DATA SCIENCE FINAL PROJECT REPORT

109550186 李嘉玲 *& 109550201* 林家輝

For the final project, the dataset that we used is ***Sleep Health and Lifestyle Dataset*** from Kaggle. The dataset encompasses diverse individual attributes, including demographic details, lifestyle factors such as sleep duration, physical activity, stress levels, and health metrics like BMI, blood pressure, heart rate, and daily steps. Additionally, it provides insights into sleep quality and disorders, categorizing individuals based on their sleep patterns—whether they exhibit no sleep disorder, struggle with insomnia, or face sleep apnea-related breathing interruptions during sleep.

	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	NaN
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN

Data Statistics

After loading the dataset, the next step would be getting to know the data and its statistics of each feature. The result that we get shows that there are several columns in object or categorical type. The statistics also show the quartile, mean and median of the numerical features.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Person ID                374 non-null    int64  
 1   Gender                   374 non-null    object 
 2   Age                      374 non-null    int64  
 3   Occupation               374 non-null    object 
 4   Sleep Duration           374 non-null    float64
 5   Quality of Sleep         374 non-null    int64  
 6   Physical Activity Level  374 non-null    int64  
 7   Stress Level             374 non-null    int64  
 8   BMI Category             374 non-null    object 
 9   Blood Pressure           374 non-null    object 
 10  Heart Rate               374 non-null    int64  
 11  Daily Steps              374 non-null    int64  
 12  Sleep Disorder           155 non-null    object 
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000
mean	187.500000	42.184492	7.132086	7.312834	59.171123	5.385027	70.165775	6816.844920
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676	1617.915679
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000	3000.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000	5600.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000	7000.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000	8000.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000	10000.000000

Handling Missing Data

After grasping insights from the dataset statistics, the subsequent step involves identifying duplicated, unique, and missing values. The 'Sleep Disorder' feature is the sole attribute with missing data, totaling 219 instances. Subsequently, following a dataset examination, all 'NaN' values are replaced uniformly with 'None.'

Data Preprocessing

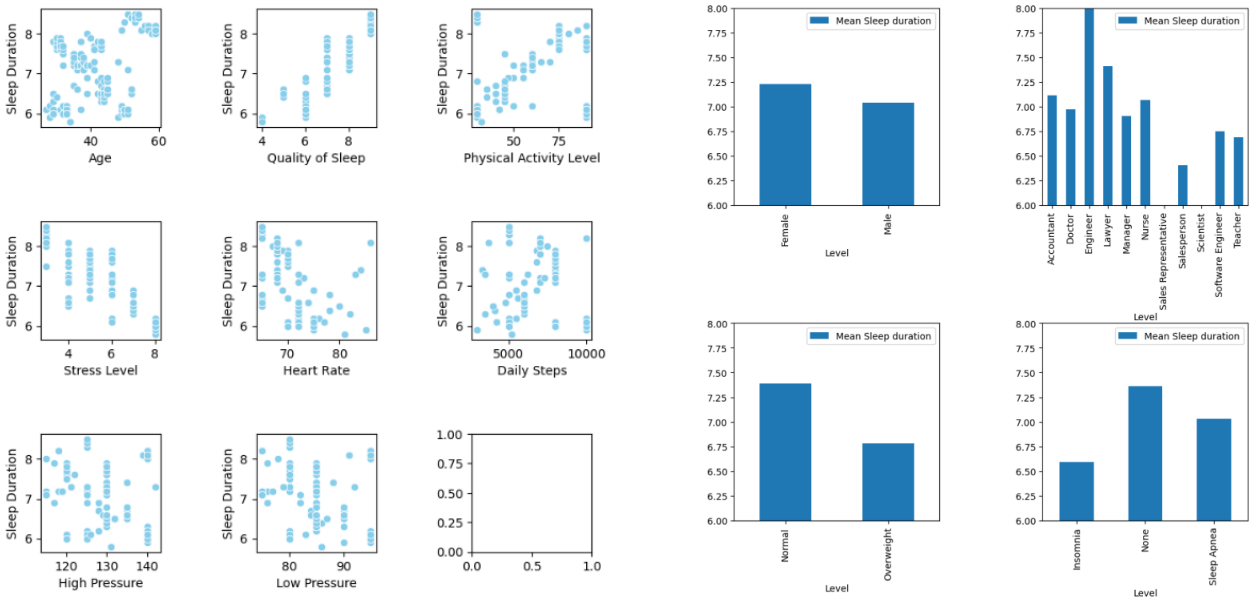
As the ‘Blood Pressure’ feature is in a string data type, it is hard to be processed for the next tasks. Therefore, it is separated into different features of ‘High Pressure’ and ‘Low Pressure’ in integer type. In addition, the unique features in the BMI Category shows ['Overweight' 'Normal' 'Obese' 'Normal Weight']. Since ‘Normal Weight’ equals ‘Normal’ and ‘Obese’ has the same meaning as ‘Overweight’, they are then combined resulting in two unique values.

```
print('Unique Values of BMI Category are', df['BMI Category'].unique())
print('Unique Values of Sleep Disorder are', df['Sleep Disorder'].unique())
```

```
Unique Values of BMI Category are ['Overweight' 'Normal']
Unique Values of Sleep Disorder are ['None' 'Sleep Apnea' 'Insomnia']
```

