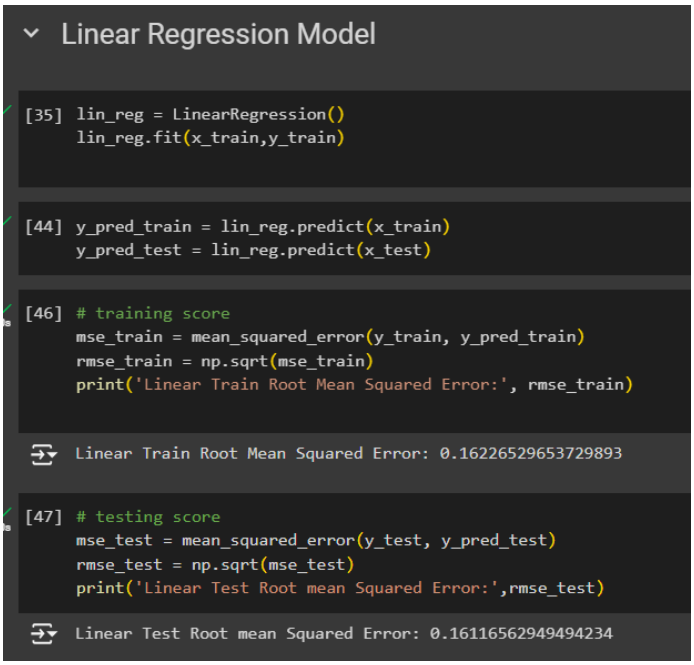
Model Development Phase Template

Date	4 July 2024
Team ID	SWTID1720090524
Project Title	Garment Worker Productivity Prediction
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:



Lasso Regression Model

```
[48] lasso = Lasso(alpha = 0.001)
     lasso.fit(x_train, y_train)
```

```
Lasso(alpha=0.001)
```

```
[49] y_pred_train = lasso.predict(x_train)
     y_pred_test = lasso.predict(x_test)
```

```
[50] # training score
     mse_train = mean_squared_error(y_train, y_pred_train)
     rmse_train = np.sqrt(mse_train)
     print('Lasso Train Root Mean Squared Error:', rmse_train)
```

```
Lasso Train Root Mean Squared Error: 0.16246420183571206
```

```
[51] # testing score
     mse_test = mean_squared_error(y_test, y_pred_test)
     rmse_test = np.sqrt(mse_test)
     print('Lasso Test Root Mean Squared Error:', rmse_test)
```

```
Lasso Test Root Mean Squared Error: 0.16121034106828316
```

Ridge Regression Model

```
[52] ridge = Ridge(alpha = 1.9)
     ridge.fit(x_train, y_train)
```

```
Ridge(alpha=1.9)
```

```
[53] y_pred_train = ridge.predict(x_train)
     y_pred_test = ridge.predict(x_test)
```

```
[54] # training score
     mse_train = mean_squared_error(y_train, y_pred_train)
     rmse_train = np.sqrt(mse_train)
     print('Ridge Train Root Mean Squared Error:', rmse_train)
```

```
Ridge Train Root Mean Squared Error: 0.16226837609384914
```

```
[55] # testing score
     mse_test = mean_squared_error(y_test, y_pred_test)
     rmse_test = np.sqrt(mse_test)
     print('Ridge Test Root mean Squared Error:', rmse_test)
```

```
Ridge Test Root mean Squared Error: 0.16115218890295427
```

Decision Tree Regressor Model

```
[56] dtr = DecisionTreeRegressor(max_depth= 4, min_samples_split= 3, min_samples_leaf= 2)
     dtr.fit(x_train,y_train)
```

```
DecisionTreeRegressor(max_depth=4, min_samples_leaf=2, min_samples_split=3)
```

```
[57] y_pred_train = dtr.predict(x_train)
     y_pred_test = dtr.predict(x_test)
```

```
[58] # training score
     mse_train = mean_squared_error(y_train, y_pred_train)
     rmse_train = np.sqrt(mse_train)
     print('Decision Tree Train Root Mean Squared Error:', rmse_train)
```

```
Decision Tree Train Root Mean Squared Error: 0.13187559206436333
```

```
# testing score
mse_test = mean_squared_error(y_test, y_pred_test)
rmse_test = np.sqrt(mse_test)
print('Decision Tree Test Root mean Squared Error:', rmse_test)
```

```
Decision Tree Test Root mean Squared Error: 0.12918875831022705
```

