Course: DATS 6401 - Visualisation of Complex Data

Instructor **: Dr Reza Jafari**

Final Term Project

Topic : Analyze Energy Imbalance of Prosumers in Estonia

Abstract

The goal of the project is to analyze the energy imbalance of Estonian Prosumers. Prosumers are organizations, businesses and individuals who consume and produce energy. The concept represents a shift from the traditional model where consumers simply purchase energy from utilities and rely on centralized power generation sources.

The imbalance problem indicates that there exists an imbalance between production and consumption of energy. That is the generated electricity mismatches the demanding side. We'll be analyzing the energy consumption and production patterns of prosumers given the auxiliary features like electricity, gas prices, installed solar panel capacity. This can be formulated as a Time Series Regression Problem

The Project work also focuses on developing an interactive Dash app to analyze the data statistically and visually. The dash app 5 tabs namely - Data Visualization, Normality Tests, PCA analysis, Consumption analysis Dashboard and Know your data

Introduction

Dash apps give a point-&-click interface to models written in Python, vastly expanding the notion of what's possible in a traditional "dashboard." The dynamic dashboard created using dash helps the user to visualize different features like installed solar panel capacity, electricity consumption, EIC count and all the input features used in the analysis of energy imbalance among prosumers.

The app layout has been designed in such a way the end user can navigate easily to the desired part of analysis. The dataset is preprocessed by removing null values. Since the dataset is time series some of the data imputation was done using backfill and forward fill methods and some nan values were imputed with 0’s . One of the important tasks in analyzing a dataset is to detect and remove the outliers. The outliers can be identified using different methods and this project uses box plots for outlier identification and IQR method for outlier removal. The cleaned dataset is then tested for normality and PCA is done to reduce the dimensions of the original feature space. PCA analysis helps in finding the best number of feature components for modeling as well as finding the correlation between the variables. The heatmap shows the correlation matrix of the dataset which helps in finding the collinearity between the variables.
The plots like histograms, line, box, reg, pie, Kde and scatter plot with regression line are developed to study the dataset in a wider view. The final dashboard has subplots of dependent
variable to analyze the impact of other variables in its prediction.
A detailed description about these techniques is discussed in the next chapter.

Methods and Theory

Dash:
Dash is an open-source framework for building analytical applications, with no Javascript
required, and it is tightly integrated with the Plotly graphing library. Dash is a python framework
mostly used for building data visualization apps.

Dash Tabs:
The dcc.Tabs and dcc.Tab components can be used to create tabbed sections in your app. The dcc.Tab component controls the style and value of the individual tab and the dcc.Tabs Component holds a collection of dcc.Tab components.

Dash Callbacks:
functions that are automatically called by Dash whenever an input component's property
changes, to update some property in another component (the output).

Dash Core Components:
The Dash Core Components module (dash.dcc) can be imported and used with dash import dcc and gives you access to many interactive components,including dropdowns, checklists, and sliders.

Dash HTML Components:
Dash is a web application framework that provides pure Python abstraction around HTML, CSS,and JavaScript. Instead of writing HTML or using an HTML

templating engine, you compose your layout using Python with the Dash HTML Components module (dash.html).

Line Plot:
A line plot is a graph that displays data using a number line.
Histogram:
A histogram is a graphical representation that organizes a group of data points into user-specified ranges. Similar in appearance to a bar graph, the histogram condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges or bins.

Pie Chart:
Pie charts can be used to show percentages of a whole and represent percentages at a set point in time. Unlike bar graphs and line graphs, pie charts do not show changes over time.

Dropdown:
To create a basic dropdown, provide options and a value to dcc.Dropdown in that order.

Graph:
The dcc.Graph component can be used to render any plotly-powered data visualization, passed as the figure argument.

Important components of Analysis :

Correlation Matrix:
A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable (Xi) in the table is correlated with each of the other values in the table (Xj). This allows you to see which pairs have the highest correlation.

Principal Component Analysis:

Reducing the number of input variables for a predictive model is referred to as dimensionality reduction. Fewer input variables can result in a simpler predictive model that may have better performance when making predictions on new data. Perhaps the most popular technique for dimensionality reduction in machine learning is Principal Component Analysis. This is a technique that comes from the field of linear algebra and can be used as a data preparation technique to create a projection of a dataset prior to fitting a model.

Outlier Detection and Removal:

Outliers are data points that are far from other data points. With outlier detection and treatment, anomalous observations are viewed as part of different populations to ensure stable findings for the population of interest.

Outlier Detection- Interquartile Range (IQR) - A commonly used rule says that a data point is an outlier if it is more than 1.5*IQR above the third quartile or below the first quartile. IQR is calculated as : Q3-Q1. Low outliers are below Q1 − 1.5*IQR and High outliers are above Q3 + 1.5 * IQR where Q1 is the first quartile and Q3 is the third quartile.

Kolmogorov-Smirnov (K-S) Test:

The Kolmogorov-Smirnov (K-S) Test compares your data with a known distribution and lets you know if they have the same distribution. The K-S test is a non-parametric test. It is commonly used as a test for normality to see if your data is normally distributed.

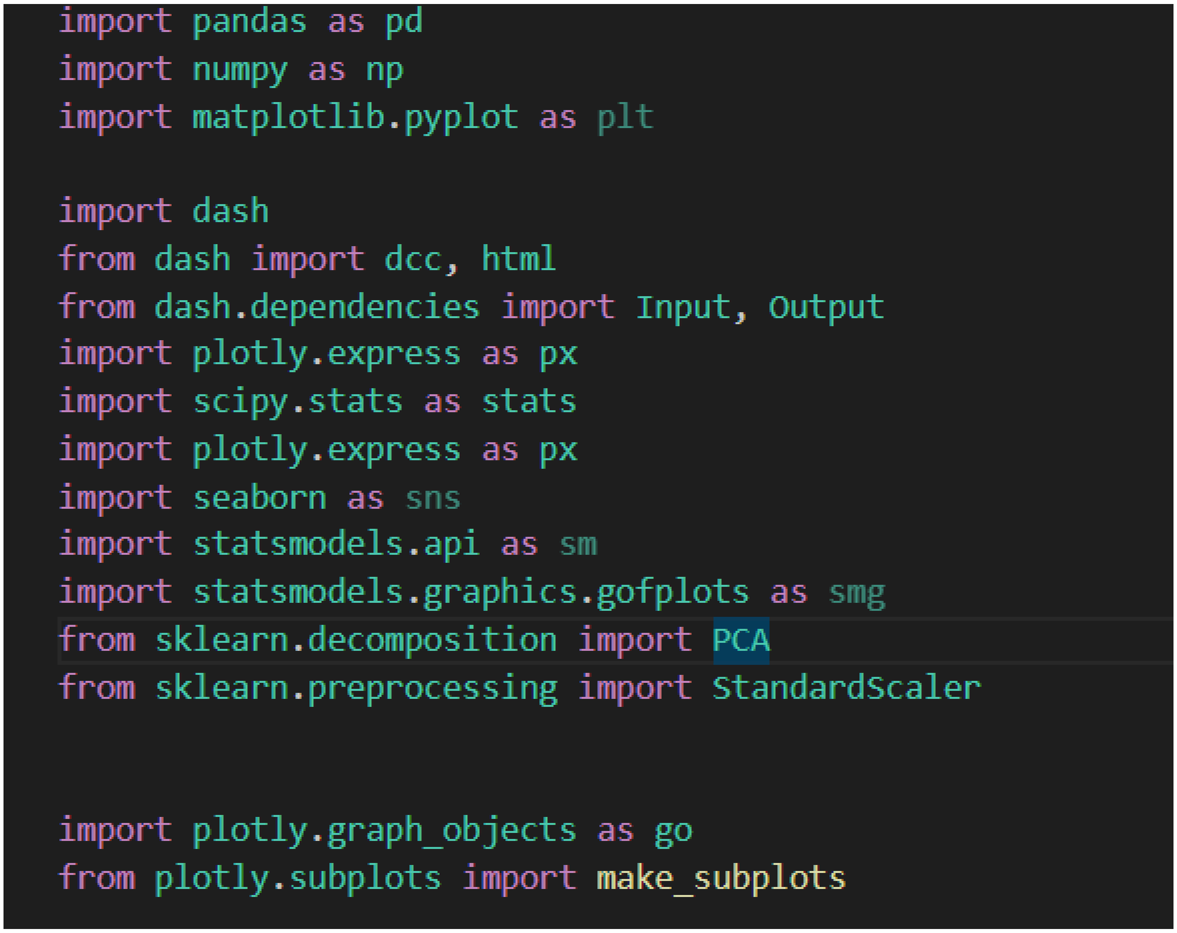
H0: The data is Normally distributed. p-value > alpha
H1: The data is not Normally distributed. p-value < alpha

Shapiro-Wilk Test:

The Shapiro-Wilk test is a way to tell if a random sample comes from normal distribution. In
practice, the Shapiro-Wilk test is believed to be a reliable test of normality, although there is
some suggestion that the test may be suitable for smaller samples of data.
H0: The data is Normally distributed. p-value > alpha
H1: The data is not Normally distributed. p-value < alpha

D’Agostino’s K2 test:
D’Agostino’s K2 test is a goodness-of-fit measure of departure from normality, that is the test aims to establish whether or not the given sample comes from a normally distributed population.
H0: The data is Normally distributed. p-value > alpha
H1: The data is not Normally distributed. p-value < alpha

