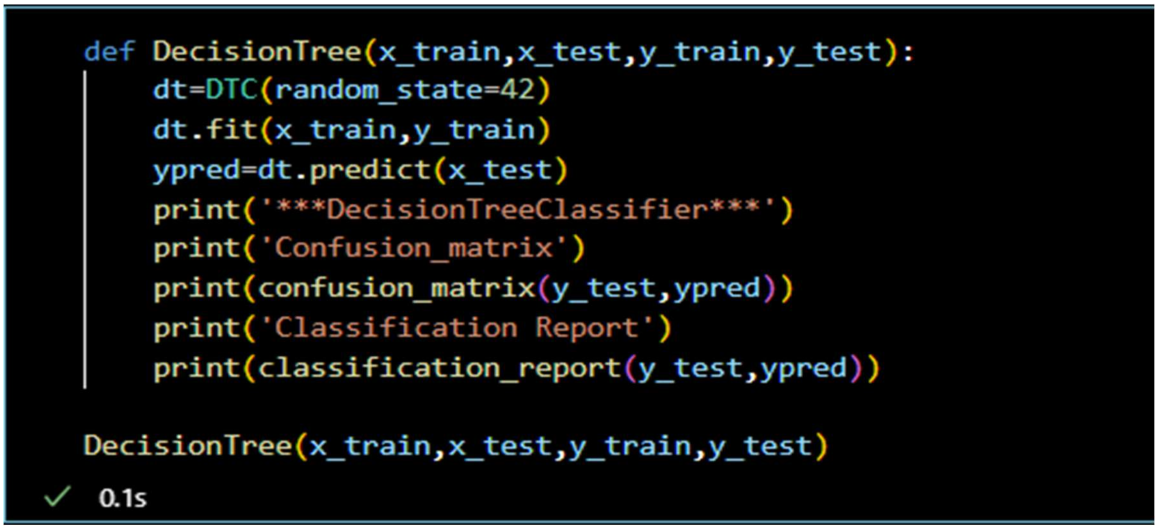
Model Development Phase Template

Date	11 July 2024
Team ID	xxxxxx
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.

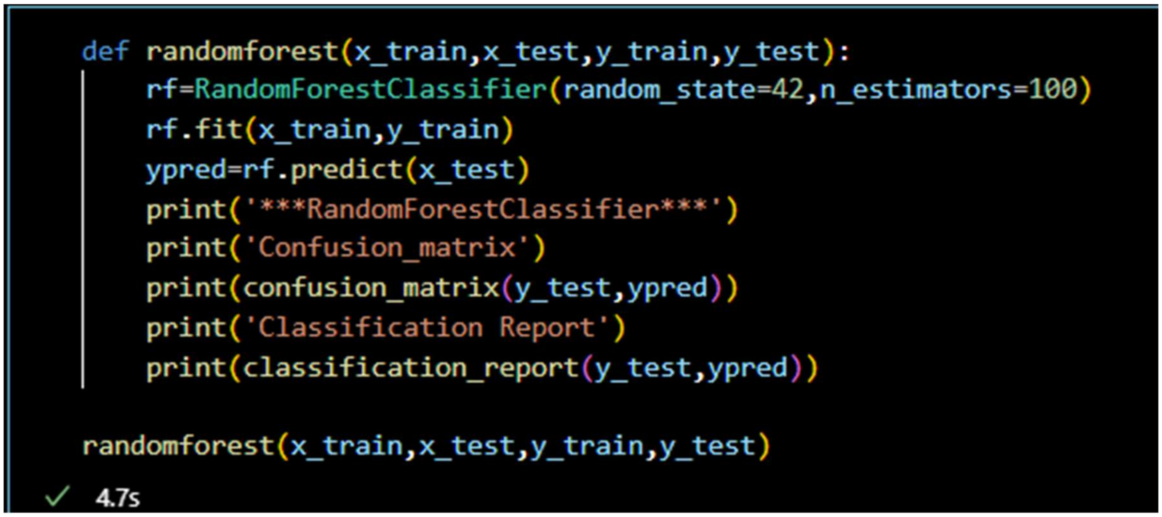
Initial Model Training Code: S



```
def DecisionTree(x_train,x_test,y_train,y_test):
    dt=DTC(random_state=42)
    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))

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

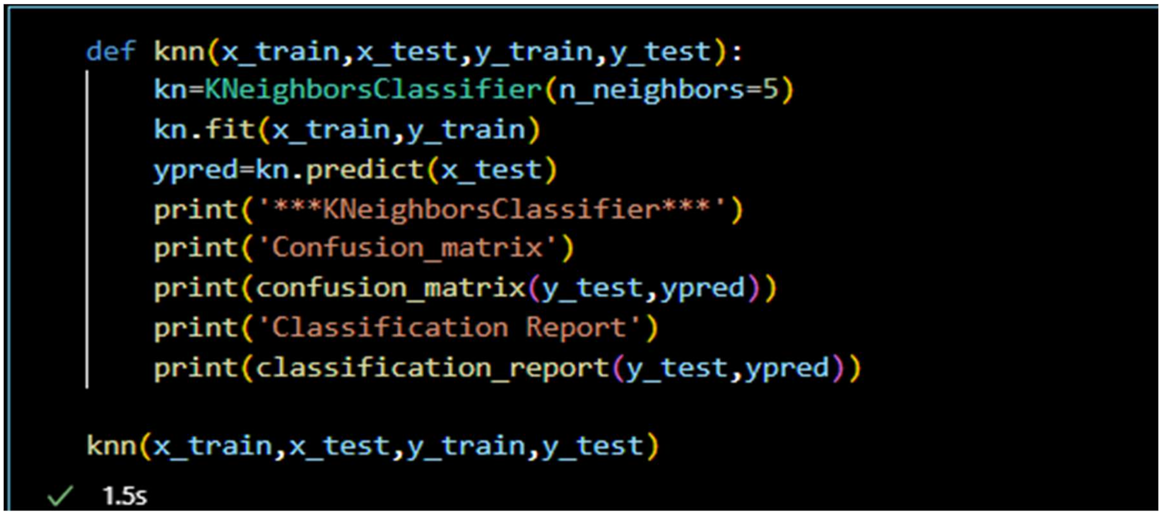
✓ 0.1s



