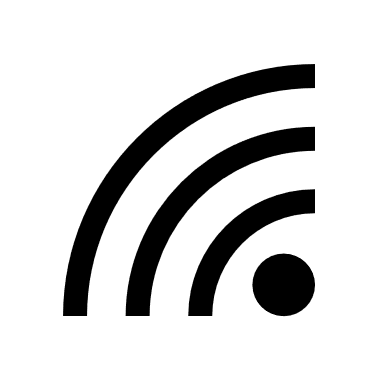
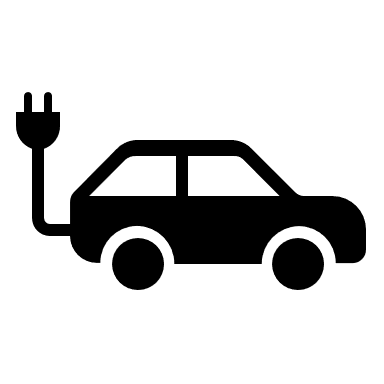


Power Consumption Analysis using IBM Watson Machine Learning Service

Project Report





BY: EEE-03

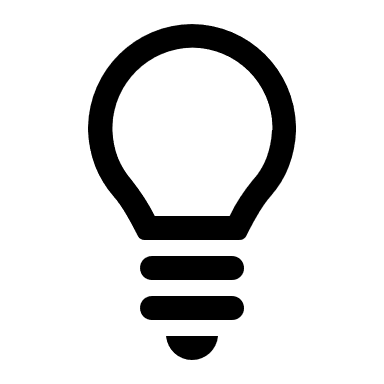
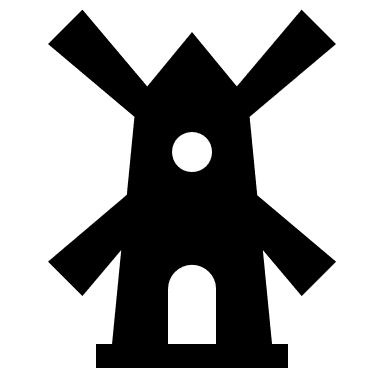
Udith

Sai Kumar

Uber

Sandeep

Vinay

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S No** | **TOPIC** | **PAGE** **No** |
| **1** | **INTRODUCITON** | **3** |
| **2** | **LITERATURE SURVEY** | **4** |
| **3** | **THEORETICAL SURVEY** | **4** |
| **4** | **EXPERIMENTAL INVESTIGATION** | **7** |
| **5** | **FLOWCHART** | **8** |
| **6** | **RESULT** | **10** |
| **7** | **ADVANTAGES AND DISADVANTAGES** | **10** |
| **8** | **APPLICATIONS** | **11** |
| **9** | **CONCLUSION** | **11** |
| **10** | **FUTURE SCOPE** | **12** |
| **11** | **BIBILOGRAPHY** | **12** |

1.INTRODUCTION

* 1. **Preview**

This project is done by using IBM Cloud’s Watson Machine Learning Service. As the name of the project suggests it is about the power consumption of households. With the increase in population and rapidly depleting natural non-renewable resources the management of resources plays an important part. One such important sector managing the resources is power sector. Predicting the required usage will definitely give a push to use the resources carefully. Sustainable use of resources is the end target which is currently not the scenario in this growing world. The prediction is done by using the past data; which is all stored in a dataset.

The dataset used for this project is a multivariate time series dataset that describes the electricity consumption for a single household over four years. The data was collected between December 2006 and November 2010 and observations of power consumption within the household were collected every minute. It is a multivariate series comprised of seven variables (besides the date and time) they are:

Global Active Power: The total active power consumed by the household (kilowatts).

Global Reactive Power: The total reactive power consumed by the household (kilowatts).

Voltage: Average voltage (volts).

Global Intensity: Average current intensity (amps).

Sub Metering 1: Active energy for kitchen (watt-hours of active energy).

Sub Metering 2: Active energy for laundry (watt-hours of active energy).

Sub Metering 3: Active energy for climate control systems (watt-hours of active energy).

Now, using this dataset after completing the required steps (building model, writing HTML, Python codes etc..) the Global Active Power is calculated by using the other required variables in the dataset which is coded in the app.

