**AIR QUALITY ANALYSIS AND**

**PREDICTION IN MACHINE LEARNING**

**Phase-4 submission document**

**Project Title**: Air Quality Analysis and Prediction in Tamil Nadu

**Phase 4:** Development Part 2

**Topic**: Start building the: Air Quality Analysis and prediction in Tamil Nadu model by by feature engineering, model training, and evaluation

A factory with smoke coming out of it

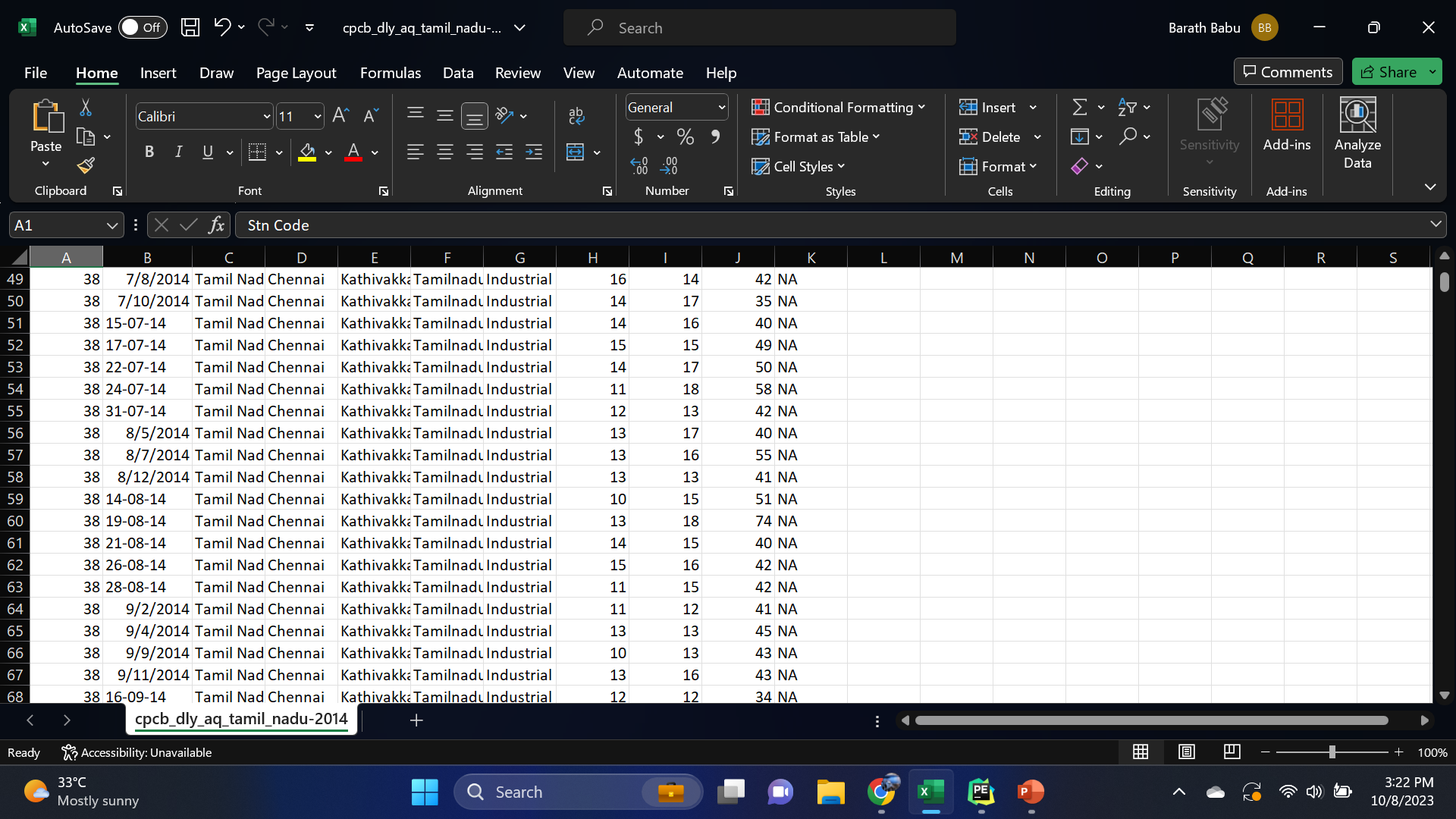
Description automatically generated

**Air Quality Analysis and Prediction in machine learning**

**Introduction:**

* Air quality is a critical environmental factor that directly affects the health and well-being of individuals and communities. Poor air quality can lead to a range of health problems, including respiratory diseases, cardiovascular issues, and even premature death.
* In this project, we aim to conduct a comprehensive analysis of air quality data in the state of Tamil Nadu, India. Our goal is not only to understand historical air quality patterns but also to build a predictive model that can help forecast air quality in the future.
* The importance of this project lies in its potential to improve air quality monitoring and prediction in Tamil Nadu. By providing accurate and timely air quality forecasts, we can empower residents to make informed decisions about outdoor activities and health precautions. Additionally, government agencies and environmental organizations can use this information to develop and implement targeted interventions to reduce air pollution and its associated health risks.
* Through this project, we aim to contribute to a healthier and more sustainable environment for the people of Tamil Nadu, and by extension, inspire similar initiatives in other regions facing air quality challenges.

