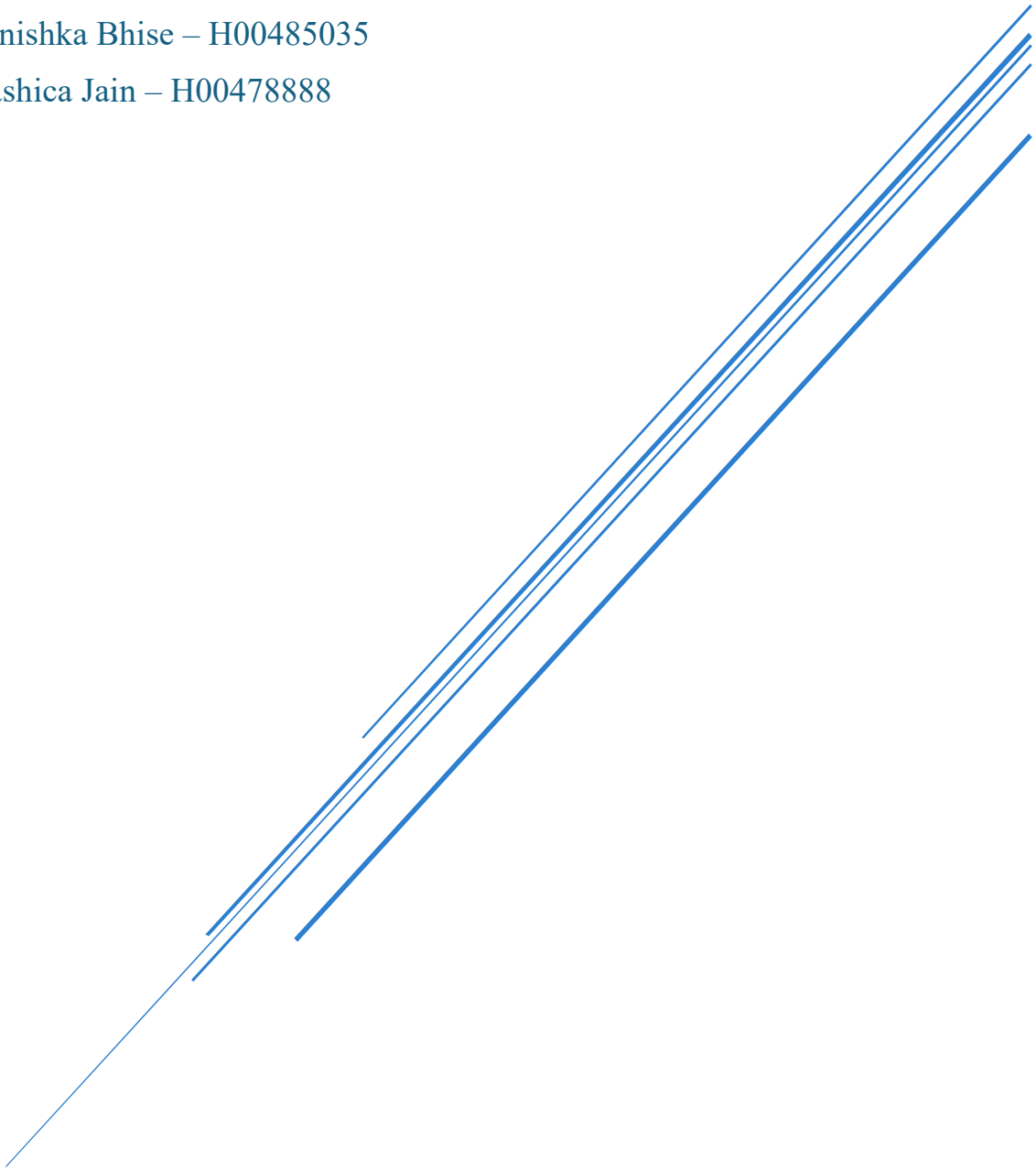
F21DL COURSEWORK

Title: Food Calorie Estimation

Group Members:

- Alfiamol Ajimshan Semeena – H00485185
- Alfiya Aziz Tamboli – H00476084
- Rukhsana Parilakathoott Shajahan – H00483791
- Tanishka Bhise – H00485035
- Yashica Jain – H00478888



Introduction

This report presents an analysis of both tabular and image datasets using a combination of data mining and machine learning techniques. After conducting exploratory data analysis and preprocessing steps such as data normalization and outlier removal, feature selection was performed to improve model performance. Clustering techniques, including K-Means and hierarchical clustering, were employed to derive meaningful groupings. Additionally, Convolutional Neural Networks (CNN) were applied to an image dataset, enhancing the analysis with deep learning techniques. Model evaluation was carried out using multiple algorithms, hyperparameter tuning, and cross-validation to ensure robust performance. The code, data, and documentation for this project are available in the GitHub repository.

