

## Model Optimization and Tuning Phase Template

Date	03 October 2024
Team ID	LTVIP2024TMID24974
Project Title	Analysis Of Amazon Cell Phone Reviews
Maximum Marks	10 Marks

### Model Optimization and Tuning Phase


The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. Although various models were considered, the primary focus was on **Random Forest**, as it is well-suited for text classification tasks due to its simplicity and effectiveness. 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># Step 1: Convert text data into numerical features using TF-IDF # You can adjust ngram_range and max_features as per your requirement vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=3000) X = vectorizer.fit_transform(merged_df['cleaned_review'])  # Convert ratings to sentiment categories for Random Forest model def convert_rating_to_sentiment(rating):     if rating &gt;= 4:         return 'positive'     elif rating == 3:         return 'neutral'     else:         return 'negative'  y = merged_df['rating_x'].apply(convert_rating_to_sentiment)</pre>	<div>Classification Report:</div> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>negative</td><td>0.80</td><td>0.77</td><td>0.79</td><td>3350</td></tr><tr><td>neutral</td><td>0.91</td><td>0.07</td><td>0.14</td><td>926</td></tr><tr><td>positive</td><td>0.87</td><td>0.97</td><td>0.92</td><td>9322</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>13598</td></tr><tr><td>macro avg</td><td>0.86</td><td>0.60</td><td>0.61</td><td>13598</td></tr><tr><td>weighted avg</td><td>0.86</td><td>0.86</td><td>0.83</td><td>13598</td></tr></table>		precision	recall	f1-score	support	negative	0.80	0.77	0.79	3350	neutral	0.91	0.07	0.14	926	positive	0.87	0.97	0.92	9322	accuracy			0.86	13598	macro avg	0.86	0.60	0.61	13598	weighted avg	0.86	0.86	0.83	13598
	precision	recall	f1-score	support																																	
negative	0.80	0.77	0.79	3350																																	
neutral	0.91	0.07	0.14	926																																	
positive	0.87	0.97	0.92	9322																																	
accuracy			0.86	13598																																	
macro avg	0.86	0.60	0.61	13598																																	
weighted avg	0.86	0.86	0.83	13598																																	

	<pre>[ ] rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)  # Step 4: Train the Random Forest Classifier on the training data rf_classifier.fit(X_train, y_train)  # Step 5: Predict sentiment on the test set y_pred = rf_classifier.predict(X_test)  # Step 6: Evaluate the model accuracy = accuracy_score(y_test, y_pred) classification_report_output = classification_report(y_test, y_pred)  print("Random Forest Accuracy: ", (accuracy * 100)) print(f"Classification Report:\n{classification_report_output}")</pre>	
--	--	--

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

Model	Optimized Metric																																			
Model-1  Random Forest	<div><div></div><div>Random Forest Confusion Matrix</div></div> <div><pre>[[2580    2   768]  [ 327    69  530]  [ 315     5 9002]]</pre></div>																																			
	<div>Classification Report:</div> <table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>negative</td><td>0.80</td><td>0.77</td><td>0.79</td><td>3350</td></tr><tr><td>neutral</td><td>0.91</td><td>0.07</td><td>0.14</td><td>926</td></tr><tr><td>positive</td><td>0.87</td><td>0.97</td><td>0.92</td><td>9322</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>13598</td></tr><tr><td>macro avg</td><td>0.86</td><td>0.60</td><td>0.61</td><td>13598</td></tr><tr><td>weighted avg</td><td>0.86</td><td>0.86</td><td>0.83</td><td>13598</td></tr></table>		precision	recall	f1-score	support	negative	0.80	0.77	0.79	3350	neutral	0.91	0.07	0.14	926	positive	0.87	0.97	0.92	9322	accuracy			0.86	13598	macro avg	0.86	0.60	0.61	13598	weighted avg	0.86	0.86	0.83	13598
		precision	recall	f1-score	support																															
	negative	0.80	0.77	0.79	3350																															
	neutral	0.91	0.07	0.14	926																															
	positive	0.87	0.97	0.92	9322																															
	accuracy			0.86	13598																															
	macro avg	0.86	0.60	0.61	13598																															
	weighted avg	0.86	0.86	0.83	13598																															

**NOTE:** Although various models were considered, the primary focus was on Random Forest, as it is well-suited for text classification tasks due to its simplicity and effectiveness and ability to handle complex datasets without overfitting.

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

Final Model	Reasoning
Random Forest	Random Forest was chosen over other models because of its ability to handle complex, nonlinear relationships in the text data, which can improve sentiment prediction accuracy. It aggregates multiple decision trees, reducing the risk of overfitting and providing more robust, generalizable results compared to models like Logistic Regression or SVM.