Flab or Fab

Table of Contents

1. Project Title and Team Members: 3
1.1 Project Title: Flab or Fab? 3
1.2 Team Members 3
2. Goal and Objectives 3
2.1 Abstract 3
2.2 Motivation 3
2.3 Significance 4
2.4 Objectives 4
2.5 Features 4
2.5.1 Model 5
2.5.2 Attributes 7
2.5.3 Visualizations 8
3. Project Increment-II 8
3.1 Introduction 8
3.1.1 Domain 8
3.1.2 Workflow Diagram 8
3.2 Background 9
3.3 Data Abstraction 10
3.3.1 Dataset 10
3.3.2 Types and Attributes 10
3.3.3 Detail Design of features with diagram 14
3.3.4 Data Transformation 17
3.4 Task Abstraction 19
3.4.1 Target 19
3.4.2 Actions 20
3.5 Implementation Using Tools 22
3.5.1 Flask Architecture 22
3.5.2 Python libraries: Pandas, NumPy, Math, Joblib. 24
3.5.3 Use of different tools in the Project: 30
3.5.4 Project Run Instructions 31

3.6 Preliminary Results for Analysis 31
3.6.1 Visualization Graphs 31
3.6 Story Telling 71
3.6.1 Who, what, where? 71
3.6.2 What, When, How, and Why? 72
3.6.3 Data 74
3.6.4 Users 75
3.7 Project Management 76
3.7.1 Implementation status report 76
4. References 77

1. Project Title and Team Members:

1.1 Project Title: Flab or Fab?

1.2 Team Members

2. Neha Goud Baddam (https://github.com/nehabaddam)
3. Panduga Raja Tejasvi Prasad (https://github.com/tejasvi06)
4. Yasmeen Haleem (https://github.com/yasmeenha)
5. Siddhartha Kundaram (https://github.com/siddu7147)

GitHub Project Repository Link:

https://github.com/nehabaddam/SDV-Project.git

2. Goal and Objectives

2.1 Abstract

AI is rapidly integrating into various aspects of our world, contributing to its improvement. We hope to create a web application "Flab or Fab" which can serve as a preventative measure by providing users with insights into their health and lifestyle habits based on user input such as age, weight, height, and other lifestyle habits. In addition, the tool will have the option to redirect users to a page that visually demonstrates how these factors can contribute to obesity and help improve health literacy and encourage users to adopt healthy habits.

2.2 Motivation

Given the rising incidence of diseases associated with obesity resulting from sedentary lifestyles and unhealthy habits, it is crucial to address this issue by providing people with AI-based tools to assess their health from the comfort of their homes without the need for hospital visits.

Furthermore, “Flab or Fab" can assist consumers in better grasping the complicated connection between lifestyle choices and health outcomes by illustrating the components that contribute to obesity. Individuals may be able to make beneficial changes in their lives as a result, lowering their risk of acquiring obesity-related health issues. Finally, “Flab or Fab" demonstrates AI's potential to change healthcare by providing individualized and accessible solutions for better health outcomes.

2.3 Significance

Obesity is a chronic disease with several causes that lead to excessive body fat accumulation. And obesity is linked to several serious life-threatening diseases like diabetes, hypertension, osteoarthritis, knee pain, hip issues, depression, mental health, and cancer which reduce the life expectancy in humans.[5]

Worldwide, obesity has nearly tripled in the last 50 years. The United States and Mexico have large numbers of teens and middle-aged people who are obese and thus leading to serious life-threatening diseases. Japan has taken decisive steps to ensure that the obesity level is at the lowest levels and thus Japan has successfully achieved the highest life expectancy of 84.3 years.[4]

Obesity can be easily controlled by educating the public and creating awareness about the advantages of maintaining a healthy weight by adopting healthy food habits, and daily importance of exercise and regular mindful eating, and adopting other methods like Meditation to reduce Stress rather than resorting to eating high-calorie junk food.

Though Obesity is a huge problem, there are no web applications that predict obesity levels and types. Many BMI calculators just calculate the Body Mass Index by taking in the height and weight of the user. For example, the Centre for Disease Control has developed an Adult BMI calculator, which does not predict obesity type.[2] Another such example is National Heath, Lung, and Blood Institutes Calculate Your Body Mass Index. [1]

So, with the help of Fab or Fab, they can determine their obesity score and can take important steps in reducing their obesity and maintaining a healthy weight and thus protecting themselves from serious life-threatening diseases and thus increasing their life expectancy.

2.4 Objectives

The primary aim of "Flab or Fab" is to gather information from users about the factors that contribute to obesity through a series of basic questions. This information is then used to calculate the user's risk of developing heart disease. Even if an individual is currently in good health, taking the assessment provided by "Flab or Fab" can help them determine their obesity score and make positive changes to their lifestyle to maintain or improve their health.

2.5 Features

The primary goal of "Flab or Fab" is to accurately predict an individual's obesity score. This is achieved by training the model using existing data and validating and testing it with the latest data available. Multiple models are tested to determine the one that produces the highest level of accuracy, which is then used to predict the risk factor for obesity. Once the model is trained, the website can be hosted to allow users to input their information dynamically, and the obesity score will be produced as output. Additionally, the tool will use visualization techniques to demonstrate how various factors contribute to obesity.

2.5.1 MODEL

We are using Random Forest Classifier for prediction.
Below are the model parameters we have chosen.

```
df_cleaned = df

df_scaled=df_cleaned.copy()

columns_to_scale = ['Gender','Age','Height','Weight','family_history_with_overweight','FAVC','FCVC','NCP','CAEC','SMOKE','CH2O','

scaler = StandardScaler()
df_scaled[columns_to_scale] = scaler.fit_transform(df_cleaned[columns_to_scale])

df_scaled.head()

#we perform some Standardization using minmaxscaler
df_scaled_mm=df_cleaned.copy()

columns_to_scale_mm = ['Gender','Age','Height','Weight','family_history_with_overweight','FAVC','FCVC','NCP','CAEC','SMOKE','CH2(

mmscaler = MinMaxScaler()
df_scaled_mm[columns_to_scale_mm] = mmscaler.fit_transform(df_cleaned[columns_to_scale_mm])

df_scaled_mm.head()


X = df_cleaned.drop(['NObeyesdad'], axis=1) #features
y = df_cleaned['NObeyesdad']  #target feature

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle = True)


# Hyperparameter Tuning using GridSearchCV
param_grid = {
    'n_estimators': [500, 1000, 1500],
    'max_depth': [10, 20, 30],
    'max_features': ['sqrt', 'log2'],
    'max_leaf_nodes': [100, 500, 1000]
}
rf = RandomForestClassifier()
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
best_params = grid_search.best_params_
print("Best parameters:", best_params)
```

```
Best parameters: {'max_depth': 20, 'max_features': 'log2', 'max_leaf_nodes': 500, 'n_estimators': 500}
```

After using the above best parameters, we have an accuracy of 95%.

```
Accuracy: 0.9527186761229315
Confusion Matrix:
 [[56  6  0  0  0  0  0]
 [ 4 50  2  0  0  0  0]
 [ 0  3 46  1  0  0  0]
 [ 0  0  0 76  2  0  0]
 [ 0  0  0  1 57  0  0]
 [ 0  0  0  0  0 63  0]
 [ 1  0  0  0  0  0 55]]
True Positive Rate: 0.9259259259259259
Classification Report:
               precision    recall  f1-score   support

           1       0.92      0.90      0.91        62
           2       0.85      0.89      0.87        56
           3       0.96      0.92      0.94        50
           4       0.97      0.97      0.97        78
           5       0.97      0.98      0.97        58
           6       1.00      1.00      1.00        63
           7       1.00      0.98      0.99        56

    accuracy                           0.95       423
   macro avg       0.95      0.95      0.95       423
weighted avg       0.95      0.95      0.95       423
```

2.5.2 ATTRIBUTES

Input:

We are to consider below inputs:

1. Gender (Female, Male)
2. Age (numeric)
3. Height (numeric)
4. Weight (numeric)
5. family_history_with_overweight (Yes/No)
6. FAVC: Whether the individual consumes high-calorie food frequently (Yes/No)
7. FCVC: Frequency of consumption of vegetables by the individual weekly (numeric).
8. NCP: Number of main meals consumed by the individual per day(numeric).
9. CAEC: Consumption of food between main meals (Yes/No).
10. SMOKE: Whether the individual smokes tobacco (Yes/No).
11. CH2O: Water consumption of the individual in liters per day (numeric)
12. SCC: Whether the individual monitors their calorie intake (Yes/No)
13. FAF: Physical activity level of the individual (numeric).
14. TUE: Time spent by the individual on sedentary activities per day in hours(numeric).
15. CALC: Consumption of alcohol by the individual (Yes/No).
16. MTRANS: Mode of transportation used by the individual. (Automobile, Motorbike, Bike, Public_Transportation, Walking)
17. NObeyesdad: Type of Obesity
18. (Insufficient_Weight, Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_Type_I, Obesity_Type_II, Obesity_Type_III)

Output:

A calculated obesity type and visual representation of how inputs contribute to obesity.

2.5.3 VISUALIZATIONS

We are using many different types of charts to visualize data.

1. Visualizing Obesity concerning factors affecting obesity.
2. Visualizing the highest, lowest, and regional obesity numbers.
3. Visualizing Obesity concerning gender.

3. PROJECT INCREMENT-II

3.1 INTRODUCTION

3.1.1 DOMAIN

Healthcare or public health is the domain of an obesity prediction project. This is because obesity is a medical condition that can cause several health issues, including high blood pressure, diabetes, and heart disease. To decrease the prevalence of obesity and the related health hazards, healthcare practitioners and public health officials can establish preventative and intervention measures with the use of obesity prediction. The project also involves gathering and evaluating health data, which calls for training in healthcare or public health.

The domain for an obesity-type prediction project typically involves analyzing various factors such as body mass index (BMI), body fat distribution, genetics, lifestyle choices, and medical history.

The following are some potential domains for an obesity-type prediction:

1. Healthcare: Predicting the type of obese individuals would allow for more individualized treatment strategies and better health results.

2. Medical research: Examining the connection between various forms of obesity and the dangers that they bring with them.

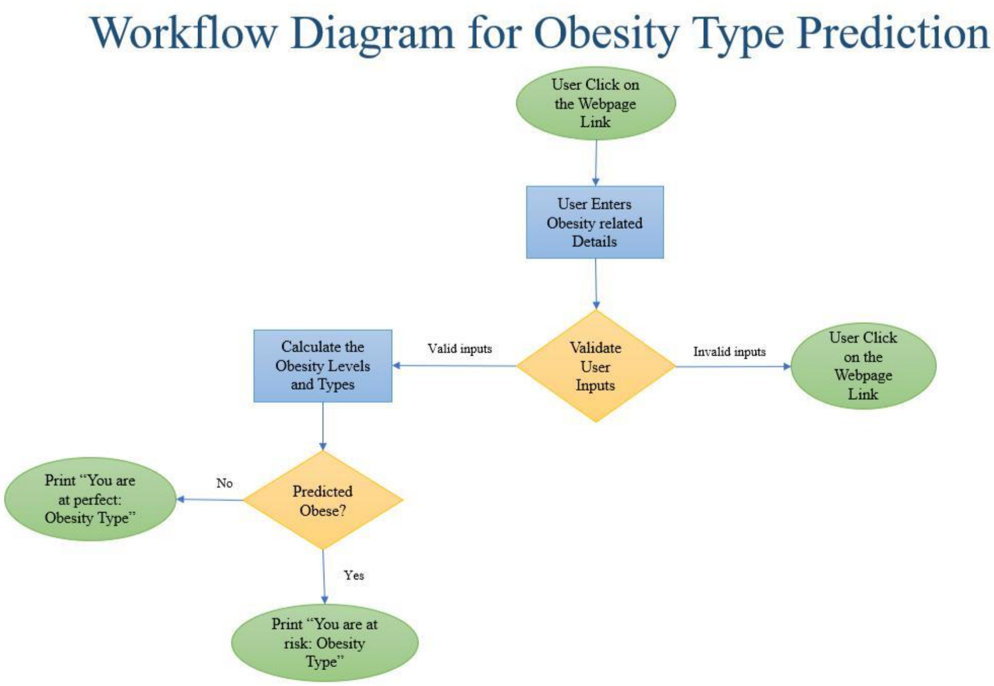
3. Nutrition: Determining dietary habits, nutrient intake, and various types of obesity.

4. Fitness: Determining the type of obesity in people based on their levels of physical activity and creating

5. Technology: Creating algorithms and models that use data from wearables, mobile apps, and other health data sources to predict the kind of obesity.

3.1.2 WORKFLOW DIAGRAM

Ongoing research in the medical field has aimed to pinpoint the risk factors associated with obesity. Some factors that have been identified include genetics, overeating, a sedentary lifestyle, an unhealthy diet, medical conditions, medications, sleep deprivation, and stress. Despite progress in understanding these factors, there is still room for improvement in accurately predicting the risk of obesity and informing individuals about their health. [3] The "Flab or Fab" web application utilizes previous and current research data on these factors to predict the type of obesity using AI algorithms and present the risk information to users in an easily understandable visual format.



3.2 BACKGROUND

During the research for the obesity-related data and projects we came across the World Heart Federation[7] and which has cited that obesity is one of the rapidly increasing health problems in the world, that the most important cause leadinglife-threateningning diseases like heart attack, diabetes, and which further leads to a reduced life expectancy among the adult population. The World Heart Foundation is also mainly focused on educating and making people more aware of the serious consequences of obesity and helping the world population to track and reduce the obesity percentages in their own countries.

Obesity is a chronic disease with several causes that lead to excessive body fat accumulation. And obesity is linked to several serious life-threatening diseases like diabetes, hypertension, osteoarthritis, knee pain, hip issues, depression, mental health, and cancer which reduce the life expectancy in humans.[5]

Worldwide, obesity has nearly tripled in the last 50 years. The United States and Mexico have large numbers of teens and middle-aged people who are obese and thus leading to serious life-threatening diseases. Japan has taken decisive steps to ensure that the obesity level is at the lowest levels and thus Japan has successfully achieved the highest life expectancy of 84.3 years.[4]

Obesity can be easily controlled by educating the public and creating awareness about the advantages of maintaining a healthy weight by adopting healthy food habits, and daily importance of exercise and regular mindful eating, and adopting other methods like Meditation to reduce Stress rather than resorting to eating high-calorie junk food.

Though Obesity is a huge problem, there are no web applications that predict obesity levels and types. Many BMI calculators just calculate the Body Mass Index by taking in the height and weight of the user. For example, the Centre for Disease Control has developed an Adult BMI calculator, which does not predict obesity type.[2] Another such example is National Heath, Lung, and Blood Institutes Calculate Your Body Mass Index. [1]

So, with the help of Fab or Fab, they can determine their obesity score and can take important steps in reducing their obesity and maintaining a healthy weight and thus protecting themselves from serious life-threatening diseases and thus increasing their life expectancy.

3.3 Data Abstraction

Data abstraction deals with the steps and methods that can be taken to simplify the process of understanding the data, extracting meaningful insights from the data, and classifying the information in the data in tables or charts so that they can be easily understood by the end user.

3.3.1 Dataset

For this project, we are using 5 datasets to create dynamic and interactive data visualization to understand obesity in different countries around the world.
The 5 datasets that are to be used are in CSV format and they are named obesity, obesity_data, world, obesity_country, and world_population.

3.3.2 Types and Attributes

The five datasets that have been used in the project namely obesity, obesity_data, world, obesity_country, and world_population are further explained in detail below along with different types, columns, and attributes in each dataset.

The first dataset being used is obesity as shown below.

1	Gender	Age	Height	Weight	family_his	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
2	Female	21	1.62	64	yes	no	2	3	Sometime	no	2	no	0	1	no	Public_Tra	Normal_Weight
3	Female	21	1.52	56	yes	no	3	3	Sometime	yes	3	yes	3	0	Sometime	Public_Tra	Normal_Weight
4	Male	23	1.8	77	yes	no	2	3	Sometime	no	2	no	2	1	Frequently	Public_Tra	Normal_Weight
5	Male	27	1.8	87	no	no	3	3	Sometime	no	2	no	2	0	Frequently	Walking	Overweight_Level_I
6	Male	22	1.78	89.8	no	no	2	1	Sometime	no	2	no	0	0	Sometime	Public_Tra	Overweight_Level_II
7	Male	29	1.62	53	no	yes	2	3	Sometime	no	2	no	0	0	Sometime	Automobil	Normal_Weight
8	Female	23	1.5	55	yes	yes	3	3	Sometime	no	2	no	1	0	Sometime	Motorbike	Normal_Weight
9	Male	22	1.64	53	no	no	2	3	Sometime	no	2	no	3	0	Sometime	Public_Tra	Normal_Weight
10	Male	24	1.78	64	yes	yes	3	3	Sometime	no	2	no	1	1	Frequently	Public_Tra	Normal_Weight
11	Male	22	1.72	68	yes	yes	2	3	Sometime	no	2	no	1	1	no	Public_Tra	Normal_Weight
12	Male	26	1.85	105	yes	yes	3	3	Frequently	no	3	no	2	2	Sometime	Public_Tra	Obesity_Type_I
13	Female	21	1.72	80	yes	yes	2	3	Frequently	no	2	yes	2	1	Sometime	Public_Tra	Overweight_Level_II
14	Male	22	1.65	56	no	no	3	3	Sometime	no	3	no	2	0	Sometime	Public_Tra	Normal_Weight
15	Male	41	1.8	99	no	yes	2	3	Sometime	no	2	no	2	1	Frequently	Automobil	Obesity_Type_I
16	Male	23	1.77	60	yes	yes	3	1	Sometime	no	1	no	1	1	Sometime	Public_Tra	Normal_Weight
17	Female	22	1.7	66	yes	no	3	3	Always	no	2	yes	2	1	Sometime	Public_Tra	Normal_Weight
18	Male	27	1.93	102	yes	yes	2	1	Sometime	no	1	no	1	0	Sometime	Public_Tra	Overweight_Level_II
19	Female	29	1.53	78	no	yes	2	1	Sometime	no	2	no	0	0	no	Automobil	Obesity_Type_I
20	Female	30	1.71	82	yes	yes	3	4	Frequently	yes	1	no	0	0	no	Automobil	Overweight_Level_II
21	Female	23	1.65	70	yes	no	2	1	Sometime	no	2	no	0	0	Sometime	Public_Tra	Overweight_Level_I
22	Male	22	1.65	80	yes	no	2	3	Sometime	no	2	no	3	2	no	Walking	Overweight_Level_II
23	Female	52	1.69	87	yes	yes	3	1	Sometime	yes	2	no	0	0	no	Automobil	Obesity_Type_I
24	Female	22	1.65	60	yes	yes	3	3	Sometime	no	2	no	1	0	Sometime	Automobil	Normal_Weight
25	Female	22	1.6	82	yes	yes	1	1	Sometime	no	2	no	0	2	Sometime	Public_Tra	Obesity_Type_I

As shown above, the obesity dataset consists of 17 columns and 2112 rows of data. The data consists of both quantitative and categorical data. The different columns that are used in the dataset are Gender, Height, Weight, family_history, FAVC, FCVC NCP, Smoke, CH2O, SCC, FAF, TUE, CALC, MTRANS, NObeyesdad. So, the Gender column is categorical having either male or female, family_history, FAVC, Smoke, SCC. Are also categorical with only 2 outputs either yes or no, the remaining columns are quantitative. The two columns CAEC and CALC are also categorical with 3 values Always, Never, and Sometimes.

The second dataset that we are using is shown below which is obesity_data.

1	Gender	Age	Height	Weight	family_his	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
2	0	21	1.62	64	1	1	2	3	1	1	2	1	0	1	1	1	1
3	0	21	1.52	56	1	1	3	3	1	1	3	1	3	0	1	1	1
4	1	23	1.8	77	1	1	2	3	1	1	2	1	2	1	2	1	1
5	1	27	1.8	87	1	1	3	3	1	1	2	1	2	0	2	5	2
6	1	22	1.78	89.8	1	1	2	1	1	1	2	1	0	0	1	1	3
7	1	29	1.62	53	1	1	2	3	1	1	2	1	0	0	1	2	1
8	0	23	1.5	55	1	1	3	3	1	1	2	1	1	0	1	3	1
9	1	22	1.64	53	1	1	2	3	1	1	2	1	3	0	1	1	1
10	1	24	1.78	64	1	1	3	3	1	1	2	1	1	1	2	1	1
11	1	22	1.72	68	1	1	2	3	1	1	2	1	1	1	1	1	1
12	1	26	1.85	105	1	1	3	3	2	1	3	1	2	2	1	1	4
13	0	21	1.72	80	1	1	2	3	2	1	2	1	2	1	1	1	3
14	1	22	1.65	56	1	1	3	3	1	1	3	1	2	0	1	1	1
15	1	41	1.8	99	1	1	2	3	1	1	2	1	2	1	2	2	4
16	1	23	1.77	60	1	1	3	1	1	1	1	1	1	1	1	1	1
17	0	22	1.7	66	1	1	3	3	3	1	2	1	2	1	1	1	1
18	1	27	1.93	102	1	1	2	1	1	1	1	1	1	0	1	1	3
19	0	29	1.53	78	1	1	2	1	1	1	2	1	0	0	1	2	4
20	0	30	1.71	82	1	1	3	4	2	1	1	1	0	0	1	2	3
21	0	23	1.65	70	1	1	2	1	1	1	2	1	0	0	1	1	2
22	1	22	1.65	80	1	1	2	3	1	1	2	1	3	2	1	5	3
23	0	52	1.69	87	1	1	3	1	1	1	2	1	0	0	1	2	4
24	0	22	1.65	60	1	1	3	3	1	1	2	1	1	0	1	2	1

This dataset consists of 17 columns and 2112 rows of data. The different columns that are used in the dataset are Gender, Height, Weight, family_history, FAVC, FCVC NCP, Smoke, CH2O, SCC, FAF, TUE, CALC, MTRANS, NObeyesdad. The columns Gender, family_history, FAVC, Smoke, and SCC have values of either 0 or 1 and the remaining columns have quantitative values. This data set consists of numerical values, so it is easy to make graphs and charts with it. The details in the obesity_data dataset are about the different genders their weight, height, family history, hereditary habits like smoking, and their food eating patterns ncp, favc, fcvc like if they eat high-calorie foods if they eat vegetables and the physical activity done by each person and the mode of transportation taken by everyone. The third dataset being used for the Project is Obesity_country.

A1 *fx*

	A	B	C	D	E	F	G
0595	20593	Spain	1993	14.4 [11.5	Male		
0596	20594	Spain	1993	17.2 [14.4	Female		
0597	20595	Spain	1994	16.3 [14.3	Both sexes		
0598	20596	Spain	1994	14.8 [11.9	Male		
0599	20597	Spain	1994	17.5 [14.7	Female		
0600	20598	Spain	1995	16.6 [14.6	Both sexes		
0601	20599	Spain	1995	15.2 [12.3	Male		
0602	20600	Spain	1995	17.8 [15.0	Female		
0603	20601	Spain	1996	17.0 [14.9	Both sexes		
0604	20602	Spain	1996	15.6 [12.8	Male		
0605	20603	Spain	1996	18.1 [15.3	Female		
0606	20604	Spain	1997	17.3 [15.3	Both sexes		
0607	20605	Spain	1997	16.0 [13.2	Male		
0608	20606	Spain	1997	18.3 [15.6	Female		
0609	20607	Spain	1998	17.6 [15.6	Both sexes		
0610	20608	Spain	1998	16.4 [13.6	Male		
0611	20609	Spain	1998	18.6 [15.8	Female		
0612	20610	Spain	1999	18.0 [15.9	Both sexes		
0613	20611	Spain	1999	16.9 [14.0	Male		
0614	20612	Spain	1999	18.8 [16.1	Female		
0615	20613	Spain	2000	18.3 [16.3	Both sexes		
0616	20614	Spain	2000	17.3 [14.5	Male		
0617	20615	Spain	2000	19.1 [16.3	Female		
0618	20616	Spain	2001	18.6 [16.6	Both sexes		
0619	20617	Spain	2001	17.7 [14.9	Male		

The above dataset obesity_country consists of 24571 rows and 5 columns. The columns are Country, year, obesity percentage, and sex. The sex column is categorical and shows whether the person is male, female, or both sexes. The year column details the year in which the details were recorded. Then in obesity percentages, we have the obesity percentages of different countries recorded in different years which are quantitative values.

The below screenshot shows the details of the World dataset.

