A STUDENT'S READING GUIDE

Co-funded by the
Erasmus+ Programme
of the European Union

ICT-4MRPQ

DATA ANALYTICS, VISUALIZATION AND DECISION MAKING FOR:
POSTGRADUATE DIPLOMA IN BUSINESS INTELLIGENCE AND DATA ANALYTICS (PBDA) STUDENTS

Abdal Kasule, PhD
Prof. Samali Violet Mlay, PhD

Title: Data Analytics, Visualization And Decision Making, A Student's Reading Guide For: Postgraduate Diploma In Business Intelligence And Data Analytics (PBDA) Students.

Copyright: © 2025 All rights reserved - Strengthening Capacity for ICT Usage and Policy Reforms for Relevant and Quality Masters Research Process Management in Uganda's HEIs (ICT-4MRPQ)

Funding and support: This book was produced with funding from the European Union, through the Erasmus+ programme, and it's a deliverable of the ICT-4MRPQ Project. Its content is the sole responsibility of the authors and do not reflect the views of the European Union.

Authors: Abdal Kasule, PhD, Prof. Samali Violet Mlay, PhD

Editing: RSET – Associação Técnico-científica para o Desenvolvimento, Lisbon Portugal

Design: Teresa Vaz

Distribution: This material is intended for free distribution only and not be sold or used for commercial purposes.

Further information: Find out more about the project at https://ict-4mrpq.uccsri.com/

ISBN:

Title: Data Analytics, Visualization And Decision Making, A Student's Reading Guide
For: Postgraduate Diploma In Business Intelligence And Data Analytics (PBDA) Students.

Copyright: © 2025 All rights reserved - Strengthening Capacity for ICT Usage and Policy Reforms for Relevant and Quality Masters Research Process Management in Uganda's HEIs (ICT-4MRPQ)

Funding and support: This book was produced with funding from the European Union, through the Erasmus+ programme, and it's a deliverable of the ICT-4MRPQ Project. Its content is the sole responsibility of the authors and do not reflect the views of the European Union.

Authors: Abdal Kasule, PhD, Prof. Samali Violet Mlay, PhD

Editing: RSET – Associação Técnico-científica para o Desenvolvimento, Lisbon Portugal

Design: Teresa Vaz

Distribution: This material is intended for free distribution only and not be sold or used for commercial purposes.

Further information: Find out more about the project at https://ict-4mrpq.uccsri.com/

ISBN:

MAKERERE UNIVERSITY BUSINESS SCHOOL

FACULTY OF COMPUTING AND INFORMATICS

DEPARTMENT OF APPLIED COMPUTING

AND INFORMATION TECHNOLOGY

DATA ANALYTICS, VISUALIZATION AND DECISION MAKING

A STUDENT'S READING GUIDE

For:

POSTGRADUATE DIPLOMA IN BUSINESS INTELLIGENCE AND DATA ANALYTICS (PBDA)

PROGRAMME

November, 2025

This student guide is a product of the ICT-4MRPQ Project,

ICT-4MRPQ

Co-funded by the European Union

A project implemented by

Co-funded by the Erasmus+ Programme of the European Union

Contents

DATA ANALYTICS, VISUALIZATION AND DECISION MAKING.....1
TOPIC ONE: ESSENTIAL CONCEPTS OF DATA, DATA PROBLEMS AND SOLUTIONS ...1
1.1 What is Data Analytics?.....1
1.1.1. Applications of Data Analytics:.....1
1.1.2. Common Tools and Languages:.....1
1.2. Essential Concepts of Data.....1
1.3. Types of Data.....2
1.3.1 Qualitative (Categorical) Data.....2
1.3.2 Quantitative (Numerical) Data.....2
1.3.3 Time Series Data.....2
1.3.4 Spatial (Geospatial) Data.....2
1.3.5 Text Data.....2
1.4. Typical Data Problems and Solutions.....2
1.5. Skill Set Needed for Data Analytics.....3
1.6. The Data Analytics Process.....3
1.7. Topic One Summary.....3
1.8. End of Topic One Exercises: Essential Concepts of Data, Data Problems and Solutions.....3
1.8.1. Conceptual Questions.....3
1.8.2. Analytical Questions.....4
1.8.3. Practical / Applied Questions.....4
1.8.4. Scenario-Based Questions.....4
1.8.5. Mini-Project.....4
1.9. Topic One References.....5
TOPIC TWO: VARIABLES, POPULATION, SAMPLES AND STATISTICAL MODELLING ...6
2.1. Variables.....6
2.2. Populations and Samples.....6
2.3. Summary Measures.....7
2.4. Categorical Data.....7
2.5. Statistical Modelling.....7
2.5.1. Linear Regression.....8
2.5.2. Logistic Regression.....9
2.5.3. Time Series Models..... .9
2.5.4. ANOVA (Analysis of Variance).....10
2.5.5 Summary Table of Statistical Models.....10
2.6. Descriptive Modelling.....11
2.6.1. Clustering (e.g., K-Means).....11
2.6.2. Association Rule Mining.....12
2.6.3. Principal Component Analysis (PCA).....12

2.6.4 Summary Table of Descriptive Modelling Techniques....13
2.7. Predictive Modelling....13
2.7.1. Regression (Linear and Logistic)....14
2.7.2. Decision Trees....15
2.7.3. Su pport Vector Machines (SVM)....15
2.7.4. Neural Networks....16
2.7.5 Comparison Table of Predictive Modelling Techniques....17
2.8. Fitting a Model....17
2.9. Topic Two Summary....17
2.10 End of Topic Two Exercises: Variables, Population, Samples, and Statistical Modelling....17
2.10.1. Conceptual Questions....17
2.10.2. Analytical Questions....18
2.10.3. Practical / Applied Questions....18
2.10.4. Scenario-Based Questions....18
2.10.5. Mini-Project / Practical Exercise....18
2.11 Topic Two References....19
TOPIC THREE: EXPLORATORY DATA ANALYSIS (EDA)....20
3.1. Philosophy of Exploratory Data Analysis....20
3.2. Basic EDA Tools....20
3.3. Summary Statistics....20
3.4.Histograms....21
3.5. Boxplots....21
3.6. Scatter Plots....22
3.7. Correlation Matrix....22
3.8 Topic Three Summary....22
3.9 End of Topic Three Exercises: Exploratory Data Analysis (EDA)....23
3.9.1. Conceptual Questions....23
3.9.2. Analytical Questions....23
3.9.3. Practical / Applied Questions....23
3.9.4. Scenario-Based Questions....24
3.9.5. Mini-Project / Practical Exercise....24
3.10 Topic Three References....24
TOPIC FOUR: DATA VISUALIZATION AND DECISION MAKING....,....25
4.1. Basic Principles of Data Visualization....25
4.2. Popular Data Visualization Tools....25
4.2.1 Matplotlib....25
4.2.2 Seaborn....25
4.2.3 Tableau....26
4.2.4 Power BI....26
4.2.5 Plotly....26
4.3. Case Studies of Inspiring Projects....26

4.4. Dashboards and Storytelling....26
4.5. Creating a Simple Dashboard in Python....26
4.6. Decision Making....27
4.7. How Dashboards Aid Decision Making....27
4.8. Topic Four Summary....28
4.9 End-of-Topic Exercises: Data Visualization and Decision Making....28
4.9.1. Conceptual Questions....28
4.9.2. Tool-Based Questions....28
4.9.3. Practical / Applied Exercises....28
4.9.4. Scenario-Based Questions....29
4.9.5. Mini-Project / Practical Exercise....29
4.10 Topic Four References....29
TOPIC FIVE: PREDICTIVE AND PRESCRIPTIVE ANALYTICS....30
5.1. Overview....30
5.1.1 Predictive Analytics (PA)....30
5.1.2 Prescriptive Analytics (PSA)....31
5.1.3 Use Case Example: E-commerce....33
5.1.4 Key Differences Between PA and PSA....34
5.2. Connecting Predictive Analytics to Business Objectives....34
5.3. Gathering Critical Business Insights....34
5.4. Making a Business Decision....35
5.5. Identifying Market Trends....35
5.6. Approaches to a Business Challenge....35
5.7. Forecasting Future Demand....36
5.8. Topic Five Summary....36
5.9 End-of-Topic Exercises: Predictive and Prescriptive Analytics....36
5.9.1. Conceptual Questions....36
5.9.2. Tool and Python-Based Questions....36
5.9.3. Scenario-Based Questions....37
5.9.4. Mini-Practical / Applied Exercise....37
5.9.5. Business Insight and Decision-Making Questions....38
5.10. Topic Five References....38

TOPIC ONE:

ESSENTIAL CONCEPTS OF DATA, DATA PROBLEMS AND SOLUTIONS

1.1 What is Data Analytics?

Data analytics is the process of examining, organizing, transforming, and interpreting raw data to uncover meaningful insights, patterns, and trends that support decision-making. It involves techniques such as data cleaning, statistical analysis, visualization, and predictive modeling to extract value from data.

