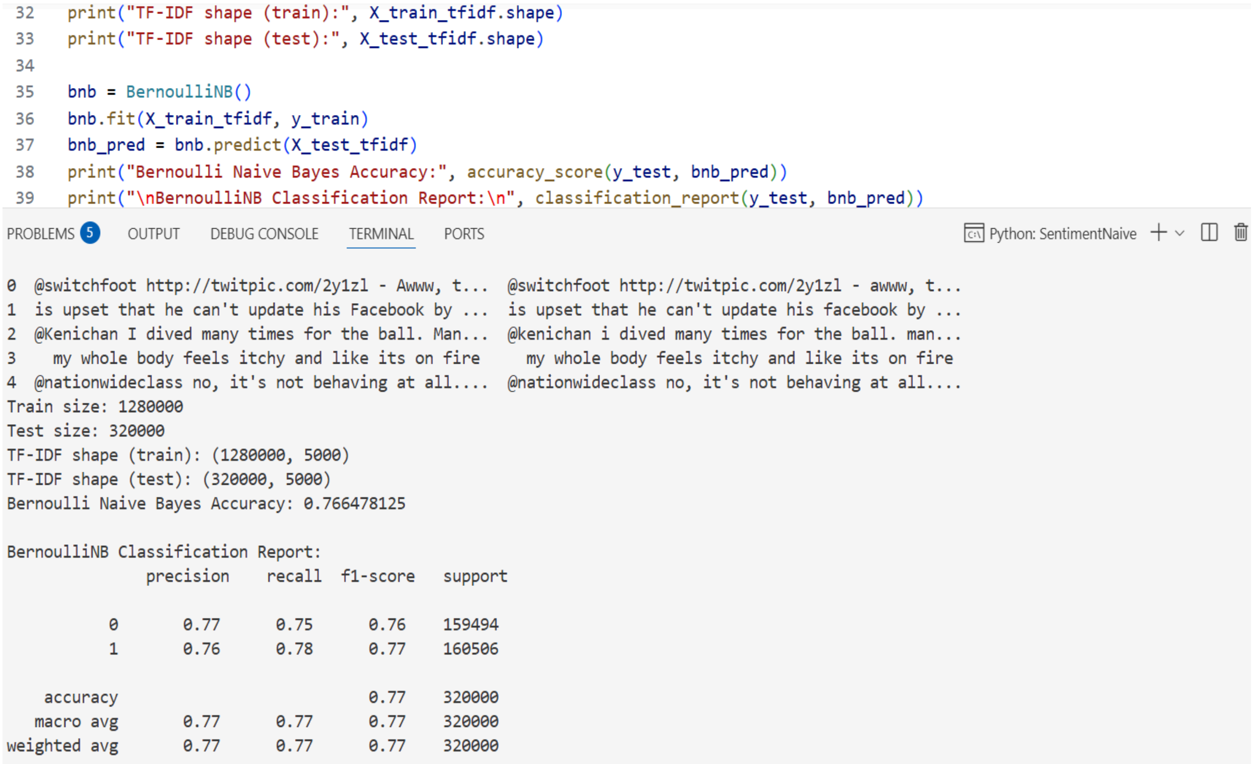
Earlier we tried to understand how to use Python for analysis for the dataset used in our code which is important to study the co-relation between the various values (pollutants & the calculated AQI) and also to understand the accuracy (MSE : mean squared error) of the technique / model being used in an “**AirPollutionDataset**”. Then, we saw how to perform **Web Scraping** which is a real-time application & is used by businesses for market research and in product price comparisons. Let us try to understand more about **Machine Learning using Python** and explore further with “**Sentiment Analysis**” dataset & going further understand about forecasting such as Earthquake Prediction using Machine Learning.

