DEPARTMENT: Computer Science

Module Code: CCS6610

Academic Year: 2024

Submission Deadline: 25/02/2024

Actual Submission Date: 25/02/2024

Student Registration Number
23032

Declaration: *I have completed and submitted this work by myself without assistance from or communication with another person either external or fellow student or any AI type of content generator. I understand that not working on my own will be considered grounds for unfair means and will result in a fail mark for this work and might invoke disciplinary actions. This piece of assessment will be continuously checked for its academic integrity until my graduation and the mark will be revised if it is found to breach the unfair means policy. It is at the instructor's discretion to conduct an oral examination which will result in the award of the final grade for that particular piece of work.*

Abstract

This report analyses the given data, which are about Obesity based on eating habits and physical condition. Classification and clustering technics were used. The chosen algorithms are Random Forests and K-Means. Two Random Forests models were created for the classification task, each with distinct hyperparameters. The models were assessed using performance metrics such as accuracy, precision, recall, and F1-score. Moreover, potential overfitting was checked by comparing the performance of the models on training and testing datasets. GridSearchCV was utilized for hyperparameter tuning to enhance model performance. Confusion matrices, classification reports, and precision-recall curves were used to analyze model outcomes and evaluate their performance. K-Means clustering was used for clustering, in order to identify unique clusters using specific features. The ideal number of clusters was identified by utilizing the elbow method and silhouette score. Clustering quality and cluster separation were performed using evaluation metrics like silhouette score and Davies–Bouldin Index. Plots like scatter plot, were used to visualize the distribution and separation of clusters.

Table of contents

1.Introduction 4
2.Preprocessing and visualization 4
3. Classification with Random Forests 6
1.1 Random Forests creation 6
1.2 Confusion Matrix and Classification Report 7
1.3 Model Overfitting Check 8
4. Clustering with K-Means 11
4.1 K-Means Creation 11
4.2 Clustering Evaluation 11
4.2.1 Elbow method 11
4.2.2 Silhouette score 12
5. Conclusion 13
6. References 14

1.Introduction

This report provides an analysis of data focused on predicting obesity levels in individuals, using information about their dietary habits and physical health. The dataset contains a lot of features that possibly are related to obesity levels, such as dietary habits, physical activity, family history, and socio-demographic characteristics. 23% of the dataset was collected directly from users through a survey conducted by Fabio Mendoza Palechor and Alexis de la Hoz Manotas using a web platform. 77% of the dataset was created synthetically using the Weka tool and the SMOTE filter.

Tools like sorting and grouping with machine learning are great for looking at hard sets of info. They pull out points that can help change health actions and rules. In this text, we'll first look at the data set. We will make it neat and show it in ways to see how the data spreads and how different parts link up. Then we will dig into two big jobs: sorting with Tree Nets and grouping with Key Piles. By looking into these, we plan to make models that guess fat rates well and find clear groups of people from what they are like.

When this report is done, we will know more about what affects how many people are too heavy. We will also have looked at good ways to use smart computer programs to study and make sense of big sets of health data.

2.Preprocessing and visualization

To begin with, I started to analyze the data visualizing some features. The first one presents the individuals having a family history of being overweight and being obese.

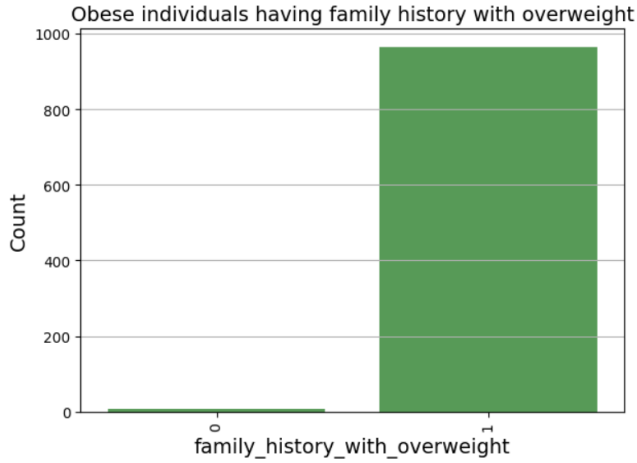

Fig. 1: Individuals having a family history of being overweight.

As the plot in Fig. 1 shows, most of the obese individuals appear to have family history with being overweight, we can assume that there is a good relation between these two features [1].

Next, the plot below represents the distribution of age for obese individuals.

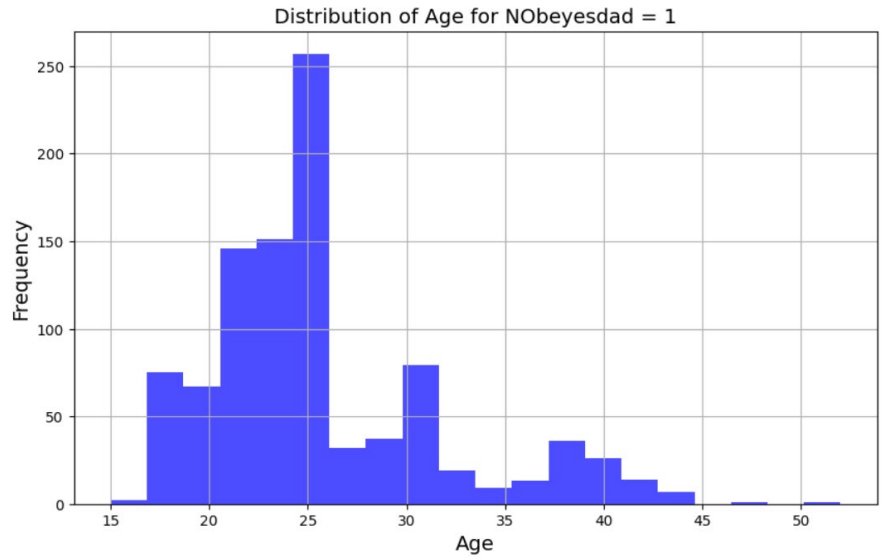

Fig. 2: Distribution of age for obese individuals.

Based on this histogram in Fig. 2, most of the obese individuals seem to be around 25 years old.

First, after a visualization in of the dataframe in Fig. 3, there is a feature that has different (higher) values.

	Gender	Age	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	Automobile	Bike	Motorbike
0	0	21.0	1	0	2.0	3.0	1	0	2.0	0	0.0	1.0	0	0	0	0
1	0	21.0	1	0	3.0	3.0	1	1	3.0	1	3.0	0.0	1	0	0	0
2	1	23.0	1	0	2.0	3.0	1	0	2.0	0	2.0	1.0	2	0	0	0
3	1	27.0	0	0	3.0	3.0	1	0	2.0	0	2.0	0.0	2	0	0	0
4	1	22.0	0	0	2.0	1.0	1	0	2.0	0	0.0	0.0	1	0	0	0

Fig. 3: Visualization of the dataframe.

To improve the training results, scaling of the features has been applied as presented in Fig. 4 and Fig. 5.

```
[18] sc = StandardScaler() # Setup the scaler object

[19] X_train = sc.fit_transform(X_train) # Transform the train data
     print(X_train) # Take a look at the values

     [[-1.01311923 -0.53264595  0.46961161 ... -0.07719764  0.59012101
       -0.1729054 ]
      [-1.01311923 -0.54423543  0.46961161 ... -0.07719764  0.59012101
       -0.1729054 ]
```

Fig. 4: Feature scaling for the training.

```
X_test = sc.transform(X_test) # Transform the test data
print(X_test)  # Take a look at the values
```

```
[[-1.01311923 -0.62424632  0.46961161 ... -0.07719764  0.59012101
  -0.1729054 ]
 [-1.01311923  0.2395332   0.46961161 ... -0.07719764  0.59012101
  -0.1729054 ]
```

Fig. 5: Feature scaling for the test

3. Classification with Random Forests

1.1 Random Forests creation

Two random forests models were created. The hyperparameters were defined for both models as max_leaf_nodes=10, criterion = 'gini', random_state=42 [2][3].

