AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU

ABSTRACT

- *Abstract. Prediction of air pollution index may help in traffic routing and identifying serious pollutants.*
- *Modeling of the complex relationships between these variables by sophisticated methods in machine learning is a promising field.*
- *The quality of air in Alandur, Chennai is polluted*
- *by Particulate Matter (PM2.5) over the years. Reports prove*

INTRODUCTION

- *Technological advancements lead to the emissions of airpollutants over the decades.*
- *Air pollutants such as sulphur dioxide (SO2), nitrogenoxide (Nox), nitric oxide (NO), nitrogen dioxide (NO2),*
- *carbon monoxide (CO), Ozone (O3), respirable suspendedparticulates (RSPs) are some of the major airborne pollutants*
- *which exerts impact on physical and biological environmen*
- *Chennai, the capital of Tamil Nadu in India is located onthe Coromandel Coast off the Bay of Bengal. It is theeconomic and educational centre of south India*

AIM

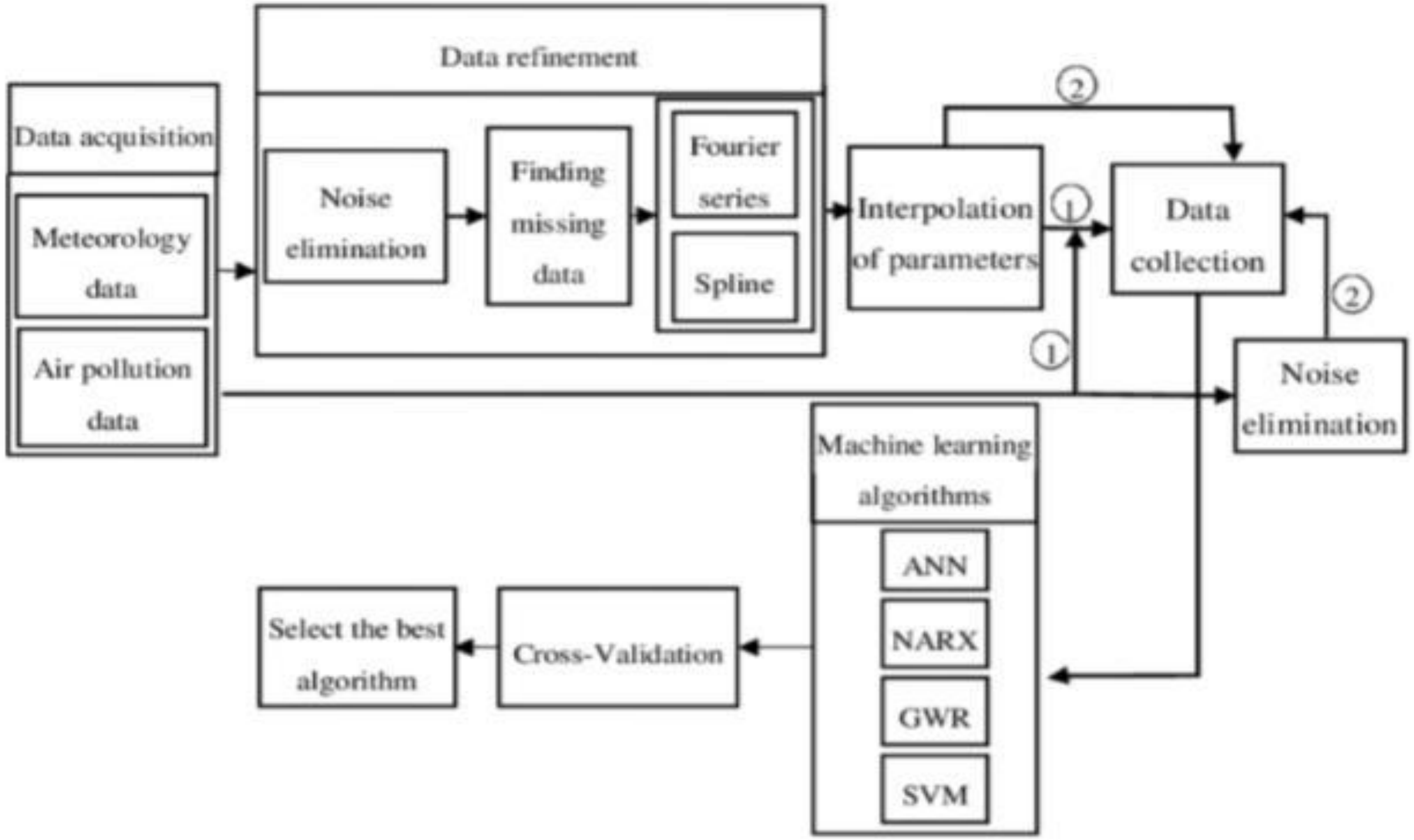
- *The objective of study is to implement air quality prediction*
- *with machine learning algorithes namely logistic regressio*

and deep learning techniques such as Neural Network

CONCLUSION

- *TheproposedapproachforAirqualitypredictionbasedon meteorologicalandhistoricalpollutantdata.*
- *usingamodelbasedonthepreviousmeteorologicaldata.Machinelearningalg orithmLogisticregressionisusedfor prediction.*
- *results,logisticregressionpredictswithaccuracyof66%*

AIR PREDICTION ARCHITECTURE BLACK DIAGRAM



CODE IMPLEMENTATION

- Import matplotlib.pyplot as plt
- From matplotlib.animation import FuncAnimation
- Import random

- # Generate some sample air quality data (replace with your own data)
- Timestamps = range(1, 11)
- Pm25_values = [random.randint(0, 50) for _ in timestamps]

- # Initialize the plot
- Fig, ax = plt.subplots()
- Ax.set_xlim(1, 10)
- Ax.set_ylim(0, 50)
- Line, = ax.plot([], [], lw=2)

- # Function to update the animation at each time step
- Def animate(i):
-     line.set_data(timestamps[:i], pm25_values[:i])
-     return line,

PREPROCESSING DATASET EXAMPLE: CODE

- *# importing pandas module for data frame*
- *Import pandas as pd*

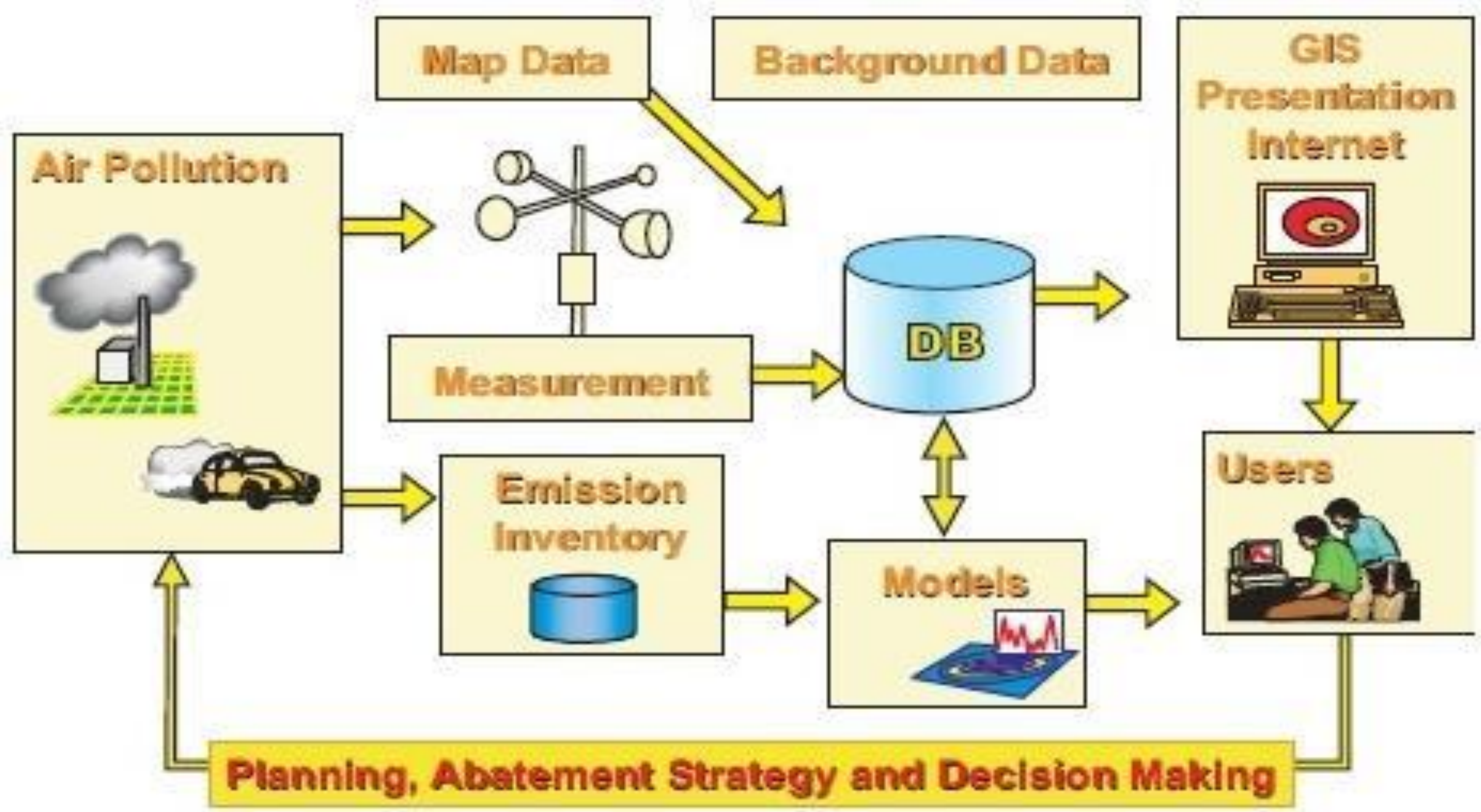
- *# loading dataset and storing in train variable*
- *Train=pd.read_csv('AQI.csv')*

