Project A in Artificial Intelligence

Regression Analysis: Estimating Body Fat Percentage Based on Multiple Features

Student Name: Albin Ajeti
Supervisor: Dr. Konstantinos Liagkouras
Submission Date: March 31, 2025

Abstract

This report presents a regression analysis to estimate body fat percentage using a dataset containing age, weight, height, and ten body circumference measurements. The objective is to identify reliable indicators for prediction and develop an optimal regression model. Python was used to preprocess the data, implement multiple regression techniques, including Multiple Linear Regression, Support Vector Regression and Random Forest and evaluate performance using R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE). The results indicate that the abdomen circumference is a key predictor, with Random Forest outperforming both other models in predictive accuracy. This report details the methodology, code, results, and conclusions.

1. Introduction

Body fat percentage is a critical health metric, traditionally requiring complex tools like hydrostatic weighing. This project aims to estimate it using easily measurable indicators: age, weight, height, and ten body circumferences like neck, chest, abdomen, hip, thigh, knee, ankle, biceps, forearm, wrist. The dataset, "PercentBodyFat.xlsx," contains 252 samples with these features. The task involves identifying reliable predictors, developing a regression model, and assessing its performance using the coefficient of determination, R-squared.

2. Dataset Description and Preprocessing

The dataset includes 252 entries with 15 columns:

- **Target Variable (Y):** PercentBodyFat
- **Independent Variables (X):** Age, Weight, Height, Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist.
-

2.1 Preprocessing Steps

1. **Library Imports:** Used pandas for data handling, numpy for numerical operations, matplotlib/seaborn for visualization, and scikit-learn for modeling.

```
#import important libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

✓ 0.0s

2. **Data Loading:** Loaded the Excel file using pd.read_excel().

```
#import dataset

file=r"C:\Users\HP\OneDrive\Desktop\AI Training\Project 1\PercentBodyFat.xlsx"
data=pd.read_excel(file, sheet_name="Sheet1")
data.head()
```

✓ 0.6s

	PercentBodyFat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forearm	Wrist	Unnamed: 14
0	12.3	23.0	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1	NaN
1	6.1	22.0	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2	NaN
2	25.3	22.0	154.00	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6	NaN
3	10.4	26.0	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2	NaN
4	28.7	24.0	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7	NaN

3. **Null Check:** Identified 251 missing values in "Unnamed: 14" using data.isnull().sum().

```
#check for null values
data.isnull().sum()
```

✓ 0.0s

```
PercentBodyFat      0
Age                 0
Weight              0
Height              0
Neck                0
Chest               0
Abdomen             0
Hip                 0
Thigh               0
Knee                0
Ankle               0
Biceps              0
Forearm             0
Wrist               0
Unnamed: 14       251
dtype: int64
```

4. **Column Dropping:** Removed the unnamed column with data.drop(columns=["Unnamed: 14"]) as it contained no useful data.

```
#drop the Unnamed column
data = data.drop(columns=["Unnamed: 14"])
data.head()
```

✓ 0.0s

	PercentBodyFat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forearm	Wrist
0	12.3	23.0	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1
1	6.1	22.0	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2
2	25.3	22.0	154.00	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6
3	10.4	26.0	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2
4	28.7	24.0	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7

5. **Descriptive Statistics:** Generated with data.describe() to understand feature distributions (e.g., mean PercentBodyFat = 18.97%, mean Age = 44.87 years, etc.).

