**Body Fat Data Report**

**1. Introduction CLO1**

This project studies a dataset with body fat percentages and body measurements. The goal is to predict body fat percentage using python and analyse patterns created from visualisations and clustering in the data. This analysis is done using Python and includes regression, classification, and clustering models.

**2. Tools CLO1**

* Language: Python.
* Libraries: listed at the last page.
* Version Control: GitHub.

**3. Data CLO1**

**Dataset: bodyfat.csv**

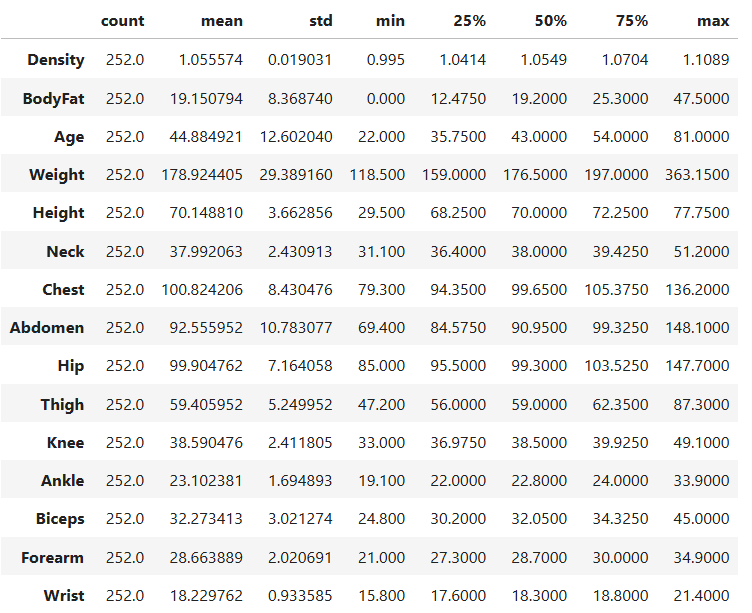
* **Records:** 252.
* **Variables:** 15 (Density, BodyFat, Age, Weight, Height, Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist).
* **Dependent Variable:** BodyFat.
* **Independent Variables:** (Density, Age, Weight, Height, Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist).
* **Purpose:** Predict body fat from body measurements.
* **Chosen Machine Learning Algorithms:** Linear Regression, Logistic Regression, KNN, Naive Bayes, Decision Tree.

**Top correlations with BodyFat:**

* **Abdomen:** 0.81
* **Chest:** 0.70
* **Hip:** 0.63
* **Weight:** 0.61

**4. Descriptive Analysis and Sampling (CLO2)**

Descriptive statistics such as mean, std, min, max, and quartiles were computed.



The count suggests complete data. The mean of the bodyfat which is 19% suggests the people in this dataset have a lower than average bodyfat percentage. There seems to be an error in the dataset itself because it states that the minimum bodyfat is 0 and this is impossible. The max bodyfat is generally considered healthy.

**Sampling Techniques:**

* **Random sampling:** 150 records were selected randomly. The average body fat is 19.08, minimum body fat is 0, which is impossibly low, could be an error or missing data. The median is19.9

A screenshot of a cell phone

AI-generated content may be incorrect.

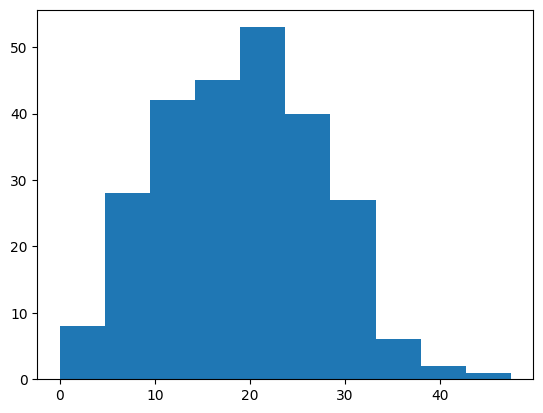
* **Systematic sampling:** Selected every kth element (every 1 element) because our dataset is small. This sample includes 150 people with an average body fat of about 19. The values range from 3.7 to 40.1, which is more realistic than the previous sample. Most people have body fat between 13 and 25 percent, with the median at 19.5.