A1 fx name

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	name	alpha-2	alpha-3	country-cc	iso_3166-2	region	sub-region	intermedia	region-coc	sub-region	intermediate-region-code		
2	Afghanista	AF	AFG	4	ISO 3166-2	Asia	Southern Asia		142	34			
3	Åland Islar	AX	ALA	248	ISO 3166-2	Europe	Northern Europe		150	154			
4	Albania	AL	ALB	8	ISO 3166-2	Europe	Southern Europe		150	39			
5	Algeria	DZ	DZA	12	ISO 3166-2	Africa	Northern Africa		2	15			
6	American S	AS	ASM	16	ISO 3166-2	Oceania	Polynesia		9	61			
7	Andorra	AD	AND	20	ISO 3166-2	Europe	Southern Europe		150	39			
8	Angola	AO	AGO	24	ISO 3166-2	Africa	Sub-Sahara	Middle Afr	2	202	17		
9	Anguilla	AI	AIA	660	ISO 3166-2	Americas	Latin Amer	Caribbean	19	419	29		
10	Antarctica	AQ	ATA	10	ISO 3166-2:AQ								
11	Antigua an	AG	ATG	28	ISO 3166-2	Americas	Latin Amer	Caribbean	19	419	29		
12	Argentina	AR	ARG	32	ISO 3166-2	Americas	Latin Amer	South Ame	19	419	5		
13	Armenia	AM	ARM	51	ISO 3166-2	Asia	Western Asia		142	145			
14	Aruba	AW	ABW	533	ISO 3166-2	Americas	Latin Amer	Caribbean	19	419	29		
15	Australia	AU	AUS	36	ISO 3166-2	Oceania	Australia and New Ze		9	53			
16	Austria	AT	AUT	40	ISO 3166-2	Europe	Western Europe		150	155			
17	Azerbaijan	AZ	AZE	31	ISO 3166-2	Asia	Western Asia		142	145			
18	Bahamas	BS	BHS	44	ISO 3166-2	Americas	Latin Amer	Caribbean	19	419	29		
19	Bahrain	BH	BHR	48	ISO 3166-2	Asia	Western Asia		142	145			
20	Banglades	BD	BGD	50	ISO 3166-2	Asia	Southern Asia		142	34			
21	Barbados	BB	BRB	52	ISO 3166-2	Americas	Latin Amer	Caribbean	19	419	29		
22	Belarus	BY	BLR	112	ISO 3166-2	Europe	Eastern Europe		150	151			
23	Belgium	BE	BEL	56	ISO 3166-2	Europe	Western Europe		150	155			
24	Belize	BZ	BLZ	84	ISO 3166-2	Americas	Latin Amer	Central Am	19	419	13		
25	Benin	BJ	BEN	204	ISO 3166-2	Africa	Sub-Sahara	Western A	2	202	11		

The above dataset World consists of 250 rows and 11 columns. The different columns are country name, alpha-2, alpha-3, country-cc, iso_3166, region, sub-region, intermediate, region-code, sub-region, intermediate -region code. The data set consists of both categorical and numerical data. The dataset has information about different country names and their codes and the region and subregions the different countries belong to.

The below screenshot gives the details about the world_population dataset.

	A	B	C	D
1	Country N	Year	Count	
2	Aruba	1960	54211	
3	Afghanista	1960	8996973	
4	Angola	1960	5454933	
5	Albania	1960	1608800	
6	Andorra	1960	13411	
7	United Ara	1960	92418	
8	Argentina	1960	20481779	
9	Armenia	1960	1874121	
10	American !	1960	20123	
11	Antigua an	1960	54131	
12	Australia	1960	10276477	
13	Austria	1960	7047539	
14	Azerbaijan	1960	3895397	
15	Burundi	1960	2797932	
16	Belgium	1960	9153489	
17	Benin	1960	2431622	
18	Burkina Fa	1960	4829288	
19	Banglades	1960	48013504	
20	Bulgaria	1960	7867374	
21	Bahrain	1960	162427	
22	Bahamas,	1960	109534	
23	Bosnia anc	1960	3225668	

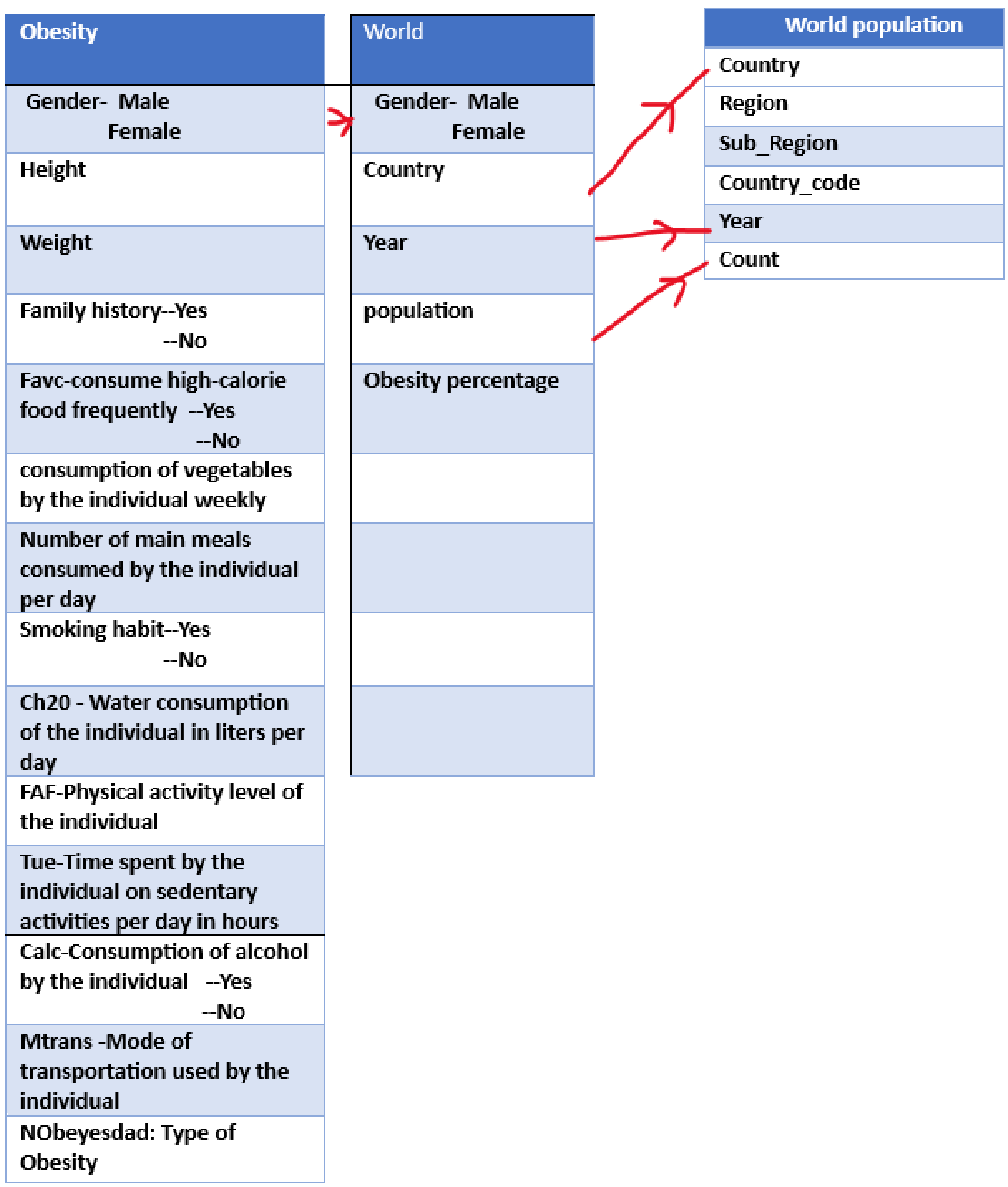
The above dataset's world population consists of 12596 rows and 3 columns. The three columns are country, year, and count. This dataset has numerical and categorical data types. The country column is categorical showing the names of different countries. The year and Count columns are numerical, and they show the different values of the population for different countries in different years.

3.3.3 DETAIL DESIGN OF FEATURES WITH DIAGRAM

The below design shows the interaction between the three different datasets that are used in the project.

The important features from the three datasets obesity, world, and world population are the obesity percentage that is indirectly dependent on the different factors that are affecting the value of obesity in both genders Male and female, like the daily physical habit, drinking habits, smoking habits, genetics, physical activity, genetic factors, travel method, consumption of natural unprocessed food and consumption of water.

Then additionally based on the obesity count and population in different countries we find the obesity percentage in both, males and females in different countries. And another consideration we take while calculating obesity in different genders in different countries is the different years during which obesity and the population have been counted. Thus, we get a correlation and interdependence between different important features in the three datasets obesity, world, and world population.



Detail design of Features with diagram

Input:
We are to consider below inputs:

1. Gender (Female, Male)
2. Age (numeric)
3. Height (numeric)
4. Weight (numeric)
5. family_history_with_overweight (Yes/No)
6. FAVC: Whether the individual consumes high-calorie food frequently (Yes/No)
7. FCVC: Frequency of consumption of vegetables by the individual weekly (numeric).
8. NCP: Number of main meals consumed by the individual per day(numeric).
9. CAEC: Consumption of food between main meals (Yes/No).
10. SMOKE: Whether the individual smokes tobacco (Yes/No).
11. CH2O: Water consumption of the individual in liters per day (numeric)
12. SCC: Whether the individual monitors their calorie intake (Yes/No)
13. FAF: Physical activity level of the individual (numeric).
14. TUE: Time spent by the individual on sedentary activities per day in hours(numeric).
15. CALC: Consumption of alcohol by the individual (Yes/No).
16. MTRANS: Mode of transportation used by the individual. (Automobile, Motorbike, Bike, Public_Transportation, Walking)
17. NObeyesdad: Type of Obesity
18. (Insufficient_Weight, Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_Type_I, Obesity_Type_II, Obesity_Type_III)

Output:
A calculated obesity type and visual representation of how inputs contribute to obesity.

We are using many different types of charts to visualize data.

1. Visualizing Obesity concerning factors affecting obesity.
2. Visualizing the highest, lowest, and regional obesity numbers.
3. Visualizing Obesity concerning gender.

   s

3.3.4 DATA TRANSFORMATION

The primary goal of "Flab or Fab" is to accurately predict an individual's obesity score. This is achieved by training the model using existing data and validating and testing it with the latest data available. Multiple models are tested to determine the one that produces the highest level of accuracy, which is then used to predict the risk factor for obesity. Once the model is trained, the website can be hosted to allow users to input their information dynamically, and the obesity score will be produced as output. Additionally, the tool will use visualization techniques to demonstrate how various factors contribute to obesity. We are using Random Forest Classifier for prediction. Below are the model parameters we have chosen. CSCE: 5320.003 Increment-I Scientific Data Visualization After using the above best parameters, we have an accuracy of 95%.

We are using Random Forest Classifier for prediction.
Below are the model parameters we have chosen.

```
df_cleaned = df

df_scaled=df_cleaned.copy()

columns_to_scale = ['Gender','Age','Height','Weight','family_history_with_overweight','FAVC','FCVC','NCP','CAEC','SMOKE','CH2O','

scaler = StandardScaler()
df_scaled[columns_to_scale] = scaler.fit_transform(df_cleaned[columns_to_scale])

df_scaled.head()

#we perform some Standardization using minmaxscaler
df_scaled_mm=df_cleaned.copy()

columns_to_scale_mm = ['Gender','Age','Height','Weight','family_history_with_overweight','FAVC','FCVC','NCP','CAEC','SMOKE','CH2O

mmscaler = MinMaxScaler()
df_scaled_mm[columns_to_scale_mm] = mmscaler.fit_transform(df_cleaned[columns_to_scale_mm])

df_scaled_mm.head()


X = df_cleaned.drop(['NObeyesdad'], axis=1) #features
y = df_cleaned['NObeyesdad']  #target feature

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle = True)


# Hyperparameter Tuning using GridSearchCV
param_grid = {
    'n_estimators': [500, 1000, 1500],
    'max_depth': [10, 20, 30],
    'max_features': ['sqrt', 'log2'],
    'max_leaf_nodes': [100, 500, 1000]
}
rf = RandomForestClassifier()
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
best_params = grid_search.best_params_
print("Best parameters:", best_params)
```

```
Best parameters: {'max_depth': 20, 'max_features': 'log2', 'max_leaf_nodes': 500, 'n_estimators': 500}
```

After using the above best parameters, we have an accuracy of 95%.

```
Accuracy: 0.9527186761229315
Confusion Matrix:
 [[56  6  0  0  0  0  0]
 [ 4 50  2  0  0  0  0]
 [ 0  3 46  1  0  0  0]
 [ 0  0  0 76  2  0  0]
 [ 0  0  0  1 57  0  0]
 [ 0  0  0  0  0 63  0]
 [ 1  0  0  0  0  0 55]]
True Positive Rate: 0.9259259259259259
Classification Report:
              precision    recall  f1-score   support

           1       0.92      0.90      0.91        62
           2       0.85      0.89      0.87        56
           3       0.96      0.92      0.94        50
           4       0.97      0.97      0.97        78
           5       0.97      0.98      0.97        58
           6       1.00      1.00      1.00        63
           7       1.00      0.98      0.99        56

    accuracy                           0.95       423
   macro avg       0.95      0.95      0.95       423
weighted avg       0.95      0.95      0.95       423
```

3.4 Task Abstraction

A web application for obesity prediction is a useful tool for encouraging healthy living and preventing obesity. The dashboard which we created for the obesity level prediction shows the specific aspects that influence obesity.

3.4.1 Target

The tool is used to target people who wish to track and control their weight. By using the web application users will be able to have a complete understanding of the factors that lead to an increase in obesity. Depending on the predictions users can change their lifestyle to lead a healthy life. This is an early detection of the high risk of obesity in individuals. The web application could also assist users in setting weight loss goals and tracking their success over time. This can help users stay motivated and on track.

3.3.1 Trends

The web application is used to show trends in obesity data. We have used data visualization techniques like bar charts, scatter plots, etc. to learn the patterns and narrow the factors that are most effective for increasing obesity levels in individuals. We have plotted graphs showing the highest and lowest obesity levels in different countries. We have distinguished the male and female obesity levels along with several other factors like age, weight, height, smoking habits, and lifestyle. From using the plots, we can see the difference in the habits of normal people and people who are obese.

From the results obtained, we can observe that the lack of exercise and bad diets are important contributions to the obesity epidemic in terms of risk factors. Physical inactivity, such as using mobile devices or sitting in front of a screen for lengthy periods, has become increasingly widespread in modern life. Age is also a component of the obesity epidemic, with obesity rates increasing with age. Obesity may also be influenced by a family history of obesity and genetic factors. Furthermore, the intake of high-calorie, low-nutrient meals has increased dramatically in recent years, contributing to an increase in obesity rates. We can also see that obesity affects men more than women due to less physical activity.

3.3.2 Features

Below are the main features that we are going to visualize using different kinds of encodings.

1. Visualizing Obesity concerning factors affecting obesity.
2. Visualizing the highest, lowest, and regional obesity numbers.
3. Visualizing Obesity concerning gender.

3.4.2 ACTIONS

3.4.2.1 Query

Users may wish to conduct the following task abstractions on this type of application:

- **Obesity Type:**

To gain insight into how much weight they have and design an appropriate approach to weight control, individuals may choose to keep track of their obesity type.

- **Age factor:**

Users may want to keep track of their age and observe how their age influences their physique and general well-being.

- **Height factor:**

Users may want to keep track of their height while trying to calculate their body mass index (BMI) and assess their weight status.

- **Family history with overweight:**

Users may want to track their family history of obesity to identify which individuals are in greater danger of developing obesity and take steps to avoid it.

- **Number of main meals:**

Users may want to log the number of main meals individuals have to control their calorie intake and avoid overeating.

- **Consumption of food between meals:**

Users might want to keep track of their food consumption between meals to manage their snacking habits and avoid overeating.

- **Smoking habit:**

Smoking Habit: Users should keep track of their smoking habits to see how it impacts their physique and health.

- **Daily water intake:**

Users should keep track of their daily water consumption to stay hydrated and control their liquid calorie intake.

- **Calories monitoring:**

Users may choose to keep track of their calorie intake and output to monitor their energy balance and manage their weight.

- **Physical activity:**

Users may desire to measure their physical activity to check their exercise levels and enhance their general health.

Time spent on devices:

Users may want to keep track of how much time they spend on their devices.

- **Monitoring Gender:**

Users may wish to track their gender to examine how it impacts their physique and whole health, as well as to compare their weight and health condition to others of the same gender.

3.4.2.2 Analyze

We are presenting the below charts, the consumer can discover, and enjoy the charts:

- **Choropleth Graph:**

Users may wish to investigate and compare obesity prevalence statistics across various regions or countries to get insights into global obesity trends and highlight areas of concern. They may also want to examine obesity prevalence statistics by age, height, or gender to uncover discrepancies and patterns.

Example: Country, Continent, Obesity Perveance, Gender, Year.

- **Bar Graph:**

Users may wish to compare the average obesity rates across different countries or regions to find locations with high or low obesity rates. They may also wish to compare obesity rates across demographic groups, such as age, and gender to uncover inequalities and patterns.

Example: Country, Continent, Obesity Perveance, Gender, Year.

- **Stacked Graph:**

To better understand the underlying causes of obesity and design an effective management strategy, users may want to visualize and compare the contribution of several aspects to obesity, such as calorie intake, physical activity, and smoking.

Examples: Caloric Intake, Physical Activity, Smoking, Obesity Prevalence, Gender, and Age.

- **Horizontal Graph:**

Users may want to visualize and compare the relationship between several parameters and obesity prevalence, such as caloric consumption and obesity prevalence or physical activity and obesity prevalence. They may also wish to analyze these associations across demographic groupings, such as age, gender, or race/ethnicity, to uncover patterns and inequalities.

Examples: Caloric Intake, Physical Activity, Smoking, Obesity Prevalence, Gender, and Age.

An obesity level predictor dashboard that includes a variety of graphs and allows users to investigate various aspects that lead to obesity could provide significant insights and assist users in better understanding the root causes of obesity and developing suitable treatment plans.

3.5 Implementation Using Tools

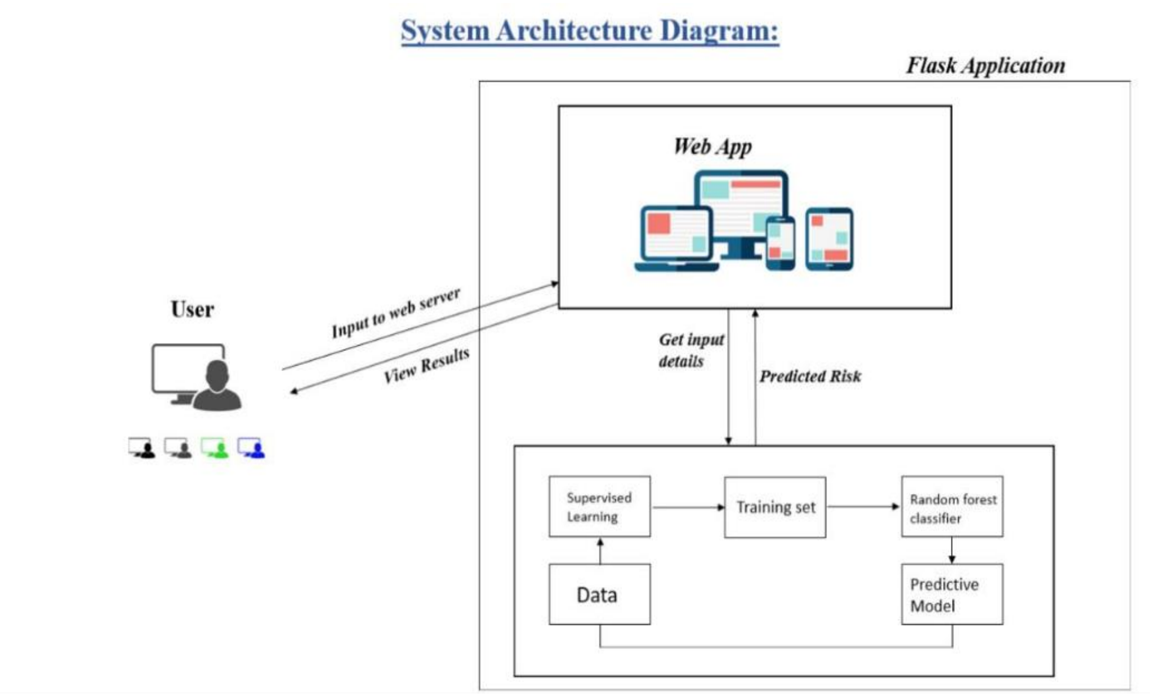
The different tools that are used for the project are :

1. Flask
2. Python libraries math, NumPy, pandas, joblib.
3. Gridsearchcv
4. Sklearn for Random Forest
5. Plotly, plotly.express, plotly.graphobjs
6. Matplotlib for geographical dataset

3.5.1 Flask Architecture

1. User: The user enters information on the website in answer to the questions. When the user inputs all his information and clicks the proceed button, the model predicts the result and displays it on the webpage.
2. Flask Application: The flask application is made up of three main parts.
3. Websites: Flask hosts the home page and the result page.

4. Data: The available data is used to train the model.
5. Model: The model is trained using the data using a random forest classifier. It is compiled, run with the flask run command, and then used to generate predictions.



3.5.1.1 Flask

Developers can create web projects quickly and easily using the Flask framework. The Flask web application framework is made in Python. It was developed by Armin Ronacher, the team leader of Poocco, a global community of Python enthusiasts. The Flask is built on top of the Jinja2 template engine and the Werkzeg WSGI toolkit.

3.5.1.2 WSGI

The de facto industry standard for creating Python online applications is the Web Server Gateway Interface (WSGI). The WSGI specification outlines a standard interface for web servers and online applications.

Utilizing a WSGI toolkit dubbed Werkzeug, requests, response objects, and utility actions are implemented. On top of it, a web frame may now be constructed. Werkzeg is one of the foundations of the Flask framework.

3.5.1.3 Jinja2

A well-liked Python template engine is jinja2. A web template system renders a dynamic web page by fusing a template with a particular data source.

3.5.1.4 Benefits of Flask

Scalable

We can use Flask, a microframework, to quickly expand a project like a web app. This is the best option if you want to create an app that starts small but has the potential to grow quickly and in areas that you have not fully figured out. It can operate continuously even as it scales up and higher due to how easy it is to use and how independent it is.

Flexible:

One of Flask's main benefits and a crucial element. While developing a web application, simplicity is preferable to complexity since it can be restructured and moved around more easily.

Simple to navigate:

Web developers can save time and effort by using the microframework since it is inherently simple to understand and gives them more control over their code and the possibilities.

Lightweight:

Flask enables modular programming, which allows for the division of its functionality into several swappable modules. Each module serves as a standalone construction block that can carry out a specific functionality. Together, this indicates that all the structure's component elements are adaptable, movable, and tested on their own.[10]

3.5.2 PYTHON LIBRARIES: PANDAS, NUMPY, MATH, JOBLIB.

3.5.2.1 Pandas

Panda is an open-source library designed primarily for working quickly and logically with relational or labeled data. It offers a range of data structures and procedures for working with time series and numerical data. The NumPy library serves as the foundation for this library. Pandas is quick and offers its users exceptional performance & productivity.

Advantages

- It is a quick and effective way to edit and assess data.
- Various file objects from which data can be loaded.
- Simple handling of missing data; both floating-point and non-floating-point values are represented as NaN.

- Size mutability: Dataframe columns and higher-dimensional items can be added or removed.
- merging and connecting data sets.
- Modular data set pivoting and reshaping
- Time-series features are available.
- Practical group by features that let you split, use, and combine data sets.

3.5.2.2 NumPy

Use the NumPy Python library to manipulate arrays. There are also functions for working with linear algebra, matrix operations, and the Fourier transform. The acronym for numerical Python is NumPy. The Python equivalent of arrays is listed, although lists are slow to run. NumPy will make array objects up to 50 times faster than normal Python lists. Utilizing an ND array is rather simple thanks to a variety of supporting methods provided by the NumPy array object, sometimes known as an array. In data research, when effectiveness and resourcefulness are crucial, arrays are widely utilized. Since they are stored in a single continuous area of memory, NumPy arrays are easier to use than lists.

This is the primary factor that makes NumPy faster than lists. Additionally, it is enhanced to function with modern CPU architectures.

3.5.2.3 Math

Math module for Python. You can use a built-in module in Python to do mathematical operations. There are several techniques and constants in the math module.

3.5.2.4 Joblib

To train a machine learning model with a large-size dataset, we also need a decent amount of time. Machine learning models need huge datasets to achieve high accuracy. As a result, we use the joblib library to avoid training the model again. Instead, we train the model only once, save it using the joblib library, and then reuse it.

3.5.2.5 Library Joblib

This article will examine how to save and load machine learning models using Python's joblib library. The use of Google Colab is made for this project.
Python's Joblib package enables the simultaneous execution of computationally demanding jobs. It offers a collection of tools for carrying out parallel operations on massive data sets.

3.5.2.6 Matplotlib

A well-liked Python data visualization library is Matplotlib. Users can make a wide range of graphs using this software, including line plots, scatter plots, bar plots, histograms, and more. For these plots, Matplotlib offers a variety of customization choices, including control over colors, typefaces, labels, and legends.

Matplotlib is a crucial tool for producing plots of publication quality and is widely used in the scientific and data analytic sectors. It can be used in a range of settings, including desktop GUIs, online applications, and Jupyter notebooks.

Matplotlib has several important features, including:

- Wide variety of plot types and styles are supported.
- Colors, fonts, labels, and legends that can be changed.
- ability to divide a figure into many figures and create subplots.
- Including NumPy, Pandas, and other scientific computing libraries in the integration assistance with interactive charting tools like zooming and panning
- Access to extensibility via a potent object-oriented API
- With the help of Python's package management, pip, Matplotlib can be installed for free and without restriction.

3.5.2.7 Plotly

Plotly is a data visualization package for Python, R, and other programming languages that enables the creation of interactive, web-based plots. Users can design numerous interactive graphs with Plotly, including line plots, scatter plots, bar plots, histograms, 3D surface plots, and more.

The ability to produce interactive plots that can be integrated into websites, notebooks, and other applications is one of Plotly's primary advantages. There are many other ways to personalize these plots, including changing the colors, labels, and annotations.

Through its web-based platform, Plotly Online, Plotly offers assistance for data exploration and analysis in addition to its web-based plotting features. Users of this platform can upload data, make graphs, and publish their work online.

Plotly has several important features, including:

- Wide variety of interactive plot kinds and styles are supported.
- Colors, fonts, labels, and annotations that can be changed.
- Python, R, and other programming languages integration.
- the capability to develop interactive graphs for dashboards and web apps.
- assistance with data analysis and exploration with Plotly Online.
- Adaptability via a potent API [11]

Plotly.express

On top of Plotly, Plotly Express is a high-level data visualization library. For constructing a range of interactive graphs, including line plots, scatter plots, bar plots, histograms, and more, it offers a clear and basic syntax.

Its simplicity and ease of usage are two of Plotly Express's standout qualities. It offers a variety of methods for building various plot kinds, each with practical default values that are easily customizable.

Additionally, Plotly Express supports a broad variety of data formats, such as CSV files, Pandas data frames, and more. Additionally, it interacts with Plotly Online, Plotly's web-based platform, making it simple for users to share their work with others.

Plotly Express has several important features, such as:

- syntax for generating at a high level.
- a high-level syntax for building different interactive plots.
- simple and simple customization possibilities
- large range of data formats supported.
- Extensibility via a potent API Integration with Plotly Online for sharing and collaboration [11]

Plotly.graph_objs

Plotly's plotly.graph_objs sub-module offers a low-level interface for creating and modifying specific plot components including traces, layouts, and annotations. Users that need more precise control over the appearance and behavior of their plots should utilize it.

Users can construct a variety of interactive graphs Plotly.graph_objs, including line plots, scatter plots, bar plots, heatmaps, and more. Numerous customization choices, including different colors, markers, and line types, are available for each trace.

Plotly.graph_objs has a Layout class in addition to traces for altering a plot's general design, including axis labels, titles, and annotations. The Annotations class allows users to add unique annotations to their plots.

Plotly.graph_objs has several important features, including:

- Low-level interface for building and altering specific plot components.
- Controlling a plot's appearance and actions with detail.
- Wide variety of interactive plot kinds and styles are supported.
- Colors, fonts, labels, and annotations that can be changed.

- Extensibility via a potent API Integration with Plotly Online for sharing and collaboration.

3.5.2.8 GridSearchCV

The scikit-learn library's GridSearchCV function is used to do hyperparameter tuning on machine learning models. It is applied to find the ideal set of hyperparameters for a particular machine-learning method.

Hyperparameters are variables that are chosen in advance of the learning process and are not discovered through data analysis. The regularization parameter, the number of hidden layers in a neural network, and the learning rate are a few examples of hyperparameters. The values of a machine learning model's hyperparameters may have a significant impact on how well it performs.

In an extensive search over a given hyperparameter space, GridSearchCV trains and evaluates a model for each set of hyperparameters. It estimates each model's performance using cross-validation and chooses the set of hyperparameters that result in the best performance.

GridSearchCV has several important features, including:

- For machine learning models, automatic hyperparameter tuning.
- thorough investigation of a certain hyperparameter space
- estimation of model performance by cross-validation
- choosing the ideal collection of hyperparameters for the model and data at hand

The Scikit-Learn library includes GridSearchCV, which may be imported by executing the command from sklearn.model_selection import GridSearchCV. It is an effective method for enhancing machine learning model performance and lowering the danger of overfitting.

3.5.2.9 Sklearn

A well-liked open-source machine-learning library for Python is called Scikit-learn (often referred to as sklearn). It offers several tools for implementing and using machine learning algorithms and is built on top of NumPy, SciPy, and matplotlib.

Several algorithms for supervised and unsupervised learning tasks are included in Scikit-learn, such as:

- Regression techniques include logistic and linear regression.
- Classification techniques include decision trees, support vector machines, and k-nearest neighbors.
- Hierarchical clustering, k-means clustering, etc.
- Dimensional reduction techniques include PCA, etc.
- Scikit-learn additionally contains tools for data preprocessing, model selection, and evaluation in addition to these techniques. For instance, it offers functions for scaling and normalizing data, dividing data into training and testing sets, and choosing the ideal collection of hyperparameters for a particular model.
- Scikit-learn's salient characteristics include:
- Implementing machine learning algorithms using a straightforward and reliable API
- extensive instructions and documentation
- Numerous tasks and algorithms are supported.
- Matplotlib, SciPy, and NumPy integration
- Using optimized C code, many algorithms are implemented efficiently.
- A large user community and ongoing development

3.5.2.10 Random Forest Classifier

A well-liked supervised learning approach for classification tasks is the Random Forest Classifier. It uses the ensemble learning technique to merge various decision trees into a potent model that can easily generalize to new data.

A random subset of the training data and a random subset of the features are used to construct each decision tree in a Random Forest Classifier. This increases the diversity of the trees and lessens overfitting. The forecasts of all the trees are combined to get the model's final prediction.

The Random Forest Classifier's major characteristics include:

- The capacity to handle both continuous and categorical data.
- Robustness to missing data and noise.
- High performance and accuracy across a variety of datasets.
- Automatic selection of key characteristics.
- Overfitting resistance thanks to the ensemble approach.
- capacity for working with high-dimensional and huge datasets.

The RandomForestClassifier class in sci-kit-learn can be used to build a Random Forest Classifier model. The class offers a wide range of hyperparameters, such as the number of trees, the depth of each tree, and the number of features utilized at each split, which can be modified to enhance the performance of the model.

3.5.3 USE OF DIFFERENT TOOLS IN THE PROJECT:

- We wrote the code of the project in Visual Studio Code. We created Python files named app.py, model.py, and obesity.py.
- Firstly, we created the flask environment. Then we wrote the different HTML codes in the template folder. In the template folder, we have index.html, analyze.html, charts.html, and obesity.html.
- The index.html contains the code for the opening page of the web application.
- In index.html under the head> tag we give the title as Obesity Type Predictor. Then in the style tag, we give the style for the webpage.
- Then in the .header and .footer> tag we give the details of dimension and color and different positionings. Then we give the input commands for entering inputs and then we create a button > tag to give the details of the buttons that the user clicks and then in the output box we give the details for displaying the output.
- Then in the body>tag we give the names of all the different input features like age, weight, gender, family history, smoking, transportation, etc.
- Then in charts.html we give the same style.css for the background of the page and then in body>tag we give the text-align commands and the titles for different charts and the colors and the submit button etc.
- In the chart.html page we link the image to the source page.
- Then in the analyze.html page we use the same styles.css as in index.html and then in body>tag we give the different button types as obesity factors, highest obese countries, lowest obese countries, continental obesity indicator, gender-specific obesity, etc,
- Then in obesity.html we use the same styles.css and then in the class>tag we give all the different feature names that are used to predict obesity.

- The app.py file contains the code for the website that is created that predicts obesity.
- For creating the obesity prediction model, we used the sklearn library. Firstly, we cleaned the data frame by doing data preprocessing, then we scaled the data frame, then we chose the columns needed, and then normalized the data frame.
- Then we created a split and created a testing and training dataset. Then using a random forest classifier and grid search we created a machine learning model that is trained to predict obesity.

- In the model.py we write the Python code for creating the globe visualization showing different countries using matplotlib and Plotly.express.We gave the details of different countries to be drawn on the globe and linked the values of obesity percentages of different countries.

- Then in obesity.py we write the code for creating different plots and graphs using Plotly. express. Using Plotly. express we write the code to create different bar charts for obesity, height, gender, and weight, and for different features fcvc, favc, NCP, smoking and water consumption, and violin graph and scatter plot for alcohol consumption and other features.

3.5.4 PROJECT RUN INSTRUCTIONS

To run the Flask application, follow the below instructions:

1. Extract the project zip folder.
2. from the file directory of the project open the terminal.
3. Run the below commands.
   i. py -m venv env (to set up the Python environment)
   ii. Set-ExecutionPolicy Unrestricted -Scope Process
   iii. .\env\Scripts\activate (to activate the environment)
   iv. iv. flask run (to run flask application)

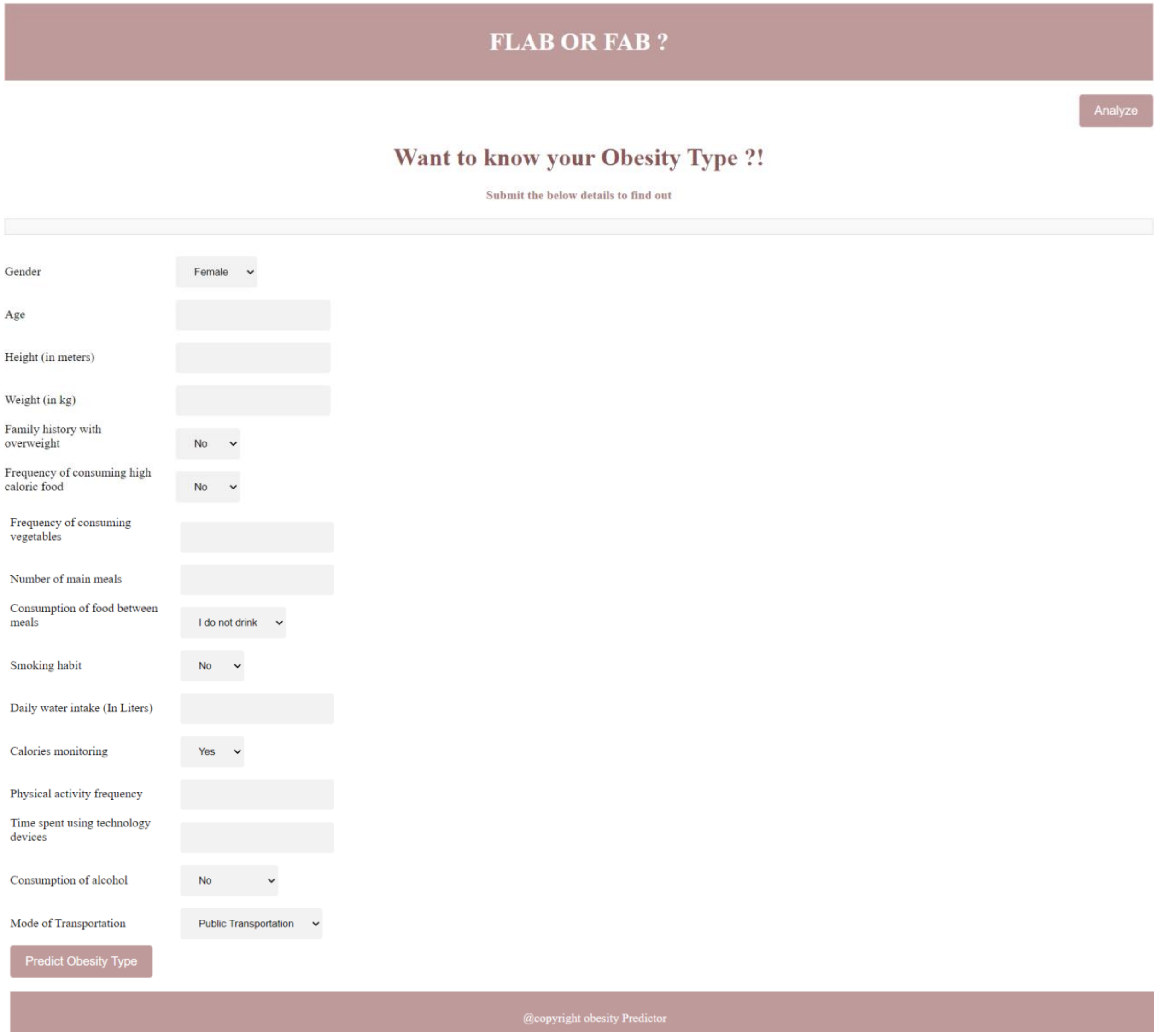
4. We can run the application by clicking on the local host path that is displayed on the terminal.

3.6 PRELIMINARY RESULTS FOR ANALYSIS

3.6.1 VISUALIZATION GRAPHS

Initially, we launched the webpage of our web application “Flab or Fab”. The webpage is designed with the main page which is an obesity type predictor. The below screenshot shows the main page of the web app created for obesity-type prediction.

3.6.1.1 Main Page



Screenshot: Main page of the web application Flab or Fab

The main page involves a questionnaire consisting of drop-down fields and text fields which can enable users to manually select a choice of input based on several factors like age, weight, and other day-to-day habits. The Obesity type predictor will give the response on the likelihood of a user becoming overweight.

Main Page Components

- Title of the page.
- Input drop-down fields and text fields (expected input from the user).
- Predict obesity type button.
- Analyze Button

Title of the main page

We have given the title of the initial page as an obesity type predictor where we have varied fields which need to be filled by the user who wants to check his/her obesity type.

User input fields in the Obesity Predictor

1. Drop-down

Below are the types of input fields designed on the main page which involve drop-down fields. The drop-down will enable the user to easily select their answers from the different options listed in the drop-down menu.

For instance, there is a gender field, with a male or female option and a yes/no dropdown option for the "family history of obesity" field. Similarly, dropdown menus are available for the other fields such as frequency of consuming high caloric food/consumption of food between means which has drop-down options such as I do not drink, I drink. There are other fields such as smoking habits, calorie monitoring, and consumption of alcohol which have yes or no as drop-down options, and another field which is a mode of transportation that involves choices of public or private.

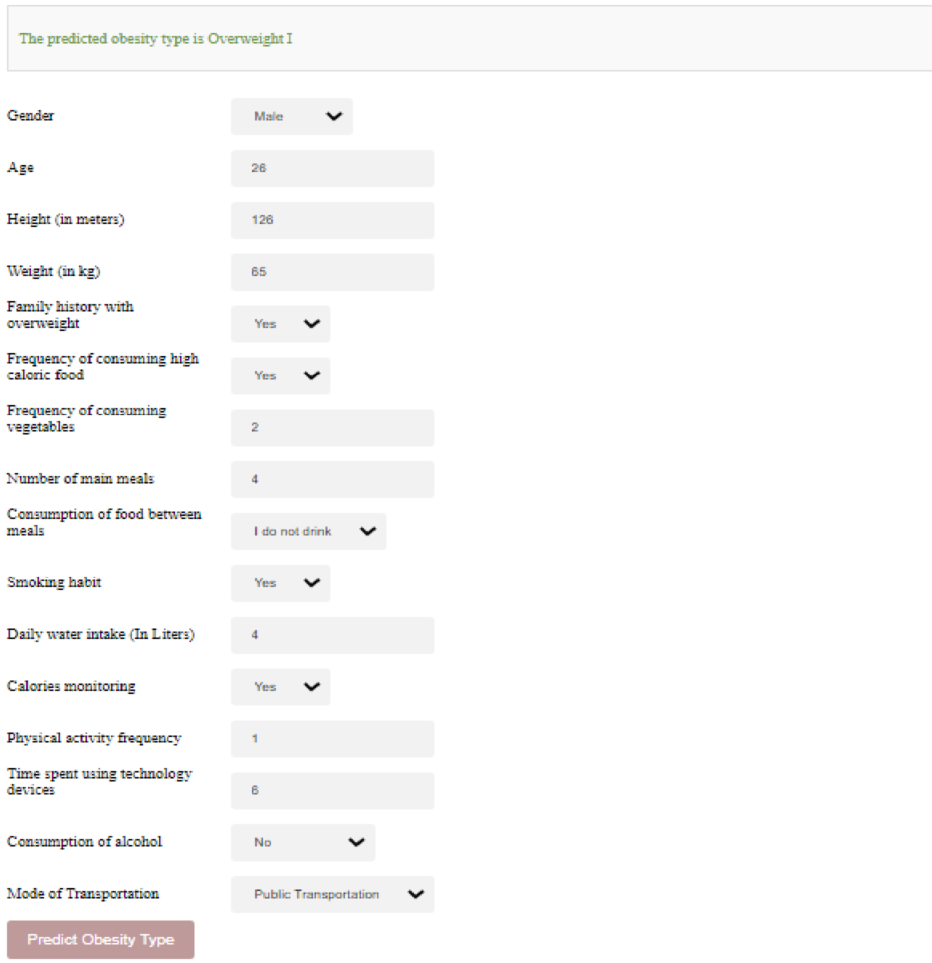
2. Text fields

Text fields on a web page are the input fields where users can enter text, such as age, height, weight, frequency of consuming vegetables, number of main meals, daily water intake (in liters), physical activity frequency, and time spent using technology devices.

3. Predict obesity Button.

After filling in all the required fields and then when we click on the predict obesity button, the webpage gives the response on the type of obesity. In the backend, we have a prediction model that takes in the user input and based on the values and model created it will decide the output and prints the response on the screen.

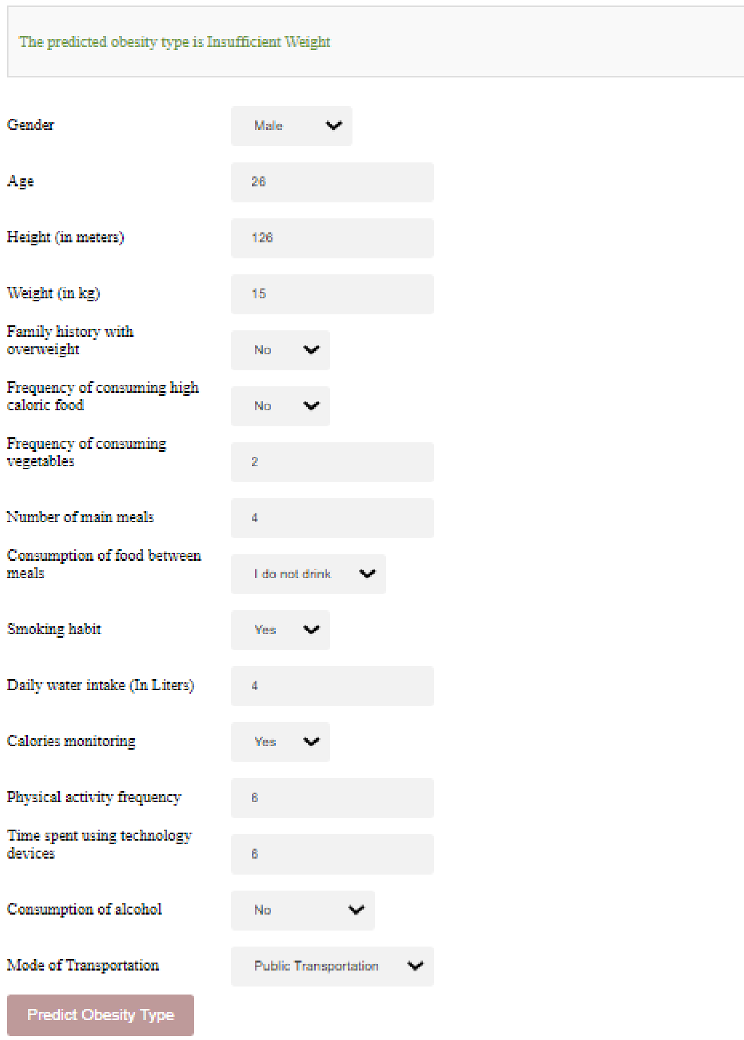
User test case: Overweight



Screenshot: The response of the predictor for the user test case.

User test case: Insufficient Weight

We can see in the screenshot that a user has answered all the questions required and the obesity predictor gives a response that the Male user is Overweight based on the factors.



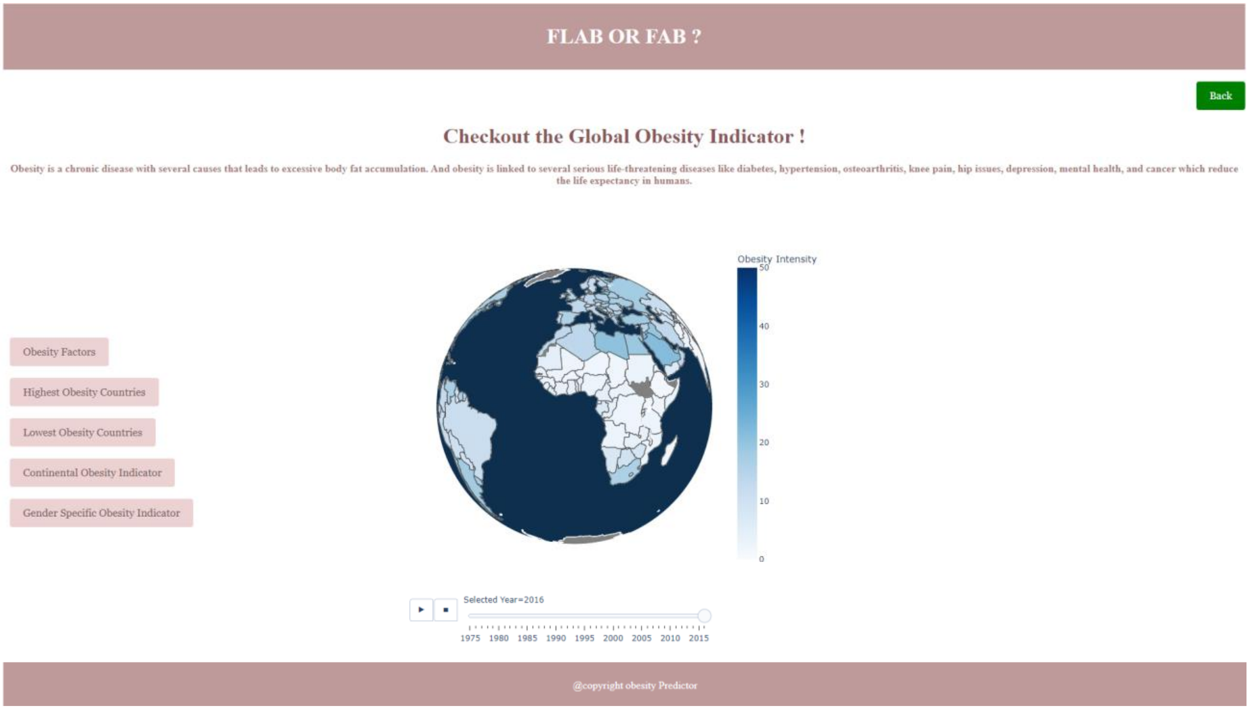
Screenshot: The response of the predictor for the user test case

We can see in the screenshot that a user has answered all the questions required but the weight is insufficient, and the obesity predictor gives a response that the weight is insufficient.

3.5.1.2 Analyze Page

There is also an analyze button at the right corner of the main page. When we click the button, it will further navigate to another dynamic web page that gives more in detail information about the factors that are responsible for obesity type.

Analyze



Screenshot: The response of the predictor for the user test case

The analysis page shows a globe-level indicator and a world map with various countries shown in various color intensities. It is a useful approach to show the obesity level and year of obesity. The values of the vertical and horizontal parameters are used to gauge the degree of obesity in various nations which will decide the color shade of each nation.

The choropleth graph is shown as an interactive graphic design with each nation labeled and colored by the severity of its obesity problem based on the selected year. For instance, nations with greater rates of obesity can be highlighted in dark blue, whereas nations with lower rates might be highlighted in lighter shades of the globe.

The user can change the year and rotate the globe by moving the cursor over the globe which is an interactive aspect as shown in the screenshot above.

Navigation buttons

We have a few other buttons to navigate to different pages which show different visualization graphs for different factors of the dataset.

Here we have 5 buttons for the user to navigate and check the effect of factors in detail that determine the type of obesity.

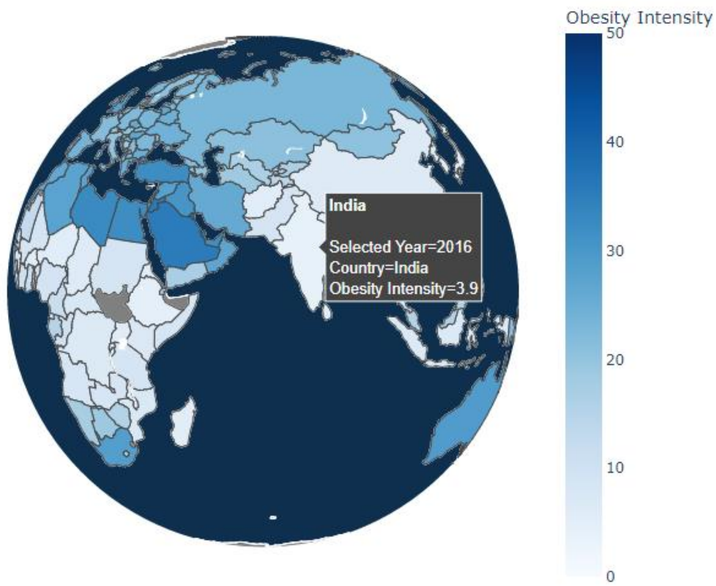
1. Obesity Factors
2. High-obesity Countries
3. Lowest obesity Countries
4. Continental obesity Indicator

5. Gender-specific obesity indicators.

Back button

There is also a back button for the user to navigate to the main page. As seen in the above screenshot, the back button is green in color.

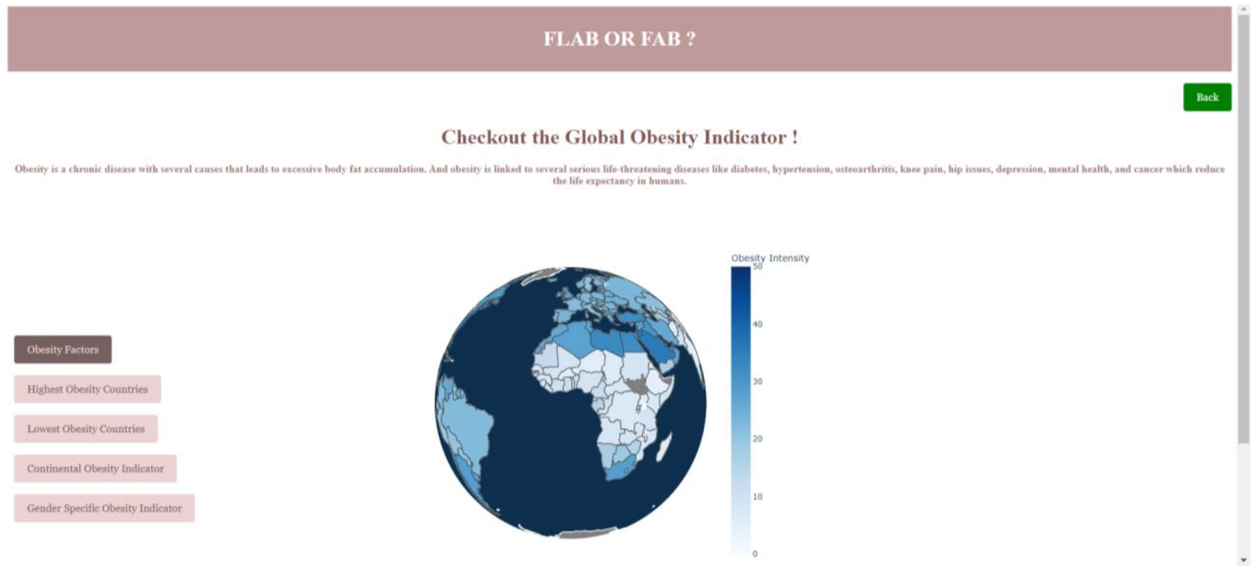
User test case



Screenshot: Tool tip showing the country's obesity details.

As seen in the above screenshot as the user selects a particular year, he can hover over the country and check in detail information about obesity in the tooltip. The details involve factors such as country name, selected year, country, and obesity intensity.

Here we can see the example of Niger, which is in Africa, the selected year is 2016, the country is Niger, and the obesity intensity is 5.5.



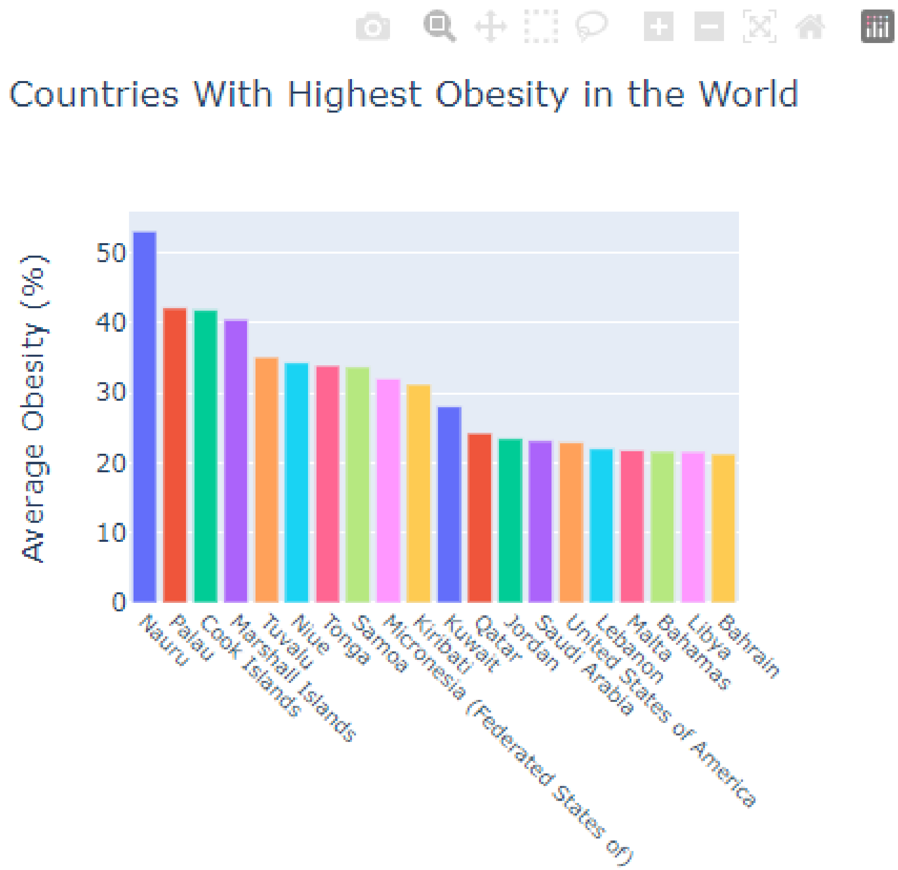
Screenshot: User clicking on the obesity factors button

As seen in the above screenshot, the user clicks on the obesity factors button. To acknowledge the user about the button, click the color of the button is changed to darker color on the click action. This will provide good visibility for the user.

Highest Obesity Countries

The second button on analyze page is the highest obesity countries.

Screenshot: Dashboard complete view



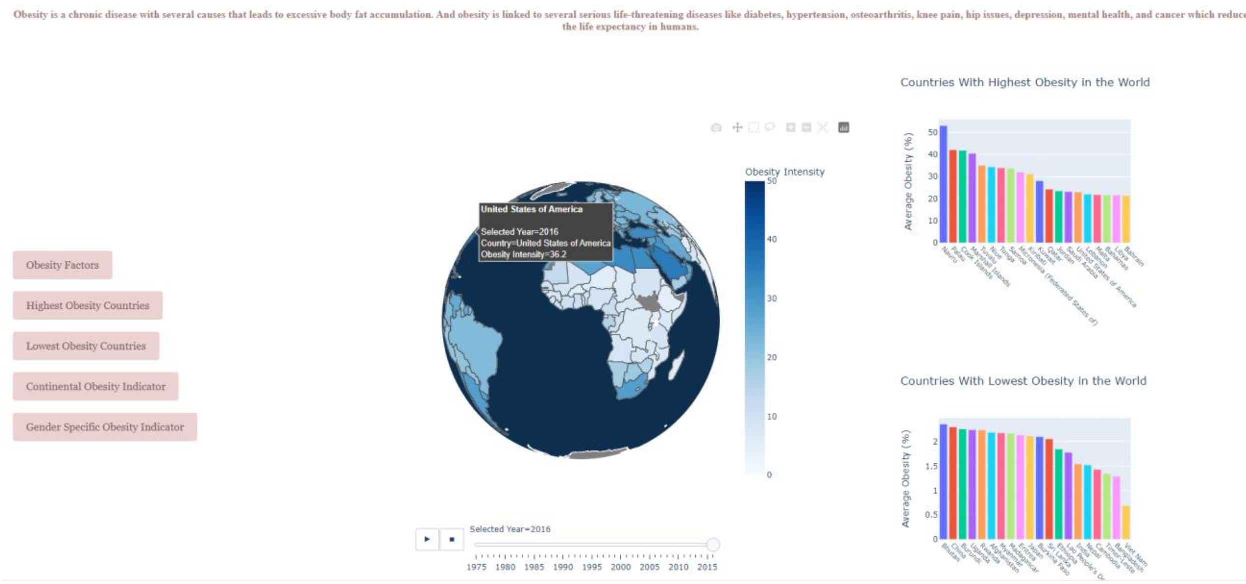
Screenshot: Countries with the highest obesity level

On the main page or dashboard of our web application, you can also see the bar graph interaction of the highest obesity rates along with the global map.

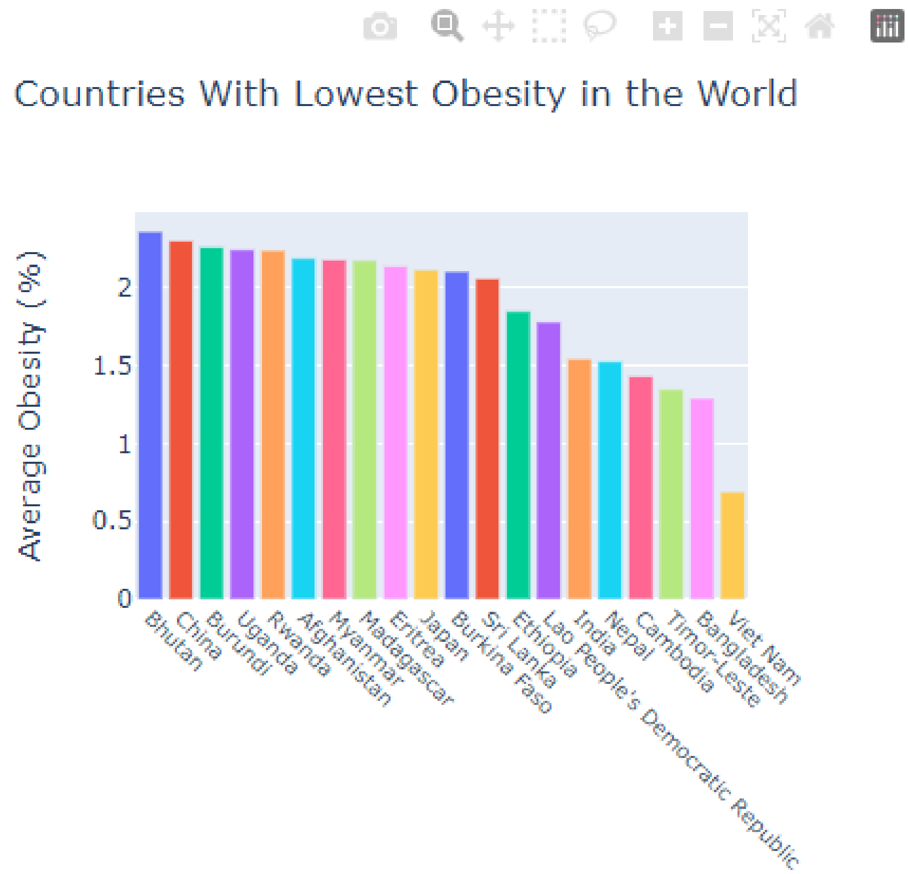
In this, we analyze the average obesity rate for different countries. The x-axis of the graph will show the names of the countries being compared, while the y-axis will show the average obesity rates for each country. Here we use different colors to interpret different countries to help the user easily understand. The bar graph can also be used to compare the average obesity of different countries and to identify any trends in the data. We can observe that Kuwait has the highest average obesity whereas Bahrain has the lowest average obesity.

Lowest Obesity Countries

Along with the highest obesity rates we have also added one more chart of the lowest obesity countries. This will help the user get a better view of analyzing the global map. A global map alone is not enough for the user to know many details. These bar graphs along with the global map will enable users to view countries' information and their obesity details.

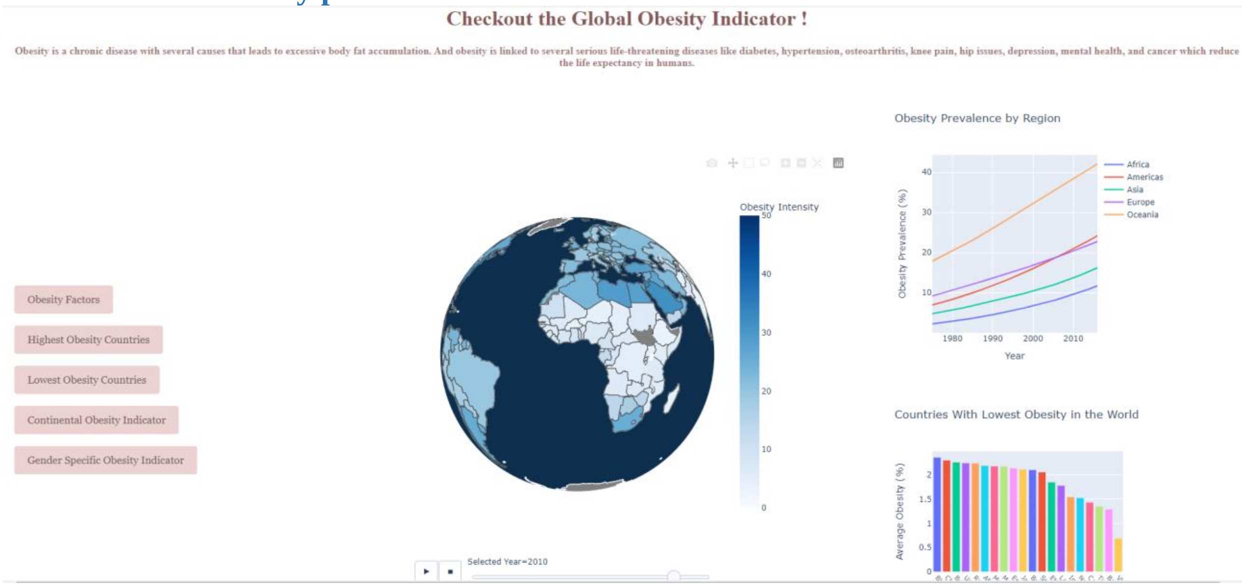


Screenshot: Countries with the lowest obesity level



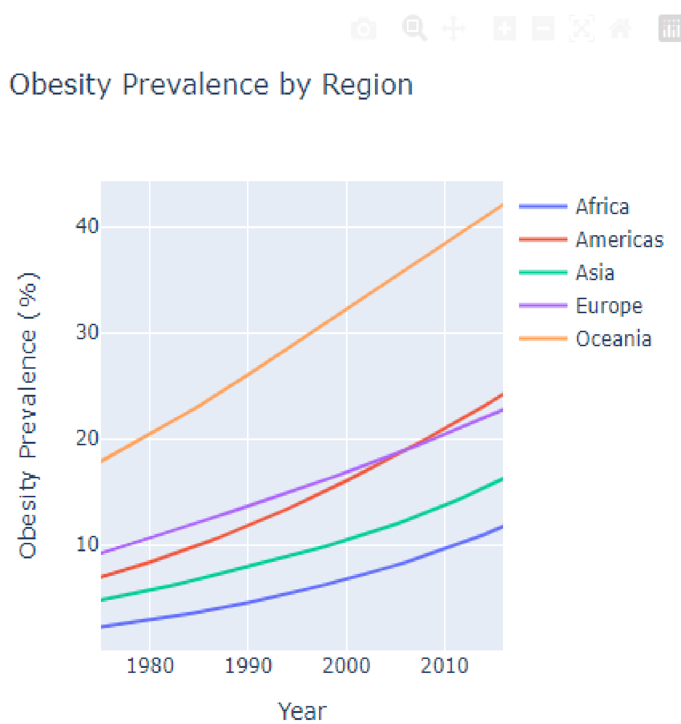
The graph's x-axis will display the names of the countries being compared, while the y-axis will display the average obesity rates for each country. To help the user comprehend, we utilize different colors to signify different countries. The bar graph can also be used to compare average obesity levels across countries and discover any trends in the data. Vietnam has the lowest average obesity rate, while Bhutan has the lowest average obesity rate.

Continent-wise obesity prevalence



Screenshot: Complete dashboard view with line and bar graphs

In the dashboard, we can also load the line graph of obesity levels by region. Displaying different graphs on the dashboard will help users gain more insights from the easy information on the screen without much complexity.



A line graph shows the continent-wise obesity prevalence over the years, we have added color coding to distinguish between the continents. The x-axis of the graph will show the years being compared and the y-axis shows the obesity prevalence for each continent. To make it simpler to identify the continents, each continent might be represented on the graph by a distinct colored line. The height of the line depicts the annual prevalence of obesity in that continent. We can see that the lowest obesity. We can see that Africa has the lowest obesity prevalence over the years while America has the highest. We can conclude that most Americans are prone to obesity.

3.5.1.3 Obesity Factors Section

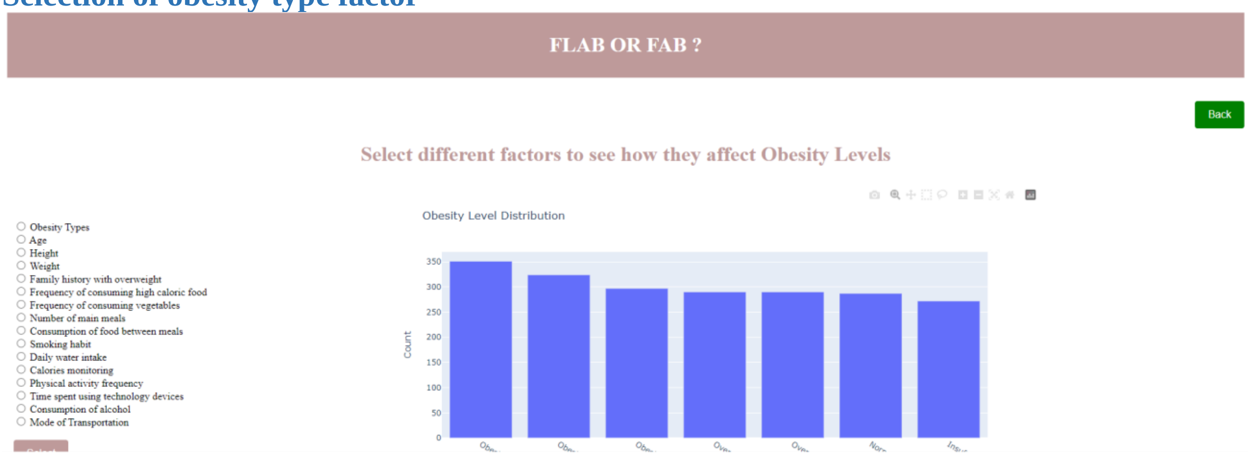
On clicking the obesity factors button, we can see various visualization options as shown in the screenshot below.

Different factors that affect obesity levels

On the left side of the page, we can see various factors based on which the obesity level varies as shown in the above screenshot.

These are the same factors that we asked about in the main page fields. This will let the user know the relevance of each field and the user can independently check each field's significance in deciding the obesity level across the globe. The user would be able to check the visualization and effect for all the fields globally.

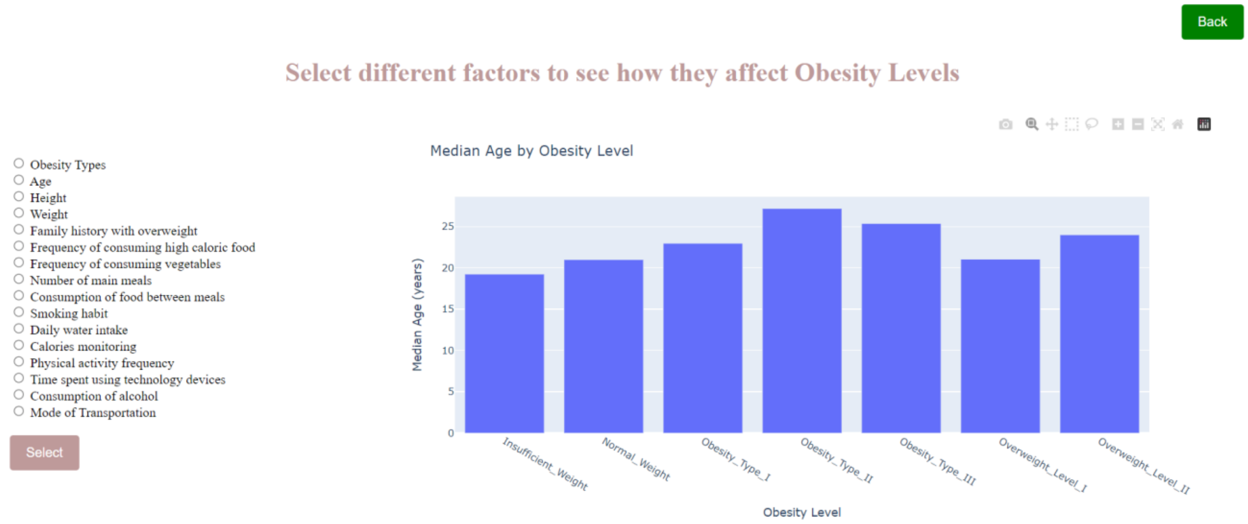
Selection of obesity type factor



Screenshot: Obesity factors section

In this section, we can see a graph which shows the bar graph plotted for the obesity type and the count of people. As we can see in the screenshot, type-1 obesity has the highest count of 350. While the lowest count in the dataset involves people with insufficient weight. There are other types for which the count is checked such as obesity type 2 and 3, overweight type 1 to 3, and normal weight count.

Selection of age factor



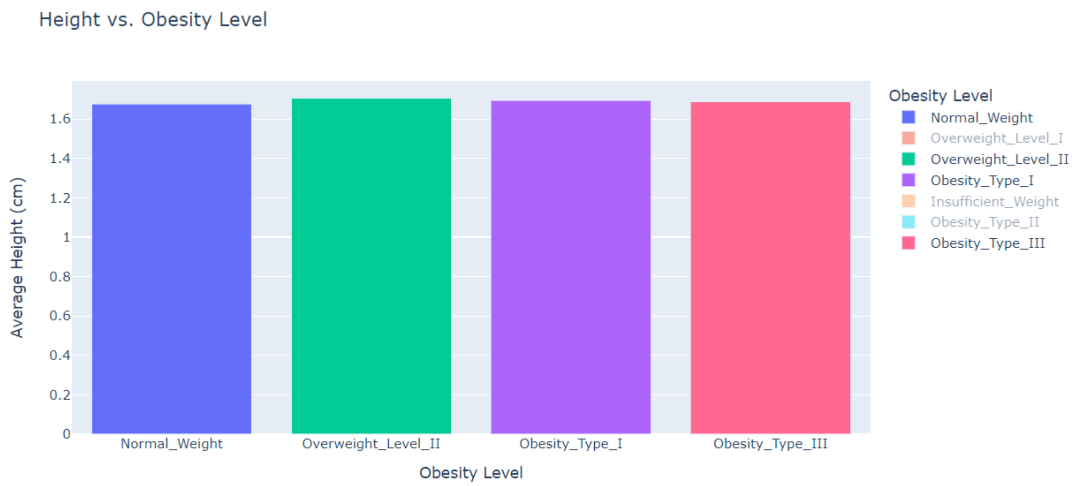
Screenshot: age factors

The bar graph shows the age group that falls into each obesity category with age groups on the x-axis and the count of individuals in each obesity category on the y-axis. The bars for each age group will indicate the number of people in that age group who fall into each category of obesity.

As we can see in the bar graph the obese type-1 people fall between the age group 20- 25, the maximum of people has obesity type-2, and they fall in the age group greater than 25 while the minimum count of people with overweight type 2 is age 20.

Selection of height factor

Select different factors to see how they affect Obesity Levels



The bar graph shows the Count of people in each obese group is shown on the y-axis of the bar graph with height categories on the x-axis. The bars for each height category would represent the number of people in each obesity category for that height category.

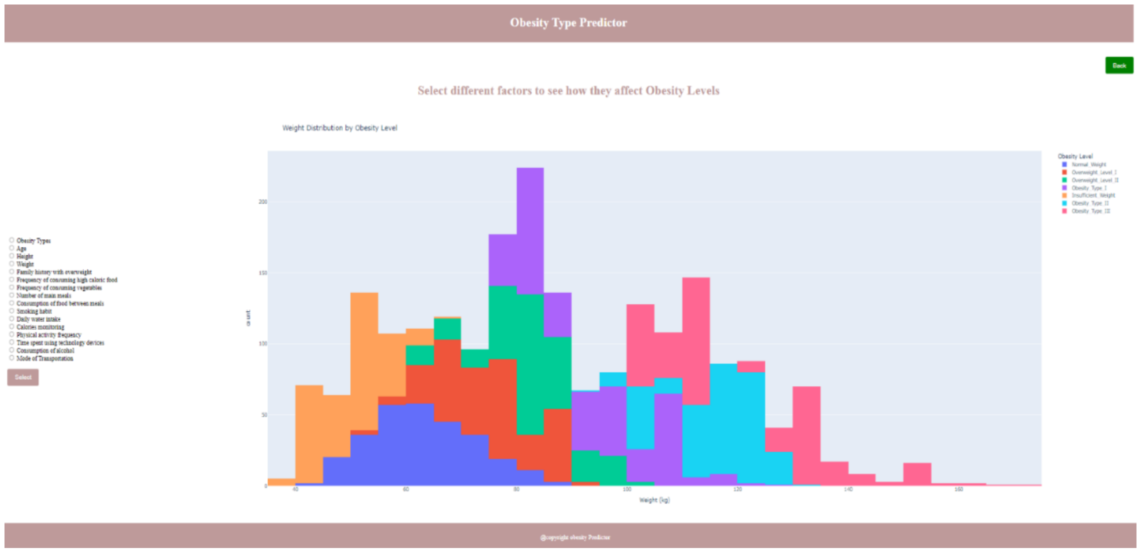
As seen in the above screenshot there is not much height difference among people in various obesity levels. The average height of people with obesity type 2 is greater than 1.6 whereas for normal weight, height is 1.65.

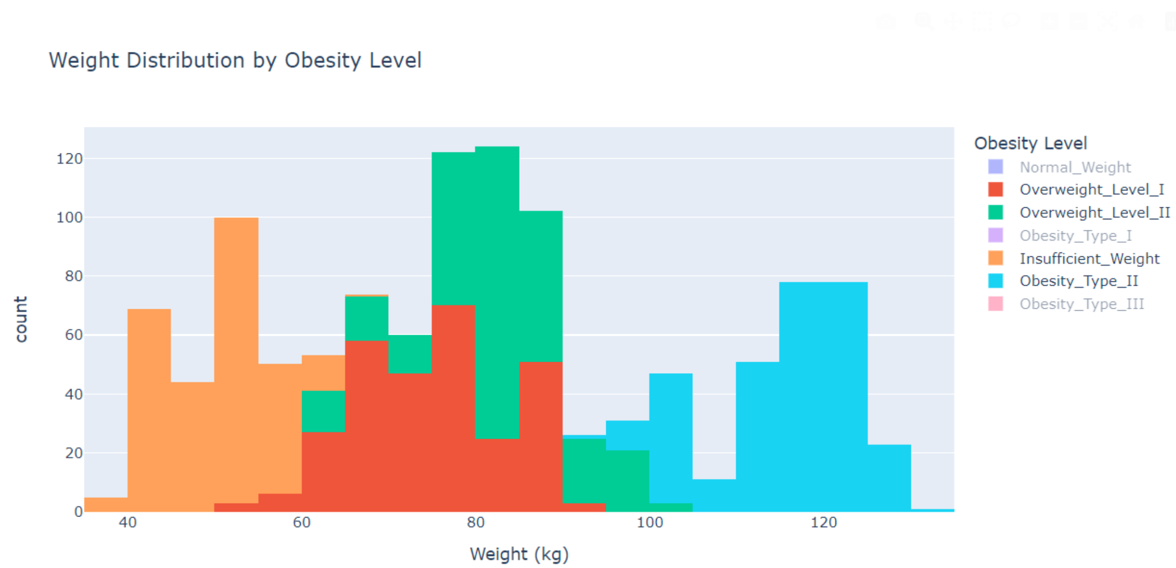
Here each obesity type is represented by different colors for visualization.

User interaction:

As we can see in the second screenshot, it has more interactions where users can. particularly selecting which type of obesity, they want to view. Here we have selected only normal weight, overweight level 2, obesity type-1, and obesity type-3 from the legend on the left side and only those attribute bar graphs are highlighted. We can also see that the average height is the same in all these cases. This interaction enables users to see only the information they want to highlight rather than all the details.

Selection of weight factor





Given the weight, we can group the weights into weight ranges and count the people falling into each category. The bar graph shows that the maximum number of people who have a weight 50-80 are obesity type-1, while the least people who weigh above 130 fall into obesity type-3.

User interaction:

In the above screenshot, we have selected only overweight levels 1 and 2, insufficient weight, and obesity type-2 from the legend on the left side and only those attributes are highlighted in the graph. This interaction enables users to see only the information they want to highlight rather than all the details.

Selection of Family history with an overweight factor

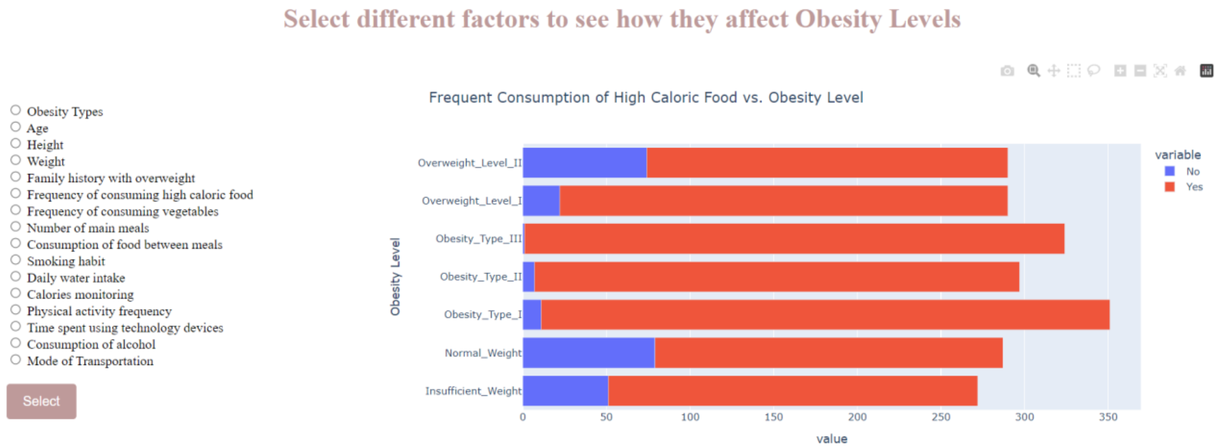
Screenshot: Family history with an overweight factor

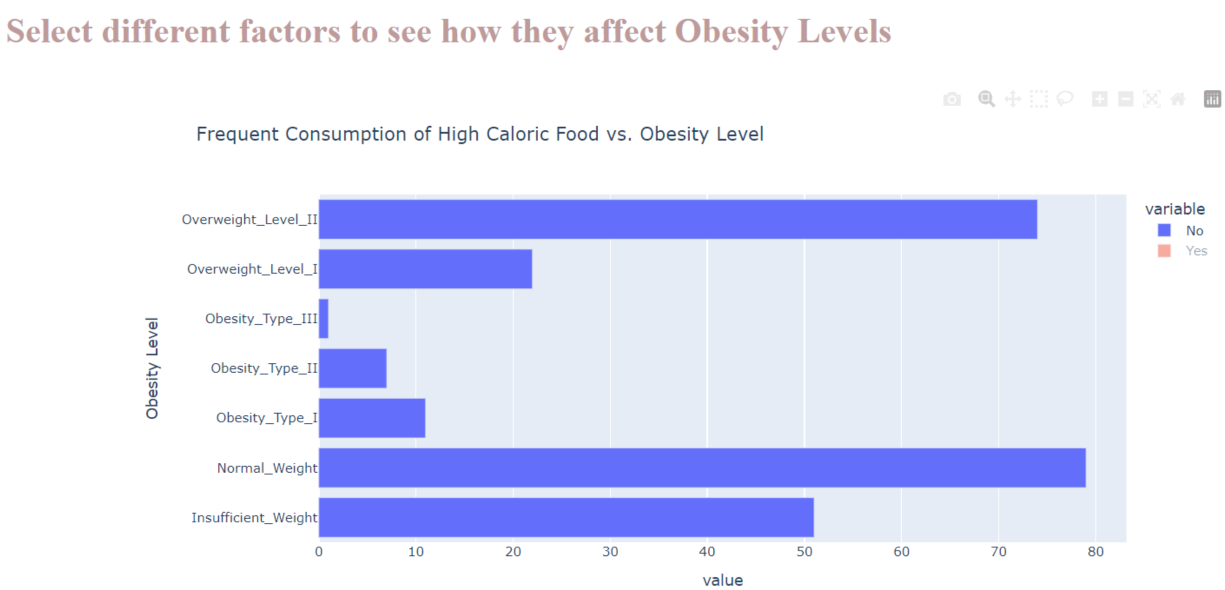
The bar graph consists of family history with an overweight factor which is yes or no depicted in different colors, orange color for yes, and blue representing no. We can see in the bar graph that the maximum number of people who come in the obesity type-1 category and their response to the question of having a family history of being overweight is yes while for insufficient weight 270 people responded both yes and no.

User interaction:

In the above screenshot, we have two options in the legend, yes or no. When we click on the yes option from the legend, the attributes that have a family history of being overweight are alone heightened in every obesity category rather than distinguishing the bar graph into yes or no. Users can alone view all the data points having a yes value or no value. This will help the user make quick decisions and analyze the graph in less time.

Selection of frequency of consuming high caloric food factor





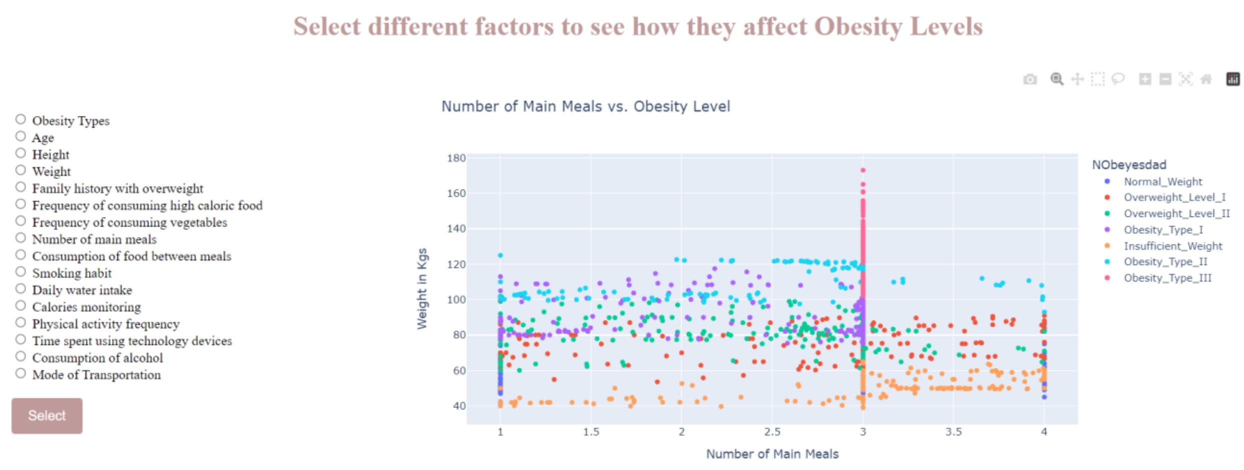
Screenshot: Selection of frequency of consuming high caloric food factor

It is used to depict the density and distribution of a continuous variable. Here we plot the frequency of high-caloric food vs obesity on the violin graph. The x-axis consists of the obesity level and the y-axis consists of the frequency of high-caloric food. As seen in the above screenshot the kernel density estimates the data for both the attributes, frequency of consuming high-calorie foods within each obesity level group that has a corresponding violin plot for every obesity level category on the x-axis.

User interaction:

In the above screenshot, we have two options in the legend, yes or no. When we click on the yes option from the legend, the attributes that have a family history of being overweight are alone heightened in every obesity category rather than distinguishing the bar graph into yes or no. Users can alone view all the data points having a yes value or no value. This will help the user make quick decisions and analyze the graph in less time.

Selection of several main meal factors



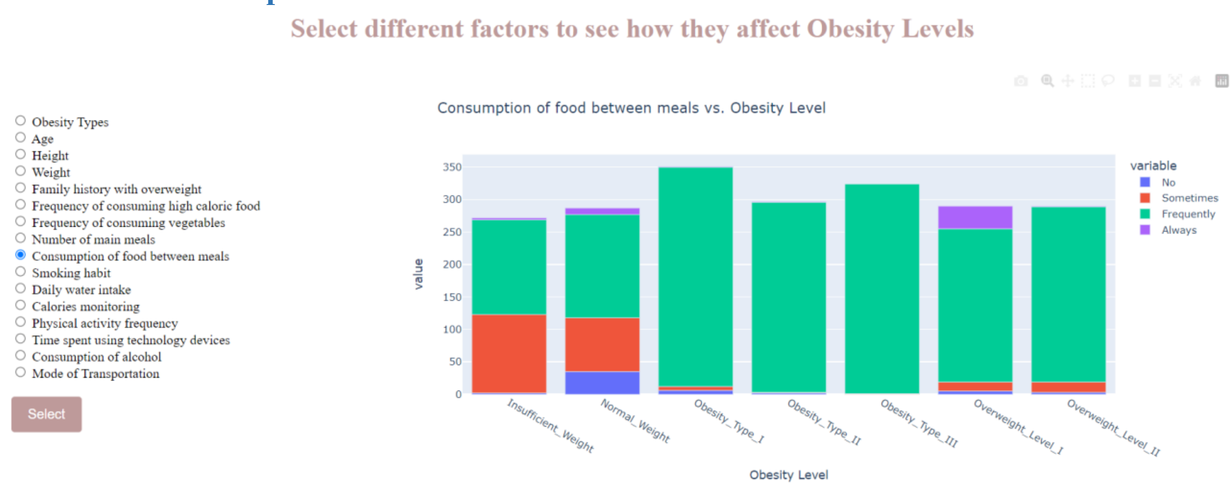
Screenshot: Selection of several main meals factor

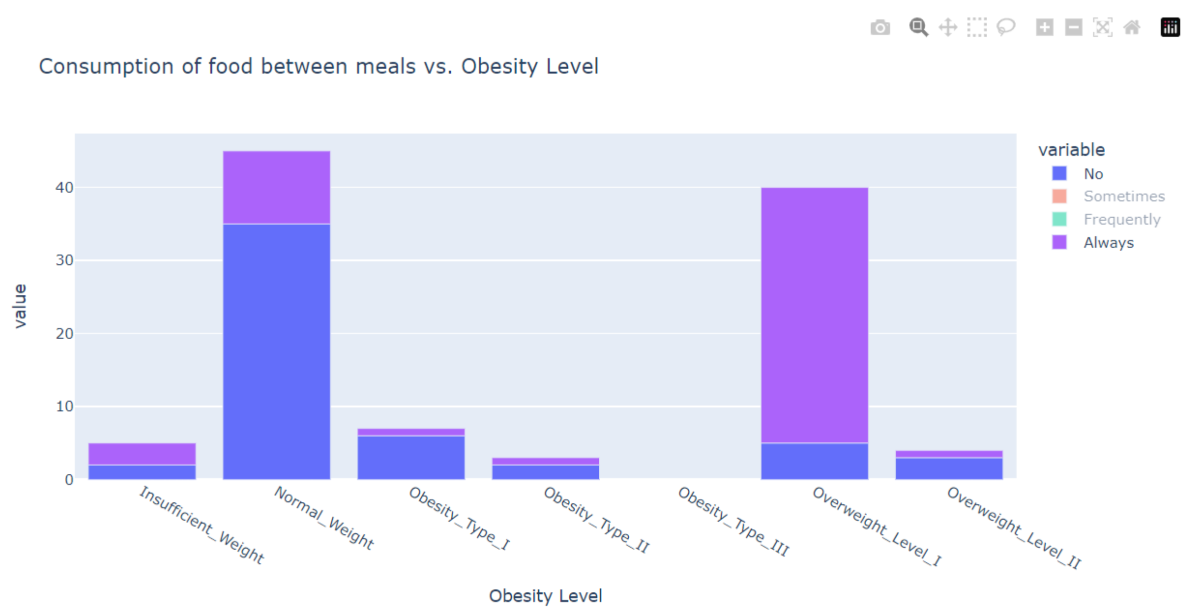
The scatter plot shows the number of main meals vs obesity level. The x-axis consists of the number of main meals and the y-axis consists of weight in kgs. As observed, people who come under the category of overweight level-1 and level-2, and insufficient weight eat more than 3 times a day and the rest of the people in other categories eat less than 3 times a day. So, we can conclude from the graph that most people who eat more than 3 times a day are obese or underweight.

User interaction:

In the above screenshot, we can see that the scatter plot marks of only normal weight and overweight level-2 are highlighted. The scatter plot is then given a transform_filter to display only the chosen data points from the legend on the left side. The maximum data points are for overweight level-2 from the specific attributes which are selected.
Focusing on specific data points or categories of interest can be accomplished by selecting specified features from a legend and only viewing them in the scatter plot.

Selection of consumption of food between meals factor





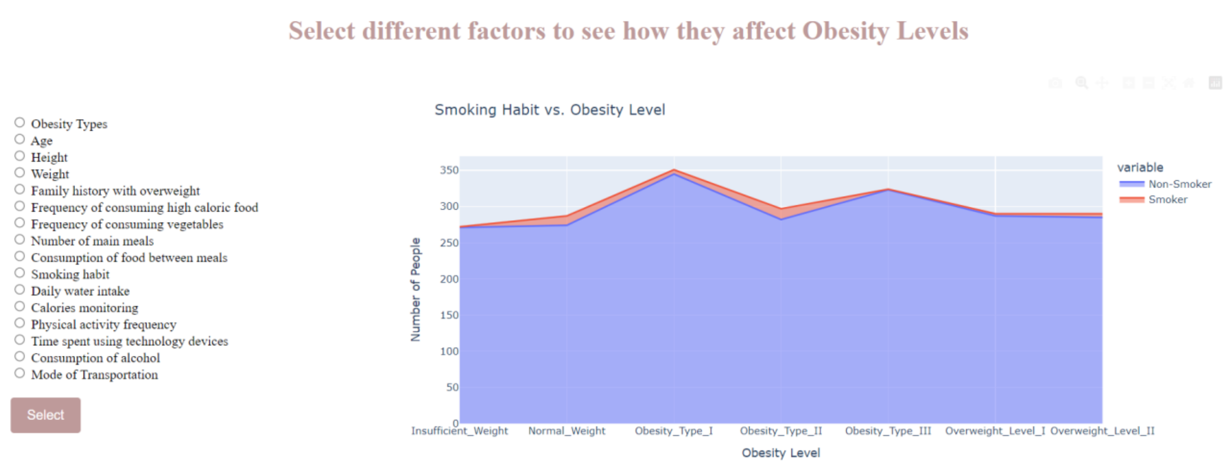
Screenshot: Selection of consumption of food between meals factor

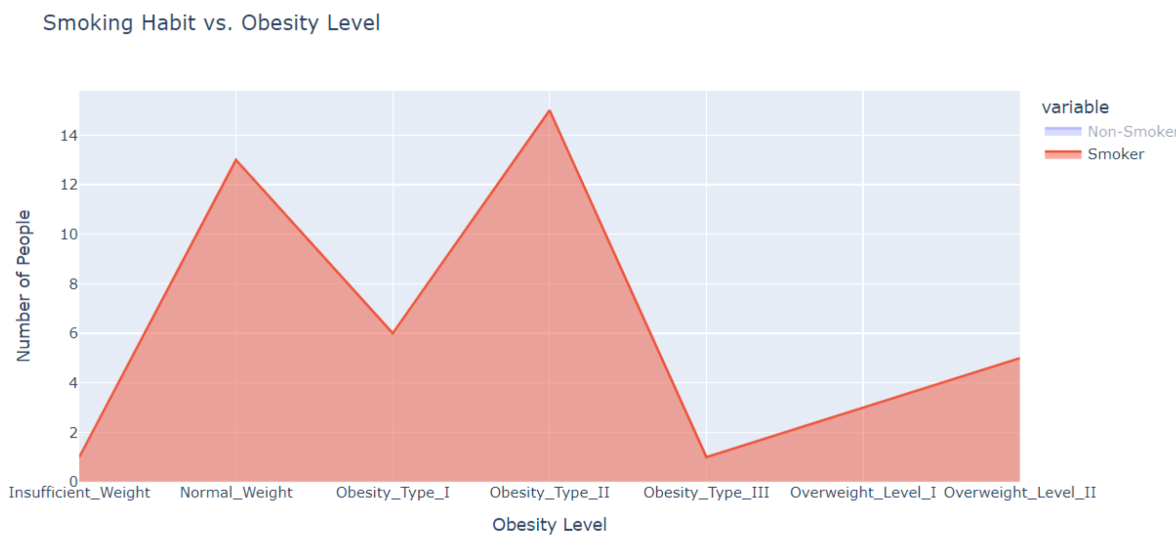
The above screenshot consists of a scatter plot that shows the relation between the number of meals and the weight of the person, and also the data points representing obesity level. The weight is plotted on the x-axis and the number of meals is plotted on the y-axis. To represent each obesity level corresponding to the data points, every category is assigned a color as shown on the left-side legend. This enables the user to visually identify the trends in the dataset. As seen, the people consuming 3 meals in a day are the most weighted and fall in the category of obesity type 1. While the least consuming meals are weight between 80-100 and fall in overweight type 2.

User interaction:

In the above screenshot, the legend has four variable attributes no, sometimes, frequently, and always. The user selects the variables ‘no’ and ‘always’ and only those obesity categories are displayed in the bar chart. We can observe that the obesity type –3 has no values consisting of the ‘no’ and ‘always’ attributes. The highest data points having these attributes are in normal weight and overweight level 1.

Selection of smoking habit factor





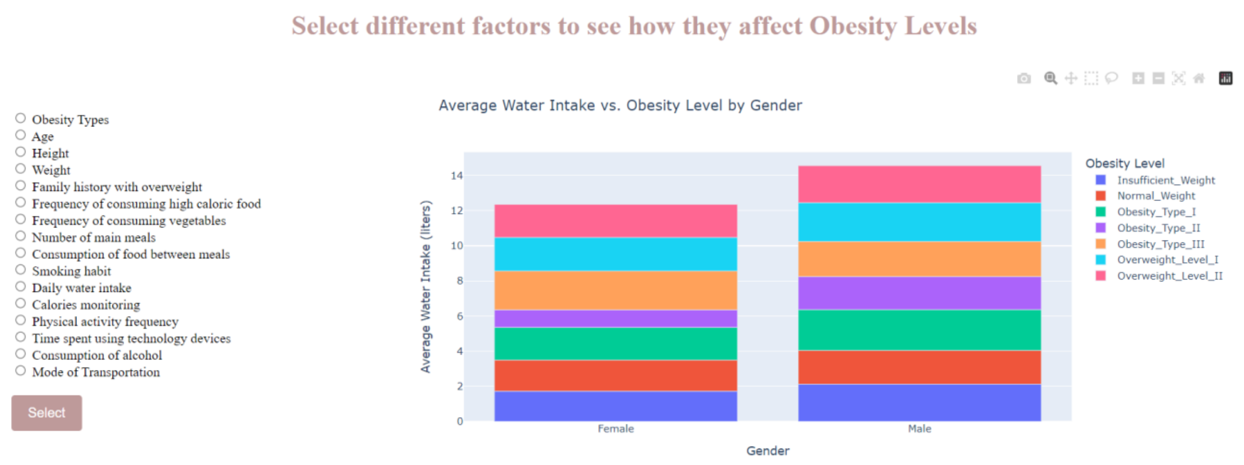
Screenshot: Selection of smoking habit factor

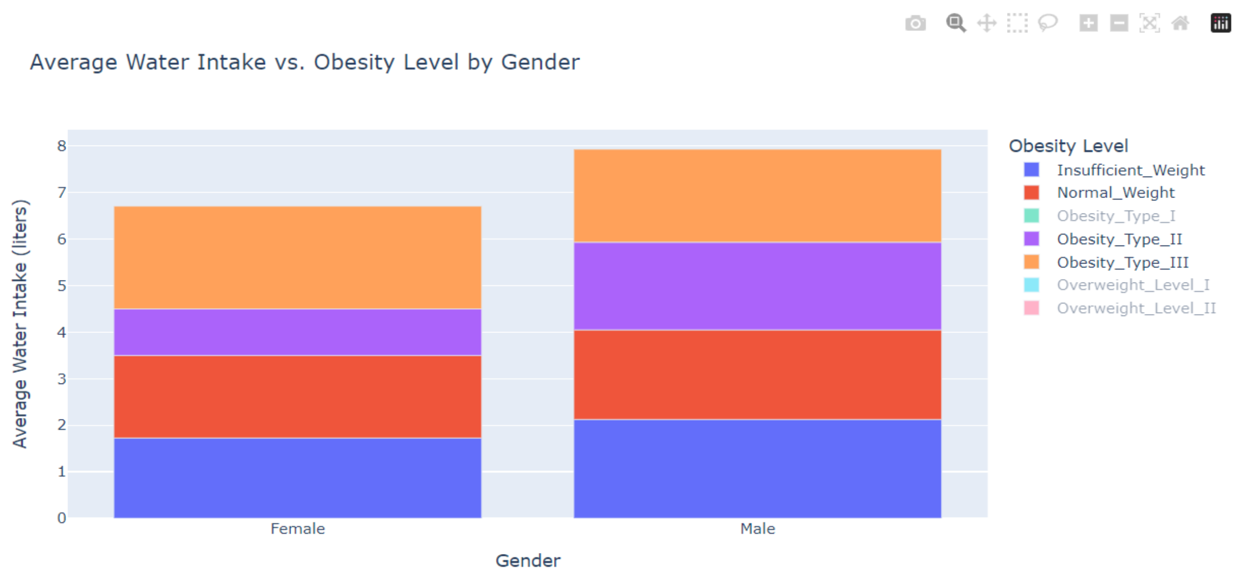
The area graph shows the smoking habit of people with the corresponding obesity level they belong to. As seen, the greatest number of people are non-smokers. While people who smoke fall in the category of obesity level-1 and level-2.

User interaction:

For the area graph, we selected only the smoker attribute to be displayed in the graph. Here users can specifically check the number of smokers in each obesity category. This enables the user to distinguish between smokers and non-smokers in different categories independently.

Selection of Daily water intake factor





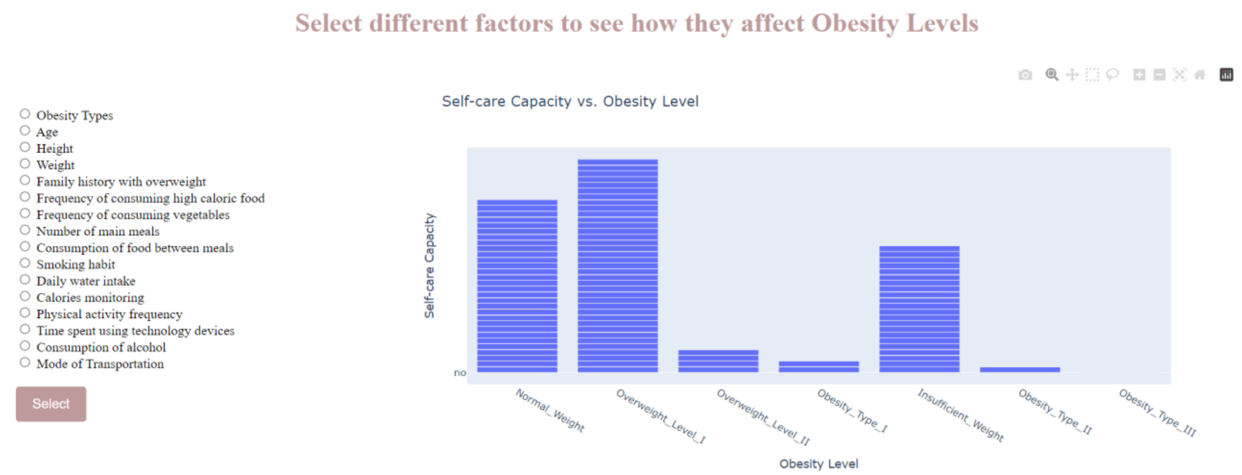
Screenshot: Selection of Daily water intake factor

The stacked bar graph shows the average water intake by females and males. As seen, the average intake of water is slightly higher for males than females. A person of normal weight consumes 2-3 liters of water whereas people belonging to the overweight type-2 category consume more than 12 liters of water. We can identify the water intake comparison between both females and males in different obesity categories.

User interaction:

For the stacked bar graph in the above screenshot, we selected only the specific obesity level from the legend which are insufficient weight, normal weight, and obesity type 1 and 3. The same is replicated in the graph, only those attribute data points are marked in the stacked bar graph. We can observe that from these specific categories also male has the highest water intake than female.

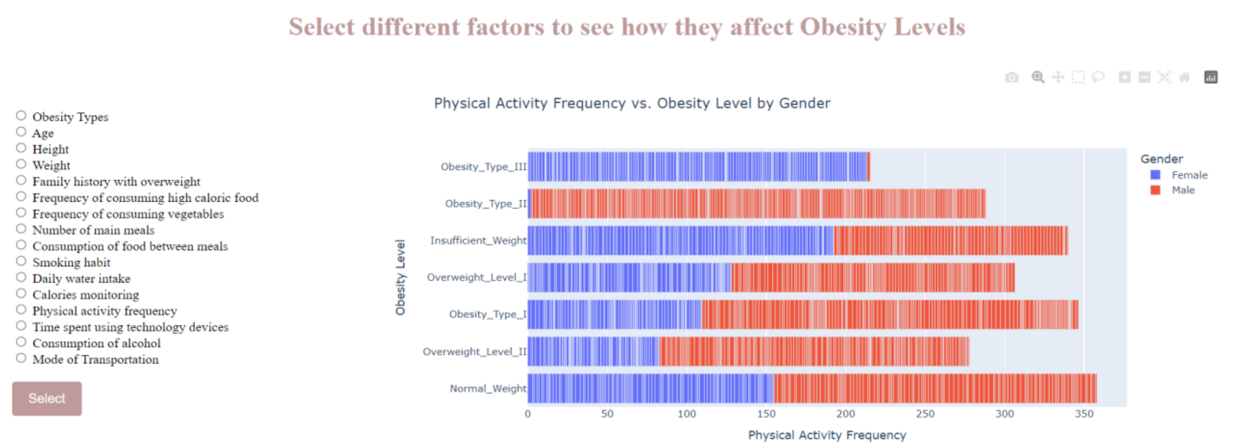
Selection of Calories monitoring factor



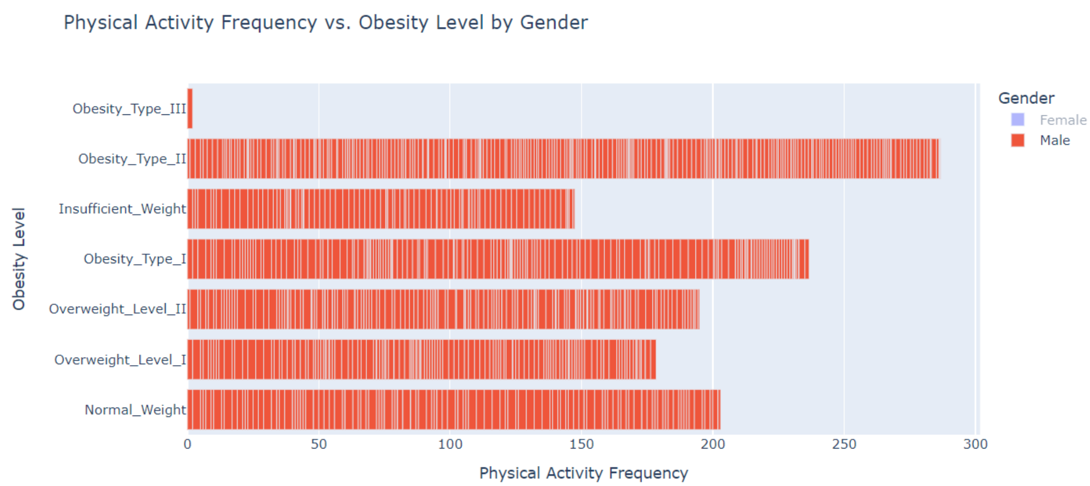
Screenshot: Selection of Calories monitoring factor

In this case, we can see the self-care capacity vs obesity level. Here self-care indicates the number of calories consumed by every person. As shown in the graph, the people belonging to the overweight level-1 have high self-care of calorie intake since they are already obese while the second highest self-care can be seen in normal-weight people. Whereas the least self-care is projected by people belonging to obesity type-3 which is zero. They are prone to becoming obese and easily jump to level -1.

Selection of Physical activity frequency factor



Select different factors to see how they affect Obesity Levels



Screenshot: Selection of Physical activity frequency factor

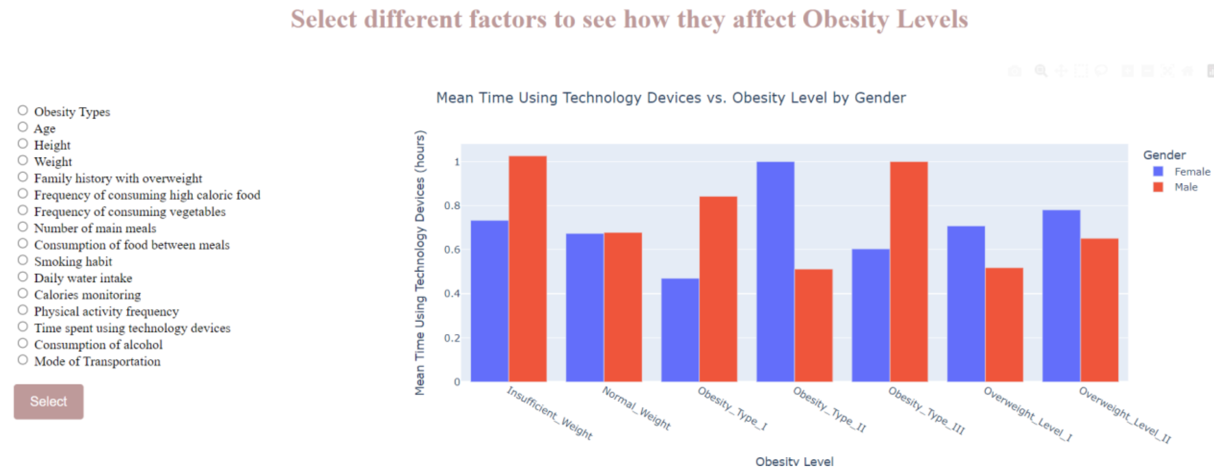
The horizontal scatter plot is used to display the physical activity frequency on the x-axis and obesity level on the y-axis concerning the male and female categories.

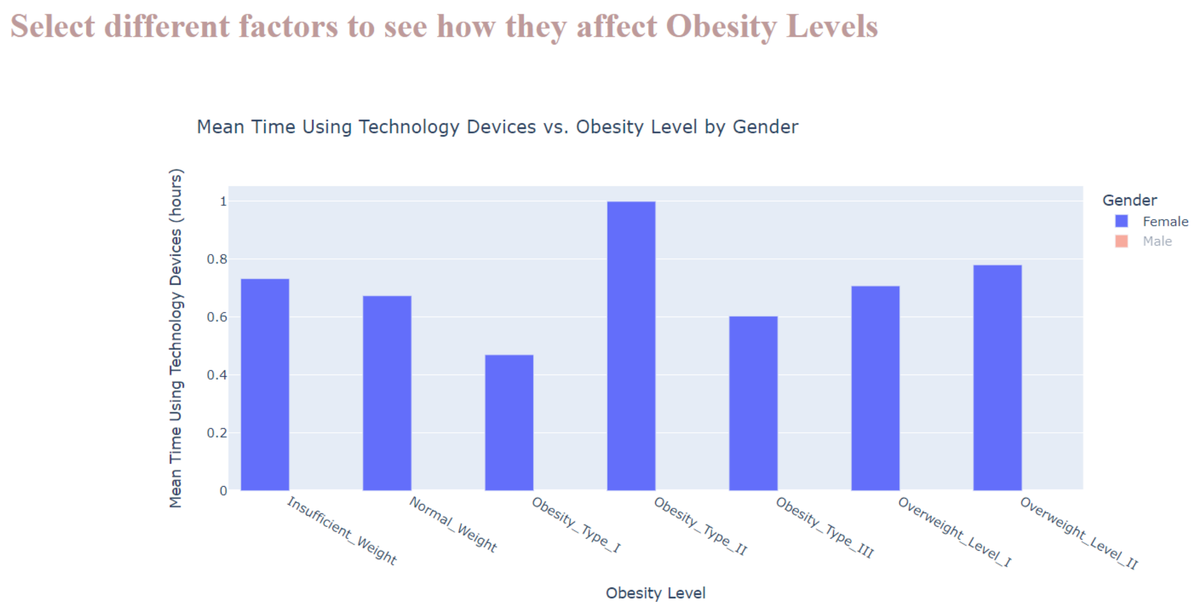
We can observe that the male has more physical activity than the female. Obesity type-2 consists of 99% males, and they have a physical activity frequency of 290 while obesity type 3 consists of females, and they have a physical activity frequency of 230.

User interaction:

In the horizontal bar graph, the legend consists of 2 categories male and female. We can particularly check the physical activity frequency of males alone across the obesity levels by clicking on the male category in the legend which is on the left side of the graph. As seen male who are in obesity type-2 has the highest physical activity frequency followed by male in obesity type-1.

Selection of time spent using technology devices factor.





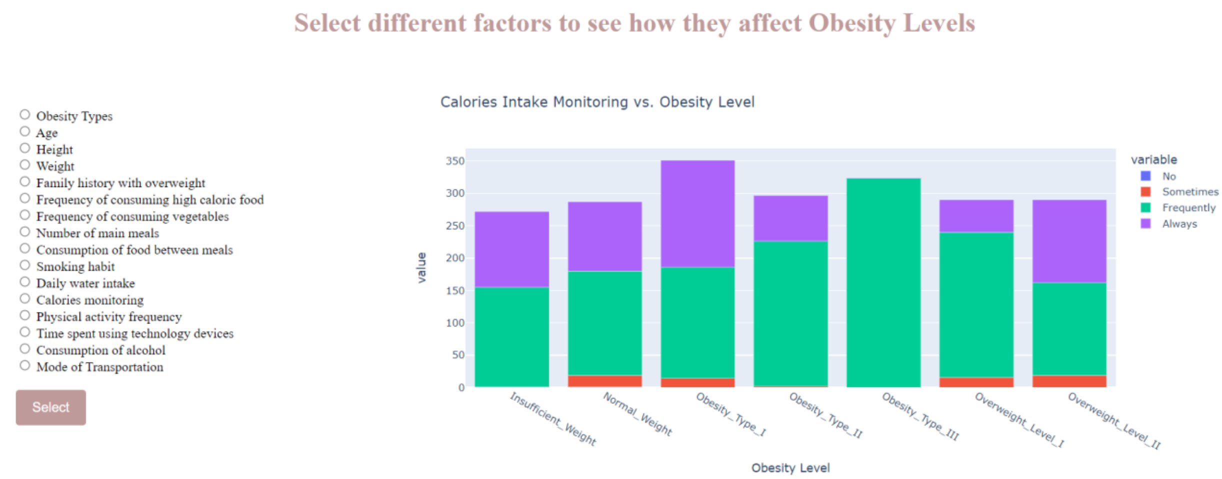
Screenshot: Selection of time spent using technology devices factor

We can observe the meantime using technology devices vs obesity levels in males and females. As seen for obesity type-1 and type-3, males spend more time using devices than females. Whereas obesity type-2, overweight level-1, and level-2 involve more percentage of females using the devices. For normal weight, the average time is 0.6 hrs. The highest time spent is 1 hour by men who fall in obesity 3. The female who falls into obesity type-2 also spends 1hr on average on devices. We can see that the use of devices affects human weight.

User interaction:

The previous graph shows the information of time spent by males and females on devices. If the user wants to check the time spent by females alone and their obesity levels, the user can click on a female category in the legend and the graph displays the time spent by females on technology devices. A female who falls into obesity type 2 spent the maximum time on the devices followed by a female in overweight type –2 categories.

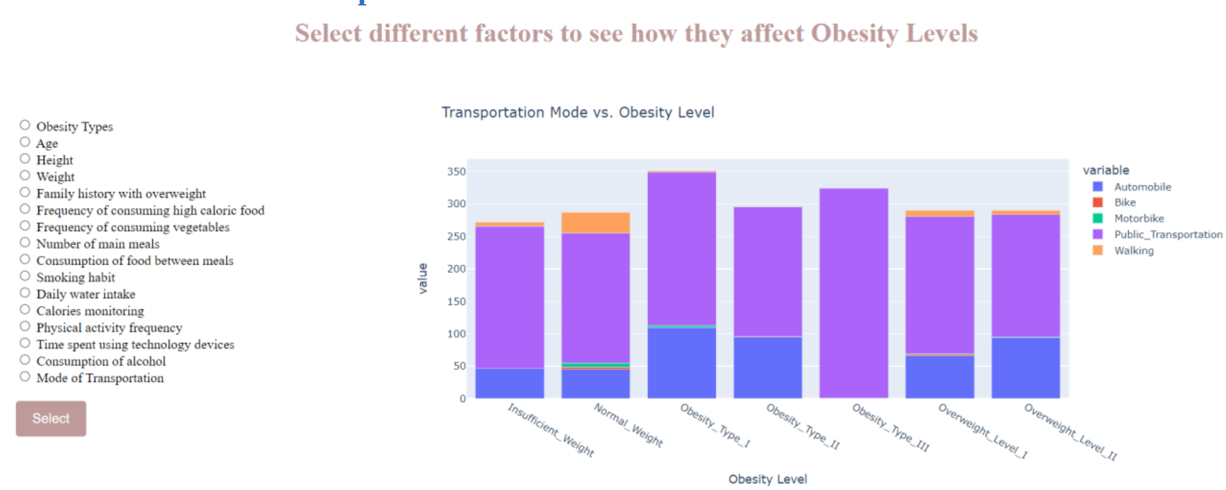
Selection of calories intake factor

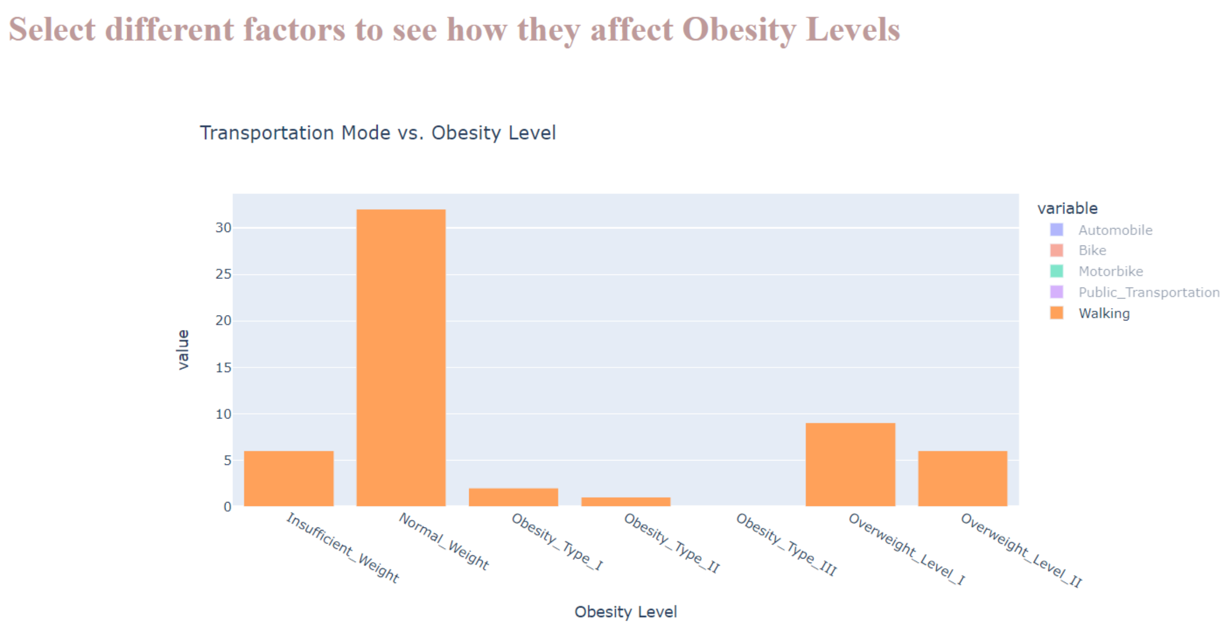


Screenshot: Selection of calories intake factor

The bar graph shows the people's calorie intake in terms of their answers as no, sometimes, frequently, or always belonging to different obesity level categories. As seen, obesity type-1 has the highest calorie intake which is the answer to the calorie intake is always. Whereas the orange color associated with the answer was no, belonging to normal weight, obesity type-1, overweight level-1, and level-2.

Selection of Mode of transportation factor





Screenshot: Selection of Mode of transportation factor

The bar graph shows the mode of transportation automobile, bike, motorbike, public transportation, and walking. We can observe that the mode of transportation has significant effects on obesity, as people belonging to normal weight are walking whereas people who fall in the other obesity levels use public transport.

User interaction:

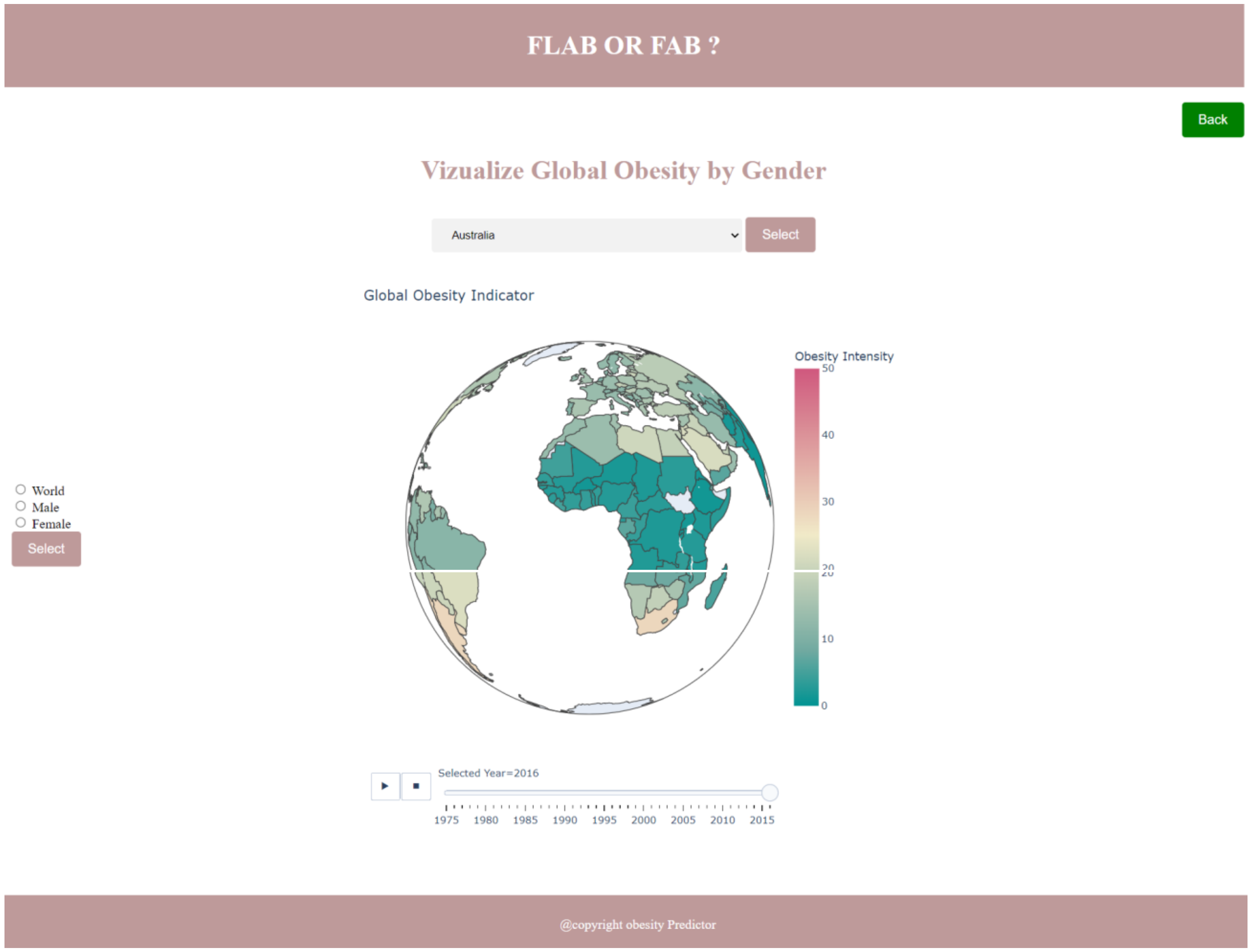
From the above screenshot, we can select only the walking transportation mode and analyze the number of people with a particular obesity level. As observed, a maximum number of people belonging to normal weight walk. We can thus conclude that people who walk more fall in the normal weight. Selection helps to analyze each attribute separately and provide detailed insights.

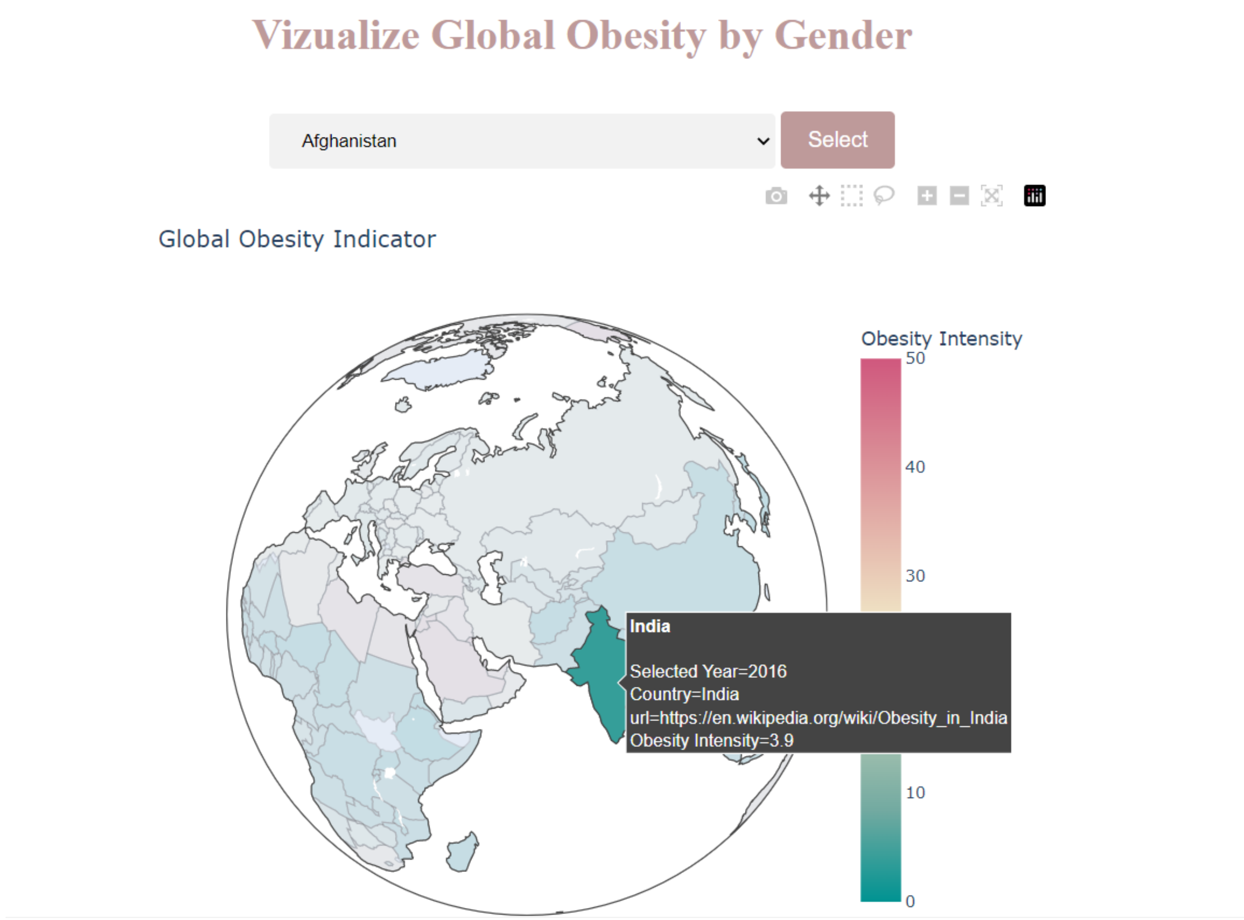
3.5.1.4 World Obesity by Gender

We can use a global map to display the gender-specific obesity rates in various countries. The user can choose the year they want to view by using a global indicator on the map. Each country's level of obesity can be determined by the hue of the globe.

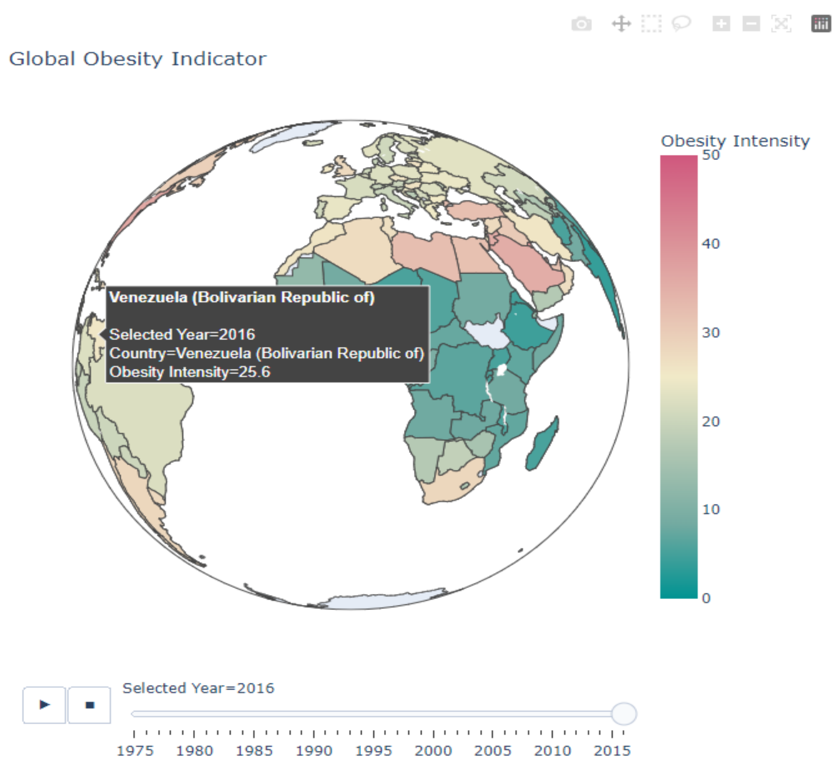
To make it simpler to identify each country, a distinct color shade is used to represent each one on the map. The obesity rate in that country increases with the color's intensity. For instance, a country with a high percentage of obesity will have a darker shade, whereas a country with a low prevalence of obesity will have a lighter shade of color.

s


FLAB OR FAB ?
Back
Vizualize Global Obesity by Gender
Australia
Select
Global Obesity Indicator
Obesity Intensity
50
40
30
20
10
0
World
Male
Female
Select
Selected Year=2016
1975
1980
1985
1990
1995
2000
2005
2010
2015
@copyright obesity Predictor



Screenshot: Visualization of global obesity by Gender

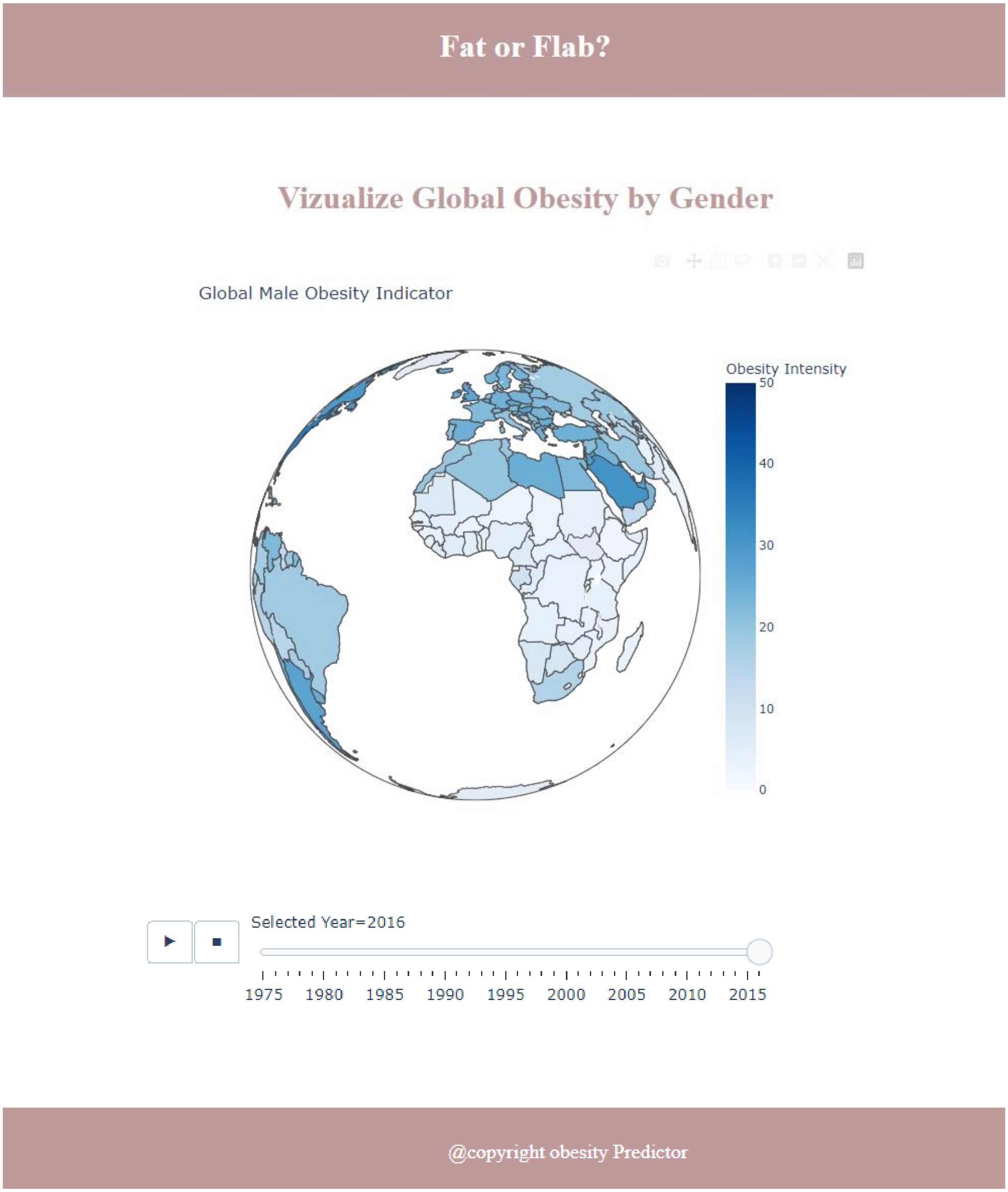


Screenshot: World obesity global indicator

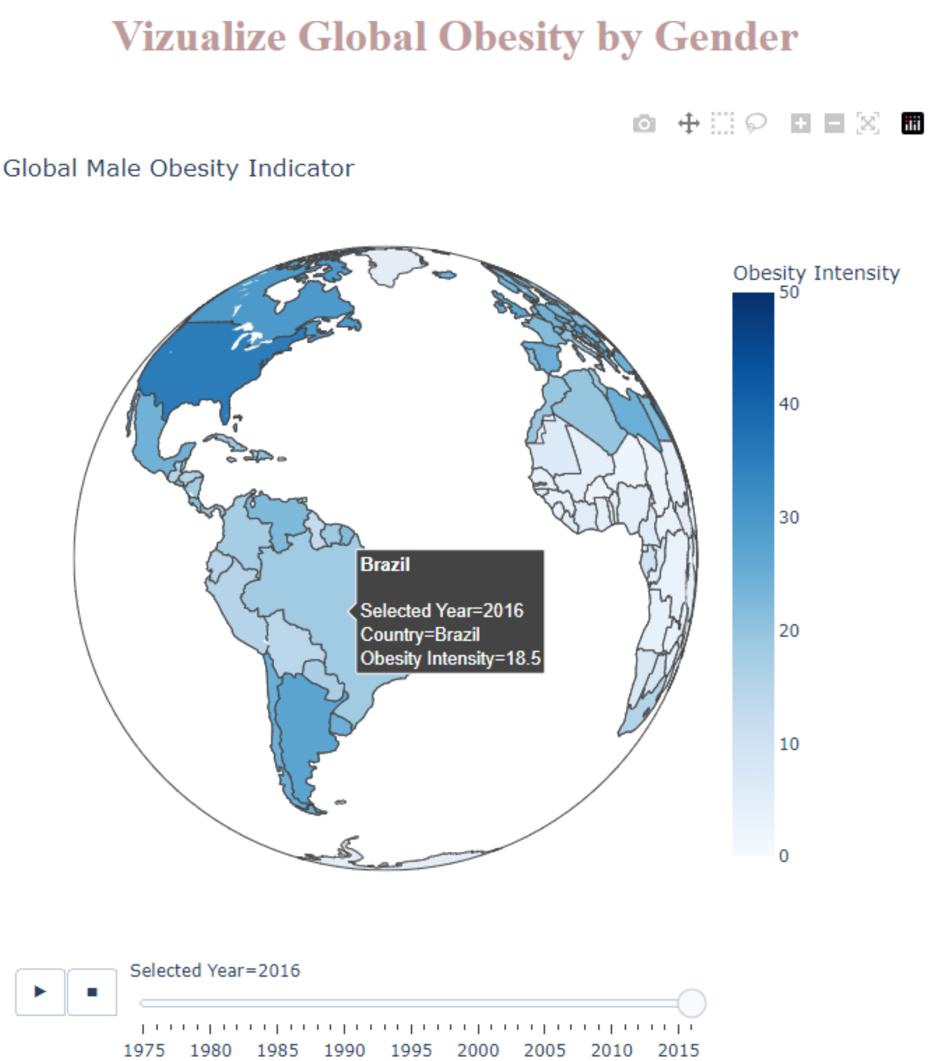
As seen in the screenshot above we have 3 options to choose from on the left side, we can view the world obesity intensity, female and male intensity alone.

The image shows the selected "world" option shows the obesity intensity of the whole world. The color indicates the intensity of obesity as shown in the line meter on the left side. The intensity increase concerning the color shade and is represented by the values and color intensity. We can also see the details of the countries when we hover over the countries. As shown, we can see the information on Venezuela's country with selected year, country, and obesity intensity level details.

Male



Screenshot: Visualization of global obesity by Gender (Male)



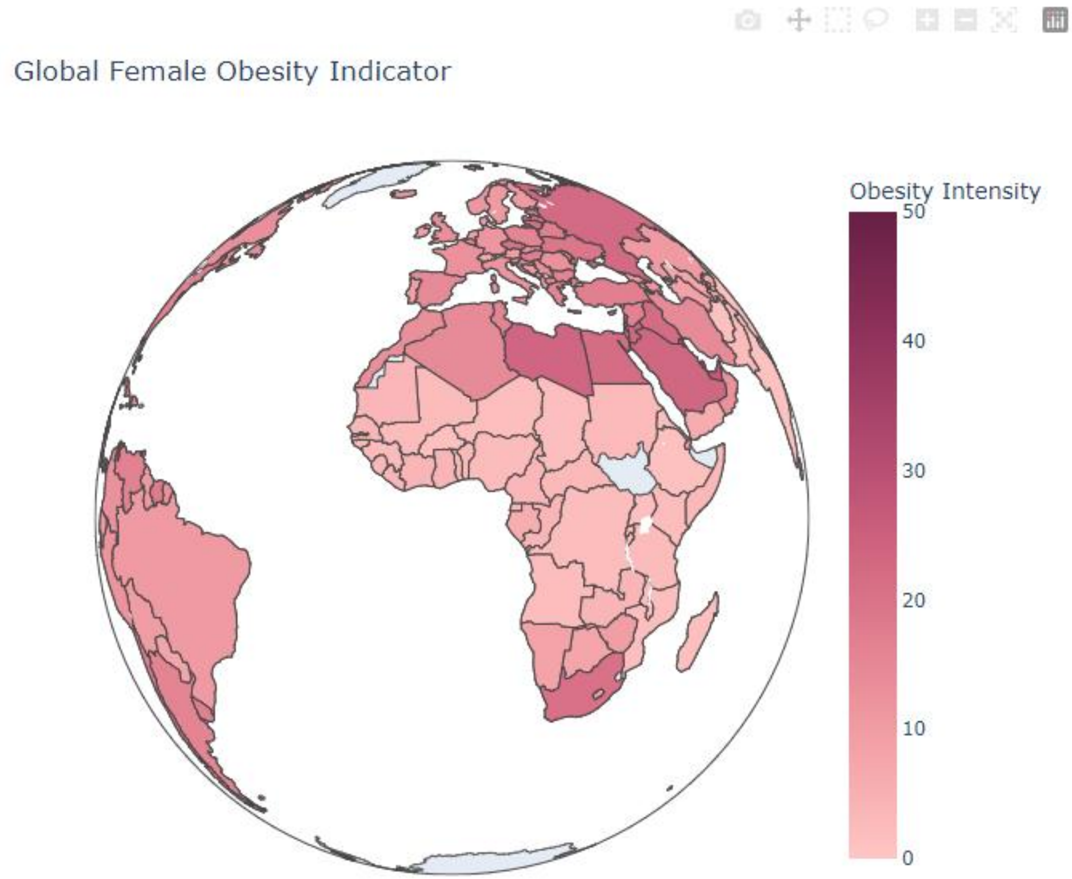
We can particularly select gender. Here when we click on male, the globe map shows the intensity of obesity in men across different countries. As seen here countries like India has the lowest obesity rate and America has the highest obesity rate for male.

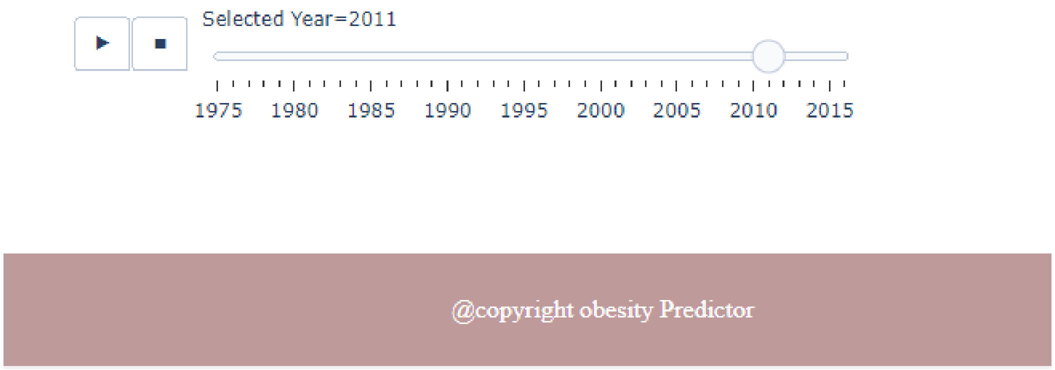
The tooltip gives information about Brazil. Here we can see the severity of obesity in males in Brazil alone. The obesity intensity of females in Brazil is on the lesser side of the intensity meter which is 18.5.

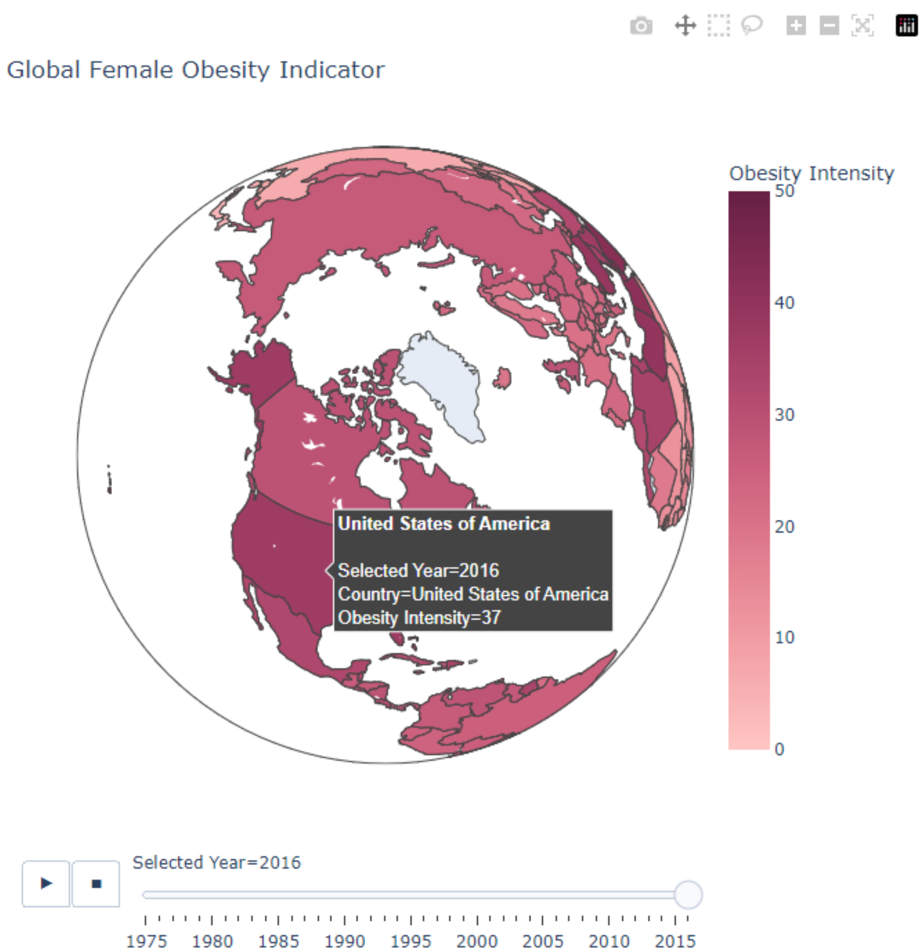
Female

Fat or Flab?

Vizualize Global Obesity by Gender





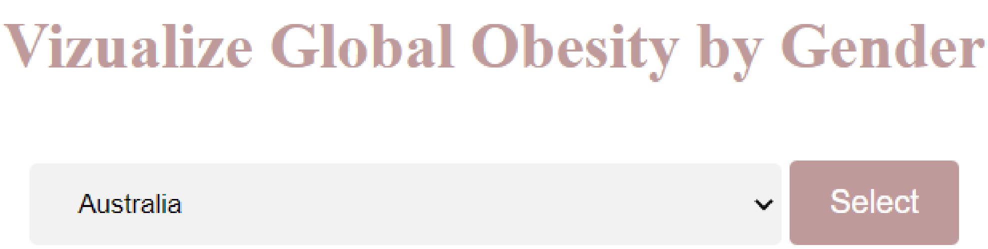


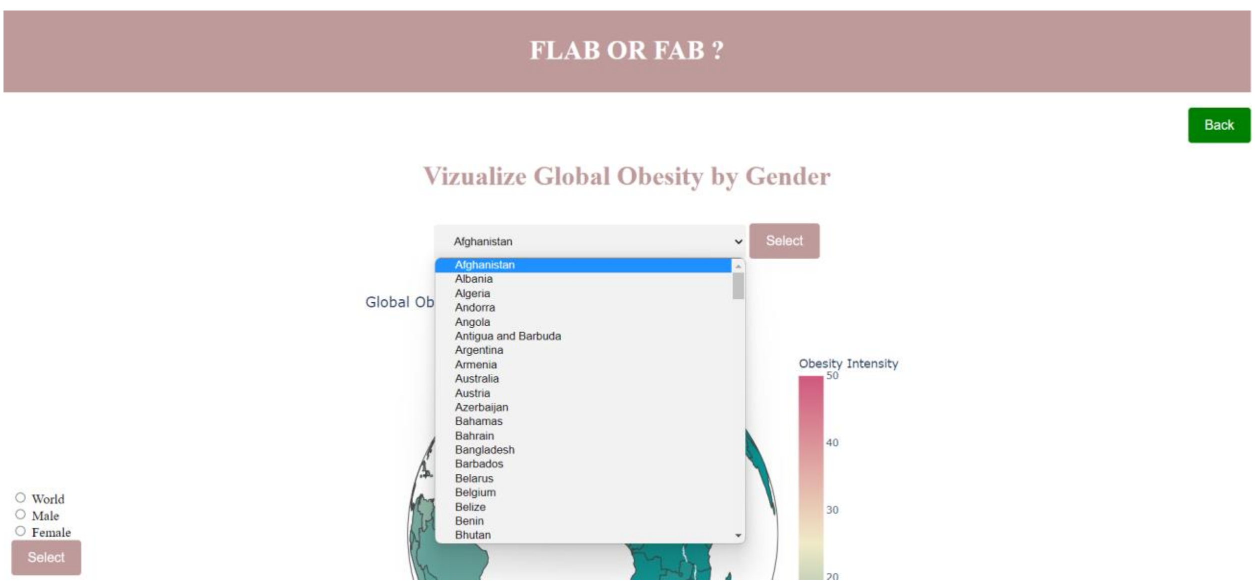
Screenshot: Visualization of global obesity by Gender (Female)

We can specifically choose the gender. When we select "Female" in this section, a world map displays the prevalence of obesity among females in various nations. It can be observed, the map shows the intensity of obesity in the African continent.

The screenshot shows the tooltip of the intensity of obesity in the USA, the intensity is comparatively very much higher in America showing 37 intensity measures.

3.5.1.5 World Obesity by Specific Country



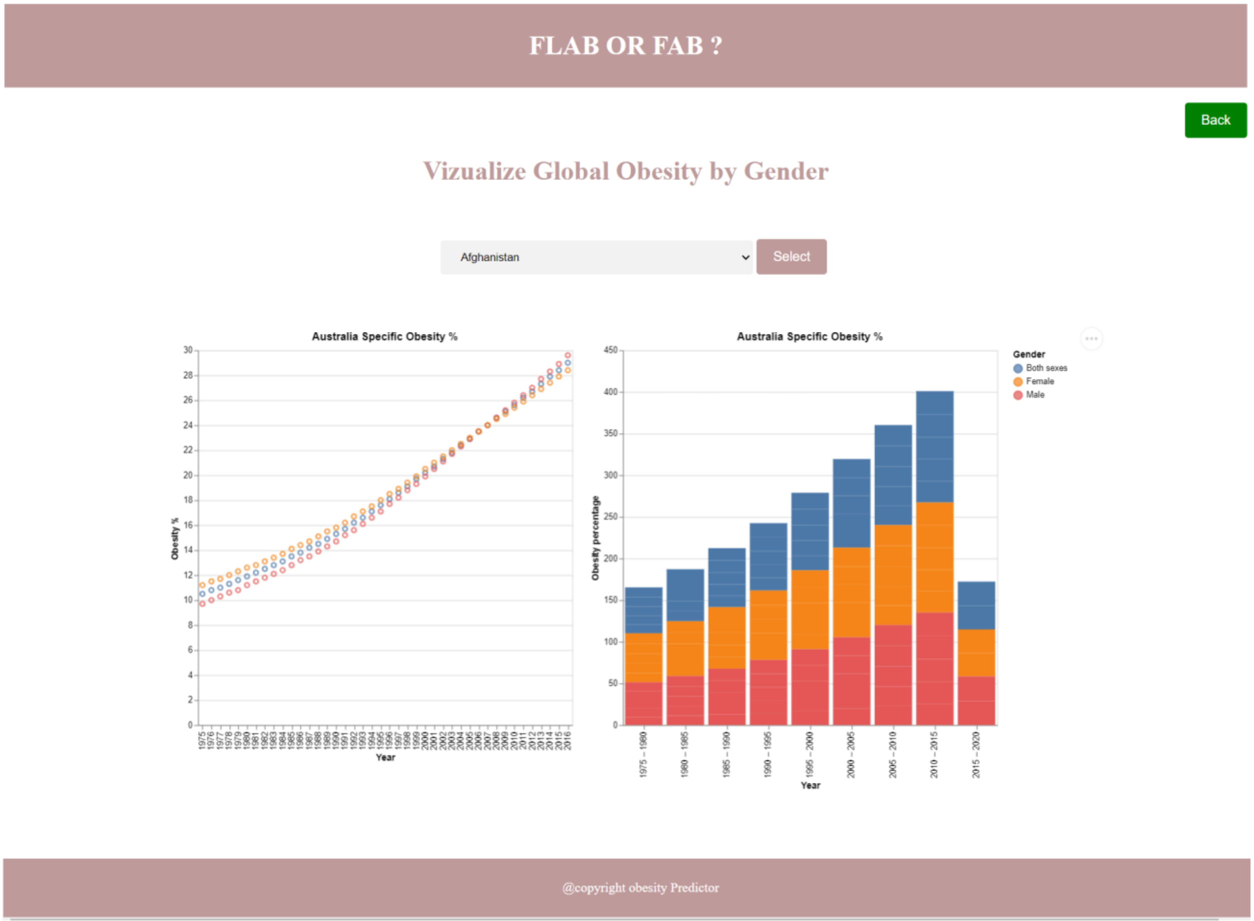


These screenshots show the use **of brush, linking, and adding href as a tooltip** to the visualizations.

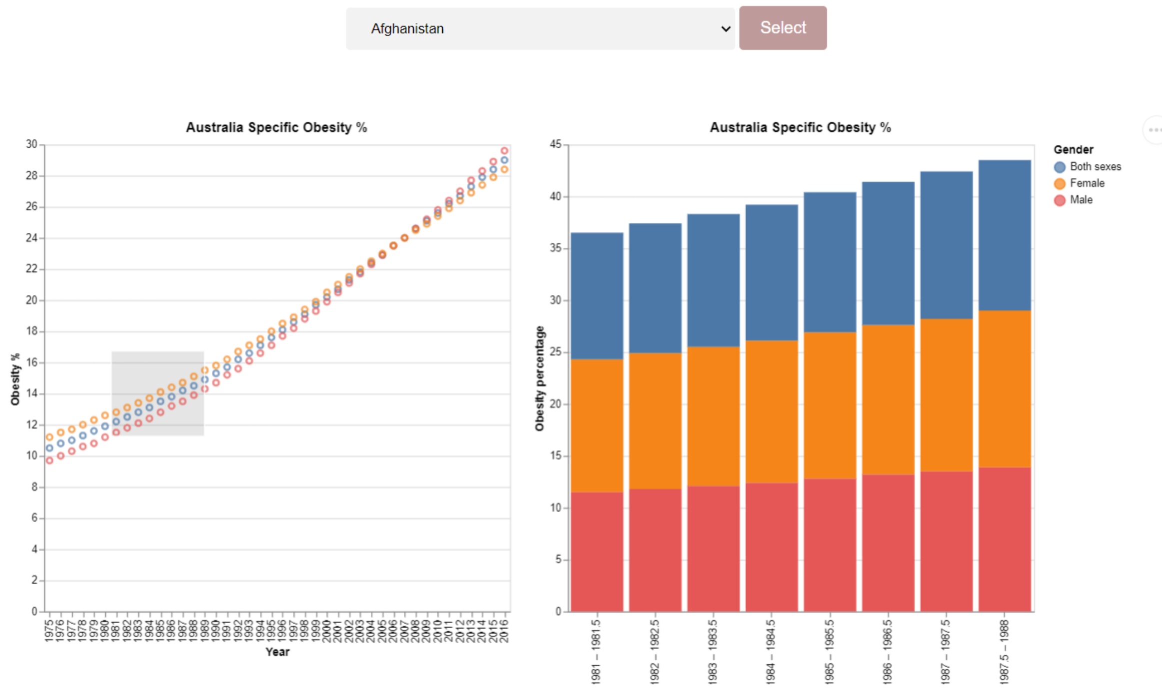
Firstly, in the above screenshot, we are using a drop-down menu and giving the option of choosing different countries of the world from the drop-down menu. And we are giving the selection buttons to select either the male or female category.

In the above example, we are using Australia as a country.

And in the background, we have the globe of the world map which is a choropleth map that shows luminous and saturation intensity coded color map to show the obesity percentage intensity of different countries of the world.

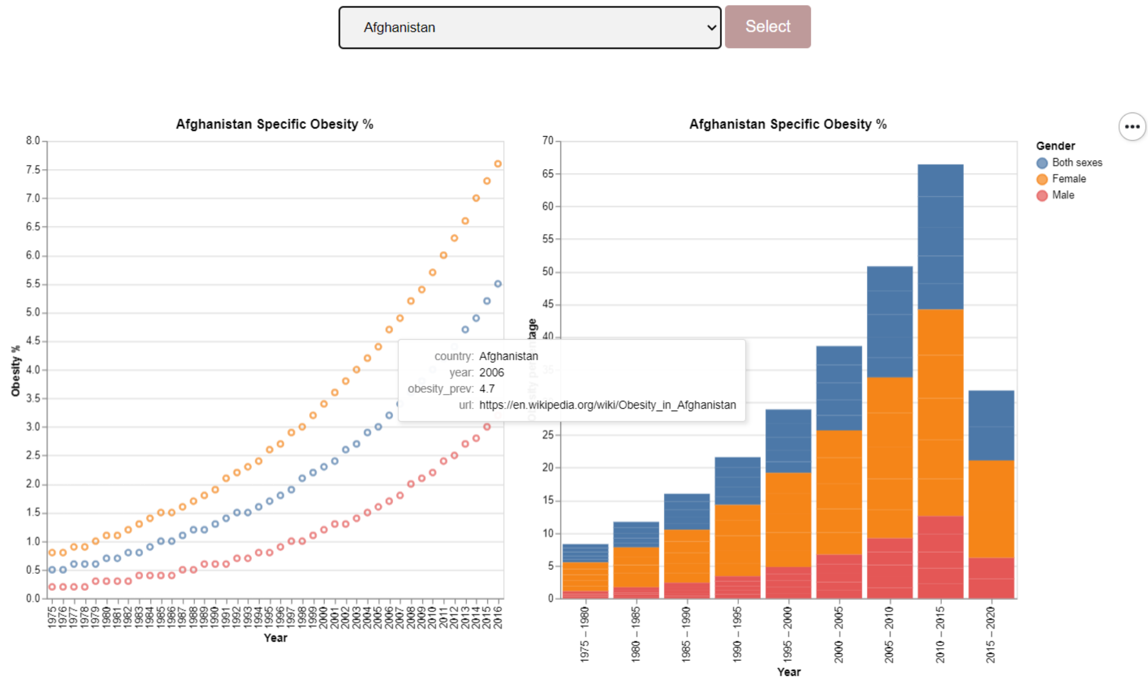


In the above screenshot we represent the scatterplot and the bar graph that will be used further for brushing and linking and to create the href-linked tooltip. For the scatterplot, we have used the obesity percentage data on the y-axis, and on the x-axis, we are using different years from 1975 to 2016. Similarly, for the bar graph, we are using corresponding years from 1975-2016 on the x-axis and we are using obesity percentages on the y-axis, and we are representing the data for two genders male and female respectively and both, the red color shows male obesity percentage and orange color shows female obesity percentage and blue color shows both. In the above graph we can see that when we choose Australia then the obesity percentage has risen in Australia from 1975 to 2016 from 12% to 30%.then in the vertical bar graph we can see that obesity in males and females increased the highest during the years 2015-2016 and the lowest increase was during the years 2018-2020.

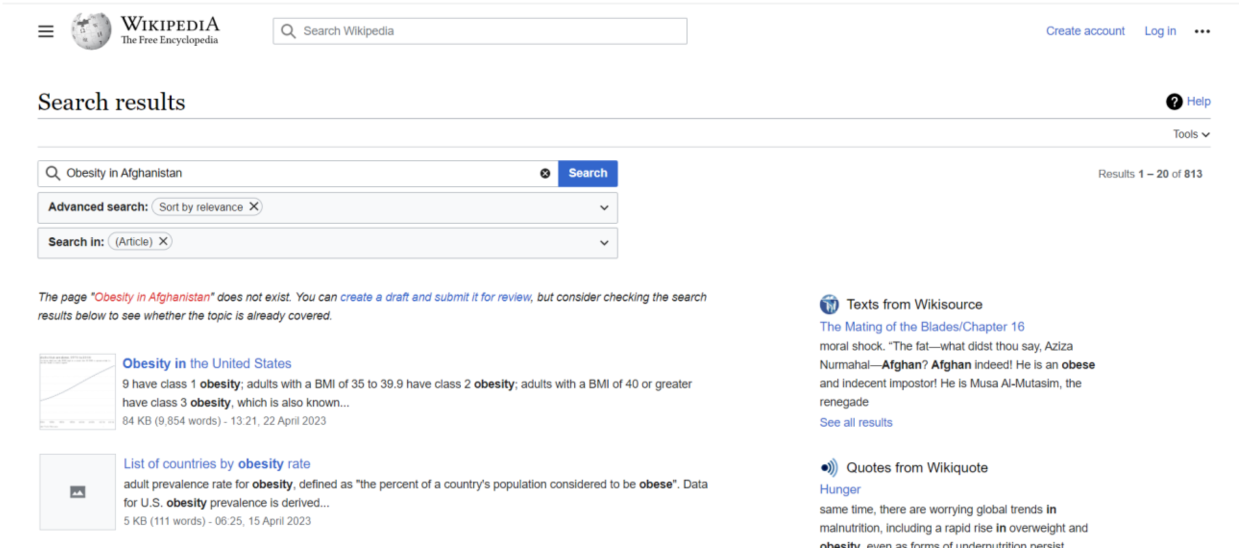


In the above screenshot, we show the use of brushing and linking the scatterplot and the bar graph such that when we zoom and select a portion of the scatterplot then the corresponding effects are shown in the bar graph. When we select a small portion of the scatterplot, then the bar graph shows the obesity percentages of females and males during the selected corresponding years from 1981 to 1988. So, we can see that for the year 1981 the male obesity percentage is the highest at around 12% and the female obesity percentage is around 10 % and both percentages are around 15 %.
Similarly, the bar graph shows the obesity percentages for all the zoomed-in years. The bar graph shows that for the years 1987-1986, the female obesity percentage was 15 % and the male obesity percentage we 12 %, and both percentages were around 15 %.
The advantage of using brushing and linking is that it makes the two plots interactive, and it makes it easy for the user to focus on just the required no of values rather than interpreting the whole graph.

In the above screenshot we show the addition of the href tooltip to the scatterplot such that when we click on the href URL in the tooltip, it directly redirects us to the URL page shown in the tooltip. Below is shown an example, when we click on the href URL in the tooltip for Afghanistan it directly redirects us to the link that is the Wikipedia page showing the details of Afghanistan as shown below., and thus we can know more additional details a href redirected URL site. For the code we have used Altair and we are using the tooltip command and giving the href link and the URL page.



Similarly, we have one more screenshot showing the brushing and linking feature used for visualization. In this screenshot, we are selecting the years from 2000 to 2010. So, as it has become easy to read and interpret the two graphs simultaneously, we can easily read and interpret that year 2009-2009.5 saw the largest increase in the obesity percentage in males and females, and the year 2002-2001 saw the lowest increase in the obesity percentage.
Thus, linking and brushing have made both the subplots interactive and easy to interpret.
For the code we used Altair and in that we used the brush command and then we added and linked it to the bar graph as well as the scatterplot.



3.6 Story Telling

3.6.1 Who, what, where?

One of the biggest health issues in the world today is obesity, which no longer just affects wealthy nations but affects people of all income levels. Every year, 4.7 million premature deaths are caused by obesity. Obesity, which is characterized by having a high body mass index, increases the chance of developing several of the world's major killers, such as heart disease, stroke, diabetes, and different types of cancer. None of these health effects are specifically caused by obesity, although it can raise the likelihood that they will manifest.[6]

The Global Burden of Disease report states that in 2017, 4.7 million premature deaths were attributed to obesity to put this into perspective, nearly four times as many people perished in traffic accidents, and nearly five times as many people died of HIV/AIDS in

2017. Obesity caused 8% of deaths worldwide in 2017; this is an increase from 4.5% in 1990. Around the world, this percentage varies greatly.[6]

More than 15% of deaths in numerous middle-income nations in 2017 were related to obesity, especially in Eastern Europe, Central Asia, North Africa, and Latin America. This is probably because we have a high prevalence of obesity but inferior general health and healthcare systems in comparison to high-income nations with comparable high levels of obesity.[7]

This percentage ranges between 8 and 10% in most high-income nations. The percentage of many middle-income countries is roughly half of this. In contrast to other wealthy nations, only about 5% of premature deaths in Japan and South Korea are related to obesity. Less than 5% of fatalities in low-income nations, particularly in Sub-Saharan Africa, are caused by obesity.[6]

Both for men and women, the WHO Americas area continues to have the highest obesity rates. However, the numbers in Africa are likely to quadruple by 2030, from 8 million males (2010) to 27 million men (2030), and from 26 million women (2010) to 74 million women, whilst the Americas are predicted to see a 1.5-fold increase between 2010 and 2030. (2030).[7]

Almost 75 percent of all overweight children worldwide reside in Asia and Africa. About two-thirds of children under five who were overweight or obese in 2019 lived in Asia, and the number of overweight children under five in Africa has increased by almost 24% since 2000.[7]

A survey from the Global Obesity Federation predicts that by 2035, more than 4 billion people, or 51% of the global population aged 5 and older, would be overweight or obese. In contrast, 38% of the world's population, or 2.6 billion individuals, were overweight or obese in 2020. By 2035, 24 percent of the world's population, or over 2 billion people, are predicted to be obese, up from 14 percent in 2020. The rate of obesity is forecast to double in boys, from 10 to 20 percent, and more than double in girls, from 8 to 18 percent, among children and adolescents aged 5 to 19, representing the sharpest increase.[8]

This indicates not only a reduction in the quality of life for those who are obese but also higher direct medical costs and indirect costs from lost productivity because of associated illnesses. These expenses are thought to total EUR1.8 trillion annually or 2.8% of world GDP. If the obesity trend is not reversed, health systems will not only have to deal with increased expenditures associated with aging societies but also with a significant additional financial burden.[9]

3.6.2 What, When, How, and Why?

An imbalance between calories ingested and calories burned is the primary cause of obesity. When we consume more calories than we burn, our bodies produce surplus energy, which they then store as fat. Although there are several potential causes for this imbalance in energy intake and output, including hereditary abnormalities, living in a setting that promotes excessive consumption of foods high in calories, prolonged inactivity, and low levels of physical activity is the most common one.[7]

One of the main causes of the present obesity epidemic is the expanding availability of ultra-processed meals, which have high quantities of sugars, salt, saturated fats, and refined carbs. Consumption of foods high in fat and free sugars has increased over the past few decades as worldwide diets have altered. Due to the more sedentary character of many occupations, evolving transit options, and escalating urbanization, there has also been an increase in physical inactivity. A lack of supportive policies in areas including health, agriculture, transportation, urban planning, environment, food processing, distribution, marketing, and education contributes to these shifts in dietary and physical activity patterns in many societies.[7]

There are many techniques to measure and identify it, but the Body Mass Index (BMI), which uses a person's height and weight to determine whether their weight is healthy, is the most popular one. A person's BMI is calculated by dividing their weight in kilograms by the square of their height in meters. A BMI of over 25 in adults is regarded as overweight, and one of over 30 as obese.[7]

This indicates that there are two possible causes for the rise in obesity rates in recent decades: either an increase in calorie intake, i.e., eating more; or a decrease in energy expenditure due to decreased levels of daily activity. Both factors are probably contributing to the growth in obesity. Interventions that address both energy intake and expenditure will probably be required to combat obesity.[6]

To prevent overweight and obesity, supportive environments and communities are crucial in influencing people's decisions. They do this by deciding to eat healthier foods and engage in regular physical activity the simplest one (the choice that is most accessible, available, and affordable).

On a personal level, people can Reduce the amount of energy they consume from total fats and sugars; increase their consumption of fruit and vegetables; add legumes, whole grains, and nuts; and get regular exercise.

At the societal level, it is crucial to encourage people to heed the advice by persistently implementing population-based, evidence-based policies that make regular physical activity and healthier eating options readily available, affordable, and easily accessible to everyone, especially the poorest people. One illustration of such a measure is a levy on beverages with added sugar. The food industry can significantly contribute to the promotion of healthy diets by lowering the fat, sugar, and salt content of processed foods, ensuring that all consumers have access to affordable, healthy options, restricting the marketing of foods high in sugar, salt, and fat, especially those marketed to children and teenagers, and promoting regular consumption of healthy foods.[6]

3.6.3 Data

Five datasets are being used in this project. The five datasets are obesity, obesity_data, world, obesity_country, and world_population.

For the World dataset, the lists are the result of merging data from two sources, the Wikipedia ISO 3166-1 article for alpha and numeric country codes, and the UN Statistics site for countries' regional, and sub-regional codes. In addition to countries, it includes dependent territories. The International Organization for Standardization (ISO) site provides partial data (capitalized and sometimes stripped of non-Latin ornamentation) but sells the complete data set as a Microsoft Access 2003 database. Other sites give you the numeric and character codes, but there appeared to be no sites that included the associated UN-maintained regional codes in their data sets. The data was scraped from the above two websites which are all publicly available already to produce some ready-to-use complete data sets.

The World dataset consists of

- name - Country name in English
- alpha-2 - ISO code formed of 2 letters.
- alpha-2 - ISO code formed of 3 letters (use this in your plotly maps ;))
- country code - unique
- region - the continent of provenience
- sub-region - subcontinent
- intermediate region
- codes for region/ subregion/ intermediate region
-

The World dataset was taken from lukes on GitHub: https://github.com/lukes/ISO-3166-Countries-with-Regional-Codes/blob/master/all/all.csv.

Obesity and Obesity_data recorded the Obesity among adults dataset by WHO, from the year 1975 to the year 2016. The dataset is obtained from https://apps.who.int/gho/data/node.main.A900A?lang=en.

The columns for the obesity dataset:

- Gender (Female, Male)
- Age (numeric)
- Height (numeric)
- Weight (numeric)
- family_history_with_overweight (Yes/No)
- FAVC: Whether the individual consumes high-calorie food frequently (Yes/No)

- FCVC: Frequency of consumption of vegetables by the individual weekly (numeric).
- NCP: Number of main meals consumed by the individual per day(numeric).
- CAEC: Consumption of food between main meals (Yes/No).
- SMOKE: Whether the individual smokes tobacco (Yes/No).
- CH2O: Water consumption of the individual in liters per day (numeric)
- SCC: Whether the individual monitors their calorie intake (Yes/No)
- FAF: Physical activity level of the individual (numeric).
- TUE: Time spent by the individual on sedentary activities per day in hours(numeric).
- CALC: Consumption of alcohol by the individual (Yes/No).
- MTRANS: Mode of transportation used by the individual. (Automobile, Motorbike, Bike, Public_Transportation, Walking)
- NObeyesdad: Type of Obesity
- (Insufficient_Weight, Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_T ype_I, Obesity_Type_II, Obesity_Type_III)

Prevalence of obesity among adults, BMI >= 30 (age-standardized estimate) (%)

The World_Population dataset contains population data of different countries/regions from 1960 to 2018 and the dataset has been downloaded from

https://data.worldbank.org/indicator/SP.POP.TOTL

There is condensed and region-wise data in the population dataset.

Mainly adults both male and female were sampled in the above dataset and the period during which the data was collected was between 1960 to 2018. The dataset is around 4 years old.

In the obesity dataset eating habits, smoking habits, calorie intake, physical activity, sedentary activities, and alcohol consumption were recorded among adults. Based on it the obesity prediction was generated.

In the World dataset and world population dataset the data was collected from 250 countries all over the world and for each country population was recorded for different years. Obesity is a global phenomenon, so it is geographical coverage is the whole world. This is a longitudinal dataset. This dataset has GIS.

The obesity dataset mainly focuses on adults. Obesity happens due to unhealthy eating and lifestyle choices by humans.

3.6.4 USERS

The targeted users are the adult population who are at high risk of obesity due to the unhealthy choices made by them and it is important to make them aware and conscious and help them in making better life choices for better health and long life.

The application is deployed on the web, so anyone with access to the internet can access the obesity prediction model Flab or Fab which will predict obesity.

Firstly, the website app helps to give individual obesity prediction, and secondly, we have around 16 visualizations that show global obesity patterns the visualizations make it easy to analyze the different causes of obesity and the most dominant causes, so the user can understand better and then the user can make required changes in the lifestyle to reduce obesity and improve the health and staying fit and fabulous.

The visualizations are useful to the user using the web application because the user becomes more aware and conscious about the different causes of obesity and the user also becomes serious about obesity while seeing different visualizations showing the harmful effects and reduced life expectancy caused by obesity and with the web app the user can have a personal monitoring system to check the obesity every few months and thus maintain good health.

If all adults in the community start using this web app, the community as a whole will be better aware of the dangers of obesity, and thus community as a whole can implement healthy lifestyle habits and thus can reverse the adverse side effects of obesity and thus resulting in better quality life and a higher life expectancy for the community and reduce the health expenditure budget.

3.7 PROJECT MANAGEMENT

3.7.1 IMPLEMENTATION STATUS REPORT

3.7.1.1 Work completed.

Description:

We have gathered the data to complete the training and testing purpose. We have eliminated the duplicate values. Outliers were found and eliminated as well. Using standardized data, we have trained the model using different nodes. We have selected the random forest classifier with the nodes that produce the maximum accuracy. Predictions have been made after the training using test data. We are also visualizing obesity based on different factors using 16 charts.

Improved the existing model's accuracy to 95% and added more of the interaction to already existing charts.

Responsibility (Task, Person):

a. Neha Goud Baddam: Improving the model metrics and adding more visualizations.
b. Panduga Raja Tejasvi Prasad: Creating Charts and Testing the web application.
c. Yasmeen Haleem: Creating Charts, Documentation, and hosting of the website.
d. Siddarth Kundaram: Documentation and testing of the website.

Contributions (members/percentage):
a. Neha Goud Baddam: 25%
b. Panduga Raja Tejasvi Prasad: 25%
c. Yasmeen Haleem: 25%
d. Siddarth Kundaram: 25%

4. REFERENCES

1. Adult BMI Calculator: This site uses Height and Weight to calculate the BMI.
2. National Heart, Lung, and Blood Institute: Calculate your Body Mass Index: This site takes in Height and weight and visualizes the BMI chart for easy understanding.
3. Obesity prediction using Machine Learning: This summary of machine learning algorithms provides a unique overview of the state of data analysis applied specifically to obesity.
4. An Analysis of Indonesian Basic Health Research 2018: This article suggests multiple machine-learning techniques that predict obesity levels.
5. We are using an existing dataset for this project: UCI Obesity Level Dataset
6. https://ourworldindata.org/obesity
7. https://world-heart-federation.org/what-we-do/obesity/
8. https://www.washingtonpost.com/wellness/2023/03/20/obesity-overweight-increasing-worldwide/
9. https://www.allianz-trade.com/en_global/news-insights/economic-insights/obesity-a-costly-epidemic.html
10. https://careerfoundry.com/en/blog/web-development/what-is-flask/#:~:text=Flask%20is%20a%20microframework%20for,web%20apps%20quickly%20and%20simply.
11. https://www.analyticsvidhya.com/blog/2021/06/tricks-for-data-visualization-plotly-library/