PROJECT REPORT ON

“**Churn prediction on telecom data**”

**Submitted By:**

PRODDATUR MAHAMMAD SOHAIL

**MAY,2024**



INFOSYS SPRINGBOARD 4.0

VIRTUAL INTERNSHIP

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE-NO** |
| **1** | **Overview** | **2-5** |
| **2** | **INTRODUCTION** | **5-6** |
| **3** | **MILESTONE-1** | **7-10** |
| **4** | **MILESTONE-2** | **10-20** |
| **5** | **MILESTONE-3** | **20-23** |
| **6** | **MILESTONE-4** | **23-26** |
| **7** | **conclusion** | **26** |

**Overview**

**Churn Prediction**

**Importance of Churn Prediction**

Churn prediction helps businesses identify customers likely to stop using their products or services, allowing them to take proactive measures to retain these customers. This is particularly crucial in subscription-based industries like telecommunications.

**Why Churn Prediction Matters**

Saving Money:Retaining current customers is cheaper than acquiring new ones.

Understanding Customers: Analyzing churn helps improve services.

Staying Competitive: Reducing churn helps maintain a competitive edge.

**Milestone 1: Data Collection and Understanding**

Data Collection

A comprehensive dataset was collected, containing customer behavior, transactions, demographics, interactions, and feedback.

Data Understanding

Data Overview: Reviewed dataset structure and data types.

Descriptive Statistics: Calculated basic statistics for numerical features.

Data Visualization: Used box plots and other visualizations to identify patterns and outliers.

Correlation Analysis: Identified relationships between features.

Missing Values and Data Quality: Addressed missing values and ensured data reliability.

Feature Importance: Assessed the significance of various features in predicting churn.

Tools

Anaconda Software: Installed for comprehensive data analysis and machine learning.

Jupyter Notebook: Used for creating and sharing documents with live code, equations, and visualizations.

**Milestone 2: Data Pre-processing**

Key Steps

Data Type Conversion: Corrected misclassified data types.

Duplicate Records Removal: Ensured data integrity by removing duplicates.

Unique Value Variables Removal: Checked for and addressed variables with unique values.

Zero Variance Variables Removal: Ensured variables contribute to predictive power.

Outlier Treatment: Identified and handled outliers using various techniques.

Missing Value Treatment: Addressed missing values through imputation or deletion.

Highly Correlated Variables Removal: Mitigated multicollinearity by removing correlated variables.

Multicollinearity (VIF > 5): Addressed multicollinearity by calculating and addressing high VIF values.

**Milestone 3: Model Building**

Decision Tree Classifier

Data Collection: Gathered customer data from various sources.

Data Splitting: Split data into training and testing sets (75-25 split).

Feature Scaling: Normalized numerical features.

Model Training and Evaluation: Trained and evaluated the Decision Tree model.

Confusion Matrix: Analyzed model performance.

Accuracy Calculation: Calculated model accuracy.

Visualization: Visualized the trained Decision Tree.

Hyperparameter Tuning: Improved performance through Grid Search and cross-validation.

**Milestone 4: Additional Models and Documentation**

Random Forest

Data Collection and Preprocessing: Similar steps as Decision Tree.

Model Training and Evaluation: Trained and evaluated the Random Forest model.

Confusion Matrix: Analyzed model performance.

Accuracy Calculation: Calculated model accuracy.

Documentation

Comprehensive Documentation: Summarized objectives, data collection, preprocessing, model training, and evaluation.

Clarity and Reproducibility: Ensured the process is transparent and can be reproduced.

**Results**

Decision Tree Classifier

Train Accuracy: 77.01%

Test Accuracy: 76.66%

Random Forest Classifier

Train Accuracy: 98.26%

Test Accuracy: 73.78%

**INTRODUCTION:**

Churn prediction**:** Churn prediction is a process used by businesses to identify customers who are likely to stop using their products or services within a certain period of time. It is a key component of customer relationship management and is particularly important in subscription-based businesses, such as telecommunications, SaaS (Software as a Service), and financial services**.**

In the highly competitive telecom industry, predicting and managing customer churn is crucial for maintaining profitability and market share. Churn, in this context, refers to the percentage of subscribers who switch from one service or provider to another within a given time period. Accurate churn prediction allows telecom companies to identify customers at risk of leaving and take proactive measures to retain them.

The main goal here is to predict churn for a major telecom company. This involves analyzing user data, including call and data usage patterns as well as demographic information. By leveraging state-of-the-art machine learning techniques, telecom companies can create predictive models that identify customers likely to churn.

This introduction focuses on the application of churn prediction for two major Indian telecom giants: Airtel and Jio. Both companies have large subscriber bases and operate in a fiercely competitive market, making effective churn prediction and management essential for their success. By using advanced data analytics and machine learning, Airtel and Jio can enhance customer satisfaction and loyalty, ensuring sustained growth and profitability.

**Why Churn Prediction Matters**

Churn prediction is important for several reasons:

1.Saving Money: It's cheaper to keep current customers than to find new ones. By predicting who might leave, companies can take action to keep them.