* 1. **Purpose**

This project can be used to find the Global Active power. Global active power is the realpower consumption i.e., the power consumed by electrical appliances other than the sub metered appliances. This will help to give a check on household whether they’re using the electric power sustainably or not.

2. LITERATURE SURVEY

* 1. **Existing problem**

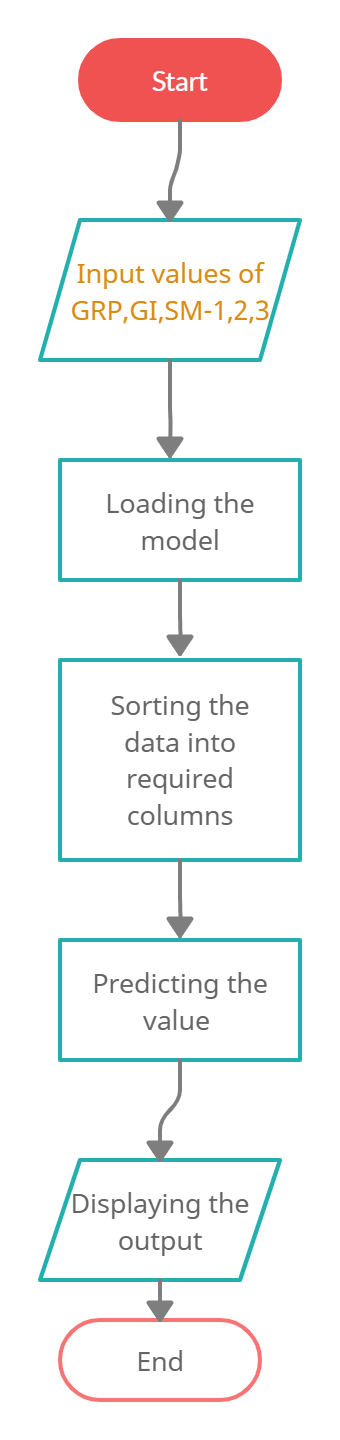
Energy management systems are being used to control this problem of wastage of power, which includes energy usage analysis, energy tariff optimization, Asset Management optimization, alarm and event logging. Building the Automation and Control systems is he engineering way of dealing with the problem. Even the power factor control, Lighting control and HVAC control are widely used to solve the problem. Power measurement is one of the widely used method to solve this problem.

* 1. **Proposed Solution**

Most of the methods used require lot of time and few of also require a bit of money put on it. Our model helps to do it easily by using the global active power value predicted from the dataset. We used the linear regression model and accuracy was 99%, so, it’s trustworthy too.

1. **THEORITICAL ANALYSIS**

**3.1 Block diagram**

****

GRP- Global Reactive Power

GI- Global Intensity

SM 1,2,3- Sub Metering 1(2,3)

**3.2 Hardware/Software Design**

Software:

* We used Python 3 for writing the codes.
* Anaconda is the prompt used for opening the Jupyter Notebook.
* Jupyter notebook is used to write and execute the code to develop the model.
* We used spyder to save .py files and run the project.
* Notepad for writing the HTML codes
* Google Chrome for running on the local host

Hardware:

* Processor: Intel® Core™ i3-2350M CPU @ 2.30GHz
* Installed memory (RAM):4.00GB
* System Type: 64-bit Operating System

