



NARNARAYAN SHASTRI INSTITUTE OF TECHNOLOGY
INSTITUTE OF FORENSIC SCIENCES & CYBER SECURITY

AFFILIATED TO

NATIONAL FORENSIC SCIENCES UNIVERSITY
(INSTITUTE OF NATIONAL IMPORTANCE, MHA, GOVT. OF INDIA)



PROJECT REPORT

ON

“End to End Data Processing and Analytics”

Submitted To

School of Cyber Security & Digital Forensics,

National Forensic Sciences University

For partial fulfilment for the award of degree

MASTER OF SCIENCE

In

CYBER SECURITY

Submitted by

Dream Johnson

(002300105001009)

FORENSIC SCIENCES AND CYBER SECURITY

Under the Supervision of

**Ms. Divya Patel (School of Cyber Security & Digital
Forensics)**

**National Forensic Sciences University, Gandhinagar
Campus,**

Gandhinagar-382009, Gujrat, India

APRIL 2025

DECLARATION

I certify that

- a. The work contained in the dissertation **“End to End Data Processing and Analytics”**, is original and has been done by me under the supervision of **“Ms. Divya Patel”**.
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references.
- e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.
- f. From the plagiarism test, it is found that the similarity index of whole dissertation within 25% and single paper is less than 10 % as per the university guidelines.

Date: 5th May 2025

Place: Jetalpur

Dream Johnson

Enrollment No: 002300105001009

Narnarayan Shastri Institute of Technology

Institute of Forensic Science & Cyber Security

M.Sc. Cyber Security, Semester-4th

CERTIFICATE

This is to certify that the work contained in the dissertation entitled “**End to End Data Processing and Analytics**”, submitted by **Dream Johnson (002300105001009)** for the award of the degree of **Master of Science in Cyber Security** to the **National Forensic Sciences University, Gandhinagar Campus**, is a record of bonafide research works carried out by him under my direct supervision and guidance.



CERTIFICATE

It is certified that **Mr. Dream Johnson** has worked for the dissertation in M.Sc. Cyber Security, as a bona fide student of the **Narnarayan Shastri Institute of Forensic Sciences and Cyber Security, affiliated with National Forensic Sciences University**, Gandhinagar, Gujarat, during the fourth Semester from January 2025 to April 2025. The dissertation work was carried out by the student under the guidance of **Ms. Divya Patel**, Assistant Professor, NSIT-IFSCS.



ACKNOWLEDGEMENTS

I am truly grateful to my supervisor, Ms. Divya Patel, for their professional guidance, motivation, and precious feedback throughout this study. My appreciation also goes to the Head of the Department, Mr. Aakash Khunt, and faculty members of MSc Cyber Security for their academic guidance and mentorship.

I thank my lab group members at Narnarayan Shastri Institute of Technology for their cooperation, valuable discussions, and friendship.

I thank the technical and administrative staff of Narnarayan Shastri Institute of Technology for providing hassle-free access to resources. Lastly, gratitude to my family members and friends for their constant encouragement and patience.

This achievement is a team effort, and I still owe gratitude to all those who helped.

With Sincere Regards,

Dream Johnson

M.Sc.Cyber Security

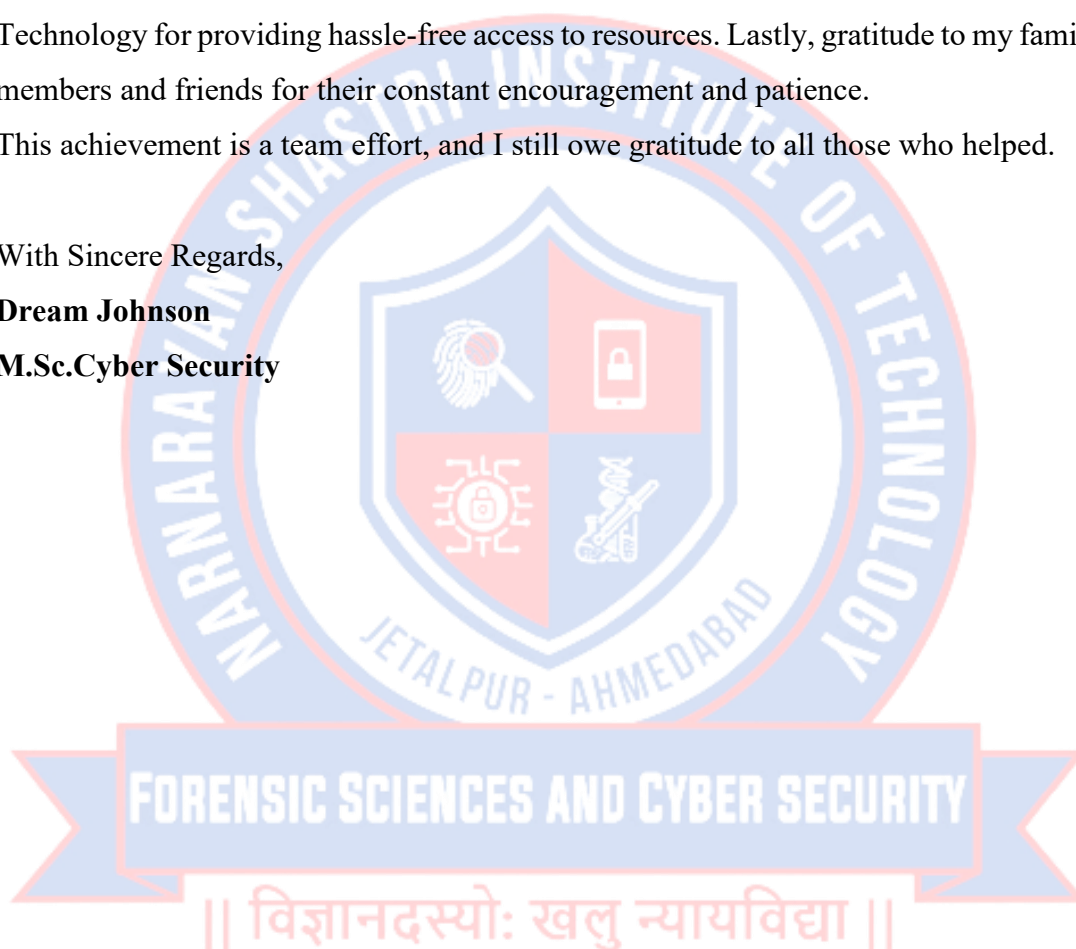


TABLE OF CONTENTS

Declaration		i
Certificate		ii
Certificate		iii
Acknowledgement		iv
List of Figures		ix
Abstract		
Chapter -1	INTRODUCTION	1-3
1.1	Project Profile	2
1.2	Scope of Work	2
1.3	Tools-Technologies Requirements	2
1.4	Languages	3
1.5	Libraries	3
Chapter-2	LITERATURE REVIEW	4-12
2.1	Exploring the Significance of Statistics in Research	5
2.2	Use of Statistics in Research	5
2.3	Basic statistical tools in research and data analysis	6
2.4	Importance of statistics to data science	7
2.5	Exploratory Data Analysis	8
2.6	Role of Exploratory Data Analysis in Data Science	9
2.7	What is Exploratory Data Analysis	9
2.8	Feature Engineering Tools and Techniques for Better Classification Performance	10
2.9	The implications of Statistical analysis and feature engineering for model building using Machine Learning Algorithms	11
2.10	An Empirical Analysis of Feature Engineering for Predictive Modeling	12
Chapter-3	Statistics in Data Analysis	13-16
3.1	Importance of Statistics in Data Analysis	14
3.2	Descriptive Statistics in Data Analysis	14
3.3	Feature Engineering	15

Chapter 4	Data Analysis Tools	17-19
4.1	SQL (Structured Query Language)	18
	4.1.1 Purpose	
	4.1.2 Role in Feature Engineering	
4.2	Python	18
	4.2.1 Purpose	
	4.2.2 Role in Feature Engineering	
4.3	Microsoft Excel	18
	4.3.1 Purpose	
	4.3.2 Role in Feature Engineering	
4.4	Power BI	19
	4.4.1 Purpose	
Chapter-5	Exploratory Data Analysis	20-30
5.1	Exploratory Data Analysis (EDA)	21
5.2	Steps Involved in EDA	21
Chapter-6	Storage Platforms and Tools	31-44
6.1	Amazon Web Services (AWS)	32
6.2	Snowflake	33
6.3	Power BI	37
Chapter-7	Impact and Future Scope	45-46
Conclusion		47
References		48

FORENSIC SCIENCES AND CYBER SECURITY

॥ विज्ञानदस्योः खलु न्यायविद्या ॥

LIST OF FIGURES

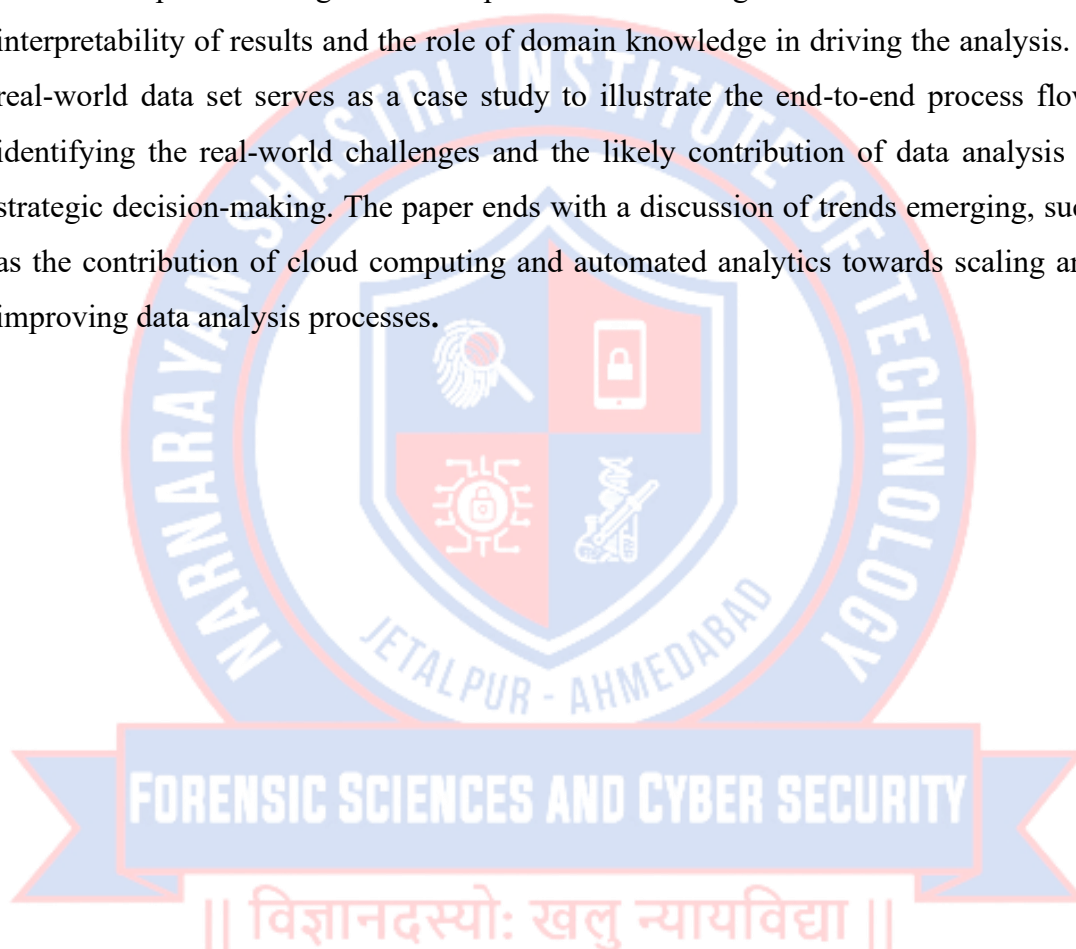
	Chapter-3	Page No.
Figure 3.1	Key Objectives of Statistical data analysis	15
Figure 3.2	Feature Engineering in Data Analysis	16
	Chapter-4	
Figure 4.1	Data Flow	19
	Chapter-5	
Figure 5.1	Features of the dataset	22
Figure 5.2	Importing libraries	23
Figure 5.3	Sampling the dataset	23
Figure 5.4	Info about Dataset	24
Figure 5.5	Describing the dataset	25
Figure 5.6	Splitting values for columns	25
Figure 5.7	Splitted columns	26
Figure 5.8	Describing the dataset	26
Figure 5.9	Changing datatype	26
Figure 5.10	Changed datatype	27
Figure 5.11	Dropping columns	27
Figure 5.12	Splitting Hours and minutes	28
Figure 5.13	Splitted columns into hours and minutes	28
Figure 5.14	Changing datatype	28
Figure 5.15	Conforming datatype	28
Figure 5.16	Viewing dataset	29
Figure 5.17	Viewing datatypes	29
Figure 5.18	Unique values	30
Figure 5.19	Viewing null values	30
Figure 5.20	Unique values	30

Chapter-6

Figure 6.1	Amazon S3 Bucket in Data Analysis	32
Figure 6.2	Snowflake in Data Engineering	33
Figure 6.3	Snowflake dashboard	34
Figure 6.4	Creating bucket	34
Figure 6.5	Uploading file to bucket	34
Figure 6.6	Creating role	35
Figure 6.7	ARN value on snowflake	35
Figure 6.8	ARN values in Amazon S3	35
Figure 6.9	Creating Schema	36
Figure 6.10	Creating stage	36
Figure 6.11	Copying data set into PBI_UPI_Dataset	37
Figure 6.12	Viewing dataset on snowflake	37
Figure 6.13	Importing dataset from snowflake	38
Figure 6.14	Credential Screen	38
Figure 6.15	Uploaded dataset in Power BI	39
Figure 6.16	Transforming data	40
Figure 6.17	Transformed column	40
Figure 6.18	Transforming column (2)	41
Figure 6.19	Transformed column (2)	41
Figure 6.20	Transforming columns (3)	42
Figure 6.21	Transformed columns (3)	42
Figure 6.22	Attributes from dataset	43
Figure 6.23	Visualization	43
Figure 6.24	Attributes used for visualization	43
Figure 6.25	Selecting custom attributes	44
Figure 6.26	Visuals according to custom attributes	44

ABSTRACT

In the age of big data, successful data analysis is critical to deriving useful insights and informing data-driven decision-making in numerous fields. This paper is a thorough examination of contemporary data analysis methods, combining statistical approaches, machine learning algorithms, and visualization tools to analyze and interpret large datasets. We cover the data preprocessing pipeline, such as data cleaning, transformation, and feature selection, followed by comparative assessment of analytical models for pattern recognition and predictive modeling. There is a focus on the interpretability of results and the role of domain knowledge in driving the analysis. A real-world data set serves as a case study to illustrate the end-to-end process flow, identifying the real-world challenges and the likely contribution of data analysis in strategic decision-making. The paper ends with a discussion of trends emerging, such as the contribution of cloud computing and automated analytics towards scaling and improving data analysis processes.