```
def randomforest(x_train,x_test,y_train,y_test):
    rf=RandomForestClassifier(random_state=42,n_estimators=100)
    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))

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

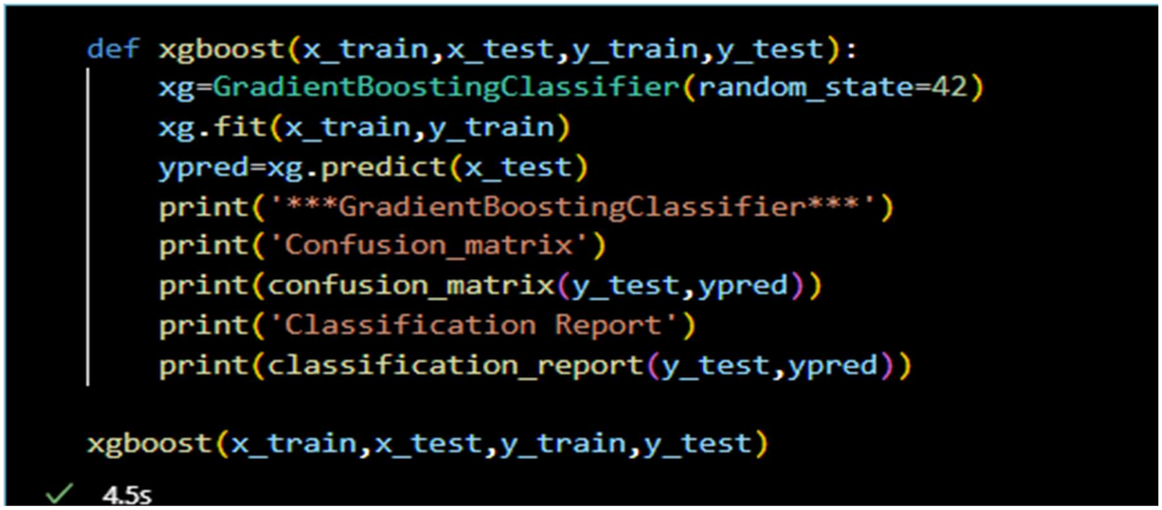
✓ 4.7s



```
def knn(x_train,x_test,y_train,y_test):
    kn=KNeighborsClassifier(n_neighbors=5)
    kn.fit(x_train,y_train)
    ypred=kn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion_matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification Report')
    print(classification_report(y_test,ypred))

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

✓ 1.5s

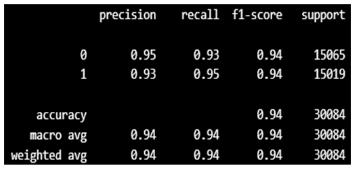
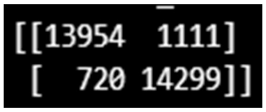
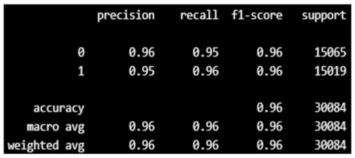
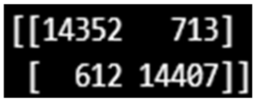
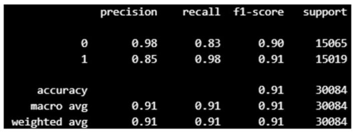
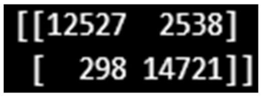
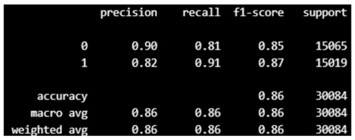
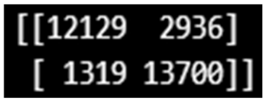


```
def xgboost(x_train,x_test,y_train,y_test):
    xg=GradientBoostingClassifier(random_state=42)
    xg.fit(x_train,y_train)
    ypred=xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion_matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification Report')
    print(classification_report(y_test,ypred))

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

✓ 4.5s

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
DecisionTree Classifier	 precision recall f1-score support 0 0.95 0.93 0.94 15065 1 0.93 0.95 0.94 15019 accuracy 0.94 30084 macro avg 0.94 0.94 0.94 30084 weighted avg 0.94 0.94 0.94 30084	93.913%	 [[13954 1111] [720 14299]]
RandomForestClassifier	 precision recall f1-score support 0 0.96 0.95 0.96 15065 1 0.95 0.96 0.96 15019 accuracy 0.96 30084 macro avg 0.96 0.96 0.96 30084 weighted avg 0.96 0.96 0.96 30084	95.595%	 [[14352 713] [612 14407]]
KNeighborsClassifier	 precision recall f1-score support 0 0.98 0.83 0.90 15065 1 0.85 0.98 0.91 15019 accuracy 0.91 30084 macro avg 0.91 0.91 0.91 30084 weighted avg 0.91 0.91 0.91 30084	90.573%	 [[12527 2538] [298 14721]]
GradientBoostingClassifier	 precision recall f1-score support 0 0.90 0.81 0.85 15065 1 0.82 0.91 0.87 15019 accuracy 0.86 30084 macro avg 0.86 0.86 0.86 30084 weighted avg 0.86 0.86 0.86 30084	85.856%	 [[12129 2936] [1319 13700]]