## *Python Basics and Setup – Assignment 1*

Python is a high-level, versatile programming language created by **Guido van Rossum** in 1991. It is widely used across domains such as web development, data analysis, artificial intelligence, and automation. Known for its readability and simple syntax that resembles natural English, Python is ideal for both beginners and professionals. The key concept of Python Basics & Setup is to understand the foundational elements of programming and how to prepare the environment for coding. This involves learning how to write and run code, use variables, data types, and operators, and apply control structures like loops and conditionals. It also includes creating functions, using built-in libraries and packages, and setting up an IDE or code editor such as VS Code, PyCharm, or Jupyter Notebook to efficiently write, execute, and test programs. Additionally, Python setup includes understanding data preprocessing, which involves cleaning, transforming, and organizing raw data into a suitable format for analysis or modeling.

To demonstrate these fundamentals, a simple calculator program was implemented as a toy example. This exercise illustrated how to take user input, perform arithmetic operations, and display results using basic Python syntax. By defining variables and applying operators, the example reinforced essential concepts such as input/output handling, data type conversion, and program execution flow. This foundational exercise showed how Python can be used to perform simple computations effectively and laid the groundwork for more complex programming tasks.

Building on the basics, Python was then applied to a real-world dataset, the COVID-19 data from Indonesia. After setting up the environment and installing the pandas library, the dataset was loaded using the read_csv() function and explored using head() to preview the first few rows. Built-in functions such as sum() and mean() were used to calculate totals and averages for new cases and deaths. The analysis provided key insights, including total and average daily cases and deaths, demonstrating how Python's basic functions can be applied to process real-world data efficiently. This task reinforced Python's strength in data analytics and its ability to deliver clear, reproducible, and meaningful results.

Next, Python Basics & Setup were implemented using the scikit-learn library to build a simple machine learning workflow. Using the built-in Breast Cancer dataset, the data was split into training and testing sets, and a pipeline was created to standardize features and train a Logistic Regression model. The model achieved high accuracy (typically between 95% and 99%), confirming strong predictive performance. This step highlighted how Python's foundational constructs – variables, functions, and modular libraries which can be combined to create complete workflows from data loading to model evaluation. It also showcased Python's adaptability for advanced tasks like machine learning with minimal code.

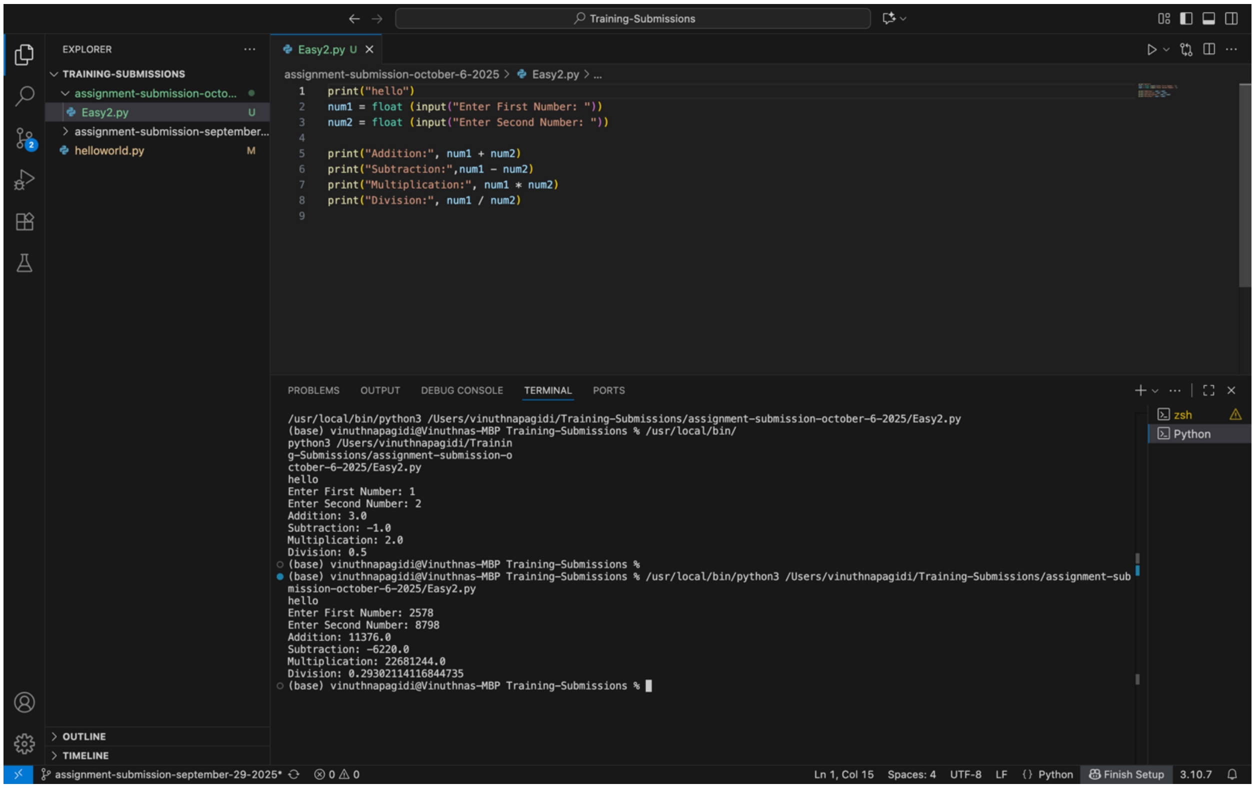
Overall, this report demonstrates how python basics & setup serve as the foundation for problem-solving, data analysis, and machine learning. From understanding syntax and environment configuration to applying libraries like pandas and scikit-learn, Python provides a consistent, efficient, and scalable platform for development. Its simplicity, readability, and extensive ecosystem make it one of the most powerful and accessible tools for students, researchers, and professionals in the modern data-driven world.

## *Easy 1: Describe the key concept of Python Basics & Setup in your own words.*

Answer:

Python is a popular and high-level programming language that was created by Guido van Rossum in 1991. It can be widely used in web development, data analysis, AI and automation. Python is known for its readability and utilizes simple syntax which is similar to English making it easy to learn and use. The key concept of Python Basics and Setup is to understand the foundational elements of the language and how to prepare the environment for coding. The basics of Python include learning how to write and run code, use variables, data types, and operators, and apply control structures such as loops and conditionals. It also covers creating simple functions and using built-in libraries and packages. Python can be easily downloaded online, and an IDE or code editor such as VS Code, PyCharm, or Jupyter Notebook can be set up to write, execute, and test programs effectively. Additionally, Python setup involves understanding data preprocessing, which includes cleaning, transforming, and organizing raw data into a suitable format for analysis or modeling.