- *# display top 5 data*
- *Train.head()*

OUTPUT:

PM2.5 AVG	PM10 AVG	NO2 AVG	NH3 AVG	SO2 AG	CO	OZONE AVG
190	131	107	4	42	0	63
188	131	110	4	40	0	62
280	174	155	2	37	0	52
302	181	144	2	39	0	78

AIR QUALITY ANALYSIS ANIMATION CIRCUIT



PREPROCESSING DATASET

- *Data Preprocessing:*

Clean and preprocess the data to remove any inconsistencies, missing values, and outliers. You may need to handle date and time data, as well as convert categorical variables into numerical representations.

- *Data Exploration:*

Perform exploratory data analysis (EDA) to understand the characteristics of the dataset. Visualizations and summarstatistics can help you identify patterns and trends.

“Feature Engineering Create relevant features that can be used in your air quality prediction model. This might include weather data, geographical coordinates, or historical air quality mmeasurement

- *Split the Data:*

Split your dataset into training, validation, and testing sets. The training set is used to train your model, the validation set is used for tuning hyperparameters, and the testing set is used to evaluate the model's performance.

LOADING DATASET

- *Data Collection:*

Gather historical air quality data for Tamilnadu. You can obtain this data from government agencies, research institutions, or environmental monitoring stations.

Collect meteorological data such as temperature, humidity, wind speed, and wind direction, which can impact air quality.

- *Exploratory Data Analysis (EDA):*

Visualize the data to identify trends, patterns, and correlations. EDA helps in understanding the data and identifying potential factors influencing air quality.

- *Feature Engineering:*

Create relevant features based on domain knowledge. For example, consider lag variables, seasonal patterns, and interactions between variables.