2.Understanding Customers: Analyzing churn helps companies know what their customers want and need, allowing them to improve their services.

3.Staying Competitive: In a crowded market, reducing churn helps companies stay ahead of their competitors.

**MILESTONE 1:**

**DATA COLLECTION**

<https://drive.google.com/file/d/1PZOGzky097lsrmsbE_Ywygr2Up9G8d9K/view?usp=sharing>

For churn prediction, we collected comprehensive data on customer behavior, transactions, demographics, interactions, and feedback. This diverse dataset provides the necessary insights and patterns to accurately predict which customers are at risk of leaving.

**DATA UNDERSTANDING**

Data understanding in churn prediction is a crucial step that involves thoroughly examining and interpreting the dataset to uncover insights that influence customer churn. When working with an Dataset containing churn prediction data, this process typically includes the following steps:

Data Overview: Reviewing the structure of the dataset, including the number of rows (customers) and columns (features), and understanding the type of data each column contains (e.g., numerical, categorical, date/time).

Descriptive Statistics: Calculating basic statistics such as mean, median, standard deviation, and distribution for numerical features. This helps in understanding the central tendencies and variability of the data.

Data Visualization: Creating visualizations such as box plots to identify patterns, trends, and outliers in the data. Visualization aids in comprehending the relationships between different features and their impact on churn.

Correlation Analysis: Analyzing correlations between features to identify which factors are closely related to churn. This can involve calculating correlation coefficients for numerical features.

Missing Values and Data Quality: Identifying and addressing missing values, inconsistencies, or errors in the data. This ensures the dataset is clean and reliable for further analysis and model building.

Feature Importance: Assessing the importance of various features in predicting churn. This can involve preliminary modeling or statistical tests to identify which features have the most significant impact on whether a customer will churn.

By thoroughly understanding the data, businesses can identify key factors driving customer churn and develop targeted strategies to improve customer retention. This foundational step ensures that subsequent predictive modeling efforts are based on well-informed insights.

**DATA ANALYSIS**

Data Overview and Structure:

df.info(): This command provided a concise summary of the DataFrame, including the number of non-null entries, data types of each column, and memory usage. This allowed us to quickly assess the overall structure and integrity of the data.

df.columns: We listed all column names to identify the features present in the dataset. Understanding the available features is crucial for planning the subsequent analysis steps.

df.dtypes: We checked the data types of each column to understand the nature of the data (e.g., integers, floats, objects). This also helped us identify any necessary type conversions.

Initial Data Inspection:

df.head(): This command displayed the first few rows of the DataFrame, providing a quick glance at the data. This helped us verify the content and structure of the dataset.

df.tail(): Similar to df.head(), this command displayed the last few rows of the DataFrame, ensuring that the data at the end of the dataset was also consistent and correctly formatted.

Missing Values Analysis:

df.isna().sum(): We analyzed the presence of missing values in the dataset by checking each column for null values and counting their occurrences. Identifying missing data is crucial for deciding how to handle these gaps in subsequent data cleaning processes.

**TOOL INSTALLATION AND SETUP**

In the first milestone of our Churn Prediction project, we successfully installed the necessary tools required for our analysis:

Anaconda Software: We downloaded and installed Anaconda, a comprehensive data science platform that includes a wide range of pre-installed packages and tools essential for data analysis and machine learning.

Jupyter Notebook: Within Anaconda, we utilized Jupyter Notebook, an interactive web application that allows us to create and share documents containing live code, equations, visualizations, and narrative text. Jupyter Notebook was chosen for its versatility and ease of use, making it an ideal environment for conducting and documenting our data analysis process.

**MILESTONE 2:**

Data pre-processing

1. Converting data types of variables which are misclassified.

In our data preprocessing phase, we identified and corrected several misclassified data types to ensure accurate and efficient analysis. Misclassified data types can lead to incorrect results and inefficient memory usage. Here are the key conversions we performed:

Numeric Data Stored as Strings: We converted numeric columns stored as strings to their appropriate numeric types (integers or floats) using pd.to\_numeric().

Categorical Data Stored as Strings: Categorical columns stored as strings were converted to the category dtype to save memory and enhance performance.

Boolean Data Stored as Integers or Strings: Boolean columns were converted to the boolean type using astype ('bool') to ensure correct logical operations.

RESULT: There are no variables with misclassified data types.

2. Removing Duplicate records

In our data preprocessing phase, we checked for the presence of duplicate records to ensure data integrity and accuracy. Duplicate records can distort analysis results and lead to incorrect conclusions. Here’s a brief overview of the steps we took:

Identification: We identified duplicate records using the df.duplicated() function, which flagged all duplicate rows.

Removal: We removed duplicate records using the df.drop\_duplicates() method, ensuring only the first occurrence of each duplicate was retained.

RESULT: There are no duplicate records in the dataset.

3.Removing Unique value variables

In our data preprocessing phase, we checked for variables with unique values to ensure the dataset's relevance and efficiency. Variables with unique values are often uninformative for analysis as they do not provide variability or insights into data patterns. Here’s a brief overview of our findings and steps:

Identification: We used the nunique() function to count the number of unique values in each column.

