Exploring Sleep Health and Lifestyle Factors:

Predicting BMI Category Based on Age, Gender, and Occupation

Lim Jacob

Abstract

The Sleep Health and Lifestyle Dataset covers a wide range of variables related to sleep and daily habits. However, the data will be cleaned and filtered, focusing on Gender, Age, Occupation and BMI Category only.

I began by exploring the relationships between 1. Gender and BMI category, 2. Age and BMI category, 3.Occupation and BMI category. The main methods applied here are **Grouped Bar Charts, Distribution Comparison using Boxplot and Count Plot.**

Given that there are correlations between Gender, Age, Occupation and BMI Category, I then approached the central question of whether we can predict the BMI category based on age, gender, and occupation using **Decision Tree Classification(Scikit).**

With a high level of accuracy, the results suggest that we can predict the BMI category based on Age, Gender, and Occupation.

Motivation

Recently, I came across an research article titled "Changes in body mass index by age, gender, and socio-economic status among a cohort of Norwegian men and women (1990–2001)", which highlights the dynamic nature of BMI and how they vary across different age groups, genders, and socio-economic backgrounds. The article sparked the inspiration for this presentation.

The motivation behind this presentation stems from the intriguing correlations observed between gender, age, occupation, and BMI category within the context of sleep, health, and lifestyle factors. As these aspects play crucial roles in individuals' well-being, understanding the relationships between them can provide valuable insights into overall health and potentially guide personalized interventions or preventive measures.

Understanding the correlations between gender, age, occupation, and BMI category can aid in the development of targeted strategies for maintaining healthy weight ranges, providing tailored recommendations for different demographic groups, and fostering an evidence-based approach towards overall well-being.

Dataset

Sleep Health and Lifestyle Dataset from Kaggle

Dataset Overview:
The Sleep Health and Lifestyle Dataset comprises 400 rows and 13 columns, covering a wide range of variables related to sleep and daily habits. It includes details such as gender, age, occupation, sleep duration, quality of sleep, physical activity level, stress levels, BMI category, blood pressure, heart rate, daily steps, and the presence or absence of sleep disorders.

Key Features of the Dataset:
Comprehensive Sleep Metrics: Explore sleep duration, quality, and factors influencing sleep patterns.
Lifestyle Factors: Analyze physical activity levels, stress levels, and BMI categories.
Cardiovascular Health: Examine blood pressure and heart rate measurements.
Sleep Disorder Analysis: Identify the occurrence of sleep disorders such as Insomnia and Sleep Apnea.

Data Preparation and Cleaning

Dropping unwanted columns: Unnecessary columns that did not contribute to the analysis were removed from the dataset. This was done to simplify the data and focus only on the relevant variables.

Handling missing values: The dataset was assessed for missing values, and appropriate actions were taken to address them.

Research Questions

Is there a relationship between gender, age, occupation, and BMI category?

Can we predict the BMI category based on age, gender, and occupation using Decision Tree Classification?

Methods

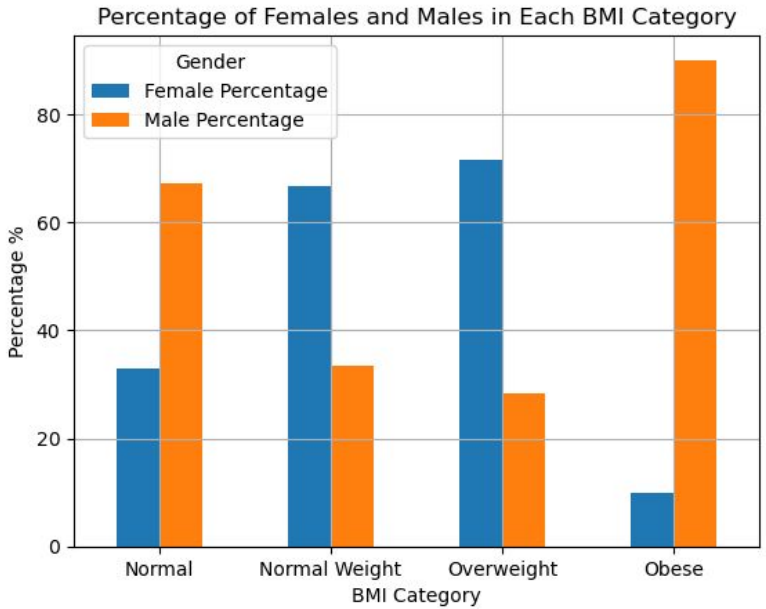
The **grouped bar chart** is a suitable for visualizing the distribution and comparison of categorical variables, in this case, gender and BMI category.

