


Model Optimization and Tuning Phase Report

| | |
|---------------|----------------------------------|
| Date | 15 July 2024 |
| Team ID | 740054 |
| Project Title | Doctors Annual Salary 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 |
|-------------------|--|
| Linear Regression |  <pre> [20] g_train_pred[0] array([[276.8001007], [200.5407019], [627.4714024], [200.1043205], [1267.38001743]]) [21] r2_score(y_train, y_train_pred)*100 67.461420775808 [22] Mean square error for training data mean_squared_error(y_train, y_train_pred) 3485.21118075111 </pre> |

| | |
|---------------|--|
| Random Forest | <pre>[18] y_train_pred = rf.predict(x_train) y_test_pred = rf.predict(x_test) [19] r2_score(y_train,y_train_pred)*100 88.11481511617%</pre> <pre>[20] mean_squared_error(y_train,y_train_pred) 47796.71077777777%</pre> |
| Decision Tree | <pre>[17] y_train_pred[0] array([1016., 1076., 1016., 1006., 1700.]) [18] y_test_pred[0] array([1700., 1106., 1506., 1006., 1506.])</pre> <pre>[19] r2_score(y_train,y_train_pred)*100 100.0</pre> <pre>[20] mean_squared_error(y_train,y_train_pred) 0.0</pre> |
| XGBRegressor | <pre>[46] y_train_pred = xg_reg.predict(x_train) y_test_pred = xg_reg.predict(x_test) [47] r2_score(y_train,y_train_pred)*100 91.88888888888889%</pre> <pre>[48] mean_squared_error(y_train,y_train_pred) 6.888888888888889</pre> <pre>[49] r2_score(y_test,y_test_pred)*100 91.17647058823529%</pre> |

Performance Metrics Comparison Report (2 Marks):

| | |
|-------------------|--|
| | |
| Linear Regression | <pre>[23] Accuracy for with testing data (linear regression) r2_score(y_test,y_test_pred)*100 27.200000000000004%</pre> <pre>[24] Mean square error for testing data mean_squared_error(y_test,y_test_pred) 371596.56125000003</pre> |

| | |
|---------------|---|
| Random Forest | <pre> [31] F2_score(y_test, y_test_pred)*100 37.40340080709866 [32] mean_squared_error(y_test, y_test_pred) 336638.10881127775 </pre> |
| Decision Tree | <pre> [31] F2_score(y_test, y_test_pred)*100 30.264116470888916 [32] mean_squared_error(y_test, y_test_pred) 334512.5 </pre> |
| XGBRegressor | <pre> from sklearn.metrics import mean_squared_error mean_squared_error(y_test, y_test_pred) 336638.10881127775 [31] F2_score(y_test, y_test_pred)*100 30.170687951188874 </pre> |

Final Model Selection Justification (2 Marks):

| Final Model | Reasoning |
|---------------|--|
| Decision Tree | Decision trees can be a good starting point for predicting doctors' annual salaries due to their interpretability and ability to handle non-linear relationships. By carefully tuning hyperparameters and evaluating performance, you can build a robust model. For better generalization, consider using ensemble methods like random forests or gradient boosting if decision trees alone do not provide satisfactory results. |