Table of Contents

1.0 Project Background

Over the last few decades, obesity has become so prevalent and is one of the most significant public health challenges all around the globe. Along with increasing obesity rates, chronic diseases that are related to obesity such as diabetes, cardiovascular conditions, and certain cancers have intensified, putting a heavy burden on healthcare systems. To tackle this growing epidemic, governments and healthcare organizations have implemented a variety of strategies, including public health campaigns and policy interventions, but progress has been slow. This is due to Obesity's multifactorial nature, which can be caused by lifestyle, genetics, and environmental factors. This makes it difficult to address through traditional healthcare approaches alone.

On the other hand, the rapid development in computing resources and machine learning algorithms has revolutionized healthcare research by enabling more sophisticated and data-driven analysis. Predictive analytics that are powered by machine learning offer the ability to identify patterns and predict outcomes in large, complex datasets that might be missed in traditional approaches. This has opened up new approaches for tackling complex health problems like obesity, where many interrelated factors contribute to individual risk. Machine learning models have been increasingly applied in areas like disease diagnosis, personalized treatment, and preventive care, transforming how healthcare providers predict and manage patient outcomes.

In the context of obesity, by incorporating variables from different aspects such as demographics, lifestyle habits and clinical data into machine learning models, researchers can adopt a more personalized and precise approach to predicting obesity risk in patients. Several studies have demonstrated the potential application and usage of these models to improve the accuracy of obesity risk forecasts, leading to better prevention strategies and targeted interventions. However, gaps remain in developing models that are generalizable across different populations and accurately account for the wide array of factors contributing to obesity.

This project builds on these foundations by focusing on the application of machine learning to predict obesity risk and understand its driving factors. Incorporating a comprehensive dataset sourced from Kaggle (Koklu & Sulak, 2024), which includes demographic and lifestyle information from 1610 individuals in Türkiye, this project aims to develop advanced predictive models. These models will support healthcare professionals in identifying high-risk individuals and tailoring preventive strategies more effectively, contributing to the broader goal of mitigating the obesity epidemic through data-driven solutions.

2.0 Project Objective

1. To identify heart-failure predictors based on health indicators of cardiovascular disease patients.
2. To build and compare multiple predictive models with varied modelling techniques for heart-failure outcome prediction.

3. To test and verify the efficacy of each model, identifying the most accurate modelling technique for heart-failure outcome prediction.

3.0 Data Cleaning

3.1 Checking for NA values

The dataset was evaluated for missing values across all the features as part of the data cleaning process. Figure 3.1 indicates that none of the columns contain missing values, with each feature reporting zero missing records. This includes demographic attributes (Sex, Age, Height), lifestyle factors (Smoking, Physical_Exercise, Type_of_Transportation_Used), dietary habits (Consumption_of_Fast_Food, Food_Intake_Between_Meals), and the target variable (Class). As a result, no imputation or additional handling of missing data is required, ensuring the dataset is complete and ready for further preprocessing and analysis.

```
Check NA value
 Sex : 0
 Age : 0
 Height : 0
 Overweight_Obese_Family : 0
 Consumption_of_Fast_Food : 0
 Frequency_of_Consuming_Vegetables : 0
 Number_of_Main_Meals_Daily : 0
 Food_Intake_Between_Meals : 0
 Smoking : 0
 Liquid_Intake_Daily : 0
 Calculation_of_Calorie_Intake : 0
 Physical_Excercise : 0
 Schedule_Dedicated_to_Technology : 0
 Type_of_Transportation_Used : 0
 Class : 0
```

Figure 3.1 Identifying missing values

3.2 Outliers Detection

The Age column ranges from 18 to 54, with quartiles (25th, 50th, and 75th percentiles) at 25, 32, and 41 respectively, suggesting a relatively symmetrical distribution without extreme outliers. The Height column ranges from 150 to 193, with quartiles (25th, 50th, and 75th percentiles) at 161, 168, and 174 respectively, suggesting a balanced distribution. Both of the columns are further categorized into age and height groups where no significant or extreme outliers are presented.

	Age
count	1610.000000
mean	33.115528
std	9.835076
min	18.000000
25%	25.000000
50%	32.000000
75%	41.000000
max	54.000000

Age_Group	count
18-26	548
27-35	426
36-44	372
45-54	242

	Height
count	1610.000000
mean	167.741615
std	7.979873
min	150.000000
25%	161.000000
50%	168.000000
75%	174.000000
max	193.000000

Height_Group	count
168-176	576
159-167	565
177-185	232
150-158	222
186-195	15

Figure 3.3 Outliers Detection in Age and Height

4.0 Exploratory Data Analysis (EDA)

EDA is the process of analyzing and summarizing the key features to discover underlying patterns, relationship between the variables as well as meaningful insights. This process typically involves data visualization through bar chart, pie chart and histograms.

4.1 Distribution of Weight Class

Pie chart is used to illustrate the distribution of weight class in their respective proportions. The dataset shows a majority of individuals with “Normal” (40.9%) and “Overweight” (36.8%) weight classification, followed by “Obesity” (17.8%) and “Underweight” (4.5%).

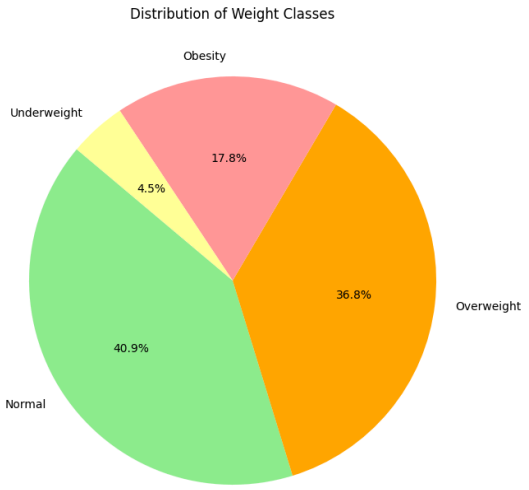


Figure 4.1 Visualization of Weight Class using Pie chart

4.2 Distribution of Key Features by Target Variable

Bar chart is used to visualize distributions across various key features segmented by weight class. Based on the histograms overweight and obesity are more prevalent among older age groups (36-44 and 45-54), shorter height ranges (159-167 cm), and among males compared to females. Individuals in overweight and obesity weight classes are also more commonly associated with automobile use, suggesting a less active lifestyle.

Additionally, a family history of being overweight or obese which indicates a potential genetic or environmental influence might not be strongly associated with higher counts in the Overweight and Obesity as shown in the bar chart. Lastly, it is also shown that dietary habits play a significant role, with those who “Sometimes”, “Usually” and “Always” eat between meals showing a higher proportion of overweight and obese individuals.