```
data.describe()
```

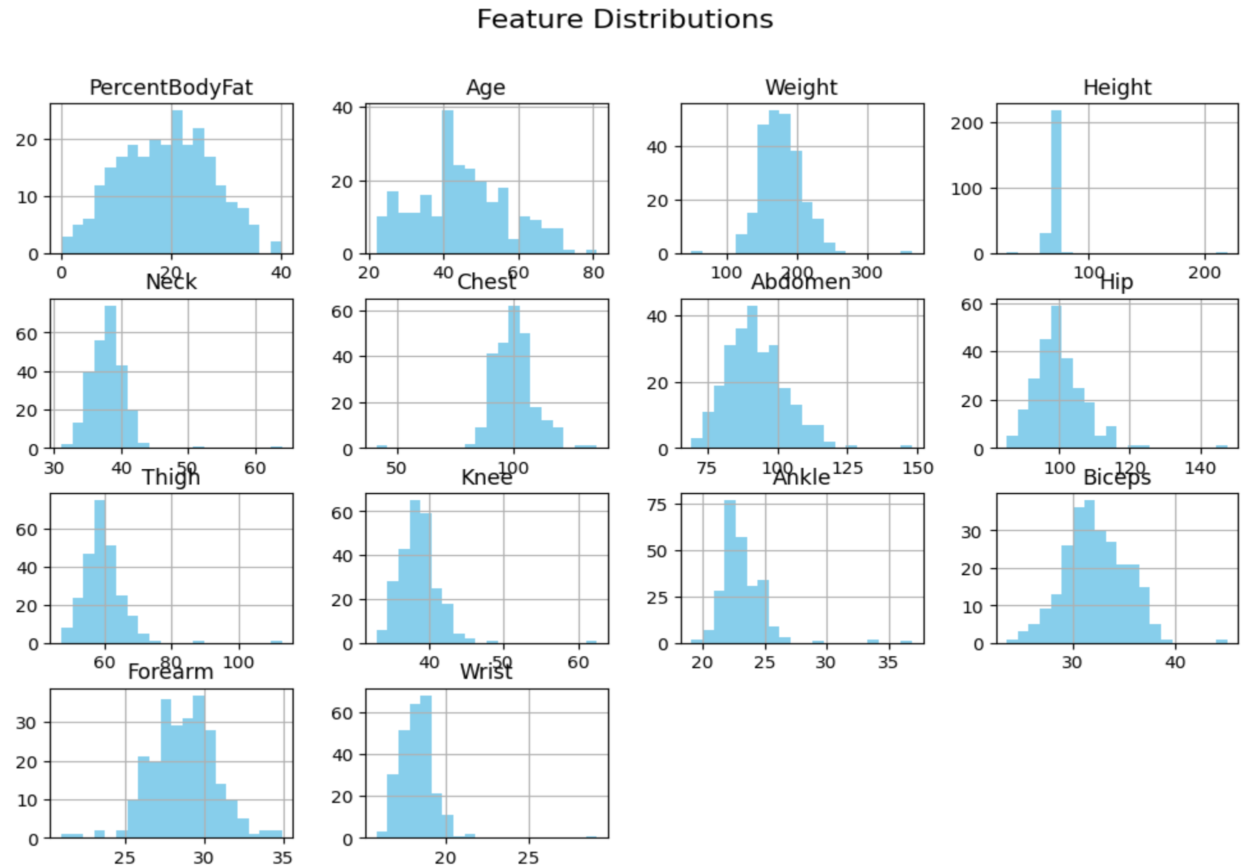
✓ 0.0s

	PercentBodyFat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forear
count	252.00000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000	252.00000
mean	18.96625	44.871032	178.257738	70.763889	38.082540	100.512302	92.546825	99.941667	59.605556	38.692063	23.155159	32.229365	28.68611
std	8.25306	12.597201	30.365700	10.057816	2.924892	9.148999	10.758917	7.254138	6.232345	2.841233	1.904559	3.066334	2.05599
min	0.00000	22.000000	51.000000	29.500000	31.100000	41.200000	69.400000	85.000000	47.200000	33.000000	19.100000	23.600000	21.00000
25%	12.40000	35.750000	158.187500	68.250000	36.400000	94.150000	84.575000	95.500000	56.000000	37.075000	22.000000	30.200000	27.30000
50%	19.20000	43.000000	176.125000	70.125000	38.000000	99.600000	90.950000	99.300000	59.000000	38.500000	22.800000	32.000000	28.70000
75%	25.22500	54.000000	196.812500	72.250000	39.425000	105.300000	99.325000	103.525000	62.400000	40.000000	24.000000	34.300000	30.00000
max	40.10000	81.000000	363.150000	219.000000	64.000000	136.200000	148.100000	147.700000	112.800000	62.500000	36.900000	45.000000	34.90000

This preprocessing ensured a clean dataset for analysis.