The first model was defined with ten(10) decision trees (n_estimator), as the Fig. 6 indicates.

```
# @title Model 1
randf_1 = RandomForestClassifier(n_estimators=10, max_leaf_nodes=10, criterion = 'gini', random_state=42)
randf_1.fit(X_train, y_train)
y_pred_1 = randf_1.predict(X_test)
```

Fig. 6: Random forests model 1

The second model was defined with five hundred (500) decision trees (n_estimator), as we can observe in the Fig. 6.

```
# @title Model 2
randf_2 = RandomForestClassifier(n_estimators=500, max_leaf_nodes=10, criterion = 'gini',random_state=42)
randf_2.fit(X_train, y_train)
y_pred_2 = randf_2.predict(X_test)
```

Fig. 6: Random forests model 2

The bar plot in Fig. 7 below, compares the accuracy of both Random Forests models.

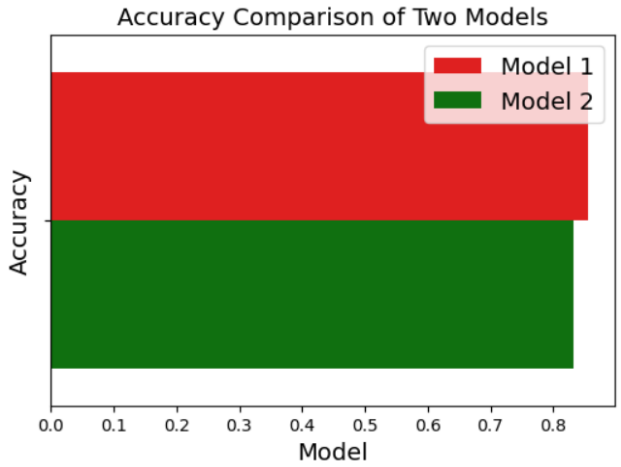

Fig. 7: Accuracy comparison of the two models

As we can see in Fig. 7, the first model has more accuracy with the percentage of 85.579% and the second one follows with 83.215%.

1.2 Confusion Matrix and Classification Report

Moreover, a confusion matrix and a classification report was created for each model. The confusion matrix for Model 1 presented in Fig. 8, reveals the performance of the classifier in terms of true positive, true negative, false positive, and false negative predictions. From the confusion matrix, we can observe that Model 1 achieved an accuracy of 86%, with a slightly higher precision for the "Not Obese" class compared to the "Obese" class.

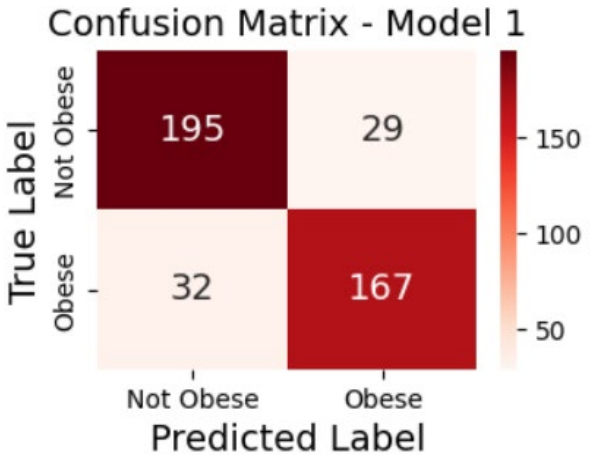

Fig. 8: Confusion matrix for model 1

The classification report in Table 1, provides a detailed breakdown of precision, recall, and F1-score for each class. Precision indicates the proportion of true positive predictions among all positive predictions, while recall represents the proportion of true positives correctly identified by the model. The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance.

Table 1: Classification Report for model 1

	precision	recall	f1-score	support
Not Obese	0.86	0.87	0.86	224
Obese	0.85	0.84	0.85	199
accuracy			0.86	423
macro avg	0.86	0.85	0.86	423
weighted avg	0.86	0.86	0.86	423

Similarly, the confusion matrix in Fig. 9 and classification report for Model 2 in Table 2, provide insights into its performance. While Model 2 achieved a slightly lower accuracy compared to Model 1, it still demonstrates competitive performance across precision, recall, and F1-score metrics.

