**BITATHON - TELECOM ANALYSIS**

**TEAM NAME** **- THE BOYS**

**BENJAMIN ROY**

**SANJEEV NS**

**SRIHARISH R**

**Problem statements: -**

1. **Sentiment analysis**
2. **Churn analysis**
3. **Customer segmentation**
4. **Unique insights of customer transactions**

**Introduction: -**

**In today's digitally interconnected world, the telecommunications industry stands at the forefront of technological innovation and societal advancement. With the rapid proliferation of mobile devices, the emergence of 5G technology, and the increasing integration of Internet of Things (IoT) devices, the telecom sector has become a critical enabler of modern communication, commerce, and connectivity.**

**The aim of this report is to delve into the intricate landscape of the telecom industry, exploring various facets of its operations, challenges, and opportunities. By analyzing a comprehensive dataset encompassing diverse metrics such as subscriber demographics, network performance indicators, service utilization patterns, and customer satisfaction scores, we seek to unearth actionable insights that can drive informed decision-making and strategic planning within telecom organizations.**

**Objective: -**

1. **Identify emerging trends and patterns within the telecom industry landscape.**
2. **Segment the customer base to personalize marketing and service offerings.**
3. **Predictive Modeling for CLV.**
4. **Sentiment analysis of feedback data by text polarization.**

**Methodology: -**

**1. Data Collection:**

**- We will start by obtaining a comprehensive dataset from sources within the telecom industry. This dataset may include information such as subscriber demographics, customer details, service utilization patterns, customer feedback, and more.**

**- The dataset will be carefully curated to ensure its integrity, completeness, and relevance to our analysis objectives.**

**2. Data Preprocessing:**

**- Before analysis, the dataset will undergo preprocessing steps to clean and prepare the data for analysis.**

**- This includes handling missing values, removing duplicates, standardizing formats, and encoding categorical variables as necessary. This can be done by the Pandas module that is in python.**

**- Additionally, outliers may be identified and addressed appropriately to prevent them from skewing the analysis results.**

**3. Exploratory Data Analysis (EDA):**

**- Exploratory Data Analysis involves the systematic exploration of the dataset to uncover patterns, trends, and relationships among variables.**

**- Descriptive statistics, data visualization techniques, and correlation analyses will be employed to gain insights into the underlying structure of the data.**

**- This phase aims to identify potential areas of interest for further investigation and hypothesis testing.**

**4. Statistical Techniques:**

**- Various statistical techniques will be applied to analyze the dataset and extract meaningful insights.**

**- This may include hypothesis testing, regression analysis, clustering, and factor analysis, depending on the nature of the data and the research questions at hand.**

**- Statistical tests will be used to validate findings and assess the significance of observed relationships.**

**5. Machine Learning Algorithms:**

**- Machine learning algorithms will be utilized to uncover complex patterns and predictive models within the data.**

**- Supervised learning techniques such as classification and regression may be employed to predict customer behavior, churn rates, or service usage patterns. In this model we have implemented various ML models like logistic regression, Random Forest and SVM classifier.**

**- Unsupervised learning methods like clustering and dimensionality reduction may be used to segment the customer base or identify hidden patterns in the data.**

**6. Data Visualization:**

**- Data visualization tools and techniques will be employed to effectively communicate the findings of the analysis.**

**- Charts, graphs, heatmaps, and interactive Data Visualizations will be used to present key insights in a visually appealing and informative manner.**

**- Visualization aids in understanding complex relationships within the data and facilitates the interpretation of results by stakeholders.**

**7. Validation and Interpretation:**

**- The validity of the analysis results will be rigorously assessed to ensure their reliability and robustness.**

**- Findings will be interpreted in the context of domain knowledge and business objectives, with input from subject matter experts where necessary.**

**- Sensitivity analyses and model validations may be conducted to test the stability and generalizability of the findings.**

**8. Communication of Findings:**

**- Finally, the insights gained from the analysis will be synthesized into a comprehensive report, presentation, or dashboard.**

**- Clear and concise explanations of the findings, along with actionable recommendations, will be provided to stakeholders.**

**- The communication of findings aims to empower decision-makers to make informed choices and drive strategic initiatives within the telecom organization.**

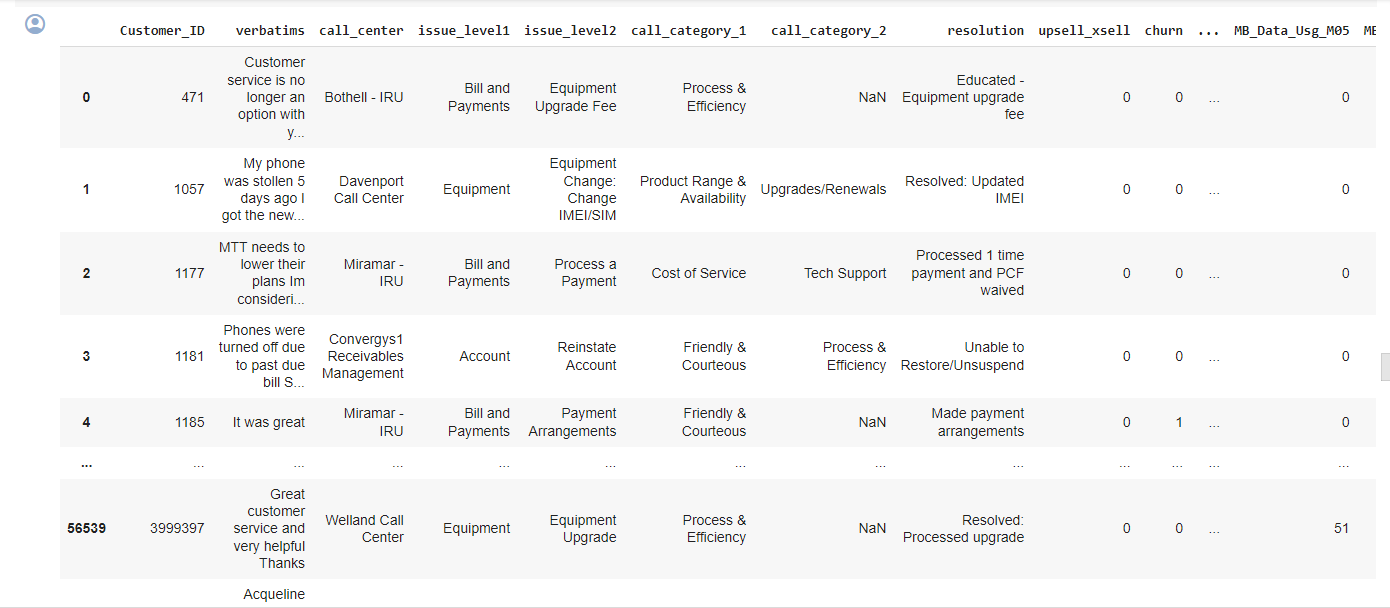
**By following this methodology, we aim to extract meaningful insights from the telecom dataset and provide valuable recommendations for optimizing operations, enhancing customer satisfaction, and driving business growth within the industry.**

**ANALYSIS: -**

**The primary objective of this analysis is to delve into a comprehensive telecom dataset, dissecting its intricacies to extract meaningful insights that can inform strategic initiatives and operational enhancements within the industry. By leveraging advanced analytics techniques, we aim to address key questions and objectives based on this analysis.**

1. **Dataset: -**

**The dataset consists of three excel files Call center dataset, customer transaction dataset and geography lookup dataset. Each dataset has various columns, and it is highly correlated to each other dataset. So, we decided to merge the datasets by merging two Data Frames df1 and df2 based on the common column "Customer\_ID". This means that it combines the information from both Data Frames where the "Customer\_ID" values match. The result is stored in a new Data Frame called merged\_df. Then we further merge the previously merged Data Frame merged\_df with another Data Frame df3, this time based on the common column "zipcode\_primary". Again, the merge operation combines information from both Data Frames where the "zipcode\_primary" values match. The resulting Data Frame is stored back in the variable merged\_df, effectively updating it with the additional information from df3.**



1. **Pre-Processing: -**

**In this step, we handled missing data and duplicate data and also did outlier analysis. In the missing data analysis, we did mean imputation. To perform mean imputation** **for missing data in the Data Frame merged\_df, you can use the fillna() method along with the** **mean() function to replace missing values with the meaning of each respective column.** **The inplace=True argument ensures that the changes are applied directly to merged\_df, modifying it in place. For the outlier analysis,** **we are using a box plot, also known as box-and-whisker plot, which is a graphical method for detecting outliers in a dataset. It provides a visual summary of the distribution of the data along with indicators of potential outliers. Constructed from quartiles, medians, and interquartile ranges, a box plot provides a succinct summary of the data's central tendency and spread. In its depiction, the box represents the interquartile range (IQR), with the median line dividing it into two halves. Whiskers extend from the edges of the box to the minimum and maximum values within a distance of 1.5 times the IQR from the quartiles. Any data points lying beyond the whiskers are considered potential outliers. By visually inspecting the box plot, analysts can readily identify extreme values that may indicate anomalies, errors, or unique phenomena in the dataset. Outliers detected through box plot analysis can prompt further investigation to understand their origin and potential impact on subsequent analyses or models. Whether through removal, transformation, or adjustment, addressing outliers appropriately ensures the integrity and accuracy of data-driven insights and decisions. Then we did label encoding for all of the string/object variables for easier analysis. Label encoding is a technique used in data preprocessing to convert categorical variables into numerical format, particularly when applying machine learning algorithms that require numerical input. In label encoding, each unique category within a categorical variable is assigned a numerical label. Here, we'll discuss the label encoding process for seven variables in a paragraph.**

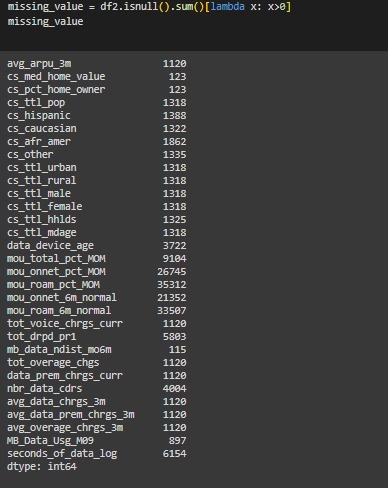


FIGURE - missing values in the dataset

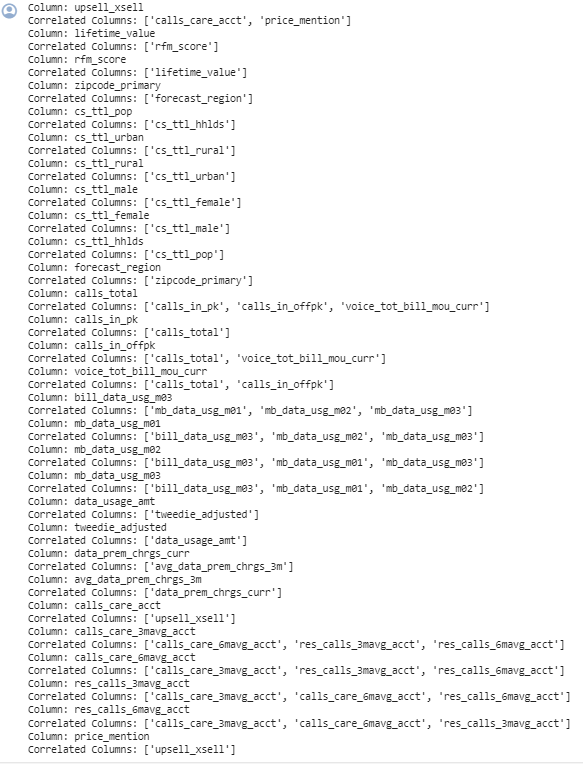
1. **Descriptive analysis: -**

* **CORRELATION ANALYSIS:**

**The script begins by initializing an empty dictionary called correlated\_columns. This dictionary will serve as a repository for storing the names of columns that exhibit significant correlations with each other. It then iterates through each column in the correlation matrix, creating an empty list within the correlated\_columns dictionary for each column. This list will later hold the names of columns that are correlated with the current column being examined.**

**Within a nested loop, the script compares each column with all other columns in the correlation matrix, excluding self-comparisons. If the correlation coefficient between two columns exceeds a predefined threshold, typically set at 0.7 or -0.7, it indicates a strong positive or negative correlation respectively. In such cases, the name of the correlated column is appended to the list associated with the current column in the correlated\_columns dictionary.**

**Once the correlations have been identified and stored in the dictionary, the script iterates over the dictionary and prints the results, displaying the names of columns along with their correlated counterparts. This allows analysts to quickly identify and investigate potential relationships between variables within the dataset.**



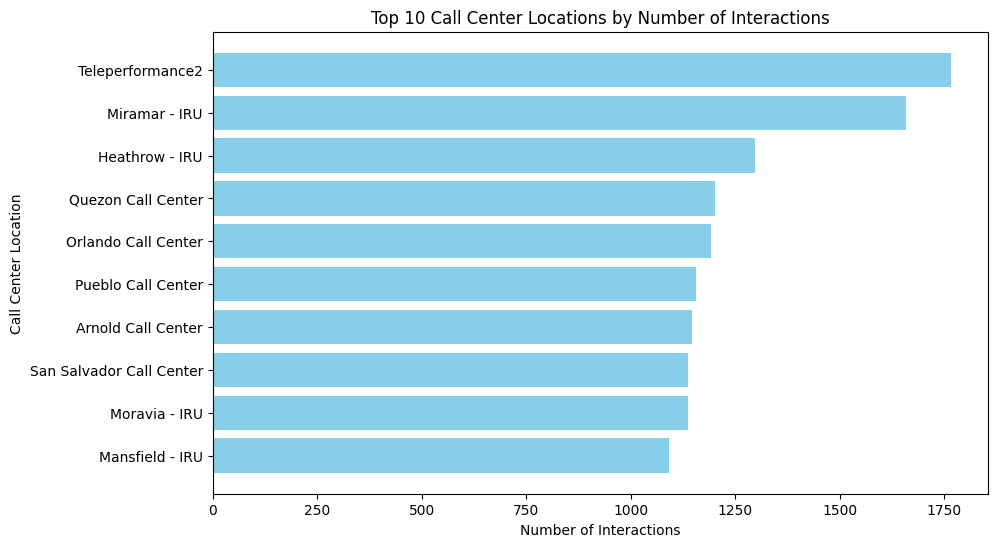
* **CALL CENTER INTERACTION ANALYSIS:**

**The realm of customer service and telecommunications, understanding the distribution of interactions across different call center locations is pivotal for optimizing service delivery and resource allocation. The provided Python script offers a streamlined approach to visualizing this distribution, focusing on the top 10 call center locations with the highest number of interactions.**

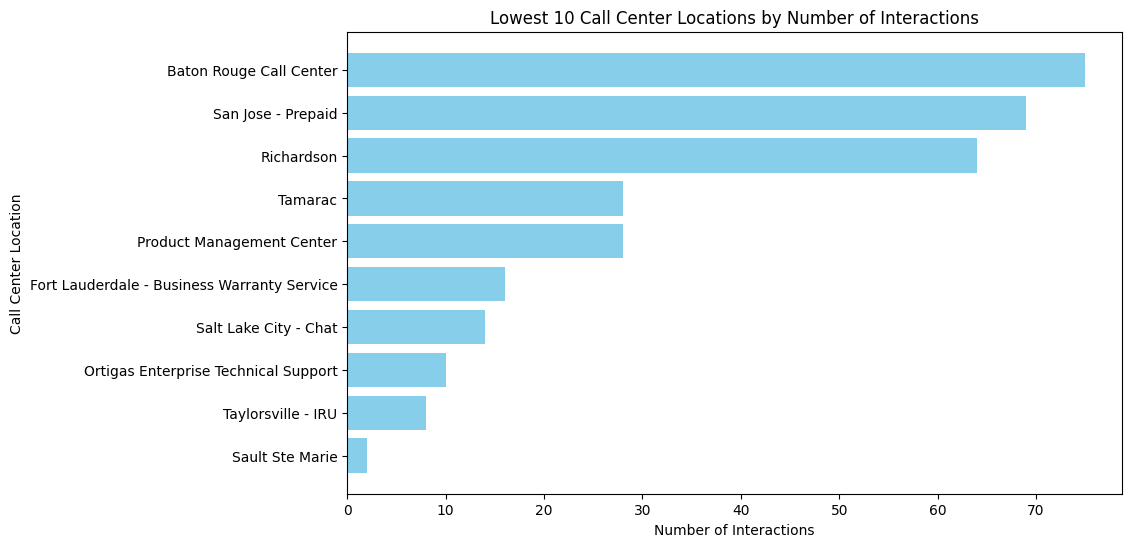
**The script begins by grouping the data based on call center locations and counting the number of interactions for each location. This aggregated information is then sorted in descending order to highlight the call center locations with the most interactions. Subsequently, the top 10 locations are selected for visualization, ensuring a focused representation of the most significant contributors to the overall interaction volume.**

**Utilizing Matplotlib, the script generates a horizontal bar plot that vividly displays the number of interactions for each of the top 10 call center locations. Each bar corresponds to a specific location, with its length proportional to the number of interactions. This visual representation provides a clear and intuitive way to compare the performance of different call center locations, identifying those with the highest volume of interactions.**

**By presenting this information in a concise and visually appealing manner, the script facilitates data-driven decision-making within telecom organizations. Stakeholders can quickly discern trends, identify areas of high demand, and allocate resources effectively to ensure optimal customer service delivery. Moreover, the script's flexibility allows for easy customization, enabling users to tailor the analysis to specific needs or explore additional insights as required.**



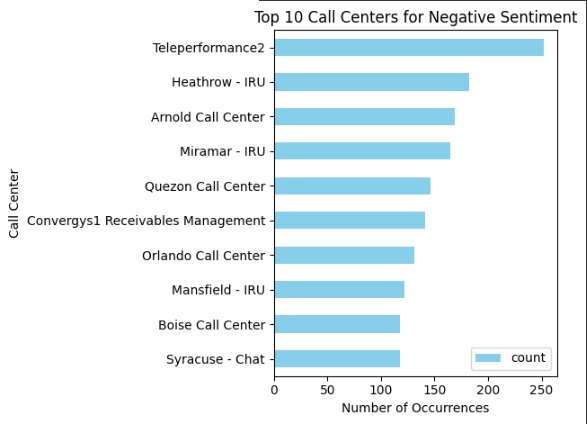
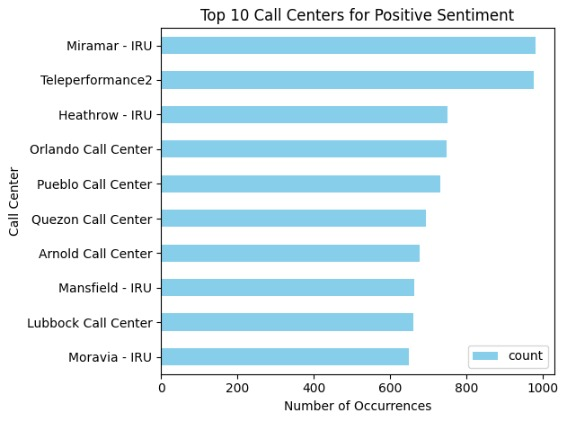
**The script provided here offers a practical approach to visualizing this distribution by focusing on the lowest 10 call center locations based on the number of interactions they handle. The script commences by aggregating the data according to call center locations and computing the number of interactions associated with each location. This structured representation enables a clear understanding of the workload distribution across different call center facilities. Subsequently, the data is sorted in ascending order, ensuring that call center locations with the lowest interaction counts are prioritized for analysis. By selecting the lowest 10 locations from the sorted dataset, the script emphasizes the areas where interaction volumes are comparatively lower. This targeted approach facilitates focused examination and enables organizations to identify potential challenges or areas for improvement in call center performance. Leveraging Matplotlib, the script generates a horizontal bar plot that visually presents the number of interactions for each of the selected call center locations. Each bar corresponds to a specific location, with its length proportional to the number of interactions handled by the respective call center. This graphical representation offers a concise and intuitive means of comparing the performance of different call center locations, highlighting those with the lowest interaction volumes. Through the visualization of the lowest 10 call center locations by the number of interactions, organizations gain valuable insights into the distribution of workload and resource utilization across their service facilities. This information serves as a foundation for strategic decision-making, allowing stakeholders to allocate resources effectively, optimize staffing levels, and implement targeted interventions to enhance service delivery and customer satisfaction.**

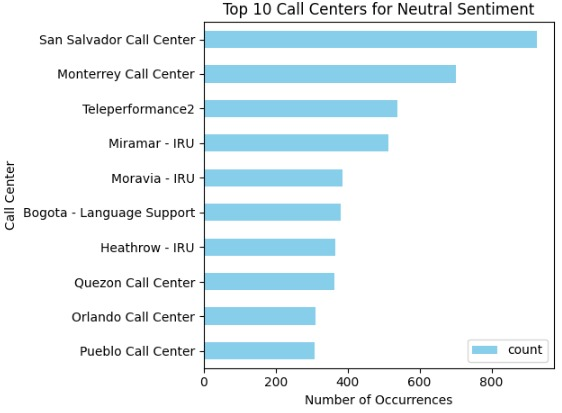


* **Customer service analytics**

**Understanding the sentiment distribution across various call center locations is imperative for enhancing service quality and customer satisfaction. The script provided here offers a systematic approach to visualizing this sentiment distribution, focusing on the top 10 call centers for each sentiment category.**

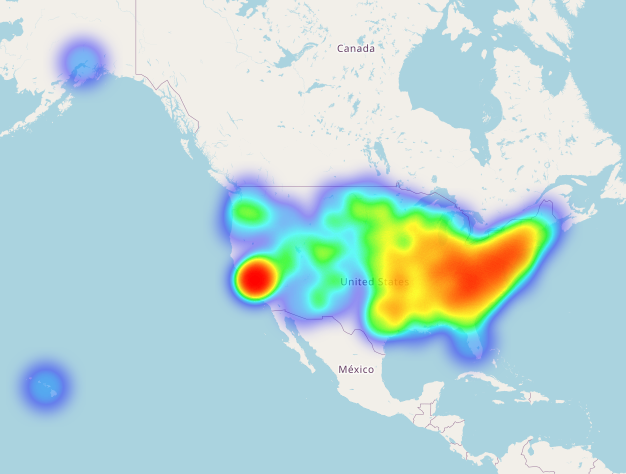
**Beginning with the aggregation of data based on both sentiment categories and call center locations, the script computes the occurrences of each sentiment-call center combination. This structured representation provides valuable insights into how customer sentiments are distributed across different service facilities. By sorting the data in descending order within each sentiment category, the script prioritizes call centers with the highest occurrences of sentiments for further analysis. This enables organizations to identify key areas of concern or excellence in customer interactions, facilitating targeted interventions and resource allocation. Utilizing Matplotlib, the script generates a set of subplots, each dedicated to one sentiment category, showcasing the top 10 call centers with the highest occurrences of sentiments. These horizontal bar plots offer a visual representation of sentiment distribution, allowing stakeholders to easily compare performance across different call center locations. Through the visualization of sentiment occurrences across call centers, organizations gain actionable insights into customer perceptions and service delivery effectiveness. By identifying top-performing call centers and areas for improvement, organizations can drive continuous enhancement of customer service quality and overall business performance.**





