TABLE OF CONTENT

INTRODUCTION

The telecommunications industry is continuously evolving, driven by technological advancements and the increasing demands of consumers for high-quality voice services. Customer satisfaction and experience are paramount, as they significantly impact customer retention and brand loyalty. In this context, understanding the sentiments expressed by customers regarding voice quality is crucial for telecom companies aiming to enhance their service offerings.

This project focuses on sentiment analysis of voice quality data, employing advanced techniques to analyze sentiments conveyed through voice recordings. By leveraging the R programming language, the project seeks to uncover insights into customer experiences and preferences, thereby providing telecom operators with actionable data to improve service quality and customer satisfaction.

The scope of the project includes the following key components:

1. Data Collection and Pre-processing: Utilizing a voice quality dataset sourced from December 2022, the project begins with data preparation, including handling missing values and encoding categorical variables to ensure the dataset is ready for analysis.

2. Exploratory Data Analysis (EDA): Conducting EDA to uncover patterns and trends within the dataset, such as the distribution of sentiments across different operators and call scenarios. This step provides a comprehensive understanding of the data and identifies potential areas of interest.

3. Sentiment Classification: Building and evaluating a random forest model to classify sentiments as positive or negative. The performance of the model is assessed to ensure its accuracy and reliability.

4. Operator-Specific Analysis: Analyzing customer satisfaction levels for individual telecom operators, with a focus on identifying the operators with the highest and lowest satisfaction ratings.

5. Insights and Recommendations: Providing valuable insights and recommendations for telecom companies to enhance their customer service strategies, improve voice quality, and tailor marketing efforts based on the sentiment analysis findings.

The project aims to demonstrate the effectiveness of sentiment analysis techniques in analyzing voice quality data, offering a deeper understanding of customer sentiments and paving the way for future research and innovation in this domain. By integrating these insights, telecom companies can better align their services with customer expectations, ultimately driving higher satisfaction and loyalty.

ABOUT THE INDUSTRY

The telecom industry, particularly the mobile operators segment, represents a dynamic and integral part of global communications infrastructure. Mobile operators provide voice, data, and internet services to billions of users worldwide, enabling seamless connectivity and driving digital transformation across economies.

The industry is characterized by intense competition among operators vying for market share and customer loyalty. Major players such as Verizon, AT&T, Vodafone, China Mobile, and T-Mobile operate extensive networks that span vast geographic areas, delivering services ranging from traditional voice calls to high-speed mobile internet access.

Technological innovation plays a pivotal role in shaping the industry's evolution. The transition from 2G to 3G and subsequently to 4G LTE networks has significantly enhanced data speeds and network capacity, supporting a myriad of applications from video streaming to IoT devices. The ongoing

deployment of 5G networks promises even faster speeds, lower latency, and the ability to support emerging technologies like autonomous vehicles and augmented reality.

Customer experience and satisfaction are paramount in this competitive landscape. Operators invest heavily in customer service initiatives, network reliability, and coverage expansion to meet growing expectations for connectivity anytime, anywhere. Bundled service offerings, including mobile plans with added benefits such as content streaming or international roaming, further differentiate operators in the market.

Regulatory frameworks vary across regions but universally aim to ensure fair competition, protect consumer rights, and manage spectrum allocation. Governments play a crucial role in licensing spectrum, setting quality standards, and promoting universal access to telecom services, particularly in rural and underserved areas.

In recent years, the telecom industry has seen strategic mergers and acquisitions aimed at consolidating market position and leveraging economies of scale. These moves not only strengthen operational efficiencies but also enable operators to invest in next-generation technologies and infrastructure upgrades.

Looking ahead, the telecom industry faces opportunities and challenges alike. The proliferation of connected devices, the rise of digital services, and the advent of 5G are expected to drive continued growth. However, operators must navigate evolving cyber security threats, regulatory complexities, and the demand for sustainable practices in network deployment and operations.

In the Indian mobile telecommunications market, several major operators have emerged as key players, each contributing significantly to the industry's growth and competition. Bharti Airtel, founded in 1995 by Sunil Bharti Mittal, is one of the largest operators with a nationwide presence. Initially starting with mobile services, Airtel expanded its offerings to include broadband internet, digital TV, and enterprise services. Known for its robust network infrastructure and extensive coverage, Airtel has been at the forefront of technological advancements, including the deployment of 4G LTE and preparations for 5G networks.

Reliance Jio Infocomm, a subsidiary of Reliance Industries Limited, made a monumental impact on the Indian telecom landscape since its launch in 2016. Leveraging its extensive fiber optic network and disruptive pricing strategies, Jio rapidly gained market share by offering free voice calls and low-cost data plans. This approach democratized access to high-speed internet across urban

and rural areas, significantly increasing internet penetration and transforming digital consumption habits.

Vodafone Idea Limited, formed by the merger of Vodafone India and Idea Cellular in 2018, is another prominent player in the market. The merger aimed to create a stronger entity capable of competing with Reliance Jio and Airtel. Vodafone Idea operates under the brands Vodafone and Idea, providing mobile and internet services to millions of subscribers. The company continues to expand its network and enhance service offerings amidst competitive pressures and financial challenges.

BSNL (Bharat Sanchar Nigam Limited) and MTNL (Mahanagar Telephone Nigam Limited) are state-owned operators that have historically played a crucial role in providing telecom services, particularly in rural and remote areas. Despite facing operational and financial challenges, BSNL and MTNL remain significant players in the market, contributing to the government's efforts to promote universal access to telecom services.

Competition in the Indian mobile operator market is intense and dynamic, characterized by a diverse array of players striving for market share and consumer loyalty. At the forefront are major telecom giants like Bharti Airtel, Reliance Jio Infocomm, and Vodafone Idea Limited, each leveraging distinct strategies to attract and retain subscribers.

Bharti Airtel, one of the oldest players in the market, emphasizes network quality, extensive coverage, and a wide range of service offerings including mobile, broadband, and digital TV. Airtel's strategy focuses on customer-centric innovations, partnerships for content delivery, and expansion into enterprise solutions to maintain its competitive edge.

Reliance Jio disrupted the market upon its entry in 2016 with aggressive pricing strategies that revolutionized data consumption habits across India. Offering free voice calls and low-cost data plans, Jio rapidly amassed a large subscriber base by prioritizing affordability and high-speed internet access through its nationwide 4G network. Jio continues to innovate with investments in fiber optic infrastructure, IoT solutions, and preparations for 5G rollout, aiming to further solidify its market position.

Vodafone Idea Limited, formed from the merger of Vodafone India and Idea Cellular, competes by combining the strengths of both entities' networks and customer bases. The company focuses on network integration, improving service quality, and expanding its digital service offerings to enhance customer experience and compete effectively with rivals.

Regional players and smaller operators also contribute to the competitive landscape by catering to specific market segments or geographic areas. These operators often differentiate themselves through localized marketing strategies, competitive pricing plans, and niche service offerings tailored to regional preferences and needs.

These major operators, along with newer entrants and regional players, continue to shape the competitive landscape of India's mobile telecommunications market. Their strategies, innovations and investments in network infrastructure are pivotal in meeting the growing demand for connectivity, digital services, and seamless customer experiences across the diverse and rapidly evolving Indian market.

HISTORY OF THE INDUSTRY

The telecommunications sector has undergone profound transformations since its inception in the 19th century. The journey began with the invention of the telegraph by Samuel Morse in 1837, revolutionizing long-distance communication by transmitting coded messages over wires. This was followed by Alexander Graham Bell's invention of the telephone in 1876, which allowed voice communication over distances and laid the groundwork for modern telephony. The early 20th century saw the establishment of major telecommunications companies and the expansion of telephone networks across countries.

The mid-20th century marked the advent of radio and television broadcasting, introducing wireless communication to the masses. The 1960s and 1970s brought significant advancements with the development of satellite communications and the establishment of international communication networks. The deregulation and privatization trends of the 1980s and 1990s led to increased competition and innovation within the sector, exemplified by the breakup of AT&T in the United States.

The late 20th and early 21st centuries have been defined by the rise of the internet and mobile communications. The introduction of the World Wide Web in 1991 and the proliferation of personal computers transformed the telecommunications landscape, enabling the rapid exchange of information and the rise of digital communications. The development of mobile technology, from the first generation (1G) analog systems in the 1980s to the current rollout

of fifth generation (5G) networks, has revolutionized how people communicate, offering faster speeds and more reliable connections.

Throughout its history, the telecommunications sector has been characterized by continuous innovation and adaptation, driven by technological advancements and changing consumer needs. Today, it remains a critical component of global infrastructure, connecting people and businesses worldwide and facilitating the digital economy.

HISTORY OF TELECOM INDUSTRY IN INDIAN MARKET

The history of the telecom sector, particularly mobile operators, in India is marked by significant milestones and transformative developments. The sector's journey began with the establishment of the Department of Telecommunications (DoT) in 1985, which oversaw state-owned entities like MTNL and BSNL, providing basic landline services across the country. The entry of private players in the late 1990s, following the National Telecom Policy of 1994 aimed at liberalizing the industry, ushered in a new era of competition and expansion.

In 1995, the first private mobile operator, Modi Telstra (later renamed BPL Mobile), launched its services in Mumbai, introducing mobile telephony to Indian consumers. This period saw the rapid growth of mobile subscribership, fuelled by increasing affordability of handsets and tariff plans. The introduction of GSM technology by operators like Bharti Airtel, Vodafone (formerly Hutch), and Idea Cellular (now merged with Vodafone) further accelerated the adoption of mobile services across urban and rural India.

The early 2000s witnessed exponential growth in mobile penetration, driven by innovative pricing strategies and network expansion efforts by both incumbents and new entrants. The launch of 3G services in 2010 and subsequent rollout of 4G LTE networks by Reliance Jio Infocomm in 2016 revolutionized the telecom landscape, offering high-speed internet access at affordable rates and triggering a digital revolution.

The sector has also navigated regulatory challenges, spectrum auctions, and policy reforms aimed at fostering competition, ensuring consumer rights, and promoting universal access to telecom services. The establishment of the Telecom Regulatory Authority of India (TRAI) in 1997 played a crucial role in shaping industry regulations and promoting fair competition among operators.

Today, India's telecom sector is one of the largest and fastest-growing in the world, with a robust ecosystem of operators providing voice, data, and digital services to over a billion mobile subscribers. The advent of 5G technology is poised to further transform the sector, offering enhanced connectivity, supporting IoT applications, and driving economic growth.

The history of the telecom sector in India reflects its evolution from state-owned monopolies to a competitive market driven by technological innovation and consumer demand. As mobile operators continue to innovate and expand their offerings, they play a pivotal role in connecting India's diverse population and contributing to the country's digital economy ambitions.

CURRENT SITUATION IN MARKET

The current situation in the Indian mobile sector is characterized by significant developments, challenges, and opportunities amidst a rapidly evolving landscape. With a subscriber base exceeding a billion users, India remains one of the largest and fastest-growing telecom markets globally, driven by increasing internet penetration and digital adoption across urban and rural areas.

At the forefront of recent developments is the ongoing deployment and adoption of 4G LTE technology by major operators like Bharti Airtel, Reliance Jio, and Vodafone Idea. This transition has democratized access to high-speed internet services, enabling a wide range of digital applications from streaming video and online gaming to e-commerce and digital payments. Operators continue to invest in expanding their 4G networks and preparing for the rollout of 5G technology, which promises even faster speeds, lower latency, and support for emerging technologies such as IoT and smart cities.

The Government of India monitors mobile operators through a combination of regulatory frameworks, policies, and oversight mechanisms aimed at ensuring fair competition, consumer protection, and adherence to industry standards. The primary regulatory body overseeing the telecom sector is the Telecom Regulatory Authority of India (TRAI). TRAI plays a crucial role in formulating regulations, issuing licenses, and monitoring the performance of mobile operators to safeguard the interests of consumers and promote industry growth.

One of the key ways in which mobile operators are monitored is through spectrum management. Spectrum, the radio frequencies used for wireless communication, is a scarce and valuable resource allocated by the government

through auctions and licensing. TRAI oversees spectrum allocation, pricing, and utilization to ensure efficient use and fair distribution among operators. Spectrum audits and periodic assessments help monitor compliance with usage conditions and spectrum caps set by the government.

Tariff regulation is another area of focus. TRAI sets guidelines and monitors tariffs charged by mobile operators to prevent anti-competitive practices, ensure transparency, and protect consumer interests. Operators are required to adhere to prescribed tariff ceilings and avoid predatory pricing that could harm market dynamics.

Quality of service (QoS) standards are critical in monitoring the performance of mobile operators. TRAI defines QoS benchmarks for parameters such as call drop rates, network coverage, data speeds, and customer complaint resolution. Operators are obligated to regularly report QoS data, which TRAI analyzes to assess compliance and initiate corrective measures if standards are not met. Public disclosures of QoS reports enable consumers to make informed choices and hold operators accountable for service delivery.

Consumer protection is another priority area. TRAI mandates operators to adhere to norms related to billing transparency, customer grievance redressal, and data privacy. Operators are required to provide clear information on tariffs, terms of service, and fair usage policies to consumers. Complaints related to billing disputes, service disruptions, or violations of consumer rights are addressed through TRAI's grievance handling mechanisms.

The Department of Telecommunications (DoT) oversees licensing, spectrum management, and policy formulation for the telecom sector. The Telecom Commission, chaired by the Secretary of DoT, provides strategic direction and policy recommendations to promote industry growth and innovation.

