Machine Learning Internship Assessment

Customer Churn Prediction

By Yuvraj Mor

Table of Contents

1. Introduction

Objective

The primary objective of this project is to develop a predictive model for customer churn.

Customer churn, also known as customer attrition, is a critical concern for businesses across various industries. It refers to the phenomenon where customers cease doing business with a company, and it can have a significant impact on a company's revenue and profitability.

The aim of this project is to leverage machine learning techniques to proactively identify customers at risk of churning, enabling the business to take targeted actions to retain them.

Data Description

The dataset consists of customer information for a customer churn prediction problem. It includes the following columns:

CustomerID: Unique identifier for each customer.
Name: Name of the customer.
Age: Age of the customer.
Gender: Gender of the customer (Male or Female).
Location: Location where the customer is based, with options including Houston, Los Angeles, Miami, Chicago, and New York.
Subscription_Length_Months: The number of months the customer has been subscribed.
Monthly_Bill: Monthly bill amount for the customer.
Total_Usage_GB: Total usage in gigabytes.
Churn: A binary indicator (1 or 0) representing whether the customer has churned (1) or not (0).

Approach

I followed a typical machine learning project pipeline, from data preprocessing to model deployment.

2. Data Preprocessing

Exploratory Data Analysis (EDA)

The first step involves loading the dataset and exploring it to understand its structure and characteristics:

- The dataset contains information about 100,000 customers with 9 variables.
- Out of the 9 variables, 6 are numerical and 3 are categorical.
- All variables are uniformly distributed which we can confirm by plotting their distribution using histograms.
- The correlation heatmap shows that there is almost no relationship between any pairs of variables.
- The unnecessary columns 'Customer_ID' and 'Name' are dropped as they won't have any effect in the analysis.

Handling missing values and outliers

- There are no null or duplicate values in the dataset.
- There are no outliers in the dataset which we confirmed by using the IQR method.

Label Encoding

- The categorical columns 'Gender' and 'Age' are converted into numerical variables using Label Encoding.

The pre-processed data looks like this:

	Age	Gender	Location	Subscription_Length_Months	Monthly_Bill	Total_Usage_GB	Churn
0	63	1	2	17	73.36	236	0
1	62	0	4	1	48.76	172	0
2	24	0	2	5	85.47	460	0
3	36	0	3	3	97.94	297	1
4	46	0	3	19	58.14	266	0
...	...	...	...	...	...	...	...
99995	33	1	1	23	55.13	226	1
99996	62	0	4	19	61.65	351	0
99997	64	1	0	17	96.11	251	1
99998	51	0	4	20	49.25	434	1
99999	27	0	2	19	76.57	173	1

100000 rows × 7 columns

3. Feature Engineering

Generating relevant features

I generated the following features from the dataset that can help improve the model's prediction accuracy.

1. **Age Group:** This feature categorizes customers into age groups such as "Young," "Middle-aged," and "Senior" based on predefined age bins.

2. **Tenure in Years:** It calculates the customer's tenure in years by dividing the "Subscription_Length_Months" by 12, providing a more interpretable measure of the customer's relationship with the service.

3. **Billing to Usage Ratio:** This feature represents the ratio of the customer's monthly bill to their total usage, helping to identify customers who may not be using the service efficiently.

4. **Usage per Billing Cycle:** It calculates the average usage per billing cycle by dividing "Total_Usage_GB" by "Subscription_Length_Months," providing insight into the customer's usage behavior.

5. **Churn History Count:** This feature tracks the cumulative count of churn events for each customer based on their age, revealing how many times a customer has churned in the past.

These features aim to capture various aspects of customer behavior, tenure, and churn history, which can be valuable for building a predictive model for customer churn. They provide different dimensions of information that may be relevant in understanding and predicting churn patterns.

Feature Scaling

Feature scaling is applied to ensure all variables were on the same scale using MinMaxScaler.

Data Splitting

The dataset is divided into training and testing sets in the ratio 80:20 to enable model training and evaluation.