1.1.1. Applications of Data Analytics:

- **Business:** Understanding customer behavior, sales forecasting, marketing optimization.
- **Healthcare:** Predicting patient outcomes, disease diagnosis, hospital resource management.
- **Finance:** Risk management, fraud detection, investment analysis.
- **Technology:** Recommender systems, usage analytics, product improvement.

1.1.2. Common Tools and Languages:

- Programming: **Python**, **R**, **SQL**
- Libraries: **Pandas**, **NumPy**, **scikit-learn**, **Matplotlib**, **Seaborn**
- Platforms: **Excel**, **Tableau**, **Power BI**

1.2. Essential Concepts of Data

Data refers to raw facts, figures, or symbols collected from observations, measurements, or processes. Data becomes meaningful when it is processed and analysed.

Key Concepts:

- **Data Types:**
    - *Numerical:* Quantitative measurements (e.g., sales, height).
    - *Categorical:* Qualitative information (e.g., gender, education level).
    - *Text:* Unstructured written content (e.g., emails, reviews).
- **Data Structures:** Tables, arrays, data frames, databases.
- **Data Formats:** CSV, JSON, Excel, SQL databases.
- **Data Quality:** Accuracy, completeness, consistency, timeliness.
- **Data Processing:** Collection, storage, cleaning, normalization, handling missing values.
- **Data Privacy & Security:** Ensuring data protection and ethical use.

1.3. Types of Data

1.3.1 Qualitative (Categorical) Data

- Non-numeric data describing qualities or categories.
- **Nominal:** No natural order (e.g., gender, color).
- **Ordinal:** Has meaningful order (e.g., education level, customer rating).
- **Analysis Techniques:** Counts, proportions, bar charts, pie charts.

1.3.2 Quantitative (Numerical) Data

- Numeric data representing measurable quantities.
- **Discrete:** Countable (e.g., number of students).
- **Continuous:** Measurable (e.g., temperature, salary).
- **Analysis Techniques:** Mean, median, standard deviation, histograms, scatter plots.

1.3.3 Time Series Data

- Sequential data collected at regular intervals.
- Examples: Stock prices, daily sales, temperature trends.
- Analysis Methods: Moving averages, forecasting, trend analysis, time series modeling.

1.3.4 Spatial (Geospatial) Data

- Data representing physical locations or shapes on Earth.
- Examples: Maps, satellite images, GIS data.
- Applications: Urban planning, navigation, environmental monitoring.
- Visualizations: Maps, spatial plots, choropleth maps.

1.3.5 Text Data

- Unstructured natural language content.
- Examples: Emails, social media posts, product reviews.
- Analysis Techniques: Natural Language Processing (NLP), sentiment analysis, keyword extraction, topic modeling.

1.4. Typical Data Problems and Solutions

Common Issues:

- Missing data
- Duplicate entries
- Inconsistent formats
- Outliers and noise
- Irrelevant features

Solutions:

- **Data Cleaning:** Impute missing values (mean, median, predictive models).
- **Duplicates Removal:** Filtering and deduplication.
- **Format Standardization:** Date, numeric, and string normalization.
- **Outlier Detection:** Statistical methods or domain-specific rules.
- **Feature Selection:** Remove irrelevant or redundant variables.

Tools: Python libraries such as **Pandas**, **NumPy**, **scikit-learn** automate these processes efficiently.

1.5. Skill Set Needed for Data Analytics

- **Technical Skills:** Python/R programming, SQL, Excel, data visualization (Tableau, Power BI), basic machine learning.
- **Analytical Skills:** Statistics, probability, data preprocessing, EDA, modeling.
- **Communication Skills:** Present insights via dashboards, reports, and presentations.
- **Additional Tools:** Jupyter Notebook, Git, cloud platforms (AWS, Google Cloud).
- **Soft Skills:** Problem-solving, critical thinking, curiosity.

1.6. The Data Analytics Process

1. **Problem Definition:** Clarify objectives and questions.
2. **Data Collection:** Gather data from internal/external sources.
3. **Data Cleaning & Preprocessing:** Address missing values, outliers, and inconsistencies.
4. **Exploratory Data Analysis (EDA):** Use statistical summaries and visualizations to understand data patterns.
5. **Modelling:** Apply statistical or machine learning models for insights or predictions.
6. **Evaluation:** Validate results for accuracy, reliability, and relevance.
7. **Communication:** Present insights using visualizations, dashboards, and reports.
8. **Decision Making:** Use insights to inform strategies and actions.

1.7. Topic One Summary

Data analytics is a critical discipline that transforms raw data into actionable insights. By understanding types of data, addressing data quality challenges, applying analytical techniques, and communicating results effectively, organizations can leverage data for strategic advantage.

1.8. End of Topic One Exercises: Essential Concepts of Data, Data Problems and Solutions

1.8.1. Conceptual Questions

1. Define data analytics and explain its importance in at least three domains (business, healthcare, and finance). (6 marks)
2. Differentiate between qualitative (categorical) and quantitative (numerical) data, providing two examples of each. (6 marks)

3. Explain the difference between discrete and continuous numerical data with real-world examples. (4 marks)

4. What is time series data? Give two examples and explain why special analysis methods are needed. (4 marks)

5. Define spatial (geospatial) data and describe one application in urban planning or environmental monitoring. (4 marks)

6. Explain the importance of data quality in analytics and list four key dimensions of data quality. (6 marks)

1.8.2. Analytical Questions

7. Identify at least three common data problems in a dataset containing customer sales data, and suggest appropriate solutions using Python libraries like Pandas or NumPy. (6 marks)

8. A dataset has missing values in the “Age” column. Describe two methods to handle these missing values and discuss when each method would be appropriate. (4 marks)

9. Explain why removing duplicate entries and standardizing data formats are critical steps in data preprocessing. (4 marks)

10. Describe the skills needed for a data analyst and categorize them into technical, analytical, communication, and soft skills. Provide one example under each category. (6 marks)

1.8.3. Practical / Applied Questions

11. You are given a CSV file containing sales data with columns: Date, Product, Quantity, Price. Write pseudocode or Python steps to:
    a) Load the data.
    b) Identify and fill missing values in Quantity.
    c) Remove duplicate rows.
    d) Standardize the date format to YYYY-MM-DD. (8 marks)

12. For the same dataset, describe how you would perform exploratory data analysis (EDA) to:
    a) Find the top 5 best-selling products.
    b) Visualize monthly sales trends.
    c) Detect outliers in the Price column. (6 marks)

1.8.4. Scenario-Based Questions

13. Your company wants to predict customer churn. Describe the steps of the data analytics process you would follow from problem definition to decision making. Include the type of data you would need at each step. (10 marks)

14. A social media platform wants to analyze user comments to determine sentiment (positive, negative, neutral).
    a) Identify the type of data involved.
    b) Suggest preprocessing steps to clean the text data.
    c) Mention one Python library or tool that can be used for sentiment analysis. (6 marks)

1.8.5. Mini-Project

15. You are provided with a dataset containing patient information: PatientID, Age, Gender, Diagnosis, VisitDate.

- Perform data cleaning: handle missing values, remove duplicates, standardize VisitDate.

- Identify the number of patients per diagnosis and visualize the result using a bar chart.
- Calculate basic statistics (mean, median, standard deviation) for Age.
- Submit a report describing your process, code, and findings. (15 marks)

1.9. Topic One References

- Dasu, T., & Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*. Wiley.
- Davenport, T., & Harris, J. (2007). *Competing on Analytics: The New Science of Winning*. Harvard Business Review Press.
- Dhar, V. (2013). *Data Science and Prediction*. Communications of the ACM, 56(12), 64–73.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
- Kelleher, J., Mac Carthy, M. (2018). *Data Science and Artificial Intelligence: A Guide for Non-Experts*. CRC Press.
- Marr, B. (2016). *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. Wiley.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media.