Python Intermediate (Applications)

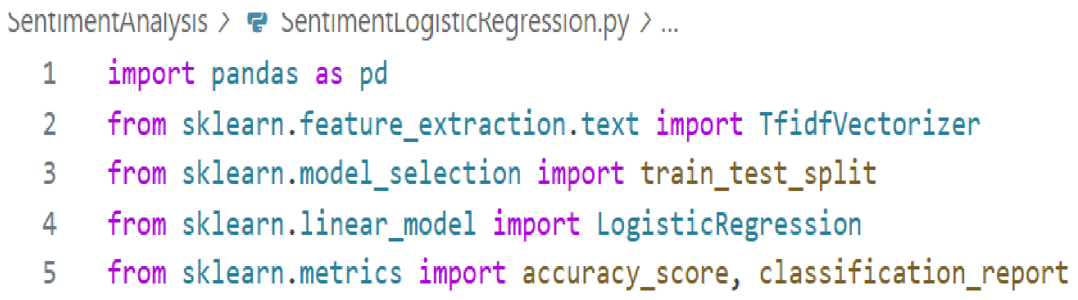
Let us try to understand how to use Python for “Sentiment Analysis” by exploring the Naive Bayes, Support Vector Classifier & Logistic Regression **machine learning techniques.**



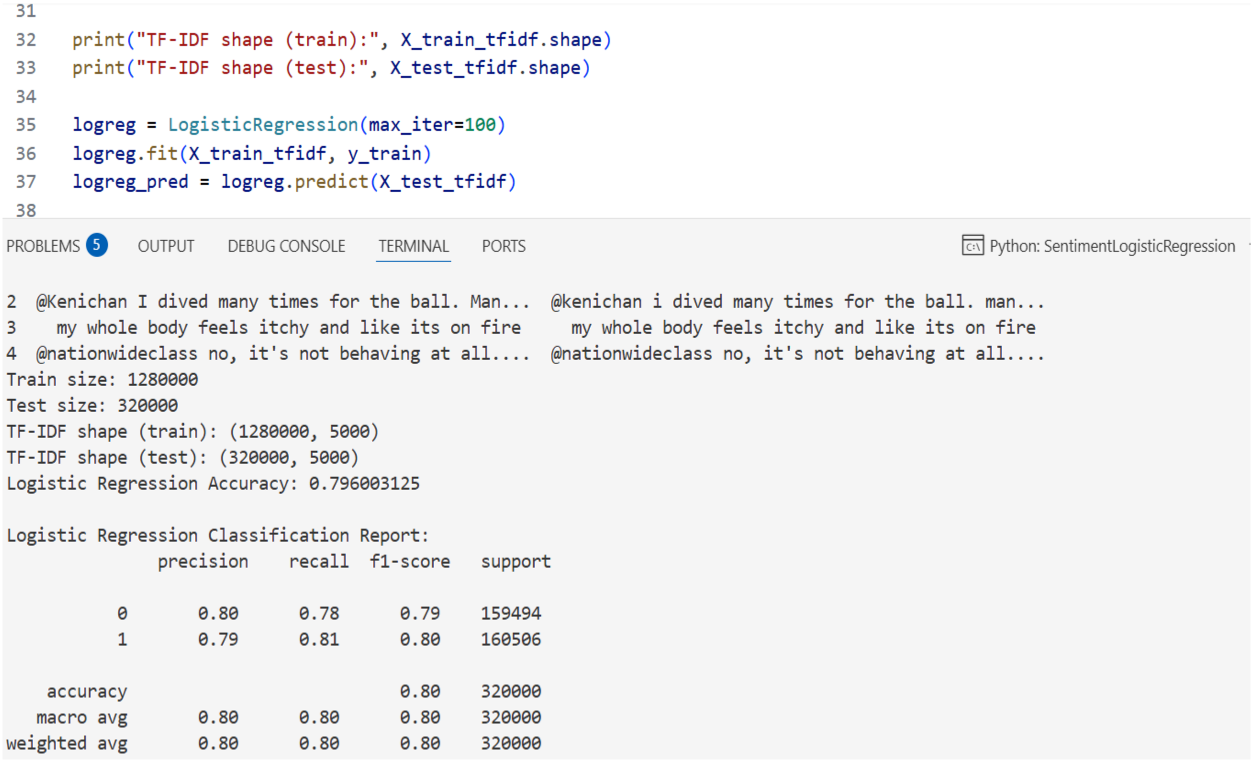
```
print("TF-IDF shape (train):", X_train_tfidf.shape)
print("TF-IDF shape (test):", X_test_tfidf.shape)

bnb = BernoulliNB()
bnb.fit(X_train_tfidf, y_train)
bnb_pred = bnb.predict(X_test_tfidf)
print("Bernoulli Naive Bayes Accuracy:", accuracy_score(y_test, bnb_pred))
print("\nBernoulliNB Classification Report:\n", classification_report(y_test, bnb_pred))
```

```
0  @switchfoot http://twitpic.com/2y1zl - Awww, t...  @switchfoot http://twitpic.com/2y1zl - awww, t...
1  is upset that he can't update his Facebook by ...  is upset that he can't update his facebook by ...
2  @Kenichan I dived many times for the ball. Man...  @kenichan i dived many times for the ball. man...
3    my whole body feels itchy and like its on fire     my whole body feels itchy and like its on fire
4  @nationwideclass no, it's not behaving at all....  @nationwideclass no, it's not behaving at all....
Train size: 1280000
Test size: 320000
TF-IDF shape (train): (1280000, 5000)
TF-IDF shape (test): (320000, 5000)
Bernoulli Naive Bayes Accuracy: 0.766478125

BernoulliNB Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.75      0.76    159494
           1       0.76      0.78      0.77    160506

    accuracy                           0.77    320000
   macro avg       0.77      0.77      0.77    320000
weighted avg       0.77      0.77      0.77    320000
```

