

## Model Development Phase Template

Date	6th July 2024
Team ID	739719
Project Title	Garment Workers Productivity Predictions
Maximum Marks	10 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 a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

### Initial Model Training Code (5 marks):

```
df = pd.read_csv(r'C:\Users\srira\Downloads\miniProject\garments_worker_productivity.csv')
df.head()
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=42)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(823, 12)
(353, 12)
(823,)
(353,)
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt
from sklearn.metrics import mean_absolute_percentage_error
```

**Model Validation and Evaluation Report (5 marks)**

	<b>Summary</b>	<b>Training and Validation Performance Metrics</b>
--	----------------	--

<p>Model 1</p> <p><b>Random Forest Regressor</b></p>	<h2>Random Forest Regressor Summary</h2> <h3>Model Parameters</h3> <ul style="list-style-type: none"> <li>• <b>Number of Trees:</b> Optimal number of trees determined through hyperparameter tuning.</li> <li>• <b>Max Depth:</b> Maximum depth of the trees, optimized to prevent overfitting.</li> <li>• <b>Min Samples Split:</b> Minimum number of samples required to split an internal node.</li> <li>• <b>Min Samples Leaf:</b> Minimum number of samples required to be at a leaf node.</li> </ul> <h3>Training Process:</h3> <ul style="list-style-type: none"> <li>• <b>Data Preprocessing:</b> Standardized or normalized input features.</li> <li>• <b>Bootstrapping:</b> Random sampling with replacement to create multiple training sets for the trees.</li> <li>• <b>Feature Selection:</b> Random selection of features at each split to ensure diverse trees.</li> </ul> <h3>Evaluation Metrics:</h3> <ul style="list-style-type: none"> <li>• <b>Mean Absolute Error (MAE):</b> Measures the average magnitude of the errors in the predictions.</li> <li>• <b>Mean Squared Error (MSE):</b> Measures the average of the squares of the errors, penalizing larger errors.</li> <li>• <b>R<sup>2</sup> Score:</b> Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.</li> </ul>	<pre>from sklearn.ensemble import RandomForestRegressor randf = RandomForestRegressor(random_state=42) randf.fit(x_train,y_train) pred_randf = randf.predict(x_test) print("MAE :", mean_absolute_error(y_test, pred_randf)) print("MSE :", mean_squared_error(y_test, pred_randf)) print("RMSE :",sqrt(mean_squared_error(y_test, pred_randf))) print("MAPE :",mean_absolute_percentage_error(y_test, pred_randf))</pre> <p>MAE : 0.08366785595438364 MSE : 0.015441874867015823 RMSE : 0.12426534057015183 MAPE : 0.14067390864389964</p>
--	--	---

<p>Model 2</p> <p>Gradient Boosting Regressor</p>	<h3>Gradient Boosting Regressor Summary</h3> <p><b>Model Parameters:</b></p> <ul style="list-style-type: none"> <li>• <b>Number of Estimators:</b> Total number of boosting stages (trees).</li> <li>• <b>Learning Rate:</b> Shrinks the contribution of each tree.</li> <li>• <b>Max Depth:</b> Maximum depth of the individual regression estimators (trees).</li> <li>• <b>Min Samples Split:</b> Minimum number of samples required to split an internal node.</li> <li>• <b>Min Samples Leaf:</b> Minimum number of samples required to be at a leaf node.</li> <li>• <b>Subsample:</b> Fraction of samples used for fitting the individual base learners.</li> </ul> <p><b>Training Process:</b></p> <ul style="list-style-type: none"> <li>• <b>Data Preprocessing:</b> Standardized or normalized input features.</li> <li>• <b>Initialization:</b> Starts with an initial prediction, often the mean of the target values.</li> <li>• <b>Sequential Training:</b> Each tree is trained on the residuals of the previous trees' predictions.</li> <li>• <b>Loss Function:</b> Mean Squared Error (MSE) to minimize the difference between predicted and actual values.</li> </ul> <p><b>Evaluation Metrics:</b></p> <ul style="list-style-type: none"> <li>• <b>Mean Absolute Error (MAE):</b> Average magnitude of the errors in the predictions.</li> <li>• <b>Mean Squared Error (MSE):</b> Average of the squares of the errors, penalizing larger errors.</li> <li>• <b>R<sup>2</sup> Score:</b> Proportion of the variance in the dependent variable that is predictable from the independent variables.</li> </ul>	<pre>from sklearn.ensemble import GradientBoostingRegressor gb = GradientBoostingRegressor(random_state=42) gb.fit(x_train,y_train) pred_gb = gb.predict(x_test) print("MAE :", mean_absolute_error(y_test, pred_gb)) print("MSE :", mean_squared_error(y_test, pred_gb)) print("RMSE :",sqrt(mean_squared_error(y_test, pred_gb))) print("MAPE :",mean_absolute_percentage_error(y_test, pred_gb))</pre> <p>MAE : 0.08052610453252707 MSE : 0.013325110632581337 RMSE : 0.11543444300806123 MAPE : 0.1351165000418134</p>
---	---	--

--	--	--