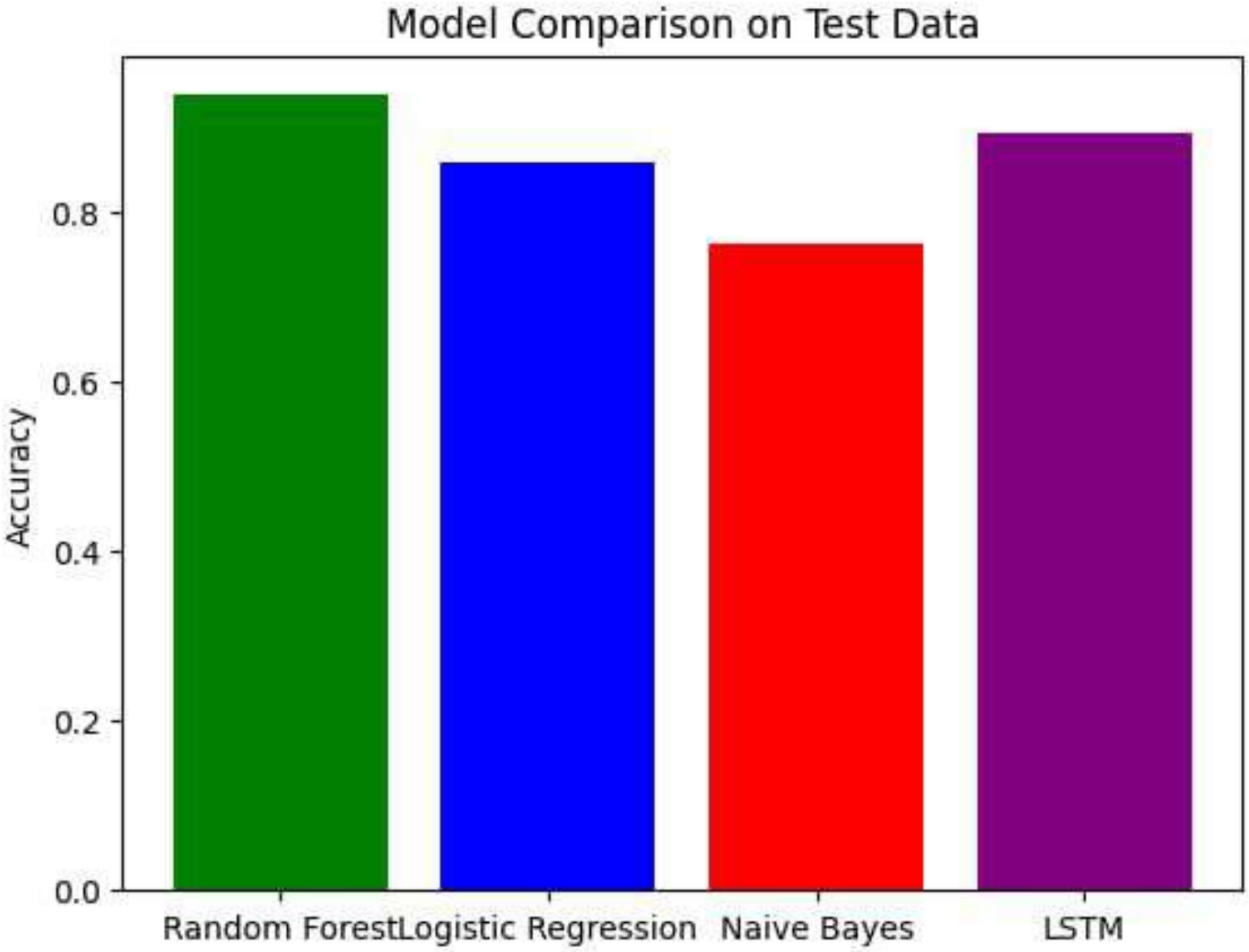