Accuracy	**Precision**	**Recall**	**f1-score**	**support**
Accuracy is the proportion of all classifications that were correct, whether positive or negative	Precision is the proportion of all the model's positive classifications that are actually positive	The true positive rate (TPR), or the proportion of all actual positives that were classified correctly as positives, is also known as recall.	A single metric that provides a balance between precision and recall.	component for calculating the F1 score, especially in multiclass scenarios.



```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

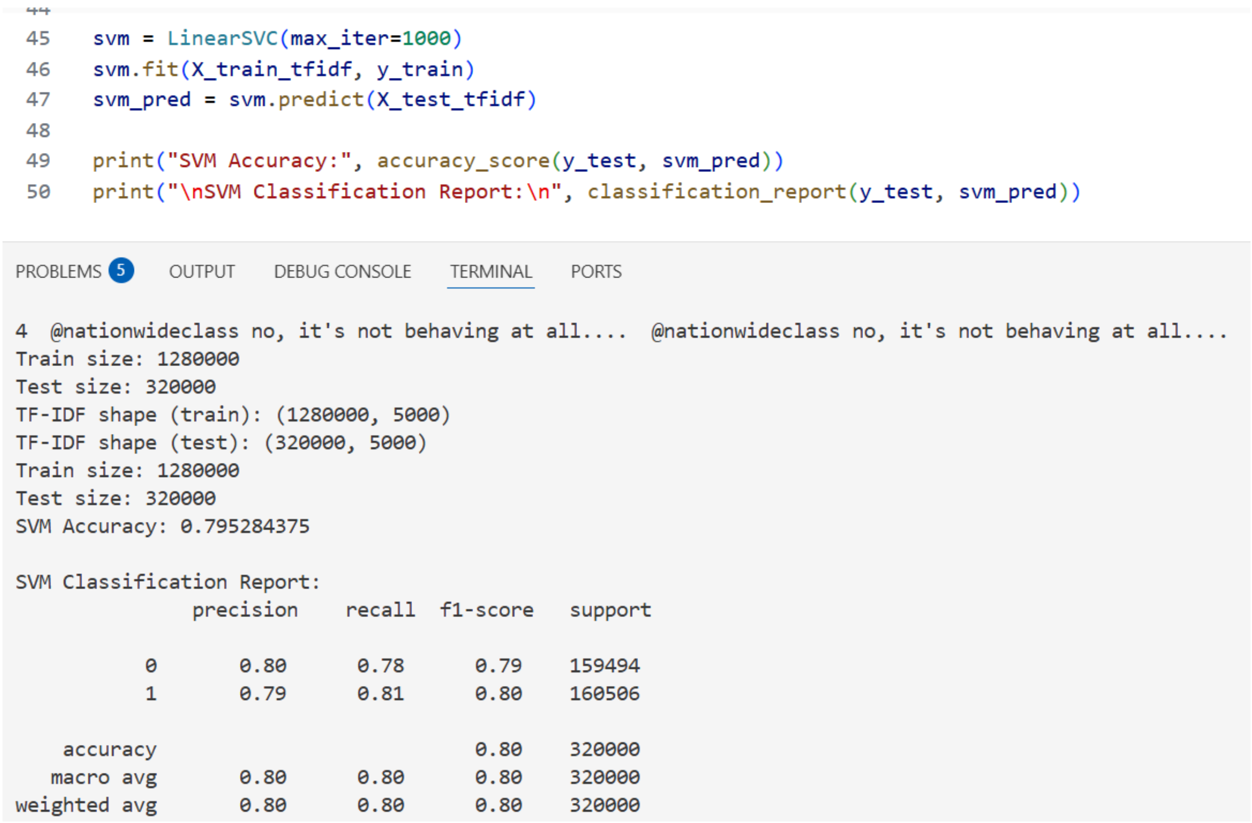