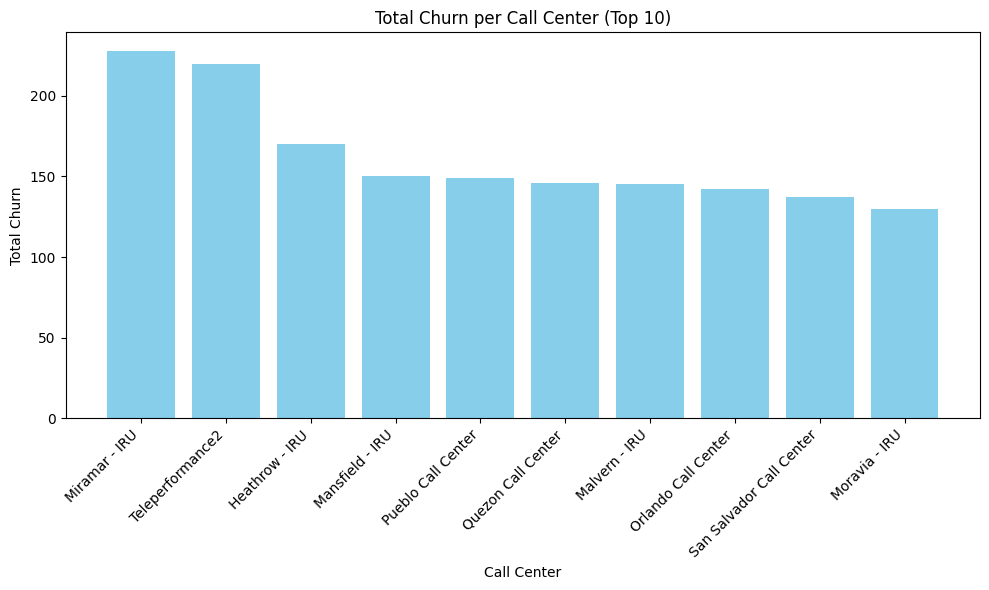
* **Customer density Heatmap:**

**The creation of a customer density heatmap serves as a powerful tool for understanding the spatial distribution of customer interactions and service usage. By visualizing where customer activities are concentrated geographically, telecom companies can gain valuable insights into network performance, service demand, and potential areas for improvement. Typically, telecom datasets contain information such as call records, network signal strength measurements, or customer locations. Leveraging this data, the process begins with extracting the geographical coordinates of customer activities, whether it's making calls, using data services, or experiencing network connectivity. Once the coordinates are obtained, the data is aggregated to calculate customer density metrics within specific geographical regions. This aggregation could involve grouping data into grid cells, administrative boundaries, or other predefined spatial units. Metrics such as the number of calls, data usage, or signal strength measurements are then calculated for each region. Using Python libraries like Folium, Matplotlib, or Plotly, the aggregated data is visualized as a heatmap. The heatmap provides a clear and intuitive representation of customer density, with colors indicating varying levels of activity or usage across different geographical areas. Interpreting the heatmap allows telecom companies to identify hotspots of customer activity, network congestion, or areas with high service demand. This information can inform strategic decisions related to network planning, infrastructure deployment, marketing campaigns, and customer service initiatives.**



* **Churn Rate per call center:**

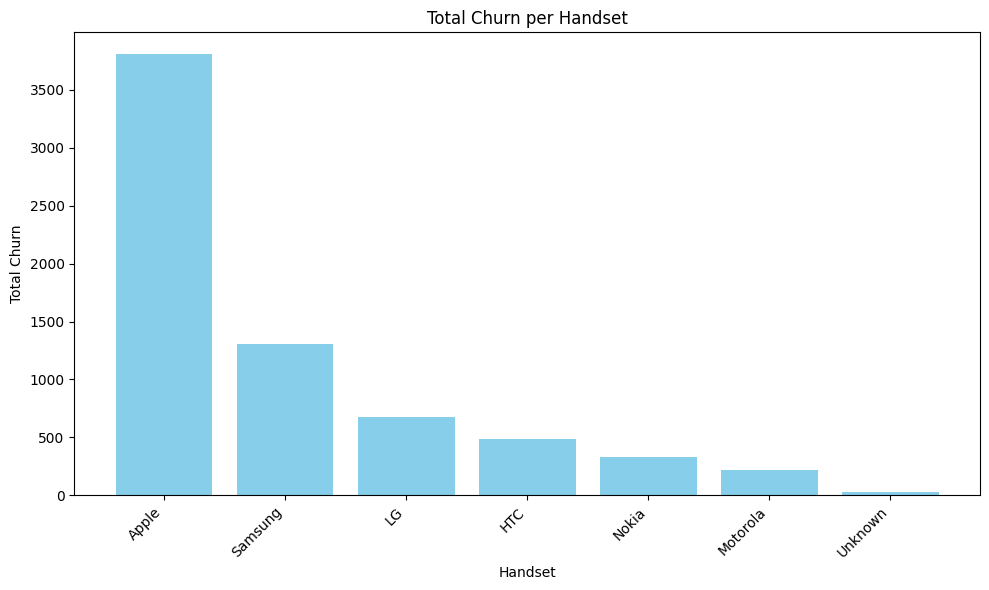
**A systematic approach to analyze and visualize the total churn per call center within a telecommunications dataset. By focusing on the top 10 call centers with the highest churn rates, businesses gain valuable insights into areas of potential concern and opportunities for intervention to mitigate churn. The process begins by aggregating the data based on call center identifiers and summing the churn values, yielding the total churn per call center. This step provides a comprehensive overview of the churn situation across different service facilities. Subsequently, the data is sorted in descending order based on the total churn values, ensuring that call centers with the highest churn rates are prominently featured. This sorting facilitates the identification of the most critical areas requiring attention within the telecom network. The script then selects the top 10 call centers with the highest total churn values for visualization, enabling stakeholders to focus their analysis on the most significant contributors to overall churn. Utilizing Matplotlib, the script generates a bar chart that vividly depicts the total churn per call center for the top 10 call centers. Each bar represents a call center, with its height proportional to the total churn value. This visual representation offers a clear and intuitive way to compare churn rates across different service facilities, aiding in the identification of hotspots and areas requiring targeted interventions.**



* **churn rates associated with different handset models:**

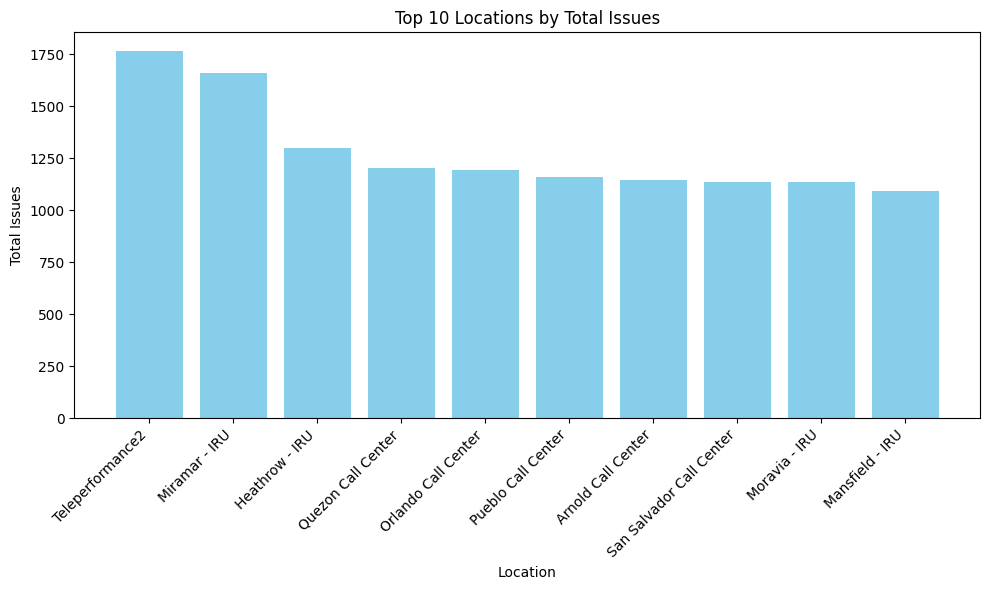
**The churn rates associated with different handset models. By leveraging this script, telecom companies can effectively identify the top-performing handsets in terms of churn, enabling strategic decision-making to mitigate customer attrition and enhance overall service quality. The script begins by aggregating the dataset based on handset identifiers and computing the total churn for each handset model. This step provides a comprehensive overview of churn patterns across various handset models within the network.**

**Following the aggregation, the data is sorted in descending order based on total churn values, ensuring that handsets with the highest churn rates are prioritized for analysis. This sorting enables telecom companies to pinpoint the handsets contributing most significantly to churn, thereby facilitating targeted interventions and improvement initiatives. Subsequently, the script selects the top 10 handsets with the highest churn rates for visualization, enhancing the interpretability and actionable insights derived from the analysis. This visualization, in the form of a bar chart, vividly depicts the churn rates associated with each handset model, thereby enabling telecom companies to identify trends, patterns, and outliers within the dataset.**