Keywords - Data, Collection, Cleaning, EDA, SQL, AWS, Snowflake



1.1 Project Profile

Analysis of data is key to converting unprocessed data into insightful information guiding well-informed decisions. Organizations and researchers are increasingly utilizing analytical methods for hidden pattern and trend identification in the current age of data reliance, making fact-based predictions. Data analysis comprises various activities: collection, cleansing, exploration, modelling, and interpretation.

This project follows a formal methodology to analysis of data, starting from the data acquisition and pre-processing to get quality and consistent data. Exploratory Data Analysis (EDA) is utilized to provide a summary of the overall features of the data set, mostly using visual techniques to facilitate comprehension of intricate relationships. Based on the study aims, statistical methods and machine learning algorithms are used to analyze the data, identify patterns, and make predictions.

The value of data analysis in the project is that it offers actionable insights and confirms hypotheses on the basis of empirical evidence. In addition, the use of contemporary tools like Python, SQL, and data visualization libraries improves the analytical process in terms of efficiency, reproducibility, and interpretability.

This report segment describes the research methodology used when carrying out data analysis and serves as the framework for interpreting findings in the light of the broader project goals.

1.2 Scope of Work

- Learn Data Cleaning
- Learn SQL
- Learn Power BI
- Use cleaned dataset to create visualizations.

1.3 Tool- Technologies Requirements

- Google colab
- jupyter notebook
- VSCode
- PowerBI
- SQL

- Snowflake
- AWS S3

1.4 Language

- Python
- SQL
- DAX
- Power Query

1.5 Libraries

- Pandas
- Matplotlib.pyplot
- Numpy
- Seaborn





D P Singh, J S Jassi, Sunaina, September 16, 2024

2.1 Exploring the Significance of Statistics in Research

Statistics is a fundamental tool used for the analysis of quantitative data, making inferences, and facilitating accurate predictions, hence a critical component in evidence-based decision-making in both engineering and non-engineering fields. This paper consolidates past research on the use of statistical tools, with a focus on their applicability in policymaking, scientific investigation, and trend prediction. Significant contributions comprise an overview of methodologies which employ statistical methods to test relationships between variables, refine results, and promote research reliability. The use of current computational tools—like Python and machine learning (ML) is recognized as a revolutionary boost that can perform scalable analysis on big data and enhance the precision of estimates and predictions. The authors highlight the importance of choosing proper statistical methods to support rigor in collecting, analyzing, and interpreting data, which is important for creating generalized results. Although quantitative methods prevail as they can cope with structured data, qualitative evidence and algorithmic methods supplement the approaches, presenting an integrated analysis of intricate problems. The work also recognizes strengths and weaknesses of statistical methods, calling for domain-independent frameworks ensuring methodological parity across disciplines.

Finally, the research confirms the necessity of statistical literacy in research, contending that knowledge of statistical methods and newer computational methods is imperative for the verification of findings, pushing innovation, and ensuring sound solutions in both applied fields and academic. This integration sits within wider scholarly debate around the changing role of statistics in a data-driven research era.

Dr. Seema Amit Agarwal, 09 November 2021

2.2 Use of Statistics in Research

Statistics is central to research in that it offers methodological techniques for study design, data analysis, and interpretation. It responds to the issue of handling large raw data through systematic classification, tabulation, and the construction of summary indices, allowing researchers to condense complex datasets into usable information. Two fundamental branches support this process: *descriptive statistics*, which summarizes data into understandable measures (e.g., means, distributions), and

inferential statistics, which extrapolates results from samples to larger populations, considering uncertainty and error. Statistical methods are applied correctly to ensure reliability at each step—data collection, analysis, and conclusion—while reducing biases. Outside of academics, statistical literacy is essential in all fields, from scientific investigation to policy, industry, since it equips professionals with the ability to cross-check findings, construct sound studies, and vet data-based claims.

Even individuals without special statistical training can apply basic statistical competence to critically utilize research products. On the other hand, statisticians are highly demanded across varied industries (government, healthcare, technology) for improving methodologies as well as confirming analytical rigor. Therefore, statistics is both a basic framework for empirical investigation and an interface between theoretical research and practical application, reinforcing its inescapable contribution to evidence-based knowledge. This integration emphasizes statistics' dual role as a tool for accuracy and a discipline for interdisciplinary collaboration, a requirement of any master's-level research undertaking.

Ali, Zulfiqar & Bhaskar, SBala. September (2016)

2.3 Basic statistical tools in research and data analysis

Statistical techniques form the core of well-defined research, informing each stage of a study right from preliminary planning and design to the collection of data, analysis, interpretation, and reporting of results. Statistical techniques turn unstructured raw data into useful information, allowing researchers to make informed conclusions and assist in evidence-based decision-making. A solid grounding in fundamental statistical principles is essential, such as the difference between quantitative (numeric) and qualitative (categorical) variables, measures of central tendency (e.g., mean, median, mode) summarizing data distributions, and principles of sample size estimation and power analysis to provide assurance that studies are sufficiently powered to detect substantial effects. No less crucial is an awareness of potential statistical errors, Type I (false positives) and Type II (false negatives) errors, that undermine research findings' validity.

The selection of statistical tests, parametric (e.g., t-tests, ANOVA) or non-parametric (e.g., Mann-Whitney U, Wilcoxon tests), should be based on the type of data and research hypotheses to make inferences that are valid. Misuse of such methods, for instance, applying inappropriate tests or omitting assumptions such as normality and

homogeneity of variance, may result in wrong conclusions, which destroy the validity of the study. Poor statistical practices not only distort results but also disseminate unethical research results because incorrect conclusions can impact clinical practice guidelines or policy decisions, ultimately damaging public confidence in scientific research.

On the other hand, familiarity with statistical principles improves study designs' strength by providing reproducibility and reliability. For example, meticulous calculation of sample size reduces wastage of resources and maximizes the chances of detecting the true effect, while power analysis prevents underpowered studies that may ignore relevant findings. Clear reporting of statistical techniques and outcomes further enhances research credibility, allowing peer criticism and replication. In medical research, good statistical application is a necessity for producing translatable evidence that can inform clinical practice and public health policy.

Therefore, researchers need to place great emphasis on statistical literacy to understand the intricacies of data analysis, avoid biases, and ensure ethical standards. Incorporating rigorous statistical training in academic curricula and professional development programs is critical to develop a culture of methodological excellence. By cultivating a stronger appreciation of these tools, scientists can improve the quality of studies, minimize avoidable mistakes, and help bring about the evolution of reliable, actionable knowledge to guide innovation and enhance outcomes in evidence-based practice. This synthesis highlights the role of statistics in closing the divide between empirical inquiry and practical application.

Jalajakshi V, Myna A N, 2 April 2022

2.4 Importance of statistics to data science

This research emphasizes the key contribution of statistics in the development of data science as a seminal support for solving real-world issues with extensive data processing. The research elucidates how statistical approaches facilitate sound data analysis at significant steps, ranging from data collection to optimization, interpretation, modeling, testing, and visualization. Through combining quantitative statistical metrics, data scientists are able to improve precision in predictive modeling, pattern detection, and decision-making. The paper points out that statistical methods—like hypothesis testing, regression analysis, and probability theory—supplement computational algorithms, making it possible to strike a balance when solving intricate

problems. The study also highlights challenges of using statistics with data science, including handling data heterogeneity and model interpretability.

It contends that statistics fills the gaps in data science by bringing mathematical rigor to managing large sets of data, especially in real-time applications where projections must be precise. Though other fields (e.g., computer science) support data science, the paper asserts that statistics is crucial in performing activities such as data enrichment, sophisticated modeling, and arriving at actionable conclusions. In conclusion, the integration of statistical approaches and computational software is shown to be crucial for maximizing data-driven solutions, further validating statistics as an indispensable aspect in the development of data science research and applications. This examination sets statistics not only as a supporting tool but as the fundamental framework that makes data science capable of changing raw data into interpretable, scalable results.

**Matthieu Komorowski, Dominic C. Marshall, Justin D. Saliccioli
and Yves Crutain, 2016**

2.5 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an important first step in research processes, especially in discovering structural patterns, outliers, and distributions in data to guide hypothesis development and testing. Developed by John Tukey in 1977, EDA focuses more on graphical methods—like plots and visualizations—than on formal statistical inference, allowing researchers to ask questions of data freely and naturally. Howard Seltman (Carnegie Mellon University) defines EDA as including all the non-inferential methods, differing from confirmatory analysis through its absence of strict assumptions and focus on open-ended discovery. After data collection and data preprocessing, EDA allows for quality checking, feature selection, and initial model building by exploiting tools such as histograms, scatterplots, and boxplots to uncover trends, outliers, and relationships. Its visual emphasis leverages human pattern detection, enabling analysts to extract insights that might be missed by quantitative approaches alone.

While universally applicable across fields, EDA is especially important in healthcare analytics, including electronic health record (EHR) research, where high-dimensional, complex datasets need to be scrutinized exhaustively to detect clinically significant variables or create new hypotheses. For instance, visualizing treatment outcomes or patient demographics using EDA can reveal disparities, data input mistakes, or surprise correlations, informing later statistical modeling. The strength of the methodology is its

flexibility: it supports a variety of data types (e.g., spatial, temporal, categorical) and scales well across small to big data sizes. Informal though it is, EDA is still a gold standard for first-pass data inspection, bridging raw data and organized analysis. By combining visualization with simple statistical summaries, it enables researchers to sharpen questions, test assumptions, and streamline analytical pipelines prior to investing in resource-hungry modeling.

In short, EDA's long-term value lies in its double function as a diagnostic tool (e.g., identifying data flaws) and a generative model for hypothesis formulation. Its blending of creativity and rigor renders it invaluable in contemporary data-driven research, especially in areas such as healthcare, where strong preliminary analysis is crucial to ethical and accurate conclusions.

Dr. A Suresh Rao, Dr. B. Vishnu Vardhan, Hafeezuddin Shaik, 2021

2.6 Role of Exploratory Data Analysis in Data Science

Exploratory Data Analysis (EDA) is an integral but oft-underappreciated step in data science that is essential for taking raw, heterogeneous, and usually biased data and turning it into useful information. Though indispensable in improving the accuracy of models, detecting business-critical patterns, and informing resource-effective workflows, practitioners too often skip EDA to get projects done on time. This omission risks significant downstream consequences, including flawed algorithmic predictions, increased rework, and elevated operational costs due to inefficient resource allocation. Surveyed literature underscores that EDA's integration of intuitive visualization, statistical techniques, and specialized tools (e.g., Python libraries like Pandas, Matplotlib, or R packages) ensures robust data understanding, directly influencing model performance and business decision-making. Overlooking EDA not only undermines analytical accuracy but also organizational effectiveness, highlighting the importance of giving priority to this stage to reduce risks and maximize results in data-driven initiatives.

2.7 What is Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an important phase in the data science process that involves understanding and summarizing the primary features of a dataset using visualization, statistical methods, and data manipulation. The article highlights that EDA is not a structured process but an iterative, creative process to discover patterns,

identify anomalies, test hypotheses, and verify assumptions prior to using sophisticated models. Some of the most important techniques are examining data distributions, detecting outliers, missing value handling, and correlation between variables. Python libraries (Pandas, Matplotlib, Seaborn) and R packages (ggplot2, dplyr) are popularly used to create histograms, box plots, scatter plots, and heatmaps, which help in visualizing trends and relationships. The article emphasizes EDA's function in directing feature engineering, model selection, and preprocessing choices to ensure strong and reliable results. Skipping EDA potentially disregards data quality problems (e.g., skewness, unnecessary features) that result in defective models and erroneous conclusions. By rigorously investigating data, EDA closes the gap between raw data and meaningful insights, thus being the key to effective and accurate data-driven decision-making in business, healthcare, and research.

Tara Rawat, Dr. Vineeta Khemchandani, April 2017

2.8 Feature Engineering Tools and Techniques for Better Classification Performance

Exploratory Data Analysis (EDA) is a crucial step in the data science pipeline aimed at grasping and summarizing the key features of a dataset by visualizing, applying statistical methods, and manipulating the data. The paper stresses that EDA is not a systematic process but a iterative and creative process in discovering patterns, identifying anomalies, checking hypotheses, and verifying assumptions prior to using sophisticated models. Key techniques include analyzing data distributions, identifying outliers, handlingThe surveyed paper presents a comprehensive overview of feature engineering methodologies, tools, and techniques aimed at enhancing classifier accuracy in machine learning. It underscores the critical role of feature engineering in addressing the limitations posed by raw data, particularly in domains such as text classification, clinical text analysis, social network link prediction, fraud detection, and knowledge base construction. The work focuses on two fundamental processes: (1) feature engineering, which creates more relevant features from available data sets, and (2) feature selection, which removes redundant or irrelevant features to achieve an independent minimal set of features representing underlying data patterns. Experimental results on various applications prove that engineered features achieve dramatically better model performance than non-engineered methods, in terms of complexity reduction and accuracy enhancement. Yet, the paper emphasizes an

ongoing challenge: repeated testing of feature inputs to determine their usefulness in model training. This survey places feature engineering as an essential step toward optimizing predictive algorithms, promoting systematic methods to reconcile feature richness with computational efficiency. Its conclusions concur with more extensive literature regarding the need for domain-specific feature engineering approaches to handle complexities and enhance generalizability in machine learning pipelines. Missing values, and correlations between variables are typically investigated using tools such as Python libraries (Pandas, Matplotlib, Seaborn) and R packages (ggplot2, dplyr) to create histograms, box plots, scatter plots, and heatmaps, which help visualize trends and relationships. The article is pointing out the importance of EDA in directing feature engineering, model choice, and preprocessing, leading to strong and stable results. Omitting EDA can lead one to overlook data quality problems (e.g., biased distributions, irrelevant features) affecting resultant models and conclusions. By rigorously investigating data, EDA closes the gap between raw data and informative results, thereby making it a necessity for effective and precise data-driven decision-making in fields such as business, healthcare, and research.

Swayanshu Shanti Pragnya and Shashwat Priyadarshi

2.9 The implications of Statistical analysis and feature engineering for model building using Machine Learning Algorithms

This paper highlights the important role that statistical analysis and feature engineering have in improving predictive accuracy in machine learning models. Highlighting the importance of correlation significance, heat map visualization, and feature selection methods, authors show how consistent data preprocessing—in this case, the reduction of datasets to 9 optimal features—can advance model performance. By contrasting logistic regression (accuracy of 0.70) with K-nearest neighbor (KNN) algorithms (accuracy of 0.78 after optimization), the research points to an 11.42% accuracy improvement, emphasizing the significance of feature selection and dimensionality reduction in model performance. Major contributions involve methodological understanding of how statistical techniques (e.g., analysis of categorical data distribution) can be combined with machine learning processes to improve model results. But the small boost implies that other aspects apart from feature engineering—like hyperparameter tuning or balancing the data—could yet further increase accuracy. The work situates feature engineering as an essential initial step in model creation and

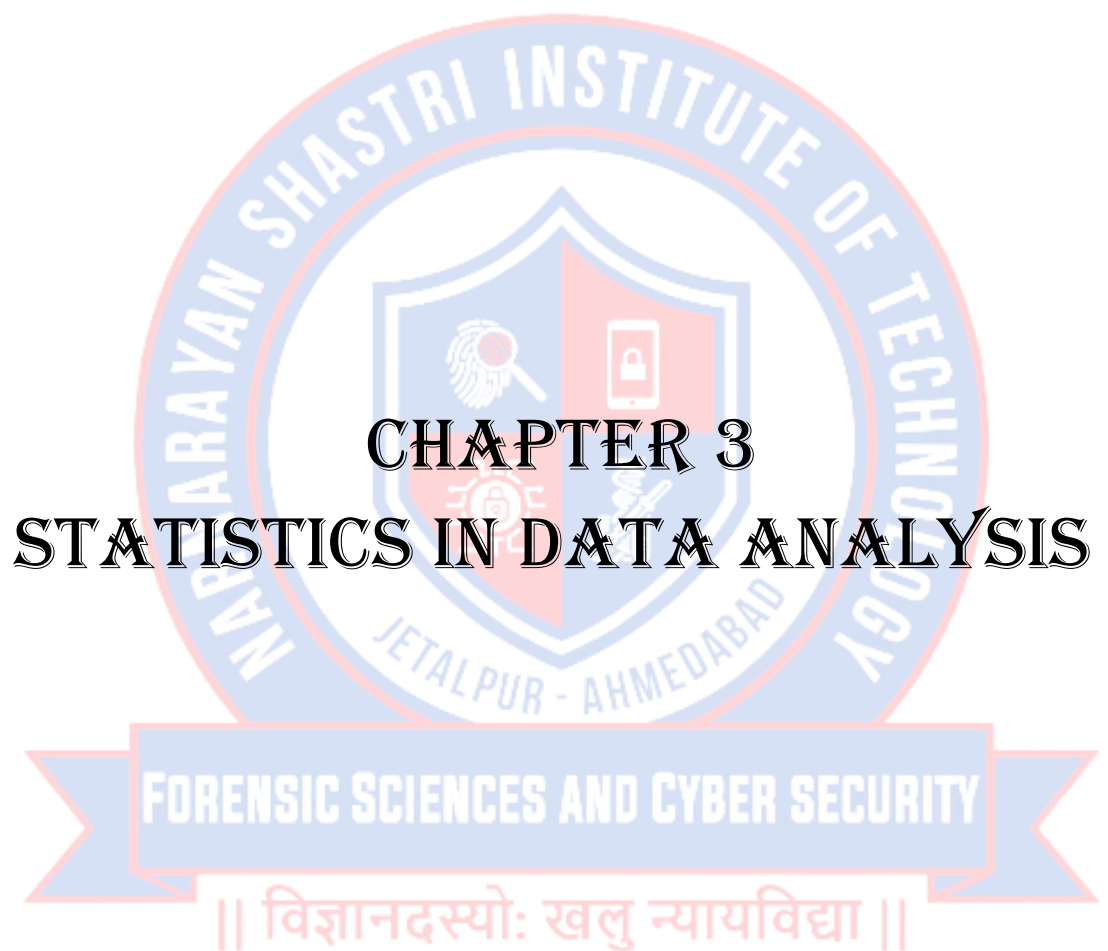
calls on subsequent studies to find additional optimization methods. The research is aligned with general debate within data science on preprocessing's effect and challenges further investigation into hybrid methods for more efficient predictive systems.

Jeff Heaton, 1 November 2020

2.10 An Empirical Analysis of Feature Engineering for Predictive Modeling

This empirical work investigates the interaction between feature engineering and model performance in the case of neural networks (NNs), support vector machines (SVMs), random forests (RFs), and gradient boosting machines (GBMs). Through experimentation with specially crafted datasets that bias towards particular engineered features, the authors show that model architectures differ considerably in their capacity to independently synthesize useful features. The NNs and SVMs revealed co-interpretable feature preferences tending to rely on analogous engineered inputs, but RFs and GBMs dominated with a disparate set of features, which infers model-sensitive optimization tactics. Interestingly, features based on ratios (e.g., differences in ratios) weren't adequately crafted by any given model, affirming the value of traditional human feature engineering on such occasions. The research also showcases the opportunity of heterogeneous ensembles (e.g., NNs/SVMs paired with RFs/GBMs) in exploiting complementary feature synthesis abilities, amplifying performance. Although generic hyperparameters were employed to focus on generalizability over optimized tuning, the research emphasizes the need for coordinated feature engineering practice with model structure and promotes deliberate ensemble design to take advantage of varied feature representations. These findings offer actionable advice for optimizing feature engineering processes and model choice in machine learning systems.

॥ विज्ञानदस्योः खलु न्यायविद्या ॥



3.1 Importance of Statistics in Data Analysis

Statistics is the backbone of data analysis, giving a systematic approach to gathering, organizing, interpreting, and presenting data in an effective manner. In data analysis, statistics aid in converting raw data into meaningful information by condensing data patterns, recognizing trends, and validating evidence-based conclusions. Whether one aims to describe a dataset or predict, statistical techniques are needed to guarantee accuracy and credibility in analysis.

3.2 Descriptive Statistics in Data Analysis

Descriptive statistics are used to summarize and describe the main features of a dataset. They provide simple but powerful insights into the distribution, central tendency, and variability of data. Common descriptive statistics include:

- **Measures of Central Tendency:** Mean, median, and mode.
- **Measures of Dispersion:** Range, variance, and standard deviation.
- **Shape and Distribution:** Skewness.

These measures allow analysts to quickly understand how data is distributed and to detect any anomalies or patterns that may influence further analysis.

Among the most fundamental descriptive statistics are the mean, median, and mode—collectively known as measures of central tendency.

- **Mean (Average):** The mean is the arithmetic average of all data points. It is widely used but sensitive to outliers. It gives a quick overview of the general value in a dataset.
- **Median:** The median is the middle value when data is arranged in order. It is especially useful in skewed distributions, where the mean may be misleading due to extreme values.
- **Mode:** The mode is the value that appears most frequently in the dataset. It is useful for identifying common occurrences, especially in categorical data.

Understanding these measures allows analysts to grasp where most data points lie and how they are spread out.

key objective of Statistical data analysis?



Figure 3.1 Key Objectives of Statistical data analysis

3.3 Feature Engineering

Feature engineering is the activity of converting raw data into useful input variables that improve the performance of machine learning models and statistical analysis. It entails choosing, altering, or designing new features from available data to more accurately reflect the underlying patterns and relationships. Good feature engineering has a direct influence on the quality of insights extracted from data and the accuracy of predictive models.

Some of the most important steps in feature engineering are:

- Handling missing values
- Encoding categorical variables
- Building interaction features
- Binning and discretization
- Scaling and normalization
- Date/time transformations
- Domain-specific transformations

The purpose is to put the data in a form that makes patterns more visible to the model or analysis being performed.

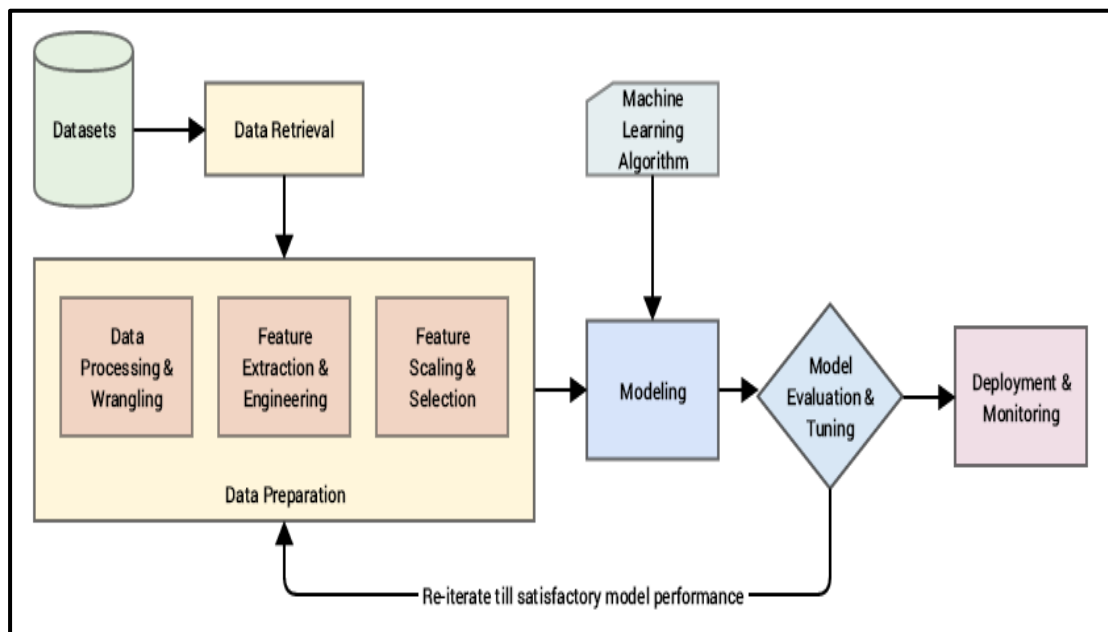
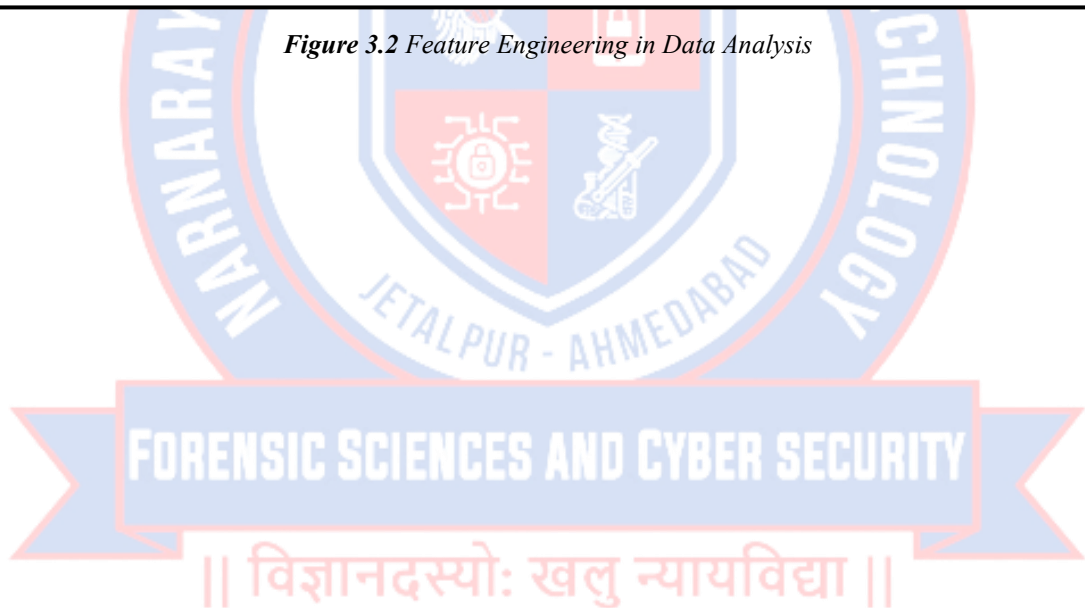


Figure 3.2 Feature Engineering in Data Analysis





A few tools are critical in data preparation, feature engineering, and analysis. Each tool has different capabilities that support various steps in the data pipeline.

