

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import svm, datasets
from sklearn.metrics import accuracy_score
```

```
In [2]: data=pd.read_csv("twitter_training.csv")
```

```
In [3]: data.head()
```

Out[3]:

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74681 entries, 0 to 74680
Data columns (total 4 columns):
#   Column                                Non-Null Count  Dtype
---  -
0    2401                                74681 non-null  int64
1    Borderlands                        74681 non-null  object
2    Positive                          74681 non-null  object
3    im getting on borderlands and i will murder you all , 73995 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.3+ MB
```

```
In [5]: data.describe()
```

Out[5]:

	2401
count	74681.000000
mean	6432.640149
std	3740.423819
min	1.000000
25%	3195.000000
50%	6422.000000
75%	9601.000000
max	13200.000000

```
In [6]: data.isnull().sum()
```

```
Out[6]: 2401                                0
Borderlands                             0
Positive                                0
im getting on borderlands and i will murder you all , 686
dtype: int64
```

In [7]: ▶ data.dropna()

Out[7]:

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...
...
74676	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74677	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74678	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74679	9200	Nvidia	Positive	Just realized between the windows partition of...
74680	9200	Nvidia	Positive	Just like the windows partition of my Mac is I...

73995 rows × 4 columns

```

In [8]: # Importing the necessary libraries
import numpy as np
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score

# Importing the dataset from the sklearn library into a local variable called dataset
dataset = load_digits()

# Splitting the data test into train 70% and test 30%.
# x_train, y_train are training data and Labels respectively
# x_test, y_test are testing data and Labels respectively
x_train, x_test, y_train, y_test = train_test_split(dataset.data, dataset.target, test_size=0.30, random_state=42)

# Making the SVM Classifier
Classifier = SVC(kernel="linear")

# Training the model on the training data and Labels
Classifier.fit(x_train, y_train)

# Using the model to predict the Labels of the test data
y_pred = Classifier.predict(x_test)

# Evaluating the accuracy of the model using the sklearn functions
accuracy = accuracy_score(y_test, y_pred)*100
confusion_mat = confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Printing the results
print(f"Accuracy for SVM is: {accuracy:.4f}")
print("Confusion Matrix")
print(confusion_mat)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)

```

Accuracy for SVM is: 97.9630

Confusion Matrix

```

[[53  0  0  0  0  0  0  0  0  0]
 [ 0 50  0  0  0  0  0  0  1  0]
 [ 0  0 54  0  0  0  0  0  0  0]
 [ 0  0  0 54  0  1  0  0  0  0]
 [ 0  1  0  0 53  0  0  0  0  0]
 [ 0  0  0  0  0 57  0  0  0  1]
 [ 0  0  0  0  0  0 51  0  1  0]
 [ 0  0  0  0  0  0  0 54  1  0]
 [ 0  4  0  0  0  0  0  0 51  0]
 [ 0  0  0  0  0  0  0  0  1 52]]

```

Precision: 0.9803030303030303

Recall: 0.9796296296296296

F1 Score: 0.979804640683546

```

In [9]: ► from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score

# Load data
data = pd.read_csv("twitter_training.csv")

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(dataset.data, dataset.target, test_size=0.2)

# Create a Gaussian Naive Bayes classifier
clf = GaussianNB()

# Train the classifier
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Print the accuracy
print(f"Accuracy of the Naive Bayes classifier: {accuracy:.4f}")
print("Confusion Matrix")
print(confusion_mat)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)

```

Accuracy of the Naive Bayes classifier: 0.8361

Confusion Matrix

```

[[40  0  0  0  0  0  0  0  0  0]
 [ 0 23  1  0  0  0  0  0  5  4]
 [ 0  1 22  0  0  0  0  0  9  0]
 [ 0  0  0 20  0  1  0  2  4  3]
 [ 0  0  0  0 30  1  1  4  0  0]
 [ 0  0  0  1  0 34  0  2  0  0]
 [ 0  0  0  0  0  1 33  0  0  0]
 [ 0  0  0  0  0  1  0 45  0  0]
 [ 0  0  1  0  0  1  0  5 25  0]
 [ 0  0  0  3  0  0  1  3  4 29]]

```

Precision: 0.8595828097065599

Recall: 0.8361111111111111

F1 Score: 0.8383944512764792

```

In [10]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score
from sklearn.feature_extraction.text import CountVectorizer

# Load the dataset
data = pd.read_csv('twitter_training.csv')

# Split the data into features and target
X = data.iloc[:, 2] # Assuming the text data is in the 4th column (index 3)
y = data.iloc[:, 1] # Assuming the sentiment is in the 3rd column (index 2)

# Convert X to strings
X = X.astype(str)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train the Random Forest Classifier
rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train_vec, y_train)

# Predict the sentiment on the test set
y_pred = rf_classifier.predict(X_test_vec)

# Calculate the accuracy
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print("Accuracy:", accuracy)
print("Confusion Matrix")
print(confusion_mat)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)

```

C:\Users\KIIT\anaconda3\Lib\site-packages\sklearn\metrics_classification.py:1469: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))

Accuracy: 0.06955881368414006
 Confusion Matrix
 [[242 0 67 ... 0 0 0]
 [204 0 138 ... 0 0 0]
 [22 0 270 ... 0 0 0]
 ...
 [192 0 156 ... 0 0 0]
 [75 0 150 ... 0 0 0]
 [207 0 48 ... 0 0 0]]
 Precision: 0.008501101288488954
 Recall: 0.06955881368414006
 F1 Score: 0.015080324491006613

```

In [11]: ► import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score
from sklearn.feature_extraction.text import CountVectorizer

# Load the dataset
data = pd.read_csv('twitter_training.csv')

# Split the data into features and target
X = data.iloc[:, 2] # Assuming the text data is in the 4th column (index 3)
y = data.iloc[:, 1] # Assuming the sentiment is in the 3rd column (index 2)

# Convert X to strings
X = X.astype(str)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Train the Neural Network Classifier
nn_classifier = MLPClassifier()
nn_classifier.fit(X_train_vec, y_train)

# Predict the sentiment on the test set
y_pred = nn_classifier.predict(X_test_vec)

# Calculate the accuracy
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print("Accuracy:", accuracy)
print("Confusion Matrix")
print(confusion_mat)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)

```

C:\Users\KIIT\anaconda3\Lib\site-packages\sklearn\metrics_classification.py:1469: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))

Accuracy: 0.06971658112028566
 Confusion Matrix
 [[354 0 99 ... 0 0 0]
 [311 0 206 ... 0 0 0]
 [41 0 424 ... 0 0 0]
 ...
 [316 0 214 ... 0 0 0]
 [109 0 222 ... 0 0 0]
 [311 0 73 ... 0 0 0]]
 Precision: 0.008567789932028138
 Recall: 0.06971658112028566
 F1 Score: 0.015196180467998007

In []: ►