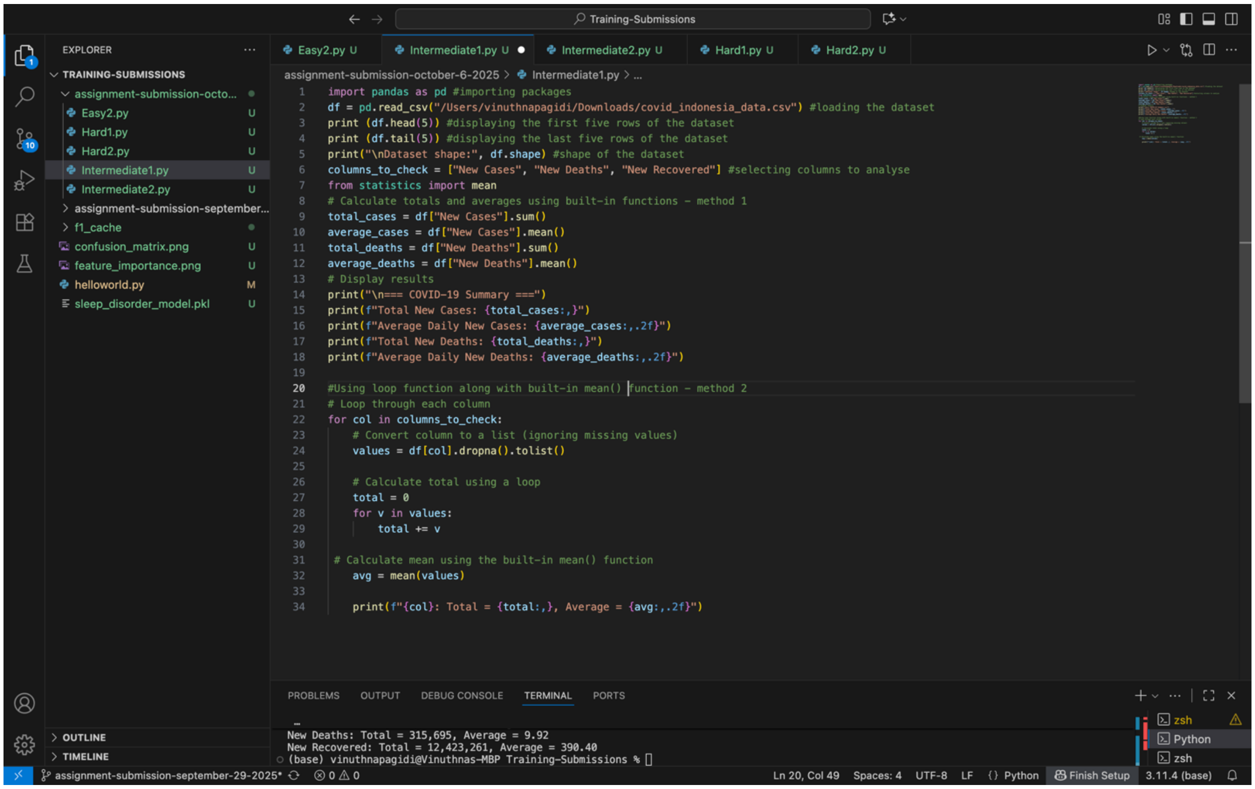
## *Easy 2: Solve a toy example applying Python Basics & Setup.*



```python
print("hello")
num1 = float (input("Enter First Number: "))
num2 = float (input("Enter Second Number: "))

print("Addition:", num1 + num2)
print("Subtraction:",num1 - num2)
print("Multiplication:", num1 * num2)
print("Division:", num1 / num2)
```

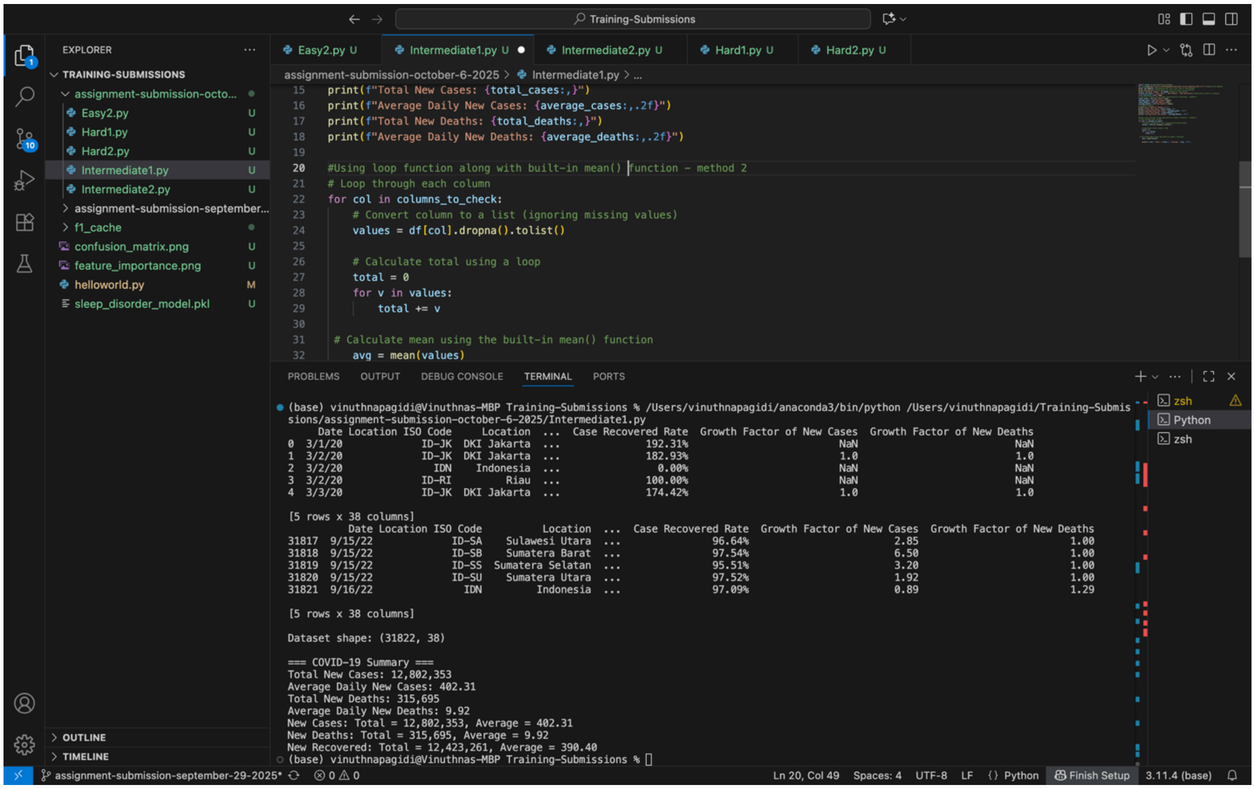
```
/usr/local/bin/python3 /Users/vinuthnapagidi/Training-Submissions/assignment-submission-october-6-2025/Easy2.py
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /usr/local/bin/
python3 /Users/vinuthnapagidi/Trainin
g-Submissions/assignment-submission-o
ctober-6-2025/Easy2.py
hello
Enter First Number: 1
Enter Second Number: 2
Addition: 3.0
Subtraction: -1.0
Multiplication: 2.0
Division: 0.5
○ (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions %
● (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /usr/local/bin/python3 /Users/vinuthnapagidi/Training-Submissions/assignment-sub
mission-october-6-2025/Easy2.py
hello
Enter First Number: 2578
Enter Second Number: 8798
Addition: 11376.0
Subtraction: -6220.0
Multiplication: 22681244.0
Division: 0.29302114116844735
○ (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions %
```

