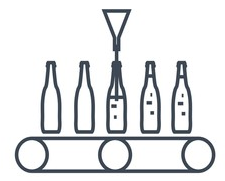


Energy Management in

Beverage Industry

Using IoT sensor



**CONTENTS**

1. Project objective……………………………………………………………………………….………. 2
2. Initiation and Data Preparation…….………………………………………………….………. 6
3. Data Visualization…………………………………………………………………………….……… 10
4. Model Building...………………………………….……………………………………………………. 16
5. Deployment…….……………………………………………………………………………….………. 21

1. Conclusion..………………………………………………………………………………………………. 24

**PROJECT OBJECTIVE**

Project Background

Energy Management

Energy Management is the process of tracking and optimizing energy consumption to conserve usage in the Industry. Energy consumption by various units are monitored timely and the valuable insights helps in understanding the pattern and hence timely user intervention can save a lot of energy.

Overview of Beverage Industry

The beverage Industry also known as drink industry is the one which manufactures drinks. It produces alcoholic such as distilled spirits, wine and brewing as well as non-alcoholic drinks such as fruit juices, tea or coffee and other flavored drinks etc.

Benefits of Energy Management in Beverage Industry

* It increases overall productivity of the manufacturing plant.
* It helps to detect malfunctioning electrical devices timely.
* It is eco-friendly practice so makes the technology more constructive.

Aim:

* Main aim of this project is to manage the energy consumption by the overall plant
* To monitor the energy consumption on real time basis by individual plant units and overall plant using IoT sensor.
* To find whether at any instance energy consumption is low or high
* To find whether all the electric motors are working at optimum performance.
* To minimize power losses and overloading

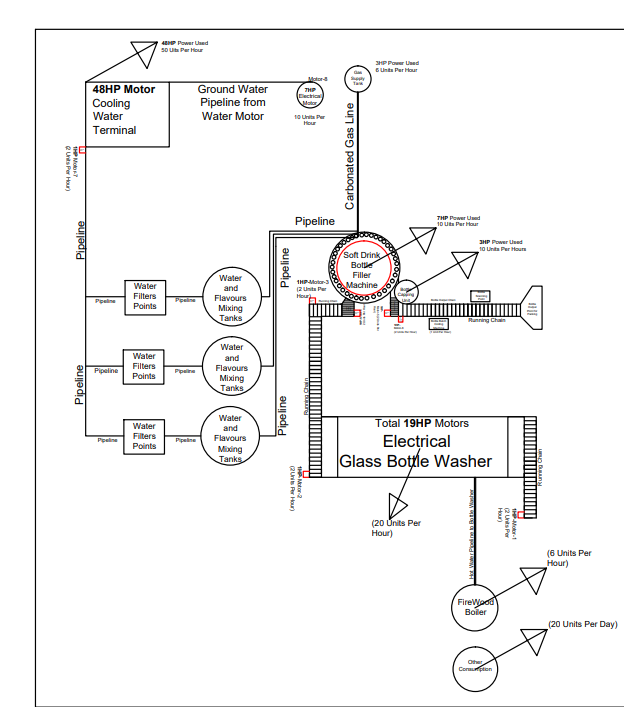
Business objective:

* Minimize: Energy losses, overloading
* Maximize: Energy efficiency

Business Constraint:

* To improve the performance of large amounts of data.
* Security of the data and working of the IoT sensor.

Flow of process in Beverage Plant:



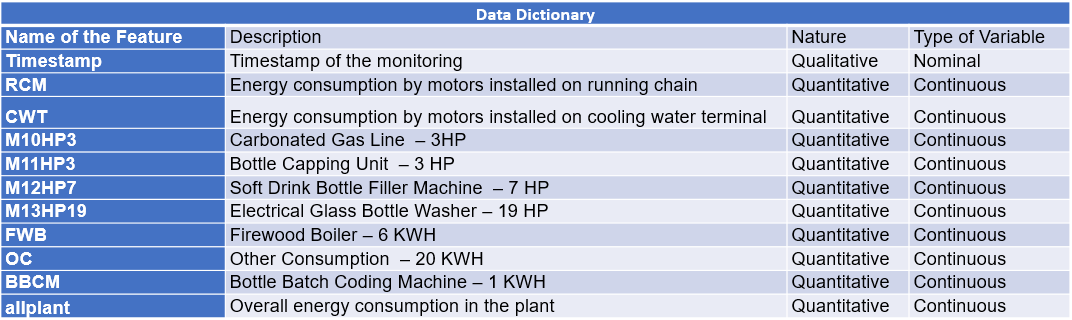
Project Dataset:

Data is collected on real time basis with the help of IoT sensor installed in the various electric motors in each of unit of the manufacturing plant. For every 10 seconds the data of energy consumption by all these individual units is obtained from the IoT sensor.