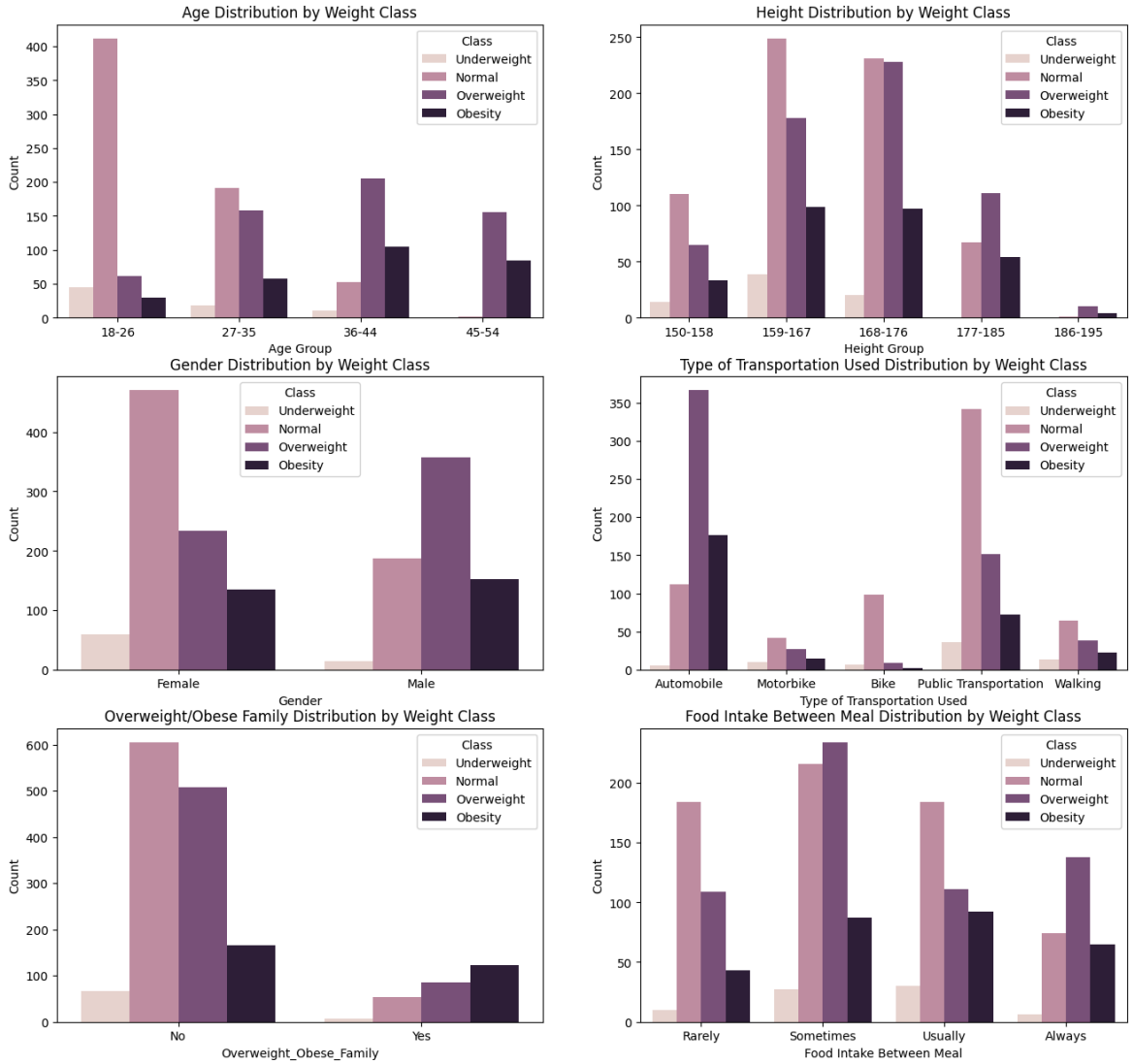
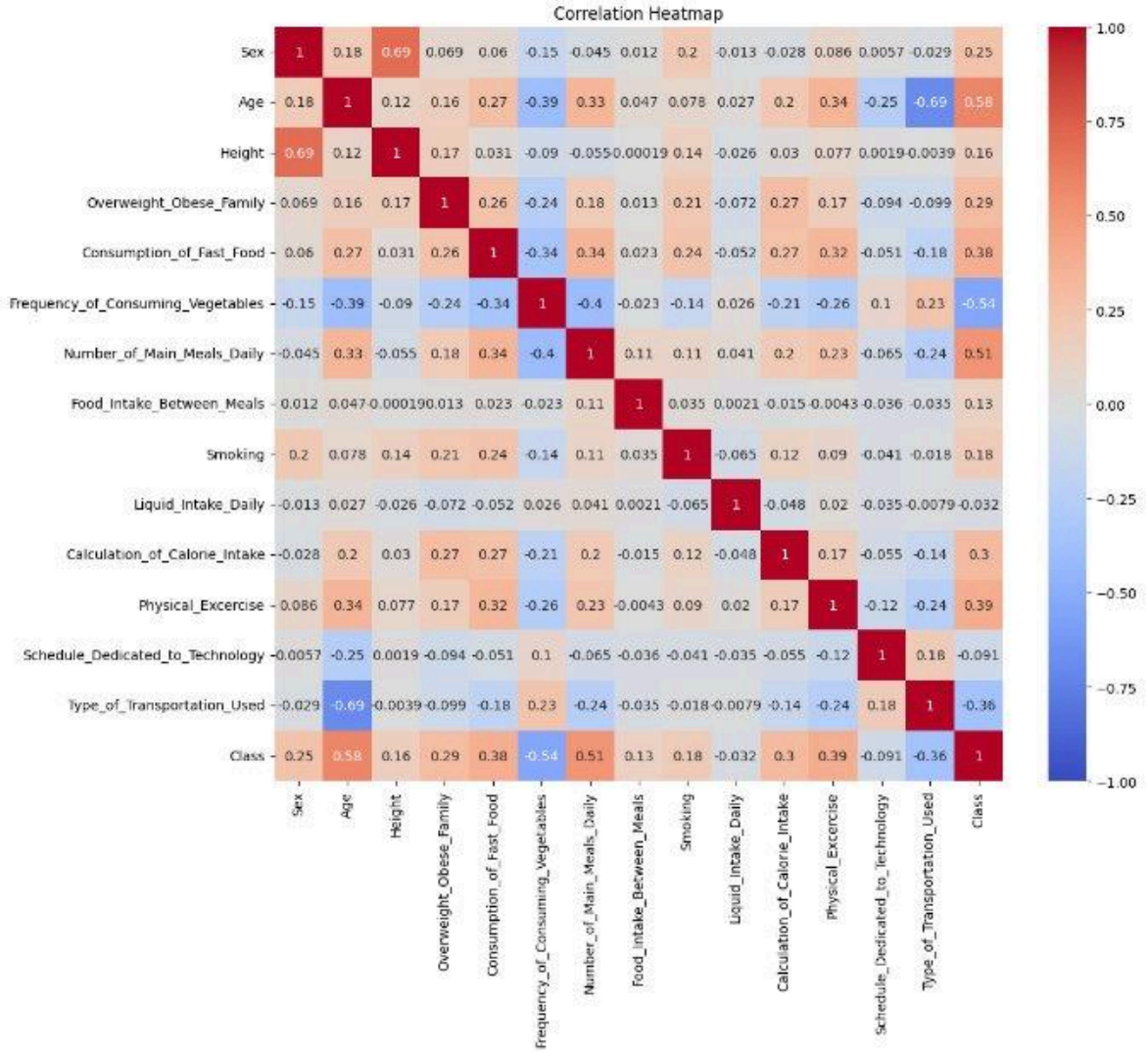