TOPIC TWO:

VARIABLES, POPULATION, SAMPLES AND STATISTICAL MODELLING

2.1. Variables

Variables are symbolic names that store data values for processing, analysis, and modeling. They can hold numbers, strings, or complex data structures like lists, dictionaries, or DataFrames.

Key Points:

- Essential for storing **input data**, intermediate results, and outputs.
- In **Python**, variables are dynamically typed: the type is inferred from the value assigned.
- Proper variable naming improves **code readability** and **maintainability**.

Examples in Python:

```
# Numeric variable
age = 25
# String variable
name = “Amina”
# List variable
scores = [85, 90, 78, 92]
# DataFrame variable using Pandas
import pandas as pd
data = pd.DataFrame({
    ‘Name’: [‘Amina’, ‘Brian’],
    ‘Score’: [90, 88]
})
print(name, age)
print(scores)
print(data)
```

2.2. Populations and Samples

- **Population:** The complete set of items or individuals under study.
- **Sample:** A subset of the population used for analysis, often due to constraints of time or cost.
- Sampling allows **statistical inference** about the population from observed data.

Example in Python:

```
import numpy as np

population = np.arange(1, 101)  # Population from 1 to 100

sample = np.random.choice(population, size=10, replace=False)

print("Sample:", sample)
```

2.3. Summary Measures

Summary measures describe key aspects of a dataset such as **central tendency** and **variability**.

Common Measures:

- **Mean:** Average value
- **Median:** Middle value
- **Mode:** Most frequent value
- **Standard Deviation (Std Dev):** Spread of data
- **Range:** Difference between max and min

Example in Python:

```
import pandas as pd

data = pd.Series([10, 15, 20, 25, 30])

print("Mean:", data.mean())

print("Median:", data.median())

print("Std Dev:", data.std())
```

2.4. Categorical Data

- **Categorical variables** represent data that can take a limited set of categories.
- **Types:**
    - **Nominal:** No order (e.g., gender, color)
    - **Ordinal:** Ordered (e.g., education level, satisfaction rating)
- **Analysis:** Frequency counts, group statistics, visualizations (bar charts, pie charts)

Example in Python:

```
df = pd.DataFrame({'Color': ['Red', 'Blue', 'Red', 'Green', 'Blue']})

print(df['Color'].value_counts())  # Frequency of each category
```

2.5. Statistical Modelling

Statistical modelling is the process of using **mathematical representations** to describe relationships between variables, quantify patterns in data, and make predictions or inferences. It allows analysts to **understand complex systems**, test hypotheses, and support decision-making by summarizing observed data into structured models.

Purpose of Statistical Modelling:

1. **Understanding Relationships:** Identify how independent variables influence dependent variables.
2. **Prediction:** Forecast outcomes for new or unseen data.
3. **Inference:** Test hypotheses and quantify uncertainty about relationships.
4. **Decision Support:** Provide evidence-based guidance for strategy, operations, or research.

Key Components:

- **Dependent Variable (Response):** The outcome we want to predict or explain.
- **Independent Variables (Predictors):** Factors that influence the outcome.
- **Model Parameters:** Quantitative measures that describe the strength and direction of relationships.
- **Error Term:** Represents variability not explained by the model.

2.5.1. Linear Regression

Predict a continuous dependent variable based on one or more independent variables. Assumes a **linear relationship** between predictors and outcome.

Mathematical Form:

$y=\beta_0+\beta_1x_1+\beta_2x_2+\square+\epsilon$

Where y = dependent variable, βi = coefficients, xi = predictors, ϵ = error term.

Python Example:

```
from sklearn.linear_model import LinearRegression

import numpy as np


# Sample data: size of house vs price

X = np.array([[600], [800], [1000], [1200], [1400]])

y = np.array([150, 200, 250, 300, 350])


model = LinearRegression()

model.fit(X, y)


# Predict price for a new house

predicted_price = model.predict([[1100]])

print(f”Predicted price: {predicted_price[0]}”)
```

Applications:

- Forecasting sales
- Predicting housing prices
- Estimating operational costs

2.5.2. Logistic Regression

Predict a **categorical outcome**, typically binary (e.g., Yes/No, 0/1), based on independent variables. Unlike linear regression, the output is modeled using the **logistic function** to yield probabilities.

Mathematical Form:

$z=\beta_0+\beta_1X_1+...+\beta_nX_n$

Python Example:

```
from sklearn.linear_model import LogisticRegression

import numpy as np

# Example: hours studied vs pass/fail

X = np.array([[1], [2], [3], [4], [5]])

y = np.array([0, 0, 0, 1, 1])  # 0=fail, 1=pass

model = LogisticRegression()

model.fit(X, y)


pred_prob = model.predict_proba([[3.5]])[0,1]

print(f"Probability of passing: {pred_prob:.2f}")
```

Applications:

- Predicting customer churn
- Credit scoring (default risk)
- Disease diagnosis (presence/absence)

2.5.3. Time Series Models

Analyse data collected over **time intervals** to detect trends, seasonal patterns, and forecast future values.

Common Models:

- **ARIMA (AutoRegressive Integrated Moving Average):** Captures trends and seasonality.
- **Exponential Smoothing:** Assigns exponentially decreasing weights to past observations.

Python Example:

```
import pandas as pd

from statsmodels.tsa.arima.model import ARIMA
```

```
# Sample monthly sales data
data = pd.Series([120, 130, 125, 140, 150, 160])
model = ARIMA(data, order=(1,1,0))
model_fit = model.fit()
forecast = model_fit.forecast(steps=2)
print("Forecasted sales:", forecast.values)
```

Applications:

- Sales and demand forecasting
- Stock market analysis
- Energy consumption prediction

2.5.4. ANOVA (Analysis of Variance)

Determine whether there are **statistically significant differences** between the means of three or more independent groups. Useful for testing hypotheses about group differences.

Python Example:

```
import pandas as pd
from scipy.stats import f_oneway

# Example: exam scores for 3 study methods
group1 = [85, 88, 90]
group2 = [78, 82, 80]
group3 = [92, 95, 91]

f_stat, p_val = f_oneway(group1, group2, group3)
print(f"F-statistic: {f_stat:.2f}, p-value: {p_val:.3f}")
```

Applications:

- Comparing effectiveness of teaching methods
- Testing product quality across production lines
- Clinical trial comparisons

2.5.5 Summary Table of Statistical Models

Model Type	Purpose	Outcome Type	Example Use
Linear Regression	Predict continuous variable	Continuous	House price prediction
Logistic Regression	Predict categorical outcome	Binary/Categorical	Customer churn prediction
Time Series	Forecast future trends	Continuous over time	Monthly sales forecasting
ANOVA	Compare group means	Categorical groups	Effect of training programs on performance

2.6. Descriptive Modelling

Descriptive modelling is a **data analysis approach** that focuses on summarizing and interpreting historical data to uncover **patterns, relationships, clusters, or associations**. Unlike predictive or prescriptive modelling, descriptive models **do not forecast future outcomes**; instead, they provide insights into what has already happened. This is critical for understanding past behaviour, segmenting data, and identifying trends or anomalies.

Purpose of Descriptive Modelling:

1. **Data Summarization:** Convert raw data into structured, understandable summaries.
2. **Pattern Discovery:** Identify recurring behaviors or trends in historical data.
3. **Segmentation:** Group similar observations together for analysis.
4. **Decision Support:** Provide foundational insights to guide predictive or prescriptive analytics.

2.6.1. Clustering (e.g., K-Means)

Clustering groups similar data points together based on their attributes, revealing **natural patterns or segments** in the dataset. It is widely used for **customer segmentation, market analysis, and anomaly detection**.

K-Means Clustering:

- A popular unsupervised learning algorithm.
- Divides the dataset into k clusters based on similarity.
- Each cluster has a **centroid** representing its center.

