CAPSTONE - FINAL PROJECT

DSBA

Krishnabhamini Sinha

Contents

Data Dictionary (Customer Churn Data)

- AccountID - account unique identifier
- Churn - account churn flag (Target)
- Tenure - Tenure of account
- City_Tier - Tier of primary customer's city
- CC_Contacted_L12m - How many times all the customers of the account has contacted customer care in last 12months
- Payment - Preferred Payment mode of the customers in the account
- Gender - Gender of the primary customer of the account
- Service_Score - Satisfaction score given by customers of the account on service provided by company
- Account_user_count - Number of customers tagged with this account
- account_segment - Account segmentation on the basis of spend
- CC_Agent_Score - Satisfaction score given by customers of the account on customer care service provided by company
- Marital_Status - Marital status of the primary customer of the account
- rev_per_month - Monthly average revenue generated by account in last 12 months
- Complain_l12m - Any complaints has been raised by account in last 12 months
- rev_growth_yoy - revenue growth percentage of the account (last 12 months vs last 24 to 13 month)
- coupon_used_l12m - How many times customers have used coupons to do the payment in last 12 months
- Day_Since_CC_connect - Number of days since no customers in the account has contacted the customer care
- cashback_l12m - Monthly average cashback generated by account in last 12 months
- Login_device - Preferred login device of the customers in the account

Problem 1 – Introduction

Defining the problem statement and the need for solving it.

Solution:

Defining the problem statement:

Context

An DTH provider is facing a lot of competition in the current market, and it has become a challenge to retain the existing customers in the current situation. Hence, the company wants to develop a model through which they can perform churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers. Hence by losing one account the company might be losing more than one customer.

Objective

We have been assigned to develop a churn prediction model for this company and provide business recommendations on the campaign.

Need for the study/project

The need for this project is essential to the success of the company since customer churn prediction will help the company optimize it sales by identifying loyal and risky customers.

While loyal customers can generate revenue for the company, risky customers might simply hinder the company's growth by causing deviations and riff-raff in business plans like subscriptions and account segments. This kind of behavior simply consumes time, resources and revenue of the company – putting the sales figures in loss.

This project, by predicting the churn rate of customers, can help the company identify and place importance on its loyal customers via curated offers and discounts that may trigger the customer to invest more in the company's products.

- <u>Business Opportunity:</u>

1. Revenue Optimization - retaining existing customers more cheaper than acquiring new ones.
2. Customer Lifetime Value (CLV) Enhancement - businesses can personalize engagement and extend CLV through tailored offers and loyalty programs.

3. Reduced Marketing Costs - businesses can particularly focus efforts on customers with a higher risk of leaving.
4. Competitive Advantage - allows companies to differentiate themselves through proactive customer support and better service.
5. Data-Driven Decision-Making - companies can refine and modify product offerings, pricing, and service models by analyzing customer behavior patterns.

- Social Opportunity:

1. Enhanced Customer Experience - predicting churn can help businesses address service gaps that ensures that customers feel valued and heard.
2. Job Creation & Upskilling - data-driven churn prediction creates demand for data analysts, data scientists, and business analysts.
3. Ethical Customer Engagement - companies can adopt a customer-centric approach, providing value-based engagement instead of pressuring customers to stay.
4. Industry-wide Improvement - insights from churn analysis can influence better industry policies, improving transparency and service standards.

Problem 2 – Exploratory Data Analysis and Business Implication

Solution:

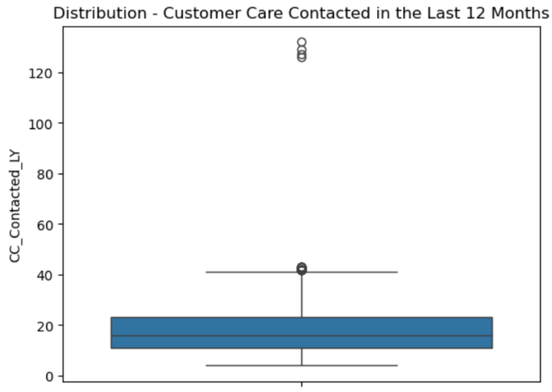
Univariate Analysis:

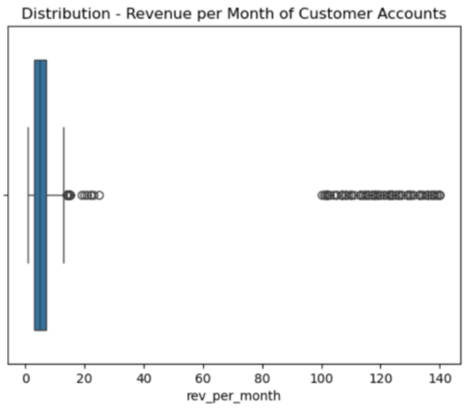
Given below are the observations made on categorical variables after univariate analysis.

