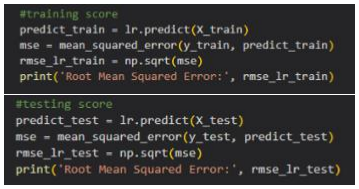
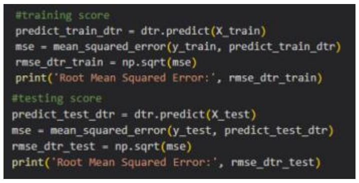
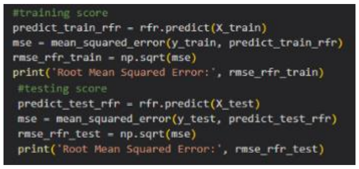
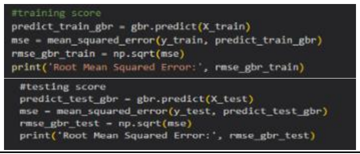
Project Development Phase
Model Performance Test

Date	09 November 2023
Team ID	Team-592545
Project Name	Garment worker productivity prediction
Maximum Marks	10 Marks

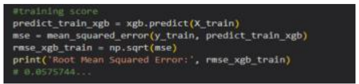
Model Performance Testing:

Project team shall fill the following information in model performance testing template.

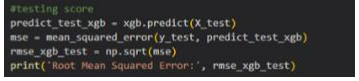
S.No.	Parameter	Values	Screenshot

1.	Metrics	**Regression Model:**
		RMSE score
			Linear Regression Model
			
			#training score
			predict_train = lr.predict(X_train)
			mse = mean_squared_error(y_train, predict_train)
			rmse_lr_train = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_lr_train)
			#testing score
			predict_test = lr.predict(X_test)
			mse = mean_squared_error(y_test, predict_test)
			rmse_lr_test = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_lr_test)
			Decision Tree Regressor Model
			
			#training score
			predict_train_dtr = dtr.predict(X_train)
			mse = mean_squared_error(y_train, predict_train_dtr)
			rmse_dtr_train = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_dtr_train)
			#testing score
			predict_test_dtr = dtr.predict(X_test)
			mse = mean_squared_error(y_test, predict_test_dtr)
			rmse_dtr_test = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_dtr_test)
			Random Forest Regressor Model
			
			#training score
			predict_train_rfr = rfr.predict(X_train)
			mse = mean_squared_error(y_train, predict_train_rfr)
			rmse_rfr_train = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_rfr_train)
			#testing score
			predict_test_rfr = rfr.predict(X_test)
			mse = mean_squared_error(y_test, predict_test_rfr)
			rmse_rfr_test = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_rfr_test)
			Gradient Boosting Regressor Model
			
			#training score
			predict_train_gbr = gbr.predict(X_train)
			mse = mean_squared_error(y_train, predict_train_gbr)
			rmse_gbr_train = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_gbr_train)
			#testing score
			predict_test_gbr = gbr.predict(X_test)
			mse = mean_squared_error(y_test, predict_test_gbr)
			rmse_gbr_test = np.sqrt(mse)
			print('Root Mean Squared Error:', rmse_gbr_test)

Extreme Gradient Boost Regressor Model

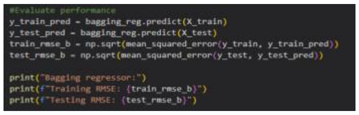


```
#training score
predict_train_xgb = xgb.predict(X_train)
mse = mean_squared_error(y_train, predict_train_xgb)
rmse_xgb_train = np.sqrt(mse)
print('Root Mean Squared Error:', rmse_xgb_train)
# 0.0575744...
```



```
#testing score
predict_test_xgb = xgb.predict(X_test)
mse = mean_squared_error(y_test, predict_test_xgb)
rmse_xgb_test = np.sqrt(mse)
print('Root Mean Squared Error:', rmse_xgb_test)
```

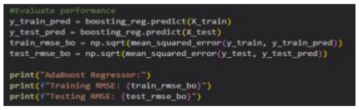
Bagging Regressor Model



```
#Evaluate performance
y_train_pred = bagging_reg.predict(X_train)
y_test_pred = bagging_reg.predict(X_test)
train_rmse_b = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse_b = np.sqrt(mean_squared_error(y_test, y_test_pred))

print("Bagging regressor:")
print(f"Training RMSE: {train_rmse_b}")
print(f"Testing RMSE: {test_rmse_b}")
```

Boosting Regressor Model



```
#Evaluate performance
y_train_pred = boosting_reg.predict(X_train)
y_test_pred = boosting_reg.predict(X_test)
train_rmse_bo = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse_bo = np.sqrt(mean_squared_error(y_test, y_test_pred))

print("AdaBoost Regressor:")
print(f"Training RMSE: {train_rmse_bo}")
print(f"Testing RMSE: {test_rmse_bo}")
```