1. **EXPERIMENTAL INVESTIGATIONS**

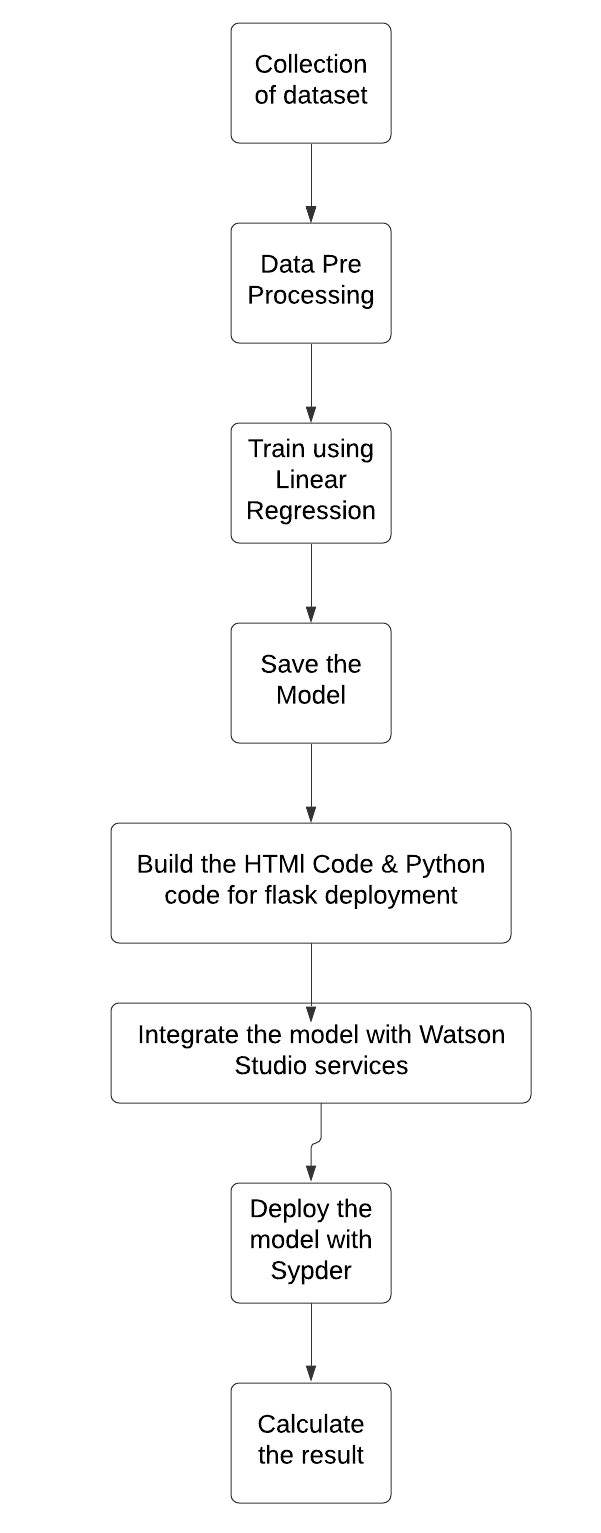
The base which we took is Artificial Intelligence and Machine Learning.

Artificial Intelligence (AI) is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristics of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience.

