ASSIGNMENT FINAL REPORT

Qualification	**Pearson BTEC Level 5 Higher National Diploma in Computing**		
Unit number and title	**Unit 26: Big Data Analytics and Visualization**		
Submission date	December 11th, 2024	**Date Received 1st submission**	December 11th, 2024
Re-submission Date		**Date Received 2nd submission**	
Group number:	**Student names & codes**	**Final scores**	**Signatures**
	Hoang Anh Quy BS00311	P	Quy
	Nguyen Đinh Tien BS00493	P	Tien
	Tran Ngoc Bao Han BS00666	P	Han
Class	DA06201	**Assessor name**	Mr. Nguyen Van Quang
Plagiarism			

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Student Declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I declare that the work submitted for assessment has been carried out without assistance other than that which is acceptable according to the rules of the specification. I certify I have clearly referenced any sources and any artificial intelligence (AI) tools used in the work. I understand that making a false declaration is a form of malpractice.

Student's signature	Quy

Grading grid

P1	P2	P5	M1	M4	D1	D3

ASSIGNMENT FINAL REPORT

Qualification	**Pearson BTEC Level 5 Higher National Diploma in Computing**		
Unit number and title	**Unit 26: Big Data Analytics and Visualization**		
Submission date	December 11th, 2024	**Date Received 1st Submission**	December 11th, 2024
Re-submission Date		**Date Received 2nd Submission**	
Student Name	Hoang Anh Quy	**Student ID**	BS00311
Class	DA06201	**Assessor name**	Mr. Nguyen Van Quang

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Student Declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I declare that the work submitted for assessment has been carried out without assistance other than that which is acceptable according to the rules of the specification. I certify I have clearly referenced any sources and any artificial intelligence (AI) tools used in the work. I understand that making a false declaration is a form of malpractice.

Student’s signature	Quy

Grading grid

P3	P4	M2	M3	D2

OBSERVATION RECORD

<table>
<tr><td><b>Student</b></td><td><b>Hoang Anh Quy</b></td></tr>
<tr><td colspan="2"><b>Description of activity undertaken</b></td></tr>
<tr><td colspan="2">P2. Examine the processes of data-driven decision-making (DDDM) when using complex data sets:
• Define Clear Objectives and Goals.
• Data Collection and Integration.
• Data Preprocessing
• Exploratory Data Analysis (EDA)
• Modeling and Analysis
• Decision-Making Criteria and Scenarios.
• Visualization to Share Insights.
• Implement and Monitor Decisions.
• Continuous Feedback and Iteration.</td></tr>
<tr><td colspan="2"><b>Assessment & grading criteria</b></td></tr>
</table>

How the activity meets the requirements of the criteria			
Student signature:	**Quy**	**Date:**	December 11th, 2024
Assessor signature:		**Date:**	
Assessor name:	Mr. Nguyen Van Quang		

r Summative Feedback:

r Resubmission Feedback:

Grade:	Assessor Signature:	Date:

Internal Verifier’s Comments:
Signature & Date:

Table of Contents

List of Figures:

List of Tables

Introduction.

In today's data-driven world, big data plays a crucial role in decision-making processes for organizations and end-users alike. This assignment aims to explore the fundamental concepts of big data, emphasizing its relevance to data-driven decision-making (DDDM). It will also examine the processes involved in DDDM, the roles and responsibilities of data specialists, the challenges they face, and the tools and techniques used in industry for big data manipulation and visualization. By investigating these aspects, this assignment provides insights into how big data influences business decisions and how organizations can leverage it to drive growth and efficiency. Furthermore, the role of statistical and graphical tools, as well as real-time analytics, will be analyzed to highlight the importance of scalable systems for big data processing and decision-making. Through case studies and real-life applications, this paper demonstrates how data manipulation and automation contribute to informed decision-making, particularly in the context of predictive modeling and visualization.

ASM part 1.

1 Explain the fundamental concepts of big data and its value in decision-making for end users and organizations. P1

1.1 The fundamental concepts of big data

The "Vs" of big data describes the key characteristics that differentiate big data from traditional data management. These elements are foundational in understanding the complexities and potential of big data. (segment, n.d.), (Firican, 2017), (Staff, 2024), (Technologies, 2023)

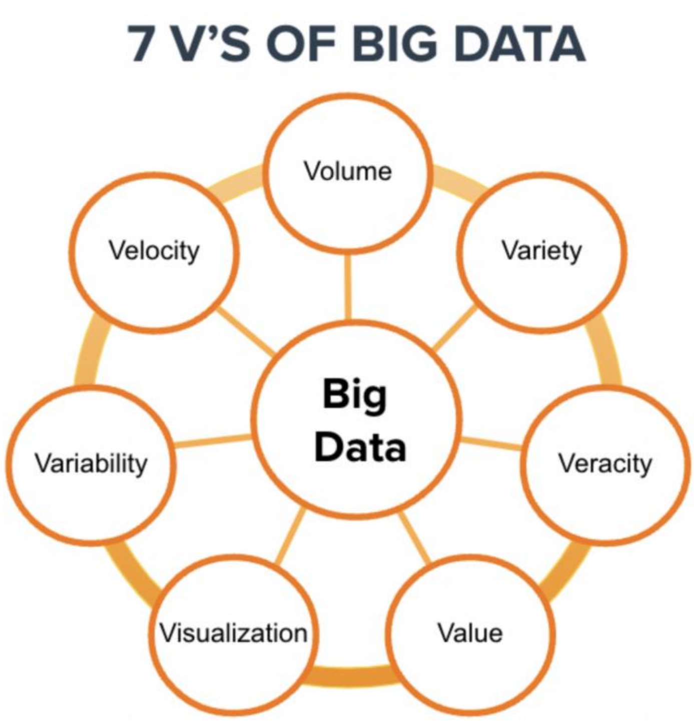


Figure 1 7 "Vs" of big data.

- **Volume**
  - Volume refers to the amount of data generated (at a minimum, many terabytes but as much as petabytes). Refers to the vast amount of data generated every second. For example, social media platforms, IoT devices, and online transactions produce enormous volumes of data.
  - The staggering amount of data available today can create a significant resource burden on organizations. Storing, cleaning, processing, and transforming data requires time, bandwidth, and money.
  - For data engineers, this increased volume will require them to consider scalable data architectures and appropriate storage solutions, as well as how to handle temporary data spikes (like what an e-commerce company might experience during holiday sales).
- **Velocity**
  - The word velocity means "speed," and in this context, the speed at which data is being generated and processed. Real-time data processing plays an important role in this regard, as it processes data as it's generated for instantaneous (or near-instantaneous) insight. Weather alerts, GPS tracking, sensors, and stock prices are all examples of real-time data at work. Of course, when working with huge datasets, not everything should be processed in real-time. This is one of the considerations an organization would have to think through, what should be processed in real time vs. batch processing?
  - Distributed computing frameworks and streaming processing frameworks like Apache Kafka or Apache Flink have become useful in managing data velocity.
- **Variety**

    - Data diversity is another attribute of big data, encompassing structured, unstructured, and semi-structured data (e.g., social media feeds, images, audio, shipping addresses). Organizations will need to map out:
        - How they plan to integrate these different data types (e.g., ETL or ELT pipelines).
        - Schema flexibility (e.g., NoSQL databases).
        - Data lineage and metadata management.
        - How data will be made accessible to the larger organization via business reports, data visualizations, etc.

- **Variability**: Variability in big data is an important aspect to understand. It is the change and inconsistency of data over time and context. Some specific points about Variability include:
    - Time Variation: Data can vary by hour, day, season, or special events. For example, sales data can spike during holidays or drop during periods when there are no special events.
    - Structure Variation: Data can come from many different sources in different formats such as text, images, video, or audio. The structure of the data can change as the source systems or data collection processes change.
    - Quality Inconsistency: Data can have anomalies, omissions, or biases, which requires cleaning and normalization of the data before analysis.
    - Context Variation: Data can change in meaning and value depending on the context in which it is collected. For example, a user's opinion of a product may change over time and over the situations in which the product is used.
- **Veracity**
    - For all the effort that goes into data collection, processing, and storage, if there are any inconsistencies or errors (like data duplicates, missing data, or high latencies) then data’s usefulness quickly erodes.
    - Veracity refers to the accuracy, reliability, and cleanliness of these large data sets. Ensuring data veracity comes down to good data governance, and implementing best practices like:
        - Automating QA checks and flagging data violations in real time
        - Adhering to a single tracking plan
        - Standardizing naming conventions
- **Visualization**
    - Data visualization is essential for data analytics since it involves displaying the analyzed data in a way that is understandable to a visual audience. In today’s world, visualization is essential. Large

volumes of complex data can be visualized using graphical representations far more effectively than using reports and Excel sheets that are filled with figures and calculations.

  - Raw data must be presented appropriately to be used and leveraged. Numerous forms, such as Word documents, Excel files, graphs, and others, can be used to present data. Data visualization is vital for making the information understandable, accessible, and easy to access, regardless of its format.

- **Value**
  - As the name suggests, Value refers to the actionable insights that can be gleaned from large data sets. While it may seem like massive amounts of data will automatically lead to even greater insights, without the right processing, validation, and analytics frameworks, it can be difficult to extract value. (Hence the need for the four Vs above.)
  - This is where AI and machine learning can come in, helping to extract lessons learned and actionable items at a rapid pace (e.g. predictive analytics or prescriptive analytics).
  - Another important aspect of making data valuable is making it accessible across teams, as with self-service analytics.

1.2 The relevance of basic concepts of big data to processes of data-driven decision-making (DDDM)

Volume

- With large volumes of data, organizations can gain detailed insights into customers and markets. This rich data helps create predictive models and deeper analysis.
- However, to handle large volumes, powerful storage and analysis technologies such as Big Data and Cloud Computing are needed.

Velocity (Speed of Data)

- Real-time data supports organizations in rapid decisions such as fraud detection, supply chain management, and real-time price optimization.
- This speed also requires systems that can collect and process data immediately, helping to create timely responses.

Variety

- Variety provides decision-makers with a comprehensive and multidimensional picture. For example, an organization can use data from customer reviews, email responses, and social media interactions to gain a better understanding of customer expectations.
- However, this variety also requires sophisticated analytics tools that can combine and clean different formats to create a unified data set for analysis.

Variability

- Predictive models in DDDM need to be able to recognize and adapt to this variability in order to make accurate forecasts. For example, retail businesses need to prepare for increased demand during the holiday shopping season.
- To mitigate the risk of variability, companies need to rely on historical data analysis and have flexible models that can adapt quickly to changes.

Veracity

- Data accuracy is the foundation of DDDM. If data is inaccurate, decisions can be wrong, causing financial loss and reputational damage.
- To ensure reliability, organizations need to have a data cleaning and auditing process that helps eliminate errors or noise and improve the reliability of input data.

Visualization

- Visualization helps decision-makers easily grasp information and trends from complex data. A good visualization report can help stakeholders quickly understand key points and draw conclusions.
- In DDDM, visualization also helps uncover previously unseen trends and relationships, which form the basis for strategic decisions.

Value

- The value of data lies in its ability to improve operational efficiency, increase revenue, and optimize strategies. For example, data on market trends can help a company capture new opportunities or optimize products.
- The DDDM process leverages the value of data to make impactful decisions that maximize organizational benefits and increase competitiveness.

2 Examine the processes of data-driven decision-making (DDDM) when using complex data sets. P2

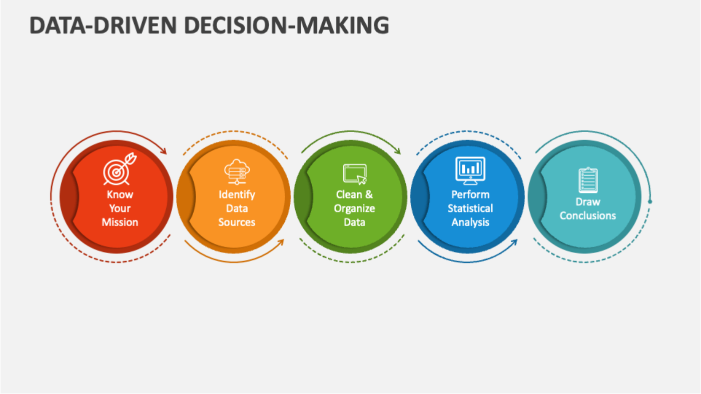


Figure 2 the processes of data-driven decision-making (DDDM).

Data-Driven Decision Making (DDDM) relies on the scientific analysis of complex data sets to guide decisions, drawing out insights that can deliver tangible value to an organization. Here is a 9-step breakdown to understand how DDDM works. These nine steps collectively enable robust DDDM, using complex datasets to inform strategic choices that propel organizational growth and success.

2.1 Define Clear Objectives and Goals.

The foundation of effective DDDM lies in setting clear goals. Before you begin collecting or analyzing data, identify specific business goals and objectives that will guide the entire process. The goal might be to increase revenue, improve performance, or optimize a process. For example, if the goal is to increase customer satisfaction, all steps from data collection to analysis should be geared toward making decisions that improve the customer experience. Clear goals prevent the collection and analysis of irrelevant data, aligning every stage of DDDM with actionable results. (Ticong, 2024)

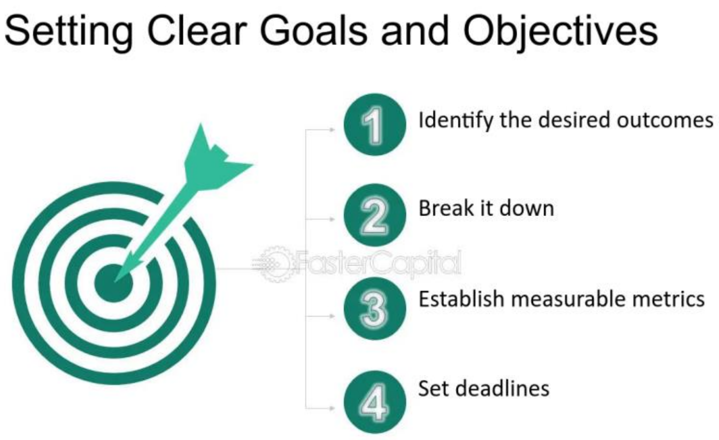

Setting Clear Goals and Objectives
1
Identify the desired outcomes
2
Break it down
3
Establish measurable metrics
4
Set deadlines
FasterCapital

Figure 3 Define Clear Goals and Objectives.

2.2 Data Collection and Integration.

In this phase, data from multiple sources such as databases, online surveys, CRM systems, or transaction logs is gathered. Since data often originates from different systems with unique formats, integration is essential. (Mucci, 2024)This involves:

- Data Cleaning: Removing duplicates, filling missing values, and correcting inaccuracies.
- Data Transformation: Converting data into a consistent format and structure, enabling compatibility.
- Handling Inconsistencies: Addressing discrepancies to create a unified dataset suitable for analysis. This step is crucial for obtaining a complete and reliable dataset to inform decision-making.

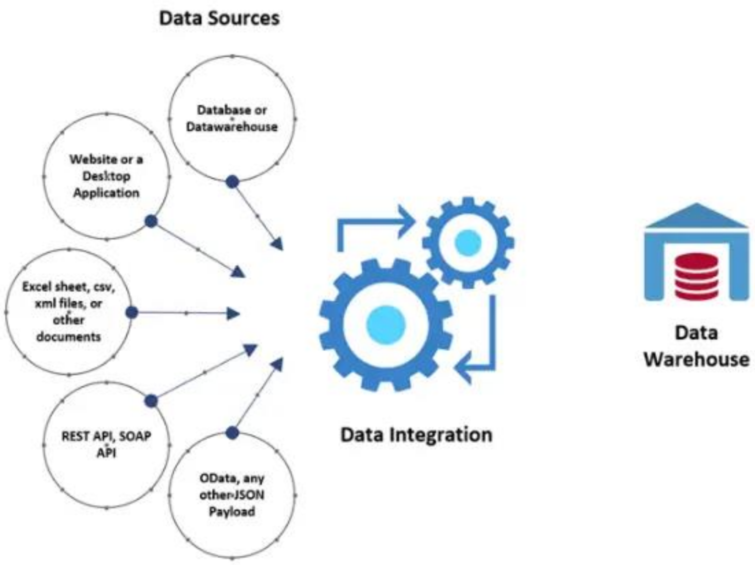

Data Sources
Database or Datawarehouse
Website or a Desktop Application
Excel sheet, csv, xml files, or other documents
REST API, SOAP API
OData, any other JSON Payload
Data Integration
Data Warehouse

Figure 4 Data Collection and Integration.

2.3 Data Preprocessing

Data preprocessing ensures that data is suitable for analysis and helps maintain quality:

- Data cleaning: Dealing with missing, duplicate, or incorrect values that can distort insights.
- Data transformation: Converting raw data into forms that are more conducive to analysis, such as normalizing or encoding data by category.
- Feature engineering: Creating new, insightful features from raw data, such as day of the week, month, or quarter from date data. Or calculating customer lifetime value from transaction data, providing more nuanced input for analysis.

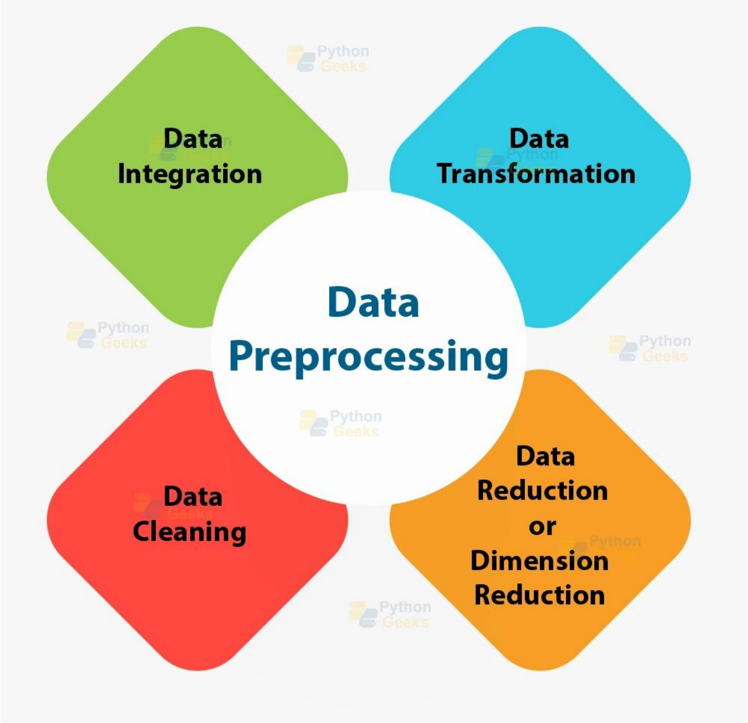
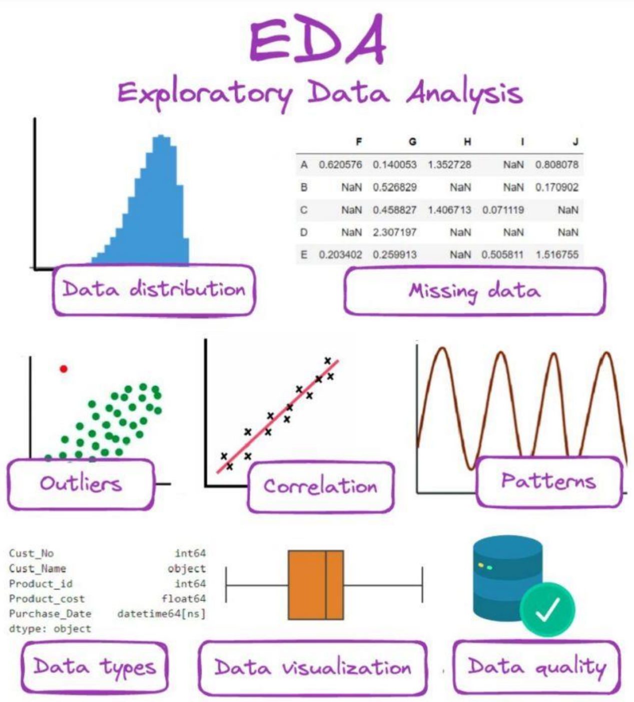


Figure 5 Data Preprocessing.

2.4 Exploratory Data Analysis (EDA)

EDA enables a deeper understanding of data sets and reveals key patterns: (Insights, 2024)

- Identify patterns and trends: Visualizations such as histograms, box plots, and scatter plots make patterns clear and intuitive, helping to discover patterns, trends, and relationships in data.
- Dimensionality reduction: Techniques such as Principal Component Analysis (PCA) help simplify complex data sets, focusing on the most informative variables. This step is crucial to making high-dimensional data more meaningful and insightful.



	F	G	H	I	J
A	0.620576	0.140053	1.352728	NaN	0.808078
B	NaN	0.526829	NaN	NaN	0.170902
C	NaN	0.458827	1.406713	0.071119	NaN
D	NaN	2.307197	NaN	NaN	NaN
E	0.203402	0.259913	NaN	0.505811	1.516755

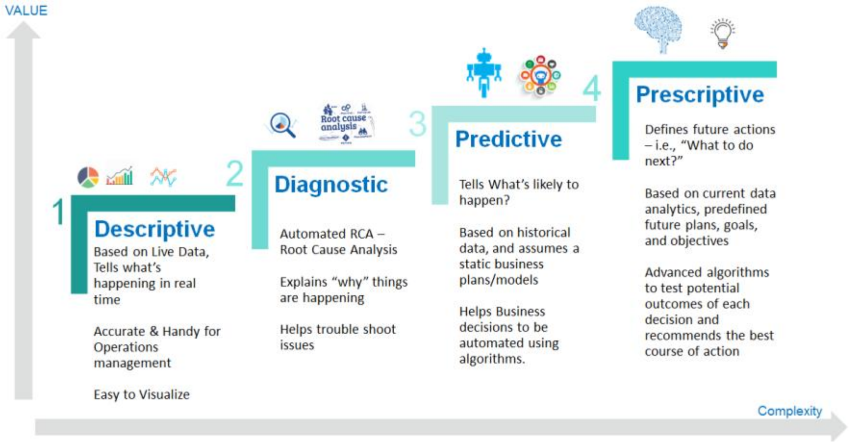
```
Cust_No                    int64
Cust_Name                 object
Product_id                 int64
Product_cost             float64
Purchase_Date     datetime64[ns]
dtype: object
```

Figure 6 Exploratory Data Analysis (EDA).