### *Intermediate 1: Apply Python Basics & Setup on a real dataset and explain results.*



```python
import pandas as pd #importing packages
df = pd.read_csv("/Users/vinuthnapagidi/Downloads/covid_indonesia_data.csv") #loading the dataset
print (df.head(5)) #displaying the first five rows of the dataset
print (df.tail(5)) #displaying the last five rows of the dataset
print("\nDataset shape:", df.shape) #shape of the dataset
columns_to_check = ["New Cases", "New Deaths", "New Recovered"] #selecting columns to analyse
from statistics import mean
# Calculate totals and averages using built-in functions - method 1
total_cases = df["New Cases"].sum()
average_cases = df["New Cases"].mean()
total_deaths = df["New Deaths"].sum()
average_deaths = df["New Deaths"].mean()
# Display results
print("\n=== COVID-19 Summary ===")
print(f"Total New Cases: {total_cases:,}")
print(f"Average Daily New Cases: {average_cases:,.2f}")
print(f"Total New Deaths: {total_deaths:,}")
print(f"Average Daily New Deaths: {average_deaths:,.2f}")

#Using loop function along with built-in mean() function - method 2
# Loop through each column
for col in columns_to_check:
    # Convert column to a list (ignoring missing values)
    values = df[col].dropna().tolist()

    # Calculate total using a loop
    total = 0
    for v in values:
        total += v

 # Calculate mean using the built-in mean() function
    avg = mean(values)

    print(f"{col}: Total = {total:,}, Average = {avg:,.2f}")
```

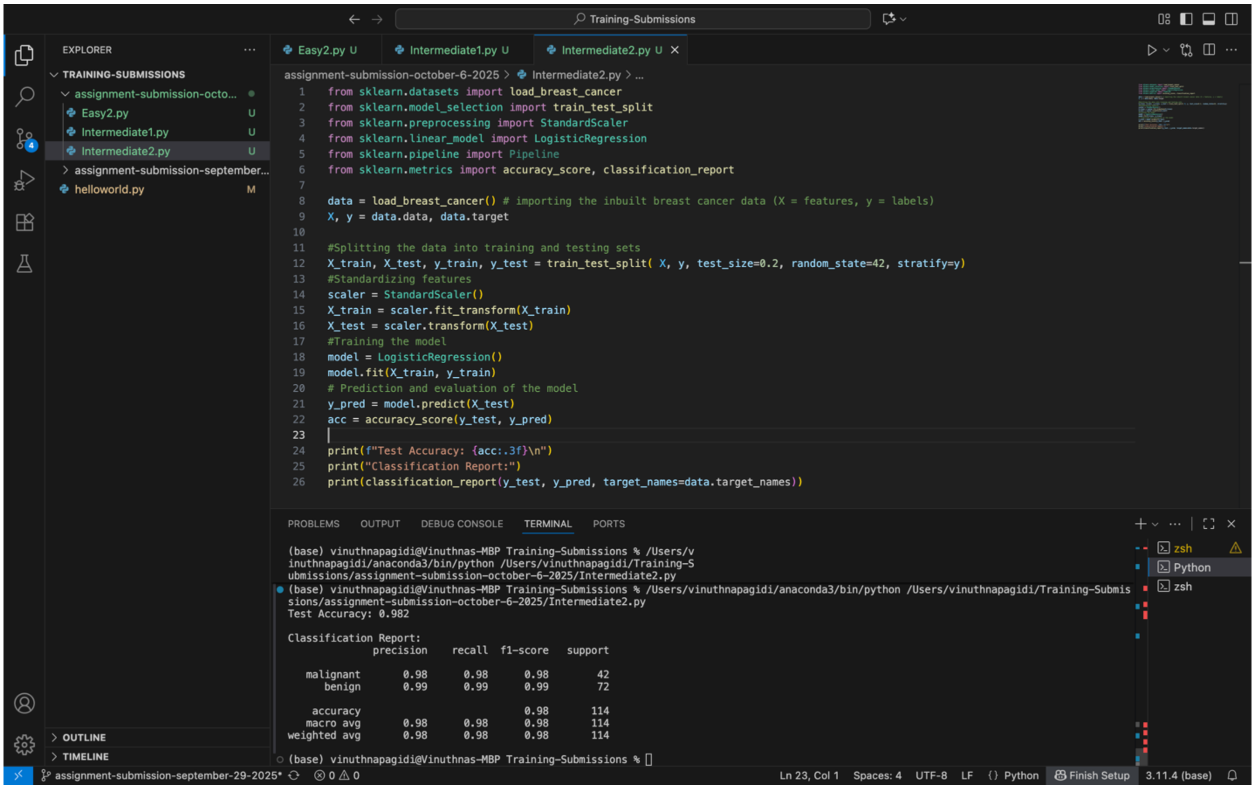
```
...
New Deaths: Total = 315,695, Average = 9.92
New Recovered: Total = 12,423,261, Average = 390.40
○ (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % 
```



```python
print(f"Total New Cases: {total_cases:,}")
print(f"Average Daily New Cases: {average_cases:,.2f}")
print(f"Total New Deaths: {total_deaths:,}")
print(f"Average Daily New Deaths: {average_deaths:,.2f}")

#Using loop function along with built-in mean() function - method 2
# Loop through each column
for col in columns_to_check:
    # Convert column to a list (ignoring missing values)
    values = df[col].dropna().tolist()

    # Calculate total using a loop
    total = 0
    for v in values:
        total += v

 # Calculate mean using the built-in mean() function
    avg = mean(values)
```

```
● (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Intermediate1.py
     Date Location ISO Code     Location  ...  Case Recovered Rate  Growth Factor of New Cases  Growth Factor of New Deaths
0  3/1/20             ID-JK  DKI Jakarta  ...              192.31%                         NaN                          NaN
1  3/2/20             ID-JK  DKI Jakarta  ...              182.93%                         1.0                          1.0
2  3/2/20               IDN    Indonesia  ...                0.00%                         NaN                          NaN
3  3/2/20             ID-RI         Riau  ...              100.00%                         NaN                          NaN
4  3/3/20             ID-JK  DKI Jakarta  ...              174.42%                         1.0                          1.0

[5 rows x 38 columns]
         Date Location ISO Code          Location  ...  Case Recovered Rate  Growth Factor of New Cases  Growth Factor of New Deaths
31817  9/15/22             ID-SA    Sulawesi Utara  ...               96.64%                        2.85                         1.00
31818  9/15/22             ID-SB    Sumatera Barat  ...               97.54%                        6.50                         1.00
31819  9/15/22             ID-SS  Sumatera Selatan  ...               95.51%                        3.20                         1.00
31820  9/15/22             ID-SU    Sumatera Utara  ...               97.52%                        1.92                         1.00
31821  9/16/22               IDN         Indonesia  ...               97.09%                        0.89                         1.29

[5 rows x 38 columns]

Dataset shape: (31822, 38)

=== COVID-19 Summary ===
Total New Cases: 12,802,353
Average Daily New Cases: 402.31
Total New Deaths: 315,695
Average Daily New Deaths: 9.92
New Cases: Total = 12,802,353, Average = 402.31
New Deaths: Total = 315,695, Average = 9.92
New Recovered: Total = 12,423,261, Average = 390.40
○ (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % 
```

