# Mumbai Power Demand Forecasting

Problem Statement = Analyze the Pattern affecting the power consumption In Mumbai Region anf forecasting upcoming Power consumption result

**Dataset Source** [**: https://www.kaggle.com/datasets/twinkle0705/state-wise-power-consumptionin-india**](file:///D:\Round_2_hackathon\:%20https:\www.kaggle.com\datasets\twinkle0705\state-wise-power-consumptionin-india)

* **Flow Structure for Building Predictive Model**

Data Extraction 🡪 Data Cleaning 🡪 Exploring Patterns 🡪Statistical Testing 🡪 Building Predictive Model 🡪 Forecasting Results

Index –

1. Extracting Data from Mysql Database to CSV file

import pymysql

import  mysql.connector as mysql\_connection

import pandas as pd

import os

#fetching the Database credentials for environmen file

MYSQL\_USERNAME = os.getenv("MYSQL\_USERNAME")

MYSQL\_PASSWORD = os.getenv("MYSQL\_PASSWORD")

MYSQL\_SERVER = os.getenv("MYSQL\_SERVER")

MYSQL\_PORT = os.getenv("MYSQl\_PORT")

MYSQL\_DATABASE = os.getenv("MYSQL\_DATABASE")

#checking the Database credentials by creating an list

Database\_credentials = [

    MYSQL\_USERNAME,MYSQL\_PASSWORD,MYSQL\_SERVER,MYSQL\_PORT,MYSQL\_DATABASE

                        ]

if all(Database\_credentials):

    print(" Database credentials is imported")

else:

    print(" Database credentials not imported")

#making an connection to database with its credentials

Connection = pymysql.connect(

    host=MYSQL\_SERVER,port=int(MYSQL\_PORT),database=MYSQL\_DATABASE,

    user=MYSQL\_USERNAME,password=MYSQL\_PASSWORD)

if Connection:

    print("Database connected successfully")

else:

    print("Database not connected successfully")

#creating an cursor object for reading and SQL Query

cursor\_connection = Connection.cursor()

if cursor\_connection:

    sql\_statement = "SELECT \* FROM long\_data\_ WHERE States = 'Maharashtra'"

    cursor\_connection.execute(sql\_statement)

    result = cursor\_connection.fetchall()

    desc = [desc[0] for desc in cursor\_connection.description]

    Connection.commit()

    cursor\_connection.close()

#converting the reuslt into pandas dataframe

try:

    demand\_forecasting\_df = pd.DataFrame(result,columns=desc)

except Exception as e:

    print("error : {e}")

finally:

    print(demand\_forecasting\_df)

#saving these dataframe into folder structure

demand\_forecasting\_df.to\_csv('dataset.csv')

1. Basic Information regarding Dataset

**Data Type of each columns**

|-- \_c0: integer (nullable = true)

|-- States: string (nullable = true)

|-- Regions: string (nullable = true)

|-- latitude: double (nullable = true)

|-- longitude: double (nullable = true)

|-- Dates: string (nullable = true)

|-- Usage: double (nullable = true)