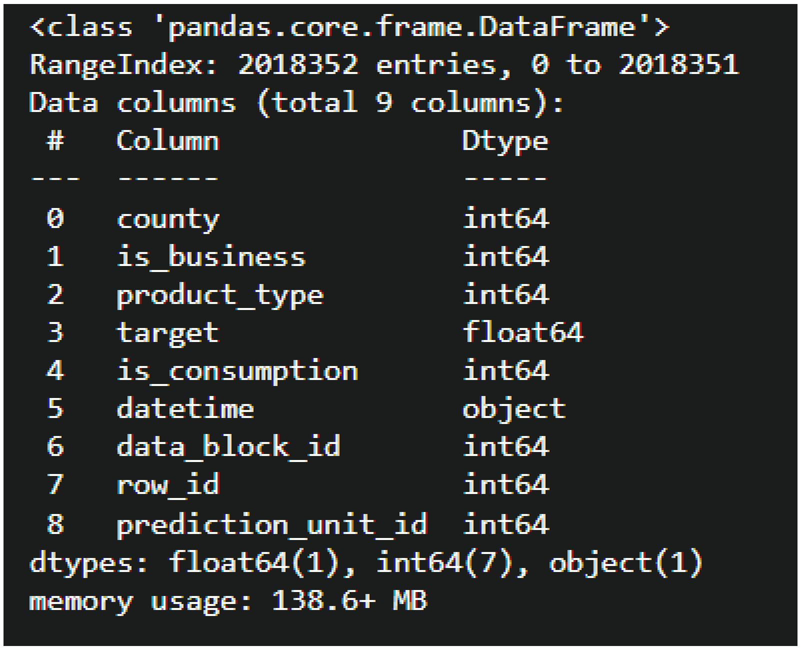
Experimental Setup : Below libraries are required for doing analysis



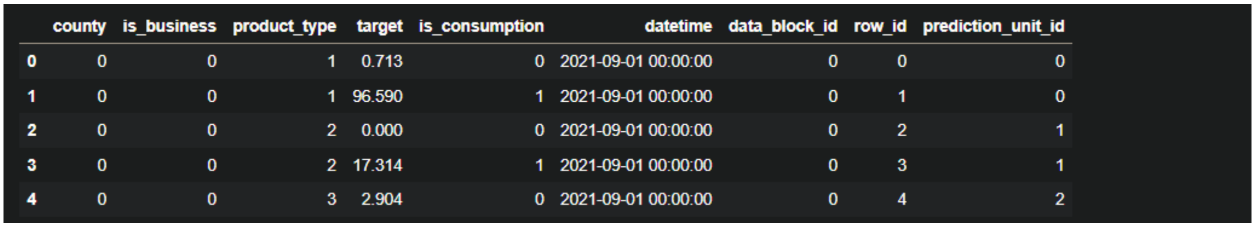
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import plotly.express as px
import scipy.stats as stats
import plotly.express as px
import seaborn as sns
import statsmodels.api as sm
import statsmodels.graphics.gofplots as smg
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler


import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

DATASET DESCRIPTION

1.The project deals with 8 datasets namely - Training dataset(containing target variable i.e. electricity consumption or production)



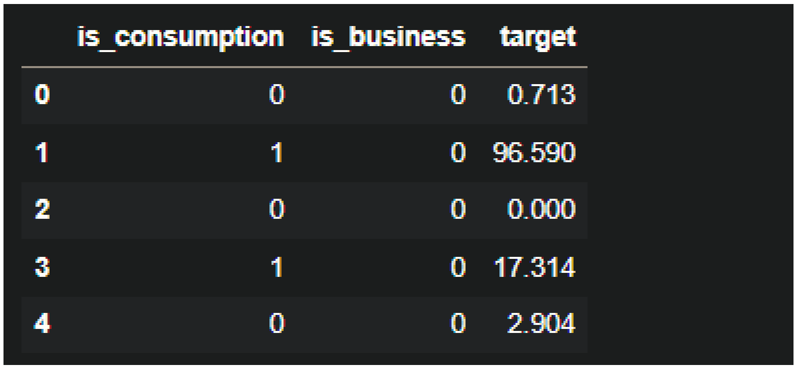
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2018352 entries, 0 to 2018351
Data columns (total 9 columns):
 #   Column              Dtype
---  ------              -----
 0   county              int64
 1   is_business         int64
 2   product_type        int64
 3   target              float64
 4   is_consumption      int64
 5   datetime            object
 6   data_block_id       int64
 7   row_id              int64
 8   prediction_unit_id  int64
dtypes: float64(1), int64(7), object(1)
memory usage: 138.6+ MB
```



	county	is_business	product_type	target	is_consumption	datetime	data_block_id	row_id	prediction_unit_id
0	0	0	1	0.713	0	2021-09-01 00:00:00	0	0	0
1	0	0	1	96.590	1	2021-09-01 00:00:00	0	1	0
2	0	0	2	0.000	0	2021-09-01 00:00:00	0	2	1
3	0	0	2	17.314	1	2021-09-01 00:00:00	0	3	1
4	0	0	3	2.904	0	2021-09-01 00:00:00	0	4	2

The shape of dataset is (2018352, 9)

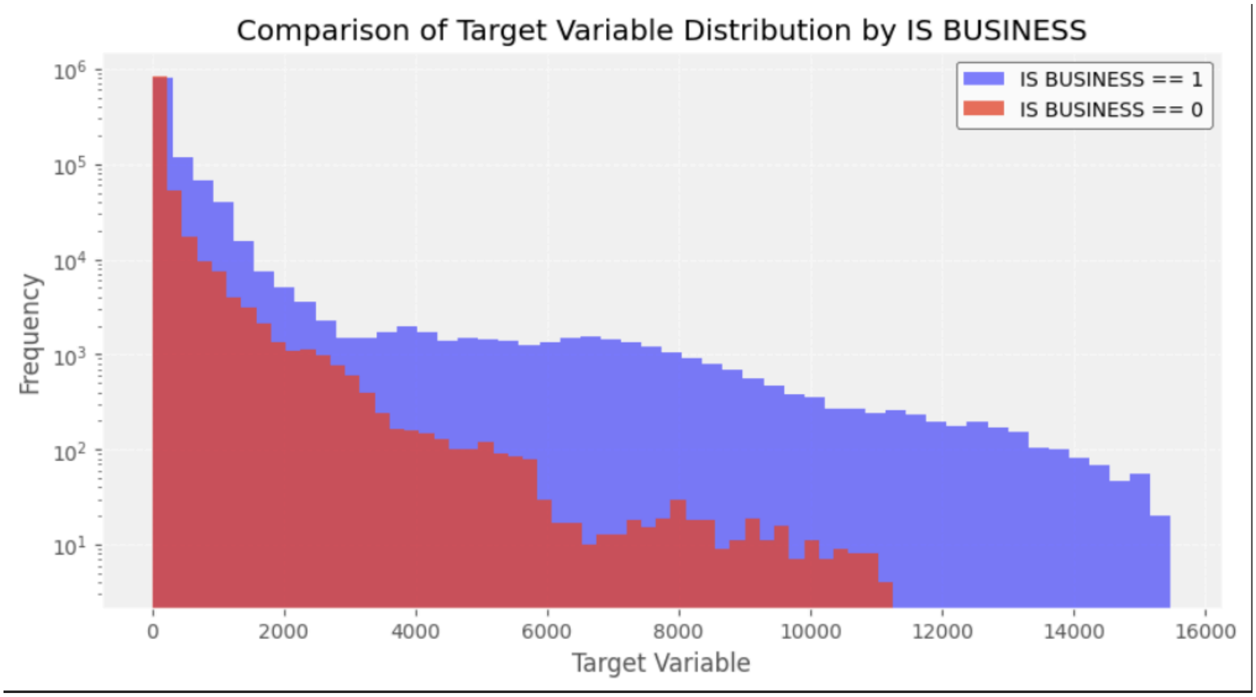
Important boolean flags on which the the target is dependent are - is_consumption and is_business flags



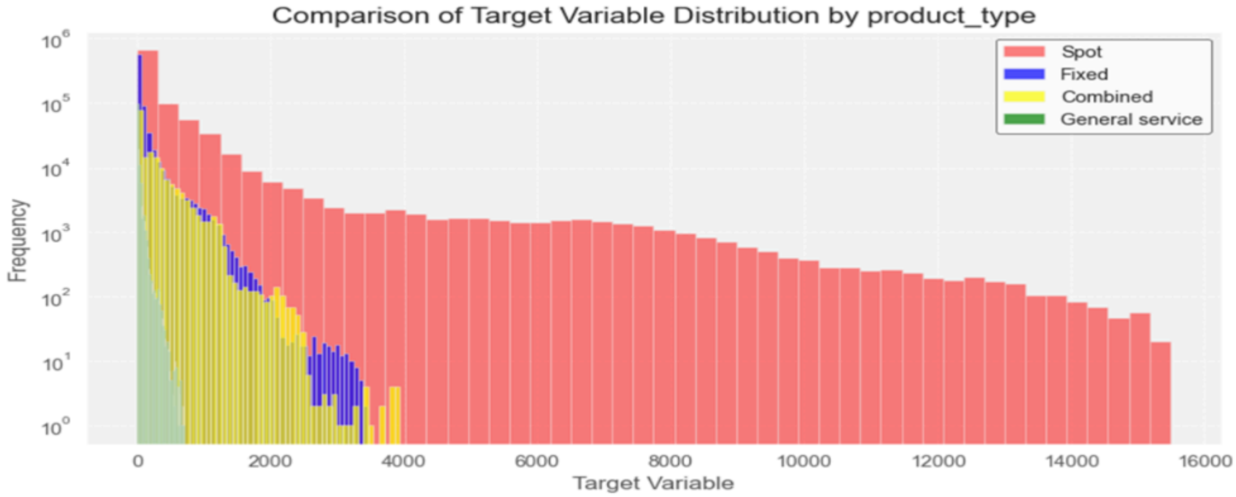
	is_consumption	is_business	target