Figure 4.2 Visualization of Key Features by Weight Class using Bar chart

4.4 Correlation Map

Correlation map is also used to identify patterns of variables which are strongly linked to obesity risk. It also assists in selecting features for model building by identifying highly correlated predictors or variables that might provide unique insights into the target variable. These statistical correlations can be used to compare with a machine learning model's feature importance to further support the output of the model.



5.0 Data Preprocessing

5.1 Label-Encoding of Binary Variables

The dataset includes features with binary data that were standardized to ensure the consistency. Specifically, binary variables such as Sex, Overweight_Obese_Family, Consumption_of_Fast_Food, Smoking, and Calculation_of_Calorie_Intake were adjusted to use a uniform representation of 0 and 1. For instance, 0 represents female and 1 represents male for Sex, whereas 0 represents No and 1 represents Yes for the rest of the binary variables.

```
Unique values in Sex: [0 1]
Unique values in Overweight_Obese_Family: [0 1]
Unique values in Consumption_of_Fast_Food: [0 1]
Unique values in Smoking: [0 1]
Unique values in Calculation_of_Calorie_Intake: [0 1]
```

Figure 5.1 Converting Binary Variables to 0 and 1

5.2 One Hot Encoding for Categorical Variables

One Hot Encoding is also applied to convert the categorical variables with more than two categories to a binary format, making the variables more suitable for machine learning algorithms that require numerical inputs. For instance, the category “Rarely”, “Sometimes”, “Always” in Frequency_of_Consuming_Vegetables column will be converted into three new columns, with each row will have a binary value (0 or 1) in these columns, indicating the frequency of consuming vegetables.

4.0 Data Modelling

One Hot Encoding in Predictive Models

One hot encoding is a technique which can help improve the model performance by correctly representing the categorical features, improving feature interactions and preventing ordinal misinterpretations.

StratifiedKFold

The StratifiedFold split the dataset into K parts (K=5 in our case) for cross-validation. This KFold technique can ensure the proportion of each class in both training and test data mirror distribution of classes of the original dataset. StratifiedKFold can reduce the selection bias which avoids overrepresented or underrepresented certain classes in some folds. A more interpretable evaluation result can be achieved as the variance in metrics such as accuracy, precision, recall, and F1 score is reduced across folds with consistent class proportions.

By using StratifiedKFold, we can also explore the best combination of training and test dataset to increase the accuracy of the model. Furthermore, it also can be used to determine the optimized hyperparameter for models

```
def model_training(X, y, model, sampler = None):
  skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

  fold_accuracies = []

  for fold, (train_idx, test_idx) in enumerate(skf.split(X, y), 1):
      X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
      y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
```

```
Fold 1: Accuracy = 0.85
Fold 2: Accuracy = 0.89
Fold 3: Accuracy = 0.87
Fold 4: Accuracy = 0.88
Fold 5: Accuracy = 0.86
Average Accuracy: 0.87

Best Model is from Fold 2 with Accuracy = 0.89
```

SMOTE

The target variable “Class” distribution shows serious class imbalance where Class 1 is significantly undersampled (4.5%) whereas Class 2 is highly dominant (40.9%). The imbalance of the dataset will skew the predictive model performance by favouring the majority class. To address class imbalance effectively, SMOTE (Synthetic Minority Over-sampling Technique) is applied to achieve an equal proportion for each class. This is an oversampling technique to oversample the minority class before training. SMOTE generates synthetic samples for the minority class by using interpolation between existing minority classes. From the graphs below, we can see after SMOTE is applied on the training dataset, it successfully balances the number of samples across all classes.

```python
from imblearn.over_sampling import SMOTE
import warnings
warnings.filterwarnings("ignore", category = FutureWarning)
smote = SMOTE(random_state=42)
model_training(X, y, rfc, smote)
```

Class	Count	Proportion (%)
1	73	4.5
2	658	40.9
3	592	36.8
4	287	17.8

```python
from sklearn.model_selection import StratifiedKFold
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

fold_accuracies = []
smote = SMOTE(random_state=42)

for fold, (train_idx, test_idx) in enumerate(skf.split(X, y), 1):
      X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
      y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
      X_train, y_train = smote.fit_resample(X_train, y_train)
      print(y_train.value_counts())
```

```
Class
2    526
3    526
4    526
1    526
Name: count, dtype: int64
Class
2    526
3    526
4    526
1    526
Name: count, dtype: int64
Class
2    526
3    526
4    526
1    526
Name: count, dtype: int64
Class
2    527
3    527
4    527
1    527
Name: count, dtype: int64
Class
2    527
3    527
4    527
1    527
```

RandomOverSampler

The RandomOverSampler is an oversampling technique which randomly duplicates the existing samples for minority classes until balance is achieved. Unlike SMOTE, RandomOverSample will not create synthetic data hence cannot increase diversity of

minority class and may lead to overfitting risk. However, it is a straightforward and compatible technique with a wide range of classifiers without requiring special adjustment.

```
from imblearn.over_sampling import RandomOverSampler
oversampler = RandomOverSampler(sampling_strategy = {1: 300})
```

Predictive Model

Several predictive models such as Random Forest Classifier, XGBClassifier, Support Vector Machine(SVM), Decision Tree Classifier are utilized in our analysis. Techniques such as hyperparameter tuning, SMOTE, and RandomOverSampling are applied to enhance the performance of all these models. After comparing these models using various techniques, we concluded that the Random Forest model, combined with RandomOverSampler and hyperparameter tuning, achieves the highest accuracy.

