

## Model Development Phase Template

Date	15 March 2024
Team ID	XXXXXX
Project Title	Human Resource Management: Predicting Employee Promotions Using Machine Learning
Maximum Marks	10 Marks

### Initial Model Training Code, Model Validation and Evaluation Report

The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models. The details will be presented through respective screenshots.

#### Initial Model Training Code:

```
# Decision Tree
def decisionTree(x_train, x_test, y_train, y_test):
    dt = DecisionTreeClassifier()
    dt.fit(x_train, y_train)
    yPred = dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test, yPred))
    print('Classification report')
    print(classification_report(y_test, yPred))
```

```
# Random Forest
def randomForest(x_train, x_test, y_train, y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train, y_train)
    yPred = rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test, yPred))
    print('Classification report')
    print(classification_report(y_test, yPred))
```

```
# KNN
def KNN(x_train, x_test, y_train, y_test):
    knn = KNeighborsClassifier()
    knn.fit(x_train, y_train)
    yPred = knn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test, yPred))
    print('Classification report')
    print(classification_report(y_test, yPred))
```

```
# Xg Boost Model
def xgboost(x_train, x_test, y_train, y_test):
    xg= GradientBoostingClassifier()
    xg.fit(x_train, y_train)
    yPred = xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test, yPred))
    print('Classification report')
    print(classification_report(y_test, yPred))
```

### Model validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decision Tree	<pre>***DecisionTreeClassifier*** Confusion matrix [[14090  985]  [   798 14211]] Classification report precision    recall  f1-score   support     0       0.95     0.93     0.94     15075    1       0.94     0.95     0.94     15009   accuracy          0.94  macro avg         0.94 weighted avg         0.94</pre>	94%	<pre>Confusion matrix [[14090  985]  [   798 14211]]</pre>

Random Forest	<pre> ***RandomForestClassifier*** Confusion matrix [[15016   59]  [ 1036 13973]] Classification report       precision    recall  f1-score   support       0       0.94      1.00      0.96      15075      1       1.00      0.93      0.96      15009   accuracy: 0.97  macro avg: 0.97      0.96      0.96      30084  weighted avg: 0.97      0.96      0.96      30084 </pre>	96%	<pre> Confusion matrix [[15016   59]  [ 1036 13973]] </pre>
KNN	<pre> ***KNeighborsClassifier*** Confusion matrix [[ 9694  5381]  [ 1300 13709]] Classification report       precision    recall  f1-score   support       0       0.88      0.64      0.74      15075      1       0.72      0.91      0.80      15009   accuracy: 0.78  macro avg: 0.80      0.78      0.77      30084  weighted avg: 0.80      0.78      0.77      30084 </pre>	91%	<pre> Confusion matrix [[ 9694  5381]  [ 1300 13709]] </pre>
Xgboost	<pre> ***GradientBoostingClassifier*** Confusion matrix [[15028   47]  [ 1092 13917]] Classification report       precision    recall  f1-score   support       0       0.93      1.00      0.96      15075      1       1.00      0.93      0.96      15009   accuracy: 0.96  macro avg: 0.96      0.96      0.96      30084  weighted avg: 0.96      0.96      0.96      30084 </pre>	96%	<pre> Confusion matrix [[15028   47]  [ 1092 13917]] </pre>