Feature Selection

Identifying important features helps streamline the model and improve its interpretability.
I used Random Forest Feature Importance rank features based on their contribution to the target variable:

	Feature	Importance
0	Churn_History_Count	0.152454
1	Monthly_Bill	0.149973
2	Billing_to_Usage_Ratio	0.147522
3	Usage_Per_Billing_Cycle	0.137945
4	Total_Usage_GB	0.129485
5	Age	0.107590
6	Tenure_Years	0.050268
7	Subscription_Length_Months	0.050161
8	Location	0.042758
9	Gender	0.019041
10	Age_Group	0.012802

I calculate the optimal number of features to be used i.e., 8 by plotting a graph for cumulative importance.

Multicollinearity using Variance Inflation Factor

The training data contains multicollinearity because Tenure_Years and Subscription_Length_Months the have a VIF value way more than 5. So, out of the 8 features that I was going to use, I will drop Subscription_Length_Months and use Location which is the next most important feature.

4. Model Building

- Several machine learning algorithms were trained and evaluated using the dataset.
- Algorithms include Logistic Regression, Decision Tree, K-Nearest Neighbours, Gaussian Naive Bayes, AdaBoost, Gradient Boosting, Random Forest, XGBoost, and Support Vector Classifier (SVC).
- Training and test data performance metrics were calculated, revealing the strengths and weaknesses of each algorithm.

Training data result:

	Algorithm	Accuracy	Precision	Recall	F1-score	Building Time (s)
0	LogisticRegression	0.503238	0.503039	0.503238	0.500175	1.934268
1	DecisionTreeClassifier	1.000000	1.000000	1.000000	1.000000	1.874611
2	KNeighborsClassifier	0.687375	0.687376	0.687375	0.687367	6.280308
3	GaussianNB	0.504487	0.504456	0.504487	0.485971	0.283656
4	AdaBoostClassifier	0.518725	0.518750	0.518725	0.517327	4.954256
5	GradientBoostingClassifier	0.537075	0.537565	0.537075	0.534740	18.377644
6	RandomForestClassifier	1.000000	1.000000	1.000000	1.000000	28.024447
7	XGBClassifier	0.651563	0.651907	0.651563	0.651296	7.335256
8	SVC	0.503925	0.505431	0.503925	0.431035	754.212243

- Decision Tree and Random Forest achieved perfect accuracy on the training data, suggesting potential overfitting.
- Logistic Regression, GaussianNB, Gradient Boosting and AdaBoost have relatively low accuracy and F1-score.
- KNeighborsClassifier and XGBClassifier have moderate accuracy and F1-score.
- SVC has low accuracy but high building time.

Testing data result:

	Algorithm	Accuracy	Precision	Recall	F1-score
0	LogisticRegression	0.50390	0.503375	0.50390	0.500633
1	DecisionTreeClassifier	0.49645	0.496449	0.49645	0.496449
2	KNeighborsClassifier	0.49400	0.493994	0.49400	0.493996
3	GaussianNB	0.50600	0.505274	0.50600	0.486792
4	AdaBoostClassifier	0.50210	0.501707	0.50210	0.500532
5	GradientBoostingClassifier	0.50455	0.504109	0.50455	0.502186
6	RandomForestClassifier	0.50285	0.502697	0.50285	0.502571
7	XGBClassifier	0.49920	0.499004	0.49920	0.498805
8	SVC	0.50370	0.501817	0.50370	0.431097

- Most algorithms have mediocre performance on the test data, with accuracy close to random guessing (around 0.5).
- No algorithm is performing well.

Multiple neural network architectures were tried but none of them provided good results. Same case with the ensembles of Random Forest, which is why I decided to further optimize the XGBClassifier as it is the best-performing algorithm on both the training and test data. It has the highest accuracy, precision, recall, and F1-score on all three.

5.Model Optimization

Hyperparameter Tuning

I performed hyperparameter tuning using recall as a metric to focus on as false negatives are more important to reduce.

After hyperparameter tuning, the performance of the model improved slightly. Hence, I will use the best parameters in the XGB classifier model

Cross-Validation

Cross-validation is performed to validate the model's performance and ensure it generalized well to new data.

1. **Cross-Validation Scores (Accuracy)**: [0.49875 0.4993125 0.49925 0.5010625 0.504]

   Mean Accuracy Score: 0.500475

2. **Cross-Validation Scores (Recall)**: [0.76053688 0.54703964 0.26718515 0.29557145 0.49705181]

   Mean Recall Score: 0.47347698502908464

Finding Optimal Threshold

The threshold for classification is fine-tuned to strike a balance between accuracy, sensitivity, specificity, and F1-score. The value is 0.5.