```python

print("TF-IDF shape (train):", X_train_tfidf.shape)
print("TF-IDF shape (test):", X_test_tfidf.shape)

logreg = LogisticRegression(max_iter=100)
logreg.fit(X_train_tfidf, y_train)
logreg_pred = logreg.predict(X_test_tfidf)

```

```
2  @Kenichan I dived many times for the ball. Man...  @kenichan i dived many times for the ball. man...
3    my whole body feels itchy and like its on fire     my whole body feels itchy and like its on fire
4  @nationwideclass no, it's not behaving at all....  @nationwideclass no, it's not behaving at all....
Train size: 1280000
Test size: 320000
TF-IDF shape (train): (1280000, 5000)
TF-IDF shape (test): (320000, 5000)
Logistic Regression Accuracy: 0.796003125

Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.78      0.79    159494
           1       0.79      0.81      0.80    160506

    accuracy                           0.80    320000
   macro avg       0.80      0.80      0.80    320000
weighted avg       0.80      0.80      0.80    320000
```

Accuracy	**Precision**	**Recall**	**f1-score**	**support**
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$2\times\frac{(Precision \times Recall)}{(Precision+Recall)}$	The number of true instances for each label in the dataset.

TP/N : True positives/negatives
FP : False positives are actual negatives that were misclassified
FN : False Negatives (False negatives are actual positives that were misclassified as negatives)



```python
svm = LinearSVC(max_iter=1000)
svm.fit(X_train_tfidf, y_train)
svm_pred = svm.predict(X_test_tfidf)

print("SVM Accuracy:", accuracy_score(y_test, svm_pred))
print("\nSVM Classification Report:\n", classification_report(y_test, svm_pred))
```

```
4  @nationwideclass no, it's not behaving at all....  @nationwideclass no, it's not behaving at all....
Train size: 1280000
Test size: 320000
TF-IDF shape (train): (1280000, 5000)
TF-IDF shape (test): (320000, 5000)
Train size: 1280000
Test size: 320000
SVM Accuracy: 0.795284375

SVM Classification Report:
               precision    recall  f1-score   support

           0       0.80      0.78      0.79    159494
           1       0.79      0.81      0.80    160506

    accuracy                           0.80    320000
   macro avg       0.80      0.80      0.80    320000
weighted avg       0.80      0.80      0.80    320000
```

A brief overview of some of the algorithms in machine learning

Algorithms /Models	**Naive Bayes**	**Logistic Regression**	**SVM**
Model works by,	A generative, probabilistic model to calculate the probability of a class given its features, assuming the features are conditionally independent of each other.	A generative, probabilistic model to calculate the probability of a class given its features, assuming the features are conditionally independent of each other.	Finds the optimal separating hyperplane with the largest margin between classes, making it robust to outliers & good for high-dimensional data.
Observations (Accuracy of algo/model)	~77%	~80%	~80%

AIM of sentiment analysis is to classify the input tuples into two values (0 // no sentiment detected or 4 // sentiment detected). We use the dataset to train the classifier & check the accuracy of the models on the dataset.