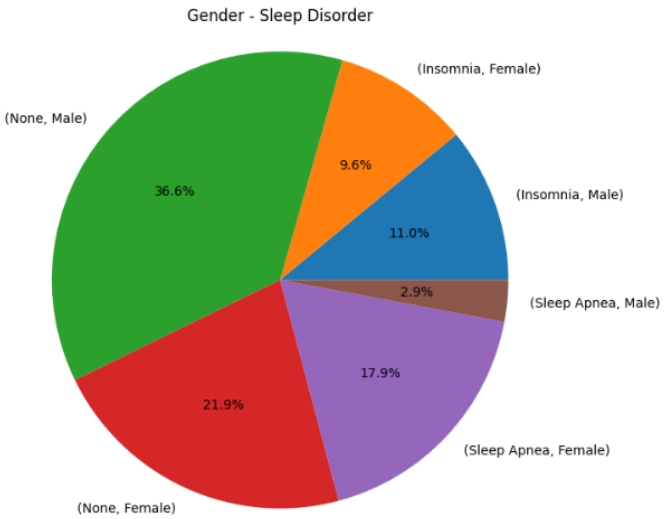
Data Visualization and Analysis

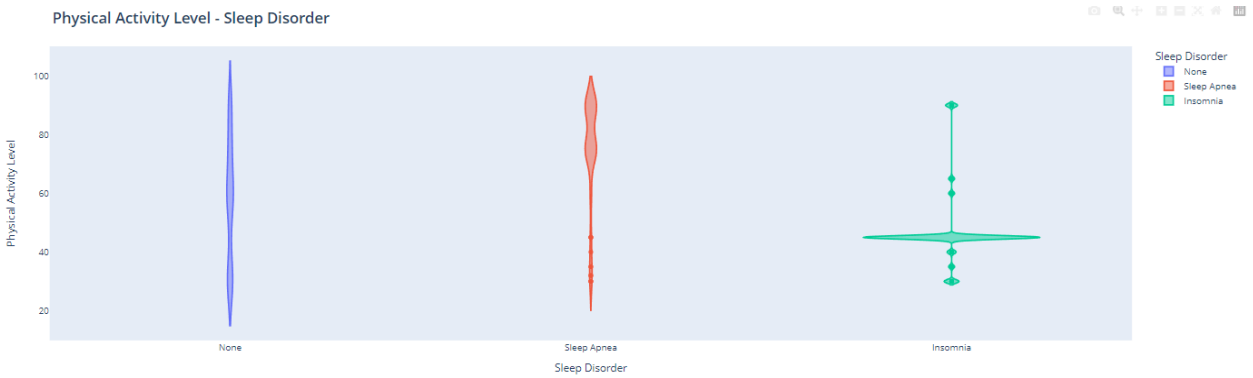
The libraries that are mainly used through this project are Seaborn, Matplotlib, and Plotly. In the beginning, the data are divided and visualized by their data types. For the categorical features, visualizations are made to compare the average sleep duration between these features. While for numerical features, they are also compared with the sleep duration and shown in a scatter plot.

