SUMMER INTERNSHIP REPORT

CUSTOMER CHURN PREDICTION

In partial fulfillment for the award of the degree of:

MSC

(Data Science)

Submitted to:

Amity Institute of Integrative Sciences and Health, Amity University, Haryana

Submitted by:

TANIYA CHAUHAN

A525117724021:

Under the Supervision of:

Dr. Nirmal Punetha

Assistant Professor,

Amity Institute of Integrative Sciences and Health,

Amity University, Haryana, Gurugram - 122413

(Nov, 2025)

Dedicated To

To my parents, whose steadfast support laid the groundwork for my journey, I dedicate this project. I appreciate gratitude my mentors and teachers for their guidance, my friends for being so motivating and for me throughout. This work is also dedicated to all those in the data science community whose relentless pursuit of knowledge has inspired and nurtured my learning.

Declaration

Amity Institute of Integrative Sciences and Health

Amity University, Haryana

Declaration of the Candidate

I hereby declare that the content included in this thesis titled “CUSTOMER CHURN PREDICTION” by “Taniya Chauhan” in partial fulfilment of re- quirements of the degree of MSC in Data Science submitted to the Amity Institute of Integrative Sciences and Health, Amity University Haryana, is an authentic work record of my own carried out during a period of January 2025 to May 2025, under the supervision of Dr. Nirmal Punetha. The matter submitted in this thesis has not been submitted by me in any other University / Institute for the award of any Degree

Taniya Chauahan

(A525117724021)

Amity University, Haryana

Taniya Chauhan

InternshipReport-Churn_prediction_Taniya.pdf

1 dec - 7 dec 2025

1 dec - 7 dec 2025

Amity University, Noida

Document Details

Submission ID trn:oid:::16158:123344506

Submission Date Dec 1, 2025, 4:27 PM GMT+5:30

Download Date Dec 1, 2025, 5:53 PM GMT+5:30

File Name InternshipReport-Churn_prediction_Taniya.pdf

File Size 1.3 MB

20 Pages

3,054 Words

17,926 Characters

Submission ID
Submission ID trn:oid:::16158:123344506
trn:oid:::16158:123344506

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 14 words)

Match Groups

6 Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
3% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Page 2 of 23 - Integrity Overview

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin

Submission ID trn:oid:::16158:123344506

Match Groups

- **6** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	
University of Hertfordshire on 2025-01-04		1%
2	Internet	
eprints.utm.my		<1%
3	Submitted works	
Liverpool John Moores University on 2021-06-05		<1%
4	Internet	
www.numberanalytics.com		<1%
5	Submitted works	
The Robert Gordon University on 2024-12-12		<1%
6	Publication	
Voulgari, Evangelia. "Data Analysis and Prediction Algorithms with Python", Univ...		<1%

Acknowledgments

I wish to place on record my heartfelt appreciation of my faculty at Amity University for guiding me during the entire phase of this CUSTOMER CHURN PREDICTION Project. Their suggestions helped improve the quality of this work.

My thanks extend to the customer churn prediction dataset that was made publicly available, along with the other public use tools that aided in the analysis.

Most of all, I want to appreciate my parents and family who supported me through-
out the process, my friends who offered their help and positive remarks, and everyone
who encouraged me along the way.

TABLE OF CONTENTS

5. **Findings and Conversation**
6. **Implementation of Code**
7. **Prospects**
8. **Conclusion**
9. **Citations**

1.Introduction

As markets become more competitive and consumers have more options than ever, retaining customers has emerged as one of the top priorities for contemporary businesses. These days, businesses rely on data analytics to understand how consumers use their services and to spot early warning signs that a customer might cancel their account or stop their subscription. Businesses can prevent needless losses and take prompt action by identifying these patterns.

I worked on a project that used data-driven methods to forecast customer attrition during my summer internship. The project's objectives were to investigate customer behaviour, investigate churn-causing factors, and create machine-learning models that can identify customers who are likely to leave. . I was able to obtain practical experience in feature engineering, model development, exploratory analysis, and data cleaning through this project.

I learned how churn prediction aids in business decision-making while working on this problem. The analysis's insights can assist businesses in developing customer-focused strategies, raising the caliber of their services, and improving retention strategies. All things considered, this project improved my technical abilities and expanded my comprehension of how predictive analytics can help address actual business problems.

1.1 The Study's Background

In the fiercely competitive business world of today, keeping existing clients has become just as crucial as finding new ones. Businesses in a variety of sectors gather enormous volumes of consumer data, which can be examined to spot trends in behaviour and determine which clients are most likely to discontinue using a service. We call this phenomenon "customer churn." Organizations can take proactive measures like tailored offers, improved customer service, or focused marketing campaigns by anticipating churn at the appropriate time. As machine learning has grown, churn prediction has become more precise and data-driven, assisting companies in strengthening their long-term client relationships.

1.2 Problem Description

Even with sizable customer bases, many businesses suffer large losses as a result of unforeseen customer attrition. Most businesses find it difficult to spot churn trends early enough to take appropriate action. This study aims to solve the following problem: How can machine learning models be used to accurately predict customer churn based on past behaviour and service attributes? Analysing various customer characteristics, finding hidden patterns, and developing a model that can accurately predict churn are the challenges.

1.3 The Study's Goals

This project's primary goal is to create and assess a predictive model that identifies clients who are likely to leave.

Particular goals consist of:

Principal Goals

1. To examine consumer data and pinpoint important variables that affect churn behaviour.
2. To prepare and preprocess the dataset in order to create a successful churn prediction model.
3. To categorize clients as either churn or non-churn using machine learning algorithms.
4. To assess which model offers the best prediction accuracy by comparing various models.
5. To assist companies in comprehending patterns in consumer behaviour and assisting in the formulation of retention strategies

Extra Objectives

6. To use data exploration techniques to visualize churn-related patterns and trends. 7. To create insights that can help businesses lower attrition and enhance

---

1.4 Project Scope

The application of machine learning and data analytics methods to a structured customer dataset is the main goal of this project. Data preprocessing, exploratory analysis, feature engineering, model development, and performance comparisons are all included. Real-time deployment and sophisticated deep learning architectures are not included in the study. Rather, it seeks to highlight significant business insights and develop a precise, comprehensible churn prediction model.

2.Literature Review:

Customer churn, or the gradual loss of customers, is a major problem for companies trying to sustain growth and revenue. Businesses can take proactive steps to keep high-risk clients by accurately forecasting churn. Conventional methods, like statistical analysis and logistic regression, were frequently employed to determine the variables that affected customer attrition, such as service contracts, usage trends, and customer demographics. However, these techniques frequently had trouble handling big datasets and intricate consumer behavior.

Models like decision trees, random forests, and gradient boosting have been used more frequently for churn prediction as machine learning techniques have grown. These models provide improved predictive performance by capturing non-linear relationships and patterns that statistical methods might overlook.

In order to increase prediction accuracy, large and complex datasets, such as transaction histories and customer interactions, have recently been analysed using deep learning techniques, such as neural networks.

Researchers also emphasize the significance of explainable AI and customer segmentation. Models can make more precise predictions by classifying consumers according to their behaviour or demographics, and interpretability strategies assist companies in comprehending the causes of possible attrition. All things considered, firms can lower attrition, improve customer satisfaction, and optimize retention tactics by fusing sophisticated algorithms with practical insights.

3. Technologies and Tools

Python is the main programming language used for modelling and data analysis. Pandas and NumPy are used for numerical calculations and data manipulation.

- Seaborn and Matplotlib: For displaying patterns and trends in data.
- Scikit-learn: This tool is used to implement machine learning algorithms. • Jupyter Notebook: An interactive code writing, testing, and documentation environment.
- Machine Learning Algorithms: Logistic Regression, Random Forests, Decision Trees, and Gradient Boosting

4. Methodology

This project's methodology uses a methodical approach to forecast customer attrition. Understanding and preparing the dataset, conducting exploratory analysis, creating machine learning models, and assessing them to choose the best-performing model are the first steps in the process. For customer retention, this methodical approach guarantees accurate forecasts and useful insights.

4.1 Description of the Dataset

The dataset includes comprehensive customer data, such as account details, service usage trends, and demographic characteristics. The target variable shows whether a customer has churned, and each record represents a unique customer. The basis for analysis, feature engineering, and model development is this dataset.

4.2 Understanding and Loading Data

This project's dataset came from Kaggle, which offers a customer churn dataset that is accessible to the general public (Kaggle Link). The Pandas library is used to load the dataset into Python. To comprehend the structure, data types, and general distribution of features, a preliminary analysis is carried out. This step offers a basic understanding of customer patterns and assists in identifying potential problems like incorrect data types or inconsistencies.

4.3 Preprocessing Data

To get the dataset ready for machine learning models, data preprocessing is done. This step consists of:

4.3.1 Dealing with Missing Data

The dataset's missing or null values are found and handled appropriately. To maintain the dataset's completeness and dependability, missing values are either eliminated or substituted using statistical techniques like mean, median, or mode.

4.3.2 Categorical Variable Encoding

Encoding methods like one-hot encoding and label encoding are used to translate categorical features—which machine learning models are unable to use directly—into numerical values. This guarantees that every feature is in a modeling-ready format.

4.3.3 Scaling Features

StandardScaler and MinMaxScaler are two methods for scaling numerical features to a standard range. Scaling enhances the performance of algorithms by preventing features with higher values from controlling the learning process.

4.4 Exploratory Data Analysis (EDA)

EDA is performed to understand the characteristics of the dataset and identify patterns that may influence churn:

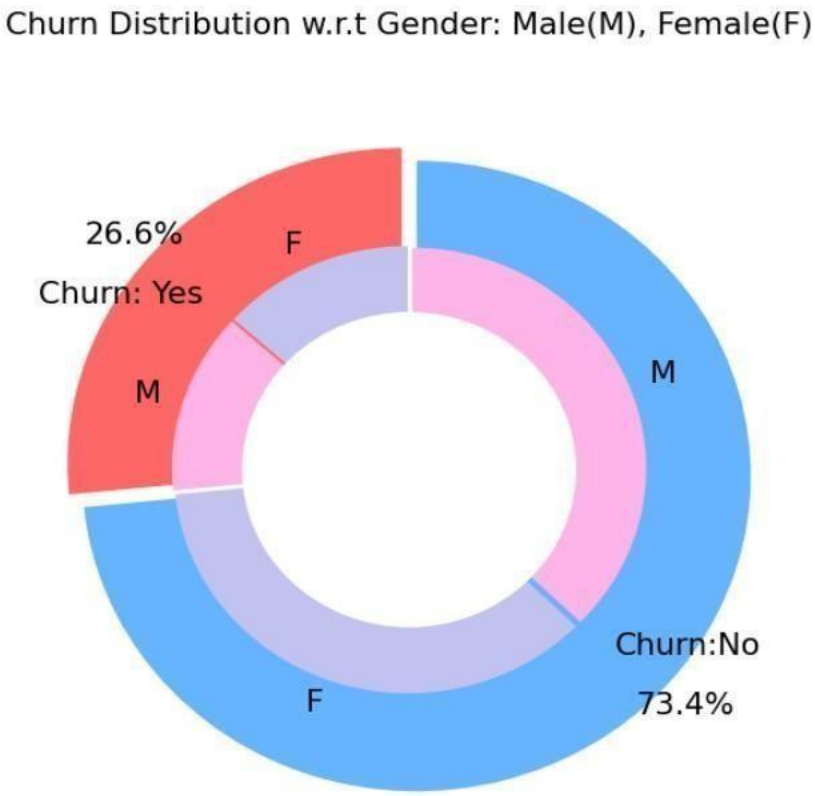


Figure 4.4.1: Screenshot Showing the Customer Churn or not churn between genders

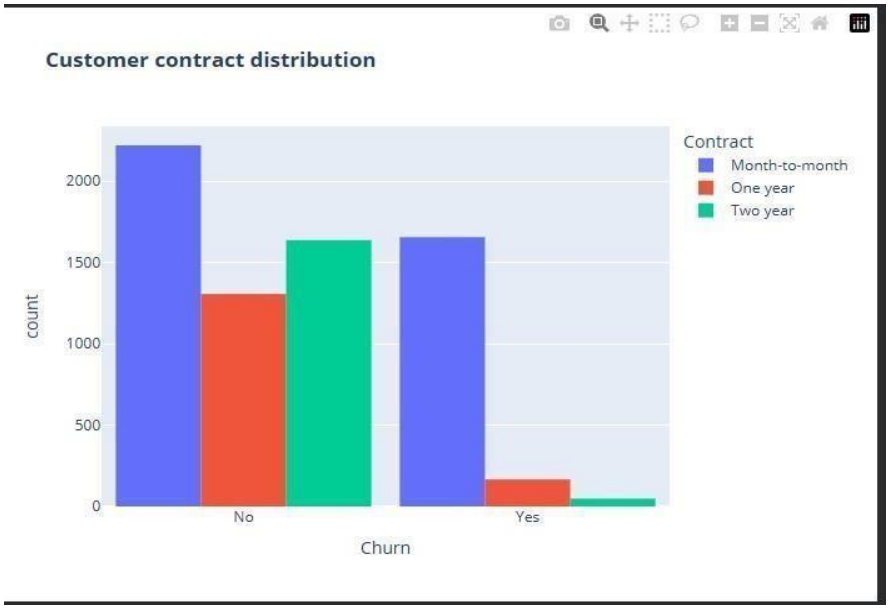


Figure 4.4.2: Screenshot Showing the Customer Contract distribution

Almost 75% of clients with month-to-month contracts eventually stopped using the service. By contrast, only roughly 3% of clients with a two-year contract and 13% of those with a one-year contract chose to cancel.

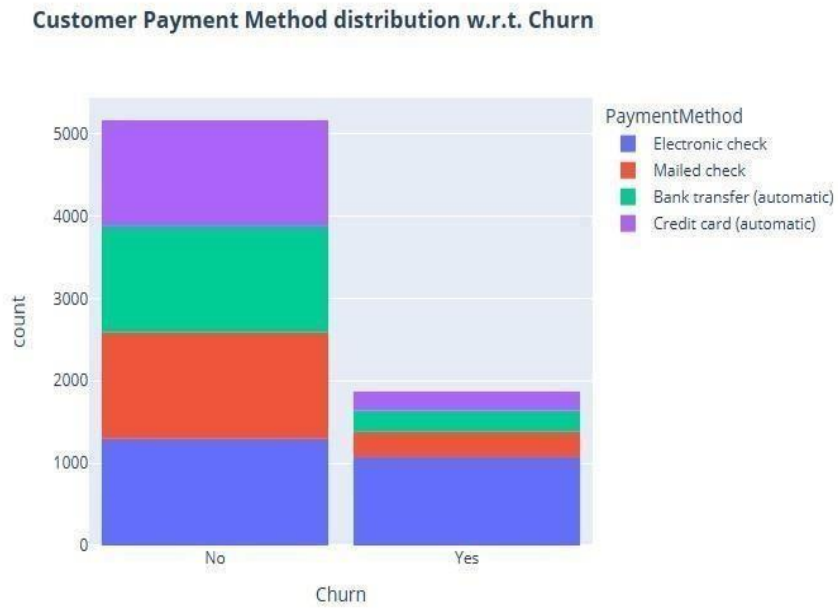


Figure 4.4.3: Screenshot Showing the Customer Payment Method distribution w.r.t churn

Electronic checks were used by a significant portion of clients who stopped using the service. Customers who paid by bank auto-transfer, credit card auto-pay, or mailed check, on the other hand, had a significantly lower tendency to churn.

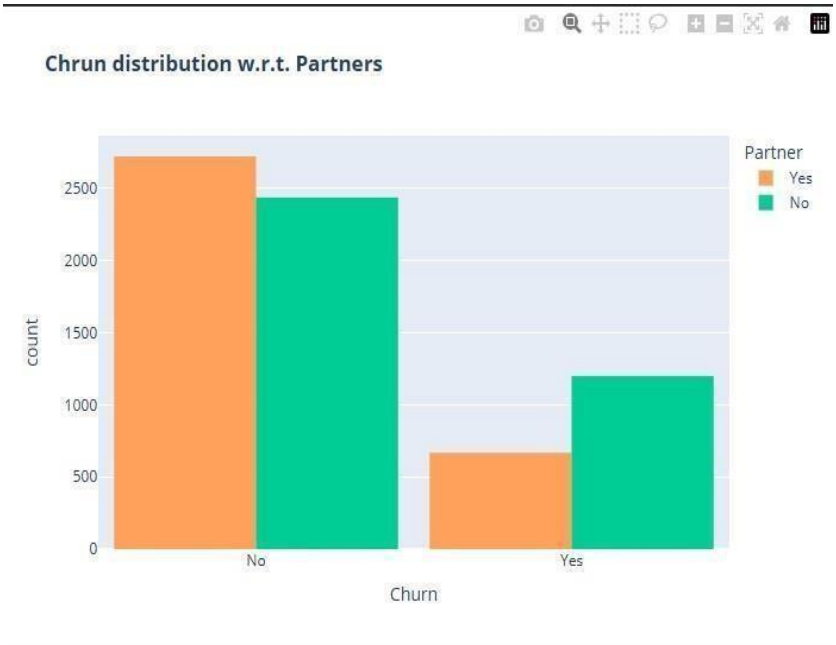


Figure 4.4.4: Screenshot Showing the Customer distribution w.r.t Partners

Compared to those who are partnered, customers without partners are more likely to churn.

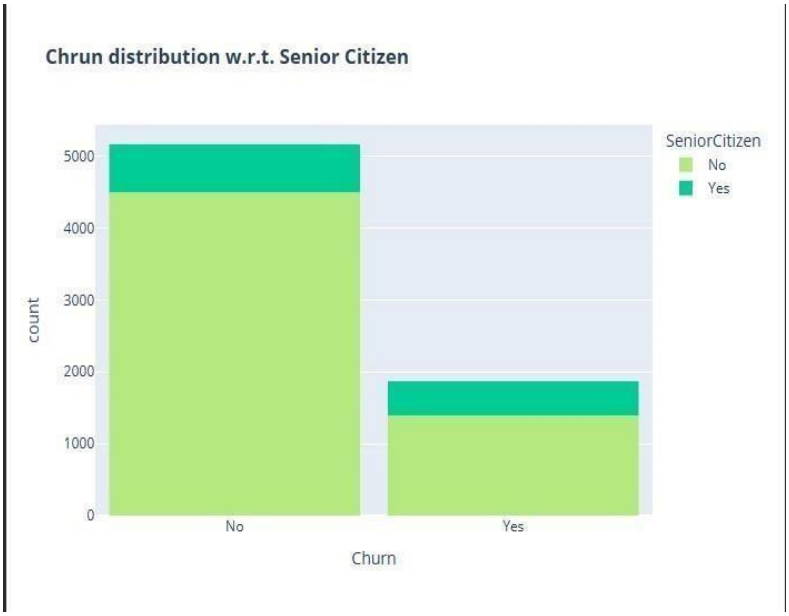


Figure 4.4.5: Screenshot Showing the Customer distribution w.r.t Senior citizen

The proportion of senior citizens in the dataset is quite small. However, among those who are senior citizens, a majority tend to churn.

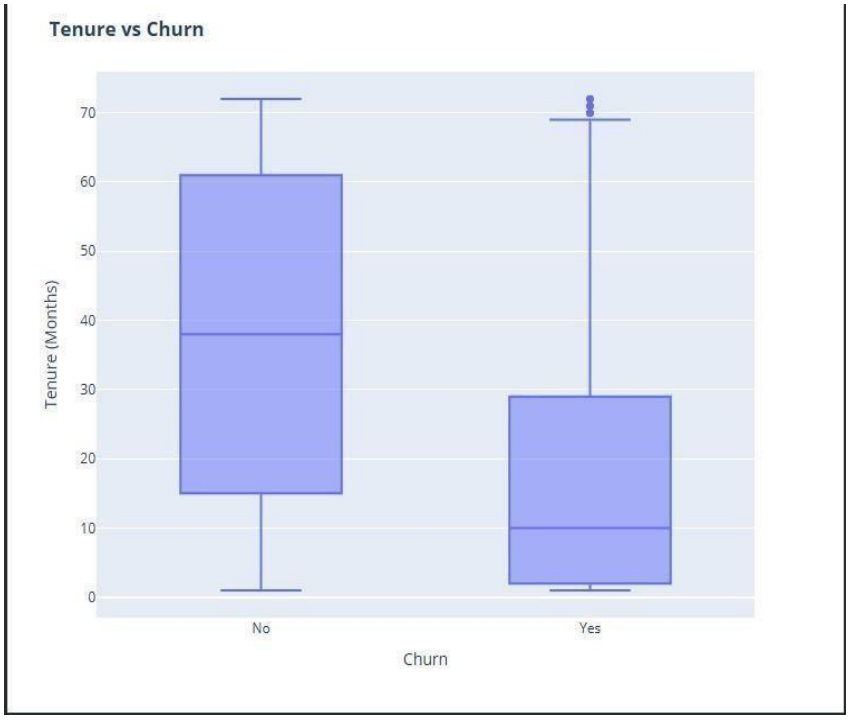


Figure 4.4.6: Screenshot Showing the Tenure vs churn

Customers who are new to the service have a higher chance of churning compared to longterm customers.

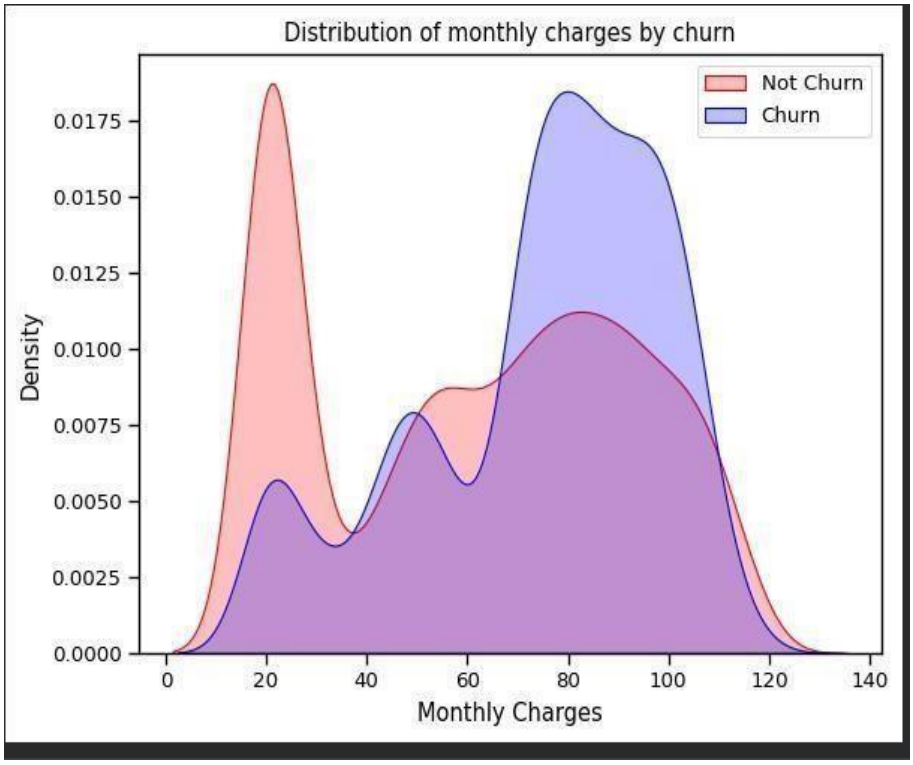


Figure 4.4.7: Screenshot Showing the Distribution of monthly charges by churn

Monthly fees are typically higher for customers who leave and lower for those who stay. The density curve demonstrates that customers who pay more each month are more likely to churn.

