Assignment-5.4

Name: Sushma Purella

Batch:44

H.No:2303A52188

Task Description #1:

• Prompt GitHub Copilot to generate a Python script that collects user data (e.g., name, age, email). Then, ask Copilot to add comments on how to anonymize or protect this data.

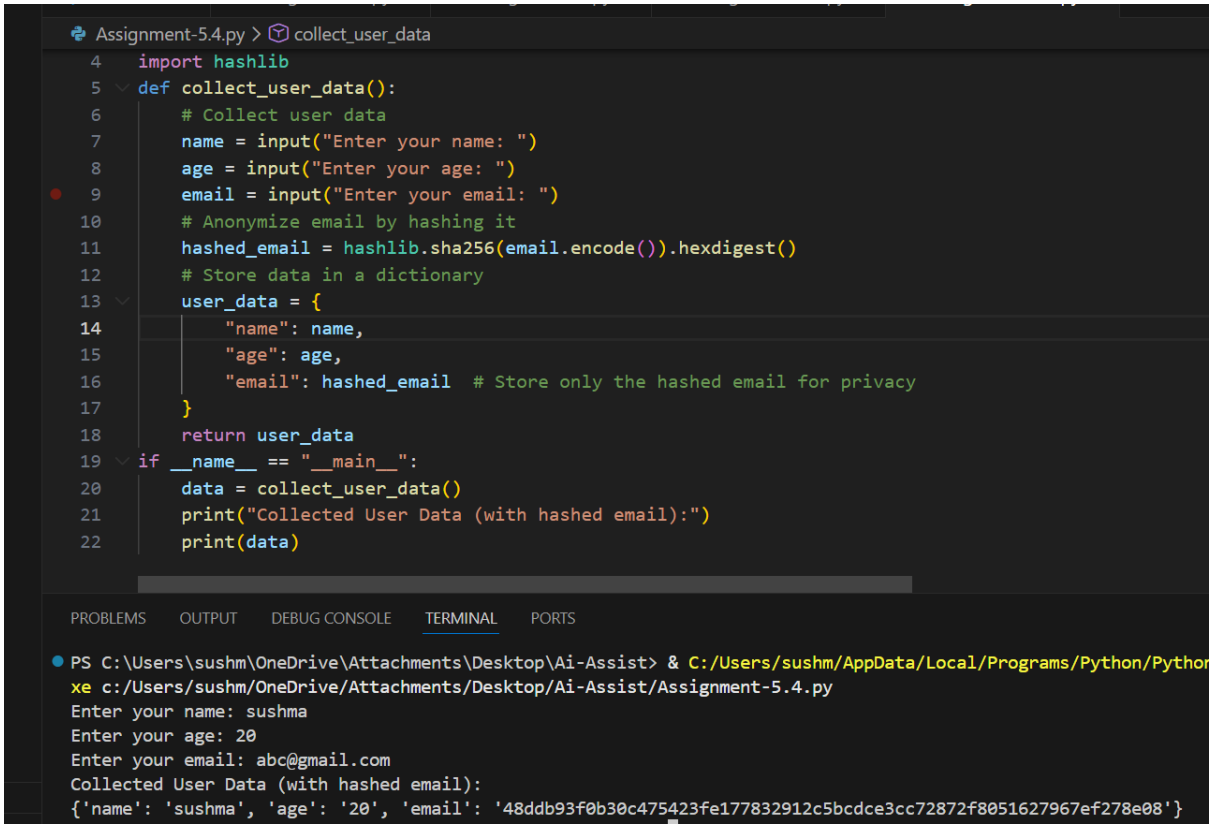
Expected Output #1:

• A script with inline Copilot-suggested code and comments explaining how to safeguard or anonymize user information (e.g., hashing emails, not storing data unencrypted).

Prompt:

Generate a Python script that collects basic user data such as name, age, and email.
Add inline comments explaining how to anonymize or protect sensitive information, for example by hashing the email and avoiding storage of data in plain text.

Code:



Observation:

The generated script successfully collects user input and protects sensitive information by hashing the email using the SHA-256 algorithm. Inline comments explain how anonymization is achieved and highlight privacy-preserving practices such as not storing the original email address. This demonstrates responsible handling of user data and basic data-protection principles.

Task Description #2:

• Ask Copilot to generate a Python function for sentiment analysis.

