SMARTBRIDGE Let's Bridge the Gap a Veranda Enterprise

Smart Internz

Model Optimization and Tuning Phase Template

Date	18 June 2025
Team ID	SWTID1749618778
Project Title	Rising Waters: A Machine Learning Approach To Flood Prediction
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Decision Tree	max_depth, min_samples_split, criterion	5, 4, 'gini'
Random Forest	n_estimators, max_depth, max_features	100, 6, 'sqrt'
KNN	n_neighbors, weights, metric	5, 'uniform', 'minkowski'
Logistic Regression	penalty, C, solver, max_iter	'l2', 1.0, 'lbfgs', 100

Performance Metrics Comparison Report (2 Marks):

Model	**Optimized Metric**

Decision Tree	```print(classification_report(y_test,p1))```
	```print(confusion_matrix(y_test,p1))```
	
	``` precision recall f1-score support```
	``` 0 1.00 1.00 1.00 20```
	``` 1 1.00 1.00 1.00 3```
	``` accuracy 1.00 23```
	``` macro avg 1.00 1.00 1.00 23```
	```weighted avg 1.00 1.00 1.00 23```
	
	```[[20 0]```
	``` [0 3]]```

Random Forest	```print(classification_report(y_test,p2))```
	```print(confusion_matrix(y_test,p2))```
	
	``` precision recall f1-score support```
	``` 0 1.00 1.00 1.00 20```
	``` 1 1.00 1.00 1.00 3```
	``` accuracy 1.00 23```
	``` macro avg 1.00 1.00 1.00 23```
	```weighted avg 1.00 1.00 1.00 23```
	
	```[[20 0]```
	``` [0 3]]```

<table>
<tr><td>KNN</td><td>

```
print(classification_report(y_test,p3))
print(confusion_matrix(y_test,p3))
```

```
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        20
           1       1.00      0.33      0.50         3

    accuracy                           0.91        23
   macro avg       0.95      0.67      0.73        23
weighted avg       0.92      0.91      0.89        23
```

```
[[20  0]
 [ 2  1]]
```

</td></tr>
<tr><td>Logistic Regression</td><td>

```
y_pred = pipeline.predict(X_test)
print("📊 Classification Report:\n")
print(classification_report(y_test, y_pred))

print("🧩 Confusion Matrix:\n")
print(confusion_matrix(y_test, y_pred))


with open('../models/logreg_pipeline.pkl', 'wb') as f:
    pickle.dump(pipeline, f)
```

```
📊 Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        20
           1       1.00      1.00      1.00         3

    accuracy                           1.00        23
   macro avg       1.00      1.00      1.00        23
weighted avg       1.00      1.00      1.00        23

🧩 Confusion Matrix:

[[20  0]
 [ 0  3]]
```

</td></tr>
</table>

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Logistic Regression	We selected Logistic Regression as the final model for flood prediction due to its simplicity, interpretability, and reliable performance. Unlike more complex models that risk overfitting, Logistic Regression provided consistent, realistic accuracy while offering clear insights into how each feature (such as rainfall, humidity, and temperature) impacts flood risk. Its ability to handle binary classification tasks effectively, combined with its low computational cost, made it the most suitable and practical choice for our flood prediction system.