- *Data Splitting:*

Split the dataset into training, validation, and test sets to evaluate the performance of predictive mo

Dataset contains in features

- *Stn_code : Station code. A code is given to each station that recorded the Data.*
- *Sampling_date: The date when the data was recorded.*
- *State: It represents the states whose air quality data is measured.*
- *Location: It represents the city whose air quality data is measured.*
- *Agency: Name of the agency that measured the data.*

DATA EXPLORATION

- *Let us get some insights about the data — the number of entries in each*
- *column, the type of entry in each column*
- *It represents the type of area where the data was recorded like industrial, residential, etc.*
- *Let us see how many types of area*
- *NO, because the agency's name has nothing to do with how much polluted the*
- *state is. Similarly, stn_code is also unnecessary.*
- *It is given in the data description that date*

DATA EXPLORATION CODE :

- *# Rows with missing “types*
- *null_data = tn[tn.isnull().any(axis=1)]*
- *null_data.head(20)*

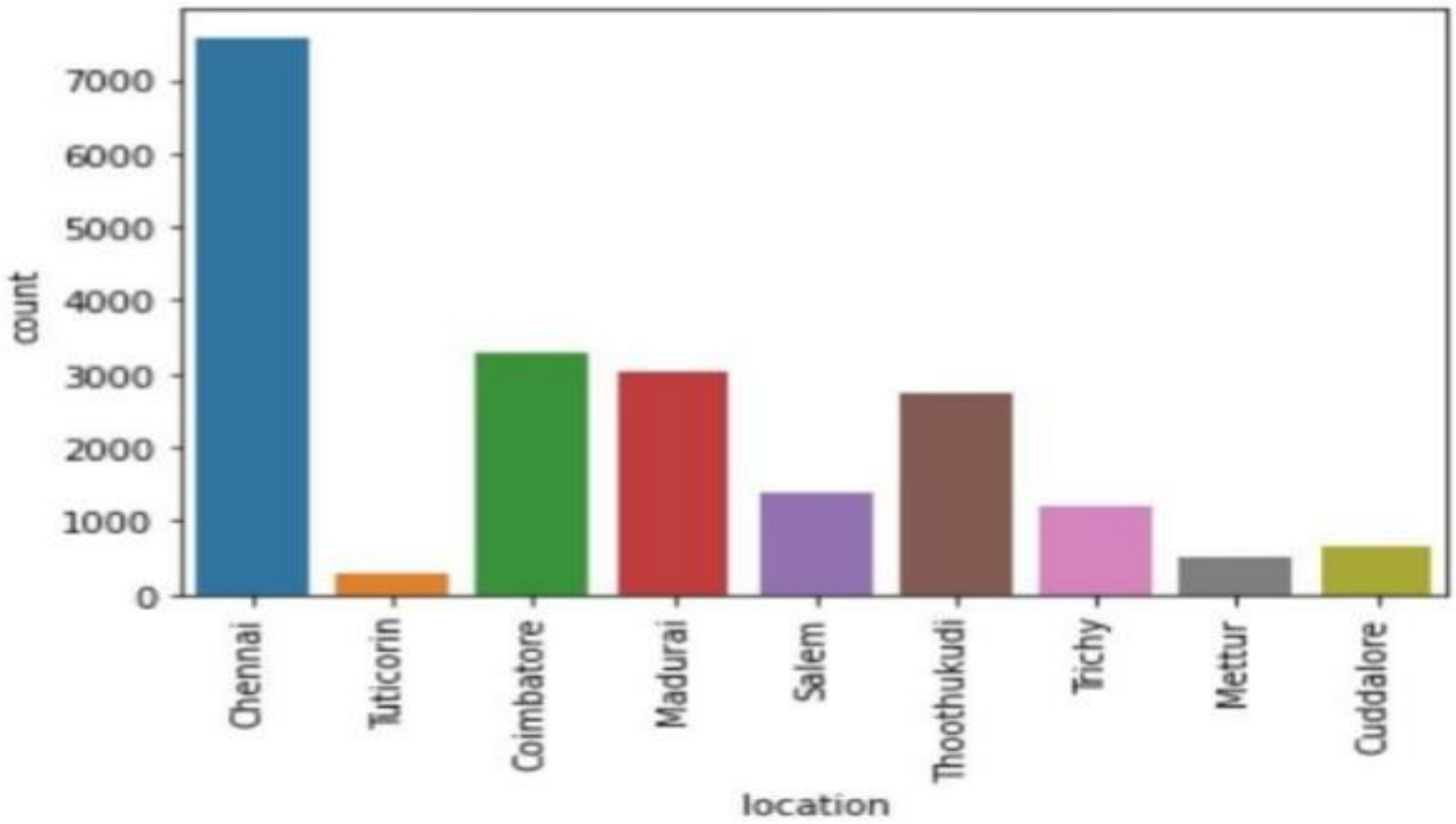
OUTPUT :