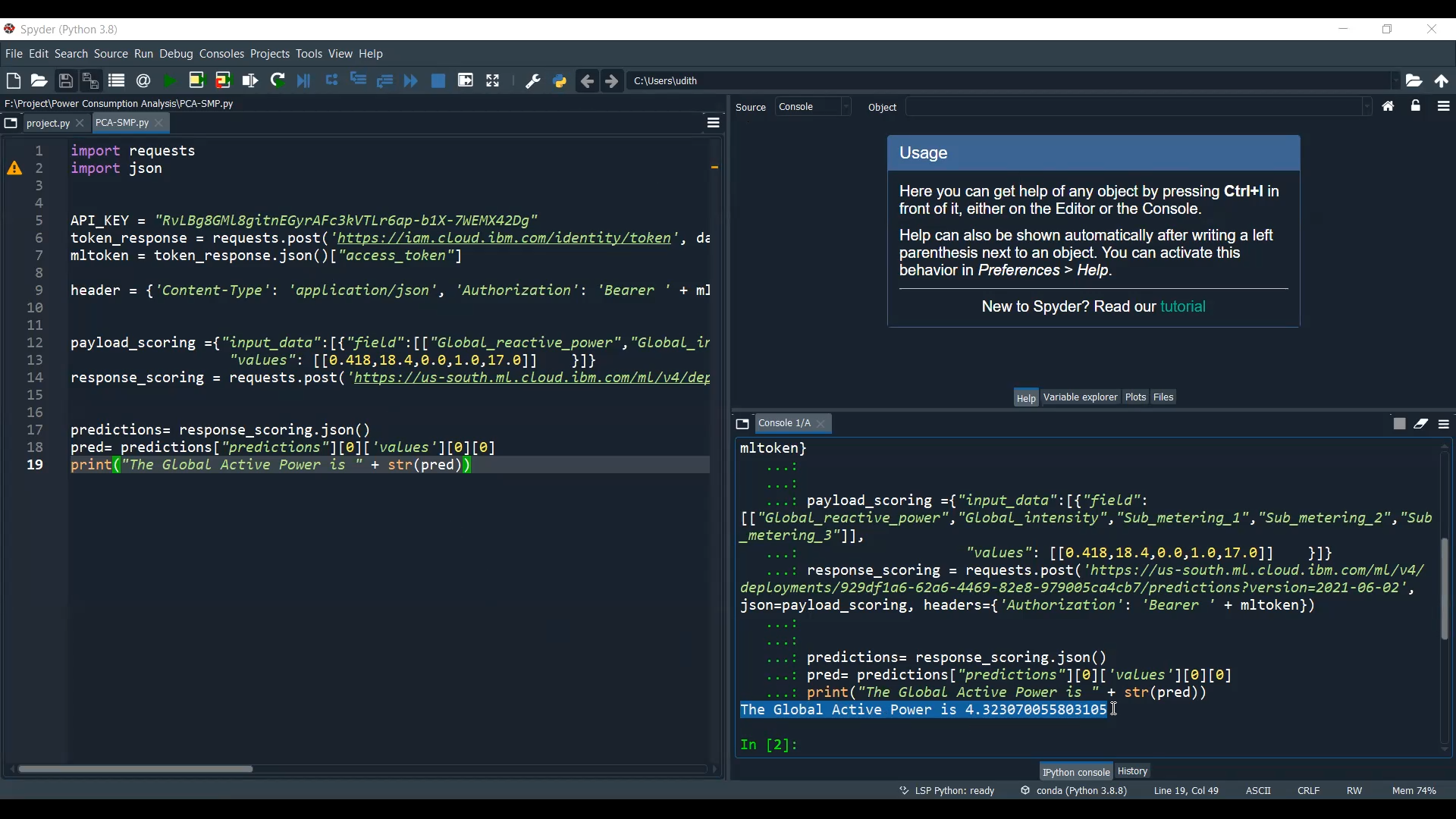
Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it to learn for themselves. We decided to use linear regression in machine learning.

