🧠 titanic-data-science-pipeline – Report

Author: Ahmad
Date: May 1, 2025
Main Goal: To create a **modular, interactive, and reusable Exploratory Data Analysis (EDA) pipeline** that works not only for Titanic but for **any tabular dataset**.

🎯 Project Objective

The objective of this project is to **automate and standardize the EDA process** through a Python class-based pipeline. The pipeline is designed to allow data scientists and analysts to quickly:

- Understand the structure of any dataset.
- Handle missing values.
- Perform one-hot encoding.
- Normalize or scale features.
- Visualize data distributions and correlations.
- Remove outliers.
- Save the cleaned data for further use.

This system aims to **speed up EDA workflows**, especially in iterative or multiple-dataset environments.

🏗️ Core Component: `EDA` Class

Implemented in `eda.py`, the `EDA` class provides a comprehensive toolkit for:

- 📊 **Data Summary**: Viewing dataset shape, column names, types, and statistical summaries.

- 📈 **Visualization**:
    - Histograms for all features.
    - Correlation heatmaps (global or selective).
    - Histplot for variable comparison.
    - Boxplots for outlier detection.
- 🧹 **Data Cleaning**:
    - Handle missing values using various strategies.
    - Drop specified columns or duplicates.
    - Normalize or scale columns using MinMaxScaler.
    - Apply one-hot encoding to categorical features.
    - Remove outliers using the IQR method.
- 💾 **Export**:
    - Save cleaned datasets to CSV.

✅ Reusability Features

- Fully reusable by simply providing a different CSV path.
- Built-in CLI-like options make it dataset-agnostic.
- Can be expanded with new cleaning or visualization methods easily.

⚙️ How It Works

```
from eda import EDA

# Example usage on any dataset:
eda_tool = EDA("YourDataset.csv")
eda_tool.run()  # CLI-based method to explore and clean the dataset
```

🧪 Demonstration on Titanic Dataset

To test the EDA pipeline, the Titanic dataset was used as a case study. Key results:

- Missing values were handled successfully.
- Outliers in age/fare were detected and removed.
- Data was normalized.
- Dataset was encoded and saved to `cleaned_titanic.csv`.

Modeling was later applied using `model.py` to validate the usefulness of cleaned data.

📁 Project Files

File	Description
`eda.py`	Reusable EDA class with full CLI logic
`model.py`	Model training and evaluation script
`Titanic-Dataset.csv`	Example dataset for testing the pipeline
`cleaned_titanic.csv`	Output of the EDA pipeline

🌟 Achievements

- ✅ Created a highly **modular, extendable EDA class**.
- ✅ Pipeline works on **any dataset** without rewriting code.
- ✅ Reduced time for data inspection and preparation.
- ✅ Demonstrated pipeline usability on Titanic dataset.

📌 Future Enhancements

- Wrap the `EDA` class into a **Python package (PyPI)** for easier import and sharing.
- Add **logging** and **automated report generation** (PDF or HTML).
- Add support for **time series** or **text-based datasets**.
- Build a **GUI version** using Streamlit or Tkinter.