4.1 SQL (Structured Query Language)

4.1.1 Purpose

Extraction, filtering, joining, and aggregation of data from databases.

4.1.2 Role in Feature Engineering

- Joining multiple tables to extract new features.
- Aggregation of counts, averages, and totals by categories.
- Execution of conditional logic through CASE WHEN to generate derived features.
- Dealing with missing values or formatting fields at the source level.

4.2 Python

4.2.1 Purpose

Detailed data manipulation, automation, machine learning, and statistical analysis.

4.2.2 Role in Feature Engineering

- Leveraging libraries such as pandas, NumPy, and scikit-learn for data wrangling and transformation.
- Developing custom features using domain logic.
- Dealing with missing values and outliers.
- Encoding methods such as one-hot, label encoding, or frequency encoding.
- Automating scaling, normalization, and feature selection methods.

4.3 Microsoft Excel

4.3.1 Purpose

Rapid exploration, manual data entry, and basic analysis.

4.3.2 Role in Feature Engineering

- Formulas for creating derived columns (e.g., IF, VLOOKUP, TEXT, DATE).
- Using pivot tables and charts to do exploratory data analysis (EDA).
- Filtering and sorting data to identify patterns.

4.4 Power BI

4.4.1 Purpose

- Interactive visualization of data and dashboard creation.
- Role in Feature Engineering:
- Calculated columns and measures with DAX (Data Analysis Expressions).
- Building time-based or category-based aggregations.
- Visual analysis that can lead to new feature ideas.
- Using Power Query (M language) for data shaping by applying transformations and filters.

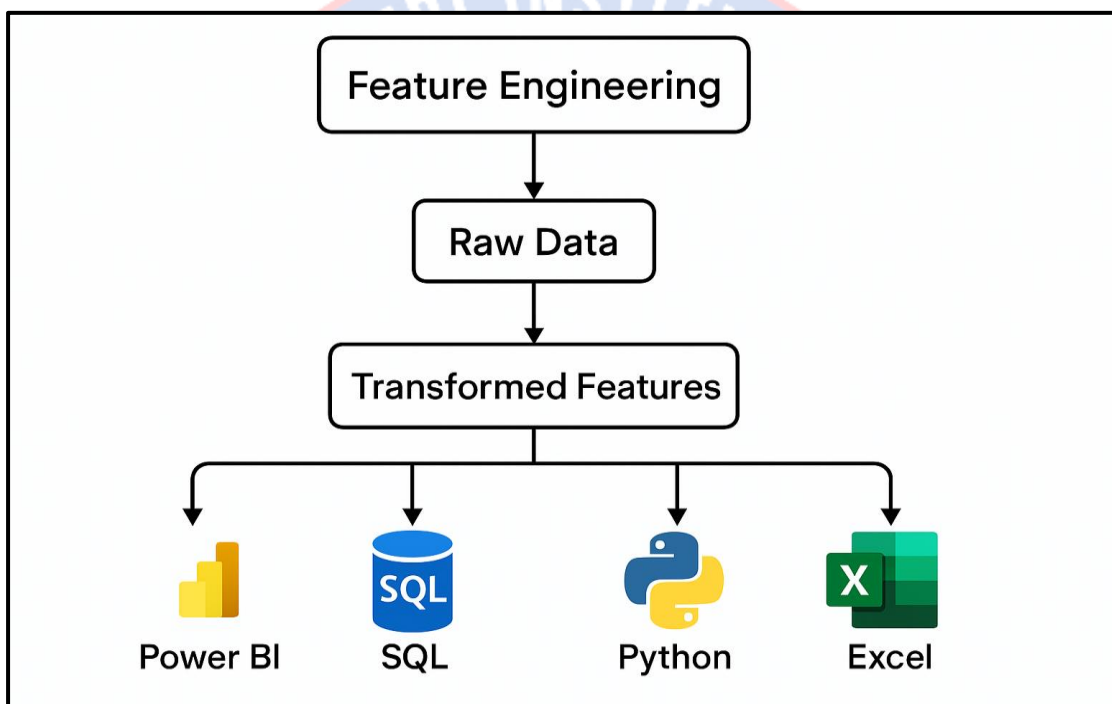


Figure 4.1 Data Flow

In real-world projects, these tools are often used in combination:

- SQL extracts and pre-processes raw data from relational databases.
- Python is used for deeper transformations and modeling.
- Excel allows for quick review and experimentation with small subsets.
- Power BI helps visualize the data and identify trends or relationships that suggest new features.

This integrated approach ensures data is well-prepared, insights are accessible, and models are informed by meaningful, well-constructed features.



5.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an essential process in data analysis that seeks to comprehend the structure, quality, and underlying patterns in a data set prior to any rigorous modeling or hypothesis testing.

The key goals of EDA are:

- Detection of trends, patterns, and outliers.
- Verification of data distributions and correlations.
- Management of missing values and outliers.
- Formulation of hypotheses and informing feature engineering.

EDA is a visual and iterative process that depends heavily on summary statistics and data visualization methods to obtain an intuitive feel for the data.

5.2 Steps Involved in EDA

General EDA involves the following important steps:

- **Data Collection and Loading:** Collecting data from places like databases, APIs, or CSV files.
- **Data Cleaning:** Correcting the wrong data types, processing missing values and deleting duplicates.
- **Descriptive Statistics:** Computing measures like mean, median, mode, standard deviation, skewness, and kurtosis.
- **Visualization:** Employing charts (histograms, box plots, scatter plots, heatmaps) to identify distributions, trends, and relationships.
- **Correlation Analysis:** Identifying relationships between numeric variables through correlation matrices or pair plots.
- **Data Profiling:** Looking at value distributions, frequency counts, and category balances.

The following dataset in the image consists of Flight Prices. EDA has been performed to understand the data and prepare it keeping in mind that the dataset could be used to give a machine learning model for training.

The following are the columns that are present in the raw dataset in the below image.

FEATURES

The various features of the cleaned dataset are explained below:

1. Airline: The name of the airline company is stored in the airline column. It is a categorical feature having 6 different airlines.
2. Flight: Flight stores information regarding the plane's flight code. It is a categorical feature.
3. Source City: City from which the flight takes off. It is a categorical feature having 6 unique cities.
4. Departure Time: This is a derived categorical feature obtained created by grouping time periods into bins. It stores information about the departure time and have 6 unique time labels.
5. Stops: A categorical feature with 3 distinct values that stores the number of stops between the source and destination cities.
6. Arrival Time: This is a derived categorical feature created by grouping time intervals into bins. It has six distinct time labels and keeps information about the arrival time.
7. Destination City: City where the flight will land. It is a categorical feature having 6 unique cities.
8. Class: A categorical feature that contains information on seat class; it has two distinct values: Business and Economy.
9. Duration: A continuous feature that displays the overall amount of time it takes to travel between cities in hours.
- 10) Days Left: This is a derived characteristic that is calculated by subtracting the trip date by the booking date.
10. Price: Target variable stores information of the ticket price.

Figure 5.1: Features of the dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

✓ 0.0s

Figure 5.2: Importing libraries

Firstly we import the necessary libraries required to assess the dataset more can be added if we need them.

```
df = pd.read_excel('flight_price.xlsx')
df.head()
```

✓ 1.2s

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

Figure 5.3: Sampling the dataset

In this case the dataset which loaded is in an excel sheet format. The pandas library in python has a function called [read_excel(file_name)] to load an excel file as a dataframe.

॥ विज्ञानदस्योः खलु न्यायविद्या ॥

```
# Basic info about data
df.info()

✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Airline                10683 non-null  object
1   Date_of_Journey        10683 non-null  object
2   Source                 10683 non-null  object
3   Destination            10683 non-null  object
4   Route                  10682 non-null  object
5   Dep_Time               10683 non-null  object
6   Arrival_Time           10683 non-null  object
7   Duration               10683 non-null  object
8   Total_Stops            10682 non-null  object
9   Additional_Info        10683 non-null  object
10  Price                  10683 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

Figure 5.4: Info about Dataset

The .info() function is used on the dataset to get basic information about the columns or dataset. According to the image above we can see that the data types for all the columns are in object only the Price column has the datatype as integer. We might need to change some of the data types as we go ahead.

```
df.describe()
```

✓ 0.0s

	Price
count	10683.000000
mean	9087.064121
std	4611.359167
min	1759.000000
25%	5277.000000
50%	8372.000000
75%	12373.000000
max	79512.000000

Figure 5.5: Describing the dataset

The `df.describe()` shows us the statistical values. The following image only shows one column as the `.describe` function only works with numerical or integer values, as we have seen in the previous image only the Price column has integer values.

Now to make feature I have decided to use the **Date_of_Journey** column and split it into individual columns of itself mainly day, month, year.

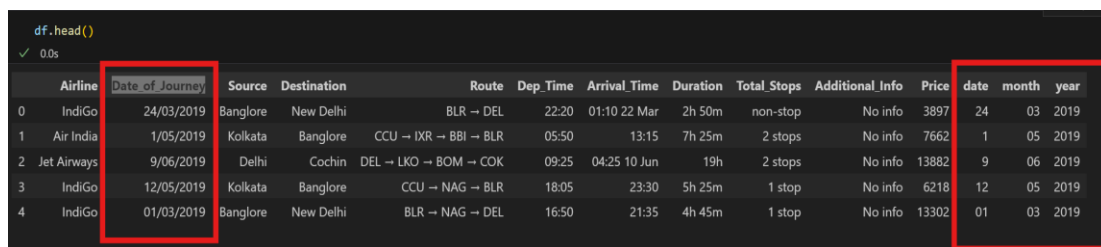
```
#feature
df['date']=df['Date_of_Journey'].str.split('/').str[0]
df['month']=df['Date_of_Journey'].str.split('/').str[1]
df['year']=df['Date_of_Journey'].str.split('/').str[2]
```

✓ 0.1s

Figure 5.6: Splitting values for columns

The image above shows code for creating a new column namely date, month, year

Using the Date_of_Journey column and using the split function to select the values respectively.

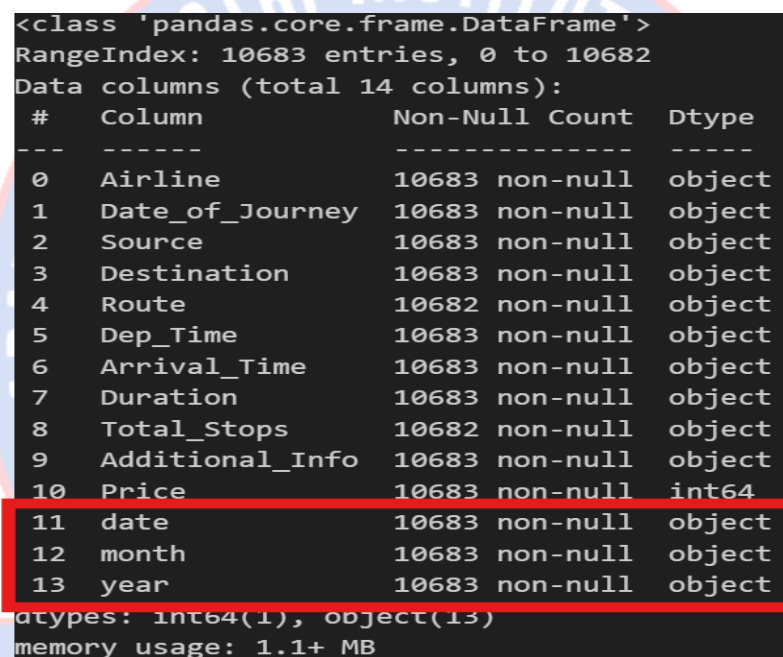


```
df.head()
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	date	month	year
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	03	2019
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	05	2019
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882	9	06	2019
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	12	05	2019
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	01	03	2019

Figure 5.7: Splitted columns

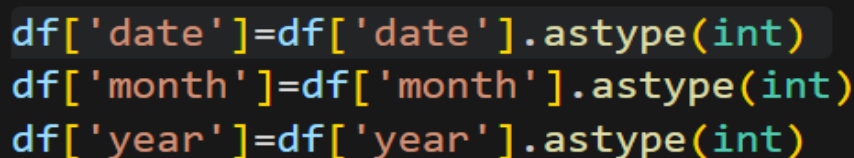
Now when we use the function df.info()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 14 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Airline              10683 non-null  object
1   Date_of_Journey      10683 non-null  object
2   Source               10683 non-null  object
3   Destination          10683 non-null  object
4   Route               10682 non-null  object
5   Dep_Time             10683 non-null  object
6   Arrival_Time         10683 non-null  object
7   Duration             10683 non-null  object
8   Total_Stops          10682 non-null  object
9   Additional_Info      10683 non-null  object
10  Price                10683 non-null  int64
11  date                 10683 non-null  object
12  month                10683 non-null  object
13  year                 10683 non-null  object
dtypes: int64(1), object(13)
memory usage: 1.1+ MB
```

Figure 5.8: Describing the dataset

We can see that the new columns have the data type as 'object', we need to turn it into an interger datatype se we can feed the data into a model.



