Model Optimization and Tuning Phase Template

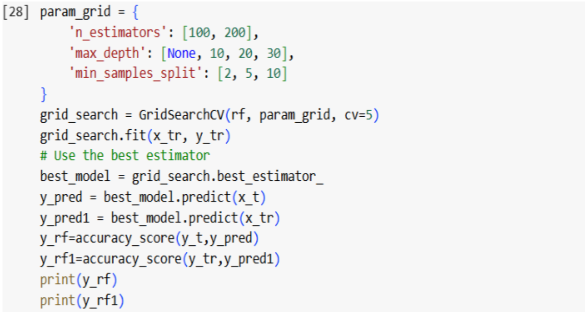
Date	3 October 2024
Team ID	LTVIP2024TMID24876
Project Title	Rising Waters: A Machine Learning Approach to Flood Prediction
Maximum Marks	10 Marks

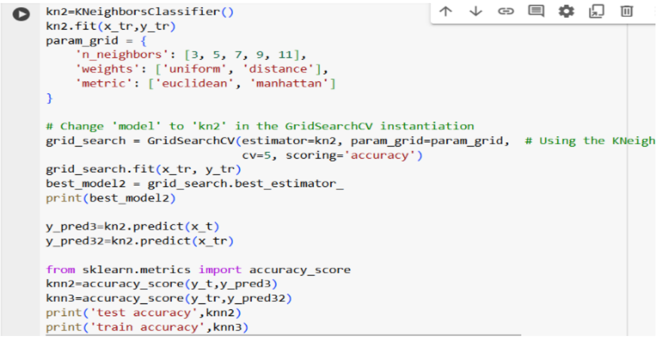
Model Optimization and Tuning Phase

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):

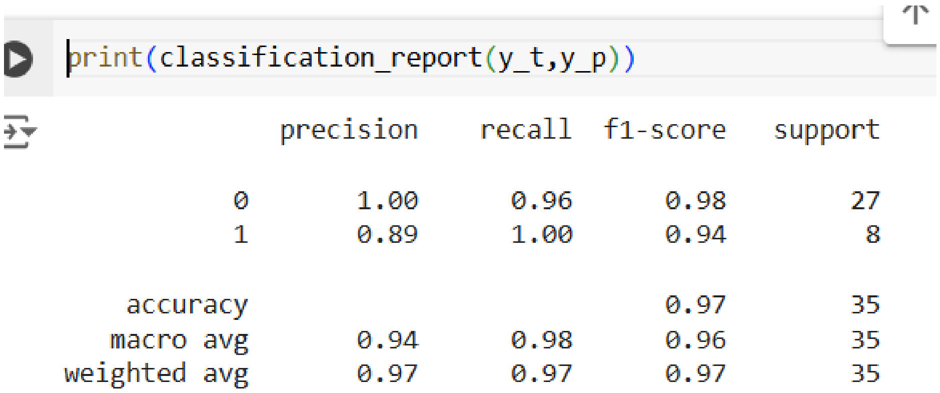
Model	Tuned Hyperparameters	Optimal Values

Random Forest	
	[28] param_grid = {
	'n_estimators': [100, 200],
	'max_depth': [None, 10, 20, 30],
	'min_samples_split': [2, 5, 10]
	}
	grid_search = GridSearchCV(rf, param_grid, cv=5)
	grid_search.fit(x_tr, y_tr)
	# Use the best estimator
	best_model = grid_search.best_estimator_
	y_pred = best_model.predict(x_t)
	y_pred1 = best_model.predict(x_tr)
	y_rf=accuracy_score(y_t,y_pred)
	y_rf1=accuracy_score(y_tr,y_pred1)
	print(y_rf)
	print(y_rf1)
		0.9714285714285714
		1.0

KNN	
	kn2=KNeighborsClassifier()
	kn2.fit(x_tr,y_tr)
	param_grid = {
	'n_neighbors': [3, 5, 7, 9, 11],
	'weights': ['uniform', 'distance'],
	'metric': ['euclidean', 'manhattan']
	}
	
	# Change 'model' to 'kn2' in the GridSearchCV instantiation
	grid_search = GridSearchCV(estimator=kn2, param_grid=param_grid, # Using the KNeigh
	cv=5, scoring='accuracy')
	grid_search.fit(x_tr, y_tr)
	best_model2 = grid_search.best_estimator_
	print(best_model2)
	
	y_pred3=kn2.predict(x_t)
	y_pred32=kn2.predict(x_tr)
	
	from sklearn.metrics import accuracy_score
	knn2=accuracy_score(y_t,y_pred3)
	knn3=accuracy_score(y_tr,y_pred32)
	print('test accuracy',knn2)
	print('train accuracy',knn3)
		KNeighborsClassifier(metric='manhattan', n_neighbors=3, weights='distance')
		test accuracy 0.9428571428571428
		train accuracy 0.95

Performance Metrics Comparison Report (2 Marks):

<table>
<tr><th>Model</th><th>Baseline Metric Optimized Metric</th></tr>
<tr><td>Random Forest</td><td>



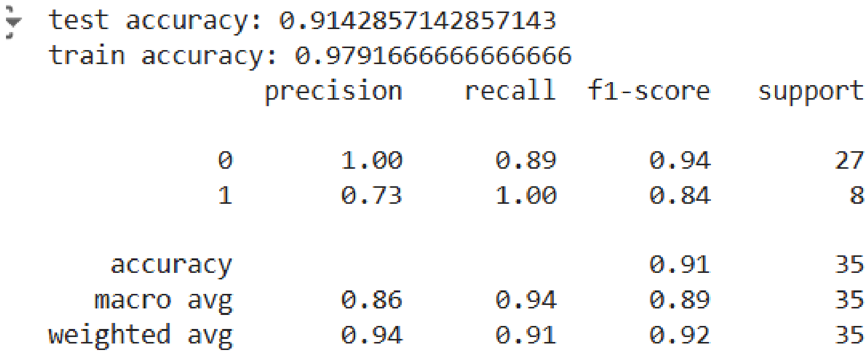
```
print(classification_report(y_t,y_p))
```

```
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        27
           1       0.89      1.00      0.94         8

    accuracy                           0.97        35
   macro avg       0.94      0.98      0.96        35
weighted avg       0.97      0.97      0.97        35
```

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



```
test accuracy: 0.9142857142857143
train accuracy: 0.9791666666666666
              precision    recall  f1-score   support

           0       1.00      0.89      0.94        27
           1       0.73      1.00      0.84         8

    accuracy                           0.91        35
   macro avg       0.86      0.94      0.89        35
weighted avg       0.94      0.91      0.92        35
```

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

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
KNN	• According to the above data the KNN has been selected for prediction. • Random Forest was not selected because the training data accuracy is 1.0 which overfits the model. **Final Model: KNN**