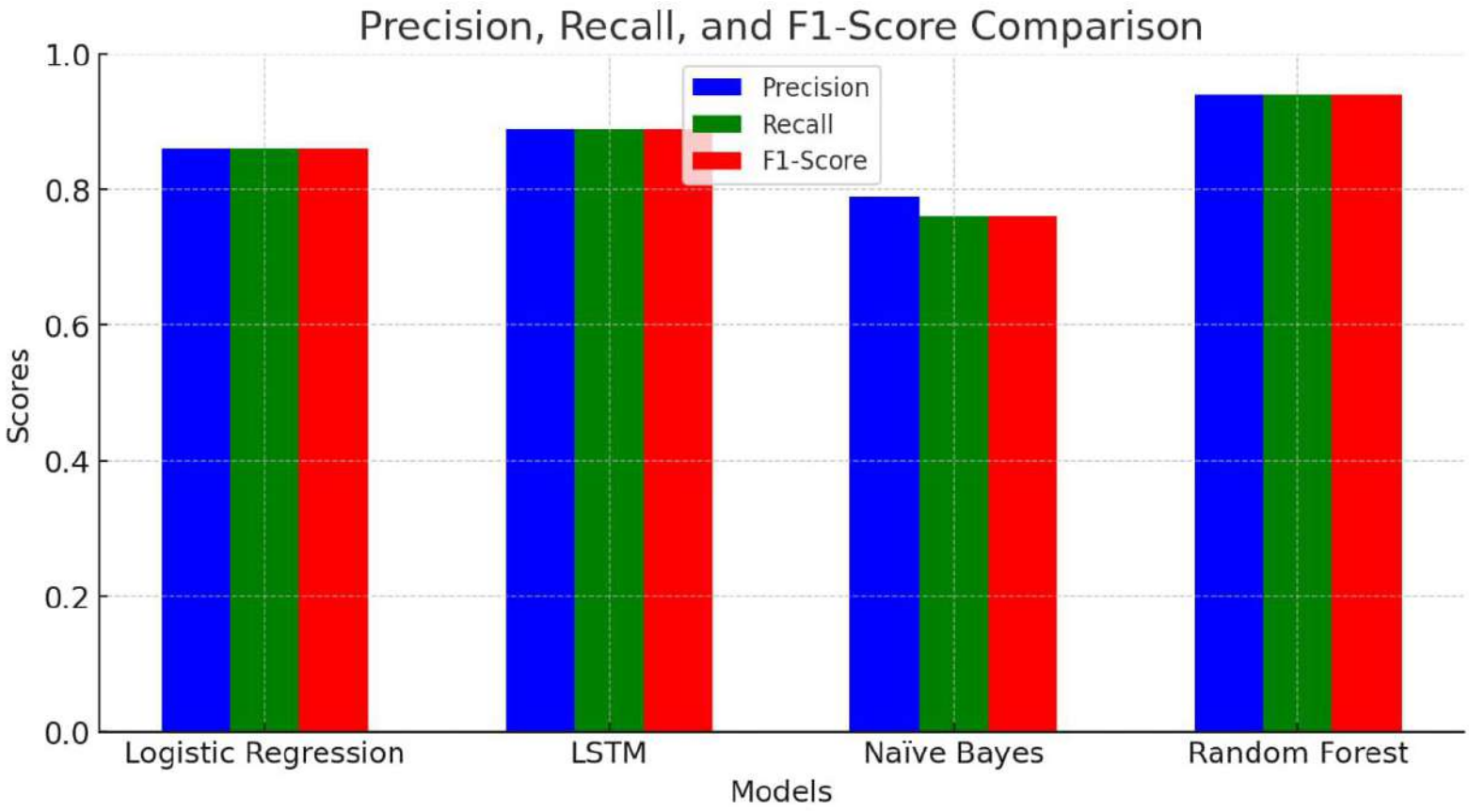
