# IBMHC Project Template

## 0) Project Scope

Problem Statement:
Predicting the energy output of wind turbine based on weather condition

Overview:

Wind power generation is increasing rapidly and the availability of wind energy depends on the wind speed, which is a random variable.This highly depends on the weather conditions at that place. In our project, we propose an intelligent technique for forecasting wind speed and power output of a wind turbine from several hours up to 72 hours ahead. We will carry out this problem on publicly available weather and energy data sets correlating and considering different features in our project. This will enable us to cut down on production costs and collaborate on different energy sources more efficiently.

Reference:
1.M. Hayashi and K. Nagasaka, "Wind speed prediction and determination of wind power output with multi-area weather data by deterministic chaos," Proceedings of the 2014 International Conference on Advanced Mechatronic Systems, Kumamoto, 2014, pp. 192-197, doi: 10.1109/ICAMechS.2014.6911649.
2.https://hpi.de/friedrich/docs/paper/RE1.pdf

Project Team: Our team consists of four people:

1. Aditya Mahajan
2. Noel Jaymon
3. Tarun Agarwal
4. Benjamin

Project Deliverables:

We are going to deliver a time series model to predict the power output of a wind farm based on the weather conditions in the site (1Hr prediction to 72Hrs. prediction).We are going to build an application with the help of django to recommend the Power Grid to suggest the best time to utilize the energy from wind farm.

Technical and Software Requirements

Plotly with Dash, Django, Deep learning model(RNN), IBM ML services, IBM Bluemix, IBM Cloudant, Python 3, MDBootstrap.

**References**:

https://hpi.de/friedrich/docs/paper/RE1.pdf
https://www.ibm.com/in-en/cloud/cloudant
https://ieeexplore.ieee.org/document/6911649

# 1) Data Collection

https://data.open-power-system-data.org/time_series/
https://www.elia.be/en/grid-data/power-generation
https://data.open-power-system-data.org/weather_data/

**References:**

https://openei.org/datasets/dataset/miller-keith-2018-windplantdata
https://data.open-power-system-data.org/weather_data/2019-04-09/weather_data.csv
http://www.sotaventogalicia.com/en/technical-area/real-time-data/historical/
https://www.nrel.gov/grid/eastern-wind-data.html

# 2) Data Preprocessing

REF - https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825
When we talk about data, we usually think of some large datasets with a huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos, etc..

- Data Quality Assessment
    a) Missing values
    b) Inconsistent values
    c) Duplicate values
- Feature Aggregation
    a) Reduction of memory consumption and processing time
    b) High-level view of the data
- Feature Sampling
    a) Sampling without Replacement
    b) Sampling with Replacement
- Dimensionality Reduction
- Feature Encoding
    a) Nominal
    b) Ordinal
    c) Interval
    d) Ratio
- Train / Validation / Test Split

# 3) Feature engineering

a) Correlation of features
b) Date Related Features
c) Time-based Features
d) Lag features
e) Rolling/Expanding Window features
f)

REF - https://www.analyticsvidhya.com/blog/2019/12/6-powerful-feature-engineering-techniques-time-series/

# 4) