IBM AI:101 Measure Energy Consumption

Mohan Krishna.N ([mohankrish3027@gmail.com](mailto:mohankrish3027@gmail.com))

Prakash.A ([prakashmokesh2004@gmail.com](mailto:prakashmokesh2004@gmail.com))

Raguram.R ([raguramrramanan@gmail.com](mailto:raguramrramanan@gmail.com))

Reegan Anto.J ([reegananto21@gmail.com](mailto:reegananto21@gmail.com))

Phase 2: Innovation

Project Goal:

The goal of our project is to predict the future energy consumption based on the hourly energy consumptions from 2004 to 2018.

Approach:

First we need to investigate the given dataset; the dataset contains what appears to be various cities with the amount of electrical energy consumed alongside it.

Our initial plan was to build a linear regression model around the dataset, but after visualization we concluded that linear regression wouldn’t suit the given dataset.

Our second approach was to find any form of seasonal growth or trends within the dataset, and we seem to find some seasonal pattern within the dataset.

We concluded that univariate time series analysis would be the best method to solve the problem.

Analysis:

We then started to analyze the best algorithm within the time series analysis which would suit the given dataset perfectly.

The algorithms we compared are listed below:

* Seasonal Auto regressive Integrated Moving Average (SARIMA)
* Single Exponential Smoothing (SES)
* Prophet
* XGBoost
* LSTM

Our analysis on each machine (and deep) learning algorithms are given below:

Seasonal Auto Regressive Integrated Moving Average:

SARIMA is an improved version of ARIMA which adds seasonal moving average to the previous autoregressive model which is vital for our given time series dataset. The additional hyperparameters would aid us in predicting the measure of electrical energy.

Single Exponential Smoothing:

This technique uses the weighted sum of all values obtained below to predict the approximate future value. This technique also applies exponentially decreasing weights to prevent exploding gradient problem. The ability to control the alpha value of the problem would be useful to tune the model based on what timeline the required model should focus on thus tuning its time-accuracy combination.

Prophet:

This is an open-source library developed by meta for machine learning on the python programming language. The automated forecasting system can reduce the need for human interference but would make it hard to debug.

LSTM:

Long Short Term Memory deep learning algorithm has been rejected as we don’t have as much knowledge on memory bounded algorithms compared to other machine learning algorithms.

XGBoost:

This algorithm contains gradient boosted trees in order to quicky and efficiently calculate tabular data. The high popularity within data science contests, speed and the amount of online sources to learn the working of this algorithm has made us choose this algorithm over all others mentioned before.

Plan:

The algorithm we are going to be using here is Extreme Gradient Boosting (XGBoost).

Our plan is to split 80% of the first half of the dataset as training dataset and the remainder 20% of the dataset as testing dataset in order to train and verify the model.

Currently the result and accuracy of the model can’t be estimated but we aim to obtain at least 70% accuracy.

Conclusion:

The algorithm we are going to be using is decided to be XGBoost, the dataset will be split as per the industry standard 80% training and 20% testing and we hope to obtain an estimated average accuracy rating of at least 20% when our project has been completed.