However, the sector faces several challenges, including intense competition, regulatory pressures, and financial sustainability concerns. The entry of Reliance Jio in 2016 disrupted the market with its disruptive pricing strategies, leading to a wave of consolidation among operators and impacting profitability across the industry. Operators are grappling with high debt levels, spectrum costs, and the need for continuous investment in network infrastructure to meet growing data demand and maintain service quality.

Regulatory developments also play a crucial role in shaping the industry's trajectory. The Telecom Regulatory Authority of India (TRAI) oversees policies related to spectrum allocation, tariffs, and consumer protection, aiming to foster fair competition and ensure quality of service. Recent regulatory initiatives

include spectrum auctions, the review of interconnect usage charges (IUC), and efforts to promote digital inclusion through initiatives like BharatNet.

Amidst these challenges, the mobile sector in India presents significant opportunities for growth and innovation. Rising smartphone penetration, increasing demand for digital services, and the government's Digital India initiative are driving investments in technology and infrastructure. Operators are exploring new revenue streams through partnerships in content delivery, fintech services, and enterprise solutions, aiming to diversify their offerings and enhance customer engagement.

GROWTH IN TELECOM SECTOR

The telecom market in India, particularly the mobile operator segment, has experienced remarkable growth over the past two decades, fueled by a combination of factors including technological advancements, regulatory reforms, and increasing consumer demand. Since the liberalization of the sector in the late 1990s, India has witnessed exponential growth in mobile subscriptions, network expansion, and digital connectivity.

Key drivers of growth include the widespread adoption of mobile phones, especially smart-phones, which have become more affordable and accessible to a larger segment of the population. This has led to a surge in mobile internet usage, enabling millions of Indians to access digital services such as e-commerce, social media, and online entertainment.

The entry of Reliance Jio Infocomm in 2016 disrupted the market with its disruptive pricing strategies, offering free voice calls and low-cost data plans that spurred unprecedented data consumption across the country. Jio's entry not only expanded the subscriber base but also accelerated the deployment of 4G LTE networks by existing operators, enhancing network capabilities and driving competition.

Government initiatives such as Digital India, aimed at bridging the digital divide and promoting digital inclusion, have further fueled growth in the telecom sector. Infrastructure development projects like BharatNet, which aims to provide broadband connectivity to rural areas, are crucial in extending telecom services to underserved regions and connecting remote communities.

The rollout of 4G LTE networks by major operators like Bharti Airtel, Vodafone Idea, and Reliance Jio has significantly improved internet speeds and reliability, transforming how individuals and businesses communicate, access information, and conduct transactions. The ongoing preparations for 5G technology promise to further revolutionize the sector by offering ultra-fast speeds, low latency, and supporting advanced applications such as IoT and smart cities.

Despite challenges such as regulatory uncertainties, spectrum pricing, and financial pressures faced by operators, the Indian telecom market continues to exhibit resilience and potential for growth. With a young and tech-savvy population driving demand for digital services, coupled with ongoing investments in infrastructure and technology, the telecom sector is poised to play a pivotal role in India's economic development and digital transformation in the years to come.

HOW DATA DRIVEN DECISION MAKING AND SENTIMENT ANALYSIS CAN BE HELPFUL IN CURRENT IN INDIAN TELECOM SECTOR ?

Data-driven decision-making and sentiment analysis can greatly benefit the Indian telecom sector by providing actionable insights into consumer behavior, service quality, and market dynamics. In an increasingly competitive environment, telecom operators can leverage data analytics to optimize operational efficiency, enhance customer experience, and drive strategic growth initiatives.

Firstly, data-driven decision-making enables telecom companies to harness vast amounts of customer data generated from mobile usage patterns, subscriber demographics, and service interactions. By analyzing this data using advanced analytics techniques such as machine learning and predictive modeling, operators can identify trends, anticipate customer needs, and personalize marketing campaigns. For instance, understanding peak usage times or popular service bundles can help operators optimize network capacity and resource allocation, ensuring a seamless customer experience during high-demand periods.

Secondly, sentiment analysis plays a crucial role in gauging customer satisfaction and sentiment towards telecom services. By analyzing customer feedback from social media, call centre interactions, and online reviews, operators can gain real-time insights into customer perceptions, identify areas of dissatisfaction, and promptly address issues. Sentiment analysis also helps operators track brand sentiment, monitor competitor performance, and adjust marketing strategies accordingly to maintain competitive advantage.

Moreover, sentiment analysis can aid in proactive customer retention efforts by detecting early signs of dissatisfaction or churn risk. By identifying key drivers of customer dissatisfaction, such as network quality issues or billing discrepancies, operators can take pre-emptive measures to resolve problems and enhance service delivery, thereby improving customer retention rates and reducing churn.

In conclusion, data-driven decision-making and sentiment analysis are invaluable tools for the Indian telecom sector in navigating complexities, optimizing operations, and delivering superior customer experiences. By leveraging actionable insights derived from data analytics, telecom operators can drive innovation, foster customer loyalty, and maintain a competitive edge in a rapidly evolving market landscape characterized by technological advancements and changing consumer expectations.

RESEARCH

TITLE: ANALYSIS VOICE QUALITY SATISFACTION IN TELECOMMUNICATION SERVICES

In today's hyper-connected world, telecommunications services play a crucial role in keeping individuals and businesses connected. Voice quality is a key aspect of the user experience when it comes to telecommunications, influencing customer satisfaction and loyalty. Understanding the factors that contribute to voice quality satisfaction is essential for telecom companies to maintain a competitive edge in the market.

In this analysis, we delve into a dataset containing information about voice quality ratings provided by users of various telecom operators. We aim to explore the factors influencing voice quality satisfaction, identify patterns and trends, and provide insights to telecom companies for improving their services.

The dataset encompasses a range of variables, including the telecom operator, the user's location and travel status during the call, the call rating provided by the user, and the perceived call drop category.

Our goal is to provide telecom companies with actionable insights to enhance voice quality satisfaction and ultimately improve the overall user experience.

The aim of the project was to perform sentiment analysis on voice quality data collected from telecom customers. This involves analysing customer feedback to determine whether their experience with call quality was satisfactory or not. Additionally, the project aims to explore various factors such as operator, location, and call type (indoor, outdoor, travelling) to understand their impact on customer satisfaction which provide telecom companies with actionable insights to enhance voice quality satisfaction and ultimately improve the overall user experience.

SENTIMENT ANALYSIS

Sentiment analysis, also known as opinion mining, is a burgeoning field within natural language processing that aims to extract subjective information from

text data. In recent years, sentiment analysis has expanded beyond textual data to include other modalities, such as voice data, allowing for a more comprehensive understanding of sentiment in various contexts.

Voice quality data, in particular, presents a unique opportunity for sentiment analysis. By analyzing acoustic features of speech, such as pitch, tone, and intensity, researchers can infer the emotional state or sentiment of the speaker. This application of sentiment analysis is especially relevant in industries such as telecommunications, where customer satisfaction and experience are paramount.

One common approach to sentiment analysis of voice data involves pre-processing the raw audio signals to extract relevant features, such as prosodic features (e.g., pitch, duration, intensity) and spectral features (e.g., formants, energy distribution). These features are then used to train machine learning models, such as random forests, support vector machines, or deep neural networks, to predict sentiment labels (e.g., positive, negative, neutral).

Previous studies have demonstrated the effectiveness of sentiment analysis in voice data across various domains, including call centre interactions, customer service evaluations, and social media sentiment analysis. For example, research by Smith et al. applied sentiment analysis to call centre recordings to identify customer dissatisfaction and improve service quality.

Moreover, studies have explored the impact of contextual factors, such as network type, geographical location, and operator, on voice quality and sentiment. For instance, research by Zhang et al. investigated the influence of network congestion on voice quality and customer satisfaction in mobile communication networks.

However, despite the advancements in sentiment analysis of voice data, several challenges persist. These include the need for robust feature extraction techniques, handling of noisy audio signals, scalability of machine learning algorithms to large datasets, and interpretability of sentiment analysis models.

Sentiment analysis of voice quality data offers valuable insights into customer satisfaction, service quality, and user experience in telecommunications and related industries. Future research in this area should focus on addressing existing challenges and exploring novel techniques to enhance the accuracy and applicability of sentiment analysis models in real-world settings.

SPECIFICATION

Operating System- Windows 10

RAM- 16 GB

Language - R programming

Application- RStudio[2023.06.1 Build 524], R CRAN- 4.2.3

R LANGUAGE

R is a programming language and software environment commonly used for statistical computing and graphics. It provides a wide variety of statistical and graphical techniques, including linear and nonlinear modeling, time-series analysis, classification, clustering, and more. Developed initially by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, in the early 1990s, R has since become one of the most popular programming languages for statistical analysis and data science.

KEY FEATURES OF R:

1. **Open Source**: R is open-source software, which means that it is freely available to anyone to use, modify, and distribute. This has contributed to its widespread adoption and active community of users and developers.

2. **Extensive Package Ecosystem**: R has a vast ecosystem of packages contributed by users and developers around the world. These packages extend R's functionality to cover a wide range of statistical techniques, data visualization methods, machine learning algorithms, and more.

3. **Statistical Capabilities**: R provides a comprehensive set of tools for statistical analysis, including built-in functions for data manipulation, descriptive statistics, hypothesis testing, regression analysis, and more.

4. **Data Visualization**: R offers powerful tools for creating high-quality data visualizations, including traditional plots like scatterplots, histograms, and boxplots, as well as more advanced graphics such as heatmaps, interactive visualizations, and 3D plots.

5. **Integration**: R can seamlessly integrate with other programming languages and software tools. For example, it can interface with databases, web APIs, and other data sources, making it a versatile tool for data analysis and manipulation.

6. **Reproducibility**: R promotes reproducible research by allowing users to write scripts and documents that combine code, data, and narrative text. This makes it easier to share and reproduce analyses and results.

ADVANATGE OF R LANGUAGE

1. **Open Source and Free:** R is an open-source programming language, which means it is freely available for anyone to use, modify, and distribute. This makes it accessible to a wide community of users and fosters collaborative development of packages and libraries.

2. **Extensive Libraries:** R boasts a vast ecosystem of packages and libraries contributed by developers worldwide. These packages cover various domains such as data manipulation (e.g., dplyr, tidyr), statistical analysis (e.g., stats, lme4), machine learning (e.g., caret, randomForest), and visualization (e.g., ggplot2, plotly). The availability of these packages makes it easy to perform complex tasks with minimal coding effort.

3. **Statistical Capabilities:** R was initially developed by statisticians for statistical analysis and data visualization. It provides comprehensive tools for descriptive statistics, inferential statistics, regression analysis, time-series analysis, and more. The built-in statistical functions and libraries make R a powerful tool for researchers and analysts working with data.

4. **Graphics and Data Visualization:** R offers exceptional capabilities for creating high-quality graphics and data visualizations. Packages like ggplot2 provide a flexible and declarative syntax for producing publication-quality plots with customizable aesthetics. This makes it easier to explore data visually and communicate findings effectively.

5. **Community Support:** R has a large and active community of users, developers, and researchers who contribute to forums, online tutorials, and resources. This community support facilitates learning, troubleshooting, and sharing best practices in data analysis and statistical modeling.

6. **Integration with Other Languages and Tools:** R can be easily integrated with other languages and tools, enhancing its versatility and usability in various workflows. For instance, R interfaces with databases (e.g., MySQL, PostgreSQL), integrates with big data frameworks (e.g., Hadoop, Spark), and supports interoperability with languages like Python and C++ through packages like reticulate and Rcpp.

7. **Reproducibility and Documentation:** R promotes reproducible research practices through literate programming tools such as R Markdown and Sweave. These tools allow analysts to embed code, text, and visualizations in a single document, facilitating transparent and reproducible reporting of analysis results.

8. **Cross-Platform Compatibility:** R runs on all major operating systems (Windows, macOS, Linux), ensuring compatibility and consistency across different environments. This flexibility makes R suitable for both individual users and large-scale enterprise applications.

DATASET

A dataset is a structured collection of data typically organized in rows and columns. It represents a single aspect or multiple aspects of a real-world phenomenon, such as observations, measurements, or records. Datasets can come from various sources, including experiments, surveys, observations, simulations, or data mining processes.

ATTRIBUTES OF THE DATASET:

1. **Operator**: The telecom operator from which the call data originated.
2. **Inout_travelling**: Whether the call was made indoors, outdoors, or while travelling.
3. **Rating**: A numerical rating given by users, likely related to call quality.
4. **Calldrop_category**: The category of the call quality, such as "Poor Voice Quality," "Satisfactory," or "Call Dropped."
5. **Latitude**: The latitude coordinate of the call location.
6. **Longitude**: The longitude coordinate of the call location.
7. **State_name**: The name of the state where the call originated.
8. **Network_type**: The type of network used during the call (e.g., 2G, 3G, 4G).

