1. Loading and Checking the Data

First, we loaded the dataset into a Pandas DataFrame. We checked the size of the dataset, the data types of each column, and looked at the first few rows. This helped us understand what kind of data we are working with and which columns are numerical or categorical.

This step is important because it gives a general idea about the dataset before doing any processing or modeling.

---

2. Handling Missing Values

After checking the data, we handled missing values.

- For numerical columns, we replaced missing values with the mean of each column.
- For categorical columns, we replaced missing values with the most common value (mode).

We did this to make sure there are no empty values in the dataset, because machine learning models cannot work properly with missing data.

---

3. Encoding Categorical Data

Since machine learning models only work with numbers, we converted all categorical columns into numerical form using One-Hot Encoding.

This step created new columns for each category and allowed the models to understand the categorical information correctly.

---

4. Checking Outliers

We used box plots to visually check if there are extreme values in numerical columns.

The goal of this step was only to understand the data better. We did not remove any outliers to avoid losing important information.

---

5. Splitting the Data

We split the dataset into two parts: - 80% for training - 20% for testing

This allows us to train the models on one part of the data and test their performance on unseen data.

---

6. Feature Scaling

We applied feature scaling using StandardScaler.

Only the input features were scaled, while the target columns were kept unchanged. This step helps improve the performance of models like Logistic Regression and K-Means, which are sensitive to feature size.

III. Models Implementation and Results

Data Preprocessing and Exploratory Data Analysis (EDA)

1. Data Loading and Initial Inspection

The dataset was loaded into a Pandas DataFrame. Initial inspection steps included: - Checking dataset dimensions using `df.shape` - Reviewing data types and missing values using `df.info()` - Displaying sample records using `df.head()`

This step was essential to identify numerical versus categorical variables and to confirm that the dataset was suitable for regression, classification, and clustering tasks.

---

2. Handling Missing Values

Real-world datasets often contain missing or incomplete data, which can negatively affect model performance if not handled correctly.

- **Numerical Features**: Missing values in numerical columns were replaced using **mean imputation**. This approach preserves the overall distribution and minimizes distortion of the data.
- **Categorical Features**: Missing values in categorical columns were replaced with the **mode (most frequent value)**, ensuring consistency within each category.

After imputation, the dataset was verified to contain no remaining missing values.

---

3. Feature Encoding

Machine learning algorithms require numerical inputs. Therefore, categorical variables were transformed using **One-Hot Encoding**. This technique converts each category into a binary column, allowing the model to process categorical information without introducing unintended ordinal relationships.

This step increased the dimensionality of the dataset but ensured compatibility with all applied algorithms.

---

4. Outlier Detection and Visualization

Outliers can significantly impact model performance, particularly for distance-based algorithms.

To detect potential outliers, **box plots** were generated for all numerical features. These visualizations helped identify extreme values and understand feature distributions.

Outliers were not removed in this project to preserve data completeness and because no strong evidence suggested that these values were data errors.

---

5. Feature Scaling

Feature scaling was applied using **StandardScaler**, which standardizes features to have a mean of 0 and a standard deviation of 1.

Scaling was necessary because: - Logistic Regression is sensitive to feature magnitudes - K-Means relies on distance calculations

Target variables (`condition` and `age`) were not scaled to avoid data leakage and preserve interpretability.

III. Model Implementation and Evaluation

A. Classification – Logistic Regression

For the classification task, we used Logistic Regression to predict the `condition` column.

First, we separated the features from the target variable. Then, we split the data into training and testing sets. After that, we applied feature scaling to the input data.

We trained the Logistic Regression model using the training data and then used it to make predictions on the test data.

To evaluate the model, we calculated: - Accuracy to see how many predictions were correct - Confusion Matrix to understand correct and wrong predictions - Precision to measure how accurate the positive predictions are - Recall to measure how well the model finds positive cases

The results showed that the model was able to classify the condition reasonably well.

B. Clustering – K-Means

For clustering, we used K-Means to group similar data points without using any target labels.

We removed the `condition` column and scaled the remaining features. To choose the best number of clusters, we used the Elbow Method, which helped us decide that 3 clusters is a good choice.

After fitting the K-Means model, we assigned each data point to a cluster. We then evaluated the clustering using the Silhouette Score.

Finally, we visualized the clusters using a scatter plot, which helped us see how the data is grouped.

C. Regression – Linear Regression

For the regression task, we used Linear Regression to predict the `age` column.

We separated the features and the target, then split the data into training and testing sets. The model was trained using the training data and then tested on unseen data.

To evaluate the regression model, we calculated: - Mean Squared Error (MSE) - Root Mean Squared Error (RMSE) - R^2 score

These metrics helped us understand how close the predicted ages are to the real values.

IV. Conclusion

Summary of Findings

- Logistic Regression provided reliable classification performance
- K-Means Clustering successfully identified underlying data groupings
- Linear Regression demonstrated effective continuous prediction

Challenges Encountered

- Managing mixed data types
- Preventing data leakage during scaling
- Selecting an optimal number of clusters

Future Work

- Experiment with advanced models such as Random Forest or Gradient Boosting
- Apply dimensionality reduction techniques like PCA
- Perform hyperparameter tuning
- Investigate class imbalance handling techniques

V. Reproducibility and Documentation

All steps in this project were implemented in a structured and reproducible manner. Code organization, preprocessing logic, and evaluation methods follow standard machine learning best practices. This ensures that results can be replicated and extended in future work.

This project fulfills all requirements of the Machine Learning Capstone assignment and demonstrates a clear understanding of the complete machine learning pipeline.