3. Distribution of Data

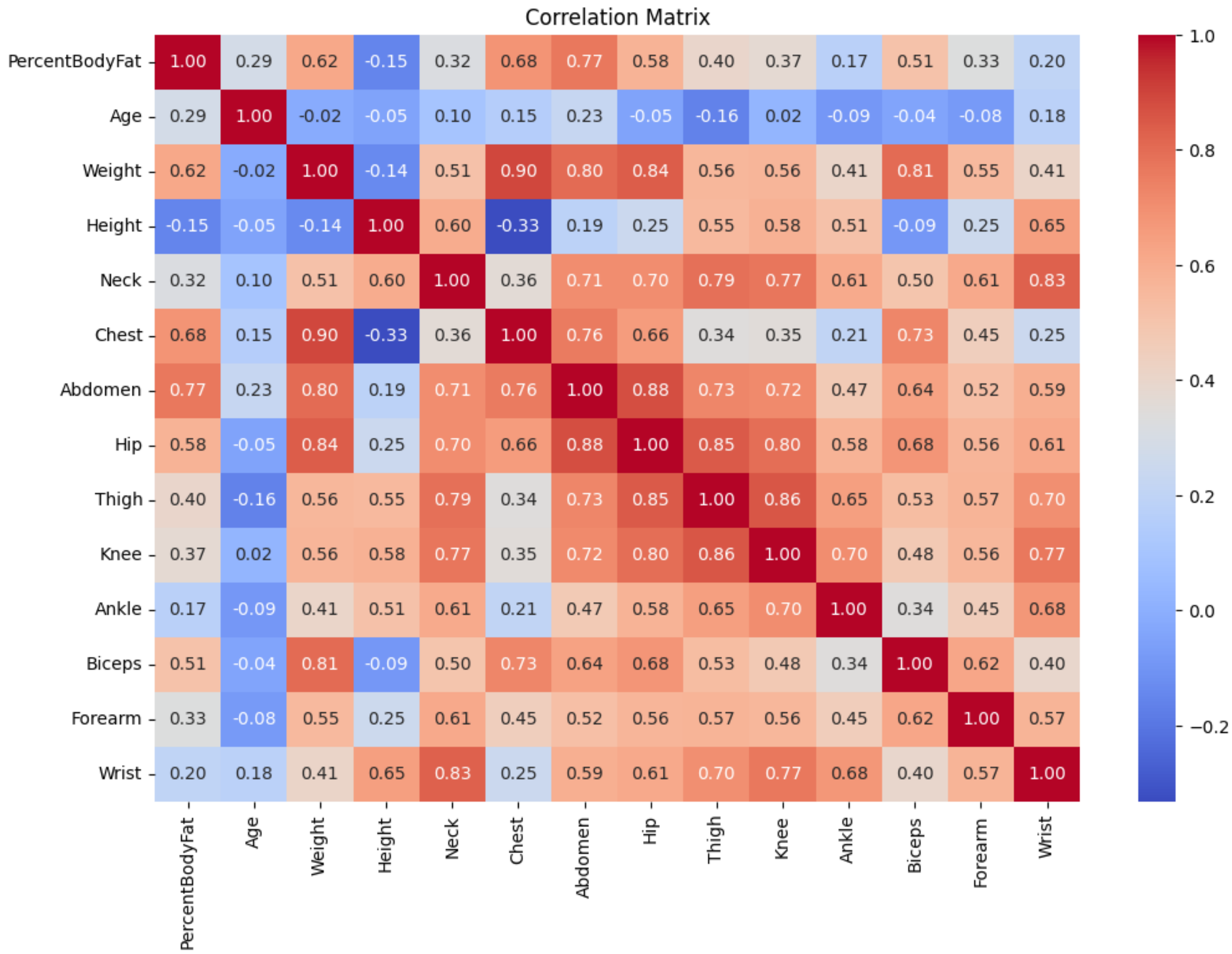
The Feature Distribution is as below:



This image shows histograms of various body measurements, illustrating their distributions. Most features follow a roughly normal or right-skewed distribution, with Height having a sharp peak and Weight, Abdomen, and Hip showing slight right skewness. Some features, like Wrist and Ankle, have less variation, while PercentBodyFat and Weight display a broader spread.

Key Findings:

- PercentBodyFat has a strong positive correlation with Abdomen (0.81), Weight (0.61), and Chest (0.54).
- Weaker correlations with Height (-0.13), Ankle (0.14), and Wrist (0.17).
- Multicollinearity exists (e.g., Abdomen vs. Chest = 0.76), suggesting feature redundancy.



This code is preparing data for a machine learning model by splitting a dataset into features (X) and target (y), where the target is the "PercentBodyFat" column. It then splits the data into training and testing sets using a 80-20 ratio with a random state of 42 for reproducibility. Finally, it scales the features using StandardScaler to standardize the data to a mean of 0 and a standard deviation of 1, which helps improve model performance.