* Calculation = ((energy in kWh x 1000)/60 x 60) x 10

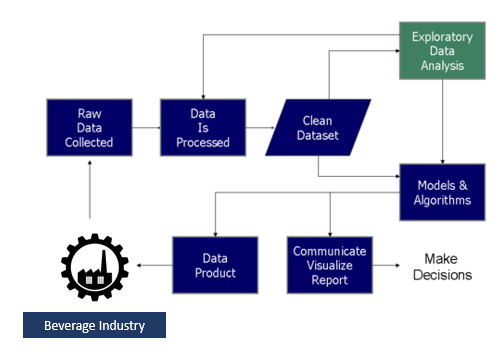
Electrical Energy consumed by individual motors installed in individual unit of the production system acts as independent variables and another meter is installed at the outlet of production system that measures overall electricity consumption acts as a target variable.

The features present in the data are:

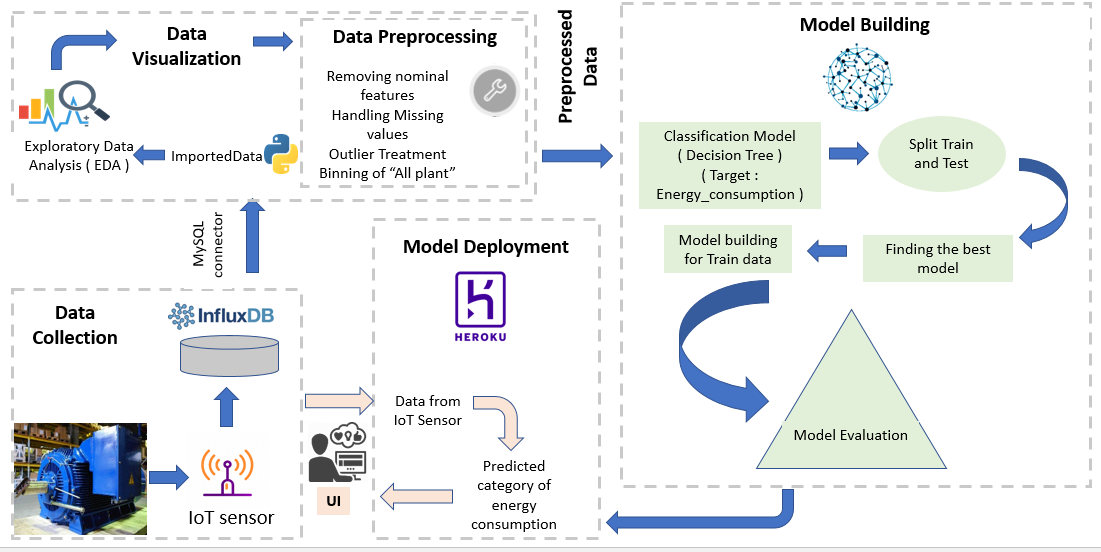


**INITIATION AND DATA PREPARATION**

High level Flowchart of the process:



Detailed Project Architecture:



Data Preparation:

This preprocessing step involves exploratory data analysis, data visualization, checking null and duplicate values, outlier treatment , removing undesirable nominal features such as “Timestamp” and usage of binning to convert output variable “allplant” into categorical variable so that data can be made to be used for classification model.

Exploratory Data Analysis:

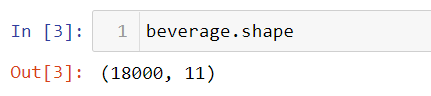
This step is used to identify and summarize the main characteristics in a data set through data analysis and investigation. We used data visualizations to better understand how best to manipulate data sources, to determine which statistical techniques are most appropriate for data analysis, and for choosing the right features for a model.

EDA includes fallowing steps:

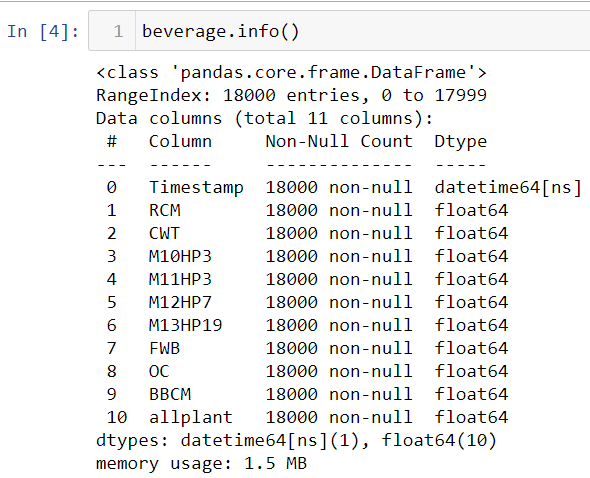
* Details About Data
* Data cleaning
* Statistical Insights
* Univariate analysis
* Bi-variate/Multivariate analysis

Details about Data:

The first and foremost step of any data analysis, after loading the data file, should be about checking few introductory details like, no. Of columns, no. of rows, types of features (categorical or Numerical), data types of column entries.



* Data has 18000 rows and 11 columns

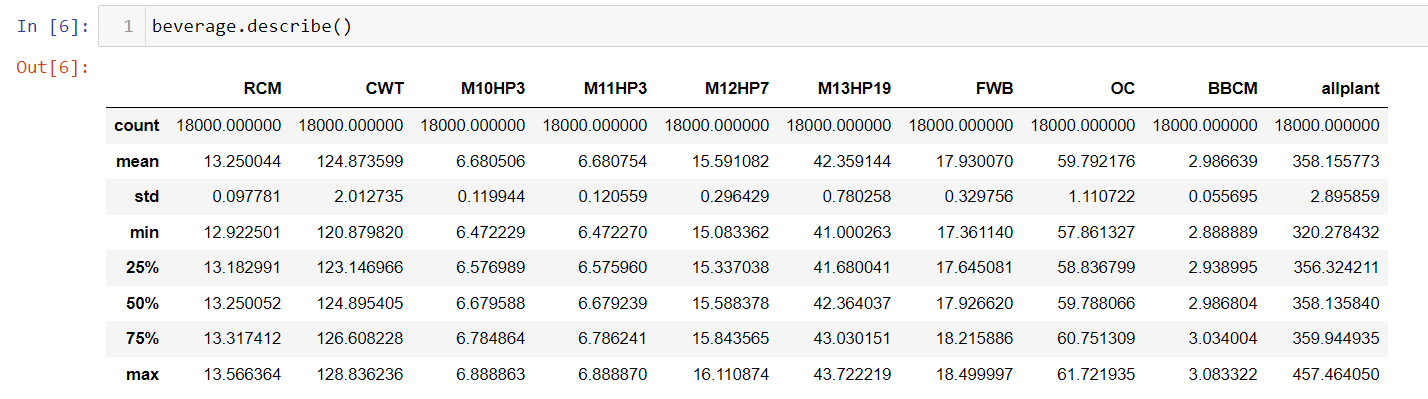


Data Cleansing:

* **Nominal Features:** The “Timestamp” is just nominal data which is only showing the timestamp of the monitoring and as no relevance with other features, so it is removed from the dataset.
* **Missing Values:** There are no missing values and null values in the dataset.
* **Outliers:** Here in this Dataset, There are outliers in the features like “RCM” and “allplant” so the outliers were replaced with the max threshold of the feature.
* **Binning:** The target variable “allplant” is continuous data and to make the data compatible to classification problem it is then converted to categorical by the concept of binning. “allplant” is converted into two categories namely high and low. From 250 to 358 watts per 10 seconds is treated as one category i.e. low consumption and from 358 watts per 10 seconds onwards it is treated as another category i.e. high consumption.

Statistical Insights:

Below is the count, mean, standard deviation, minimum and maximum values and the quantiles of the data.