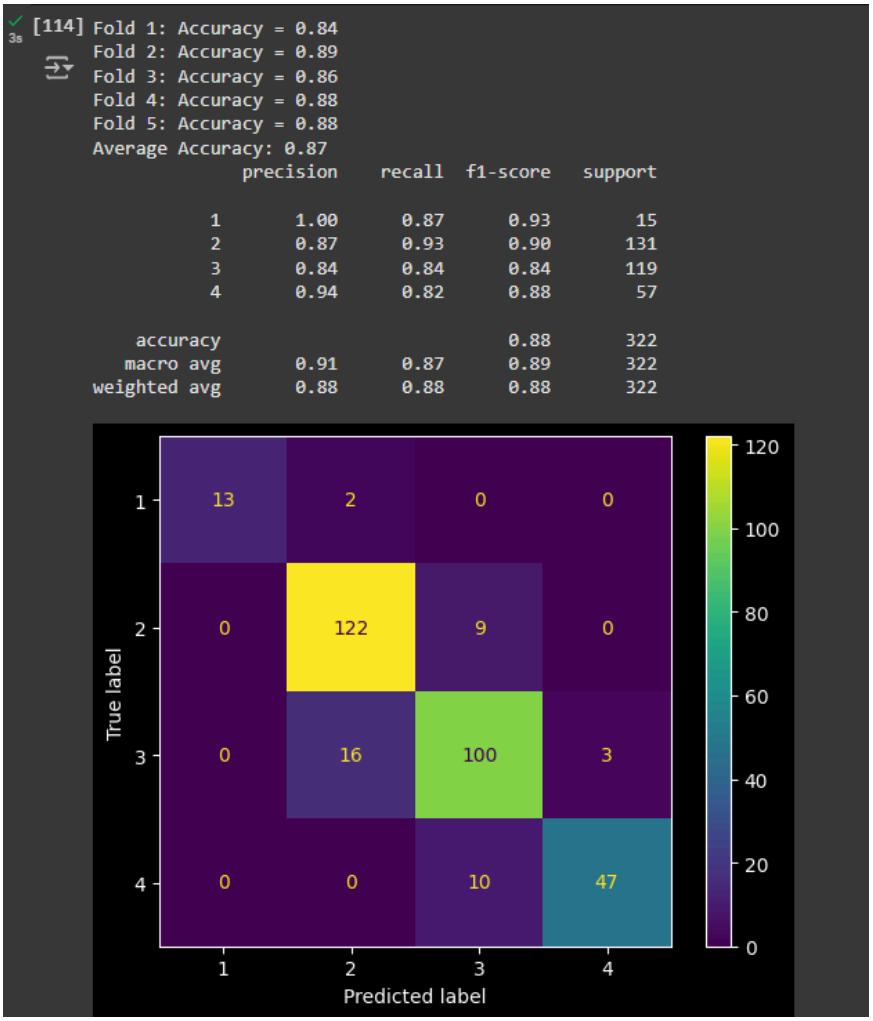
Model Comparison

Predictive Model	Accuracy	
	SMOTE	RandomOverSampler
Random Forest Classifier	0.87	0.88
Best Random Forest Classifier (Tuning)	0.85	0.89
Best Random Forest Classifier (Tuning and One Hot)	0.86	0.89
XGBClassifier	0.86	0.87
Support Vector Machine(SVM)	0.74	0.73
Decision Tree Classifier	0.65	0.66

From the graph above, we can see that the base random forest classifier with both SMOTE and RandomOverSampler techniques have accuracy of **0.87 and 0.88** respectively. After hyperparameter tuning, the performance of this model is improved to an accuracy of **0.89** with a RandomOverSampler technique, which makes it become the best performing model. However, the accuracy of Random Forest Classifiers with the hyperparameter tuning using the SMOTE decreases to **0.85**. We only apply one-hot encoding to the Best Random Forest in order to see if the performance of the best predictive model can improve or not, the accuracy with SMOTE slightly increases **(0.86)** and remains the same for RandomOverSampler. This is because the Random Forest is robust and handles categorical variables well which does not require one hot encoding to help with interpretability and interactions between categorical variables. The XGBClassifier model has a slightly lower performance compared to the Random Forest, with an accuracy of **0.86** using SMOTE and **0.87** with RandomOverSampler. The Support

Vector Machine(SVM) is underperformed compared to the first few models where accuracy with SMOTE is **0.74** and with RandomOverSampler is **0.73.** The Decision Tree Classifier shows the lowest accuracy among all the predictive models which achieve **0.65** with SMOTE and **0.66** with RandomOverSampler. It indicates that the Decision Tree Classifier model is not suitable for our dataset. This outlines the importance of appropriate selection on model, effective preprocessing and optimization technique.

Random OverSampler often performs better than the SMOTE especially in tree-based models such as Random Forest and XGBClassifier. It indicates that a simpler oversampling method is more effective than the complex oversampling method in our case. However, for the model that requires nuanced boundaries like SVM the result will be better by using SMOTE. It suggests that SMOTE may be suitable to enhance models that need nuanced boundaries but RandomOverSampler will be better for tree-based models.



5.0 Data Interpretation

From the previous section, the Random Forest Classifier achieved an accuracy of approximately 88%, making it the most accurate model in this study for predicting obesity

risk. This section delves into the interpretation of the model's results, focusing on feature importance, comparison with statistical insights, and visualization to provide a comprehensive understanding of the model's decisions.

5.1 Feature Importance Analysis

To understand which features contributed most to the model's performance in predicting obesity risk, feature importance scores were generated by the best Random Forest model trained as follow:

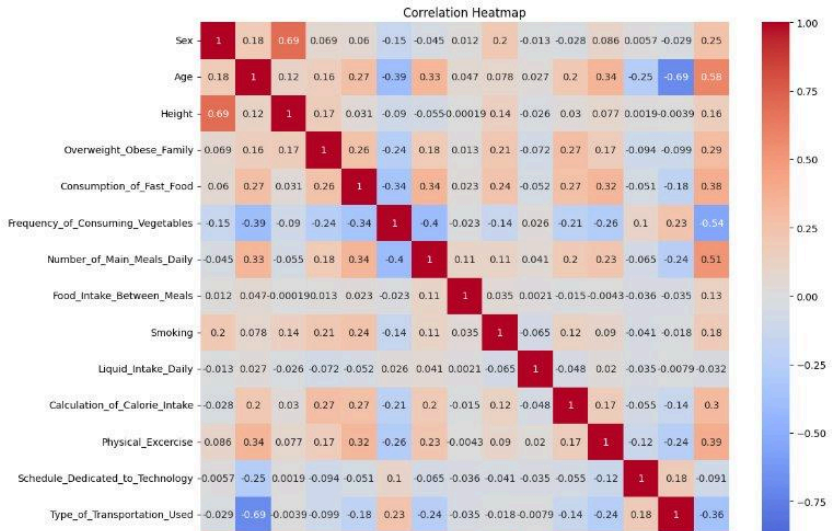
Top 10 Important Features:

	Feature	Importance
1	Age	0.184339
11	Physical_Excercise	0.156542
2	Height	0.105192
5	Frequency_of_Consuming_Vegetables	0.092654
6	Number_of_Main_Meals_Daily	0.086899
13	Type_of_Transportation_Used	0.065210
9	Liquid_Intake_Daily	0.053110
8	Smoking	0.047173
7	Food_Intake_Between_Meals	0.046871
12	Schedule_Dedicated_to_Technology	0.043225

The feature importance analysis identified Age as the most significant factor influencing obesity risk, contributing 18.4% to the model's decisions. This was followed by Physical Exercise (15.7%) and Height (10.5%), emphasizing the importance of physical attributes and activity levels. Additionally, dietary factors such as Frequency of Consuming Vegetables (9.3%) and Number of Main Meals Daily (8.7%) were also among the top contributors, highlighting the critical role of diet in obesity prediction. These top five features provide key insights into the primary drivers of obesity risk in the dataset.

5.2 Comparison with Statistical Results

In this section, we compare the Random Forest model's feature importance with the correlations shown in the heatmap. This analysis highlights consistencies and discrepancies, revealing how the model captures patterns beyond simple statistical relationships to better explain obesity risk.



When compared with the correlation heatmap, Age, Physical Exercise, Type of Transportation Used, and Number of Main Meals Daily from the top 5 correlated features also appear in the model's top 10 important features, showing significant alignment. However, Overweight_Obese_Family, which has the highest correlation with the class, did not rank in the model's top 10. This suggests that the Random Forest model may deprioritize redundant features or capture non-linear patterns involving other predictors.

	Feature	Chi-Square Statistic	P-Value	Degrees of Freedom	Critical Value (5%)	Hypothesis
11	Type_of_Transportation_Used	226.752421	6.965562e-49	4	9.487729	Dependent
2	Consumption_of_Fast_Food	182.299693	2.810739e-39	1	3.841459	Dependent
9	Physical_Excercise	158.233243	4.409566e-34	4	9.487729	Dependent
1	Overweight_Obese_Family	150.585483	1.969996e-32	1	3.841459	Dependent
3	Frequency_of_Consuming_Vegetables	136.917148	1.746329e-29	2	5.991465	Dependent
6	Smoking	128.780698	9.904355e-28	1	3.841459	Dependent
8	Calculation_of_Calorie_Intake	128.444546	1.170204e-27	1	3.841459	Dependent
4	Number_of_Main_Meals_Daily	95.433615	1.489793e-20	2	5.991465	Dependent
0	Sex	87.107677	9.155437e-19	1	3.841459	Dependent
7	Liquid_Intake_Daily	20.387221	1.410942e-04	2	5.991465	Dependent
5	Food_Intake_Between_Meals	15.642558	1.342285e-03	3	7.814728	Dependent
10	Schedule_Dedicated_to_Technology	5.654038	1.297103e-01	2	5.991465	Independent

A brief comparison with the chi-square test results further supports these findings. Features like Physical Exercise, Type of Transportation Used, and Number of Main Meals Daily ranked high in both chi-square significance and feature importance, indicating their strong dependence on the target class. However, Consumption of Fast Food and Overweight Obese Family, which ranked highly in the chi-square test, were less prominent in the model's feature importance. This discrepancy highlights how the model may capture complex interactions or redundancies that statistical methods like chi-square do not account for.

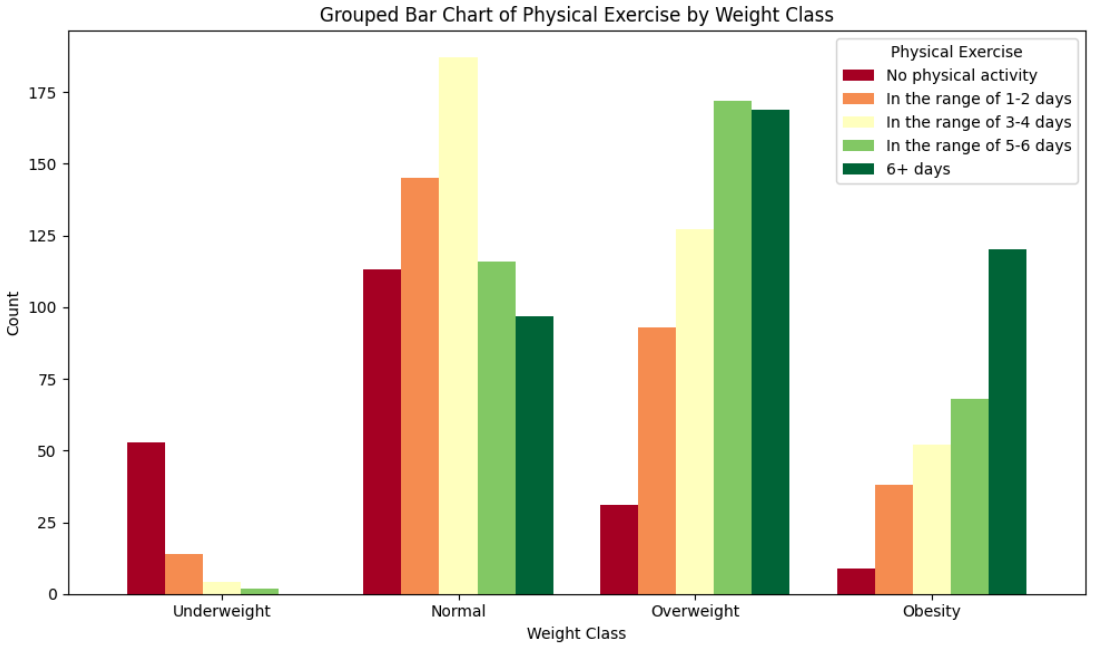
5.3 Visualization of Key Features

In this section, we examine the top 5 features identified by the Random Forest model as the most significant predictors of obesity risk. Through targeted visualizations, we analyze their distributions across weight classes and their relationship with the target variable, offering deeper insights into the patterns and trends influencing obesity risk.

5.3.1 Age

The box plot illustrates the distribution of Age across the four weight classes: Underweight, Normal, Overweight, and Obesity. The median age increases progressively from the Underweight class to the Obesity class, indicating a potential trend where individuals in higher weight categories tend to be older. The spread of ages is wider in the Overweight and Obesity classes compared to the Underweight and Normal classes, suggesting greater variability in age for these categories. Outliers are observed in the Underweight and Normal classes, indicating a few individuals who deviate significantly from the typical age range for these groups. This visualization reinforces the importance of age as a significant predictor in the obesity risk model.