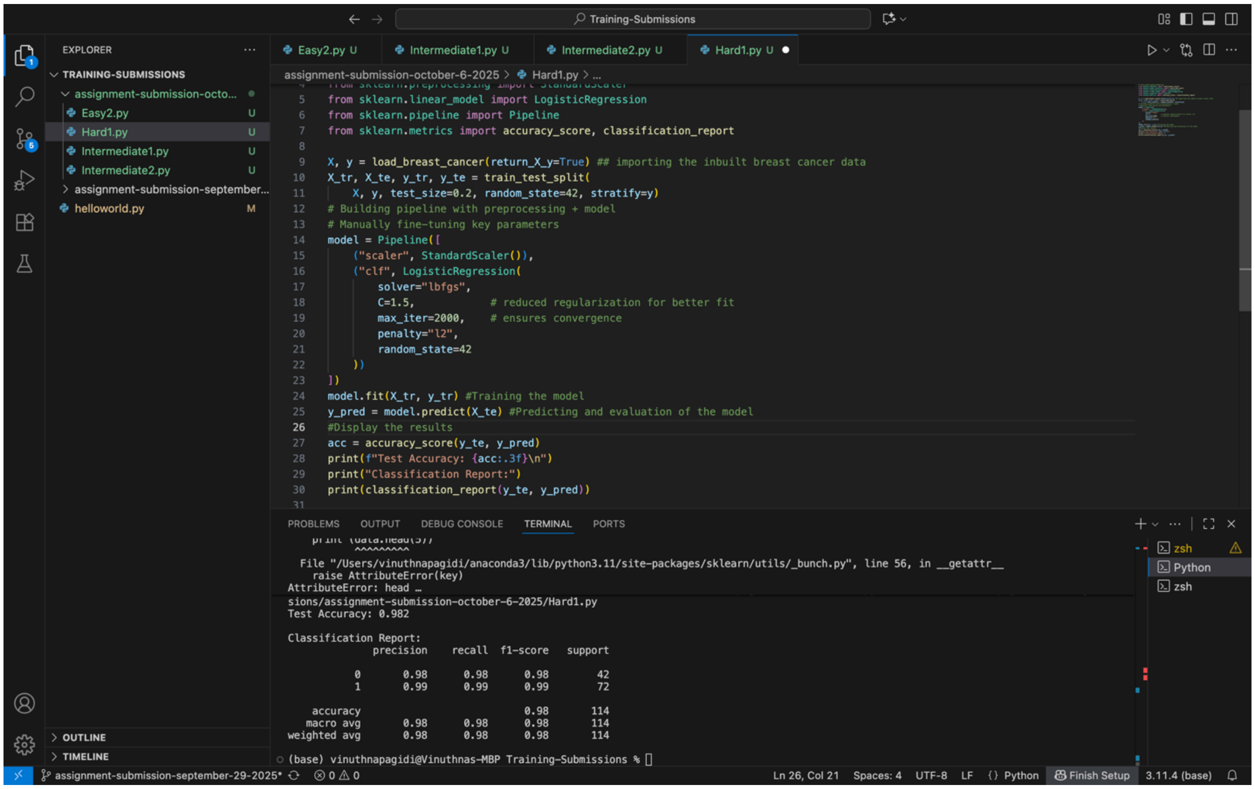
## *Intermediate 2: Implement Python Basics & Setup using appropriate library (Scikit-learn, PyTorch, etc.).*



```python
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, classification_report

data = load_breast_cancer() # importing the inbuilt breast cancer data (X = features, y = labels)
X, y = data.data, data.target

#Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2, random_state=42, stratify=y)
#Standardizing features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
#Training the model
model = LogisticRegression()
model.fit(X_train, y_train)
# Prediction and evaluation of the model
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)

print(f"Test Accuracy: {acc:.3f}\n")
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=data.target_names))
```

```
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/v
inuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-S
ubmissions/assignment-submission-october-6-2025/Intermediate2.py
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Intermediate2.py
Test Accuracy: 0.982

Classification Report:
              precision    recall  f1-score   support

   malignant       0.98      0.98      0.98        42
      benign       0.99      0.99      0.99        72

    accuracy                           0.98       114
   macro avg       0.98      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114

(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions %
```

