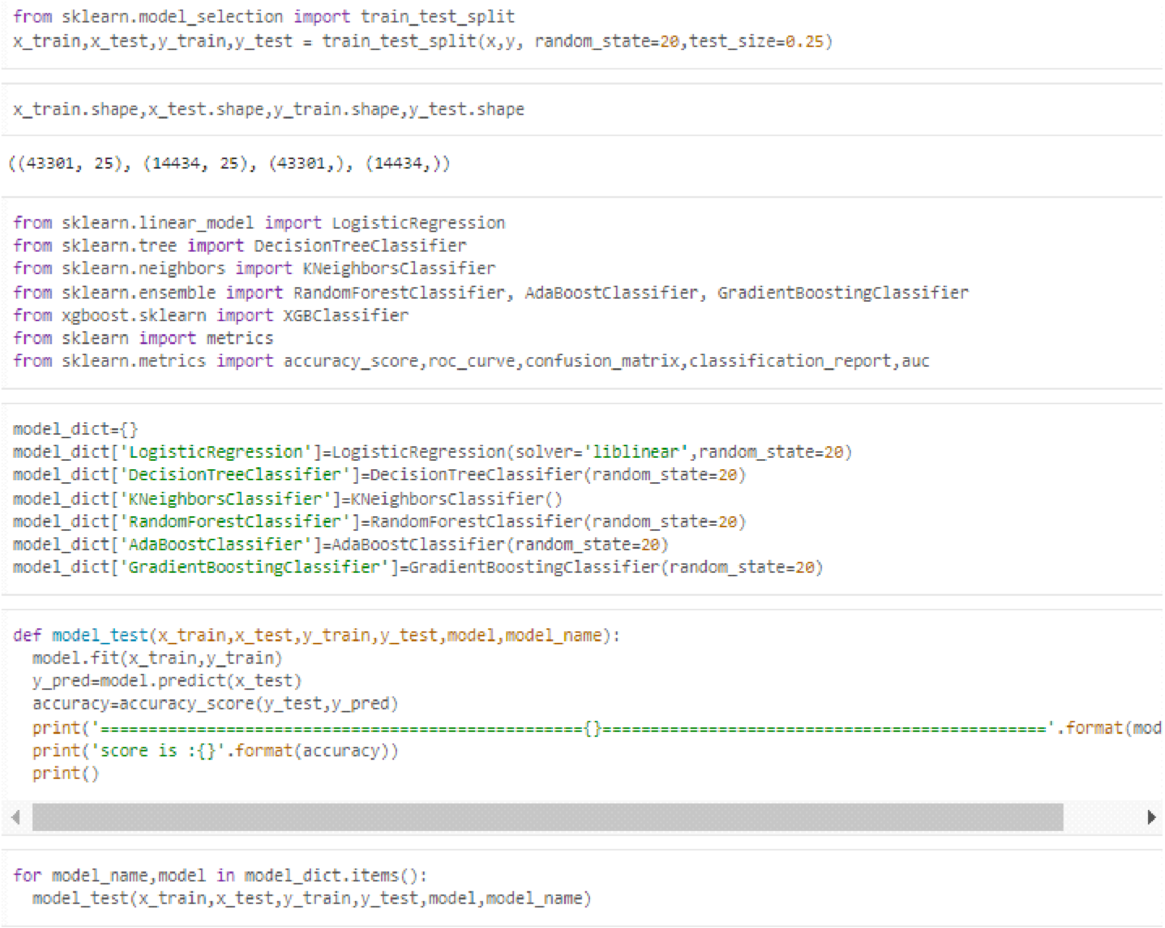
Model Development Phase Template

Date	June 22,2024
Team ID	team-740066
Project Title	Hospital Readmission Prediction 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:



```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y, random_state=20,test_size=0.25)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((43301, 25), (14434, 25), (43301,), (14434,))
```

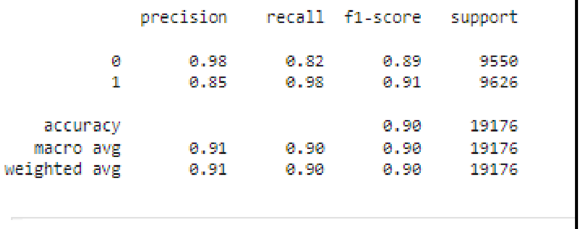
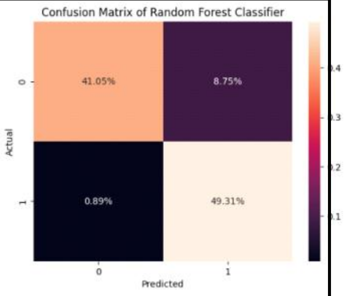
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from xgboost.sklearn import XGBClassifier
from sklearn import metrics
from sklearn.metrics import accuracy_score,roc_curve,confusion_matrix,classification_report,auc
```

```
model_dict={}
model_dict['LogisticRegression']=LogisticRegression(solver='liblinear',random_state=20)
model_dict['DecisionTreeClassifier']=DecisionTreeClassifier(random_state=20)
model_dict['KNeighborsClassifier']=KNeighborsClassifier()
model_dict['RandomForestClassifier']=RandomForestClassifier(random_state=20)
model_dict['AdaBoostClassifier']=AdaBoostClassifier(random_state=20)
model_dict['GradientBoostingClassifier']=GradientBoostingClassifier(random_state=20)
```

```
def model_test(x_train,x_test,y_train,y_test,model,model_name):
  model.fit(x_train,y_train)
  y_pred=model.predict(x_test)
  accuracy=accuracy_score(y_test,y_pred)
  print('=================================================={}=============================================='.format(mod
  print('score is :{}'.format(accuracy))
  print()
```

```
for model_name,model in model_dict.items():
  model_test(x_train,x_test,y_train,y_test,model,model_name)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	 precision recall f1-score support 0 0.98 0.82 0.89 9550 1 0.85 0.98 0.91 9626 accuracy 0.90 19176 macro avg 0.91 0.90 0.90 19176 weighted avg 0.91 0.90 0.90 19176	93%	
Decision Tree	-	87%	-
KNN	-	79%	-
Gradient Boosting	-	90%	-