Python Example:

```
from sklearn.cluster import KMeans

import numpy as np


# Sample dataset: customer features

X = np.array([[1, 2], [1, 4], [1, 0],

              [10, 2], [10, 4], [10, 0]])


# Create and fit K-Means model

kmeans = KMeans(n_clusters=2, random_state=0).fit(X)


print("Cluster labels:", kmeans.labels_)

print("Cluster centers:", kmeans.cluster_centers_)
```

Applications:

- Market segmentation
- Identifying patterns in customer purchase behavior
- Fraud detection

2.6.2. Association Rule Mining

Association rule mining identifies **relationships or correlations between variables** in large datasets. It is particularly used in **retail, marketing, and recommendation systems** to understand which items or behaviors occur together frequently.

Key Concepts:

- **Support:** Frequency of the itemset in the dataset.
- **Confidence:** Likelihood of occurrence of the consequent given the antecedent.
- **Lift:** Measures the strength of the association relative to random chance.

Python Example (using mlxtend library):

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules

# Sample transactions
dataset = pd.DataFrame({
    'Milk': [1,0,1,1,0],
    'Bread': [1,1,0,1,1],
    'Butter':[0,1,1,0,1]
})

# Frequent itemsets
frequent_itemsets = apriori(dataset, min_support=0.6, use_colnames=True)

# Generate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
print(rules)
```

Applications:

- Market basket analysis (e.g., "customers who buy milk also buy bread")
- Cross-selling and upselling strategies
- Web clickstream pattern analysis

2.6.3. Principal Component Analysis (PCA)

PCA is a **dimensionality reduction technique** used to simplify large datasets by transforming variables into a smaller number of **principal components** while preserving as much variance as possible. It helps identify **key patterns** and reduces computational complexity.

Python Example:

```
from sklearn.decomposition import PCA
import numpy as np
```

```
# Sample dataset
X = np.array([[2.5, 2.4],
              [0.5, 0.7],
              [2.2, 2.9],
              [1.9, 2.2],
              [3.1, 3.0]])

# Apply PCA to reduce to 1 component
pca = PCA(n_components=1)
principal_component = pca.fit_transform(X)

print("Principal Component:\n", principal_component)
```

Applications:

- Reducing data dimensionality for visualization
- Feature extraction in machine learning
- Identifying latent patterns in customer or sensor data

2.6.4 Summary Table of Descriptive Modelling Techniques

Technique	Purpose	Example Use
Clustering	Segment data into similar groups	Customer segmentation, anomaly detection
Association Rule Mining	Discover frequent patterns and relationships	Market basket analysis, recommendations
PCA	Reduce dimensionality and highlight patterns	Feature extraction, data visualization

2.7. Predictive Modelling

Predictive modelling is the process of **using historical and current data** to make informed predictions about **future outcomes**. It applies **statistical, machine learning, and data mining techniques** to identify patterns, quantify relationships, and forecast unknown or future events. Predictive models help organizations **anticipate trends, behaviors, risks, and opportunities**, supporting evidence-based decision-making.

Purpose of Predictive Modelling:

1. **Forecasting:** Anticipate future events, such as sales, demand, or customer churn.
2. **Classification:** Categorize data points into predefined classes, like fraud vs. non-fraud.
3. **Decision Support:** Provide insights for strategic, operational, or marketing decisions.
4. **Risk Management:** Predict potential risks or failures before they occur.

2.7.1. Regression (Linear and Logistic)

2.7.1.1 Linear Regression:

- Predicts a **continuous numeric outcome** based on one or more predictors.
- Assumes a linear relationship between independent variables and the dependent variable.

Example in Python:

```
from sklearn.linear_model import LinearRegression
import numpy as np

# Historical data: house size vs price
X = np.array([[600], [800], [1000], [1200], [1400]])
y = np.array([150, 200, 250, 300, 350])

model = LinearRegression()
model.fit(X, y)

predicted_price = model.predict([[1100]])
print(f"Predicted price: {predicted_price[0]}")
```

2.7.1.2 Logistic Regression:

- Predicts a **categorical outcome**, usually binary.
- Outputs probabilities that an instance belongs to a particular class.

Example in Python:

```
from sklearn.linear_model import LogisticRegression
import numpy as np

# Example: hours studied vs pass/fail
X = np.array([[1], [2], [3], [4], [5]])
y = np.array([0, 0, 0, 1, 1])  # 0=fail, 1=pass

model = LogisticRegression()
model.fit(X, y)

pred_prob = model.predict_proba([[3.5]])[0,1]
```

print(f"Probability of passing: {pred_prob:.2f}")

Applications:

- Sales forecasting, revenue prediction
- Credit scoring, risk assessment
- Customer churn prediction

2.7.2. Decision Trees

Decision trees are **tree-like structures** that split data based on feature values to predict an outcome. They are easy to interpret and can handle both **categorical and continuous variables**.

Python Example:

```
from sklearn.tree import DecisionTreeClassifier

# Sample data: feature vs class
X = [[0], [1], [2], [3]]
y = ['no', 'no', 'yes', 'yes']

model = DecisionTreeClassifier()
model.fit(X, y)

print("Prediction for 1.5:", model.predict([[1.5]])[0])
```

Applications:

- Fraud detection
- Customer segmentation
- Loan approval

2.7.3. Support Vector Machines (SVM)

SVM is a supervised learning technique used for **classification and regression**. It works by finding a **hyperplane** that best separates classes in the feature space. SVMs are effective for **high-dimensional data** and cases where classes are not linearly separable using kernels.

Python Example:

```
from sklearn.svm import SVC
import numpy as np

# Sample data
X = np.array([[1,2], [2,3], [3,3], [6,5], [7,8]])
y = ['A','A','A','B','B']
```

```
model = SVC(kernel='linear')
model.fit(X, y)

print("Prediction for [4,4]:", model.predict([[4,4]])[0])
```

Applications:

- Image classification
- Text categorization
- Bioinformatics (gene expression analysis)

2.7.4. Neural Networks

Neural networks are **computational models inspired by the human brain**, capable of capturing complex non-linear relationships. They consist of **layers of interconnected nodes (neurons)**. Neural networks are widely used in deep learning for tasks like **image recognition, natural language processing, and time-series forecasting**.

Python Example (using Keras):

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np

# Sample data
X = np.array([[0], [1], [2], [3]])
y = np.array([0, 0, 1, 1])

model = Sequential([
    Dense(4, input_dim=1, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X, y, epochs=100, verbose=0)

print("Prediction for 1.5:", model.predict([[1.5]])[0][0])
```

Applications:

- Predicting stock prices
- Speech and image recognition
- Customer behavior modeling

2.7.5 Comparison Table of Predictive Modelling Techniques

Technique	Purpose	Data Type	Pros	Applications
Linear Regression	Predict continuous outcomes	Numeric	Simple, interpretable	Sales, prices, revenue
Logistic Regression	Predict binary outcomes	Numeric/Categorical	Probabilistic output	Churn, disease detection
Decision Trees	Classification & regression	Numeric/Categorical	Easy to interpret, handles missing values	Fraud, loan approval
Support Vector Machines	Classification & regression	Numeric	Effective in high dimensions	Text, images, bioinformatics
Neural Networks	Complex non-linear prediction			

2.8. Fitting a Model

Fitting a model involves training a statistical or machine learning model on data to **learn relationships between variables**. This often involves splitting the dataset into training and test sets.