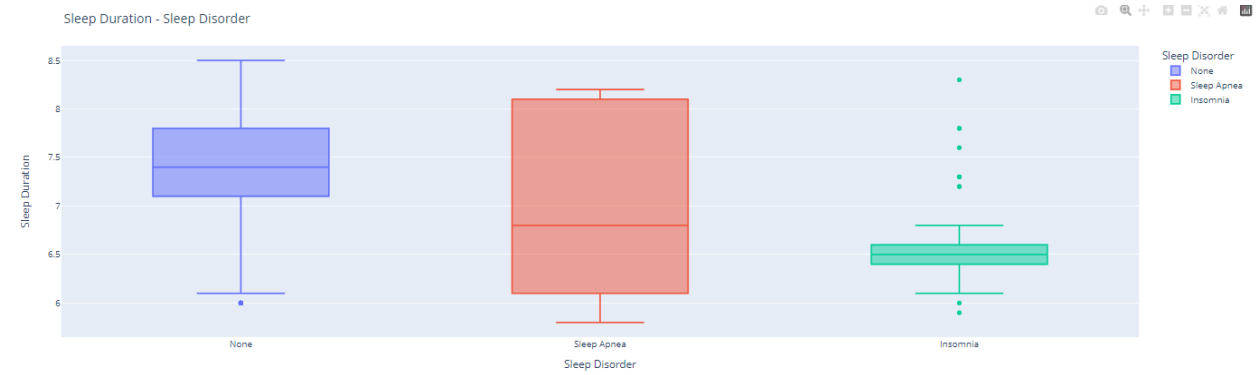


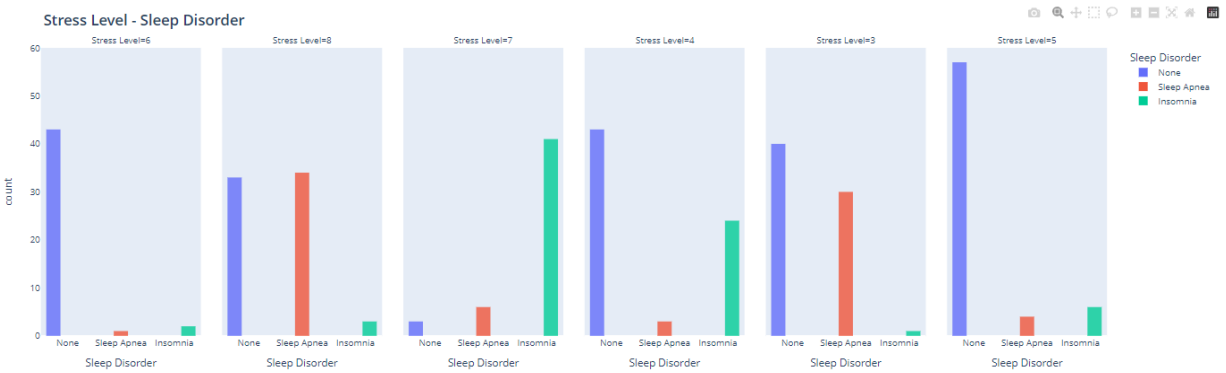
The relationships between most numerical features and 'Sleep Duration' appear to be approximately linear. For example, when the heart rate increases, the sleep duration decreases. While in the bar charts for categorical features, some insights can be taken into considerations, such as salesperson has the least average sleep duration and people with insomnia tend to have a lower sleep duration.