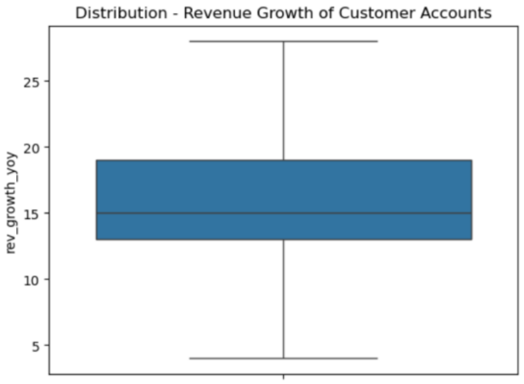
Name of feature/variable in dataset.	**Maximum value/values**	**Minimum value/values**
City_tier	Most customers reside in tier 1 cities.	Number of customers residing in tier 2 cities is the least.
Payment_mode	Most payments are done via debit/credit cards.	Payments are done least using UPI, COD and E-wallet.
Gender	Majority customers are male.	Minority customers are male.
Account_user_count	Majority of accounts have 4 and 3 customers tagged to them.	The accounts tagged with 1, 2 and 6 customers are least in number.
Account_segment	Most customers have Regular Plus and Super accounts.	Customers have Regular and Super Plus accounts in the least.
Marital_status	Most of the customers are married.	Least number of customers are divorced.

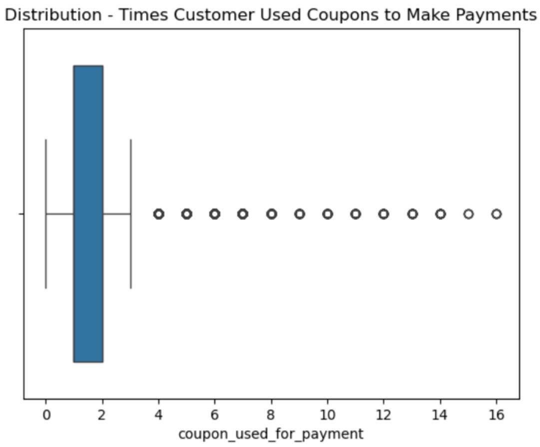
Login_device	Most customers use their mobile phones to log into the company website/app.	Less number of customers use computers to login to the company website/app.

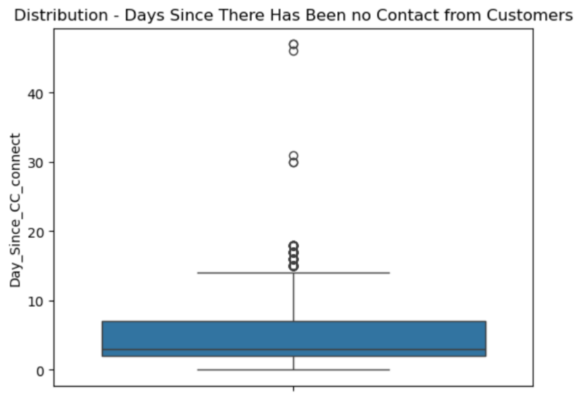
Univariate Analysis done on numerical features:

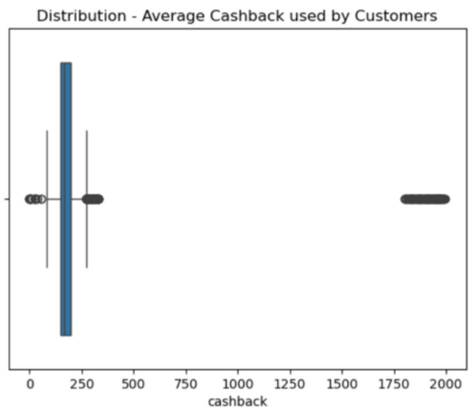


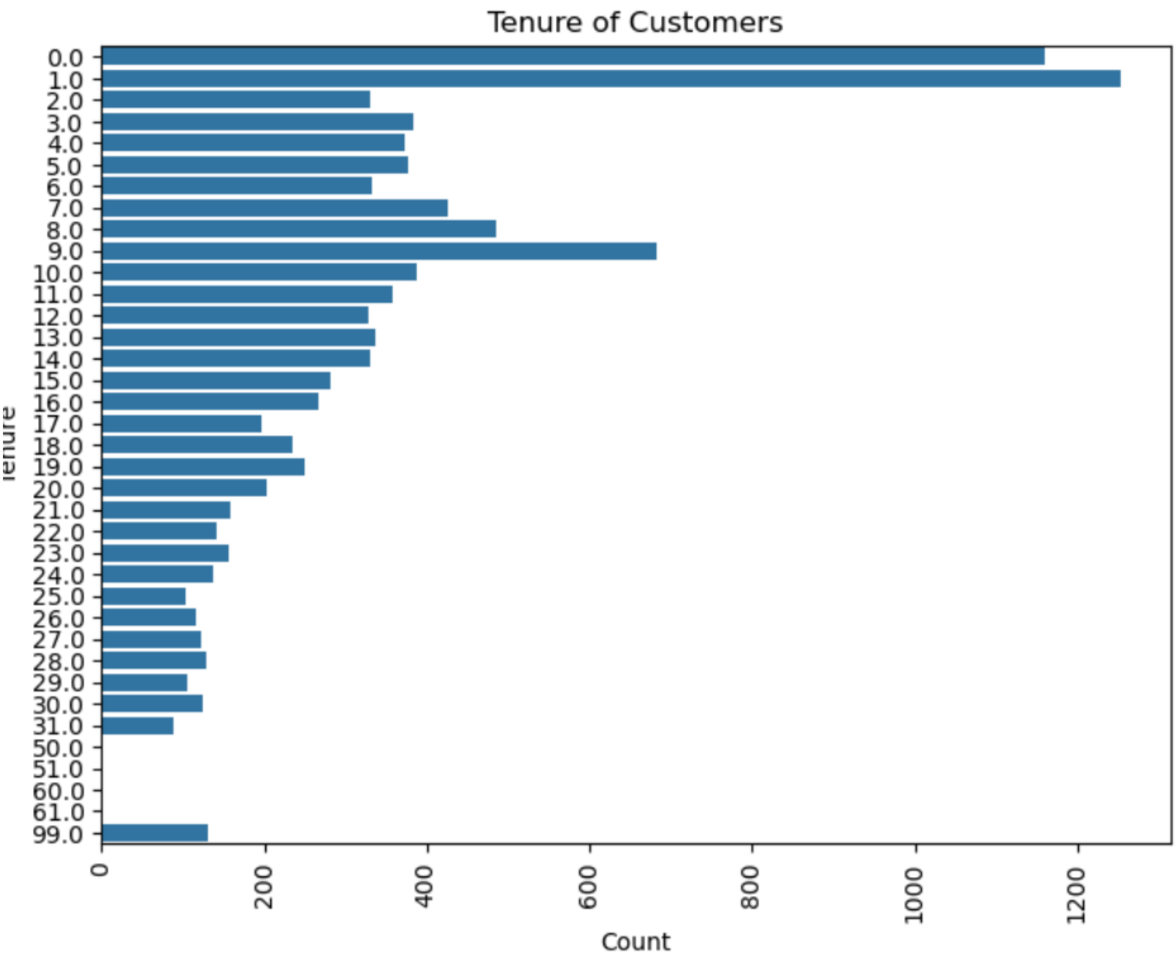






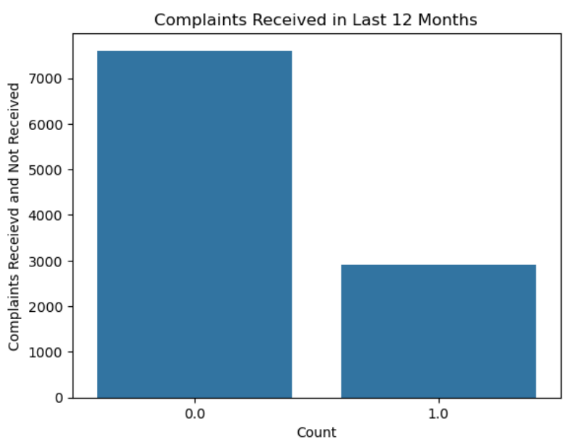


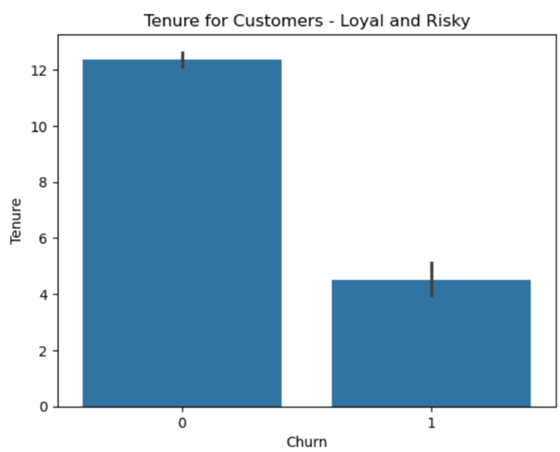


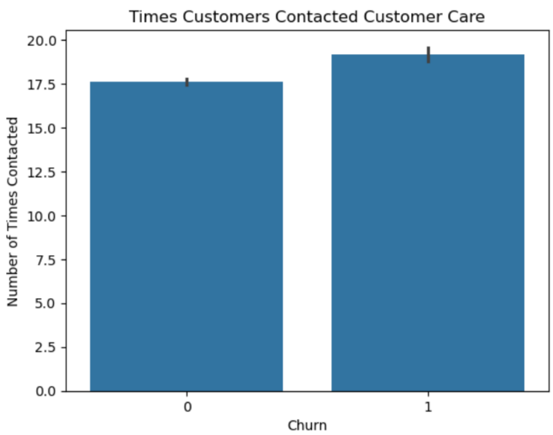


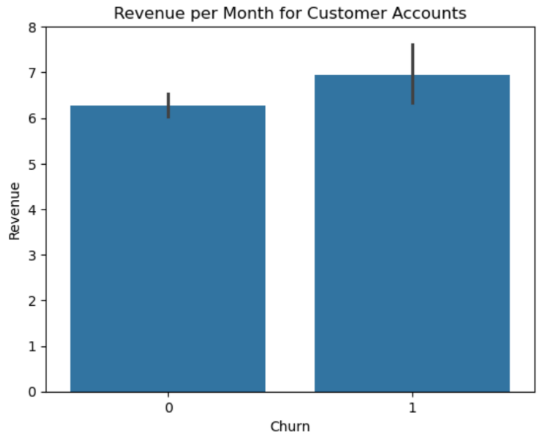
Bivariate Analysis:

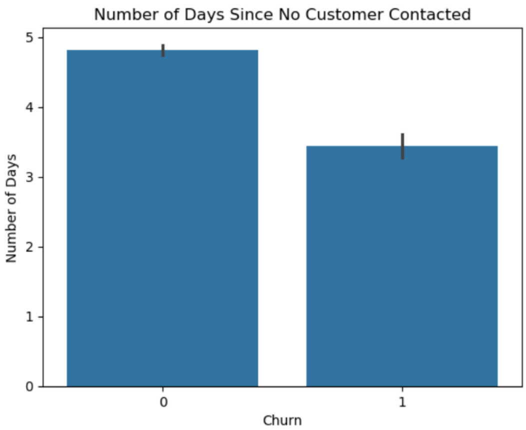
Here, we will look at the comparison of various features to each other – based on the concept of churn prediction. We will see how these observations help us draw business implications for our problem in the project.

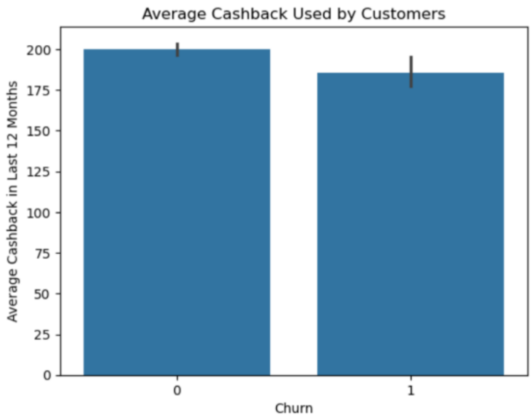












Multivariate analysis:

In this section we have created three clusters using K-prototype clustering to find patterns and relationships among features and variables to find how churn prediction is affected. The following table will give us an idea of how features interact with each other while deciding churn prediction.

Cluster 0	**Properties**	**Churn**
108 rows	Gender – Male Login device – Mobile (88) Marital status – majority married Payment mode – majority debit card & credit card Service score – 2.92 Tenure – 10.29 Cashback – 1903.79 Coupons used for payment – 1.72 Revenue growth year-on-year – 16 Revenue per month – 5.31	Yes – 22 No – 86 Percentage of churn = 20%
Cluster 1 2116	Gender – Male Login device – Mobile (88) Marital status – majority married	Yes – 190 No – 1926