Example: Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

X = [[0], [1], [2], [3]]
y = ['no', 'no', 'yes', 'yes']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5)
model = DecisionTreeClassifier().fit(X_train, y_train)
print("Predicted:", model.predict(X_test))
```

2.9. Topic Two Summary

Understanding variables, populations, samples, and modeling techniques is foundational for **data analytics**. These concepts enable analysts to summarize, describe, and predict data patterns effectively, forming the basis for decision-making.

2.10 End of Topic Two Exercises: Variables, Population, Samples, and Statistical Modelling

2.10.1. Conceptual Questions

1. Define a variable in Python and explain why variable naming conventions are important. Provide two examples. (4 marks)

2. Differentiate between population and sample, giving one real-world example for each. (4 marks)

3. List and briefly explain three common summary measures used to describe a dataset. (3 marks)

4. Explain the difference between nominal and ordinal categorical data with examples. (4 marks)

5. Define statistical modeling and describe its four main purposes. (6 marks)

2.10.2. Analytical Questions

6. A dataset contains the monthly sales figures of a company for one year. Identify two potential data quality issues that might arise and suggest methods to address them. (4 marks)

7. Explain the difference between descriptive, predictive, and inferential statistical modeling, providing one example use case for each. (6 marks)

8. Compare linear regression and logistic regression in terms of:
   a) Dependent variable type
   b) Typical applications
   c) Python libraries used
   (6 marks)

2.10.3. Practical / Applied Questions

9. Write Python code to:
   a) Create a list of ages: [18, 20, 22, 25, 30]
   b) Calculate and print the mean, median, and standard deviation.
   c) Add a new age 28 and recalculate the statistics. (6 marks)

10. Using Pandas, create a DataFrame with columns Name, Gender, Score for 5 students.
    a) Display the frequency of each gender.
    b) Find the highest and lowest scores.
    c) Sort the DataFrame by Score in descending order. (6 marks)

11. Given a population of integers from 1 to 100, write Python code to select a random sample of 10 elements without replacement. (4 marks)

2.10.4. Scenario-Based Questions

12. A bank wants to predict whether a customer will default on a loan (Yes/No) based on their credit score and income.
    a) Identify an appropriate predictive modeling technique.
    b) Name the dependent and independent variables.
    c) Mention one Python library that can be used to implement the model. (5 marks)

13. A retail store wants to analyze historical sales data to forecast demand for the next month.
    a) Identify which type of model is most suitable.
    b) Give two example methods or algorithms used for this type of modeling.
    c) Explain why modeling is helpful for the store. (6 marks)

2.10.5. Mini-Project / Practical Exercise

14. You are provided with a dataset of house prices with columns: Size, Bedrooms, Price.
    a) Perform basic descriptive statistics: mean, median, and standard deviation for Price.
    b) Fit a linear regression model to predict Price using Size as the predictor.

c) Predict the price for a house of size 1500 sq.ft.
d) Visualize the relationship between Size and Price using a scatter plot. (15 marks)

15. Using the following dataset of transactions (items bought by customers):

Milk	Bread	Butter
1	1	0
1	0	1
0	1	1
1	1	1

a) Identify frequent itemsets with minimum support of 0.5.

b) Generate association rules with minimum confidence of 0.7.

c) Explain one insight that could be useful for marketing decisions. (10 marks)

2.11 Topic Two References

- Freedman, D., Pisani, R., & Purves, R. (2007). *Statistics*. W. W. Norton & Company.
- Hair, J., Black, W., Babin, B., & Anderson, R. (2018). *Multivariate Data Analysis*. Cengage Learning.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning*. Springer.
- McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.
- Montgomery, D., Peck, E., & Vining, G. (2012). *Introduction to Linear Regression Analysis*. Wiley.
- Ross, S. (2014). *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2019). *Introduction to Data Mining*. Pearson.

TOPIC THREE:

EXPLORATORY DATA ANALYSIS (EDA)

3.1. Philosophy of Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical phase in the **data science workflow** focused on **understanding, visualizing, and summarizing data** before formal modeling.

Key Principles:

- Introduced by **John Tukey** (1977), EDA emphasizes **flexibility and data-driven discovery** rather than strict adherence to formal statistical models.
- Encourages analysts to **explore data visually and statistically**, uncover patterns, detect anomalies, and test assumptions.
- Focuses on **data understanding first**, helping analysts generate hypotheses and decide on appropriate modeling approaches.

Key Objectives of EDA:

1. Identify data structure and quality issues.
2. Detect outliers and anomalies.
3. Understand distributions and variable relationships.
4. Formulate hypotheses for further analysis or modeling.

3.2. Basic EDA Tools

EDA combines **summary statistics** and **visualization** to explore datasets.

Common Tools:

- **Summary Statistics:** Mean, median, variance, standard deviation, range.
- **Visualizations:**
  - **Histograms:** Frequency distribution of numerical data.
  - **Boxplots:** Highlight median, quartiles, spread, and outliers.
  - **Scatter Plots:** Show relationships between two continuous variables.
  - **Correlation Matrices & Heatmaps:** Measure linear relationships between variables.

These tools guide **decision-making, hypothesis generation**, and **data preprocessing**.

3.3. Summary Statistics

Summary statistics provide **numerical summaries** to describe **central tendency** and **variability**.

Python Example:

```
import pandas as pd


# Sample dataset
```

```
data = pd.DataFrame({
    'Age': [25, 30, 45, 35, 22],
    'Score': [88, 92, 95, 70, 85]
})

# Summary statistics
print(data.describe())
```

Key Measures in describe() output:

- count: Number of observations
- mean: Average value
- std: Standard deviation
- min/max: Minimum and maximum
- 25%, 50%, 75%: Quartiles

3.4. Histograms

Histograms visualize **frequency distribution** of numerical variables.

Python Example:

```
import matplotlib.pyplot as plt

data['Age'].hist(bins=5, color='skyblue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

Insights from Histograms:

- Detect **skewness** (left or right)
- Identify **modality** (unimodal, bimodal)
- Spot potential **outliers**

3.5. Boxplots

Boxplots summarize data through **quartiles**, showing **median, spread, and outliers**.

Python Example:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(y=data['Score'], color='lightgreen')
plt.title('Score Boxplot')
```

plt.show()

Insights from Boxplots:

- Identify **median and quartiles**
- Detect **outliers**
- Compare distributions across groups

3.6. Scatter Plots

Scatter plots reveal **relationships between two continuous variables**.

Python Example:

plt.scatter(data['Age'], data['Score'], color='purple')

plt.title('Age vs Score')

plt.xlabel('Age')

plt.ylabel('Score')

plt.show()

Insights:

- Identify **linear or non-linear relationships**
- Detect **clusters or patterns**
- Spot potential **outliers**

3.7. Correlation Matrix

Correlation matrices measure **pairwise linear relationships** between variables.

- Values range from **-1 (perfect negative)** to **+1 (perfect positive)**.
- Often visualized as **heatmaps**.

Python Example:

corr_matrix = data.corr()

sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

Insights:

- Understand **which variables are strongly related**
- Identify potential **multicollinearity** issues

3.8 Topic Three Summary

EDA is the **foundation of any data analytics process**, combining statistical summaries and visual exploration to **understand data deeply**, detect anomalies, and guide model selection.

- Philosophy emphasizes **flexibility and discovery**.
- Basic EDA tools allow analysts to **identify trends, patterns, and relationships** before applying complex models.

3.9 End of Topic Three Exercises: Exploratory Data Analysis (EDA)

Notes for Students:

- Use Python libraries: Pandas, Matplotlib, and Seaborn.
- Provide code and outputs for all practical exercises.
- Interpret visualizations carefully and relate findings to the context of the data.
- Mini-project emphasizes combining statistical summaries and visual exploration to support data understanding.

3.9.1. Conceptual Questions

1. Explain the philosophy of Exploratory Data Analysis (EDA) and why it is considered important in the data science workflow. (4 marks)
2. List three key objectives of EDA and give a practical example for each. (6 marks)
3. Differentiate between summary statistics and visualizations in EDA, providing two examples of each. (4 marks)
4. Describe the main insights that can be obtained from a boxplot. (3 marks)
5. Explain the significance of a correlation matrix and how it can influence further modeling. (3 marks)

3.9.2. Analytical Questions

6. A dataset of 100 students contains their Age and Test Scores. Explain how histograms and scatter plots could help identify:
   a) Skewness in the distribution of ages
   b) Relationships between age and scores
   (4 marks)
7. You notice several extreme values in your data. Which EDA tools would you use to detect these outliers and why? (3 marks)
8. Explain the difference between univariate, bivariate, and multivariate EDA with examples. (4 marks)

3.9.3. Practical / Applied Questions

9. Using Python and Pandas, create a DataFrame for the following data and perform summary statistics:

Age	Score
20	85
22	90
19	78
24	92
21	88

   a) Display count, mean, standard deviation, min, and max.
   b) Identify the 25th, 50th, and 75th percentiles. (6 marks)

10. Plot the following using Python:
    a) A histogram of Age with 4 bins.
    b) A boxplot of Score.

c) A scatter plot of Age vs Score.
d) Provide one insight from each plot. (8 marks)

11. Using the same dataset, compute the correlation matrix and visualize it using a heatmap. Interpret the correlation value between Age and Score. (5 marks)

3.9.4. Scenario-Based Questions

12. A marketing team wants to understand the purchasing behavior of customers. They have a dataset with Customer Age, Annual Income, and Monthly Expenditure.
a) Suggest two EDA techniques they could use to explore the relationships between these variables.
b) Explain what insights they might gain from each technique. (6 marks)

13. A healthcare analyst has patient data including Blood Pressure, Cholesterol, and BMI. Describe how EDA could help detect anomalies or errors in this dataset before predictive modeling. (4 marks)

3.9.5. Mini-Project / Practical Exercise

14. You are provided with a CSV file student_data.csv containing the columns Age, Score, Hours_Studied. Perform the following in Python:
a) Load the data into a Pandas DataFrame.
b) Display summary statistics for all numerical columns.
c) Plot histograms for Age and Hours_Studied.
d) Create a scatter plot of Hours_Studied vs Score.
e) Generate a correlation heatmap for all numerical variables.
f) Write a short paragraph interpreting the patterns, trends, or anomalies you observe. (15 marks)

3.10 Topic Three References

- Behrens, J. T. (1997). *Principles and Procedures of Exploratory Data Analysis*. Psychological Methods, 2(2), 131–160.
- Freedman, D., Pisani, R., & Purves, R. (2007). *Statistics*. W. W. Norton & Company.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning*. Springer.
- McKinney, W. (2017). *Python for Data Analysis*. O'Reilly Media.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- Wickham, H., & Grolemund, G. (2016). *R for Data Science*. O'Reilly Media.

TOPIC FOUR:

DATA VISUALIZATION AND DECISION MAKING

4.1. Basic Principles of Data Visualization

Effective data visualization transforms data into **meaningful insights** by presenting it in a clear and interpretable format.

Core Principles:

1. **Clarity:** Avoid clutter; visuals should be easy to understand.
2. **Simplicity:** Use straightforward charts and limit unnecessary elements.
3. **Accuracy:** Represent data truthfully; avoid misleading scales or distortions.
4. **Relevance:** Display only information that supports decision-making.
5. **Emphasis:** Use color, size, or annotations to highlight key insights without distraction.

4.2. Popular Data Visualization Tools

4.2.1 Matplotlib

- Foundational Python library for **static, animated, and interactive plots**.
- Provides **fine control** over plot elements.

Example:

```
import matplotlib.pyplot as plt


plt.plot([1, 2, 3, 4], [10, 20, 25, 30])
plt.title('Matplotlib Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

4.2.2 Seaborn

- Built on Matplotlib, provides a **high-level interface** for statistical graphics.
- Simplifies creation of attractive visualizations with **minimal code**.

Example:

```
import seaborn as sns
import matplotlib.pyplot as plt


tips = sns.load_dataset('tips')
sns.scatterplot(data=tips, x='total_bill', y='tip', hue='day')
```

```
plt.title('Seaborn Scatter Plot')
plt.show()
```

4.2.3 Tableau

- Commercial, **interactive dashboard tool**.
- Drag-and-drop interface, supports multiple data sources.
- Can integrate Python via **TabPy** for advanced analytics.

4.2.4 Power BI

- Microsoft's **business analytics service** for interactive dashboards.
- Supports Python integration for advanced visualizations.

4.2.5 Plotly

- Python library for **interactive, web-based graphs**.
- Features **zoom, hover, and click interactivity**.

Example:

```
import plotly.express as px

df = px.data.iris()
fig = px.scatter(df, x='sepal_width', y='sepal_length', color='species', title='Plotly Interactive Scatter Plot')
fig.show()
```

4.3. Case Studies of Inspiring Projects

- **Gapminder:** Animated bubble charts visualizing global development trends.
- **COVID-19 Dashboards:** Interactive global tracking of pandemic spread.
- **Takeaways:** Storytelling combined with visualization makes complex data **accessible and actionable**.

4.4. Dashboards and Storytelling

Dashboards

- Visual displays consolidating **key metrics and trends**.
- Designed for **quick comprehension** and **decision-making**.
- Components: Charts, gauges, tables, filters, KPIs.

Storytelling

- Combines visuals and narrative to explain **context and implications**.
- Guides the audience through data insights, making them **memorable and actionable**.

4.5. Creating a Simple Dashboard in Python

- Use **Streamlit** for interactive dashboards with minimal code.

Example Code:

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = pd.DataFrame({
    'Month': ['Jan', 'Feb', 'Mar', 'Apr'],
    'Sales': [150, 200, 250, 300]
})

# Dashboard layout
st.title("Simple Sales Dashboard")
st.write("Monthly Sales Overview")

# Bar chart
fig, ax = plt.subplots()
ax.bar(data['Month'], data['Sales'], color='skyblue')
plt.ylabel("Sales")
st.pyplot(fig)

# Filter
month = st.selectbox("Select a month", data['Month'])
filtered_sales = data[data['Month'] == month]['Sales'].values[0]
st.write(f"Sales in {month}: {filtered_sales}")
```

To run: streamlit run dashboard.py

4.6. Decision Making

- **Definition:** Selecting the best course of action among alternatives.
- **Data-driven decision making:** Guided by insights from data analysis, visualization, and predictive modeling.
- **Benefits:** Reduces uncertainty, minimizes risks, improves outcomes, enhances accountability.

4.7. How Dashboards Aid Decision Making

1. **Centralized view:** Consolidates key performance metrics.
2. **Real-time monitoring:** Allows quick detection of trends and anomalies.

3. **Simplifies complex data:** Reduces cognitive load via charts and tables.
4. **Customizable:** Tailored views for different stakeholders.
5. **Supports action:** Highlights areas needing attention, guiding prioritization.

4.8. Topic Four Summary

Effective data visualization **combines clear principles, proper tools, and storytelling**. By leveraging Python libraries (Matplotlib, Seaborn, Plotly), commercial tools (Tableau, Power BI), and dashboards, analysts can transform raw data into **insightful visuals** that guide decision-making and strategic action.

4.9 End-of-Topic Exercises: Data Visualization and Decision Making

Notes for Students:

- Use Python libraries: Matplotlib, Seaborn, Plotly, and Streamlit for practical exercises.
- Include code, plots, and short interpretations for all exercises.
- Focus on clarity, accuracy, and relevance when creating visualizations.
- Scenario-based exercises assess your ability to apply visualization principles for decision-making.

4.9.1. Conceptual Questions

1. Define data visualization and explain its importance in decision-making. (3 marks)
2. List and explain any four basic principles of effective data visualization. Provide an example for each principle. (8 marks)
3. Differentiate between dashboards and storytelling in data visualization. (4 marks)
4. Explain how data visualization reduces cognitive load for decision-makers. (3 marks)

4.9.2. Tool-Based Questions

5. Compare Matplotlib, Seaborn, and Plotly in terms of functionality and interactivity. (5 marks)
6. Describe one advantage of using Tableau or Power BI over Python libraries for data visualization. (2 marks)
7. Explain the role of Streamlit in creating dashboards with Python. Include one practical benefit. (3 marks)

4.9.3. Practical / Applied Exercises

8. Using Python and Matplotlib, create a bar chart for the following sales data:

Month	Sales
Jan	150
Feb	200
Mar	250
Apr	300

a) Label the axes and give the chart a title.
b) Highlight the month with the highest sales using color or annotation. (5 marks)

9. Using Seaborn, plot a scatter plot for the tips dataset showing total_bill vs tip colored by day.
a) Interpret the pattern or relationship between the variables.

b) Identify any outliers visible in the plot. (5 marks)

10. Using Plotly, create an interactive scatter plot for the iris dataset showing sepal_length vs sepal_width and colored by species.
    a) Explain how interactivity can aid in exploring the dataset. (4 marks)

4.9.4. Scenario-Based Questions

11. A company wants to monitor monthly sales, customer complaints, and website visits on a dashboard.
    a) Suggest three types of visualizations that would be appropriate for this dashboard.
    b) Explain how the dashboard can support management decision-making. (6 marks)

12. During a product launch, the marketing team wants to identify which regions are performing best in sales and which are underperforming.
    a) Recommend a data visualization technique to display this information effectively.
    b) Explain why your chosen visualization helps in prioritizing decisions. (4 marks)

4.9.5. Mini-Project / Practical Exercise

13. You are provided with a CSV file company_data.csv containing Month, Revenue, Expenses, Profit.
    a) Load the data into a Pandas DataFrame.
    b) Create a simple Streamlit dashboard showing:
        - A bar chart of Revenue and Expenses by month
        - Display Profit for a selected month using a filter
    c) Write a short paragraph interpreting trends, highlighting any months with unusually high or low performance. (15 marks)

4.10 Topic Four References

- Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media.
- Few, S. (2012). *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press.
- Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90–95.
- Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.
- Microsoft. (2022). *Power BI Documentation*. Microsoft Docs.
- Plotly Technologies. (2023). *Plotly Python Open Source Graphing Library*.
- Rosling, H., Rosling, O., & Rönnlund, A. R. (2018). *Factfulness: Ten Reasons We're Wrong About the World*. Flatiron Books.
- Streamlit Documentation. (2023). *Streamlit: The fastest way to build data apps*. Streamlit.
- Tableau Software. (2023). *Tableau Desktop User Guide*. Tableau.
- Waskom, M. (2021). *Seaborn: Statistical Data Visualization*. Journal of Open Source Software, 6(60), 3021.
- Yau, N. (2013). *Data Points: Visualization That Means Something*. Wiley.
- Yigitbasioglu, O. M., & Velcu, O. (2012). *A review of dashboards in performance management: Implications for design and research*. International Journal of Accounting Information Systems, 13(1), 41–59

PREDICTIVE AND PRESCRIPTIVE ANALYTICS

5.1. Overview

5.1.1 Predictive Analytics (PA)

Predictive analytics is the process of using **historical and current data** combined with **statistical models and machine learning techniques** to forecast future events or outcomes. It helps organizations anticipate what is likely to happen based on past patterns, trends, and relationships.

Key Features:

1. **Data-Driven Forecasting:** Uses structured and unstructured data to identify patterns that can inform future outcomes.
2. **Statistical Modeling:** Involves regression analysis, time series analysis, clustering, and other statistical techniques to understand relationships between variables.
3. **Machine Learning:** Employs algorithms like decision trees, support vector machines, neural networks, and ensemble methods to predict outcomes from complex data.
4. **Decision Support:** Provides organizations with predictive insights that can guide strategic and operational decisions.

Questions Answered by Predictive Analytics:

- What is likely to happen in the future?
- Which customers are at risk of churn?
- What products are likely to be in demand next quarter?
- Which machines are likely to fail soon?

Applications:

- **Retail:** Forecasting sales and inventory requirements.
- **Finance:** Credit scoring, fraud detection, and stock price forecasting.
- **Healthcare:** Predicting disease outbreaks or patient readmission risk.
- **Manufacturing:** Predictive maintenance to prevent equipment failure.

5.1.1.1 Examples of Predictive Analytics in Python

5.1.1.1.1 Predicting house prices based on size using linear regression:

```
from sklearn.linear_model import LinearRegression
import numpy as np


# Sample data: house sizes (sq ft) and prices ($1000)
X = np.array([[600], [800], [1000], [1200], [1400]])  # Features
```

```
y = np.array([150, 200, 250, 300, 350])            # Target

# Create and train the model
model = LinearRegression()
model.fit(X, y)

# Predict price for a new house
new_size = np.array([[1100]])
predicted_price = model.predict(new_size)
print(f"Predicted price for a house of size {new_size[0][0]} sq ft is ${predicted_price[0]*1000:.2f}")
```

Insights:

- Model **learns from historical data**.
- Forecasts outcomes for **new, unseen inputs**.

5.1.1.1.2 Forecasting customer purchase amounts based on historical data:

```
from sklearn.linear_model import LinearRegression
import numpy as np

# Historical purchase data (in $)
X = np.array([[1], [2], [3], [4], [5]])  # Month
y = np.array([100, 150, 200, 250, 300]) # Purchase amount

# Create and train the model
model = LinearRegression()
model.fit(X, y)

# Predict next month's purchase
next_month = np.array([[6]])
predicted_amount = model.predict(next_month)
print(f"Predicted purchase amount for month 6: ${predicted_amount[0]:.2f}")
```

5.1.2 Prescriptive Analytics (PSA)

Prescriptive analytics builds on predictive analytics by not only forecasting future outcomes but also **recommending the best actions** to achieve specific business objectives. It helps organizations decide **what should be done** to optimize performance.

Key Features:

1. **Optimization Algorithms:** Linear programming, integer programming, and other mathematical optimization techniques to maximize or minimize a target variable (e.g., profit, cost, efficiency).

2. **Business Rules:** Integrates organizational constraints and policies into decision-making.
3. **Simulation and Scenario Analysis:** Tests different courses of action to identify the most effective strategy.
4. **Action-Oriented:** Focuses on recommendations and interventions, not just predictions.

Questions Answered by Prescriptive Analytics:

- What actions should we take to achieve a specific goal?
- How many products should we produce to maximize profit?
- Which marketing strategy will minimize churn?
- How should we allocate resources to optimize outcomes?

Applications:

- **Supply Chain:** Optimizing inventory levels and delivery routes.
- **Healthcare:** Recommending treatment plans based on predicted patient outcomes.
- **Finance:** Portfolio optimization and risk mitigation strategies.
- **E-commerce:** Personalized promotions to increase retention and sales.

5.1.2.1 Example of Prescriptive Analytics in Python

5.1.2.1.1 Optimizing product production to maximize profit:

```
import pulp
# Define the optimization problem
problem = pulp.LpProblem("Maximize_Profit", pulp.LpMaximize)

# Decision variables: units of product A and B
x = pulp.LpVariable('Product_A', lowBound=0, cat='Continuous')
y = pulp.LpVariable('Product_B', lowBound=0, cat='Continuous')

# Objective function: maximize profit
problem += 20*x + 30*y, "Total_Profit"

# Constraints: resource limitations
problem += 2*x + y <= 100
problem += x + 3*y <= 90

# Solve
problem.solve()

print(f"Produce {x.varValue} units of Product A")
print(f"Produce {y.varValue} units of Product B")
```

```
print(f"Maximum Profit: ${pulp.value(problem.objective)}")
```

5.1.2.2 Optimizing decisions using linear programming with PuLP:

```
import pulp

# Define the problem: Maximize profit
problem = pulp.LpProblem("Maximize_Profit", pulp.LpMaximize)

# Decision variables
x = pulp.LpVariable('Product_A', lowBound=0, cat='Continuous')
y = pulp.LpVariable('Product_B', lowBound=0, cat='Continuous')