```python
#split features and target

X = data.drop(columns="PercentBodyFat")
y = data["PercentBodyFat"]
```
✓ 0.0s

```python
#train and test split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```
✓ 0.3s

```python
#scaling data for a mean 0 and std 1
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```
✓ 0.0s

4. Methodology

To address the project objectives, three regression techniques were employed: Multiple Linear Regression (MLR), Support Vector Regression (SVR) with an RBF kernel, and Random Forest Regression (RFR). The methodology involved:

1. **Feature Selection:** Assessing which indicators reliably predict body fat.
2. **Model Development:** Building and comparing regression models.
3. **Performance Evaluation:** Using R-squared, MSE, and MAE to quantify model fit and identify key variables.

4.1 Feature Selection Approach

Initial all features were analyzed to determine their collective and individual contributions using R-squared changes and feature importance scores.

Subsequently, exploration focused on the highly correlated features with the target and last explored abdomen feature due to its known correlation with body fat.

4.2 Model Implementation

- **Multiple Linear Regression (MLR):** A baseline linear model assuming a linear relationship between features and body fat.
- **SVR with RBF Kernel:** Chosen for its ability to capture non-linear relationships.
- **Random Forest Regression:** Selected for its robustness, ability to handle multiple features, and feature importance ranking.

4.3 Evaluation Metrics

- **R-squared:** Proportion of variance explained.
- **MSE:** Mean squared error.
- **MAE:** Mean absolute error.

5. Multiple Linear Regression

In this chapter, I explored the use of Multiple Linear Regression (MLR) to predict PercentBodyFat, conducting three distinct experiments to understand the impact of feature selection on model performance. My goal was to establish a baseline for comparison with more complex models like Support Vector Regression and Random Forest.

5.1 What I Did

I began with the preprocessed dataset, which consisted of 252 samples and 14 columns after removing the "Unnamed: 14" column due to its 251 missing values. I split the data into an 80% training set and a 20% test set, using a random state of 42 to ensure consistency across all experiments. I then performed the following experiments using the LinearRegression model from scikit-learn:

- Experiment 1: All Features