	Payment mode – majority debit card & credit card Service score – 2.98 Tenure – 18.04 Cashback – 262.42 Coupons used for payment – 2.69 Revenue growth year-on-year – 16.26 Revenue per month – 7.11	Percentage of churn = 9%
Cluster 2 9036	Gender – Male Login device – Mobile (88) Marital status – majority married Payment mode – majority debit card & credit card Service score – 2.88 Tenure – 9.34 Cashback – 160.34 Coupons used for payment – 1.58 Revenue growth year-on-year – 16.17 Revenue per month – 6.17	Yes – 1684 No – 7352 Percentage of churn = 19%

Business Implications based on Exploratory Data Analysis:

- Customer care contact with customers ranges from 11 to 20 times.
- The year-on-year revenue growth of customers ranges from 12 to 20.
- Most customers have used coupons 1-2 times.
- Revenue per month and times customer care contacted are more in value for customers who churn.
- Complaints received in last 12 months and tenure for churning customers are significantly less than those who do not churn.

Problem 3 – Data Cleaning and Pre-processing

Solution:

We can see that many category variables have been assigned integer and decimal data types. Also, Many continuous variables have been assigned object data types. The former needs to be specified as categories and the latter need junk values in the columns treated.

- **Renaming:**

Some of the column names do not give a clear idea about what the column contains. Some Column names have been named using lowercases and uppercases haphazardly that might hinder the consistency of variables names while working.

Some changed variable names are:

1. ‘Payment’ has been changed to ‘Payment_mode’
2. ‘City_Tier’ has been changed to ‘City_tier’
3. ‘Service_Score’ has been changed to ‘Service_score’
4. ‘account_segment’ has been changed to ‘Account_segment’

- **Removal of unwanted variables (if applicable)** - along with the variables mentioned in multivariate analysis we have dropped “AccountID” since its contribution to customer churn is rudimentary.

- **Missing Value treatment (if applicable)** – the following imputations have been done:

1. Numerical variables have been imputed with their respective mean and median values
2. Gender, Account_user_count, Account_segment, coupon_used_for_payment, Login_device, City_tier, Payment_mode, Service_score, CC_Agent_Score, Marital_Status, Complain_ly have been imputed with their respective mode values since the most highest occuring value within these variables makes sense.

- **Outlier treatment (if required)** – no need since we want to retain the original data for accurate predictions.

- **Variable transformation (if applicable)** – most variables are unique and do not contribute to formation of relevant transformations.

- **Addition of new variables (if required)** – not required.

Problem 4 – Model Building

Solution:

Since the dataset given to us is imbalanced with lot of categorical variables, and we need to figure out the churn rate, we will be building the following models on both the original dataset and the balanced dataset. The balanced dataset is created using SMOTE (Synthetic Minority Oversampling Technique) and top of it, we have applied Bayesian Search for hyperparameter tuning of the models. The following six models have been built.

1. Decision Tree Classifier Model
2. Random Tree Classifier Model
3. AdaBoost Classifier Model
4. Naïve Bayes Classification Model
5. XGBoost Classifier Model
6. Support Vector Machine Model

Given below is a table of comparison for various metrics on the models created on the original dataset – both without and with Bayesian Search Hyperparameter tuning of models.

	Metrics on Original Dataset					Metrics on Bayesian Dataset			
Model Name	Precision	Recall	f1-score	Accuracy	Model Name	Precision	Recall	f1-score	Accuracy
Decision Tree					**Decision Tree**				
Class 0 - Train	1.00	1.00	1.00		Class 0 - Train	1.00	1.00	1.00	
Class 1 -Train	1.00	1.00	1.00	1.00	Class 1 -Train	1.00	0.98	0.99	1.00
Class 0 - Test	0.96	0.97	0.97		Class 0 - Test	0.96	0.97	0.97	
Class 1 - Test	0.84	0.82	0.83	0.94	Class 1 - Test	0.894	0.81	0.83	0.94
Random Tree Classifier					**Random Tree Classifier**				
Class 0 - Train	0.96	1.00	0.98		Class 0 - Train	1.00	1.00	1.00	
Class 1 -Train	0.98	0.79	0.87	0.96	Class 1 -Train	1.00	1.00	1.00	1.00
Class 0 - Test	0.94	0.99	0.96		Class 0 - Test	0.97	1	0.99	
Class 1 - Test	0.94	0.68	0.79	0.94	Class 1 - Test	0.99	0.86	0.92	0.98
AdaBoost Classifier					**AdaBoost Classifier**				
Class 0 - Train	0.86	0.98	0.92		Class 0 - Train	0.92	0.96	0.94	
Class 1 -Train	0.74	0.22	0.34	0.86	Class 1 -Train	0.77	0.6	0.68	0.9
Class 0 - Test	0.87	0.98	0.92		Class 0 - Test	0.92	0.96	0.94	
Class 1 - Test	0.72	0.25	0.37	0.86	Class 1 - Test	0.77	0.58	0.65	0.9
Naïve Bayes					**Naïve Bayes**				
Class 0 - Train	0.87	0.97	0.92		Class 0 - Train	0.87	0.97	0.92	
Class 1 -Train	0.66	0.26	0.38	0.85	Class 1 -Train	0.66	0.26	0.38	0.85
Class 0 - Test	0.87	0.97	0.92		Class 0 - Test	0.87	0.97	0.92	
Class 1 - Test	0.69	0.28	0.4	0.86	Class 1 - Test	0.69	0.28	0.4	0.86
XGBoost Model					**XGBoost Model**				
Class 0 - Train	1.00	1.00	1.00		Class 0 - Train	1.00	1.00	1.00	
Class 1 -Train	1.00	1.00	1.00	1.00	Class 1 -Train	1.00	1.00	1.00	1.00
Class 0 - Test	0.97	0.99	0.98		Class 0 - Test	0.98	1	0.99	
Class 1 - Test	0.96	0.86	0.91	0.97	Class 1 - Test	0.98	0.9	0.94	0.98
Support Vector Model					**Support Vector Model**				
Class 0 - Train	0.94	0.99	0.96		Class 0 - Train				
Class 1 -Train	0.95	0.66	0.78	0.94	Class 1 -Train				
Class 0 - Test	0.92	0.99	0.95		Class 0 - Test				
Class 1 - Test	0.92	0.59	0.72	0.92	Class 1 - Test				

The observations after building the models are as follows:

- Decision Tree Classifier

Overfitting is evident as training accuracy is 100%, while test accuracy drops to 94%.
Class 1 performance on the test set is relatively lower (F1-score: 0.83) compared to Class 0 (F1-score: 0.97), suggesting potential imbalance.

After Optimization:
No significant changes in accuracy and F1-score.
Minor improvement in Class 1 precision (0.894 vs. 0.84 before), but recall remains similar.
Inference: Bayesian optimization did not significantly impact the Decision Tree model. Overfitting remains a concern.

- Random Tree Classifier

Still exhibits some overfitting (Train: 96%, Test: 94%).

Class 1 recall is lower (0.68 on test set), which means it's missing many instances of Class 1.

After Optimization:
Train Accuracy: 100% (previously 96%)
Test Accuracy: 98% (previously 94%)
Class 1 Test F1-score improved to 0.92 (previously 0.79)
Inference: Bayesian optimization significantly improved test accuracy and Class 1 recognition, reducing bias towards Class 0.

- AdaBoost Classifier

Struggles with Class 1 predictions (Train F1-score: 0.34, Test F1-score: 0.37).
Overall test accuracy is 86%, meaning the model is not performing well on the minority class.

After Optimization:
Train Accuracy: 90% (previously 86%)
Test Accuracy: 90% (previously 86%)
Class 1 Test F1-score improved to 0.65 (previously 0.37)
Inference: Bayesian optimization notably improved Class 1 performance, making the model more balanced.

- Naïve Bayes

Similar behavior to AdaBoost, with Class 1 suffering (F1-score: 0.38 on Train, 0.40 on Test).
Accuracy is 85% on train and 86% on test, suggesting generalization but poor minority class handling.

After Optimization:
No significant improvements in any metric.
Inference: Bayesian optimization had no significant impact on Naïve Bayes, possibly due to the model's inherent simplicity.

- Support Vector Model

Good balance with Train: 94%, Test: 92%.
Class 1 recall is low (0.59 on test), indicating misclassification of the minority class.

Given below are the reasons for choosing the XGBoost Model as a valid model on the original dataset:

- Best performer overall, with minimal overfitting (Train: 100%, Test: 97%).
- Class 1 test F1-score is 0.91, significantly better than AdaBoost and Naïve Bayes.
- After Optimization:
- Train Accuracy: 100% (same as before)
- Test Accuracy: 98% (previously 97%)
- Class 1 Test F1-score improved to 0.94 (previously 0.91)

- Inference: Bayesian optimization improved generalization and Class 1 performance, making XGBoost the best-performing model.

Since, the dataset is not balanced in terms of the churn prediction classes, we have balanced the dataset using SMOTE, as discussed previously. Given below is the table for metrics on the balanced dataset without and with Bayesian Search hyperparameter tuning.

	Metrics on Balanced Dataset					Metrics on Bayesian Dataset			
Model Name	Precision	Recall	f1-score	Accuracy	Model Name	Precision	Recall	f1-score	Accuracy
Decision Tree Classifier					**Decision Tree Classifier**				
Class 0 - Train	1.00	1.00	1.00		Class 0 - Train	0.99	0.99	0.99	
Class 1 -Train	1.00	1.00	1.00	1.00	Class 1 -Train	0.99	0.99	0.99	0.99
Class 0 - Test	0.94	0.93	0.93		Class 0 - Test	0.93	0.93	0.93	
Class 1 - Test	0.93	0.94	0.93	0.93	Class 1 - Test	0.93	0.93	0.93	0.93
Random Tree Classifier					**Random Tree Classifier**				
Class 0 - Train	0.94	0.94	0.94		Class 0 - Train	1.00	1.00	1.00	
Class 1 -Train	0.94	0.94	0.94	0.94	Class 1 -Train	1.00	1.00	1.00	1.00
Class 0 - Test	0.92	0.92	0.92		Class 0 - Test	0.97	0.97	0.97	
Class 1 - Test	0.92	0.91	0.92	0.92	Class 1 - Test	0.97	0.97	0.97	0.97
AdaBoost Classifier		0.70	0.73		**AdaBoost Classifier**				
Class 0 - Train	0.76	0.70	0.73	0.74	Class 0 - Train	0.91	0.91	0.91	
Class 1 -Train	0.72	0.78	0.75		Class 1 -Train	0.91	0.91	0.91	0.91
Class 0 - Test	0.75	0.71	0.73		Class 0 - Test	0.9	0.91	0.91	
Class 1 - Test	0.72	0.77	0.75	0.74	Class 1 - Test	0.91	0.9	0.9	0.9
Naïve Bayes					**Naïve Bayes**				
Class 0 - Train	0.76	0.70	0.73		Class 0 - Train	0.77	0.69	0.72	
Class 1 -Train	0.72	0.77	0.75	0.74	Class 1 -Train	0.72	0.79	0.75	0.74
Class 0 - Test	0.75	0.69	0.72		Class 0 - Test	0.76	0.68	0.72	
Class 1 - Test	0.71	0.77	0.74	0.73	Class 1 - Test	0.71	0.79	0.75	0.73
XGBoost Model					**XGBoost Model**	1.00	1.00	1.00	
Class 0 - Train	1.00	1.00	1.00		Class 0 - Train	1.00	1.00	1.00	1.00
Class 1 -Train	1.00	1.00	1.00	1.00	Class 1 -Train				
Class 0 - Test	0.98	0.97	0.98		Class 0 - Test	0.98	0.98	0.98	
Class 1 - Test	0.97	0.98	0.98	0.98	Class 1 - Test	0.98	0.98	0.98	0.98
Support Vector Model					**Support Vector Model**				
Class 0 - Train	0.95	0.90	0.93		Class 0 - Train				
Class 1 -Train	0.91	0.95	0.93	0.93	Class 1 -Train				
Class 0 - Test	0.93	0.89	0.91		Class 0 - Test				
Class 1 - Test	0.90	0.94	0.92	0.91	Class 1 - Test				

Given below are the observations based on the above models:

- Decision Tree Classifier

Minimal change in test performance after Bayesian optimization.
Slight reduction in training accuracy (1.00 → 0.99), indicating reduced overfitting.
Business Takeaway: Overfitting is still present. Consider pruning or ensemble methods.

- Random Tree Classifier

Significant improvement after Bayesian optimization.

Test accuracy increased (0.92 → 0.97), reducing bias towards majority class.
Business Takeaway: A strong contender for deployment due to better balance across classes.

- AdaBoost Classifier

Major improvement in test performance after Bayesian optimization.
Test F1-score increased from 0.75 to 0.91, significantly reducing bias.
Business Takeaway: Improved generalization makes AdaBoost a viable model for classification tasks.

- Naïve Bayes

Minimal improvement post-Bayesian tuning.
Test performance saw slight improvement in recall (Class 1: 0.77 → 0.79).
Business Takeaway: Bayesian search does not significantly impact Naïve Bayes. Better suited for simpler datasets.

Reasons for finalizing the XGBoost Model after Bayesian Search on balanced dataset:

- Consistently the best performer both before and after Bayesian optimization.
- Test accuracy remains 98%, showing strong generalization.
- Business Takeaway: If computational resources allow, XGBoost is the best choice for deployment.
- We will therefore, choose XGBoost Classifier Model as our optimum since it can clearly demarcate the target classes.

Bayesian search on support vector model consumes a lot of time and resources. Hence, optimizing the SVM model has been avoided. The final model that will be used is the XGBoost model on the balanced dataset.

Problem 5 – Model Validation

Solution:

The model has been validated on the balanced dataset based on the following three metrics. The reasons as to why they have been focused upon are also given.

1. **Accuracy** as a fundamental metric has been used to evaluate the performance of classification models, as it measures the proportion of correctly predicted instances (both true positives and true negatives) among all instances in the dataset. Our objective is to predict the churn rate i.e., the probability of positive cases.

2. **Precision** is a critical metric used to assess the quality of positive predictions made by a classification model since it quantifies the proportion of true positive predictions

(correctly predicted positive instances) among all instances predicted as positive, whether they are true positives or false positives.

3. **Recall**, also known as sensitivity or true positive rate, assesses a model's ability to correctly identify all positive instances within a dataset. It quantifies the proportion of true positive predictions (correctly predicted positive instances) among all instances that are actually positive.

4. The **F1-Score** combines both precision and recall into a single value. It provides a balanced assessment of a model's performance, especially when there is an imbalance between the classes being predicted. Our model definitely has imbalance between classes being predicted.

As shown in the tables that compare the metrics of all the models, we can see that without Byaesian Search on the original dataset there is a lot of overfitting and underfitting issues in most of the above metrics. Bayesian Search increases the performance of the models, and the best model that stands out is the XGBoost model. The metrics are given as follows:

XGBoost Model				
Class 0 - Train	1.00	1.00	1.00	
Class 1 -Train	1.00	1.00	1.00	1.00
Class 0 - Test	0.98	1	0.99	
Class 1 - Test	0.98	0.9	0.94	0.98

After using SMOTE to balance the dataset, we ran the model building process again. This time, although some models performed good, there were overfitting and underfitting issues in some models. The hyperparameter tuning with Bayesian Search increased the performance of these models to a great extent, and out of all the best performing model turned out to be the XGBoost model. The metrics for the final model are as follows:

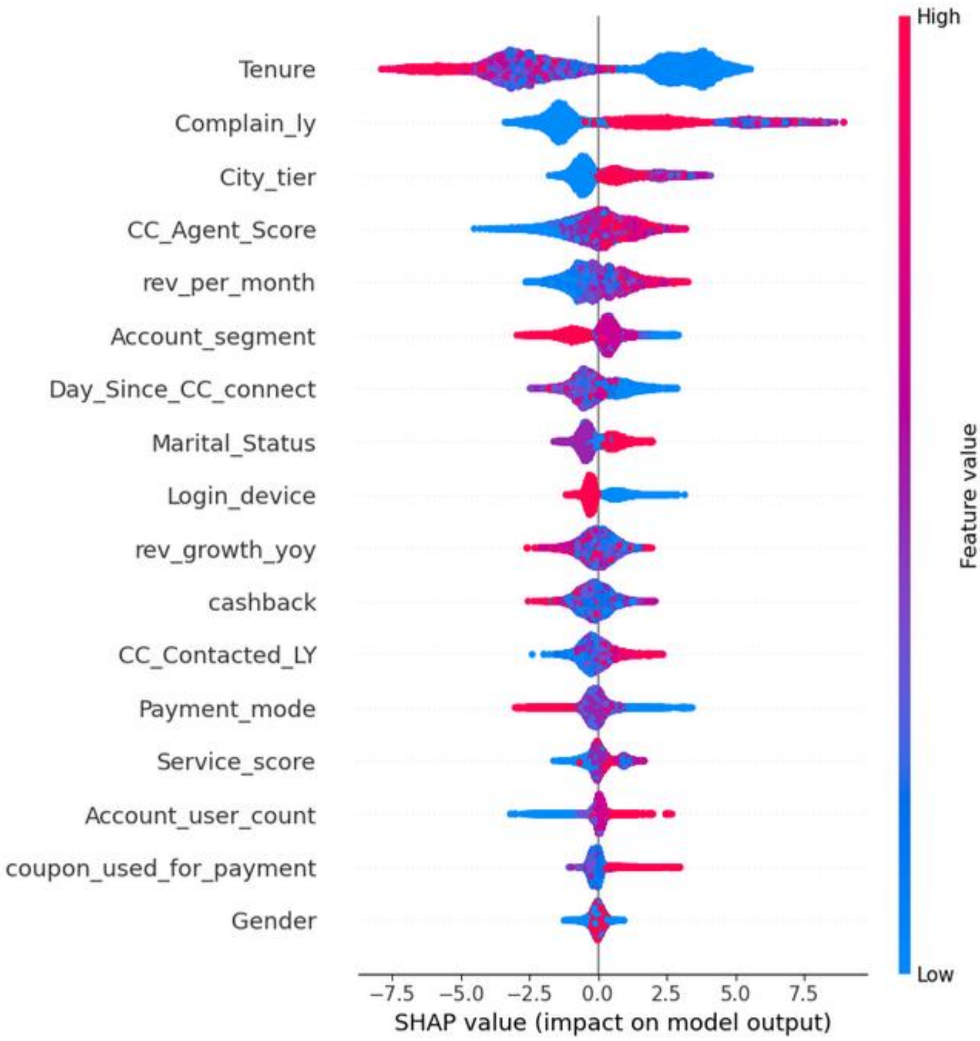
XGBoost Model	1.00	1.00	1.00	
Class 0 - Train	1.00	1.00	1.00	1.00
Class 1 -Train				
Class 0 - Test	0.98	0.98	0.98	
Class 1 - Test	0.98	0.98	0.98	0.98

Problem 6 – Final Interpretation/Recommendations

Solution:

To find out how the variables and features are influencing the model, we use SHAP (SHapley Additive exPlanations) which is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

The figure below illustrates the contribution of variables and features to the accurate prediction of churn probabilities.



Inferences based on the SHAP values for variables:

- New customers are more likely to churn.

- Complaints increase churn risk.
- Low tier (tier 2 and 3) cities have customers more likely to churn.
- Poor agent performance surprisingly is related to customers not churning.
- High-paying customers are at risk of churning.
- Regular, HNI and some Regular Plus account holders are at risk of churning.
- Less contacted customers by Customer Care are more likely to churn.
- People who use computers as a login device are more likely to churn than those who use mobile phones (apps).
- People using COD and Credit as payment modes are observed to churn easily.
- More number of people using the account leads to churning.
- Customers who use more coupons churn easily.

Recommendations based on interpretations and observations:

- Improve onboarding experience & engagement for new customers.
- Address customer concerns quickly and improve support services.
- Increase support and educative programs for customers in tier 2 and tier 3 cities.
- Inspect if high ratings received by customers are really deserving or not – because wherever the customer gives honest review the churn rate is less.
- Some people seem to leave after buying certain expensive products – high revenue individuals who churn might be customers who might have joined to take advantage of joining bonuses or may have found difficult to navigate through the system.
- Segmented offers should be offered according to the financial and preferences of Regular, Regular Plus and HNI customers.
- The user interface of computer users should be made more comfortable for users.
- Either the payment experience for COD and Credit Card customers should be made easier or offers should be given on these payment modes more.