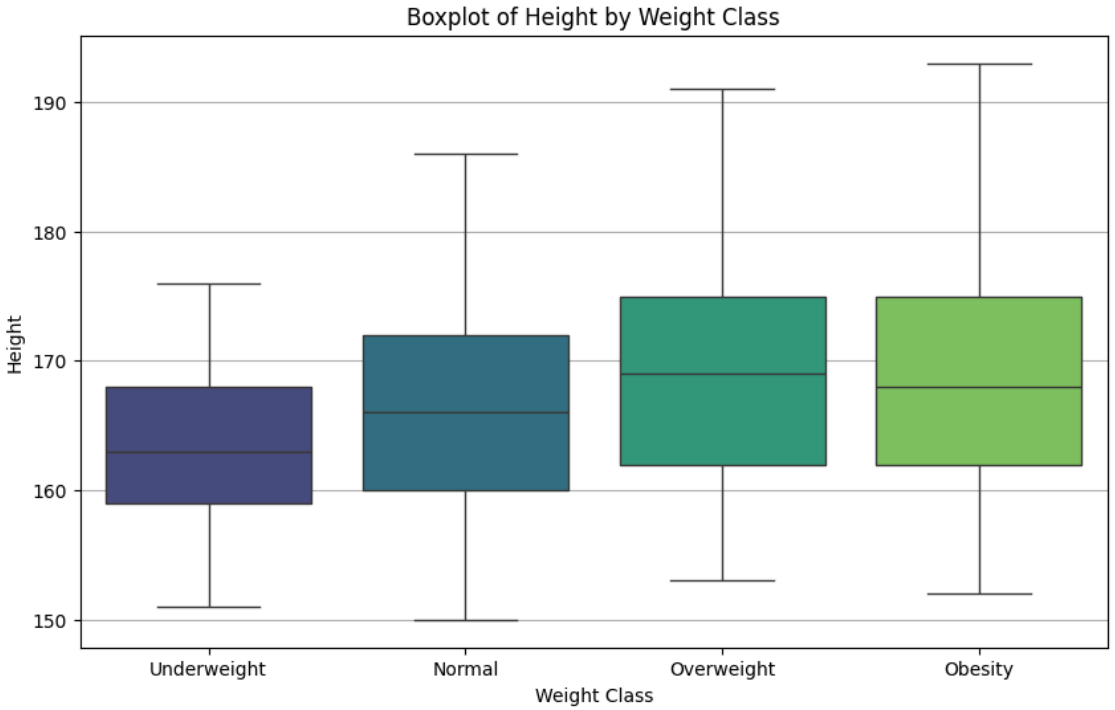
5.3.2 Physical Exercise



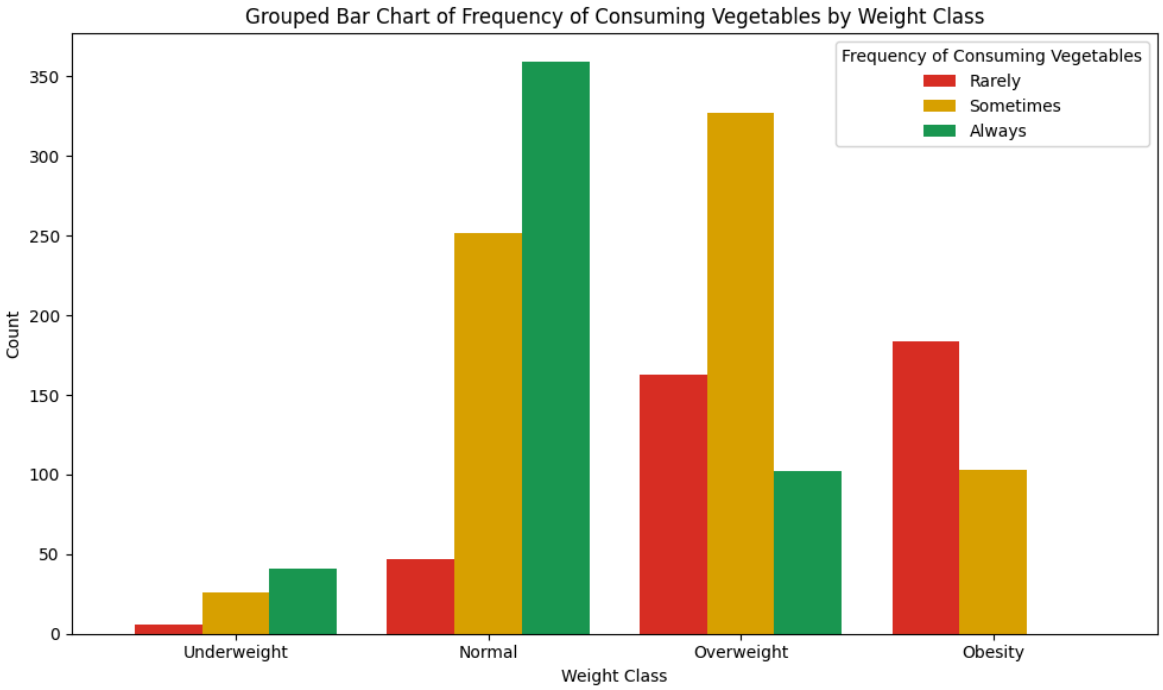
The grouped bar chart shows an unexpected trend where individuals in heavier weight classes, such as Overweight and Obesity, report engaging in physical exercise more frequently compared to those in the Underweight and Normal weight classes. This pattern could reflect several factors. First, individuals in higher weight categories may be more likely to engage in regular exercise as part of weight management efforts or following health advice. Additionally, the data may include a self-reporting bias, where individuals overestimate their activity levels, particularly if they perceive exercise as a desirable behavior. It's also possible that other factors, such as dietary habits or metabolic differences, play a significant role in weight, regardless of exercise frequency. Furthermore, the metric may not differentiate between light and intense exercise, with heavier individuals potentially engaging in more frequent but less strenuous activities. These factors combined suggest the need for further analysis to fully understand the relationship between physical activity and weight class.

5.3.3 Height

The boxplot shows the distribution of Height across the four weight classes. Individuals in the Underweight class generally have shorter heights, with the median height being notably lower than in other classes. As height increases, there is a trend toward the Normal, Overweight, and Obesity classes, with these classes showing higher medians and a broader range of values. The Normal and Overweight classes exhibit slightly wider interquartile ranges, while the Obesity class has a consistent distribution similar to the Overweight group. This pattern suggests that height, while influential, interacts with other factors to determine weight classification.



