCRIPTION OF THE DATASET

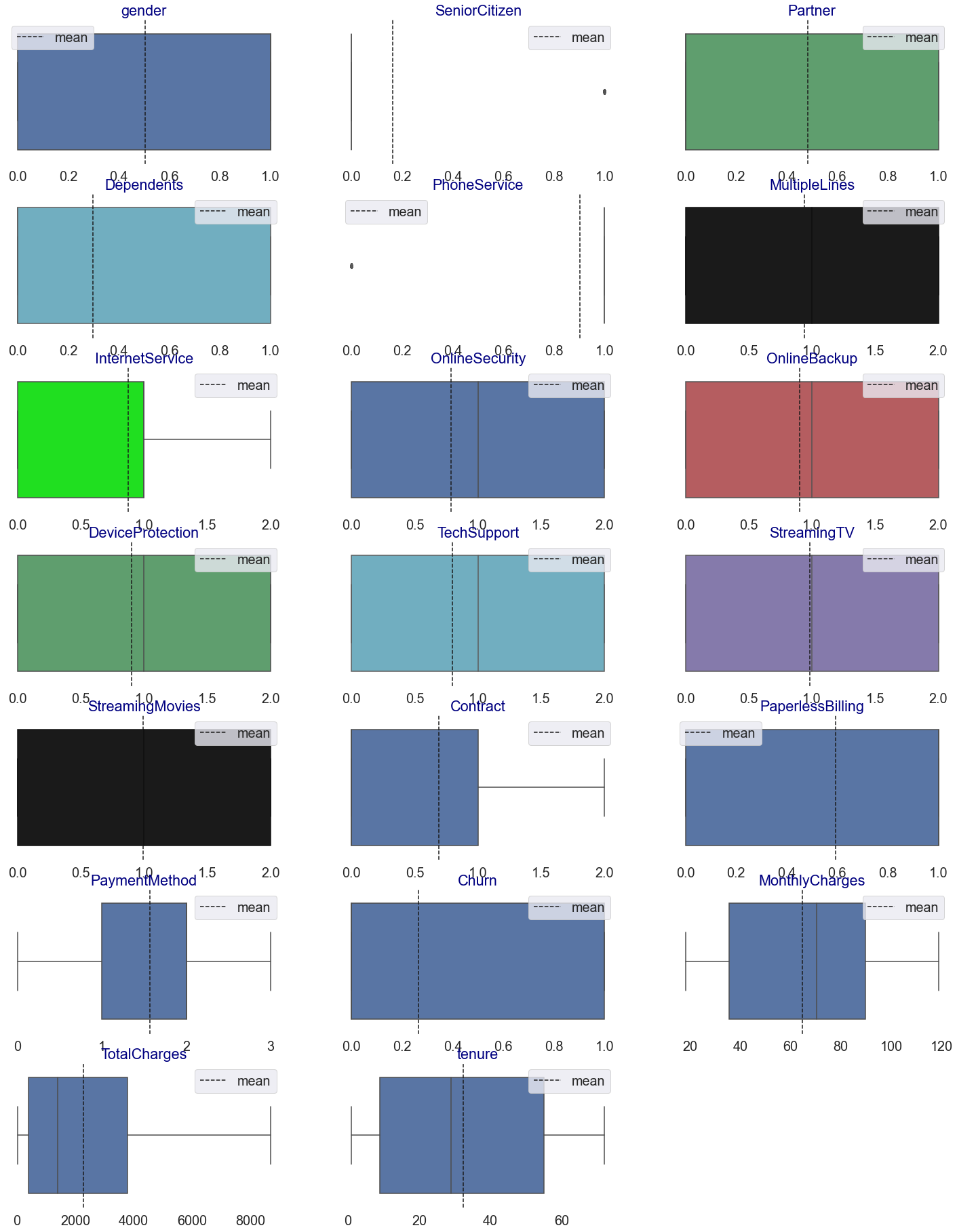


# Detecting outliers:

Outliers are the silent killer of accuracy, that’s why studying them and removing them is necessary.

