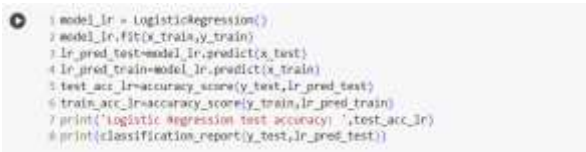
Model Optimization and Tuning Phase Report

Date	12 JULY 2024
ID	740022
Project Title	Lymphography Classification using ML
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.

Model	Tuned Hyperparameters	Optimal Values

Logistic Regression	
	model_lr = LogisticRegression()
	model_lr.fit(x_train,y_train)
	lr_pred_test=model_lr.predict(x_test)
	lr_pred_train=model_lr.predict(x_train)
	test_acc_lr=accuracy_score(y_test,lr_pred_test)
	train_acc_lr=accuracy_score(y_train,lr_pred_train)
	print('Logistic Regression test accuracy: ',test_acc_lr)
	print(classification_report(y_test,lr_pred_test))
		
		# Evaluate the performance of the tuned model
		accuracy = accuracy_score(y_test, y_pred)
		print("Optimal Hyperparameters: {best_params}")
		print("Accuracy on Test Set: {accuracy}")
		Optimal Hyperparameters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'best'}
		Accuracy on Test Set: 0.3333333333333

SVM	```from sklearn.svm import SVC model_svm = SVC() model_svm.fit(x_train,y_train) svm_pred_test=model_svm.predict(x_test) svm_pred_train=model_svm.predict(x_train) test_acc_svm=accuracy_score(y_test,svm_pred_test) train_acc_svm=accuracy_score(y_train,svm_pred_train) print('SVM Test Accuracy: ',test_acc_svm) print(classification_report(y_test,svm_pred_test))``` 	

Hyperparameter Tuning Documentation (6 Marks):

KNN		
Gradient Boosting	```from sklearn.ensemble import GradientBoostingClassifier GBC=GradientBoostingClassifier() GBC.fit(x_train,y_train) GBC_pred_test=GBC.predict(x_test) GBC_pred_train=GBC.predict(x_train) GBC_test_acc=accuracy_score(y_test,GBC_pred_test) GBC_train_acc=accuracy_score(y_train,GBC_pred_train) print('GB accuracy is: ',GBC_test_acc) print(classification_report(y_test,GBC_pred_test))``` 	

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric

Logistic Regression	

```
Logistic Regression test accuracy:  0.8
              precision    recall  f1-score   support

         2.0       0.71      1.00      0.83        12
         3.0       1.00      0.71      0.83        17
         4.0       0.00      0.00      0.00         1

    accuracy                           0.80        30
   macro avg       0.57      0.57      0.55        30
weighted avg       0.85      0.80      0.80        30
```

```
[ ]  1 confusion_matrix(y_test,lr_pred_test)
```

```
array([[12,  0,  0],
       [ 4, 12,  1],
       [ 1,  0,  0]])
```

SVM	

```
SVM Test Accuracy:  0.8
              precision    recall  f1-score   support

         2.0       0.73      0.92      0.81        12
         3.0       0.87      0.76      0.81        17
         4.0       0.00      0.00      0.00         1

    accuracy                           0.80        30
   macro avg       0.53      0.56      0.54        30
weighted avg       0.78      0.80      0.79        30
```

```
[ ]  1 confusion_matrix(y_test,svm_pred_test)
```

```
array([[11,  1,  0],
       [ 4, 13,  0],
       [ 0,  1,  0]])
```

KNN	81.35593220338984 ----- Regular Training Set Used ----- Classification report for KNeighborsClassifier(): precision recall f1-score support 2.0 0.87 0.93 0.90 14 3.0 0.80 0.86 0.83 14 4.0 0.00 0.00 0.00 2 accuracy 0.83 30 macro avg 0.56 0.60 0.57 30 weighted avg 0.78 0.83 0.80 30 Accuracy score: 83.33333333333334
Gradient Boosting	GB accuracy is: 0.8333333333333334 precision recall f1-score support 2.0 0.73 0.92 0.81 12 3.0 0.93 0.82 0.87 17 4.0 0.00 0.00 0.00 1 accuracy 0.83 30 macro avg 0.56 0.58 0.56 30 weighted avg 0.82 0.83 0.82 30 1 GBC.predict([[4.0,2.0,1.0,1.0,1.0,1.0,1.0,2.0,1.0,2.0,2.0,2.0,4.0,8.0,1.0,1.0,2.0,2.0]]) array([3.])

Final Model Selection Justification (2 Marks):

Final Model	Reasoning

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