Deploying Data processing and Sentiment Analysis Model

- To deploy the dataProcessing.py and SentimentAnalysis model, you can create a prediction module, similar to your existing crime.py, that imports and uses these modules. Here's an example of how you can structure the prediction module and deploy the sentiment analysis and data processing models:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sentiment_analysis import SentimentAnalysis  # Import your SentimentAnalysis class
from dataProcessing import DataProcessor  # Import your DataProcessor class

# Initialize the SentimentAnalysis and DataProcessor classes
sentiment_analyzer = SentimentAnalysis()
data_processor = DataProcessor()
```

```
# Function to get suspect information and calculate scores
def calculate_scores(name, age, gender, question1, question2, question3):
    obedient_result = sentiment_analyzer.is_obedient(question1, name, age, gender)
    emotion_score = sentiment_analyzer.calculate_emotion_score(question2)
    consistency_score = sentiment_analyzer.calculate_consistency_score(question2, question3)
    confidence_score = sentiment_analyzer.calculate_confidence_score(emotion_score, consistency_score)
    return obedient_result, emotion_score, consistency_score, confidence_score

# Example usage:
name = "John"
age = 30
gender = "Male"

question1 = "I am very obedient."
question2 = "I refuse to cooperate."
question3 = "I saw the crime happen."

# Calculate scores for the suspect
```

```
obedient_result, emotion_score, consistency_score,
confidence_score = calculate_scores(name, age, gender,
question1, question2, question3)


# Use the DataProcessor to prepare the data for prediction

input_data = data_processor.prepare_data(name, age, gender,
obedient_result, emotion_score, consistency_score)


# Load the trained machine learning model (e.g., from
'crime.py')

model = load_trained_model()  # Implement a function to load
your trained model


# Make a prediction using the loaded model

prediction = model.predict(input_data)


# Print the result (Y for criminal, N for not a criminal)

if prediction == 'Y':
    print(f"{name} is a criminal.")
else:
    print(f"{name} is not a criminal.")
```

- In this code, we import the SentimentAnalysis and DataProcessor classes and initialize them. We then create a function to calculate scores for a suspect based

on their responses to questions. After calculating the scores, we use the DataProcessor to prepare the data for prediction and load the trained machine learning model.

❖ You should replace the example usage with your actual data and machine learning model. Additionally, implement a function to load your trained machine learning model (similar to what you did in crime.py).