Model Development Phase Template

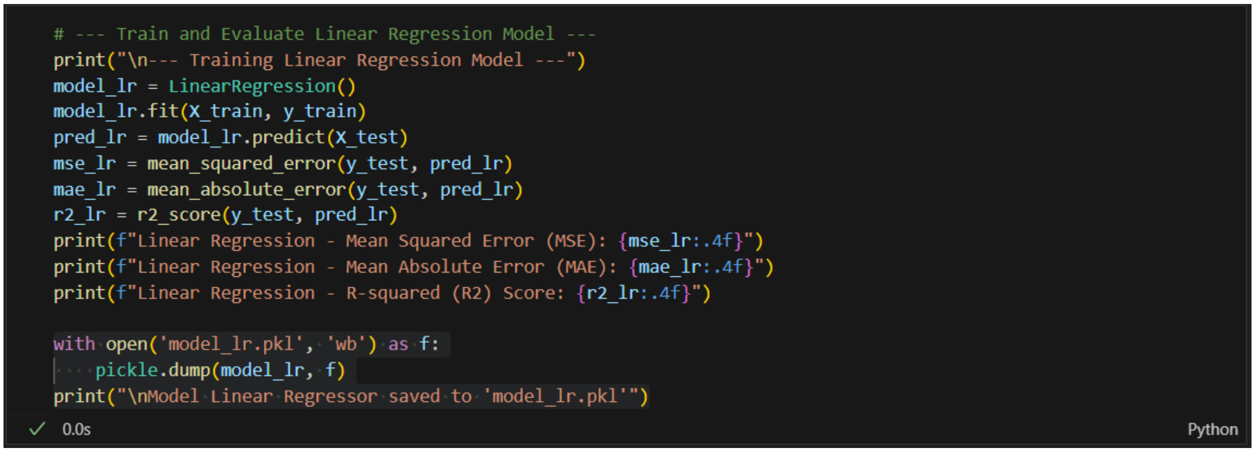
Date	05 August 2025
Skillwallet ID	SWUID20250186419
Project Title	Employee Productivity Prediction Application
Maximum Marks	4 Marks

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:

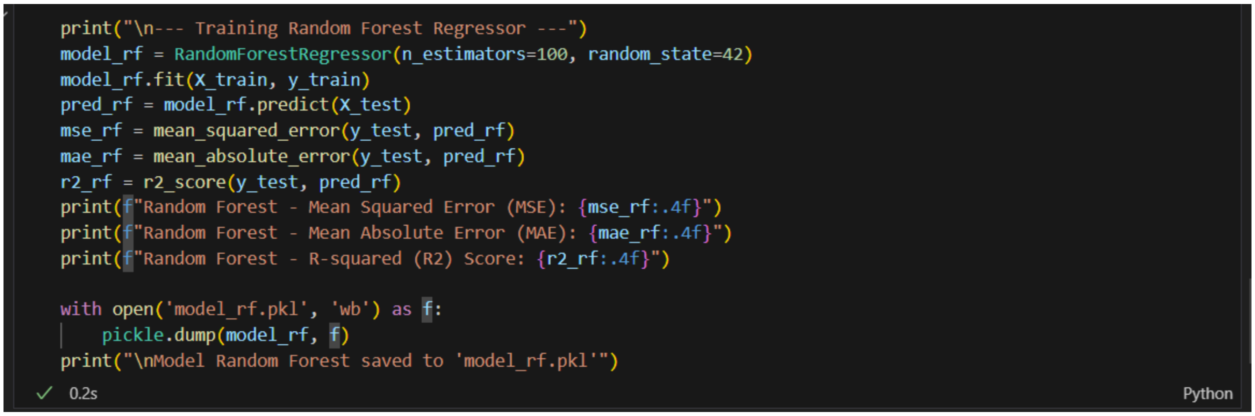
- Linear Regression :



```python
# --- Train and Evaluate Linear Regression Model ---
print("\n--- Training Linear Regression Model ---")
model_lr = LinearRegression()
model_lr.fit(X_train, y_train)
pred_lr = model_lr.predict(X_test)
mse_lr = mean_squared_error(y_test, pred_lr)
mae_lr = mean_absolute_error(y_test, pred_lr)
r2_lr = r2_score(y_test, pred_lr)
print(f"Linear Regression - Mean Squared Error (MSE): {mse_lr:.4f}")
print(f"Linear Regression - Mean Absolute Error (MAE): {mae_lr:.4f}")
print(f"Linear Regression - R-squared (R2) Score: {r2_lr:.4f}")

with open('model_lr.pkl', 'wb') as f:
    pickle.dump(model_lr, f)
print("\nModel Linear Regressor saved to 'model_lr.pkl'")
```

