Phase 5: Project Documentation & Submission

Customer Churn Prediction

# Project's objective:

* Introduction
* Data Collection
* Data Analysis
* Model Development
* Prediction and Interpretation
* Reporting and Communication
* Conclusion

# Introduction:

The purpose of customer churn prediction is to proactively identify and predict which customers are at risk of leaving or discontinuing their relationship with a business, product, or service. This predictive analysis aims to help organizations retain customers and reduce attrition by allowing them to take targeted actions to retain at-risk customers. By understanding the factors that contribute to churn and using predictive models, businesses can develop strategies to improve customer satisfaction, address issues, and increase customer loyalty, ultimately leading to enhanced customer retention and long-term profitability. Customer churn prediction is a valuable tool in customer relationship management and business sustainability.

## Data Collection:

Customer Databases: Customer databases are fundamental for customer churn prediction. They contain customer information such as demographics, contact details, purchase history, subscription details, and transaction records. This data provides insights into a customer's historical behavior and interactions with the business.

User Logs: User logs, especially for online businesses or services, capture detailed information about customer interactions. These logs may include website usage patterns, app usage data, clickstream data, and login/logout records. Analyzing user logs can help identify behavioral patterns that may indicate a customer's intent to churn.

Customer Feedback: Customer feedback is a valuable source of data, including surveys, reviews, comments, and complaints. Sentiment analysis of this feedback can help identify dissatisfied customers or those who have expressed an intention to leave. Feedback can be collected through online surveys, social media, and customer support interactions.

Data Analysis:

Data Overview: Begin by loading the dataset and obtaining a high-level overview of its structure and format.

Data Summary: Calculate summary statistics for numerical variables, such as mean, median, standard deviation, minimum, and maximum.

Missing Data: Identify and count missing values in the dataset for each variable.

Decide on a strategy for handling missing data

Model Development:

Ensemble Models:

Ensemble methods combine predictions from multiple machine learning models to produce a more accurate and robust prediction. Some popular ensemble techniques include:

**a. Random Forests:** This ensemble method builds multiple decision trees and combines their predictions. It is effective for both classification and regression tasks.

**b. Gradient Boosting Machines (GBM):** Algorithms like XGBoost, LightGBM, and CatBoost use boosting techniques to build an ensemble of decision trees sequentially. They are known for their high predictive accuracy.

**c. Stacking:** Stacking combines the predictions of multiple base models using another model, called a meta-learner or blender. It can be an effective way to leverage the strengths of different algorithms.

**d. Voting:** In voting ensembles, multiple models make predictions, and the final prediction is based on a majority vote (for classification) or an average (for regression).

Prediction and Interpretation:

Churn Prediction:

Apply the trained machine learning models to make predictions on the dataset, indicating which customers are likely to churn.

Model Outputs:

Obtain prediction scores or probabilities for each customer, representing their likelihood of churning.

Convert these scores into binary churn/non-churn predictions using a chosen threshold

Model Performance Evaluation:

Assess the performance of the predictive models using relevant evaluation metrics b. Use techniques like cross-validation to validate the model's performance and ensure its generalizability.

Post-Processing:

If necessary, apply post-processing techniques to the predictions, such as adjusting the prediction threshold to optimize specific performance metrics or balancing precision and recall.

Python Program:

import pandas as pd

df = pd.read\_csv('/media/johnzavax/Data/WA\_Fn-UseC\_-Telco-Customer-Churn.csv')

df.head()

df.info()

df["Churn"].value\_counts()

import matplotlib.pyplot as plt import seaborn as sns import numpy as np cols = ['gender','SeniorCitizen',"Partner","Dependents"] numerical = cols plt.figure(figsize=(20,4)) for i, col in enumerate(numerical): ax = plt.subplot(1, len(numerical), i+1) sns.countplot(x=str(col), data=df) ax.set\_title(f"{col}")

sns.boxplot(x='Churn', y='MonthlyCharges', data=df)

cols = ['InternetService',"TechSupport","OnlineBackup","Contract"]

plt.figure(figsize=(14,4))

for i, col in enumerate(cols):

ax = plt.subplot(1, len(cols), i+1)

sns.countplot(x ="Churn", hue = str(col), data = df)

ax.set\_title(f"{col}")

df['TotalCharges'] = df['TotalCharges'].apply(lambda x: pd.to\_numeric(x, errors='coerce')).dropna()

cat\_features = df.drop(['customerID','TotalCharges','MonthlyCharges','SeniorCitizen','tenure'],axis=1)

cat\_features.head()