Random Forest Regressor

```
[60] rfr = RandomForestRegressor(n_estimators=100, max_depth=6, min_weight_fraction_leaf=0.05, max_features=0.8, random_state=42)
     rfr.fit(x_train,y_train)
```

```
RandomForestRegressor(max_depth=6, max_features=0.8,
                      min_weight_fraction_leaf=0.05, random_state=42)
```

```
[63] y_pred_train = rfr.predict(x_train)
     y_pred_test = rfr.predict(x_test)
```

```
[64] # training score
     mse_train = mean_squared_error(y_train, y_pred_train)
     rmse_train = np.sqrt(mse_train)
     print('Random Forest Train Root Mean Squared Error:', rmse_train)
```

```
Random Forest Train Root Mean Squared Error: 0.13062916799216504
```

```
# testing score
mse_test = mean_squared_error(y_test, y_pred_test)
rmse_test = np.sqrt(mse_test)
print('Random Forest Test Root mean Squared Error:', rmse_test)
```

```
Random Forest Test Root mean Squared Error: 0.1272406384012021
```

Gradient Boosting Regressor

```
[66] gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=1, random_state=42)
     gbr.fit(x_train,y_train)
```

```
GradientBoostingRegressor
GradientBoostingRegressor(max_depth=1, random_state=42)
```

```
[67] y_pred_train = gbr.predict(x_train)
     y_pred_test = gbr.predict(x_test)
```

```
[68] # training score
     mse_train = mean_squared_error(y_train, y_pred_train)
     rmse_train = np.sqrt(mse_train)
     print('Gradient Boosting Train Root Mean Squared Error:', rmse_train)
```

```
Gradient Boosting Train Root Mean Squared Error: 0.1424427737607694
```

```
[69] # testing score
     mse_test = mean_squared_error(y_test, y_pred_test)
     rmse_test = np.sqrt(mse_test)
     print('Gradient Boosting Test Root mean Squared Error:', rmse_test)
```

```
Gradient Boosting Test Root mean Squared Error: 0.13953081336729123
```

Extreme Gradient Boost Regressor

```
[70] xgb = XGBRegressor (n_estimators=300, learning_rate=0.05, max_leaves = 3, random_state = 1)
     xgb.fit(x_train, y_train)
```

```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.05, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=3,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=300, n_jobs=None,
             num_parallel_tree=None, random_state=1, ...)
```

