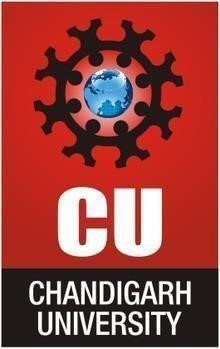
**CHANDIGARH UNIVERSITY**

**UNIVERSITY INSTITUTE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



|  |  |
| --- | --- |
| **Submitted by:** | **Submitted To:** Ajay Kumar (E13141) |
| **Subject Name:** | Machine Learning Lab |
| **Subject Code:** | 20CSP-317 |
| **Branch:** | CSE |
| **Semester:** | 5th |

LAB INDEX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr.  No | Program | Date |  | Evaluation | | | Sign |
| LW  (12) | VV  (8) | FW  (10) | Total (30) |
| 1. |  |  |  |  |  |  |  |
| 2. |  |  |  |  |  |  |  |
| 3. |  |  |  |  |  |  |  |
| 4. |  |  |  |  |  |  |  |
| 5. |  |  |  |  |  |  |  |

# 

# Experiment-2

Aim/Overview of the practical: To perform Data Visualization

Code and output:

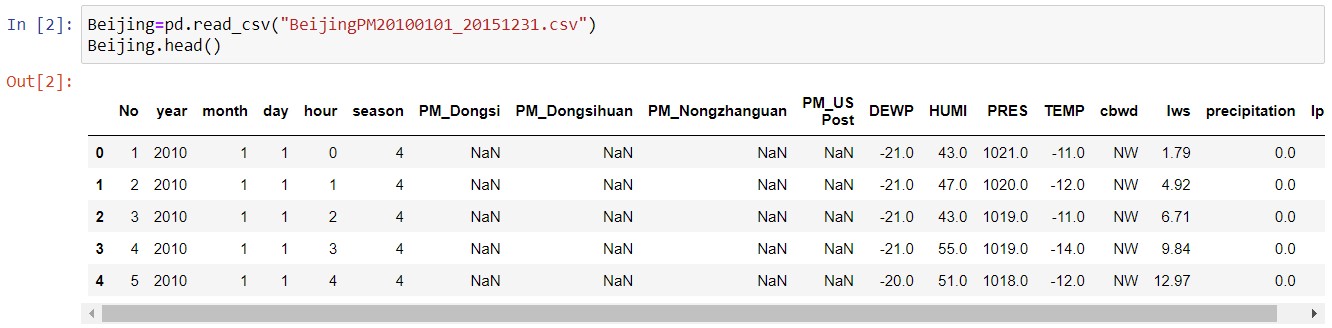
# Importing Libraries

import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns

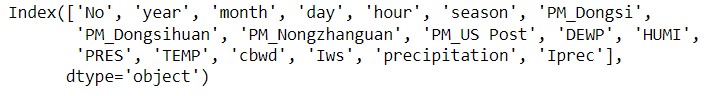
# Reading the data

Beijing=pd.read\_csv("BeijingPM20100101\_20151231.csv") Beijing.head()

Beijing.shape



Beijing.columns



# Calculating the percentage of NaN values in the Data set

Beijing.isnull().sum()



Beijing.isnull().mean()\*100



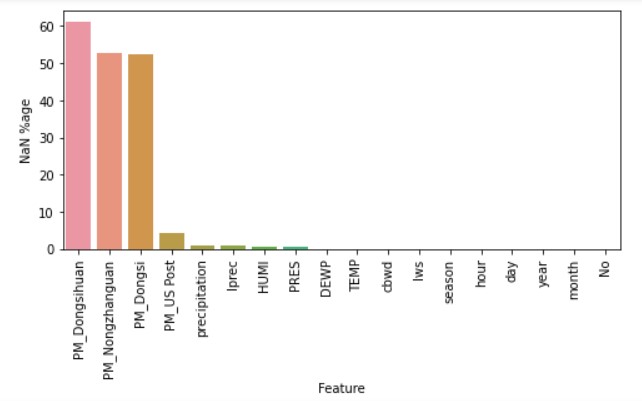
NaN\_percentage = pd.DataFrame(Beijing.isnull().mean()\*100,columns=["NaN %age"]).reset\_index().sort\_values(by='NaN %age',ascending=False)