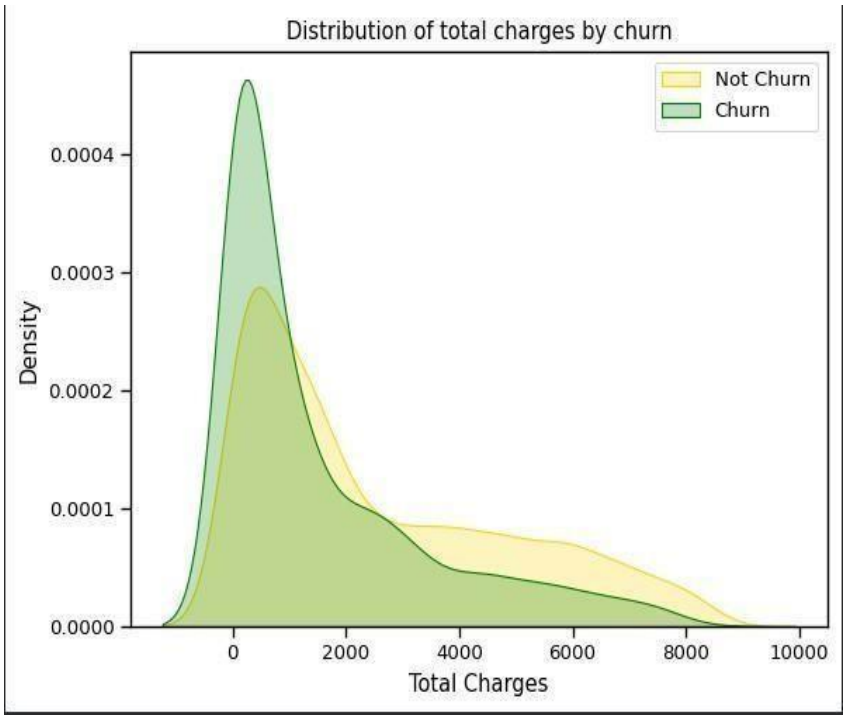


Figure 4.4.7: Screenshot Showing the Distribution of total charges by churn

Generally speaking, churning customers have lower total charges, indicating that they used the service for a shorter period of time. Higher total charges for non-churned customers reflect longer tenure and continuous usage.

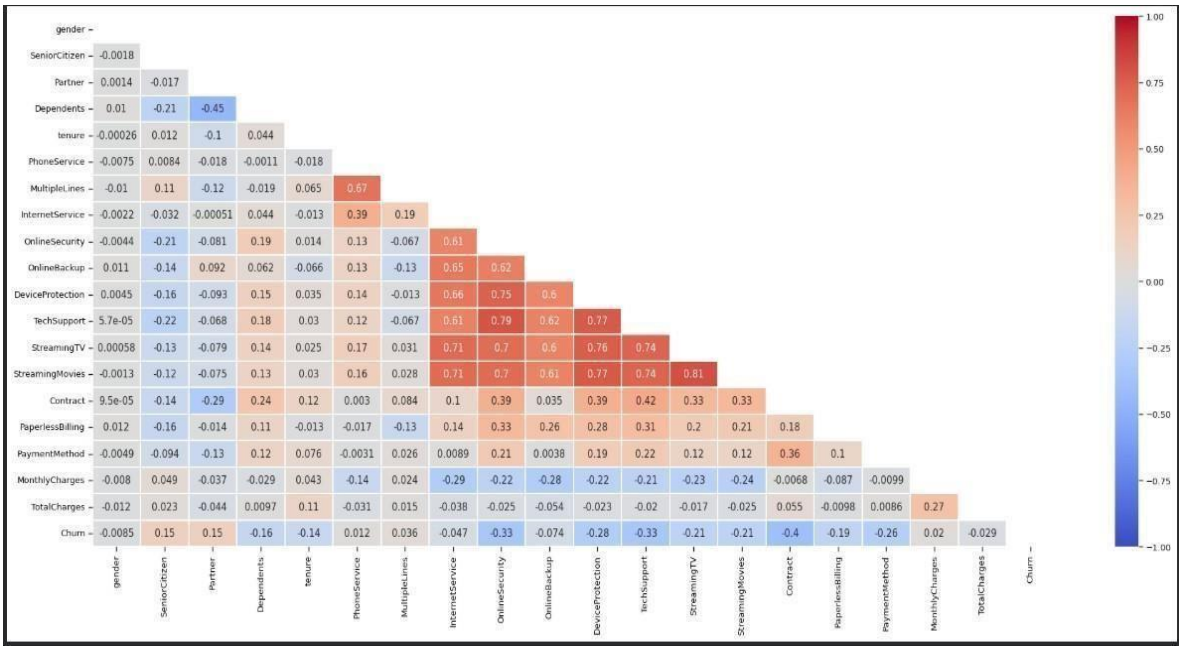


Figure 4.4.7: Screenshot Showing the Correlation Heatmap

Tenure and security-related services are negatively correlated with churn, according to the heatmap, indicating that users of these features are less likely to depart. Churn is positively correlated with monthly fees, suggesting that higher-paying clients are more likely to leave. The majority of other characteristics exhibit weak correlations.

4.5 Engineering Features

To improve the predictive ability of models, feature engineering entails developing new features or altering preexisting ones. To assist the model in identifying patterns linked to churn, for instance, features that represent customer engagement, tenure, or usage patterns are generated.

4.6 Model Creation

To predict churn, a number of machine learning models are created and evaluated:

- **Logistic regression:** This model provides a starting point for estimating the likelihood of churn.
- **Random Forest Classifier**: An ensemble method that increases accuracy and stability by combining several decision trees.

- **XGBoost Classifier:** A gradient boosting technique that efficiently manages intricate patterns and iteratively enhances predictions.

4.7 **Model Assessment**

Several metrics are used to assess each model's performance:

- Confusion Matrix: Indicates how many predictions were right and wrong for each class.

- **Classification Report:** Offers accuracy, precision, recall, and F1-score for a thorough evaluation.

- **ROC-AUC Score:** Indicates how well the model can differentiate between customers who have left and those who have stayed.

4.**8 Comparing and Choosing Models**

The evaluation metrics of each model are used to compare them. The final churn prediction model is the one that strikes the best balance between accuracy, precision, recall, and ROC-AUC score. This guarantees accurate forecasts and permits companies to execute successful retention tactics.

5. Findings and Conversation

The results of the churn prediction analysis are shown in this section, along with an interpretation of the outcomes from various machine learning models. The objective is to assess the models' capacity to precisely identify clients who are likely to leave and offer information that can direct retention tactics.

5.1 Performance of the Model

Three machine learning models—Random Forest Classifier, XGBoost Classifier, and Logistic Regression—were created and evaluated. Metrics like accuracy, precision, recall, F1-score, and ROC-AUC score were used to evaluate their performance.

- Logistic Regression: This model was used as the baseline and had a moderate level of accuracy. Although it could identify general trends in the data, it performed poorly when dealing with intricate patterns, which led to a greater number of false negatives.

- Random Forest Classifier: The model was able to effectively take feature interactions into account thanks to the ensemble approach. When compared to Logistic Regression, Random Forest demonstrated greater accuracy and a better balance between precision and recall. It worked especially well at accurately identifying clients who might leave.

- XGBoost Classifier: In every evaluation metric, this model performed better than Random Forest and Logistic Regression. With the highest accuracy, F1-score, and ROC-AUC score, XGBoost was able to capture complex relationships in the data thanks to its gradient **boosting** mechanism.

5.2 Analysis of confusion matrix

Each model's predictions were shown in detail by the confusion matrices. More misclassifications, especially false negatives, were revealed by logistic regression, suggesting that some churned customers were mistakenly predicted to be retained. XGBoost had the lowest misclassification, indicating its superiority over Random Forest in reducing these errors.

5.3 Analysis of ROC-AUC A model's capacity to differentiate between churned and retained customers is gauged by the ROC-AUC metric. Strong discriminative power was demonstrated by XGBoost's highest AUC score. Logistic Regression performed relatively poorly, but Random Forest also did well.

5.4 Conversation

The analysis shows that when it comes to predicting customer churn, ensemble and gradient boosting models perform better than conventional techniques. The predictive power of the models was improved by effective feature engineering, which included the development of features pertaining to customer engagement, tenure, and usage patterns.

Businesses can reduce revenue loss, increase overall customer satisfaction, and implement proactive retention strategies by accurately identifying high-risk customers. The best model for this task turned out to be XGBoost, which offers decision-makers accurate forecasts and useful information.

In conclusion, the study shows that sophisticated machine learning methods can effectively predict customer attrition when paired with appropriate preprocessing, feature engineering, and assessment. The best performance is provided by XGBoost, which can be a useful tool for companies to predict consumer behavior and enhance retention tactics.

7. Prospects

The churn prediction model created for this project offers businesses useful information. Nonetheless, there are numerous chances to expand and improve the work:

- **Including Real-Time Data**: By incorporating streaming data from customer interactions, real-time churn prediction can be made, allowing for quick retention actions.

- **Additional Features:** Prediction accuracy may be enhanced by elements like customer feedback, social media activity, and support ticket history

.

- **Use of Advanced Deep Learning Models:** For improved prediction, methods like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks can identify sequential patterns in consumer behaviour.

- **Automated Model Deployment:** Businesses can make prompt decisions by creating a dashboard or API for automated churn prediction monitoring.

- **Cross-Industry Application:** To enhance customer retention tactics, the model can be applied to other sectors like banking, e-commerce, and subscription-based services.

6. Implementation of Code

1. Loading libraries and data

```
import pandas as pd
import numpy as np
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import warnings
import missingno as msno
warnings.filterwarnings('ignore')
```

```
!pip install catboost
```

```
Requirement already satisfied: catboost in /usr/local/lib/python3.12/dist-packages (1.2.8)
Requirement already satisfied: graphviz in /usr/local/lib/python3.12/dist-packages (from catboost) (0.21)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from catboost) (3.10.0)
Requirement already satisfied: numpy<3.0,>=1.16.0 in /usr/local/lib/python3.12/dist-packages (from catboost) (2.0.2)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.12/dist-packages (from catboost) (2.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-packages (from catboost) (1.16.3)
Requirement already satisfied: plotly in /usr/local/lib/python3.12/dist-packages (from catboost) (5.24.1)
Requirement already satisfied: six in /usr/local/lib/python3.12/dist-packages (from catboost) (1.17.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=0.24->catboos
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=0.24->catboost) (2025.2
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=0.24->catboost) (2025
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (1.3.
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (4.6
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (1.4
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->catboost) (3.2.
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.12/dist-packages (from plotly->catboost) (9.1.2)
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
from sklearn import metrics
from sklearn.metrics import roc_curve
from sklearn.metrics import recall_score, confusion_matrix, precision_score, f1_score, accuracy_score, classification_repor
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
import pandas as pd
df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecuri
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Y
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Y
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Y
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	

5 rows × 21 columns

---

3. Undertanding the data

Each row represents a customer, each column contains customer's attributes described on the column Metadata.

