Project Objectives

Each group will:

1. **Acquire, clean, and wrangle a real-world dataset (Task for 1 group Member)**
    a. Research and present 5 challenges for the team to choose from
    b. Provide Dataset for each of the challenges mentioned above for review (Kaggle, Datacamp, etc.,)
    c. Import the data into your environment.
    d. Save an untouched copy for reference.D

2. **Clean the Data (Data Cleaning) - Task for 1 group Member**

Activities to correct or remove dirty, inconsistent, or incomplete data.

Task	**Description**
Remove duplicates	Eliminate exact or near-identical rows
Handle missing values	Drop, impute (mean/median), or flag missing data
Correct data types	Convert to appropriate types (e.g., date, numeric, category)
Fix typos/inconsistencies	E.g., "N/A" vs "na", "male" vs "Male"
Strip extra spaces/characters	Clean string fields (e.g., .strip(), .replace())
Normalize data formats	Dates, currency, percentages, etc.
Filter invalid entries	Remove or fix rows that don't make sense (e.g., age = 900)

3. **Wrangle the Data (Data Structuring & Enrichment)**

Activities to reshape and organize the data into a format suitable for analysis or modelling.

Task	**Description**
Reshape data	Pivot, melt, unstack, or stack data tables
Merge/join datasets	Combine related tables using keys (e.g., merge, join)

Create new variables	Feature engineering (e.g., age from birthdate, revenue per user)
Encode categorical variables	Convert categories to numeric (e.g., one-hot, label encoding)
Parse complex fields	Break apart nested fields (e.g., splitting full names)
Group/aggregate data	Create summaries by category, date, region, etc.
Handle outliers	Remove or treat extreme values
Save cleaned version	Export to new file (e.g., cleaned_data.csv) for analysis

4. Perform comprehensive Exploratory Data Analysis (EDA)

(Task for 2 group Member)

4.1. Understand the Data Context

- Read the data dictionary or documentation (if available)
- Know the source, purpose, and structure of the dataset

4.2. Load and Inspect the Data

- Load the data into a tool (e.g. Python/pandas, R, Excel)
- Use .head(), .info(), and .describe() to get a quick overview
- Check data types and column names

4.3. Handle Missing Data

- Identify missing values (NaN, empty strings, etc.)
- Visualize missing data patterns (e.g., with heatmaps)
- Decide on a strategy: drop, impute (mean, median, etc.), or flag

4.4. Univariate Analysis (Single Variable)

- Analyse distribution of individual variables
    - Numerical: histograms, box plots, summary stats (mean, median, std)
    - Categorical: bar charts, value counts
- Detect outliers

4.5. Bivariate & Multivariate Analysis

- **Numerical vs Numerical**: scatter plots, correlation matrix
- **Categorical vs Numerical**: box plots, groupby aggregations
- **Categorical vs Categorical**: stacked bar charts, contingency tables

4.6. Identify and Handle Outliers

- Use IQR, Z-score, or visual methods to find extreme values
- Decide whether to cap, transform, or remove them

4.7. Check Data Quality and Consistency

- Duplicate records
- Inconsistent formats (e.g., "Male" vs "male")
- Unexpected or invalid values

4.8. Examine Relationships and Trends

- Visual exploration: pairplots, correlation heatmaps
- Time-based trends (if applicable)
- Grouping and summarizing to detect patterns

4.9. Document Findings and Hypotheses

- Take notes or prepare visuals to share observations
- List potential variables of interest for modeling
- Highlight anomalies or risks in the dataset

5.0 Build models (predictive, prescriptive, classification, or time series as relevant)
Task for 2 group Member

5.1. Define the Problem and Goal

- Identify whether it's a **classification**, **regression**, **clustering**, or other task.
- Define your **target variable** and **success criteria** (e.g., accuracy, RMSE).

5.2. Select Features

- Choose relevant features based on EDA and business context.
- Optionally perform **feature selection**:
    - Remove irrelevant or redundant features.

- Use techniques like correlation analysis, mutual information, or feature importance.

5.3. Split the Data

- Split into **training**, **validation**, and **test** sets (commonly 70/15/15 or 80/20).
- Use train_test_split() from sklearn.model_selection in Python.

5.4. Choose Model(s)

- Select appropriate algorithms based on the problem:
  - **Classification**: Logistic Regression, Random Forest, XGBoost, SVM
  - **Regression**: Linear Regression, Ridge/Lasso, Decision Trees
  - **Clustering**: K-Means, DBSCAN, Hierarchical Clustering

5.5. Train the Model

- Fit the model to the training data.
- Use .fit() methods or equivalent.

6. Evaluate the Model - Task for 2 group Member

- Predict on the validation/test set.
- Use appropriate evaluation metrics:
  - **Classification**: Accuracy, Precision, Recall, F1-score, ROC-AUC
  - **Regression**: MAE, RMSE, R^2
  - **Clustering**: Silhouette Score, Davies-Bouldin index

7. Tune Hyperparameters - Task for 2 group Member

- Use techniques like:
  - **Grid Search** (GridSearchCV)
  - **Randomized Search**

    - **Bayesian Optimization**
- Cross-validation helps avoid overfitting.

8. Interpret the Model - Task for 2 group Member

- Analyze feature importance, decision boundaries, and coefficients.
- Use tools like SHAP, LIME, or model-specific .feature_importances_.

8.1. Validate and Test - Task for 1 group Member

- Test final model on unseen data.
- Check for **generalization performance**.

9. Communicate findings via visualizations and compelling narratives

Task for 2 group Member

9.1. Understand Your Audience

- Identify who you're presenting to:
    - Technical (data scientists, engineers): prefer detailed insights.
    - Non-technical (execs, stakeholders): want business impact and clarity.
- Tailor the language, visuals, and level of detail accordingly.

9.**2. Define the Key Message**

- Ask: What do I want the audience to remember?
- Focus on 1–3 main insights supported by data.
- Avoid overwhelming the audience with everything you discovered.

9.3. Select the Right Visualizations

Data Type or Insight	**Recommended Visual**
Distribution	Histogram, Boxplot

Comparisons	Bar chart, Column chart
Trends over time	Line chart, Area chart
Relationships	Scatter plot, Bubble chart
Part-to-whole	Pie chart (with caution), Donut chart, Stacked bar
Categories vs metrics	Heatmap, Grouped bar
Geospatial insights	Maps, Choropleths

Use libraries like matplotlib, seaborn, plotly, Power BI, or Tableau.

9.4. Clean and Simplify Visuals

- Avoid clutter: minimal text, clean fonts, proper spacing.
- Label axes, titles, and values clearly.
- Use colour consistently and with purpose (highlight key points).
- Use annotations to emphasize important points.

9. 5. Craft a Narrative (Data Storytelling)

- Follow a story structure:
    - Context: What's the problem or question?
    - Data: What did we analyse?
    - Insight: What did we find?
    - Impact: Why does it matter?
    - Action: What should we do next?
- Use headlines and annotations to guide the viewer through the story.

9.6. Combine Visuals and Text

- Don’t rely on charts alone. Use short, meaningful captions or bullets.
- Combine visuals with compelling commentary in reports, slides, or dashboards.

9.7. Create a Final Output

- Could be:
    - A slide deck (PowerPoint, Google Slides)
    - A dashboard (Power BI, Tableau, Excel)
    - A written report (Word, PDF, Jupyter Notebook)

    - An interactive story (e.g., Shiny app, Streamlit, web app)

9.8. Review and Polish

- Double-check all figures, labels, and interpretations.
- Ask a colleague or test on a non-expert for clarity.

• Demonstrate teamwork, accountability, and technical growth

• Sub/it a well-documented GitHub repository with reproducible code