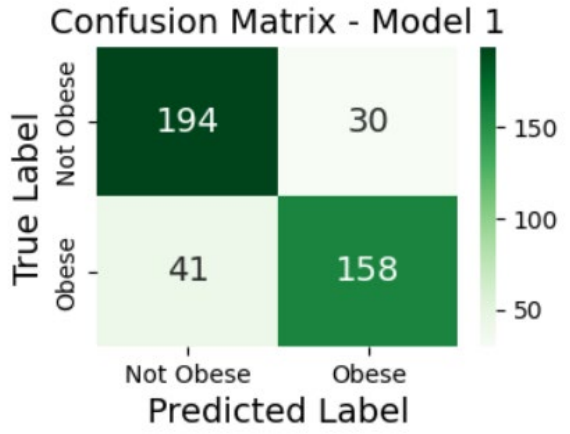

Fig. 9: Confusion matrix for model 2

Table 2: Classification Report for model 2

	precision	**recall**	**f1-score**	**support**
Not Obese	0.83	0.87	0.85	224
Obese	0.84	0.79	0.82	199
accuracy			0.83	423
macro avg	0.83	0.83	0.83	423
weighted avg	0.83	0.83	0.83	423

1.3 Model Overfitting Check

To assess potential overfitting, we compare the performance metrics of the models on both training and testing datasets as the Table 3 demonstrates [4]. While Model 1 exhibits slightly higher metrics on the training set compared to the testing set, the difference is not substantial, indicating relatively stable performance.

Table 3: Initial Model Metrics

Training Accuracy:	0.8364928909952607
Testing Accuracy:	0.8321513002364066

Training Precision:	0.8094645080946451
Testing Precision:	0.8404255319148937
Training Recall:	0.8408796895213454
Testing Recall:	0.7939698492462312
Training F1 Score:	0.8248730964467005
Testing F1 Score:	0.8165374677002584

Potential strategies to mitigate overfitting have been explored, including feature selection, regularization techniques, and hyperparameter tuning [2]. Additionally, visualizations such as precision-recall curves and correlation matrices aid in understanding the data and refining the models as the Fig. 10 and the Table 4 declare.

The plots below show the precision-recall curve.

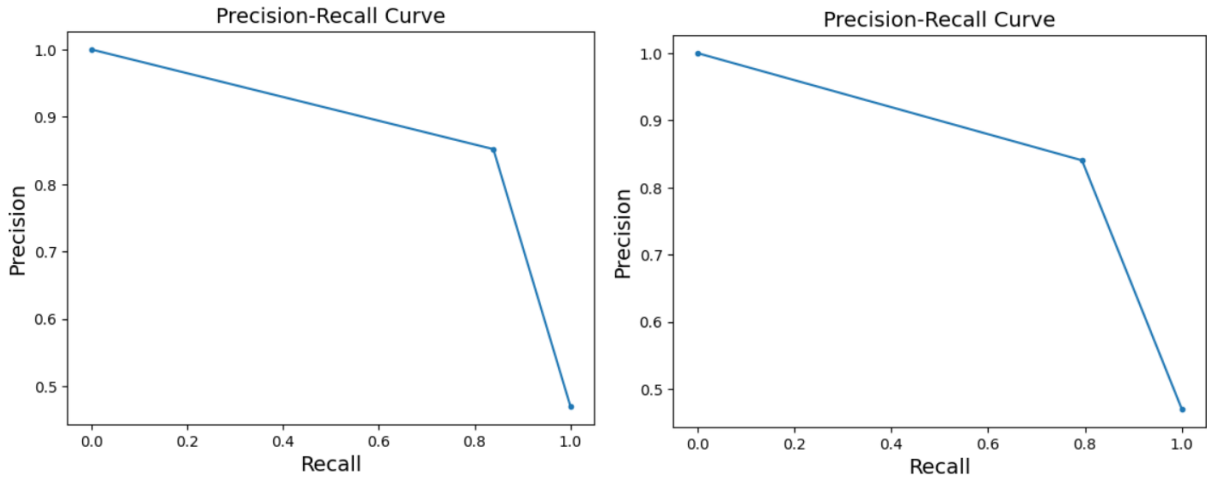

Fig. 10: precision-recall curves for the two models

Table 4: Precision-recall for each model

	Model 1			Model 2		
Precision	0.470	0.852	1.	0.470	0.840	1.
Recall	1.	0.839	0.	1.	0.79	0.

Moreover, GridSearchCV function has been applied for the first Model- displayed in Fig. 11, to identify the best combination of decision trees and max depth.

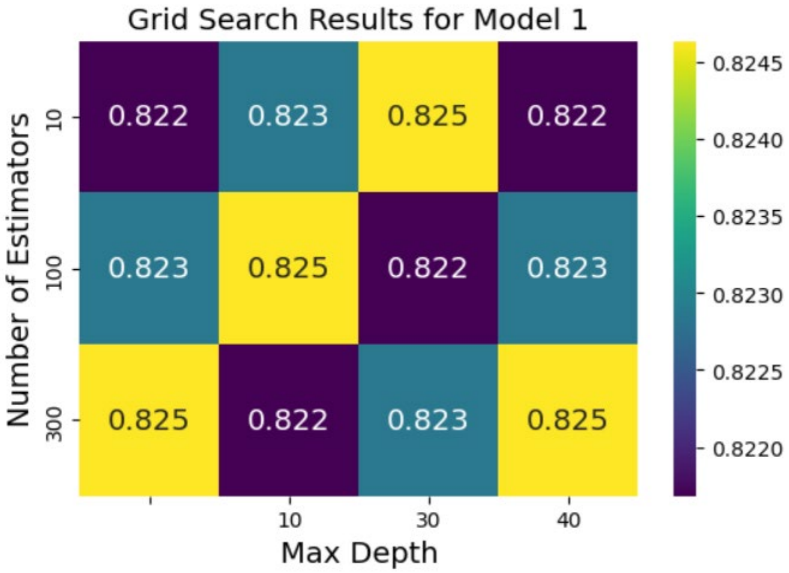

Fig. 11: Visualization for GridSearchCV function

Hyperparameter tuning of machine learning models is out of the scope of this assignment, but hypothetically, different combinations of decision trees and max depth could be used to regularize the hyperparameters, like decision trees=100 and max depth=10 as the plot above shows.

Finally, a visualization was performed with the selected features to be Walking and family_history_with_overweight as presented in Fig. 12.

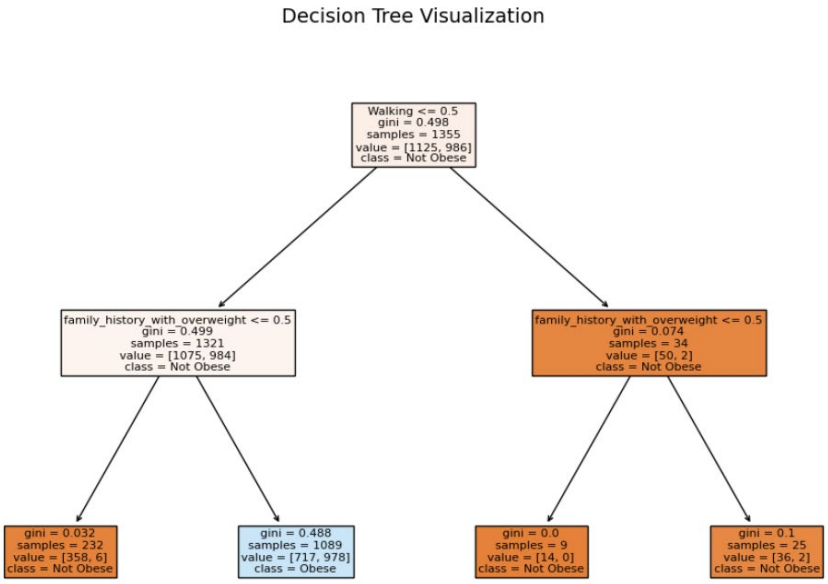

Fig. 12: Decision tree visualization

4. Clustering with K-Means

4.1 K-Means Creation

The K-Means clustering algorithm was picked to apply clustering to the dataframe. K-Means clustering is employed to identify distinct clusters within the dataset based on selected features [2][3]. The features that were selected in this case were FAF which represents how much physical activity a respondent does on a scale of 0 to 3 and NCP which represents how many main meals a respondent has daily (0 for 1-2 meals, 1 for 3 meals, and 2 for more than 3 meals).

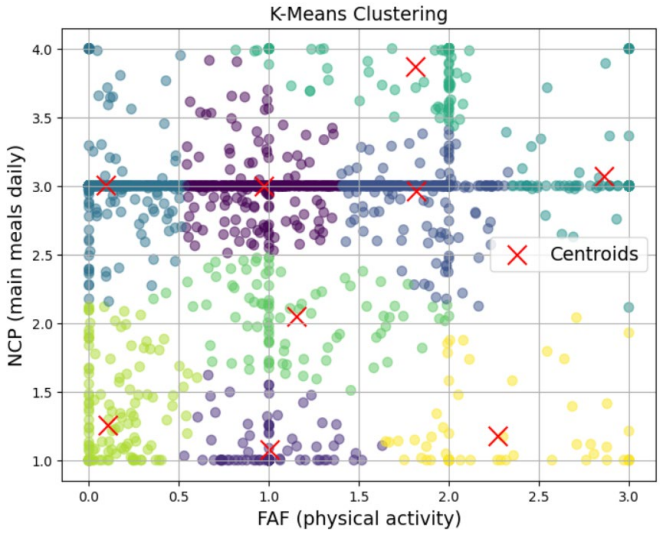

Fig. 13: Clustering Visualization

The number of the clusters presented in Fig. 13, was defined as 9, the random state 42 and the n_init parameter that controls the number of times the k-means algorithm will be run with different centroid seeds, as 10. Visualizations such as scatter plots and silhouette scores were used to provide insights into the clustering quality and cluster separation.

4.2 Clustering Evaluation

4.2.1 Elbow method

The elbow method is utilized to determine the optimal number of clusters (k) by evaluating the within-cluster sum of squares (inertia) for different values of k. The elbow point indicates an optimal trade-off between clustering quality and the number of clusters [5] and displayed in Fig. 14.

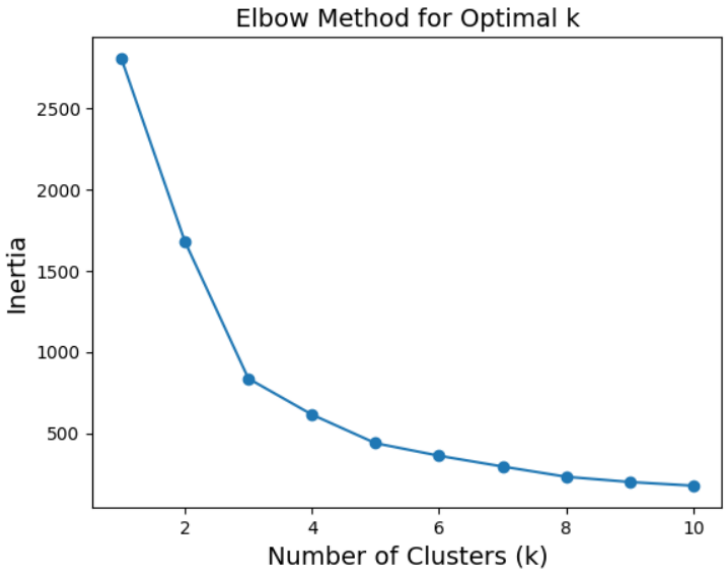

Fig. 14: Elbow Method

The Elbow method doesn't necessarily mean that it will be correctly identifying the correct number of k, so the next step was to visualize the Silhouette score.

4.2.2 Silhouette score

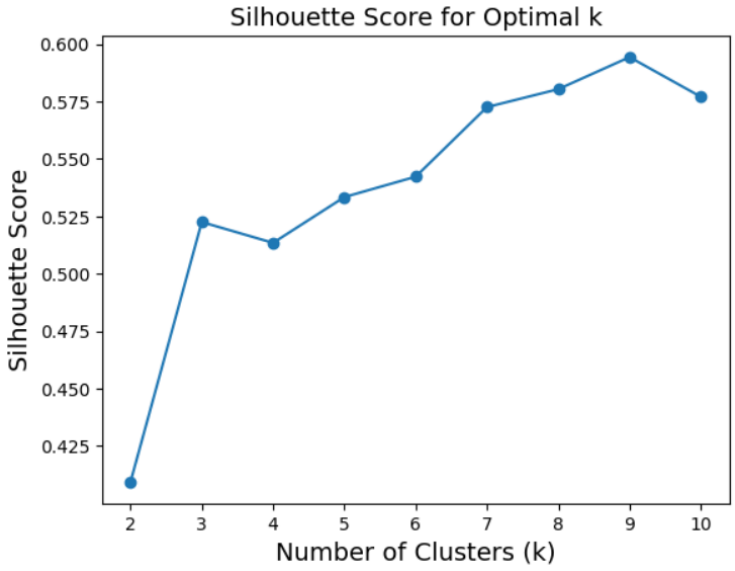

Fig. 15: Silhouette score

The shadow score rates how tight and apart the packs are, with more top scores meaning the packs are clearer and more split up. By looking at shadow scores for other k values, we can find the best count of packs that make the pack quality the best [6]. As the Silhouette score plot in Fig. 15 shows, the optimal number of clusters was not 3, as the Elbow method showed, but the optimal number of clusters is 9, which is the number that was used in our K-Mean model. To evaluate the clustering algorithm, two different metrics were used, the Silhouette Score and the Davies–Bouldin Index.

The silhouette score metric, which is used to evaluate the quality of clusters. This metric describes the similarity of an object to its own cluster (cohesion), compared to other clusters (separation). In this case, the silhouette score was -0.157. This indicates that the clusters are not so well separated, and probably there is a degree of overlap between clusters or incorrect assignment of data points to clusters. This indicates that the clustering algorithm probably have a good performance.

The Davies–Bouldin Index (DBI) is a metric used to evaluate the performance of clustering algorithms. It measures the average similarity between each cluster and its most similar cluster, relative to the average dissimilarity between points in different clusters [6].

The Davies–Bouldin Index for our clustering model was 7.373. This metric indicates good clustering, with lower values indicating tighter clusters with good separation.

5. Conclusion

In conclusion, this report has provided a comprehensive analysis of a dataset aimed at estimating obesity levels. After data analysis, preprocessing, and visualization, insights gained into the distribution of the features and identified potential relationships between them.

Our analysis included two main tasks: classification using Random Forests and clustering using K-Means. For classification, we developed Random Forest models to predict obesity levels based on various attributes. These models demonstrated competitive performance, with Model 1 achieving an accuracy of 86% and Model 2 achieving 83%.

For clustering, we employed K-Means to identify distinct clusters of individuals based on their characteristics. The optimal number of clusters was determined using the elbow method and silhouette scores. While the silhouette score indicated some overlap between clusters, the Davies–Bouldin Index suggested good clustering quality.

In summary, our study gives useful ideas about what causes fatness and points out ways to look more into it and step in. Using smart computer ways, we can learn more and deal with the hard parts linked to being very fat. This can help make people's health better.

Through continued exploration and refinement of models, we can enhance our understanding of obesity dynamics and develop targeted interventions to mitigate its impact on individuals and communities.

6. References

1. Molinaro, A. M., Simon, R., & Pfeiffer, R. M. (2005). Prediction error estimation: a comparison of resampling methods. Bioinformatics, 21(15), 3301-3307.

2. Lee, J. W., & Lee, J. B. (2021). A review of feature selection techniques in bioinformatics. Genomics & Informatics, 19(3), e30.

3.Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.

4. Scikit-learn developers. (2020). Random State. Scikit-learn: Machine Learning in Python, 0.24.1 Documentation. Retrieved from https://scikit-learn.org/stable/glossary.html#term-random-state

5. Halkidi, Maria, et al. "Clustering validity assessment: Finding the optimal partitioning of a data set." Data mining and knowledge discovery handbook. Springer, 2010.

6. Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. Prentice-Hall.