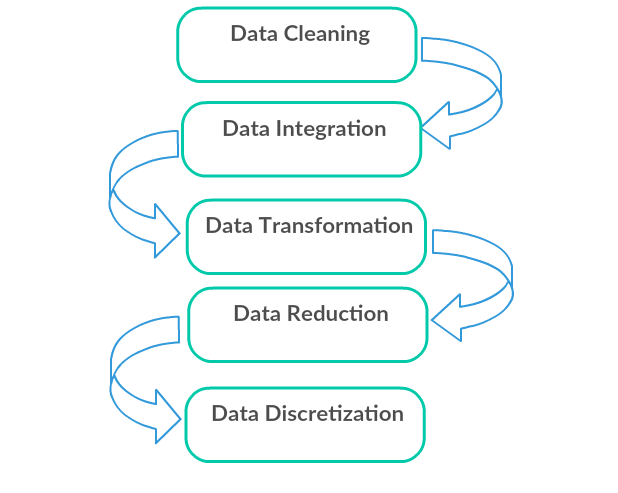






**4. Pre-Processing Pipeline.**

Data pre-processing can refer to the manipulation or dropping of data before it is used to ensure or enhance performance, and is an important step in the data mining process.



1. Data Cleaning: First we clean the data which have no use in prediction like the Over18 column, and then we drop the data which has a high no of missing percentages.
2. Data Integration: then we do some EDA process for finding out the meaning full insights of the data.
3. Data transformation is the process of changing the format, structure, or values of data; we use a labeled encoder for coding the object data into integer data.
4. Data Reduction: it is the process of finding the most correlated columns, and combining them because the machine does not understand which feature columns impact the most on accuracy.
5. Data discretization converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy, using box plots is makes a clear understanding of the data.

**5. Building Machine Learning Models**.

After analyzing the dataset, I observe that many of the feature columns are object types so first, we have to convert them into integer or float types so that the machine interprets the data and for that we do label encode all the features column.

After label encoding, we find that many feature columns have Nan values so we use mean and median for filling that missing data,

Then find the correlation between the columns with target columns and delete the non-related feature columns.

We observe that the target column is skewed so we remove the skewness of the target column because normal data gives better results when we make the M.L model.

The target column is object type so we start work on Classification Modelling building.

* Testing of Identified Approaches (Algorithms)

1. Logistic Regression
2. Regurgitation:

Ridge Classification

1. Ensemble techniques

DecisionTreeClassifier

# Random Forest Classifier

# 4. Support Vector Classifier

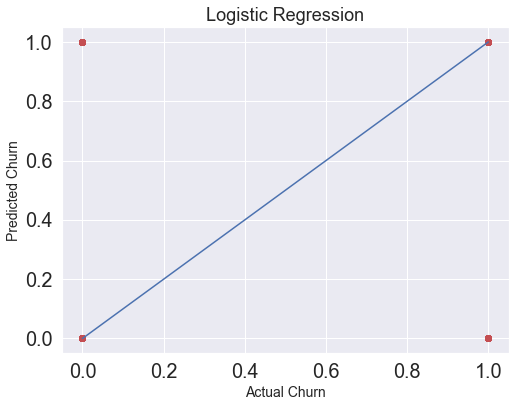
5. K-nearest Neighbour Classifier

**Logistic Regression Model**

• Logistic Regression is a machine learning algorithm based on supervised learning.

• It performs a regression task. Regression models a target prediction value based on independent variables.

• It is mostly used for finding out the relationship between variables and forecasting.



**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**[[859 96]**

**[147 169]]**

**0.8088119590873328**

**precision recall f1-score support**

**0 0.85 0.90 0.88 955**

**1 0.64 0.53 0.58 316**

**accuracy 0.81 1271**

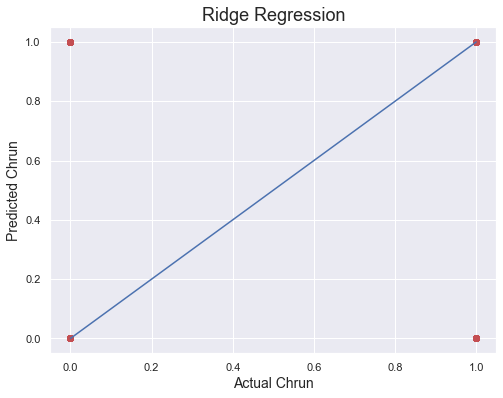
**macro avg 0.75 0.72 0.73 1271**

**weighted avg 0.80 0.81 0.80 1271**

# 1. Ridge

The [**Ridge**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html#sklearn.linear_model.Ridge) regressor has a classifier variant: **[RidgeClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html" \l "sklearn.linear_model.RidgeClassifier" \o "sklearn.linear_model.RidgeClassifier)**. This classifier first converts binary targets to {-1, 1} and then treats the problem as a regression task, optimizing the same objective as above. The predicted class corresponds to the sign of the regressor’s prediction. For multiclass classification, the problem is treated as multi-output regression, and the predicted class corresponds to the output with the highest value.