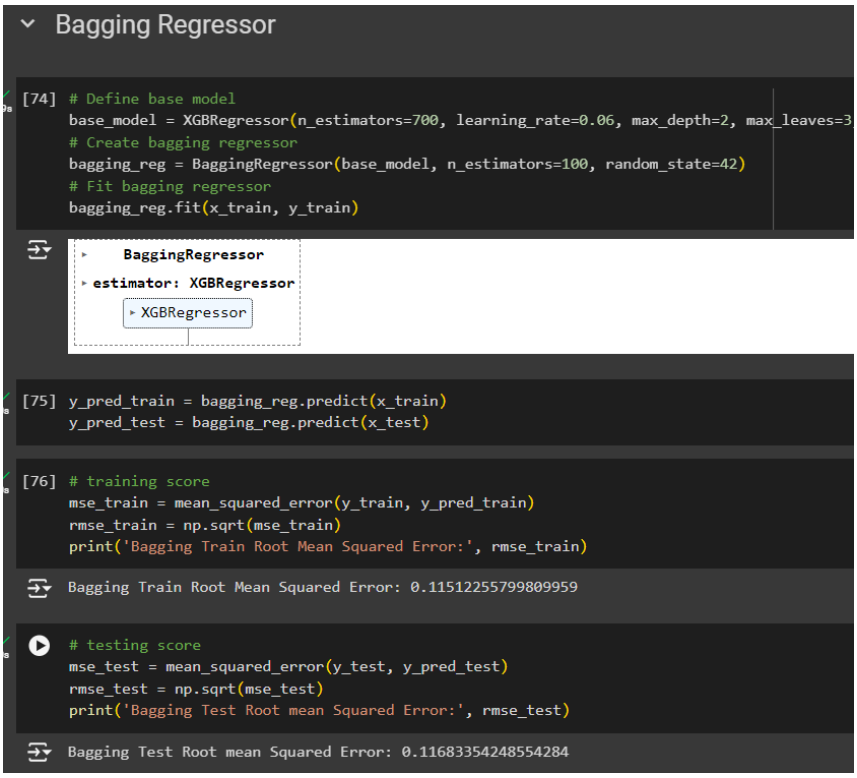
```
[71] y_pred_train = xgb.predict(x_train)
     y_pred_test = xgb.predict(x_test)
```

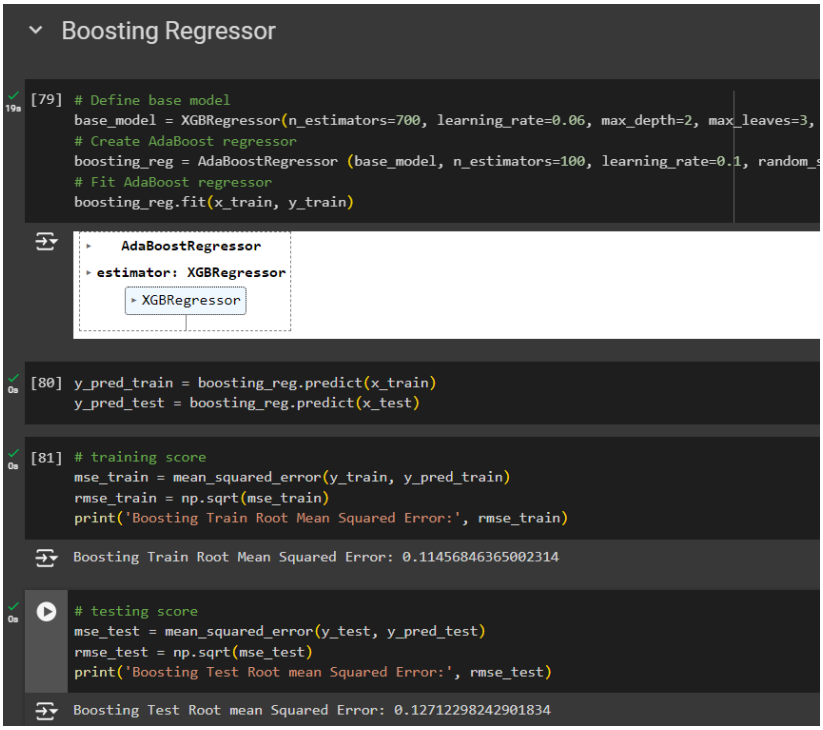
```
# training score
mse_train = mean_squared_error(y_train, y_pred_train)
rmse_train = np.sqrt(mse_train)
print('Extreme Gradient Boosting Train Root Mean Squared Error:', rmse_train)
```

```
Extreme Gradient Boosting Train Root Mean Squared Error: 0.12323119231675213
```