```
df['date']=df['date'].astype(int)
df['month']=df['month'].astype(int)
df['year']=df['year'].astype(int)
```

Figure 5.9: Changing datatype

The above code in the image typecasts the data type for the entire column.

```
df.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Airline                10683 non-null  object
1   Date_of_Journey        10683 non-null  object
2   Source                 10683 non-null  object
3   Destination            10683 non-null  object
4   Route                 10682 non-null  object
5   Dep_Time              10683 non-null  object
6   Arrival_Time          10683 non-null  object
7   Duration              10683 non-null  object
8   Total_Stops           10682 non-null  object
9   Additional_Info       10683 non-null  object
10  Price                 10683 non-null  int64
11  date                 10683 non-null  int64
12  month                10683 non-null  int64
13  year                 10683 non-null  int64
dtypes: int64(4), object(10)
memory usage: 1.1+ MB
```

Figure 5.10: Changed datatype

Now that we have changed the data type df.info() function can be used to check if it has worked.

We can drop the column as we no longer need it.

```
#Since we have striped the date column we can remove the column as the table no longer needs the column
##drop dateof journey column
df.drop('Date_of_Journey',axis=1,inplace=True)
```

Figure 5.11: Dropping columns

Furthermore we can use the Arrival_Time column to extract the hour and minuite into its respective columns.

```
df['Arrival_hour']=df['Arrival_Time'].str.split(':').str[0]
df['Arrival_min']=df['Arrival_Time'].str.split(':').str[1]
```

Figure 5.12: Splitting Hours and minutes

```
df.head()
```

	Airline	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	date	month	year	Arrival_hour	Arrival_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	01:10	2h 50m	non-stop	No info	3897	24	3	2019	01	10
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662	1	5	2019	13	15
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25	19h	2 stops	No info	13882	9	6	2019	04	25
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218	12	5	2019	23	30
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302	1	3	2019	21	35

Figure 5.13: Splited columns into hours and minutes

We need to further change the data type of the newly made columns.

```
df['Arrival_hour']=df['Arrival_hour'].astype(int)
df['Arrival_min']=df['Arrival_min'].astype(int)
```

Figure 5.14: Changing datatype

We can check if the data type has been changed by using df.info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 15 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   Airline              10683 non-null  object
1   Source               10683 non-null  object
2   Destination          10683 non-null  object
3   Route                10682 non-null  object
4   Dep_Time             10683 non-null  object
5   Arrival_Time         10683 non-null  object
6   Duration              10683 non-null  object
7   Total_Stops          10682 non-null  object
8   Additional_Info      10683 non-null  object
9   Price                10683 non-null  int64
10  date                 10683 non-null  int64
11  month                10683 non-null  int64
12  year                 10683 non-null  int64
13  Arrival_hour         10683 non-null  int64
14  Arrival_min          10683 non-null  int64
dtypes: int64(6), object(9)
memory usage: 1.2+ MB
```

Figure 5.15: Conforming datatype

The same can be done to departure time and data type can be changed.
The final table looks like this.

```
df.head()
```

✓ 0.0s Python

	Airline	Source	Destination	Route	Dep_Time	Duration	Total_Stops	Additional_Info	Price	date	month	year	Arrival_hour	Arrival_min	Departure_hour	Departure_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	22:20	2h 50m	non-stop	No info	3897	24	3	2019	1	10	22	20
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	7h 25m	2 stops	No info	7662	1	5	2019	13	15	05	50
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	19h	2 stops	No info	13882	9	6	2019	4	25	09	25
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	18:05	5h 25m	1 stop	No info	6218	12	5	2019	23	30	18	05
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	16:50	4h 45m	1 stop	No info	13302	1	3	2019	21	35	16	50

Figure 5.16: Viewing dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Airline                               10683 non-null  object
1   Source                                10683 non-null  object
2   Destination                           10683 non-null  object
3   Route                                 10682 non-null  object
4   Dep_Time                              10683 non-null  object
5   Duration                              10683 non-null  object
6   Total_Stops                           10682 non-null  object
7   Additional_Info                       10683 non-null  object
8   Price                                 10683 non-null  int64
9   date                                  10683 non-null  int64
10  month                                 10683 non-null  int64
11  year                                  10683 non-null  int64
12  Arrival_hour                          10683 non-null  int64
13  Arrival_min                           10683 non-null  int64
14  Departure_hour                        10683 non-null  int64
15  Departure_min                         10683 non-null  int64
dtypes: int64(8), object(8)
memory usage: 1.3+ MB
```

Figure 5.17: Viewing datatypes

Now we check if any of our columns contain null values.

```
df['Total_Stops'].unique()
✓ 0.0s
array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
      dtype=object)
```

Figure 5.18: Unique values

We can see that there are 'nan' values in the 'Total_Stops' column. We can use the isNull() function to check for null values.

```
df[df['Total_Stops'].isnull()]
✓ 0.0s
#only one record with nan
```

	Airline	Source	Destination	Route	Du
9039	Air India	Delhi	Cochin	NaN	23

Figure 5.19: Viewing null values

Only one record comes with 'nan' value.

We can further change the stops into a numerical value.