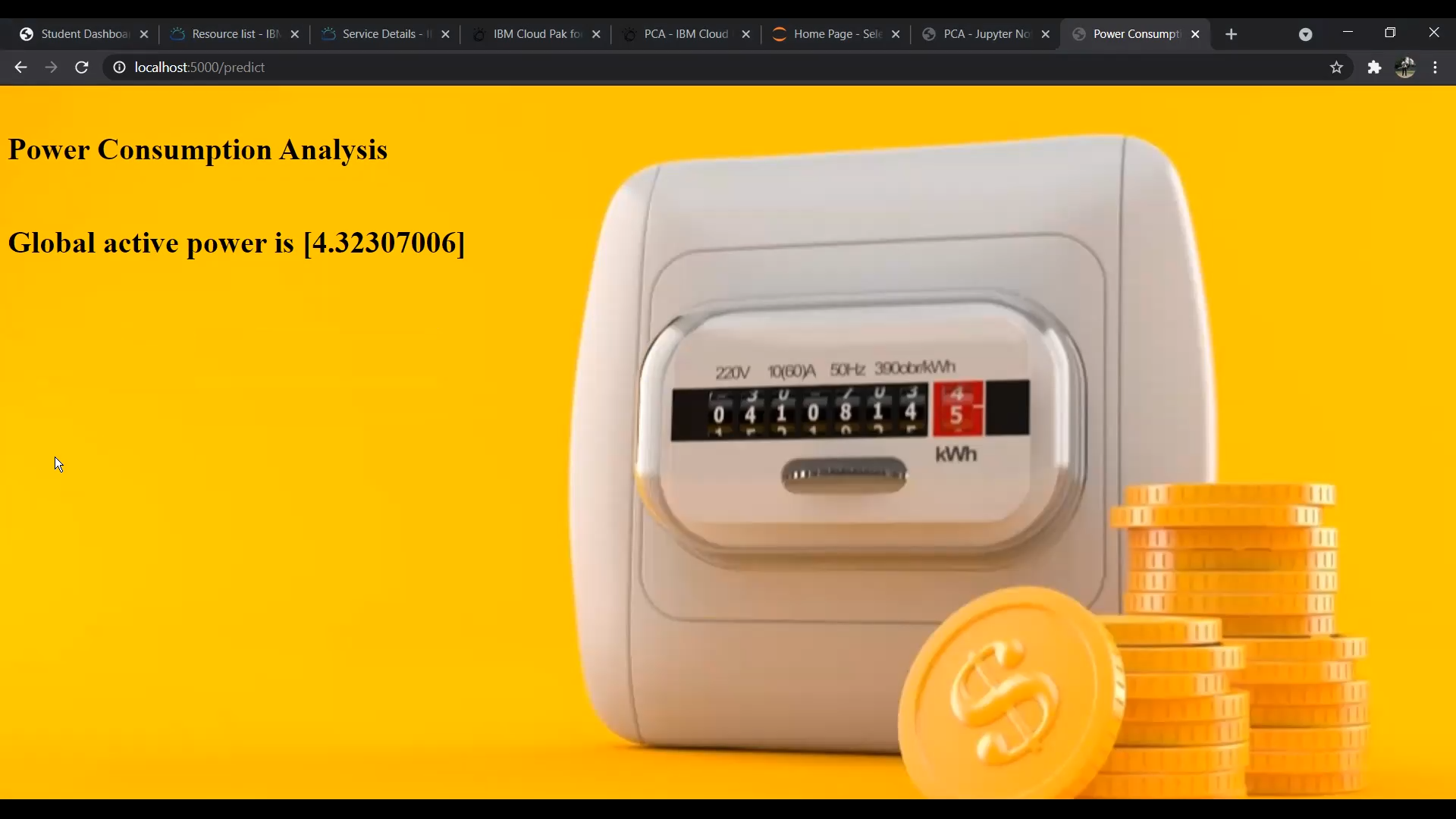
**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

**5.FLOWCHART**



**6.RESULT**





**7.ADVANTAGES & DISADVANTAGES**

Advantages:

* Easier to implement, interpret and efficient to train
* Overfitting can be reduced by regularization
* Linear regression performs exceptionally well for linearly separable data
* It handles overfitting pretty well using dimensionally reduction techniques, regularization, and cross-validation
* One more advantage is the extrapolation beyond a specific data set

Disadvantages:

* It is prone to multicollinearity
* The assumption of linearity between dependent and independent variables
* It is often quite prone to noise and overfitting
* Linear regression is quite sensitive to outliers

**8.APPLICATIONS**

As mentioned earlier using this project, the global active power can be calculated which in turn helps in power consumption analysis.

With such data, the power consumption of individual households can be tracked in almost real-time. Such prediction can help power companies regulate their supply; also, the consumer can use this information to make better decisions both financially and environment-consciously.

**9.CONCLUSION**

To summarize the whole findings by starting with its advantages:

The results of the project can be applied in different fields. For example, knowing the amounts of energy consumption is of great importance for several reasons. First of all, for consumers of electrical energy knowledge about the electric load and the targeted is important for understanding their bills and better controlling their consumption. For organizations it is also useful to know periods of minimum and maximum of the consumption for planning the

technological cycles, for planning budget costs. Secondly analysing the data of energy consumption is useful for in energy sales companies to predict probable future consumption and applying the costs for electrical units on the opt market of electrical energy. Thirdly it is useful for power grid companies to regulate and determine the optimal loading of transformer substations. And at last, analysing the consumption of electrical consumption can be applied in governmental sector for calculating the optimal tariff schemes for different groups of consumers.

**10.FUTURE SCOPE**

Linear regression model can be replaced by many other models which are currently existing and giving a try with larger data set might change the model’s accuracy more. Even training the model by using the many inner techniques like Feature scaling etc. If new column transforms technique come up and suits the data, then those can definitely be applied. Once good number of people use it the basing and outline of the project can be changed considering the latest coding platforms available. The app’s performance should be tested across various CPU’s considering all the CPU’s don’t give similar output. The app can also be directly linked to the electrical meter’s of every house which is unsupervised learning, hence developing the model more.