2.5 Modeling and Analysis

Different models and analytical techniques are applied to achieve various decision-making objectives:

- **Predictive Modeling:** Utilizes algorithms like linear regression or decision trees to forecast future outcomes (e.g., sales predictions).
- **Descriptive Modeling:** Summarizes patterns in historical data, giving an overview of trends or customer behavior.
- **Diagnostic Analysis:** Examines cause-and-effect relationships to understand why certain trends occur. For example, diagnosing why sales dropped can inform corrective actions.



Descriptive
Based on Live Data, Tells what's happening in real time

Accurate & Handy for Operations management

Easy to Visualize

Diagnostic
Automated RCA – Root Cause Analysis

Explains "why" things are happening

Helps trouble shoot issues

Predictive
Tells What's likely to happen?

Based on historical data, and assumes a static business plans/models

Helps Business decisions to be automated using algorithms.

Prescriptive
Defines future actions – i.e., "What to do next?"

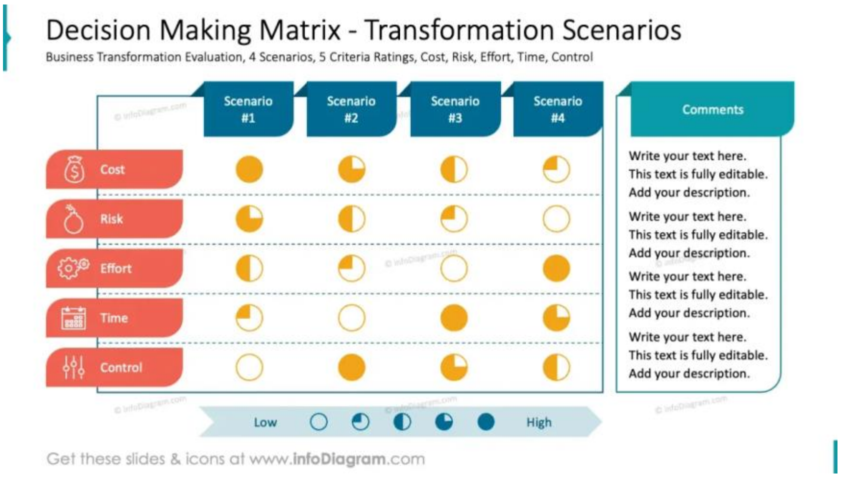
Based on current data analytics, predefined future plans, goals, and objectives

Advanced algorithms to test potential outcomes of each decision and recommends the best course of action

Figure 7 Modeling and Analysis.

2.6 Decision-Making Criteria and Scenarios.

Establishing decision-making criteria ensures that insights are actionable. Criteria could include cost efficiency, impact on customer satisfaction, or revenue growth. By setting clear scenarios (e.g., "If customer churn exceeds 5%, consider loyalty programs"), decisions become structured and guided by data-based thresholds.



Decision Making Matrix - Transformation Scenarios

Business Transformation Evaluation, 4 Scenarios, 5 Criteria Ratings, Cost, Risk, Effort, Time, Control

	Scenario #1	Scenario #2	Scenario #3	Scenario #4	Comments
Cost	●	◕	◑	◔	Write your text here. This text is fully editable. Add your description.
Risk	◕	◑	◔	○	Write your text here. This text is fully editable. Add your description.
Effort	◑	◔	○	●	Write your text here. This text is fully editable. Add your description.
Time	◔	○	●	◕	Write your text here. This text is fully editable. Add your description.
Control	○	●	◕	◑	Write your text here. This text is fully editable. Add your description.

Low ○ ◔ ◑ ◕ ● High

Get these slides & icons at www.infoDiagram.com

Figure 8 Decision-Making Criteria and Scenarios.

2.7 Visualization to Share Insights.

Visualizations, such as dashboards or interactive reports, make insights accessible to stakeholders. For complex data, tools like Tableau or Power BI can highlight trends and comparisons. Charts, tables, and dashboards help communicate information clearly and effectively. Clear visualizations help communicate findings to a non-technical audience, making data-driven insights actionable and transparent across departments.

Visual Analytics: Simplify Complex Data with Insightful Visuals

Figure 9 Visualization to Share Insights.

2.8 Implement and Monitor Decisions.

Once a decision is made based on data insights, it is implemented, and outcomes are monitored over time. For instance, if a data-driven marketing campaign is launched, tracking its performance with metrics like conversion rates or engagement ensures it aligns with original objectives. Monitoring also allows for immediate adjustments if outcomes diverge from expectations.

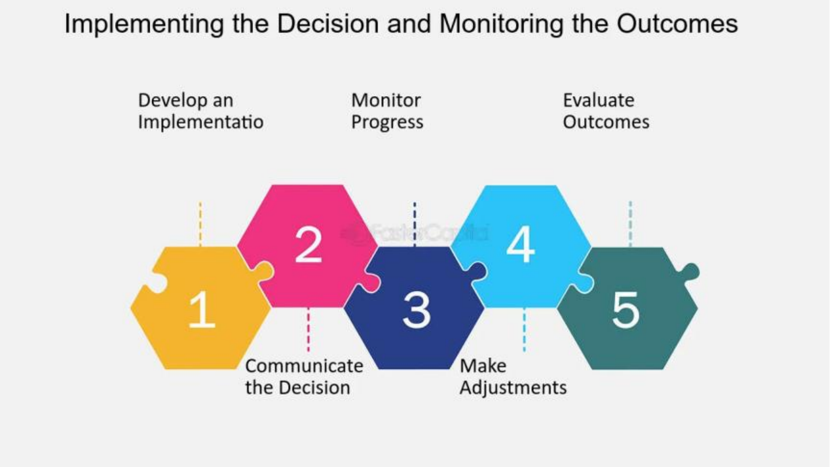


Figure 10 Implement and Monitor Decisions.

2.9 Continuous Feedback and Iteration.

DDDM is iterative; decisions are continuously refined as new data emerges. Feedback loops help assess the effectiveness of each decision, enabling adjustments and identifying areas for improvement. This cycle of continuous learning drives ongoing alignment with business goals, ensuring that each iteration leads to more accurate and effective outcomes.

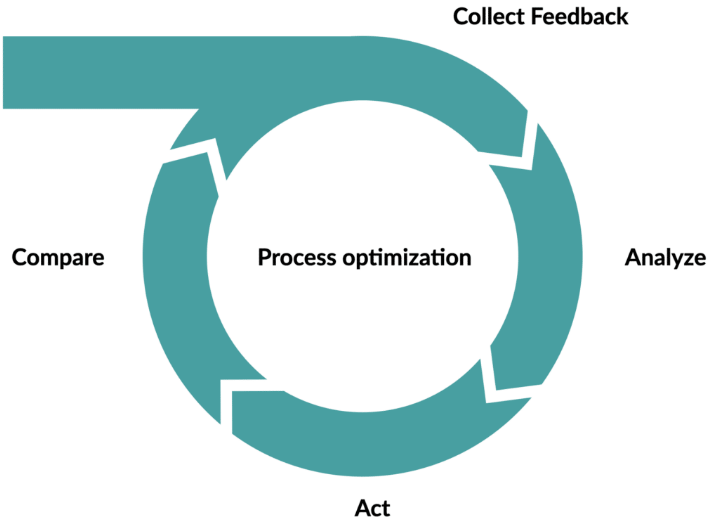


Figure 11 Continuous Feedback and Iteration.

3 Investigate the roles, responsibilities, and key issues faced by data specialists in their day-to-day roles at Express Shipping. P5

3.1 Explain roles in a data-driven industry.

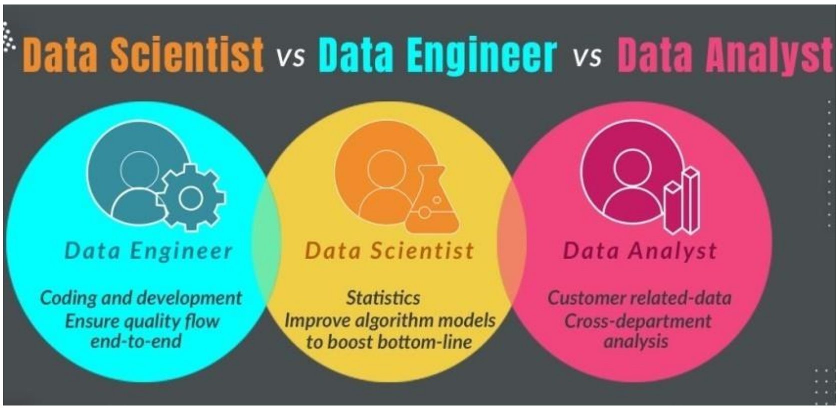

Data Scientist vs Data Engineer vs Data Analyst
Data Engineer
Coding and development
Ensure quality flow end-to-end
Data Scientist
Statistics
Improve algorithm models to boost bottom-line
Data Analyst
Customer related-data
Cross-department analysis

Figure 12 The roles of a data specialist.

Data Analyst: Analyzes large data sets to identify patterns, trends, and insights that can support decision-making. For Express Shipping, data analysts might look into trends like shipment delays, customer inquiries, or seasonal demand.

Data Scientist: Uses machine learning and statistical methods to build predictive models that forecast shipping demands or optimize delivery routes.

Data Engineer: Develops and maintains data pipelines, ensuring that data flows smoothly between systems. At Express Shipping, data engineers might set up pipelines to collect data from customer interactions, shipping logs, and inventory systems.

Data Visualization Specialist: Creates dashboards and visual reports to present data in a way that is easy for stakeholders to interpret. This could involve visualizing KPIs like on-time delivery rates or customer satisfaction scores.

Data Administrator: Manages the storage, accessibility, and security of data. This role is essential for protecting sensitive customer data and ensuring compliance with data regulations.

Business Analyst: Acts as the bridge between data teams and business units, identifying business needs and translating them into data requirements. In logistics, this role can help set priorities for data projects based on business objectives. (Keshari, 2023)

3.2 Explore the responsibilities of a data specialist.

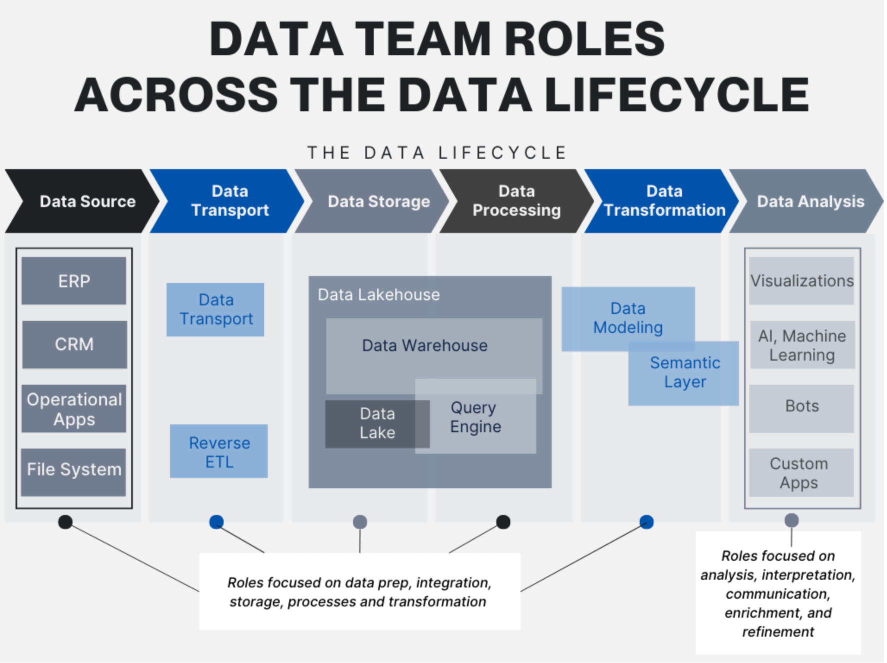


Figure 13 The responsibilities of a data specialist.

Data Preparation: Involves cleaning and organizing raw data to ensure it is accurate and ready for analysis, a crucial step for reliable insights.

Data Analysis: Reviewing data to identify trends or anomalies, enabling the company to make informed business decisions, such as identifying potential causes for delivery delays.

Data Modeling: Building statistical or machine learning models to predict trends and outcomes, such as expected delivery times based on traffic and weather data.

Data Management: Overseeing the collection, storage, and quality of data, including maintaining storage solutions that comply with data security and privacy regulations.

Data Visualization: Transforming complex data into visual formats, allowing for quick insights and decision-making.

Storage and Access Rights: Implementing and enforcing policies to manage data access rights, ensuring that only authorized personnel can access sensitive data.

3.3 Understand challenges.

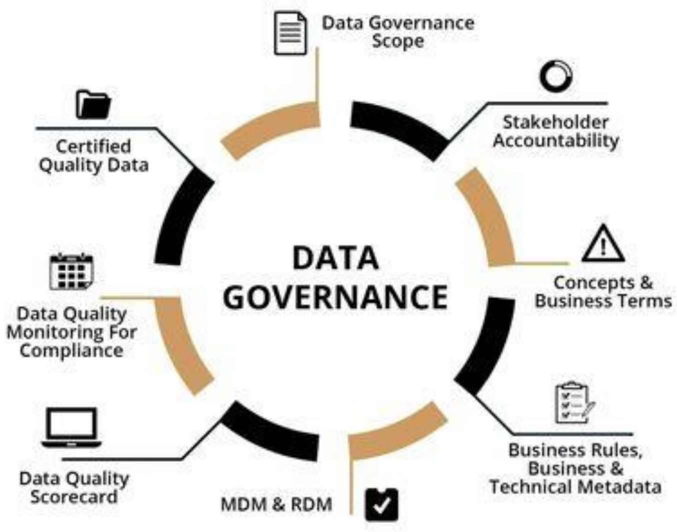

Data Governance Scope
Stakeholder Accountability
Concepts & Business Terms
Business Rules, Business & Technical Metadata
MDM & RDM
Data Quality Scorecard
Data Quality Monitoring For Compliance
Certified Quality Data
DATA GOVERNANCE

Figure 14 The challenges of a data specialist.

Data Governance Framework: Implementing and maintaining a data governance framework that ensures the value of outcomes, accountability, and trust. This involves defining roles, responsibilities, and standards for data usage.

Accountability and Trust: Ensuring that data is accurate and handled responsibly to build trust within the company and with customers.

Collaboration and Transparency: Promoting cross-departmental collaboration, as data often needs to flow seamlessly between multiple departments to optimize processes.

Security and Risk Management: Protecting data against breaches and ensuring compliance with regulations like GDPR, which is especially important given the sensitive nature of customer data.

Role of the Data Steward: A data steward plays a vital role in maintaining data quality, implementing governance policies, and ensuring data consistency and compliance.

ASM part 2.

4 Discuss statistical and graphical tools and techniques used in industry for big data manipulation and visualization. P3

4.1 Statistical Tools and Techniques.

Statistics play a vital role in big data analytics, helping solve complex problems, from summarizing data to predicting trends. These techniques provide accurate information and create a scientific basis for decision-making and explaining phenomena. (Valcheva, n.d.), (Ninja, 2024)

4.1.1 Descriptive Statistics.

Descriptive statistics is a technique used to summarize and organize data, helping convey information visually and easily. (Bhandari, 2023)

- Common tools include Mean, Median, Mode.
- Frequency distribution: Visualized by graph or table.
- Standard deviation and variance: Assess the level of variation.

Importance and uses:

- Increase analytical efficiency: Help grasp data quickly.
- Support reporting and decision-making: Create a foundation to understand data before applying more complex techniques.

4.1.2 Inferential Statistics

Inferential statistics allows conclusions to be drawn about a population based on sample data, thereby supporting decision making when it is not possible to analyze the entire data. (Bhandari, 2023)

Main methods and tools

- Parameter estimation: Confidence Interval: Determines the range of values within which the population parameter can fall.
- Hypothesis Testing: Testing relationships or differences between groups (e.g. t-test, ANOVA).
- Samples and sampling distributions: Using sample data to create representative models of the population.

Importance and significance

- Saving resources: No need to analyze all large data, reducing costs and time.

- Scientific decision making: Making decisions based on analysis instead of guesswork.

Practical applications

- Finance: Predicting market trends from trading patterns.
- Marketing: Identify customer behavior from surveys.

4.1.3 Regression Analysis

Regression analysis studies the relationships between variables. It is beneficial for predicting the value of one variable based on another.

Common Types of Regression Analysis

- Linear Regression: A linear model for predicting continuous values (e.g., sales).
- Logistic Regression: Used for categorical data (e.g., whether a purchase was made).
- Multiple Regression: Examines the relationship between a dependent variable and multiple independent variables.

Importance and Significance

- Understanding Causality: Determining the Impact of One Factor on Another.
- Predicting Outcomes: Applications in Planning and Strategy.

Applications of Regression Analysis

- Prediction: Forecasting home prices, sales, weather, etc.
- Optimization: Find out the important factors that affect the outcome (dependent variable).
- Decision making: Support business, marketing, medical, financial analysis.
- Medical: Analyze the impact of diet on health.

4.1.4 Clustering

Clustering is a data grouping technique in machine learning and data analysis. The goal of clustering is to divide data into groups (clusters) such that points in the same group are most similar to each other, and most dissimilar to points in other groups. (Lien, 2024)

Main clustering methods:

- K-Means Clustering: The most popular method. Focuses on grouping data into k pre-specified clusters based on distance to the cluster center. It works in the following steps:
  - Choose k initial centroids.
  - Assign each data point to the nearest cluster based on distance (usually Euclidean distance).
  - Update the centroid of each cluster.
  - Repeat this process until the clusters no longer change.
- Hierarchical Clustering: Hierarchically divide data into a clustering tree (dendrogram). No need to pre-select the number of clusters. There are two types:
  - Agglomerative: Each starting point is a separate cluster, then gradually merges smaller clusters.
  - Divisive: The entire data starts as a large cluster, then gradually divides into smaller clusters.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Find irregular clusters based on the density of data points. Identify clusters based on high density areas, ignoring noise points. No need to pre-select the number of clusters.

Importance and significance

- Detect hidden patterns: Find potential relationships between objects.
- Automatic classification: Reduce manual intervention when working with big data.

Practical applications

- Marketing: Segment customers to personalize campaigns.
- Medicine: Classify patients based on symptoms.

4.1.5 Dimensionality Reduction

Dimensionality Reduction is a technique used in data processing and machine learning to reduce the number of variables or attributes (dimensions) in a dataset, while still trying to retain as much important information as possible. (Jacob Murel Ph.D., Eda Kavlakoglu, 2024)

Main Methods

- Principal Component Analysis (PCA): Transforms data from the original space to a new space by finding the principal components - the directions in which the data has the largest dispersion. PCA reduces dimensionality by retaining the k principal components with the largest variance. Applied in image processing, data analysis.

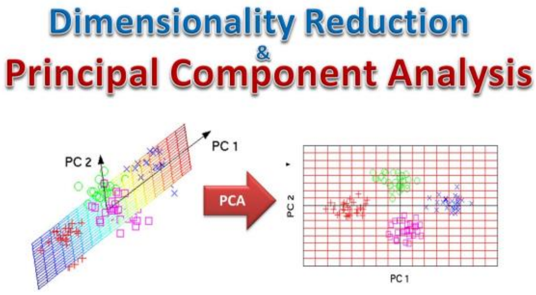


Figure 15 Principal Component Analysis (PCA).

- t-SNE (t-Distributed Stochastic Neighbor Embedding): Reduces the dimensionality of nonlinear data into 2D or 3D space, focusing on keeping the distance relationship between points. Good for visualizing complex data. However, the computation time is long and not suitable for large datasets.

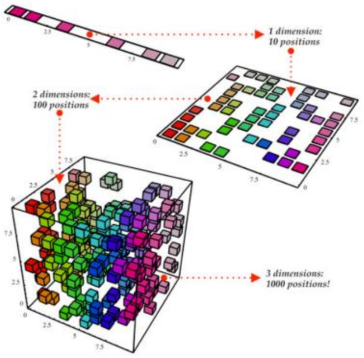


Figure 16 t-SNE (t-Distributed Stochastic Neighbor Embedding).

Importance and Significance

- Avoid redundancy: Eliminate unnecessary information.
- Reduce complexity: High-dimensional data is difficult to process, difficult to visualize, and prone to the curse of dimensionality.
- Increase efficiency: Reducing dimensionality reduces computation time and improves model performance.
- Remove noise: Reduce unimportant or highly correlated attributes, making data easier to analyze.

Practical applications

- Scientific research: Analyze genomic data.
- E-commerce: Accelerate customer behavior analysis.