0	0	0	0.713
1	1	0	96.590
2	0	0	0.000
3	1	0	17.314
4	0	0	2.904

So, if the ‘if_consumption’ flag is 0 - the intersection corresponding to this row is a producer and if the flag is 1 - the intersection corresponds to consumption and the target is electricity amount

Analysis of Target variable



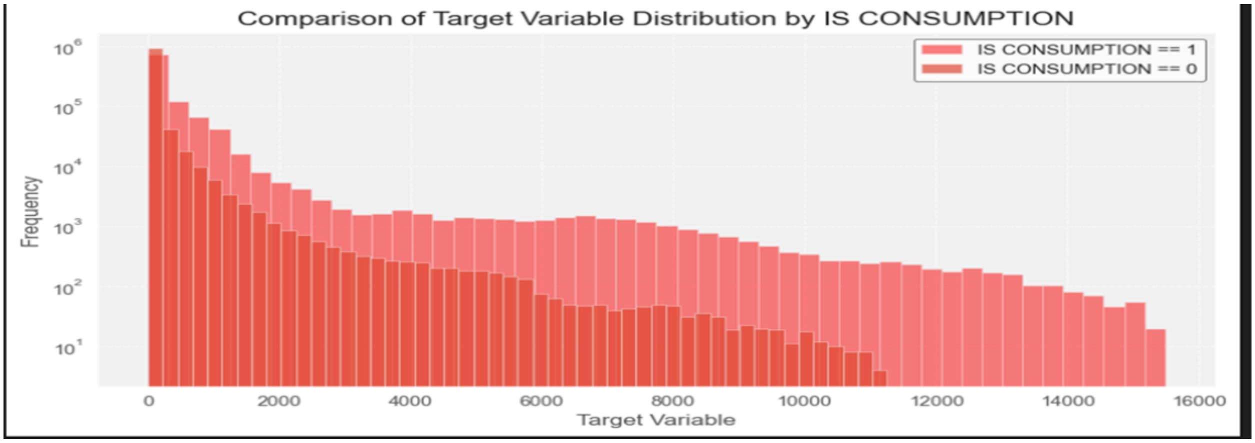
Inference - The graph shows the distribution of target variable based upon the business boolean flag, it is quite evident that the organizations or people who are into business have a wider spread and have higher variability



This is electricity amount analysis based upon different product types. The type here corresponds to mapping of different contract types

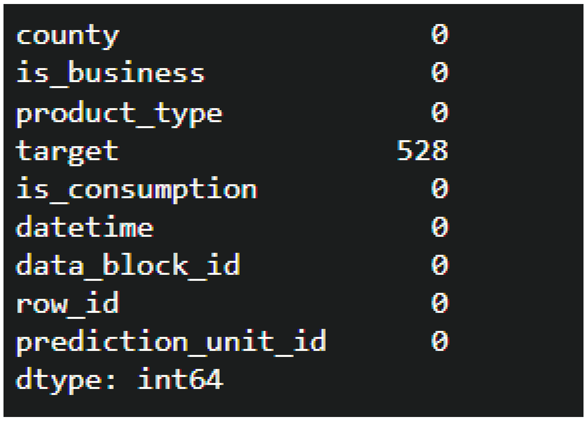
{0 : ‘Combined’, 1 : ‘Fixed’, 2 : ‘General Service’ 3 : ‘Spot’’}

Inference - From the above graph it is deduced that the product Spot has the widest spread and the maximum variability



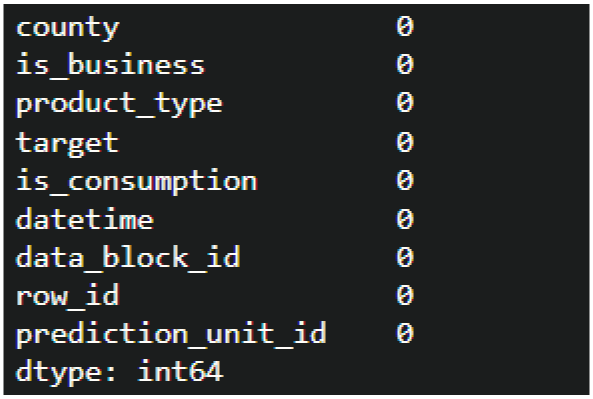
The above is the depiction of variation of target variable i.e. electricity amount versus the consumption flag. It is deduced that the organizations who are consumers have a wider spread as compared to organizations who are non consumers

Treatment of Null values in training data -



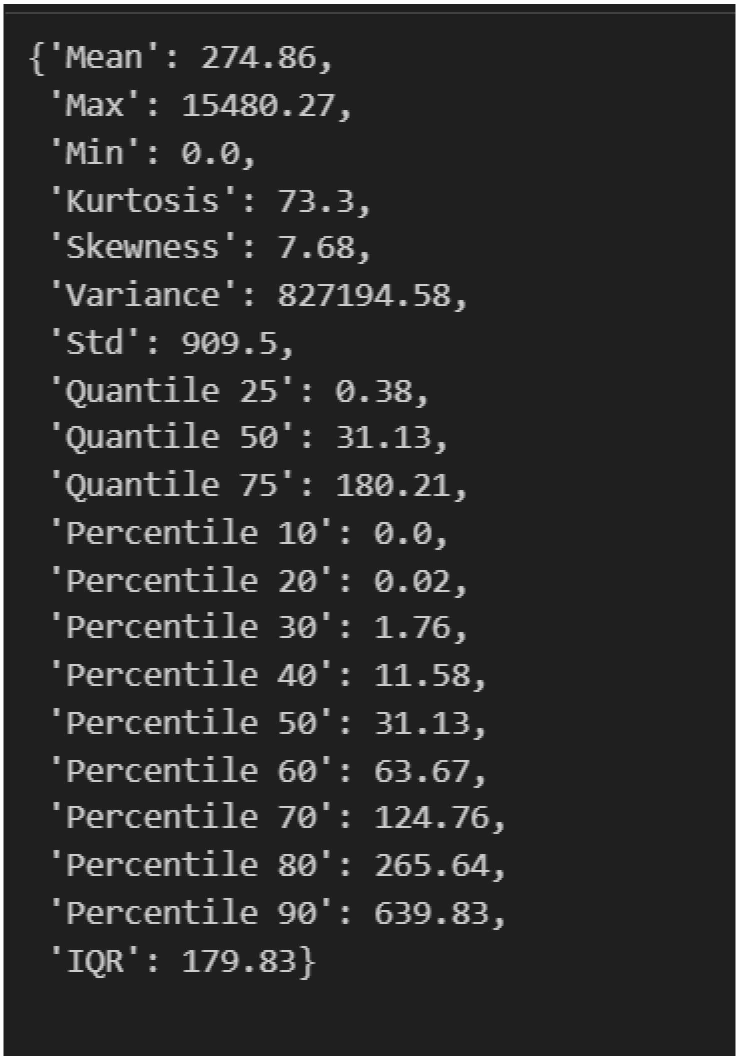
```
county                  0
is_business             0
product_type            0
target                528
is_consumption          0
datetime                0
data_block_id           0
row_id                  0
prediction_unit_id      0
dtype: int64
```

Inference - From the above snippet it is clear that only target column has the nan values so to deal with this i removed the nan values because the count i.e. 528 <<< total of samples ~ 2 million so i just dropped the nan values



```
county                0
is_business           0
product_type          0
target                0
is_consumption        0
datetime              0
data_block_id         0
row_id                0
prediction_unit_id    0
dtype: int64
```

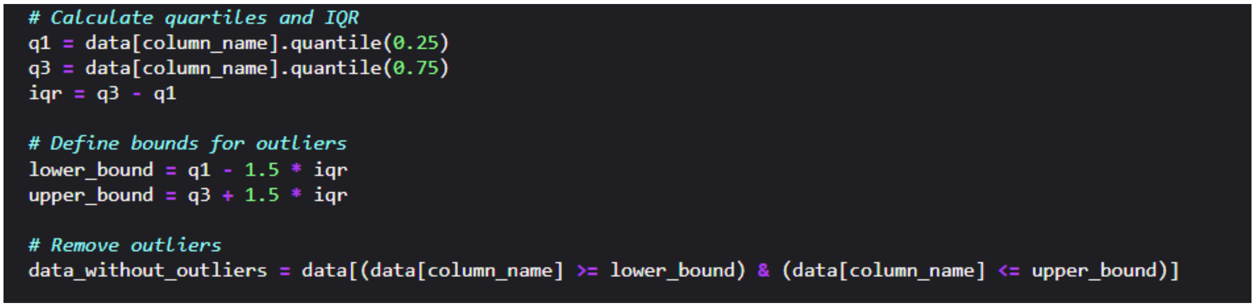
Statistics of Target variable



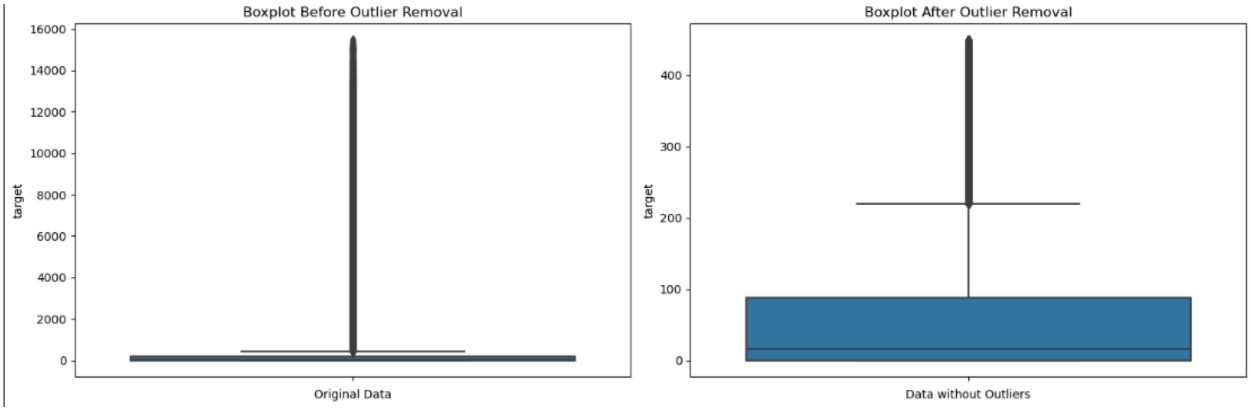
```
{'Mean': 274.86,
 'Max': 15480.27,
 'Min': 0.0,
 'Kurtosis': 73.3,
 'Skewness': 7.68,
 'Variance': 827194.58,
 'Std': 909.5,
 'Quantile 25': 0.38,
 'Quantile 50': 31.13,
 'Quantile 75': 180.21,
 'Percentile 10': 0.0,
 'Percentile 20': 0.02,
 'Percentile 30': 1.76,
 'Percentile 40': 11.58,
 'Percentile 50': 31.13,
 'Percentile 60': 63.67,
 'Percentile 70': 124.76,
 'Percentile 80': 265.64,
 'Percentile 90': 639.83,
 'IQR': 179.83}
```

Inference - Mean electricity consumption/production(based on the flag) is ~275. The variable has a high kurtosis value indicating that it has extreme values as compared to a normal distribution.

Outlier Treatment of Target - I used 1.5 IQR that is InterQuantile method to remove outliers.



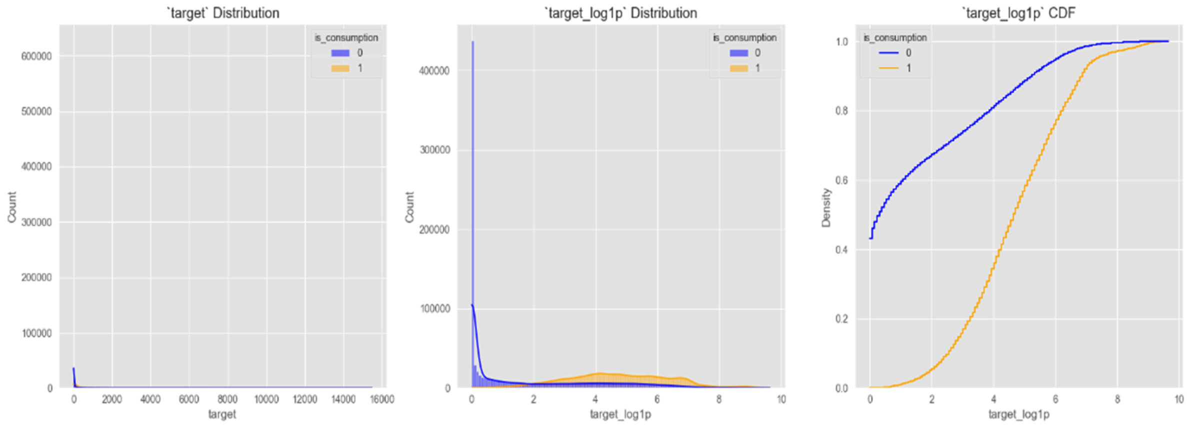
```
# Calculate quartiles and IQR
q1 = data[column_name].quantile(0.25)
q3 = data[column_name].quantile(0.75)
iqr = q3 - q1

