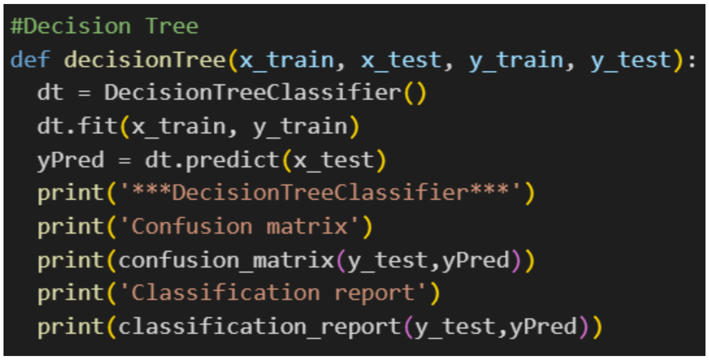
Model Development Phase Template

Date	25 June 2025
Team ID	SWTID1750155746
Project Title	Human Resource Management: Predicting Employee Promotions using Machine Learning
Maximum Marks	4 Marks

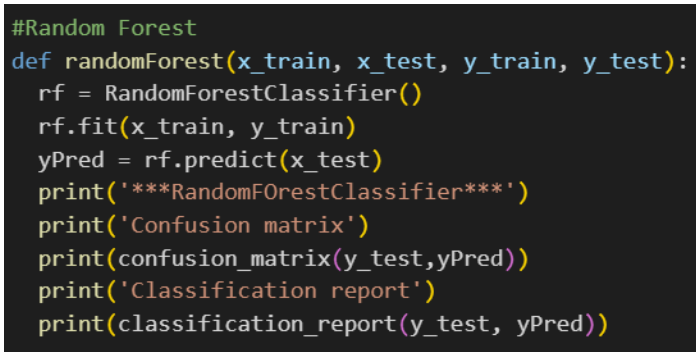
Initial Model Training Code, Model Validation and Evaluation Report

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.

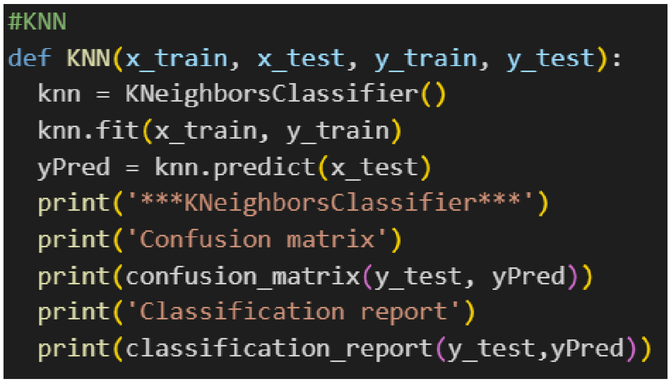
Initial Model Training Code:



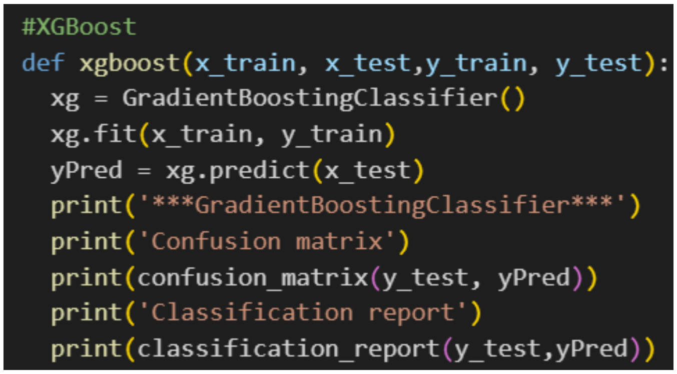
```
#Decision Tree
def decisionTree(x_train, x_test, y_train, y_test):
  dt = DecisionTreeClassifier()
  dt.fit(x_train, y_train)
  yPred = dt.predict(x_test)
  print('***DecisionTreeClassifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test,yPred))
  print('Classification report')
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



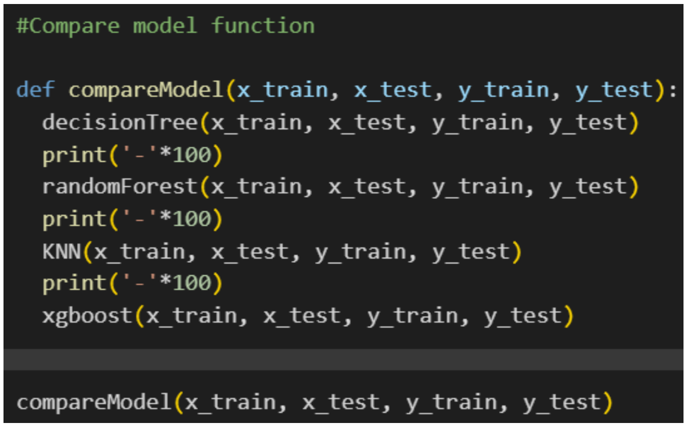
```
#Random Forest
def randomForest(x_train, x_test, y_train, y_test):
  rf = RandomForestClassifier()
  rf.fit(x_train, y_train)
  yPred = rf.predict(x_test)
  print('***RandomFOrestClassifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test,yPred))
  print('Classification report')
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



```
#XGBoost
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))
```