**Given data set:**



2880 Rows x 11 Columns

**Overview of the process:**

The following is an overview of the process of Air Quality Analysis and prediction in Tamil Nadu model by feature selection, model training, and

evaluation:

1. **Prepare the data:** This includes cleaning the data, removing

outliers, and handling missing values.

2. **Perform feature selection:** This can be done using a variety of

methods, such as correlation analysis, information gain, and recursive feature elimination.

3. **Train the model:** There are many different machine learning learning algorithms that can be used for Air Quality Analysis prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.

4. **Evaluate the model:** This can be done by calculating the mean

squared error (MSE) or the root mean squared error (RMSE) of the

model's predictions on the held-out test set.

5. **Deploy the model:** Once the model has been evaluated and found

to be performing well, it can be deployed to production so that it can be used to predict the Air Quality Analysis and prediction in Tamil Nadu.

**PROCEDURE:**

**Feature selection:**

1. **Identify the target variable.** This is the variable that you want to

predict, such as Air Quality Analysis.

2. **Explore the data.** This will help you to understand the

relationships between the different features and the target variable. You can use data visualization and correlation analysis to identify features that are highly correlated with the target variable.

3. **Remove redundant features.** If two features are highly correlated with each other, then you can remove one of the features, as they are likely to contain redundant information.

4. **Remove irrelevant features.** If a feature is not correlated with the target variable, then you can remove it, as it is unlikely to be useful for prediction

**Model training:**

1. **Choose a machine learning algorithm.** There are a number of

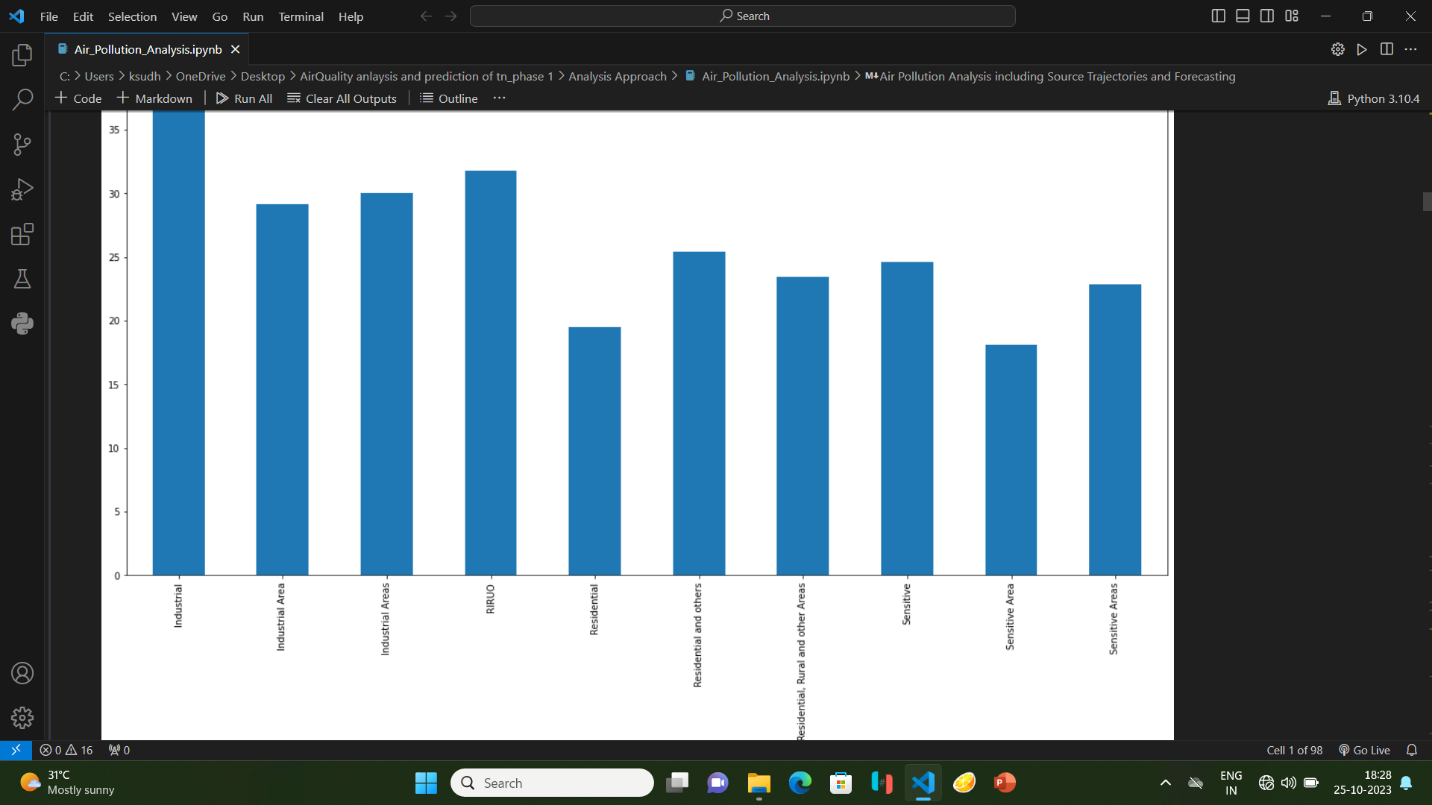
different machine learning algorithms that can be used for Air Quality Analysis prediction, such as NO2, SO2, RSPM and PM10, across station ,cities, areas are Covered.