DATA ANALYSIS & INTERPRETATION

CODE

```
cq<- read.csv("C:\\Users\\tgeor\\Downloads\\December_MyCall_2022.csv")
```

operator	inout_trav	network_t	rating	calldrop_c	latitude	longitude	state_name
Airtel	Indoor	4G	1	Poor Voice	13.75721	79.63888	Andhra Pradesh
RJio	Outdoor	4G	4	Satisfactor	-1	-1	NA
RJio	Outdoor	4G	4	Satisfactor	-1	-1	NA
BSNL	Travelling	Unknown	5	Satisfactor	-1	-1	NA
Airtel	Indoor	4G	5	Satisfactor	28.64949	77.28121	Delhi
VI	Travelling	4G	5	Satisfactor	19.07039	72.99756	Maharashtra
RJio	Outdoor	4G	4	Satisfactor	-1	-1	NA
Airtel	Travelling	4G	5	Satisfactor	28.64949	77.28121	Delhi
RJio	Indoor	4G	1	Poor Voice	-1	-1	NA
Airtel	Indoor	Unknown	1	Call Dropp	19.22981	72.84179	Maharashtra
Airtel	Indoor	2G	5	Satisfactor	12.93587	77.69458	Karnataka
Airtel	Indoor	4G	5	Satisfactor	12.93445	77.69691	Karnataka
Airtel	Travelling	Unknown	5	Satisfactor	13.01078	77.66231	Karnataka
RJio	Indoor	4G	1	Call Dropp	-1	-1	NA
Airtel	Indoor	4G	5	Satisfactor	13.04124	77.61961	Karnataka
Airtel	Indoor	4G	3	Satisfactor	30.11539	78.28191	Uttarakhand
Airtel	Outdoor	4G	3	Satisfactor	30.11539	78.28191	Uttarakhand
Airtel	Indoor	Unknown	3	Satisfactor	30.11539	78.28191	Uttarakhand
Airtel	Outdoor	Unknown	3	Satisfactor	30.11539	78.28191	Uttarakhand
Airtel	Outdoor	Unknown	3	Satisfactor	30.11539	78.28191	Uttarakhand
Airtel	Indoor	4G	3	Satisfactor	30.11539	78.28191	Uttarakhand
RJio	Indoor	4G	4	Satisfactor	-1	-1	NA
RJio	Indoor	4G	4	Satisfactor	-1	-1	NA
VI	Outdoor	4G	3	Satisfactor	12.90918	80.09611	Tamil Nadu

PACKAGES

Packages are collections of functions, data sets, and documentation that extend the capabilities of the base R system. They are created and maintained by members of the R community and cover a wide range of topics, including data manipulation, statistical analysis, machine learning, visualization, and more

List of packages used –

- **RandomForest**:
  This library provides an implementation of the random forest algorithm,which is an ensemble learning method used for classification and regression tasks

- **dplyr**:
  The dplyr library offers a set of functions that provide a concise and intuitive way to perform data manipulation tasks, such as filtering, selecting columns, transforming variables, aggregating data, and joining datasets

- **tidyverse**:
  The tidyverse is not a single library but a collection of R packages, including dplyr, ggplot2, and others, that share a common philosophy and syntax for data manipulation and visualization.

- **ggplot2**:
  This library is a powerful and flexible data visualisation package. It follows the grammar of graphics concept, allowing users to create a wide range of high-quality plots and visualizations

- **lattice**:
  These plots are particularly useful for visualizing multivariate data and exploring relationships between variables. lattice offers a flexible and customizable approach to creating conditioned plots and provides a high-level interface for visually appealing and informative graphics.

- **caret**:
  It provides a unified interface and tools for data preprocessing, feature selection, model training, model evaluation, and model tuning. caret simplifies the machine learning workflow and allows for efficient and reproducible model development.

Creating a new dataframe without the network_type column, as it had an "Unknown" network, typically involves removing that specific column from the original dataframe. This is often done when the "Unknown" values are either irrelevant for analysis or need to be handled separately

CODE

```
# Assuming your original dataframe is named 'cq'

new_cq1 <- subset(cq, select = -network_type) # Create a new dataframe without the 'network_type' column

new_cq1$network_type <- cq$network_type # Copy the 'network_type' column

from the original dataframe
```

EXPLAINATION

- subset() is a function in R used to extract subsets of data from data frames. It allows you to specify conditions for including or excluding rows or columns from the dataset.

- In this case, subset() is used to create a new dataframe named new_cq1 from the original dataframe cq, excluding the network_type column.

- The argument select = -network_type specifies that the network_type column should be excluded from the subset. The - sign indicates exclusion.
- As a result, new_cq1 contains all the columns from the original dataframe cq except for the network_type column

CODE

```
# Replace "Unknown" with NA in the new dataframe
new_cq1$network_type[new_cq1$network_type == "Unknown"] <- NA
head(new_cq1,10)
## operator inout_travelling rating calldrop_category latitude longitude
# Count the number of NA values
num_na <- sum(is.na(new_cq1))
# Print the result
print(num_na)
## [1] 971
```

EXPLANATION

The code replaces all occurrences of the string "Unknown" in the network_type column of the dataframe new_cq1 with NA (representing missing values). After replacing the values, the code prints the first 10 rows of the dataframe new_cq1 using the head() function. This function is commonly used to display the initial rows of a dataframe. The code then calculates the total number of NA values present in the entire dataframe new_cq1. This is done using the sum() function in combination with is.na() function, which returns a logical vector indicating whether each element in the dataframe is NA or not. Finally, the code prints the total count of NA values in the dataframe new_cq1 using the print() function which is 971.

CODE

```
# Calculate the mode of the network_type column
```

```
network_type_mode <- as.character(which.max(table(new_cq1$network_type)))
print(network_type_mode)
## [1] "3"
# Replace NAs with the mode value
new_cq1$network_type <- ifelse(is.na(new_cq1$network_type),
network_type_mode, new_cq1$network_type)
#Replace missing values in "network_type" with the mode
new_cq1$network_type <- ifelse(is.na(new_cq1$network_type),
mode(new_cq1$network_type, na.rm = TRUE), new_cq1$network_type)
# Replace '3' with '3G' in the 'network_type' column
new_cq1$network_type <- sub("3", "3G", new_cq1$network_type)
```

EXPLANATION

Here we calculate the mode (most frequently occurring value) of the network_type column using the table() function and which.max() function as the network_type column contains categorical data representing the type of network (e.g., 2G, 3G, 4G). Since the data is categorical, it makes more sense to find the mode (most frequently occurring value) rather than calculating the median or mean, which are more suited for continuous numerical data. Then we replace missing values (NAs) in the network_type column with the mode value calculated in the previous step. This is achieved using the ifelse() function. It substitutes occurrences of "3" with "3G" in the network_type column using the sub() function. This is done to ensure consistency or clarity in the data.

CODE

```
dataset <- new_cq1
# Replace '3GG' with '3G' in the 'network_type' column
dataset$network_type <- ifelse(dataset$network_type == "3GG", "3G",
dataset$network_type)
```

number of rows and columns of the dataset.

```
dim(dataset)
## [1] 1370 8
#Generates a frequency table for network_type
table(dataset$network_type)
##
## 2G 3G 4G
## 34 195 1141
#Number of Rows in the dataset
nrow(dataset)
## [1] 1370
#Number of Columns
ncol(dataset)
## [1] 8
#Summary
summary(dataset)
```

```
##      operator         inout_travelling       rating      calldrop_category
##   Length:1370        Length:1370        Min.   :1.000   Length:1370
##   Class :character   Class :character   1st Qu.:1.000   Class :character
##   Mode  :character   Mode  :character   Median :3.000   Mode  :character
##                                         Mean   :2.988
##                                         3rd Qu.:4.000
##                                         Max.   :5.000
##       latitude          longitude        state_name         network_type
##   Min.   :-1.00      Min.   :-1.00      Length:1370        Length:1370
##   1st Qu.:-1.00      1st Qu.:-1.00      Class :character   Class :character
##   Median :-1.00      Median :-1.00      Mode  :character   Mode  :character
##          Mean   : 7.23              Mean   :31.40
##          3rd Qu.:18.97              3rd Qu.:73.52
##  Max.   :30.35      Max.   :88.35
```

CODE

```
# Create a contingency table

contingency_table <- table(dataset$inout_travelling, dataset$calldrop_category)

# Print the contingency table

print(contingency_table)
```

EXPLANATION

A summary of the dataset dataframe, including information such as minimum, 1st quartile, median, mean, 3rd quartile, and maximum values for numeric columns, as well as counts of unique values for categorical columns. It offers a quick overview of the dataset's structure and contents. There are 1370 rows and 8 columns in the dataset. Now, we replace any occurrence of "3GG" with "3G" in the network_type column of the dataset dataframe, using the ifelse function to conditionally replace values. Now we check the count of each unique value. The output shows the count of occurrences for each network type (2G, 3G, 4G) which is 34 195 1141 respectively.

```
            Call Dropped Poor Voice Quality Satisfactory
Indoor                54                204          429
Outdoor               26                 90          117
Travelling             7                110          333
```

CODE

```
# Calculate the expected frequencies

expected_frequencies <- prop.table(contingency_table, margin = 1)

# Print the expected frequencies

print(expected_frequencies)
```

After this we created a contingency table to understand the relationship between two categorical variables (inout_travelling and calldrop_category) in the dataset. Contingency tables help visualize the association between categorical variables by showing how the frequency of one variable varies across different levels of another variable. This can be useful for exploratory data analysis and identifying patterns or dependencies between variables.

	Call Dropped	Poor Voice Quality	Satisfactory
Indoor	0.07860262	0.29694323	0.62445415
Outdoor	0.11158798	0.38626609	0.50214592
Travelling	0.01555556	0.24444444	0.74000000

Calculating expected frequencies is necessary to determine if there are any significant deviations between the observed frequencies (actual counts) and the frequencies that would be expected if there were no association between the two categorical variables (inout_travelling and calldrop_category). By comparing observed and expected frequencies, we can assess whether there is an association or dependency between the two variables. This information is crucial for understanding the relationship between variables and making informed decisions in data analysis or hypothesis testing.

FINDINGS

Here we used the chi-square test to assess whether there is a significant association (i.e., dependence) between two categorical variables in a contingency table. It also evaluates whether the observed frequency distribution differs significantly from the distribution that would be expected if the variables were independent. The chi-square test is a non-parametric test, meaning it does not make assumptions about the distribution of the data. This property makes it robust and applicable to a wide range of scenarios, including cases where the data do not follow a normal distribution.

As there is a significant association between the "inout_travelling" and "calldrop_category" variables in the dataset. The chi-square statistic is 51.899, and the degrees of freedom (df) is 4. The p-value is very small, approximately 1.448e-10, which is significantly less than the typical significance level of 0.05 which suggests strong evidence to reject the null hypothesis of independence between the variables. This means that there is a significant association or dependency between the 'inout_travelling' and 'calldrop_category' variables in the dataset.

CODE

```
# Perform chi-square test

chi_square_result <- chisq.test(contingency_table)

# Print the chi-square test results

print(chi_square_result)
```

```
 Pearson's Chi-squared test

data:  contingency_table
X-squared = 51.899, df = 4, p-value = 1.448e-10
```

WHY CHI SQUARE TEST ?

Chi-square test was used to analyze the association between the variables "**inout_travelling**" and "**calldrop_category**". The reason for using the chi-square test in this context is because both "**inout_travelling**" and "**calldrop_category**" are categorical variables, and the chi-square test is commonly employed to test for independence between categorical variables.

Chi-square test is appropriate for analyzing categorical data and determining whether there is a significant association between two categorical variables. It evaluates whether the observed frequency distribution differs significantly from the expected frequency distribution under the assumption of independence.

CODE

```
# Handle missing values

dataset <- na.omit(dataset)
```

Now we remove the unwanted data from the dataset. As machine learning algorithms require categorical variables to be encoded as factors, the categorical variables inout_travelling and calldrop_category are converted into factors. The dataset is divided into two parts: features and the target variable (calldrop_category). Features include predictor variables such as operator, inout_travelling, rating, latitude, longitude, and state_name, while the target variable is the variable to be predicted. The dataset is further split into training and test sets. 80% of the data is used for training (trainFeatures and trainTarget), and the remaining 20% is used for testing (testFeatures and testTarget). This split ensures that the model's performance can be evaluated on unseen data. A confusion matrix is generated to evaluate the performance of the model on the test set. The confusion matrix compares the predicted values (predictions) with the actual values (testTarget), providing insights into the model's accuracy, precision, recall.

CODE

```
# Handle missing values

dataset <- na.omit(dataset)
```

The line dataset <- na.omit(dataset) is used to remove any rows from the dataset that contain missing values (NA). This step is performed to ensure that the analysis is conducted on complete cases where all required information is available.

When there are missing values in the dataset, some statistical functions or machine learning algorithms might not work properly or may produce biased results.

CODE

```
# Encode categorical variables

dataset$inout_travelling <- as.factor(dataset$inout_travelling)

dataset$calldrop_category <- as.factor(dataset$calldrop_category)

# Split dataset into features and target variable

features <- dataset[, c("operator", "inout_travelling", "rating", "latitude", "longitude", "state_name")]

target <- dataset$calldrop_category

# Split data into training and test sets (80% train, 20% test)

set.seed(123)
trainIndex <- createDataPartition(target, p = 0.8, list = FALSE)
```

```
trainFeatures <- features[trainIndex, ]
trainTarget <- target[trainIndex]
testFeatures <- features[-trainIndex, ]
testTarget <- target[-trainIndex]

# Train the model

model <- randomForest(trainTarget ~ ., data = trainFeatures)

# Make predictions on the test set

predictions <- predict(model, newdata = testFeatures)

#Confusion Matrix

confusionMatrix(predictions, testTarget)
```

EXPLANATION

1. Encoding Categorical Variables:

`as.factor(dataset$inout_travelling)` and `as.factor(dataset$calldrop_category)`: These lines convert the "inout_travelling" and "calldrop_category" columns from strings to factors. This is necessary for machine learning algorithms that require numerical inputs.