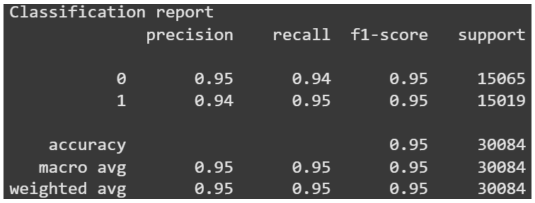
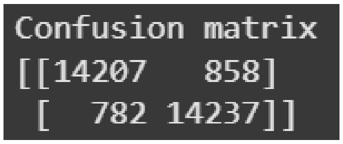
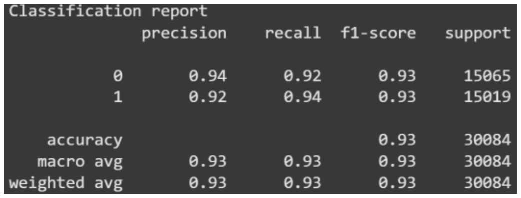
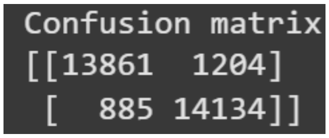
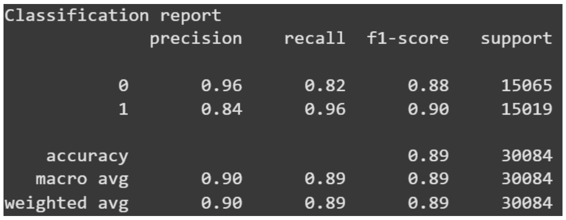
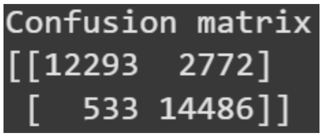
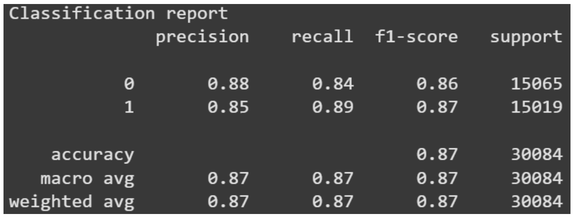
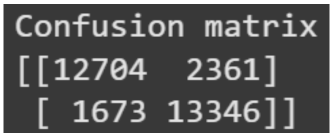


```
#Compare model function

def compareModel(x_train, x_test, y_train, y_test):
  decisionTree(x_train, x_test, y_train, y_test)
  print('-'*100)
  randomForest(x_train, x_test, y_train, y_test)
  print('-'*100)
  KNN(x_train, x_test, y_train, y_test)
  print('-'*100)
  xgboost(x_train, x_test, y_train, y_test)
```

```
compareModel(x_train, x_test, y_train, y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	 Classification report precision recall f1-score support 0 0.95 0.94 0.95 15065 1 0.94 0.95 0.95 15019 accuracy 0.95 30084 macro avg 0.95 0.95 0.95 30084 weighted avg 0.95 0.95 0.95 30084	95%	 Confusion matrix [[14207 858] [782 14237]]
Decision Tree	 Classification report precision recall f1-score support 0 0.94 0.92 0.93 15065 1 0.92 0.94 0.93 15019 accuracy 0.93 30084 macro avg 0.93 0.93 0.93 30084 weighted avg 0.93 0.93 0.93 30084	93%	 Confusion matrix [[13861 1204] [885 14134]]
KNN	 Classification report precision recall f1-score support 0 0.96 0.82 0.88 15065 1 0.84 0.96 0.90 15019 accuracy 0.89 30084 macro avg 0.90 0.89 0.89 30084 weighted avg 0.90 0.89 0.89 30084	89%	 Confusion matrix [[12293 2772] [533 14486]]
XG Boost	 Classification report precision recall f1-score support 0 0.88 0.84 0.86 15065 1 0.85 0.89 0.87 15019 accuracy 0.87 30084 macro avg 0.87 0.87 0.87 30084 weighted avg 0.87 0.87 0.87 30084	87%	 Confusion matrix [[12704 2361] [1673 13346]]