

|  |  |  |
| --- | --- | --- |
|  | **Model Development Phase Template** | |
|  |  |  |
| **Date** |  | **5th July 2024** |
|  |  |  |
| **Team ID** |  | **740071** |
|  |  |  |
| **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)**



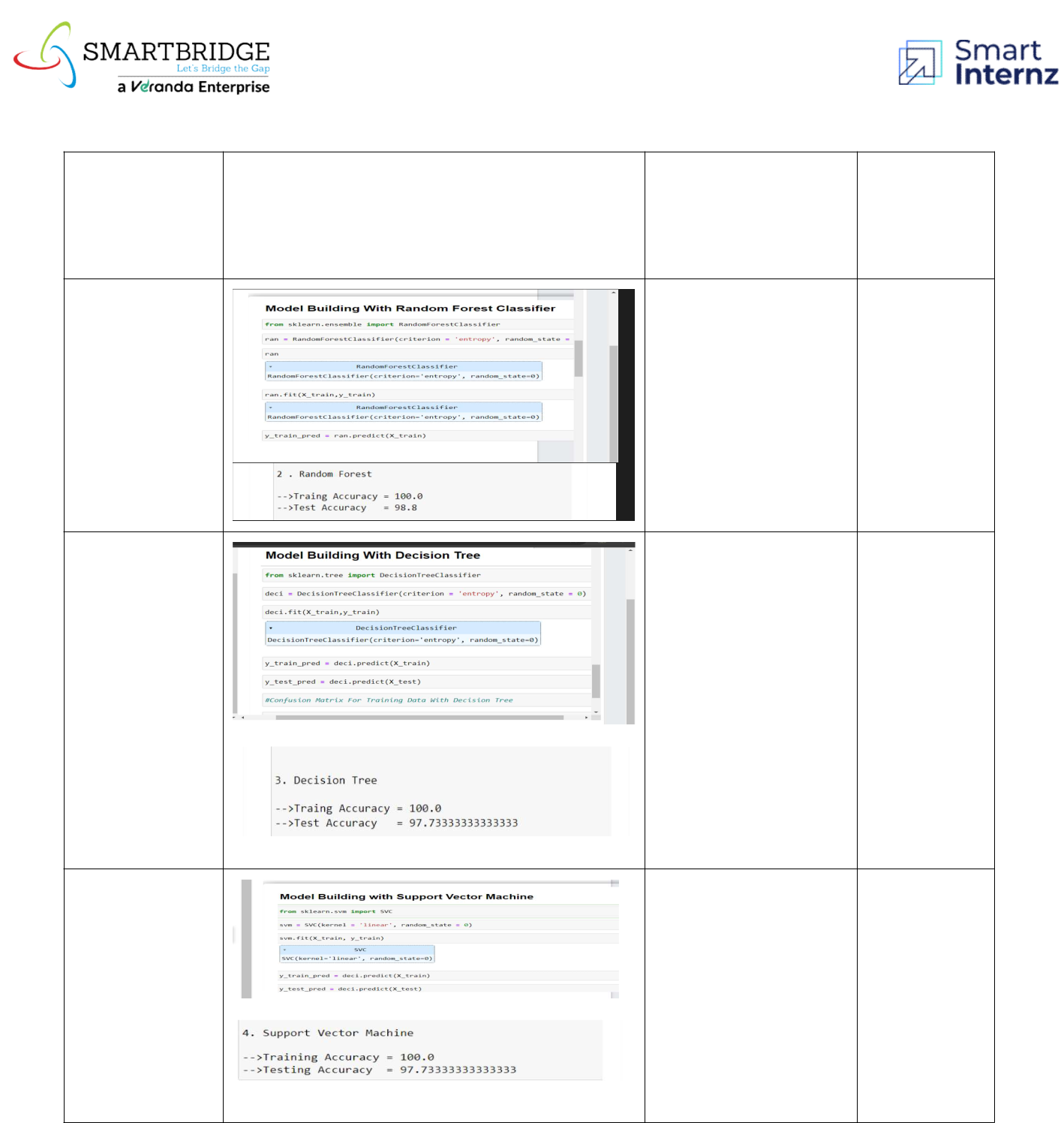
**y\_train\_pred deci.predict(X\_train)**



**y\_test\_pred deci.predict(X\_test)**



**Model Validation and Evaluation Report:**



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Confusion** |
| **Model** | **Classification Report** | **Accuracy** | **Matrix** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Random** |  |  |  |
| **Forest** | **98%** | **-** |  |
|  |  |  |
| **Regressor** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Decision** |  |  |  |
| **Tree** | **97%** | **-** |  |
|  |  |  |
| **Regressor** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Gradient** |  |  |  |
| **Boosting** | **97%** | **-** |  |
|  |  |  |
| **Regressor** |  |  |  |