**NO2:**

**station;**

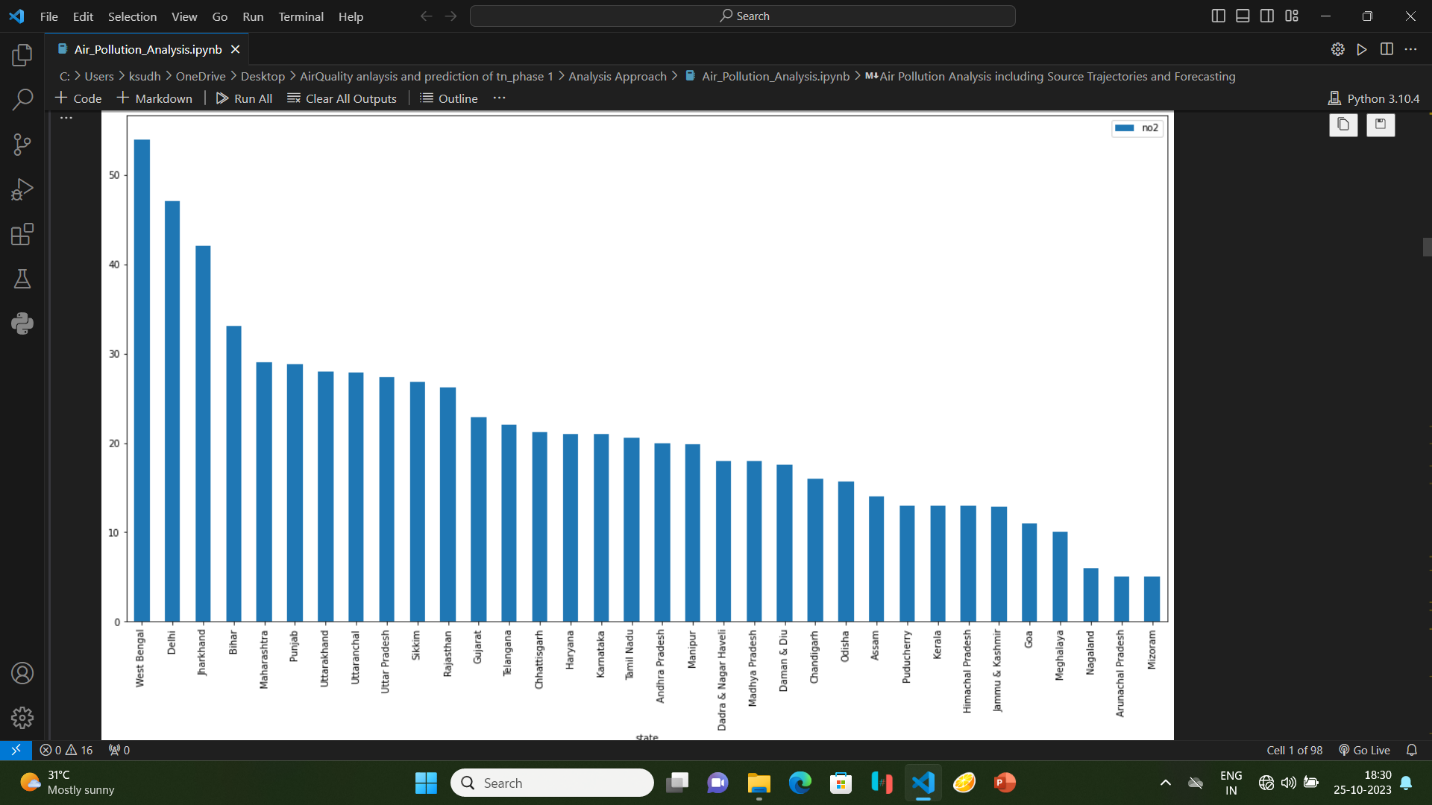
grp = df.groupby(["type"]).mean()["no2"].to\_frame()

grp.plot.bar(figsize = (20,10))