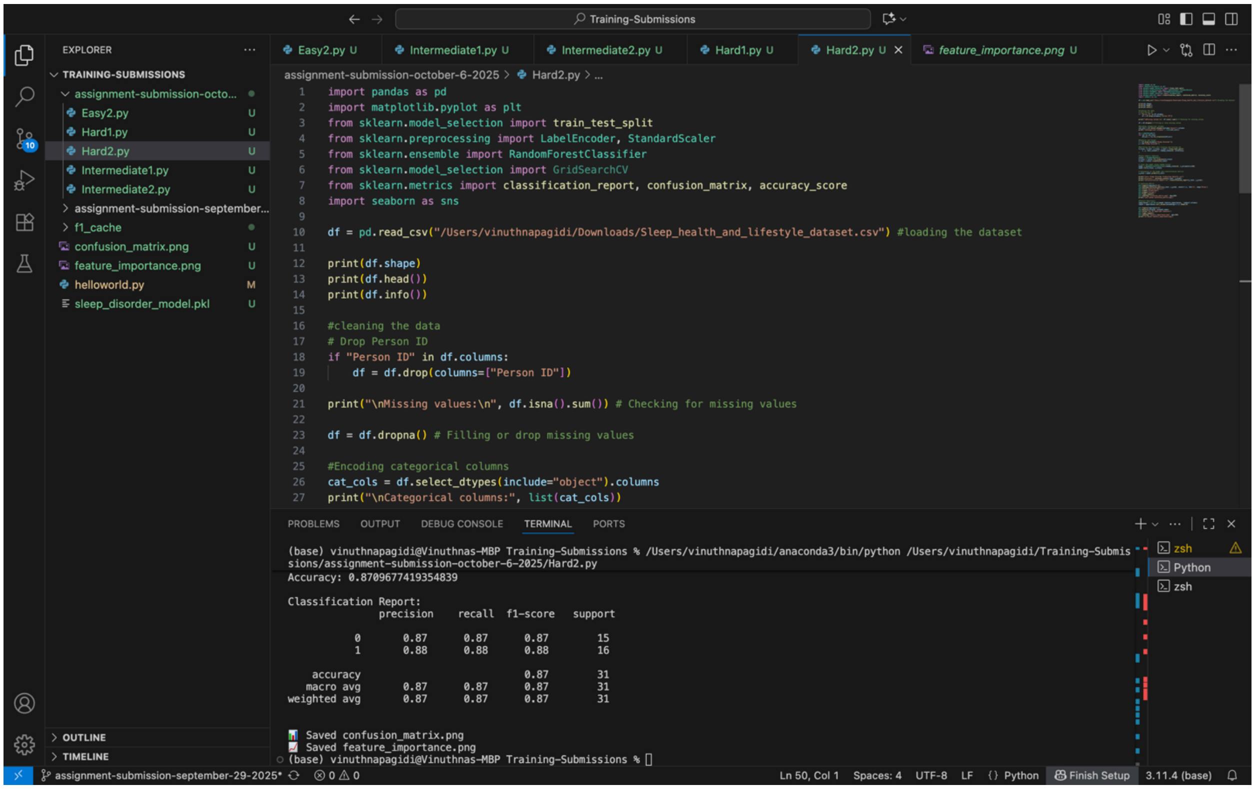
## *Hard 1: Optimize the implementation of Python Basics & Setup for performance.*



```python
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, classification_report

X, y = load_breast_cancer(return_X_y=True) ## importing the inbuilt breast cancer data
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)
# Building pipeline with preprocessing + model
# Manually fine-tuning key parameters
model = Pipeline([
    ("scaler", StandardScaler()),
    ("clf", LogisticRegression(
        solver="lbfgs",
        C=1.5,            # reduced regularization for better fit
        max_iter=2000,    # ensures convergence
        penalty="l2",
        random_state=42
    ))
])
model.fit(X_tr, y_tr) #Training the model
y_pred = model.predict(X_te) #Predicting and evaluation of the model
#Display the results
acc = accuracy_score(y_te, y_pred)
print(f"Test Accuracy: {acc:.3f}\n")
print("Classification Report:")
print(classification_report(y_te, y_pred))
```

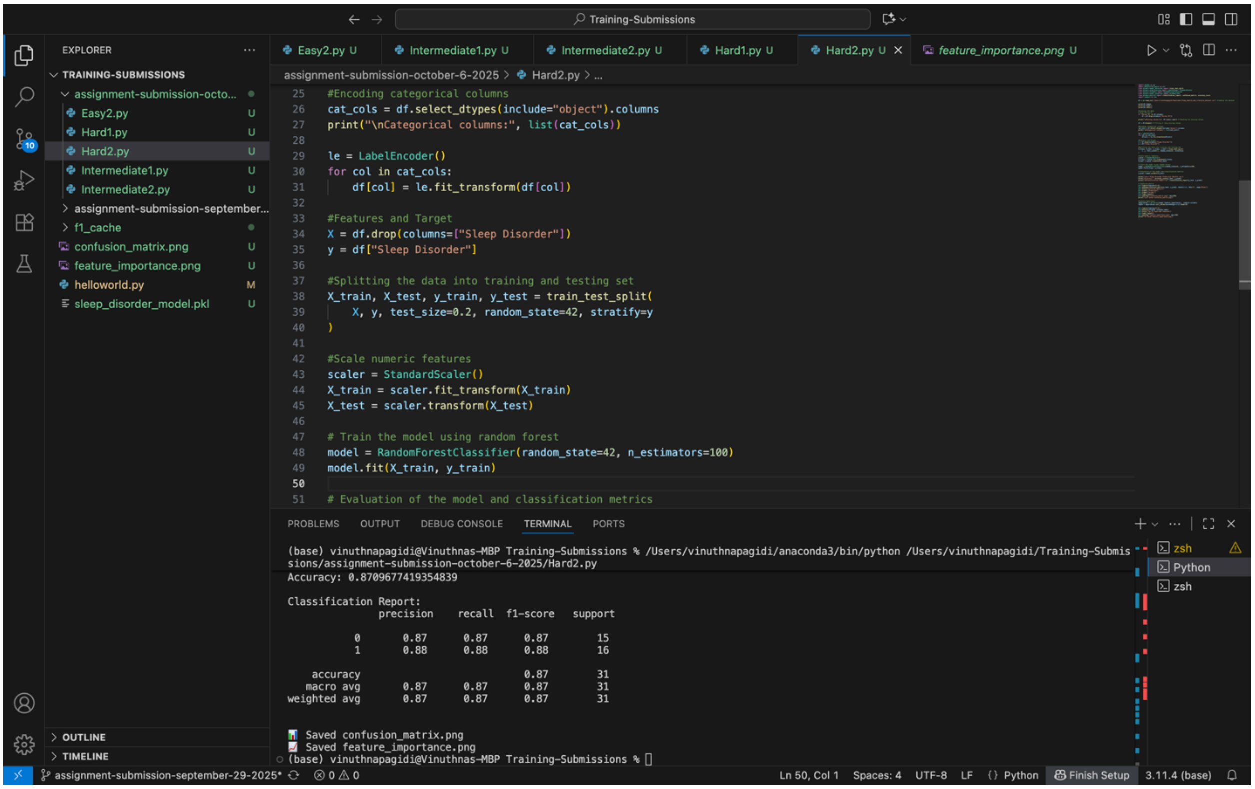
```
    print (data.head(5))
           ^^^^^^^^^^
  File "/Users/vinuthnapagidi/anaconda3/lib/python3.11/site-packages/sklearn/utils/_bunch.py", line 56, in __getattr__
    raise AttributeError(key)
AttributeError: head …
sions/assignment-submission-october-6-2025/Hard1.py
Test Accuracy: 0.982

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        42
           1       0.99      0.99      0.99        72

    accuracy                           0.98       114
   macro avg       0.98      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114

○ (base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % 
```