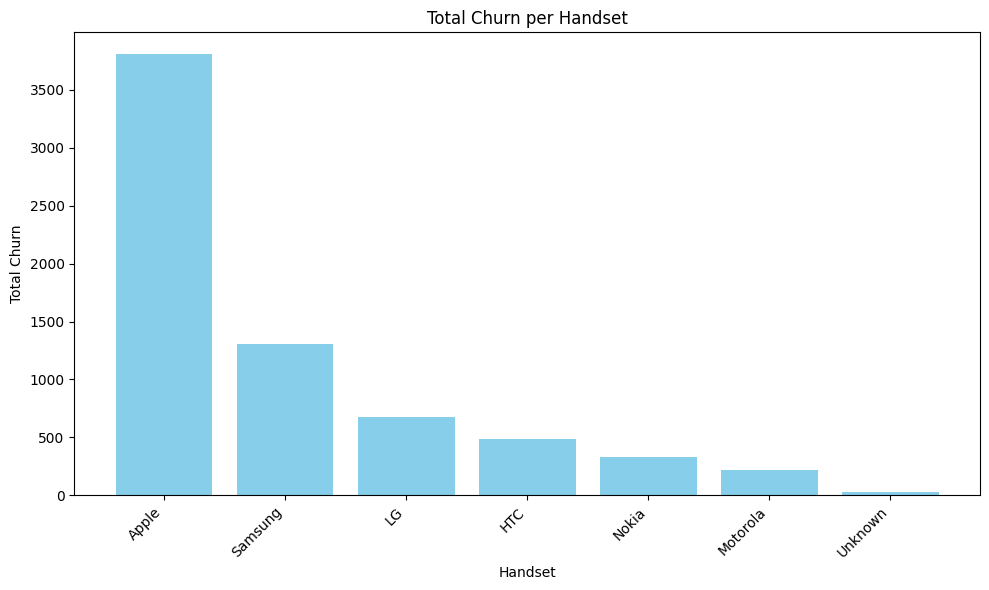


* **Analyzing the distribution of reported issues across various call center locations:**

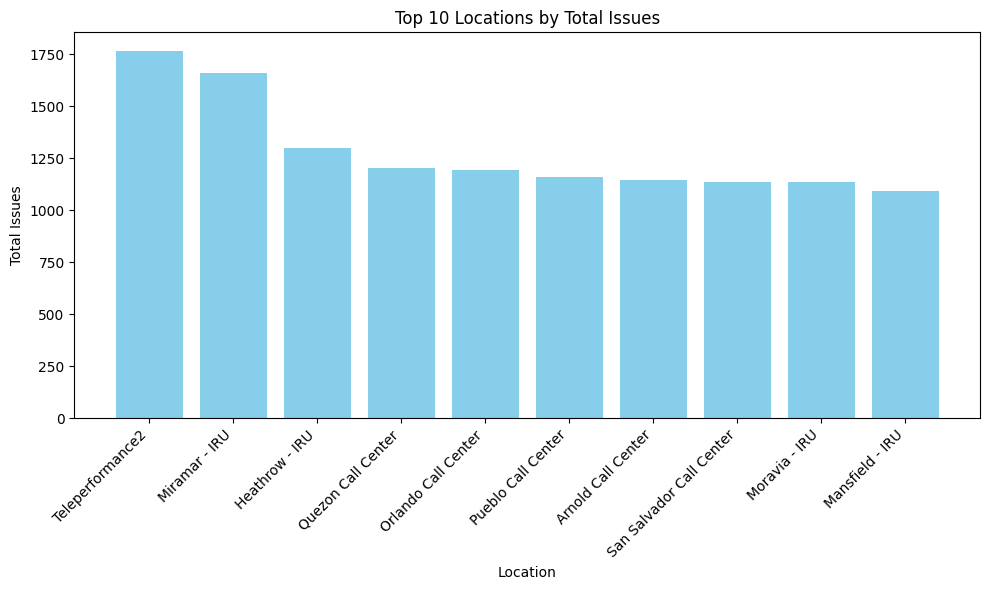
**A methodical approach to visualizing and analyzing the distribution of reported issues across various call center locations within a telecommunications dataset. By focusing on the top 10 locations with the highest total issues, telecom companies can efficiently identify and address areas requiring attention and improvement. Initially, the script aggregates the dataset based on call center locations and counts the occurrences of reported issues at each location. This step provides a comprehensive overview of the distribution of issues across different service facilities. Following data aggregation, the script sorts the locations in descending order based on the total issue count. This sorting allows telecom companies to identify the locations with the most significant number of reported issues, thereby prioritizing efforts for resolution and improvement. Subsequently, the script selects the top 10 locations with the highest total issue counts for visualization. Utilizing Matplotlib, it generates a bar chart where each bar represents a location, and the height of the bar reflects the total number of reported issues. This visualization enables stakeholders to easily identify locations with the most prevalent issues and allocate resources accordingly.**