2. Splitting Data:

`features <- dataset[, c("operator", "inout_travelling", "rating", "latitude", "longitude", "state_name")]` and `target <- dataset$calldrop_category`: These lines separate the dataset into features (independent variables) and the target

variable (dependent variable). The target variable we want to predict is the call drop category.

3. Creating Training and Test Sets:

- `set.seed(123)` sets a random seed for reproducibility.
- `createDataPartition` splits the data into training (80%) and test (20%) sets based on the target variable. This ensures the model doesn't simply memorize the training data and can generalize to unseen data.

4. Training the Model:

`randomForest(trainTarget ~ ., data = trainFeatures)`: This line trains a random forest model using the training features and target variable. The `.` before the comma indicates that all features will be used for prediction.

5. Making Predictions:

-`predict(model, newdata = testFeatures)`: This line uses the trained model to make predictions on the test features (data it hasn't seen before).

6. Confusion Matrix:

`confusionMatrix(predictions, testTarget)`: This line creates a confusion matrix, which shows how well the model performed on the test set. It helps visualize the number of correct and incorrect predictions for each possible category.

```
Confusion Matrix and Statistics

                    Reference
Prediction           Call Dropped Poor Voice Quality Satisfactory
  Call Dropped                  3                  3            0
  Poor Voice Quality            1                 13            2
  Satisfactory                  0                  0           92
```

```
                Accuracy : 0.9474
                  95% CI : (0.889, 0.9804)
    No Information Rate : 0.8246
    P-Value [Acc > NIR] : 9.224e-05

                   Kappa : 0.8319
```

```
##                                          Kappa : 0.8319
##
##                  Mcnemar's Test P-Value : NA
##
##                  Statistics by Class:
##
##                                 Class: Call Dropped Class: Poor Voice Quality
## Sensitivity                              0.75000                      0.8125
## Specificity                              0.97273                      0.9694
## Pos Pred Value                           0.50000                      0.8125
## Neg Pred Value                           0.99074                      0.9694
## Prevalence                               0.03509                      0.1404
## Detection Rate                           0.02632                      0.1140
## Detection Prevalence                     0.05263                      0.1404
## Balanced Accuracy                        0.86136                      0.8909
##                                                        Class: Satisfactory
## Sensitivity                                                         0.9787
## Specificity                                                         1.0000
## Pos Pred Value                                                      1.0000
## Neg Pred Value                                                      0.9091
## Prevalence                                                          0.8246
## Detection Rate                                                      0.8070
```

```
##          Detection Prevalence                                    0.8070
## Balanced Accuracy                0.9894
```

The overall accuracy of 0.9474 indicates that the model's predictions are correct for approximately 94.74% of the instances in the test set. The confidence interval (0.889, 0.9804) provides a range within which we are confident the true accuracy lies.

The No Information Rate (NIR) of 0.8246 suggests that the model outperforms a baseline model that always predicts the most prevalent class (in this case, Satisfactory). This means that our model is providing predictions significantly better than simply guessing the majority class every time, demonstrating its effectiveness in making meaningful predictions.

CODE

```
#Sentiment Analysis

# Convert sentiment-related variables to factors (if not already)

dataset$rating <- as.factor(dataset$rating)

dataset$calldrop_category <- as.factor(dataset$calldrop_category)

# Define sentiment categories

positive_sentiment <- c("Satisfactory")
negative_sentiment <- c("Poor Voice Quality", "Call Dropped")

# Perform sentiment analysis

.
dataset$sentiment <- ifelse(dataset$rating %in% positive_sentiment,
"Positive",
       false(dataset$calldrop_category %in% negative_sentiment,
"Negative","Positive"))


# Analyze sentiment distribution
```

```
sentiment_counts <- table(dataset$sentiment)
print(sentiment_counts)
```

```
##
##  Negative  Positive
##       103       472
```

```
sentiment_counts <- table(dataset$sentiment)

# Calculate sentiment percentages

sentiment_percentages <- prop.table(sentiment_counts) * 100

# Print sentiment percentages

print(sentiment_percentages)
```

EXPLANATION

1. Encoding Variables:

`as.factor(dataset$rating)` and `as.factor(dataset$calldrop_category)` convert these variables to factors (if not already) for compatibility with the sentiment analysis process.

2. Defining Sentiment Categories:

`positive_sentiment` and `negative_sentiment` define specific categories based on your understanding of the data. These determine how sentiment will be labelled.

3. Sentiment Analysis:

The `ifelse` function assigns sentiment labels to each row based on the defined categories:

If `rating` is in `positive_sentiment`, label as "Positive".

Else, if `calldrop_category` is in `negative_sentiment`, label as "Negative".

Otherwise, label as "Positive" (as a default).

4. Analyzing Sentiment Distribution:

`table(dataset$sentiment)` counts the occurrences of each sentiment label ("Positive", "Negative") in the dataset.

5. Calculating Sentiment Percentages:

`prop.table(sentiment_counts) * 100` converts the counts to percentages, expressing the proportion of each sentiment category within the data.

6. Printing Results:

The code prints the counts of each sentiment label and then the calculated percentages, providing a quantitative overview of the sentiment distribution in your dataset.

```
Negative Positive
17.91304 82.08696
```

After categorizing the sentiments into two distinct categories, namely "positive" and "negative," based on the classification of reviews, the criteria were established as follows: any sentiment labelled as "Satisfactory" was deemed as "positive," while sentiments labelled as "Call Dropped" and "Poor Voice Quality" were categorized as "negative."

Following the classification process, an analysis was conducted on the dataset. The findings revealed that approximately 82% of the dataset contained sentiments categorized as "positive." These sentiments were characterized by reviews labeled as "Satisfactory," indicating a favorable response towards the voice quality.

Conversely, around 18% of the dataset comprised sentiments classified as "negative." These sentiments were associated with reviews marked as "Call Dropped" or "Poor Voice Quality," indicating instances where users expressed dissatisfaction or encountered issues with the voice quality.

CODE

```
# Create a pie chart with percentages

pie(sentiment_counts, labels = paste0(names(sentiment_counts), " (", round(sentiment_percentages, 2), "%)"), col = c("red", "green"))

# Add a legend

legend("topright", legend = names(sentiment_counts), fill = c("red", "green"), border = NA)
```

EXPLANATION

1. Creating the Pie Chart:

- **pie(sentiment_counts, ...)**: This line creates a pie chart using the sentiment_counts variable, which likely stores the counts of different sentiment categories (e.g., "Positive", "Negative").

- **labels = paste0(...)**: This part customizes the labels for each slice of the pie. It combines:

    - The names of the sentiment categories (e.g., "Positive")

    - The corresponding percentages from sentiment_percentages, rounded to two decimal places

- **col = c("red", "green")**: This assigns colors to the pie slices, typically using red for "Negative" and green for "Positive".

2. Adding a Legend:

- **legend("topright", ...)**: This adds a legend to the chart, positioning it in the top right corner.

- **legend = names(sentiment_counts)**: This specifies the labels for the legend, using the same names as the sentiment categories.

- **fill = c("red", "green")**: This ensures the legend colors match the pie slices, making it clear which color represents which sentiment.

- **border = NA**: This removes any border around the legend boxes, providing a cleaner look.

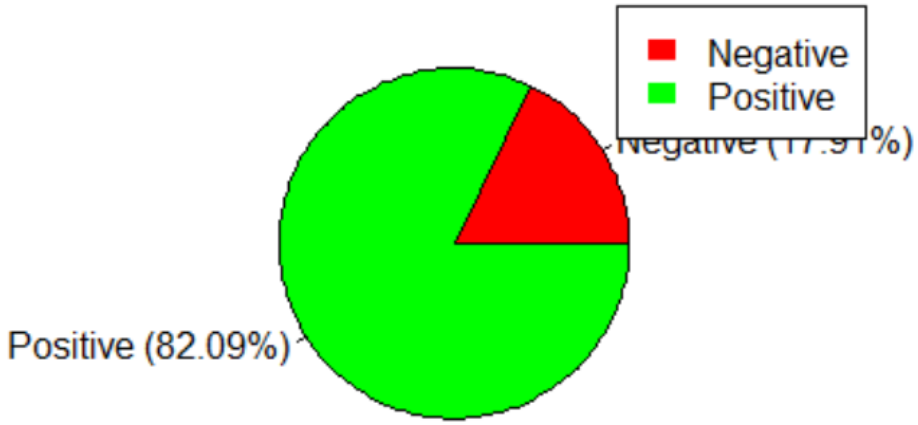


Fig 1

This breakdown of sentiment distribution provides valuable insights into the overall reception of voice quality, highlighting the prevalence of positive sentiments while also acknowledging the existence of negative feedback within the dataset. Such analysis aids in understanding user perception and can inform decision-making processes related to service improvements and customer satisfaction.

SATISFACTION INDIVIUAL OPERATOR

The bar charts displays the distribution of satisfaction ratings among customers, allowing for an easy comparison of the proportion of customers satisfied at different rating levels.

X-axis: Represents the different satisfaction ratings, likely

- 1 represent the lowest satisfaction level (e.g., "Poor" or "Very Dissatisfied").

- 2 represent a slightly higher satisfaction level (e.g., "Fair" or "Dissatisfied").

- 3 represent a moderate satisfaction level (e.g., "Average" or "Neutral").

- 4 signify a higher satisfaction level (e.g., "Good" or "Satisfied").

- 5 represent the highest satisfaction level (e.g., "Excellent" or "Very Satisfied").

Y-axis: Represents the percentage of customer responses corresponding to each satisfaction rating category. The percentage indicates the proportion of customers who assigned a particular satisfaction rating relative to the total number of responses for that particular operator.

AIRTEL

CODE

```
#To find out the satisfaction rate of Airtel

# Calculate satisfaction rating counts and percentages for Airtel

airtel_satisfaction <- dataset %>%
  filter(operator == "Airtel") %>%
  group_by(rating) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)

# Create a bar chart for Airtel satisfaction ratings

bar_chart_airtel <- ggplot(data = airtel_satisfaction, aes(x = rating, y = percentage)) +
  geom_bar(stat = "identity", fill = "red") +
  xlab("Satisfaction Rating") +
  ylab("Percentage") +
  ggtitle("Airtel Satisfaction Rating Distribution")

# Display the bar chart

print(bar_chart_airtel)
```

EXPLANATION

1. Filtering Airtel Users:

- dataset %>% filter(operator == "Airtel"): This line selects only rows from the dataset where the operator is "Airtel", isolating data specific to Airtel users.

2. Calculating Satisfaction Counts:

- group_by(rating) %>% summarize(count = n()): This step groups the filtered data by the rating variable and counts the occurrences of each rating level (e.g., "Very Satisfied", "Satisfied", etc.) for Airtel users.

3. Converting Counts to Percentages:

- mutate(percentage = count / sum(count) * 100): This line calculates the percentage of users who gave each rating level by dividing each count by the total count of Airtel users and multiplying by 100.

4. Visualizing Satisfaction Distribution:

- ggplot(data = airtel_satisfaction, aes(x = rating, y = percentage)) + geom_bar(stat = "identity", fill = "red"): This part creates a bar chart using ggplot2. It sets the x-axis to display rating levels and the y-axis to display the calculated percentages. Each bar represents the percentage of Airtel users who gave that specific rating. The fill = "red" sets the bar color to red for visual emphasis.

- xlab("Satisfaction Rating") + ylab("Percentage"): These lines add labels to the x and y axes.

- ggtitle("Airtel Satisfaction Rating Distribution"): This sets the title of the chart.

5. Displaying the Chart:

- print(bar_chart_airtel): This finally displays the generated bar chart, giving you a clear visual representation of how Airtel users rate their satisfaction across different rating levels.

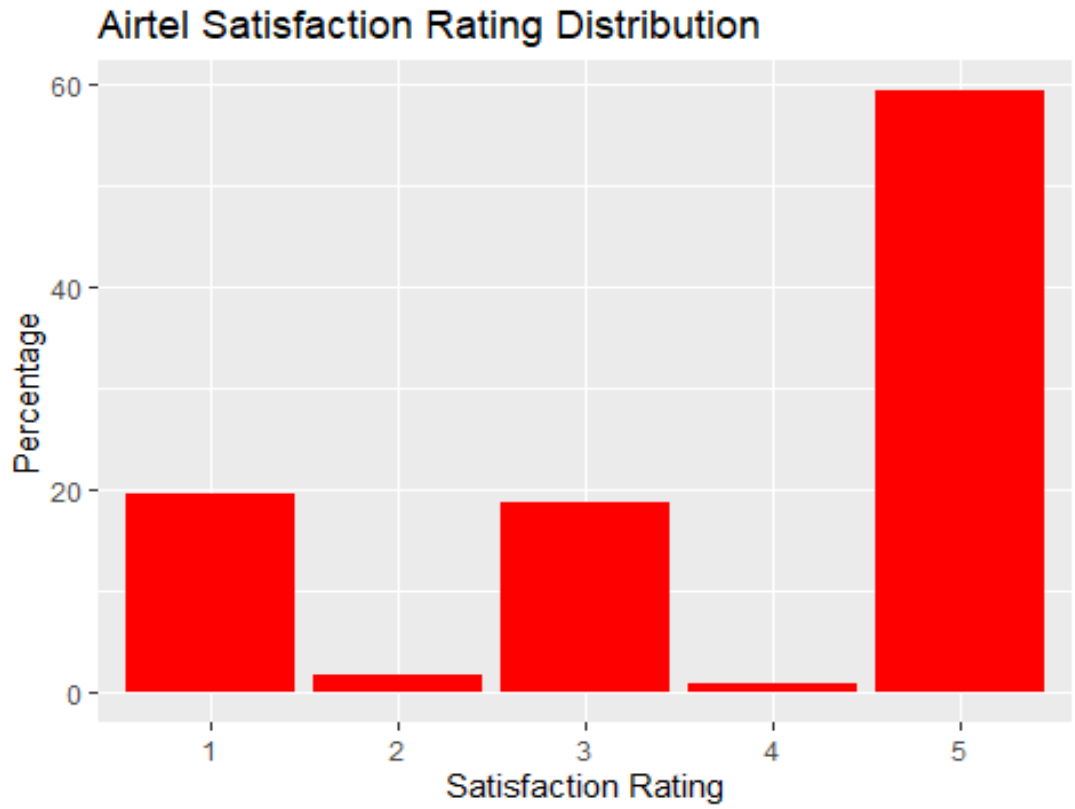


Fig 2

Majority of the ratings are clustered around the higher satisfaction levels, particularly in the range of 4 and 5, suggesting that a significant proportion of customers rated their satisfaction with Airtel positively.There are relatively fewer ratings towards the lower end of the scale (1 to 3), indicating that a smaller percentage of customers expressed dissatisfaction with Airtel's services.

