INTRODUCTION

a)Overview:

India is the world's third largest producer and third largest consumer of electricity. The gross electricity consumption in 2018-19 was 1,181 kWh per capita. Energy use can be viewed as a function of total GDP, structure of the economy and technology. The increase in household energy consumption is more significant than that in the industrial sector. To achieve reduction in electricity consumption, it is vital to have current information about household electricity use. This Project mainly focuses on applying a machine-learning algorithm to calculate the power consumed by all appliances. This will help you track the power consumed on regular intervals for all kinds of appliances which use heavy loads such as Air Conditioners, Oven or a washing machine etc.

b)Purpose:

The main purpose of this project is to help the user track the power consumed by various appliances on regular intervals in order to achieve reduction in electricity consumption.

LITERATURE SURVERY

a)Existing problem :

The increase in household electricity consumption on a regular basis all over India is a matter of concern as it causes many

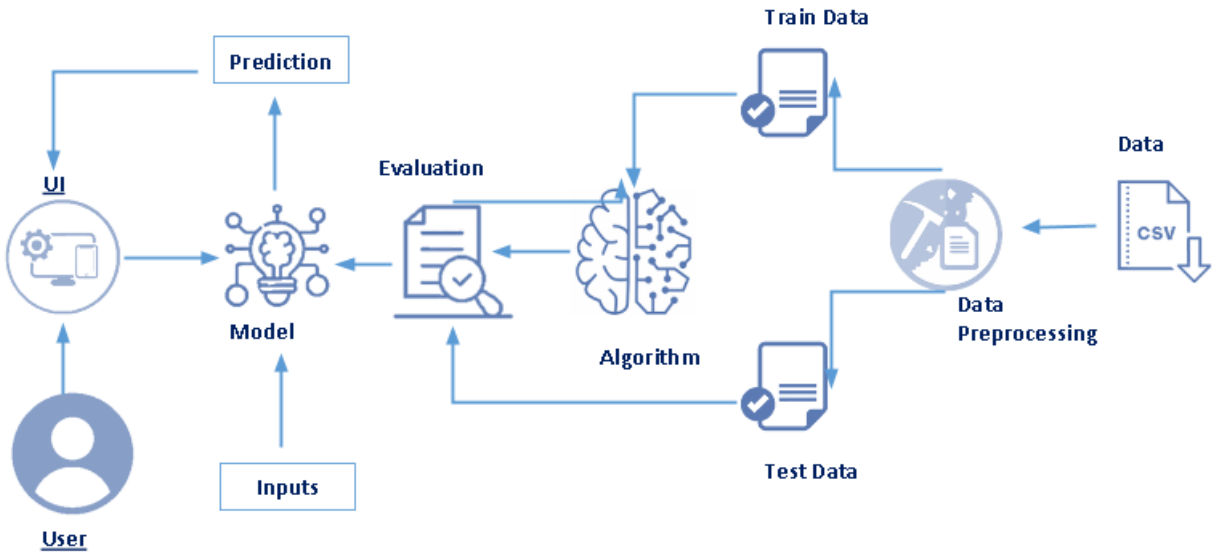
environmental problems and may also risk lowering the expected lifespan of appliances.

b)Proposed solution :

The proposed solution is to achieve reduction in the consumption of electricity.This can be achieved by having current information about household electricity household electrcity use. This project aims at calculating the power consumed by all the appliances that helps in reducing the electricity useage.

THEORITICAL ANALYSIS

a)Block diagaram:



b) Hardware/Software designing:

Hardware:

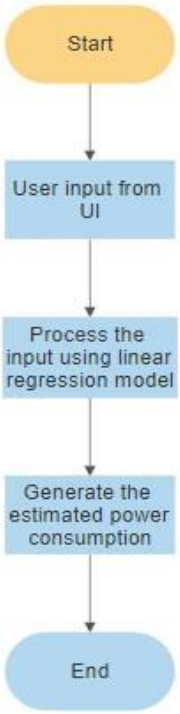
Processor- Intel core i5
Hard Disk capacity- 1TB
RAM capacity-8GB

Software:
Operating system-Windows 10
Programming language-python
Development Tool-Jupyter Notebook,Spyder

EXPERIMENTAL INVESTIGATIONS

1. The dataset considered in the experiment has 207529 rows and 7 columns
2. Mean absolute error for the model is found to be 0.027455608456450387
3. Mean squared error for the model is found to be 0.0018226463072696978
4. R squares value for the model is found to be 0.9983631695586828 which is considered to be high and falls in the accepted range.