4.1.6 Time Series Analysis

Time series analysis is a technique used to explore, model, and predict data that changes over time. Time series data is typically collected as observations in a time order, such as monthly sales, daily stock prices, or hourly temperatures. (Grewcoe, 2024)

Characteristics of a time series:

- Time order: The order of the data is important and cannot be changed or mixed up like regular panel data.
- Time dependence: Values at one point in time may depend on previous values.

Components of a time series:

- Trend: Long-term direction of increase or decrease in data.
- Seasonality: A pattern that repeats periodically (daily, weekly, monthly, or yearly).
- Noise: Random fluctuations that do not follow a regular pattern.
- Cyclic: Variations that occur without a fixed schedule, often related to economic factors.

Steps of time series analysis:

- Explore the data: Draw a line plot to observe the trend. Also check for the presence of seasonality or cycles.
- Stationarity analysis: Stationary data has a constant distribution over time, i.e. no trend or seasonality. Use tests such as ADF test (Augmented Dickey-Fuller) or KPSS test to check for stationarity. If the data is not stationary, use methods such as differencing to make it stationary.
- Separate the components of the time series: Use decomposition to separate the trend, seasonality and noise.
- Time series modeling:
  - ARIMA (AutoRegressive Integrated Moving Average): Suitable for stationary and nonlinear data.
  - SARIMA (Seasonal ARIMA): Extended ARIMA to handle seasonal data.
  - Holt-Winters: Smoothing method for forecasting with trends and seasonality.
  - Prophet (Facebook): Suitable for nonlinear and complex cyclical data.
- Prediction:
  - Split the dataset into train-test split to check the performance of the model.
  - Calculate evaluation metrics: MAE (Mean Absolute Error), RMSE (Root Mean Square Error).

- Evaluation and improvement:
    - Check residuals to ensure that they do not contain trends or seasons.
    - Adjust model parameters to improve accuracy.

Importance and significance

- Resource management: Helps plan based on future demand.
- Accurate prediction: Strong application in finance, and manufacturing.

Practical application

- Business: Forecast seasonal sales.
- Inventory management: Determine purchasing cycles.
- Financial analysis: Predict stock prices, and interest rates.
- Climate monitoring: Analyze temperature, and rainfall over time.
- Anomaly detection: Find out abnormal fluctuations.

4.2 Graphical Tools and Techniques.

4.2.1 Visualization Libraries: Python Libraries

4.2.1.1 Matplotlib.

Matplotlib is a popular graphing library in Python that helps you create static charts and visualizations from data, providing granular control over each element. The library supports a wide variety of chart types such as line, bar, scatter, histogram, heatmap, and more. (Flare, 2024)

Main Components of Matplotlib

- Figure: The entire area that contains the chart.
- Axes: The area where the data is plotted, usually a chart.
- Axis: The x or y axis in an Ax.
- Plot: The graph, such as a line, column, or scatter, that is plotted on the Axes.

Pros:

- Highly customizable (can edit individual points, axes, labels, text, etc.).
- Supports many chart types.
- Suitable for creating high-quality charts for publishing.

Cons:

- No interactivity.
- Difficult to use if advanced customization is needed.

Best Uses:

- Static charts in reports, research.
- Charts that require detailed customization.
- Basis for other advanced libraries (like Seaborn).

4.2.1.2 Seaborn.

Seaborn is a powerful Python library for data visualization, built on Matplotlib. It makes it easy to create beautiful charts and supports working with Pandas data structures like DataFrame. (Mutea, 2022)

Pros:

- Supports statistical charts like distribution plots, regression plots, boxplots, heatmaps, etc.
- Integrates well with Pandas DataFrame.
- Provides friendly chart themes.
- Automatically handles colors based on categorical or numeric data.
- Supports building complex charts like FacetGrid, Pairplot.
- Default interface is nicer than Matplotlib.

Cons:

- Less flexible than Matplotlib when high customization is needed.
- No interaction support.

Best apps:

- Statistical data charts.
- Analyze classified or aggregated data.
- Explore data in the early analysis (EDA) phase.

4.2.1.3 Plotly.

Plotly is a powerful library for creating interactive charts, data visualizations, and supports many different chart types, such as line charts, bar charts, pie charts, heat maps, 3D charts, and more, with zoom, pan, and live updates. (Lianne & Justin, 2020)

Pros:

- Plotly allows for high customization such as adding titles, axis labels, or combining multiple charts in one view (subplots). Editing the appearance, such as colors, fonts.
- Supports 3D charts, maps, and dashboards.
- Integrates easily with web applications (such as Dash).

Cons:

- Large library size, higher resource requirements.
- More complex syntax than Seaborn for simple tasks.

Best for:

- Interactive dashboards and data exploration.
- Web-based reporting and presentations.
- When interaction (hover, zoom) is needed in the chart.

4.2.1.4 Comparison between Matplotlib, Seaborn, and Plotly.

Table 1 Comparison between Matplotlib, Seaborn, and Plotly

Feature	Matplotlib	Seaborn	Plotly
Ease of Use	Moderate	Easy	Moderate
Customization	High	Moderate	High
Interactivity	None	None	High
Built-in Statistics	Limited	Extensive	Moderate
Plot Aesthetics	Basic	Enhanced	Highly Polished
Performance	Fast	Fast	Slower (due to interactivity)

Use Matplotlib for publication-quality static plots and fine-grained customizations.

Use Seaborn for quick, visually appealing statistical plots during EDA.

Use Plotly for interactive visualizations and web-based presentations.

4.2.2 Interactive Visualization Tools.

4.2.2.1 Tableau.

Tableau is a powerful data visualization and analysis software that helps users transform raw data into easy-to-understand dashboards and charts. Developed by Tableau Software, it provides a powerful platform for exploring, analyzing, and sharing data from a variety of sources such as SQL databases, Excel, Google Analytics, and cloud services like Salesforce. (Tableau, n.d.)

Notable Features:

- Easy to Use: The easy drag-and-drop interface allows users without programming skills to create interactive charts and dashboards.
- Advanced Analytics: Supports time-series analysis, classification analysis, and integration with multiple data sources.
- Customization: While Tableau allows for customizing charts and dashboards, the level of flexibility may not be as great as Power BI or D3.js. However, the filtering and drill-down features help users dig deeper into data.

Intended use: Tableau is suitable for non-technical users who need powerful tools to analyze and visualize data easily and quickly. In addition, Tableau supports sharing and collaboration via Tableau Server or Tableau Online.

Figure 17 Tableau.

4.2.2.2 Power BI.

Power BI is a data analysis and visualization tool developed by Microsoft that allows users to connect to multiple data sources, build dynamic reports and dashboards, and share them in cloud or local environments. Power BI integrates tightly with other Microsoft tools such as Excel, Azure, and SharePoint, making it easy for organizations to mine and analyze data from internal and cloud systems. (Aktualisierte, 2024)

Notable features:

- Ease of use: The interface is similar to Tableau with drag and drop capabilities, easy to get used to, especially for users familiar with the Microsoft ecosystem.
- Data analysis features: Power BI supports real-time data analysis and allows users to build data models and write analytical formulas with DAX (Data Analysis Expressions).
- Customization: Power BI offers custom charts and plugins from the community, but the level of customization may be limited compared to Tableau and D3.js.

Intended use: Power BI is well suited for organizations that are already using Microsoft tools and need a powerful data visualization and analytics solution that also offers strong sharing and collaboration options.

Figure 18 Power BI.

4.2.2.3 D3.js

D3.js is a powerful JavaScript library for creating dynamic and customizable data visualizations on the web. D3.js allows developers to create complex charts and graphics such as force graphs, hierarchical charts, and geographic maps, with powerful interactivity. Unlike Tableau and Power BI, D3.js requires users to have knowledge of web programming and JavaScript, providing maximum flexibility in designing and customizing the look and feel of charts and features. (Tutorialsteacher, n.d.)

Notable features:

- Highly flexible and customizable: D3.js gives users complete control over the look and feel of data, creating completely customized charts and graphics.
- Strong interactivity: Supports interactive features such as drag and drop, zoom, and many other dynamic effects.
- Easy integration with the web: D3.js uses HTML, SVG, and CSS to build complex visualizations that can be integrated into websites.

Intended use: D3.js is the ideal tool for web developers who want to create highly customizable and interactive charts for web applications or research projects that require flexibility and dynamism.

Figure 19 D3.js

4.3 Big Data Manipulation Tools.

4.3.1 SQL-based Systems: Apache Hive.

Apache Hive is a distributed data management system built on Hadoop. It provides a SQL-like interface that allows users to query and manage data stored on HDFS (Hadoop Distributed File System). Hive was originally developed by Facebook to solve large-scale data query problems. Hive uses a query language with a SQL-like syntax called HiveQL, which allows users to perform complex queries on distributed data without having to write MapReduce code directly. (aws, n.d.)

Figure 20 Apache Hive.

Notable features:

- SQL-like interface: Hive provides a query syntax similar to SQL, which is easy for people who are familiar with SQL to use. This helps reduce the complexity of working with Hadoop.
- Big data processing capability: Hive is designed to operate on the Hadoop system, which can process and analyze very large volumes of data that traditional database systems cannot do.
- Support for complex data types: Hive can store and query a wide range of data types, including tabular (structured), semi-structured, and unstructured data.

- Highly scalable: Hive is distributed and can scale easily thanks to its ability to run on Hadoop. This helps handle petabytes of data without performance issues.

Key features:

- HiveQL: Hive's query language has a syntax similar to SQL, allowing users to perform tasks such as querying, filtering, grouping, and creating tables.
- Hadoop integration: Hive is optimized to work with Hadoop, leveraging HDFS and the distributed computing capabilities of Hadoop MapReduce or alternatives such as Apache Tez and Apache Spark.
- Support for semi-structured and unstructured data: Hive can work with data that is not only structured, but can also process semi-structured and unstructured data, such as JSON and XML.
- Flexible scalability: Hive can be configured to run on a Hadoop cluster, making the system capable of handling increasing volumes of data without losing performance.

Advantages:

- Easy to use: With a SQL-like syntax, Hive makes it easy for users familiar with SQL to use without having to learn how to write complex MapReduce code.
- Good integration with Hadoop: Hive takes full advantage of Hadoop in distributed data processing, providing scalability and efficient parallel computing.
- Support for complex queries: Hive can handle complex queries with operations such as join, group by, filter and big data operations.

Disadvantages:

- Low performance for real-time queries: Hive is not ideal for real-time querying requirements, as it primarily uses MapReduce (or Tez, Spark) to process queries, which can cause latency in tasks that require fast responses.
- No full ACID support: While Hive may support some ACID features (such as supporting transactions on tables), it is not as robust as traditional relational database systems.
- No good support for updating and deleting data as relational database management systems: Hive is not optimized for small updates or deletions, which makes it unsuitable for applications that require frequent data updates.

Applications of Apache Hive:

- Big Data Analytics: Hive is used in big data analytics systems where the volume of data is too large to be processed by traditional database tools.
- Business Reporting and Analytics: Organizations use Hive to analyze data and generate detailed reports for business decision making.
- Long-term Data Storage: Hive can store and process data for a long time on Hadoop, suitable for cases where data analysis is required in historical data warehouses.

4.3.2 Distributed Computing Frameworks.

4.3.2.1 Apache Spark

Apache Spark is an open-source distributed computing system designed to process and analyze data at very high speed. Spark was developed by researchers at UC Berkeley and is one of the prominent technologies in the field of big data processing. Spark can run on Hadoop but can also run independently on other distributed systems. Spark stands out with its in-memory data processing ability, which reduces latency and increases processing speed compared to batch processing systems such as Hadoop MapReduce. (Sruthy, 2024)

Figure 21 Apache Spark.

Outstanding features:

- In-memory data processing: Spark stores data in memory instead of having to read and write data from disk like MapReduce. This helps reduce latency and increase processing speed.
- Parallel computing: Spark is designed to process data in parallel on multiple nodes in a cluster, making the most of the resources of distributed systems.
- Scalability: Spark can scale easily from a single machine to thousands of machines in a cluster, helping to process huge amounts of data without performance issues.
- Support for multiple programming languages: Spark supports languages such as Scala, Java, Python, and R, allowing users to choose the programming language that best suits their needs.

- Rich ecosystem: Apache Spark provides a wide range of built-in libraries and tools for tasks such as real-time data processing (Spark Streaming), machine learning (MLlib), graph processing (GraphX), and SQL analytics (Spark SQL).

Key features:

- Spark SQL: Spark SQL allows users to perform SQL queries on structured data, which can access data from a variety of sources such as HDFS, Apache Hive, Apache HBase, and relational database management systems.
- Spark Streaming: Spark Streaming enables real-time data processing, helping users analyze continuous streams of data from sources such as Kafka, Flume, or file systems.
- MLlib: The MLlib machine learning library in Spark provides popular machine learning algorithms such as classification, regression, clustering, and dimensionality reduction, helping users build efficient machine learning models at scale.
- GraphX: Spark also provides GraphX, a graph processing library, allowing users to perform operations on distributed graphs, such as calculating shortest paths, communities, and properties of graphs.
- In-memory computing: Spark stores data in memory (RAM) instead of using hard disk, optimizing processing speed and reducing latency when working with large data sets.

Advantages:

- Fast processing speed: Thanks to its in-memory data processing and parallel computing capabilities, Spark can process much faster than traditional batch processing systems like Hadoop MapReduce.
- Real-time data processing capabilities: With Spark Streaming, users can process and analyze data in real time as it is generated, a feature that Hadoop MapReduce does not support.
- Easy integration: Spark can easily integrate with Hadoop tools like HDFS, HBase, and Hive, as well as other distributed systems.
- Multiple language support: Spark supports many programming languages like Scala, Java, Python, and R, allowing users to use the language they are familiar with.
- Strong ecosystem: Built-in libraries like Spark SQL, MLlib, and GraphX help users perform various tasks without having to use external tools.

Disadvantages:

- Requires large system resources: Since Spark mainly uses RAM to process data, it requires large hardware resources, especially when working with huge data sets.

- Requires technical knowledge: Although Spark has an easy-to-use API, optimizing and handling complex tasks still requires users to have a solid knowledge of programming and distributed systems.
- Complex management and maintenance: When deploying Spark on a large cluster, system management and maintenance can become complex and require good system management tools.

Applications of Apache Spark:

- Big data processing: Spark is widely used in companies to process and analyze big data that traditional systems cannot handle.
- Real-time data: With Spark Streaming, companies can process real-time data from sources such as IoT devices, financial trading systems, or data streams from social networks.
- Machine Learning and Artificial Intelligence: Spark is a powerful tool for large-scale machine learning tasks, helping to build complex machine learning models from big data.
- Analysis and Reporting: With Spark SQL and analytics features, Spark is used to generate reports and analyze data from various data sources.

4.3.2.2 Apache Hadoop.

Apache Hadoop is an open source framework for storing and processing big data distributed across a cluster of computers. Hadoop was developed to help organizations process very large volumes of data without performance and cost issues. Hadoop provides a distributed storage system and the ability to process very large data sets in parallel, making it a very powerful tool in big data systems. (Anderson, 2016)

Outstanding features:

- Distributed Storage System (HDFS): Hadoop uses HDFS (Hadoop Distributed File System), a distributed file system, to store data across different nodes in a cluster of computers. Data in HDFS is divided into small blocks and backed up to ensure data availability and durability.
- Scalability: Hadoop can scale from a few computers to thousands of computers in a cluster, making it capable of processing data at a scale that traditional systems cannot handle.
- Parallel processing: Hadoop processes data in parallel (MapReduce), which breaks large tasks into smaller pieces, which are then executed in parallel on multiple computers in the cluster. This speeds up processing and reduces latency.
- High fault tolerance: Hadoop is designed with redundancy and disaster recovery in mind, ensuring that if one node in the cluster fails, the data can still be recovered from backups on other nodes.

- Rich ecosystem: Hadoop is not just a data storage and processing tool, but also has a large ecosystem that includes many additional tools such as Apache Hive, Apache HBase, Apache Pig, Apache Spark, and many others, which help extend the capabilities and functionality of the system.

Key Features:

- Hadoop Distributed File System (HDFS): Hadoop's distributed storage system, which helps distribute and store large data sets across multiple computers, with high backup and fault tolerance.
- MapReduce: This is a parallel computing model that Hadoop uses to process and analyze data. MapReduce breaks down jobs into small pieces (Map), then combines their results (Reduce). This process helps process data quickly and efficiently.
- Apache Hive: Hive is a data warehouse system on Hadoop that helps users query data using the SQL language. Hive simplifies querying data on Hadoop by using SQL syntax without having to work directly with MapReduce.
- Apache HBase: HBase is a distributed, unstructured database designed to run on Hadoop. HBase is well suited for applications that require processing large data tables and heterogeneous data.
- Apache Pig: Pig is a programming language that helps process data on Hadoop. Pig provides an easier-to-use syntax than MapReduce and is optimized for complex data processing tasks.
- YARN: YARN (Yet Another Resource Negotiator) is Hadoop's resource management system, which helps manage and allocate resources in a Hadoop cluster, coordinate tasks, and optimize resource usage.

Advantages:

- Large-scale data processing: Hadoop can process and analyze huge data sets that traditional systems cannot handle. This helps businesses and organizations take advantage of all their data to make more accurate business decisions.
- Scalability: Hadoop can easily scale from a single computer to thousands of machines in a cluster, allowing for large-scale data processing without performance issues.
- Low Cost: Hadoop uses commodity hardware, which reduces costs compared to expensive and specialized hardware systems. The use of low-cost computers reduces deployment and maintenance costs.
- Flexibility: Hadoop can handle many different types of data, including structured, unstructured, and semi-structured data. This makes Hadoop suitable for many different applications and industries.

- Fault Tolerance: With its backup and recovery features, Hadoop is very fault tolerant. If a node in the cluster fails, Hadoop can still recover data from backups and continue processing without interruption.

Disadvantages:

- Difficult to use for beginners: Hadoop requires solid technical knowledge to deploy and use, especially when working with additional tools such as MapReduce, Hive, or HBase. This can be difficult for beginners.
- High Latency: Although Hadoop can handle large data sets, since it primarily uses batch processing, the latency when retrieving and processing data can be quite high, making it unsuitable for real-time tasks.
- Complex Management and Maintenance: When Hadoop is deployed on a large cluster, the management and maintenance of the system can become complex, requiring constant monitoring and optimization to ensure the best performance.

Applications of Apache Hadoop:

- Big Data Analytics: Hadoop is ideal for analyzing huge data sets, including applications such as customer data analytics, user behavior analytics, and social media analytics.
- Unstructured Data Storage and Processing: Hadoop is well suited for processing unstructured data types such as text, images, videos, and other data that does not follow a specific pattern.
- Machine Learning and Artificial Intelligence: Hadoop can be used to build machine learning models, especially when working with large datasets, helping to create predictive models, analytics, and pattern recognition.
- Real-Time Data: Although Hadoop primarily processes data in batches, with the addition of tools such as Apache Storm or Apache Spark, it can process data in real-time, helping organizations analyze data immediately.

4.3.3 Data Preprocessing and Cleaning.

4.3.3.1 Pandas (Python).

Pandas is a powerful library in Python for data processing and analysis. It provides data structures such as DataFrame and Series, which help to manipulate structured data (such as table data, CSV data, Excel data, SQL database) easily and efficiently. Pandas is a popular tool in the field of data analysis and data science because of its ease of use and fast processing ability. Pandas provides very powerful tools for cleaning and preprocessing data, making the process of data analysis and model building more accurate and efficient.

In the data analysis process, one of the important steps is data preprocessing and cleaning. (George McIntire, Brendan Martin, Lauren Washington, n.d.) Here are the techniques for cleaning data with Pandas:

- Check and handle missing data: Handle by removing rows with missing data or filling in appropriate values.
- Remove duplicate data: Use the drop_duplicates() method to remove duplicates.
- Data Type Conversion: Use astype() or pd.to_datetime() to convert data types accordingly.
- Outlier Handling: Handle by identifying and removing outliers that exceed a certain range.
- Normalize and Standardize: Ensure that numeric columns have the same units of measure and range.
- Create New Features: Based on existing features, you can create new features to improve forecasting or analysis.

4.3.3.2 Apache Spark SQL

Apache Spark SQL is a component of the Apache Spark ecosystem, providing big data processing capabilities through traditional SQL queries, combined with Spark APIs. Spark SQL helps users process and analyze data from a variety of data sources (databases, distributed file systems such as HDFS, S3, or even NoSQL systems) using easy-to-understand and friendly SQL statements. (Sarkar, 2019)

Spark SQL provides a powerful and flexible approach to interacting with big data, allowing users to use SQL to query data without having to transform the data or change the data structure, while still taking advantage of Spark's capabilities such as distribution and high-speed processing.

Apache Spark SQL provides powerful tools for processing and cleaning big data, from removing missing values, handling duplicate data, filtering invalid data, to normalizing and creating new columns.

With its distributed processing capabilities and tight integration with the Spark ecosystem, Spark SQL is ideal for cleaning large-scale data.

The data cleaning process includes normalization, error removal, and creation of new computations to make the data ready for analysis.

4.4 Discuss the importance of using scalable tools for big data.

Scalability is an important factor when working with big data, especially when dealing with millions or billions of rows of data. As data grows exponentially, the ability of a system or tool to scale efficiently becomes a key factor in maintaining performance, speed, and reliability.

4.4.1 Handling Large Volumes: Big data requires scalable systems.

For organizations working with billions of rows of data, scalability ensures that systems can handle increasing volumes of data without experiencing significant performance degradation.

Without scalable tools, systems can struggle to manage large datasets, resulting in slow processing times, system crashes, or inaccurate results. Scalable tools like Apache Spark or Hadoop are designed to distribute data across multiple nodes and compute resources, making them efficient at processing data as it grows larger.