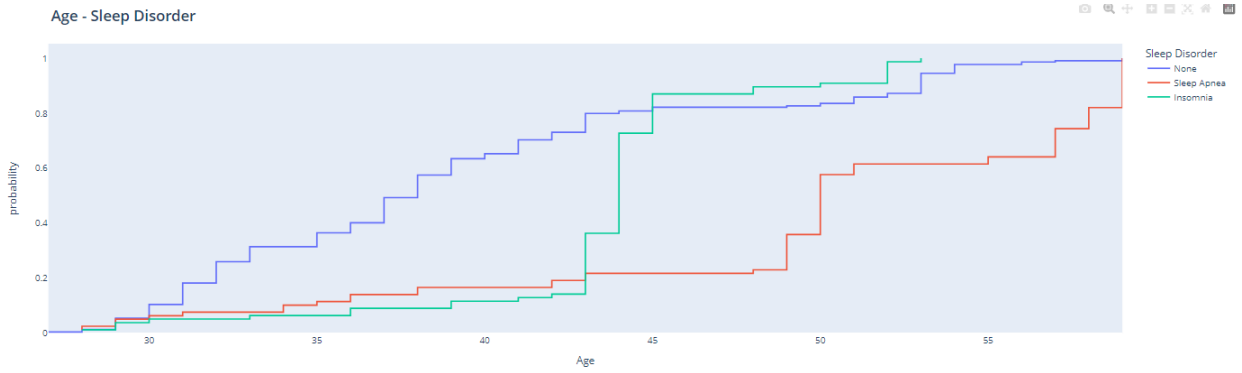
For the following visualizations are mostly related to the 'Sleep Disorder' attribute to produce analysis and insight of the correlation with other features. The visualizations below range from bar charts, pie charts, boxplots to show the relationships between 'Sleep Disorder' and other features.




Physical Activity Level - Sleep Disorder
Physical Activity Level
100
80
60
40
20
None
Sleep Apnea
Insomnia
Sleep Disorder
Sleep Disorder
None
Sleep Apnea
Insomnia


Sleep Duration - Sleep Disorder
Sleep Duration
8.5
8
7.5
7
6.5
6
None
Sleep Apnea
Insomnia
Sleep Disorder
Sleep Disorder
None
Sleep Apnea
Insomnia