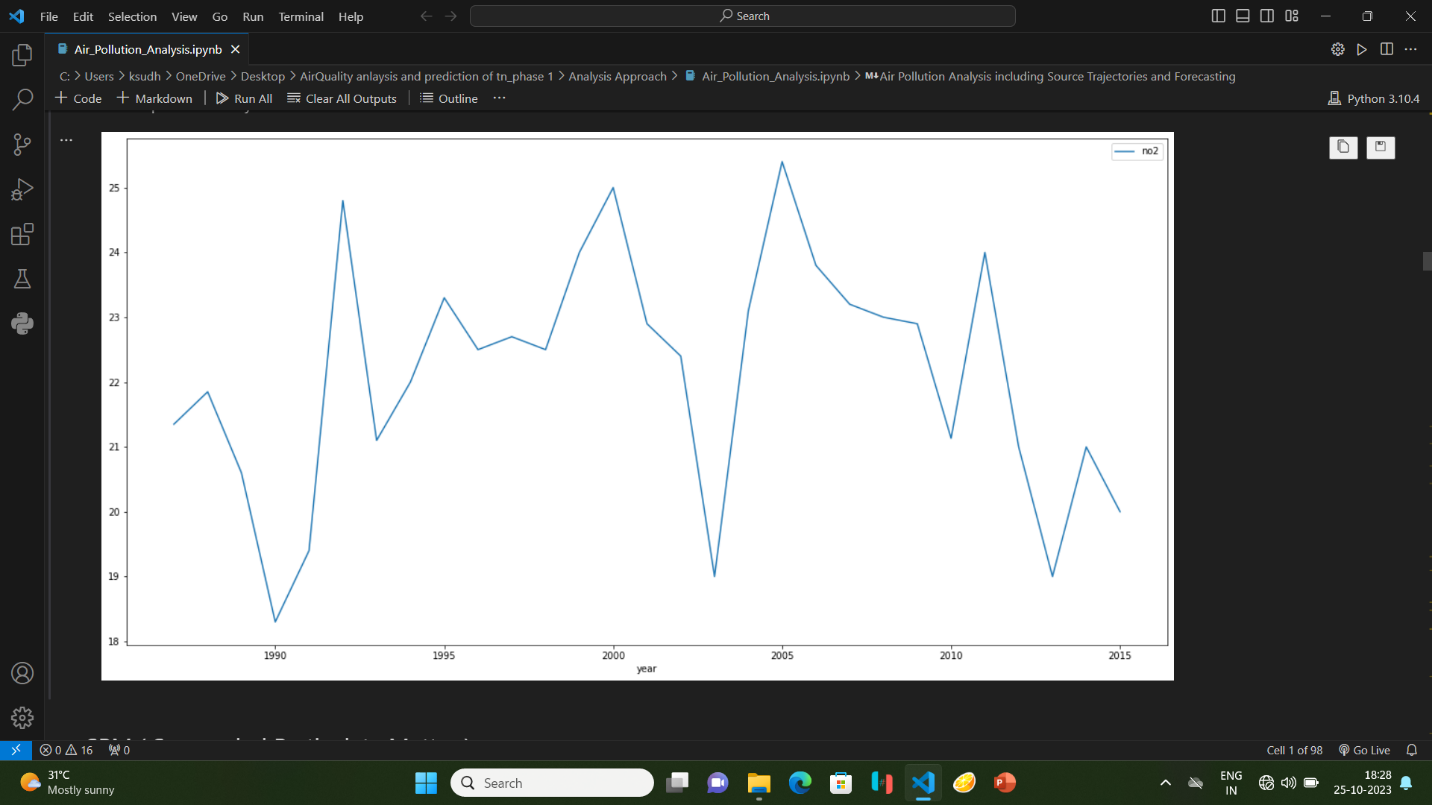
**cities;**

df[['no2', 'state']].groupby(['state']).median().sort\_values("no2", ascending = False).plot.bar(figsize=(20,10))

****

**areas;**

df[['no2','year','state']].groupby(["year"]).median().sort\_values(by='year',ascending=False).plot(figsize=(20,10))