state	location	type	so2
343962	Tamil Nadu	Tuticorin	NaN
343984	Tamil Nadu	Tuticorin	NaN
343985	Tamil Nadu	Tuticorin	NaN
343986	Tamil Nadu	Tuticorin	NaN
344142	Tamil Nadu	Tuticorin	NaN
344143	Tamil Nadu	Tuticorin	NaN
344147	Tamil Nadu	Tuticorin	NaN
344148	Tamil Nadu	Tuticorin	NaN
344155	Tamil Nadu	Coimbatore	NaN
344156	Tamil Nadu	Coimbatore	NaN
344157	Tamil Nadu	Coimbatore	NaN
344158	Tamil Nadu	Coimbatore	NaN
344159	Tamil Nadu	Coimbatore	NaN
344160	Tamil Nadu	Coimbatore	NaN
344161	Tamil Nadu	Coimbatore	NaN
344162	Tamil Nadu	Coimbatore	NaN
344163	Tamil Nadu	Coimbatore	NaN
344164	Tamil Nadu	Coimbatore	NaN
344165	Tamil Nadu	Coimbatore	NaN
344166	Tamil Nadu	Coimbatore	NaN

DATA VISUALIZATION

- *Input:*
- *datacount =sns.countplot(x =“location”,data = tn);*
- *datacount.set_xticklabels(datacount.get_xticklabels(), rotation=90);*

OUTPUT:



7000
6000
5000
4000
3000
2000
1000
0
count
Chennai
Tuticorin
Coimbatore
Madurai
Salem
Thoothukudi
Trichy
Mettur
Cuddalore
location

CALCULATE IN AIR QUALITY

- *The AQI is calculated by converting measured pollutant concentrations to a uniform index which is based on the health effects associated with a Pollutant*
- *The health benchmarks used for calculating the AQI are pollutant specific and are established by the EPA through the National Ambient Air Quality Standards*

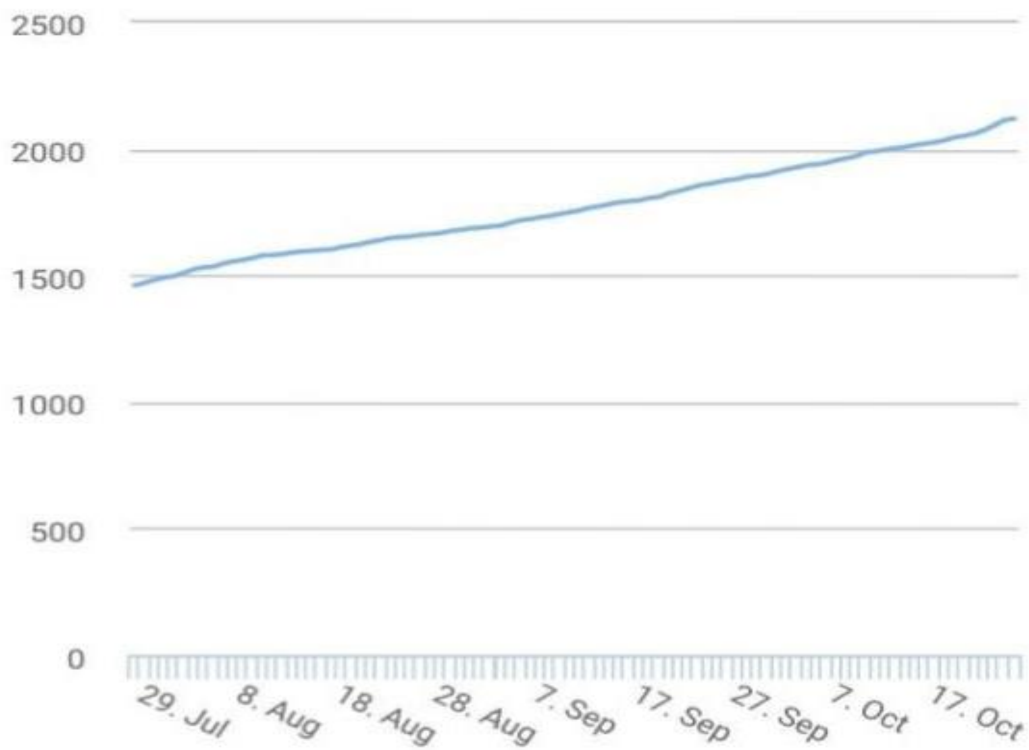
INPUT:

```
#Function to calculate  so2 individual pollutant  index(si)
def calculate_si(so2):
   si=0
   if (so2<=40):
    si= so2*(50/40)
   if (so2>40 and so2<=80):
    si= 50+(so2-40)*(50/40)
   if (so2>80 and so2<=380):
    si= 100+(so2-80)*(100/300)
   if (so2>380 and so2<=800):
    si= 200+(so2-380)*(100/800)
   if (so2>800 and so2<=1600):
    si= 300+(so2-800)*(100/800)
   if (so2>1600):
    si= 400+(so2-1600)*(100/800)
   return si
data['si']=data['so2'].apply(calculate_si)
df= data[['so2','si']]
df.head()
```