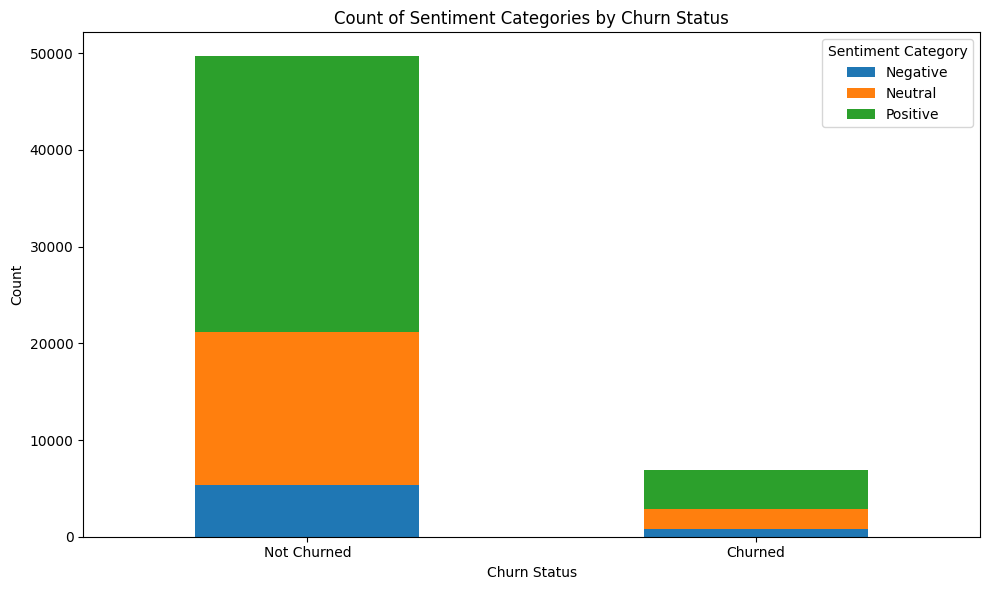
* **Total Churn per Handset**



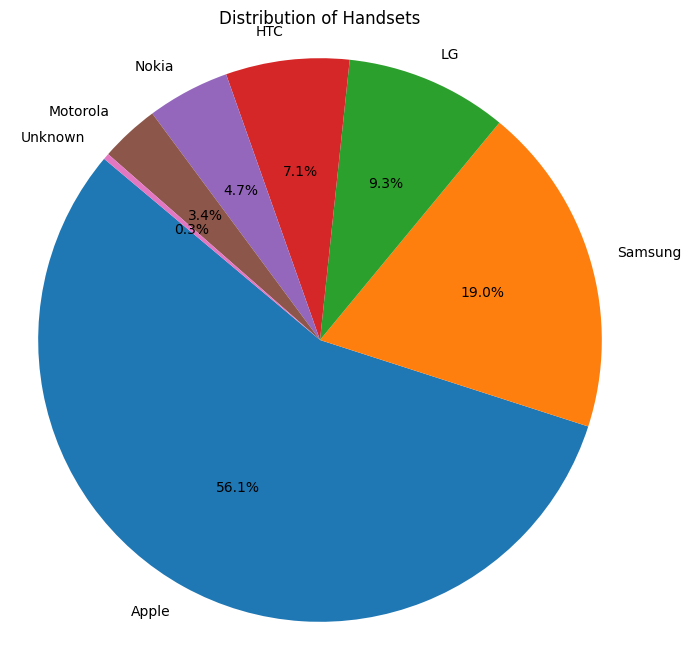
* **Top 10 Locations by Total Issues**



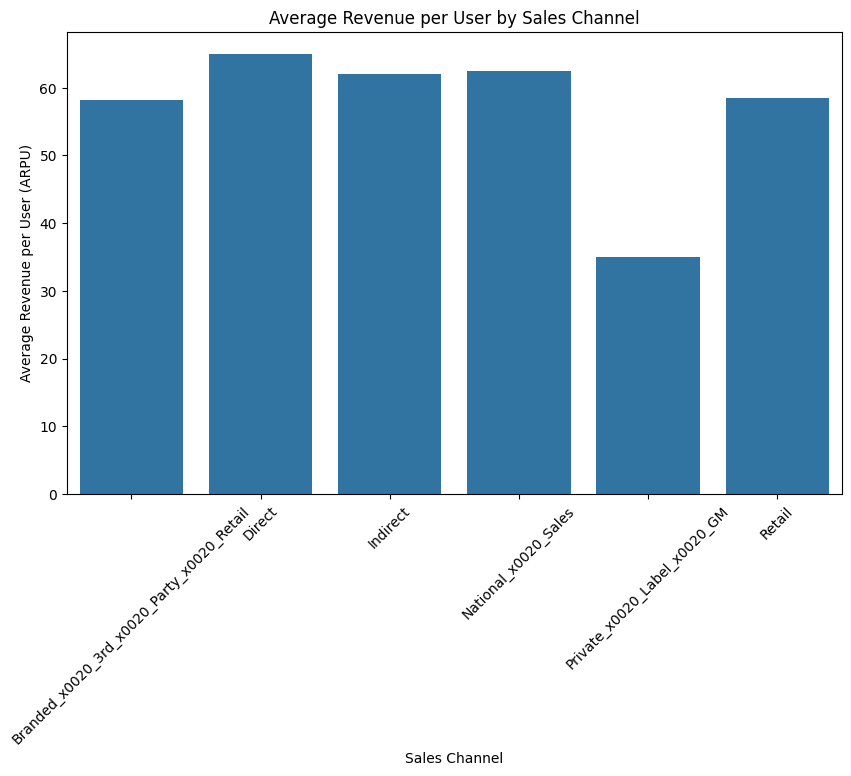
* **Count of Sentiment Categories by Churn Status**



* **Distribution of Handsets**



* **Average Revenue per User by Sales Channel**



1. **Predictive analysis: -**

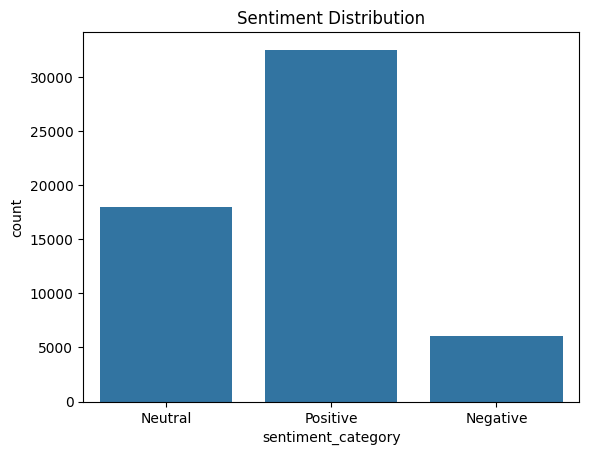
* **Sentimental Analysis:**

**Sentiment analysis on telecom datasets involves extracting insights from customer interactions, reviews, or feedback to gauge their sentiment towards the services provided. By analyzing text data from sources such as customer reviews, call transcripts, or social media mentions, sentiment analysis helps telecom companies understand customer satisfaction levels, identify areas for improvement, and tailor their services to meet customer needs more effectively. This analysis enables businesses to proactively address concerns, enhance customer experiences, and ultimately foster stronger relationships with their clientele. In this sentimental analysis we have used 3 models. They are:**

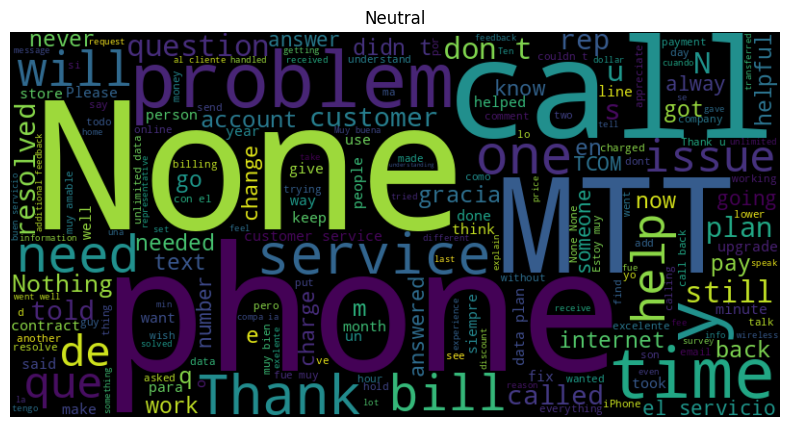
* **Logistic regression**
* **Random forest classifier**
* **SVM classifier**

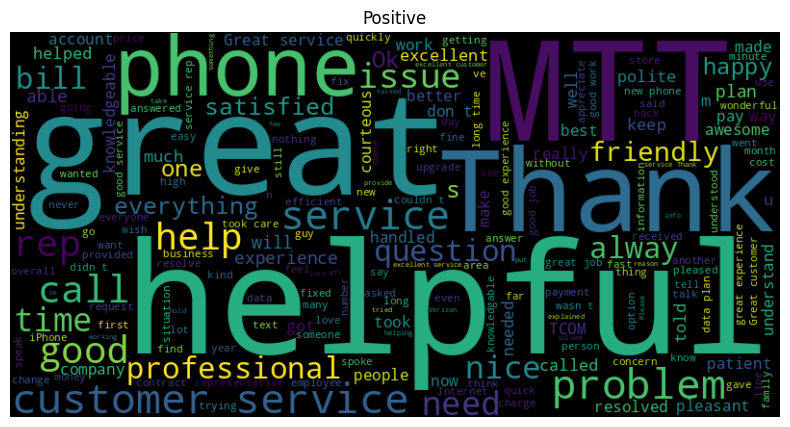
**Pre-processing for sentimental analysis:**

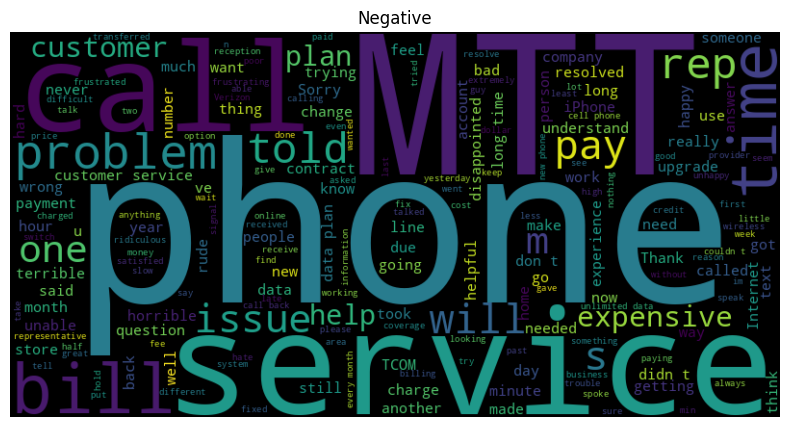
**The script employs the TextBlob library to conduct sentiment analysis on textual data extracted from a telecommunications dataset. By leveraging natural language processing techniques, the script evaluates the sentiment polarity of customer verbatims or feedback, enabling telecom companies to gauge customer sentiment towards their services. The get\_sentiment function utilizes TextBlob to analyze the sentiment polarity of each text entry, assigning a numerical score that represents the positivity or negativity of the sentiment expressed. Subsequently, the sentiment scores are appended to the dataset under the 'sentiment\_score' column. To further enhance interpretability, the categorize\_sentiment function categorizes the sentiment polarity scores into three distinct categories: 'Positive', 'Negative', or 'Neutral', allowing for a more straightforward analysis of customer sentiment trends. Overall, this approach facilitates the extraction of valuable insights from textual data within the telecom dataset, empowering companies to identify areas of customer satisfaction and areas requiring improvement. By understanding customer sentiment, telecom companies can tailor their services and strategies to better meet customer needs and enhance overall satisfaction levels.**