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

df\_cat = cat\_features.apply(le.fit\_transform)

df\_cat.head()

num\_features = df[['customerID','TotalCharges','MonthlyCharges','SeniorCitizen','tenure']]

finaldf = pd.merge(num\_features, df\_cat, left\_index=True, right\_index=True)

from sklearn.model\_selection import train\_test\_split

finaldf = finaldf.dropna()

finaldf = finaldf.drop(['customerID'],axis=1)

X = finaldf.drop(['Churn'],axis=1)

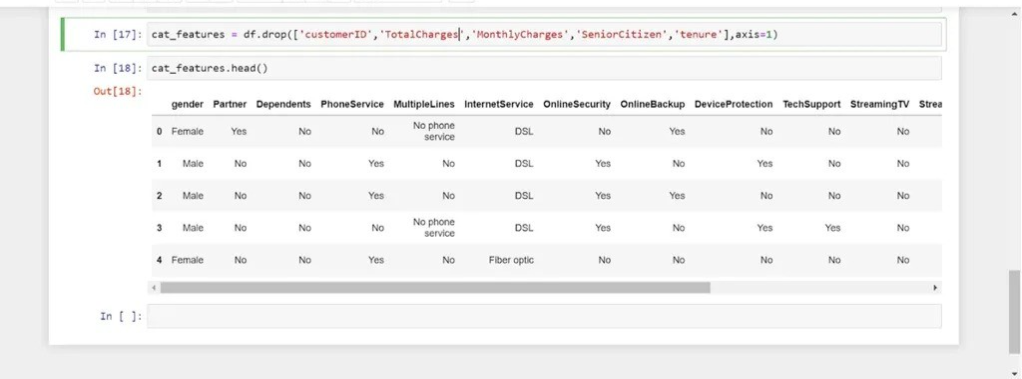
y = finaldf['Churn']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)

from sklearn.metrics import accuracy\_score preds = rf.predict(X\_test) print(accuracy\_score(preds,y\_test))

Output:

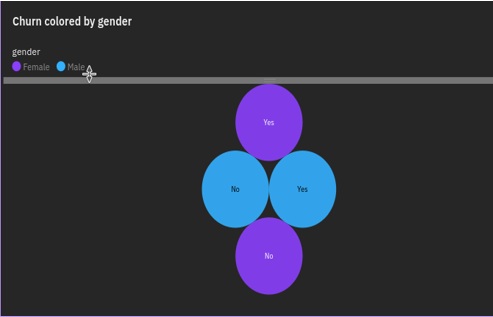
**accuracy of approximately 0.78 on the test dataset.**


# Reporting and Communication:

Churn by Gender:

* Churn No has the highest total TotalCharges due to gender Female
* Gender Male has the highest TotalCharges at over 8.1 million, out of which Churn No contributed the most at nearly 6.6 million.

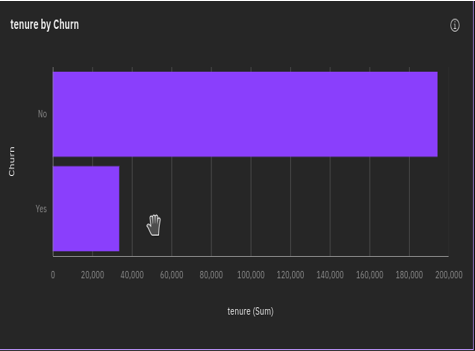


Churn By SeniorCitizen:

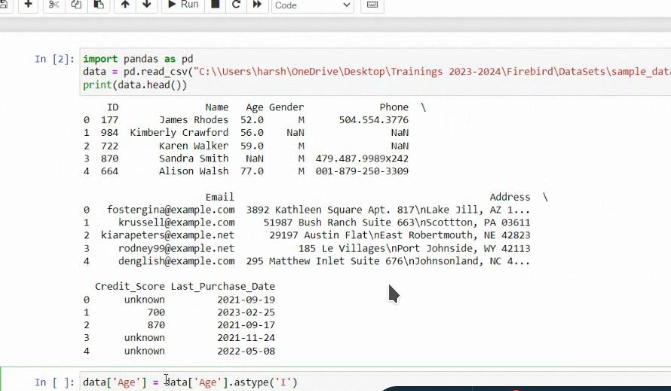
* Churn No has the highest values of both SeniorCitizen and TotalCharges.
* Across all values of Churn, the sum of SeniorCitizen is over a thousand.
* SeniorCitizen ranges from 476, when Churn is Yes, to 666, when Churn is No.