For example,

- Hadoop uses a distributed file system (HDFS), which distributes data across multiple computers, allowing for parallel storage and processing of large data.
- Apache Spark can scale horizontally by adding more nodes to the cluster, allowing for rapid processing and storage of large data.

4.4.2 Speed and efficiency: Process data faster with scalability.

As data sets grow larger, traditional tools can become inefficient, taking hours or even days to perform simple tasks. Scale-out systems like Apache Spark are optimized for parallel processing, meaning tasks are split across multiple processors or nodes in a cluster, dramatically reducing processing times.

With the ability to scale horizontally (adding more computers to the system) or vertically (increasing the capacity of a single computer), scale-out systems can maintain high performance even as data volumes increase. This is important for real-time analytics, machine learning model training, and business intelligence analysis.

4.4.3 Flexibility: Adapting to Different Data Volumes and Types.

Scalable tools can not only handle large volumes of data but can also adapt to different types of data (structured, semi-structured, unstructured) and changing workloads. This adaptability ensures that businesses can handle diverse data sets such as customer behavior data, social media posts, sensor data, and many other types of data.

4.4.4 Real-Time Analytics: Supporting Fast Decision-Making.

For many businesses, the ability to process real-time data is critical. Scalable tools like Apache Kafka for streaming data and Apache Spark for real-time data processing allow businesses to analyze and make decisions based on the latest information.

With scalable tools, businesses can ingest, process, and analyze large volumes of data in real time, helping them make faster decisions for marketing, customer service, and operations.

4.4.5 A Sustainable Future: Preparing for Growing Data.

The growth of data is inevitable. As organizations continue to collect more data, their infrastructure must change to keep up. Scalable tools allow businesses to prepare for data growth. Whether it’s adding more servers or increasing storage capacity, these tools ensure that the system can continue to scale as data increases.

For example, as IoT devices grow and generate massive amounts of data, scalable systems ensure that data can be received and processed without having to change the entire system.

4.5 The Importance of Real-Time Analytics for Rapid Decision Making.

Real-time analytics is becoming increasingly important for businesses and organizations, as it allows them to make faster and more accurate decisions based on constantly updated data. This is especially important in industries such as e-commerce, finance, healthcare, and network monitoring, where data is constantly changing and timely decision making can make a big difference. In a context where data is growing rapidly and changing in nature, the use of real-time data processing and analytics tools is essential to achieve high efficiency in business operations.

- E-commerce: Real-time customer behavior analysis helps to adjust marketing or pricing strategies immediately.
- Network and security monitoring: Detect and respond immediately to security threats or network incidents.
- Finance: Detect fraud in transactions or adjust investment strategies immediately based on market data.

Two prominent tools in this area are Apache Kafka and Apache Flink, both of which play an important role in processing and visualizing real-time data, helping businesses react promptly and make decisions based on immediately updated information.

4.5.1 Apache Kafka.

Apache Kafka is an extremely popular streaming data processing and distribution platform that helps organizations collect, process, and distribute data in real-time. Kafka is designed to handle large volumes of data and supports high scalability, can process millions of events per second, thereby providing continuous streams of data for further analysis or processing applications. (Gupta, 2024)

Benefits of Apache Kafka in real-time analytics:

- Streaming data processing: Kafka can collect and distribute streaming data from various sources (e.g. IoT sensors, social networks, online transactions, etc.) and provide it to real-time analytics systems.
- Ensuring availability and fault tolerance: Kafka supports fault tolerance, which helps ensure that data is not lost in the event of a failure.
- Powerful scalability: Kafka can scale elastically to meet the demands of processing big data and real-time streaming data.
- Integration with analytics tools: Kafka easily integrates with other analytics and data processing tools such as Apache Spark, Apache Flink, and database systems to perform rapid analysis and decision making.

Application example: An e-commerce company can use Kafka to collect data on customer shopping behavior and product inventory. This data can be analyzed in real-time to make decisions such as changing product prices or offering promotions immediately.

4.5.2 Apache Flink

Apache Flink is a real-time data processing and streaming analytics engine that excels at real-time data processing and complex computation. Flink is optimized for low-latency streaming data processing, allowing businesses to perform analytics and make decisions almost instantly as new data arrives. (Team, n.d.)

Benefits of Apache Flink in real-time analytics:

- Streaming and batch data processing: Flink supports both real-time and batch data analytics, enabling users to perform complex tasks such as aggregate calculations, distributed analytics, etc.
- Real-time data visualization: Flink helps generate reports, charts, and dashboards in real time so that users can monitor and make timely decisions.
- Accuracy and low latency: Flink can process data with very low latency and ensure high accuracy, which is important in situations that require quick response such as financial transactions or network monitoring.
- Integration with Kafka: Flink is often used in conjunction with Kafka to process streaming data and perform real-time analytics, creating a powerful system for processing and visualizing data in real time.

Application example: In the financial industry, Flink can be used to analyze online transaction data and detect fraud in real time. When a transaction shows signs of abnormality, the system can immediately alert and take preventive measures.

4.5.3 Combining Apache Kafka and Apache Flink in real-time analytics

The combination of Apache Kafka and Apache Flink provides a powerful solution for processing and analyzing streaming data in real time. Kafka can collect and distribute data, while Flink can ingest this data and perform complex analytics, such as aggregation, prediction, or event detection in real time.

Kafka and Flink workflow:

- Collecting data with Kafka: Kafka collects data from various sources, such as IoT sensors, mobile applications, or transaction systems, and distributes this data to consumers such as Flink.
- Analyzing data with Flink: Flink ingests data from Kafka and performs real-time analytics or computations. For example, Flink can perform operations such as averaging, trend analysis, or predicting future events.
- Visualize results: After analyzing data, Flink can output results to reporting systems or real-time dashboards, allowing users to track key metrics and make quick decisions.

4.6 Integration with Cloud Services.

Using tools like AWS QuickSight, Google Data Studio, and Azure Power BI helps organizations maximize the potential of their data stored in the cloud. These tools provide direct connectivity to cloud storage services, making it easy for users to access, analyze, and visualize data without having to move it out of the cloud. This not only reduces storage and bandwidth costs, but also saves time and increases data security.

4.6.1 AWS QuickSight: Powerful Cloud Data Analytics from Amazon

AWS QuickSight is a cloud data analytics and visualization service from Amazon Web Services (AWS), which makes it easy for users to explore and analyze data from various data sources, including data stored in AWS services such as Amazon S3, Amazon Redshift, and Amazon RDS.

Benefits of AWS QuickSight:

- Deep integration with AWS services: QuickSight can seamlessly integrate with other AWS services such as S3, Redshift, and Athena, making it easy for users to retrieve and analyze data from cloud storages.
- Powerful data visualization: This tool provides a variety of charts and dashboards to visualize data, helping users better understand trends and patterns in data.
- Automated Machine Learning (ML): QuickSight is capable of using automated machine learning models to analyze and make predictions, supporting decisions in situations that require complex data analysis.
- High scalability and performance: QuickSight can handle large amounts of data and deliver results in real time, suitable for organizations that need to analyze large-scale data.

Application example: An e-commerce company can use QuickSight to analyze customer data and shopping behavior from Amazon Redshift, then visualize purchasing trends to optimize marketing strategies.

4.6.2 Google Data Studio: Free tool for cloud data visualization

Google Data Studio is a free tool from Google that helps create reports and dashboards that visualize data from a variety of sources, including Google cloud services such as Google Analytics, Google BigQuery, and Google Sheets. (Egg, n.d.)

Benefits of Google Data Studio:

- Free and easy to use: Google Data Studio is a completely free tool with an easy-to-use interface, making it easy for users to create reports and data dashboards without complex programming skills.
- Integration with Google services: Data Studio can easily connect to data storage services on Google Cloud, such as BigQuery and Google Sheets, to retrieve and analyze data.
- Create custom reports and dashboards: Data Studio allows users to create custom reports and dashboards with a variety of charts, graphs, and tables, making it easy for users to track important metrics.
- Share and collaborate: This tool supports sharing reports and dashboards with others in the organization, enhancing collaboration and shared decision-making.

Application example: A business can use Google Data Studio to visualize marketing data from Google Analytics and Google Sheets, thereby evaluating the effectiveness of advertising campaigns and optimizing marketing strategies.

4.6.3 Azure Power BI: Microsoft's powerful tool for cloud data analysis and visualization

Azure Power BI is a powerful tool from Microsoft that helps analyze and visualize data from a variety of sources, including data stored on the Azure platform such as Azure SQL Database and Azure Data Lake, as well as from external services such as Salesforce and Google Analytics. (davidiseminger, mohitp930, TimSvhererWithAquent, KesemSharabi, n.d.)

Benefits of Azure Power BI:

- Deep integration with Azure services: Power BI can easily connect to data storage services on Azure such as Azure SQL Database and Azure Data Lake, allowing users to directly access and analyze data on Microsoft's cloud.

- Big Data and Complex Analytics Capabilities: Power BI can handle large data sets and perform complex analytics, providing insights into data.
- Create interactive reports and dashboards: Power BI supports the creation of interactive reports and dashboards, allowing users to track metrics and analyze data flexibly.
- Integrate with Microsoft Office 365: Power BI can integrate with Microsoft applications such as Excel and SharePoint, making it easy for users to share and collaborate on reports and dashboards.

Application example: A financial company can use Power BI to analyze transactional and financial data stored on Azure SQL Database, create visual financial reports, and track business performance.

5 Demonstrate the use of data manipulation and automation to present a visualization for a given user case. P4

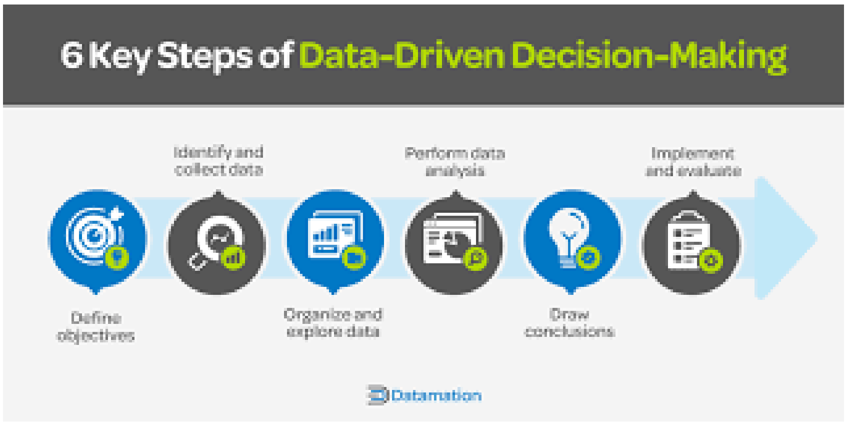

6 Key Steps of Data-Driven Decision-Making
Define objectives
Identify and collect data
Organize and explore data
Perform data analysis
Draw conclusions
Implement and evaluate
Datamation

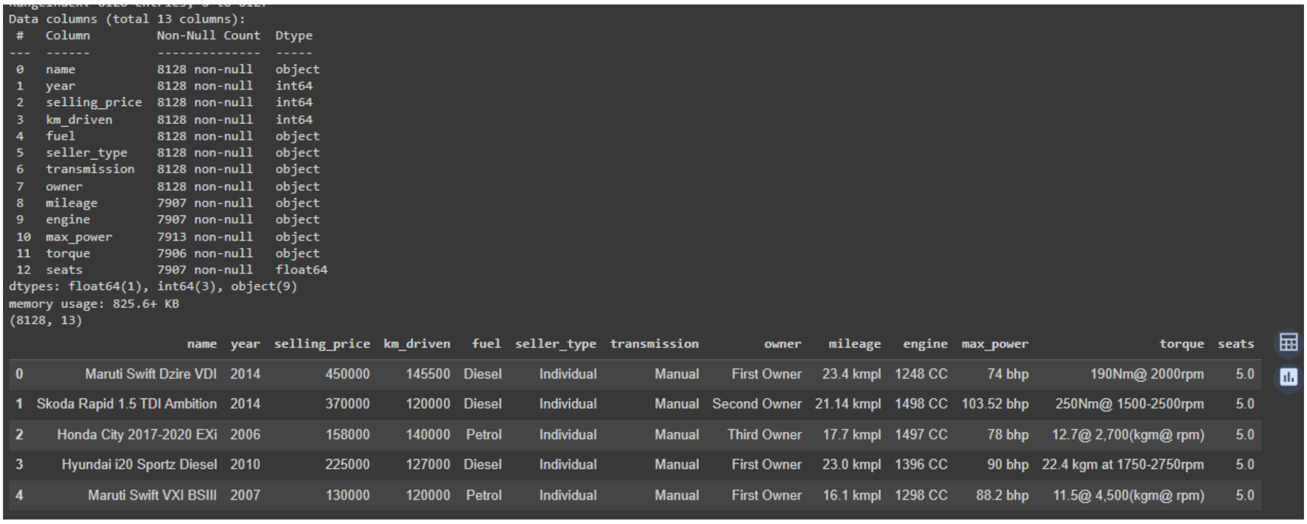
Figure 22 The processes of data-driven decision-making.

5.1 Identify the problem to be solved.

Before deciding to purchase a car for the company's employees to enhance their delivery operations, the company wants to estimate the budget for this task based on data collected from a used car dealership including information about the car's model, year, mileage, condition, and price.

5.2 Data Collection.

Data was collected from the old car shop including information on the model, year, mileage, condition, and price of the vehicles. The provided dataset consists of 13 columns and 8128 rows:



```
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   name           8128 non-null   object 
 1   year           8128 non-null   int64  
 2   selling_price  8128 non-null   int64  
 3   km_driven      8128 non-null   int64  
 4   fuel           8128 non-null   object 
 5   seller_type    8128 non-null   object 
 6   transmission   8128 non-null   object 
 7   owner          8128 non-null   object 
 8   mileage        7907 non-null   object 
 9   engine         7907 non-null   object 
 10  max_power      7913 non-null   object 
 11  torque         7906 non-null   object 
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
(8128, 13)
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74 bhp	190Nm@ 2000rpm	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52 bhp	250Nm@ 1500-2500rpm	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78 bhp	12.7@ 2,700(kgm@ rpm)	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90 bhp	22.4 kgm at 1750-2750rpm	5.0
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2 bhp	11.5@ 4,500(kgm@ rpm)	5.0

Figure 23 Dataset information.

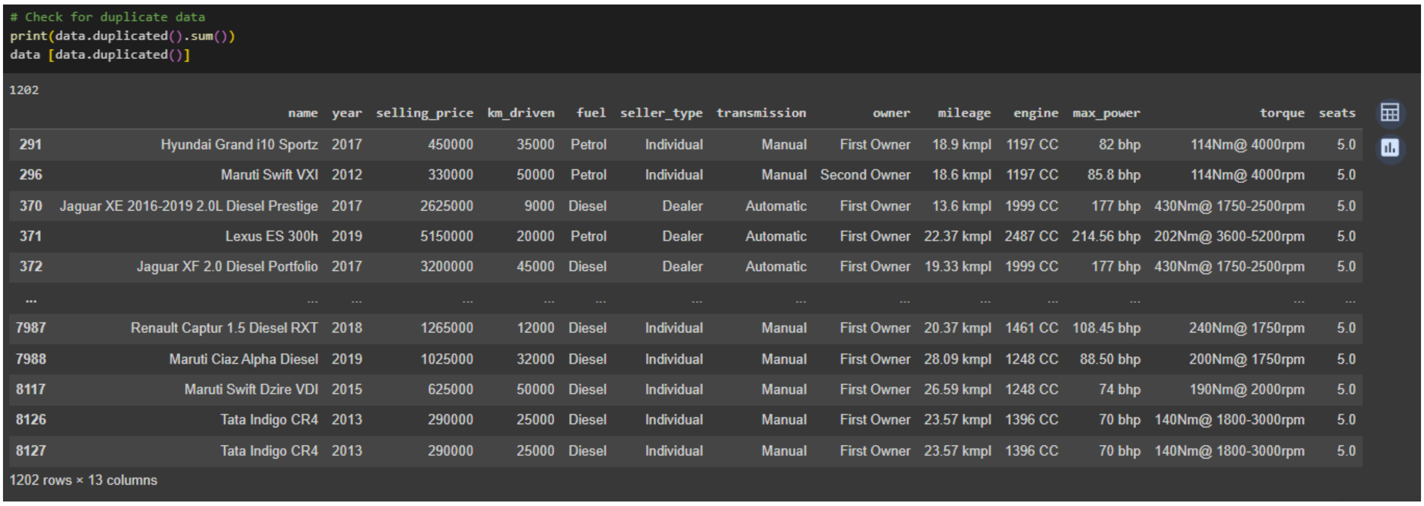
- name: The name or model of the car. For example, "Maruti Swift Dzire VDI" is the name of the car model.
- year: The year the car was manufactured. This information helps determine the age of the car. For example, 2014.
- selling_price: The selling price of the car (in money). This is the current price at which the car is being sold. For example: 450,000.
- km_driven: The number of kilometers the car has traveled. This is an important factor in assessing the level of wear and tear on the car. For example: 145,500 km.
- fuel: The type of fuel the car uses, such as Diesel, Petrol (gasoline), CNG (compressed natural gas), or Electric (electricity). For example: Diesel.
- seller_type: The type of person selling the car, such as an individual (Individual), a dealer (Dealer), or a company (Trustmark Dealer). For example: Individual.
- transmission: The vehicle's transmission, which can be automatic or manual. For example: Manual.
- owner: The number of previous owners of the vehicle, for example: "First Owner", "Second Owner".
- mileage: The vehicle's fuel consumption, usually measured in km/l (the number of kilometers the vehicle travels with 1 liter of fuel). For example 23.4 kmpl.
- engine: The vehicle's engine capacity, measured in cc (cubic centimeters). For example: 1248 cc.
- max_power: The maximum power the engine can achieve, measured in horsepower (bhp). For example: 74 bhp.
- torque: The engine's torque, measured in Newton meters (Nm) at a certain number of engine revolutions (rpm).
- seats: The number of seats in the vehicle. For example: 5.

The above information will give everyone an overview of the car, from condition, fuel type, transmission type, performance to engine specifications and number of seats.

5.3 Clean and standardize data.

5.3.1 Handling Duplicate Data.

First, I will check if the data is duplicate or not, using pandas I show the count and show duplicate rows.



```
# Check for duplicate data
print(data.duplicated().sum())
data [data.duplicated()]
```

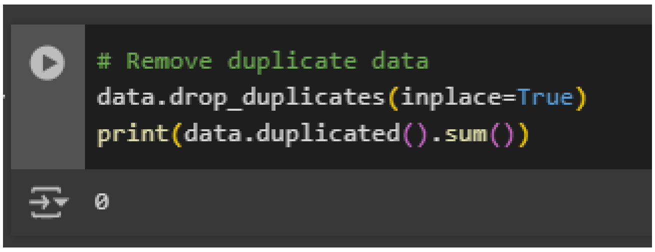
```
1202
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
291	Hyundai Grand i10 Sportz	2017	450000	35000	Petrol	Individual	Manual	First Owner	18.9 kmpl	1197 CC	82 bhp	114Nm@ 4000rpm	5.0
296	Maruti Swift VXI	2012	330000	50000	Petrol	Individual	Manual	Second Owner	18.6 kmpl	1197 CC	85.8 bhp	114Nm@ 4000rpm	5.0
370	Jaguar XE 2016-2019 2.0L Diesel Prestige	2017	2625000	9000	Diesel	Dealer	Automatic	First Owner	13.6 kmpl	1999 CC	177 bhp	430Nm@ 1750-2500rpm	5.0
371	Lexus ES 300h	2019	5150000	20000	Petrol	Dealer	Automatic	First Owner	22.37 kmpl	2487 CC	214.56 bhp	202Nm@ 3600-5200rpm	5.0
372	Jaguar XF 2.0 Diesel Portfolio	2017	3200000	45000	Diesel	Dealer	Automatic	First Owner	19.33 kmpl	1999 CC	177 bhp	430Nm@ 1750-2500rpm	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7987	Renault Captur 1.5 Diesel RXT	2018	1265000	12000	Diesel	Individual	Manual	First Owner	20.37 kmpl	1461 CC	108.45 bhp	240Nm@ 1750rpm	5.0
7988	Maruti Ciaz Alpha Diesel	2019	1025000	32000	Diesel	Individual	Manual	First Owner	28.09 kmpl	1248 CC	88.50 bhp	200Nm@ 1750rpm	5.0
8117	Maruti Swift Dzire VDI	2015	625000	50000	Diesel	Individual	Manual	First Owner	26.59 kmpl	1248 CC	74 bhp	190Nm@ 2000rpm	5.0
8126	Tata Indigo CR4	2013	290000	25000	Diesel	Individual	Manual	First Owner	23.57 kmpl	1396 CC	70 bhp	140Nm@ 1800-3000rpm	5.0
8127	Tata Indigo CR4	2013	290000	25000	Diesel	Individual	Manual	First Owner	23.57 kmpl	1396 CC	70 bhp	140Nm@ 1800-3000rpm	5.0

1202 rows × 13 columns

Figure 24 The count and show duplicate rows.

I found that there were quite a few rows of duplicate data (1202 rows). So, I'm going to proceed to remove these rows from the data frame.