The **boxplot** displays the distribution of a quantitative variable, age across different BMI categories in a concise and informative manner. It provides several key statistics, such as the median, quartiles, and any outliers, which allow for a comprehensive understanding of the data distribution.

A **count plot** is suitable for analyzing the relationship between occupation and BMI category because it visually compares the frequency or count of different occupation categories within each BMI category.

Decision Trees can handle a mix of categorical and numerical features, making them suitable for incorporating age (numerical), gender (categorical), and occupation (categorical) as predictors.

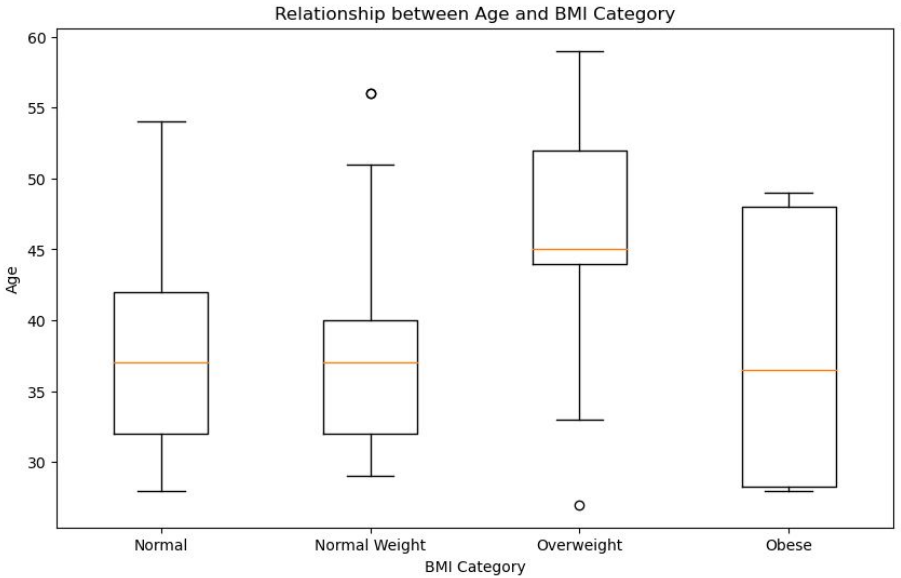
Findings



Given that the total number of males and the total number of females in the data is relatively even with a difference of 4, there are significantly more 'Normal Weight' and 'Overweight' females than males. Similarly, there are significantly more 'Normal' and 'Obese' males compared to females, especially for the category - ‘Obese’.

This result suggests that there may be gender-specific differences in BMI distribution within the dataset.

Findings

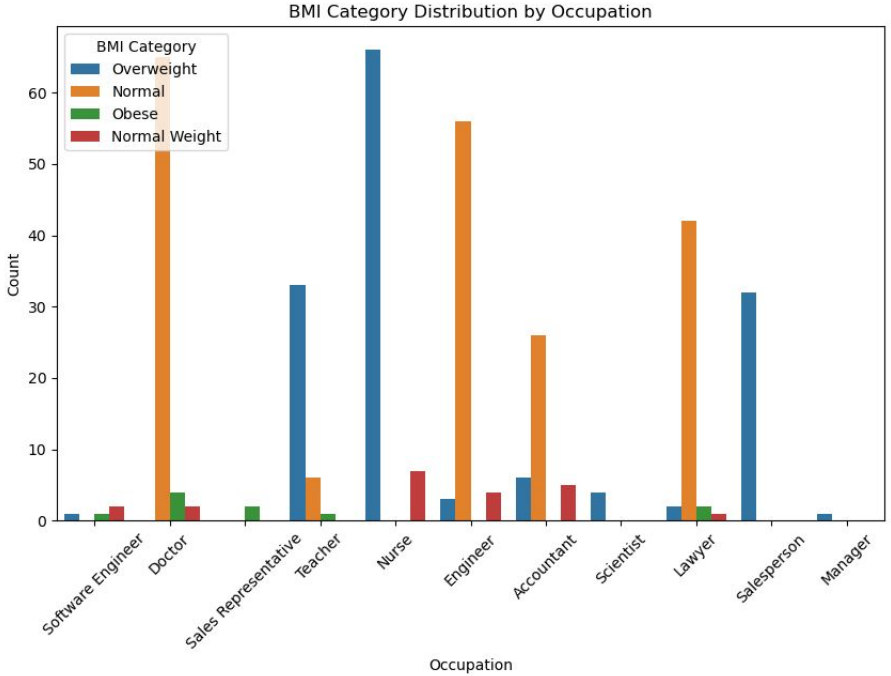


BMI Category	count	mean	std	min	25%	50%	75%	max
Normal	195.0	38.482051	7.561666	28.0	32.00	37.0	42.0	54.0
Normal Weight	21.0	38.380952	7.921339	29.0	32.00	37.0	40.0	56.0
Obese	10.0	38.000000	9.614803	28.0	28.25	36.5	48.0	49.0
Overweight	148.0	47.885135	6.859646	27.0	44.00	45.0	52.0	59.0

From the boxplots, the age ranges for 'Normal', 'Normal Weight' and 'Obese' seem to lie between 28 and 56, with a median of around 37. However, for 'Overweight', the age range seems to lie between 35 and 59, with a median of 45. Looking at the description of each boxplot, we can see that the mean for 'Normal', 'Normal Weight' and 'Obese' are similar at 38 as well, whereas 'Overweight' is much higher at 47. This suggests that being older has a correlation with being 'Overweight'.

However 1 limitation which is apparent here is that the age range in this dataset is not large enough, with missing data for people below the age of 27 and over the age of 59.

Findings



The distribution of BMI categories within each occupation indicates potential associations between occupation and weight status. Some notable examples are Doctor, Engineer, Accountant and Lawyer where a large portion of them falls under 'Normal' whilst Teacher, Nurse, Scientist, Salesperson and Manager are mostly 'Overweight'. This suggests the potential correlation between Occupation and BMI Category.

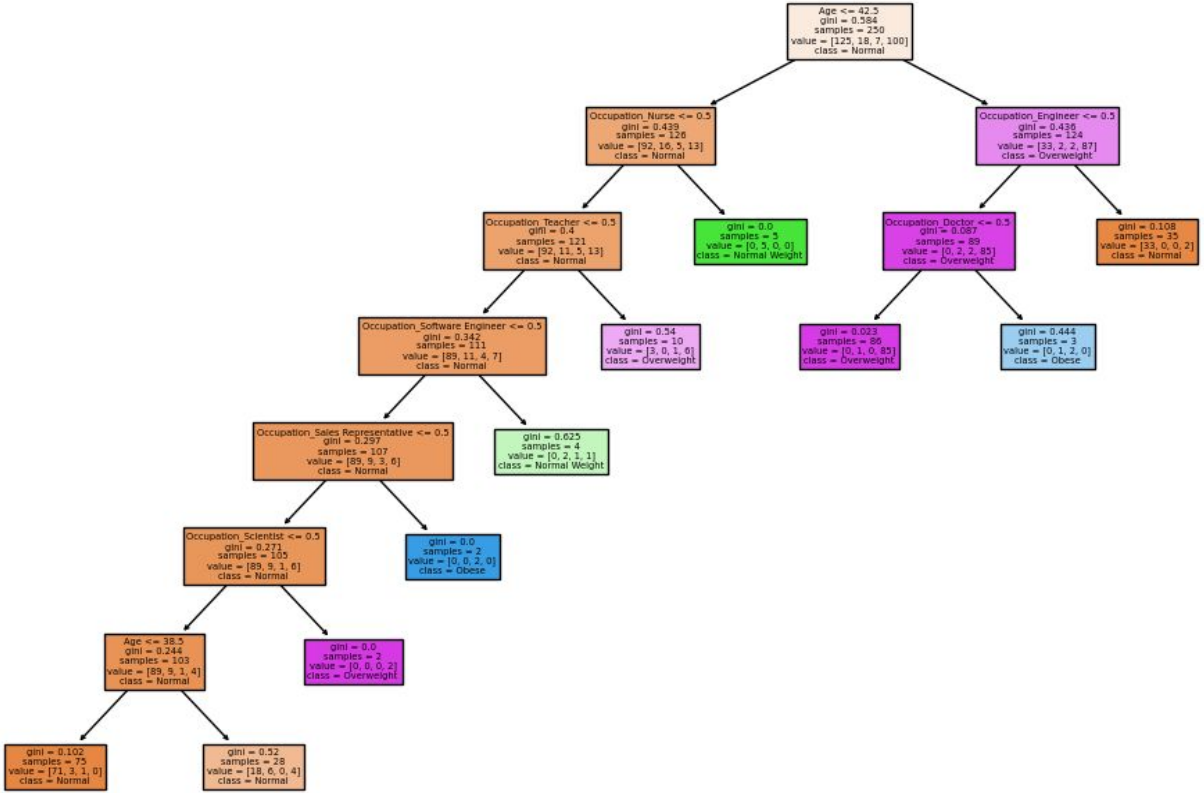
Findings

	precision	recall	f1-score	support
Normal	0.97	0.96	0.96	70
Normal Weight	0.50	0.33	0.40	3
Obese	0.67	0.67	0.67	3
Overweight	0.94	0.98	0.96	48
accuracy			0.94	124
macro avg	0.77	0.73	0.75	124
weighted avg	0.94	0.94	0.94	124

Can we predict the BMI category based on age, gender, and occupation using Decision Tree Classification?

Findings

Visualizing the Decision Tree



Accuracy: The overall accuracy of the model is 94.35%, indicating that it correctly predicted the BMI category for 94.35% of the instances in the testing set.

Precision: Precision measures the proportion of correctly predicted instances out of the total predicted instances for each class.
The precision values for the 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 0.97, 0.50, 0.67, and 0.94, respectively. It means that the model has high precision for predicting 'Normal' and 'Overweight' categories, but relatively lower precision for predicting 'Normal Weight' and 'Obese' categories.

Recall: Recall measures the proportion of correctly predicted instances out of the total actual instances for each class.
The recall values for the 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 0.96, 0.33, 0.67, and 0.98, respectively. The model achieved high recall for 'Normal' and 'Overweight' categories, but relatively lower recall for 'Normal Weight' and 'Obese' categories.

F1-score: F1-score combines precision and recall into a single metric, providing a balanced measure of the model's performance.
The F1-scores for the 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 0.96, 0.40, 0.67, and 0.96, respectively.

Support: Support indicates the number of instances in the testing set for each category.
The support values for 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 70, 3, 3, and 48, respectively.

The model achieved good accuracy and performed well for predicting the 'Normal' and 'Overweight' categories. However, it struggled with the 'Normal Weight' and 'Obese' categories, possibly due to the smaller number of instances in those categories(as shown under Support). **Overall, the results suggest that we can predict the BMI category based on Age, Gender, and Occupation with a high level of accuracy.**

Limitations

The age range in this dataset is not large enough, with missing data for people below the age of 27 and over the age of 59, which is data useful for the analysis. Additionally, our machine learning model struggled with the 'Normal Weight' and 'Obese' categories, possibly due to the smaller number of instances in those categories, reinforcing the need for a larger and more balanced dataset to improve model performance.

Sample Bias: The dataset may not be representative of the entire population or may have a biased sample selection. This can lead to inaccurate or skewed conclusions if the sample is not diverse or is not selected randomly. Even though there are more than 32 salespersons, all of them appeared to be overweight in this dataset. Additionally, certain occupations have significantly higher representation compared to others.

Conclusions

Based on the analysis of the dataset, it can be concluded that there is a relationship between age, occupation, and BMI category. The findings indicate that these factors are associated with the distribution of individuals across different BMI categories.

Based on the results from scikit's classification_report, it can be concluded that we are able to predict the BMI category based on age and occupation using Decision Tree Classification.

Acknowledgements

UCSanDiegoX DSE200x: Python for Data Science!

References

Sleep Health and Lifestyle Dataset
https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset

Reas, D.L., Nygård, J.F., Svensson, E. et al. Changes in body mass index by age, gender, and socio-economic status among a cohort of Norwegian men and women (1990–2001). BMC Public Health 7, 269 (2007). https://doi.org/10.1186/1471-2458-7-269