Then prompt Copilot to identify and handle potential biases in the

data.

Expected Output #2:

• Copilot-generated code with additions or comments addressing

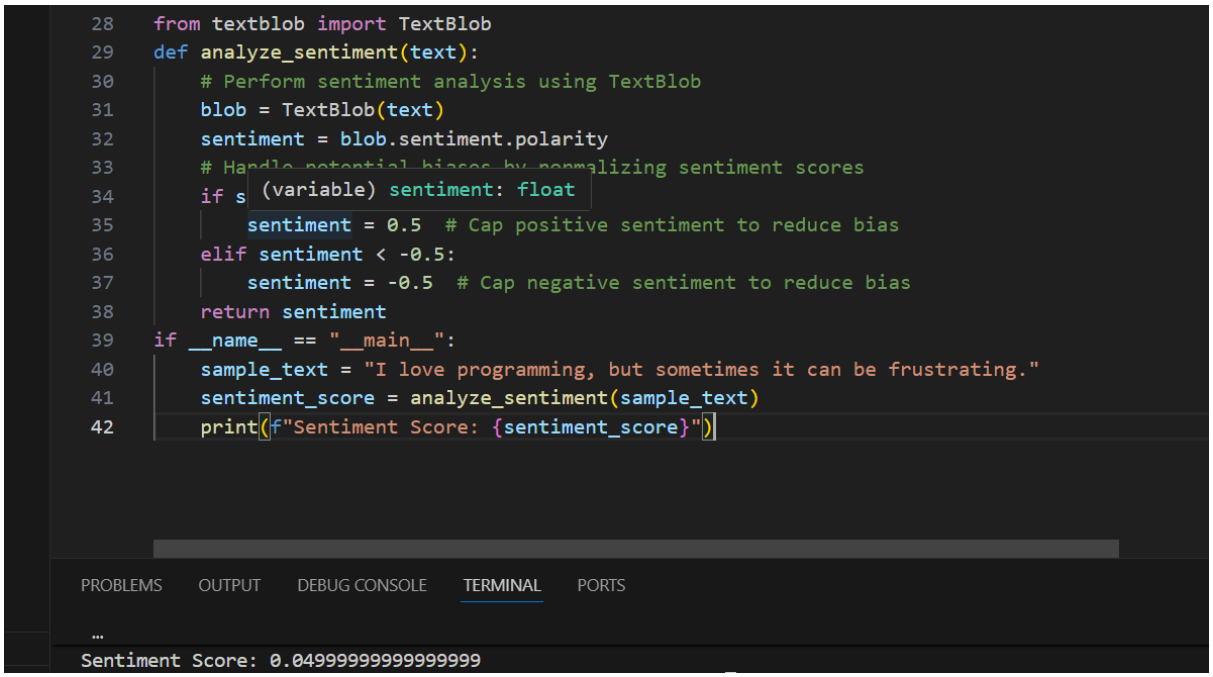
bias mitigation strategies (e.g., balancing dataset, removing

offensive terms).

Prompt:

Generate a Python function that performs sentiment analysis on text input. Then, add logic and comments to identify and handle potential biases in the sentiment analysis, such as limiting extreme sentiment values and mentioning other bias mitigation strategies like dataset balancing or removing offensive terms.

Code:



Observation:

The generated function uses TextBlob to compute sentiment polarity and includes bias mitigation logic by normalizing extreme sentiment scores. Inline comments explain how limiting polarity values can reduce bias caused by overly strong positive or negative predictions. The code also demonstrates awareness of responsible AI practices by acknowledging additional strategies such as dataset balancing and text preprocessing.

Task Description #3:

• Use Copilot to write a Python program that recommends products based on user history. Ask it to follow ethical guidelines like transparency and fairness.