It might seem questionable to use a (penalized) Least Squares loss to fit a classification model instead of the more traditional logistic or hinge losses. However, in practice, all those models can lead to similar cross-validation scores in terms of accuracy or precision/recall, while the penalized least squares loss used by the **[RidgeClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html" \l "sklearn.linear_model.RidgeClassifier" \o "sklearn.linear_model.RidgeClassifier)** allows for a very different choice of the numerical solvers with distinct computational performance profiles.



**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**0.7970102281667978**

**[[838 83]**

**[175 175]]**

**precision recall f1-score support**

**0 0.83 0.91 0.87 921**

**1 0.68 0.50 0.58 350**

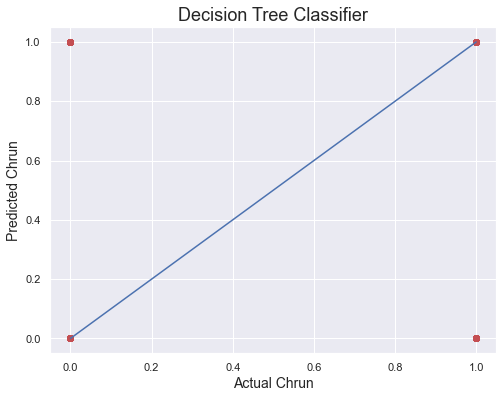
**accuracy 0.80 1271**

**macro avg 0.75 0.70 0.72 1271**

**weighted avg 0.79 0.80 0.79 1271**

**DecisionTreeClassifier**

**Decision Trees (DTs)** are a non-parametric supervised learning method used for classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

****

**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**0.7301337529504327**

**[[749 172]**

**[171 179]]**

**precision recall f1-score support**

**0 0.81 0.81 0.81 921**

**1 0.51 0.51 0.51 350**

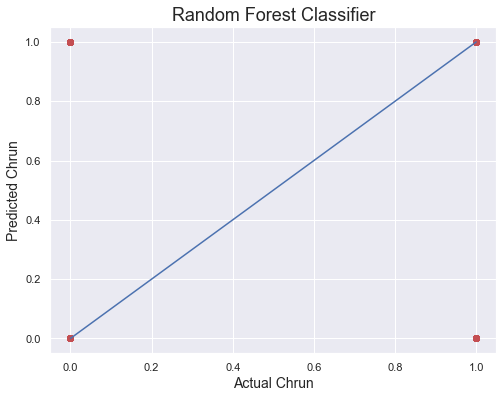
**accuracy 0.73 1271**

**macro avg 0.66 0.66 0.66 1271**

**weighted avg 0.73 0.73 0.73 1271**

### 3 Random Forest Classifiers

The core unit of random forest classifiers is the decision tree. The decision tree is a hierarchical structure that is built using the features (or the independent variables) of a data set. Each node of the decision tree is split according to a measure associated with a subset of the features. The random forest is a collection of decision trees that are associated with a set of bootstrap samples that are generated from the original data set. The nodes are split based on the entropy (or Gini index) of a selected subset of the features. The subsets that are created from the original data set, using bootstrapping, are of the same size as the original data set.

****

**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**0.7930763178599528**

**[[824 97]**

**[166 184]]**

**precision recall f1-score support**

**0 0.83 0.89 0.86 921**

**1 0.65 0.53 0.58 350**

**accuracy 0.79 1271**

**macro avg 0.74 0.71 0.72 1271**

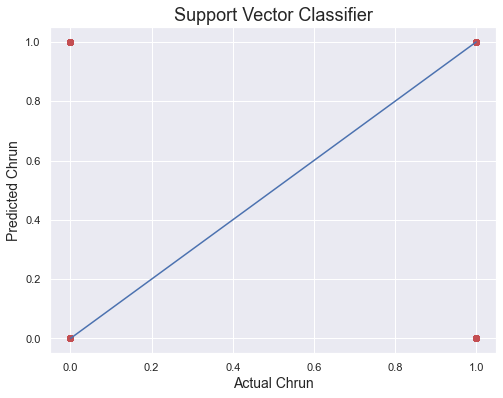
**weighted avg 0.78 0.79 0.79 1271**

**Support vector machines (SVMs)**

 are a set of supervised learning methods used for [classification](https://scikit-learn.org/stable/modules/svm.html#svm-classification).

