Model Optimization and Tuning Phase Report

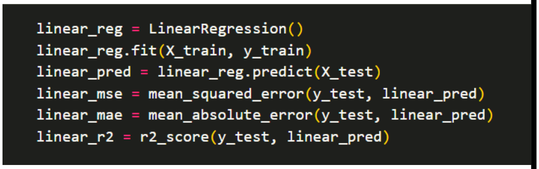
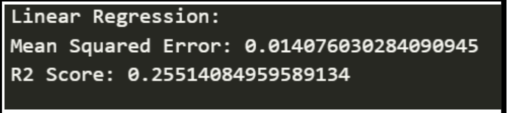
Date	11 July 2024
Team ID	SWTID1720178802
Project Title	Garment worker productivity
Maximum Marks	10 Marks

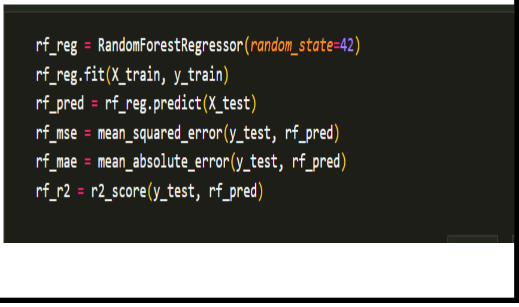
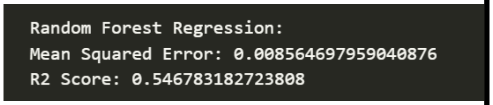
Model Optimization and Tuning Phase

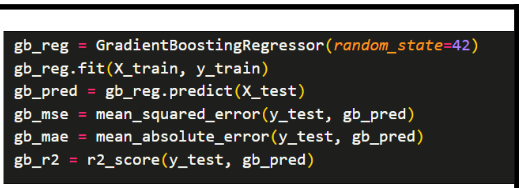
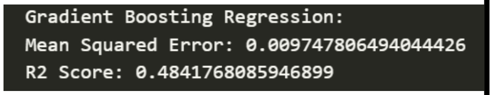
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

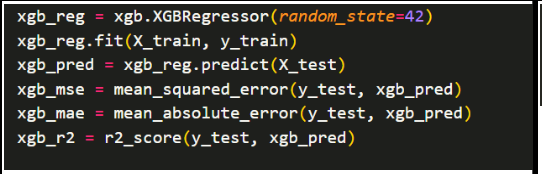
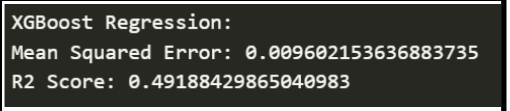
Hyperparameter Tuning Documentation (6 Marks):

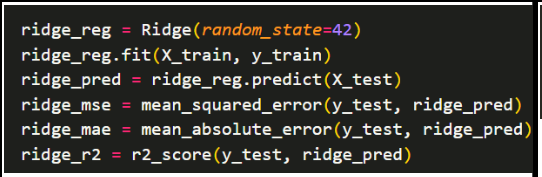
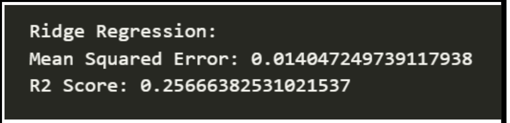
Model	Tuned Hyperparameters	Optimal Values

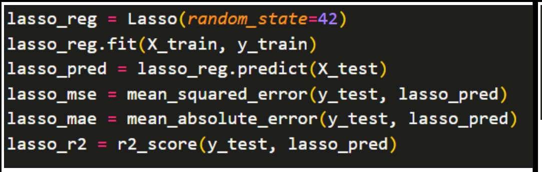
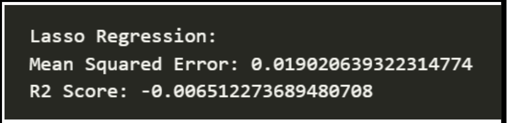
Linear Regression	
	`linear_reg = LinearRegression()`
	`linear_reg.fit(X_train, y_train)`
	`linear_pred = linear_reg.predict(X_test)`
	`linear_mse = mean_squared_error(y_test, linear_pred)`
	`linear_mae = mean_absolute_error(y_test, linear_pred)`
	`linear_r2 = r2_score(y_test, linear_pred)`
		
		`Linear Regression:`
		`Mean Squared Error: 0.014076030284090945`
		`R2 Score: 0.25514084959589134`

Random Forest	
	`rf_reg = RandomForestRegressor(random_state=42)`
	`rf_reg.fit(X_train, y_train)`
	`rf_pred = rf_reg.predict(X_test)`
	`rf_mse = mean_squared_error(y_test, rf_pred)`
	`rf_mae = mean_absolute_error(y_test, rf_pred)`
	`rf_r2 = r2_score(y_test, rf_pred)`
		
		`Random Forest Regression:`
		`Mean Squared Error: 0.008564697959040876`
		`R2 Score: 0.546783182723808`