**Memory used by each columns in dataset** :

Index 132

\_c0 2012

States 34204

Regions 29677

latitude 4024

longitude 4024

Dates 38228

Usage 4024

dtype: int64

**Memory used by entire dataset** : 116325

**Size of each columns in dataset** : 3521

**number of rows and columns in dataset** : (503, 7)

**number of rows in dataset** : 503

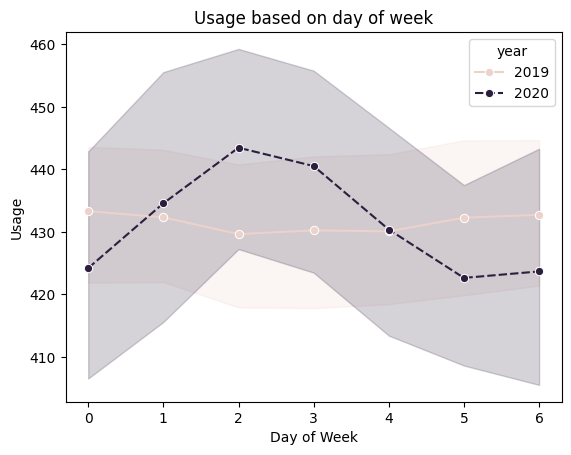
**number of columns in dataset** : 7

1. **Descriptive Statistic Result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Statistics | Index | latitude longitude Usage | Longitude | Usage |
| count | 503 | 503 | 503 | 503 |
| mean | 251 | 19.25023 | 73.16017 | 431.5702 |
| std | 145.3479 | 5.33E-14 | 7.11E-14 | 42.25227 |
| min | 0 | 19.25023 | 73.16017 | 305.6 |
| 25% | 125.5 | 19.25023 | 73.16017 | 401.7 |
| 50% | 251 | 19.25023 | 73.16017 | 429 |
| 75% | 376.5 | 19.25023 | 73.16017 | 463.75 |
| max | 502 | 19.25023 | 73.16017 | 522.1 |
|  |  |  |  |  |

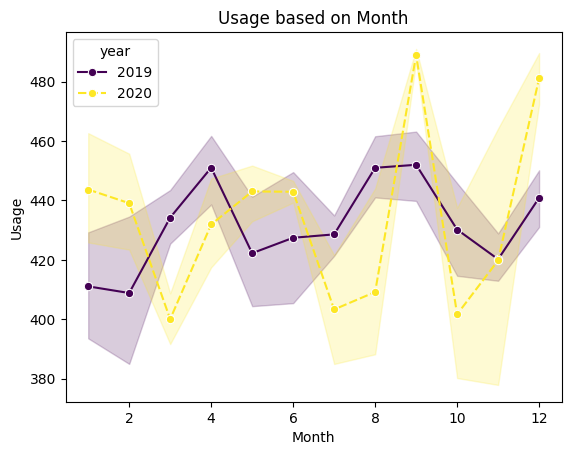
* Exploring Consumption Factors

1. Power consumption Factors Day of week Wise for each year



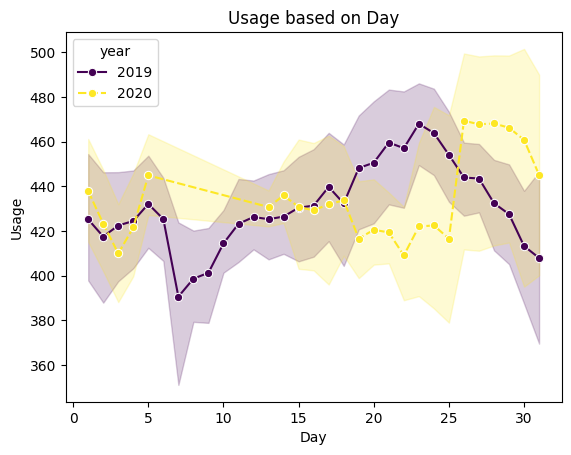
Observation = There has slightly decrease in power consumption in 2020 and increase int 2019 consumption slightly increase