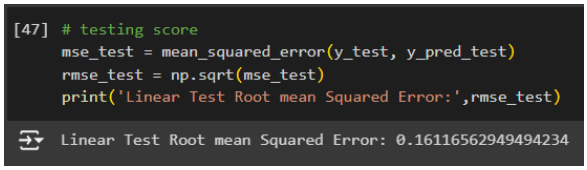
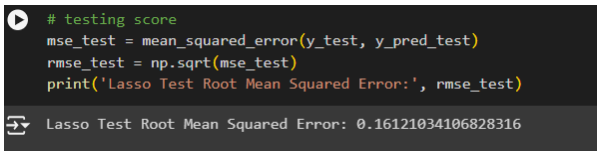
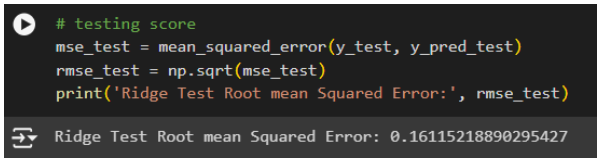
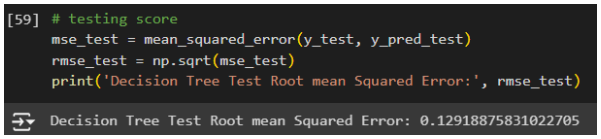
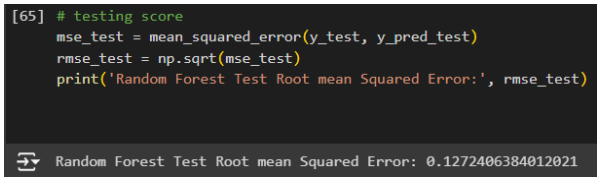
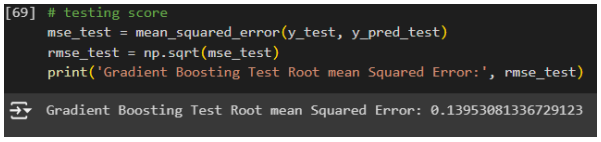
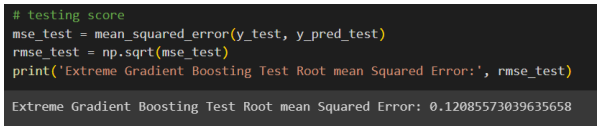
```
[73] # testing score
     mse_test = mean_squared_error(y_test, y_pred_test)
     rmse_test = np.sqrt(mse_test)
     print('Extreme Gradient Boosting Test Root mean Squared Error:', rmse_test)
```

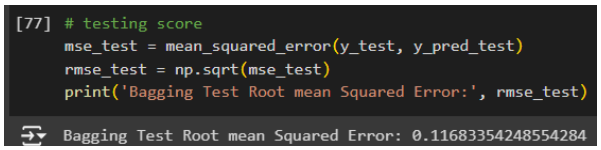
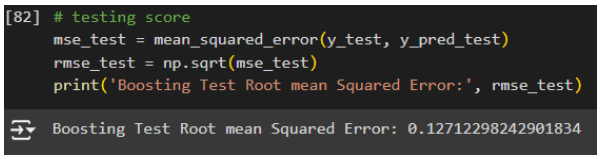
```
Extreme Gradient Boosting Test Root mean Squared Error: 0.12085573039635658
```





Model Validation and Evaluation Report:

Model	Classification/ RMSE Report	Accuracy(Test Root Mean Squared Error)	Confusion Matrix
Linear Regression Model		Test Root mean Squared Error: 0.16116562949494234	--
Lasso Regression Model		Test Root Mean Squared Error: 0.16121034106828316	--
Ridge Regression Model		Test Root mean Squared Error: 0.16115218890295427	--
Decision Tree Regressor Model		Test Root mean Squared Error: 0.12918875831022705	--
Random forest Regressor Model		Test Root mean Squared Error: 0.1272406384012021	--
Gradient Boosting Regressor Model		Test Root mean Squared Error: 0.13953081336729123	--
Extreme Gradient		Test Root mean Squared Error: 0.12085573039635658	

Boosting Model			
Bagging Regressor Model		Test Root mean Squared Error: 0.11683354248554284	--
Boosting Regressor Model		Test Root mean Squared Error: 0.12712298242901834	--