Model Development Phase Template

Date	Nov 30,2024
Team ID	739891
Project Title	Unlocking the Minds: Analyzing Mental Health with NLP
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 screenshot

Initial model Training Code:

[25]:
```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score
sv = SVC()
sv.fit(x_train,y_train)
y_pred = sv.predict(x_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.94      0.91      4271
           1       0.93      0.88      0.91      4121

    accuracy                           0.91      8392
   macro avg       0.91      0.91      0.91      8392
weighted avg       0.91      0.91      0.91      8392
```

[26]:
```
sv_acc = accuracy_score(y_test,y_pred)
sv_acc
```

[26]:
```
0.9101525262154433
```

[27]:
```
from sklearn.tree import DecisionTreeClassifier
Dt = DecisionTreeClassifier()
Dt.fit(x_train,y_train)
y_pred = Dt.predict(x_test)
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.83      0.82      4271
           1       0.82      0.81      0.82      4121

    accuracy                           0.82      8392
```

```
              0       0.82      0.83      0.82      4271
              1       0.82      0.81      0.82      4121

       accuracy                           0.82      8392
      macro avg       0.82      0.82      0.82      8392
   weighted avg       0.82      0.82      0.82      8392

0.819351763584366
```

[28]:
```
dt_acc = accuracy_score(y_test,y_pred)
dt_acc
```

[28]:
```
0.819351763584366
```

[29]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
y_pred = rf.predict(x_test)
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.91      0.89      4271
           1       0.90      0.86      0.88      4121

    accuracy                           0.89      8392
   macro avg       0.89      0.88      0.89      8392
weighted avg       0.89      0.89      0.89      8392

0.8852478551000953
```

[30]:
```
Rf_acc = accuracy_score(y_test,y_pred)
Rf_acc
```

[30]:
```
0.8852478551000953
```

[31]:
```
from sklearn.ensemble import AdaBoostClassifier
# clf = AdaBoostClassifier(algorithm='SAMME')

ab = AdaBoostClassifier(algorithm='SAMME')
ab.fit(x_train,y_train)
y_pred = ab.predict(x_test)
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.89      0.84      4271
           1       0.87      0.76      0.81      4121

    accuracy                           0.83      8392
   macro avg       0.83      0.83      0.83      8392
weighted avg       0.83      0.83      0.83      8392

0.8286463298379408
```

[32]:
```
ab_acc = accuracy_score(y_test,y_pred)
ab_acc
```

[32]:
```
0.8286463298379408
```

[33]:
```
from sklearn.ensemble import GradientBoostingClassifier
gb = GradientBoostingClassifier()
gb.fit(x_train,y_train)
y_pred = gb.predict(x_test)
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.92      0.87      4271
           1       0.90      0.81      0.85      4121

    accuracy                           0.86      8392
   macro avg       0.87      0.86      0.86      8392
weighted avg       0.87      0.86      0.86      8392

0.861415633937083
```

[34]:
```
gb_acc = accuracy_score(y_test,y_pred)
gb_acc
```

[34]:
```
0.861415633937083
```

[35]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train,y_train)
y_pred = lr.predict(x_test)
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.94      0.91      4271
           1       0.93      0.88      0.90      4121

    accuracy                           0.91      8392
   macro avg       0.91      0.91      0.91      8392
weighted avg       0.91      0.91      0.91      8392

0.9074118207816969
```

[36]:
```
lr_acc = accuracy_score(y_test,y_pred)
lr_acc
```

[36]:
```
0.9074118207816969
```