

## Model Optimization and Tuning Phase Report

Date	21 June 2024
Team ID	739832
Project Title	Startup Prophet
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 Model	-	-

Support Vector Machine	-	-
------------------------	---	---

Random Forest Model	-	-
---------------------	---	---

### Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric																														
Logistic Regression Model	<pre>[ ] #LOGISTIC REGRESSION from sklearn.linear_model import LogisticRegression lr=LogisticRegression() lr.fit(x_bal,y_bal) y_pred=lr.predict(x_test)</pre> <pre>from sklearn.metrics import confusion_matrix,accuracy_score,classification_report print(confusion_matrix(y_test,y_pred)) print(classification_report(y_test,y_pred))</pre> <pre>[[136  34]  [ 56 133]]</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>0.71</td><td>0.80</td><td>0.75</td><td>170</td></tr><tr><td>1</td><td>0.80</td><td>0.70</td><td>0.75</td><td>189</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.75</td><td>359</td></tr><tr><td>macro avg</td><td>0.75</td><td>0.75</td><td>0.75</td><td>359</td></tr><tr><td>weighted avg</td><td>0.75</td><td>0.75</td><td>0.75</td><td>359</td></tr></tbody></table> <pre>[ ] from sklearn.metrics import log_loss print(log_loss(y_test,y_pred))</pre> <pre>9.03601338445834</pre>		precision	recall	f1-score	support	0	0.71	0.80	0.75	170	1	0.80	0.70	0.75	189	accuracy			0.75	359	macro avg	0.75	0.75	0.75	359	weighted avg	0.75	0.75	0.75	359
	precision	recall	f1-score	support																											
0	0.71	0.80	0.75	170																											
1	0.80	0.70	0.75	189																											
accuracy			0.75	359																											
macro avg	0.75	0.75	0.75	359																											
weighted avg	0.75	0.75	0.75	359																											

Support Vector Machine	<pre>#SUPPORT VECTOR MACHINE from sklearn.svm import SVC svm=SVC(kernel='rbf',C=2.0,random_state=42) svm.fit(x_bal,y_bal) y_predict=svm.predict(x_test)</pre> <pre>[ ] print(confusion_matrix(y_test,y_predict)) print(classification_report(y_test,y_predict))</pre> <pre>[[135  35]  [ 30 159]]</pre> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.82</td><td>0.79</td><td>0.81</td><td>170</td></tr><tr><td>1</td><td>0.82</td><td>0.84</td><td>0.83</td><td>189</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.82</td><td>359</td></tr><tr><td>macro avg</td><td>0.82</td><td>0.82</td><td>0.82</td><td>359</td></tr><tr><td>weighted avg</td><td>0.82</td><td>0.82</td><td>0.82</td><td>359</td></tr></table>		precision	recall	f1-score	support	0	0.82	0.79	0.81	170	1	0.82	0.84	0.83	189	accuracy			0.82	359	macro avg	0.82	0.82	0.82	359	weighted avg	0.82	0.82	0.82	359																														
	precision	recall	f1-score	support																																																									
0	0.82	0.79	0.81	170																																																									
1	0.82	0.84	0.83	189																																																									
accuracy			0.82	359																																																									
macro avg	0.82	0.82	0.82	359																																																									
weighted avg	0.82	0.82	0.82	359																																																									
Random Forest Model	<pre>#RANDOM FOREST MODEL from sklearn.ensemble import RandomForestClassifier rf=RandomForestClassifier() rf.fit(x_bal,y_bal) rf_test=rf.predict(x_test) rf_train=rf.predict(x_train) print(confusion_matrix(rf_test,y_test)) print(confusion_matrix(rf_train,y_train)) print(classification_report(rf_test,y_test)) print(classification_report(rf_train,y_train))</pre> <pre>[[163  7]  [ 7 182]] [[412  14]  [ 15 394]]</pre> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.96</td><td>0.96</td><td>0.96</td><td>170</td></tr><tr><td>1</td><td>0.96</td><td>0.96</td><td>0.96</td><td>189</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.96</td><td>359</td></tr><tr><td>macro avg</td><td>0.96</td><td>0.96</td><td>0.96</td><td>359</td></tr><tr><td>weighted avg</td><td>0.96</td><td>0.96</td><td>0.96</td><td>359</td></tr></table> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.96</td><td>0.97</td><td>0.97</td><td>426</td></tr><tr><td>1</td><td>0.97</td><td>0.96</td><td>0.96</td><td>409</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.97</td><td>835</td></tr><tr><td>macro avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>835</td></tr><tr><td>weighted avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>835</td></tr></table> <pre>[ ] from sklearn.metrics import log_loss print(log_loss(rf_test,y_test))</pre> <pre>1.4056020820268529</pre>		precision	recall	f1-score	support	0	0.96	0.96	0.96	170	1	0.96	0.96	0.96	189	accuracy			0.96	359	macro avg	0.96	0.96	0.96	359	weighted avg	0.96	0.96	0.96	359		precision	recall	f1-score	support	0	0.96	0.97	0.97	426	1	0.97	0.96	0.96	409	accuracy			0.97	835	macro avg	0.97	0.97	0.97	835	weighted avg	0.97	0.97	0.97	835
	precision	recall	f1-score	support																																																									
0	0.96	0.96	0.96	170																																																									
1	0.96	0.96	0.96	189																																																									
accuracy			0.96	359																																																									
macro avg	0.96	0.96	0.96	359																																																									
weighted avg	0.96	0.96	0.96	359																																																									
	precision	recall	f1-score	support																																																									
0	0.96	0.97	0.97	426																																																									
1	0.97	0.96	0.96	409																																																									
accuracy			0.97	835																																																									
macro avg	0.97	0.97	0.97	835																																																									
weighted avg	0.97	0.97	0.97	835																																																									

### Final Model Selection Justification (2 Marks):

Final Model	Reasoning

Random Forest Model	The Random Forest Model was selected for its superior performance, exhibiting high accuracy. Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model.
---------------------	---