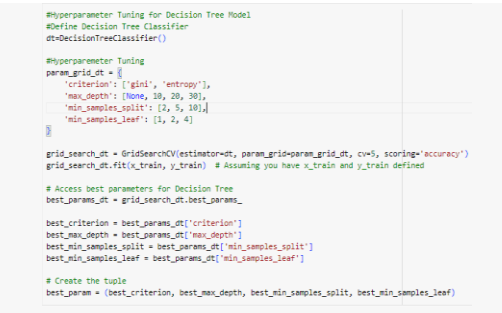
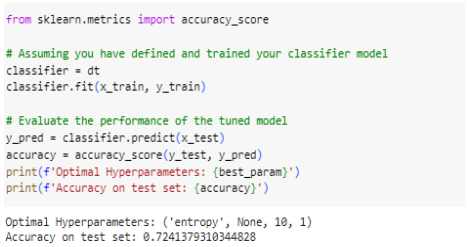
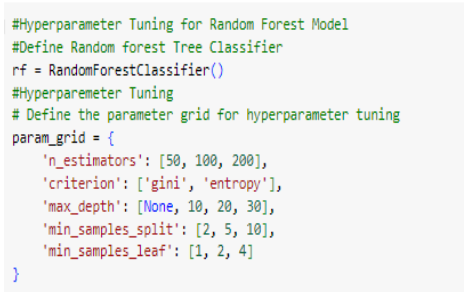
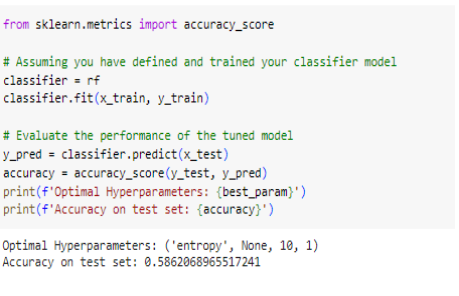
Model Optimization and Tuning Phase Report

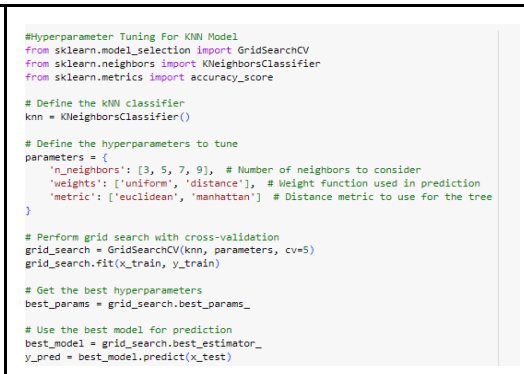
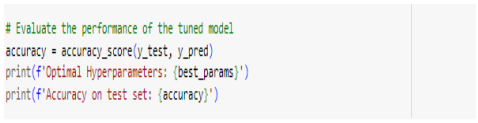
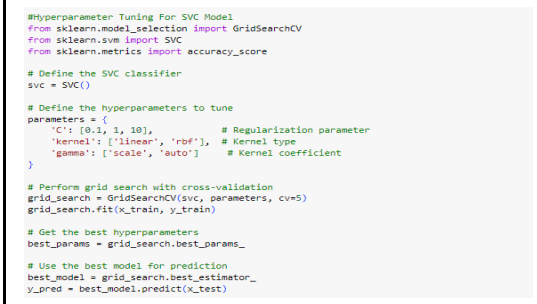
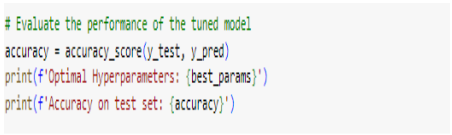
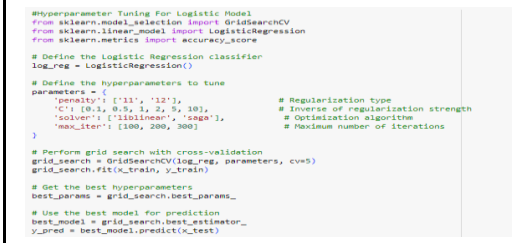
Date	21 June 2024
Team ID	739812
Project Title	Eudaimonia Engine: Machine Learning Delving into Happiness Classification
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

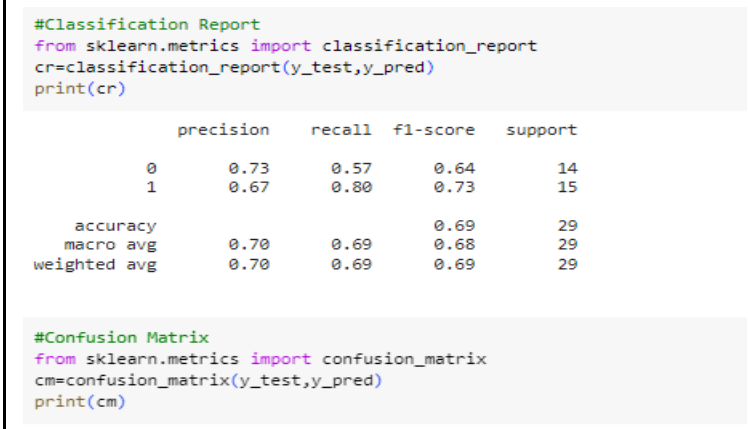
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Decision Tree		
Random Forest		

KNN		 Optimal Hyperparameters: {'metric': 'manhattan', 'n_neighbors': 7, 'weights': 'uniform'} Accuracy on test set: 0.5517241379310345
SVC		 Optimal Hyperparameters: {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'} Accuracy on test set: 0.4827586206896552
Logistic Model		 Optimal Hyperparameters: {'C': 2, 'max_iter': 100, 'penalty': 'l2', 'solver': 'liblinear'} Accuracy on test set: 0.4827586206896552

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Decision Tree	

Random Forest	`#Classification Report`
	`from sklearn.metrics import classification_report`
	`cr=classification_report(y_test,y_pred)`
	`print(cr)`
	
	`precision recall f1-score support`
	
	`0 0.56 0.36 0.43 14`
	`1 0.55 0.73 0.63 15`
	
	`accuracy 0.55 29`
	`macro avg 0.55 0.55 0.53 29`
	`weighted avg 0.55 0.55 0.54 29`
	
	`#Confusion Matrix`
	`from sklearn.metrics import confusion_matrix`
	`cm=confusion_matrix(y_test,y_pred)`
	`print(cm)`
	
	`[[ 5 9]`
	`[ 4 11]]`

KNN	`#Classification Report`
	`from sklearn.metrics import classification_report`
	`cr=classification_report(y_test,y_pred)`
	`print(cr)`
	
	`precision recall f1-score support`
	
	`0 0.25 0.14 0.18 14`
	`1 0.43 0.60 0.50 15`
	
	`accuracy 0.38 29`
	`macro avg 0.34 0.37 0.34 29`
	`weighted avg 0.34 0.38 0.35 29`
	
	`#Confusion Matrix`
	`from sklearn.metrics import confusion_matrix`
	`cm=confusion_matrix(y_test,y_pred)`
	`print(cm)`
	
	`[[ 2 12]`
	`[ 6 9]]`

SVC	`#Classification Report`
	`from sklearn.metrics import classification_report`
	`cr=classification_report(y_test,y_pred)`
	`print(cr)`
	
	`precision recall f1-score support`
	
	`0 0.25 0.14 0.18 14`
	`1 0.43 0.60 0.50 15`
	
	`accuracy 0.38 29`
	`macro avg 0.34 0.37 0.34 29`
	`weighted avg 0.34 0.38 0.35 29`
	
	`#Confusion Matrix`
	`from sklearn.metrics import confusion_matrix`
	`cm=confusion_matrix(y_test,y_pred)`
	`print(cm)`
	
	`[[ 2 12]`
	`[ 6 9]]`

Logistic Model	`#Classification Report`
	`from sklearn.metrics import classification_report`
	`cr=classification_report(y_test,y_pred)`
	`print(cr)`
	
	`precision recall f1-score support`
	
	`0 0.25 0.14 0.18 14`
	`1 0.43 0.60 0.50 15`
	
	`accuracy 0.38 29`
	`macro avg 0.34 0.37 0.34 29`
	`weighted avg 0.34 0.38 0.35 29`
	
	`#Confusion Matrix`
	`from sklearn.metrics import confusion_matrix`
	`cm=confusion_matrix(y_test,y_pred)`
	`print(cm)`
	
	`[[ 2 12]`
	`[ 6 9]]`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Decision Tree Model	The Decision Tree 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.