FLOW CHART



RESULT

The result of the model is global active power which predicts the total electricity consumption of the household by considering user inputs regarding global reactive power, global intensity, submetering.

ADVANTAGES AND DISADVANTAGES

1. **Advantages:**

- The model helps in reducing the electricity consumption.
- Helps in identifying energy cost.
- Helps in uncovering energy wastage.

2. **Disadvantages:**

- The model may not be used in all the households.
- The model might not give accurate results in some cases.

APPLICATIONS

- To estimate the power consumption in households.
- To reduce the power consumption accordingly.
- To reduce power wastage.

CONCLUSION

The increase in household energy consumption is more significant than that in the industrial sector. Therefore, considering the scarcity optimal usage and efficient management is necessary. Hence this project takes a step towards estimating the

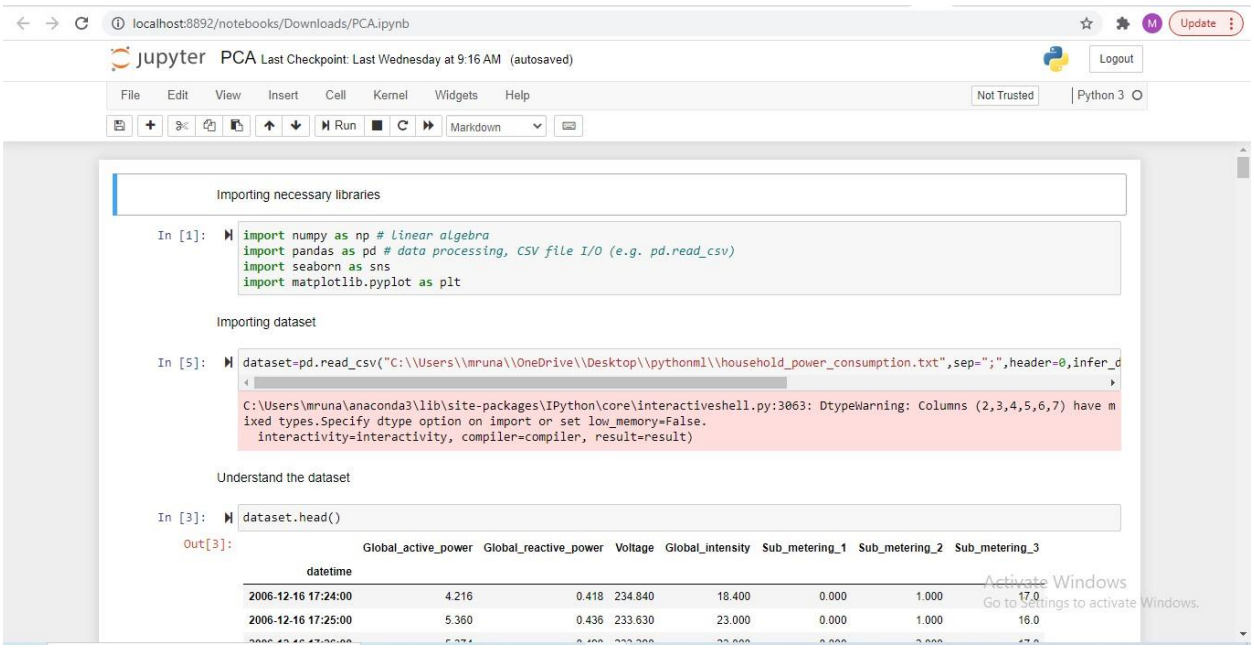
consumption of electricity by considering few user inputs which helps in optimum usage.

FUTURE SCOPE

- The model can be improved by improving the dataset being used.
- The accuracy of the model can be increased by using different algorithms.