NaN\_percentage.rename(columns={"index":"Feature"},inplace=True)

# Visualization for dropping NaN values

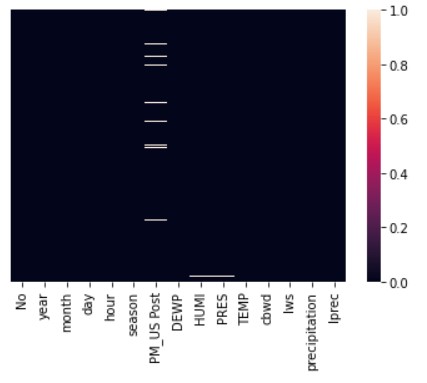
plt.figure(figsize=(8,3.5))

sns.barplot(x="Feature",y="NaN %age",data=NaN\_percentage) plt.xticks(rotation=90)



for f in Beijing.columns:

if(Beijing[f].isnull().mean()\*100>30): Beijing.drop(f,inplace=True,axis=1)



Beijing.dropna(inplace=True)

Beijing.shape

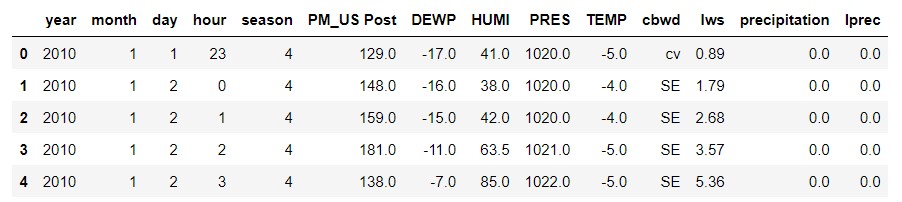


Beijing.reset\_index(inplace=True)

#Dropping unecessary features

Beijing.drop(["index","No"],axis=1,inplace=True)

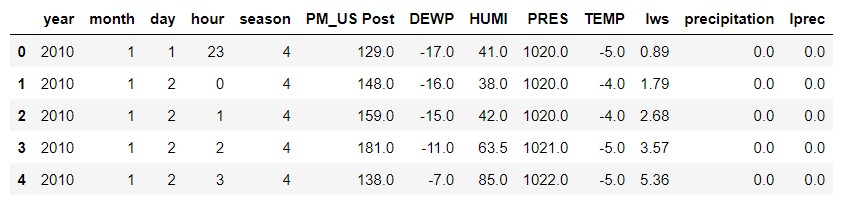
Beijing.head()



#Data processing on numerical features

Beijing\_numerical=Beijing.select\_dtypes(exclude="object").copy()

Beijing\_numerical.head()