The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where many dimensions are greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

****

**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**0.7922895357985837**

**[[835 86]**

**[178 172]]**

**precision recall f1-score support**

**0 0.82 0.91 0.86 921**

**1 0.67 0.49 0.57 350**

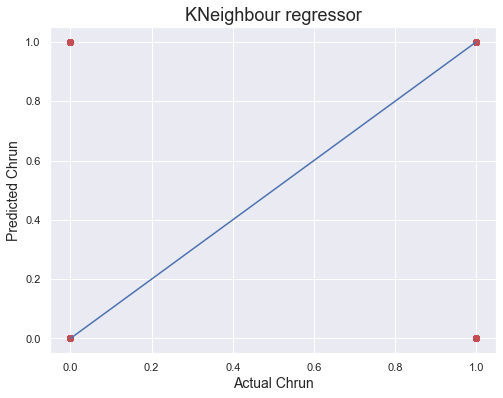
**accuracy 0.79 1271**

**macro avg 0.75 0.70 0.71 1271**

**weighted avg 0.78 0.79 0.78 1271**

K-Nearest Neighbour Classification

1. K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.
2. K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
3. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a good suite category by using K- NN algorithm.
4. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for Classification problems.
5. K-NN is a **non-parametric algorithm**, which means it does not make any assumptions on underlying data.
6. It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
7. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

****

**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**0.7946498819826908**

**[[811 110]**

**[151 199]]**

**precision recall f1-score support**

**0 0.84 0.88 0.86 921**

**1 0.64 0.57 0.60 350**

**accuracy 0.79 1271**

**macro avg 0.74 0.72 0.73 1271**

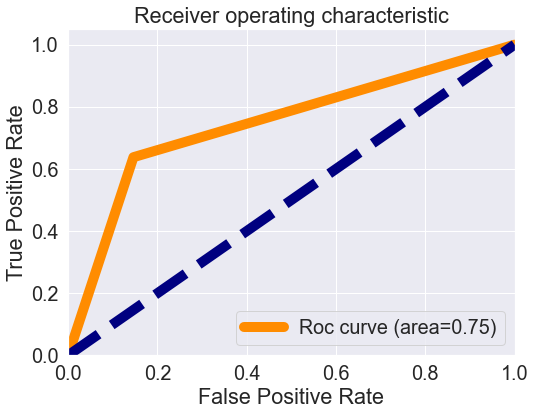
**weighted avg 0.79 0.79 0.79 1271**

**6. Concluding Remarks.**

# So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our Aim Column. It is seen overall Quality is the most effective attribute in predicting the house price and that the Logistic regression is the most effective model for our Dataset.

# We tested 6 models out of which Logistic Regression performing well:

Plotting Auc-Roc curve..with logistic regression prediction.

****

**Cross validation score is:- 80.5416144853189**

**accuracy\_score is :- 80.88119590873329**

**[[859 96]**

**[147 169]]**

**0.8088119590873328**

**precision recall f1-score support**

**0 0.85 0.90 0.88 955**

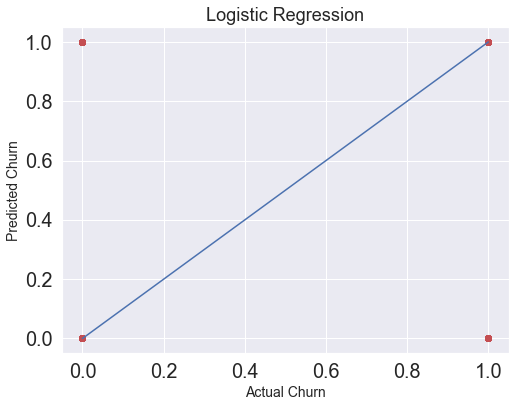
**1 0.64 0.53 0.58 316**

**accuracy 0.81 1271**

**macro avg 0.75 0.72 0.73 1271**

**weighted avg 0.80 0.81 0.80 1271**

**Our Model performs with an Accuracy Score of 80.88%.**

****

That's it! We reached the end of our exercise.

# Throughout this kernel, we put into practice many of the strategies

# for predicting the churn of Customer Churn Analysis

# .

We philosophized about the variables, we analyzed churn alone and with the most correlated variables, we dealt with missing data and outliers, we tested some of the fundamental statistical assumptions and we even transformed categorical variables into dummy variables. That's a lot of work that Python helped us make easier**.**

**Thank you**