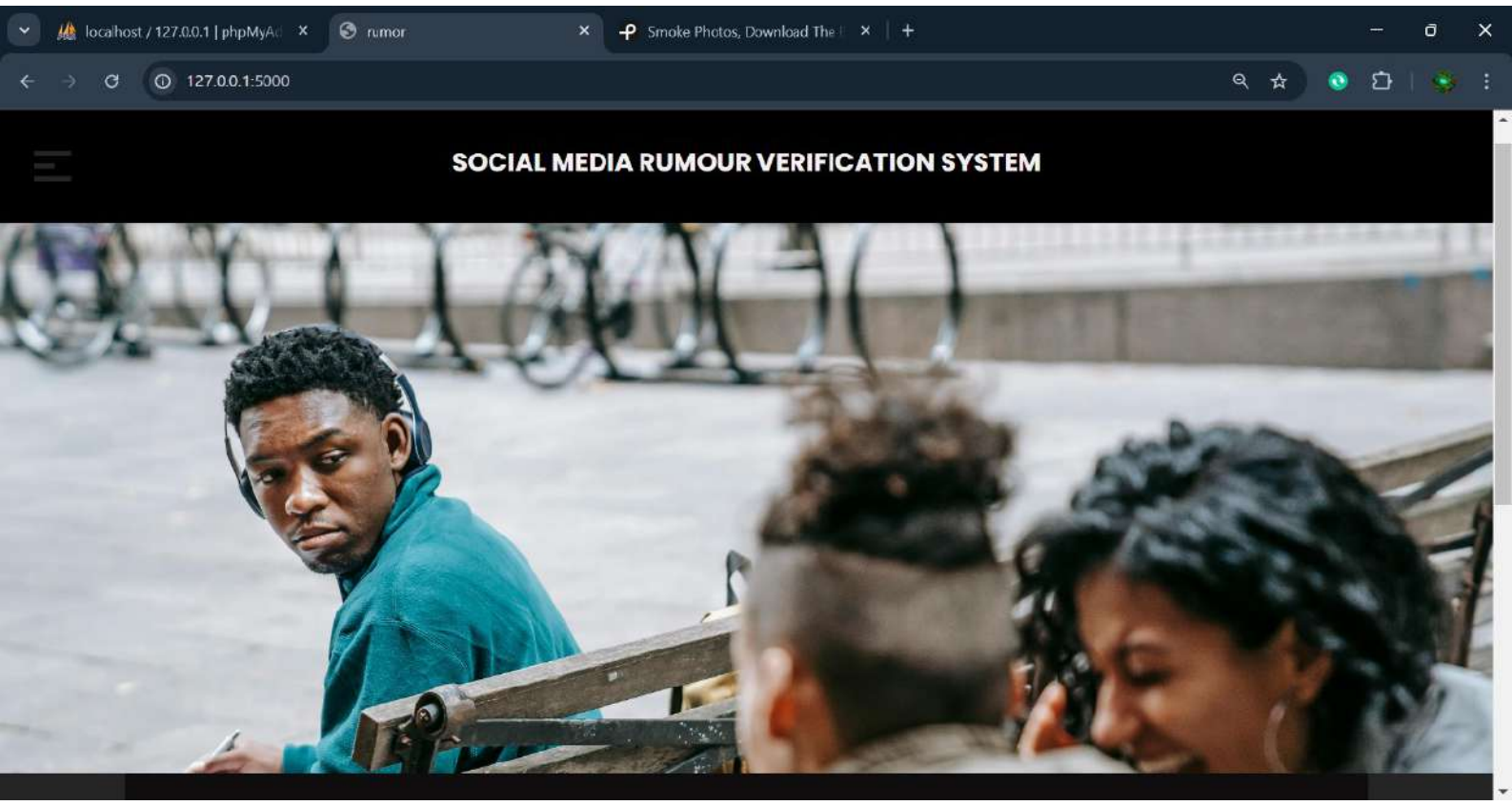