OUTPUT:

- So2 si
- 0 4.8 6.000
- 1 3.1 3.875
- 2 6.2 7.750
- 3 6.3 7.875
- 4 4.7 5.875

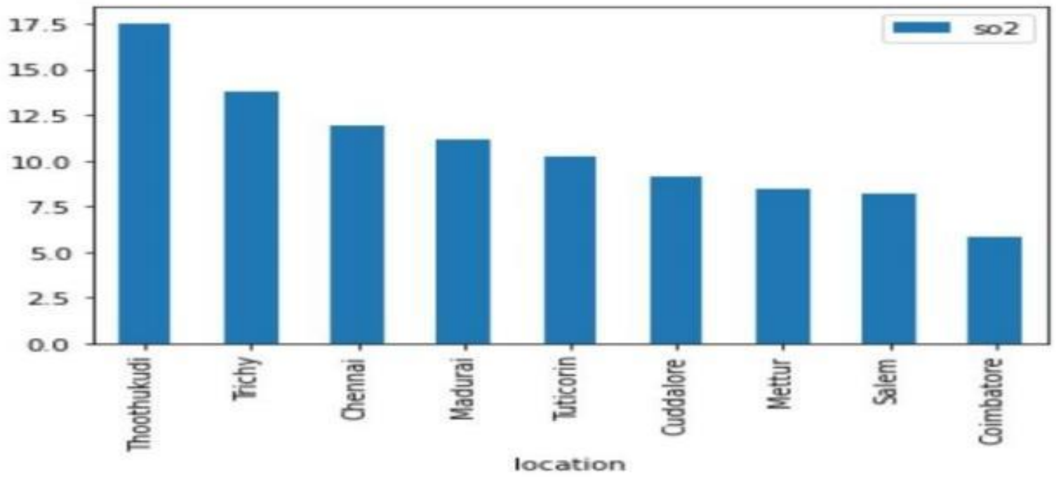
Article Metrics (level)



DISTRICT LEVEL PREDICTION

- ***INPUT : CODE***
- ***maxso2 = loc.sort_values(by='so2',ascending=False)***
- ***maxso2.loc[:,['so2']].head(10).plot(kind='bar'); # Based on average values***

OUPUT :



Air quality visualization

- *The graphical display of data – helps us understand the distribution of air pollutants in the atmosphere. This is hard to do just by looking at a modern air monitor equipment with its digital display*
  By combining real-time monitoring data with python programming, one can easily visualize air monitoring data. Interactive graphs can be created which makes it easier to check air quality, and increasingly diverse colors can visually highlight the air quality level. Visualization of data has a resilient expression (more images and more insightful) than the original data table, which is favorable for further analysis of data.

OUTPUT FOR VISUALIZATION: