

Model Development Phase Template

| | |
|---------------|-----------------------------|
| Date | 5th July 2024 |
| Team ID | 739687 |
| Project Title | Workforce Retention System. |
| 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:

RandomForest Classifier::

Model Building With Random Forest Classifier

```
from sklearn.ensemble import Random ForestClassifier
```

```
ran = Random ForestClassifier (criterion = 'entropy', random_state = 0)
```

```
ran
```



Random ForestClassifier

```
RandomForestClassifier (criterion='entropy', random_state=0)
```

```
ran.fit(X_train,y_train)
```

Random ForestClassifier

```
RandomForestClassifier (criterion='entropy', random_state=0)
```

```
y_train_pred ran.predict(X_train)
```

DecisionTree Classifier:

Model Building With Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

deci = DecisionTreeClassifier (criterion = 'entropy', random_state = 0)

deci.fit(X_train, y_train)

DecisionTreeClassifier

DecisionTreeClassifier (criterion='entropy', random_state=0)

y_train_pred deci.predict(X_train)

y_test_pred deci.predict(X_test)

#Confusion Matrix For Training Data With Decision Tree

confusion_matrix(y_train, y_train_pred)

array([[9134, 0],
       [ 0, 2865]], dtype=int64)
```

Model Building with Support Vector Machine::

```
from sklearn.svm import SVC

svm = SVC(kernel = 'linear', random_state = 0)

svm.fit(X_train, y_train)

SVC

SVC(kernel='linear', random_state=0)

y_train_pred deci.predict(X_train)

y_test_pred deci.predict(X_test)
```

Model Validation and Evaluation Report:

| Model | Classification Report | Accuracy | Confusion Matrix |
|-----------------------------|--|----------|------------------|
| Random Forest Regressor | <pre> Model Building With Random Forest Classifier from sklearn.ensemble import RandomForestClassifier ran = RandomForestClassifier(criterion = 'entropy', random_state = 0) ran # RandomForestClassifier RandomForestClassifier(criterion='entropy', random_state=0) ran.fit(X_train,y_train) # RandomForestClassifier RandomForestClassifier(criterion='entropy', random_state=0) y_train_pred = ran.predict(X_train) 2 . Random Forest -->Traing Accuracy = 100.0 -->Test Accuracy = 98.8 </pre> | 98% | - |
| Decision Tree Regressor | <pre> Model Building With Decision Tree from sklearn.tree import DecisionTreeClassifier deci = DecisionTreeClassifier(criterion = 'entropy', random_state = 0) deci.fit(X_train,y_train) # DecisionTreeClassifier DecisionTreeClassifier(criterion='entropy', random_state=0) y_train_pred = deci.predict(X_train) y_test_pred = deci.predict(X_test) #Confusion Matrix For Training Data With Decision Tree 3. Decision Tree -->Traing Accuracy = 100.0 -->Test Accuracy = 97.73333333333333 </pre> | 97% | - |
| Gradient Boosting Regressor | <pre> Model Building with Support Vector Machine from sklearn.svm import SVC svm = SVC(kernel = 'linear', random_state = 0) svm.fit(X_train, y_train) # SVC SVC(kernel='linear', random_state=0) y_train_pred = deci.predict(X_train) y_test_pred = deci.predict(X_test) 4. Support Vector Machine -->Training Accuracy = 100.0 -->Testing Accuracy = 97.73333333333333 </pre> | 97% | - |