```
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecuri
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Y
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Y
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Y
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	

5 rows × 21 columns

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've 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

```
df.shape
```

```
(7043, 21)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
```

```
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

```
df.columns.values
```

```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

```
df.dtypes
```

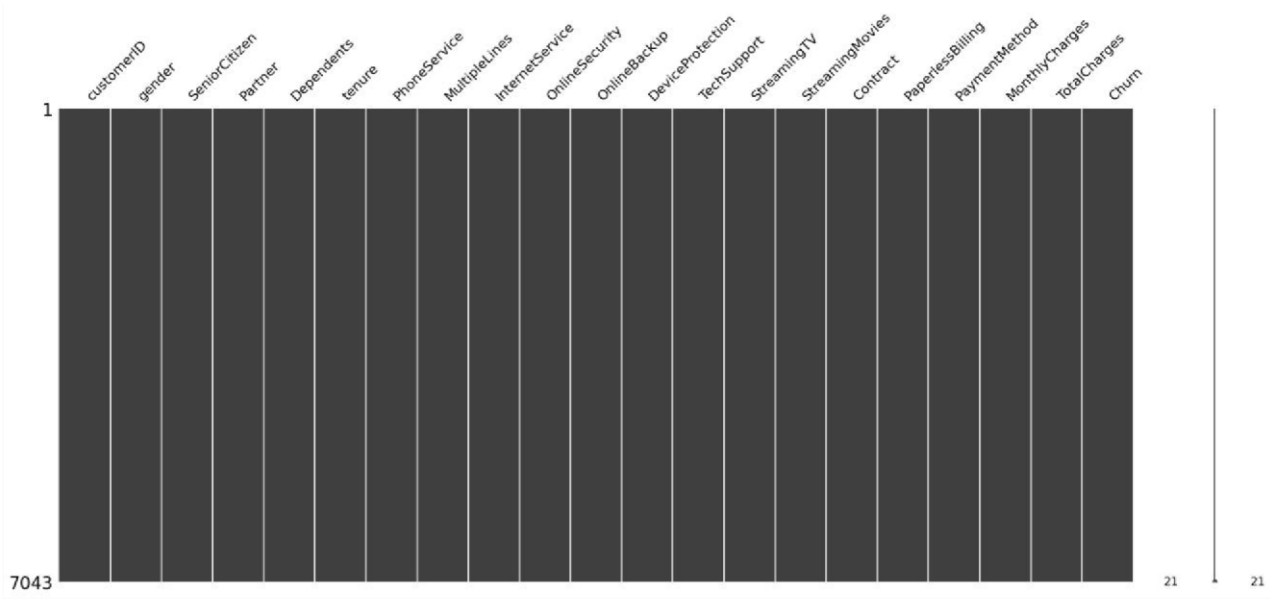
	0
customerID	object
gender	object
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
InternetService	object
OnlineSecurity	object
OnlineBackup	object
DeviceProtection	object
TechSupport	object
StreamingTV	object
StreamingMovies	object
Contract	object
PaperlessBilling	object
PaymentMethod	object
MonthlyCharges	float64
TotalCharges	object
Churn	object

dtype: object

- The target the we will use to guide the exploration is**Churn**

---

4. Visualize missing values



> Using this matrix we can very quickly find the pattern of missingness in the dataset.

- From the above visualisation we can observe that it has no peculiar pattern that stands out. In fact there is no missing data.

```
# Visualize missing values as a matrix
msno.matrix(df);
```

5. Data Manipulation

```
df = df.drop(['customerID'], axis = 1)
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBa
0	Female	0	Yes	No	1	No	No phone service	DSL	No	
1	Male	0	No	No	34	Yes	No	DSL	Yes	
2	Male	0	No	No	2	Yes	No	DSL	Yes	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	

- On deep analysis, we can find some indirect missingness in our data (which can be in form of blankspaces). Let's see that!

```
df['TotalCharges'] = pd.to_numeric(df.TotalCharges, errors='coerce')
df.isnull().sum()
```

	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	11
Churn	0

dtype: int64

- Here we see that the TotalCharges has 11 missing values. Let's check this data.

```
df[np.isnan(df['TotalCharges'])]
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	Onlin
488	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	
753	Male	0	No	Yes	0	Yes	No	No	No internet service	N
936	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	
1082	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	N
1340	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	
3331	Male	0	Yes	Yes	0	Yes	No	No	No internet service	N
3826	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	N
4380	Female	0	Yes	Yes	0	Yes	No	No	No internet service	N
5218	Male	0	Yes	Yes	0	Yes	No	No	No internet service	N
6670	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	
6754	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	

- It can also be noted that the Tenure column is 0 for these entries even though the MonthlyCharges column is not empty.

Let's see if there are any other 0 values in the tenure column.

```
df[df['tenure'] == 0].index
```

There are no additional missing values in the Tenure column.

Let's delete the rows with missing values in Tenure columns since there are only 11 rows and deleting them will not affect the data.

```
df.drop(labels=df[df['tenure'] == 0].index, axis=0, inplace=True)
df[df['tenure'] == 0].index
```

```
Index([], dtype='int64')
```

To solve the problem of missing values in TotalCharges column, I decided to fill it with the mean of TotalCharges values.

```
df.fillna(df["TotalCharges"].mean())
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	Onlin
0	Female	0	Yes	No	1	No	No phone service	DSL	No	
1	Male	0	No	No	34	Yes	No	DSL	Yes	
2	Male	0	No	No	2	Yes	No	DSL	Yes	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	
...	...	...	...	...	...	...	...	...	...	
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	
7041	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes	

7032 rows × 20 columns

```
df.isnull().sum()
```

https://colab.research.google.com/drive/1UghDCODaELghHr0QxnvHpIQzJECUorDK#scrollTo=senior-mistress&printMode=true

	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	0
Churn	0

dtype: int64

```
df["SeniorCitizen"]= df["SeniorCitizen"].map({0: "No", 1: "Yes"})
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBa
0	Female	No	Yes	No	1	No	No phone service	DSL	No	
1	Male	No	No	No	34	Yes	No	DSL	Yes	
2	Male	No	No	No	2	Yes	No	DSL	Yes	
3	Male	No	No	No	45	No	No phone service	DSL	Yes	
4	Female	No	No	No	2	Yes	No	Fiber optic	No	

```
df["InternetService"].describe(include=['object', 'bool'])
```

	InternetService
count	7032
unique	3
top	Fiber optic
freq	3096

dtype: object

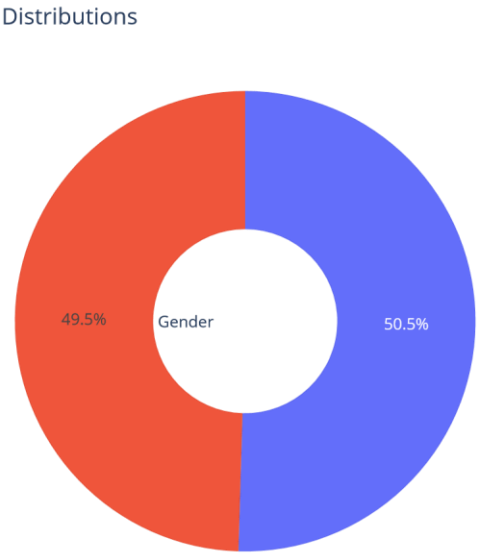
```
numerical_cols = ['tenure', 'MonthlyCharges', 'TotalCharges']
df[numerical_cols].describe()
```

	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000
mean	32.421786	64.798208	2283.300441
std	24.545260	30.085974	2266.771362
min	1.000000	18.250000	18.800000
25%	9.000000	35.587500	401.450000
50%	29.000000	70.350000	1397.475000
75%	55.000000	89.862500	3794.737500
max	72.000000	118.750000	8684.800000

6. Data Visualization

```
g_labels = ['Male', 'Female']
c_labels = ['No', 'Yes']

fig = make_subplots(
    rows=1, cols=2,
    specs=[[{'type': 'domain'}, {'type': 'domain'}]]
)

# Gender Pie
fig.add_trace(
    go.Pie(labels=g_labels, values=df['gender'].value_counts(), hole=0.4, name="Gender"),
    row=1, col=1
)

# Churn Pie
fig.add_trace(
    go.Pie(labels=c_labels, values=df['Churn'].value_counts(), hole=0.4, name="Churn"),
    row=1, col=2
)

fig.update_traces(hoverinfo="label+percent")

fig.update_layout(
    title_text="Gender and Churn Distributions",
    annotations=[
        dict(text='Gender', x=0.18, y=0.5, showarrow=False),
        dict(text='Churn', x=0.82, y=0.5, showarrow=False)
    ]
)

fig.show()
```

https://colab.research.google.com/drive/1UghDCODaELghHr0QxnvHpIQzJECUorDK#scrollTo=senior-mistress&printMode=true



- 26.6 % of customers switched to another firm.
- Customers are 49.5 % female and 50.5 % male.

```
df["Churn"][df["Churn"]=="No"].groupby(by=df["gender"]).count()
```

gender	Churn
Female	2544
Male	2619

dtype: int64

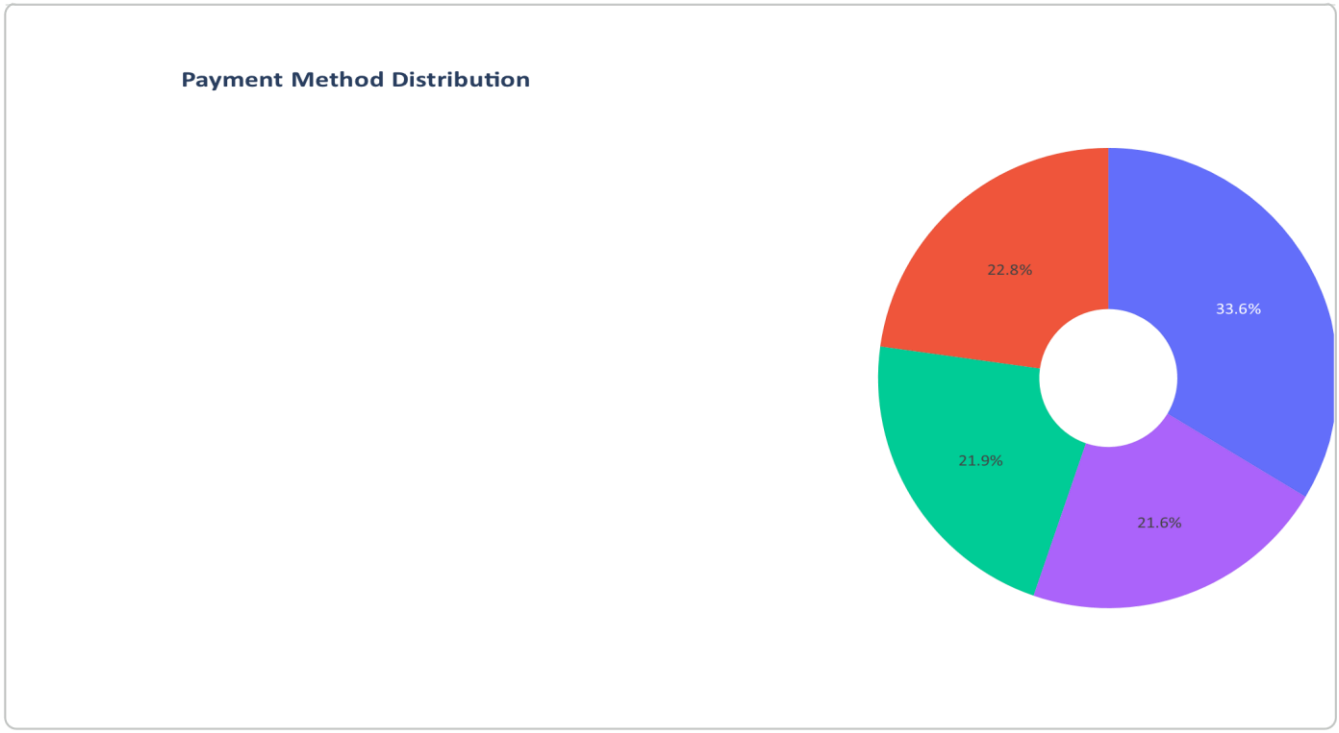
```
df["Churn"][df["Churn"]=="Yes"].groupby(by=df["gender"]).count()
```

