Model Development Phase Template

Date	9th July 2024
Team ID	SWTID1720080033
Project Title	Anemia Sense: Leveraging Machine Learning For Precise Anemia Recognitions
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:

Splitting Data Into Train and Test

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=20)

print(x_train.shape)

print(x_test.shape)

print(y_train.shape)

print(y_test.shape)
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
logistic_regression=LogisticRegression()
logistic_regression.fit(x_train,y_train)
pred=logistic_regression.predict(x_test)
pred
acc_lr=accuracy_score(y_test,pred)
confusion_matrix(y_test,pred)
acc_lr=accuracy_score(y_test,pred)
c_lr=classification_report(y_test,pred)
print('Accuracy Score: ',acc_lr)
print(c_lr)
```

Random Forest Model

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
rf.fit(x_train,y_train)
pred=rf.predict(x_test)
pred
y_test
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
acc_rf=accuracy_score(y_test,pred)
conmat=confusion_matrix(y_test,pred)
print(acc_rf)
print(conmat)
c_rf=classification_report(y_test,pred)
c_rf
```

Decision tree

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(x_train, y_train)
y_pred = decision_tree_model.predict(x_test)
y_pred
y_test
acc_dt = accuracy_score(y_test, y_pred)
acc_dt
conmat = confusion_matrix(y_test, y_pred)
conmat
c_dt = classification_report(y_test, y_pred)
print(c_dt)
```

Naive Bayes Model

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train,y_train)
pred = nb.predict(x_test)
acc_nb=accuracy_score(pred,y_test)
c_nb=classification_report(pred,y_test)
print("Accuracy Score: ",acc_nb)
print(c_nb)
```

Support Vector Machine

```
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn import metrics
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
support_vector=SVC()
support_vector.fit(x_train,y_train)
y_pred=support_vector.predict(x_test)
acc_svc=metrics.accuracy_score(y_test,y_pred)
c_svc=metrics.classification_report(y_test,y_pred)
print("accuracy_score: ",acc_svc)
print("classification_report: ")
print(c_svc)
```

Gradient Boosting Classifier

```
from sklearn.ensemble import GradientBoostingClassifier
GBC= GradientBoostingClassifier()
GBC.fit(x_train,y_train)
y_pred2=GBC.predict(x_test)
acc_gbc=metrics.accuracy_score(y_test,y_pred2)
c_gbc=metrics.classification_report(y_test,y_pred2)
print("accuracy_score: ",acc_gbc)
print("classification_report: ")
print(c_gbc)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Logistic Regression	Accuracy Score: 1.0 precision recall f1-score support 0 1.00 1.00 1.00 167 1 1.00 1.00 1.00 118 accuracy 1.00 285 macro avg 1.00 1.00 1.00 285 weighted avg 1.00 1.00 1.00 285	1.0	array([[167, 0], [0, 118]], dtype=int64)
Random Forest Tree	c_lr=classification_report(y_test,pred) print(c_lr) precision recall f1-score support 0 0.97 0.93 0.95 167 1 0.90 0.96 0.93 118 accuracy 0.94 285 macro avg 0.94 0.94 0.94 285 weighted avg 0.94 0.94 0.94 285	1.0	array([[155, 12], [5, 113]], dtype=int64)
Decision Tree	precision recall f1-score support 0 1.00 1.00 1.00 167 1 1.00 1.00 1.00 118 accuracy 1.00 285 macro avg 1.00 1.00 1.00 285 weighted avg 1.00 1.00 1.00 285	1.0	array([[167, 0], [0, 118]], dtype=int64)
Naïve Bayes Model	precision recall f1-score support 0 0.93 0.97 0.95 160 1 0.96 0.90 0.93 125 accuracy 0.94 285 macro avg 0.94 0.94 0.94 285 weighted avg 0.94 0.94 0.94 285	0.9403508771929825	array([[155, 12], [5, 113]], dtype=int64)

Support Vector Machine	classification_report: precision recall f1-score support 0 1.00 0.99 1.00 167 1 0.99 1.00 1.00 118 accuracy 1.00 285 macro avg 1.00 1.00 1.00 285 weighted avg 1.00 1.00 1.00 285	0.9964912280701754	conmat = confusion_matrix(y_test, pred) conmat array([[155, 12], [5, 113]], dtype=int64)
Gradient Boosting Classifier	classification_report: precision recall f1-score support 0 1.00 1.00 1.00 167 1 1.00 1.00 1.00 118 accuracy 1.00 285 macro avg 1.00 1.00 1.00 285 weighted avg 1.00 1.00 1.00 285	1.0	conmat = confusion_matrix(y_test, y_pred2) conmat array([[167, 0], [0, 118]], dtype=int64)