

Model Optimization and Tuning Phase Template

Date	3 October 2024
Team ID	LTVIP2024TMID24838
Project Title	Detection of Phishing Websites from URLs
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	<pre>param_grid = { 'n_estimators': [50, 100, 200], 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'criterion': ['gini', 'entropy'] }</pre>	<p>RandomForestClassifier(criterion='entropy', max_depth=20, n_estimators=200, random_state=0)</p> <p>test accuracy: 0.9665309814563546 train accuracy 0.9987562189054726</p>
KNN	<pre>param_grid = { 'n_neighbors': [3, 5, 7, 9, 11], 'weights': ['uniform', 'distance'], 'metric': ['euclidean', 'manhattan'] }</pre>	<p>KNeighborsClassifier(metric='manhattan', n_neighbors=3, weights='distance')</p> <p>test accuracy 0.6142017186793306 train accuracy 0.7800768882858435</p>
Logistic Regression	<pre>param_grid = { 'C': [0.001, 0.01, 0.1, 1, 10, 100], 'penalty': ['l1', 'l2'], 'solver': ['liblinear', 'saga'] }</pre>	<p>Best parameters: {'C': 1, 'penalty': 'l1', 'solver': 'liblinear'}</p> <p>Accuracy on test set: 0.9185888738127544 Accuracy on training set: 0.9320443238353686</p>

Decision Tree Classifier	<pre> param_grid = { 'max_depth': [None, 5, 10, 15], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'criterion': ['gini', 'entropy'] } </pre>	<p>Best parameters: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}</p> <p>Accuracy on test set: 0.9574853007688828</p> <p>Accuracy on train set: 1.0</p>
--------------------------	--	--

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric Optimized Metric																														
Random Forest	<pre>print(classification_report(y_t, y_rf))</pre> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>-1</td><td>0.97</td><td>0.95</td><td>0.96</td><td>1014</td></tr><tr><td>1</td><td>0.96</td><td>0.98</td><td>0.97</td><td>1197</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.97</td><td>2211</td></tr><tr><td>macro avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>2211</td></tr><tr><td>weighted avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>2211</td></tr></table>		precision	recall	f1-score	support	-1	0.97	0.95	0.96	1014	1	0.96	0.98	0.97	1197	accuracy			0.97	2211	macro avg	0.97	0.97	0.97	2211	weighted avg	0.97	0.97	0.97	2211
	precision	recall	f1-score	support																											
-1	0.97	0.95	0.96	1014																											
1	0.96	0.98	0.97	1197																											
accuracy			0.97	2211																											
macro avg	0.97	0.97	0.97	2211																											
weighted avg	0.97	0.97	0.97	2211																											
KNN	<pre>print(classification_report(y_t, y_pred3))</pre> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>-1</td><td>0.59</td><td>0.54</td><td>0.56</td><td>1014</td></tr><tr><td>1</td><td>0.63</td><td>0.68</td><td>0.66</td><td>1197</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.61</td><td>2211</td></tr><tr><td>macro avg</td><td>0.61</td><td>0.61</td><td>0.61</td><td>2211</td></tr><tr><td>weighted avg</td><td>0.61</td><td>0.61</td><td>0.61</td><td>2211</td></tr></table>		precision	recall	f1-score	support	-1	0.59	0.54	0.56	1014	1	0.63	0.68	0.66	1197	accuracy			0.61	2211	macro avg	0.61	0.61	0.61	2211	weighted avg	0.61	0.61	0.61	2211
	precision	recall	f1-score	support																											
-1	0.59	0.54	0.56	1014																											
1	0.63	0.68	0.66	1197																											
accuracy			0.61	2211																											
macro avg	0.61	0.61	0.61	2211																											
weighted avg	0.61	0.61	0.61	2211																											
Logistic Regression	<pre>print(classification_report(y_t, y_plr21))</pre> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>-1</td><td>0.93</td><td>0.89</td><td>0.91</td><td>1014</td></tr><tr><td>1</td><td>0.91</td><td>0.94</td><td>0.93</td><td>1197</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.92</td><td>2211</td></tr><tr><td>macro avg</td><td>0.92</td><td>0.92</td><td>0.92</td><td>2211</td></tr><tr><td>weighted avg</td><td>0.92</td><td>0.92</td><td>0.92</td><td>2211</td></tr></table>		precision	recall	f1-score	support	-1	0.93	0.89	0.91	1014	1	0.91	0.94	0.93	1197	accuracy			0.92	2211	macro avg	0.92	0.92	0.92	2211	weighted avg	0.92	0.92	0.92	2211
	precision	recall	f1-score	support																											
-1	0.93	0.89	0.91	1014																											
1	0.91	0.94	0.93	1197																											
accuracy			0.92	2211																											
macro avg	0.92	0.92	0.92	2211																											
weighted avg	0.92	0.92	0.92	2211																											

Decision Tree Classifier	<pre>print(classification_report(y_t, y_pred_t))</pre>				
		precision	recall	f1-score	support
	-1	0.95	0.95	0.95	1014
	1	0.96	0.96	0.96	1197
	accuracy			0.95	2211
	macro avg	0.95	0.95	0.95	2211
	weighted avg	0.95	0.95	0.95	2211

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Logistic Regression	<ul style="list-style-type: none"> According to the above data the model knn has the least accuracy. The Decision Tree and Random Forest models both has training accuracy of (1.0) which is overfitting of the data, even after the hyperparameter tuning of the models the models show overfitting. The most suitable models seem to be the Logistic Regression among the above models. The Logistic Regression model which has 92% accuracy score.