Model Optimization and Tuning Phase Template

Date	15 March 2024
Team ID	740003
Project Title	Disease Prediction Using Machine Learning
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
KNN	---	---
SVC	---	---
Decision Tree	---	---
Random Forest	---	---

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric

KNN

```
[75] knn=KNeighborsClassifier()
     knn.fit(X1_train, y1_train)
     y_pred_knn = knn.predict(X1_val)
```

```
[76] y_pred = rfc.predict(X1_val)
     yt_pred = rfc.predict(X1_train)
     y_pred1 = rfc.predict(x1_test)
     print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pr
     print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pre
     print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1)
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

SVC

```
[89] from sklearn.svm import SVC
     svm1=SVC(C=1)
     svm1.fit(X1_train,y1_train)
     y_pred_svc = svm1.predict(X1_val)
     y_pred = svm1.predict(X1_val)
     yt_pred = svm1.predict(X1_train)
     y_pred1 = svm1.predict(x1_test)
     print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
     print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
     print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

Random Forest	[73] `# Train a Random Forest Classifier and calculate accuracy`
	`rfc = RandomForestClassifier(random_state=42)`
	`rfc.fit(X1_train, y1_train)`
	`y_pred_rfc = rfc.predict(X1_val)`
	
	[74] `y_pred = rfc.predict(X1_val)`
	`yt_pred = rfc.predict(X1_train)`
	`y_pred1 = rfc.predict(x1_test)`
	`print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))`
	`print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))`
	`print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))`
	
	`the Training Accuracy of the algorithm is 0.9930313588850174`
	`the Validation Accuracy of the algorithm is 0.9959349593495935`
	`the Testing Accuracy of the algorithm is 1.0`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
KNN	KNN was chosen based on its demonstrated high accuracy in capturing intricate patterns within the dataset. Its ability to directly reflect the underlying structure of the data and adapt to varying complexities without assuming specific distributions made it the optimal choice. This aligns with the task's need for robust performance across different scenarios, ensuring reliable predictions with minimal computational overhead.