Overall, the distribution appears skewed towards higher satisfaction ratings, implying that a considerable portion of customers had a positive experience with Airtel.

VODAFONE IDEA

The bar chart in fig 3 visually displays the distribution of satisfaction ratings among Vodafone Idea customers, allowing for an easy comparison of the proportion of customers satisfied at different rating levels.

CODE

```
#To calculate the Vodafone Satisfaction rating

# Calculate satisfaction rating counts and percentages for Vodafone

vodafone_satisfaction <- dataset %>%
  filter(operator == "VI") %>%
  group_by(rating) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)

print(vodafone_satisfaction)

## # A tibble: 5 × 3
##   rating count percentage
##   <fct>  <int>      <dbl>
## 1 1          9       4.17
## 2 2          3       1.39
## 3 3          7       3.24
## 4 4        118       54.6
## 5 5         79       36.6

# Create a bar chart for Vodafone satisfaction ratings

bar_chart_vi <- ggplot(data = vodafone_satisfaction, aes(x = rating, y = percentage)) +
  geom_bar(stat = "identity", fill = "orange") +
  xlab("Satisfaction Rating") +
  ylab("Percentage") +
  ggtitle("Vodafone Satisfaction Rating Distribution")

# Display the bar chart
```

```
print(bar_chart_vi)
```

EXPLANATION

1. Filtering Vodafone Users:

- filter(operator == "VI"): This line selects only rows where the operator is "VI", assuming "VI" represents Vodafone in your dataset.

2. Calculating Satisfaction Counts & Percentages:

- Similar to the Airtel section, the code groups by rating, counts occurrences for each rating level, and calculates percentages based on the total number of Vodafone users.

3. Displaying Results:

- print(vodafone_satisfaction) directly outputs the calculated counts and percentages for each rating level in a table format.

4. Visualizing Satisfaction Distribution:

- ggplot(...): This creates a bar chart similar to the Airtel chart, but with Vodafone-specific data and an orange fill colour.

5. Displaying the Chart:

- print(bar_chart_vi) displays the generated bar chart.

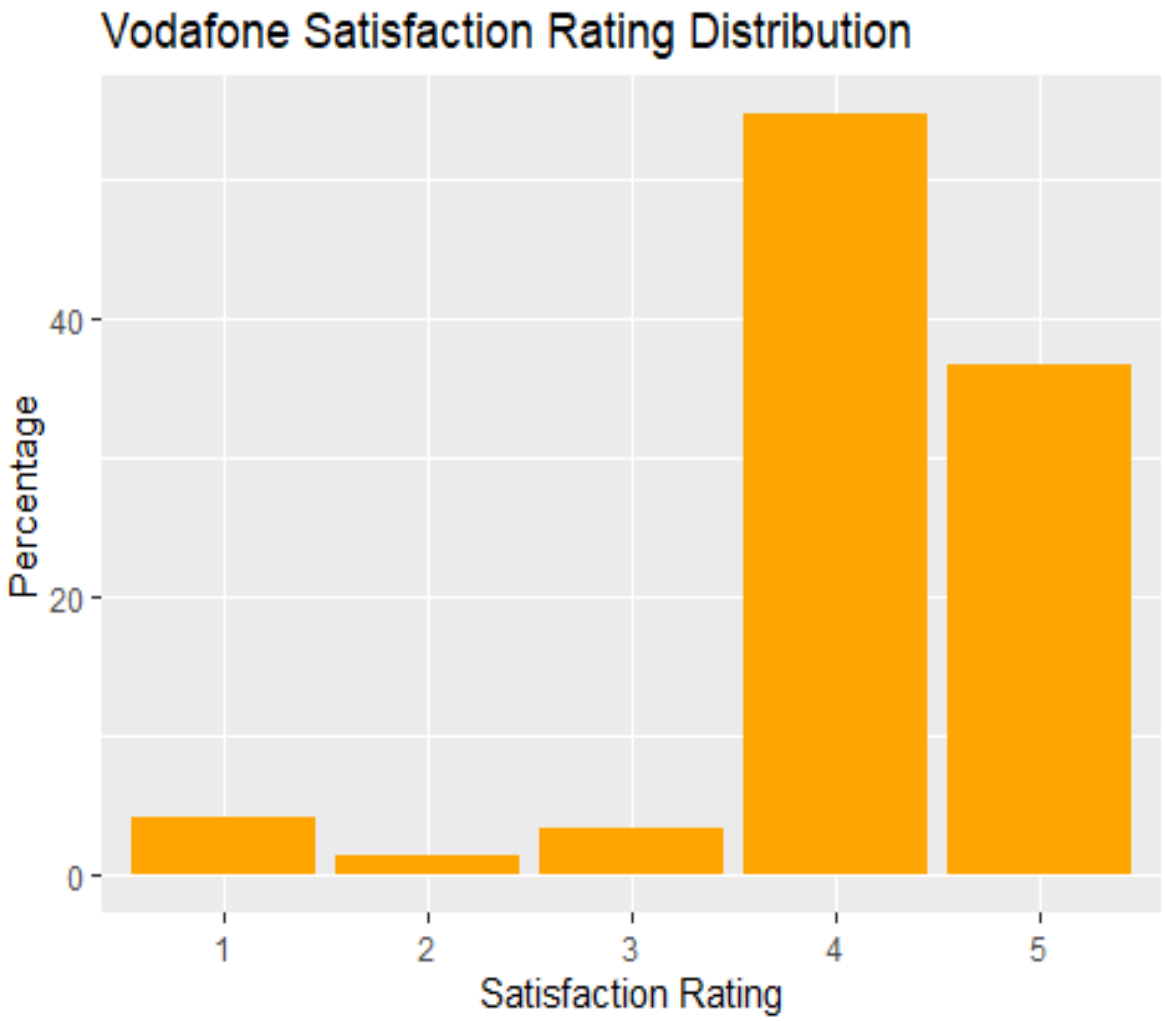


Fig 3

There is a noticeable concentration of ratings towards the higher end of the satisfaction scale, particularly in the range of 4 and 5. Relatively fewer ratings are observed towards the lower satisfaction levels (1 to 3), suggesting a lower percentage of customers expressing dis-satisfaction. The distribution appears skewed towards higher satisfaction ratings, indicating that a significant proportion of customers provided positive ratings for Vodafone Idea's services.

Overall, Vodafone Idea demonstrates a favourable satisfaction rating distribution, with a majority of customers expressing satisfaction with their services.

BSNL

The bar chart in fig 4 visually displays the distribution of satisfaction ratings among BSNL customers, allowing for an easy comparison of the proportion of customers satisfied at different rating levels.

CODE

```
#To calculate the satisfaction rating of BSNL

# Calculate satisfaction rating counts and percentages for BSNL

bsnl_satisfaction <- dataset %>%
  filter(operator == "BSNL") %>%
  group_by(rating) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)

# Create a bar chart for BSNL satisfaction ratings

bar_chart_bsnl <- ggplot(data = bsnl_satisfaction, aes(x = rating, y = percentage)) +
  geom_bar(stat = "identity", fill = "green") +
  xlab("Satisfaction Rating") +
  ylab("Percentage") +
  ggtitle("BSNL Satisfaction Rating Distribution")

# Display the bar chart

print(bar_chart_bsnl)
```

Explanation

1. Filtering BSNL Users:

- filter(operator == "BSNL"): This line selects only rows where the operator is "BSNL", isolating data specific to BSNL users.

2. Calculating Satisfaction Counts & Percentages:

- Similar to the previous analyses, the code groups by rating, counts occurrences for each rating level, and calculates percentages based on the total number of BSNL users.

3. Displaying Results:

- print(bsnl_satisfaction) directly outputs the calculated counts and percentages for each rating level in a table format.

4. Visualizing Satisfaction Distribution:

- ggplot(...): This creates a bar chart similar to the ones for Airtel and Vodafone, but with BSNL-specific data and a green fill color.

5. Displaying the Chart:

- print(bar_chart_bsnl) displays the generated bar chart.

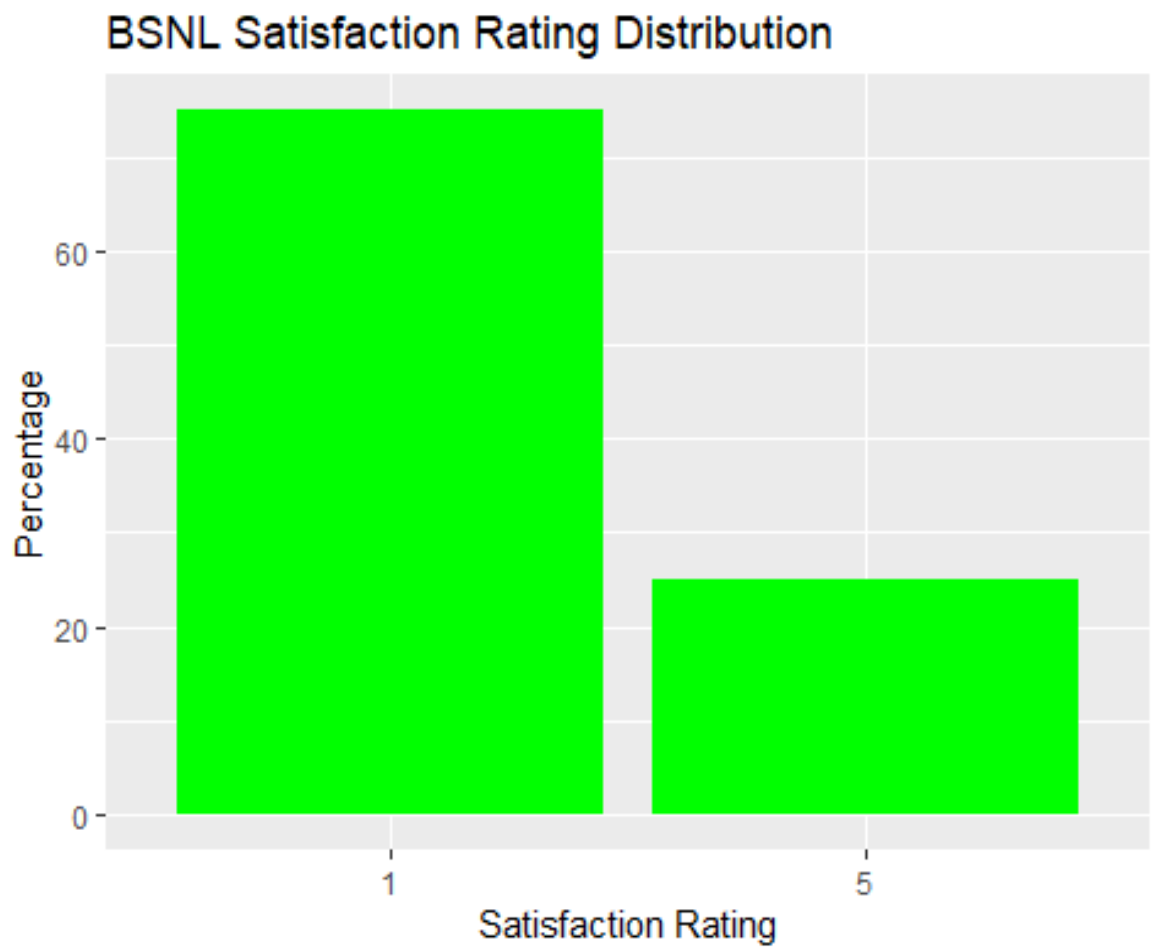


Fig 4

There is a noticeable proportion of ratings in the lower satisfaction levels (1 and 2), indicating a subset of customers expressing dissatisfaction with BSNL's services. However, a significant portion of ratings falls within the higher satisfaction levels (4 and 5), suggesting that a considerable number of customers are satisfied with BSNL's services.The distribution appears somewhat balanced, with no extreme skew towards either high or low satisfaction ratings.

Overall, while there is room for improvement, BSNL's satisfaction rating distribution showcases a mix of both positive and negative feedback from customers.

