Model Development Phase Template

Date	15 July 2024
Team ID	739901
Project Title	Thyroid disease classification using machine learning
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

This 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**

nbviewer.org.

In [ ]:
```
y_train_pred = log.predict(x_train)
y_test_pred = log.predict(x_test)
```

In [ ]:
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
```

In [ ]:
```
train_accuracy = accuracy_score(y_train,y_train_pred)
```

main sepsis_survival / Mini_project (2).ipynb ↑ Top

view Code Blame Raw

```
etricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted sam
ples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]:
```
test_accuracy = accuracy_score(y_test,y_test_pred)
test_precision = precision_score(y_test,y_test_pred,average='weighted')
test_recall = recall_score(y_test,y_test_pred,average='weighted')
test_f1score = f1_score(y_test,y_test_pred,average='weighted')
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedM
etricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted sam
ples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]:
```
print("log train accuracy:",train_accuracy)
print("log test accuracy:",test_accuracy)
print("log train precision:",train_precision)
print("log test precision:",test_precision)
print("log train recall:",train_recall)
print("log test recall:",test_recall)
print("log train f1score:",train_f1score)
print("log test f1score:",test_f1score)
```

In [ ]:
```
#checking the accuracy of the model using KNN classifier
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5,p=2)
```

In [ ]:
```
knn.fit(x_train,y_train)
```

Out[ ]:
```
KNeighborsClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with

main sepsis_survival / Mini_project (2).ipynb ↑ Top

ew Code Blame Raw

In [ ]:
```
train_accuracy = accuracy_score(y_train,y_train_pred)
train_precision = precision_score(y_train,y_train_pred,average='weighted')
train_recall = recall_score(y_train,y_train_pred,average='weighted')
train_f1score = f1_score(y_train,y_train_pred,average='weighted')
```

In [ ]:
```
test_accuracy = accuracy_score(y_test,y_test_pred)
test_precision = precision_score(y_test,y_test_pred,average='weighted')
test_recall = recall_score(y_test,y_test_pred,average='weighted')
test_f1score = f1_score(y_test,y_test_pred,average='weighted')
```

In [ ]:
```
print("knn train accuracy:",train_accuracy)
print("knn test accuracy:",test_accuracy)
print("knn train precision:",train_precision)
print("knn test precision:",test_precision)
print("knn train recall:",train_recall)
print("knn test recall:",test_recall)
print("knn train f1score:",train_f1score)
print("knn test f1score:",test_f1score)
```

SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

```
In [ ]: #checking the accuracy of the model using decision tree classifier
        from sklearn.tree import DecisionTreeClassifier
        dt = DecisionTreeClassifier(criterion='entropy',max_depth=3)

In [ ]: dt.fit(x_train,y_train)

Out[ ]: DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [ ]: y_train_pred = dt.predict(x_train)
        y_test_pred = dt.predict(x_test)

In [ ]: train_accuracy = accuracy_score(y_train,y_train_pred)
        train_precision = precision_score(y_train,y_train_pred,average='weighted')
```

main sepsis_survival / Mini_project (2).ipynb ↑ Top

view Code Blame Raw

```
    _warn_prf(average, modifier, msg_start, len(result))

In [ ]: test_accuracy = accuracy_score(y_test,y_test_pred)
        test_precision = precision_score(y_test,y_test_pred,average='weighted')
        test_recall = recall_score(y_test,y_test_pred,average='weighted')
        test_f1score = f1_score(y_test,y_test_pred,average='weighted')

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedM
etricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted sam
ples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

In [ ]: print("dt train accuracy:",train_accuracy)
        print("dt test accuracy:",test_accuracy)
        print("dt train precision:",train_precision)
        print("dt test precision:",test_precision)
        print("dt train recall:",train_recall)
        print("dt test recall:",test_recall)
        print("dt train f1score:",train_f1score)
        print("dt test f1score:",test_f1score)

In [ ]: #checking accuracy using randomforest classifier
        from sklearn.ensemble import RandomForestClassifier
        rf= RandomForestClassifier(n_estimators=2,max_depth=2,criterion='entropy',bootstrap= T

In [ ]: rf.fit(x_train,y_train)

Out[ ]: RandomForestClassifier(criterion='entropy', max_depth=2, n_estimators=2)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [ ]: y_train_pred = rf.predict(x_train)
        y_test_pred = rf.predict(x_test)

In [ ]: train_accuracy = accuracy_score(y_train,y_train_pred)
        train_precision = precision_score(y_train,y_train_pred,average='weighted')
        train_recall = recall_score(y_train,y_train_pred,average='weighted')
        train_f1score = f1_score(y_train,y_train_pred,average='weighted')

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: Undefin
etricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted
ples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

In [ ]: test_accuracy = accuracy_score(y_test,y_test_pred)
        test_precision = precision_score(y_test,y_test_pred,average='weighted')
        test_recall = recall_score(y_test,y_test_pred,average='weighted')
```

main sepsis_survival / Mini_project (2).ipynb

view Code Blame Raw

```
    _warn_prf(average, modifier, msg_start, len(result))

In [ ]: print("rf train accuracy:",train_accuracy)
        print("rf test accuracy:",test_accuracy)
        print("rf train precision:",train_precision)
        print("rf test precision:",test_precision)
        print("rf train recall:",train_recall)
        print("rf test recall:",test_recall)
        print("rf train f1score:",train_f1score)
        print("rf test f1score:",test_f1score)
```

Model Valuation And Evalution Report

Model	Classification Report	F1 Score
KNN	knn train accuracy: 0.9126849117458935 knn test accuracy: 0.9129012796494459 knn train precision: 0.8615120571673442 knn test precision: 0.8634128153243682 knn train recall: 0.9126849117458935 knn test recall: 0.9129012796494459	88%
Random Forest	rf train accuracy: 0.9232590737741729 rf test accuracy: 0.9226494776617026 rf train precision: 0.8524073173063437 rf test precision: 0.8512820586294125 rf train recall: 0.9232590737741729 rf test recall: 0.9226494776617026	89%
Decision Tree	dt train accuracy: 0.9232590737741729 dt test accuracy: 0.9226494776617026 dt train precision: 0.8524073173063437 dt test precision: 0.8512820586294125 dt train recall: 0.9232590737741729 dt test recall: 0.9226494776617026	89%
Logistic Regression	log train accuracy: 0.9232590737741729 log test accuracy: 0.9226494776617026 log train precision: 0.8524073173063437 log test precision: 0.8512820586294125 log train recall: 0.9232590737741729 log test recall: 0.9226494776617026	89%