APPENDIX

A)Source Code:



jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Markdown

In [3]:
```
dataset.head()
```

Out[3]:

datetime	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-16 17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0
2006-12-16 17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2006-12-16 17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0
2006-12-16 17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0
2006-12-16 17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0

In [4]:
```
dataset.tail()
```

Out[4]:

datetime	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2010-11-26 20:58:00	0.946	0	240.43	4	0	0	0.0
2010-11-26 20:59:00	0.944	0	240	4	0	0	0.0
2010-11-26 21:00:00	0.938	0	239.82	3.8	0	0	0.0
2010-11-26 21:01:00	0.934	0	239.7	3.8	0	0	0.0
2010-11-26 21:02:00	0.932	0	239.55	3.8	0	0	0.0

In [5]:
```
print(f"The Dataset has {dataset.shape[0]} rows and {dataset.shape[1]} columns")
```

```
The Dataset has 2075259 rows and 7 columns
```


jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Markdown

In [6]:
```
dataset.columns
```

Out[6]:
```
Index(['Global_active_power', 'Global_reactive_power', 'Voltage',
       'Global_intensity', 'Sub_metering_1', 'Sub_metering_2',
       'Sub_metering_3'],
      dtype='object')
```

Complete information of dataset

In [7]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2075259 entries, 2006-12-16 17:24:00 to 2010-11-26 21:02:00
Data columns (total 7 columns):
 #   Column                 Dtype
---  ------                 -----
 0   Global_active_power    object
 1   Global_reactive_power  object
 2   Voltage                object
 3   Global_intensity       object
