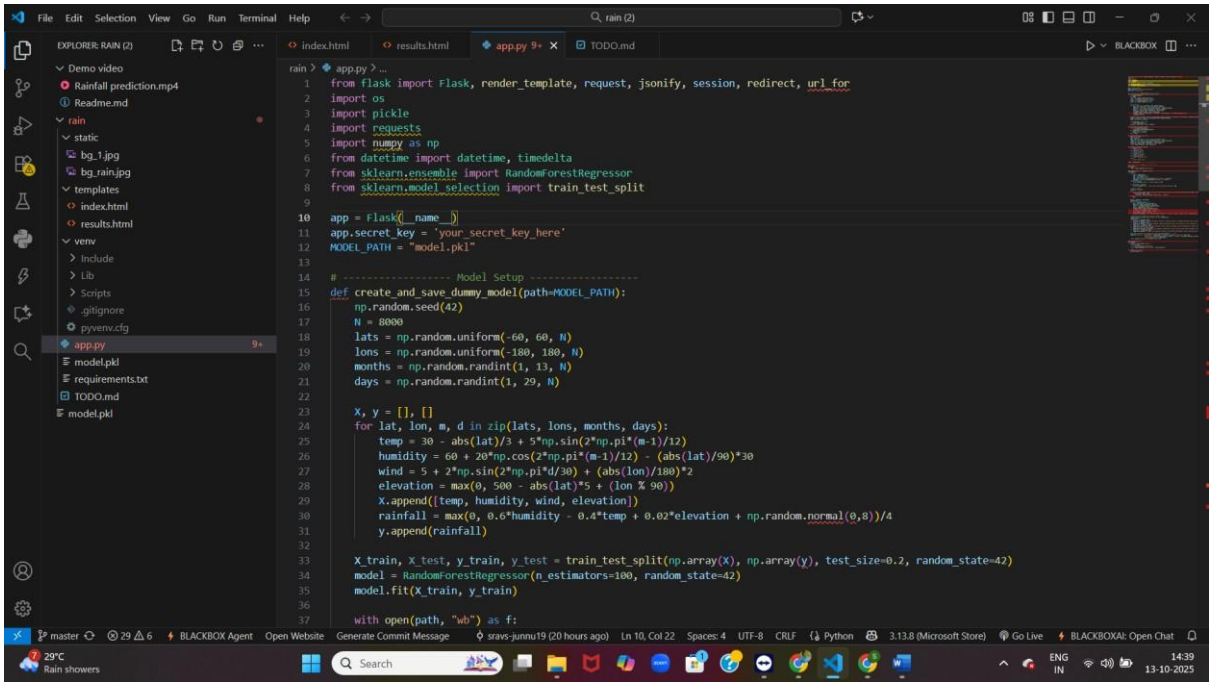
Date	3 oct 2025
Team ID	LTVIP2025TMIDS67798
Project Name	RAINFALL PREDICTION USING MACHINE LEARNING
Maximum Marks	10 Marks

1. Initial Model Training Code

This template demonstrates how to train a machine learning model for rainfall prediction using Python's scikit-learn library.



2. Model Validation and Evaluation Template

After training your model, it's crucial to validate and evaluate its performance. Here's how you can do it:

a. Cross-Validation

Implementing cross-validation helps in assessing the model's performance across different subsets of the dataset.

```
from sklearn.model_selection import cross_val_score

cv_scores = cross_val_score(model, X, y, cv=5, scoring='neg_mean_absolute_error')
print(f'Cross-validated MAE: {-cv_scores.mean()}')
```

b. Hyperparameter Tuning

Fine-tuning the model's hyperparameters can enhance its performance.

```
from sklearn.model_selection import GridSearchCV

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=3, n_jobs=-1, verbose=2
grid_search.fit(X_train, y_train)

print(f'Best Parameters: {grid_search.best_params_}')
```

c. Model Evaluation Metrics

Utilize various metrics to evaluate your model's performance comprehensively.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

y_pred = grid_search.best_estimator_.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error: {mae}')
print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

d. Residual Analysis

Analyzing residuals helps in understanding the model's prediction errors.

```
residuals = y_test - y_pred
plt.scatter(y_pred, residuals)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel('Predicted Rainfall')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.show()
```

3. Model Deployment and Monitoring

Once your model is trained and evaluated, consider the following steps for deployment and monitoring:

- **Deployment**: Use frameworks like Flask or FastAPI to create a web service for your model.
- **Monitoring**: Implement logging and monitoring to track the model's performance over time.
- **Retraining**: Periodically retrain the model with new data to maintain its accurac