Project Development Phase
Model Performance Test

Date	20nd November 2023
Team ID	592988
Project Name	Disease Prediction Using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE -, MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuracy Score- & Classification Report -	GIVEN BELOW
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	GIVEN BELOW

CONFUSION MATRIX-

```
from sklearn.metrics import confusion_matrix


new = X.drop(to_drop, axis=1)
y_new = y
X1_train, X1_val, y1_train, y1_val = train_test_split(new, y_new, test_size=0.2)
X1_test = X_test.drop(to_drop, axis=1)
y1_test = y_test


knn_new = KNeighborsClassifier()
knn_new.fit(X1_train, y1_train)

y1_pred = knn_new.predict(X1_test)

conf_matrix = confusion_matrix(y1_test, y1_pred)
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

Classification Report-

```
from sklearn.metrics import classification_report

# Assuming you already have your knn_new model trained on X1_train and y1_train
# Now, let's make predictions on the validation set (X1_val)
y1_val_pred = knn_new.predict(X1_val)

# Generate the classification report
report = classification_report(y1_val, y1_val_pred)

# Print the classification report
print(report)
```

```
                                    precision    recall  f1-score   support

(vertigo) Paroymsal  Positional Vertigo       1.00      1.00      1.00        25
                                AIDS       0.90      0.95      0.92        19
                                Acne       1.00      1.00      1.00        22
                Alcoholic hepatitis       1.00      1.00      1.00        21
                             Allergy       1.00      1.00      1.00        23
                           Arthritis       1.00      0.89      0.94        18
                  Bronchial Asthma       1.00      1.00      1.00        26
              Cervical spondylosis       1.00      1.00      1.00        31
                         Chicken pox       1.00      1.00      1.00        22
               Chronic cholestasis       1.00      1.00      1.00        23
                       Common Cold       1.00      1.00      1.00        20
```

```
                         Hepatitis B       1.00      1.00      1.00        28
                         Hepatitis C       1.00      1.00      1.00        18
                         Hepatitis D       1.00      1.00      1.00        26
                         Hepatitis E       1.00      1.00      1.00        24
                        Hypertension       1.00      1.00      1.00        19
                       Hyperthyroidism     1.00      1.00      1.00        24
                          Hypoglycemia     1.00      1.00      1.00        29
                       Hypothyroidism      1.00      1.00      1.00        25
                             Impetigo      1.00      0.91      0.95        22
                             Jaundice      1.00      1.00      1.00        24
                              Malaria      1.00      1.00      1.00        26
                             Migraine      1.00      1.00      1.00        25
                       Osteoarthristis     1.00      1.00      1.00        29
              Paralysis (brain hemorrhage)  1.00      1.00      1.00        18
                 Peptic ulcer diseae       1.00      1.00      1.00        23
                           Pneumonia       1.00      1.00      1.00        23
                           Psoriasis       1.00      1.00      1.00        18
                        Tuberculosis       1.00      1.00      1.00        31
                             Typhoid       1.00      1.00      1.00        19
          Urinary tract infection          0.81      1.00      0.90        22
                   Varicose veins          1.00      1.00      1.00        31
                      hepatitis A          1.00      1.00      1.00        27

                         accuracy                               0.99       984
                        macro avg         0.99      0.99      0.99        984
                     weighted avg         0.99      0.99      0.99        984
```

Model Summary-

```python
from keras.models import Sequential
from keras.layers import Dense

# Example neural network model
model = Sequential()
model.add(Dense(64, input_shape=(X1_train.shape[1],), activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compile the model (you may need to compile the model before using model.summary())
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Print the model summary
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                3648

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 1)                 33
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                3648

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 5761 (22.50 KB)
Trainable params: 5761 (22.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Accuracy Testing-

```
knn_new = KNeighborsClassifier()
knn_new.fit(X1_train, y1_train)
```

```
KNeighborsClassifier
KNeighborsClassifier()
```

+ Code + Text

```
[101] y_pred = knn_new.predict(X1_val)
      yt_pred = knn_new.predict(X1_train)
      y_pred1 = knn_new.predict(X1_test)
      print('The Training Accuracy of the algorithm is ', accuracy_score(y1_train, yt_pred))
      print('The Validation Accuracy of the algorithm is ', accuracy_score(y1_val, y_pred))
      print('The Testing Accuracy of the algorithm is', accuracy_score(y1_test, y_pred1))
```

```
The Training Accuracy of the algorithm is  0.9923780487804879
The Validation Accuracy of the algorithm is  0.9939024390243902
The Testing Accuracy of the algorithm is 0.9761904761904762
```

Hyperparameter Tuning-

```
a = rfc.feature_importances_
```

```
[68] col = X.columns
```

```
[69] feat_imp = {}
     for i, j in zip(a,col):
         feat_imp[j] = i
```

```
[70] feat_imp
```

```
'pain_behind_the_eyes': 0.01319824851771783,
'back_pain': 0.01689300502268126,
'constipation': 0.012293307899228953,
'abdominal_pain': 0.007626042803453611,
'diarrhoea': 0.009261739845267866,
'mild_fever': 0.01985800908553994,
'yellowing_of_eyes': 0.016565298512947028,
'swelled_lymph_nodes': 0.01657807305537706,
'malaise': 0.007187502891558174,
'blurred_and_distorted_vision': 0.008410478000376021,
'phlegm': 0.006391219937020358,
'congestion': 0.022700557966486682,
'chest_pain': 0.00952889415930193,
'weakness_in_limbs': 0.007195358230920688
```

```
[72] knn_results = []
```

```
[97] for main in [0.020,0.018,0.016,0.014,0.012,0.01,0.008]:
         to_drop = []
         for i,j in zip(feat_imp.keys(),feat_imp.values()):
             if j < main:
                 to_drop.append(i)

         X_new = X.drop(to_drop,axis = 1)
         y_new = y
         X1_train, X1_val, y1_train, y1_val = train_test_split(X_new, y_new, test_size=0.2)
         X1_test = X_test.drop(to_drop,axis = 1)
         y1_test = y_test
         knn_new = KNeighborsClassifier()
         knn_new.fit(X1_train, y1_train)
         temp1 = model_evaluation1(X1_train.shape[1],knn_new)
         knn_results.append(temp1)
```

Validation Method-

```
[53] X_train, X_val, y_train, y_val = train_test_split(X,y,test_size = 0.2)
```