Machine Learning Capstone Project Manual

This project guides students through the full machine learning pipeline: data acquisition, preprocessing, modeling with three different algorithm types, evaluation, and professional documentation.

Part 1: Project Setup and Data Acquisition

1. Project Initialization

- **Repository:** Create a new public repository on GitHub named `ML-Capstone-Project`.
- **Structure:** Set up the following folder structure: ML-Capstone-Project/
  ├── data/
  ├── notebooks/
  ├── results/
  ├── README.md
  └── project_report.pdf`
- **Environment:** Create and activate a Python virtual environment. Install required libraries: `bash     pip install pandas numpy scikit-learn matplotlib seaborn jupyter`

2. Data Selection and Download

- **Data Source:** Choose a publicly available dataset from a reputable source (e.g., **Kaggle** or **UCI Machine Learning Repository**) that supports **all three tasks**:
  - **Recommendation:** Use the **California Housing Dataset** (for Regression/Clustering) or the **Heart Disease Prediction Dataset** (for Classification/Clustering) available on Kaggle.
- **Acquisition:** Download the raw data and save it in the `data/` folder.

Part 2: Exploratory Data Analysis (EDA) and Preprocessing

3. Data Loading and Initial Inspection

- Load the data into a Pandas DataFrame.
- Check the size, column data types, and identify the **Target Variable** for your Regression/Classification tasks.

4. Data Cleaning and Preparation

- **Handle Missing Values:** Identify and address any missing values (NaN) using appropriate strategies (e.g., mean/median imputation or dropping rows).

- **Feature Encoding:** Convert any categorical features (text data) into numerical format using techniques like **One-Hot Encoding** (for nominal data) or **Label Encoding** (for ordinal data).
- **Outlier Detection (Optional):** Use box plots to visualize outliers and decide whether to cap or remove them.

5. Scaling and Splitting

- **Feature Scaling:** Standardize or Normalize the numerical features using **StandardScaler** or **MinMaxScaler**. *This is crucial for distance-based algorithms like KNN and K-Means.*
- **Data Split:** Divide the data into training and testing sets (e.g., 80% train, 20% test) using `train_test_split`.

Part 3: Model Implementation and Evaluation

Implement and evaluate the following three models on the preprocessed data.

6. Regression Task: Linear Regression (Predicting a Continuous Value)

- **Target:** Select a continuous numerical target column (e.g., house price, patient's age).
- **Implementation:**
  1. Initialize and train the **Linear Regression** model.
  2. Make predictions on the test set.
- **Evaluation:** Calculate the following metrics:
  - **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted values.
  - **Root Mean Squared Error (RMSE):** The square root of MSE, interpretable in the target variable's units.
  - R^2 Score: Measures the proportion of the variance in the dependent variable that is predictable from the independent variables.

7. Classification Task: Logistic Regression or Decision Tree

- **Target:** Select a binary target column (e.g., 0/1 for presence/absence of disease, spam/not-spam).
- **Implementation:**
  1. Initialize and train the **Logistic Regression** or **Decision Tree** model.
  2. Make predictions on the test set.
- **Evaluation:** Calculate the following metrics:
  - **Accuracy:** Overall correct predictions.
  - **Confusion Matrix:** Generate the matrix to show **True Positives (TP)**, **False Positives (FP)**, **True Negatives (TN)**, and **False Negatives (FN)**.

- **Precision and Recall:** Discuss the trade-off, especially in the context of the chosen dataset (e.g., is minimizing False Negatives crucial?).

8. Clustering Task: K-Means Clustering

- **Target:** Use the entire dataset (or a subset of features) **without the label**.
- **Implementation:**
  1. Determine the optimal number of clusters (K) using the **Elbow Method** (plotting Inertia vs. K).
  2. Fit the **K-Means** model using the optimal K.
- **Evaluation:**
  - Calculate the **Silhouette Score** for the final clustering result.
  - Visualize the clusters by plotting two key features, color-coded by the assigned cluster label.

Part 4: Reporting and Submission

9. Final Report Generation

Create a comprehensive **Project Report** (`project_report.pdf`) that addresses the following sections:

I. Introduction

- **Project Goal:** Clearly state the objective (e.g., predict house price, classify disease).
- **Data Description:** Source, size, and features used.

II. Data Preprocessing & EDA

- Summary of cleaning steps (handling missing values, encoding).
- Key findings from EDA (e.g., correlations, distribution of the target variable).

III. Modeling and Results

- **Regression Results:** State the model used, final MSE/RMSE, and interpretation of the R^2 score.
- **Classification Results:** State the model used, final Accuracy, and detailed analysis of the **Confusion Matrix**.
- **Clustering Results:** Explain the chosen K (referencing the Elbow Plot), the final Silhouette Score, and the interpretation of the resulting clusters.

IV. Conclusion

- Summarize the best-performing model for each task.
- Discuss challenges encountered and potential next steps (e.g., trying Gradient Boosting, implementing PCA).

10. Submission Instructions :

- Ensure all code notebooks (`notebooks/`), the final report (`project_report.pdf`), and the clean data (`data/`) are committed.

- Write a detailed `README.md` that explains the project, how to run the code, and links to the final report.

- Google Colab Link: You must strictly include the link to your Google Colab notebook directly inside the PDF report. Ensure the link settings allow access for the grading team.

- **Final Step:** Submit the link to the public GitHub repository. ### **8. Resources**

- **Datasets**:

    - Kaggle

    - UCI ML Repository

    - Hugging Face Datasets

    - Google Dataset Search

---

9. Final Notes for Students

> “The goal is not to build the most complex model, but to solve a meaningful problem thoughtfully, ethically, and reproducibly.”
> — ML Teaching Team

Collaborate, document, iterate—and remember that **failure is data**. A well-analyzed “failed” project often teaches more than a lucky success.

Good luck, and happy modeling! 🤖 📊