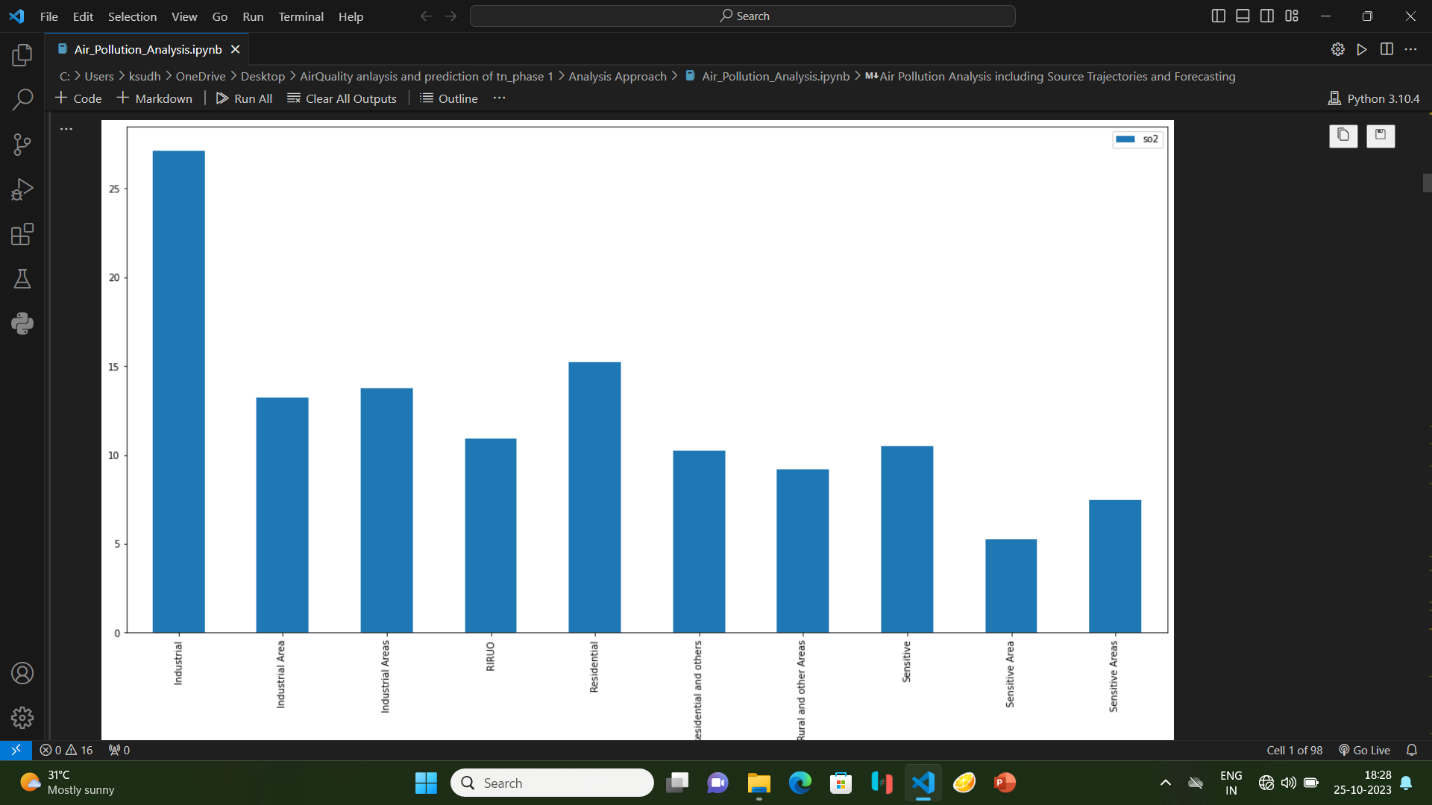


**SO2:**

**station;**

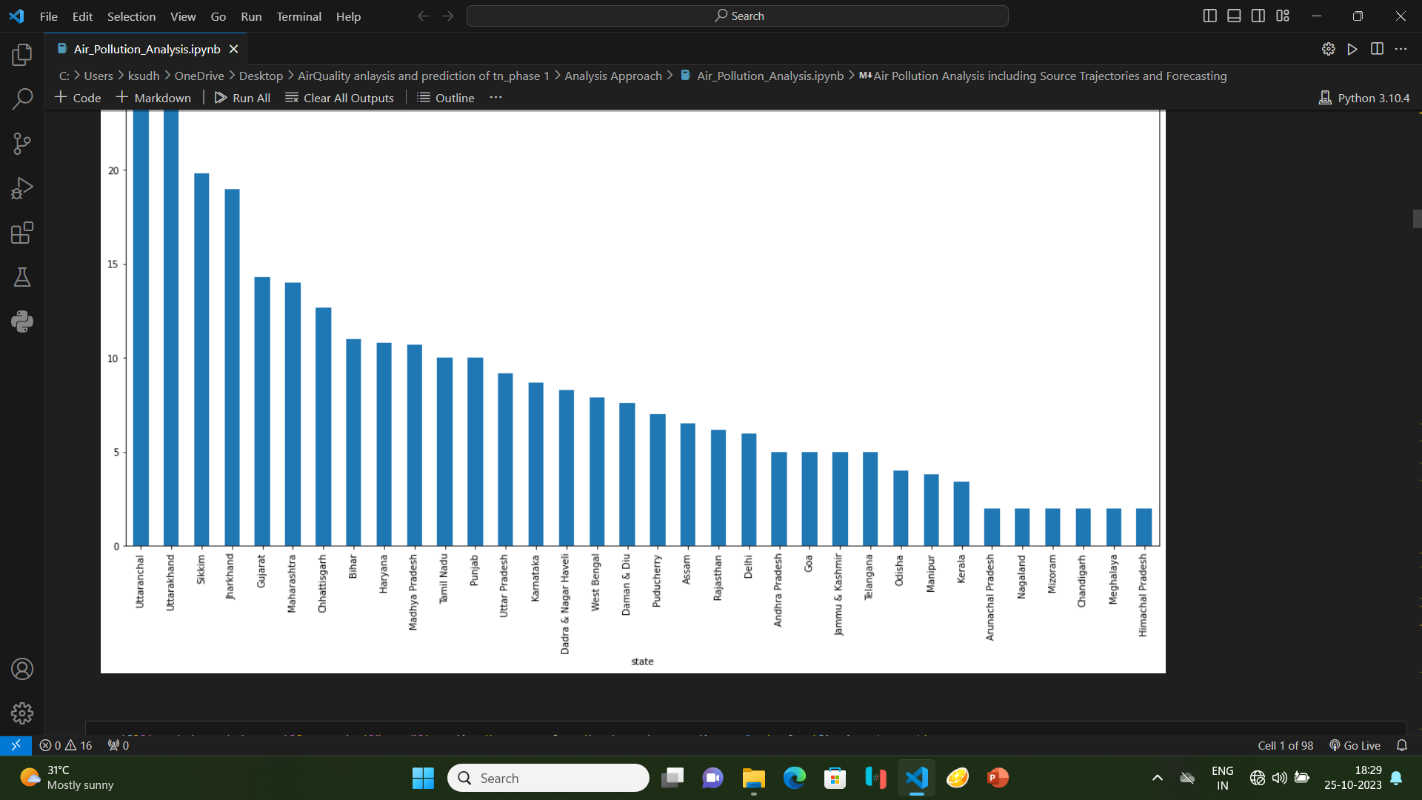
grp = df.groupby(["type"]).mean()["so2"].to\_frame()

grp.plot.bar(figsize = (20,10))

****

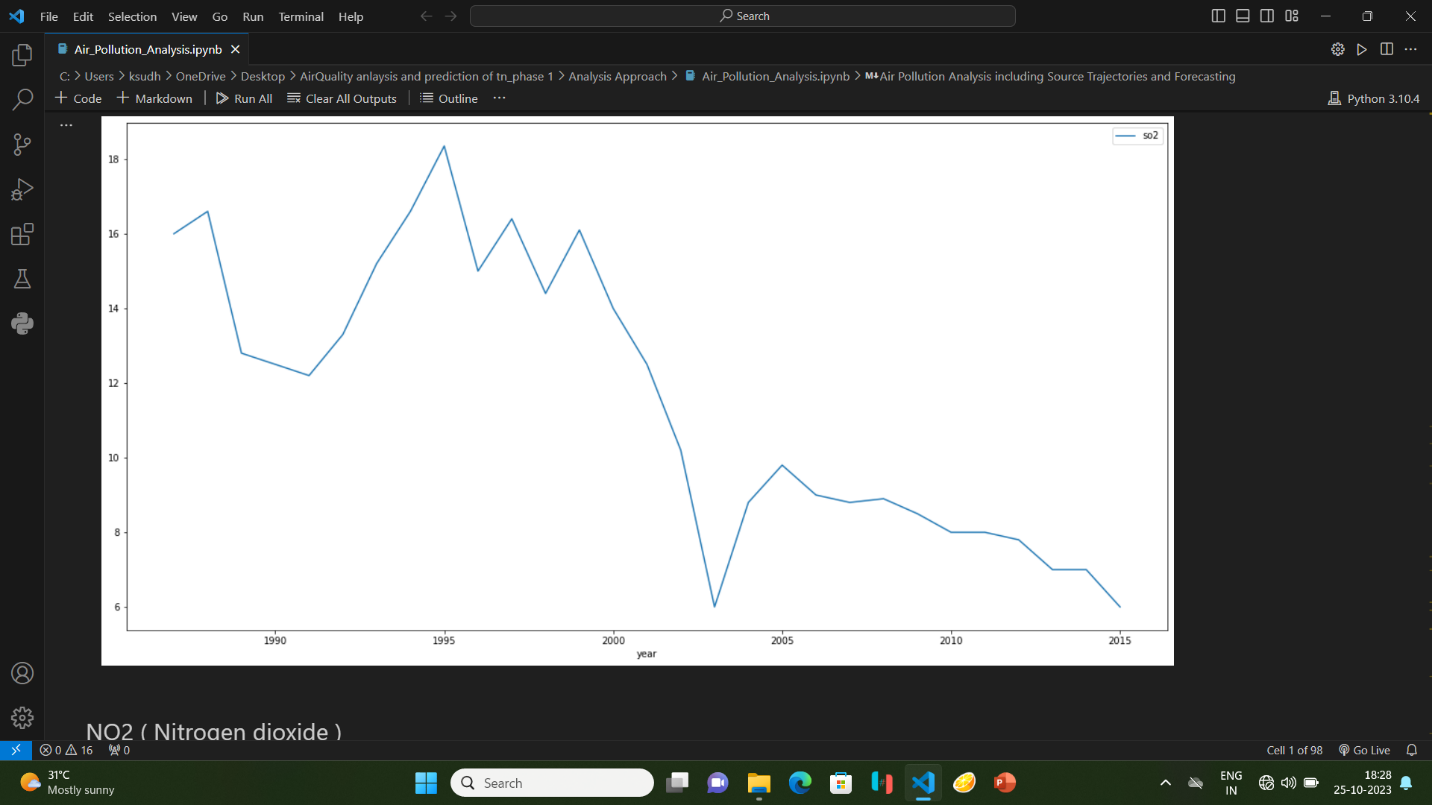
**cities;**

df[['so2', 'state']].groupby(['state']).median().sort\_values("so2", ascending = False).plot.bar(figsize=(20,10))



**areas;**

df[['so2','year','state']].groupby(["year"]).median().sort\_values(by='year',ascending=False).plot(figsize=(20,10))



**RSPM and PM10:**

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read\_csv('air.csv')

data['date'] = pd.to\_datetime(data['date'])

data = data.sort\_values('date')

plt.figure(figsize=(10, 6))

plt.plot(data['date'], data['rspm'], label='RSPM/PM10 Trend', marker='o', linestyle='-')

plt.title('RSPM/PM10 Trend Over Time')

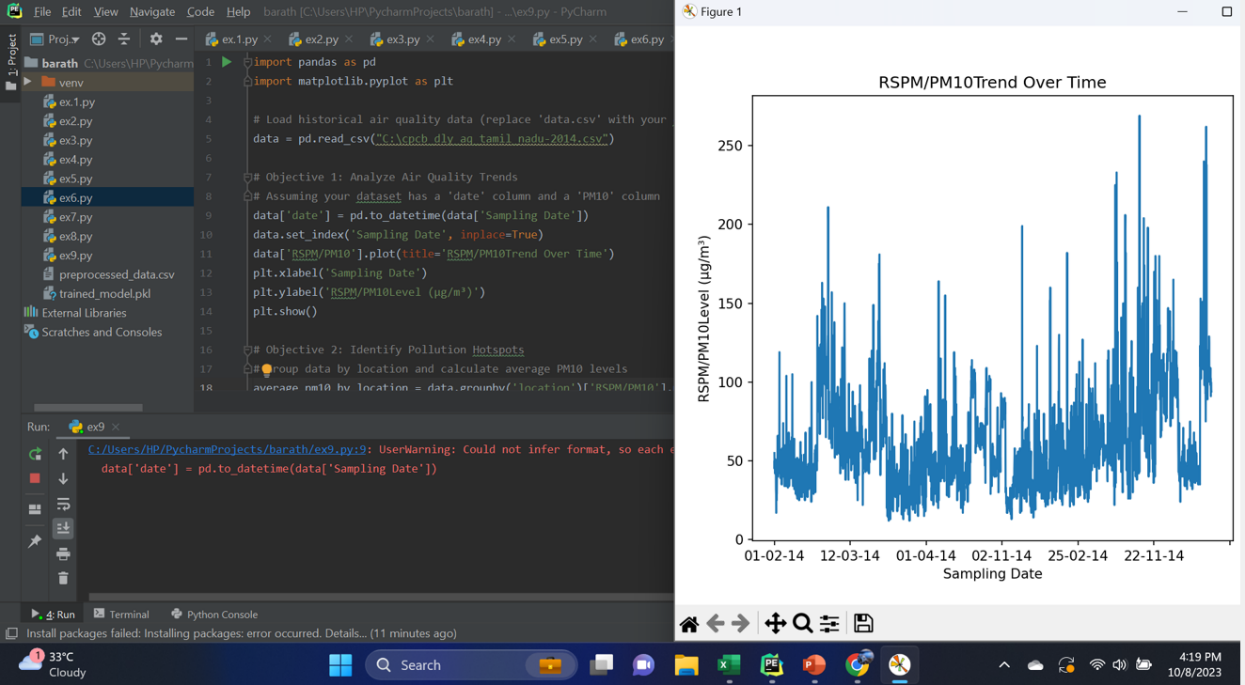
plt.xlabel('Date')

plt.ylabel('RSPM/PM10')

plt.legend()

plt.grid(True)

plt.show()

****

**Actual vs predicted RSPM and PM10;**

import pandas as pd

import matplotlib.pyplot as plt

data = pd.DataFrame({

'actual': [24, 32, 20, 28, 30, 35, 40, 42],

'predicted': [22, 30, 25, 26, 28, 34, 38, 40]

})

plt.figure(figsize=(8, 6))

plt.scatter(data['actual'], data['predicted'], color='b', label='Actual vs Predicted RSPM/PM10 level')

plt.plot(data['actual'], data['actual'], color='r', linestyle='--', label='Ideal')

plt.title('Actual vs Predicted RSPM/PM10 Level')

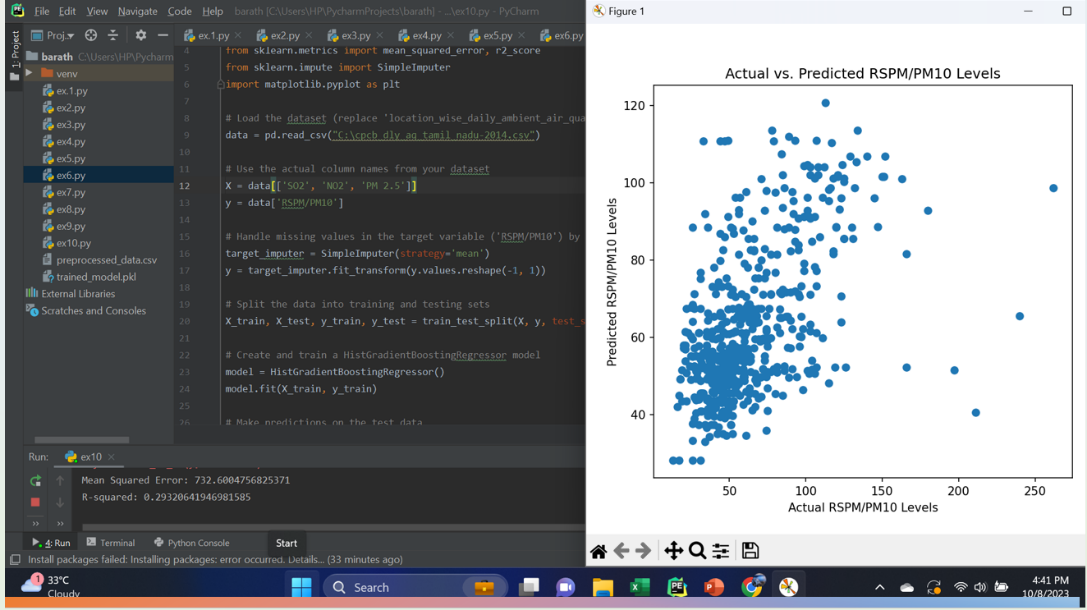
plt.xlabel('Actual RSPM/PM10 Level')

plt.ylabel('Predicted RSPM/PM10 Level')

plt.legend()

plt.grid(True)

plt.show()

****

**Model training:**

¬ Model training is the process of teaching a machine learning model to predict Air Quality Analysis. It involves feeding the model historical data on Air Quality Analysis and features, such as square footage, number of bedrooms, and location. The model then learns the relationships between these features and Air Q.

¬ Once the model is trained, it can be used to predict Air Quality Analysis for new data. For example, you could use the model to predict Air Quality Analysis that you are interested in buying.

1. **Prepare the data.** This involves cleaning the data, removing any

errors or inconsistencies, and transforming the data into a format that is compatible with the machine learning algorithm that you will be using.

2. **Split the data into training and test sets.** The training set will be

used to train the model, and the test set will be used to evaluate

the performance of the model on unseen data.

3. **Choose a machine learning algorithm.** There are a number of

different machine learning algorithms that can be used for Air Quality Analysis prediction, such as NO2, SO2, RSPM and PM10, across station ,cities, areas.

4. **Tune the hyperparameters of the algorithm.** The

hyperparameters of a machine learning algorithm are parameters that control the learning process. It is important to tune the hyperparameters of the algorithm to optimize its performance.

5. **Train the model on the training set.** This involves feeding the

training data to the model and allowing it to learn the relationships

between the features and Air Quality Analysis.

6. **Evaluate the model on the test set.** This involves feeding the test data to the model and measuring how well it predicts the Air Quality Analysis. If the model performs well on the test set, then you can be confident that it will generalize well to new data.

**Dividing Dataset in to features and target variable:**

In [12]:

X = dataset[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of

Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']]

Y = dataset['Price']

2. **Split the data into training and test sets.**

The training set will be used to train the model, and the test set will be used to evaluate the performance of the model.

In [13]:

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_st

ate=101)

In [14]:

Y\_train.head()

Out[14]:

3413 1.305210e+06

1610 1.400961e+06

3459 1.048640e+06

4293 1.231157e+06

1039 1.391233e+06

Name: air analysis, dtype: float64

In [15]:

Y\_train.shape

Out[15]:

(4000,)

In [16]:

Y\_test.head()

Out[16]:

1718 1.251689e+06

2511 8.730483e+05

345 1.696978e+06

2521 1.063964e+06

54 9.487883e+05

Name: air analysis, dtype: float64

In [17]:

Y\_test.shape

Out[17]: (1000)

3. **Train the model on the training set.** This involves feeding the

training data to the model and allowing it to learn the relationships

between the features and the target variable.

4. **Evaluate the model on the test set.** This involves feeding the test data to the model and measuring how well it predicts the target variable.

**Model evaluation:**

1. **Calculate the evaluation metrics.** There are a number of different evaluation metrics that can be used to assess the performance of a machine learning model, such as ***R-squared, mean squared error (MSE), and root mean squared error (RMSE).***

2. **Interpret the evaluation metrics.** The evaluation metrics will

give you an idea of how well the model is performing on unseen data. If the model is performing well, then you can be confident that it will generalize well to new data. However, if the model is performing poorly, then you may need to try a different model or retune the hyperparameters of the current model.

**Model evaluation:**

Model evaluation is the process of assessing the performance of a

machine learning model on unseen data. This is important to ensure

that the model will generalize well to new data.

There are a number of different metrics that can be used to evaluate

the performance of a Air Quality Analysis prediction model. Some of the most common metrics include:

• **Mean squared error (MSE):** This metric measures the average

squared difference between the predicted and actual Air Quality Analysis.

• **Root mean squared error (RMSE):** This metric is the square root

of the MSE.

• **Mean absolute error (MAE):** This metric measures the average

absolute difference between the predicted and actual Air Quality Analysis.

• **R-squared:** This metric measures how well the model explains the

variation in the actual Air Quality Analysis.In addition to these metrics, it is also important to consider the

following factors when evaluating a Air Quality Analysis prediction model:

• **Bias:** Bias is the tendency of a model to consistently over- or

underestimate Air Quality Analysis.

• **Variance:** Variance is the measure of how much the predictions of

a model vary around the true Air Quality Analysis

• **Interpretability:** Interpretability is the ability to understand how

the model makes its predictions. This is important for Air Quality Analysis prediction models, as it allows users to understand the factors that influence the predicted Air Quality Analysis.

**Various feature to perform model training:**



• **Use a variety of feature engineering techniques.**

Feature engineering is the process of transforming raw data into

features that are more informative and predictive for machine learning models. By using a variety of feature engineering techniques, you can create a set of features that will help your model to predict Air Quality analysis accurately more.

• **Use cross-validation.**

Cross-validation is a technique for evaluating the performance of a

machine learning model on unseen data. It is important to use across validation to evaluate the performance of your model during the training process. This will help you to avoid overfitting and to ensure that your model will generalize well to new data.

• **Use ensemble methods.**

Ensemble methods are machine learning methods that combine the

predictions of multiple models to produce a more accurate prediction. Ensemble methods can often achieve better performance than individual machine learning models.

• **Use cross-validation.**

Cross-validation is a technique for evaluating the performance of a

machine learning model on unseen data. It is important to use cross

validation to evaluate the performance of your model during the

evaluation process. This will help you to avoid overfitting and to ensure that the model will generalize well to new data.

• **Use a holdout test set.**

A holdout test set is a set of data that is not used to train or

evaluate the model during the training process. This data is used to

evaluate the performance of the model on unseen data after the training process is complete.

• **Compare the model to a baseline.**

A baseline is a simple model that is used to compare the

performance of your model to. For example, you could use the mean

Air Quality analysis as a baseline.

• **Analyze the model's predictions.**

Once you have evaluated the performance of the model, you can

analyze the model's predictions to identify any patterns or biases. This will help you to understand the strengths and weaknesses of the model and to improve it.

**Conclusion:**

In the quest to build an accurate and reliable Air Quality analysis

prediction model, we have embarked on a journey that encompasses

critical phases, from feature selection to model training and evaluation. Each of these stages plays an indispensable role in crafting a model that can provide meaningful insights and estimates for one of the most significant decisions individuals and businesses make—real estate transactions.

• Model training is where the model's predictive power is forged. We have explored a variety of regression techniques, fine-tuning their parameters to learn from historical data patterns. This step allows the model to capture the intricate relationships between features and Air Quality analysis, giving it the ability to generalize beyond the training dataset.

• Finally, model evaluation is the litmus test for our predictive prowess. Using metrics like *Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, and R-squared*, we've quantified the model's performance. This phase provides us with the confidence to trust the model's predictions and assess its ability to adapt to unseen data.

• In the ever-evolving world of real estate and finance, a robust Air Quality analysis prediction model is an invaluable tool. It aids *buyers, sellers, and investors* in making informed decisions, mitigating risks, and seizing opportunities. As more data becomes available and market dynamics change, the model can be retrained and refined to maintain its accuracy.