Findings: Upon analysis, we found that our dataset did not contain any variables with unique values. Therefore, no removal of such variables was necessary.

RESULT: There are no variables with unique values in the dataset.

4.Removing Zero variance variables

In our data preprocessing phase, we checked for variables with zero variance to ensure the dataset's relevance and efficiency. Zero variance variables have the same value for all observations and do not contribute to the predictive power of the model. Here’s a brief overview of our findings and steps:

Identification: We assessed the variance of each column to identify zero variance variables.

Findings: Upon analysis, we found that our dataset did not contain any zero variance variables. Therefore, no removal of such variables was necessary.

RESULT: There are no Zero variance variables in the dataset.

5.Outlier Treatment

Outlier treatment involves identifying and handling data points that significantly deviate from the rest of the dataset. These anomalies can skew analysis results and affect model performance. Common techniques for outlier treatment include:

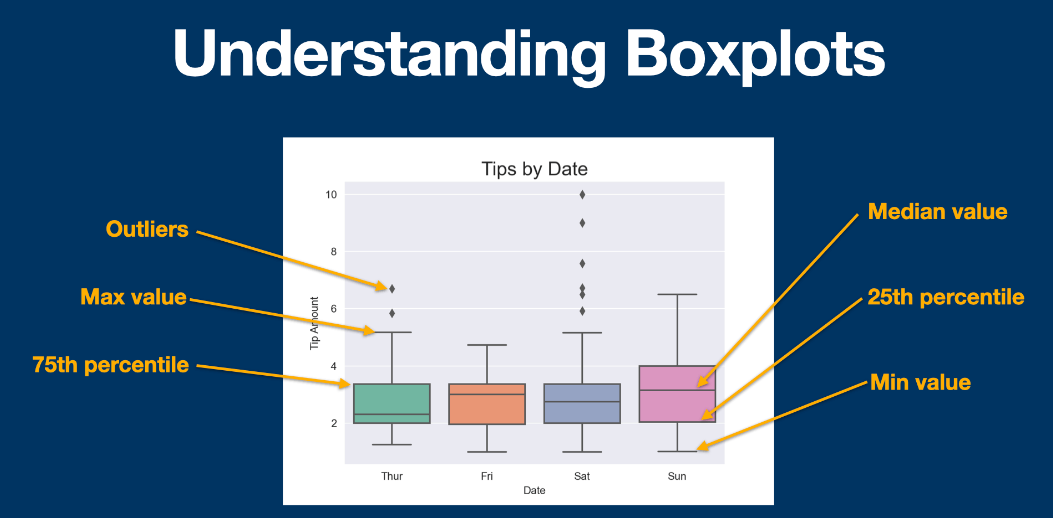
Identification: Detect outliers using statistical methods such as z-scores, the IQR method, or visualizations like box plots.

Handling: Address outliers by removing them, transforming them, or using robust statistical methods to minimize their impact on the analysis.

Effective outlier treatment ensures the accuracy of data analysis and modeling processes.

* Using Boxplot: Q3+(1.5\*IQR) & Q1-(1.5\*IQR)

Boxplots are effective visualization tools for identifying outliers in a dataset. Outliers are data points that lie significantly beyond the range of the rest of the data. One common method for outlier treatment using boxplots involves defining a threshold based on the interquartile range (IQR), which is the range between the first quartile (Q1) and the third quartile (Q3).



Here's a brief overview of the outlier treatment process using boxplots and the threshold Q3 + (1.5 \* IQR) and Q1 - (1.5 \* IQR):

Boxplot Visualization: Plot the data using a boxplot, which displays the distribution of the data, including the median, quartiles, and potential outliers.

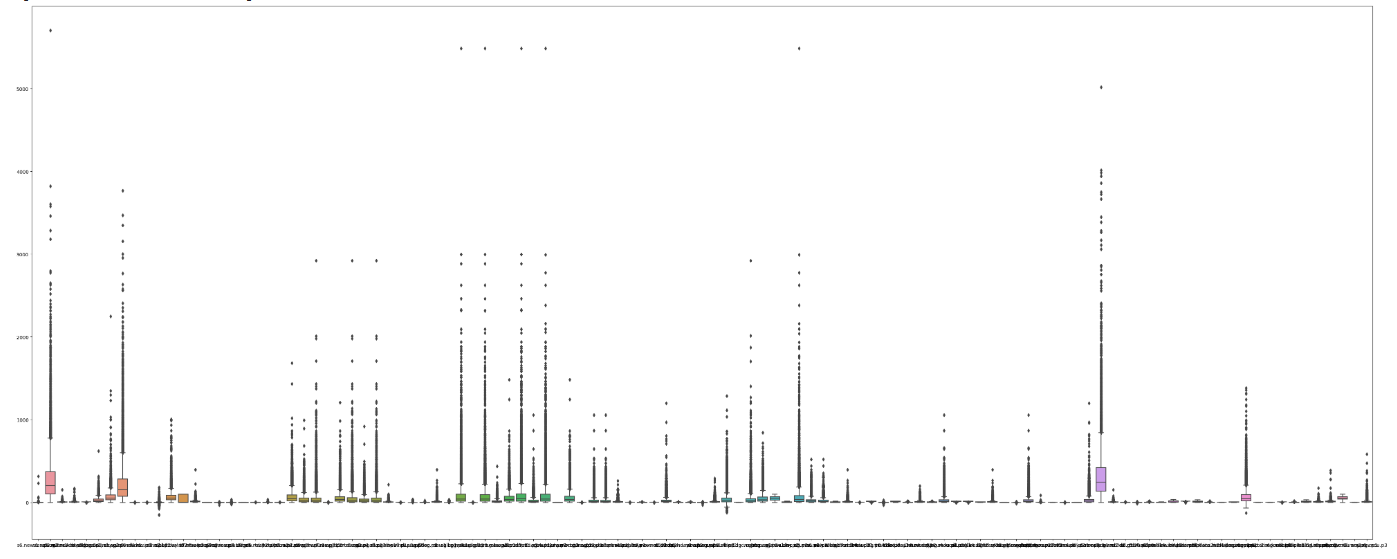
Calculate IQR: Calculate the interquartile range (IQR), which is the difference between the third quartile (Q3) and the first quartile (Q1).