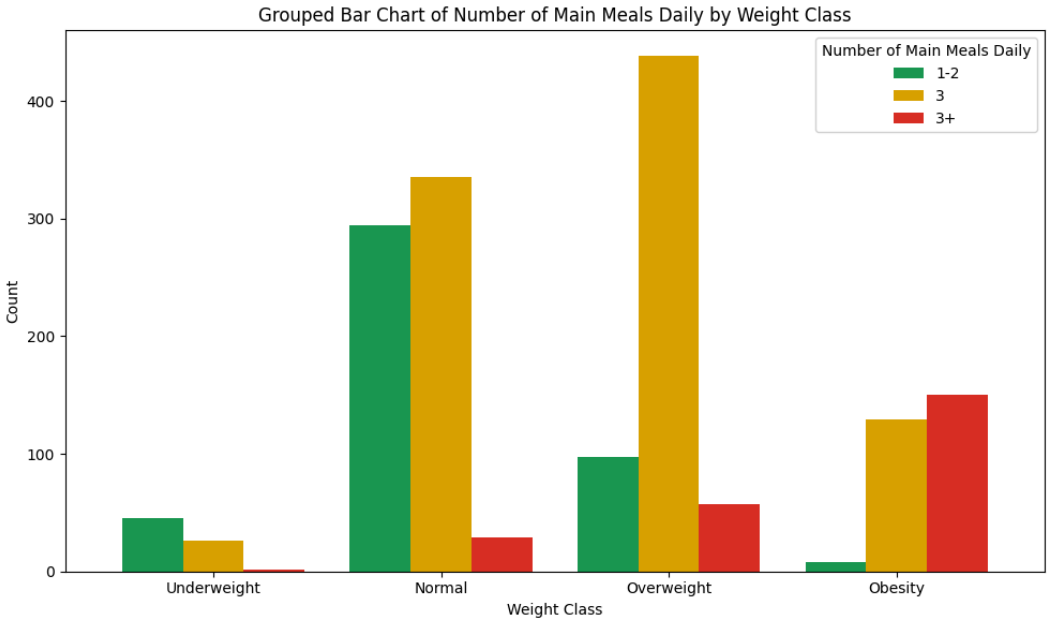
5.3.4 Frequency of Consuming Vegetables



The grouped bar chart on the frequency of consuming vegetables provides additional context to validate the possibility that dietary habits play a significant role in the observed

trends for physical exercise across weight classes. While individuals in heavier weight classes engage in more frequent physical exercise, the chart reveals that these groups are also less likely to consume vegetables regularly. This imbalance suggests that despite higher exercise levels, dietary habits—such as infrequent vegetable consumption—may still contribute to weight gain or difficulty in achieving weight loss. This reinforces the need to consider both exercise and diet as critical factors in managing weight effectively.

5.3.5 Number of Main Meals Daily



The grouped bar chart illustrates the distribution of the Number of Main Meals Daily across different weight classes. Individuals in the Normal and Overweight classes predominantly consume 3 meals daily, as represented by the tall yellow bars. Meanwhile, those in the Underweight and Obesity classes show a more varied pattern, with notable proportions consuming either fewer meals (1-2, green) or more than 3 meals (3+, red). This trend suggests that irregular meal patterns, whether consuming too few or too many meals, could be associated with underweight or obesity. In contrast, maintaining a regular pattern of 3 meals daily appears more common among individuals with healthier weight categories. This reinforces the significance of meal regularity in managing weight effectively.

6.0 Plan for Reproducible Research

Ensuring reproducibility is a critical aspect of this project, as it enhances the credibility and reliability of the research findings. The following parts outline the strategies employed to maintain reproducibility throughout the study. The project is accessible via the link: https://github.com/yukisim425/Obesity-Risk-Prediction.

6.1 Research Documentation

Comprehensive documentation throughout the project is maintained in a pdf file to ensure reproducibility. Besides, 'README.md' file in github serves as the central documentation hub, describing the project objectives, dataset details, pre-processing steps, modeling techniques, and

key outcomes. This documentation provides step-by-step instructions for setting up the environment, executing the analysis, and understanding the outputs. Additionally, each code script and Jupyter notebook will include detailed comments and markdown cells to explain the purpose and function of each step in the workflow.

6.2 Dataset Accessibility and Management

The dataset used in this study is publicly available on Kaggle (Koklu & Sulak, 2024). A link to the dataset is included in the project documentation, along with a description of the data and a summary of its attributes. Metadata files were created to document the origin, structure, and pre-processing steps applied to the dataset. The raw dataset will remain unaltered to serve as a baseline, while pre-processing scripts will generate processed data files with clear naming conventions. Steps such as handling missing values, detecting outliers, and transforming features are meticulously documented in Jupyter notebooks to ensure they can be reproduced. Version control will be applied to data-related files to track modifications effectively.

6.3 Computational Environment

To standardize the computational environment, Python is used as the main programming language. All required libraries and dependencies, including those for data analysis (e.g., Pandas, NumPy), visualization (e.g., Matplotlib, Seaborn), and machine learning (e.g., Scikit-learn, XGBoost), will be specified in a 'requirements.txt' file. A detailed guide named 'Environment Setup Guide' is provided to recreate the environment using venv or conda. Documentation of the software versions and operating system used can ensure compatibility for future replication efforts.

6.4 Code Organization and Workflow

The project files are organized into structured directories, including /scripts for modular Python scripts, /notebooks for Jupyter notebooks, and /data for raw data. Modular programming practices can be employed to separate tasks such as data cleaning, exploratory data analysis (EDA), model training, and evaluation. Automated workflows using tools like Python scripts or Makefile can be developed for repetitive processes, such as dataset pre-processing and model evaluation, to minimize errors and ensure consistency.

6.5 Reproducible Analysis

All analysis, from data exploration to model development, are documented in Jupyter notebooks. These notebooks combine code cells with markdown explanations to provide a step-by-step account of the methodology. To handle randomness in machine learning tasks, such as data splitting and model initialization, fixed random seeds will be applied. The entire analysis, including pre-processing, feature selection, modelling, hyperparameter tuning, will be reproducible using the provided code and instructions.

6.6 Version Control

Version control is implemented using Git, with the project repository hosted on GitHub. A .gitignore file will exclude unnecessary or large files, such as raw datasets and temporary outputs. Frequent commits with descriptive messages will document the progression of the project. To manage different components of the project, such as feature engineering and model experimentation, separate branches will be created. These branches will be merged into the main branch upon completion to maintain a clear development history.