# Define bounds for outliers
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

# Remove outliers
data_without_outliers = data[(data[column_name] >= lower_bound) & (data[column_name] <= upper_bound)]
```



Boxplot Before Outlier Removal

Boxplot After Outlier Removal

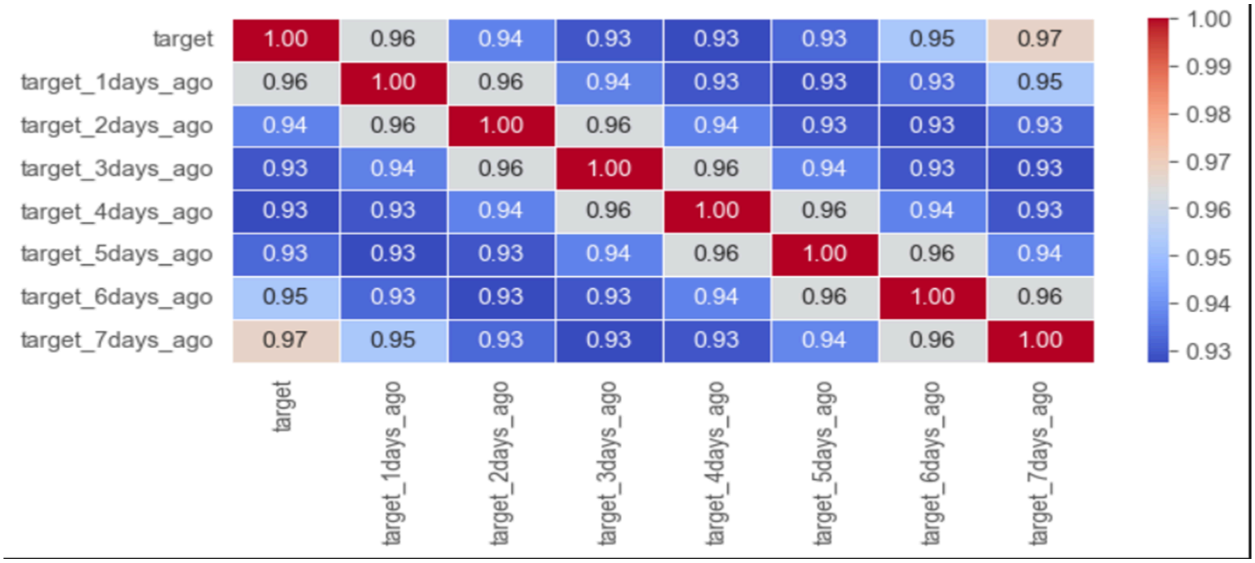
Inference - For outlier correction i used 1.5IQR method - any point which is not in range between lower bound and upper bound is termed as outlier and is negated



Inference - This subplot is 1x3 subplot of the target variable. The most salient feature is the skewness of the target variable i.e. the electricity amount so to combat that i used log transformation and plotted it. In the right most plot i have taken a Cumulative distribution function of target which shows that there is a sudden jump in production initially

Lag Analysis of Target Variable

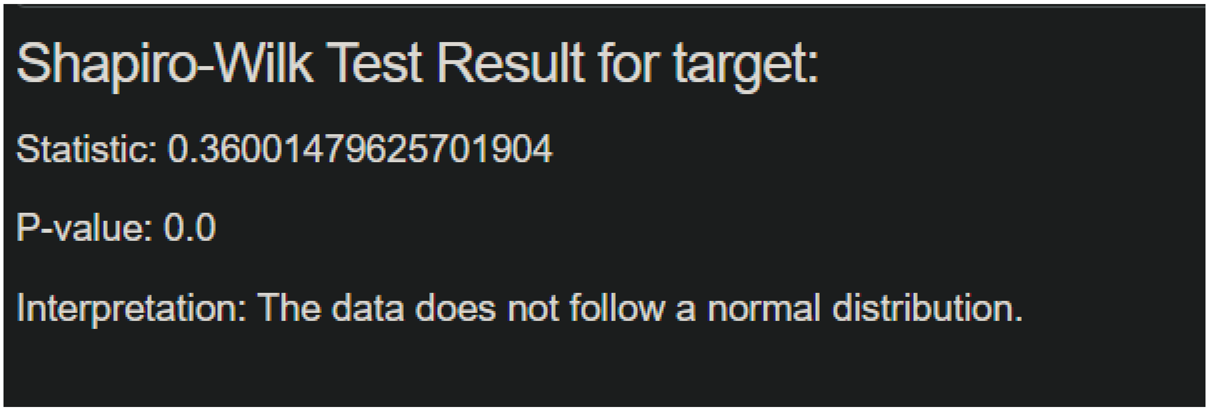
Lag analysis in time series problems is very important, and is performed to investigate the relationship between a feature and its past values over time. It involves creating lagged versions of the time series data and examining how the lagged values are correlated with the feature



	target	target_1days_ago	target_2days_ago	target_3days_ago	target_4days_ago	target_5days_ago	target_6days_ago	target_7days_ago
target	1.00	0.96	0.94	0.93	0.93	0.93	0.95	0.97
target_1days_ago	0.96	1.00	0.96	0.94	0.93	0.93	0.93	0.95
target_2days_ago	0.94	0.96	1.00	0.96	0.94	0.93	0.93	0.93
target_3days_ago	0.93	0.94	0.96	1.00	0.96	0.94	0.93	0.93
target_4days_ago	0.93	0.93	0.94	0.96	1.00	0.96	0.94	0.93
target_5days_ago	0.93	0.93	0.93	0.94	0.96	1.00	0.96	0.94
target_6days_ago	0.95	0.93	0.93	0.93	0.94	0.96	1.00	0.96
target_7days_ago	0.97	0.95	0.93	0.93	0.93	0.94	0.96	1.00