Define Thresholds: Define upper and lower thresholds for outlier detection using the formula:

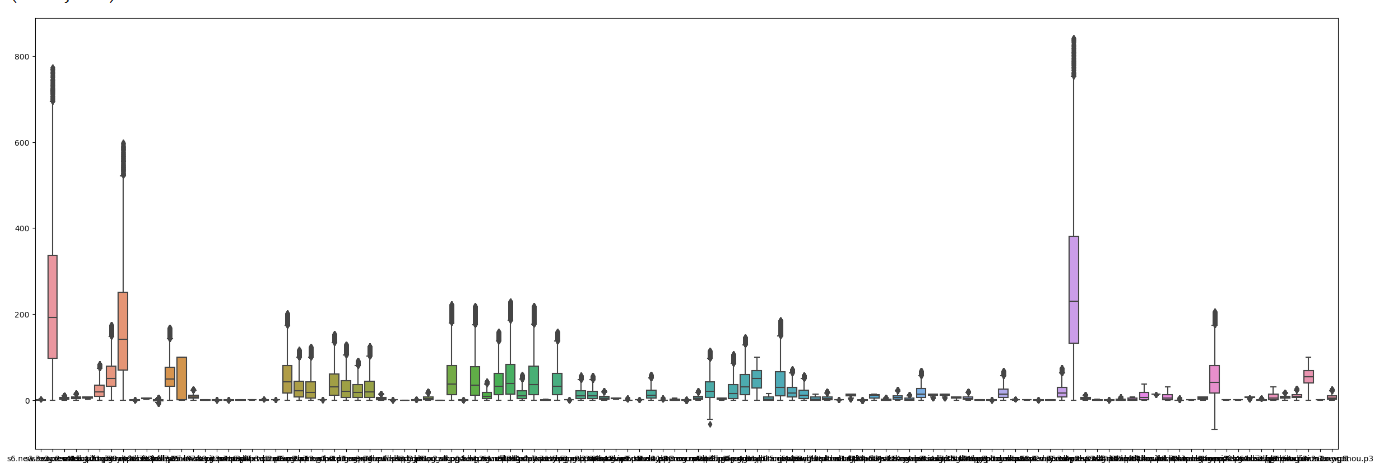
Upper threshold: Q3 + (1.5 \* IQR)

Lower threshold: Q1 - (1.5 \* IQR)

Identify Outliers: Identify outliers as data points that fall beyond the upper or lower thresholds.

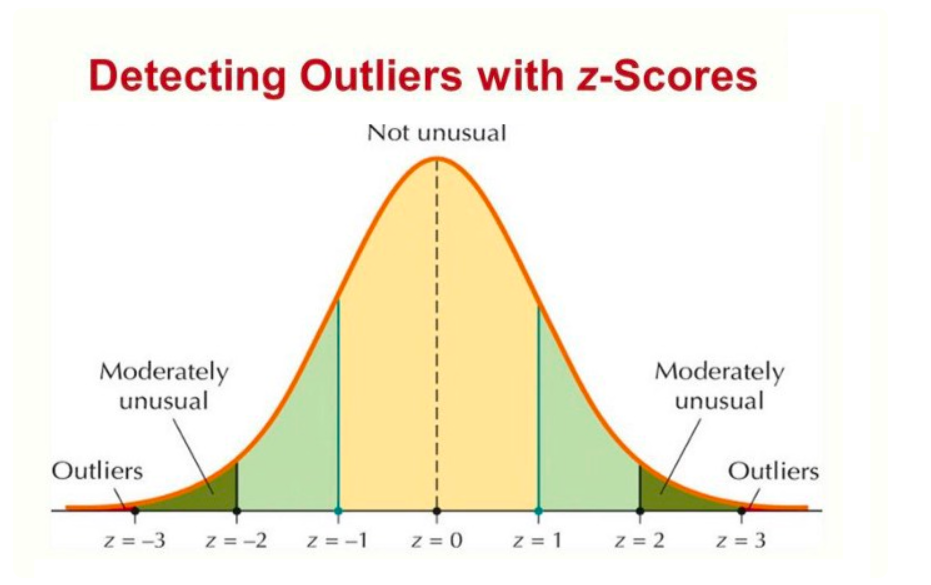


Outlier Treatment: Handle outliers by either removing them from the dataset, transforming them, or applying robust statistical methods to mitigate their impact on the analysis.



