

Importing the dataset

```
In [1]: import pandas as pd
data = pd.read_csv('updated_pollution_dataset.csv')
print(data.head())
```

	Temperature	Humidity	PM2.5	PM10	NO2	SO2	CO	\
0	29.8	59.1	5.2	17.9	18.9	9.2	1.72	
1	28.3	75.6	2.3	12.2	30.8	9.7	1.64	
2	23.1	74.7	26.7	33.8	24.4	12.6	1.63	
3	27.1	39.1	6.1	6.3	13.5	5.3	1.15	
4	26.5	70.7	6.9	16.0	21.9	5.6	1.01	

	Proximity_to_Industrial_Areas	Population_Density	Air Quality
0	6.3	319	Moderate
1	6.0	611	Moderate
2	5.2	619	Moderate
3	11.1	551	Good
4	12.7	303	Good

Determining Key Features

```
In [2]: key = ['Temperature', 'Humidity', 'Population_Density', 'Air Quality']
df = pd.DataFrame(data, columns=key)
df.head()
```

```
Out[2]:
```

	Temperature	Humidity	Population_Density	Air Quality
0	29.8	59.1	319	Moderate
1	28.3	75.6	611	Moderate
2	23.1	74.7	619	Moderate
3	27.1	39.1	551	Good
4	26.5	70.7	303	Good

```
In [3]: y = df['Air Quality']
X = df.drop(columns=['Air Quality'])
```

Converting labels, splitting and training the dataset

```
In [4]: from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split
        from sklearn.svm import SVC
```

```
In [5]: label_encoder = LabelEncoder()
        y_encoded = label_encoder.fit_transform(y)
```

```
In [6]: X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
        model = SVC(kernel='linear')
        model.fit(X_train, y_train)
```

```
Out[6]: SVC
        SVC(kernel='linear')
```

```
In [7]: y_pred = model.predict(X_test)
```

Determining the accuracy, precision, recall and f1-score

```
In [8]: from sklearn.metrics import accuracy_score
        from sklearn.metrics import precision_score, recall_score, f1_score
```

```
In [9]: accuracy=accuracy_score(y_test,y_pred)
        print("Accuracy:", accuracy)
```

Accuracy: 0.692

```
In [10]: precision=precision_score(y_test,y_pred, average="weighted")
         recall=recall_score(y_test,y_pred, average="weighted")
         f1score=f1_score(y_test,y_pred, average="weighted")
```

```
In [11]: print("Precision:", precision)
         print("Recall:", recall)
         print("F1-score:", f1score)
```

Precision: 0.6957033810627302

Recall: 0.692

F1-score: 0.689657879069599

Plotting the confusion matrix

```
In [12]: from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [13]: cm = confusion_matrix(y_test,y_pred)
cm
```

```
Out[13]: array([[355,  0, 54,  0],
               [ 0, 54,  4, 53],
               [ 69,  0, 182, 43],
               [ 7, 15, 63, 101]])
```

```
In [14]: labels = label_encoder.classes_

sns.heatmap(cm, annot=True, fmt="d", cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