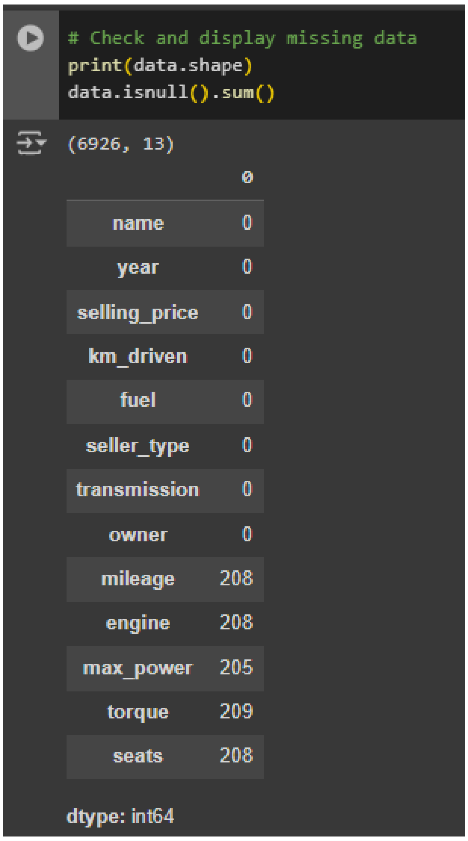
```
# Remove duplicate data
data.drop_duplicates(inplace=True)
print(data.duplicated().sum())
```

```
0
```

Figure 25 Remove duplicate data.

5.3.2 Handling Missing Values.

Check and count the number of missing values (NA/null) in each column of the DataFrame.



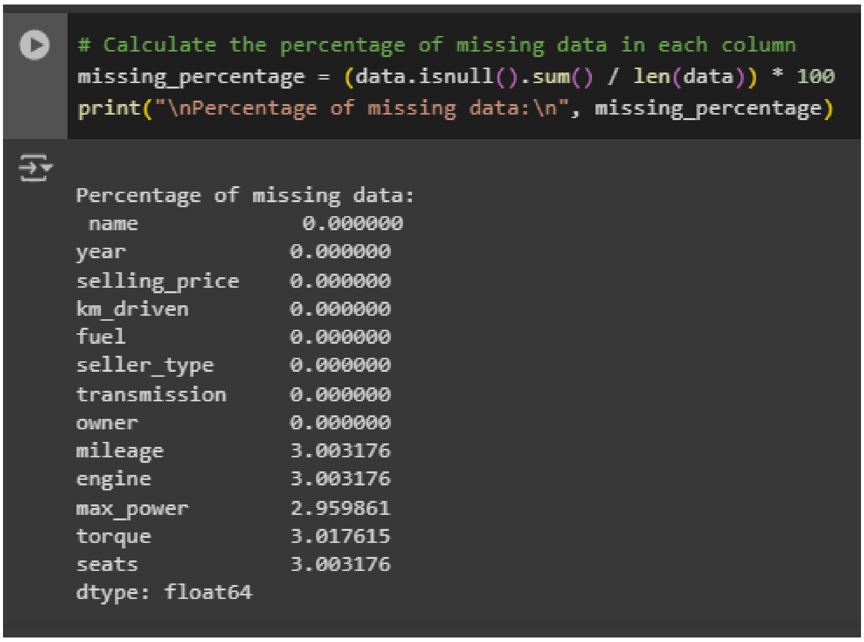
```python
# Check and display missing data
print(data.shape)
data.isnull().sum()
```

```
(6926, 13)
```

	0
name	0
year	0
selling_price	0
km_driven	0
fuel	0
seller_type	0
transmission	0
owner	0
mileage	208
engine	208
max_power	205
torque	209
seats	208

```
dtype: int64
```

Figure 26 Check and count the number of missing values.

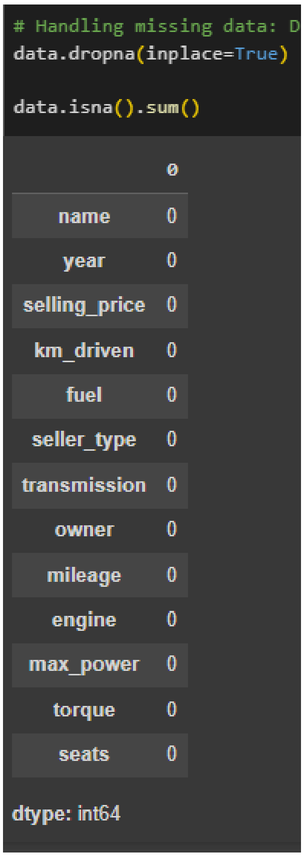


```python
# Calculate the percentage of missing data in each column
missing_percentage = (data.isnull().sum() / len(data)) * 100
print("\nPercentage of missing data:\n", missing_percentage)
```

```

Percentage of missing data:
 name            0.000000
year            0.000000
selling_price   0.000000
km_driven       0.000000
fuel            0.000000
seller_type     0.000000
transmission    0.000000
owner           0.000000
mileage         3.003176
engine          3.003176
max_power       2.959861
torque          3.017615
seats           3.003176
dtype: float64
```

Figure 27 Percentage of missing data.

Seeing that there were about 200 missing data out of a total of 6926 values, which is about 3 percent, it did not affect the analysis results, so I decided to proceed with deleting the rows containing these missing values.

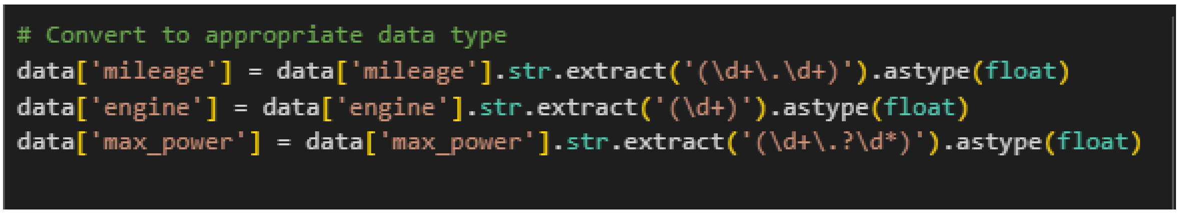


```
# Handling missing data: D
data.dropna(inplace=True)

data.isna().sum()
```

```
                0
name            0
year            0
selling_price   0
km_driven       0
fuel            0
seller_type     0
transmission    0
owner           0
mileage         0
engine          0
max_power       0
torque          0
seats           0

dtype: int64
```

Figure 28 Delete rows if missing data.

5.3.3 Convert to appropriate data type.



```
# Convert to appropriate data type
data['mileage'] = data['mileage'].str.extract('(\d+\.\d+)').astype(float)
data['engine'] = data['engine'].str.extract('(\d+)').astype(float)
data['max_power'] = data['max_power'].str.extract('(\d+\.?\d*)').astype(float)
```

Figure 29 Convert to appropriate data type.

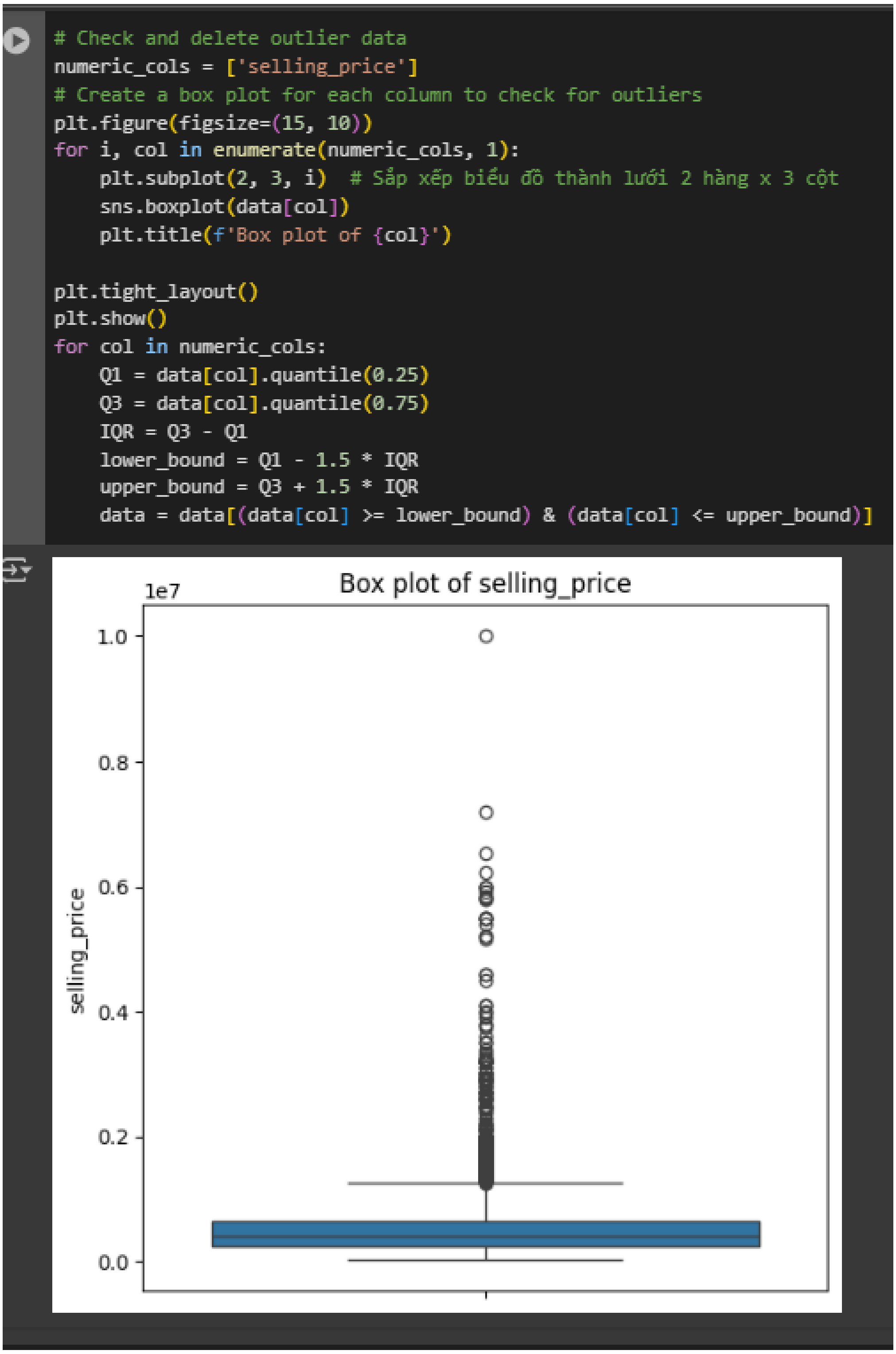
Next, we normalize the values in the mileage, engine, and max_power columns to convert them to numeric data types (float). By:

- Extracting the numeric parts from the string (removing unnecessary units and characters).
- Converting the data to float for easy analysis or calculation.

After doing this, the data in these three columns will be consistent in format. This makes it easy to analyze and calculate with the numerical values.

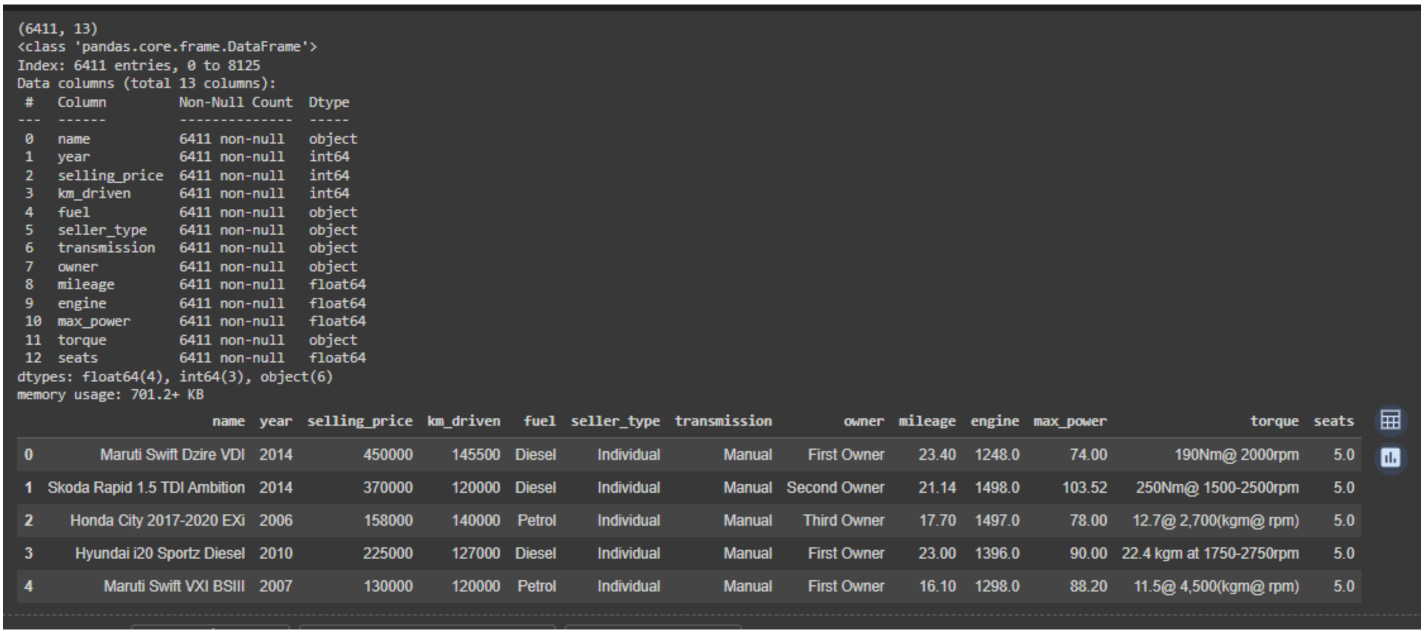
5.3.4 Check and delete outlier data.

Finally, we check and remove outlier data in the selling_price, km_driven, and mileage columns using the Interquartile Range (IQR) Method.



```
# Check and delete outlier data
numeric_cols = ['selling_price']
# Create a box plot for each column to check for outliers
plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(2, 3, i)  # Sắp xếp biểu đồ thành lưới 2 hàng x 3 cột
    sns.boxplot(data[col])
    plt.title(f'Box plot of {col}')

plt.tight_layout()
plt.show()
for col in numeric_cols:
    Q1 = data[col].quantile(0.25)
    Q3 = data[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    data = data[(data[col] >= lower_bound) & (data[col] <= upper_bound)]
```

Figure 30 Check and delete outlier data.

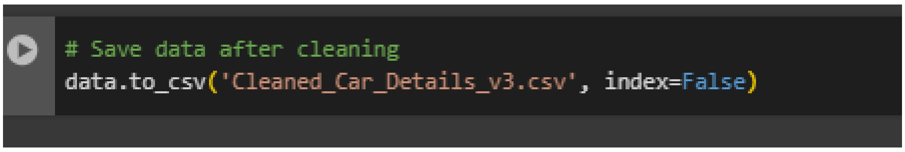


```
(6411, 13)
<class 'pandas.core.frame.DataFrame'>
Index: 6411 entries, 0 to 8125
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   name           6411 non-null   object 
 1   year           6411 non-null   int64  
 2   selling_price  6411 non-null   int64  
 3   km_driven      6411 non-null   int64  
 4   fuel           6411 non-null   object 
 5   seller_type    6411 non-null   object 
 6   transmission   6411 non-null   object 
 7   owner          6411 non-null   object 
 8   mileage        6411 non-null   float64
 9   engine         6411 non-null   float64
 10  max_power      6411 non-null   float64
 11  torque         6411 non-null   object 
 12  seats          6411 non-null   float64
dtypes: float64(4), int64(3), object(6)
memory usage: 701.2+ KB
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	torque	seats
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.40	1248.0	74.00	190Nm@ 2000rpm	5.0
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14	1498.0	103.52	250Nm@ 1500-2500rpm	5.0
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.70	1497.0	78.00	12.7@ 2,700(kgm@ rpm)	5.0
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.00	1396.0	90.00	22.4 kgm at 1750-2750rpm	5.0
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.10	1298.0	88.20	11.5@ 4,500(kgm@ rpm)	5.0

Figure 31 Cleaned data.

Cleaned data has 13 columns and 6411 rows left.