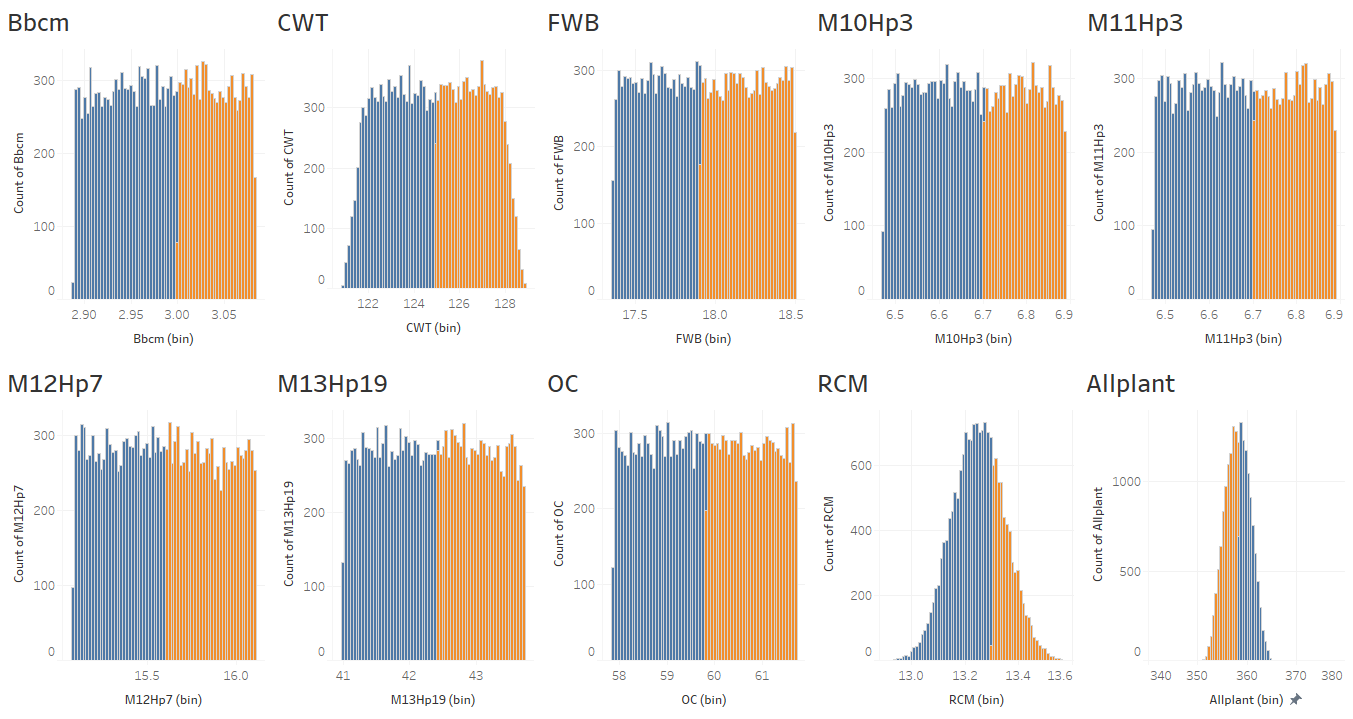
* There is not much difference between 75th %tile and max values.
* Thus observations suggests that there is no Outliers in our data set.

**Data Visualization**

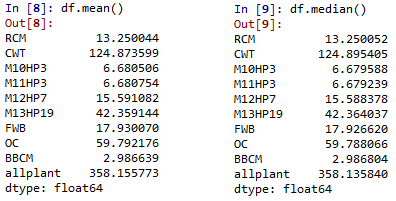
Univariate Analysis:

This shows every observation/distribution in data on a single data variable. It is shown with the help of Histogram.

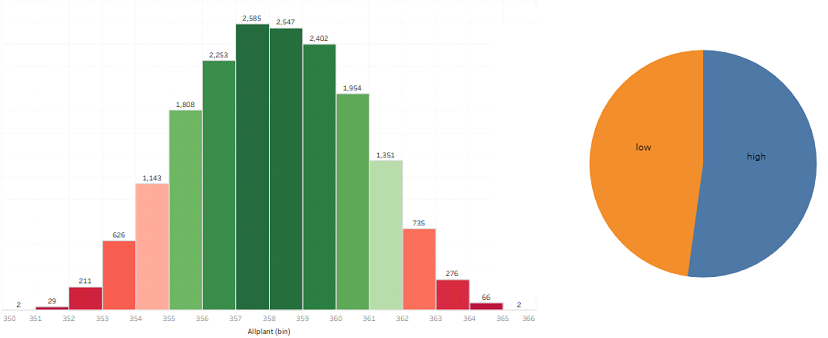
**Histogram:** A histogram is a value distribution plot of numerical columns. It basically creates bins in various ranges in values and plots it where we can visualize how values are distributed. We can have a look where more values lie like in positive, negative, or at the centre (mean).



