OBESITY PREDICTION

using random forest classifier and visualization

Irgi Rifki M.
Unaki Semarang, Central Java

ibimbing

TABLE OF CONTENT:

ABSTRACT

Obesity prediction is important in the prevention and treatment of lifestyle-related diseases. This study aims to build an obesity prediction model using the Random Forest Classifier algorithm. The data sets used include FAF (Physical Activity Frequency), TUE (Time Using Technology), FCVC (Food Consumption Frequency in Veggies), NCP (Number of Meals Per Day), FAVC (Frequent Consumption of High Caloric Food), CAEC (Caloric Consumption Between Meals), CH2O (Water Consumption) CALC (Alcohol Consumption), MTRANS (Mode of Transportation). This study provides an effective approach to support early detection of obesity.

ibimbing

TOOLS USED:

kaggle

Google colab

GitHub

Pandas

LOAD DATA SET

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

```
[ ] import pandas as pd

    df = pd.read_csv("/content/Obesity prediction.csv")
    df.head()
```

Result:

	Gender	Age	Height	Weight	family_history	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC
0	Female	21.0	1.62	64.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	0.0	1.0	no
1	Female	21.0	1.52	56.0	yes	no	3.0	3.0	Sometimes	yes	3.0	yes	3.0	0.0	Sometimes
2	Male	23.0	1.80	77.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	2.0	1.0	Frequently
3	Male	27.0	1.80	87.0	no	no	3.0	3.0	Sometimes	no	2.0	no	2.0	0.0	Frequently
4	Male	22.0	1.78	89.8	no	no	2.0	1.0	Sometimes	no	2.0	no	0.0	0.0	Sometimes

Referensi: kaggle.com

The code created shows how the “Obesity prediction.csv” dataset is accessed and displayed for initial analysis. With the help of libraries such as `numpy`, `pandas`, `matplotlib`, and `seaborn`, the data can be manipulated and visualized. This step is important to understand the structure and content of the dataset as a basis for further analysis, such as obesity prediction using machine learning algorithms.

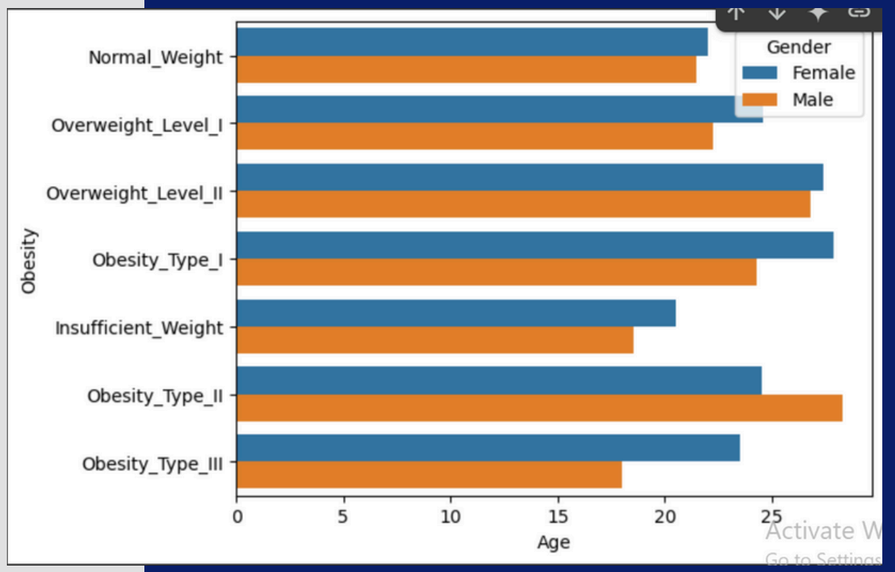
DATA VISUALIZATION (1)

This code:

1. sns.barplot(...): Visualizes the relationship between age, obesity, and gender through a bar graph.
2. df['Obesity'].value_counts(): Counts the number of each obesity category.

```
sns.barplot(data=df,x='Age',y='Obesity',hue='Gender',errorbar=None)
df['Obesity'].value_counts()
```

