Model Optimization and Tuning Phase Template

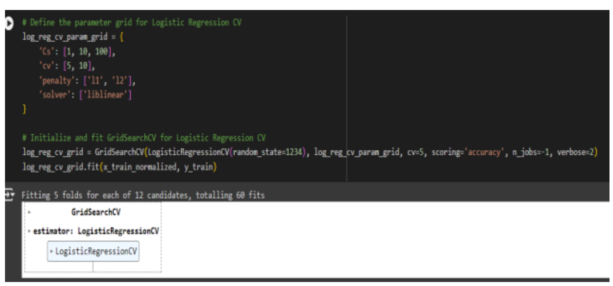
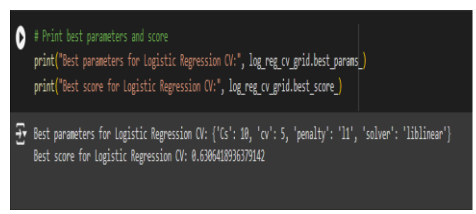
Date	7 July 2024
Team ID	SWTID1720434734
Project Title	Ecommerce shipping prediction
Maximum Marks	10 Marks

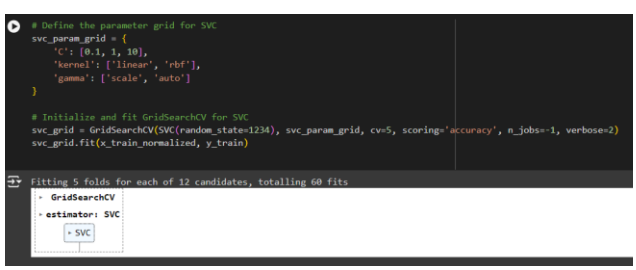
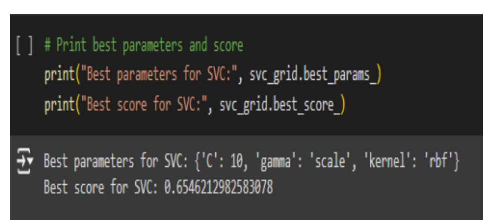
Model Optimization and Tuning Phase

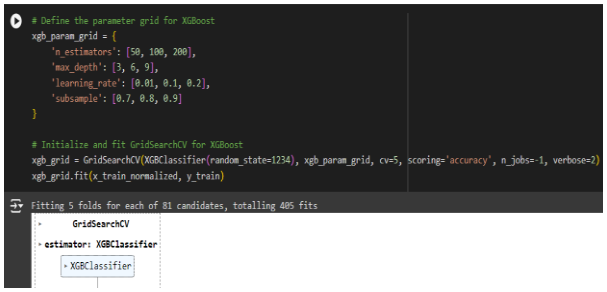
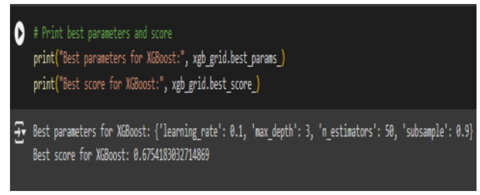
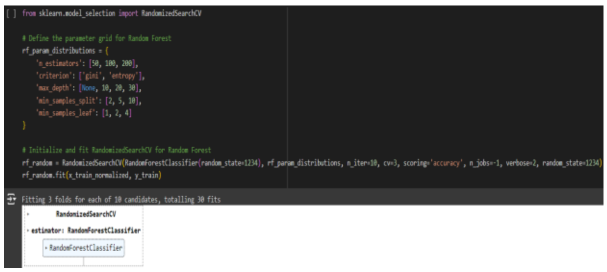
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

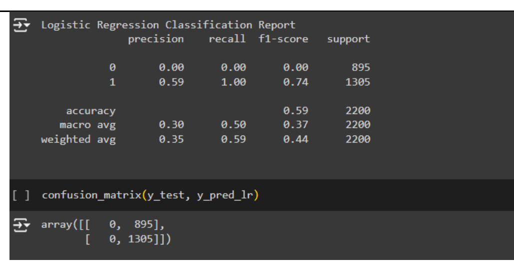
Model	Tuned Hyperparameters	Optimal Values