In [128…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn import tree
```

In [41]:
```
data = pd.read_csv('./Sleep_health_and_lifestyle_dataset.csv')
```

In [42]:
```
data
```

Out[42]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	None
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
...	...	...	...	...	...	...	...	...	...	...	...	...	...
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea

374 rows × 13 columns

Data Cleaning

Drop unwanted columns

In [43]:
```
columns_to_delete = ['Sleep Duration', 'Quality of Sleep', 'Physical Activity Level', 'Stress Level', 'Blood Pr
data.drop(columns=columns_to_delete, inplace=True)
```

In [44]:
```
data
```

Out[44]:

	Person ID	Gender	Age	Occupation	BMI Category
0	1	Male	27	Software Engineer	Overweight
1	2	Male	28	Doctor	Normal
2	3	Male	28	Doctor	Normal
3	4	Male	28	Sales Representative	Obese
4	5	Male	28	Sales Representative	Obese
...	...	...	...	...	...
369	370	Female	59	Nurse	Overweight
370	371	Female	59	Nurse	Overweight
371	372	Female	59	Nurse	Overweight
372	373	Female	59	Nurse	Overweight
373	374	Female	59	Nurse	Overweight

374 rows × 5 columns

Drop null values

In [45]:
```
before_rows = data.shape[0]
```

In [46]:
```
data = data.dropna()
```

In [47]:
```
after_rows = data.shape[0]
```

How many rows dropped due to cleaning?

In [49]:
```
before_rows - after_rows
```

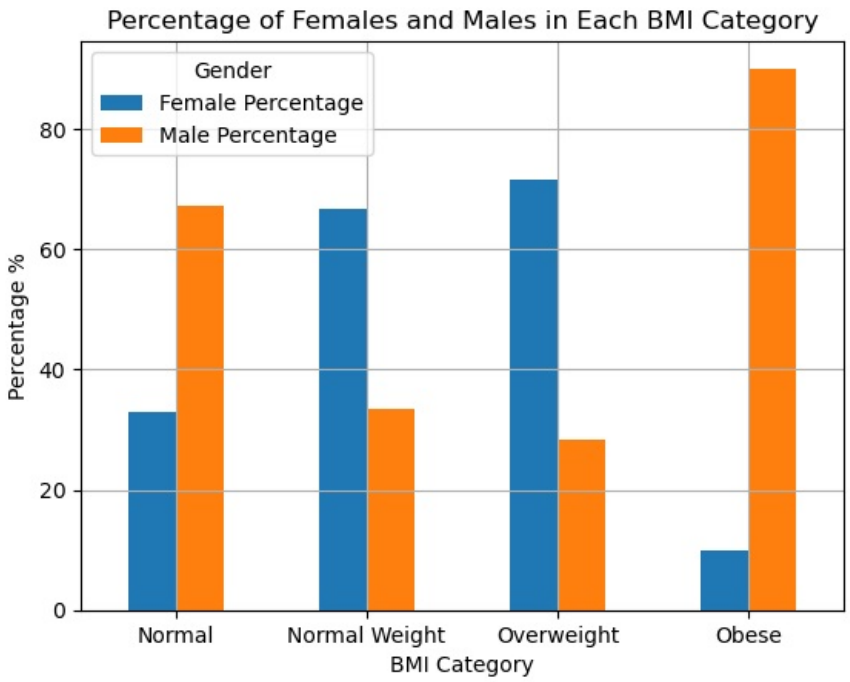
Out[49]:
```
0
```

Is there a relationship between Gender, Age, Occupation, and BMI category?

In [66]:
```
# Count the occurrences of each gender
gender_counts = data['Gender'].value_counts()

# Print the number of males and females
print("Number of Males:", gender_counts['Male'])
print("Number of Females:", gender_counts['Female'])
```