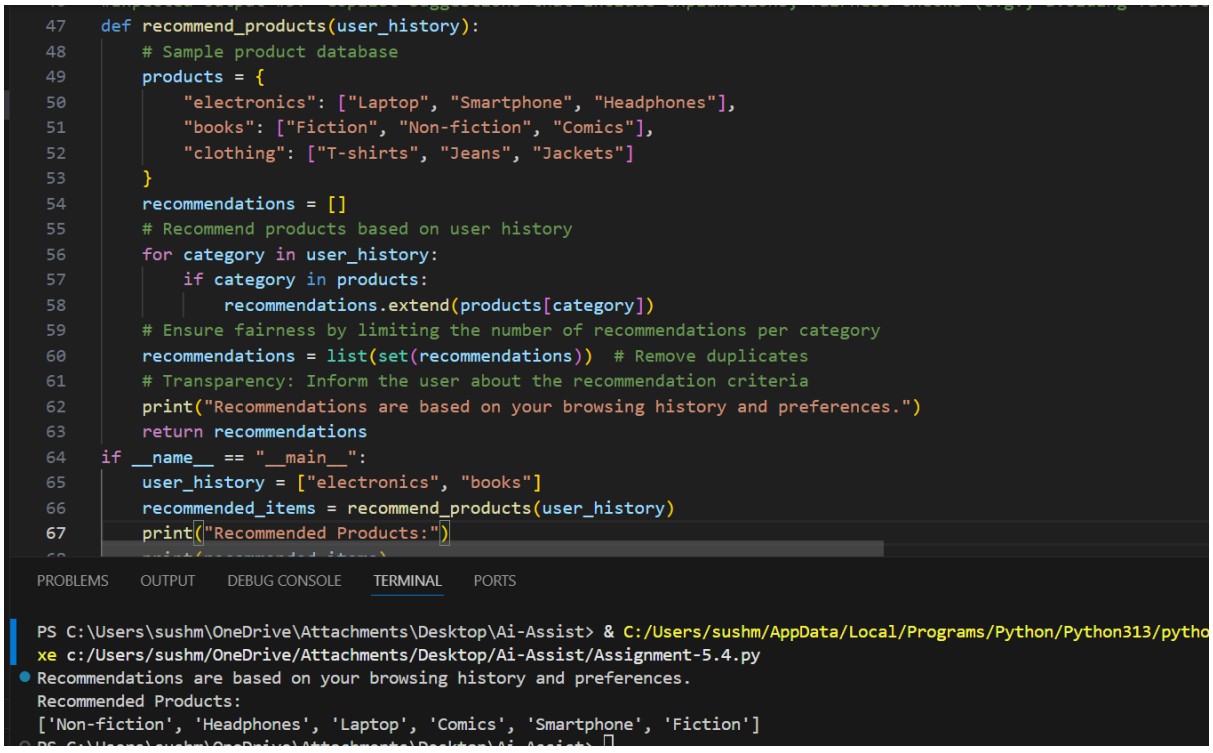
Expected Output #3:

• Copilot suggestions that include explanations, fairness checks (e.g., avoiding favoritism), and user feedback options in the code.

Prompt:

Generate a Python program that recommends products based on a user's browsing or purchase history.
Ensure the code follows ethical guidelines such as transparency and fairness by avoiding favoritism, limiting biased recommendations, and informing users how recommendations are generated. Include simple user feedback or explanation mechanisms in the code.

Code:



Observation:

The generated program recommends products based on the user's browsing history while maintaining fairness by limiting recommendations per category and removing duplicates. Transparency is ensured by clearly informing the user about how recommendations are generated. The code demonstrates ethical AI practices by avoiding favoritism toward any single category and providing explainable recommendation logic.

Task Description #4:

• Prompt Copilot to generate logging functionality in a Python web application. Then, ask it to ensure the logs do not record sensitive information.

Expected Output #4:

• Logging code that avoids saving personal identifiers (e.g., passwords, emails), and includes comments about ethical logging practices

Prompt:

Generate Python logging functionality for a web application.
Ensure that the logs follow ethical logging practices by avoiding the storage of sensitive personal information such as passwords and email addresses. Add inline comments explaining how sensitive data is filtered before logging.

Code:

Observation:

The generated logging code configures a logging system and records user actions while excluding sensitive information such as passwords and email addresses. Inline comments explain the ethical practice of filtering personal identifiers before logging. This approach ensures user privacy, prevents data leakage, and follows responsible logging and security guidelines.

Task Description #5:

• Ask Copilot to generate a machine learning model. Then, prompt it to add documentation on how to use the model responsibly (e.g., explainability, accuracy limits).