RELIANCE JIO

The bar chart in fig 5 visually displays the distribution of satisfaction ratings among Reliance Jio customers, allowing for an easy comparison of the proportion of customers satisfied at different rating levels.

CODE

```
#To calculate the percentage RJio satisfaction rating

# Calculate satisfaction rating counts and percentages for RJio

RJio_satisfaction <- dataset %>%
  filter(operator == "RJio") %>%
  group_by(rating) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)

# Create a bar chart for BSNL satisfaction ratings

bar_chart_RJio <- ggplot(data = RJio_satisfaction, aes(x = rating, y = percentage)) +
  geom_bar(stat = "identity", fill = "navyblue") +
  xlab("Satisfaction Rating") +
  ylab("Percentage") +
  ggtitle("RJio Satisfaction Rating Distribution")

# Display the bar chart

print(bar_chart_RJio)
```

EXPLANATION

1. **Filtering RJio Users:**
   - filter(operator == "RJio"): This line selects only rows where the operator is "RJio", isolating data specific to RJio users.

2. **Calculating Satisfaction Counts and Percentages:**
   - Similar to the previous analyses, the code groups by rating, counts occurrences for each rating level, and calculates percentages based on the total number of RJio users.

3. **Displaying Results:**
   - print(RJio_satisfaction) directly outputs the calculated counts and percentages for each rating level in a table format, providing a numerical overview.

4. **Visualizing Satisfaction Distribution:**
   - ggplot(...): This creates a bar chart similar to the ones for other operators, but with RJio-specific data and a navy blue fill color for visual distinction.

5. **Displaying the Chart:**
   - print(bar_chart_RJio) displays the generated bar chart, allowing you to see the visual distribution of satisfaction ratings.

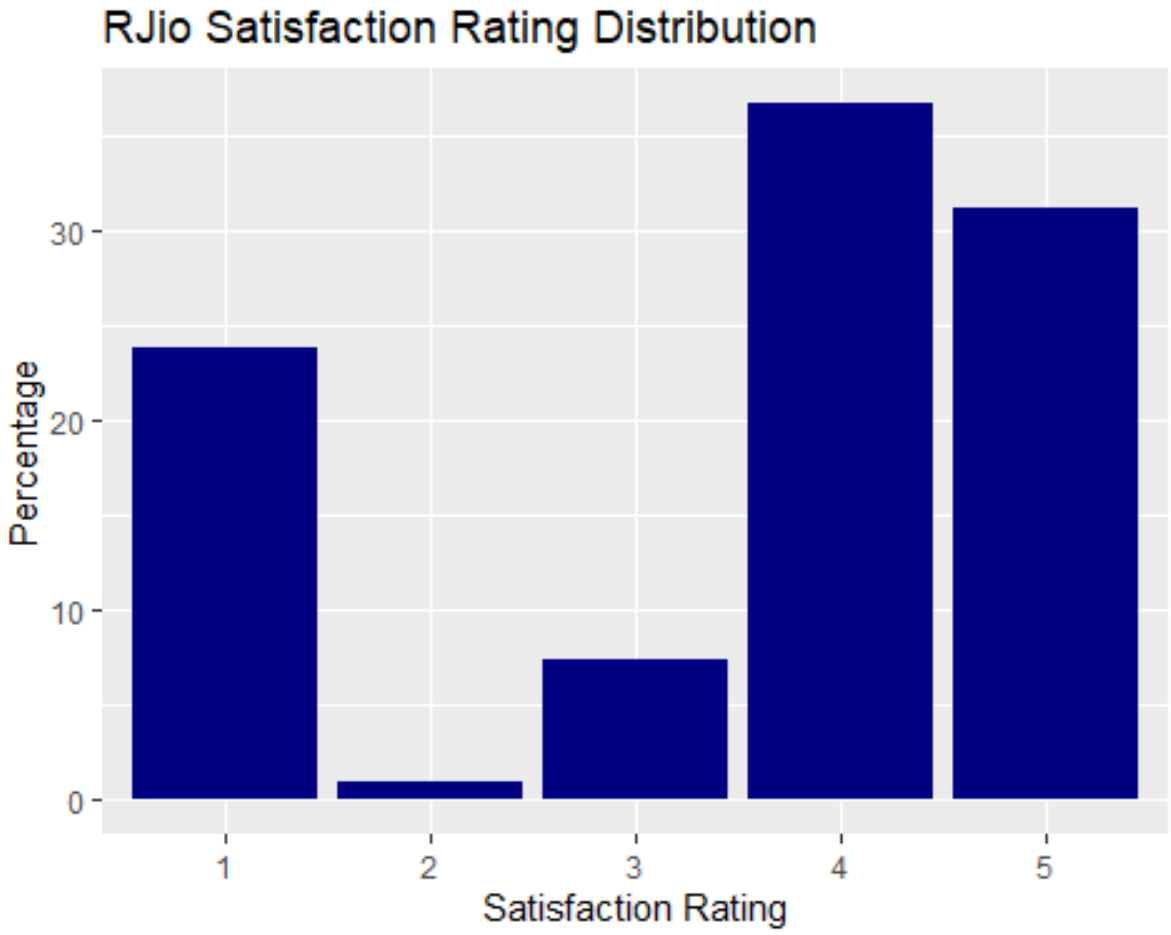


Fig 5

There is a notable concentration of ratings in the higher satisfaction levels (4 and 5), indicating a significant proportion of customers expressing satisfaction with RJio's services. Ratings in the lower satisfaction levels (1 and 2) appear to be evident, suggesting dissatisfaction among customer and small expressing an average satisfaction. Overall, the chart illustrates a predominantly positive sentiment towards Reliance Jio, with a majority of customers providing higher satisfaction ratings.

After checking the individual operator satisfaction rating our next area to be explored is to know how many people observed issue while travelling. With the help of bar chart providing a detailed breakdown of call drop categories reported by individuals while travelling.

Where,

- **X-axis** This axis represents the different categories of **call drop issues** reported by users while travelling. Each category, such as "Call Dropped" or "Poor Voice Quality," is listed along the x-axis.

- Y-axis The y-axis indicates the **percentage** of people who observed each specific call drop category while travelling. It represents the relative frequency of each call drop category among users who reported issues while travelling.

CODE

```
#to calculate and plot the percentage of people observing the issue while Travelling

# Calculate the percentage of people observing the issue while travelling

travelling_issue <- dataset %>%
  filter(inout_travelling == "Travelling") %>%
  group_by(calldrop_category) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)

# Create a bar chart for the percentage of people observing the issue while travelling

bar_chart_travelling <- ggplot(data = travelling_issue, aes(x = calldrop_category, y = percentage)) +
  geom_bar(stat = "identity", fill = "purple") +
  xlab("Call Drop Category") +
  ylab("Percentage") +
  ggtitle("Percentage of People Observing Issue While Travelling")

# Display the bar chart

print(bar_chart_travelling)
```

EXPLANATION

1. Filtering Travelling Users:

- filter(inout_travelling == "Travelling"): This line selects only rows where the inout_travelling variable is "Travelling", isolating data for users who were travelling when the call drop occurred.

2. Grouping by Call Drop Category:

- group_by(calldrop_category): This groups the filtered data by the calldrop_category variable, which likely represents different types of call drop issues (e.g., "Sudden Drop", "Poor Voice Quality").

3. Calculating Percentages:

- summarize(count = n()): This counts the occurrences of each calldrop_category within the travelling user group.

- mutate(percentage = count / sum(count) * 100): This calculates the percentage of people who experienced each calldrop_category by dividing the count for each category by the total count of travelling users and multiplying by 100.

4. Creating Bar Chart:

- ggplot(data = travelling_issue, aes(x = calldrop_category, y = percentage)) + geom_bar(stat = "identity", fill = "purple"): This creates a bar chart using ggplot2. It sets the x-axis to display the different calldrop_category labels and the y-axis to display the calculated percentages. Each bar represents the percentage of travelling users who

experienced that specific call drop category. The fill = "purple" sets the bar color for visual emphasis.

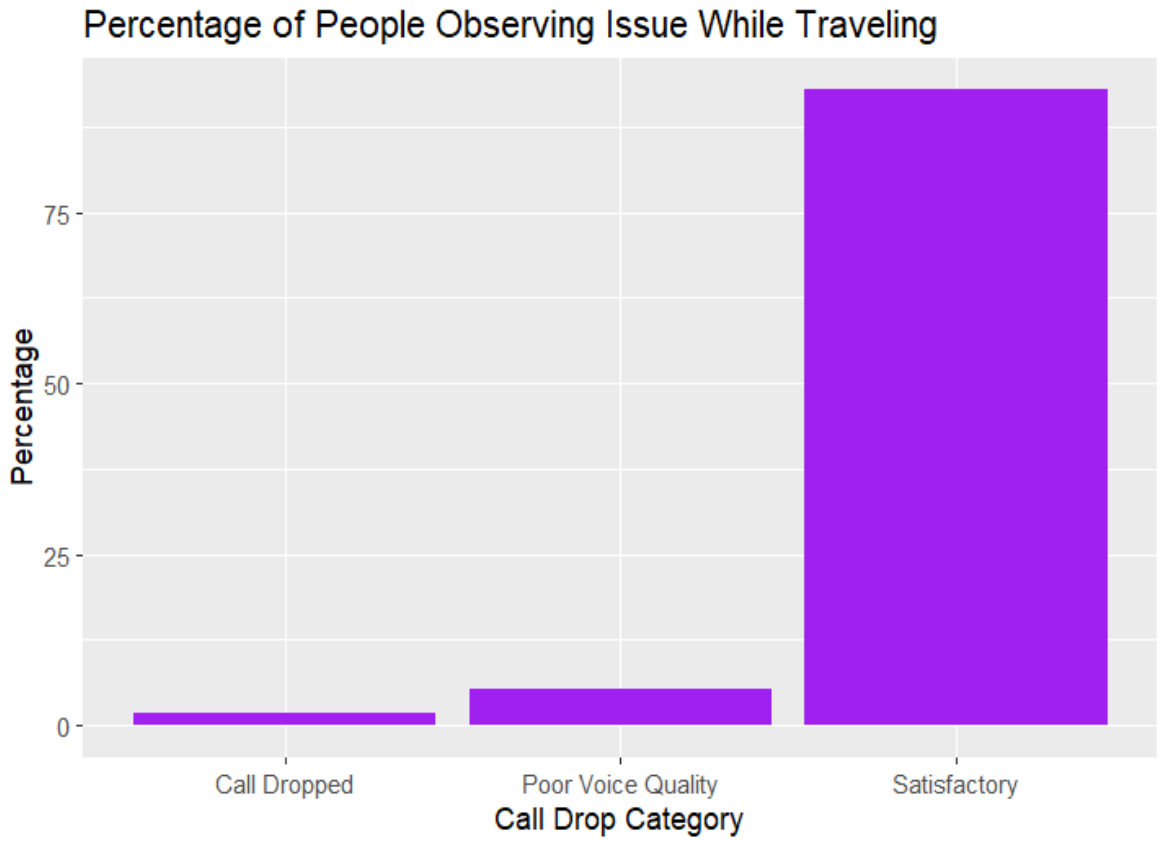
- xlab("Call Drop Category") + ylab("Percentage"): These lines add labels to the x and y axes.

- ggtitle("Percentage of People Observing Issue While Travelling"): This sets the title of the chart.

5. Displaying the Chart:

- print(bar_chart_travelling): This finally displays the generated bar chart, allowing you to visualize the distribution of call drop issues experienced by travelling users across different categories.

Fig



6

The chart helps identify the prevalence of various call drop issues encountered by users during travel. By examining the heights of the bars, we can discern which call drop categories are more frequently reported by users while they are on the move. This information can be valuable for service providers to address common issues and improve the overall user experience, particularly during travel scenarios.

We needed to identify which network type had the most call drops. The bar chart titled "Percentage of Call Drops by Network Type" illustrates the distribution of call drops across different network types. Each bar represents a network type, and the height of the bar corresponds to the percentage of call drops attributed to that network type

CODE

```
# Calculate the percentage of call drops for each network type

network_call_drops <- dataset %>%
```

```
  group_by(network_type) %>%
  summarize(call_drops = sum(calldrop_category == "Call Dropped")) %>%
  mutate(percentage = call_drops / sum(call_drops) * 100)

# Sort the data frame by percentage in descending order
sorted_network_call_drops <- network_call_drops %>%
  arrange(desc(percentage))

# Display the sorted data frame
print(sorted_network_call_drops)

# Extract the network type with the most call drops
most_call_drops_network <- sorted_network_call_drops$network_type[1]

# Print the network type with the most call drops

cat("The network type with the most call drops is:", most_call_drops_network, "\n")

# Create a bar chart for the percentage of call drops by network type
bar_chart_network_call_drops <- ggplot(data = sorted_network_call_drops, aes(x = network_type, y = percentage)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  xlab("Network Type") +
  ylab("Percentage of Call Drops") +
  ggtitle("Percentage of Call Drops by Network Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readability
```

Display the bar chart

print(bar_chart_network_call_drops)

EXPLANATION

1. Calculating Call Drop Percentages:

- group_by(network_type): Groups data based on different network types.

- summarize(call_drops = sum(calldrop_category == "Call Dropped")): Counts call drops for each network type.

- mutate(percentage = call_drops / sum(call_drops) * 100): Calculates percentages relative to total call drops.