# Objective function
problem += 20*x + 30*y, "Total_Profit"

# Constraints
problem += 2*x + y <= 100  # Resource 1
problem += x + 3*y <= 90   # Resource 2

# Solve
problem.solve()

print(f"Status: {pulp.LpStatus[problem.status]}")
print(f"Produce {x.varValue} units of Product A")
print(f"Produce {y.varValue} units of Product B")
print(f"Maximum Profit: ${pulp.value(problem.objective)}")
```

Insights:

- Prescribes **optimal production quantities** to maximize profit.
- Ensures **resource constraints** are respected.

5.1.3 Use Case Example: E-commerce

Scenario:

- **Predictive Analytics:** A retail platform forecasts which customers are likely to **churn** in the next quarter based on browsing history, purchase frequency, and engagement metrics.
- **Prescriptive Analytics:** Based on predicted churn, the system recommends **personalized retention strategies**, such as special discounts, loyalty rewards, or targeted emails, to retain customers and reduce churn.

Outcome:

- Improved customer retention
- Optimized marketing spend
- Increased revenue

5.1.4 Key Differences Between PA and PSA

Feature	Predictive Analytics	Prescriptive Analytics
Goal	Forecast future outcomes	Recommend optimal actions
Focus	What will happen?	What should we do?
Techniques	Regression, ML, time series	Optimization, simulation, business rules
Output	Predictions or probabilities	Actionable recommendations
Example	Predict customer churn	Recommend targeted retention campaigns

5.2. Connecting Predictive Analytics to Business Objectives

Predictive analytics models are most valuable when their outputs directly support an organization's **strategic objectives**. A model in isolation may produce forecasts, but unless these forecasts are tied to actionable business goals, their utility is limited. Linking predictive models to business objectives ensures **decision relevance**, **ROI**, and alignment with corporate strategy.

Key Points:

- **Alignment with Goals:** Models must address measurable outcomes that the business values, such as revenue growth, cost reduction, customer retention, or operational efficiency.
- **Actionable Insights:** Predictions should guide **tactical or strategic decisions**, not just report past behavior.
- **Resource Allocation:** Predicted trends inform where to allocate **budget, manpower, or inventory** for maximum impact.

  Example:

- A **telecom company** uses predictive analytics to forecast which customers are likely to churn in the next quarter. By identifying these customers early, marketing and customer service teams can **target retention campaigns** effectively, aiming to reduce churn by **15%**.

5.3. Gathering Critical Business Insights

Predictive analytics enables the extraction of **patterns, anomalies, and relationships** that are not immediately visible in raw data. These insights are critical for **understanding drivers of performance**, customer behavior, and operational bottlenecks.

Key Points:

- **Pattern Recognition:** Identify trends such as peak usage times, sales cycles, or seasonal demand.
- **Anomaly Detection:** Spot unusual events, fraud, or operational inefficiencies.
- **Relationship Analysis:** Understand dependencies between variables (e.g., pricing, promotions, and sales volume).

Use Case:

- A **ride-sharing platform** analyzes data on **peak hours, driver availability, and traffic patterns**. Insights guide dynamic pricing strategies, optimize driver dispatching, and **reduce customer wait times**, improving service quality and profitability.

5.4. Making a Business Decision

Data-driven decisions leverage both predictive and prescriptive analytics to **replace intuition with evidence**. This improves **accuracy, reduces risk**, and ensures resource efficiency.

Key Points:

- **Evidence-Based Decisions:** Decisions are informed by insights, forecasts, and recommended actions.
- **Application Areas:** Can involve **market entry, pricing strategies, inventory management, resource optimization, or risk mitigation**.

Use Case:

- A **healthcare provider** uses predictive models to **identify high-risk patients** for chronic diseases. Prescriptive analytics schedules **preventive appointments**, reducing emergency visits, improving patient outcomes, and lowering costs.

5.5. Identifying Market Trends

Analytics helps organizations **anticipate market shifts**, customer preferences, and emerging opportunities. Predictive tools forecast trends, while prescriptive tools suggest actionable responses.

Key Points:

- **Trend Detection:** Recognize seasonal patterns, changes in consumer behavior, or emerging market segments.
- **Actionable Recommendations:** Prescriptive analytics can guide inventory planning, marketing campaigns, or product launches based on predicted trends.

Use Case:

- A **fashion retailer** uses predictive analytics to forecast **upcoming styles**. Prescriptive insights recommend adjustments in **inventory levels and marketing campaigns** to align with predicted demand, reducing excess stock and missed sales opportunities.

5.6. Approaches to a Business Challenge

Analytics provides **structured frameworks** for solving complex business challenges, enabling organizations to test scenarios, uncover root causes, and plan effective interventions.

Key Points:

- **Scenario Modeling:** Explore “what-if” situations to assess outcomes under different conditions.
- **Root Cause Analysis:** Identify the factors driving performance issues.
- **Simulation:** Model complex systems to anticipate outcomes before implementing changes.

Use Case:

- A **manufacturing company** applies predictive analytics to detect **potential equipment failures**. Prescriptive analytics schedules **preventive maintenance** to minimize downtime and optimize production efficiency.

5.7. Forecasting Future Demand

Demand forecasting combines predictive models with prescriptive optimization to **plan inventory, staffing, and marketing strategies** efficiently.

Key Points:

- **Predictive Forecasting:** Uses historical sales, seasonality, and external factors to anticipate demand.
- **Prescriptive Optimization:** Suggests **how to act** on the forecast, such as reorder quantities, staff scheduling, and promotional planning.

Use Case:

- A **supermarket chain** predicts demand for **perishable goods** using predictive analytics. Prescriptive analytics recommends **optimal stock levels**, reducing waste and ensuring availability to maximize revenue.

5.8. Topic Five Summary

- **Predictive Analytics:** Foresees future outcomes.
- **Prescriptive Analytics:** Suggests the best course of action.
- Together, they **enhance decision-making, reduce risk, and optimize business outcomes**.

5.9 End-of-Topic Exercises: Predictive and Prescriptive Analytics

Notes for Students:

- Provide Python code, outputs, and interpretations where applicable.
- Use linear regression and PuLP for practical exercises.
- Focus on explaining how predictions inform actionable decisions.
- Scenario-based questions assess your ability to connect analytics to real business problems.

5.9.1. Conceptual Questions

1. Define predictive analytics and prescriptive analytics. Highlight one key difference between them. (4 marks)
2. Explain the importance of aligning predictive analytics models with business objectives. Provide one practical example. (4 marks)
3. List three common techniques used in predictive analytics and briefly describe their use. (6 marks)
4. Explain the role of optimization in prescriptive analytics. Include one example where this is applied in business. (4 marks)

5.9.2. Tool and Python-Based Questions

5. Consider the following historical sales data for a retail store:

Month	Sales ($)
Jan	1200
Feb	1500
Mar	1800
Apr	1600
May	2000

a) Using Python and linear regression, predict sales for June. (5 marks)
b) Explain the assumption made about the relationship between Month and Sales in this prediction. (2 marks)

6. A factory wants to optimize production of two products, A and B, to maximize profit. Resources are limited. Explain how you would use prescriptive analytics (linear programming) to determine optimal production quantities. (5 marks)

7. Using the Python PuLP library, write the objective function for maximizing total profit given:

   - Profit per unit: Product A = $25, Product B = $40
   - Resource constraints: $3A + 2B \leq 120$ and $4A + 3B \leq 150$
     (3 marks)

5.9.3. Scenario-Based Questions

8. A telecom company predicts that 20% of customers are likely to churn next quarter.
   a) Explain one predictive analytics method to identify these customers. (2 marks)
   b) Suggest two prescriptive analytics interventions to reduce churn. (4 marks)

9. A supermarket wants to forecast demand for perishable goods for the next month.
   a) Which type of analytics would you use to forecast demand? (1 mark)
   b) Once demand is forecasted, how would prescriptive analytics help in inventory planning? (3 marks)

10. A ride-sharing platform wants to reduce passenger wait times during peak hours.
    a) Describe how predictive analytics can help identify high-demand periods. (3 marks)
    b) Suggest one prescriptive analytics solution to optimize driver allocation. (3 marks)

5.9.4. Mini-Practical / Applied Exercise

11. Using Python:

- You are provided with historical customer purchase data:

Month	Purchase Amount ($)
1	100
2	150
3	200
4	250
5	300

Tasks:
a) Fit a linear regression model to predict the purchase amount for month 6. (4 marks)
b) Interpret the predicted value in a business context. (2 marks)
c) Suggest one prescriptive action based on the predicted purchase amount to increase customer spending. (2 marks)

5.9.5. Business Insight and Decision-Making Questions

12. Explain how predictive and prescriptive analytics together enhance decision-making in the following contexts:
    a) Healthcare (e.g., patient risk and treatment planning) (4 marks)
    b) Retail (e.g., product demand forecasting and inventory optimization) (4 marks)

13. A fashion retailer uses predictive analytics to forecast trends and prescriptive analytics to plan inventory and marketing. Describe a step-by-step approach for applying these analytics to:
    a) Minimize excess stock
    b) Maximize sales revenue (6 marks)

5.10. Topic Five References

- Dantzig, G. B., & Thapa, M. N. (2006). *Linear Programming 1: Introduction*. Springer.
- Davenport, T. H., & Harris, J. G. (2007). *Competing on Analytics*. Harvard Business Review Press.
- Davenport, T. H., & Harris, J. G. (2007). *Competing on Analytics: The New Science of Winning*. Harvard Business Review Press.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
- Linoff, G. S., & Berry, M. J. A. (2011). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Wiley.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business*. O'Reilly Media.
- Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2020). *Data Mining for Business Analytics*. Wiley.
- Shmueli, G., Bruce, P., Gedeck, P., & Patel, N. (2020). *Data Mining for Business Analytics*. Wiley.
- Shmueli, G., Patel, N., & Bruce, P. (2017). *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*. Wiley.
- Waller, M. A., & Fawcett, S. E. (2013). *Data Science, Predictive Analytics, and Big Data: A Revolution that will Transform Supply Chain Design and Management*. Journal of Business Logistics, 34(2), 77–84.

Implemented by: