

Model Optimization and Tuning Phase Template

Date	14 July 2024
Team ID	SWTID1720174920
Project Title	Human Resource Management: Predicting Employee Promotions 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
DecisionTree Classifier	<pre>dt = DecisionTreeClassifier(criterion='entropy',max_depth=5,min_samples_split=10, min_samples_leaf=5,random_state=42) dt.fit(x_train,y_train) y_pred_dt=dt.predict(x_test)</pre>	<pre>accuracy_score(y_test,y_pred_dt)</pre> 0.7149980055843638
RandomForest Classifier	<pre>rf=RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42) rf.fit(x_train,y_train) y_pred_rf=rf.predict(x_test)</pre>	<pre>accuracy_score(y_test,y_pred_rf)</pre> 0.7957718388512166
Kneighbours Classifier	<pre>knn=KNeighborsClassifier(n_neighbors=3, weights='uniform',algorithm='auto',leaf_size=10) knn.fit(x_train,y_train) y_pred_kn=knn.predict(x_test)</pre>	<pre>accuracy_score(y_test,y_pred_kn)</pre> 0.9032708416433984

GradientBoosting Classifier	<pre>xg=GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42) xg.fit(x_train,y_train) y_pred_xg=xg.predict(x_test)</pre>	<pre>accuracy_score(y_test,y_pred_xg)</pre> <p>0.864379736737136</p>
-----------------------------	--	--

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric																																																												
DecisionTree Classifier	<div>classification_report(y_test,y_pred)</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.95</td><td>0.93</td><td>0.94</td><td>10035</td></tr><tr><td>1</td><td>0.93</td><td>0.95</td><td>0.94</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.94</td><td>20056</td></tr><tr><td>macro avg</td><td>0.94</td><td>0.94</td><td>0.94</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.94</td><td>0.94</td><td>0.94</td><td>20056</td></tr></tbody></table> <div>confusion_matrix(y_test,y_pred)</div> <div>array([[9289, 746], [510, 9511]], dtype=int64)</div>		precision	recall	f1-score	support	0	0.95	0.93	0.94	10035	1	0.93	0.95	0.94	10021	accuracy			0.94	20056	macro avg	0.94	0.94	0.94	20056	weighted avg	0.94	0.94	0.94	20056	<div>y_pred_dt=decisionTree(x_train,x_test,y_train,y_test)</div> <div>DecisionTreeClassifier Confusion matrix [[6474 3561] [2155 7866]]</div> <div>Classification report</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.75</td><td>0.65</td><td>0.69</td><td>10035</td></tr><tr><td>1</td><td>0.69</td><td>0.78</td><td>0.73</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.71</td><td>20056</td></tr><tr><td>macro avg</td><td>0.72</td><td>0.72</td><td>0.71</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.72</td><td>0.71</td><td>0.71</td><td>20056</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.75	0.65	0.69	10035	1	0.69	0.78	0.73	10021	accuracy			0.71	20056	macro avg	0.72	0.72	0.71	20056	weighted avg	0.72	0.71	0.71	20056
		precision	recall	f1-score	support																																																									
	0	0.95	0.93	0.94	10035																																																									
	1	0.93	0.95	0.94	10021																																																									
	accuracy			0.94	20056																																																									
macro avg	0.94	0.94	0.94	20056																																																										
weighted avg	0.94	0.94	0.94	20056																																																										
	precision	recall	f1-score	support																																																										
0	0.75	0.65	0.69	10035																																																										
1	0.69	0.78	0.73	10021																																																										
accuracy			0.71	20056																																																										
macro avg	0.72	0.72	0.71	20056																																																										
weighted avg	0.72	0.71	0.71	20056																																																										
RandomForest Classifier	<div>classification_report(y_test,y_pred)</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.95</td><td>0.95</td><td>0.95</td><td>10035</td></tr><tr><td>1</td><td>0.95</td><td>0.95</td><td>0.95</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.95</td><td>20056</td></tr><tr><td>macro avg</td><td>0.95</td><td>0.95</td><td>0.95</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.95</td><td>0.95</td><td>0.95</td><td>20056</td></tr></tbody></table> <div>confusion_matrix(y_test,y_pred)</div> <div>array([[9498, 537], [477, 9544]], dtype=int64)</div>		precision	recall	f1-score	support	0	0.95	0.95	0.95	10035	1	0.95	0.95	0.95	10021	accuracy			0.95	20056	macro avg	0.95	0.95	0.95	20056	weighted avg	0.95	0.95	0.95	20056	<div>y_pred_rf=randomForest(x_train,x_test,y_train,y_test)</div> <div>RandomForestClassifier Confusion matrix [[7317 2718] [1378 8643]]</div> <div>Classification report</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.84</td><td>0.73</td><td>0.78</td><td>10035</td></tr><tr><td>1</td><td>0.76</td><td>0.86</td><td>0.81</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.80</td><td>20056</td></tr><tr><td>macro avg</td><td>0.80</td><td>0.80</td><td>0.79</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.80</td><td>0.80</td><td>0.79</td><td>20056</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.84	0.73	0.78	10035	1	0.76	0.86	0.81	10021	accuracy			0.80	20056	macro avg	0.80	0.80	0.79	20056	weighted avg	0.80	0.80	0.79	20056
		precision	recall	f1-score	support																																																									
	0	0.95	0.95	0.95	10035																																																									
	1	0.95	0.95	0.95	10021																																																									
	accuracy			0.95	20056																																																									
macro avg	0.95	0.95	0.95	20056																																																										
weighted avg	0.95	0.95	0.95	20056																																																										
	precision	recall	f1-score	support																																																										
0	0.84	0.73	0.78	10035																																																										
1	0.76	0.86	0.81	10021																																																										
accuracy			0.80	20056																																																										
macro avg	0.80	0.80	0.79	20056																																																										
weighted avg	0.80	0.80	0.79	20056																																																										