```
# Save data after cleaning
data.to_csv('Cleaned_Car_Details_v3.csv', index=False)
```

Figure 32 Save data after cleaning.

The cleaned data will be saved as "Cleaned_Car_Details_v3" in CSV format. We will use this data to visualize and analyze it using Tableau to ensure improved accuracy and clarity of charts, helping to detect trends and relationships more effectively. Clean data increases persuasiveness saves time and effort for analysts, and supports more accurate decision making. In addition, visualizing clean data improves the ability to communicate information to stakeholders and helps to detect problems early, thereby improving the ability to plan and optimize business processes.

5.4 Explore, visualize, and analyze data.

5.4.1 Explore the distribution of values in data.

First, I will explore the distribution of the data, to see what shape the distribution is.

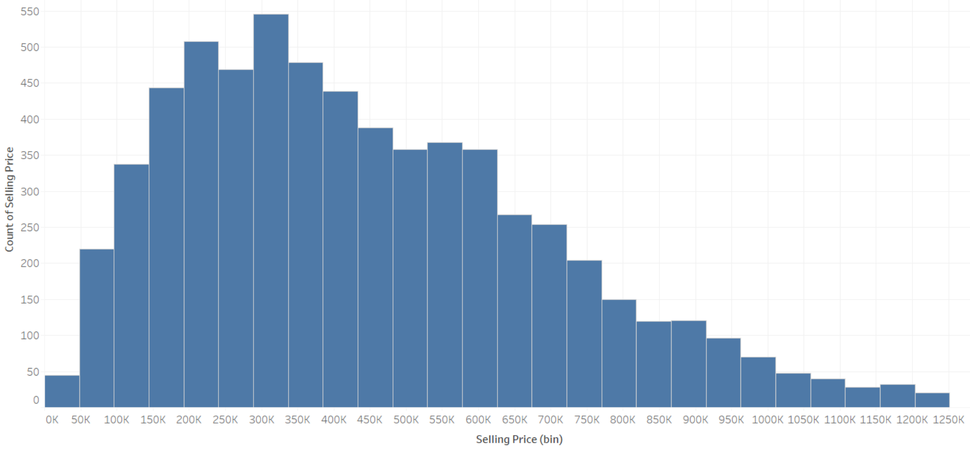


Figure 33 Histogram of Selling Price.

We see the distribution of selling prices:

- Most popular selling price: The price range from 200K to 350K has the highest number of cars sold, with a peak frequency of over 550 cars.
- Average selling price: From 350K to 500K, the number of cars sold gradually decreases but is still quite high, showing that this is the second most popular price range.
- High selling price: From 700K and up, the number of cars sold decreases sharply.

In addition, it shows that the market tends to favor cars with low to medium prices.

The chart shows a gradual decrease from low to high prices, with a clear peak in the low-price range.

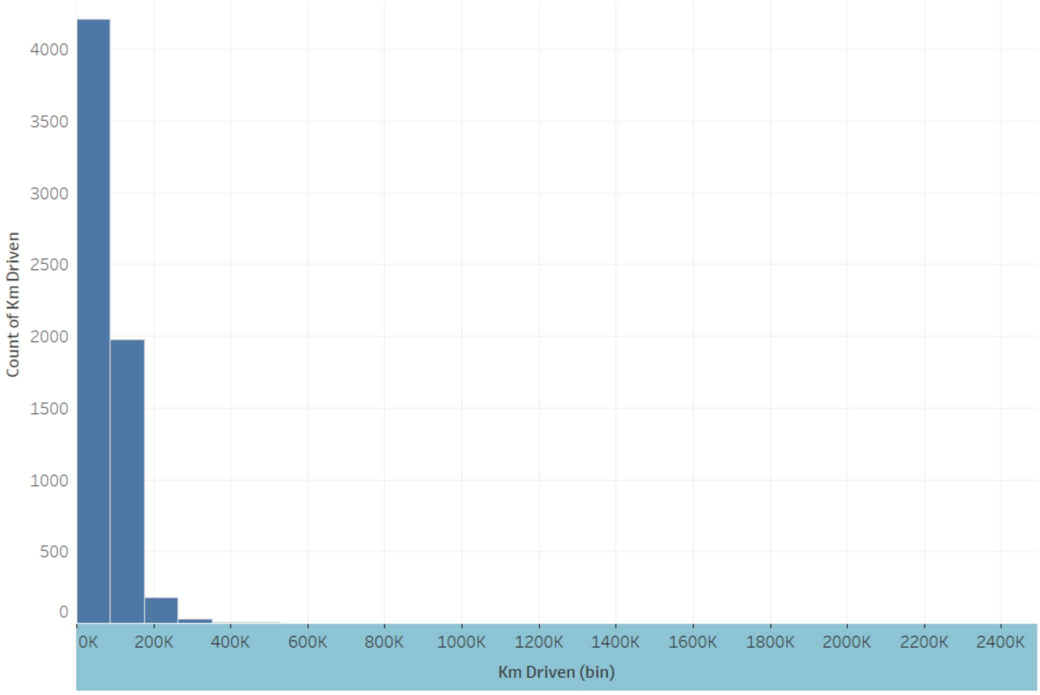


Figure 34 Histogram of Km Driven.

Mileage Distribution: The majority of cars resold have 200k km or less, with the number of cars in this range exceeding 4000. Cars with lower mileage are more likely to be purchased.

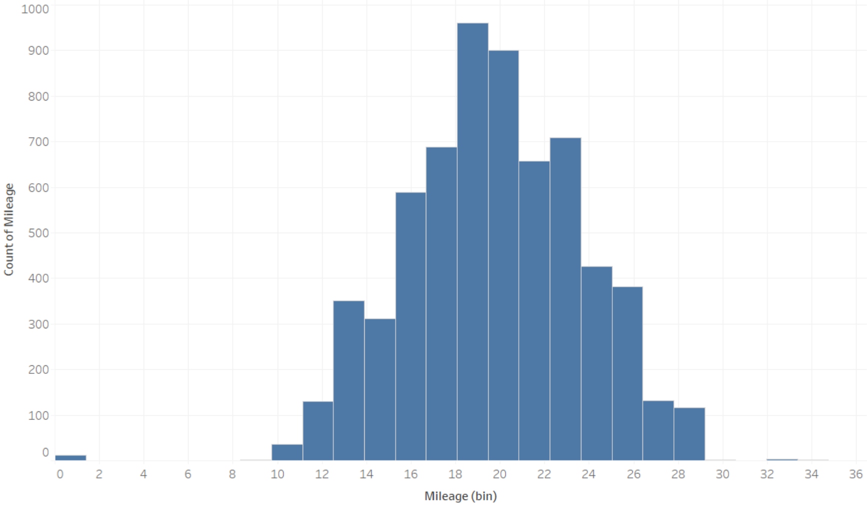
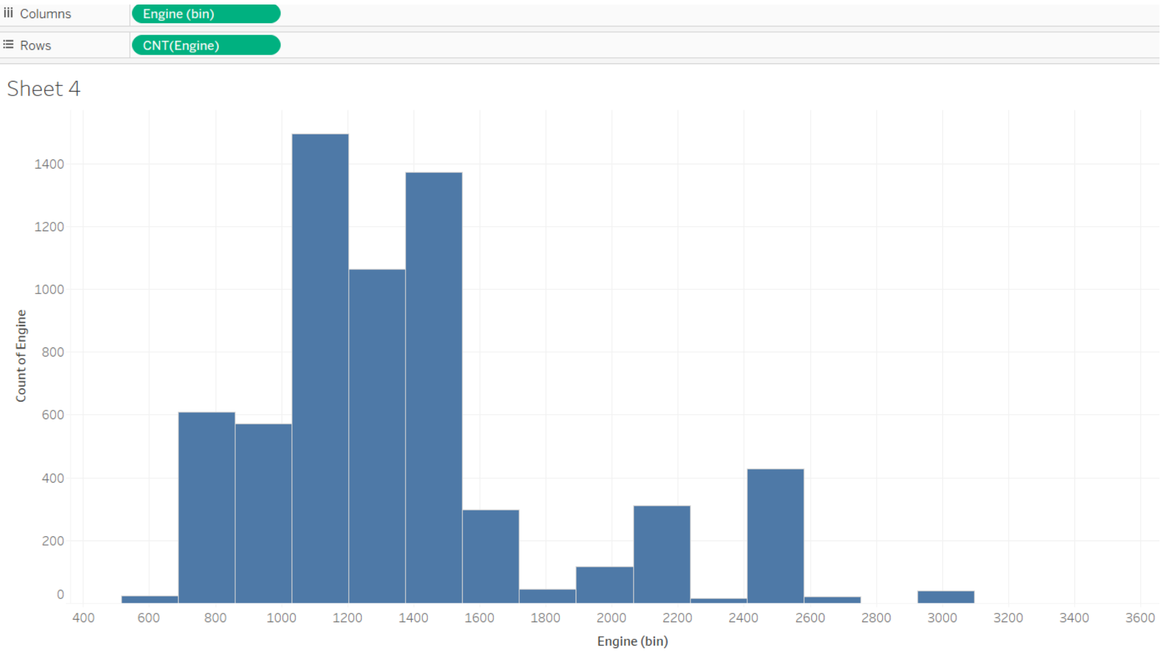


Figure 35 Histogram of Mileage.

The graph has a bell curve, indicating that most cars have average fuel efficiency.

The peak of the graph is in the 17-20 km/l range, indicating that this is the most popular fuel efficiency range. This indicates that this is the fuel efficiency level that many used car buyers prefer.

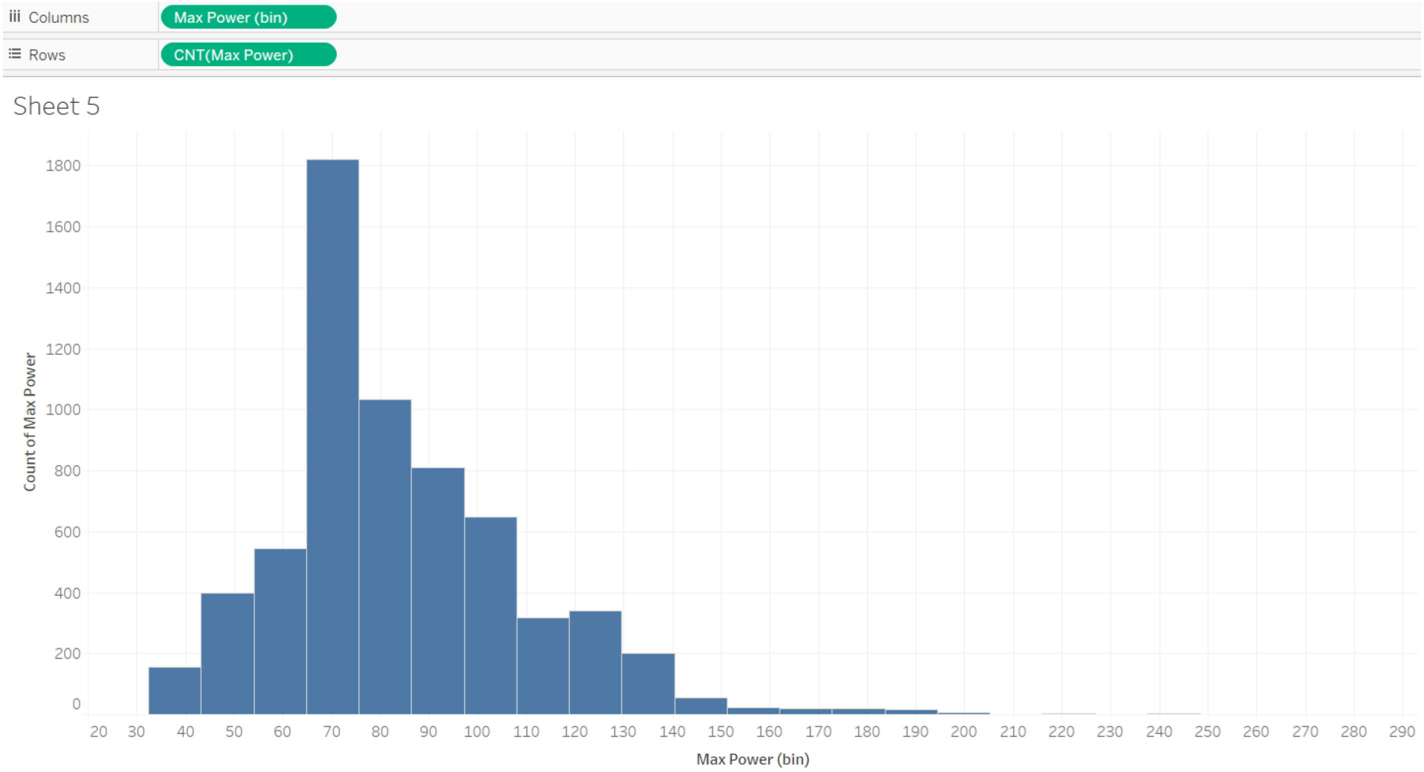
The number of cars decreases as fuel efficiency falls below or above this range.


Columns: Engine (bin)

Rows: CNT(Engine)

Figure 36 Histogram of Engine.

The graph is bell-shaped, with the highest peak in the 800-1400cc range, indicating that this is the most popular engine size. It is the engine size that many used car buyers prefer. The number of cars decreases as the engine size increases, with very few cars having very high engine sizes (over 2000cc).



Columns: Max Power (bin)

Rows: CNT(Max Power)

Figure 37 Histogram of Max Power.

The majority of used cars for resale have between 70 and 100 bhp. This suggests that this is the power level that many used car buyers prefer, perhaps due to the balance between performance and fuel economy. Medium-powered cars (70-100 bhp) are suitable for both city and highway driving, offering a balance between performance and fuel economy. Meanwhile, high-powered cars (over 100 bhp) are often sports or luxury cars, offering high performance but consuming more fuel.

The scatter in the data suggests that there are some very high- and very low-powered cars, but the number of them is small. This may reflect the diversity of customer demand, but the majority is still concentrated in mid-powered cars.

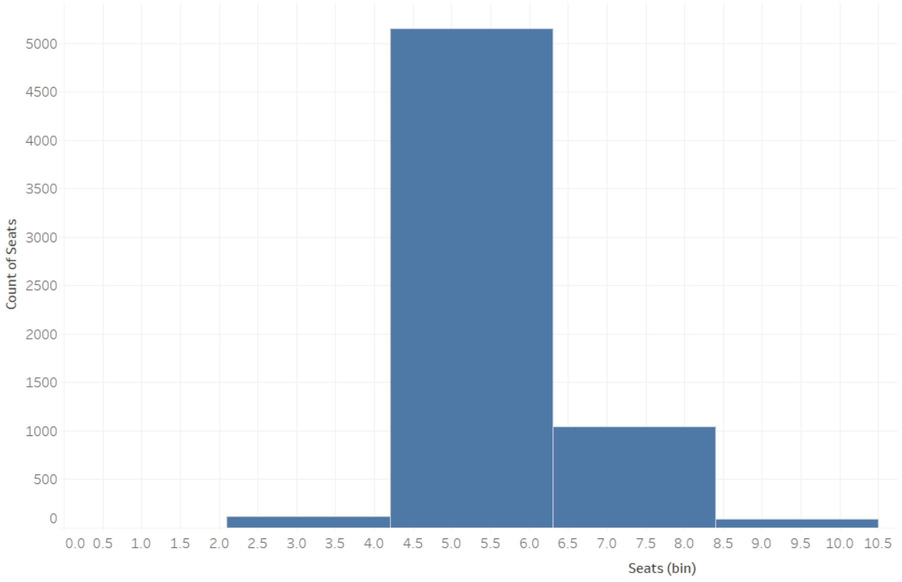
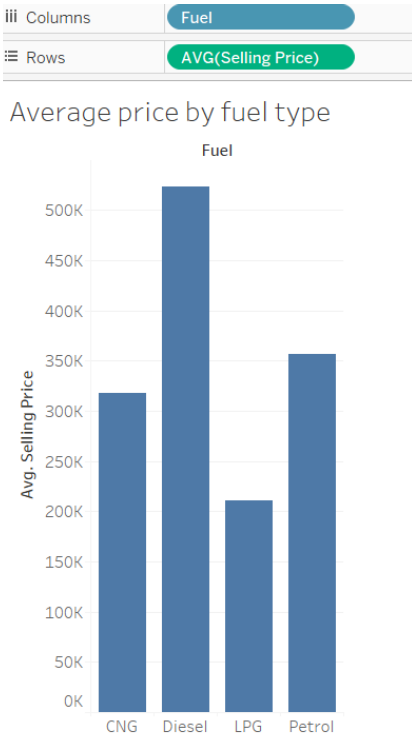


Figure 38 Histogram of Seat

Vehicles with 4 to 5 seats are the most popular, including family cars, sedans, hatchbacks and small SUVs. These vehicles are often favored for their convenience and ability to carry more people. The chart shows a high demand for family cars and regular passenger cars.

5.4.2 Cost Analysis: Compare the average prices of vehicles in different conditions to determine the potential cost savings of purchasing new vehicles.



Columns Fuel

Rows AVG(Selling Price)

Figure 39 Average price by fuel type.

The bar chart in Figure 39 shows the average selling price of used cars by fuel type:

- Diesel has the highest average selling price of all the fuels surveyed. This may be because diesel cars tend to have better fuel efficiency and longer engine life.
- Gasoline cars have a lower average selling price than diesel, but still higher than CNG and LPG. Gasoline cars are popular and easy to maintain, making them affordable for many users.
- CNG (Compressed Natural Gas): CNG cars have a lower average selling price than diesel and gasoline. CNG is an economical and environmentally friendly choice but may be less popular due to limited refueling stations.
- LPG (Liquefied Petroleum Gas): LPG cars have the lowest average selling price. LPG is also an economical and environmentally friendly fuel choice but may be less popular due to limited infrastructure.

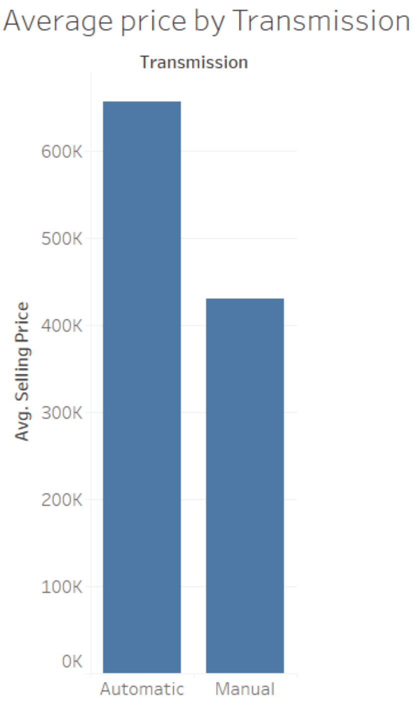


Figure 40 Average price by Transmission.

The average selling price of used cars with automatic transmission is around 600K. Used cars with automatic transmissions are usually more expensive, probably due to the convenience and popularity of this type of transmission.

The average selling price of used cars with manual transmission is around 400K, which is cheaper, probably due to lower production costs.

There is a difference of about 200K between the two types of transmission. The reason for this difference can be due to many factors such as convenience, maintenance costs, and market trends. Based on this chart, we can make a decision. If people prioritize saving costs, cars with manual transmission may be a good choice. On the contrary, if people prioritize convenience and ease of driving, automatic transmission may be a better investment.

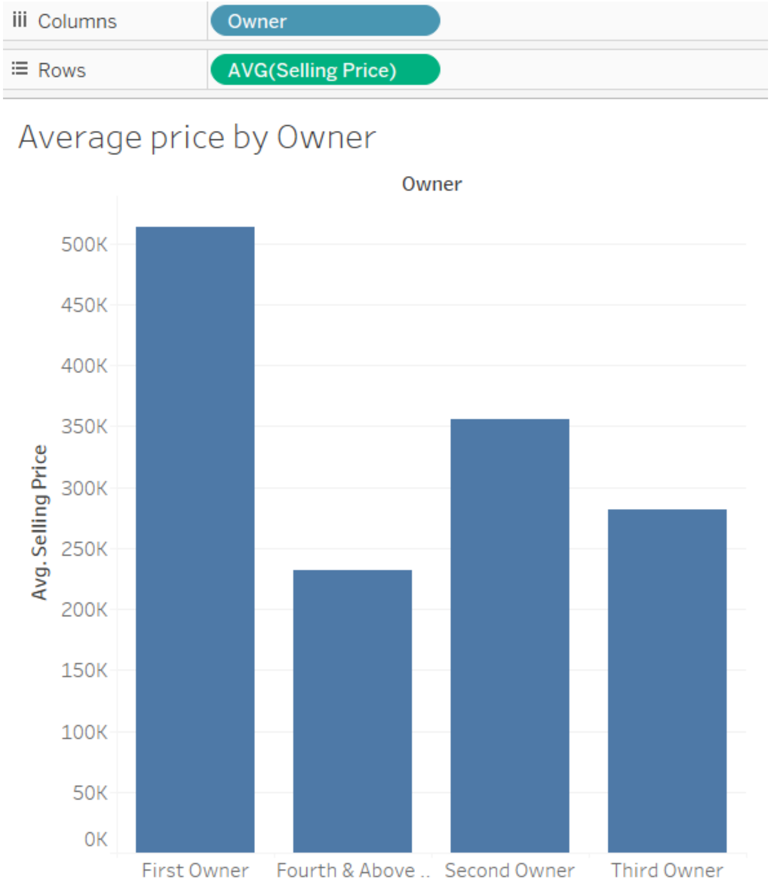


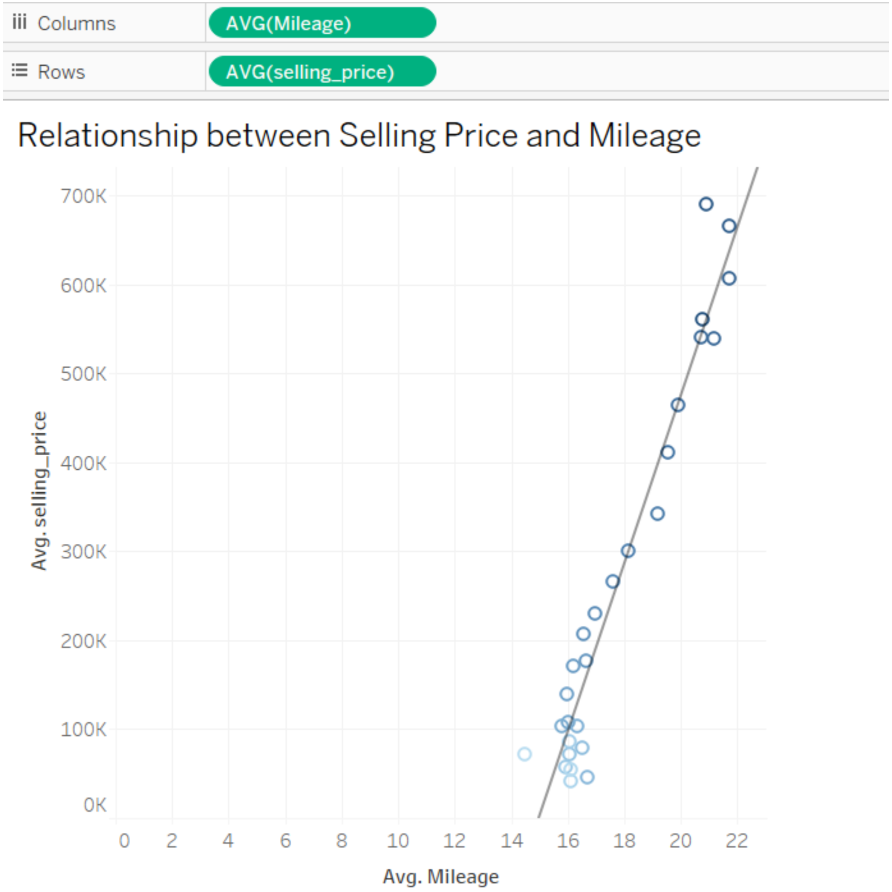
Figure 41 Average price by Owner.

The chart in Figure 41 compares the average selling price of used cars based on the number of previous owners.

Cars with only one previous owner are generally better maintained and have less wear and tear, so they sell for the highest price.

Cars with more previous owners are less expensive, have more mileage, are less used, and require more maintenance.

5.4.3 Mileage Analysis: Examine the relationship between mileage and vehicle prices.



Columns AVG(Mileage)

Rows AVG(selling_price)