Inference - The target is highly correlated with its lag values - The lag correlation consequently decreases but there is a sudden jump in correlation in last and the lag7 has second highest correlation with target

Normality tests -

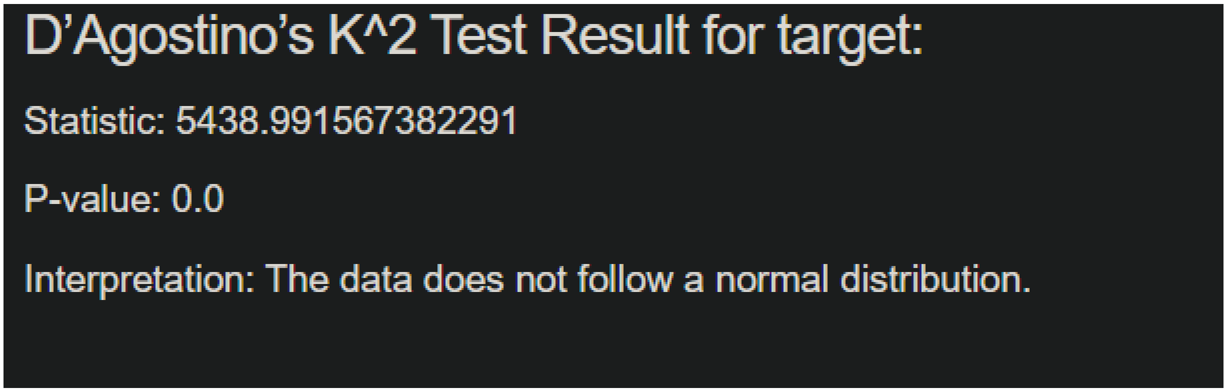


Shapiro-Wilk Test Result for target:

Statistic: 0.36001479625701904

P-value: 0.0

Interpretation: The data does not follow a normal distribution.

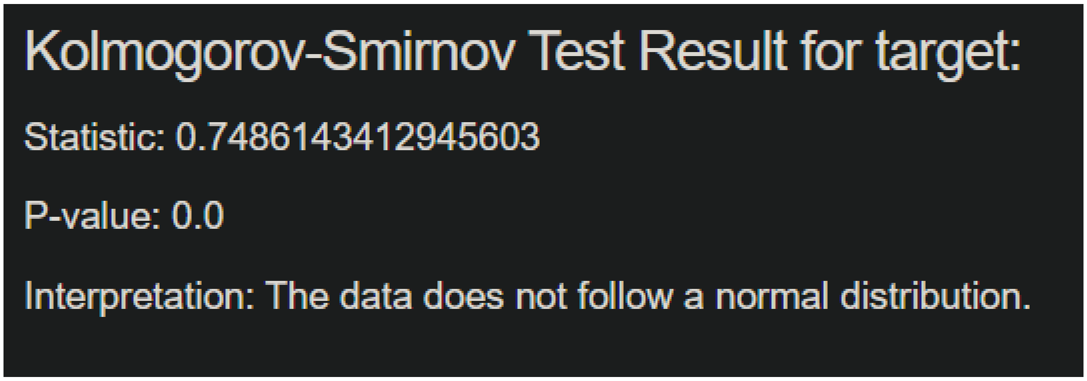


D’Agostino’s K^2 Test Result for target:

Statistic: 5438.991567382291

P-value: 0.0

Interpretation: The data does not follow a normal distribution.



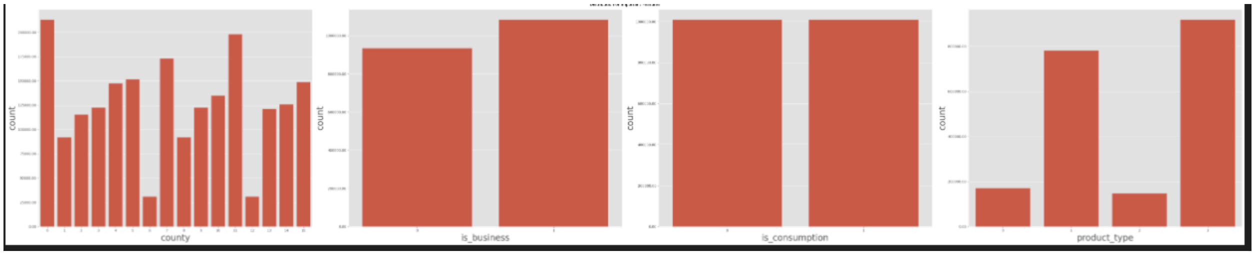
Kolmogorov-Smirnov Test Result for target:

Statistic: 0.7486143412945603

P-value: 0.0

Interpretation: The data does not follow a normal distribution.

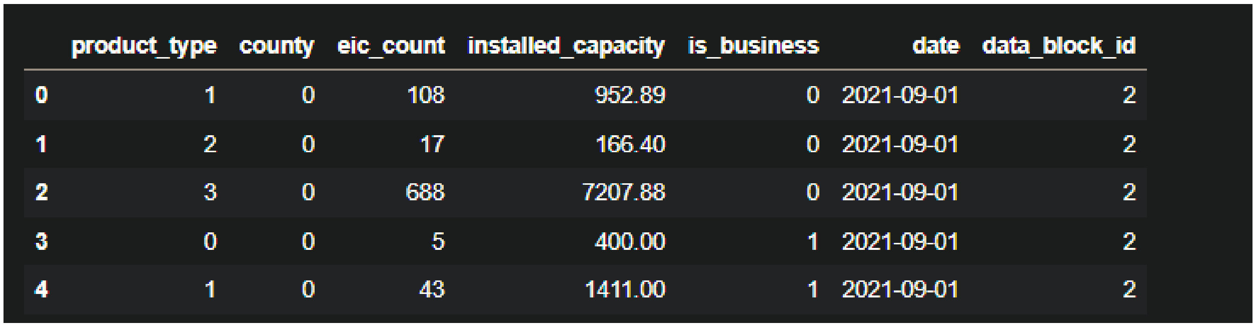
After performing KS test, Normal test and shapiro test it is clear that p value is less than the threshold value i.e. 0.01 so target distribution is not normally distributed



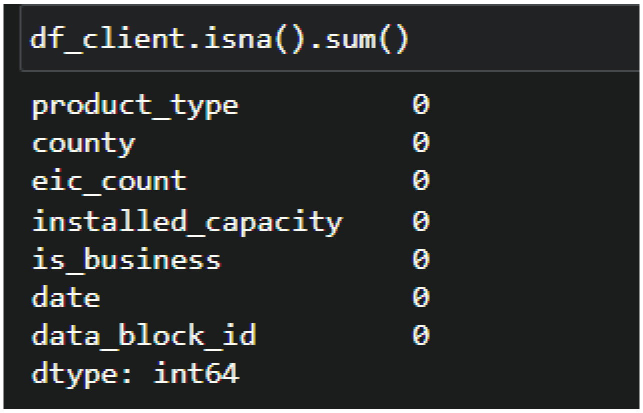
This subplot shows the segment analysis of the granularity of the train data

Analysis of client data

Snippet of client dataframe

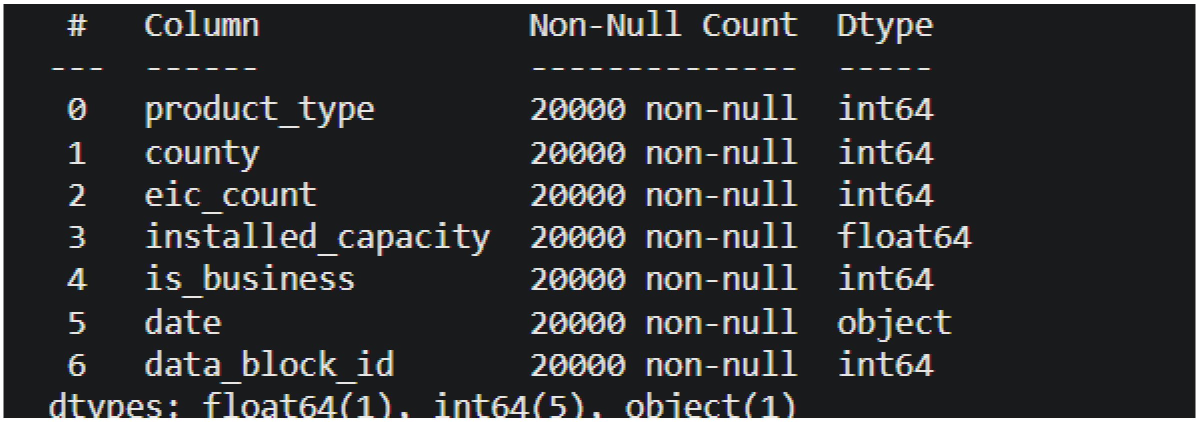


	product_type	county	eic_count	installed_capacity	is_business	date	data_block_id
0	1	0	108	952.89	0	2021-09-01	2
1	2	0	17	166.40	0	2021-09-01	2
2	3	0	688	7207.88	0	2021-09-01	2
3	0	0	5	400.00	1	2021-09-01	2
4	1	0	43	1411.00	1	2021-09-01	2



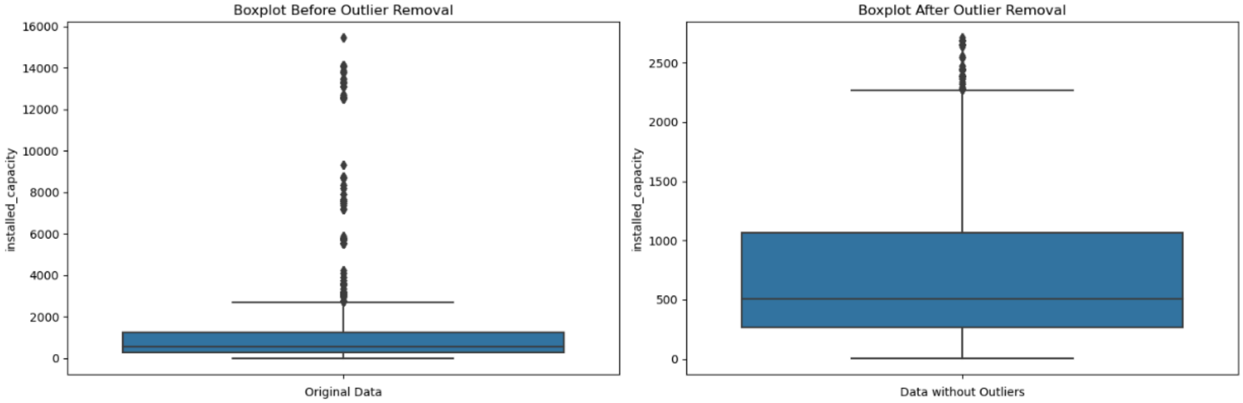
```
df_client.isna().sum()
```

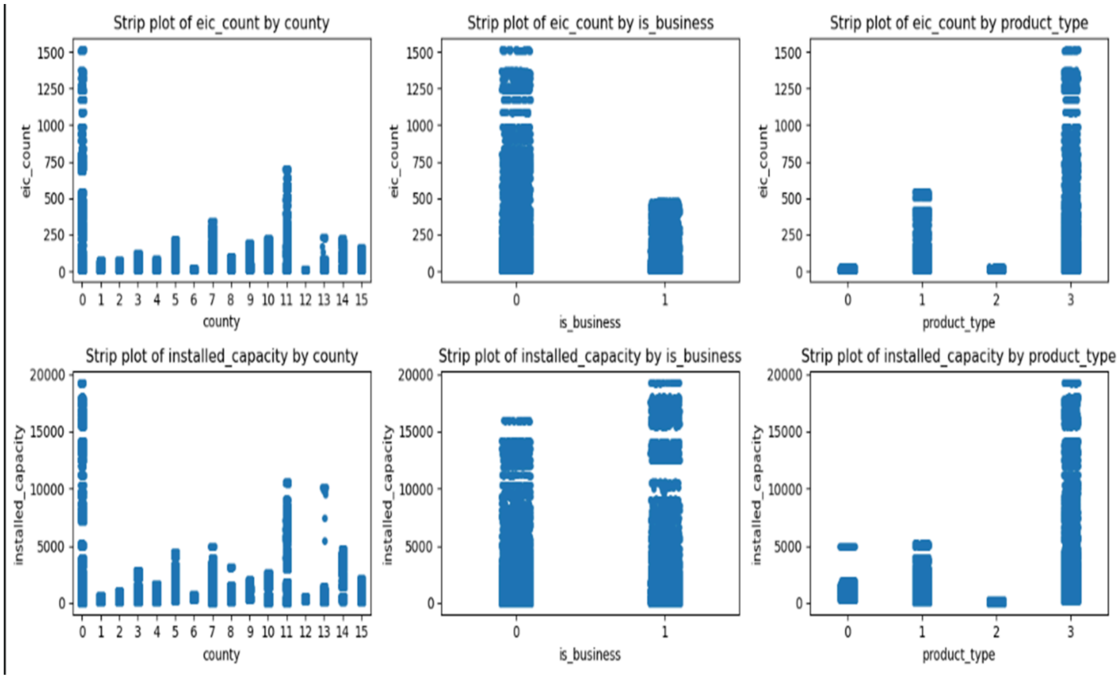
```
product_type          0
county                0
eic_count             0
installed_capacity    0
is_business           0
date                  0
data_block_id         0
dtype: int64
```