A screenshot of a cell phone

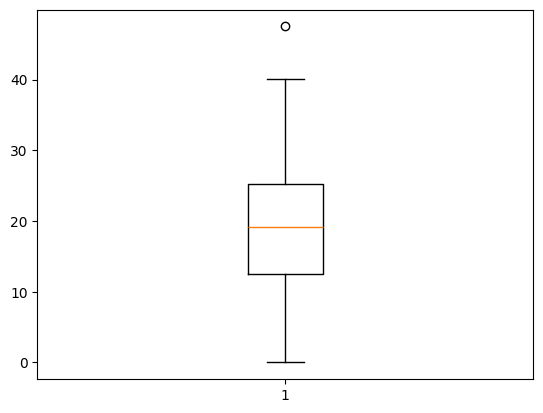
AI-generated content may be incorrect.

**Visualisation Techniques:**

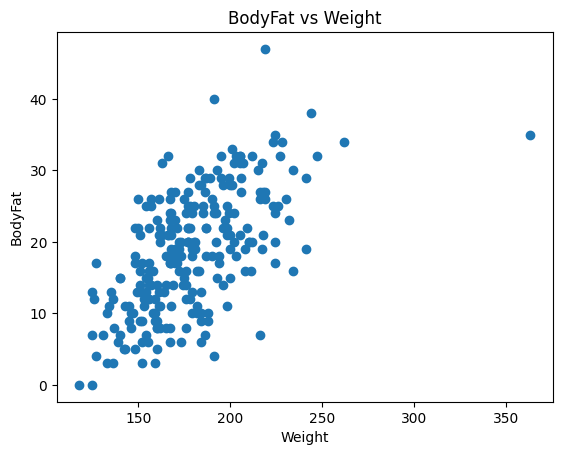
* Histogram, Box Plot, Scatter Plot, Heatmap



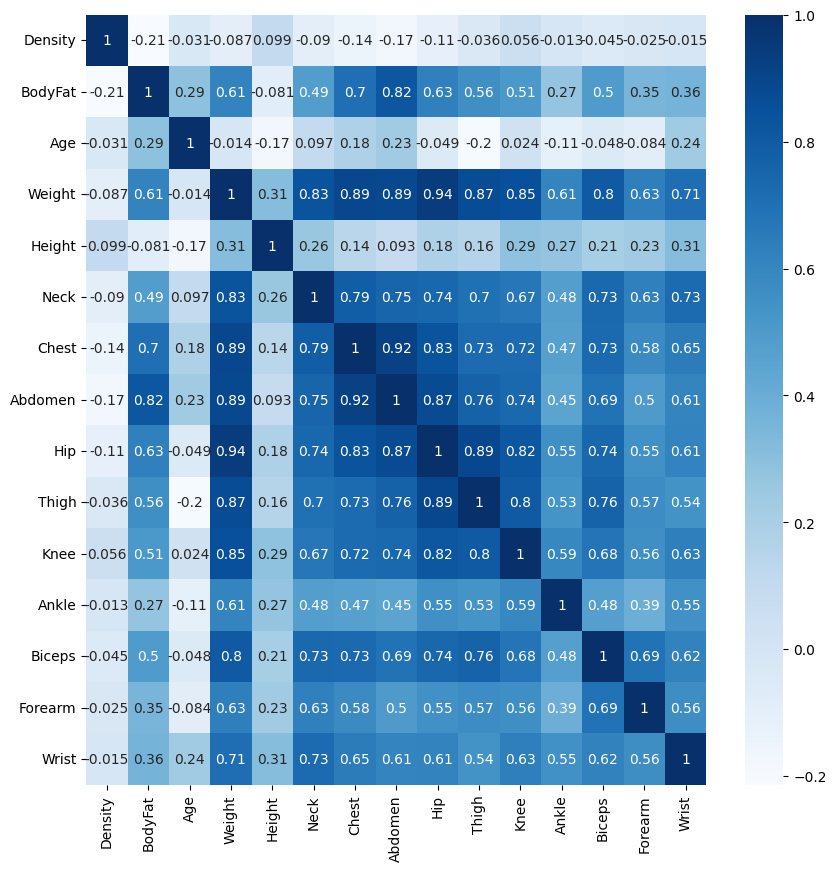
The histogram shows a normal distribution, with a slight tilt to the left. Most of the data is between 15% and 25% with some extreme cases.



The box plot shows most body fat values are between 13 and 25 percent. The average is around 19 percent. One very high value is an outlier above 45 percent. The data is mostly balanced.



We can see that as the weight increases, the bodyfat increases too, and there are some outlier points. This is a positive correlation.



- the map shows that Abdomen and Chest have a strong relationships with BodyFat.

**Hypothesis Testing:**

* **Pearson Correlation (Weight vs BodyFat):** 0.610
* **Spearman Correlation (Weight vs BodyFat):** 0.610
* **One-sample t-test:** This is a one-sample t-test checking if the average body fat in the data is significantly different from 25. The t-test statistic is -11.925 and the p-value is about 2.84e-26, which is much less than 0.05. Since the p-value is low, we reject the null hypothesis. This means the average body fat in the sample is significantly different from 25%.

**5. Regression Analysis (CLO3)**

**Model: Simple Linear Regression**

* **Independent:** age
* **Dependent:** BodyFat
* **r2 Score:** 0.105 (age alone isn’t enough to explain bodyfat, this score basically means how good my model is)

**Model: Multiple Linear Regression**

* **Independent:** knee, ankle, biceps
* **Dependent:** BodyFat
* **r2 Score:** 0.162 (score is low likely because these 3 variables don’t have a good correlation with bodyfat)

**6. Classification Models (CLO3)**

Converted BodyFat from decimal to integer. This was mainly done so the Logistic Regression would work.

**Model Accuracies:**

* **Logistic Regression:** 0.815 🡪 81.5%
* **KNN:** 0.736 🡪 73.6%
* **Naive Bayes:** 0.802 🡪 80.2%
* **Decision Tree:** 0.657 🡪 65.7%

**Model confusion matrix:**

* **Logistic Regression:**
  + **A black text on a white background

    AI-generated content may be incorrect.**
* **KNN:**
  + **A black and white text

    AI-generated content may be incorrect.**
* **Naïve Bayes:**
  + ****
* **Decision Tree:**
  + **A black and white text

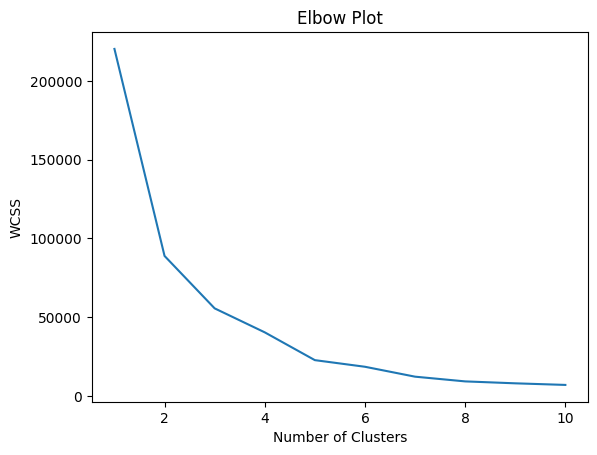
    AI-generated content may be incorrect.**

**Best Model:** Logistic Regression has the best accuracy.

**7. Clustering (CLO3)**

**Algorithm:** KMeans (on Bodyfat and Weight)

* **Number of Clusters:** 2 according to the elbow plot



* The red cluster represents people with higher weight and the blue is a lower weight.
* This diagram helps find and treat the heavier individuals so if someone is in the red cluster they give him more attention.

A diagram with red and blue dots

AI-generated content may be incorrect.

**8. Git & Version Control (CLO4)**

**Steps Taken:**

* Created GitHub repository.
* Uploaded project files.
* Invited the team members.
* Linked the github to the google colab.
* Evey time a change happens in the google colab it appears in GitHub and the word document files’ changes can be seen when someone overrides the word file.

GitHub ensured version control and collaboration between team members.

**9. Conclusion**

* Abdomen has the highest correlation with BodyFat.
* Multiple Linear Regression model performs better than simple linear (r2 = 0.162).
* Logistic Regression is the most accurate classification model (81.5%).
* KMeans clustering provided valuable insight of body types.
* GitHub was effectively used for version control and teamwork.

**10. References**

* **Dataset:** <https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset>
* **Python Libraries:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import pearsonr, spearmanr, chi2\_contingency, shapiro, ttest\_1samp

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score, recall\_score, precision\_score

from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.cluster import KMeans

from sklearn.naive\_bayes import GaussianNB