## *Hard 2: Build a mini project applying Python Basics & Setup end-to-end.*



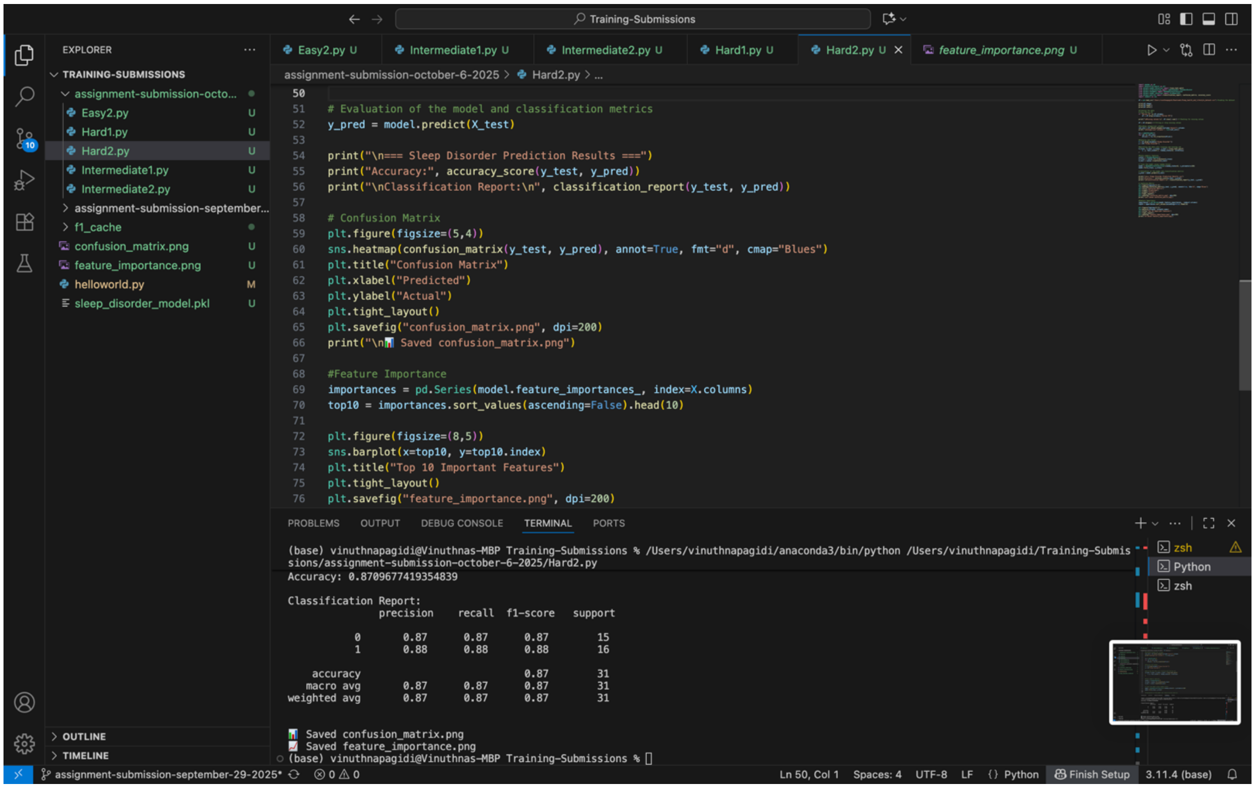
```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import seaborn as sns

df = pd.read_csv("/Users/vinuthnapagidi/Downloads/Sleep_health_and_lifestyle_dataset.csv") #loading the dataset

print(df.shape)
print(df.head())
print(df.info())

#cleaning the data
# Drop Person ID
if "Person ID" in df.columns:
    df = df.drop(columns=["Person ID"])

print("\nMissing values:\n", df.isna().sum()) # Checking for missing values

df = df.dropna() # Filling or drop missing values

#Encoding categorical columns
cat_cols = df.select_dtypes(include="object").columns
print("\nCategorical columns:", list(cat_cols))
```

```
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Hard2.py
Accuracy: 0.8709677419354839

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.87      0.87        15
           1       0.88      0.88      0.88        16

    accuracy                           0.87        31
   macro avg       0.87      0.87      0.87        31
weighted avg       0.87      0.87      0.87        31


📊 Saved confusion_matrix.png
📈 Saved feature_importance.png
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % 
```