Logistic Regression CV	
	# Define the parameter grid for Logistic Regression CV
	log_reg_cv_param_grid = {
	'Cs': [1, 10, 100],
	'cv': [5, 10],
	'penalty': ['l1', 'l2'],
	'solver': ['liblinear']
	}
	# Initialize and fit GridSearchCV for Logistic Regression CV
	log_reg_cv_grid = GridSearchCV(LogisticRegressionCV(random_state=1234), log_reg_cv_param_grid, cv=5, scoring='accuracy', n_jobs=-1, verbose=2)
	log_reg_cv_grid.fit(x_train_normalized, y_train)
	Fitting 5 folds for each of 12 candidates, totalling 60 fits
	GridSearchCV
	estimator: LogisticRegressionCV
	LogisticRegressionCV
		
		# Print best parameters and score
		print("Best parameters for Logistic Regression CV:", log_reg_cv_grid.best_params_)
		print("Best score for Logistic Regression CV:", log_reg_cv_grid.best_score_)
		Best parameters for Logistic Regression CV: {'Cs': 10, 'cv': 5, 'penalty': 'l1', 'solver': 'liblinear'}
		Best score for Logistic Regression CV: 0.6306418936379142

SVC	
	# Define the parameter grid for SVC
	svc_param_grid = {
	'C': [0.1, 1, 10],
	'kernel': ['linear', 'rbf'],
	'gamma': ['scale', 'auto']
	}
	# Initialize and fit GridSearchCV for SVC
	svc_grid = GridSearchCV(SVC(random_state=1234), svc_param_grid, cv=5, scoring='accuracy', n_jobs=-1, verbose=2)
	svc_grid.fit(x_train_normalized, y_train)
	Fitting 5 folds for each of 12 candidates, totalling 60 fits
	GridSearchCV
	estimator: SVC
	SVC
		
		# Print best parameters and score
		print("Best parameters for SVC:", svc_grid.best_params_)
		print("Best score for SVC:", svc_grid.best_score_)
		Best parameters for SVC: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
		Best score for SVC: 0.6546212982583078

<table>
<tr>
<td>XGBoost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Forest</td>
<td></td>
<td></td>
</tr>
</table>

Performance Metrics Comparison Report (2 Marks):

<table>
<tr><th>Model</th><th>Optimized Metric</th></tr>
<tr>
<td>Logistic Regression</td>
<td></td>
</tr>
</table>

KNN

```
KNN Classification Report
              precision    recall  f1-score   support

           0       0.55      0.58      0.56       895
           1       0.70      0.67      0.68      1305

    accuracy                           0.63      2200
   macro avg       0.62      0.62      0.62      2200
weighted avg       0.64      0.63      0.63      2200
```

```
confusion_matrix(y_test, y_pred_knn)
```

```
array([[519, 376],
       [432, 873]])
```

Random Forest

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.71      0.63       895
           1       0.76      0.64      0.70      1305

    accuracy                           0.67      2200
   macro avg       0.67      0.68      0.67      2200
weighted avg       0.69      0.67      0.67      2200
```

```
confusion_matrix(y_test, y_pred_rf)
```

```
array([[632, 263],
       [464, 841]])
```

Ridge Classifier

```
Ridge Classifier Classification Report
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       895
           1       0.59      1.00      0.74      1305

    accuracy                           0.59      2200
   macro avg       0.30      0.50      0.37      2200
weighted avg       0.35      0.59      0.44      2200
```

```
confusion_matrix(y_test, y_pred_rc)
```

```
array([[   0,  895],
       [   0, 1305]])
```

XGBoost	```
	XGBoost Classification Report
	              precision recall f1-score support
	
	           0 0.59 0.63 0.61 895
	           1 0.73 0.70 0.71 1305
	
	    accuracy 0.67 2200
	   macro avg 0.66 0.66 0.66 2200
	weighted avg 0.67 0.67 0.67 2200
	
	[ ] confusion_matrix(y_test, y_pred_xgb)
	
	array([[562, 333],
	       [395, 910]])
	``

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	The Random forest model was selected for its superior performance, exhibiting high accuracy during hyperparameter tuning. Efficiently processes massive amounts of data. Can work with both numerical and categorical data. Combining multiple trees reduces the risk of overfitting. Helps identify the most influential features. Can handle datasets with missing values.