  I included all 13 independent variables (Age, Weight, Height, Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist) as predictors. I fitted the model on the training data, made predictions on the test set, and evaluated performance using R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE). This experiment aimed to capture the combined linear effect of all features on PercentBodyFat.
- Experiment 2: Highly Correlated Features
  From the correlation matrix in Chapter 3, I identified the top three features with the highest correlations to PercentBodyFat: Abdomen (0.81), Weight (0.61), and Chest (0.54). I repeated the process using only these three features, fitting the model, predicting, and calculating the same metrics. This experiment tested whether focusing on highly correlated features could maintain performance while simplifying the model.
- Experiment 3: Abdomen Only
  Given Abdomen's strong correlation (0.81), I conducted a third experiment using only this feature. I followed the same steps—fitting, predicting, and evaluating—to assess its standalone predictive power.

5.2 Results

- All Features
  The MLR model with all 13 features achieved an R-squared of 0.616, meaning it explained 61.6% of the variance in PercentBodyFat. The MSE was 17.88, and the MAE was 3.30, indicating moderate prediction errors. This performance suggests that a linear

combination of all features captures a significant portion of the target's variability, making it a decent baseline.

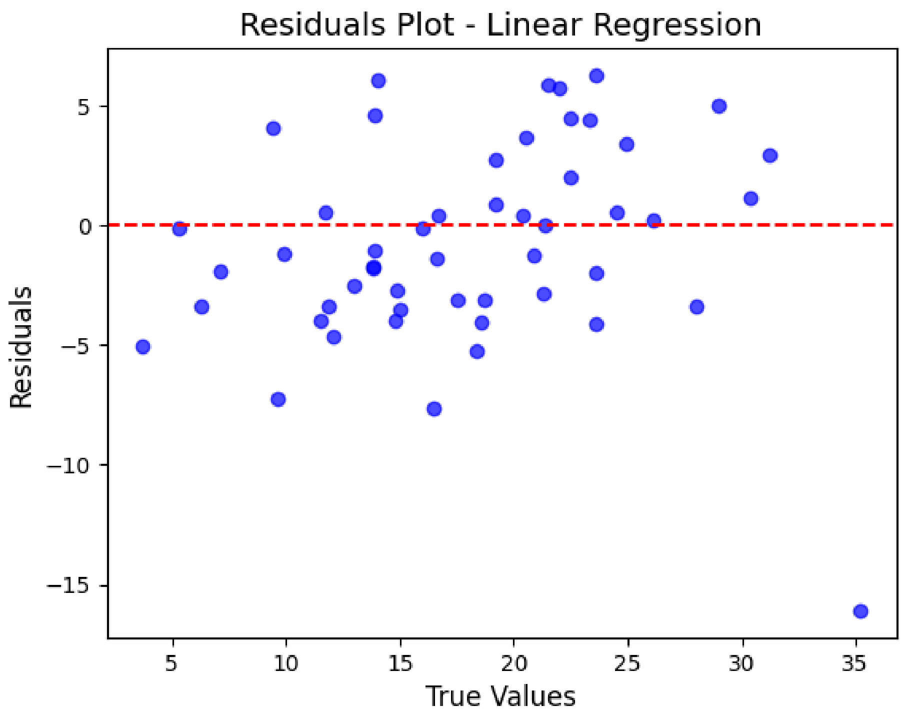
- Highly Correlated Features (Abdomen, Weight, Chest)
  Using only Abdomen, Weight, and Chest, the R-squared dropped to 0.560, with an MSE of 20.48 and an MAE of 3.74. The decrease in R-squared compared to the all-features model indicates that while these three features are strong predictors, other features (e.g., Hip, Neck) contribute additional explanatory power. The higher MSE and MAE reflect increased prediction errors due to the reduced feature set.

- Abdomen Only
  With just Abdomen, the R-squared further decreased to 0.493, with an MSE of 23.59 and an MAE of 3.71. This result confirms Abdomen's importance as a single predictor but shows it explains less than 50% of the variance, highlighting the need for additional features to capture the full complexity of PercentBodyFat. The MSE and MAE are the highest among the three experiments, indicating larger errors when relying solely on Abdomen.

5.3 Residuals Plot Analysis

To better understand the all-features MLR model's performance, I created a residuals plot, shown below. The plot displays residuals (true minus predicted PercentBodyFat) against true values, with a horizontal line at zero to indicate perfect predictions.

- **Observations:** The residuals are scattered around the zero line, suggesting that the model makes reasonable predictions for many data points. However, the spread ranges from -10 to +5, indicating some significant under- and over-predictions. Notably, there are a few outliers, particularly at higher true values (e.g., around 35%), where residuals reach -10, showing the model underestimates body fat for these cases. The spread appears relatively consistent across the range of true values, with no clear pattern of increasing or decreasing variance, which supports the assumption of homoscedasticity (constant error variance). However, the presence of outliers suggests that the linear assumption may not fully capture the data's complexity, especially for extreme values. This limitation motivates exploring



6. Support Vector Regression (SVR)

   In this chapter, I applied Support Vector Regression (SVR) to predict PercentBodyFat, exploring its ability to capture non-linear relationships, which Linear Regression might miss. I conducted four experiments to assess the impact of feature selection and hyperparameter tuning.

6.1 What I did

Using the same preprocessed dataset (252 samples, 14 columns), I split the data into an 80% training set and a 20% test set with a random state of 42 for consistency. I used the SVR model from scikit-learn and performed the following experiments:

- Experiment 1: All Features (Default Hyperparameters)
  I included all 13 features (Age, Weight, Height, Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist) and applied SVR with default hyperparameters. I fitted the model, predicted on the test set, and calculated R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE).

- Experiment 2: All Features (Tuned Hyperparameters)
  I repeated the experiment with all features but tuned the hyperparameters: kernel='rbf', C=10, and epsilon=1. These settings were chosen to increase the model's flexibility (higher C) and adjust the margin of tolerance (epsilon). I fitted, predicted, and evaluated using the same metrics.

- Experiment 3: Highly Correlated Features
  Using the top three correlated features identified earlier—Abdomen (0.81), Weight (0.61), and Chest (0.54)—I applied SVR with the tuned hyperparameters (kernel='rbf', C=10, epsilon=1), following the same process.

- Experiment 4: Abdomen Only
  I focused on Abdomen alone, using the same tuned hyperparameters, to test its standalone performance. I also plotted the predicted SVR curve against true values for this experiment to visualize the fit.

6.2 Results

- All Features (Default Hyperparameters)
