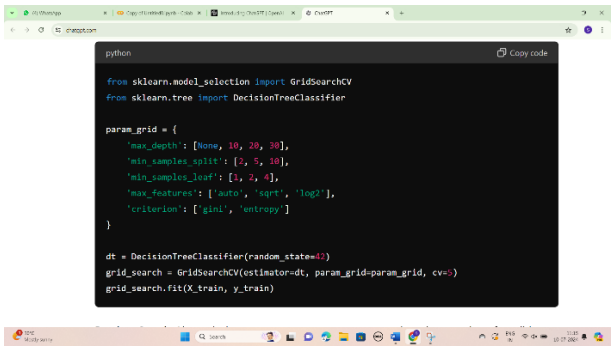
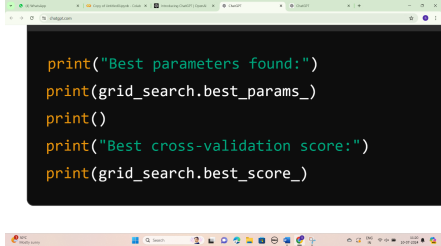
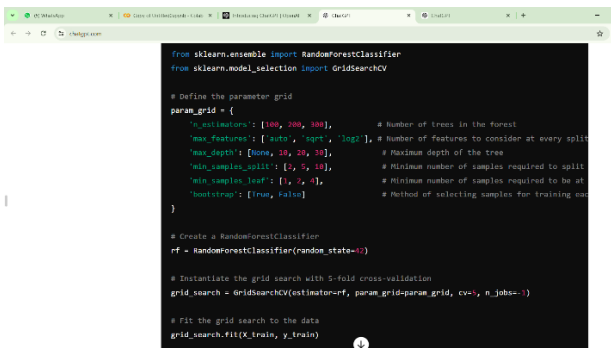
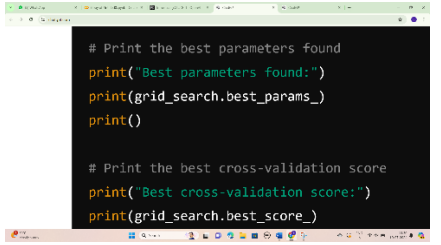


Model Optimization and Tuning Phase Report

Date	21 June 2024
Team ID	739769
Project Title	Life Style Change Due To Covid Prediction
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

Metrics Selection: Choose appropriate metrics based on the nature of your problem (e.g., accuracy, precision, recall, F1-score, AUC-ROC).

Model	Tuned Hyperparameters	Optimal Values
Decision Tree	 <pre>python from sklearn.model_selection import GridSearchCV from sklearn.tree import DecisionTreeClassifier param_grid = { 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'max_features': ['auto', 'sqrt', 'log2'], 'criterion': ['gini', 'entropy'] } dt = DecisionTreeClassifier(random_state=42) grid_search = GridSearchCV(estimator=dt, param_grid=param_grid, cv=5) grid_search.fit(X_train, y_train)</pre>	 <pre>print("Best parameters found:") print(grid_search.best_params_) print() print("Best cross-validation score:") print(grid_search.best_score_)</pre>
Random Forest	 <pre>(from sklearn.ensemble import RandomForestClassifier from sklearn.model_selection import GridSearchCV # Define the parameter grid param_grid = { 'n_estimators': [100, 200, 300], # Number of trees in the forest 'max_features': ['auto', 'sqrt', 'log2'], # Maximum number of features to consider at every split 'max_depth': [None, 10, 20, 30], # Maximum depth of the tree 'min_samples_split': [2, 5, 10], # Minimum number of samples required to split 'min_samples_leaf': [1, 2, 4], # Minimum number of samples required to be at a leaf node 'bootstrap': [True, False] # Method of selecting samples for training each tree } # Create a RandomForestClassifier rf = RandomForestClassifier(random_state=42) # Instantiate the grid search with 5-fold cross-validation grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1) # Fit the grid search to the data grid_search.fit(X_train, y_train)</pre>	 <pre># Print the best parameters found print("Best parameters found:") print(grid_search.best_params_) print() # Print the best cross-validation score print("Best cross-validation score:") print(grid_search.best_score_)</pre>

Primary Metric: Select a primary metric that aligns with your project goals (e.g., maximizing accuracy if balanced prediction is critical, optimizing recall if identifying all positive cases is crucial).

Hyperparameter Tuning Documentation (6 Marks):

Logistic Regression	-	-
---------------------	---	---

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric																														
Decision Tree	<pre>print("\nClassification Report: \n", classification_report(y_test, y_pred))</pre> <p>Accuracy: 99.57446808510639</p> <p>Classification Report:</p> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.99</td><td>1.00</td><td>0.99</td><td>85</td></tr><tr><td>1</td><td>1.00</td><td>0.99</td><td>1.00</td><td>150</td></tr><tr><td>accuracy</td><td></td><td></td><td>1.00</td><td>235</td></tr><tr><td>macro avg</td><td>0.99</td><td>1.00</td><td>1.00</td><td>235</td></tr><tr><td>weighted avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>235</td></tr></table>		precision	recall	f1-score	support	0	0.99	1.00	0.99	85	1	1.00	0.99	1.00	150	accuracy			1.00	235	macro avg	0.99	1.00	1.00	235	weighted avg	1.00	1.00	1.00	235
	precision	recall	f1-score	support																											
0	0.99	1.00	0.99	85																											
1	1.00	0.99	1.00	150																											
accuracy			1.00	235																											
macro avg	0.99	1.00	1.00	235																											
weighted avg	1.00	1.00	1.00	235																											

Random Forest

```
print("\nClassification Report: \n", classification_report(y_test, y_pred))
```

Accuracy: 97.02127659574468

Classification Report:				
	precision	recall	f1-score	support
0	0.94	0.98	0.96	85
1	0.99	0.97	0.98	150
accuracy			0.97	235
macro avg	0.96	0.97	0.97	235
weighted avg	0.97	0.97	0.97	235

Logistic Regression

```
print('\nClassification Report:',classification_report(y_test,y_pred))
```

Accuracy: 82.97872340425532

Classification Report:			precision	recall	f1-score	support
0	0.82	0.68	0.74	85		
1	0.84	0.91	0.87	150		
accuracy			0.83	235		
macro avg			0.83	0.80	0.81	235
weighted avg			0.83	0.83	0.83	235

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Gradient Boosting	The Gradient Boosting 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.