Result:



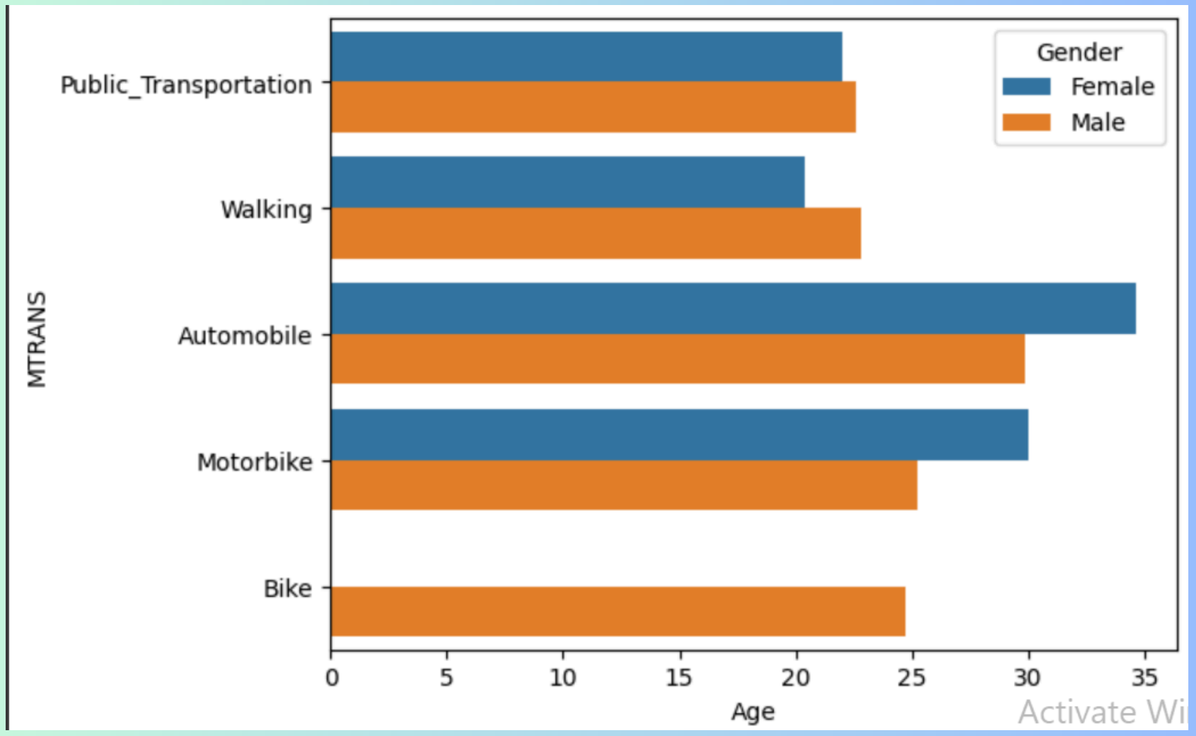
Obesity	count
Obesity_Type_I	351
Obesity_Type_III	324
Obesity_Type_II	297
Overweight_Level_I	290
Overweight_Level_II	290
Normal_Weight	287
Insufficient_Weight	272

From the data it can be concluded that men are more at risk of obesity than women, especially in old age. Obesity increases with age, especially in men, and the proportion of obesity is high, especially in old men while women are more with normal weight.

+ Kode + Teks

```
[ ]  sns.barplot(data=df,x='Age',y='MTRANS',hue='Gender',errorbar=None)
```

Result:



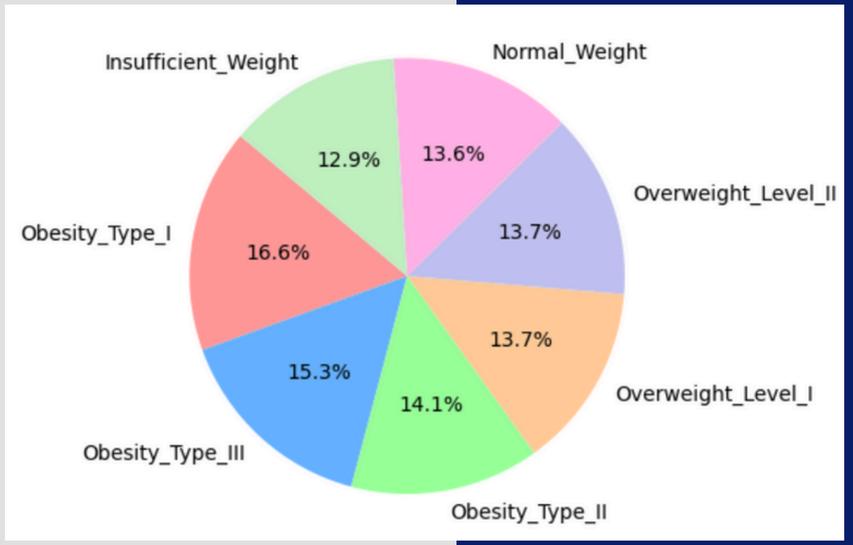
DATA VISUALIZATION (2)

- This code creates a bar plot to visualize the relationship between age (Age), mode of transportation (MTRANS), and gender (Gender) in a dataset using Seaborn.
- This graph shows the average or distribution of mode of transportation (MTRANS) used in a given age group, while comparing patterns between males and females.
- Males are more likely to use cars and motorbikes, while females are more likely to choose public transportation over walking or cycling.

DATA VISUALIZATION (3)

- This code creates a pie chart to show the distribution of obesity categories in a dataset. The result is a pie chart that visually displays the distribution of obesity categories with different colors and percentages
- The graph shows the percentage of population by weight category. Obesity types I and II dominate, followed by the overweight category. The proportion of underweight and normal weight is relatively less.

Result:



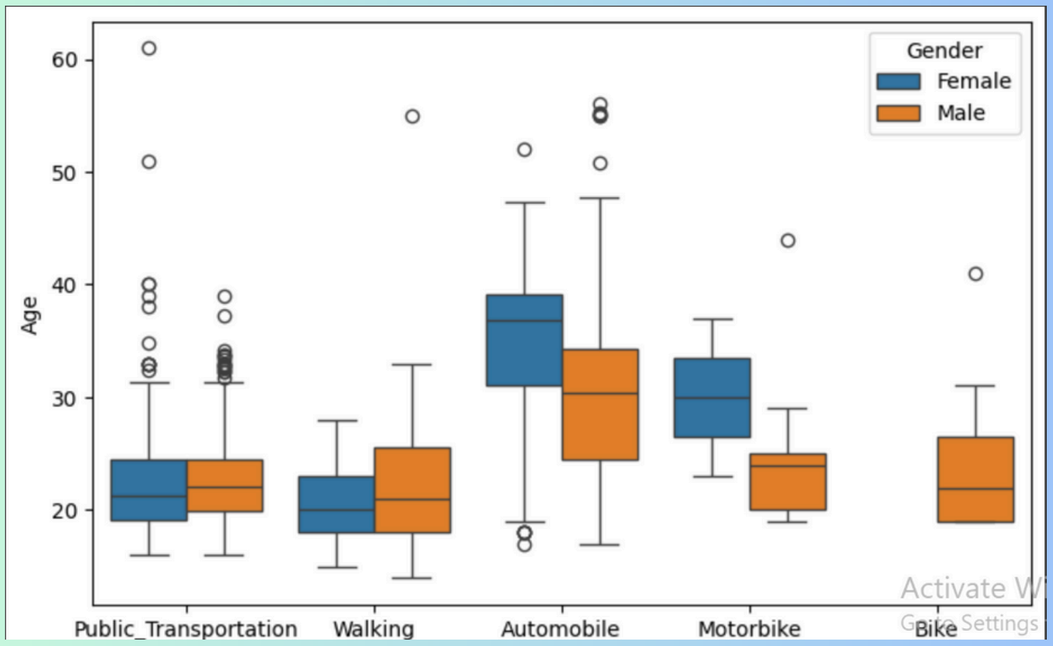

```
[ ]  Obesity_count = df['Obesity'].value_counts()
     plt.pie(Obesity_count,
             labels=Obesity_count.index,
             autopct='%1.1f%%',
             startangle=140,
             colors=['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0','#ffb3e6','#c2f0c2']
            )
```

DATA VISUALIZATION (4)

- Analyze the relationship between age, type of transportation, and gender through box plot visualization.
- Calculate the distribution of the number of users based on the type of transportation to gain quantitative insights.
- The conclusion of the image is that men predominantly use motorbikes and bicycles, women more often use cars while public transportation and walking are used evenly.s.

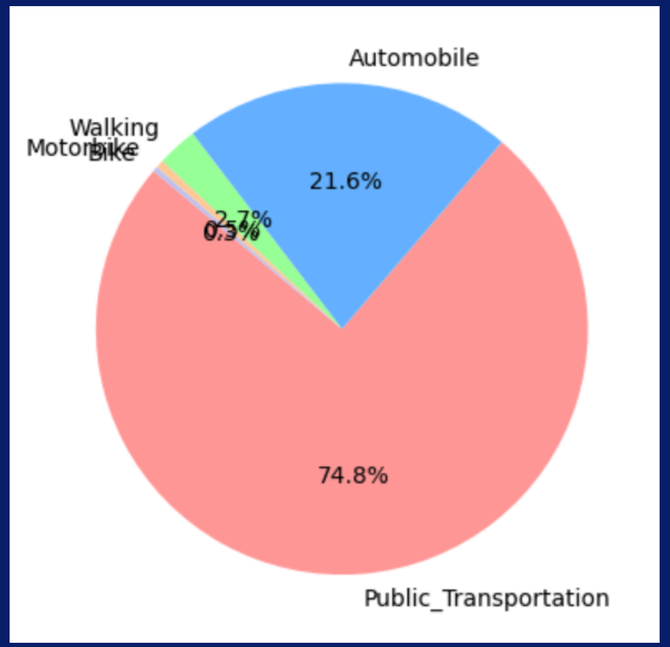
Result:

```
[ ]  plt.figure(figsize=(8,5))
     sns.boxplot(data=df,x='MTRANS',y='Age',hue='Gender')
     transport_count = df['MTRANS'].value_counts()
```



DATA VISUALIZATION (5)

```
[ ]  # plt.figure(figsize=(15,9))
     plt.pie(transport_count,
             labels=transport_count.index,
             autopct='%1.1f%%',
             startangle=140,
             # rotatelabels=45,
             colors=['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0','#ffb3e6','#c2f0c2']
            )
```



- -this code creates a pie chart to visualize the distribution of users based on transportation type. Each section shows the proportion of users for each category with different colors and percentage information.
- from the data, the conclusion is that the majority of people use public transportation as their primary means of transportation, while cars are a popular second choice. Alternative transportation such as walking, motorbikes, and bicycles are very rarely used. This may reflect people's preference for the efficiency and convenience of public transportation compared to other options.