Kneighbours Classifier	<pre>classification_report(y_test,y_pred)</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.96</td><td>0.82</td><td>0.89</td><td>10035</td></tr><tr><td>1</td><td>0.84</td><td>0.97</td><td>0.90</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.90</td><td>20056</td></tr><tr><td>macro avg</td><td>0.90</td><td>0.90</td><td>0.89</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.90</td><td>0.90</td><td>0.89</td><td>20056</td></tr></tbody></table> <pre>confusion_matrix(y_test,y_pred)</pre> <pre>array([[8242, 1793], [308, 9713]], dtype=int64)</pre>		precision	recall	f1-score	support	0	0.96	0.82	0.89	10035	1	0.84	0.97	0.90	10021	accuracy			0.90	20056	macro avg	0.90	0.90	0.89	20056	weighted avg	0.90	0.90	0.89	20056	<pre>y_pred_kn=KNN(x_train,x_test,y_train,y_test)</pre> <p>Confusion matrix</p> <pre>[[8451 1584] [356 9665]]</pre> <p>Classification report</p> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.96</td><td>0.84</td><td>0.90</td><td>10035</td></tr><tr><td>1</td><td>0.86</td><td>0.96</td><td>0.91</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.90</td><td>20056</td></tr><tr><td>macro avg</td><td>0.91</td><td>0.90</td><td>0.90</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.91</td><td>0.90</td><td>0.90</td><td>20056</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.96	0.84	0.90	10035	1	0.86	0.96	0.91	10021	accuracy			0.90	20056	macro avg	0.91	0.90	0.90	20056	weighted avg	0.91	0.90	0.90	20056
		precision	recall	f1-score	support																																																									
0	0.96	0.82	0.89	10035																																																										
1	0.84	0.97	0.90	10021																																																										
accuracy			0.90	20056																																																										
macro avg	0.90	0.90	0.89	20056																																																										
weighted avg	0.90	0.90	0.89	20056																																																										
	precision	recall	f1-score	support																																																										
0	0.96	0.84	0.90	10035																																																										
1	0.86	0.96	0.91	10021																																																										
accuracy			0.90	20056																																																										
macro avg	0.91	0.90	0.90	20056																																																										
weighted avg	0.91	0.90	0.90	20056																																																										
GradientBoosting Classifier	<pre>classification_report(y_test,y_pred)</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.88</td><td>0.84</td><td>0.86</td><td>10035</td></tr><tr><td>1</td><td>0.85</td><td>0.89</td><td>0.87</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>20056</td></tr><tr><td>macro avg</td><td>0.87</td><td>0.86</td><td>0.86</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.87</td><td>0.86</td><td>0.86</td><td>20056</td></tr></tbody></table> <pre>confusion_matrix(y_test,y_pred)</pre> <pre>array([[8409, 1626], [1094, 8927]], dtype=int64)</pre>		precision	recall	f1-score	support	0	0.88	0.84	0.86	10035	1	0.85	0.89	0.87	10021	accuracy			0.86	20056	macro avg	0.87	0.86	0.86	20056	weighted avg	0.87	0.86	0.86	20056	<pre>y_pred_xg=xgboost(x_train,x_test,y_train,y_test)</pre> <p>Confusion matrix</p> <pre>[[8409 1626] [1094 8927]]</pre> <p>Classification report</p> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.88</td><td>0.84</td><td>0.86</td><td>10035</td></tr><tr><td>1</td><td>0.85</td><td>0.89</td><td>0.87</td><td>10021</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>20056</td></tr><tr><td>macro avg</td><td>0.87</td><td>0.86</td><td>0.86</td><td>20056</td></tr><tr><td>weighted avg</td><td>0.87</td><td>0.86</td><td>0.86</td><td>20056</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.88	0.84	0.86	10035	1	0.85	0.89	0.87	10021	accuracy			0.86	20056	macro avg	0.87	0.86	0.86	20056	weighted avg	0.87	0.86	0.86	20056
	precision	recall	f1-score	support																																																										
0	0.88	0.84	0.86	10035																																																										
1	0.85	0.89	0.87	10021																																																										
accuracy			0.86	20056																																																										
macro avg	0.87	0.86	0.86	20056																																																										
weighted avg	0.87	0.86	0.86	20056																																																										
	precision	recall	f1-score	support																																																										
0	0.88	0.84	0.86	10035																																																										
1	0.85	0.89	0.87	10021																																																										
accuracy			0.86	20056																																																										
macro avg	0.87	0.86	0.86	20056																																																										
weighted avg	0.87	0.86	0.86	20056																																																										

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
RandomForest Classifier	RandomForest Classifier is selected due to its robust performance across various datasets without requiring extensive hyperparameter tuning. It effectively reduces overfitting by aggregating multiple decision trees, each trained on a random subset of features and observations. This ensemble approach improves generalization and handles outliers and noisy data well.