```
Number of Males: 189
Number of Females: 185
```

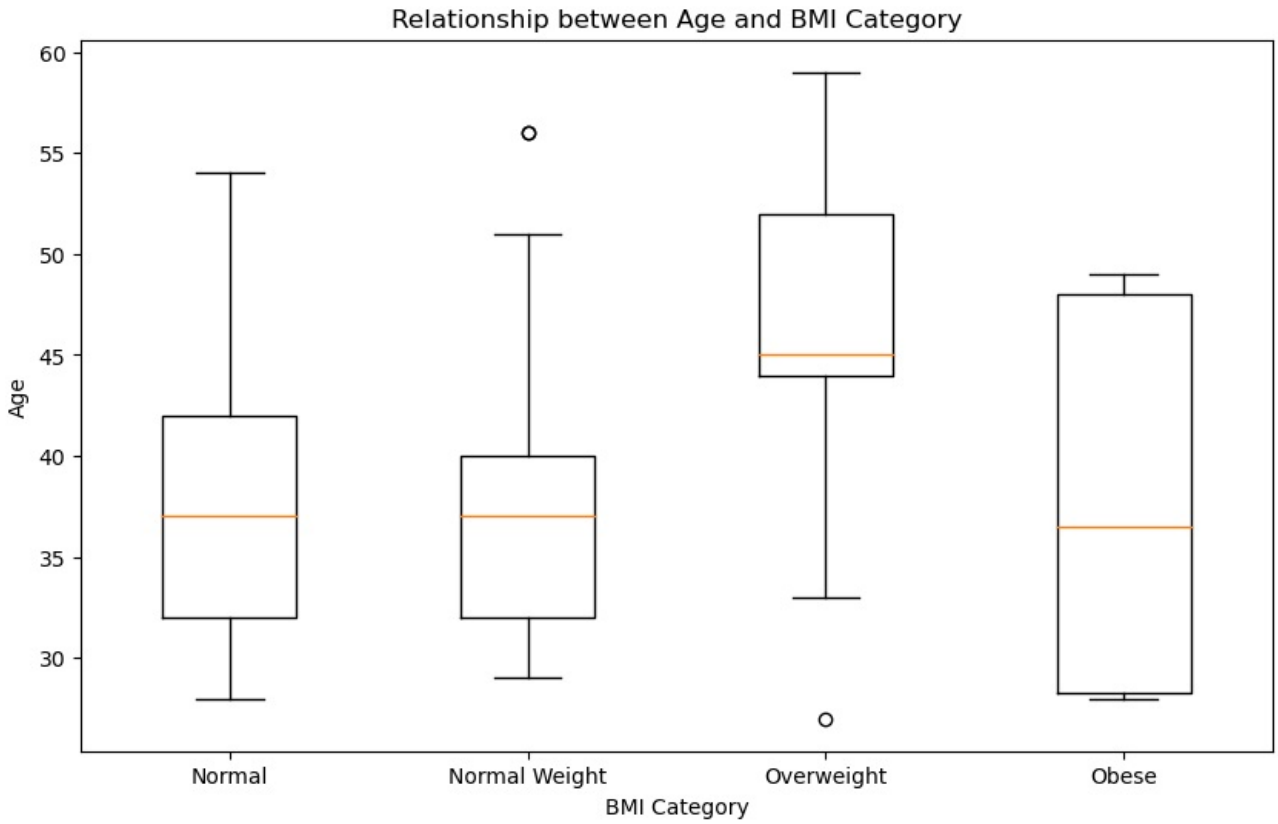
In [125…
```
# Group the data by BMI category and gender, and calculate the percentage
grouped = data.groupby(['BMI Category', 'Gender']).size().unstack()
total_counts = grouped.sum(axis=1)  # Total counts per BMI category

# Calculate the percentage of females and males in each BMI category
grouped['Female Percentage'] = grouped['Female'] / total_counts * 100
grouped['Male Percentage'] = grouped['Male'] / total_counts * 100

# Define the desired order of BMI categories
desired_order = ['Normal', 'Normal Weight', 'Overweight', 'Obese']

# Reindex the DataFrame to match the desired order
grouped = grouped.reindex(desired_order)

# Plot the percentages
grouped[['Female Percentage', 'Male Percentage']].plot(kind='bar')

# Set plot labels and title
plt.xlabel('BMI Category')
plt.ylabel('Percentage %')
plt.title('Percentage of Females and Males in Each BMI Category')
plt.grid(True)
plt.xticks(rotation=0)

# Display the plot
plt.show()
```



Given that the total number of males and the total number of females in the data is relatively even with a difference of 4, there are significantly more 'Normal Weight' and 'Overweight' females than males. Similarly, there are significantly more 'Normal' and 'Obese' males compared to females, with a stark difference for 'Obese'. This result suggests that there may be gender-specific differences in BMI distribution within the dataset.

In [93]:
```
sorted_categories = ['Normal', 'Normal Weight', 'Overweight', 'Obese']
```

```
# Create a box plot to visualize the relationship between 'Age' and 'BMI Category'
plt.figure(figsize=(10, 6))
plt.boxplot(
    [data[data['BMI Category'] == category]['Age'] for category in sorted_categories],
    labels=sorted_categories
)