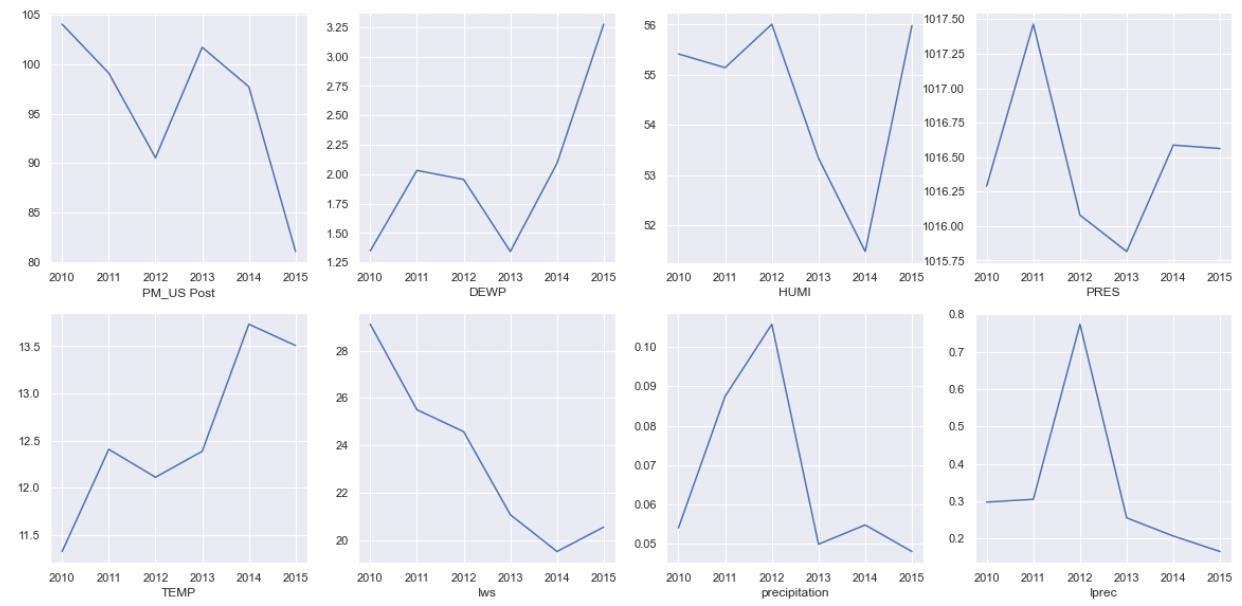


# Visualizing the Time series data for Yearly trends

f = ["year","hour","month","day","season"] sns.set() plt.figure(figsize=(20,20)) for i,c in enumerate(Beijing\_numerical.drop(f,axis=1).columns): if c not in f:

plt.subplot(4,4,i+1)

plt.plot(Beijing\_numerical.groupby("year").mean()[c]) plt.xlabel(c)



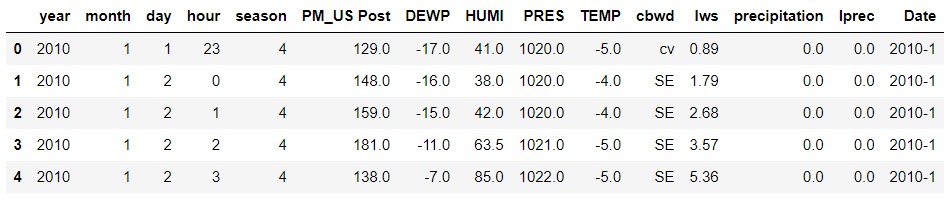
# Preparing time series data for visulizing Monthly trends dates=[]

for i in range(Beijing.shape[0]): lst=[str(Beijing["year"][i]),str(Beijing["month"][i])] st="-" s=st.join(lst)

dates.append(s)

Beijing["Date"]=dates

Beijing.head()

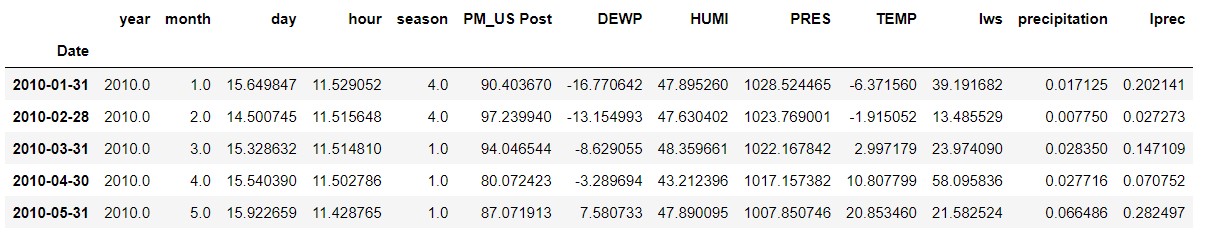


Beijing["Date"]=pd.to\_datetime(Beijing["Date"])

Beijing.head()



Beijing\_dates=Beijing.groupby(pd.Grouper(key='Date', axis=0, freq='M')).mean() Beijing\_dates.head()

 f = ["day","month","year","hour","season"]

plt.figure(figsize=(20,20)) for i,c in enumerate(Beijing\_dates.drop(f,axis=1).columns): if c not in f:

plt.subplot(4,4,i+1) plt.plot(Beijing\_dates[c]) plt.xlabel(c)