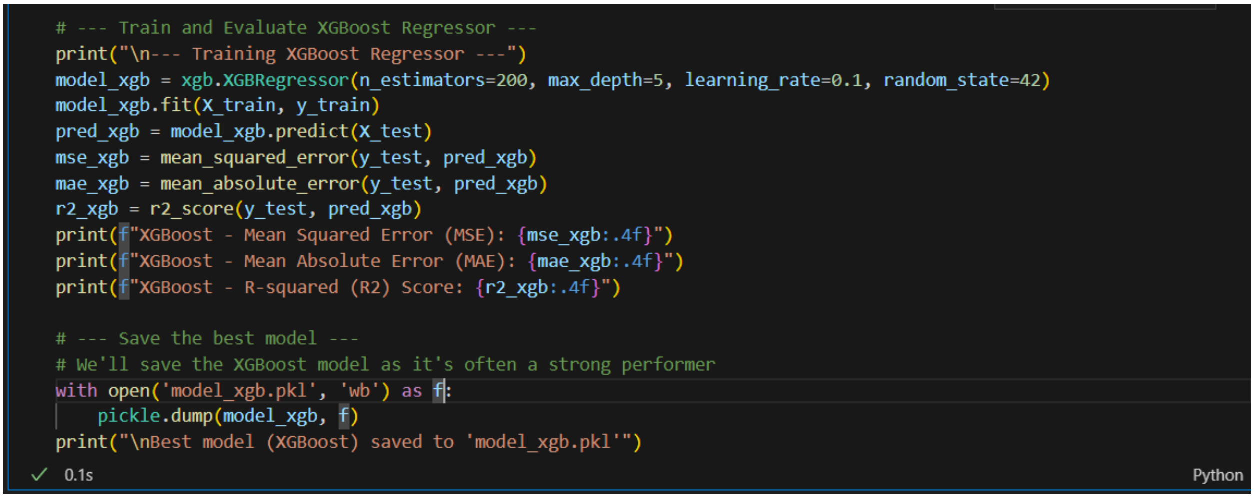
- Random Forest :



```python
print("\n--- Training Random Forest Regressor ---")
model_rf = RandomForestRegressor(n_estimators=100, random_state=42)
model_rf.fit(X_train, y_train)
pred_rf = model_rf.predict(X_test)
mse_rf = mean_squared_error(y_test, pred_rf)
mae_rf = mean_absolute_error(y_test, pred_rf)
r2_rf = r2_score(y_test, pred_rf)
print(f"Random Forest - Mean Squared Error (MSE): {mse_rf:.4f}")
print(f"Random Forest - Mean Absolute Error (MAE): {mae_rf:.4f}")
print(f"Random Forest - R-squared (R2) Score: {r2_rf:.4f}")

with open('model_rf.pkl', 'wb') as f:
    pickle.dump(model_rf, f)
print("\nModel Random Forest saved to 'model_rf.pkl'")
```

- XGBoost :



```python
# --- Train and Evaluate XGBoost Regressor ---
print("\n--- Training XGBoost Regressor ---")
model_xgb = xgb.XGBRegressor(n_estimators=200, max_depth=5, learning_rate=0.1, random_state=42)
model_xgb.fit(X_train, y_train)
pred_xgb = model_xgb.predict(X_test)
mse_xgb = mean_squared_error(y_test, pred_xgb)
mae_xgb = mean_absolute_error(y_test, pred_xgb)
r2_xgb = r2_score(y_test, pred_xgb)
print(f"XGBoost - Mean Squared Error (MSE): {mse_xgb:.4f}")
print(f"XGBoost - Mean Absolute Error (MAE): {mae_xgb:.4f}")
print(f"XGBoost - R-squared (R2) Score: {r2_xgb:.4f}")

# --- Save the best model ---
# We'll save the XGBoost model as it's often a strong performer
with open('model_xgb.pkl', 'wb') as f:
    pickle.dump(model_xgb, f)
print("\nBest model (XGBoost) saved to 'model_xgb.pkl'")
```

Model Validation and Evaluation Report:

Model	MSE	MAE	R^2 Score
Model 1 (Linear Regression)	0.8791	0.1958	-27.9999
Model 2 (Random Forest)	0.0217	0.0985	0.2856
Model 3 (XGBoost)	0.0240	0.1015	0.2068