Figure 42 Relationship between Selling Price and Mileage.

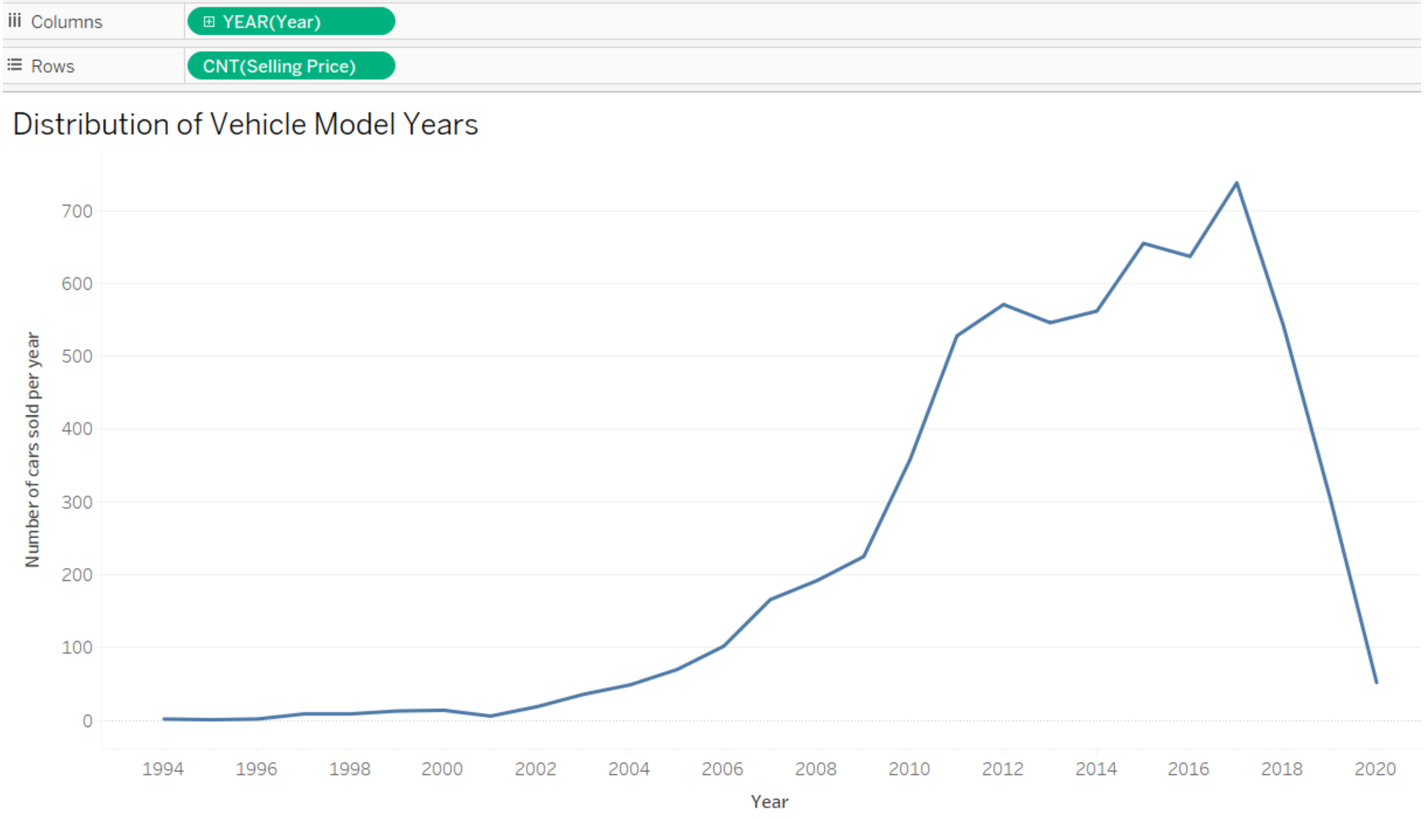
The chart in Figure 42 shows the relationship between the selling price and fuel efficiency of used cars. Looking at the graph, we see:

Upward trend line: This shows a positive relationship between fuel efficiency and selling price. That is, cars with higher fuel efficiency tend to have higher selling prices.

Data points: Are distributed evenly along the trend line, reinforcing the positive relationship between these two variables.

From the graph, buyers can consider choosing a car with high fuel efficiency to save on fuel costs in the long run, even though the initial purchase price may be higher.

5.4.4 Model Year Analysis: Analyze the distribution of vehicle model years.



Columns: YEAR(Year)

Rows: CNT(Selling Price)

Figure 43 Distribution of Vehicle Model Years.

Through the chart in Figure 43, it can be seen that:

- The number of used cars sold has increased significantly from the early 2000s to 2018: This may be due to many factors such as the development of automotive technology, the increase in demand for new cars, and the replacement of old cars. Models from these years may be more popular due to better features and performance.
- The peak around 2018: This may be the year when more cars were produced and sold on the market, leading to a larger number of used cars from this year. This may also reflect the trend of buying new cars and reselling used cars after a few years of use.
- A sharp decline after 2018: After 2018, the number of cars decreased sharply, which may be due to many reasons such as market saturation, changes in shopping trends, or an increase in keeping cars longer before selling. Especially the emergence and development of electric cars and self-driving cars.
- For buyers: Understanding this trend helps buyers know popular car models and can search for models from years with large numbers to have more choices.

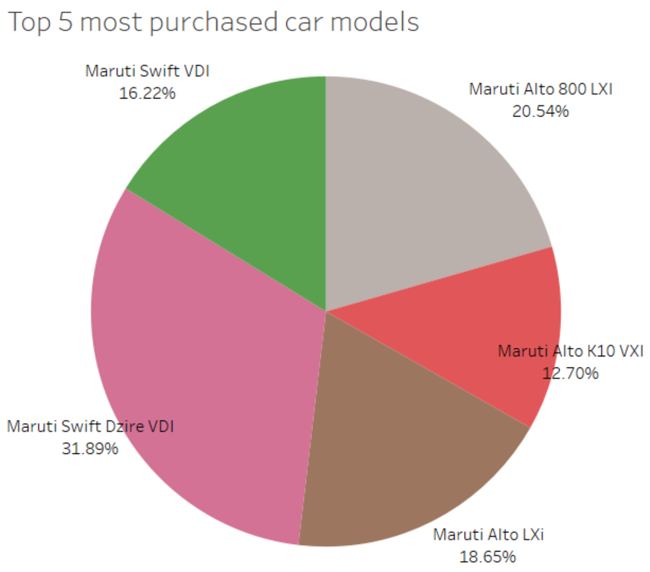


Figure 44 Top 5 most purchased car models.

5.4.5 Using the processed data by Python to create a Story in Tableau

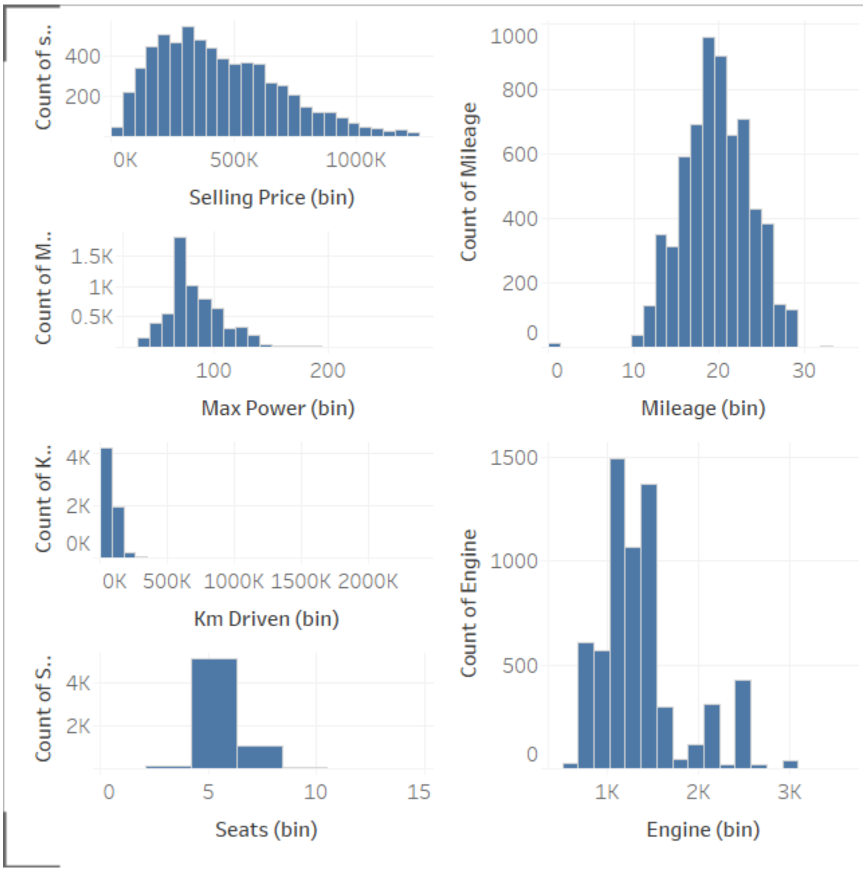


Figure 45 Explore the distribution of values in data.

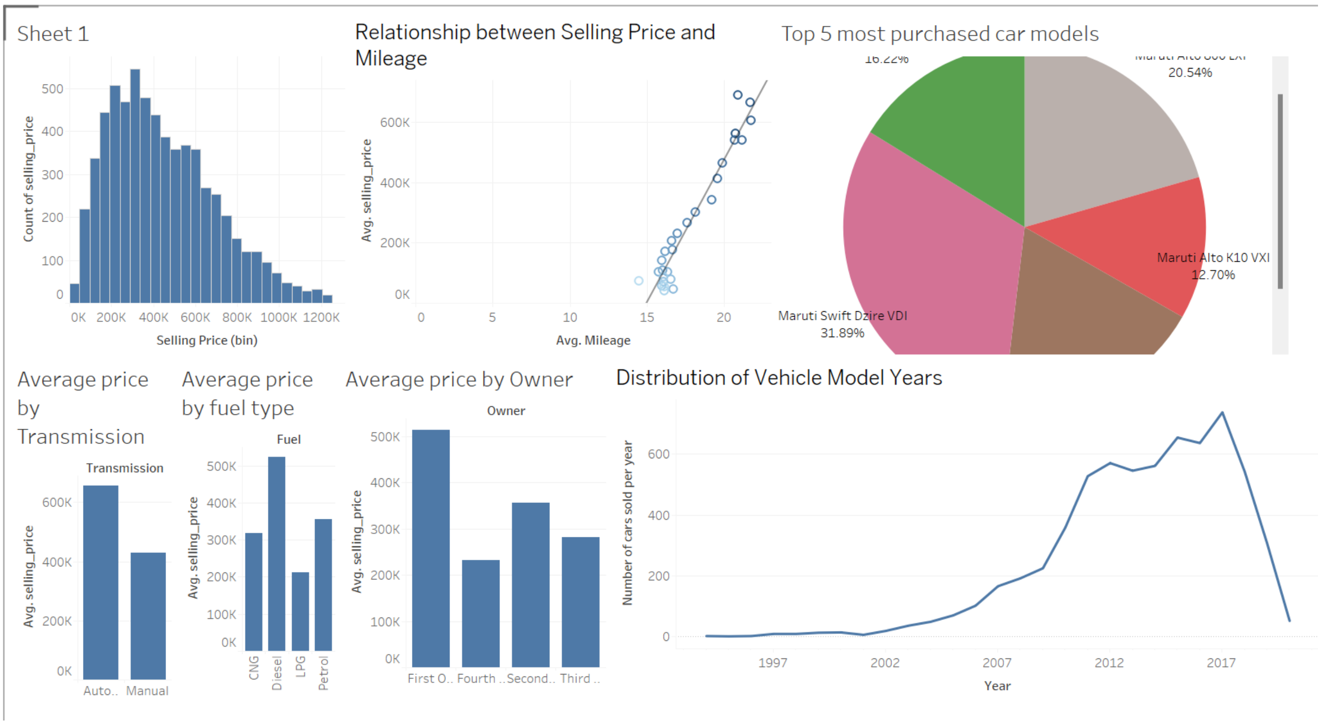
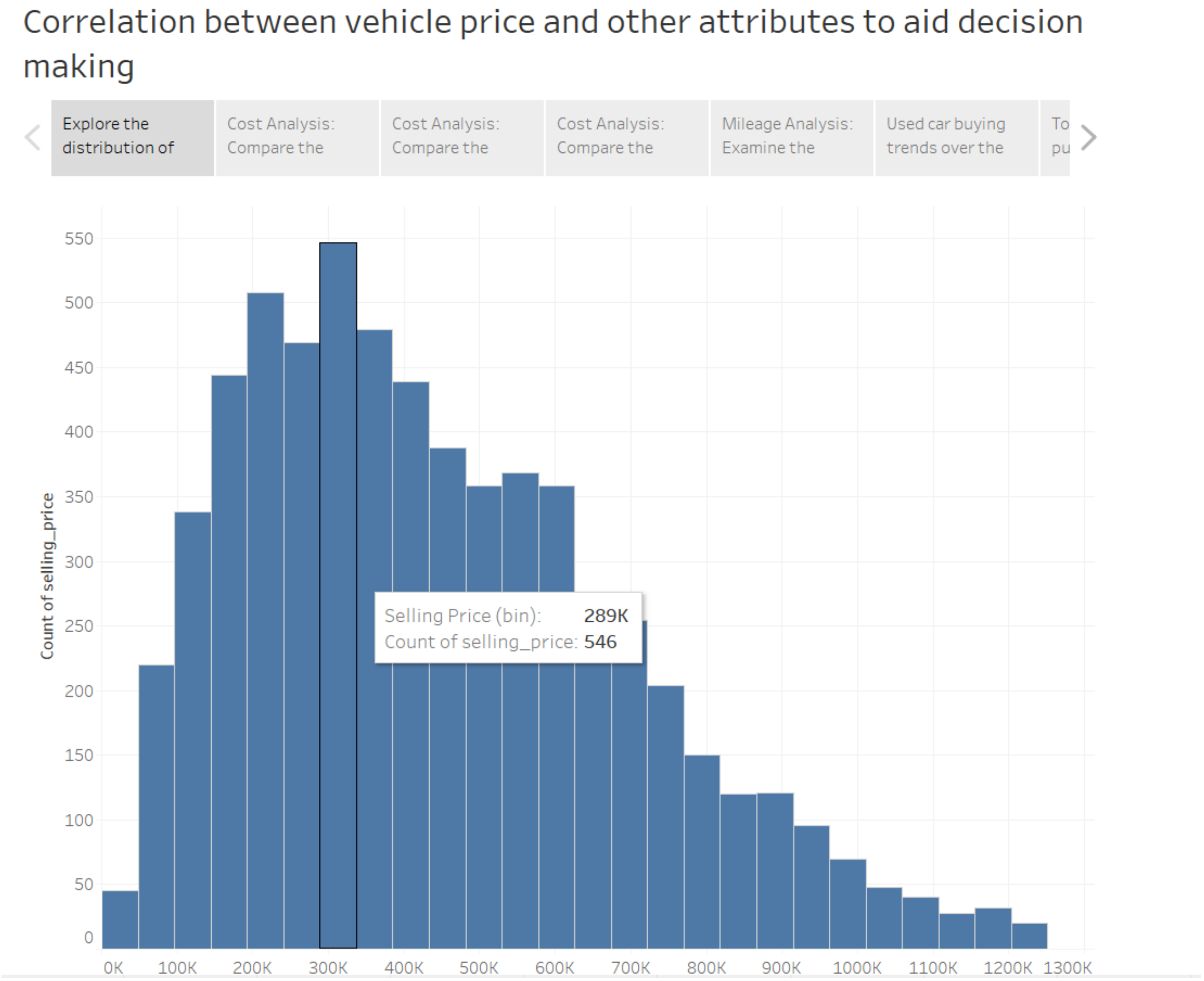


Figure 46 Compare the average prices of vehicles in different conditions.



Correlation between vehicle price and other attributes to aid decision making
Explore the distribution of
Cost Analysis: Compare the
Cost Analysis: Compare the
Cost Analysis: Compare the
Mileage Analysis: Examine the
Used car buying trends over the
To pu
Count of selling_price
550
500
450
400
350
300
250
200
150
100
50
0
0K 100K 200K 300K 400K 500K 600K 700K 800K 900K 1000K 1100K 1200K 1300K
Selling Price (bin): 289K
Count of selling_price: 546

Figure 47 Story: Correlation between car price and other attributes to aid decision-making.

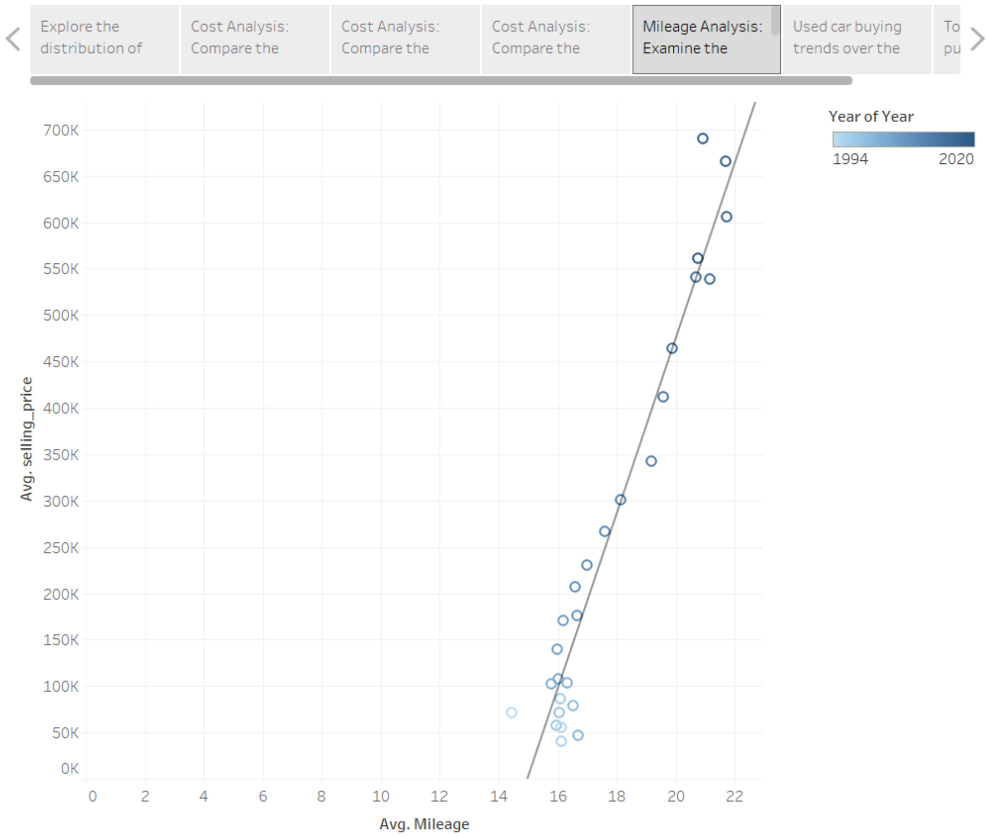


Figure 48 Story: Correlation between car price and other attributes to aid decision-making.

Decision-making: Through the charts, it can be seen that the most popular car models are cars from the Maruti brand. To save costs but ensure convenience, the company should consider buying cars from this brand, with models from 2016 onwards, prices from 250k-350k, and mileage from 150k kilometers or less to ensure long-term use and low loss and repair costs. Choose a car with fuel efficiency from 17-20 km/l of fuel to ensure a reasonable price while the car is still fuel-efficient and within the monthly fuel budget. Choose a car from 70 to 100 horsepower to balance performance and fuel economy. Prioritize choosing gasoline or diesel cars, with 5 to 7 seats to ensure convenience when moving in the city while still being able to carry a relatively large number of people or goods. Besides, to ensure the longevity and long-term use of the car, you should only choose a car that has been through 2 owners, preferably 1 owner. Finally, you can choose a car with a manual transmission to save costs.

5.5 Predict the price for a given requirements of the car provided by the company.

5.5.1 Choosing the appropriate predictive model.

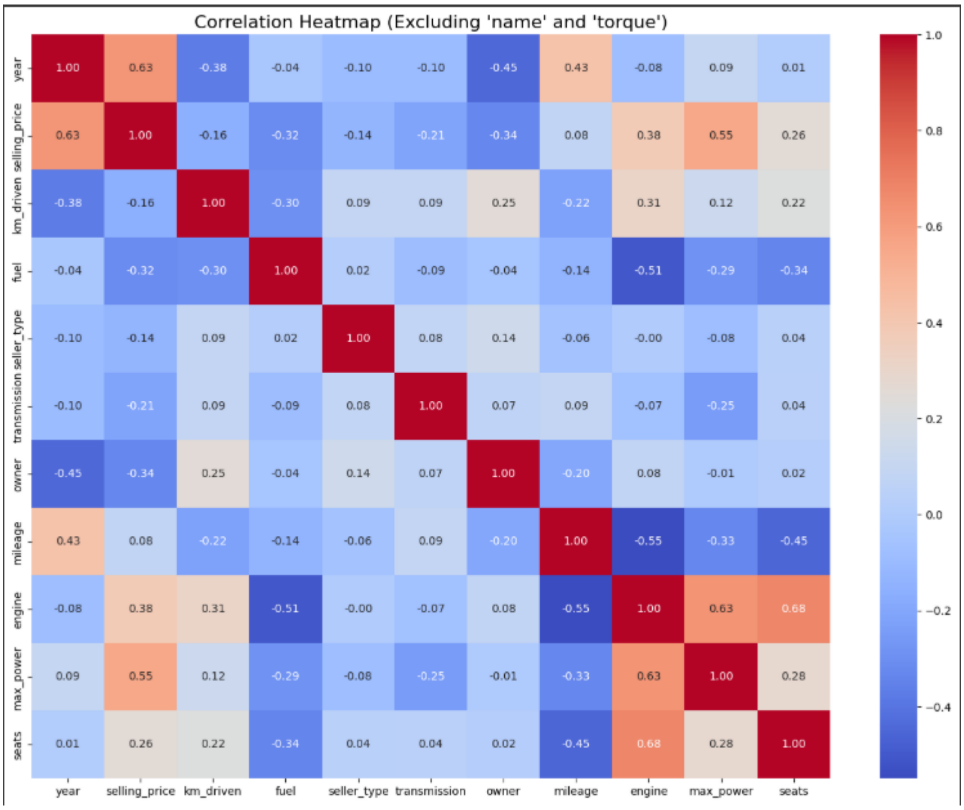


Figure 49 Correlation between features and selling price.

Through Figure 45, we see that the features and targets have a non-linear relationship with each other or there is no clear linear relationship between factors such as year of manufacture, number of kilometers driven, fuel consumption, and fuel type. At the same time, the Data has both numerical features (year, km_driven, mileage, seats, max_power) and categorical features (fuel, transmission).

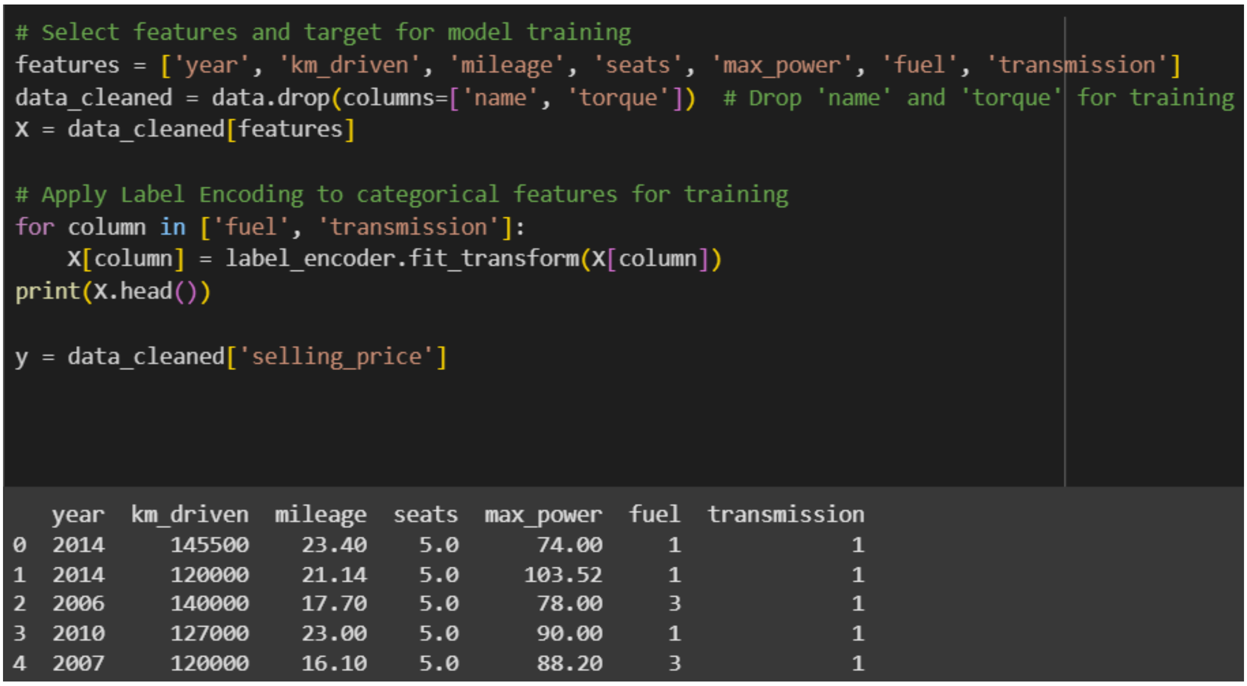
With the above characteristics, I decided to choose RandomForestRegressor to predict car prices for the following reasons:

- Random Forest is a collection of decision trees, and decision trees are very powerful in handling non-linear relationships between features and targets. Random Forest works well in capturing these complex relationships.
- Random Forest is capable of automatically handling data with many mixed features (numerical features, categorical features) without requiring much special processing. Meanwhile, linear models such as Linear Regression require the features to have a linear relationship and often require feature normalization.

- Stability and anti-overfitting: Random Forest uses the "bagging" method (Bootstrap Aggregating) to reduce the variance of the model and anti-overfitting more effectively than models based on a single decision tree.
- Good performance on medium and large data: With large data sets, Random Forest can take advantage of the power of dividing the data into many trees and calculating in parallel, thus operating effectively without taking too much time. Especially for continuous value prediction problems such as car price prediction, Random Forest often gives good results.
- No assumptions about data distribution: Unlike Linear Regression, which requires data to follow assumptions (linearity, no multicollinearity, normal distribution of residuals), Random Forest does not require any assumptions and works well on different types of data.

RandomForestRegressor is a good choice because of its ability to handle nonlinear data, high stability, and ability to work well with mixed features, especially when predicting car prices which are complex and nonlinear.

5.5.2 Build a car price prediction model using RandomForestRegressor.

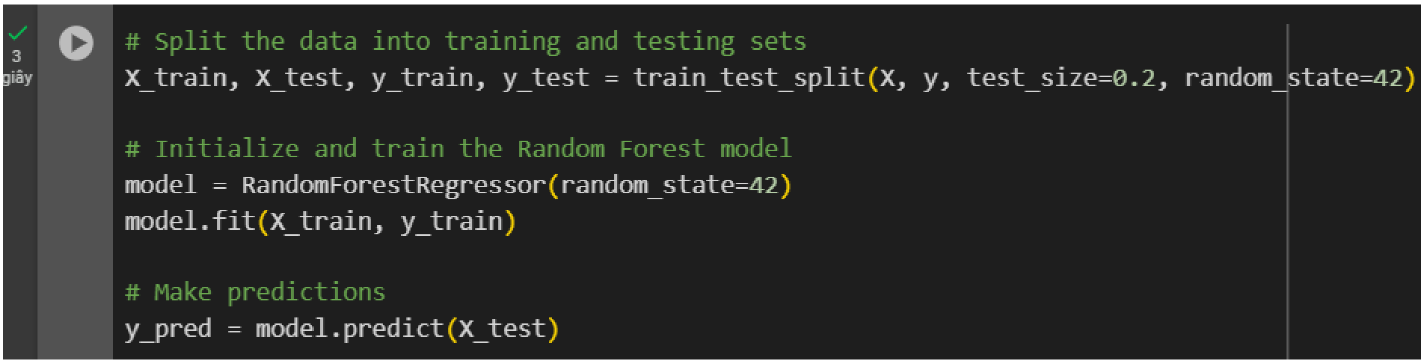