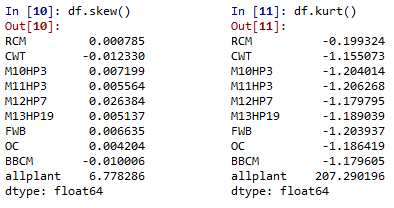
**Skewness:** Skewness refers to a distortion or asymmetry that deviates from the symmetrical bell curve, or [normal distribution](https://www.investopedia.com/terms/n/normaldistribution.asp), in a set of data. If the curve is shifted to the left or to the right, it is said to be skewed. Skewness can be quantified as a representation of the extent to which a given distribution varies from a normal distribution. A normal distribution has a skewness of zero.



Mean and median values each variable are close in range. So histogram for output and every input variable having symmetric nature with less skewness values, thereby distribution of data on both half are almost equal.



In the above figure illustrate the “Allplant” output variable data distribution in data. The histogram plot represents “Allplant” output variable continuous data distribution, and pie chart represents that of its categorical class distribution in “low” & “high” classes according to the threshold value of electrical power consumption. The continuous power consumption value of “Allplant” data that’s comes above the threshold limit treated as “high” class category and that of values comes below threshold limit treated as “low” class category.



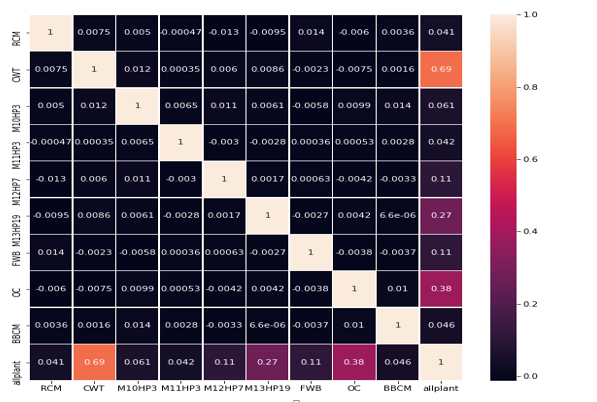
**Kurtosis (ku):** Kurtosis is a measure of relative peakedness of a distribution. It is a shape parameter that characterizes the degree of peakedness. A distribution is said to be leptokurtic when the degree of peakedness is greater than zero, it is mesokurtic when the degree of peakedness is equal to zero, and it is platykurtic when the degree of peakedness is less than zero.

As every input variables shows shorter or truncated tail ends with more like uniform distribution, they having platykurtic distribution with Ku value less than zero. Since the output variable “Allplant” having higher peakedness they showing leptokurtic distribution with Ku value greater than zero.

### Bi-Variate analysis:

Bivariate analysis displays are done to reveal the relationship between two data variables. It shown with the help of Heat map

Heat Map: A heat map is a two-dimensional representation of data in which values are represented by colors. Here heat map provides correlation between various features.



* Lighter shades represents positive correlation while darker shades represents negative correlation.
* Here we can infer that “allplant” has strong positive correlation with “CWT” and a moderate relation with M13HP19, OC.

Power Consumption by different units of plant:

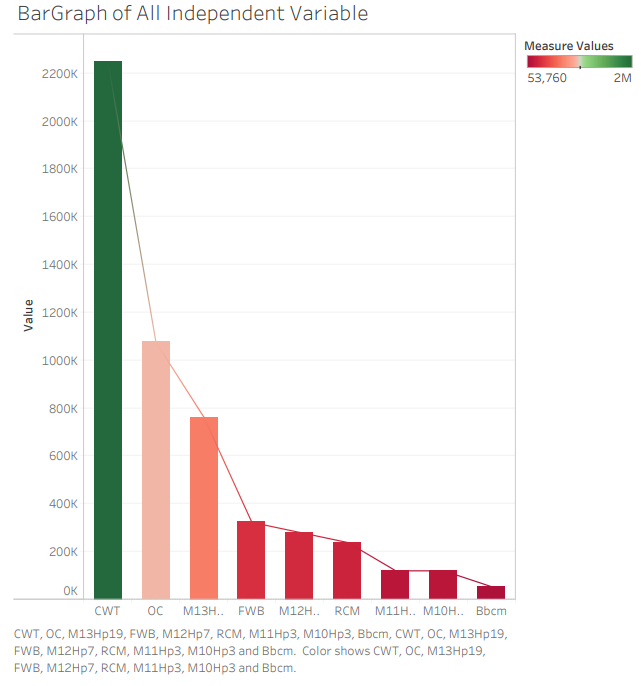


Fig. shows the total power consumption of each motor’s over a period of time. The motors with higher power ratings will consumes more power than that of motors with less ratings. So the bar plots illustrates that,