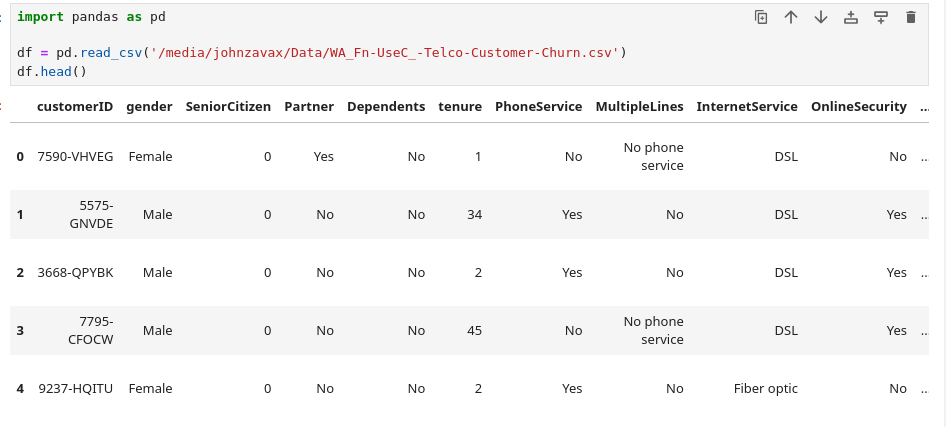
Churn By Tenure:

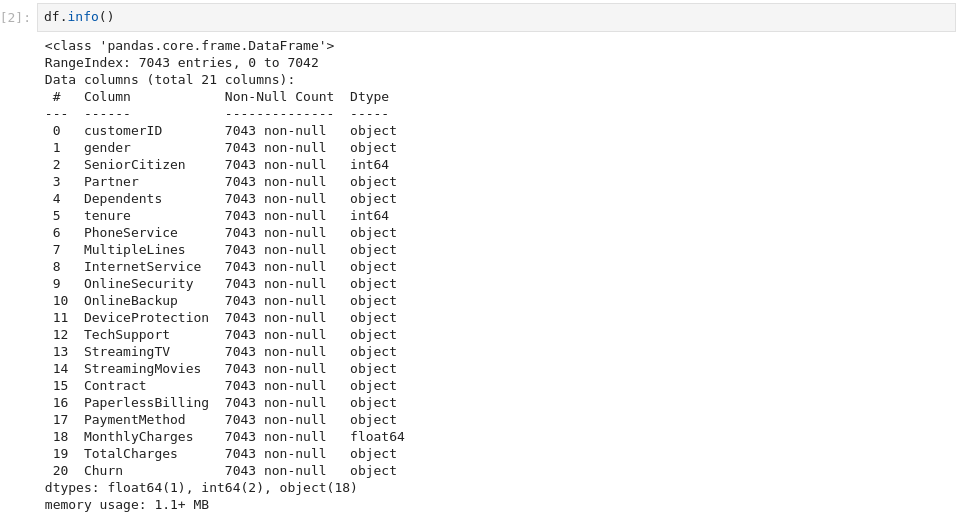
* No exceeds Yes in tenure by 160,784.
* Churn **No** has the highest values of both tenure and TotalCharges.
* Over all values of Churn, the sum of tenure is almost 228 thousand.
* Tenure ranges from almost 34 thousand, when Churn is Yes, to over 194 thousand, when Churn is No.