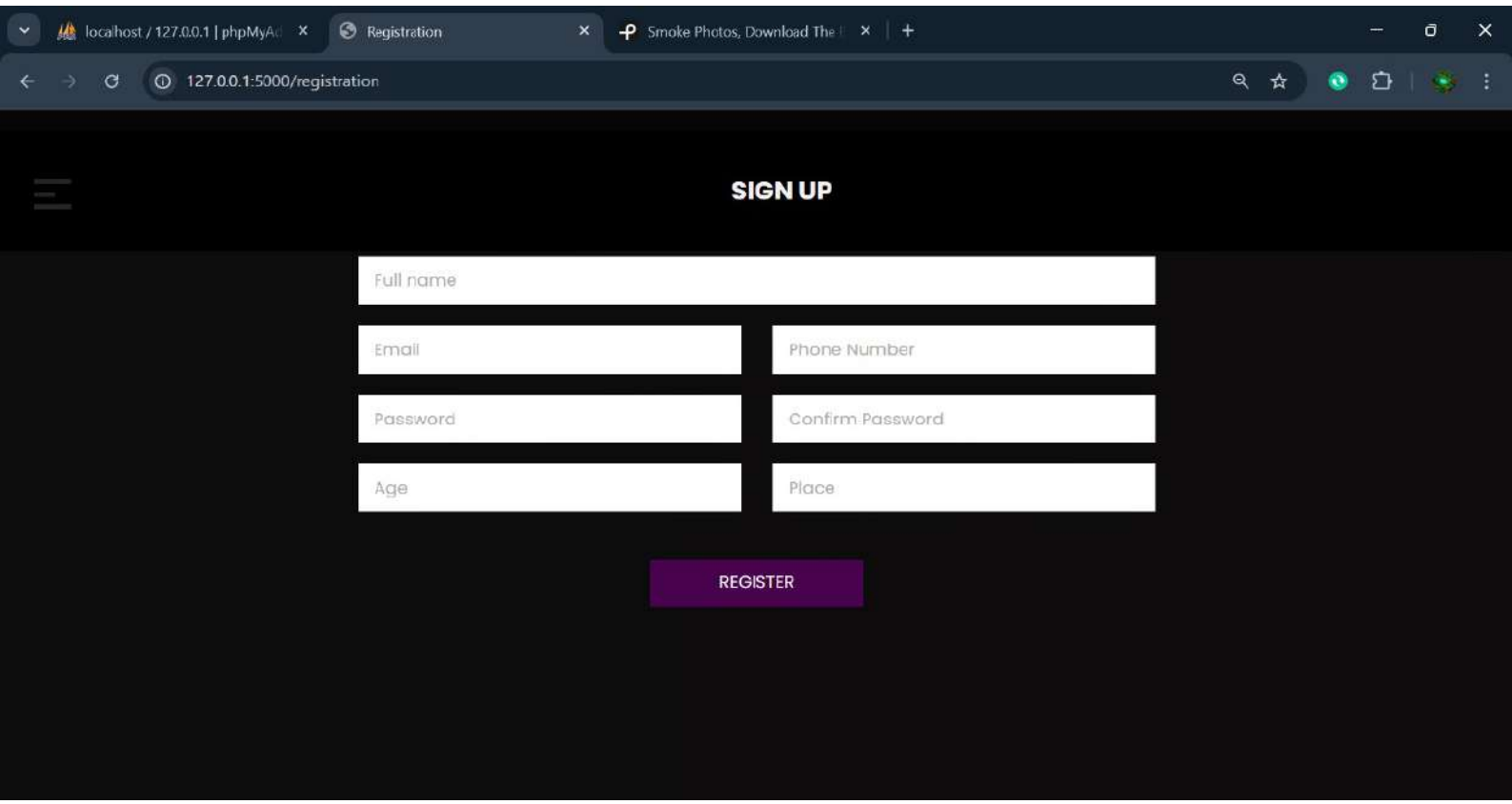