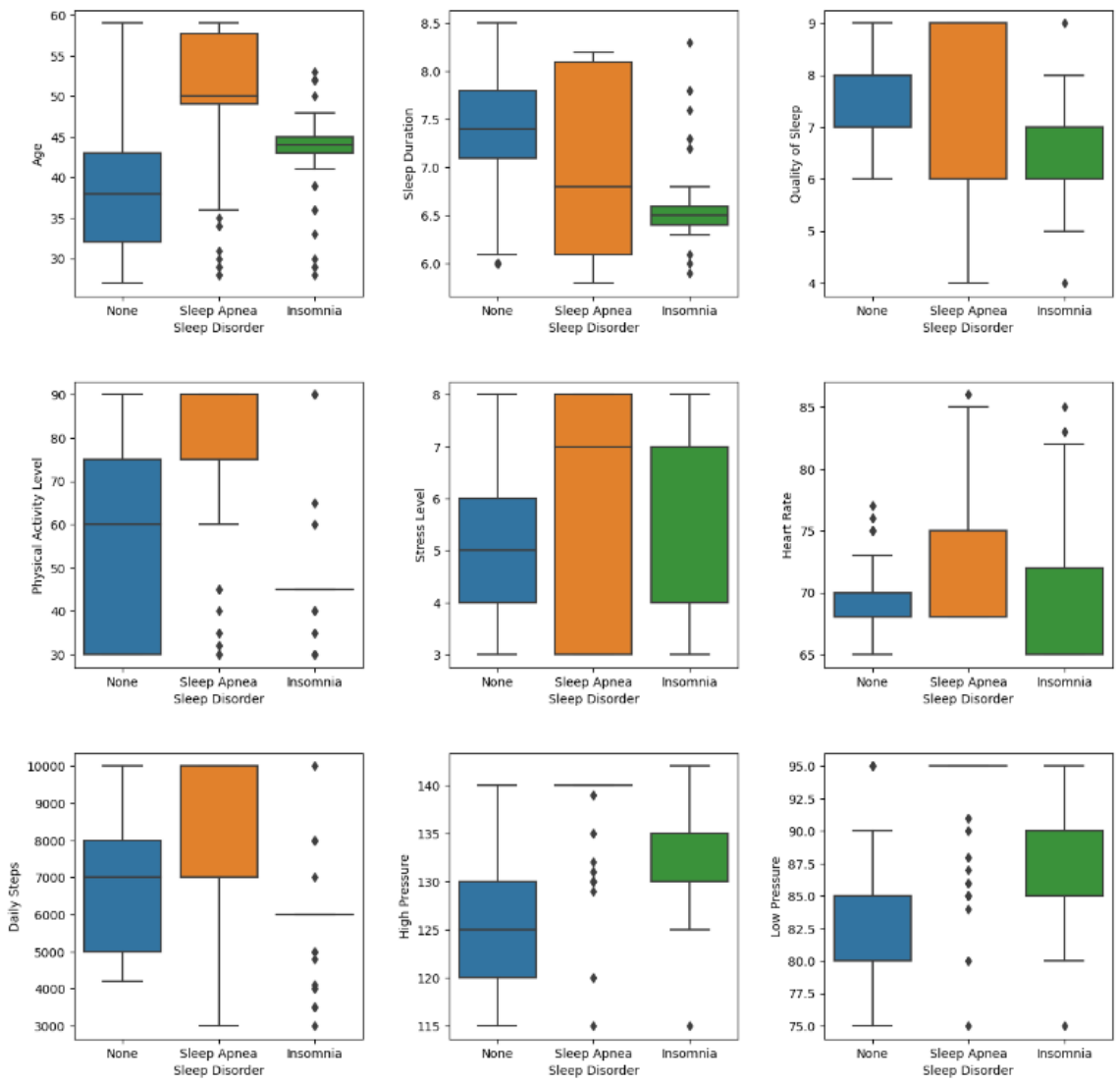

Stress Level - Sleep Disorder
Stress Level=6
Stress Level=8
Stress Level=7
Stress Level=4
Stress Level=3
Stress Level=5
count
60
50
40
30
20
10
0
None
Sleep Apnea
Insomnia
Sleep Disorder
Sleep Disorder
None
Sleep Apnea
Insomnia


Age - Sleep Disorder
probability
1
0.8
0.6
0.4
0.2
0
30
35
40
45
50
55
Age
Sleep Disorder
None
Sleep Apnea
Insomnia

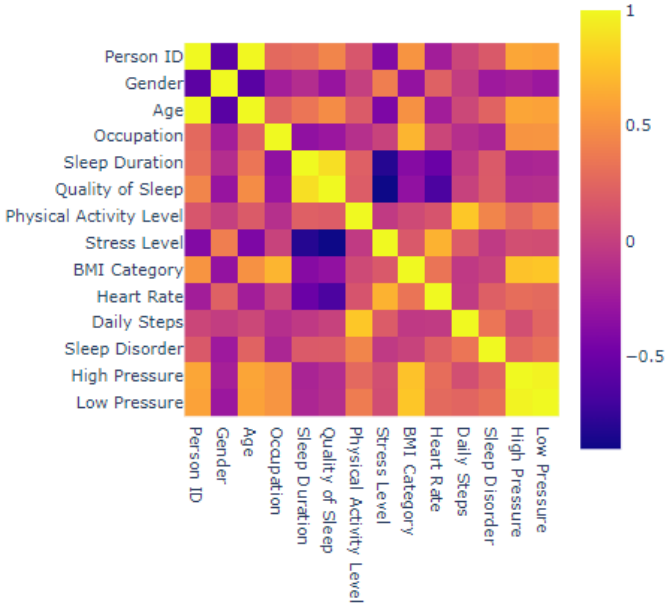
From the visualizations above, they show the following insights:

- more **Normal** men than women
- more men suffer **Insomnia** than women
- more women suffer **Sleep Apnea** than men
- most **Normal** people have high quality of sleep
- people with **Sleep Apnea** have slightly more physical activity
- **Insomnia** people sleep less than others
- **Sleep Apnea** people have high stress level
- people have higher probability to get **Sleep Apnea** on their 40s

The boxplots below show the outliers that lie between the features. It can be seen through the dots that are shown at the end of each boxplot. For example, some **Sleep Apnea** people have low physical activity level.



Correlation Analysis



This correlation matrix helps to show the correlation values between each feature by its heat scaling. The categorical features were encoded before put into process for correlation visualization.