gender	Churn
Female	939
Male	930

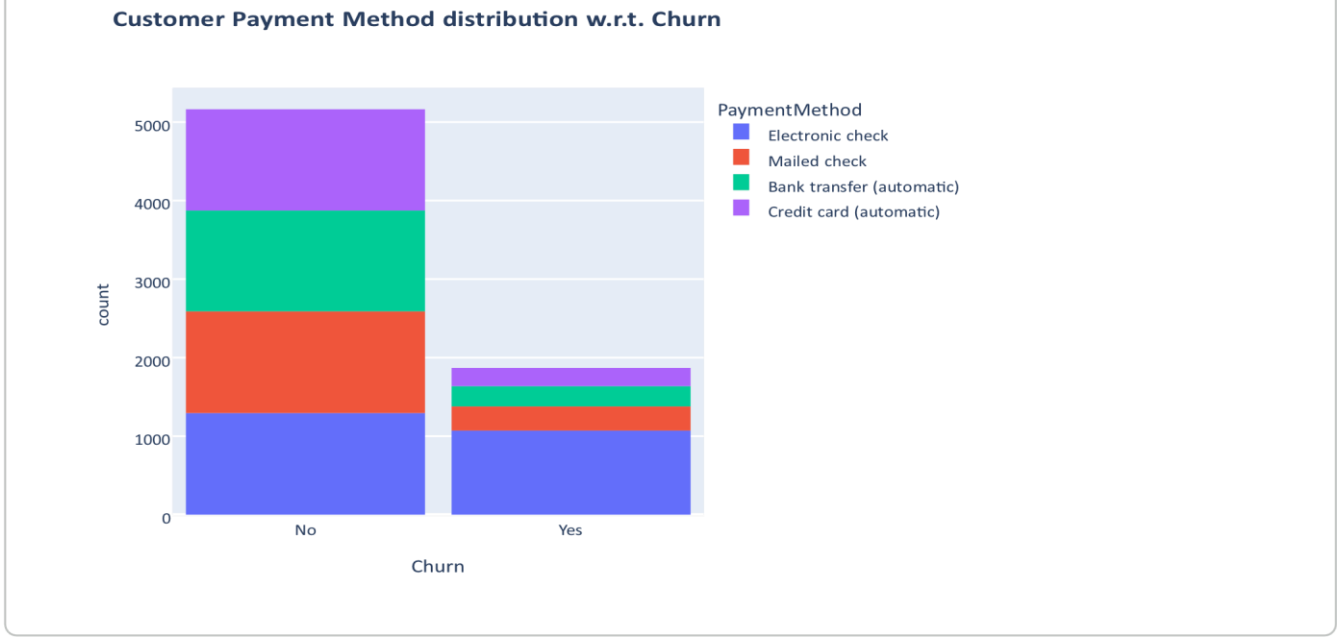
dtype: int64

```
plt.figure(figsize=(6, 6))
labels =["Churn: Yes","Churn:No"]
values = [1869,5163]
labels_gender = ["F","M","F","M"]
sizes_gender = [939,930 , 2544,2619]
colors = ['#ff6666', '#66b3ff']
colors_gender = ['#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6']
explode = (0.3,0.3)
explode_gender = (0.1,0.1,0.1,0.1)
textprops = {"fontsize":15}
#Plot
plt.pie(values, labels=labels,autopct='%1.1f%%',pctdistance=1.08, labeldistance=0.8,colors=colors, startangle=90,frame=True,
plt.pie(sizes_gender,labels=labels_gender,colors=colors_gender,startangle=90, explode=explode_gender,radius=7, textprops =te
#Draw circle
centre_circle = plt.Circle((0,0),5,color='black', fc='white',linewidth=0)
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title('Churn Distribution w.r.t Gender: Male(M), Female(F)', fontsize=15, y=1.1)

# show plot

plt.axis('equal')
plt.tight_layout()
plt.show()
```

```
plt.show()
```




```
fig = px.histogram(df, x="Churn", color="PaymentMethod", title="<b>Customer Payment Method distribution w.r.t. Churn</b>")
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



- Major customers who moved out were having Electronic Check as Payment Method.
- Customers who opted for Credit-Card automatic transfer or Bank Automatic Transfer and Mailed Check as Payment Method were less likely to move out.

```
df["InternetService"].unique()
```

```
array(['DSL', 'Fiber optic', 'No'], dtype=object)
```

```
df[df["gender"]=="Male"][["InternetService", "Churn"]].value_counts()
```

InternetService	Churn	count
DSL	No	992
Fiber optic	No	910
No	No	717
Fiber optic	Yes	633
DSL	Yes	240
No	Yes	57

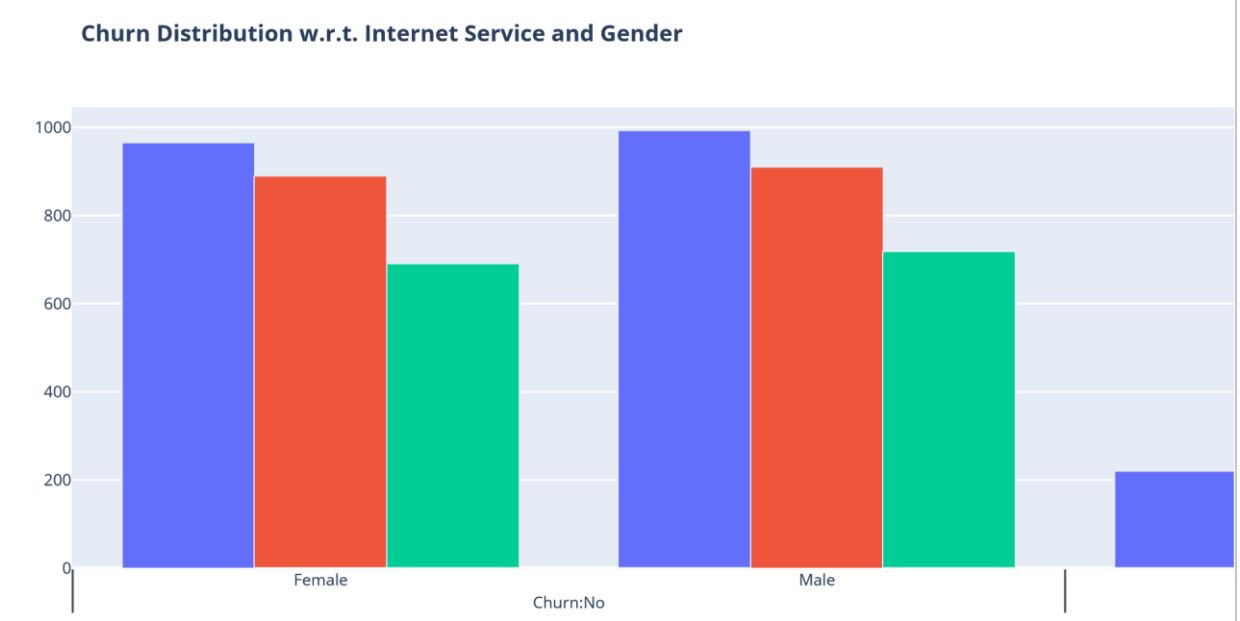
dtype: int64

```
df[df["gender"]=="Female"][["InternetService", "Churn"]].value_counts()
```

InternetService	Churn	count
DSL	No	965
Fiber optic	No	889
No	No	690
Fiber optic	Yes	664
DSL	Yes	219
No	Yes	56

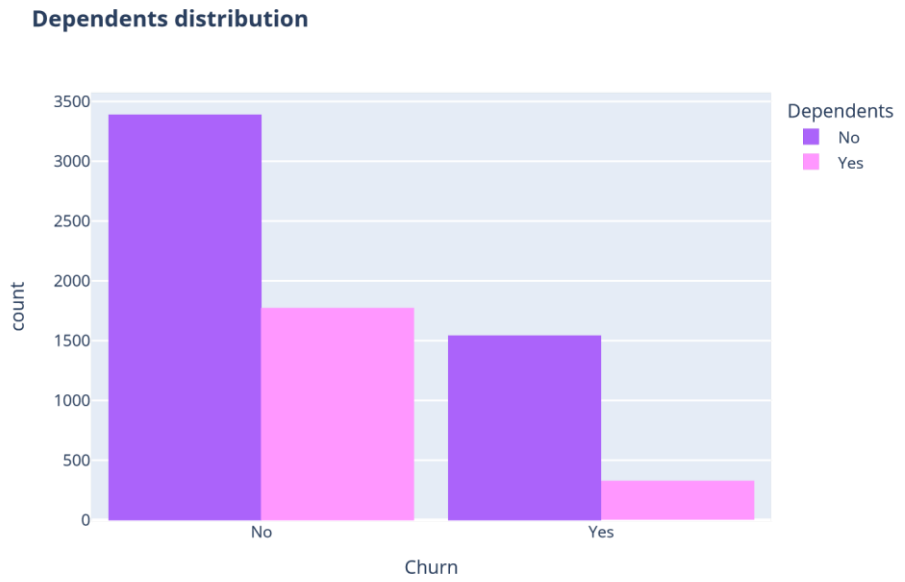
dtype: int64

```
fig = go.Figure()

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [965, 992, 219, 240],
  name = 'DSL',
))

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [889, 910, 664, 633],
  name = 'Fiber optic',
))

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [690, 717, 56, 57],
  name = 'No Internet',
))

fig.update_layout(title_text="<b>Churn Distribution w.r.t. Internet Service and Gender</b>")

fig.show()
```



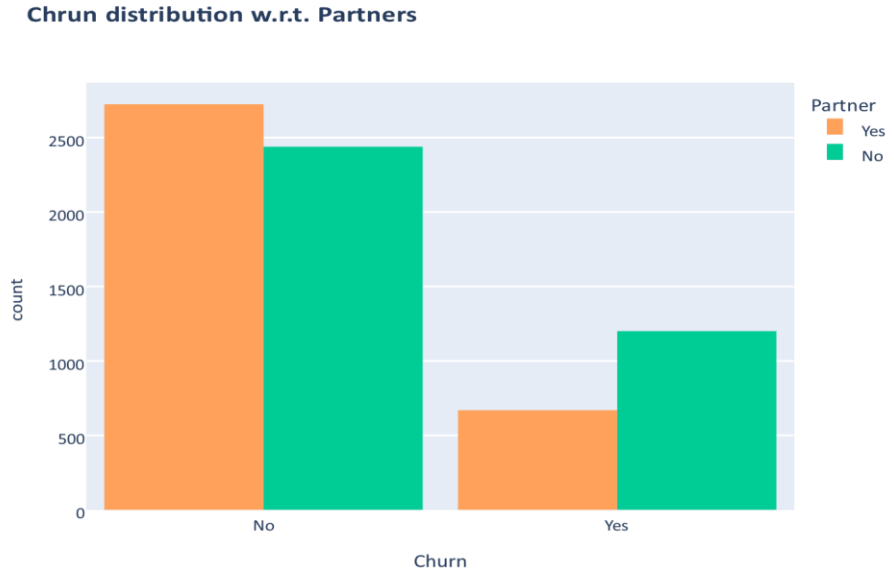
- A lot of customers choose the Fiber optic service and it's also evident that the customers who use Fiber optic have high churn rate, this might suggest a dissatisfaction with this type of internet service.
- Customers having DSL service are majority in number and have less churn rate compared to Fibre optic service.