1. Power consumption month wise in transition of 2019 to 2020



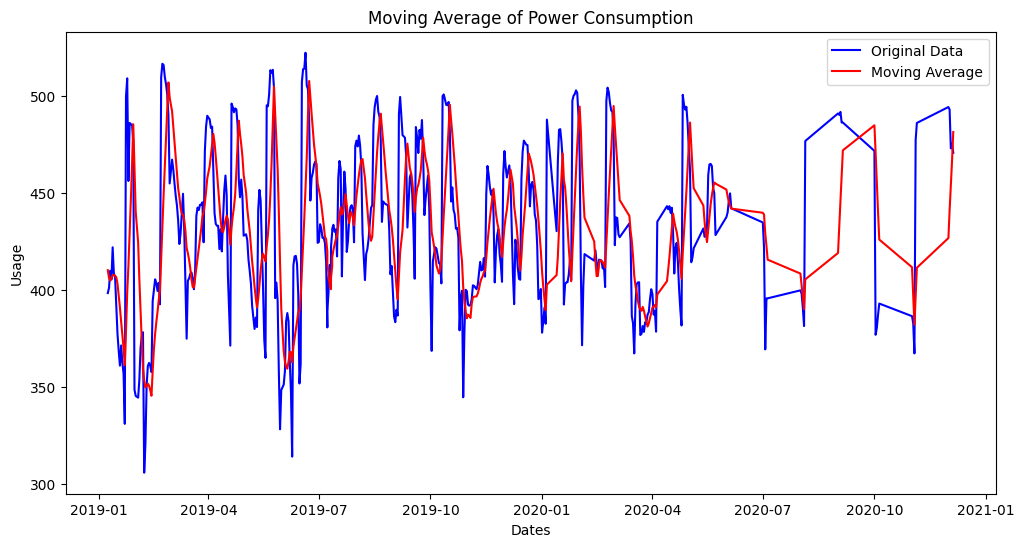
Observation = There has been drastically impact of power consumption between 8-10 month in 2020

1. Power consumption Day wise



Observation = There has been increase in Power consumption in year 2019 between 10-25 but decrease in 2020

1. Power consumption with Moving Averages for past dayas 7



Observation = During the last month of year there this fluctuation of moving Averages after 7 days in power consumption

1. Average Power consumption used in Mumbai

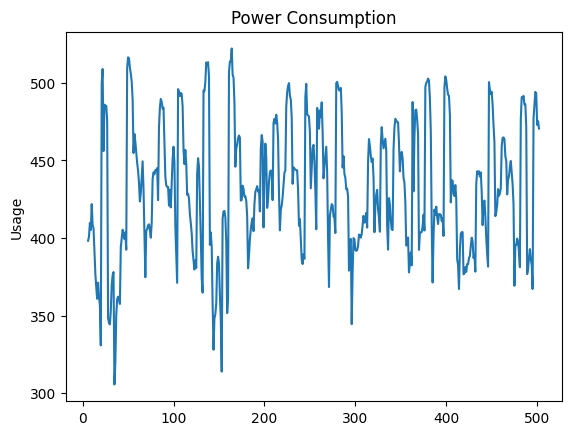
|  |  |
| --- | --- |
| States | Average\_Usage |
|  |  |
| Mumbai | 431.570179 |

* Statistical testing = Conducted ADF test to check whether power consumption is in stationary or not
  + Result for ADF statistics
* significance\_value = 0.05
* def check\_adf\_test(demand\_df):
* result = adfuller(demand\_df['Usage'])
* return result[1]
* p\_value = check\_adf\_test(demand\_df)
* if p\_value <= significance\_value:
* print("The data is stationary")
* else:
* print("The data is not stationary")

Result = data is stationary

Power consumption is moving linearly no gaps between Power consumption

Visualization of Power consumption



* Building ARIMA Model for Forecasting Power Usage

**SARIMAX Results**

**==============================================================================**

**Dep. Variable: Usage No. Observations: 503**

**Model: ARIMA(1, 1, 1) Log Likelihood -2348.835**

**Date: Sat, 01 Mar 2025 AIC 4703.671**

**Time: 08:47:50 BIC 4716.327**

**Sample: 0 HQIC 4708.636**

**- 503**

**Covariance Type: opg**

**==============================================================================**

**coef std err z P>|z| [0.025 0.975]**

**------------------------------------------------------------------------------**

**ar.L1 0.7918 0.036 22.139 0.000 0.722 0.862**

**ma.L1 -0.9987 0.040 -25.099 0.000 -1.077 -0.921**

**sigma2 673.7286 35.305 19.083 0.000 604.532 742.925**

**===================================================================================**

**Ljung-Box (L1) (Q): 2.68 Jarque-Bera (JB): 1619.78**

**Prob(Q): 0.10 Prob(JB): 0.00**

**Heteroskedasticity (H): 0.66 Skew: 1.37**

* + **Prob(H) (two-sided): 0.01 Kurtosis: 11.36**
* Visualizing the Forecasted Results