Gradient Boosting	
	gb_reg = GradientBoostingRegressor(random_state=42)
	gb_reg.fit(X_train, y_train)
	gb_pred = gb_reg.predict(X_test)
	gb_mse = mean_squared_error(y_test, gb_pred)
	gb_mae = mean_absolute_error(y_test, gb_pred)
	gb_r2 = r2_score(y_test, gb_pred)
		
		Gradient Boosting Regression:
		Mean Squared Error: 0.009747806494044426
		R2 Score: 0.4841768085946899

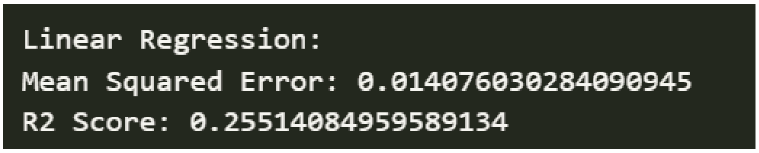
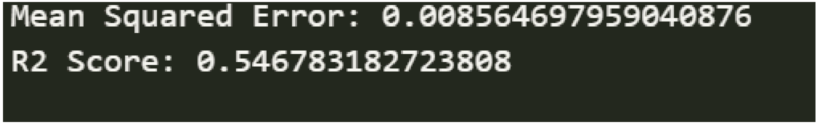
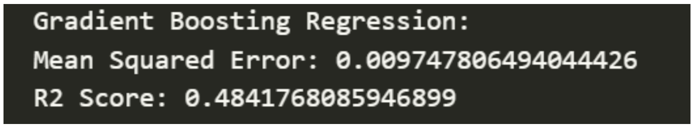
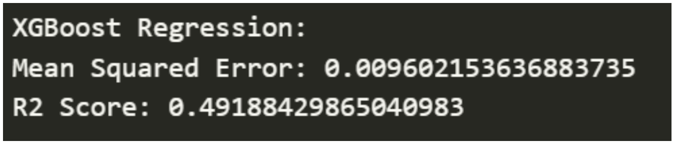
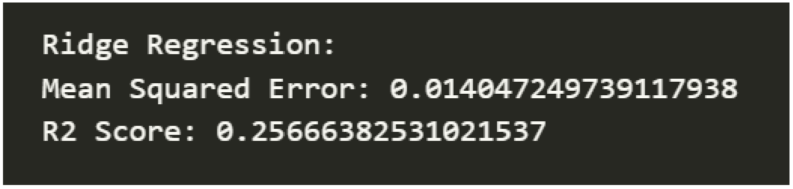
XGBoost Regression	
	xgb_reg = xgb.XGBRegressor(random_state=42)
	xgb_reg.fit(X_train, y_train)
	xgb_pred = xgb_reg.predict(X_test)
	xgb_mse = mean_squared_error(y_test, xgb_pred)
	xgb_mae = mean_absolute_error(y_test, xgb_pred)
	xgb_r2 = r2_score(y_test, xgb_pred)
		
		XGBoost Regression:
		Mean Squared Error: 0.009602153636883735
		R2 Score: 0.4918842986504098 3

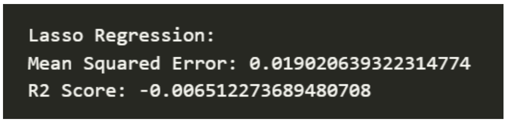
Ridge	
	ridge_reg = Ridge(random_state=42)
	ridge_reg.fit(X_train, y_train)
	ridge_pred = ridge_reg.predict(X_test)
	ridge_mse = mean_squared_error(y_test, ridge_pred)
	ridge_mae = mean_absolute_error(y_test, ridge_pred)
	ridge_r2 = r2_score(y_test, ridge_pred)
		
		Ridge Regression:
		Mean Squared Error: 0.014047249739117938
		R2 Score: 0.25666382531021537

Lasso	
	lasso_reg = Lasso(random_state=42)
	lasso_reg.fit(X_train, y_train)
	lasso_pred = lasso_reg.predict(X_test)
	lasso_mse = mean_squared_error(y_test, lasso_pred)
	lasso_mae = mean_absolute_error(y_test, lasso_pred)
	lasso_r2 = r2_score(y_test, lasso_pred)
		
		Lasso Regression:
		Mean Squared Error: 0.019020639322314774
		R2 Score: -0.006512273689480708

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric

Linear	 Linear Regression: Mean Squared Error: 0.014076030284090945 R2 Score: 0.25514084959589134
Random Forest	 Mean Squared Error: 0.008564697959040876 R2 Score: 0.546783182723808
Gradient Boosting	 Gradient Boosting Regression: Mean Squared Error: 0.009747806494044426 R2 Score: 0.4841768085946899
XGBoost Regression	 XGBoost Regression: Mean Squared Error: 0.009602153636883735 R2 Score: 0.49188429865040983
Ridge	 Ridge Regression: Mean Squared Error: 0.014047249739117938 R2 Score: 0.25666382531021537

Lasso	 Lasso Regression: Mean Squared Error: 0.019020639322314774 R2 Score: -0.006512273689480708

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest Regression	The Random Forest Regression model was selected for its superior performance. It had the least Mean Squared Error and the best R2 Score.