```
color_map = {"Yes": "#FF97FF", "No": "#AB63FA"}
fig = px.histogram(df, x="Churn", color="Dependents", barmode="group", title="<b>Dependents distribution</b>", color_discr
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



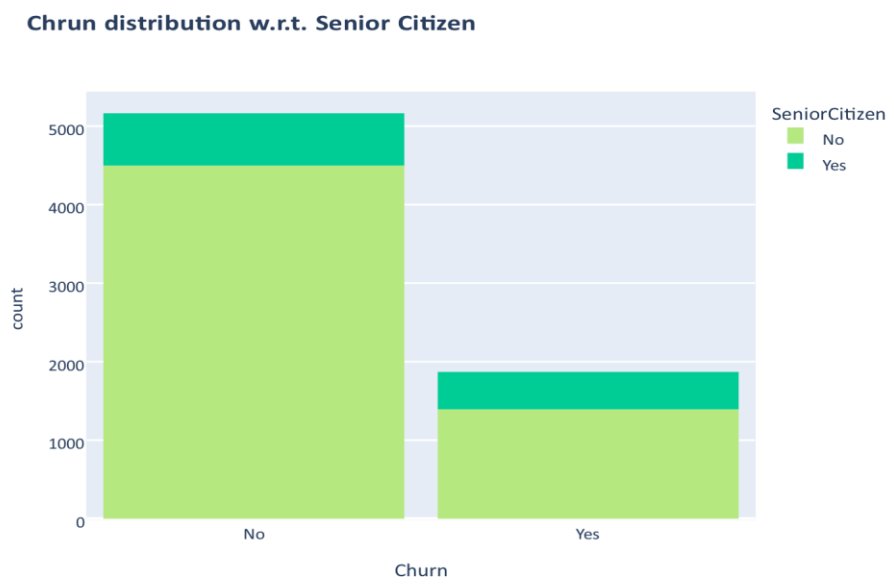
- Customers without dependents are more likely to churn

```
color_map = {"Yes": '#FFA15A', "No": '#00CC96'}
fig = px.histogram(df, x="Churn", color="Partner", barmode="group", title="<b>Chrun distribution w.r.t. Partners</b>", col
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



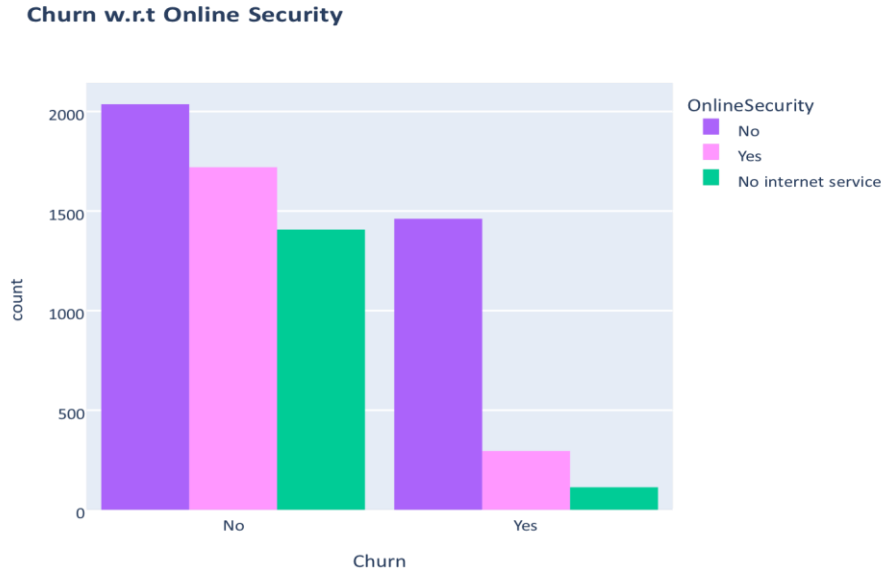
- Customers that doesn't have partners are more likely to churn

```
color_map = {"Yes": '#00CC96', "No": '#B6E880'}
fig = px.histogram(df, x="Churn", color="SeniorCitizen", title="<b>Chrun distribution w.r.t. Senior Citizen</b>", color_dis
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



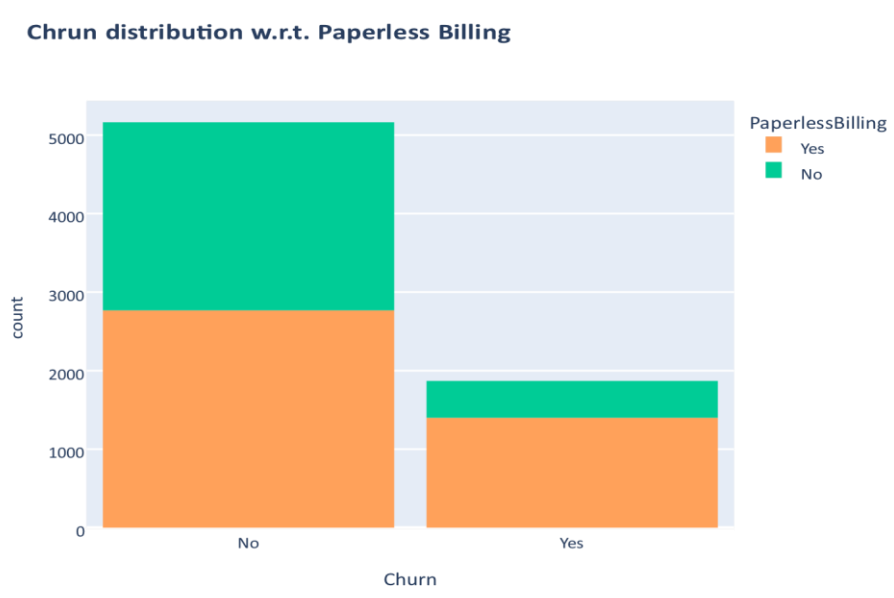
- It can be observed that the fraction of senior citizen is very less.
- Most of the senior citizens churn.

```
color_map = {"Yes": "#FF97FF", "No": "#AB63FA"}
fig = px.histogram(df, x="Churn", color="OnlineSecurity", barmode="group", title="<b>Churn w.r.t Online Security</b>", colo
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



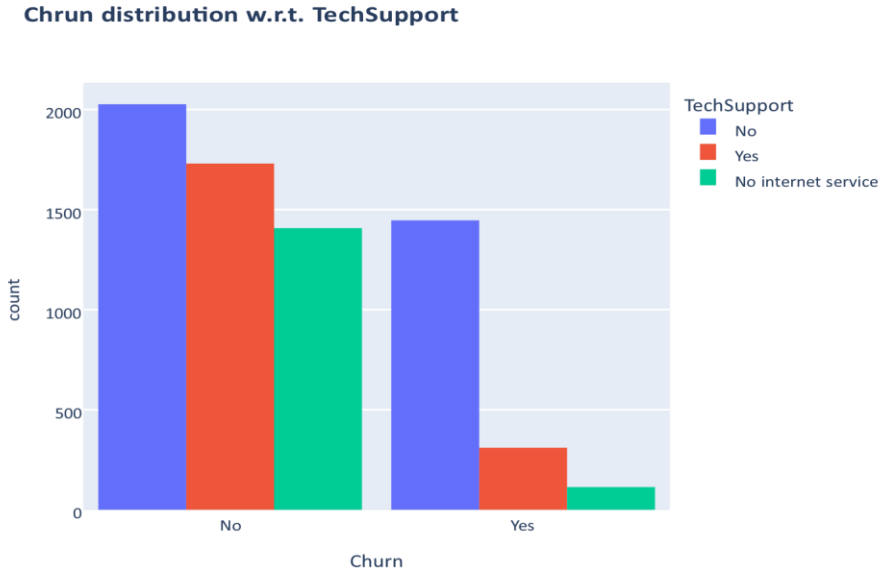
- Most customers churn in the absence of online security,

```
color_map = {"Yes": '#FFA15A', "No": '#00CC96'}
fig = px.histogram(df, x="Churn", color="PaperlessBilling",  title="<b>Chrun distribution w.r.t. Paperless Billing</b>", co
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



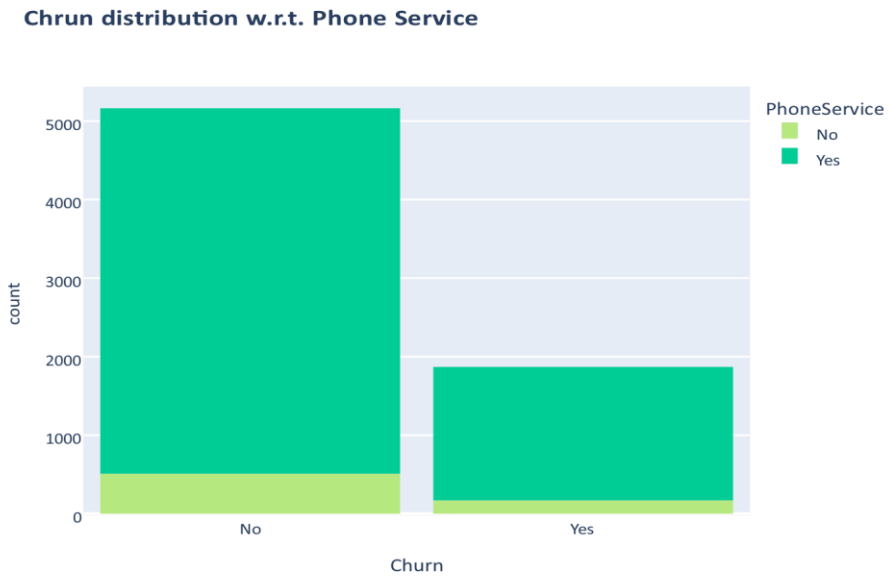
- Customers with Paperless Billing are most likely to churn.

```
fig = px.histogram(df, x="Churn", color="TechSupport",barmode="group",  title="<b>Chrun distribution w.r.t. TechSupport</b>
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



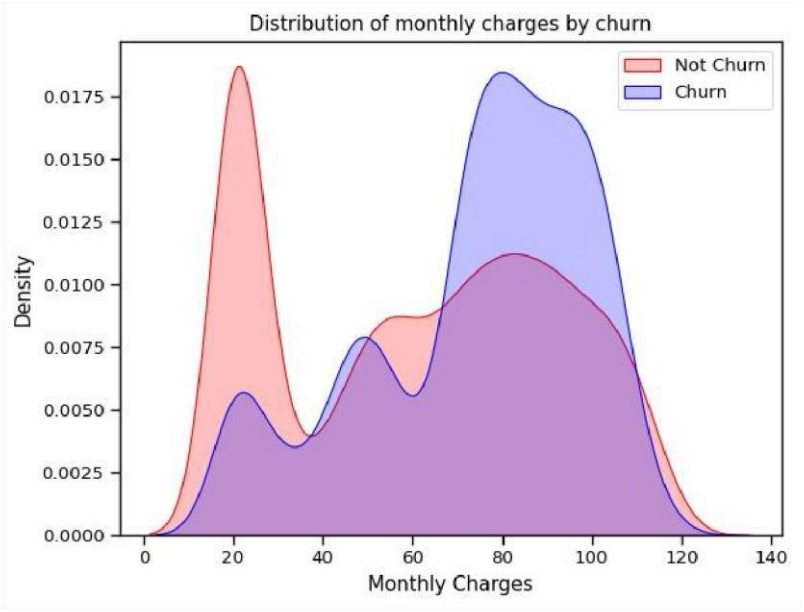
- Customers with no TechSupport are most likely to migrate to another service provider.