Data Classification Models

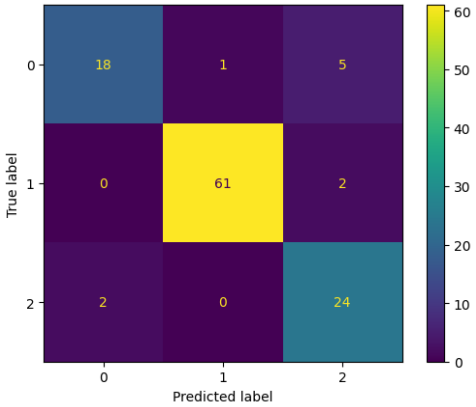
The models below will be used to predict the sleep disorder a person have given a set of data. To start it off, the data frame is split into two parts, one for the input which is all data except ‘Sleep Disorder’ and ‘person ID’ and the second one for the output which is the ‘Sleep Disorder’. Then the input is normalised by passing it into a standardScalar(). Then, the data is split again with 30% of data used for testing and the rest are used for training the model. Finally, the data is ready for the models to use.

Classification Model:

1. Decision Tree

   The first model that is tested on the data is a decision tree model. A randomised search is used to find the best hyper parameter for the model and done with a k fold of 3. Using what is found from the randomised search, A decision tree model is made for this prediction. This model is

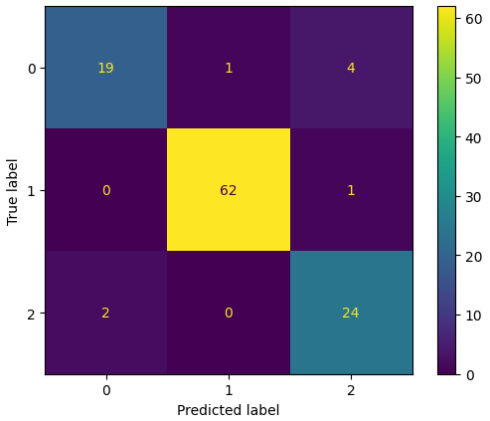
then fitted to the data and has an accuracy of 0.911 after being tested on the data.



	precision	recall	f1-score	support
0	0.90	0.75	0.82	24
1	0.98	0.97	0.98	63
2	0.77	0.92	0.84	26
accuracy			0.91	113
macro avg	0.89	0.88	0.88	113
weighted avg	0.92	0.91	0.91	113

2. Random Forest

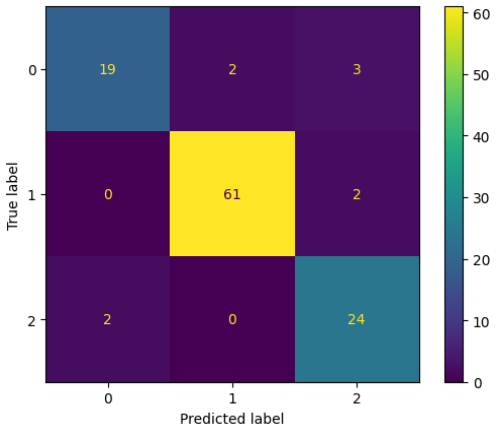
   The next model that is tested on the data is a random forest model. A randomised search is used to find the best hyper parameter for the model and done with a k fold of 5. Using what is found from the randomised search, A random forest model is made for this prediction. This model is then fitted to the data and has an accuracy of 0.929 after being tested on the data.



	precision	recall	f1-score	support
0	0.90	0.79	0.84	24
1	0.98	0.98	0.98	63
2	0.83	0.92	0.87	26
accuracy			0.93	113
macro avg	0.91	0.90	0.90	113
weighted avg	0.93	0.93	0.93	113