Model Comparison on Test Data
Accuracy
0.0
0.2
0.4
0.6
0.8
Random Forest
Logistic Regression
Naive Bayes
LSTM


Precision, Recall, and F1-Score Comparison
Precision
Recall
F1-Score
Scores
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
LSTM
Naïve Bayes
Random Forest
Models


localhost / 127.0.0.1 | phpMyAd
rumor
Smoke Photos, Download The
127.0.0.1:5000
SOCIAL MEDIA RUMOUR VERIFICATION SYSTEM


localhost / 127.0.0.1 | phpMyAd
Registration
Smoke Photos, Download The
127.0.0.1:5000/registration
SIGN UP
Full name
Email
Phone Number
Password
Confirm Password
Age
Place
REGISTER

File Edit Selection View Go Run Terminal Help rumour1.ipynb - CODE - Visual Studio Code

rumour1.ipynb

BACKEND > rumour1.ipynb > ...

+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3.12.6

Curve: The orange curve represents the ROC curve for this classifier. The area under the curve (AUC) is 0.92, indicating good performance but not as high as the first model.

Diagonal Line: The dashed diagonal line represents the line of no discrimination (AUC = 0.5).

```
# Print the classification report
logistic = classification_report(y_test, log_reg_preds)
print("Logistic Regression Classification Report:\n", logistic)
```

[47] Python

```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.86      0.86      9718
           1       0.86      0.86      0.86      9730

    accuracy                           0.86     19448
   macro avg       0.86      0.86      0.86     19448
weighted avg       0.86      0.86      0.86     19448
```

Naive_bayes

0 20 0 Spaces: 4 Cell 36 of 61 Go Live

Search ENG IN 14:45 25-02-2025

File Edit Selection View Go Run Terminal Help rumour1.ipynb - CODE - Visual Studio Code

rumour1.ipynb

BACKEND > rumour1.ipynb > ...

+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3.12.6

Diagonal Line: The dashed diagonal line represents the line of no discrimination, where the classifier performs no better than random chance (AUC = 0.5).

[43]
```python
# Print the classification report
rf = classification_report(y_test, rf_preds)
print("Random Forest Classification Report:\n", rf)
```
Python

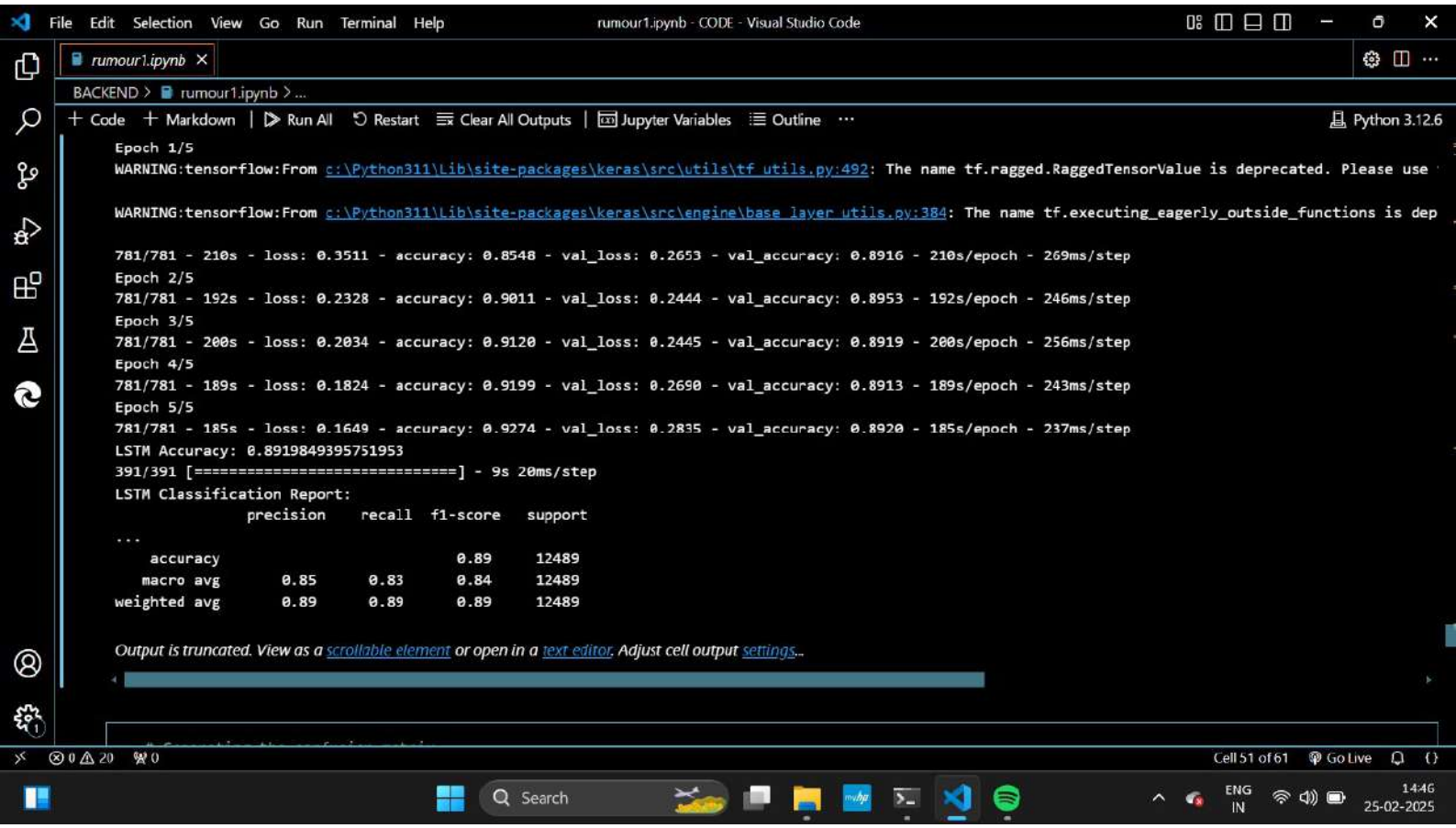
```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      9718
           1       0.93      0.95      0.94      9730

    accuracy                           0.94     19448
   macro avg       0.94      0.94      0.94     19448
weighted avg       0.94      0.94      0.94     19448
```