```
color_map = {"Yes": '#00CC96', "No": '#B6E880'}
fig = px.histogram(df, x="Churn", color="PhoneService", title="<b>Chrun distribution w.r.t. Phone Service</b>", color_discr
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



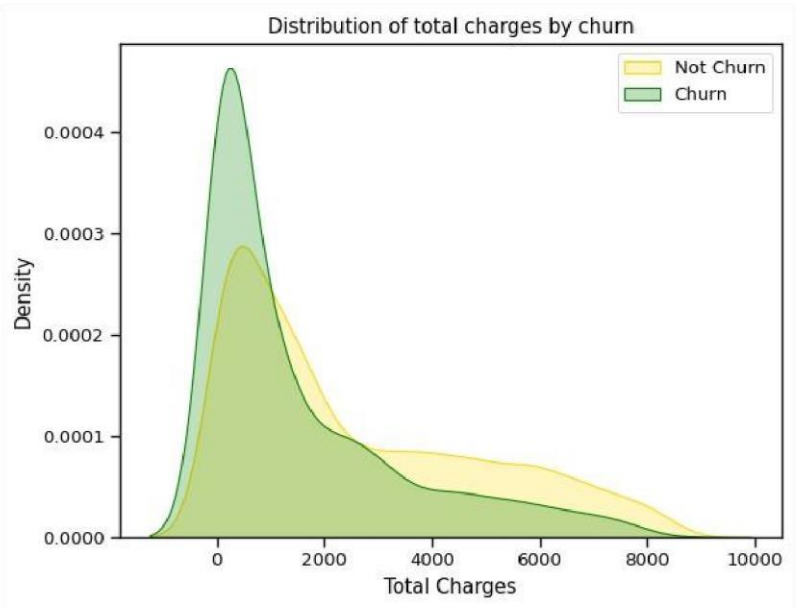
- Very small fraction of customers don't have a phone service and out of that, 1/3rd Customers are more likely to churn.

```
sns.set_context("paper",font_scale=1.1)
ax = sns.kdeplot(df.MonthlyCharges[(df["Churn"] == 'No') ],
                 color="Red", shade = True);
ax = sns.kdeplot(df.MonthlyCharges[(df["Churn"] == 'Yes') ],
                 ax =ax, color="Blue", shade= True);
ax.legend(["Not Churn","Churn"],loc='upper right');
ax.set_ylabel('Density');
ax.set_xlabel('Monthly Charges');
ax.set_title('Distribution of monthly charges by churn');
```

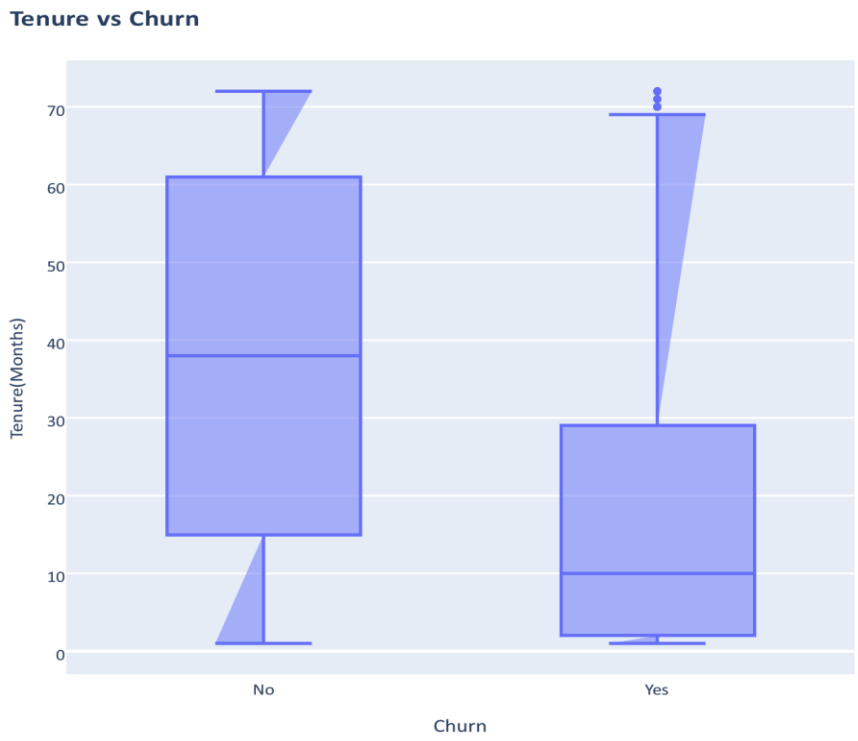


- Customers with higher Monthly Charges are also more likely to churn

```
ax = sns.kdeplot(df.TotalCharges[(df["Churn"] == 'No') ],
                 color="Gold", shade = True);
ax = sns.kdeplot(df.TotalCharges[(df["Churn"] == 'Yes') ],
                 ax =ax, color="Green", shade= True);
ax.legend(["Not Churn","Churn"],loc='upper right');
ax.set_ylabel('Density');
ax.set_xlabel('Total Charges');
ax.set_title('Distribution of total charges by churn');
```

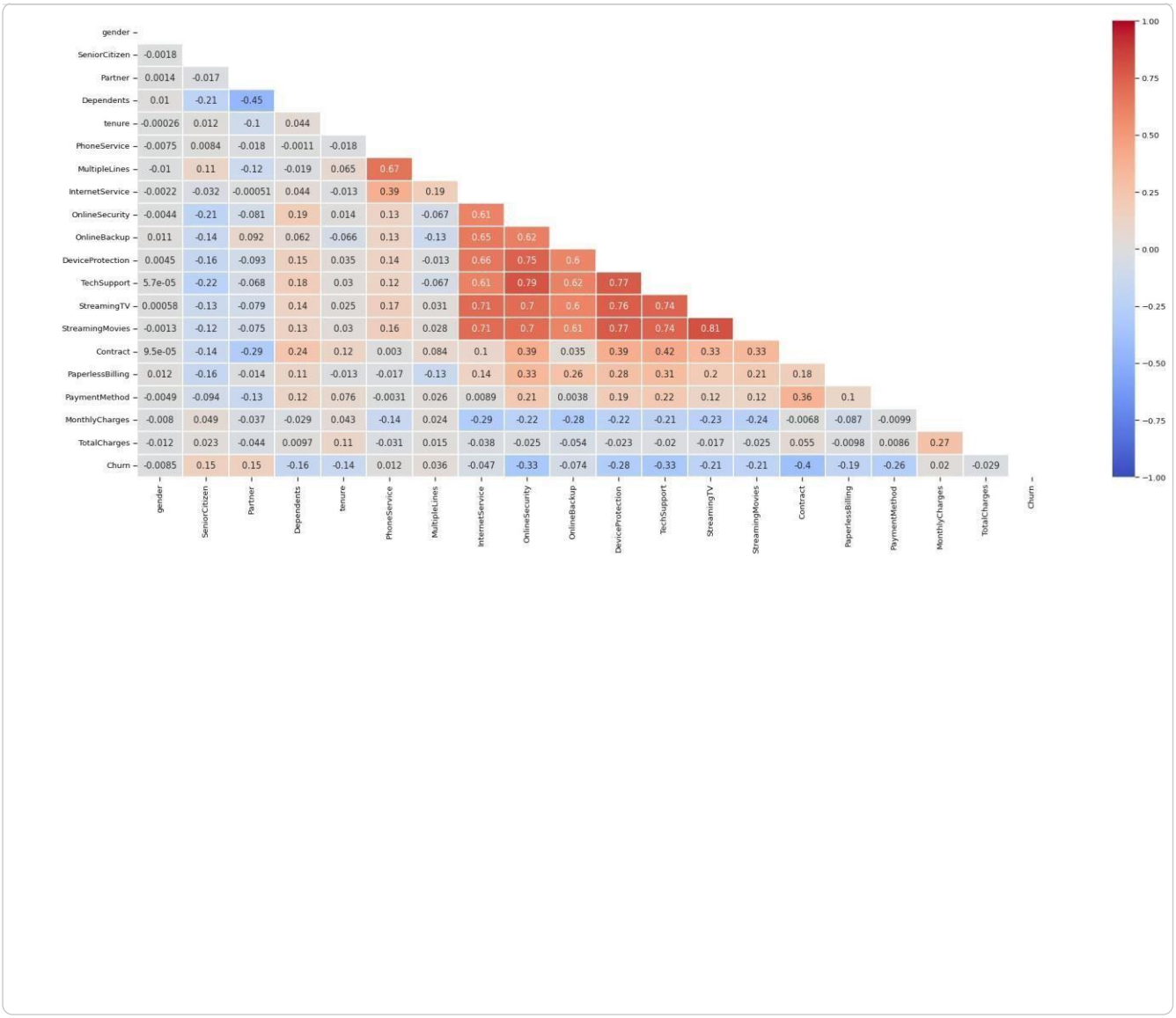



```
fig = px.box(df, x='Churn', y = 'tenure')

# Update yaxis properties
fig.update_yaxes(title_text='Tenure (Months)', row=1, col=1)
# Update xaxis properties
fig.update_xaxes(title_text='Churn', row=1, col=1)

# Update size and title
fig.update_layout(autosize=True, width=750, height=600,
    title_font=dict(size=25, family='Courier'),
    title='<b>Tenure vs Churn</b>',
)

fig.show()
```



- New customers are more likely to churn

```
plt.figure(figsize=(25, 10))

corr = df.apply(lambda x: pd.factorize(x)[0]).corr()

mask = np.triu(np.ones_like(corr, dtype=bool))

ax = sns.heatmap(corr, mask=mask, xticklabels=corr.columns, yticklabels=corr.columns, annot=True, linewidths=.2, cmap='cool
```



7. Data Preprocessing

Splitting the data into train and test sets

```
def object_to_int(dataframe_series):
    if dataframe_series.dtype=='object':
        dataframe_series = LabelEncoder().fit_transform(dataframe_series)
    return dataframe_series
```

```
df = df.apply(lambda x: object_to_int(x))
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBa
0	0	0	1	0	1	0	1	0	0	
1	1	0	0	0	34	1	0	0	2	
2	1	0	0	0	2	1	0	0	2	
3	1	0	0	0	45	0	1	0	2	
4	0	0	0	0	2	1	0	1	0	