6.7 Sharing and Accessibility

Upon completion, the project repository is made publicly accessible, excluding any sensitive or proprietary data to maintain ethical compliance. The repository includes all scripts, notebooks, and documentation required to replicate the analysis. A user-friendly guide is provided to help others execute the code and understand the workflow. Additionally, the trained Random Forest model, serialized using the pickle library, and the Flask application code is included to demonstrate the deployment process.

This plan ensures transparency, reproducibility, and accessibility, enabling other researchers and stakeholders to replicate the study and extend its findings.

7.0 Deployment of Data Product

This project aims to develop a data product that predicts the likelihood of an individual developing obesity based on various lifestyle factors. After selecting the best-performing model (Random Forest Classifier) from the evaluation stage, the model was exported using Python's 'pickle' library. This is necessary to save the trained model to a file so it can be reused without retraining. This code saves the trained model to a file named 'best_rfc.pkl'. The 'wb' mode is used to open the file in write-binary mode, and 'pickle.dump()' exports the model object into the file.

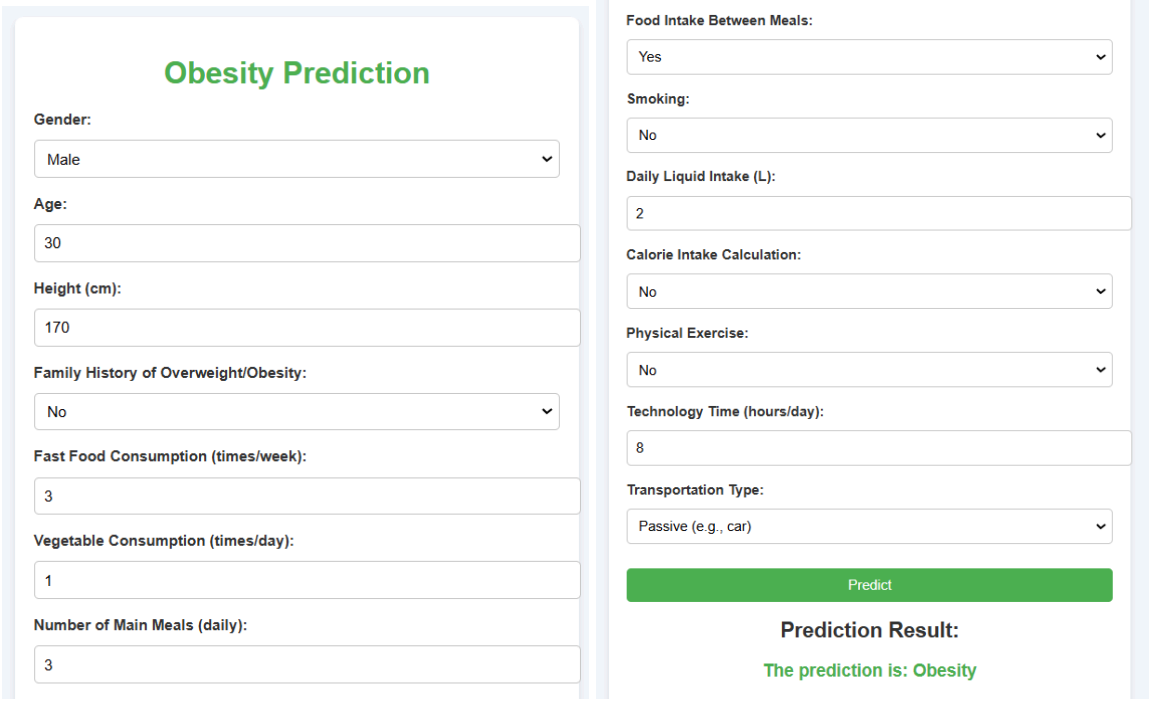
```
import joblib
import pickle

# Define file paths
joblib_model_path = "/content/drive/My Drive/WQD7001 Group 10/best_rfc.joblib"
pickle_model_path = "/content/drive/My Drive/WQD7001 Group 10/best_rfc.pkl"

# Step 1: Load the model using joblib
model = joblib.load(joblib_model_path)

# Step 2: Save the model using pickle
with open(pickle_model_path, 'wb') as file:
    pickle.dump(model, file)
```

A Flask application is set up to deploy the model. A Flask application is a web application built using the Flask framework in Python. The model is loaded into memory using pickle when the app starts. The Flask application will be running on a local machine using any browser. The application will allow users to enter their details, which are then passed to the Flask app for prediction. The result is displayed on the same page.

app.py > ...

```python
from flask import Flask, request, render_template
import pandas as pd
import pickle
import numpy as np

app = Flask(__name__)

# Load the trained RFC model
with open('best_rfc.pkl', 'rb') as file:
    model = pickle.load(file)

# Define the expected feature names for the model
model_features = [
    'Sex', 'Age', 'Height', 'Overweight_Obese_Family',
    'Consumption_of_Fast_Food', 'Frequency_of_Consuming_Vegetables',
    'Number_of_Main_Meals_Daily', 'Food_Intake_Between_Meals', 'Smoking',
    'Liquid_Intake_Daily', 'Calculation_of_Calorie_Intake',
    'Physical_Excercise', 'Schedule_Dedicated_to_Technology',
    'Type_of_Transportation_Used'
]

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Collect input data from the form
        input_data = {
            "Sex": request.form['Sex'],
            "Age": float(request.form['Age']),
            "Height": float(request.form['Height']),
```

The application collects user input, processes the data, and uses the trained model to make predictions in real-time. The Flask application was tested by providing various inputs and checking if the predictions were accurate. The model was able to correctly classify individuals into categories like "Underweight", "Normal", "Overweight", and "Obesity".



This data product of obesity prediction based on individual lifestyle can be a powerful tool for health prevention and awareness. Individuals can receive early warnings about behaviours that may increase their risk of obesity. With this information, they can take proactive steps to modify their lifestyle to prevent obesity before it becomes a serious health issue.

8.0 Insights and Conclusion

Our data product can be used to identify the significant risk factors of obesity to help in diagnosing obesity based on patients' current health condition or medical history. With the prediction model, we can assess early detection and prevention of obesity, allowing high risk individuals to take preventive measures and modify current unhealthy lifestyles, therefore reducing the severity of obesity and its complications. When risk factors of obesity are identified, patients can receive better, personalized treatments and policy makers can utilize the information to optimize resource allocation for public healthcare interventions and for research and development of better technology to treat obesity patients.

9.0 References

Koklu, N., & Sulak, S. A. (2024). Using Artificial Intelligence Techniques for the Analysis of Obesity Status According to the Individuals' Social and Physical Activities. Sinop Üniversitesi Fen Bilimleri Dergisi, 9(1), 217-239.

Appendices

ipynb?