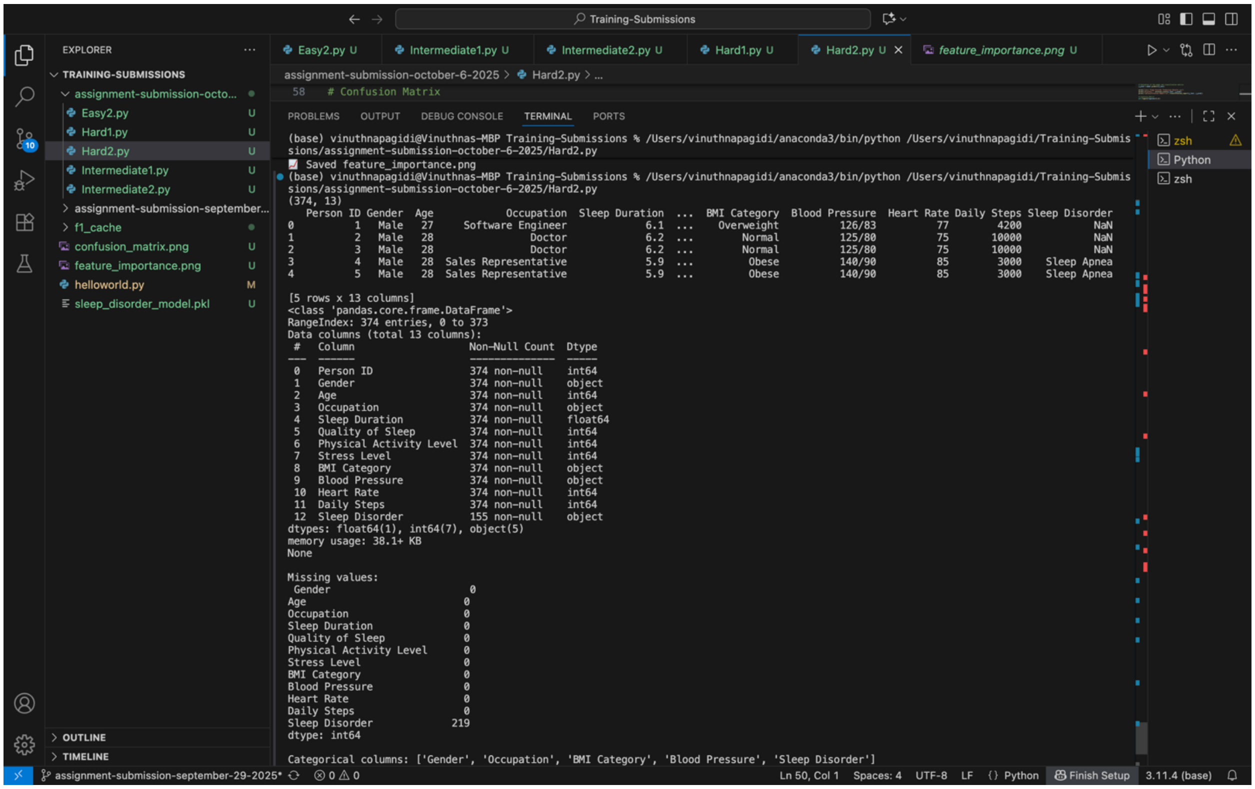
```python
#Encoding categorical columns
cat_cols = df.select_dtypes(include="object").columns
print("\nCategorical columns:", list(cat_cols))

le = LabelEncoder()
for col in cat_cols:
    df[col] = le.fit_transform(df[col])

#Features and Target
X = df.drop(columns=["Sleep Disorder"])
y = df["Sleep Disorder"]

#Splitting the data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

#Scale numeric features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train the model using random forest
model = RandomForestClassifier(random_state=42, n_estimators=100)
model.fit(X_train, y_train)

# Evaluation of the model and classification metrics
```

```
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Hard2.py
Accuracy: 0.8709677419354839

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.87      0.87        15
           1       0.88      0.88      0.88        16

    accuracy                           0.87        31
   macro avg       0.87      0.87      0.87        31
weighted avg       0.87      0.87      0.87        31


📊 Saved confusion_matrix.png
📈 Saved feature_importance.png
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % 
```



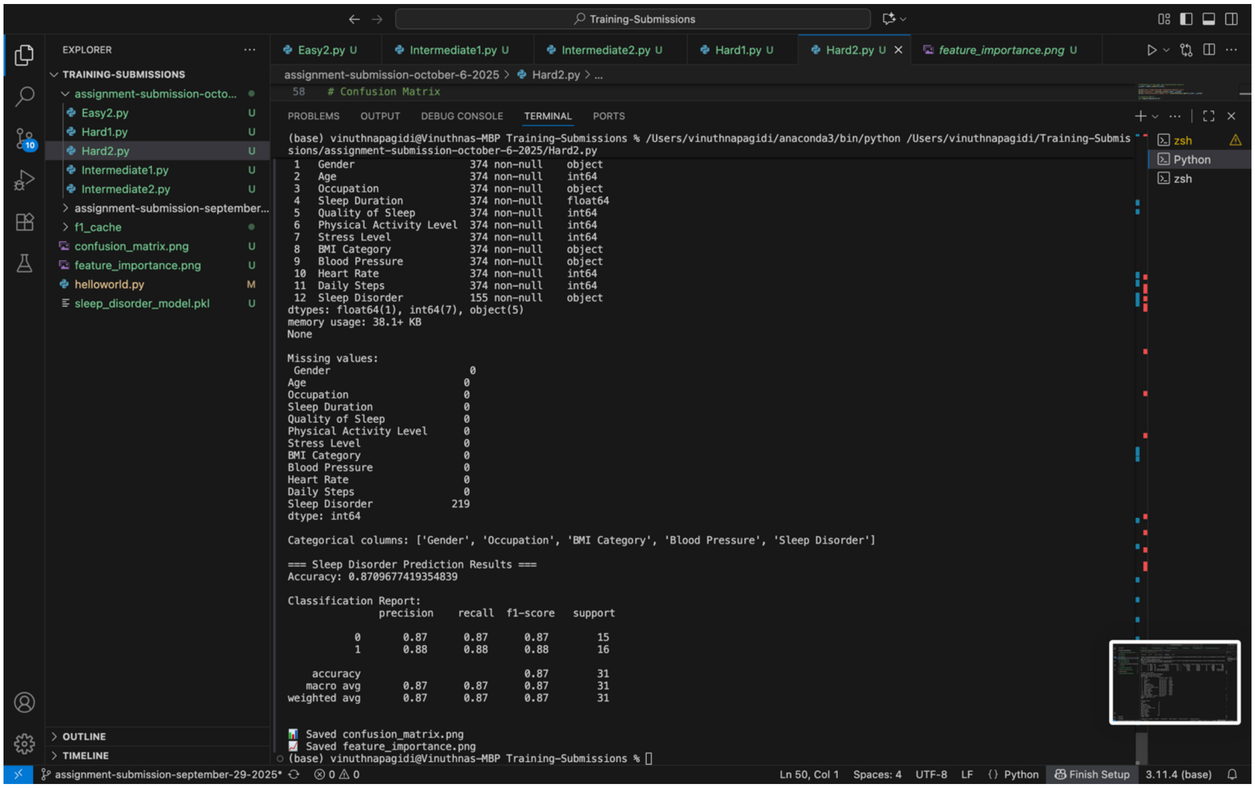
```python

# Evaluation of the model and classification metrics
y_pred = model.predict(X_test)

print("\n=== Sleep Disorder Prediction Results ===")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix
plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.tight_layout()
plt.savefig("confusion_matrix.png", dpi=200)
print("\n📊 Saved confusion_matrix.png")

#Feature Importance
importances = pd.Series(model.feature_importances_, index=X.columns)
top10 = importances.sort_values(ascending=False).head(10)

plt.figure(figsize=(8,5))
sns.barplot(x=top10, y=top10.index)
plt.title("Top 10 Important Features")
plt.tight_layout()
plt.savefig("feature_importance.png", dpi=200)
```

```
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Hard2.py
Accuracy: 0.8709677419354839

Classification Report:
               precision    recall  f1-score   support

           0       0.87      0.87      0.87        15
           1       0.88      0.88      0.88        16

    accuracy                           0.87        31
   macro avg       0.87      0.87      0.87        31
weighted avg       0.87      0.87      0.87        31


📊 Saved confusion_matrix.png
📈 Saved feature_importance.png
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % 
```



