Name : Niket Ralebhat

Section : Cse 2

Scholar Number : 211112268

In [ ]:
```
from sklearn import datasets

iris = datasets.load_iris()
x_train = iris.data
y_train = iris.target
feature_name = iris.feature_names
target_names = iris.target_names
```

In [ ]:
```
import numpy as np
from sklearn.metrics import confusion_matrix, precision_score, recall_score, roc_au

def performance_metrics(y_true, y_pred, y_prob=None):
    # 1. Confusion Matrix
    cm = confusion_matrix(y_true, y_pred)

    # 2. Precision
    precision = precision_score(y_true, y_pred , average="micro")

    # 3. Recall
    recall = recall_score(y_true, y_pred , average="micro")

    # 4. Gmean
    gmean = np.sqrt(recall * (1 - precision))

    # 5. True Positive Rate
    tpr = recall

    # 6. Area under Curve (ROC AUC Score)
    auc = roc_auc_score(y_true, y_prob) if y_prob is not None else None

    # 7. False Alarm Rate (Fallout)
    fpr = 1 - recall

    # 8. F1 Score (F-Measure)
    f1 = f1_score(y_true, y_pred , average="micro")

    # 9. Overall Accuracy
    accuracy = accuracy_score(y_true, y_pred)

    return {
        "Confusion Matrix": cm,
        "Precision": precision,
        "Recall": recall,
        "Gmean": gmean,
        "True Positive Rate": tpr,
        "Area under Curve": auc,
        "False Alarm Rate": fpr,
        "F1 Score": f1,
```

```
        "Overall Accuracy": accuracy
    }
```

Write a python program of naive bayes classifier for iris dataset classification

In [ ]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_sta
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
print("Number of mislabeled points out of a total %d points : %d"
      % (X_test.shape[0], (y_test != y_pred).sum()))

from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)

metrics = performance_metrics(y_test, y_pred)
for metric, value in metrics.items():
    print(f"{metric}: {value}")
```

```
Number of mislabeled points out of a total 75 points : 4
Confusion Matrix: [[21  0  0]
 [ 0 30  0]
 [ 0  4 20]]
Precision: 0.9466666666666667
Recall: 0.9466666666666667
Gmean: 0.2246973272847029
True Positive Rate: 0.9466666666666667
Area under Curve: None
False Alarm Rate: 0.053333333333333344
F1 Score: 0.9466666666666667
Overall Accuracy: 0.9466666666666667
```

Write a python program of naive bayes classifier for Play Tennis dataset classification

In [ ]:
```
import numpy as np
import pandas as pd
from sklearn import preprocessing
from sklearn import metrics
df=pd.read_csv("Play Tennis.csv")

le = preprocessing.LabelEncoder()
data_train_df = pd.DataFrame(df)
data_train_df_encoded = data_train_df.apply(le.fit_transform)
```

```
feature_cols = ['Outlook','Temprature','Humidity','Wind']
X = data_train_df_encoded[feature_cols ]
y = data_train_df_encoded.Play_Tennis
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
print("Number of mislabeled points out of a total %d points : %d"
      % (X_test.shape[0], (y_test != y_pred).sum()))

from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)

metrics = performance_metrics(y_test, y_pred)
for metric, value in metrics.items():
    print(f"{metric}: {value}")
```

```
Number of mislabeled points out of a total 5 points : 1
Confusion Matrix: [[2 0]
 [1 2]]
Precision: 0.8
Recall: 0.8
Gmean: 0.39999999999999997
True Positive Rate: 0.8
Area under Curve: None
False Alarm Rate: 0.19999999999999996
F1 Score: 0.8
Overall Accuracy: 0.8
```

Write a python program of naive bayes classifier for Large Movie Review Dataset dataset classification

In [ ]:

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics

df = pd.read_csv("IMDB Dataset.csv")

vectorizer = CountVectorizer(stop_words='english', max_features=5000)
X = vectorizer.fit_transform(df['review'].values.astype('U')).toarray()
y = df['sentiment'].map({'positive': 1, 'negative': 0})

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta

clf = MultinomialNB()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

```
metrics = performance_metrics(y_test, y_pred)
for metric, value in metrics.items():
    print(f"{metric}: {value}")
```

```
Confusion Matrix: [[6295 1116]
 [1218 6371]]
Precision: 0.8444
Recall: 0.8444
Gmean: 0.36247570953099734
True Positive Rate: 0.8444
Area under Curve: None
False Alarm Rate: 0.15559999999999996
F1 Score: 0.8444
Overall Accuracy: 0.8444
```