# Set plot labels and title
plt.xlabel('BMI Category')
plt.ylabel('Age')
plt.title('Relationship between Age and BMI Category')
plt.show()
```



I am using boxplots here for Distribution Comparison: Boxplots allow me to compare the distribution of age across different BMI categories. Each box represents the range between the first and third quartiles, with the line inside the box indicating the median age. This comparison helps identify any variations or similarities in age distribution among different BMI categories.

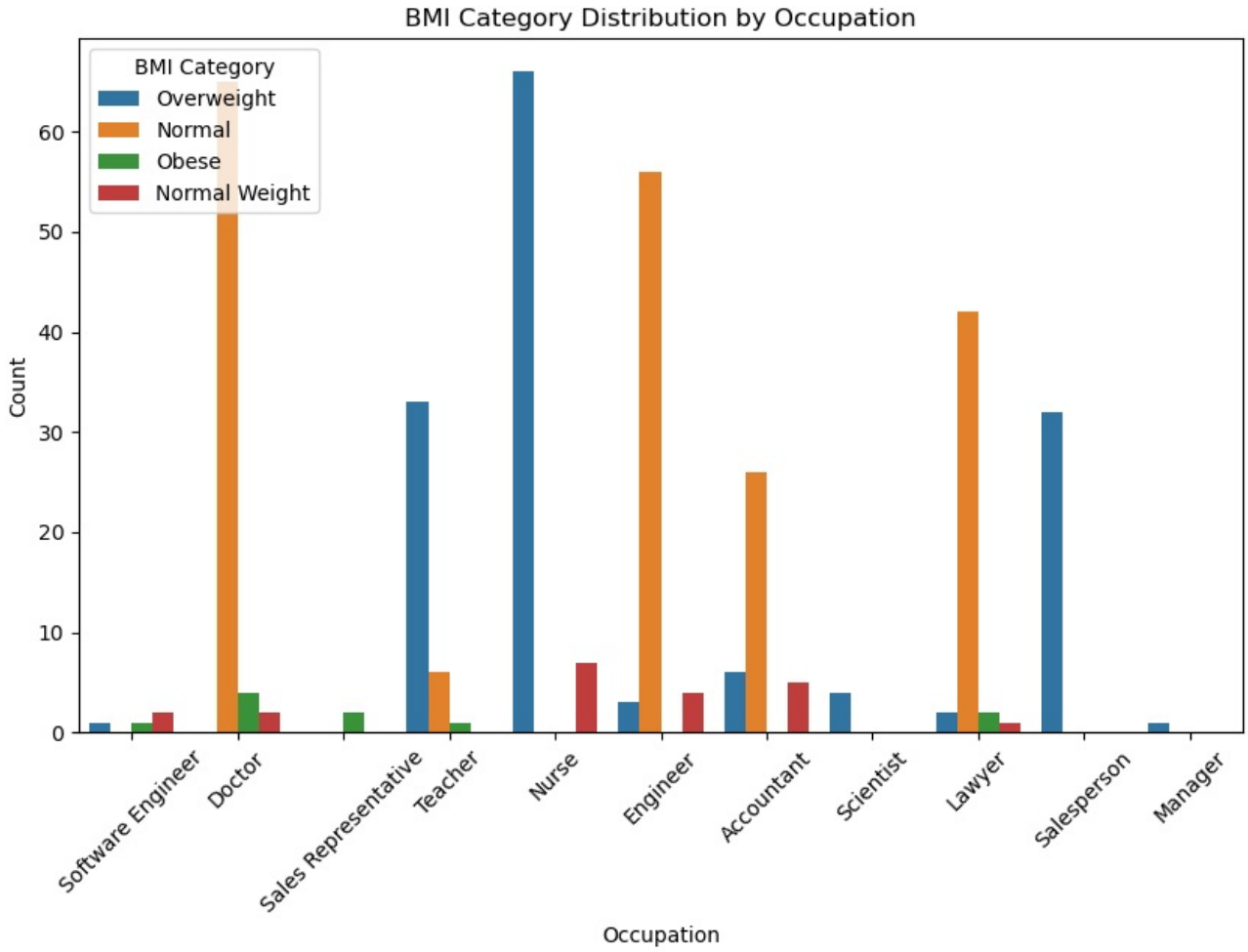
From the boxplots, the age ranges for 'Normal', 'Normal Weight' and 'Obese' seem to lie between 28 and 56, with a median of around 37. However, for 'Overweight', the age range seems to lie between 35 and 59, with a median of 45. Looking at the description of each boxplot, we can see that the mean for 'Normal', 'Normal Weight' and 'Obese' are similar at 38 as well, whereas 'Overweight' is much higher at 47. This suggests that being older has a correlation with being 'Overweight'.

However 1 limitation which is apparent here is that the age range in this dataset is not large enough, with missing data for people below the age of 27 and over the age of 59.

In [90]:
```
summary = data.groupby('BMI Category')['Age'].describe()
print(summary)
```