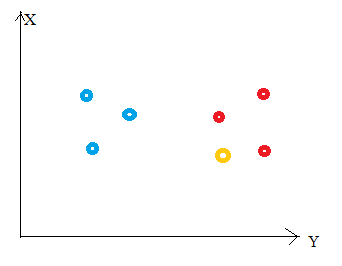
* The “CWT” motor has highest power rating, since it consumes more power compares to all others.
* The “OC” and “M13HP19” motors also have higher power ratings, since they shows significant power consumption.
* The “Bbcm” motor has the lowest power rating as its consumption is very low compares to all others.

**MODEL BUILDING**

KNN Algorithm:

KNN stands for K- Nearest Neighbor algorithm, is a non-parametric supervised machine learning problem which can be used for both classification as well as regression. It assumes that similar things are in close proximity. In other similar things are near to each other. K is the number of nearest neighbors that the algorithm considers for classification. In KNN no weights are learned instead the entire training dataset is stored in the memory. Therefore, model representation for the KNN model is the entire training dataset.

KNN model works by segregating the data points in different classes based upon the similarity identified by the distance between the data points by various distance metrics.



Here, nearest neighbors are those data points that have minimum distance in feature space from our new data point. And K is the number of such data points we consider in our implementation of the algorithm. Therefore, distance metric and K value are two important considerations while using the KNN algorithm.

Cons:

* Accuracy depends upon the quality of data
* Slow performance when data is large
* Sensitive to the scale of the data and irrelevant features
* Requires high memory as all the training data needs to be stored
* It is computationally expensive

In our model different values of K have been tried to get the best possible accuracy and various distance metrics such as Euclidean, Manhattan etc. have also been experimented upon.

Decision Tree Classification:

Decision Tree classification is non parametric supervised machine learning algorithm which makes a flowchart by learning simple decision rules from the data features and with the help of this tree like flow chart it predicts the class of the target variable. Each internal node represents a test on a feature, each leaf node represents class label i.e. decision taken after computing all the features and branches represent the conjunctions of features that leads to those class label. The path from root to leaf represents classification rules.

In this algorithm, Training data is used to build model. Tree generator determines which variable to split at a node and what will be the value of the split, decision of split and stop is then taken based on the condition and finally at the end terminal nodes are assigned to the label.

Decision tree algorithm is also called as greedy algorithm is used. Below are the steps of the algorithm:

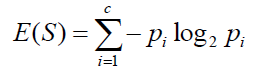
* Tree is constructed in a top down recursive divide and conquer manner
* At start, all training data are at the root
* Attributes are categorical if not then they are discretized well in advance
* Input data is partitioned recursively based on selected attributes
* Test attributes at each node are selected on the basis of statistical measure such as information gain.

Parameters used in Decision Tree:

This algorithm uses two parameters such as max\_depth and entropy as a criterion.

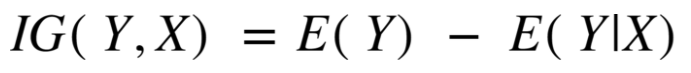
**max\_depth**: It is one of the hyperparameter used in the decision tree classification algorithm to tune the model for optimum performance i.e. to avoid overfitting. It takes an integral value and based upon that it decides till what depth the decision tree has to be made. In our model we kept max\_depth=9 which is giving optimum accuracy results.

**criterion**: It is another hyperparameter that is used in the decision three which is instrumental in deciding how the decision tree will grow. This parameter determines how the impurity of a split will be measured. Either “gini” or”entropy” can be used. In our model we used criterion=entropy. Decision trees uses information gain to split the node and gini or entropy criterion measures the impurity of the node. Greater the impurity, greater will be the information gain. The range of “entropy” is 0-1. Entropy and information gain can be calculated as below:



Where Pi is the probability that an arbitrary tuple belongs to class Ci

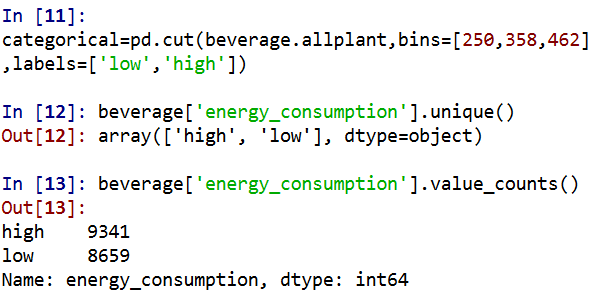
Information gain is calculated as below:



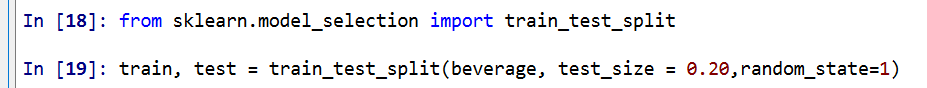
Above is information gain from X on Y.