 4   Sub_metering_1         object
 5   Sub_metering_2         object
 6   Sub_metering_3         float64
dtypes: float64(1), object(6)
memory usage: 126.7+ MB
```

Checking total null values in each column

In [8]:
```
dataset.isnull().sum()
```

jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Markdown

In [8]:
```
dataset.isnull().sum()
```

Out[8]:
```
Global_active_power         0
Global_reactive_power       0
Voltage                     0
Global_intensity            0
Sub_metering_1              0
Sub_metering_2              0
Sub_metering_3          25979
dtype: int64
```

Understanding percent of data missing

In [9]:
```
percent_missing = dataset.isnull().sum() * 100 / len(dataset)
missing_value_df = pd.DataFrame({'percent_missing': percent_missing})
```

In [10]:
```
missing_value_df
```

Out[10]:

	percent_missing
Global_active_power	0.000000
Global_reactive_power	0.000000
Voltage	0.000000
Global_intensity	0.000000
Sub_metering_1	0.000000
Sub_metering_2	0.000000
Sub_metering_3	1.251844

jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Markdown

Handling missing values

In [13]:
```
dataset.loc[dataset.Sub_metering_3.isnull()].head()
```

Out[13]:

datetime	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-21 11:23:00	?	?	?	?	?	?	NaN
2006-12-21 11:24:00	?	?	?	?	?	?	NaN
2006-12-30 10:08:00	?	?	?	?	?	?	NaN
2006-12-30 10:09:00	?	?	?	?	?	?	NaN
2007-01-14 18:36:00	?	?	?	?	?	?	NaN

In [14]:
```
dataset.replace('?', np.nan, inplace=True)
```

In [15]:
```
dataset.loc[dataset.Sub_metering_3.isnull()].head()
```

Out[15]:

datetime	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-21 11:23:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2006-12-21 11:24:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2006-12-30 10:08:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2006-12-30 10:09:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2007-01-14 18:36:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN

jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Markdown

In [16]:
```
dataset = dataset.dropna(how = 'all')
```

In [17]:
```
for i in dataset.columns:
    dataset[i] = dataset[i].astype('float64')
#dataset = dataset.astype('float32')
```

In [18]:
```
dataset.shape
```

Out[18]: (2049280, 7)

Adding another sub_metering_4 column

In [19]:
```
values = dataset.values
dataset['sub_metering_4'] = (values[:,0] * 1000 / 60) - (values[:,4] + values[:,5] + values[:,6])
```

In [20]:
```
dataset.dtypes
```

Out[20]:
```
Global_active_power      float64
Global_reactive_power    float64
Voltage                  float64
Global_intensity         float64
Sub_metering_1           float64
Sub_metering_2           float64
Sub_metering_3           float64
sub_metering_4           float64
dtype: object
```


In [21]:
```
dataset.describe()
```

localhost:8892/notebooks/Downloads/PCA.ipynb

Update

jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

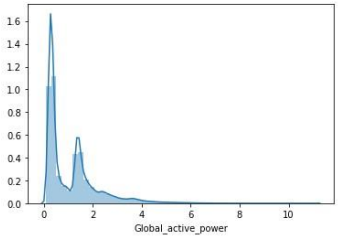
Not Trusted | Python 3

Run Markdown

Data Visualization

In [22]:
```
sns.distplot(dataset['Global_active_power'])
```

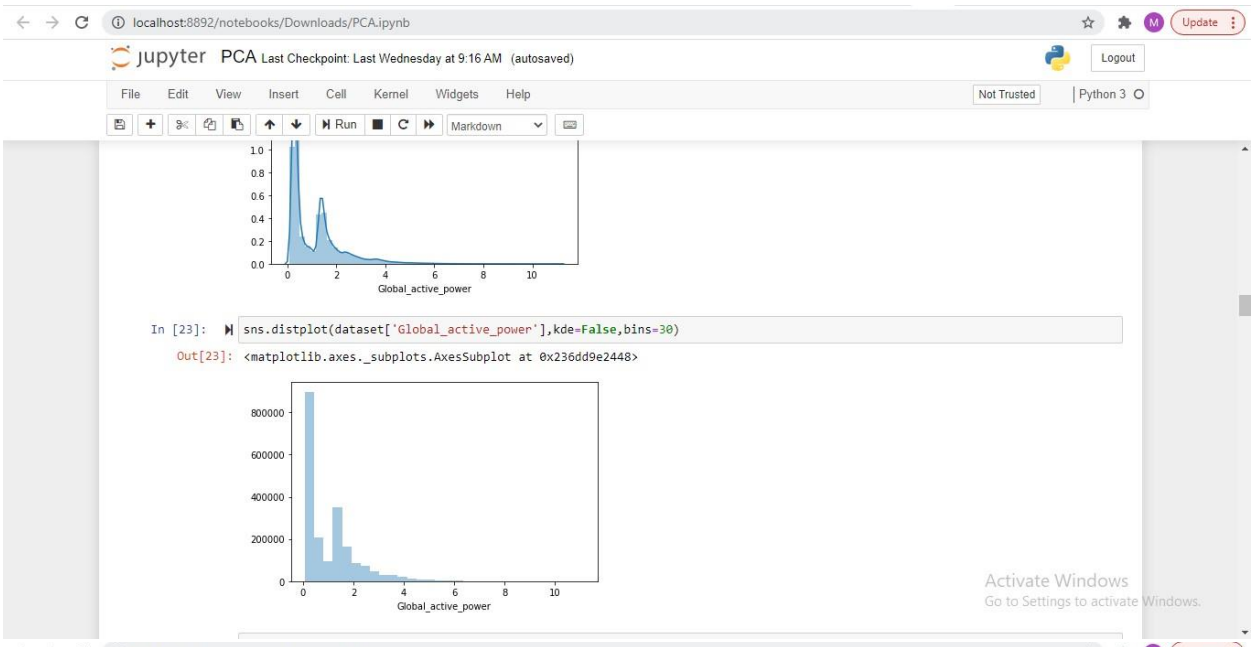
Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x236ddd3cd88>

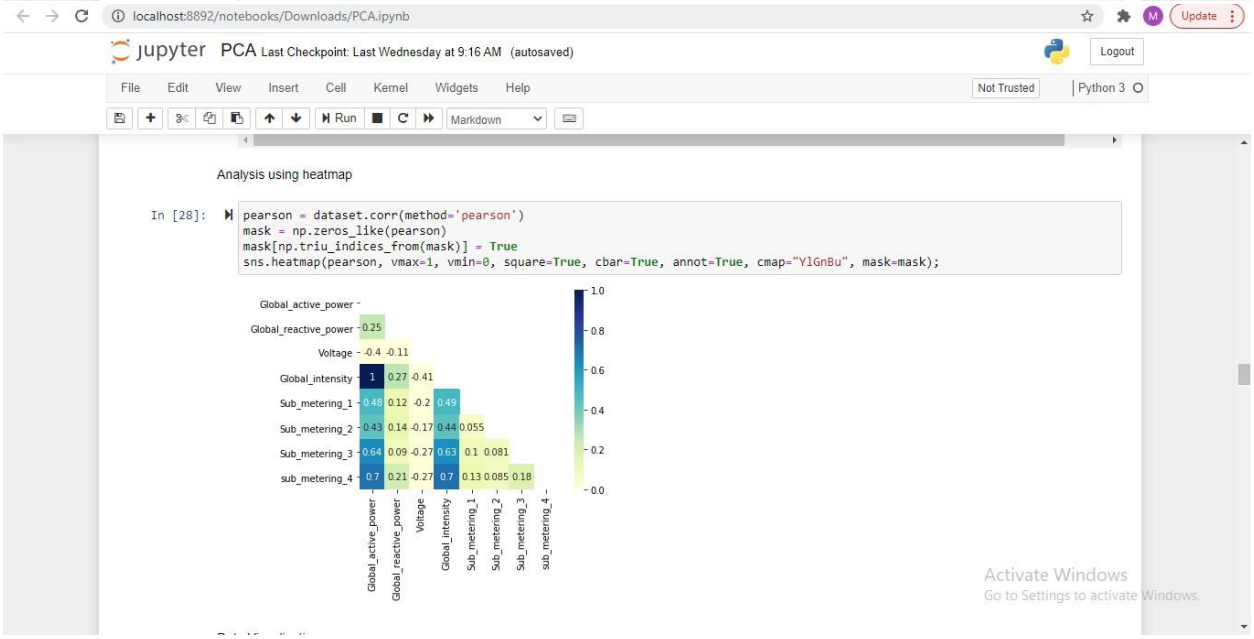


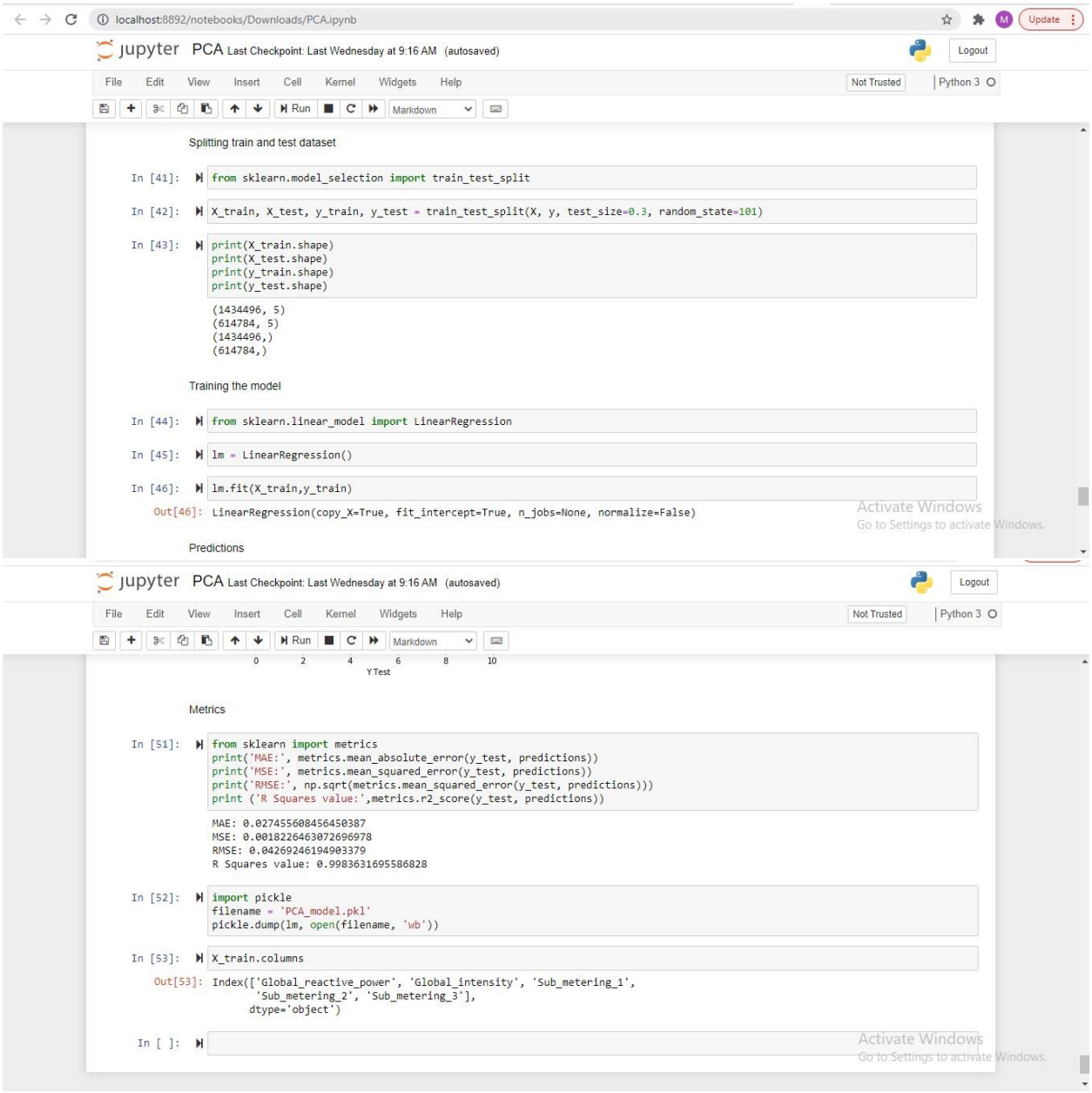
In [23]:
```
sns.distplot(dataset['Global_active_power'],kde=False,bins=30)
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x236dd9e2448>