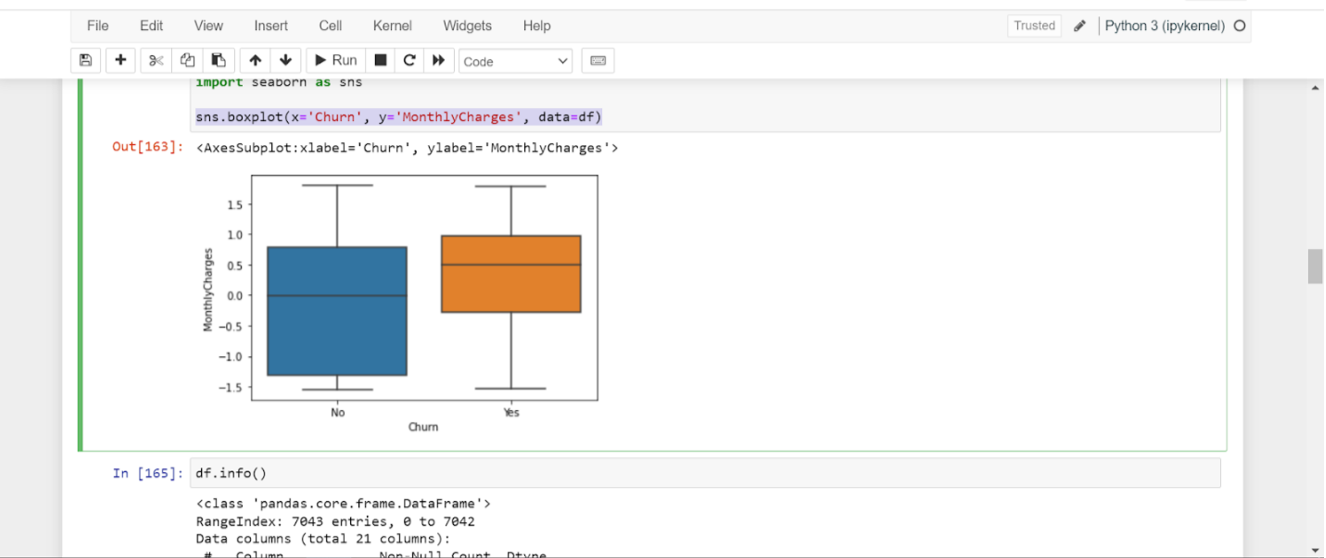


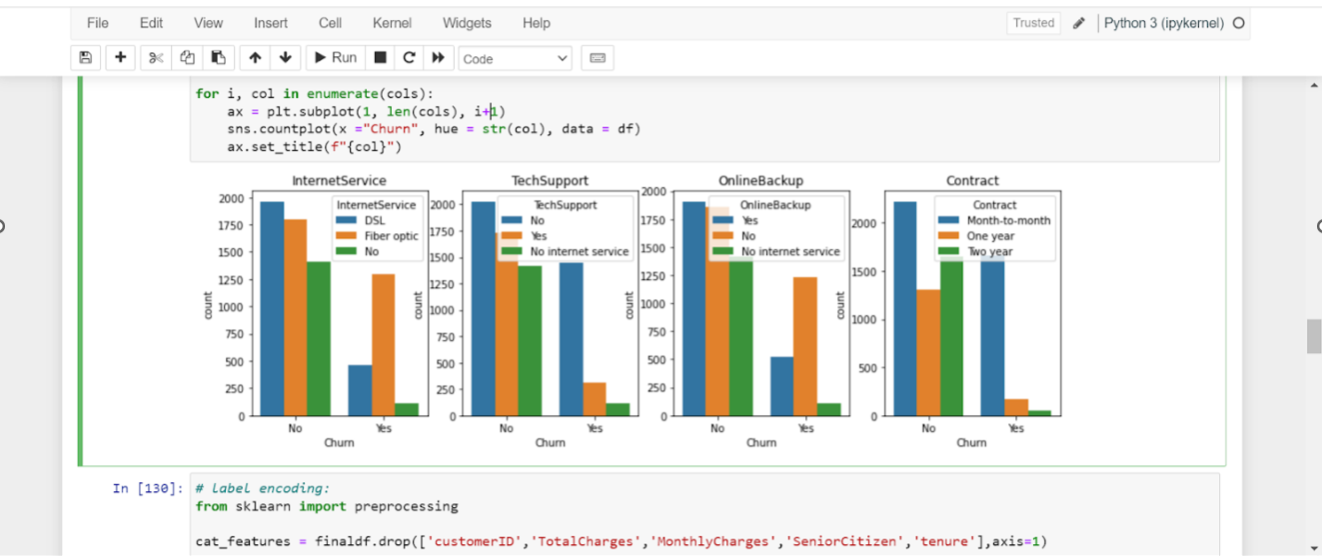
Jupyter Lab Program:











# Conclusion:

In conclusion, customer churn prediction is a IBM tool for businesses seeking to retain their customer base and enhance long-term sustainability. This predictive analysis enables organizations to proactively identify customers at risk of leaving and develop targeted strategies to reduce churn.

The success of a customer churn prediction project lies not only in the accuracy of the predictive models but also in the ability to understand the "why" behind customer behavior. By leveraging data insights, we have uncovered the key drivers of churn and have segmented the customer base to tailor retention efforts effectively.

In a rapidly evolving business landscape, customer churn prediction remains a dynamic field, offering opportunities for ongoing improvement and innovation. By embracing this continuous learning process, businesses can foster customer loyalty, drive growth, and ultimately thrive in a competitive marketplace.