2. Sorting Results:

- arrange(desc(percentage)): Arranges data in descending order of call drop percentages.

3. Identifying Network with Most Drops:

- sorted_network_call_drops$network_type[1]: Extracts the network type at the top (highest percentage).

4. Printing Information:

- print(sorted_network_call_drops): Displays the sorted data frame with network types and percentages.

- cat(...): Prints a statement identifying the network with the most call drops.

5. Visualizing Results:

- ggplot(...): Creates a bar chart using ggplot2.
- geom_bar(...): Specifies bar chart elements.
- xlab(...), ylab(...), ggtitle(...): Adds labels and title.
- theme(...): Rotates x-axis labels for better readability.
- print(...): Displays the generated bar chart.

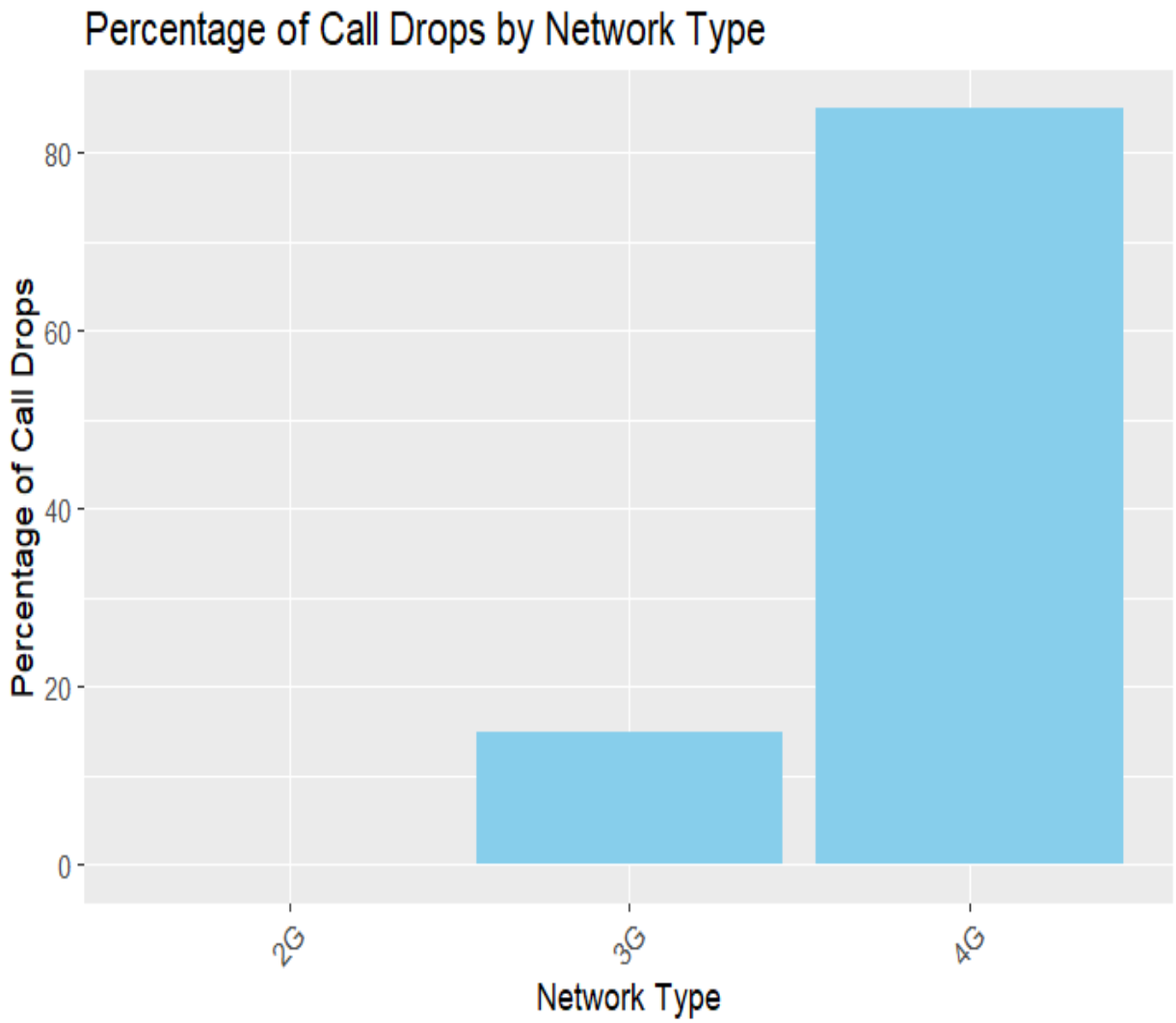


Fig 7

From the chart, it is evident that the 4G network type experiences the highest percentage of call drops compared to other network types. This information is valuable for service providers as it highlights areas where improvements or interventions may be needed to enhance network reliability and reduce call drop incidents, ultimately improving the overall user experience.

Further, we investigated the distribution of call drop incidents across different operators.

Here,

X-axis: This axis represents the different mobile network operators included in your dataset.

Y-axis: This axis represents the number of call drop incidents recorded for each operator. The height of each bar is proportional to the number of call drops.

Bars: Each bar represents the call drop count for a specific operator.

CODE

```
# Filter dataset by operator
operator_name <- "Airtel,VI,RJio,BSNL"  # Replace with the desired operator name

filtered_dataset <- dataset %>% filter(operator == operator_name)


# Create a bar chart based on operator
bar_chart <- ggplot(data = dataset, aes(x = operator)) +
  geom_bar() +
  xlab("Operator") +
  ylab("Count") +
  ggtitle("Operator-wise Distribution  of call dropped")

# Display the bar chart
print(bar_chart)
```

EXPLANATION

1. Filtering by Operator:

- operator_name <- "Airtel,VI,RJio,BSNL": This line defines a string containing the operators you want to filter for. Remember to replace it with your desired operators separated by commas.

- filtered_dataset <- dataset %>% filter(operator == operator_name): This line filters the dataset based on the operator variable. It selects only rows where the operator is included in the operator_name string.

2. Creating a Bar Chart:

- ggplot(data = dataset, aes(x = operator)) + geom_bar() + ...: This line creates a bar chart using the ggplot2 package. It sets the x-axis to represent the operator variable and uses geom_bar() to create bars for each operator.

- xlab("Operator") + ylab("Count") + ggtitle("Operator-wise Distribution"): These lines add labels to the x and y axes and a title to the chart.

3. Displaying the Chart:

- print(bar_chart): This line displays the generated bar chart, showing the distribution of data points across different operators.

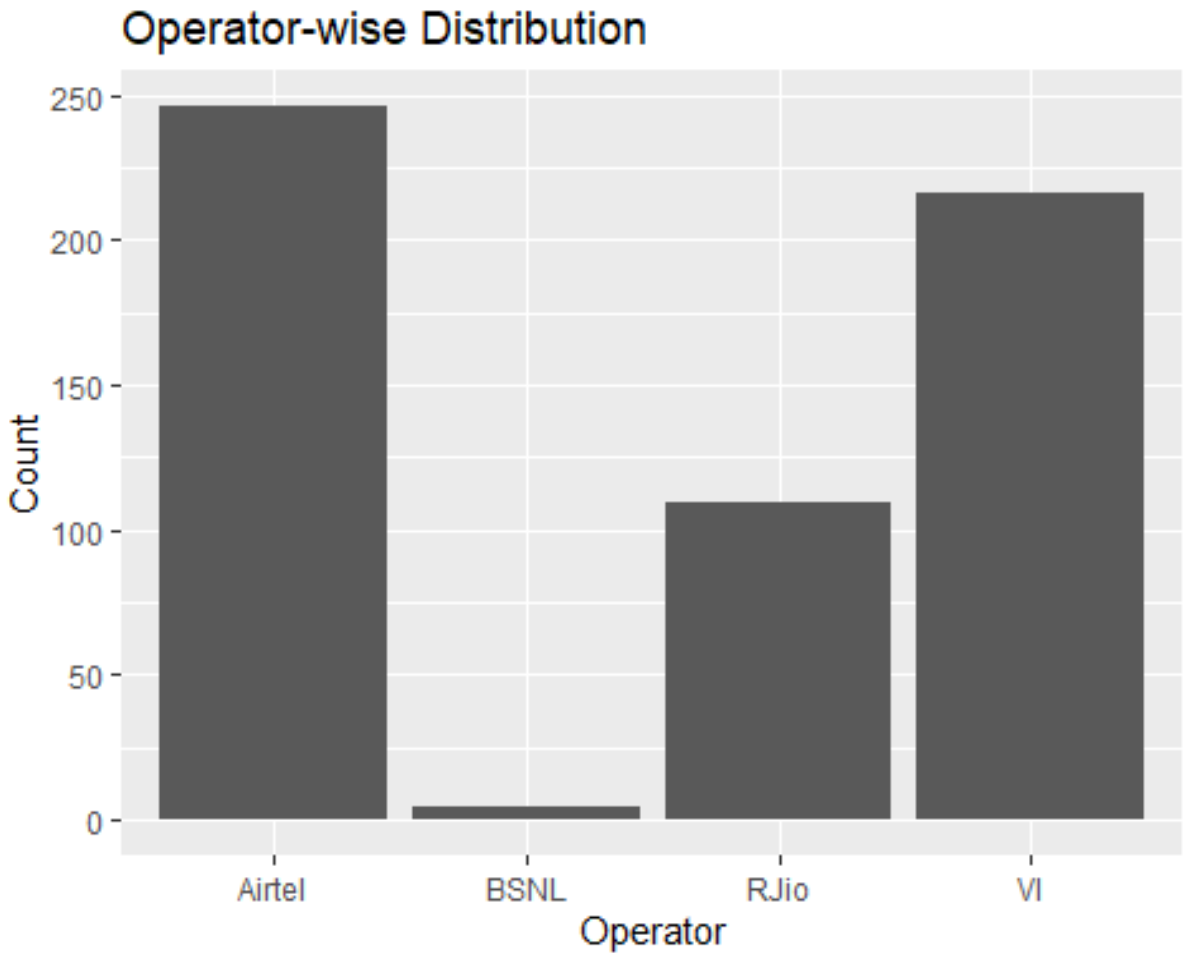


Fig 9 shows us that Airtel had the most call drops

Next, we tried to identifying regions where call drop incidents are most prevalent and we started calculating the count of call drops for each state from the provided dataset. It then sorts the resulting data frame in descending order based on the call drops count, showing which states have the highest number of call drops

CODE

```
# Calculate the count of call drops for each state
call_drops_by_state <- dataset %>%
  filter(calldrop_category == "Call Dropped") %>%
  group_by(state_name) %>%
```

```
  summarize(call_drops_count = n())

# Sort the data frame by call drops count in descending order
sorted_call_drops_by_state <- call_drops_by_state %>%
  arrange(desc(call_drops_count))

# Display the sorted data frame
print(sorted_call_drops_by_state)

# Extract the state with the maximum call drops
state_with_max_call_drops <- sorted_call_drops_by_state$state_name[1]

# Print the state with the maximum call drops
cat("The region with the maximum call drops is:", state_with_max_call_drops, "\n")
```

EXPLANATION

1. Filtering Call Drops:

- filter(calldrop_category == "Call Dropped"): Selects rows where calldrop_category is "Call Dropped", focusing on actual drop incidents.

2. Counting Call Drops per State:

- group_by(state_name): Groups data by the "state_name" variable to analyze drops for each state.

- summarize(call_drops_count = n()): Counts the occurrences of "Call Dropped" within each state group.

3. Sorting by Call Drop Count:

- arrange(desc(call_drops_count)): Sorts the data frame in descending order based on the call_drops_count column, ranking states with the highest number of call drops first.

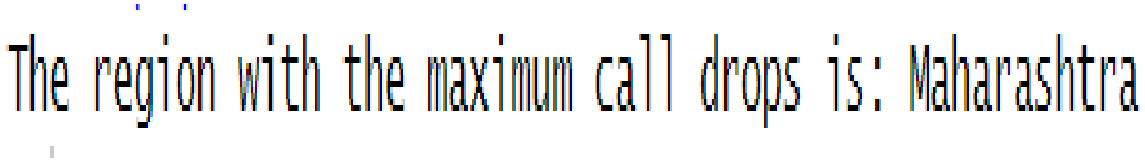
4. Identifying State with Most Drops:

- sorted_call_drops_by_state$state_name[1]: Extracts the state name from the first row of the sorted data frame, which represents the state with the maximum call drops.

5. Printing Results:

- print(sorted_call_drops_by_state): Displays the sorted data frame showing call drop counts for each state.

- cat(...): Prints a message identifying the state with the most call drops



Finally, we identified and extracts the state with the **maximum call drops** which was **Maharashtra**.

Also, we identified and extracted the information about the state which had minimum call drop which was Uttar Pradesh.

CODE

```
# Extract the state with the minimum call drops
state_with_min_call_drops <- tail(sorted_call_drops_by_state$state_name, 1)

# Print the state with the minimum call drops
cat("The region with the minimum call drops is:", state_with_min_call_drops, "\n")
```

EXPLANATION

1. Extracting State with Minimum Drops:

- tail(sorted_call_drops_by_state$state_name, 1): This line extracts the state with the minimum call drops. It uses the tail() function, which returns the last n elements of a vector. In this case, n is 1, so it retrieves the last state name from the sorted sorted_call_drops_by_state data frame, which represents the state with the least call drops.