```
# Select features and target for model training
features = ['year', 'km_driven', 'mileage', 'seats', 'max_power', 'fuel', 'transmission']
data_cleaned = data.drop(columns=['name', 'torque'])  # Drop 'name' and 'torque' for training
X = data_cleaned[features]

# Apply Label Encoding to categorical features for training
for column in ['fuel', 'transmission']:
    X[column] = label_encoder.fit_transform(X[column])
print(X.head())

y = data_cleaned['selling_price']
```

```
   year  km_driven  mileage  seats  max_power  fuel  transmission
0  2014     145500    23.40    5.0      74.00     1             1
1  2014     120000    21.14    5.0     103.52     1             1
2  2006     140000    17.70    5.0      78.00     3             1
3  2010     127000    23.00    5.0      90.00     1             1
4  2007     120000    16.10    5.0      88.20     3             1
```

Figure 50 Prepare data, define input data and target variable.

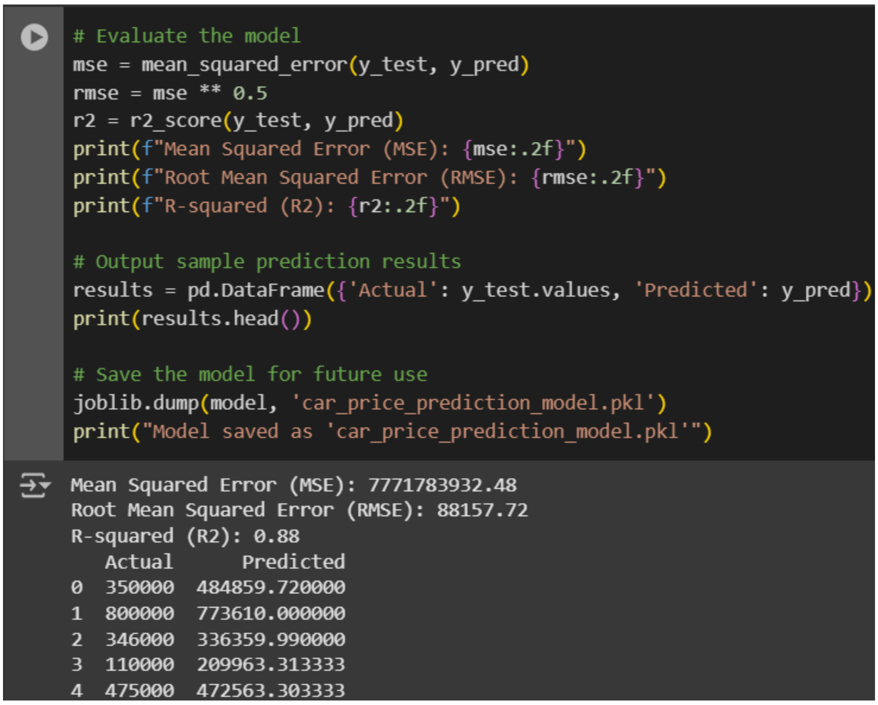
First, I prepare the input data in a suitable format to train the Random Forest Regressor model. Combine numerical features and categorical encoding to create a complete set of input features for the model. Define the target column: selling_price, which is the selling price of the car that the model will predict.



```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the Random Forest model
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
```

Figure 51 Model training and prediction.

Next, I split the data into a training set and a test set to evaluate the model fairly on a test set that has never been seen during training. Train the Random Forest model to learn the relationships between features and car prices. Predict car prices on the test set, preparing for the model performance evaluation steps.



```
# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R2): {r2:.2f}")

# Output sample prediction results
results = pd.DataFrame({'Actual': y_test.values, 'Predicted': y_pred})
print(results.head())

# Save the model for future use
joblib.dump(model, 'car_price_prediction_model.pkl')
print("Model saved as 'car_price_prediction_model.pkl'")
```

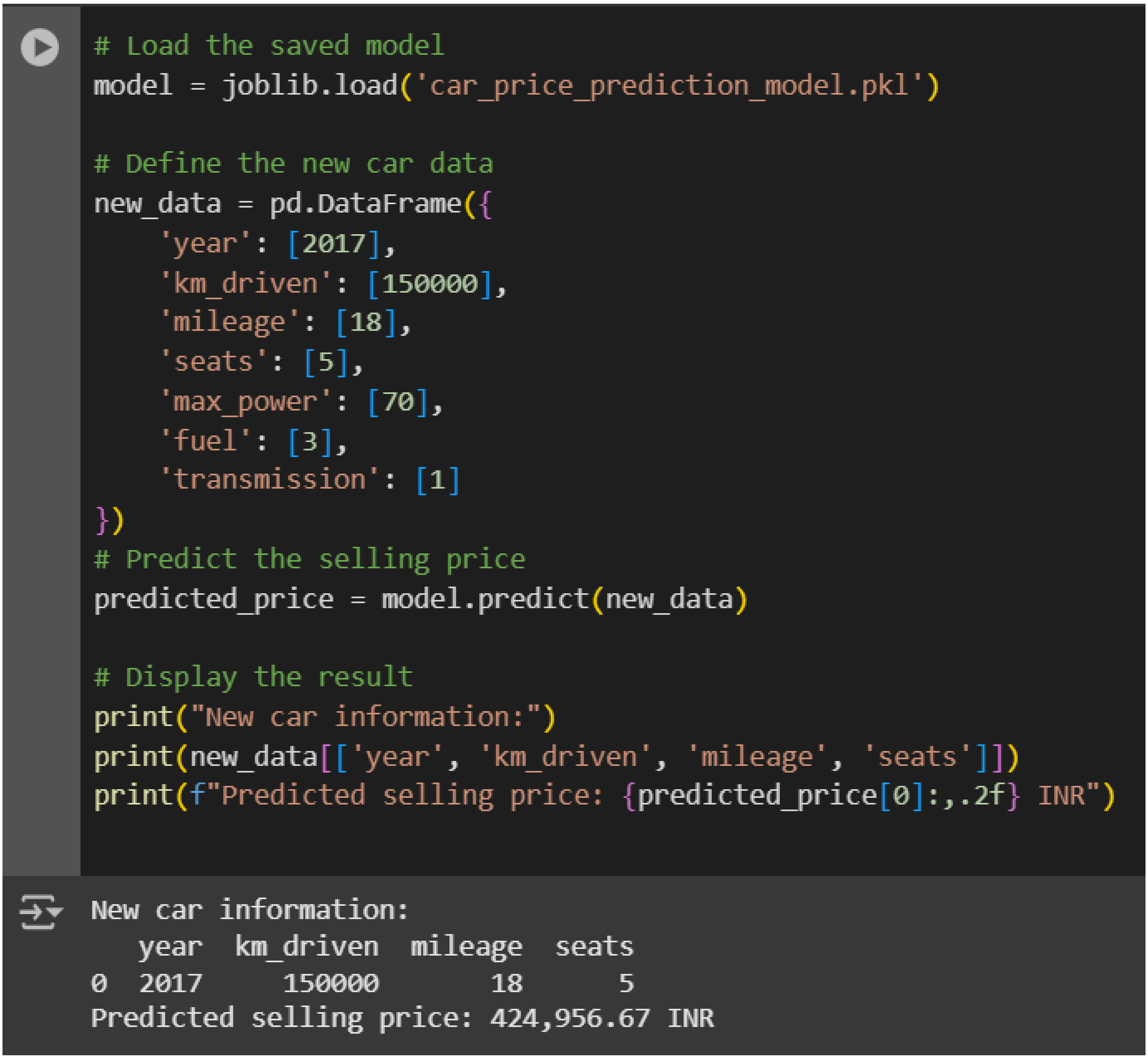
```
Mean Squared Error (MSE): 7771783932.48
Root Mean Squared Error (RMSE): 88157.72
R-squared (R2): 0.88
   Actual      Predicted
0  350000  484859.720000
1  800000  773610.000000
2  346000  336359.990000
3  110000  209963.313333
4  475000  472563.303333
```

Figure 52 Model performance evaluation.

In this step, I evaluate the model using key metrics (MSE and R²) to measure the predictive performance. R² = 0.88 indicates that about 88% of the variation in the target value (selling_price) is explained by the features in the model. Finally, Save the model for reuse in real applications or for deployment.

5.5.3 Car price prediction based on specific requirements.

I will proceed to predict the minimum price required to buy a car with the following requirements: a car manufactured from 2017 onwards, with a mileage of 150,000 km or less to ensure long-term use, low wear and tear and repair costs. Choose a car with fuel economy: 18 km / liter. Choose a car that runs on Petrol fuel, has 5 seats to ensure convenience when moving around the city while still meeting the need to transport a relatively large number of people or goods. A car with a max_power of 70 bhp. Finally, a manual transmission car.



```
# Load the saved model
model = joblib.load('car_price_prediction_model.pkl')

# Define the new car data
new_data = pd.DataFrame({
    'year': [2017],
    'km_driven': [150000],
    'mileage': [18],
    'seats': [5],
    'max_power': [70],
    'fuel': [3],
    'transmission': [1]
})
# Predict the selling price
predicted_price = model.predict(new_data)

# Display the result
print("New car information:")
print(new_data[['year', 'km_driven', 'mileage', 'seats']])
print(f"Predicted selling price: {predicted_price[0]:,.2f} INR")
```

```
New car information:
   year  km_driven  mileage  seats
0  2017     150000       18      5
Predicted selling price: 424,956.67 INR
```

Based on the model, the predicted price for buying a used car is 424,956.67 INR . This price is not the desired purchase price. However, if compared to the announced purchase price of the latest Suzuki Maruti Dzire 2024 with the standard version being INR 679,000 and the premium version starting at INR 1,014,000 (PHONG, 2024), buying a used car will help the company save INR 255,000 - INR 590,000 per car. The company can rely on this to consider the decision of whether to buy a used car or a new car.

Figure 53 Suzuki Maruti Dzire.

Conclusion.

In conclusion, the integration of big data and data-driven decision-making is essential for modern organizations to maintain a competitive edge. By understanding the fundamental concepts of big data and the processes involved in data collection, preprocessing, analysis, and visualization, businesses can make more informed decisions that align with their objectives. The role of data specialists is pivotal in ensuring the successful implementation of big data strategies, though they also face significant challenges in managing complex data sets. Additionally, the use of advanced statistical and graphical tools, combined with real-time analytics, allows organizations to process large volumes of data efficiently and make rapid, informed decisions. Ultimately, the findings of this assignment underscore the importance of adopting scalable systems and leveraging automation to handle big data effectively, paving the way for improved decision-making and business success.

References

George McIntire, Brendan Martin, Lauren Washington, n.d. *Python Pandas Tutorial: A Complete Introduction for Beginners.* [Online]
Available at: https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/
[Accessed 1 12 2024].

Aktualisierte, 2024. *Power BI Tutorial für Einsteiger.* [Online]
Available at: https://www.datacamp.com/de/tutorial/tutorial-power-bi-for-beginners
[Accessed 1 12 2024].

Anderson, M., 2016. *An Introduction to Hadoop.* [Online]
Available at: https://www.digitalocean.com/community/tutorials/an-introduction-to-hadoop
[Accessed 1 12 2024].

aws, n.d. *Apache Hive là gì?.* [Online]
Available at: https://aws.amazon.com/what-is/apache-hive/
[Accessed 1 12 2024].

Bhandari, P., 2023. *Descriptive Statistics | Definitions, Types, Examples.* [Online]
Available at: https://www.scribbr.com/statistics/descriptive-statistics/
[Accessed 1 12 2024].

Bhandari, P., 2023. *Inferential Statistics | An Easy Introduction & Examples.* [Online]
Available at: https://www.scribbr.com/statistics/inferential-statistics/
[Accessed 1 12 2024].

davidiseminger, mohitp930, TimShererWithAquent, KesemSharabi, n.d. *What is Power BI?.* [Online]
Available at: https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview
[Accessed 1 12 2024].

Egg, T., n.d. *Google Data Studio: A Five-Step Beginner's Tutorial.* [Online]
Available at: https://www.theegg.com/seo/apac/google-data-studio-a-five-step-beginners-tutorial/
[Accessed 1 12 2024].

Firican, G., 2017. *The 10 Vs of Big Data.* [Online] Available at: https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx?form=MG0AV3 [Accessed 25 10 2024].

Flare, C., 2024. *Matplotlib Full Tutorial: A Complete Guide to Data Visualization in Python.* [Online] Available at: https://consoleflare.com/blog/matplotlib/ [Accessed 1 12 2024].

Grewcoe, C., 2024. *Time-Series Analysis: What Is It and How to Use It.* [Online] Available at: https://www.timescale.com/blog/time-series-analysis-what-is-it-how-to-use-it/ [Accessed 1 12 2024].

Gupta, L., 2024. *Apache Kafka Tutorial.* [Online] Available at: https://howtodoinjava.com/kafka/apache-kafka-tutorial/ [Accessed 1 12 2024].

Insights, M., 2024. *Data-Driven Decision-Making (DDDM) explained: Making Smarter Business Decisions Using Data.* [Online] Available at: https://morethandigital.info/en/data-driven-decision-making-explained/ [Accessed 27 10 2024].

Jacob Murel Ph.D., Eda Kavlakoglu, 2024. *What is dimensionality reduction?.* [Online] Available at: https://www.ibm.com/topics/dimensionality-reduction [Accessed 1 12 2024].

Keshari, K., 2023. *Data Analyst Roles and Responsibilities : All You Need to Know.* [Online] Available at: https://www.edureka.co/blog/data-analyst-roles-and-responsibilities/ [Accessed 25 10 2024].

Lianne & Justin, 2020. *Plotly Python Tutorial: How to create interactive graphs.* [Online] Available at: https://www.justintodata.com/plotly-python-tutorial-create-graph/ [Accessed 1 12 2024].

Lien, N., 2024. *What is Clustering? Understanding the Applicability of Clustering in Database Administration.* [Online] Available at: https://fptshop.com.vn/tin-tuc/danh-gia/clustering-la-gi--174649 [Accessed 1 12 2024].

Mucci, T., 2024. *What is data-driven decision-making?.* [Online]
Available at: https://www.ibm.com/think/topics/data-driven-decision-making
[Accessed 27 10 2024].

Mutea, B., 2022. *Seaborn tutorial.* [Online]
Available at: https://www.machinelearningnuggets.com/seaborn-tutorial/
[Accessed 1 12 2024].

Ninja, N., 2024. *Inferential Statistics: Making Predictions from data.* [Online]
Available at: https://letsdatascience.com/inferential-statistics-making-predictions-from-data/
[Accessed 1 12 2024].

PHONG, Q., 2024. *Suzuki Dzire 2024 launched: Swift chassis, 5-star safety, converted price just over 200 million VND.* [Online]
Available at: https://tuoitre.vn/suzuki-dzire-2024-ra-mat-khung-gam-swift-5-sao-an-toan-gia-quy-doi-chi-hon-200-trieu-dong-20241112150025967.htm
[Accessed 1 12 2024].

Sarkar, D. (., 2019. *How to use Spark SQL: A hands-on tutorial.* [Online]
Available at: https://opensource.com/article/19/3/apache-spark-and-dataframes-tutorial
[Accessed 1 12 2024].

segment, n.d. *Understanding the 5 V's of Big Data.* [Online]
Available at: https://segment.com/data-hub/big-data/characteristics/#:~:text=Big%20data%20is%20often%20defined,variety%2C%20veracity%2C%20and%20value.%20(Accessed:%2027%20October%202024).
[Accessed 25 10 2024].

Sruthy, 2024. *Apache Spark Tutorial for Beginners: The Ultimate Guide.* [Online]
Available at: https://www.softwaretestinghelp.com/apache-spark-tutorial/
[Accessed 1 12 2024].

Staff, C., 2024. *What Are the 5 Vs of Big Data?.* [Online]
Available at: https://www.coursera.org/articles/5-vs-of-big-data?form=MG0AV3
[Accessed 25 10 2024].

Tableau, n.d. *Tutorial: Get Started with Tableau Desktop.* [Online] Available at: https://help.tableau.com/current/guides/get-started-tutorial/en-us/get-started-tutorial-home.htm
[Accessed 1 12 2024].

Team, D., n.d. *Flink Tutorial – A Comprehensive Guide for Apache Flink.* [Online] Available at: https://data-flair.training/blogs/flink-tutorial/
[Accessed 1 12 2024].

Technologies, I., 2023. *What Are The 7 V's Of Big Data?.* [Online] Available at: https://infycletechnologies.com/seven-vs-of-big-data/
[Accessed 25 10 2024].

Ticong, L., 2024. [Online] Available at: What is Data-Driven Decision-Making? 6 Key Steps (+ Examples)
[Accessed 27 10 2024].

Ticong, L., 2024. *What is Data-Driven Decision-Making? 6 Key Steps (+ Examples).* [Online] Available at: https://www.datamation.com/big-data/data-driven-decision-making/
[Accessed 27 10 2024].

Tutorialsteacher, n.d. *What is D3?.* [Online] Available at: https://www.tutorialsteacher.com/d3js/what-is-d3js
[Accessed 1 12 2024].

Valcheva, S., n.d. *Statistical Methods for Data Analysis: a Comprehensive Guide.* [Online] Available at: https://www.intellspot.com/statistical-methods/
[Accessed 1 12 2024].