```
df['Total_Stops']=df['Total_Stops'].map({'non-stop':0,'2 stops':2,'1 stop':1,'3 stops':3,'4 stops':4,np.nan:1})
#we changed the number of stops to numerical value because we might need to feed data into a model as number of stops increase the price of the fare
#nan has been replaced to 1 because we checked the mode a mode was =1
✓ 0.0s
```

Figure 5.20: Unique values

The 'nan' value has been reallocated to 1 as the mode of the dataset was 1.

This is how Python has been used to create features and clean or transform the data.



6.1 Amazon Web Services (AWS)

In the real world case scenario the data is generally kept in servers some of the popular data storage providers include AWS, AZURE etc,. In our example we have used AWS services to store the data.

Amazon Web Services (AWS) is the global leader in cloud computing, providing over 200 full-featured services from data centers all over the world. Millions of customers ranging from the most innovative startups, largest companies, and governments are building on AWS to reduce costs, become more agile, and innovate faster.

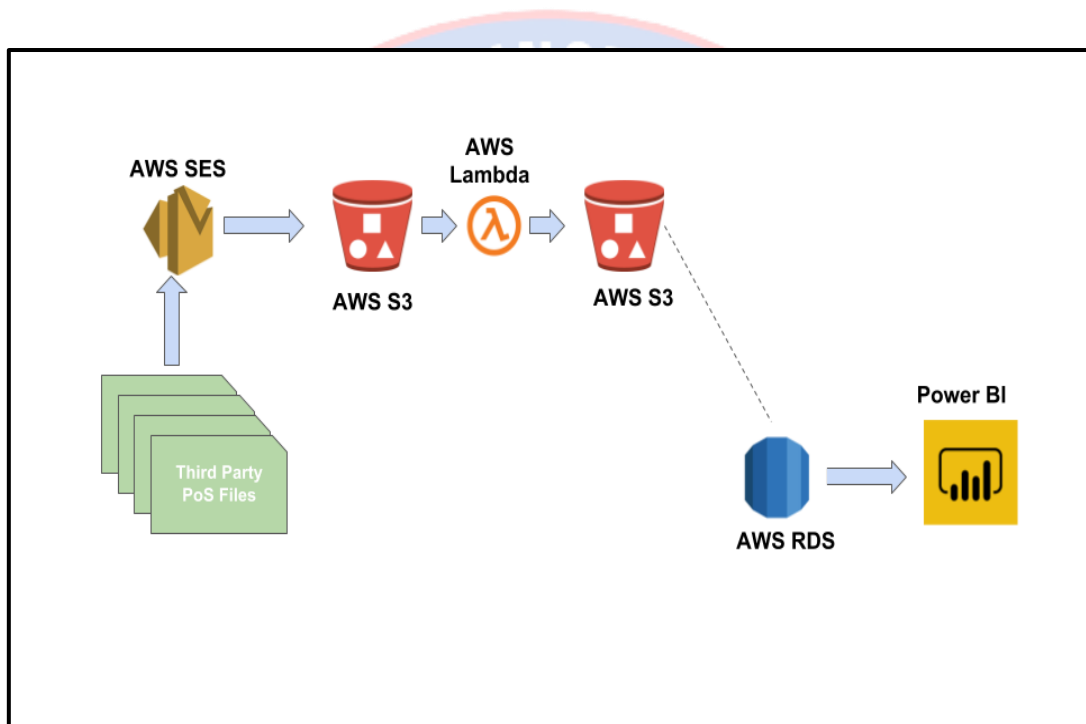


Figure 6.1: Amazon S3 Bucket in Data Analysis

Amazon Simple Storage Service (Amazon S3) is object storage with industry-leading scalability, data availability, security, and performance. All types and sizes of customers store, manage, analyze, and protect any amount of data for a wide variety of use cases like data lakes, cloud-native applications, and mobile apps. With cost-optimized storage classes and simple-to-use management features, you can optimize costs, organize and analyze.

6.2. Snowflake

Snowflake is a cloud data platform developed for data warehousing, data lakes, data engineering, data science, and data application development. Snowflake accommodates both structured and semi-structured data (such as JSON, Avro, and Parquet) and finds a broad usage in analytics because of its scalability, performance, and flexibility.

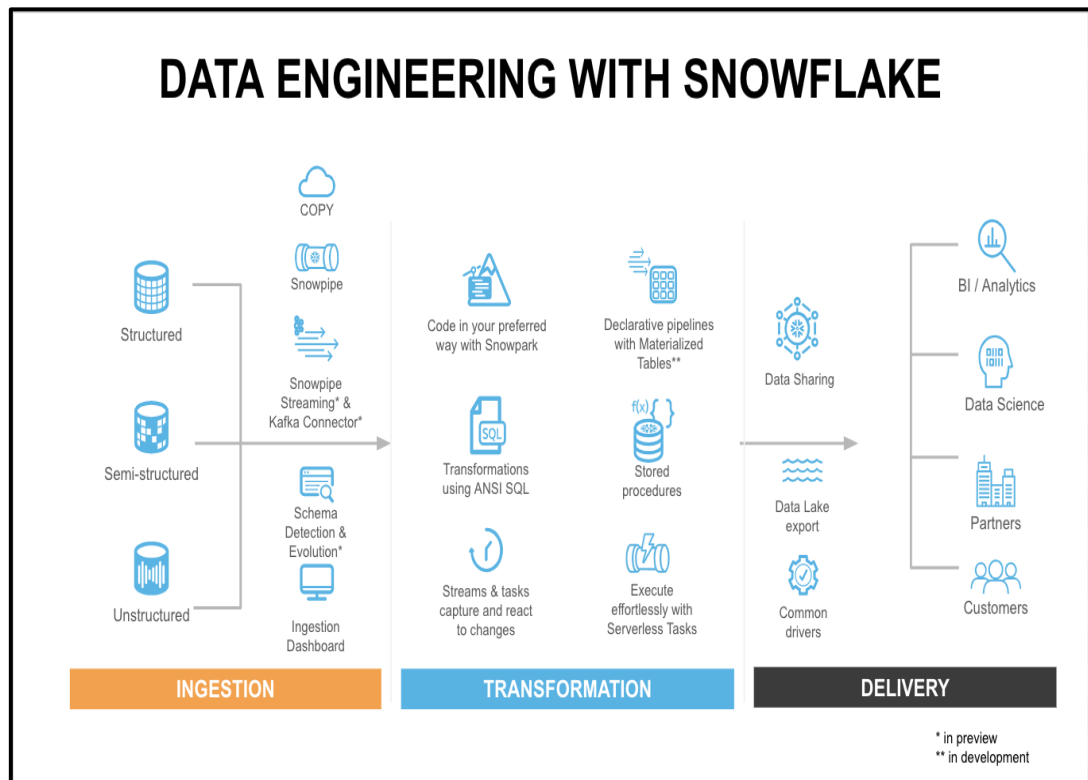


Figure 6.2: Snowflake in Data Engineering

Differing from traditional databases, Snowflake decouples storage and compute, enabling each to be scaled independently. Such a design is especially beneficial to analytics workloads, where performance and concurrency are paramount.

Firstly, we must create a AWS account which gives 12 months' free trial or S3 bucket service.

Snowflake provides a one-month free trial on its platform.

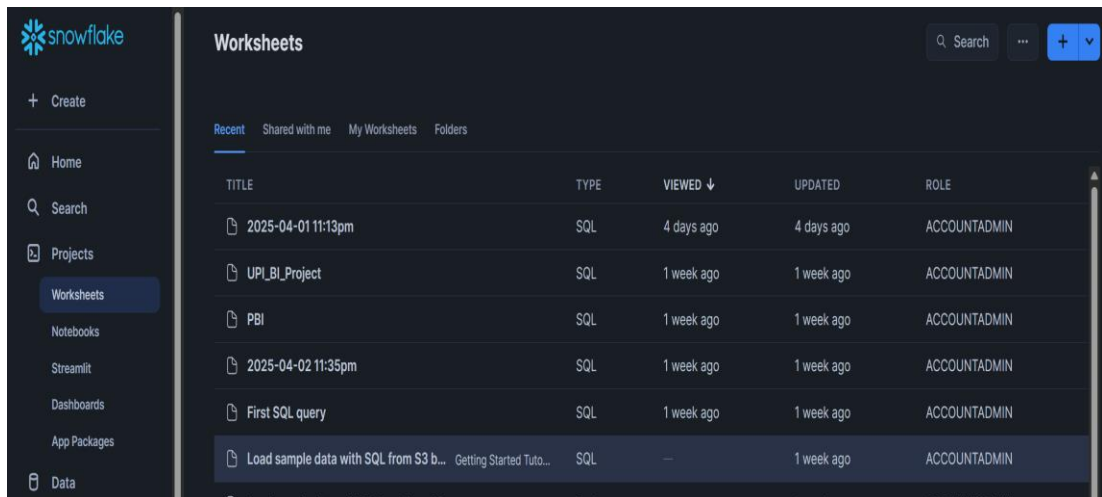


Figure 6.3: Snowflake dashboard

In this project we are going to integrate a End-to-End project using AWS S3 bucket, Snowflake and Power BI. The steps include

- Create a free AWS account.
- Select the S3 bucket and get the free 12 months free trial.
- Create Role
- Set the Identity and Access Management (IAM)
- Upload the data file into the S3 bucket

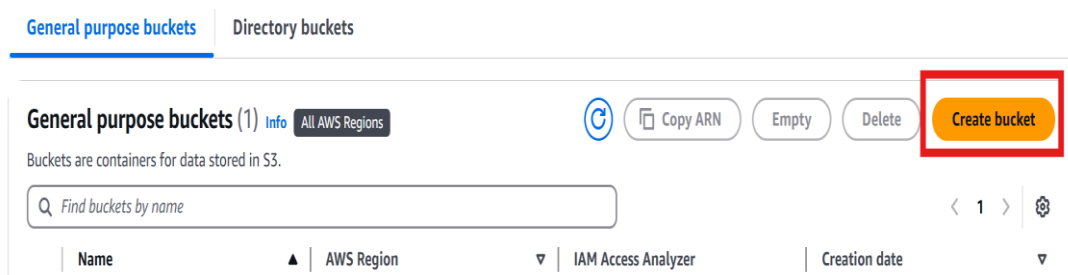


Figure 6.4: Creating bucket

After creating the bucket, we name the bucket and upload the file.

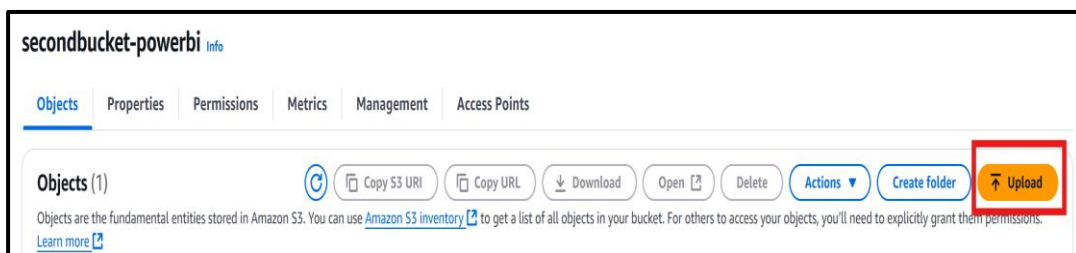


Figure 6.5: Uploading file to bucket

Next, we create a role.

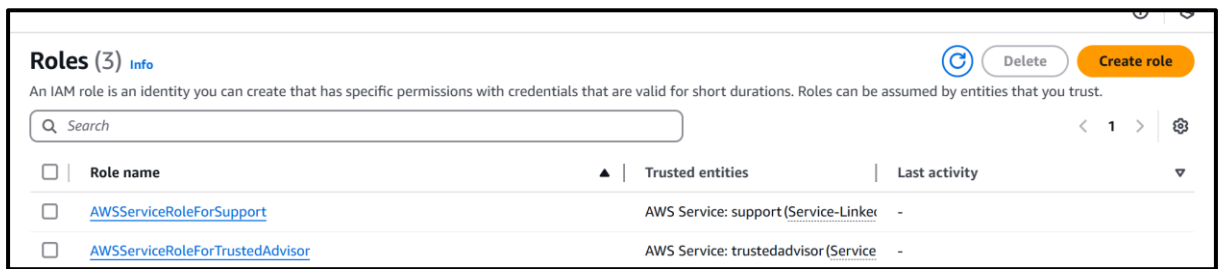


Figure 6.6: Creating role

For the project I have decided to use AmazonS3FullAccess. Amazon comes with pre-installed permissions that are industry standard that are designed for faster deployments.

Now we head over to snowflake to create an integration between AWS and snowflake.

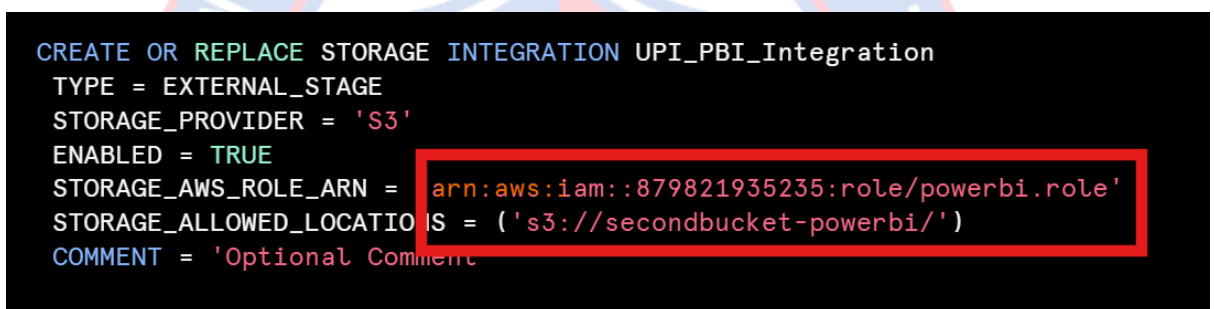


Figure 6.7: ARN value on snowflake

We create an integration name 'UPI_PBI_Integration'

The ARN can be found in the AWS dashboard and the storage location is the directory where the data file has been saved on the AWS S3 bucket.

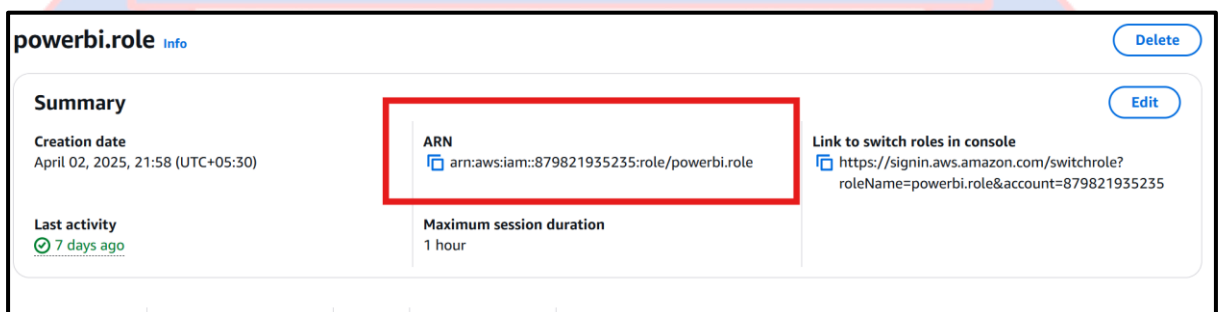


Figure 6.8: ARN values in Amazon S3

We have to create a database, a schema and an empty table on snowflake with the proper data types.

```

15 CREATE database UPI_PowerBI;
16
17 create schema UPI_PBI_Data;
18
19 CREATE OR REPLACE TABLE PBI_UPI_Dataset (
20     TransactionID STRING,
21     TransactionDate DATE,
22     Amount FLOAT,
23     BankNameSent STRING,
24     BankNameReceived STRING,
25     RemainingBalance FLOAT,
26     City STRING,
27     Gender STRING,
28     TransactionType STRING,
29     Status STRING,
30     TransactionTime TIME,
31     DeviceType STRING,
32     PaymentMethod STRING,
33     MerchantName STRING,
34     Purpose STRING,
35     CustomerAge INTEGER,
36     PaymentMode STRING,
37     Currency STRING,
38     CustomerAccountNumber STRING,
39     MerchantAccountNumber STRING
40 );

```

Figure 6.9: Creating Schema

Now that we have made an empty table, we have to import the data into the table. In snowflake we have to create a stage and load the data set into the stage first.

```

create stage UPI_PowerBI.UPI_PBI_Data.pbi_stage1
url = 's3://secondbucket-powerbi/'
storage_integration = UPI_PBI_Integration

list @pbi_stage1

```

UPI_POWERBI.UPI_PBI_DATA.PBI_STAGE1

Results Chart

name	# size	md5	last_modified
secondbucket-powerbi/UPI+T	3411609	c94e1810f1e5165d292e224e7dc	Thu, 3 Apr 2025 16:1

Figure 6.10: Creating stage

Next, we have to copy the data from the stage named 'pbi_stage1' into the empty table that we created earlier.