```
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   product_type        20000 non-null  int64
 1   county              20000 non-null  int64
 2   eic_count           20000 non-null  int64
 3   installed_capacity  20000 non-null  float64
 4   is_business         20000 non-null  int64
 5   date                20000 non-null  object
 6   data_block_id       20000 non-null  int64
dtypes: float64(1), int64(5), object(1)
```

Client data has 2 main columns - EIC count which is basically the european consumption points and installed capacity - which is the installed solar panel capacity for that particular client in that particular location for a product type

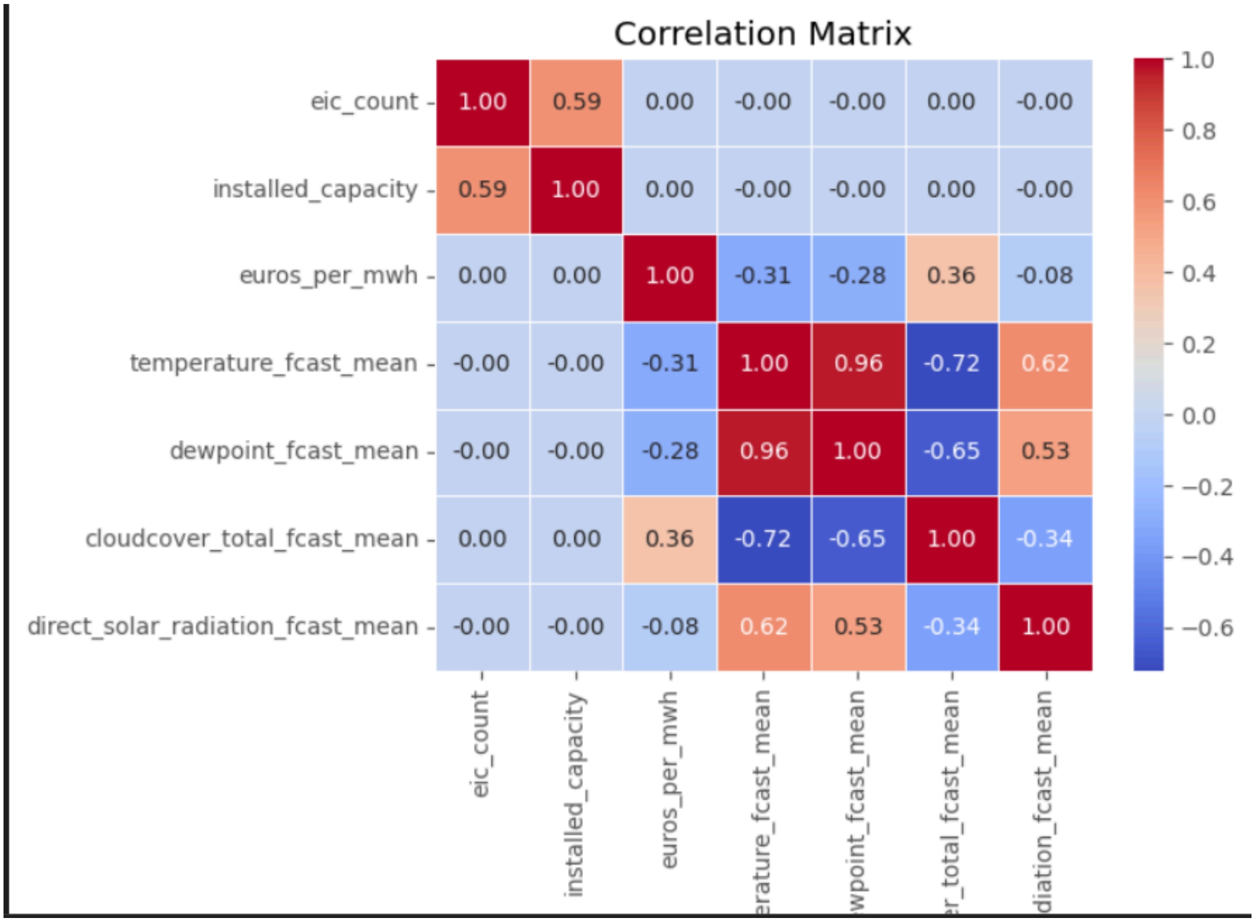




Inference - These are 2x3 strip plots for EIC count and installed capacity vs the numerical features in the client data. From the county subplot - county 0 has the maximum amount of consumption points and installed panel capacity.

Organizations which are not into business have greater EIC count as compared to organizations who are into business. Organizations who are into business have a higher installed capacity of solar panels. In products the product type 3 has the highest EIC count and installed solar panel capacity

Heatmap between important features



Inference - Eic count is highly correlated with installed capacity
Temperature variables are highly correlated with dew point temperature

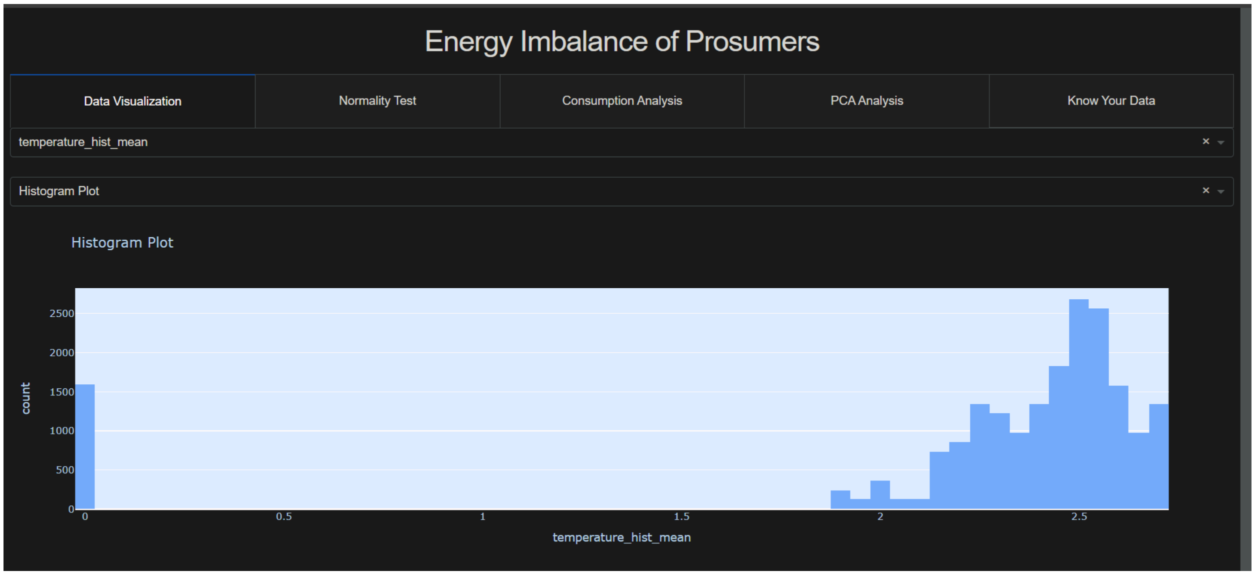
Phase 2

Dash app : The dash app has 5 tabs namely -

1.Data Visualization

2.Normality Test
3. Consumption Analysis(Time series dashboard)
4. PCA analysis
5. Know Your Data

App Layout

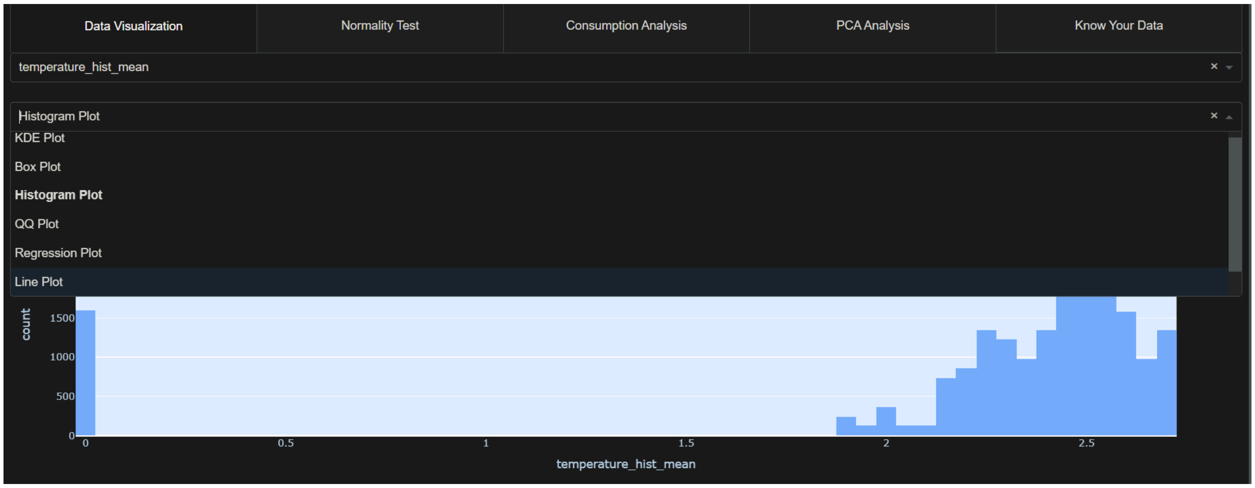


Energy Imbalance of Prosumers

temperature_hist_mean

Histogram Plot

First Tab - Data Visualization -



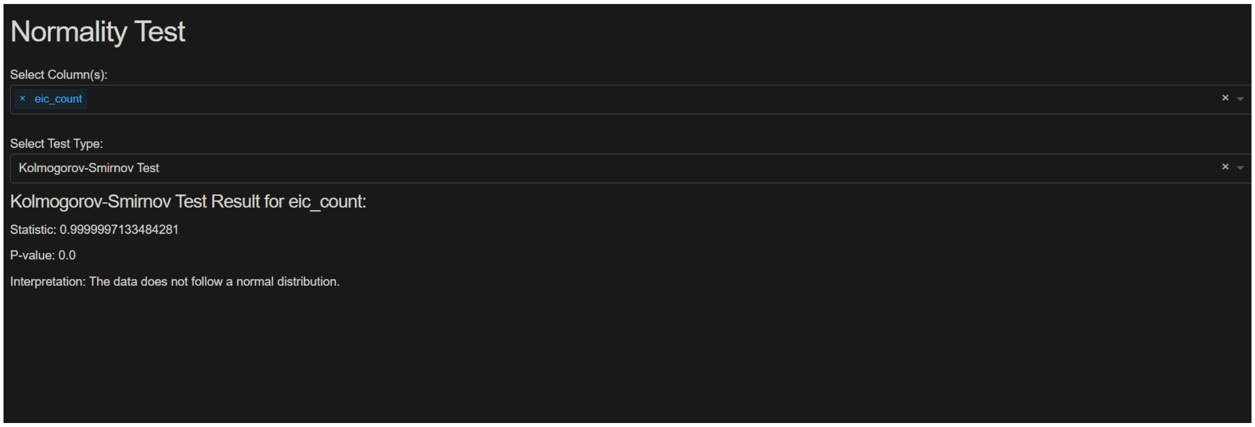
temperature_hist_mean

Histogram Plot

User can select various plots like line plot, reg plot, histogram, box plot and view the plot for the selected column

2nd tab

Normality test tab



Normality Test

Select Column(s):

× eic_count

Select Test Type:

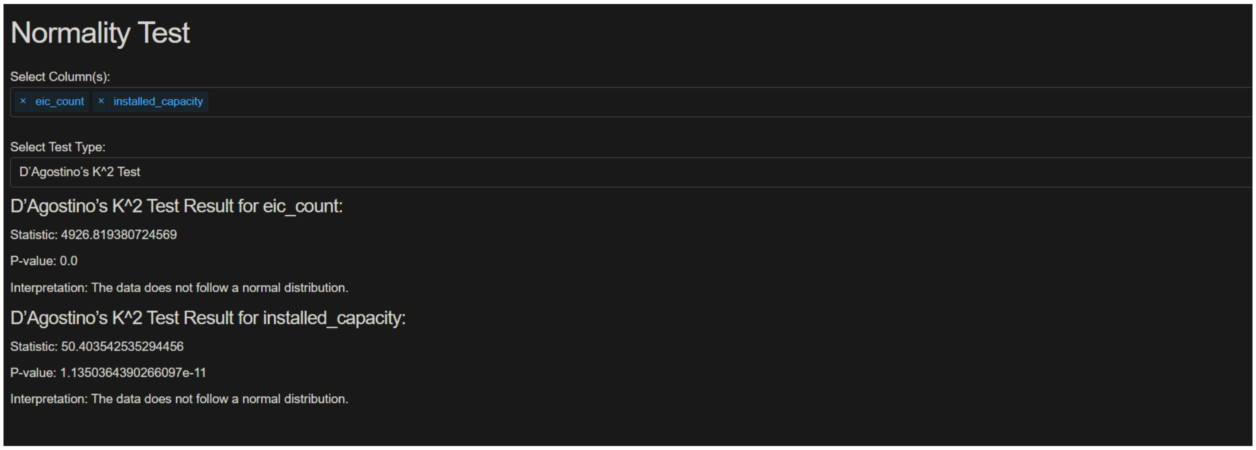
Kolmogorov-Smirnov Test

Kolmogorov-Smirnov Test Result for eic_count:

Statistic: 0.9999997133484281

P-value: 0.0

Interpretation: The data does not follow a normal distribution.



Normality Test

Select Column(s):

× eic_count × installed_capacity

Select Test Type:

D'Agostino's K^2 Test

D'Agostino's K^2 Test Result for eic_count:

Statistic: 4926.819380724569

P-value: 0.0

Interpretation: The data does not follow a normal distribution.

D'Agostino's K^2 Test Result for installed_capacity:

Statistic: 50.403542535294456

P-value: 1.1350364390266097e-11

