Capstone Project: Exploratory Data Analysis on Titanic Dataset

Objective:

The aim of this project is to perform data analysis on the Titanic dataset, which contains information about passengers aboard the Titanic and whether they survived. Through this project, students will:

- Learn how to clean and preprocess data.
- Perform exploratory data analysis (EDA) using **Pandas**, **NumPy**, **Seaborn**, and **Matplotlib**.
- Draw insights about the factors that may have influenced the survival of passengers.

Dataset Description:

The dataset used for this project will contain various features related to customers on titanic ship.

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)

1st = Upper

2nd = Middle

3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...

Sibling = brother, sister, stepbrother, stepsister

Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...

Parent = mother, father

Child = daughter, son, stepdaughter, stepson

Some children travelled only with a nanny, therefore parch=0 for them.

Project Steps and Detailed Instructions:

1. Dataset Loading and Basic Inspection

- **Task**: Load the dataset using Pandas and inspect the basic structure.
- **Instructions**:
  - Load the CSV file using pd.read_csv().
  - Inspect the dataset using df.head(), df.info(), and df.describe().
  - Understand the types of features available (e.g., categorical vs. numerical).

2. Data Cleaning

- **Task**: Clean the dataset by handling missing values and dealing with incorrect or inconsistent data.
- **Instructions**:
  - Identify missing values using isnull().sum().
  - For features like "Age" and "Fare", use imputation strategies (e.g., fill missing ages using the median age per passenger class).
  - Handle missing data in the "Cabin" and "Embarked" columns (e.g., drop or fill with appropriate values).

    - Convert categorical columns (e.g., "Sex", "Embarked") into numerical form using techniques like label encoding or pd.get_dummies().

3. Exploratory Data Analysis (EDA)

- **Task**: Perform exploratory data analysis using visualizations and statistical analysis.
- **Instructions**:
    - Use **Seaborn** and **Matplotlib** to create insightful visualizations:
        - **Bar plots**: Visualize survival rate across different classes (Pclass), gender (Sex), etc.
        - **Histograms**: Plot distributions for continuous features like "Age", "Fare".
        - **Box plots**: Show the spread and outliers in features like "Age", "Fare", grouped by survival status.
        - **Heatmaps**: Display a correlation matrix using sns.heatmap() to understand relationships between numerical features.
        - **Pair plots**: Use sns.pairplot() to explore pairwise relationships between features.

4. Feature Engineering

- **Task**: Create new features to enrich the dataset.
- **Instructions**:
    - **Family Size**: Combine "SibSp" (siblings/spouses aboard) and "Parch" (parents/children aboard) to create a new feature FamilySize = SibSp + Parch + 1.
    - **IsAlone**: Create a feature IsAlone, which is 1 if FamilySize == 1 and 0 otherwise.
    - **Age Group**: Group passengers into age buckets using pd.cut() (e.g., Child, Teen, Adult, Senior).
    - **Fare Category**: Create bins for fare prices using pd.qcut() to categorize fares.

5. Data Analysis and Insights

- **Task**: Analyze the relationships between the features and the target variable (Survival).
- **Instructions**:
    - **Survival Rate by Gender**: Compare the survival rates of male and female passengers.
    - **Survival Rate by Passenger Class**: Explore the relationship between class (Pclass) and survival.
    - **Age and Survival**: Investigate how age impacted survival rates using visualizations like a box plot or a swarm plot.
    - **Fare and Survival**: Analyze the relationship between fare prices and survival, and visualize this with a scatter plot or box plot.

6. Statistical Insights

- **Task**: Provide statistical insights and hypothesis testing, if applicable.
- **Instructions**:
    - Use **NumPy** to calculate statistical metrics (e.g., mean, median, standard deviation) for features like "Age" and "Fare".

    - Perform simple hypothesis testing (e.g., Did the passenger class significantly affect survival?).
    - Calculate correlations between features and survival to determine which features have a stronger influence.

7. Conclusion and Reporting

- **Task**: Summarize findings and present insights.
- **Instructions**:
    - Summarize the key takeaways from the analysis (e.g., "Women and children in first-class had the highest survival rate").
    - Suggest what the analysis reveals about the survival patterns and how this can be used to draw actionable insights.
    - Provide visualizations as part of the summary report for clarity.

Expected Outcomes:

- Students will gain hands-on experience in working with real-world data.
- They will be able to clean, preprocess, and analyze datasets using Python libraries like Pandas and NumPy.
- They will learn how to visualize data using Seaborn and Matplotlib, gaining insights into relationships between features.
- Through feature engineering, they will create new variables that can improve analysis or model performance.