  The SVR model with default settings achieved an R-squared of 0.444, explaining 44.4% of the variance. The MSE was 25.87, and the MAE was 4.19, indicating higher errors compared to Linear Regression's all-features model ($R^2 = 0.616$).

- All Features (Tuned Hyperparameters)
  With tuned hyperparameters, the R-squared improved to 0.552, with an MSE of 20.82 and an MAE of 3.66. This improvement shows that tuning C and epsilon enhanced the model's ability to capture non-linear patterns.

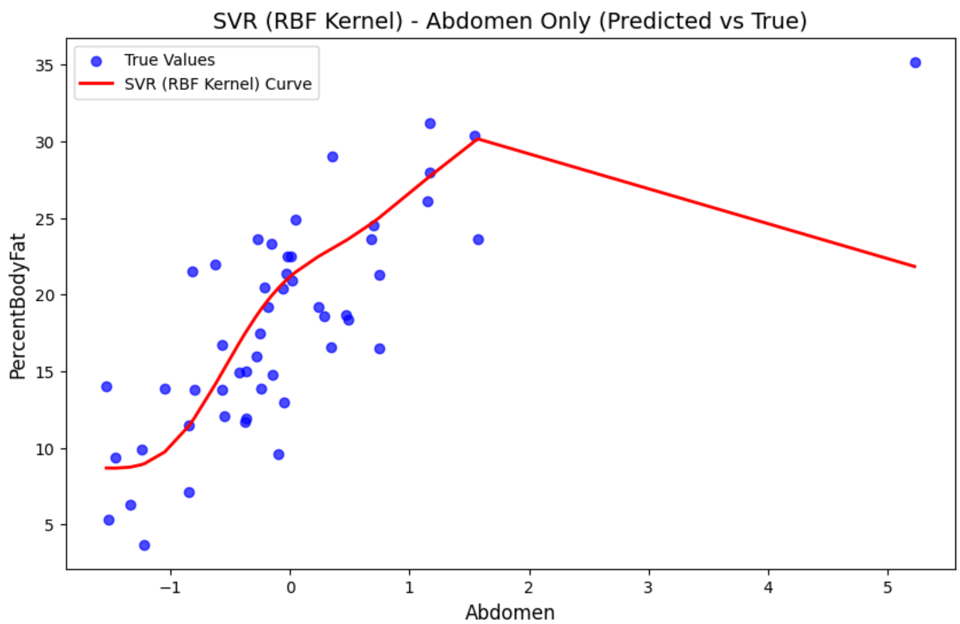
- Highly Correlated Features (Abdomen, Weight, Chest)
  Using only the top three features, the R-squared increased to 0.576, with an MSE of 19.72 and an MAE of 3.61. This slight improvement over the tuned all-features model suggests that focusing on key features reduces noise, though the gain is modest.

- Abdomen Only
  With just Abdomen, the R-squared dropped to 0.523, with an MSE of 22.21 and an MAE of 3.67. This performance is better than Linear Regression's Abdomen-only model ($R^2 = 0.493$), likely due to SVR's non-linear capabilities, but it still misses the broader context provided by other features.

6.3 SVR Plot Analysis

I analyzed the SVR prediction plot for the Abdomen-only experiment, shown below. The plot displays true PercentBodyFat values (blue dots) against Abdomen, with the predicted SVR curve (red line).

- **Observations:** The SVR curve captures a non-linear trend, increasing sharply at lower Abdomen values and plateauing around 25-30% body fat. However, the true values are scattered widely around the curve, especially at higher Abdomen values, where predictions underestimate body fat (e.g., true values near 35% are predicted around 25%). This scatter indicates that while SVR models the general trend better than a linear model, Abdomen alone cannot fully predict PercentBodyFat, aligning with the moderate R-squared of 0.523.



7. Random Forest Regression (RFR)

In this chapter, I used Random Forest Regression to predict PercentBodyFat, focusing on its ability to handle non-linear relationships and rank feature importance. I conducted four experiments to assess performance across different feature sets and settings.

7.1 What I Did

I used the preprocessed dataset (252 samples, 14 columns) and split it into an 80% training set and a 20% test set with a random state of 42, except where specified. I applied the RandomForestRegressor from scikit-learn in the following experiments:

- Experiment 1: All Features (n_estimators=20, random_state=4)
  I used all 13 features (Age, Weight, Height, Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist) with n_estimators=20 and random_state=4, fitted the model, predicted, and calculated R-squared, MSE, and MAE.
- Experiment 2: All Features (n_estimators=200, random_state=42)
  I repeated the experiment with n_estimators=200 and random_state=42, evaluated the same metrics, and extracted feature importances. I also visualized one decision tree from the forest to understand its decision-making process.
- Experiment 3: Highly Correlated Features
  I used the top three correlated features—Abdomen (0.81), Weight (0.61), and Chest (0.54)—with n_estimators=200 and random_state=42, and evaluated performance.

- Experiment 4: Abdomen Only
  I focused on Abdomen alone with n_estimators=200 and random_state=42, and calculated the metrics.

7.2 Results

- All Features (n_estimators=20, random_state=4)
  The RFR model achieved an R-squared of 0.655, with an MSE of 16.03 and an MAE of 3.33, showing strong performance.

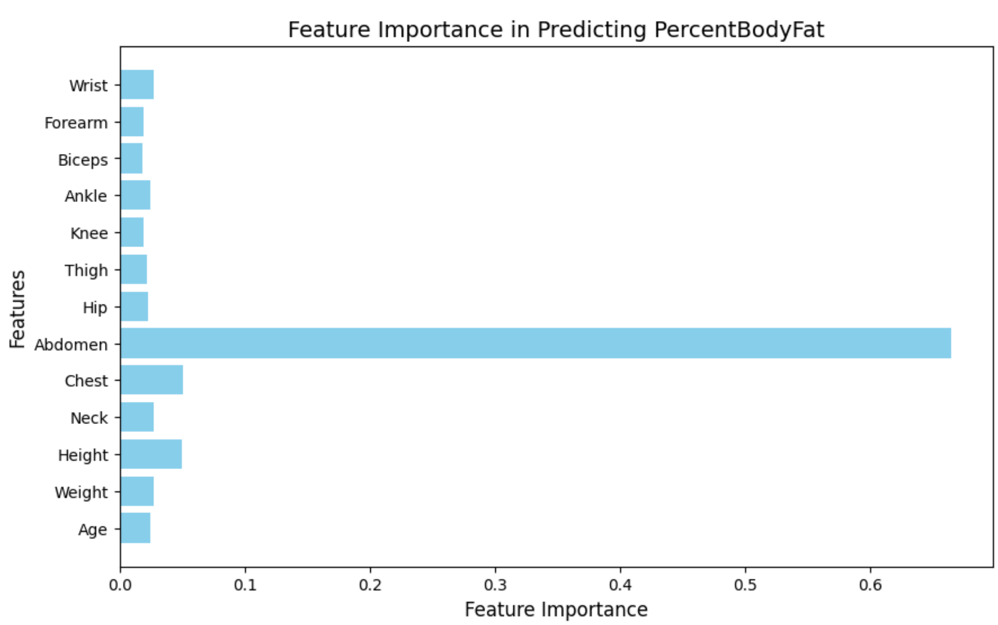
- All Features (n_estimators=200, random_state=42)
  With more trees, the R-squared remained 0.655, with the same MSE (16.03) and MAE (3.33), indicating that increasing n_estimators did not improve performance further.

- Highly Correlated Features (Abdomen, Weight, Chest)
  Using only the top three features, the R-squared dropped to 0.626, with an MSE of 17.41 and an MAE of 3.42, showing a slight performance decrease.

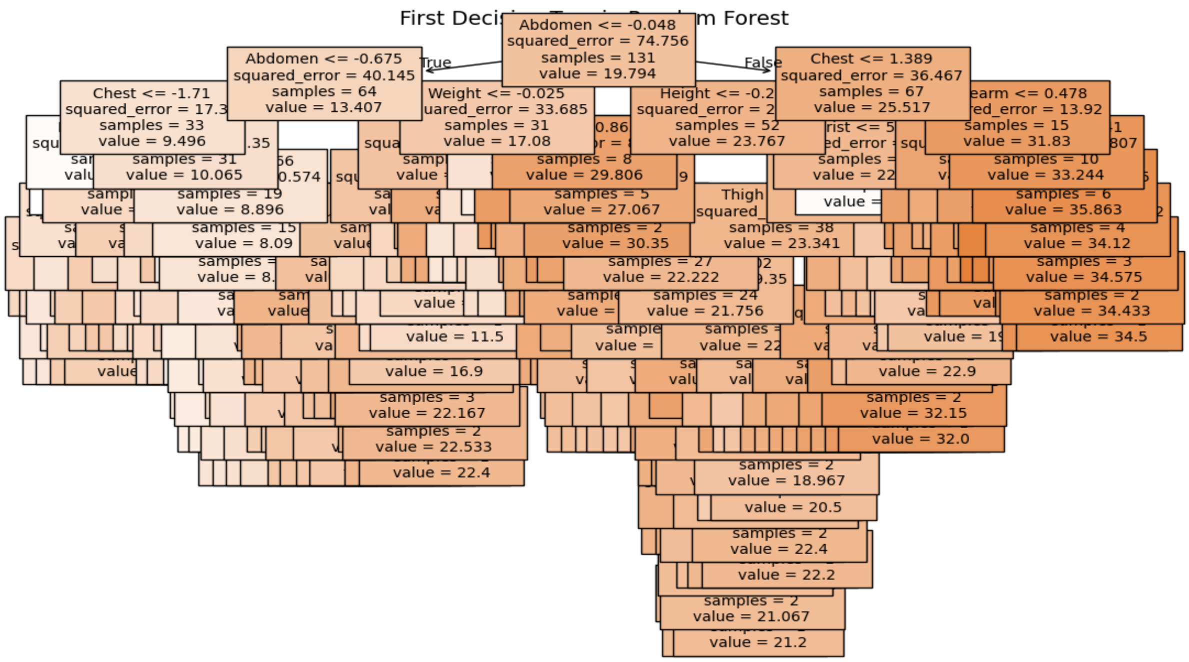
- Abdomen Only
  With just Abdomen, the R-squared was 0.313, with an MSE of 31.97 and an MAE of 4.57, indicating limited predictive power compared to SVR's Abdomen-only model (R^2 = 0.523).

7.3 Feature Importance and Decision Tree Analysis

- **Feature Importance:** The feature importance plot from the all-features model (n_estimators=200) shows Abdomen as the most important (~0.55), followed by Chest (~0.15) and Weight (~0.12). Smaller features like Wrist and Forearm contribute little (<0.05).



- **Decision Tree Analysis:** I visualized one decision tree from the forest. The tree starts by splitting on Abdomen ≤ -0.048, reflecting its high importance.



8. Conclusion

This study predicted PercentBodyFat using regression models, with Random Forest Regression (all features, n_estimators=200) achieving the highest R^2 of 0.655, explaining 65.5% of the variance. Linear Regression followed with $R^2 = 0.616$, and SVR performed slightly lower, $R^2 = 0.576$ when using highly correlated features. Abdomen emerged as a key predictor (correlation = 0.81), though using all features together maximized performance. Further improvements could be achieved through hyperparameter tuning, such as optimizing max_depth for Random Forest, C and epsilon for SVR, or applying Ridge/Lasso regularization for Linear Regression.