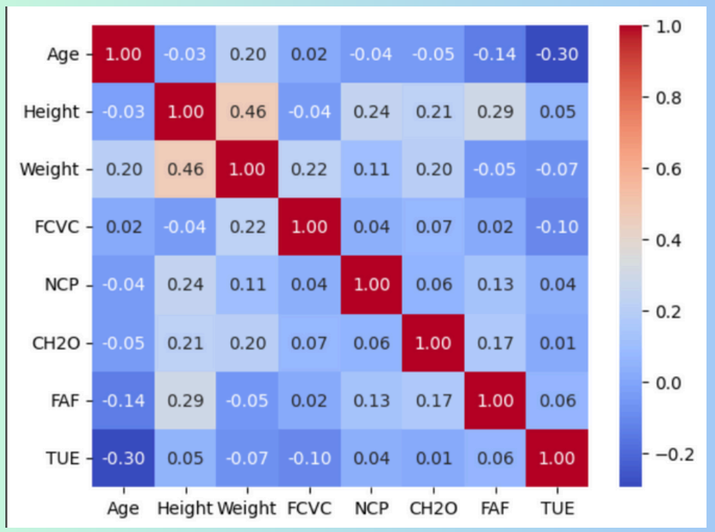
CORELATION MATRIX

A. This code is used to Select numeric columns from the dataset, Calculate the relationship between numeric columns with the correlation matrix and Visualize the correlation matrix in the form of a heatmap to understand the relationship pattern between variables more easily and intuitively.

B. Height and Weight have a strong positive relationship, while most other variables have a weak or insignificant relationship. Age tends to be negatively related to the time of use of electronic devices (TUE).

```
[ ]  numeric_df = df.select_dtypes(include=[np.number])
     corr = numeric_df.corr()

     sns.heatmap(corr,
                 annot=True,
                 cmap='coolwarm',
                 fmt='.2f'
                )
```



MACHINE LEARNING ALGORITM USING RANDOM FOREST CLASSIFIER (HASIL)

```
df['BMI'] = df['Weight'] / df['Height']
# df.head()
```

```
x = df[['Age','FAF','BMI']]
y = df['Obesity']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train,y_train)
```

```
from sklearn.metrics import accuracy_score,classification_report
y_pred = model.predict(x_test)
print("Accuracy: ",accuracy_score(y_test,y_pred)*100)
print("\nClassification_report:\n",classification_report(y_test,y_pred))
```

```
data = pd.DataFrame({'Age':[21] ,'FAF':[0.000000] ,'BMI':[39.506173]})
data_pred = model.predict(data)
print("The Obesity types is ",data_pred[0])
```

```
The Obesity types is  Normal_Weight
```

```
df['CALC'].value_counts()
```

```
Accuracy:  90.30732860520094

Classification_report:
                      precision    recall  f1-score   support

Insufficient_Weight       0.93      0.96      0.95        56
      Normal_Weight       0.87      0.87      0.87        62
     Obesity_Type_I       0.90      0.95      0.93        78
    Obesity_Type_II       0.95      0.97      0.96        58
   Obesity_Type_III       0.98      1.00      0.99        63
 Overweight_Level_I       0.82      0.80      0.81        56
Overweight_Level_II       0.84      0.72      0.77        50

           accuracy                           0.90       423
          macro avg       0.90      0.90      0.90       423
       weighted avg       0.90      0.90      0.90       423
```

CALC	count
Sometimes	1401
no	639
Frequently	70
Always	1

dtype: int64

MACHINE LEARNING ALGORITM USING RANDOM FOREST CLASSIFIER (PEMBAHASAN)

- Main objective Create a machine learning model (Random Forest Classifier) to predict obesity levels based on age (Age), frequency of physical activity (FAF), and BMI (BMI). -The main steps include, Data preprocessing is done by adding the BMI feature. Data is divided into features (x) and targets (y), then separated into training and testing data. The model is trained using the Random Forest algorithm. Evaluation is done by measuring the accuracy of the model and producing a classification report. The model is used to predict obesity levels from new data. so that the model is able to predict a person's obesity level based on input data (age, physical activity, BMI) with a certain accuracy.
- The model successfully classifies obesity types with high accuracy and good performance across most classes, making it effective for obesity prediction tasks.

THANK YOU

Irgi Rifki M.

Unaki Semarang, Central Java

CONTACT ME:

- Phone : 6283878861359
- IG : Irgirifki_27
- LinkedIn : Irgi Rifki
- Email : irgirifki5@gmail.com