3. Logistic Regression

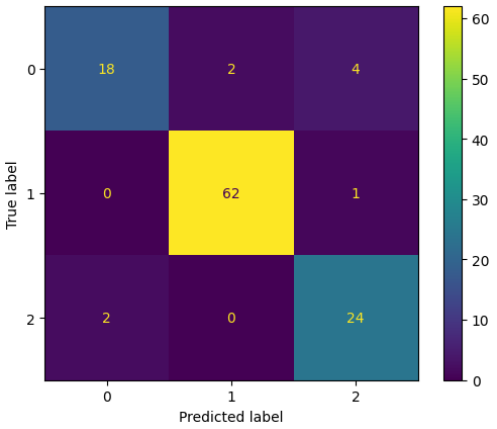
   The third model that is tested on the data is a Logistic Regression model. The hyper parameter that is used for the model is a liblinear solver, max iteration of 10000 and C of 0.1. This model is then fitted to the data and has an accuracy of 0.9203 after being tested on the data.



	precision	recall	f1-score	support
0	0.90	0.79	0.84	24
1	0.97	0.97	0.97	63
2	0.83	0.92	0.87	26
accuracy			0.92	113
macro avg	0.90	0.89	0.90	113
weighted avg	0.92	0.92	0.92	113

4. SVM

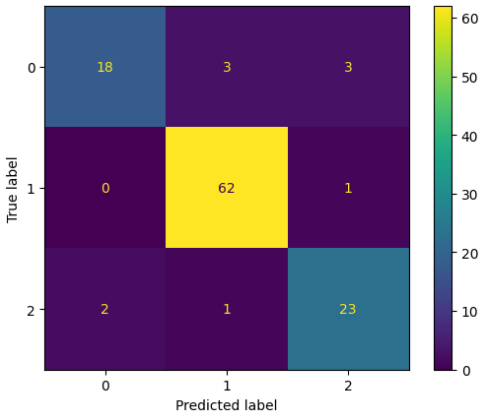
   Next, the model that is tested on the data is a Support vector classification model. The hyper parameter that is used for the model is a poly as kernel, auto for the gamma and C of 10. This model is then fitted to the data and has an accuracy of 0.9203 after being tested on the data.



	precision	recall	f1-score	support
0	0.90	0.75	0.82	24
1	0.97	0.98	0.98	63
2	0.83	0.92	0.87	26
accuracy			0.92	113
macro avg	0.90	0.89	0.89	113
weighted avg	0.92	0.92	0.92	113

5. KNN

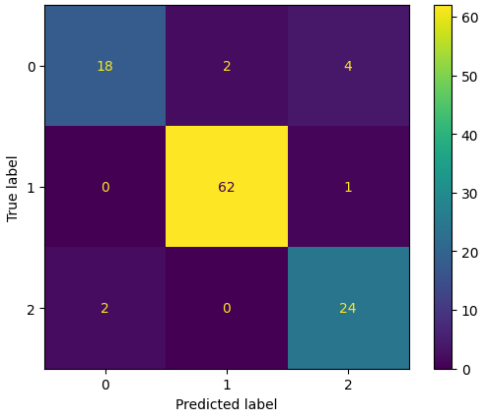
   The fifth model that is tested on the data is a K-nearest neighbour model. A grid search is used to find the best hyper parameter(the number of neighbours) for the model and done with a k fold of 2. Using what is found from the grid search, A KNN model is made for this prediction. This model is then fitted to the data and has an accuracy of 0.911 after being tested on the data.



	precision	recall	f1-score	support
0	0.90	0.75	0.82	24
1	0.94	0.98	0.96	63
2	0.85	0.88	0.87	26
accuracy			0.91	113
macro avg	0.90	0.87	0.88	113
weighted avg	0.91	0.91	0.91	113

6. Gradient Boosting

   The last model that is tested on the data is a Gradient boosting model. A randomised search is used to find the best hyper parameter for the model and done with a k fold of 5. Using what is found from the randomised search, A Gradient boosting model is made for this prediction. This model is then fitted to the data and has an accuracy of 0.9203 after being tested on the data.



	precision	recall	f1-score	support
0	0.90	0.75	0.82	24
1	0.97	0.98	0.98	63
2	0.83	0.92	0.87	26
accuracy			0.92	113
macro avg	0.90	0.89	0.89	113
weighted avg	0.92	0.92	0.92	113

The best performing model out of all these are the random forest model. It has the highest accuracy which is 0.929, slightly better from the rest of the models.