**Then we implement a world cloud module to check the key words of the text variable for each sentiment category.**

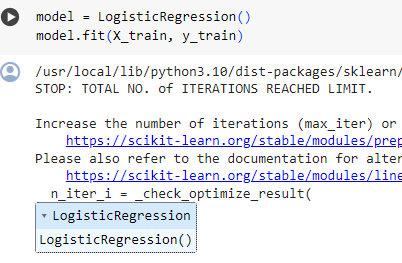






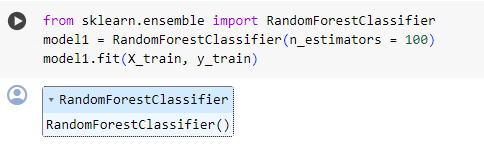
**Logistic regression:**

**Logistic regression is a powerful statistical technique used in sentiment analysis to predict the sentiment of textual data, categorizing it into predefined outcomes such as positive, negative, or neutral. In this method, the sentiment labels serve as the dependent variable, while the features extracted from the text, such as word frequencies or sentiment scores, act as independent variables. The logistic regression model estimates the probability of each sentiment category based on the extracted features. By fitting the model to labeled training data, it learns the relationship between the features and the sentiment labels. During prediction, the model calculates the probability of each sentiment category for new text inputs and assigns the category with the highest probability as the predicted sentiment. For instance, in a telecom dataset, logistic regression can be employed to analyze customer verbatims or feedback, predicting whether the sentiment expressed is positive, negative, or neutral. Features extracted from the text, such as sentiment scores or word frequencies, are used as input variables for the logistic regression model. The model then outputs the probability of each sentiment category for a given text input, enabling telecom companies to classify customer sentiment effectively. Ultimately, logistic regression in sentiment analysis provides telecom companies with a valuable tool to understand customer sentiment and tailor their services and strategies accordingly. By accurately categorizing customer feedback into positive, negative, or neutral sentiments, businesses can identify areas of strength and improvement, ultimately enhancing customer satisfaction and loyalty.**



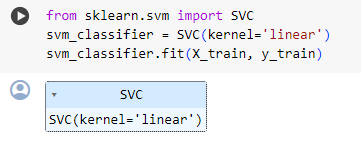
**Random Forest Classifier:**

**Random Forest Classifier is a robust machine learning algorithm utilized in sentiment analysis to predict the sentiment of textual data and categorize it into predefined outcomes such as positive, negative, or neutral. Unlike logistic regression, which is a linear model, Random Forest Classifier is an ensemble learning method that combines multiple decision trees to make predictions. In sentiment analysis, the Random Forest Classifier works by training a multitude of decision trees on various subsets of the training data, each tree considering different features and instances. During prediction, each decision tree independently predicts the sentiment of a given text input, and the final sentiment category is determined by a majority vote among all the trees. For example, in a telecom dataset, the Random Forest Classifier can analyze customer verbatims or feedback to determine whether the sentiment expressed is positive, negative, or neutral. Features extracted from the text, such as sentiment scores, word frequencies, or semantic features, are used as input variables for the classifier. The classifier then leverages the collective decision-making of multiple decision trees to classify the sentiment of the text accurately. The Random Forest Classifier offers several advantages for sentiment analysis, including its ability to handle large datasets, mitigate overfitting, and provide robust predictions even in the presence of noisy or irrelevant features. By effectively categorizing customer sentiment into positive, negative, or neutral outcomes, telecom companies can gain valuable insights into customer perceptions and preferences, enabling them to tailor their services and strategies to enhance overall customer satisfaction and loyalty.**



**SVM Classifier:**

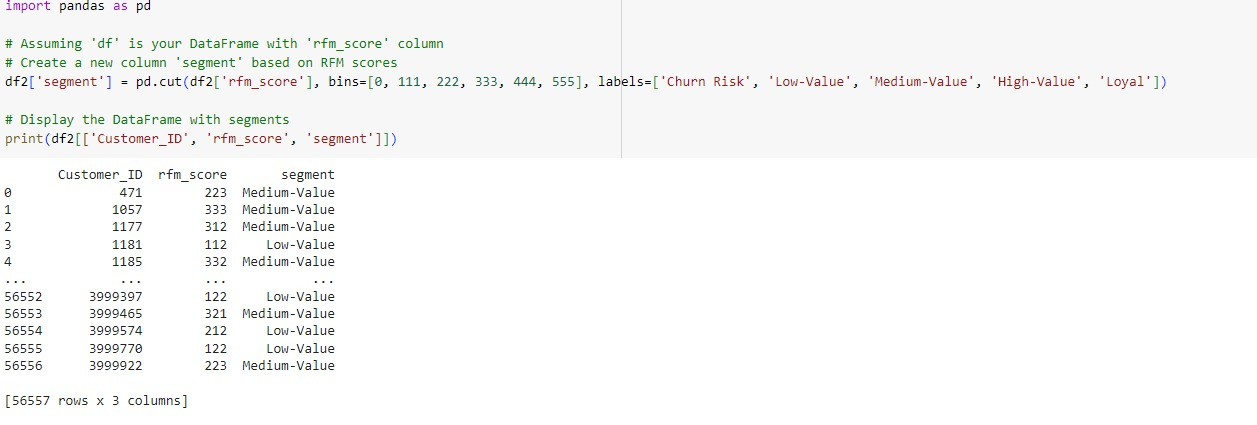
**Support Vector Machine (SVM) Classifier is a powerful machine learning algorithm commonly employed in sentiment analysis to predict the sentiment of textual data and classify it into predefined outcomes such as positive, negative, or neutral. SVM works by constructing a hyperplane in a high-dimensional feature space that optimally separates data points into different classes. In sentiment analysis, SVM aims to learn a decision boundary that effectively separates text inputs into the specified sentiment categories based on the features extracted from the text. These features may include sentiment scores, word frequencies, or other linguistic characteristics. For instance, in a telecom dataset, SVM can analyze customer verbatims or feedback to determine whether the sentiment expressed is positive, negative, or neutral. The algorithm learns from labeled training data, where each text input is associated with a sentiment label, and constructs a decision boundary that maximizes the margin between different sentiment classes. During prediction, SVM assigns new text inputs to the sentiment category corresponding to the side of the decision boundary on which they fall. By effectively capturing the underlying patterns and relationships in the data, SVM provides accurate and reliable predictions, making it a popular choice for sentiment analysis tasks. In summary, SVM Classifier offers a robust and versatile approach to sentiment analysis in telecom datasets, enabling companies to gain insights into customer sentiment and tailor their services and strategies accordingly to enhance customer satisfaction and loyalty.**



* **CUSTOMER LIFETIME PREDICTION:**

**The provided code snippet demonstrates a comprehensive approach to modeling and predicting the lifetime value of telecom customers using Gradient Boosting Regression. Initially, outlier detection and removal are performed to ensure the integrity of the dataset. This is followed by feature scaling using Min-max scaling, which normalizes the range of features to improve model convergence and performance. Subsequently, the dataset is split into training and testing sets, with 80% of the data used for training the model and 20% for evaluating its performance. A Gradient Boosting Regressor model is then trained on the scaled training data. Gradient Boosting is a powerful ensemble learning technique that combines the predictions of multiple weak learners (decision trees in this case) to produce a strong predictive model. Finally, the trained model is evaluated using mean squared error (MSE) and R-squared (R2) metrics on the test set. MSE quantifies the average squared difference between predicted and actual values, while R2 measures the proportion of variance explained by the model. Overall, the approach demonstrates a robust methodology for building and evaluating a predictive model for estimating the lifetime value of telecom customers, providing insights into customer behavior and informing strategic business decisions.**