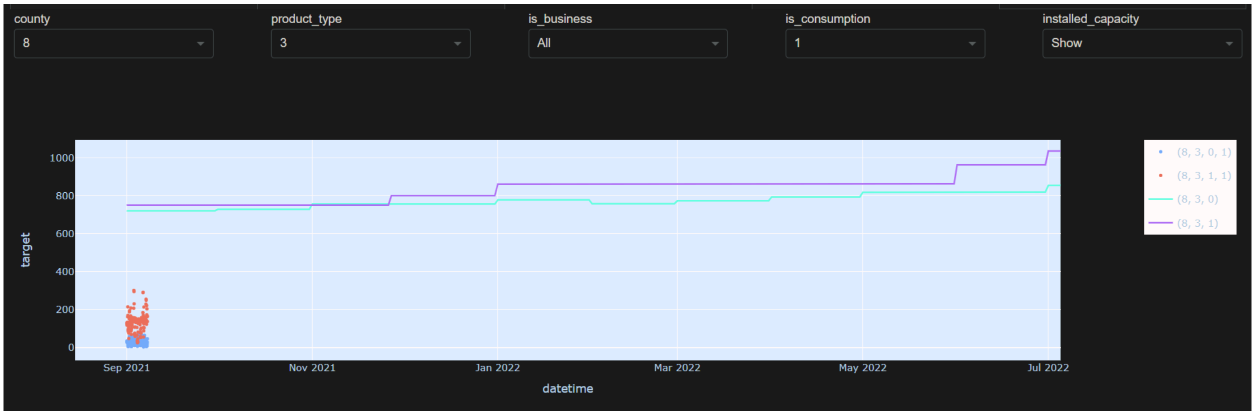
Interpretation: The data does not follow a normal distribution.

User can select any number of columns and and check the results from the following normality tests

1. KS test
2. Normal DA test
3. Shapiro wilk test

3rd tab

This is the consumption analysis tab which is a dashboard in itself and is fully customizable



county: 8

product_type: 3

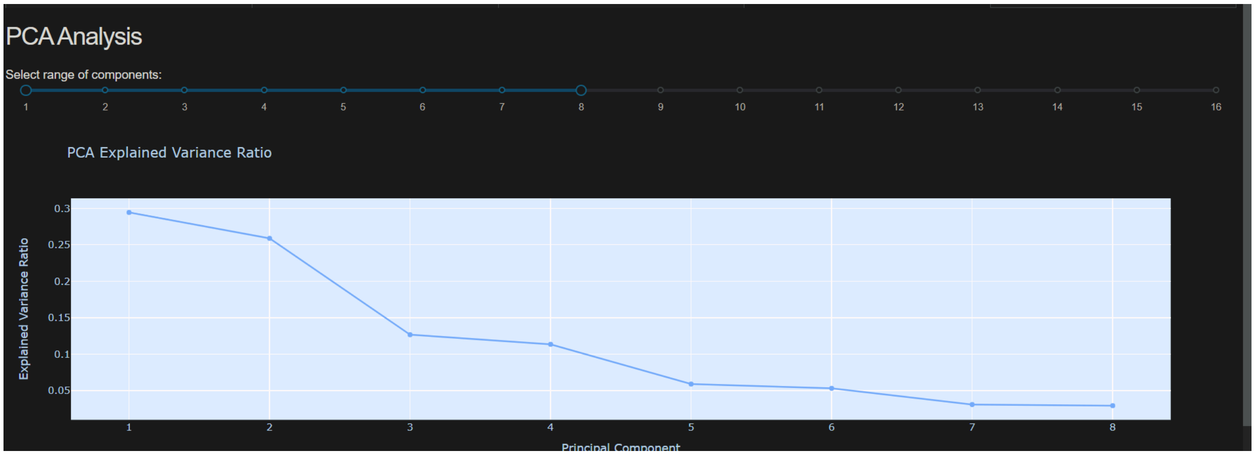
is_business: All

is_consumption: 1

installed_capacity: Show

User can select any county, any product type, any flag(business and consumption) and find the relationship between the installed capacity and the consumption/production based on the values selected

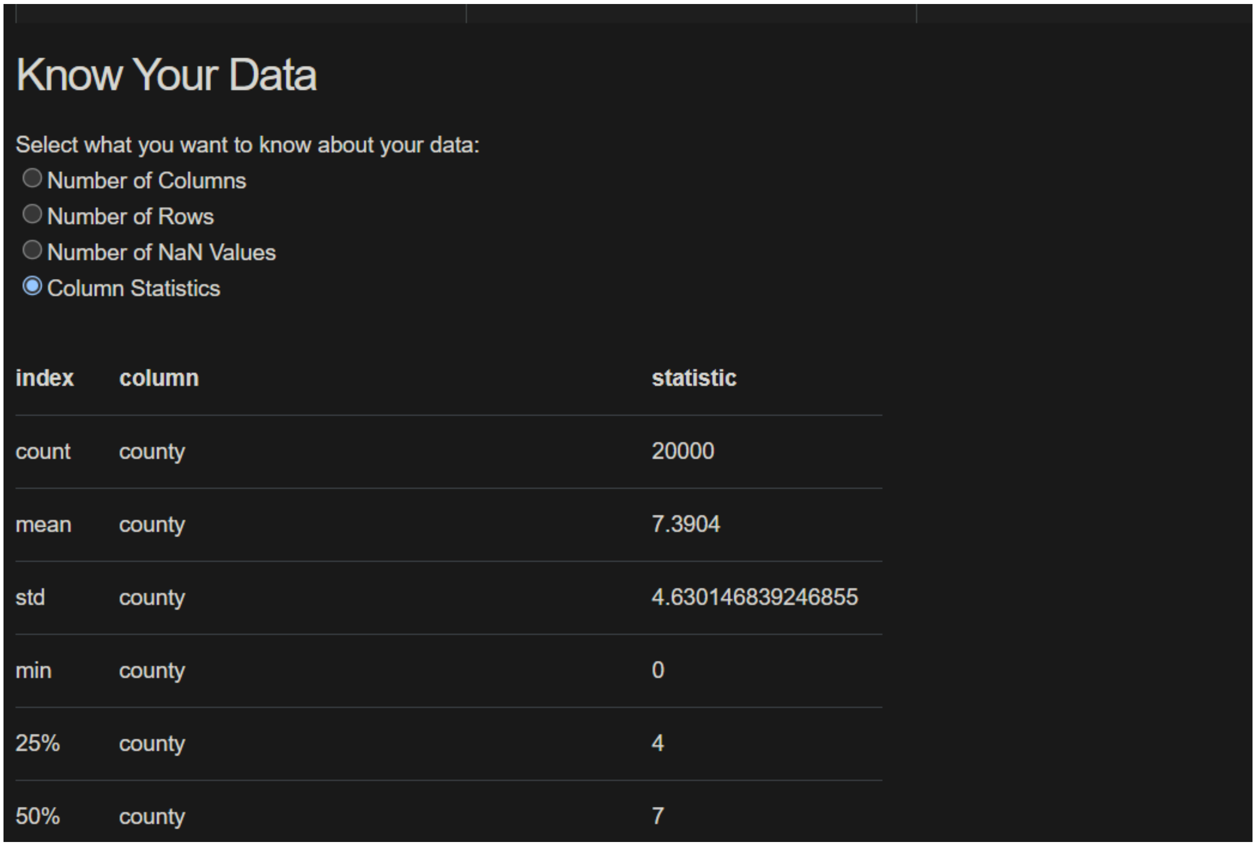
Tab4


PCA Analysis

Select range of components:

PCA analysis - this tab can be used to select any number of components and find the explained variance ratio dependence on the components

Tab5

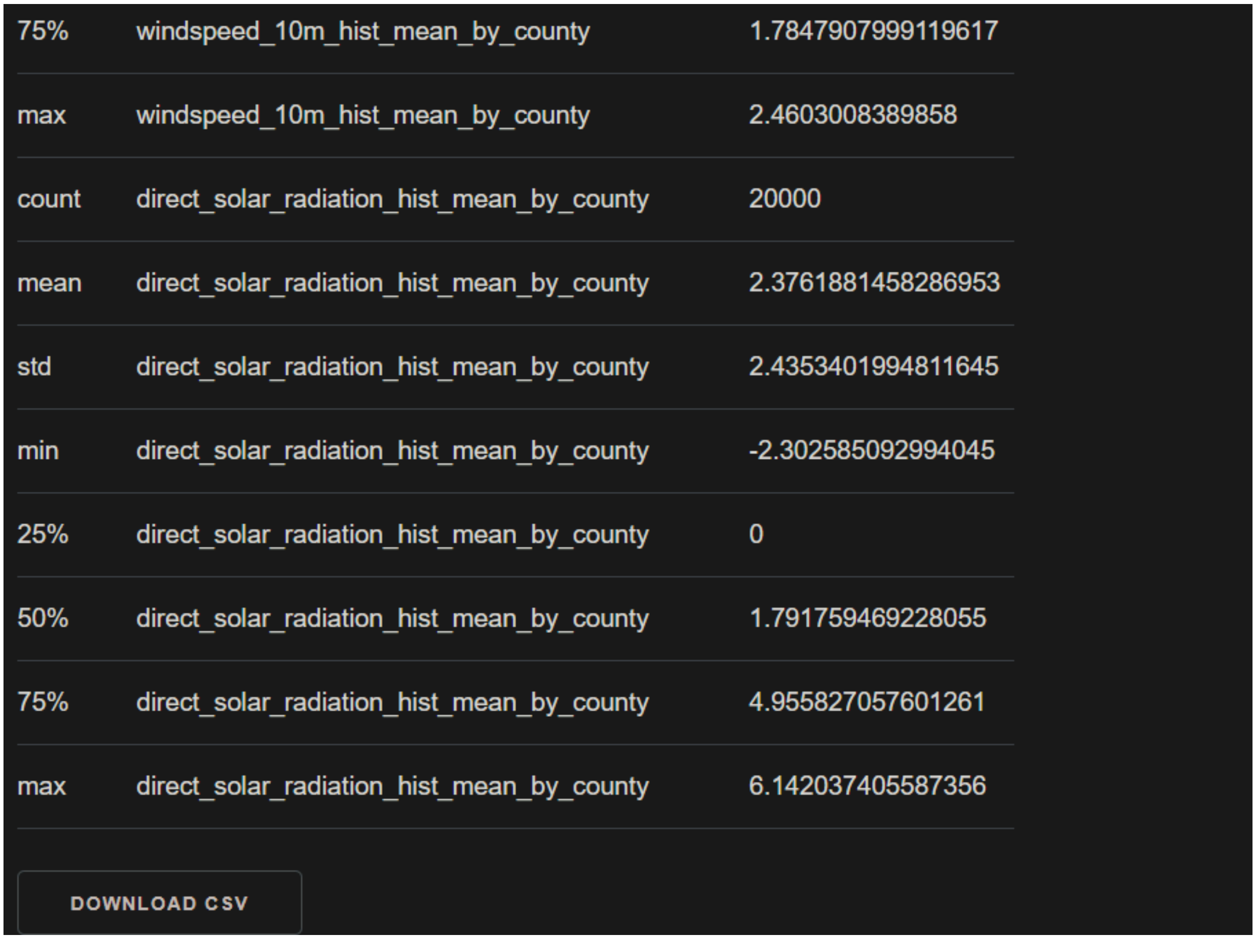


Know Your Data

Select what you want to know about your data:

○ Number of Columns
○ Number of Rows
○ Number of NaN Values
● Column Statistics

index	column	statistic
count	county	20000
mean	county	7.3904
std	county	4.630146839246855
min	county	0
25%	county	4
50%	county	7



75%	windspeed_10m_hist_mean_by_county	1.7847907999119617
max	windspeed_10m_hist_mean_by_county	2.4603008389858
count	direct_solar_radiation_hist_mean_by_county	20000
mean	direct_solar_radiation_hist_mean_by_county	2.3761881458286953
std	direct_solar_radiation_hist_mean_by_county	2.4353401994811645
min	direct_solar_radiation_hist_mean_by_county	-2.302585092994045
25%	direct_solar_radiation_hist_mean_by_county	0
50%	direct_solar_radiation_hist_mean_by_county	1.791759469228055
75%	direct_solar_radiation_hist_mean_by_county	4.955827057601261
max	direct_solar_radiation_hist_mean_by_county	6.142037405587356

This tab can be used to download the csv as well find statistics of data

Phase 3

Deployment -

Deployed Link GCP link - https://dashapp-pwuctfcgrq-ue.a.run.app/

The deployment is performed on Google cloud platform

Following are the commands used for pushing and deploying the code to production

Final-exam-420616 - Project name

```
1.docker build -f Dockerfile -t gcr.io/final-exam-420616/test:test .
```

```
2 docker push gcr.io/final-exam-420616/test:test
```

```
3.gcloud run deploy dashapp --image
gcr.io/final-exam-420616/test:test
```

Conclusion -

The project focuses on analyzing the energy imbalance problem of prosumers in estonia and uses advanced visualization techniques to check the intricacies of the problem

References -

https://dash.plotly.com/

https://www.kaggle.com/competitions/predict-energy-behavior-of-prosumers/overview

https://seaborn.pydata.org/