```
copy into PBI_UPI_Dataset
from @pbi_stage1
file_format = (type='csv' field_delimiter=',' skip_header=1)
on_error='continue'
```

Figure 6.11: Copying data set into PBI_UPI_Dataset

The screenshot shows a Snowflake SQL interface. The top part displays a query: `select * from PBI_UPI_Dataset //no data available - import from AWS S3 bucket`. Below the query, there are tabs for 'Results' and 'Chart'. The 'Results' tab is active, showing a table with 6 columns: TRANSACTIONID, TRANSACTIONDATE, AMOUNT, BANKNAMESENT, BANKNAMERECEIVED, and REMAINING. The table contains 4 rows of data.

TRANSACTIONID	TRANSACTIONDATE	AMOUNT	BANKNAMESENT	BANKNAMERECEIVED	REMAINING
TXN00001	2024-02-02	271.64	SBI Bank	HDFC Bank	
TXN00002	2024-03-03	1064.63	ICICI Bank	SBI Bank	
TXN00003	2024-04-04	144.15	Axis Bank	Axis Bank	
TXN00004	2024-05-05	612.89	HDFC Bank	ICICI Bank	

Figure 6.12: Viewing dataset on snowflake

The data has been loaded onto the empty table successfully.

The main advantage of snowflake is that it separates the storage and computational layer, which gives the freedom of increasing or decreasing according to the needs of the project.

॥ विज्ञानदस्योः खलु न्यायविद्या ॥

6.3 Power BI

The third part is to import the data into a platform that can help analyze the data.

Snowflake is compatible with most data platforms like Power BI, Tableau, My SQL etc.

To import the data into Power BI we have the Get Data button which opens up the following window.

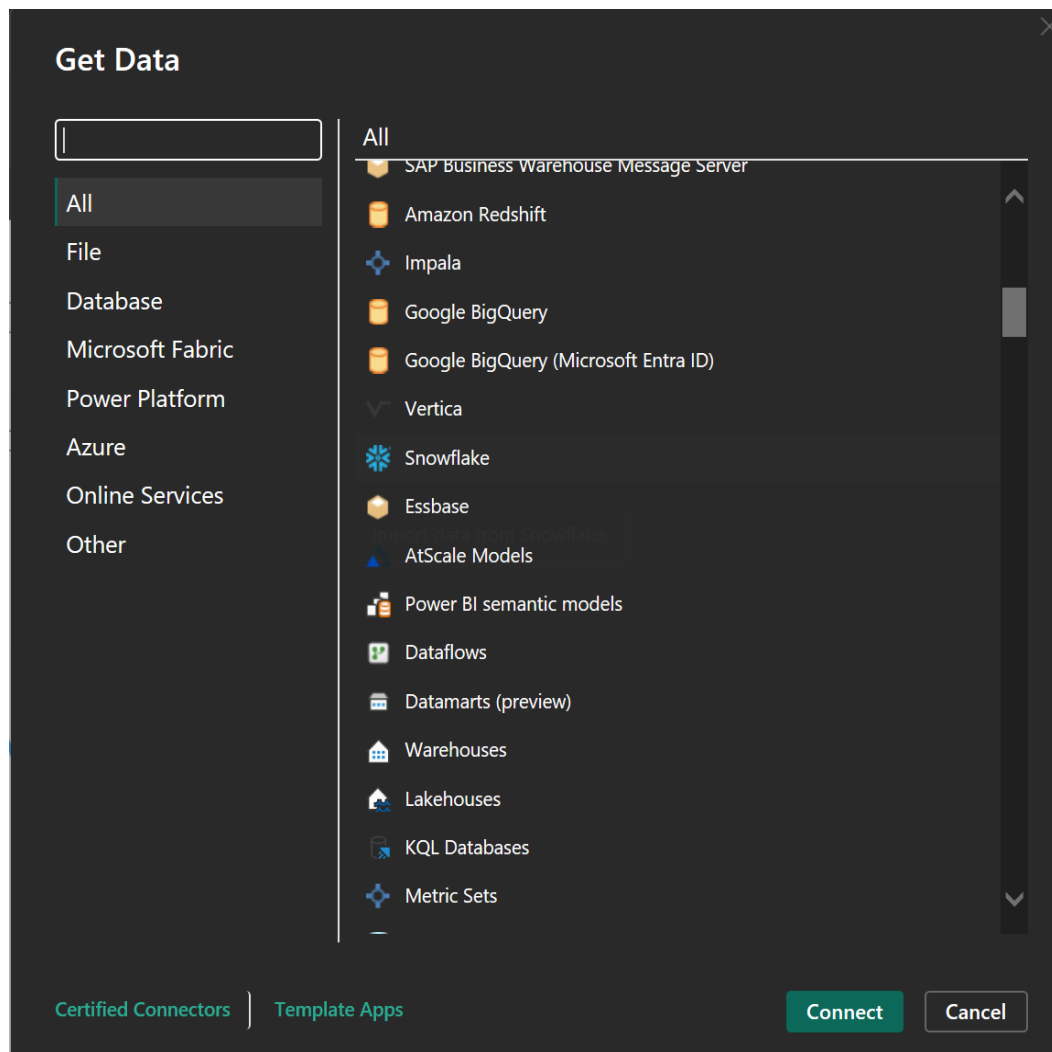


Figure 6.13: Importing dataset from snowflake

FORENSIC SCIENCES AND CYBER SECURITY

We scroll to snowflake as that is the service have used to store the data.



Figure 6.14: Credential Screen

Power BI will prompt the user to enter the credentials.

TRANSACTIONID	TRANSACTIONDATE	AMOUNT	BANKNAMESENT	BANKNAMERECEIVED	REMAININGBALANCE	CITY	GENDER	TRANSACTIONTYPE	STATUS	TRANSACTIONTIME
TXN00061	02 February 2024	96	SBI Bank	HDFC Bank	4,541	Delhi	Female	Transfer	Success	
TXN00181	02 February 2024	1,778	SBI Bank	HDFC Bank	8,294	Delhi	Female	Transfer	Success	
TXN00301	02 February 2024	1,274	SBI Bank	HDFC Bank	1,192	Delhi	Female	Transfer	Success	
TXN00661	02 February 2024	288	SBI Bank	HDFC Bank	3,405	Delhi	Female	Transfer	Success	
TXN00781	02 February 2024	1,650	SBI Bank	HDFC Bank	8,719	Delhi	Female	Transfer	Success	
TXN00901	02 February 2024	1,693	SBI Bank	HDFC Bank	1,223	Delhi	Female	Transfer	Success	
TXN01021	02 February 2024	1,318	SBI Bank	HDFC Bank	6,505	Delhi	Female	Transfer	Success	
TXN01141	02 February 2024	1,075	SBI Bank	HDFC Bank	4,906	Delhi	Female	Transfer	Success	
TXN01261	02 February 2024	1,678	SBI Bank	HDFC Bank	6,532	Delhi	Female	Transfer	Success	
TXN01381	02 February 2024	1,371	SBI Bank	HDFC Bank	9,039	Delhi	Female	Transfer	Success	
TXN01501	02 February 2024	583	SBI Bank	HDFC Bank	269	Delhi	Female	Transfer	Success	
TXN01621	02 February 2024	1,117	SBI Bank	HDFC Bank	6,373	Delhi	Female	Transfer	Success	
TXN01741	02 February 2024	741	SBI Bank	HDFC Bank	6,798	Delhi	Female	Transfer	Success	
TXN01861	02 February 2024	1,450	SBI Bank	HDFC Bank	8,484	Delhi	Female	Transfer	Success	
TXN01981	02 February 2024	77	SBI Bank	HDFC Bank	3,521	Delhi	Female	Transfer	Success	
TXN02101	02 February 2024	1,892	SBI Bank	HDFC Bank	6,825	Delhi	Female	Transfer	Success	
TXN02221	02 February 2024	1,626	SBI Bank	HDFC Bank	7,313	Delhi	Female	Transfer	Success	
TXN02341	02 February 2024	1,110	SBI Bank	HDFC Bank	1,357	Delhi	Female	Transfer	Success	
TXN02461	02 February 2024	1,784	SBI Bank	HDFC Bank	3,571	Delhi	Female	Transfer	Success	
TXN02581	02 February 2024	417	SBI Bank	HDFC Bank	6,058	Delhi	Female	Transfer	Success	
TXN02701	02 February 2024	344	SBI Bank	HDFC Bank	8,667	Delhi	Female	Transfer	Success	
TXN02821	02 February 2024	1,122	SBI Bank	HDFC Bank	2,459	Delhi	Female	Transfer	Success	
TXN02941	02 February 2024	1,487	SBI Bank	HDFC Bank	920	Delhi	Female	Transfer	Success	
TXN03061	02 February 2024	1,112	SBI Bank	HDFC Bank	8,683	Delhi	Female	Transfer	Success	

Figure 6.15: Uploaded dataset in Power BI

We can see in the table view the data has successfully been uploaded to Power BI. We can verify by looking at the top left corner. The name specified is the same name we have specified in Snowflake namely 'PBI_UPI_DATASET'.

There are 20 columns in our dataset namely TransactionID, TransactionDate, Amount, BankNameSent, BankNameReceived, RemainingBalance, City, Gender, TransactionType, Status, TransactionTime, DeviceType, PaymentMethod, MerchantName, Purpose, CustomerAge, PaymentMode, Currency, CustomerAccountNumber, MerchantAccountNumber.

By further analyzing the dataset I felt the need to make a few changes i.e., I am using Power BI's Power Query editor to make the changes.

TransactionDate
02-02-2024 00:00:00
03-03-2024 00:00:00
04-04-2024 00:00:00
05-05-2024 00:00:00
06-06-2024 00:00:00
07-07-2024 00:00:00
08-08-2024 00:00:00
09-09-2024 00:00:00
10-10-2024 00:00:00
11-11-2024 00:00:00
12-12-2024 00:00:00
13-01-2024 00:00:00

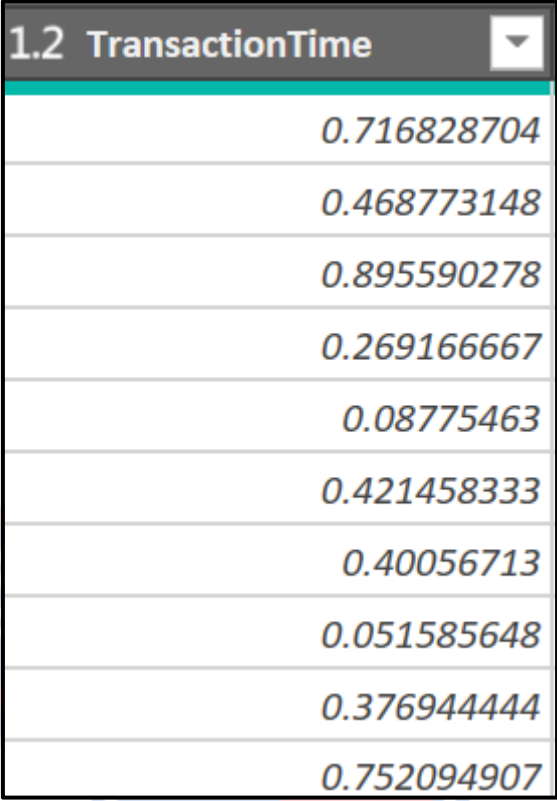
Figure 6.16: Transforming data

We do not need the date/time format. We can change the data type by clicking on the icon beside the column name, and change it to date.

TransactionDate
02-02-2024
03-03-2024
04-04-2024
05-05-2024
06-06-2024
07-07-2024
08-08-2024
09-09-2024
10-10-2024
11-11-2024

Figure 6.17: Transformed column

On further inspection we see that the TransactionTime column has float values,

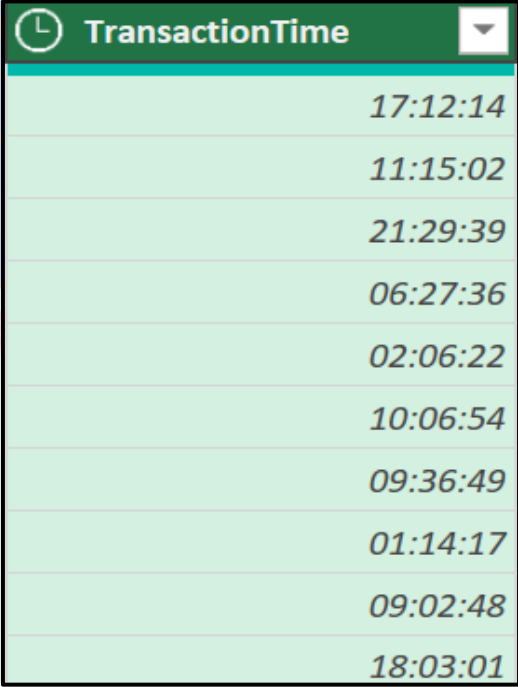


A screenshot of a data table with a dark header bar. The header bar contains the text '1.2 TransactionTime' and a dropdown arrow. The table has 10 rows, each containing a float value. The values are: 0.716828704, 0.468773148, 0.895590278, 0.269166667, 0.08775463, 0.421458333, 0.40056713, 0.051585648, 0.376944444, and 0.752094907.

1.2 TransactionTime
0.716828704
0.468773148
0.895590278
0.269166667
0.08775463
0.421458333
0.40056713
0.051585648
0.376944444
0.752094907

Figure 6.18: Transforming column (2)

We need to change the data type to time.



A screenshot of a data table with a dark header bar. The header bar contains a clock icon, the text 'TransactionTime', and a dropdown arrow. The table has 10 rows, each containing a time value in HH:MM:SS format. The values are: 17:12:14, 11:15:02, 21:29:39, 06:27:36, 02:06:22, 10:06:54, 09:36:49, 01:14:17, 09:02:48, and 18:03:01.

TransactionTime
17:12:14
11:15:02
21:29:39
06:27:36
02:06:22
10:06:54
09:36:49
01:14:17
09:02:48
18:03:01

Figure 6.19: Transformed column (2)

The above screenshot show the changed data type.

Upon further inspection we see that the CustomeAccountNumber and the MerchantAccountNumber is written in the scientific notation I.e,

1 ₂₃ CustomerAccountNumber ▼	1 ₂₃ MerchantAccountNumber ▼
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11
1.23457E+11	9.87654E+11

Figure 6.20: Transforming column (3)

Since we CustomeAccountNumber and the MerchantAccountNumber is used to identify the customer. It is a unique ID assigned to a customer. It cannot be in the scientific notation. Let's change the format to text.

A ^B _C CustomerAccountNumber ▼	A ^B _C MerchantAccountNumber ▼
123456789013	987654321013
123456789014	987654321014
123456789015	987654321015
123456789016	987654321016
123456789017	987654321017
123456789018	987654321018