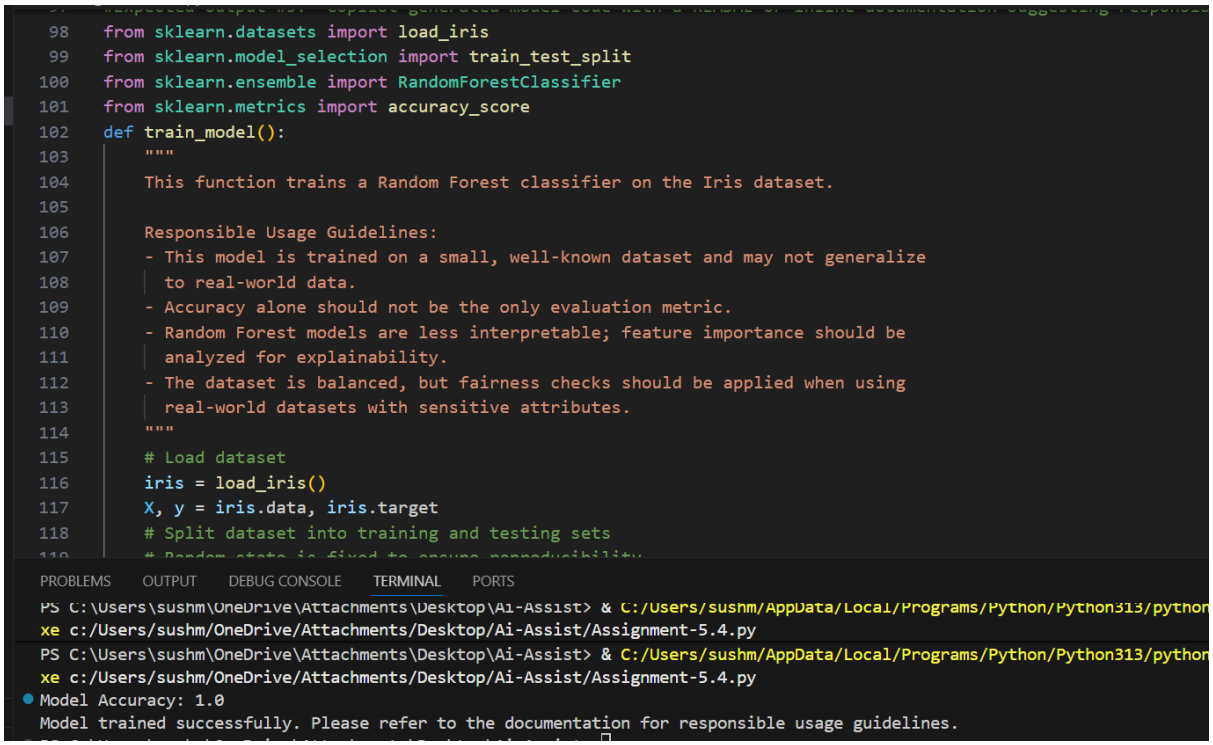
Expected Output #5:

• Copilot-generated model code with a README or inline documentation suggesting responsible usage, limitations, and fairness considerations.

Prompt:

Generate a Python machine learning model using a standard dataset.
Add inline documentation explaining how to use the model responsibly, including limitations of accuracy, explainability concerns, and fairness considerations when applying the model to real-world data.