Steps Involved:

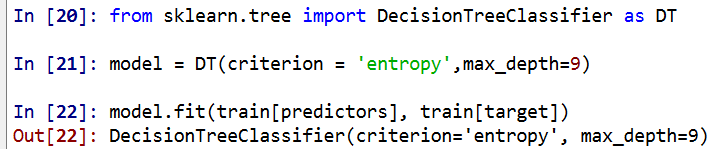
* Import the dataset
* Data preparation which includes data cleansing and binning of output variable “allplant” to convert it into categorical feature



* Setting predictors and target
* Splitting the data into test and train. Train = 80% and Test = 20%.



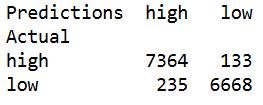
* Setting of parameters for the model and fit the data



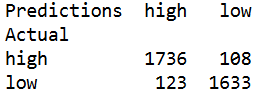
max\_depth = 9 and criterion is set as “entropy “

* Summarization of model performance using confusion matrix for train and test data.

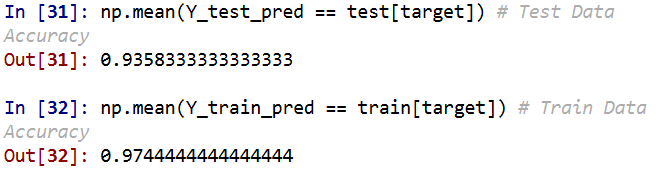
For Train:

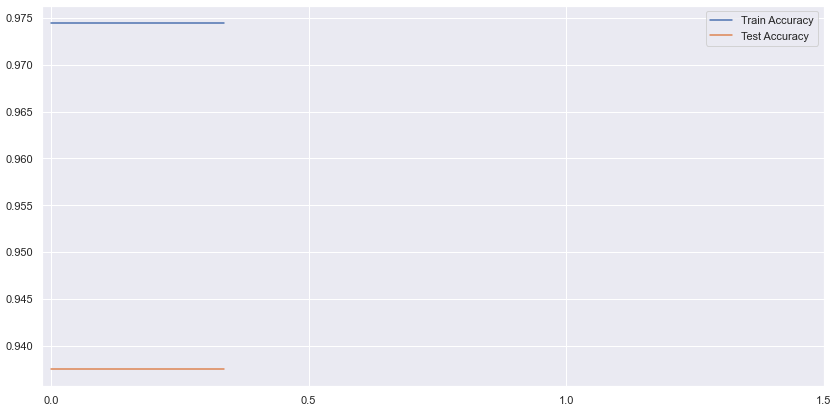


For Test:



* Model Evaluation by calculating train and test accuracy.





Above graph shows the model performance after applying post pruning which shows the model has attained the optimum performance So, based upon the performance of both the models it was decided to finalize decision tree classification algorithm as for a large dataset in the project not only it is giving faster performance but also giving optimal prediction accuracy.

**DEPLOYMENT**

In the Deployment Actually we have used the following tools:

* Python.
* Python Flask
* Html, CSS.
* Heroku
* GitHub

Python:

A tool used for Manipulation of data and model deployment using different libraries and Modules like Pandas, Flask, statsmodels.

Python Flask:

Python Flask is an open-source python library that makes it easy to create custom web apps for machine learning and data science. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third party libraries provide common functions.



Html, CSS:

HTML (Hypertext markup language) is use for making structure of the website to take inputs manually from the user.

CSS (Cascading Style sheet) is used for describing the presentation of the html document or styling the html structure to make it look good

Heroku.

Heroku is platform as a service (PaaS) that enables developers to build, run and operate application entirely in the cloud

GitHub.