```
plt.figure(figsize=(14,7))
df.corr()['Churn'].sort_values(ascending = False)
```

	Churn
Churn	1.000000
MonthlyCharges	0.192858
PaperlessBilling	0.191454
SeniorCitizen	0.150541
PaymentMethod	0.107852
MultipleLines	0.038043
PhoneService	0.011691
gender	-0.008545
StreamingTV	-0.036303
StreamingMovies	-0.038802
InternetService	-0.047097
Partner	-0.149982
Dependents	-0.163128
DeviceProtection	-0.177883
OnlineBackup	-0.195290
TotalCharges	-0.199484
TechSupport	-0.282232
OnlineSecurity	-0.289050
tenure	-0.354049
Contract	-0.396150

dtype: float64

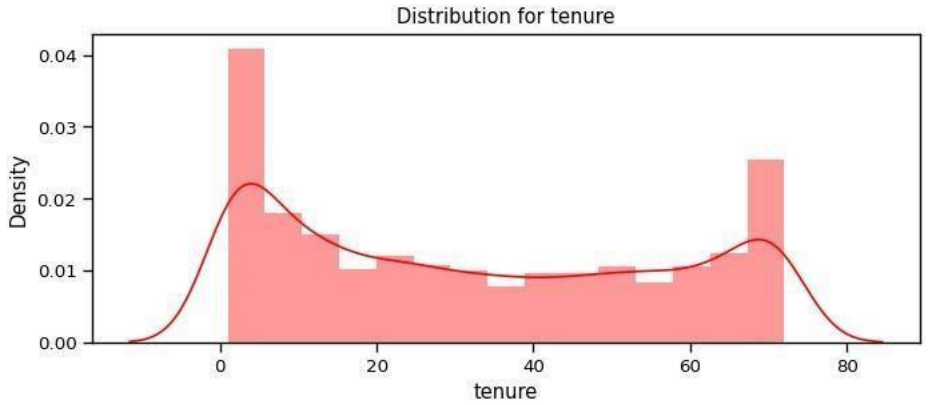
```
<Figure size 1400x700 with 0 Axes>
```

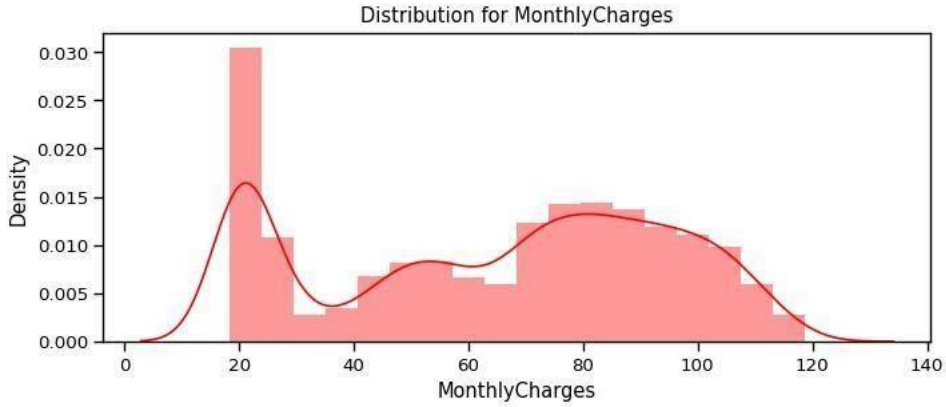
```
X = df.drop(columns = ['Churn'])
y = df['Churn'].values
```

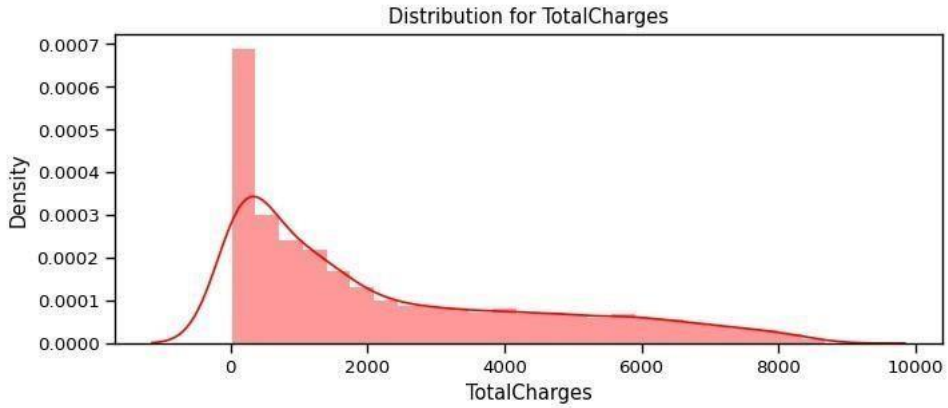
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.30, random_state = 40, stratify=y)
```

```
def distplot(feature, frame, color='r'):
    plt.figure(figsize=(8,3))
    plt.title("Distribution for {}".format(feature))
    ax = sns.distplot(frame[feature], color= color)
```

```
num_cols = ["tenure", 'MonthlyCharges', 'TotalCharges']
for feat in num_cols: distplot(feat, df)
```



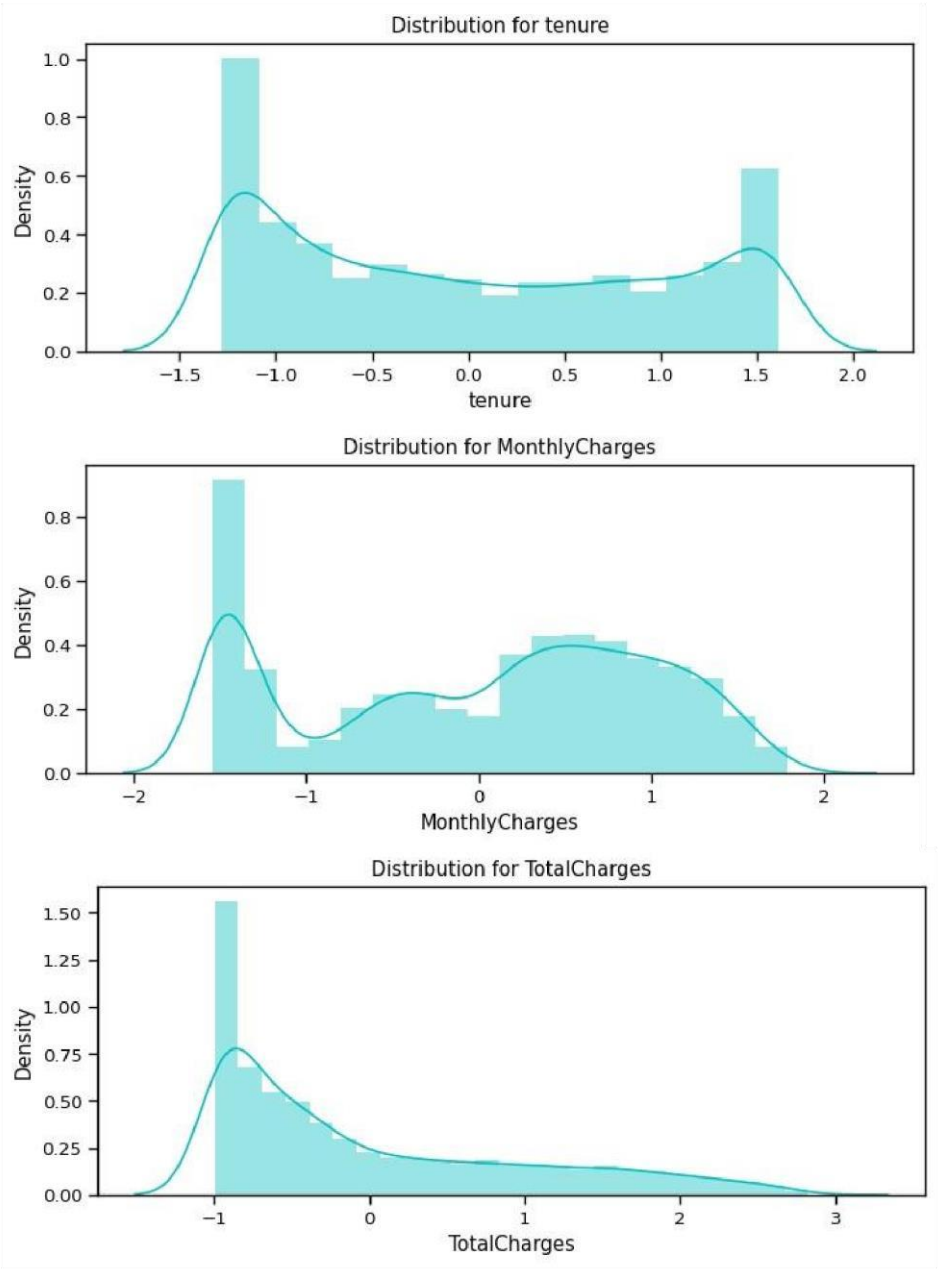




Since the numerical features are distributed over different value ranges, I will use standard scalar to scale them down to the same range.

Standardizing numeric attributes

```
df_std = pd.DataFrame(StandardScaler().fit_transform(df[num_cols].astype('float64')),
                      columns=num_cols)
for feat in numerical_cols: distplot(feat, df_std, color='c')
```




```
# Divide the columns into 3 categories, one ofor standardisation, one for label encoding and one for one hot encoding

cat_cols_ohe =['PaymentMethod', 'Contract', 'InternetService'] # those that need one-hot encoding
cat_cols_le = list(set(X_train.columns)- set(num_cols) - set(cat_cols_ohe)) #those that need label encoding
```

```
scaler= StandardScaler()

X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.transform(X_test[num_cols])
```

8. Machine Learning Model Evaluations and Predictions

AI-Workbench-Predict-propensity-churn-notebook.png

```
knn_model = KNeighborsClassifier(n_neighbors = 11)
knn_model.fit(X_train,y_train)
predicted_y = knn_model.predict(X_test)
accuracy_knn = knn_model.score(X_test,y_test)
print("KNN accuracy:",accuracy_knn)
```

```
KNN accuracy: 0.7758293838862559
```

```
print(classification_report(y_test, predicted_y))
```

```
              precision    recall  f1-score   support

           0       0.83      0.87      0.85      1549
           1       0.59      0.52      0.55       561

    accuracy                           0.78      2110
   macro avg       0.71      0.69      0.70      2110
weighted avg       0.77      0.78      0.77      2110
```

KNN

Random Forest

8.Conclusion

A major obstacle for companies looking to sustain growth and revenue is customer attrition. The goal of this project was to use machine learning techniques to predict customer churn and provide useful information for retention tactics.
Understanding the dataset, preprocessing the data, conducting exploratory data analysis, engineering features, creating several machine learning models, and assessing their effectiveness were all part of the methodology. With the highest accuracy, F1-score, and ROC-AUC score of all the models tested, XGBoost was found to be the most effective.

The findings demonstrate that sophisticated machine learning algorithms can accurately forecast customer attrition when paired with meticulous feature engineering and preprocessing. Businesses can identify high-risk clients, lower attrition, allocate resources optimally, and improve overall customer satisfaction by putting such predictive models into practice.

9. Citations

1. Bhartiprasad 17 (2023). Kaggle Predicts Customer Churn. https://www.kaggle.com/code/bhartiprasad17/customer-churn-prediction 2. J. Brownlee (2016). Python Machine Learning Mastery. mastery of machine learning.

3. Pedregosa, F., Gramfort, A., Varoquaux, G., et al. (2011). Scikit-learn: Python Machine Learning. Machine Learning Research Journal, 12, 2825–2830.
4. Guestrin, C., and Chen, T. (2016). A Scalable Tree Boosting System is called XGBoost. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Proceedings, 785–794.
5. Han, J., Pei, J., and Kamber, M. (2012). Data Mining: Principles and Methods (3rd ed.). Kaufmann, Morgan.