Activate Windows
Go to Settings to activate Windows.


localhost:8892/notebooks/Downloads/PCA.ipynb
jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Global_active_power
In [23]:
sns.distplot(dataset['Global_active_power'],kde=False,bins=30)
Out[23]:
<matplotlib.axes._subplots.AxesSubplot at 0x236dd9e2448>
Global_active_power
Activate Windows
Go to Settings to activate Windows.


localhost:8892/notebooks/Downloads/PCA.ipynb
jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Analysis using heatmap
In [28]:
pearson = dataset.corr(method='pearson')
mask = np.zeros_like(pearson)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(pearson, vmax=1, vmin=0, square=True, cbar=True, annot=True, cmap="YlGnBu", mask=mask);
Global_active_power
Global_reactive_power
Voltage
Global_intensity
Sub_metering_1
Sub_metering_2
Sub_metering_3
sub_metering_4
Activate Windows
Go to Settings to activate Windows.


localhost:8892/notebooks/Downloads/PCA.ipynb
jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Splitting train and test dataset
In [41]: from sklearn.model_selection import train_test_split
In [42]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
In [43]: print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
(1434496, 5)
(614784, 5)
(1434496,)
(614784,)
Training the model
In [44]: from sklearn.linear_model import LinearRegression
In [45]: lm = LinearRegression()
In [46]: lm.fit(X_train,y_train)
Out[46]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
Predictions
Activate Windows
Go to Settings to activate Windows.

jupyter PCA Last Checkpoint: Last Wednesday at 9:16 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
0 2 4 6 8 10
Y Test
Metrics
In [51]: from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
print ('R Squares value:',metrics.r2_score(y_test, predictions))
MAE: 0.027455608456450387
MSE: 0.0018226463072696978
RMSE: 0.04269246194903379
R Squares value: 0.9983631695586828
In [52]: import pickle
filename = 'PCA_model.pkl'
pickle.dump(lm, open(filename, 'wb'))
In [53]: X_train.columns
Out[53]: Index(['Global_reactive_power', 'Global_intensity', 'Sub_metering_1',
'Sub_metering_2', 'Sub_metering_3'],
dtype='object')
In [ ]:
Activate Windows
Go to Settings to activate Windows.

B) UI OUTPUT:


localhost:5000
Update
Power Consumption Analysis
0.418
18
0
1
17
Predict
220V 10(60)A 50Hz 3900imp/kWh
kWh


localhost:5000/predict
Update
Power Consumption Analysis
Global active power is [4.22801143]
220V 10(60)A 50Hz 3900imp/kWh
kWh