Name: - Shivam Krishnaraj Mishra

After importing all necessary libraries and provided dataset, I have followed the given machine learning pipeline

1. **Data Preprocessing**

+ Initial Exploration
  - → There are 9 features:
    - **CustomerID** – integer
    - **Name** – object
    - **Age** – Integer
    - **Gender** – object
    - **Location** – object
    - **Subscription_Length_Months** – Integer, subscription length in months
    - **Monthly_Bill** – Float
    - **Total_Usage_GB** – Integer, total usage of data in GB
    - **Churn** – Integer, 1 means it will churn else will not churn
  - → Shape of dataset is 100000 X 9
  - → Unique values of each feature:
    - **Gender** – Male and Female
    - **Location** - Los Angeles, New York, Miami, Chicago, Houston
    - **Churn** – 1 and 0
+ Handling Missing values and outliers
  - → I see have none of the features contains “NaN” values (using “Missingno”) which need to be taken care of.
+ Preparing data for ML Models

→ Converting string object data to integer data, that is, encoding

- Gender – Males are map to 0 and Females are map to 1
- Location -
  - Los Angeles: 0,
  - New York: 1,
  - Miami: 2,
  - Chicago: 3,
  - Houston: 4

→ The above conversion is done using label encoding

+ Splitting dataset

→ Splitting of data is done in ¾ ratio such that is, 75% training and 25% for testing.

→ Splitting is done randomly.

→ All non-integer/non-float datatype features are converted into integer.

2. Data Virtualization

+ Distribution curves for three continuous data features are plotted.
+ Pie Chart build on gender, location and age, shows that in our dataset

→ Population of male and female candidates are nearly same.

→ People for all the given 5 regions present in our dataset are in same quantity.

→ The population of people who will and will not churn in future are also nearly the same.

→ So, our dataset is biased to anyone of the category.

+ Missingno matrix is drawn, showing there is no null value present in dataset.

3. Feature Engineering

+ Generating relevant features
    - → Pearson Correlation matrix is made and displayed in form of heatmap, which shows that "CustomerID" and "Name" can easily be ignored due to very less impact and correlation with target feature.
    - → Dropping these features and renaming columns of our data frame.
+ Feature Scaling
    - → Normalizing dataset from 0 to 1, using **MinMaxScaler**.

4. Evaluation Metric Used

+ Confusion Matrix
+ Accuracy
+ Precision
+ Recall
+ F1-Score

5. Model Building

+ Since we must predict whether customers will churn in future or not, we need to use classification machine learning models.
+ I have trained and tested our dataset on every classification algorithm and later infer which one gives best performance and is computationally less expensive.
+ Classification Machine Learning Algorithms Performance
    - → K-Nearest Neighbor
        - Accuracy: 50.012%
        - Took less time to train and test
        - Average F1-Score: 49%
        - Average Precision: 50%
    - → Random Forest

- Accuracy: 49.59%
- Took less time to train and test
- Average F1-Score: 48%
- Average Precision: 50%
- Average Recall: 50%

→ Logistic Regression

- Accuracy: 50.40%
- Took less time to train and test
- Average F1-Score: 46%
- Average Precision: 51%
- Average Recall: 51%

→ Decision Tree

- Accuracy: 49.46%
- Took less time to train and test
- Average F1-Score: 43%
- Average Precision: 49%
- Average Recall: 50%

→ Gaussian Naïve Bayes

- Accuracy: 50.32%
- Took less time to train and test
- Average F1-Score: 47%
- Average Precision: 51%
- Average Recall: 50%

→ Support Vector Classification

- Accuracy: 49.772%
- Took less time to train and test
- Average F1-Score: 49%
- Average Precision: 50%
- Average Recall: 50%

6. Hyperparameter involved

- Decision Tree
    i. Depth of tree
- Random Forest
    i. No.of tree
- KNN
    i. Top ‘k’ neighbors
- K-Fold
    i. ‘k’ samples of datasets
- SVC
    i. Kernel to be chosen

7. Model Optimization

+ For optimization technique, I used Cross Validation, that is, training and validating our model different samples of datasets on different iterations.
+ K-Fold is used, with k value optimally 5.
+ While sampling of dataset is done randomly.
+ For instance, based models, such as KNN Optimization Result
    → KNN
    - Mean Accuracy: 50.13%
    - The optimal value of ‘k’ is 6, I found this result by iteratively training and testing on different values of ‘k’.

    → Random Forest
    - Mean Accuracy: 49.952%
    - No.of trees Is found by iterative checking on different values and at last value 6 gave best result among other values.

    → Logistic Regression
    - Mean Accuracy: 50.09%

    → Decision Tree

        - Mean Accuracy: 50.21%
        - Pre-purning is done to avoid overfitting by taking maximum depth of tree to be 5.
    - → <u>Gaussian Naïve Bayes</u>
        - Mean Accuracy: 50.2%
    - → <u>Support Vector Classifier</u>
        - Mean Accuracy: 49%
        - Linear kernel is chosen.

8. Model Selection

- Decision Tree is selected because
    - Robust to missing and outliers.
    - So, it can handle any unknown value encountered during testing.
    - Flexible
    - Handle complex relationship between features
    - Nonparametric
- Naïve can also be considered as another choice because there is very little correlation between features.
- But as we know features are not normally distributed (as I have shown in visualization part) so decision tree is good choice and while testing there may be a case customer belong to any unknown location.