123456789019	987654321019

Figure 6.21: Transformed column (3)

The rest of the columns seem to have the correct data types. We can save and load the cleaned data set into the Power BI board.

Now that we have cleaned the data we can use the data for visualizations. I have decided to give full control over the attributes so the user can filter according to the need of the information needed.

The following layout lets the user fine tune the attributes.

BANKNAMESENT <input type="checkbox"/> Axis Bank <input type="checkbox"/> HDFC Bank	BANKNAMERECEIVED <input type="checkbox"/> Axis Bank <input type="checkbox"/> HDFC Bank	CITY <input type="checkbox"/> Bangalore <input type="checkbox"/> Delhi	DEVICETYPE <input type="checkbox"/> Laptop <input type="checkbox"/> Mobile	GENDER <input type="checkbox"/> Female <input type="checkbox"/> Male
Age_groups <input type="checkbox"/> A1 <input type="checkbox"/> A2	MERCHANTNAME <input type="checkbox"/> Amazon <input type="checkbox"/> Flinkart	PAYMENTMETHOD <input type="checkbox"/> Phone Number <input type="checkbox"/> QR Code	PURPOSE <input type="checkbox"/> Bill Payment <input type="checkbox"/> Food	TRANSACTIONTYPE <input type="checkbox"/> Payment <input type="checkbox"/> Transfer

Figure 6.22: Attributes from dataset

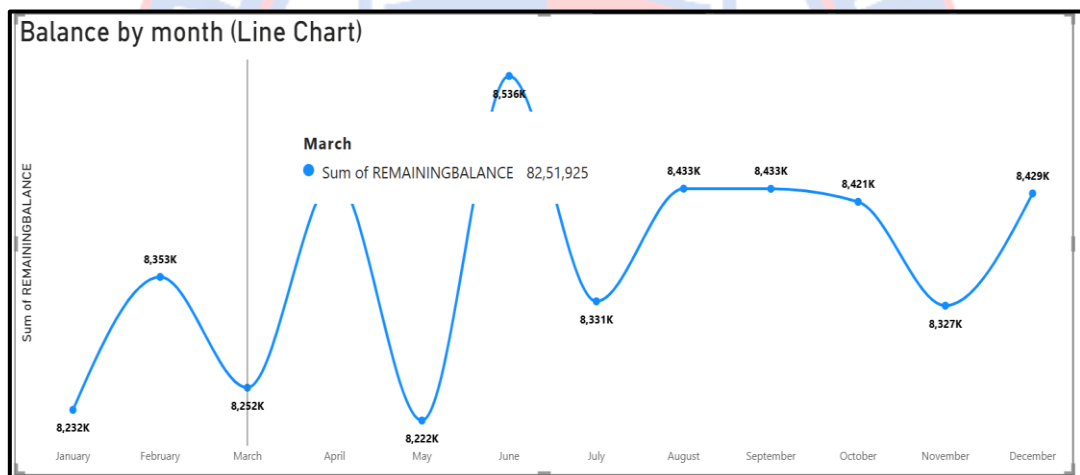


Figure 6.23: Visualization

The default line chart that shows sum of remaining balance over the year.

The chart is prepared using two columns remaining balance and transaction date.

<input type="checkbox"/>	PURPOSE
<input checked="" type="checkbox"/>	Σ REMAININGBALANCE
<input type="checkbox"/>	STATUS
<input checked="" type="checkbox"/>	> TRANSACTIONDATE

Figure 6.24: Attributes used for visualization

This can provide various insights like if the client asks what are the spendings of the female gender over the year in the category of food for example, selecting the appropriate attributes shows the spending pattern over the year.

The image shows a dashboard with four filter panels. The 'DEVICETYPE' panel has checkboxes for 'Laptop' and 'Mobile'. The 'GENDER' panel has a selected 'Female' option (indicated by a black square) and an unchecked 'Male' option. The 'PURPOSE' panel has a selected 'Food' option (indicated by a black square) and an unchecked 'Others' option. The 'TRANSACTIONTYPE' panel has an unchecked 'Transfer' option.

Figure 6.25: Selecting custom attributes

The following shows the spending pattern of female customers in the food category.

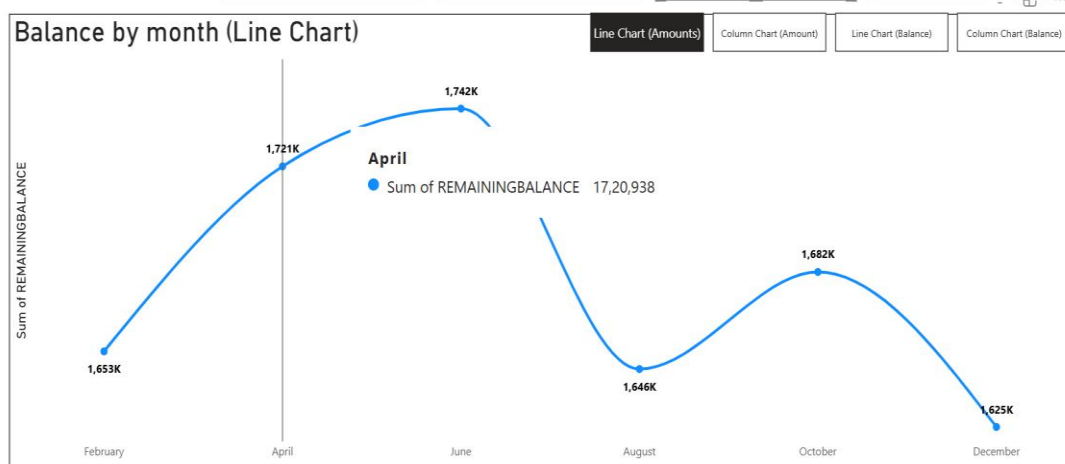
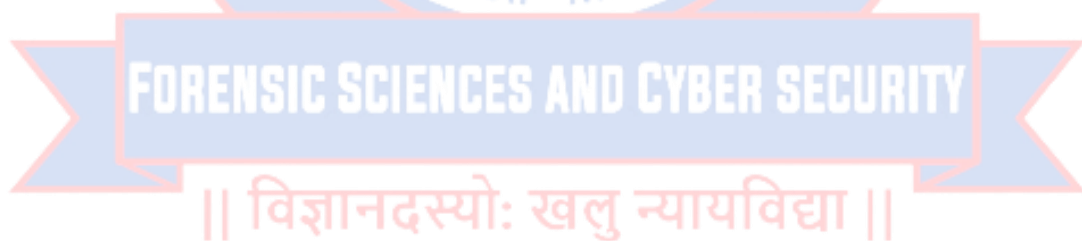


Figure 6.26: Visuals according to custom attributes



The data analytics transformation goes well beyond the research environment, redefining the way organizations operate and make decisions based on insights derived from data. Through the use of tools such as customer behavior analysis, companies can adjust marketing efforts to meet changing needs, desires, and trends. In medicine, analytics reveals trends in patient information, allowing practitioners to improve diagnostic accuracy and provide personalized treatment. But this revolution is not without ethical dilemmas. The exponential expansion of data gathering fuels privacy and security concerns, requiring strong guarantees of ethical use. Imperatively, even as analytics provides unparalleled support for decision-making, it must be a supplement to, not a substitute for, human judgment, acting as an aid to enhance critical thinking and not dominate it. The power of data analytics over future research is deep, allowing actionable insights to be gleaned from massive datasets and transforming methodologies across sectors.

Yet, its integration in society depends on confrontation of ethical threats openly. As analytics becomes inseparable from research and organizational strategy, stakeholders will need to set accountability as their first priority while ensuring its usage remains in conformity with ethical principles and supports well-informed, balanced choice. The future direction is in balancing innovation with responsibility and realizing that the true value of data analytics materializes only when applied with integrity and foresight.

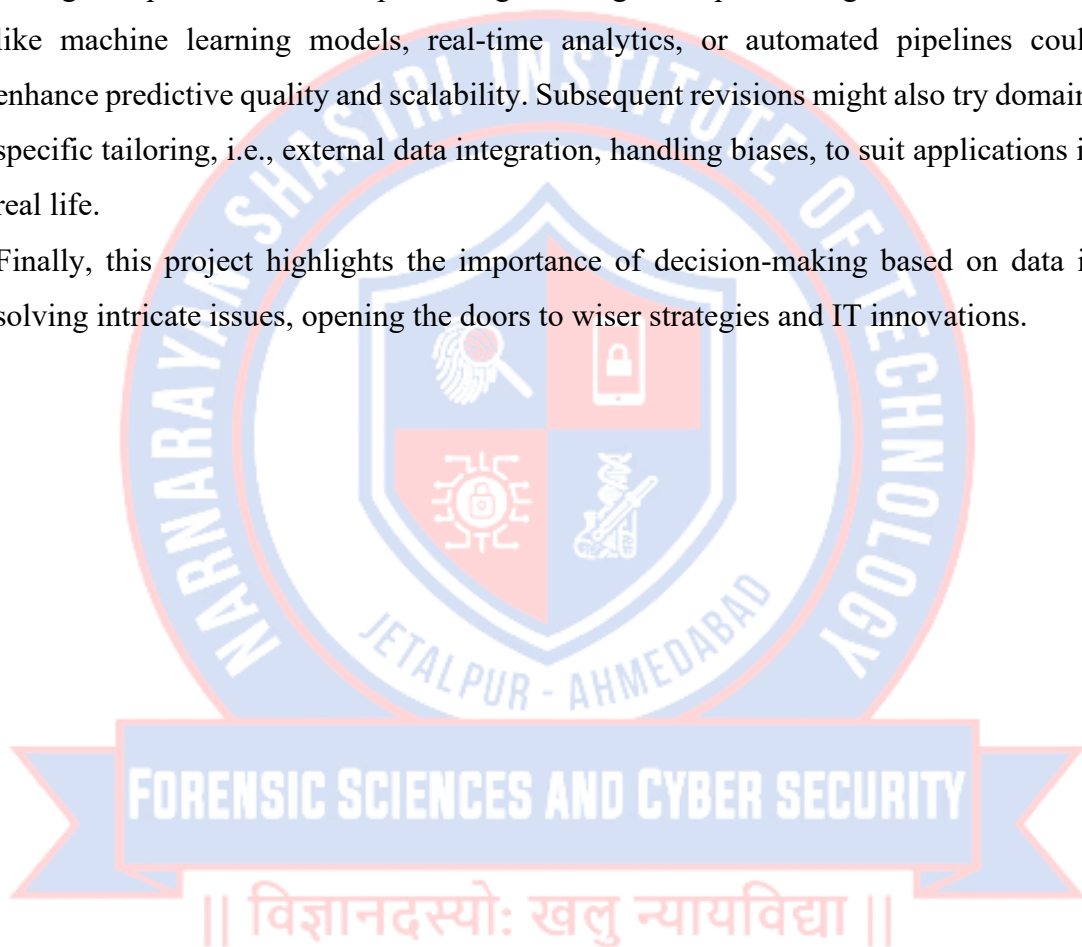


Conclusion

This project showed the strength of data analytics in turning raw data into actionable knowledge through systematic exploratory data analysis (EDA), feature engineering, and statistical validation. Through the discovery of latent patterns, solving data quality problems, and engineering relevant features, the analysis built a solid basis for comprehending main trends and associations in the dataset. Statistical and visualizations tests also verified hypotheses, ensuring the reliability of the results.

Though the present workflow provides great insights, implementing advanced methods like machine learning models, real-time analytics, or automated pipelines could enhance predictive quality and scalability. Subsequent revisions might also try domain-specific tailoring, i.e., external data integration, handling biases, to suit applications in real life.

Finally, this project highlights the importance of decision-making based on data in solving intricate issues, opening the doors to wiser strategies and IT innovations.



References

1. Challa, Narayana. (2023). DATA ANALYTICS AND ITS IMPACT ON FUTURE. Corrosion and Protection. 51. 1.
2. Taherdoost, Hamed, Different Types of Data Analysis; Data Analysis Methods and Techniques in Research Projects (August 1, 2022).
3. Dibekulu, Dawit. (2020). An Overview of Data Analysis and Interpretations in Research. 1-27. 10.14662/IJARER2020.015.
4. <https://docs.aws.amazon.com/s3/>
5. Singh, Dr & Jassi, Jasbir. (2023). Exploring the Significance of Statistics in the Research: A Comprehensive Overview Section A -Research paper. European Chemical Bulletin. 12. 2089-2102. 10.31838/ecb/2023.12.s2.262).
6. Komorowski, Matthieu & Marshall, Dominic & Saliccioli, Justin & Crutain, Yves. (2016). Exploratory Data Analysis. 10.1007/978-3-319-43742-2_15.
7. Dr. Seema Amit Agarwal. Use of Statistics in Research. International Journal for Modern Trends in Science and Technology
8. 2021, 7, pp. 98-103. <https://doi.org/10.46501/IJMTST0711017>.
9. Ali, Zulfiqar & Bhaskar, SBala. (2016). Basic statistical tools in research and data analysis. Indian Journal of Anaesthesia. 60. 662. 10.4103/0019-5049.190623.
10. Ali Z, Bhaskar SB. Basic statistical tools in
11. research and data analysis. Indian J Anaesth 2016;60:662-9.
12. <https://doi.org/10.1016/j.gltp.2022.03.019->
13. Shaik, Hafeezuddin & Rao, A & Vardhan, B. (2021). Role of Exploratory Data Analysis in Data Science. 10.1109/ICCES51350.2021.9488986.
14. <https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/>
15. Rawat, Tara & Khemchandani, Vineeta. (2019). Feature Engineering (FE) Tools and Techniques for Better Classification Performance. International Journal of Innovations in Engineering and Technology. 10.21172/ijiet.82.024.
16. Shanti Pragnya, Swayanshu & Priyadarshi, Shashwat. (2019). The Implication of Statistical Analysis and Feature Engineering for Model Building Using Machine Learning Algorithms. International Journal of Computer Science & Engineering Survey. 10. 01-11. 10.5121/ijcses.2019.10301.