* Standardization: +/- 3 Sigma approach

Standardization, also known as z-score normalization, is a common technique used to rescale data to have a mean of 0 and a standard deviation of 1. The +/- 3 sigma approach is a method of outlier treatment based on z-scores, where data points that fall outside the range of +/- 3 standard deviations from the mean are considered outliers.



Here's a brief overview of the standardization process using the +/- 3 sigma approach:

Calculate Z-Scores: Compute the z-scores for each data point using the formula:

z= (x−μ)/ σ

Where:

𝑥-x is the individual data point.

𝜇-μ is the mean of the dataset.

𝜎-σ is the standard deviation of the dataset.

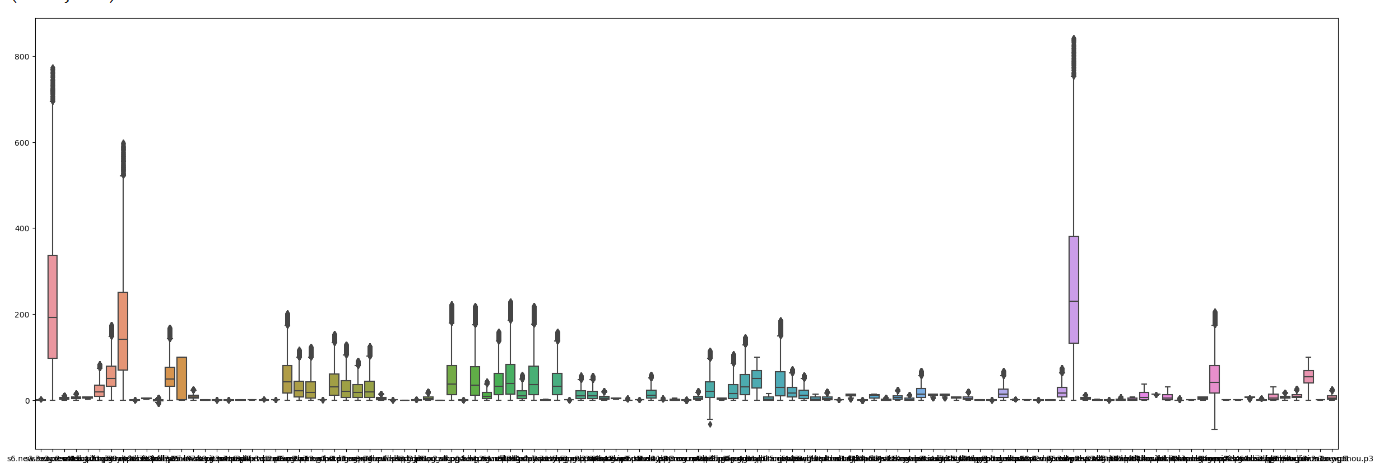
Define Thresholds: Determine the upper and lower thresholds for outlier detection based on z-scores:

Upper threshold: 𝜇+3𝜎

Lower threshold: μ−3σ

Identify Outliers: Data points that fall beyond the upper or lower thresholds are considered outliers.

Outlier Treatment: Handle outliers by either removing them from the dataset, transforming them, or applying robust statistical methods to mitigate their impact on the analysis.



* Capping & Flooring

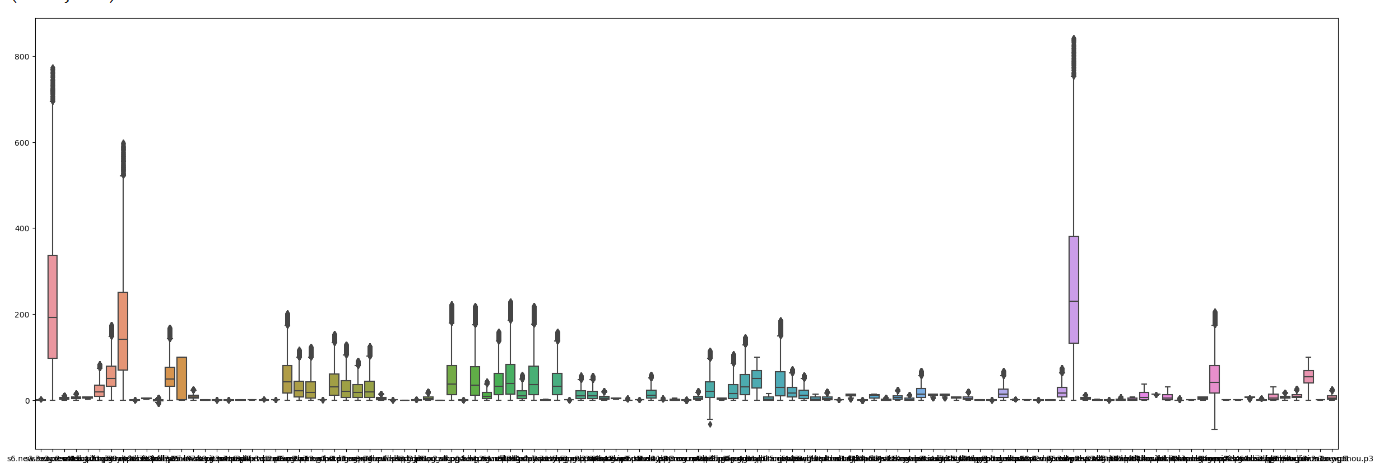
Capping and flooring, also known as winsorization, is a technique used to handle outliers by setting a threshold beyond which extreme values are replaced with a predefined value. This approach helps prevent extreme values from disproportionately influencing analysis results while preserving the overall distribution of the data.

Here's a brief overview of capping and flooring:

Identification of Outliers: Identify outliers in the dataset using statistical methods such as z-scores, box plots, or domain knowledge.

Define Thresholds: Determine upper and lower thresholds beyond which data points are considered outliers. These thresholds can be based on statistical measures like percentiles or standard deviations.

Capping & Flooring: Replace outlier values beyond the defined thresholds with predefined values known as caps (for upper outliers) and floors (for lower outliers). Commonly used values for caps and floors include the 95th percentile for upper outliers and the 5th percentile for lower outliers.



Implementation: Apply the capping and flooring technique to the dataset, replacing outlier values with the predefined caps and floors.

6.Missing Value Treatment

Missing value treatment involves identifying and handling missing data points in the dataset. This process includes identifying missing values, assessing their extent and pattern, and applying appropriate handling techniques such as imputation or deletion. Effective treatment ensures the accuracy and reliability of data analysis results.

* Remove records if NA’ s are less than 5%

Missing value treatment involves identifying and handling missing data points in the dataset. This process includes identifying missing values, assessing their extent and pattern, and applying appropriate handling techniques such as imputation or deletion. Effective treatment ensures the accuracy and reliability of data analysis results.

RESULT: After performing missing value treatment by removing records with NA values less than 5%, I retained 32 columns.

* Remove if NA’ s are 50% in any variable

During data preprocessing, we removed variables with missing values if the percentage of missing values in any variable exceeded 50%. This decision was made to ensure that variables with a significant amount of missing data do not bias our analysis or models. By removing such variables, we maintain the quality and reliability of the dataset for further analysis.

* Impute with Mean/Median, if variable is numeric and with Mode if variable is categorical

In our data preprocessing phase, we imputed missing values differently based on the type of variable:

For numeric variables, we imputed missing values with the mean or median of the respective variable. This approach helps preserve the central tendency of the numeric data and minimizes the impact of missing values.

For categorical variables, we imputed missing values with the mode (most frequent value) of the respective variable. Imputing with the mode ensures that the imputed values align with the most common category in the dataset, maintaining the integrity of categorical data.

RESULT: After imputing missing values with the mean or median for numeric variables and with the mode for categorical variables, I retained 32 columns.

7.Removing the highly correlated variables

During our data preprocessing phase, we identified and removed highly correlated variables from the dataset. Highly correlated variables exhibit strong linear relationships with each other, which can introduce multicollinearity issues in statistical models and reduce the interpretability of results. Here's a brief overview of our approach:

Identification: We calculated the correlation matrix for all pairs of variables in the dataset using methods like Pearson correlation coefficient or Spearman rank correlation coefficient.

Threshold Definition: We defined a threshold (e.g., 0.8 or -0.8) beyond which variables are considered highly correlated.

Removal: Variables exceeding the defined threshold were removed from the dataset to mitigate multicollinearity issues and streamline subsequent analysis.

RESULT:

|  |  |  |
| --- | --- | --- |
| S.NO | THRESHOLD VALUE | COLUMNS |
| 1 | 0.8 | 16 |
| 2 | 09 | 20 |
| 3 | 0.7 | 12 |

8.Multicollinearity (VIF>5)

In our data preprocessing phase, we addressed multicollinearity issues by identifying and mitigating variables with high variance inflation factors (VIFs) exceeding a threshold of 5. Multicollinearity occurs when independent variables in a regression model are highly correlated with each other, leading to unstable parameter estimates and reduced interpretability of the model.

Here's a brief overview of our approach:

VIF Calculation: We calculated the VIF for each independent variable in the dataset. VIF quantifies the severity of multicollinearity by measuring how much the variance of an estimated regression coefficient is inflated due to collinearity.

Threshold Definition: We set a threshold of 5 for VIF values. Variables with VIF greater than 5 were considered to exhibit multicollinearity.

Treatment: To address multicollinearity, we applied techniques such as removing one of the correlated variables, combining correlated variables into composite variables, or using regularization techniques like ridge regression.

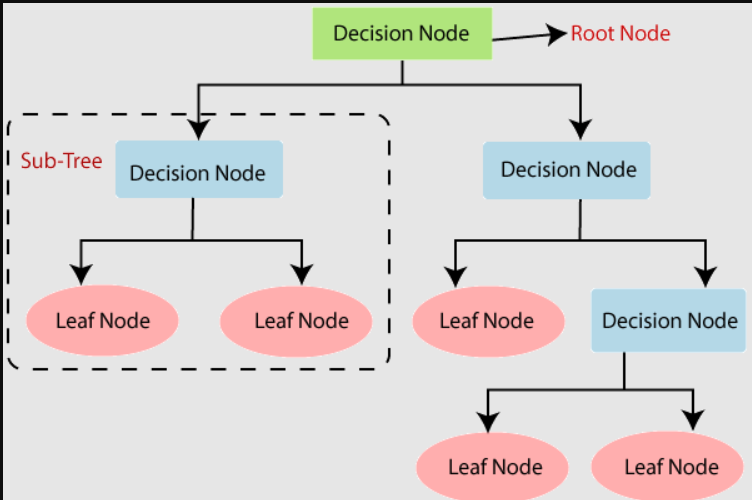
RESULT: After performing the Variance Inflation Factor (VIF) calculation, I identified and retained 8 columns in the dataset.

**MILESTONE-3**

Model Building

Decision Tree Classifier

We considered several machine learning algorithms for this classification problem, including logistic regression, decision trees and random forests. A Decision Tree classifier was chosen for its simplicity and interpretability. Decision Trees can handle both numerical and categorical data, making them a suitable choice for this churn prediction task.



Data Collection: Customer data was collected from various sources, including transaction records, customer service logs, and demographic information. After gathering the data, preprocessing steps were applied to ensure its quality and relevance.

Data Splitting: The processed data was split into training and testing sets, typically using an 75-25 split to ensure that the model is trained on a majority of the data but tested on a separate subset to evaluate performance.

Feature Scaling: We applied feature scaling to normalize the numerical features, ensuring that all features contribute equally to the model's decision-making process.

Model Training and Evaluation

Importing Decision Tree Classifier: We imported the Decision Tree Classifier model from a machine learning library (such as scikit-learn).

Model Training: The Decision Tree model was trained on the training dataset. We used the CART (Classification and Regression Trees) algorithm to learn the decision rules that lead to churn.

Model Evaluation:

Confusion Matrix: We evaluated the model's performance using a confusion matrix, which provides a summary of prediction results on the test set. The matrix includes True Positives, True Negatives, False Positives, and False Negatives.

RESULT:

* Confusion matrix for train data.

[[9091 1050]

[2346 2285]]

* Confusion matrix for test data.

[[3006 366]

[ 783 769]]

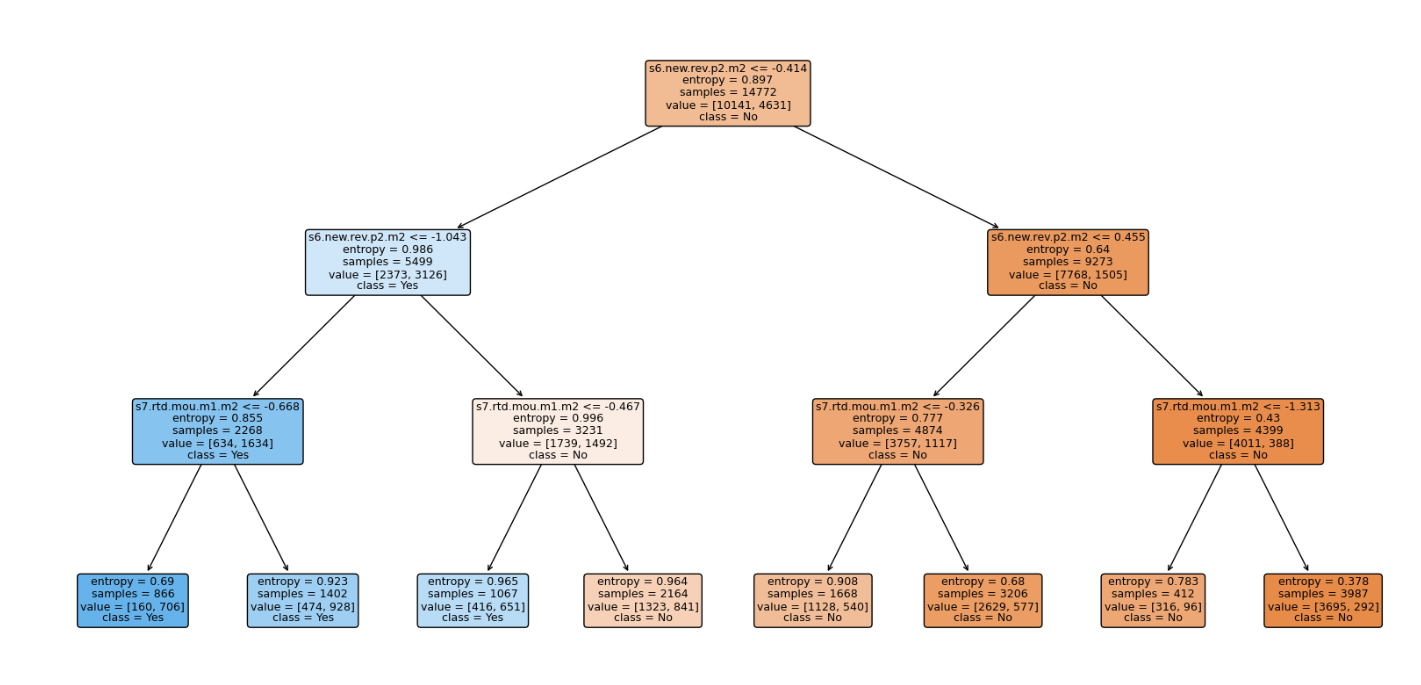
Accuracy Calculation: The accuracy of the model was calculated as the ratio of correctly predicted instances (both churn and no churn) to the total instances in the test set.

RESULT:

* Train Accuracy of the Decision Tree Classifier: 0.7701056051990252
* Test Accuracy of the Decision Tree Classifier: 0.7666531275385865

Visualization

Decision Tree Visualization: The trained Decision Tree was visualized to understand the decision rules and the hierarchy of features used by the model. This visualization helps in interpreting the model and explaining the predictions.

RESULT: 

Hyper Parameter Tuning

Hyperparameter tuning significantly improved the performance of the Decision Tree Classifier. By systematically exploring a range of hyperparameters, the best combination was identified, leading to enhanced accuracy and generalization. The use of Grid Search with cross-validation ensured a thorough and robust optimization process. This approach can be applied to various machine learning models to achieve better performance and reliability.

**MILESTONE 4:**

For the fourth milestone, we have completed the remaining models, namely Random Forest and Logistic Regression, and conducted thorough performance evaluations on each model to assess their effectiveness in addressing the project objectives. Additionally, we have documented the implementation process.

**Random Forest**

We explored various machine learning algorithms for tackling the classification problem at hand, including logistic regression, decision trees, and random forests. Opting for a Random Forest classifier, we aimed for a blend of robustness and interpretability. Random Forests offer the advantage of handling both numerical and categorical data effectively, making them well-suited for our churn prediction task.

Data Collection: Customer data was sourced from diverse channels, comprising transaction records, customer service logs, and demographic information. Following data aggregation, preprocessing steps were meticulously applied to ensure data quality and relevance.

Data Splitting: The processed dataset underwent partitioning into training and testing subsets, typically adhering to a 75-25 split. This allocation strategy ensured the model's exposure to a significant portion of the data during training while reserving a distinct subset for evaluating performance.

Feature Scaling: To maintain uniform influence across all features, we conducted feature scaling, normalizing the numerical features within the dataset. This step aimed to prevent any feature from dominating the model's decision-making process.

Model Training and Evaluation

Importing Random Forest Classifier: The Random Forest Classifier model was imported from a machine learning library, such as scikit-learn, to commence the training process.

Model Training: The Random Forest model underwent training on the designated training dataset. Leveraging the ensemble learning approach, the model utilized the random forest algorithm to amalgamate multiple decision trees, each contributing to the collective predictive power.

Model Evaluation:

Confusion Matrix Analysis: Performance assessment was conducted using a confusion matrix, providing a concise overview of the model's prediction outcomes on the test dataset. The matrix delineated True Positives, True Negatives, False Positives, and False Negatives.

RESULT:

* Confusion matrix for the training data:

[[10111 30]

[227 4404]]

* Confusion matrix for the test data:

[[2940 432]

[859 693]]

Accuracy Calculation: Model accuracy, indicative of its predictive prowess, was computed as the ratio of correctly predicted instances (comprising both churn and non-churn cases) to the total instances within the test set.

RESULT:

* Training Accuracy of the Random Forest Classifier: 0.9826022204170052
* Test Accuracy of the Random Forest Classifier: 0.7378147847278635

Documentation:

In the fourth milestone of our churn prediction project, we focused on comprehensive documentation to ensure clarity, reproducibility, and transparency. We summarized the project's objectives and significance, detailed data collection and preprocessing steps, and justified selecting the Decision Tree classifier for its simplicity and interpretability. The documentation covers model training, including data splitting, hyperparameter tuning, and evaluation using accuracy and confusion matrices.

**conclusion**

In our churn prediction project, we processed comprehensive customer data through meticulous data collection, understanding, and preprocessing. We explored and compared various machine learning models, including Decision Tree, Random Forest, and Logistic Regression. The Decision Tree model achieved a train accuracy of 77% and test accuracy of 76.67%, while the Random Forest model achieved a train accuracy of 98.26% and test accuracy of 73.78%. Hyperparameter tuning further enhanced the model performance. Overall, this project provides valuable insights and robust predictive capabilities for customer churn