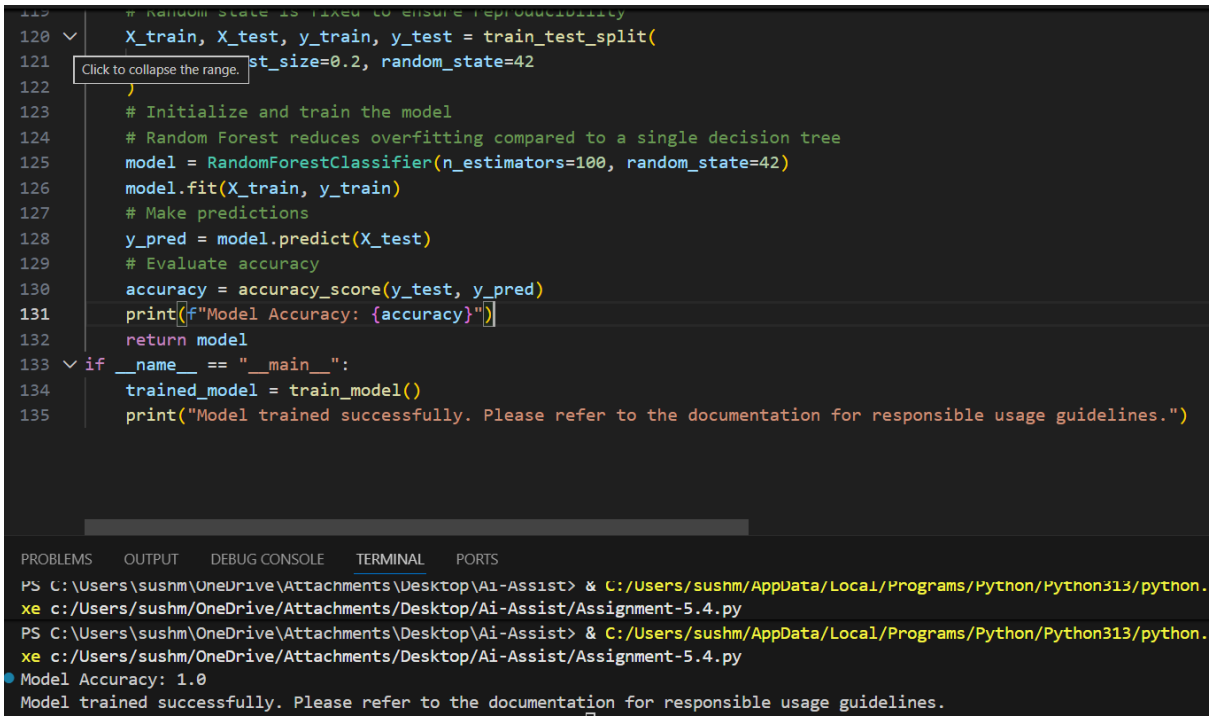
Code:



```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
def train_model():
    """
    This function trains a Random Forest classifier on the Iris dataset.

    Responsible Usage Guidelines:
    - This model is trained on a small, well-known dataset and may not generalize
      to real-world data.
    - Accuracy alone should not be the only evaluation metric.
    - Random Forest models are less interpretable; feature importance should be
      analyzed for explainability.
    - The dataset is balanced, but fairness checks should be applied when using
      real-world datasets with sensitive attributes.
    """
    # Load dataset
    iris = load_iris()
    X, y = iris.data, iris.target
    # Split dataset into training and testing sets
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\sushm\OneDrive\Attachments\Desktop\Ai-Assist> & C:/Users/sushm/AppData/Local/Programs/Python/Python313/python
xe c:/Users/sushm/OneDrive/Attachments/Desktop/Ai-Assist/Assignment-5.4.py
PS C:\Users\sushm\OneDrive\Attachments\Desktop\Ai-Assist> & C:/Users/sushm/AppData/Local/Programs/Python/Python313/python
xe c:/Users/sushm/OneDrive/Attachments/Desktop/Ai-Assist/Assignment-5.4.py
Model Accuracy: 1.0
Model trained successfully. Please refer to the documentation for responsible usage guidelines.
```



```
    X_train, X_test, y_train, y_test = train_test_split(
        st_size=0.2, random_state=42
    )
    # Initialize and train the model
    # Random Forest reduces overfitting compared to a single decision tree
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    # Make predictions
    y_pred = model.predict(X_test)
    # Evaluate accuracy
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Model Accuracy: {accuracy}")
    return model
if __name__ == "__main__":
    trained_model = train_model()
    print("Model trained successfully. Please refer to the documentation for responsible usage guidelines.")
```

Click to collapse the range.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\sushm\OneDrive\Attachments\Desktop\Ai-Assist> & C:/Users/sushm/AppData/Local/Programs/Python/Python313/python.
xe c:/Users/sushm/OneDrive/Attachments/Desktop/Ai-Assist/Assignment-5.4.py
PS C:\Users\sushm\OneDrive\Attachments\Desktop\Ai-Assist> & C:/Users/sushm/AppData/Local/Programs/Python/Python313/python.
xe c:/Users/sushm/OneDrive/Attachments/Desktop/Ai-Assist/Assignment-5.4.py
Model Accuracy: 1.0
Model trained successfully. Please refer to the documentation for responsible usage guidelines.
```

Observation:

The generated code trains a Random Forest classifier on the Iris dataset and includes detailed inline documentation describing responsible usage guidelines. The comments highlight limitations such as restricted generalization, reliance on accuracy as a single metric, reduced interpretability of ensemble models, and the importance of fairness checks when handling real-world datasets. This demonstrates responsible and ethical use of machine learning models.