Final Model Evaluation

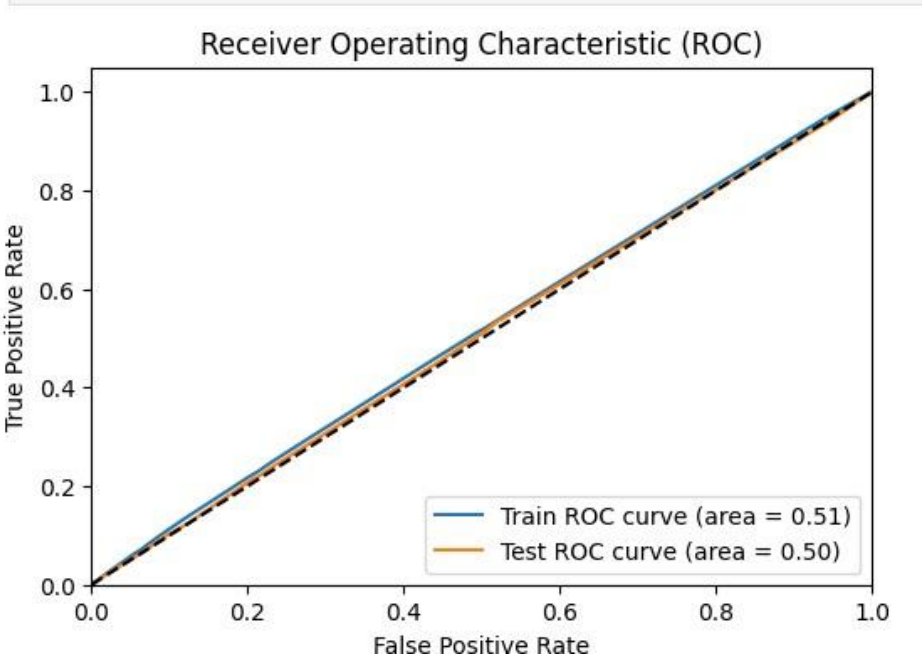
The final XGBoost model's performance is evaluated using various metrics on both the trainin g and test datasets.

	Dataset	Accuracy	Precision	Recall	F1-score
0	Train	0.651563	0.651931	0.651466	0.651259
1	Test	0.504850	0.504907	0.504906	0.504845

A confusion matrix is also plotted:

	Training Set	Test Set
True Positive (%)	24.78625	25.085
True Negative (%)	25.39125	25.310
False Positive (%)	23.78375	24.205
False Negative (%)	26.03875	25.400

ROC-AUC curve:

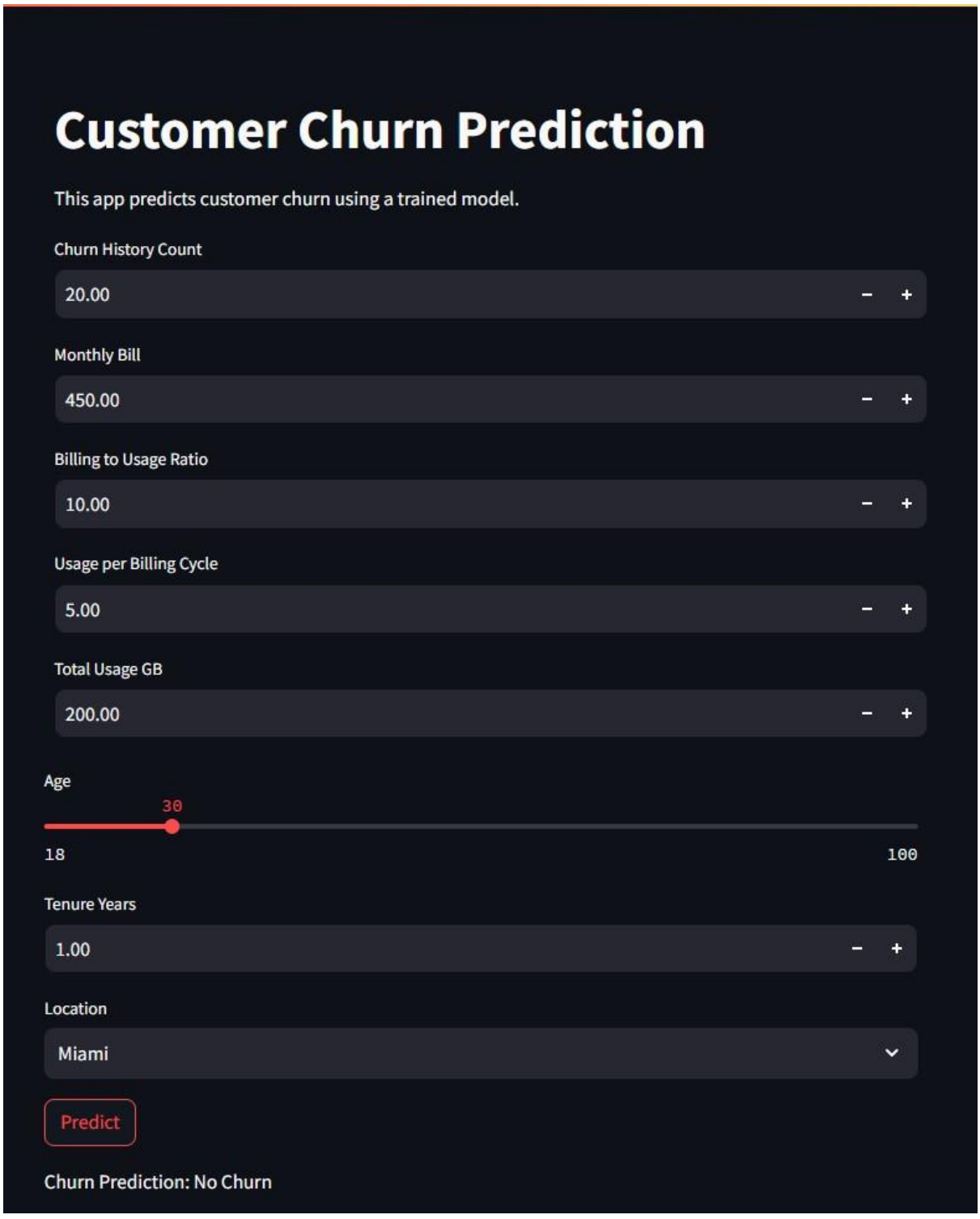


The model’s can be further improved but for the scope of this project I will stop here and save the model as a pickle file for model deployment on Streamlit.

6. Model Deployment

Here’s the link to the deployed model on Streamlit: Customer Churn Predictor

Here’s an example of inputting new customer data and getting the churn prediction for it from the app:



7. Conclusion

Outcome

Customer churn prediction is a project that uses data to predict which customers are likely to stop using a company's products or services.

This project involved thorough exploratory data analysis, pre-processing, and the evaluation of various machine learning algorithms. The XGBoost Classifier was selected as the final model because it performed better than other models on metrics such as accuracy and recall.

While it is challenging to achieve perfect accuracy and recall, the insights gained from this project can be used to guide the company's strategies for customer retention and business growth.

Future Work

In the future, I would like to improve the customer churn prediction model in the following ways:

- **Gather more data**. I would like to gather more data from a variety of sources, such as customer surveys, social media, and website analytics. This would allow me to build a more comprehensive model that can better capture the factors that contribute to customer churn.
- **Explore advanced techniques.** I would like to explore advanced machine learning techniques to see if they can improve the performance of the model.
- **Segment my customers**. I would like to segment my customers based on their characteristics and behavior. This would allow me to build more targeted churn prediction models for each segment.
- **Develop real-time churn prediction**. I would like to develop a real-time churn prediction system that can identify customers who are at risk of churning as soon as they exhibit certain behaviors. This would allow the company to intervene early and try to retain these customers.
- **Explore the reasons for churn**. Once I have identified customers who are at risk of churning, I would like to try to understand why they are churning. This information can be used to develop more effective churn prevention strategies.

I believe that these future work items will help me to develop a more robust and accurate customer churn prediction model. This model can then be used to improve customer retention and business growth.