LogisticRegression

[44]
```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

0 ⚠ 20 0 Spaces: 4 Cell 36 of 61 Go Live

Search ENG IN 14:44 25-02-2025


File Edit Selection View Go Run Terminal Help
rumour1.ipynb - CODE - Visual Studio Code
rumour1.ipynb
BACKEND > rumour1.ipynb > ...
+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ...
Python 3.12.6
Epoch 1/5
WARNING:tensorflow:From c:\Python311\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use
WARNING:tensorflow:From c:\Python311\Lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functions is dep
781/781 - 210s - loss: 0.3511 - accuracy: 0.8548 - val_loss: 0.2653 - val_accuracy: 0.8916 - 210s/epoch - 269ms/step
Epoch 2/5
781/781 - 192s - loss: 0.2328 - accuracy: 0.9011 - val_loss: 0.2444 - val_accuracy: 0.8953 - 192s/epoch - 246ms/step
Epoch 3/5
781/781 - 200s - loss: 0.2034 - accuracy: 0.9120 - val_loss: 0.2445 - val_accuracy: 0.8919 - 200s/epoch - 256ms/step
Epoch 4/5
781/781 - 189s - loss: 0.1824 - accuracy: 0.9199 - val_loss: 0.2690 - val_accuracy: 0.8913 - 189s/epoch - 243ms/step
Epoch 5/5
781/781 - 185s - loss: 0.1649 - accuracy: 0.9274 - val_loss: 0.2835 - val_accuracy: 0.8920 - 185s/epoch - 237ms/step
LSTM Accuracy: 0.8919849395751953
391/391 [==============================] - 9s 20ms/step
LSTM Classification Report:
precision recall f1-score support
...
accuracy 0.89 12489
macro avg 0.85 0.83 0.84 12489
weighted avg 0.89 0.89 0.89 12489
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
Cell 51 of 61
Go Live
Search
ENG
IN
14:46
25-02-2025

File Edit Selection View Go Run Terminal Help rumour1.ipynb - CODE - Visual Studio Code

rumour1.ipynb

BACKEND > rumour1.ipynb > ...

+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3.12.6

+ Code + Markdown

```
# Print the classification report
naive_bayes = classification_report(y_test, naive_bayes_preds)
print("Naive Bayes Classification Report:\n",naive_bayes )
```

[49] Python

```
Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.62      0.72      9718
           1       0.70      0.91      0.79      9730

    accuracy                           0.76     19448
   macro avg       0.79      0.76      0.76     19448
weighted avg       0.79      0.76      0.76     19448
```

```
#Generate confusion matrix
cm = confusion_matrix(y_test, naive_bayes_preds)
print("Confusion Matrix:")
print(cm)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
```

[50]

0 20 0 Spaces: 4 Cell 36 of 61 Go Live

Search ENG IN 14:45 25-02-2025