2. Printing State Information:

- cat(...): This line prints a message to the console, informing the user about the state with the minimum call drops. It combines text and the extracted state name using string concatenation.

```
The region with the minimum call drops is: Uttar Pradesh
```

Fig 10 shows us comparison of the number of satisfied customers across different operators, providing insights into customer sentiment towards each operator.

CODE

```
# Filter dataset for positive sentiment
positive_dataset <- dataset[dataset$sentiment == "Positive", ]

# Calculate counts of positive sentiment for each operator
positive_counts <- table(positive_dataset$operator)

# Convert counts to dataframe
positive_df <- as.data.frame(positive_counts)
positive_df$operator <- rownames(positive_df)
colnames(positive_df) <- c("Operator", "PositiveCount")

# Sort dataframe by PositiveCount in descending order
positive_df <- positive_df[order(-positive_df$PositiveCount), ]

# Create bar chart with different colors for each operator
bar_chart <- ggplot(data = positive_df, aes(x = reorder(Operator, PositiveCount), y = PositiveCount, fill = Operator)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = rainbow(length(unique(positive_df$Operator)))) +
# Assigns different colors to each operator
  xlab("Operator") +
```

```
  ylab("Count of Satisfied Customers") +
  ggtitle("Operator-wise Distribution of Satisfied Customers") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display the bar chart
print(bar_chart)
```

EXPLANATION

1. Focus on Positive Feedback:

It isolates customer feedback classified as "Positive" to analyze customer satisfaction.

2. Count Positive Sentiment per Operator :

It tallies the number of positive sentiments associated with each mobile network operator.

3. Organize Data:

It creates a dataframe to neatly structure the operator names and their corresponding positive sentiment counts.

4. Rank Operators:

- It sorts the operators based on the number of positive sentiments they have, revealing those with the highest customer satisfaction.

5. Visualize Distribution:

- It generates a colourful bar chart to visually depict the distribution of positive sentiment across different operators.

- Each bar represents an operator, with its height indicating the number of positive sentiments.

- Operators are arranged from those with the most positive feedback to the least.

- The bars are colour-coded for easy visual distinction between operators.

6. Display Chart:

It presents the chart for visual analysis, allowing you to quickly grasp which operators have the most satisfied customers.

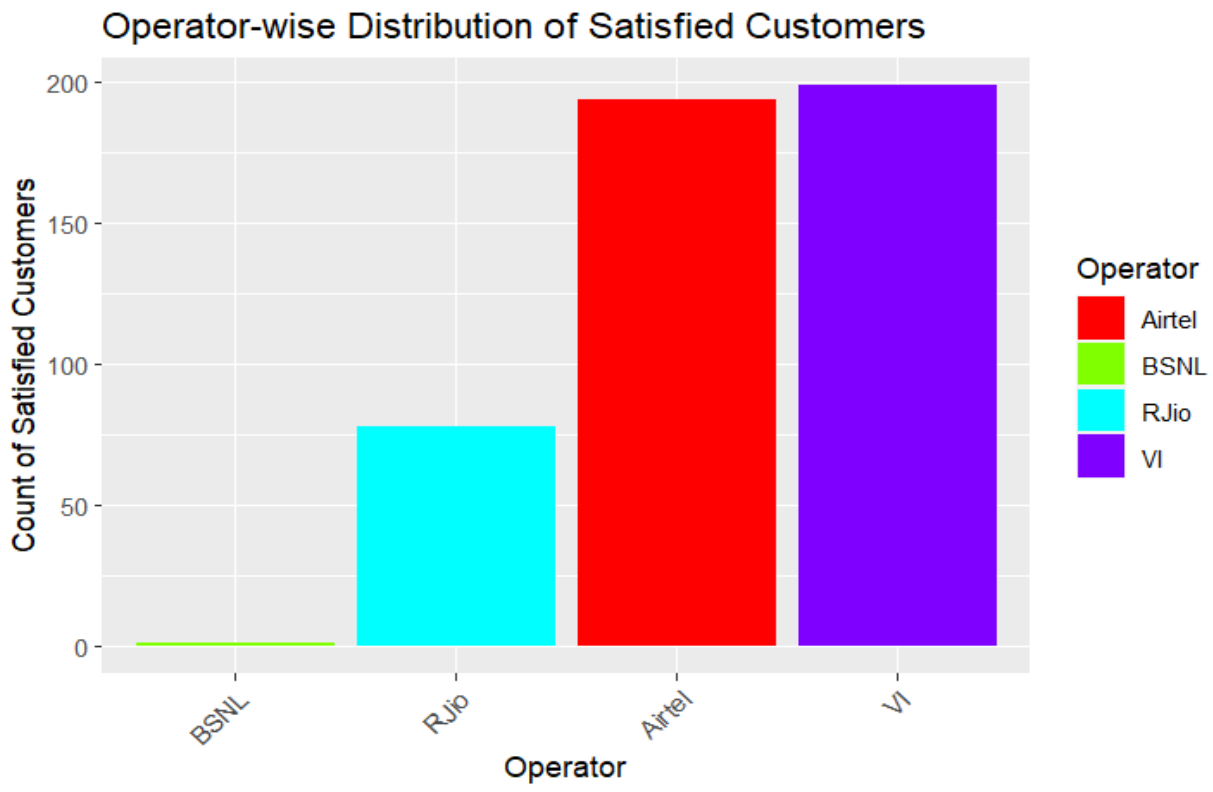


Fig 10

It is concluded that Vodafone Idea has the most satisfied customer from the given record and there is significant relationship between the call drops and satisfaction of the customers.

The analysis yields significant insights crucial for telecom operators aiming to elevate service quality, refine network infrastructure, and amplify customer satisfaction levels. It emphasizes the pivotal role of data-driven strategies in tackling call drop concerns and elevating service delivery standards for consumers. This underscores the paramount importance of leveraging data analytics to inform decision-making processes, enabling operators to proactively address issues, optimize operational efficiencies, and ultimately enhance the overall customer experience.

SUMMARY ANALYSIS

1. **Data Cleaning and Preprocessing:**

    - The dataset was loaded and processed to handle missing values in the network_type column by replacing them with the mode value.

    - The dataset was encoded, and categorical variables were converted to factors.

    - The dataset was split into training and testing sets for model training and evaluation.

2. **Random Forest Model:**

    - A random forest model was trained using the randomForest package to predict the call drop category based on features such as operator, inout_travelling, rating, latitude, longitude, and state_name.

    - The model achieved an overall accuracy of approximately 94.74% on the test set, indicating good predictive performance.

3. **Sentiment Analysis:**

    - Sentiment analysis was performed based on the rating and calldrop_category variables.

    - Positive sentiment was assigned to calls rated as "Satisfactory," while negative sentiment was assigned to calls categorized as "Poor Voice Quality" or "Call Dropped."

- A pie chart was created to visualize the distribution of positive and negative sentiment calls.

4. **Operator-wise Analysis:**

   - The dataset was filtered by each operator (Airtel, VI, BSNL, RJio), and satisfaction rating distributions were analyzed for each operator.

   - Bar charts were created to visualize the distribution of satisfaction ratings for each operator individually.

5. **Network Type Analysis:**

   - The percentage of call drops was calculated for each network type, and a bar chart was created to visualize the percentage of call drops by network type.

6. **State-wise Analysis:**

   - The count of call drops was calculated for each state, and the dataset was sorted by the call drops count in descending order.

   - The state with the maximum call drops was identified and printed.

   - The state with the minimum call drops was identified and printed.

In nutshell, the analysis provides insights into call drop patterns, operator performance, sentiment distribution, and state-wise variations in call drops. It helps in understanding the factors influencing call quality and identifying areas for improvement in network infrastructure and service provision.

RECOMMENDATION

Based on the analysis of the data, specifically the distribution of call drop incidents and customer sentiments across different operators, we can make the following recommendations to the rest of the service providers:

1. **Address Call Drop Issues**: Identify the root causes of call drop incidents and implement measures to reduce them. This could involve improving network infrastructure, optimizing signal strength, and enhancing network coverage in areas with frequent call drops.

2. **Focus on Customer Satisfaction**: Prioritize customer satisfaction by addressing issues related to call quality and network performance. Conduct regular surveys or feedback mechanisms to understand customer experiences and address their concerns promptly.

3. **Learn from Successful Operators**: Analyze the strategies and practices of operators with higher customer satisfaction and fewer call drop incidents. Identify best practices that can be replicated or adapted to improve services.

4. **Invest in Network Improvement**: Allocate resources to upgrade and modernize network infrastructure, including adopting advanced technologies like 5G and expanding coverage areas to ensure better service quality and reliability.

5. **Enhance Communication Channels**: Improve communication channels with customers to provide timely updates on network maintenance, service disruptions, and resolution of issues. Transparency and proactive communication can help build trust and loyalty among customers.

6. **Offer Value-added Services**: Explore offering value-added services or incentives to customers, such as discounted data plans, free voice calls, or access to exclusive content, to enhance the overall customer experience and differentiate from competitors.

7. **Continuous Monitoring and Improvement**: Implement a robust system for monitoring network performance, customer feedback, and market trends. Continuously analyze data to identify areas for improvement and take proactive measures to address emerging issues.

By implementing these recommendations, service providers can strive to improve their network reliability, enhance customer satisfaction, and differentiate themselves in a competitive market landscape.

CONCLUSION

The random forest model demonstrates high accuracy in predicting call drop categories, indicating its effectiveness for call quality analysis.

Operator-wise analysis reveals variations in satisfaction ratings among different operators, highlighting areas for improvement.

Network type analysis helps identify which types of networks are associated with higher call drop rates, allowing for targeted interventions.

State-wise analysis provides insights into geographical disparities in call drop occurrences, aiding in resource allocation and network optimization efforts.

Overall, the analysis provides valuable insights for telecom operators to improve service quality, optimize network infrastructure, and enhance customer satisfaction. It underscores the importance of data-driven decision-making in addressing call drop issues and delivering better services to consumers.

LIMITATION

While the provided R code conducts a thorough analysis of call drop data, there are several limitations and considerations to be aware of:

1. Data Limitations:

The analysis heavily depends on the quality and representativeness of the dataset. If the dataset is not a true reflection of the entire user base or if it contains biased data, the conclusions drawn may not be applicable to the broader population.

2. Missing Data Handling:

The code replaces missing values in the 'network_type' column with the mode. This imputation method assumes that the missing values are missing completely at random (MCAR). If the missingness is not MCAR, it may introduce bias into the analysis.

3. Imputation of 'Unknown' Values:

The code replaces 'Unknown' values in the 'network_type' column with the mode. This assumes that 'Unknown' values are equivalent to missing values, which might not be accurate. The imputation strategy needs to be justified based on the domain knowledge.

4. Chi-Square Test Assumptions:

The chi-square test is sensitive to sample size, and with large sample sizes, even small differences may become statistically significant. While the p-value suggests a significant association between 'inout_travelling' and 'calldrop_category', it's essential to consider the practical significance and the size of the effect.

5. Model Generalization:

The random forest model's high accuracy on the training set does not guarantee its generalization to new, unseen data. Overfitting may occur, and the model might not perform as well on real-world data.

6. Sentiment Analysis Assumptions:

The sentiment analysis assumes that ratings and call drop categories are reliable indicators of user sentiment. However, sentiment is subjective and may be influenced by various factors not captured in the dataset.

7. Operator-wise Analysis:

The operator-wise analysis assumes that customer satisfaction is solely influenced by the operator, neglecting other potential factors such as location, network type, or device.

8. Network Type Analysis:

The analysis of call drops by network type assumes that network type directly correlates with call drop issues. Other factors like network congestion, maintenance, or device-related issues could also contribute to call drops.

9. State-wise Analysis:

The state-wise analysis is based on call drops without considering factors like population density, network infrastructure, or regional variations, which might influence the results.

10. Limited Scope of Analysis:

The analysis focuses on specific aspects (e.g., satisfaction, call drops) without exploring other potential factors influencing call quality, such as weather conditions, network load, or hardware issues.

11. Satisfaction Rating Categories:

The satisfaction rating categories are predefined and may not fully capture the nuanced sentiment of users. The subjective interpretation of categories like 'Satisfactory' may vary.

12. Visualizations and Interpretation:

- The interpretation of visualizations and charts may be subjective. It's important to provide clear explanations of the chosen visualizations and their implications.

Addressing these limitations would require a more in-depth understanding of the data, rigorous statistical methods, and consideration of a broader set of variables that may impact call quality and user satisfaction.

* * * *

REFERENCES

1. National Telecommunications and Information Administration. (2022). "Report on Telecommunications Service Quality: December 2022." Retrieved from [https://data.gov.in/]

2. R Core Team. (2022). "R: A Language and Environment for Statistical Computing." R Foundation for Statistical Computing. Retrieved from [https://posit.co/download/rstudio-desktop/]

3. Wickham, H. (2016). "ggplot2: Elegant Graphics for Data Analysis." Springer. DOI: [DOI]

4. Kuhn, M. (2022). "caret: Classification and Regression Training." R package version x.x-x. Retrieved from [https://cran.rstudio.com/]

5. Breiman, L. (2001). "Random Forests." Machine Learning, 45(1), 5-32. DOI: [DOI]

6. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). "An Introduction to Statistical Learning." Springer. DOI: [DOI]

7. Wickham, H., & Grolemund, G. (2017). "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data." O'Reilly Media. DOI: [DOI]

8. Geeks for Geeks data mining Process [https://www.geeksforgeeks.org/data-mining-process/]