Dataset Details

The project includes one tabular dataset and two image datasets, all sourced from Kaggle:

- Tabular Dataset: Food Nutrition Dataset (combines 5 datasets)

  The Food Nutrition Dataset offers detailed information on the calorific content and nutritional composition of various food items. It includes data on macronutrients such as fats (saturated, monounsaturated, polyunsaturated), carbohydrates, protein, and essential vitamins and minerals, providing a comprehensive view of food's nutritional value for each food item.

- Image Datasets
  - Fruits 100: A collection of images featuring 100 different types of fruits, useful for image classification and recognition tasks.
  - Fruit and Vegetable Image Recognition: A set of images of fruits and vegetables aimed at supporting machine learning models for food recognition and classification.

Related work

Image-based Calorie Prediction: Studies like **Rastegari et al. (2015)** and **Rudraraju et al. (2020)** used CNNs to estimate food calories from images, focusing on feature extraction from food pictures for accurate calorie estimation.

Nutrient and Calorie Prediction from Databases: **Ning et al. (2020)** applied machine learning on food nutrition databases to predict calorie content based on ingredients and portion sizes, using structured tabular data.

Consumer Records for Calorie Estimation: **Kuznetsov et al. (2019)** used collaborative filtering techniques on consumer food records to predict calorie intake based on consumption patterns.

These studies highlight the application of machine learning in food calorie prediction, both through images and structured data, showcasing its potential in health and nutrition fields.

Dataset Description and Analysis

Exploratory Data Analysis:

In exploratory data analysis, we checked for null values, calculated descriptive statistics, plotted histograms and scatter plots and determined correlation values between various attributes.

Data Normalization and Outlier Detection:

Yeo-Johnson is used to transform the dataset as it converts any distribution to normal distribution and deals with data that includes both positive and negative values. Outliers are detected using Z-scores (threshold value = |3|), and removed to improve model robustness.

Results - Yeo-Johnson transformation normalized the dataset. 67 outliers were removed, and feature correlation analysis reduced redundancy.

Pearson's correlation coefficient:

Features with correlation > 0.1 or < -0.1 are analyzed. A threshold of 0.1 is chosen to include more features as lot of the features are highly correlated to each other and will be removed later. Additional features, such as ‘Protein+Carbs+Fat’ and ‘Total Fats’, are manually added for improved target correlation.

Feature selection:

A correlation matrix is plotted to observe dependence between pairs of features. A linear regression model with backward sequential feature selector is used, with RMSE and R^2 as performance metrics.

Results - Sequential feature selection identified a 10-feature subset that minimized RMSE and maximized R^2, indicating a good fit and generalization capability.

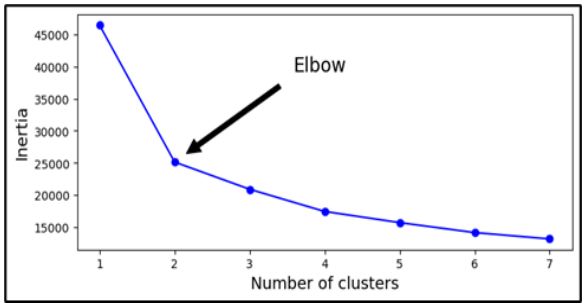
Table: Results of Sequential Feature Selection on Linear Regression

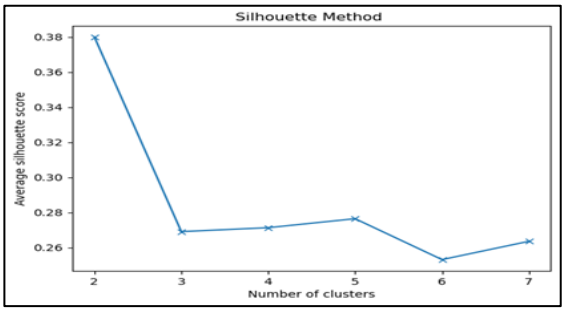
Features to select	Mean Squared Error	R^2 Score
7	0.354	0.869
8	0.350	0.872
9	0.338	0.880
10	0.335	0.882

Conclusion - Sequential feature selection reduced the feature set to 10, improving model performance ($R^2 = 0.882$, RMSE = 0.335).

Clustering

Two methods were used to determine the optimal number of clusters: elbow method and silhouette method.





Elbow method:

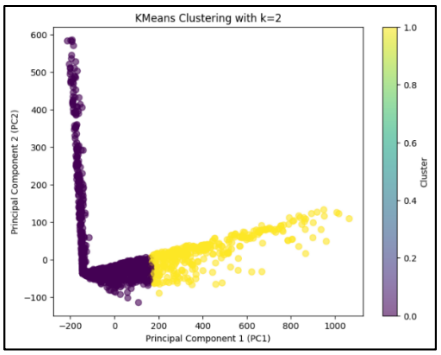
The optimal number of clusters are determined by plotting inertia for K-values from 1 to 7. The following graph shows the relationship between the number of clusters and the inertia.

Result - The plot of inertia values against the number of clusters clearly shows a sharp decrease from 1 to 2 clusters, after which the rate of reduction slows. This confirms that the optimal number of clusters for the dataset is 2.

Silhouette method:

The plot demonstrates the relationship between number of clusters and average silhouette score for each cluster count. A high silhouette score indicates a better-defined cluster.
Conclusion - The highest silhouette score is observed for cluster 2.

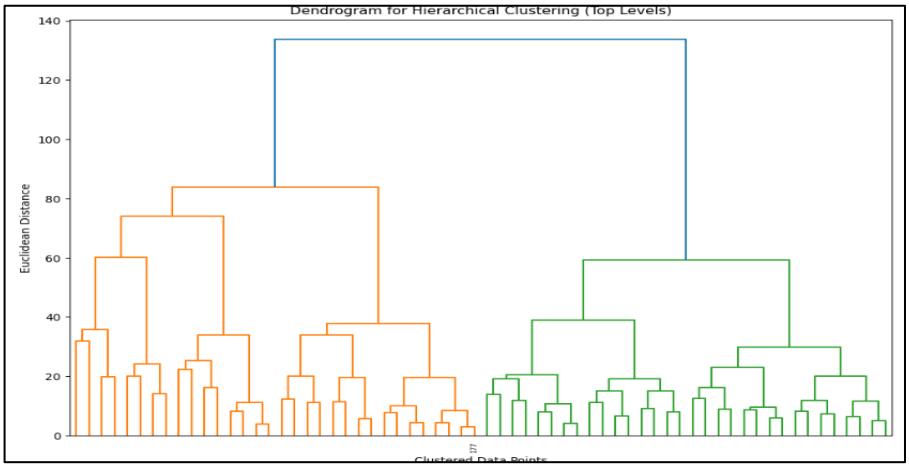
Kmeans:



Best choice for clustering. PCA visualization shows two clear and interpretable clusters, with one being tightly packed and the other more dispersed. k=2 provides the most distinct, meaningful, and interpretable clustering for the dataset.

Hierarchical Clustering:

It aims to cluster the data points in order to group similar food items into clusters based on their nutritional values. The dendrogram below represents the hierarchical clustering of the data points based on the Euclidean distance.



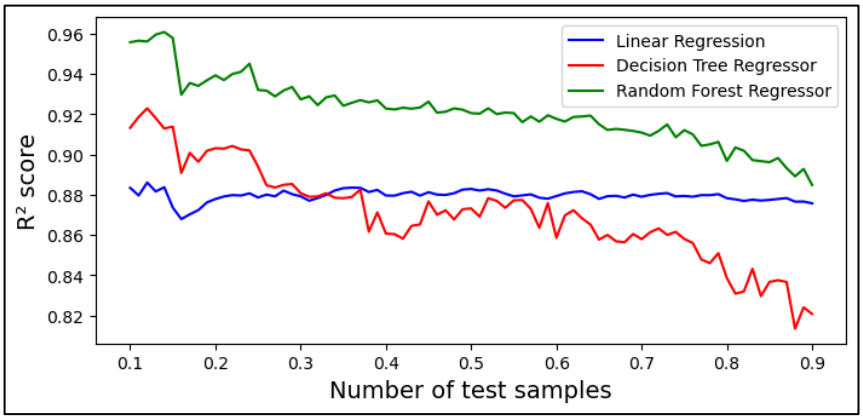
Result: In the above dendrogram, the largest distance at which clusters are merged (about 130 on the vertical axis) separates the entire dataset into two main clusters represented by orange and green colour. The orange has more subclusters than green suggesting it may contain more diverse or numerous food items.

Baseline Training and Evaluation Experiments

Comparison of models against train-test split:

Linear Regression, Decision Tree, and Random Forest models are compared using a train-test split. GridSearchCV is applied to optimize hyperparameters for Decision Tree and Random Forest, with R^2 used for performance evaluation across test sizes from 0.01 to 0.99.

Below graph is plotted for all three models for train test ratio:



Result - For smaller test sizes, models perform well due to larger training sets. However, as the test size increases, all models show a performance decline. Random Forest provides the most consistent results, while Decision Tree tends to overfit with smaller test sizes. Linear Regression maintains stable performance.

Conclusion - A test size between 0.10 and 0.20 provides high model accuracy for the Random Forest model, balancing a reasonable amount of data in both the training and testing sets.

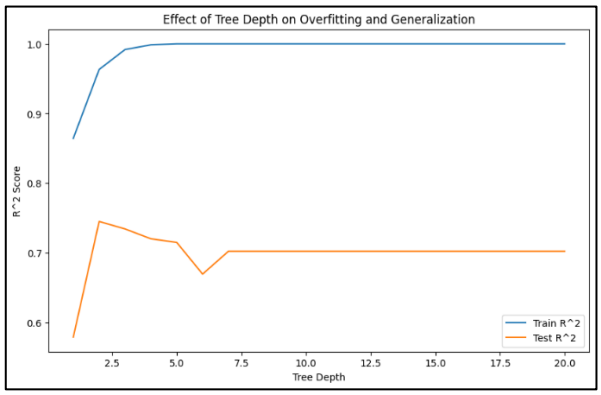
Hyperparameter Tuning:

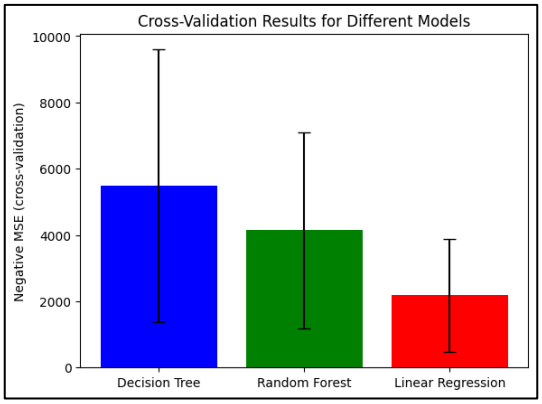
Hyperparameter Tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. GridSearchCV was the approach used for the parameter search.

{'C': 100, 'epsilon': 0.1, 'gamma': 0.01} hyperparameter combination gave the best performance with cross validation mean square error as 0.20895278292487732 and test mean square error as 0.2393170255044958.

Overfitting and Generalization:

The Decision Tree overfits as the tree depth increases, evident from the growing gap between training and test R^2 scores. Random Forest, though prone to overfitting, generalizes better than Decision Tree with a lower MSE on the test set.

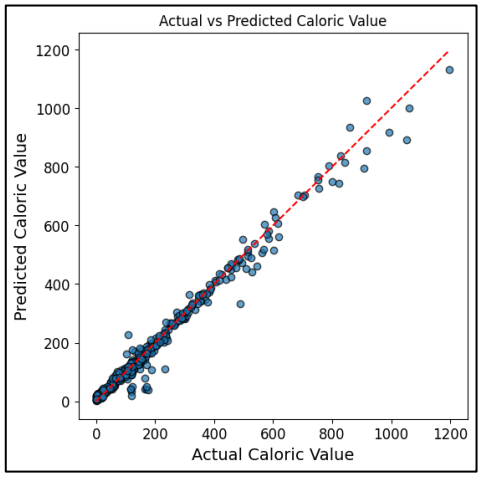


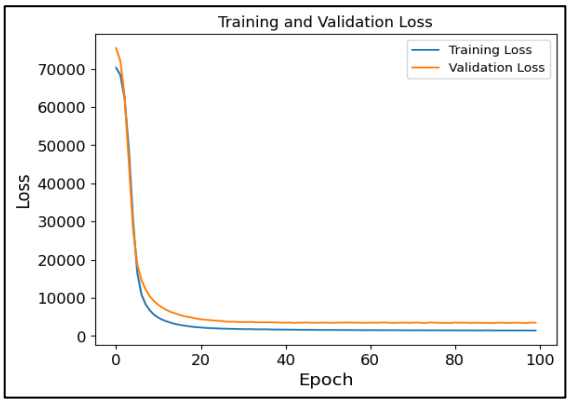


Cross-Validation Analysis:

Cross-validation revealed that Random Forest outperformed the Decision Tree in terms of generalization, with a lower test MSE. The Decision Tree's overfitting was more pronounced, which was addressed through regularization but still resulted in higher MSE.

Multiple Layer Perceptron



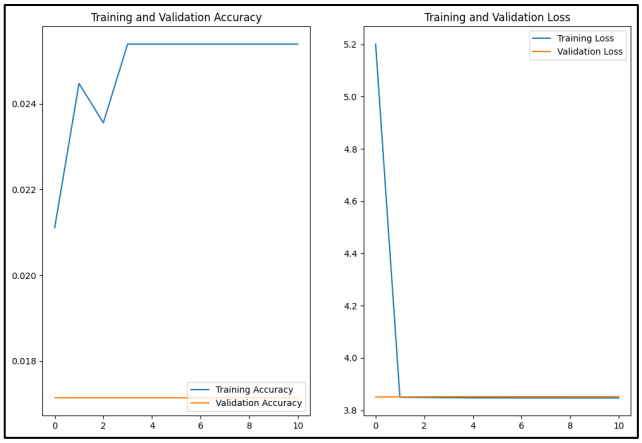


The MLP showed decreasing training loss, but the validation loss plateaued, indicating potential overfitting. Ideal performance would be characterized by lower validation loss and a higher R^2 score.

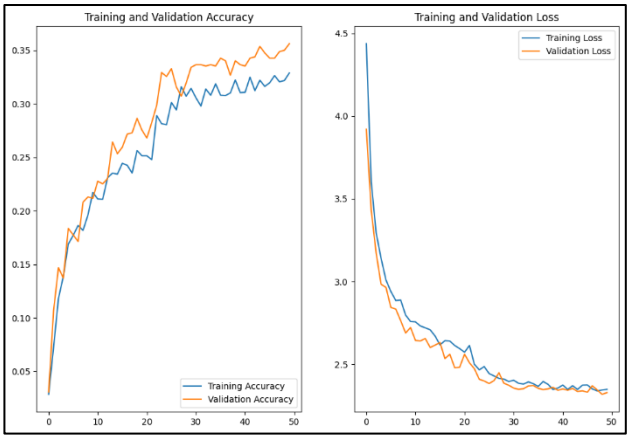
Convolutional Neural Network

Convolutional Neural Network has been applied to classify a small image data set.

Below are the graphs for ‘training-validation accuracy’ and ‘training-validation loss’ for classifications using multi layer perceptron and convolutional neural network.

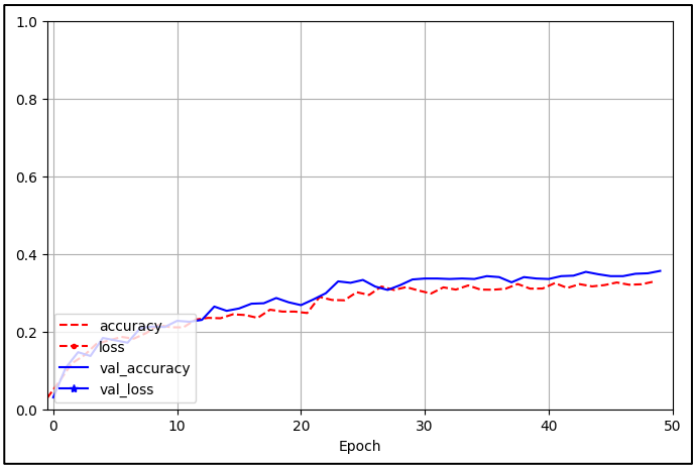


MLP



CNN

Both CNNs and MLPs exhibit overfitting when trained on the dataset, but CNNs demonstrate better learning and higher accuracy compared to MLPs



The above graph shows how the training accuracy and validation accuracy increases with every epoch for CNN.

So, the graphs indicate that CNNs are better at classifying images than MLP.

Summary

In conclusion, K-Means and Random Forest emerged as the most effective methods, with K-Means offering clear clusters and Random Forest providing robust generalization. Future work could explore additional hyperparameter tuning for further model optimization, as well as testing on additional datasets for broader applicability.

References

Scikit-learn (2024). *Scikit-learn Documentation*. Available at: https://scikit-learn.org/1.5/modules/

GeeksforGeeks (2019). *GeeksforGeeks Website*. Available at: https://www.geeksforgeeks.org

TensorFlow (n.d.). *Classifying images with TensorFlow*. Available at: https://www.tensorflow.org/tutorials/images/classification

Group Declaration

The following is a summary of each team member's contributions to the project:

- Alfiamol Ajimshan Semeena – Hierarchical Clustering, Hyperparameter Tuning
- Alfiya Aziz Tamboli – K-means, Overifitting and generalization in Decision Trees, Multi-Layer Perceptrons (MLP)
- Rukhsana Parilakathoott Shajahan – Exploratory Data Analysis, Silhouette method
- Tanishka Bhise – Convolutional Neural Networks
- Yashica Jain – Feature Selection, Elbow method, Model comparison with train test split