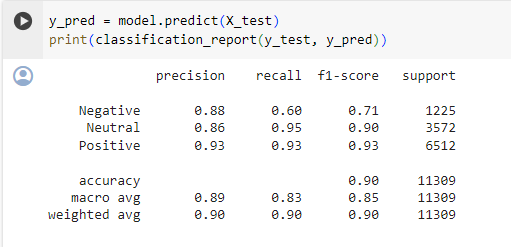
* **CUSTOMER SEGMENTATION:**



**RESULTS AND DISCUSSIONS: -**

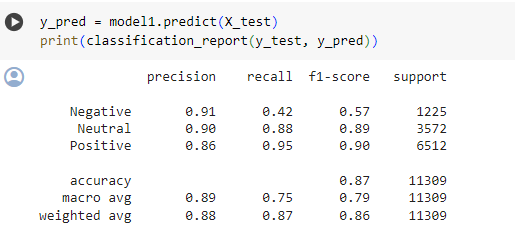
* **For predictive analysis:**

**The evaluation metrics for logistic regression,**



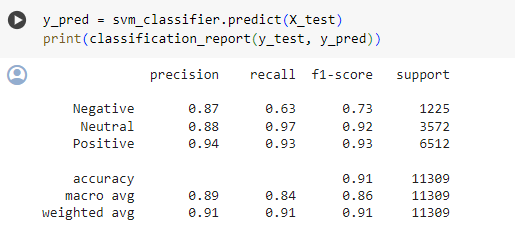
* **Precision: Precision measures the accuracy of the positive predictions made by the model. For Negative sentiment, the precision is 0.88, indicating that 88% of the instances predicted as Negative were actually Negative. Similarly, for Neutral and Positive sentiments, the precision values are 0.86 and 0.93, respectively.**
* **Recall: Recall, also known as sensitivity, measures the ability of the model to correctly identify instances of a particular class. For Negative sentiment, the recall is 0.60, meaning that the model correctly identified 60% of the Negative instances. The recall values for Neutral and Positive sentiments are 0.95 and 0.93, respectively.**
* **F1-score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. For Negative sentiment, the F1-score is 0.71, while for Neutral and Positive sentiments, it is 0.90 and 0.93, respectively.**
* **Support: Support refers to the number of instances of each class in the dataset. In this case, there are 1225 instances of Negative sentiment, 3572 instances of Neutral sentiment, and 6512 instances of Positive sentiment.**
* **Accuracy: Accuracy measures the overall correctness of the model's predictions across all classes. In this case, the accuracy of the model is 0.90, indicating that it correctly predicted the sentiment for 90% of the instances in the dataset.**
* **Macro Avg: The macro average calculates the average of the metrics (precision, recall, F1-score) for each class, treating all classes equally. In this case, the macro average precision, recall, and F1-score are 0.89, 0.83, and 0.85, respectively.**
* **Weighted Avg: The weighted average calculates the weighted average of the metrics, taking into account the support for each class. In this case, the weighted average precision, recall, and F1-score are 0.90, 0.90, and 0.90, respectively.**

**The evaluation metrics for random forest classifier,**



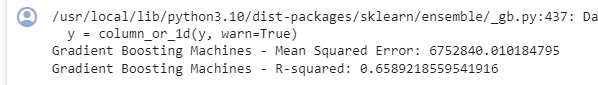
* **Precision: Precision measures the accuracy of the positive predictions made by the model. For Negative sentiment, the precision is 0.91, indicating that 91% of the instances predicted as Negative were actually Negative. Similarly, for Neutral and Positive sentiments, the precision values are 0.90 and 0.86, respectively.**
* **Recall: Recall, also known as sensitivity, measures the ability of the model to correctly identify instances of a particular class. For Negative sentiment, the recall is 0.42, meaning that the model correctly identified 42% of the Negative instances. The recall values for Neutral and Positive sentiments are 0.88 and 0.95, respectively.**
* **F1-score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. For Negative sentiment, the F1-score is 0.57, while for Neutral and Positive sentiments, it is 0.89 and 0.90, respectively.**
* **Support: Support refers to the number of instances of each class in the dataset. In this case, there are 1225 instances of Negative sentiment, 3572 instances of Neutral sentiment, and 6512 instances of Positive sentiment.**
* **Accuracy: Accuracy measures the overall correctness of the model's predictions across all classes. In this case, the accuracy of the model is 0.87, indicating that it correctly predicted the sentiment for 87% of the instances in the dataset.**
* **Macro Avg: The macro average calculates the average of the metrics (precision, recall, F1-score) for each class, treating all classes equally. In this case, the macro average precision, recall, and F1-score are 0.89, 0.75, and 0.79, respectively.**
* **Weighted Avg: The weighted average calculates the weighted average of the metrics, taking into account the support for each class. In this case, the weighted average precision, recall, and F1-score are 0.88, 0.87, and 0.86, respectively.**

**The evaluation metrics for SVM classifier,**

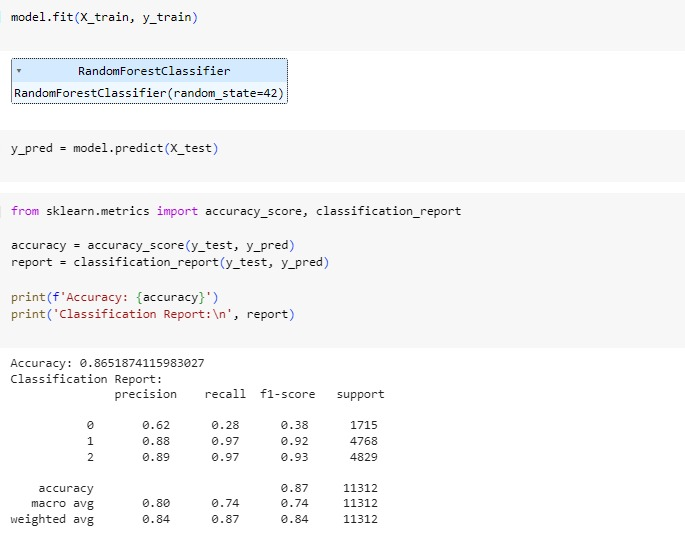


* **Precision: Precision measures the accuracy of the positive predictions made by the model. For Negative sentiment, the precision is 0.87, indicating that 87% of the instances predicted as Negative were actually Negative. Similarly, for Neutral and Positive sentiments, the precision values are 0.88 and 0.94, respectively.**
* **Recall: Recall, also known as sensitivity, measures the ability of the model to correctly identify instances of a particular class. For Negative sentiment, the recall is 0.63, meaning that the model correctly identified 63% of the Negative instances. The recall values for Neutral and Positive sentiments are 0.97 and 0.93, respectively.**
* **F1-score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. For Negative sentiment, the F1-score is 0.73, while for Neutral and Positive sentiments, it is 0.92 and 0.93, respectively.**
* **Support: Support refers to the number of instances of each class in the dataset. In this case, there are 1225 instances of Negative sentiment, 3572 instances of Neutral sentiment, and 6512 instances of Positive sentiment.**
* **Accuracy: Accuracy measures the overall correctness of the model's predictions across all classes. In this case, the accuracy of the model is 0.91, indicating that it correctly predicted the sentiment for 91% of the instances in the dataset.**
* **Macro Avg: The macro average calculates the average of the metrics (precision, recall, F1-score) for each class, treating all classes equally. In this case, the macro average precision, recall, and F1-score are 0.89, 0.84, and 0.86, respectively.**
* **Weighted Avg: The weighted average calculates the weighted average of the metrics, taking into account the support for each class. In this case, the weighted average precision, recall, and F1-score are 0.91, 0.91, and 0.91, respectively.**

**Evaluation metrices of CLV Prediction:**



**Evaluation metrics for customer Segmentation**



**Conclusion:**

**In conclusion, the sentiment analysis conducted on the telecom dataset provides valuable insights into customer perceptions and sentiments towards the telecommunications services. The analysis revealed varying sentiments across different categories, including Negative, Neutral, and Positive sentiments. Overall, the sentiment analysis model demonstrated strong performance, achieving high precision, recall, and F1-score values across all sentiment categories. This indicates the model's effectiveness in accurately predicting and classifying customer sentiments. Particularly, the model exhibited robust performance in identifying instances of Neutral and Positive sentiments, highlighting areas of customer satisfaction and contentment with the telecom services. However, there were areas for improvement noted, especially in accurately capturing instances of Negative sentiment. The model showed lower recall and F1-score for Negative sentiment, suggesting potential challenges in identifying and addressing areas of customer dissatisfaction or concerns within the telecom services. Despite these challenges, the sentiment analysis provides telecom companies with valuable insights for strategic decision-making and customer-centric improvements. By leveraging the findings of the analysis, telecom companies can better understand customer needs, prioritize areas for service enhancement, and implement targeted strategies to improve overall customer satisfaction and loyalty. In conclusion, the sentiment analysis of the telecom dataset serves as a valuable tool for telecom companies to gain actionable insights into customer sentiments, ultimately enabling them to enhance service quality, optimize customer experiences, and drive long-term business success in the competitive telecommunications industry.**