Heroku integrates with GitHub to make it easy to deploy code living on GitHub to apps running on Heroku. When GitHub integration is configured for the Heroku app, Heroku can automatically build and release ( is the build is successful) pushes to specified GitHub Repository

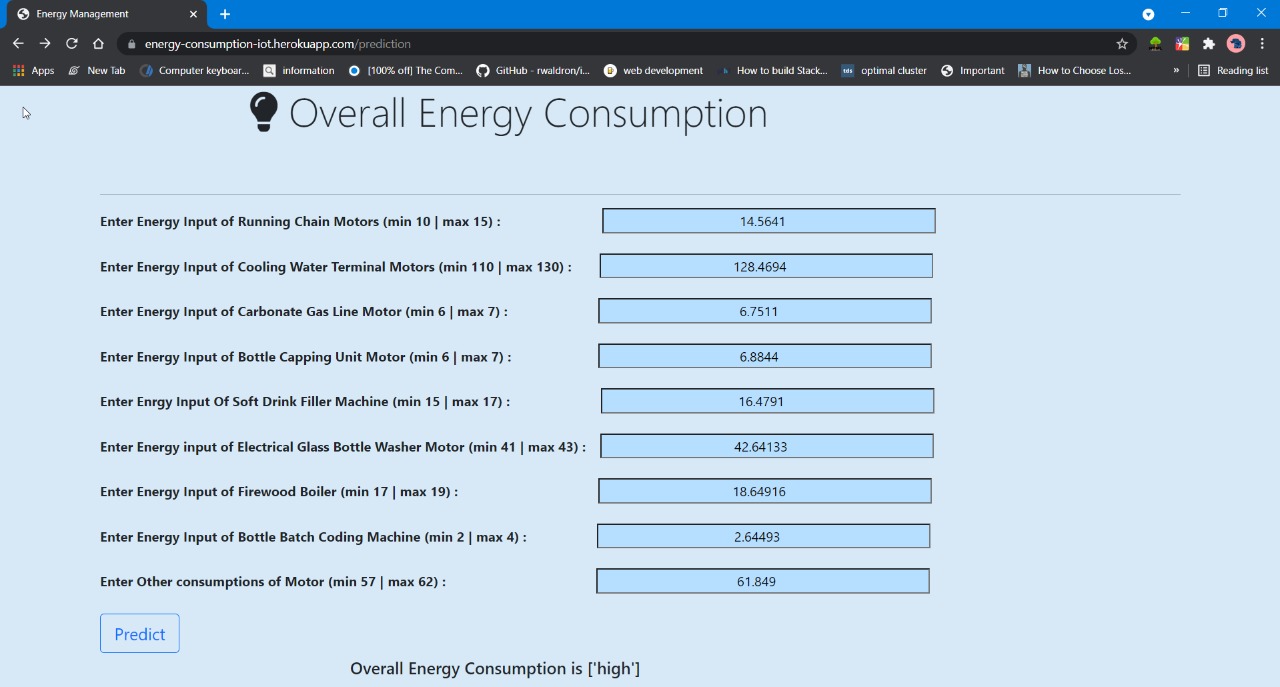
Output Graphical user interface, text, application

Description automatically generated

We have to enter Energy Consumption of motors in watts per 10 seconds that are collected from the IoT used in the industry and after Clicking Predict Button we get output if the consumption is Low or High

Graphical user interface

Description automatically generated



Links

Link to the Web app: <https://energy-consumption-iot.herokuapp.com/>

Link to the GitHub repository: [https://sgithub.com/Aniket240811/Energy-Management-Using-Iot-Sensors](https://github.com/Aniket240811/Energy-Management-Using-Iot-Sensors)

**CONCLUSION**

* The model will helps to predict all plant electrical power consumption meter reading as "high" & "low" in accordance with a threshold value for power consumption under constant production rate.
* At constant production rate, if all plant power consumption crosses the threshold value limit and comes with higher values, that indicating increased power losses happening in plant due to less efficient electrical part or devices. The model will predict such situations as output power consumption category in "high".

* Such predictions will enable the plant operators to enforce predictive maintenance on inefficient electrical parts of plant and rectify extra power losses issues.
* IoT enabled monitoring will reduce the manual task of keeping a check on electricity consumption and with this model whenever the energy consumption will be high , user will automatically get the report and hence immediate steps then can be taken to decrease the further consumption
* It provides energy consumption data for each unit so this will give information about the unit at a particular instance which is decreasing the overall production efficiency
* By the individual electricity consumption it also will help in keeping a check over functionality of the electrical component and hence will add positively towards the overall efficiency of the production system