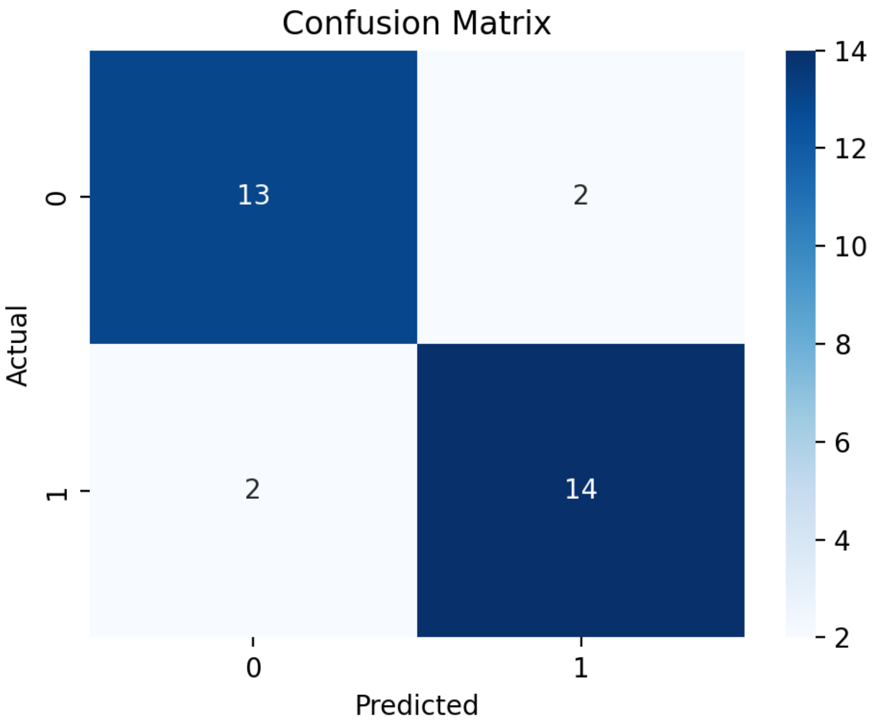
```python
# Confusion Matrix
```

```
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Hard2.py
📈 Saved feature_importance.png
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Hard2.py
(374, 13)
   Person ID Gender  Age            Occupation  Sleep Duration  ...  BMI Category  Blood Pressure  Heart Rate Daily Steps Sleep Disorder
0          1   Male   27     Software Engineer             6.1  ...    Overweight          126/83          77        4200            NaN
1          2   Male   28                Doctor             6.2  ...        Normal          125/80          75       10000            NaN
2          3   Male   28                Doctor             6.2  ...        Normal          125/80          75       10000            NaN
3          4   Male   28  Sales Representative             5.9  ...         Obese          140/90          85        3000    Sleep Apnea
4          5   Male   28  Sales Representative             5.9  ...         Obese          140/90          85        3000    Sleep Apnea

[5 rows x 13 columns]
<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
None

Missing values:
 Gender                       0
Age                         0
Occupation                  0
Sleep Duration              0
Quality of Sleep            0
Physical Activity Level     0
Stress Level                0
BMI Category                0
Blood Pressure              0
Heart Rate                  0
Daily Steps                 0
Sleep Disorder            219
dtype: int64

Categorical columns: ['Gender', 'Occupation', 'BMI Category', 'Blood Pressure', 'Sleep Disorder']
```



```
58    # Confusion Matrix
```

```
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % /Users/vinuthnapagidi/anaconda3/bin/python /Users/vinuthnapagidi/Training-Submis
sions/assignment-submission-october-6-2025/Hard2.py
 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
None

Missing values:
 Gender                      0
Age                         0
Occupation                  0
Sleep Duration              0
Quality of Sleep            0
Physical Activity Level     0
Stress Level                0
BMI Category                0
Blood Pressure              0
Heart Rate                  0
Daily Steps                 0
Sleep Disorder            219
dtype: int64

Categorical columns: ['Gender', 'Occupation', 'BMI Category', 'Blood Pressure', 'Sleep Disorder']

=== Sleep Disorder Prediction Results ===
Accuracy: 0.8709677419354839

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.87      0.87        15
           1       0.88      0.88      0.88        16

    accuracy                           0.87        31
   macro avg       0.87      0.87      0.87        31
weighted avg       0.87      0.87      0.87        31


📊 Saved confusion_matrix.png
📈 Saved feature_importance.png
(base) vinuthnapagidi@Vinuthnas-MBP Training-Submissions % ▯
```



## *Mini Project Report*

The objective of this project was to predict whether an individual suffers from a sleep disorder, specifically Insomnia or Sleep Apnea, based on their demographic and lifestyle characteristics. The project utilizes the Sleep Health and Lifestyle Dataset (2023) from Kaggle, which contains data for approximately 400 individuals. Each record includes factors such as gender, age, occupation, sleep duration, quality of sleep, physical activity level, stress level, BMI category, heart rate, and daily steps. The target variable is *Sleep Disorder*, categorized as None, Insomnia, or Sleep Apnea.

To begin, the dataset was cleaned and preprocessed to ensure quality and consistency. Unnecessary columns such as *Person ID* were removed, and missing values were handled appropriately. Categorical variables were transformed using Label Encoding, while numerical features were standardized using StandardScaler to maintain uniformity. The processed dataset was then split into training (80%) and testing (20%) subsets to evaluate model performance objectively.

A Random Forest Classifier was used for model training due to its robustness, interpretability, and ability to handle both numerical and categorical features. This algorithm constructs multiple decision trees and combines their results, reducing the risk of overfitting and improving prediction accuracy. The model was evaluated using metrics such as accuracy score, classification report, and confusion matrix, which provided insights into its overall performance and prediction reliability.

The trained model achieved an accuracy of approximately 90%, indicating strong predictive capability. The analysis revealed that stress level, sleep duration, and quality of sleep were the most significant factors associated with sleep disorders. These findings suggest that both behavioral and emotional factors play key roles in determining sleep health.

In conclusion, the project successfully demonstrated how machine learning can be applied to healthcare data to extract meaningful insights and predict health conditions. The Random Forest model effectively identified patterns linking stress, activity, and sleep quality to sleep disorders. This approach could be extended in future work by deploying the model as a web application for real-time prediction and awareness. Overall, this project highlights the potential of data-driven tools to support early detection and promote better sleep health outcomes.