

Model Development Phase Template

Date	12 July 2024
Team ID	SWTID1720174957
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)  
    y_pred=dt.predict(x_test)  
    print("DecisionTreeClassifier")  
    print('Confusion matrix')  
    print(confusion_matrix(y_test,y_pred))  
    print('Classification report')  
    print(classification_report(y_test,y_pred))  
    return y_pred
```

Random Forest:-

```
def randomForest(x_train,x_test,y_train,y_test):  
    rf=RandomForestClassifier()  
    rf.fit(x_train,y_train)  
    y_pred=rf.predict(x_test)  
    print("RandomForestClassifier")  
    print('Confusion matrix')  
    print(confusion_matrix(y_test,y_pred))  
    print('Classification report')  
    print(classification_report(y_test,y_pred))  
    return y_pred
```

KNN:-

```
from sklearn.neighbors import KNeighborsClassifier  
def KNN(x_train,x_test,y_train,y_test):  
    knn=KNeighborsClassifier()  
    knn.fit(x_train,y_train)  
    y_pred=knn.predict(x_test)  
    print("KNeighboursClassifier")  
    print('Confusion matrix')  
    print(confusion_matrix(y_test,y_pred))  
    print('Classification report')  
    print(classification_report(y_test,y_pred))  
    return y_pred
```

Xgboost:-

```
def xgboost(x_train,x_test,y_train,y_test):  
    xg=GradientBoostingClassifier()  
    xg.fit(x_train,y_train)  
    y_pred=xg.predict(x_test)  
    print("GradientBoostingClassifier")  
    print('Confusion matrix')  
    print(confusion_matrix(y_test,y_pred))  
    print('Classification report')  
    print(classification_report(y_test,y_pred))  
    return y_pred
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decision Tree	<pre>classification_report(y_test,y_pred)</pre> <pre> Classification report precision recall f1-score support 0 0.95 0.93 0.94 10035 1 0.93 0.95 0.94 10021 accuracy 0.94 0.94 0.94 20056 macro avg 0.94 0.94 0.94 20056 weighted avg 0.94 0.94 0.94 20056 </pre>	93.73%	<pre>confusion_matrix(y_test,y_pred)</pre> <pre> array([[9289, 746], [510, 9511]], dtype=int64) </pre>
random Forest	<pre>classification_report(y_test,y_pred)</pre> <pre> Classification report precision recall f1-score support 0 0.95 0.95 0.95 10035 1 0.95 0.95 0.95 10021 accuracy 0.95 0.95 0.95 20056 macro avg 0.95 0.95 0.95 20056 weighted avg 0.95 0.95 0.95 20056 </pre>	94.94%	<pre>confusion_matrix(y_test,y_pred)</pre> <pre> array([[9498, 537], [477, 9544]], dtype=int64) </pre>
KNN	<pre>classification_report(y_test,y_pred)</pre> <pre> Classification report precision recall f1-score support 0 0.96 0.82 0.89 10035 1 0.84 0.97 0.90 10021 accuracy 0.90 0.90 0.90 20056 macro avg 0.90 0.90 0.89 20056 weighted avg 0.90 0.90 0.89 20056 </pre>	89.5%	<pre>confusion_matrix(y_test,y_pred)</pre> <pre> array([[8242, 1793], [308, 9713]], dtype=int64) </pre>

xgboost	<code>classification_report(y_test,y_pred)</code>					86.43%	<code>confusion_matrix(y_test,y_pred)</code> array([[8409, 1626], [1094, 8927]], dtype=int64)
	Classification report						
		precision	recall	f1-score	support		
	0	0.88	0.84	0.86	10035		
	1	0.85	0.89	0.87	10021		
	accuracy			0.86	20056		
	macro avg	0.87	0.86	0.86	20056		
	weighted avg	0.87	0.86	0.86	20056		