

Date	15 July 2024
Team ID	739961
Project Title SmartLender -	Automotive Kickstart
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
Logistic Regression	—	<pre> pred=lr.predict(x_test) pred  array([1, 1, 1, ..., 3, 1, 1])  lg_ac=accuracy_score(y_test,pred) lg_f1 = f1_score(y_test, pred, average='weighted') # Options: 'micro', 'macro', 'weighted' lg_r2=r2_score(y_test,pred) print(lg_ac) print(lg_f1) print(lg_r2)  0.8842576462898987 0.7568252728477291 0.5704852951733073 </pre>
RandomForest	—	<pre> kpred=knn.predict(x_test) kpred  array([1, 3, 1, ..., 3, 3, 1])  knn_ac=accuracy_score(y_test,kpred) knn_f1 = f1_score(y_test, kpred, average='weighted') # Options: 'micro', 'macro', 'weighted' knn_r2=r2_score(y_test,kpred) print(knn_ac) print(knn_f1) print(knn_r2)  0.827373250990383 0.7963496160215812 0.6793182337664017 </pre>

KNN	<div>—</div>	<pre>rpred=rf.predict(x_test) rpred  array([1, 1, 1, ..., 3, 3, 1])  rf_ac=accuracy_score(y_test, rpred) rf_f1 = f1_score(y_test, rpred, average='weighted') # Options: 'micro', 'macro', 'weighted' rf_r2=r2_score(y_test, rpred) print(rf_ac) print(rf_f1) print(rf_r2)  0.8595590294914034 0.8253275461000338 0.7824806763876369</pre>
SVM	<div>—</div>	<pre>spred=rf.predict(x_test) spred  array([1, 1, 1, ..., 3, 3, 1])  svm_ac=accuracy_score(y_test, spred) svm_f1 = f1_score(y_test, spred, average='weighted') # Options: 'micro', 'macro', 'weighted' svm_r2=r2_score(y_test, spred) print(svm_ac) print(svm_f1) print(svm_r2)  0.8595590294914034 0.8253275461000338 0.7824806763876369</pre>



Performance Metrics Comparison Report (2 Marks):

Model	Optimised Metric
Logistic regression	<pre>print('\n\n\n', classification_report(y_test,pred))</pre>

Random Forest	<pre>print('\n\n', classification_report(y_test,kpred))</pre> <table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>0</td><td>0.22</td><td>0.07</td><td>0.11</td><td>7766</td></tr><tr><td>1</td><td>0.81</td><td>0.91</td><td>0.86</td><td>39471</td></tr><tr><td>2</td><td>0.12</td><td>0.01</td><td>0.01</td><td>552</td></tr><tr><td>3</td><td>0.92</td><td>0.95</td><td>0.93</td><td>26831</td></tr><tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>351</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.83</td><td>74971</td></tr><tr><td>macro avg</td><td>0.41</td><td>0.39</td><td>0.38</td><td>74971</td></tr><tr><td>weighted avg</td><td>0.78</td><td>0.83</td><td>0.80</td><td>74971</td></tr></table>		precision	recall	f1-score	support	0	0.22	0.07	0.11	7766	1	0.81	0.91	0.86	39471	2	0.12	0.01	0.01	552	3	0.92	0.95	0.93	26831	4	0.00	0.00	0.00	351	accuracy			0.83	74971	macro avg	0.41	0.39	0.38	74971	weighted avg	0.78	0.83	0.80	74971
	precision	recall	f1-score	support																																										
0	0.22	0.07	0.11	7766																																										
1	0.81	0.91	0.86	39471																																										
2	0.12	0.01	0.01	552																																										
3	0.92	0.95	0.93	26831																																										
4	0.00	0.00	0.00	351																																										
accuracy			0.83	74971																																										
macro avg	0.41	0.39	0.38	74971																																										
weighted avg	0.78	0.83	0.80	74971																																										
KNN	<pre>print('\n\n', classification_report(y_test,rpred))</pre> <table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>0</td><td>0.28</td><td>0.07</td><td>0.12</td><td>7766</td></tr><tr><td>1</td><td>0.83</td><td>0.94</td><td>0.88</td><td>39471</td></tr><tr><td>2</td><td>0.26</td><td>0.03</td><td>0.06</td><td>552</td></tr><tr><td>3</td><td>0.95</td><td>0.99</td><td>0.97</td><td>26831</td></tr><tr><td>4</td><td>0.10</td><td>0.01</td><td>0.03</td><td>351</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>74971</td></tr><tr><td>macro avg</td><td>0.49</td><td>0.41</td><td>0.41</td><td>74971</td></tr><tr><td>weighted avg</td><td>0.81</td><td>0.86</td><td>0.83</td><td>74971</td></tr></table>		precision	recall	f1-score	support	0	0.28	0.07	0.12	7766	1	0.83	0.94	0.88	39471	2	0.26	0.03	0.06	552	3	0.95	0.99	0.97	26831	4	0.10	0.01	0.03	351	accuracy			0.86	74971	macro avg	0.49	0.41	0.41	74971	weighted avg	0.81	0.86	0.83	74971
	precision	recall	f1-score	support																																										
0	0.28	0.07	0.12	7766																																										
1	0.83	0.94	0.88	39471																																										
2	0.26	0.03	0.06	552																																										
3	0.95	0.99	0.97	26831																																										
4	0.10	0.01	0.03	351																																										
accuracy			0.86	74971																																										
macro avg	0.49	0.41	0.41	74971																																										
weighted avg	0.81	0.86	0.83	74971																																										
SVM	<pre>print('\n\n', classification_report(y_test,spred))</pre> <div>ggle output scrolling</div> <table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>0</td><td>0.28</td><td>0.07</td><td>0.12</td><td>7766</td></tr><tr><td>1</td><td>0.83</td><td>0.94</td><td>0.88</td><td>39471</td></tr><tr><td>2</td><td>0.26</td><td>0.03</td><td>0.06</td><td>552</td></tr><tr><td>3</td><td>0.95</td><td>0.99</td><td>0.97</td><td>26831</td></tr><tr><td>4</td><td>0.10</td><td>0.01</td><td>0.03</td><td>351</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>74971</td></tr><tr><td>macro avg</td><td>0.49</td><td>0.41</td><td>0.41</td><td>74971</td></tr><tr><td>weighted avg</td><td>0.81</td><td>0.86</td><td>0.83</td><td>74971</td></tr></table>		precision	recall	f1-score	support	0	0.28	0.07	0.12	7766	1	0.83	0.94	0.88	39471	2	0.26	0.03	0.06	552	3	0.95	0.99	0.97	26831	4	0.10	0.01	0.03	351	accuracy			0.86	74971	macro avg	0.49	0.41	0.41	74971	weighted avg	0.81	0.86	0.83	74971
	precision	recall	f1-score	support																																										
0	0.28	0.07	0.12	7766																																										
1	0.83	0.94	0.88	39471																																										
2	0.26	0.03	0.06	552																																										
3	0.95	0.99	0.97	26831																																										
4	0.10	0.01	0.03	351																																										
accuracy			0.86	74971																																										
macro avg	0.49	0.41	0.41	74971																																										
weighted avg	0.81	0.86	0.83	74971																																										

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
RandomForest	The RandomForest model was selected for its superior performance, exhibiting high accuracy during the predict and texting.Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model.