**11.BIBILOGRAPHY**

1. [machinelearningmastery.com/how-to-load-and-explore-household-electricity-usage-data/](https://machinelearningmastery.com/how-to-load-and-explore-household-electricity-usage-data/)

2. <https://creately.com/lp/block-diagram-maker/>

3. <https://www.britannica.com/technology/artificial-intelligence>

4. <https://www.expert.ai/blog/machine-learning-definition/>

5. <https://www.geeksforgeeks.org/ml-linear-regression/>

6. <https://www.edureka.co/blog/linear-regression-for-machine-learning/>

7. <https://medium.com/analytics-vidhya/prediction-of-hourly-household-power-consumption-c680a9eb9190>

8.https://www.researchgate.net/publication/270528692\_Data\_Set\_Analysis\_of\_Electric\_Power\_Consumption

**APPENDIX**

Flask Code

from flask import Flask,request,render\_template

import numpy as np

import pandas as pd

import pickle

import os

app = Flask(\_\_name\_\_)

model = pickle.load(open('PCA\_modelm.pkl', 'rb'))

@app.route('/')

def home():

return render\_template("pca.html")

@app.route('/predict',methods=["POST","GET"])

def predict():

input\_features = [float(x) for x in request.form.values()]

features\_value = [np.array(input\_features)]

features\_name = ['Global\_reactive\_power', 'Global\_intensity','Sub\_metering\_1',

'Sub\_metering\_2', 'Sub\_metering\_3']

df = pd.DataFrame(features\_value, columns=features\_name)

output = model.predict(df)

return render\_template('result1.html', prediction\_text=output)

if \_\_name\_\_=="\_\_main\_\_":

#port = int(os.getenv('PORT', 8080))

#app.run(host='0.0.0.0', port=port, debug=False)

app.run(debug=True)

IBM Watson’s spyder code

import requests

import json

API\_KEY = "RvLBg8GMl8gitnEGyrAFc3kVTLr6ap-b1X-7WEMX42Dg"

token\_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API\_KEY, "grant\_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token\_response.json()["access\_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

payload\_scoring ={"input\_data":[{"field":[["Global\_reactive\_power","Global\_intensity","Sub\_metering\_1","Sub\_metering\_2","Sub\_metering\_3"]],

"values": [[0.418,18.4,0.0,1.0,17.0]] }]}

response\_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/929df1a6-62a6-4469-82e8-979005ca4cb7/predictions?version=2021-06-02', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

predictions= response\_scoring.json()

pred= predictions["predictions"][0]['values'][0][0]

print("The Global Active Power is " + str(pred))

Jupyter Notebook

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

dataset=pd.read\_csv(r"Household\_power\_consumption.csv")

dataset.head()

dataset.tail()

print(f"The Dataset has {dataset.shape[0]} rows and {dataset.shape[1]} columns ")

dataset.columns

dataset.isnull().sum()

dataset.describe()

sns.displot(dataset["Global\_active\_power"])

sns.displot(dataset["Global\_reactive\_power"])

sns.displot(dataset["Voltage"])

sns.displot(dataset["Global\_intensity"])

sns.jointplot(x="Global\_reactive\_power", y="Global\_active\_power", data= dataset, kind='scatter')

sns.lineplot(x='Voltage',y='Global\_intensity',data=dataset)

sns.barplot(x='Global\_active\_power',y='Global\_reactive\_power',data=dataset)

sns.pairplot(dataset)

sns.heatmap(dataset.corr(),annot=True)

x = dataset.iloc[:,[2,4,5,6,7]].values

y = dataset.iloc[:,1].values

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y , test\_size = 0.3 , random\_state = 0)

print(x\_train.shape)

print(y\_train.shape)

print(x\_test.shape)

print(y\_test.shape)

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

lr.fit(x\_train,y\_train)

ypred = lr.predict(x\_test)

from sklearn.metrics import r2\_score

accuracy = r2\_score(ypred,y\_test)

import pickle

prj='PCA\_modelm.pkl'

pickle.dump(lr,open(prj,'wb'))