```
                count       mean       std   min    25%   50%   75%   max
BMI Category
Normal          195.0  38.482051  7.561666  28.0  32.00  37.0  42.0  54.0
Normal Weight    21.0  38.380952  7.921339  29.0  32.00  37.0  40.0  56.0
Obese            10.0  38.000000  9.614803  28.0  28.25  36.5  48.0  49.0
Overweight      148.0  47.885135  6.859646  27.0  44.00  45.0  52.0  59.0
```

In [104…
```
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='Occupation', hue='BMI Category')

# Set plot labels and title
plt.xlabel('Occupation')
plt.ylabel('Count')
plt.title('BMI Category Distribution by Occupation')

plt.xticks(rotation=45)
plt.show()
```



The distribution of BMI categories within each occupation indicates potential associations between occupation and weight status. Some notable examples are Doctor, Engineer, Accountant and Lawyer where a large portion of them falls under 'Normal' whilst Teacher, Nurse, Scientist, Salesperson and Manager are mostly 'Overweight'. This suggests the potential correlation between Occupation and BMI Category.

Can we predict the BMI category based on age, gender, and occupation?

In [119…

```
# Prepare the data
X = data[['Age', 'Gender', 'Occupation']]
y = data['BMI Category']

# Convert categorical variables to one-hot encoding
X_encoded = pd.get_dummies(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.33, random_state=324)

# Initialize and train the decision tree classifier
clf = DecisionTreeClassifier(max_leaf_nodes=10, random_state=0)
clf.fit(X_train, y_train)

# Make predictions on the testing set
y_pred = clf.predict(X_test)

# Evaluate the model
print(classification_report(y_test, y_pred))
accuracy_score(y_true = y_test, y_pred = y_pred)
```

```
              precision    recall  f1-score   support

      Normal       0.97      0.96      0.96        70
Normal Weight       0.50      0.33      0.40         3
       Obese       0.67      0.67      0.67         3
  Overweight       0.94      0.98      0.96        48

    accuracy                           0.94       124
   macro avg       0.77      0.73      0.75       124
weighted avg       0.94      0.94      0.94       124
```

Out[119]:

```
0.9435483870967742
```

Accuracy: The overall accuracy of the model is 94.35%, indicating that it correctly predicted the BMI category for 94% of the instances in the testing set.

Precision: Precision measures the proportion of correctly predicted instances out of the total predicted instances for each class. The precision values for the 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 0.97, 0.50, 0.67, and 0.94, respectively. It means that the model has high precision for predicting 'Normal' and 'Overweight' categories, but relatively lower precision for predicting 'Normal Weight' and 'Obese' categories.

Recall: Recall measures the proportion of correctly predicted instances out of the total actual instances for each class. The recall values for the 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 0.96, 0.33, 0.67, and 0.98, respectively. The model achieved high recall for 'Normal' and 'Overweight' categories, but relatively lower recall for 'Normal Weight' and 'Obese' categories.

F1-score: The F1-score combines precision and recall into a single metric, providing a balanced measure of the model's performance. The F1-scores for the 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 0.96, 0.40, 0.67, and 0.96, respectively.

Support: The support indicates the number of instances in the testing set for each category. The support values for 'Normal', 'Normal Weight', 'Obese', and 'Overweight' categories are 70, 3, 3, and 48, respectively.

Overall, the model achieved good accuracy and performed well for predicting the 'Normal' and 'Overweight' categories. However, it struggled with the 'Normal Weight' and 'Obese' categories, possibly due to the smaller number of instances in those categories(as shown under Support). The precision, recall, and F1-scores for these categories are relatively lower compared to the 'Normal' and 'Overweight' categories. \ **The results suggest that we can predict the BMI category based on Age, Gender, and Occupation with a high level of accuracy.**

In [129…

```
from sklearn import tree

# Visualize the decision tree
plt.figure(figsize=(12, 8))
tree.plot_tree(clf, feature_names=X_encoded.columns, class_names=clf.classes_, filled=True)
plt.show()
```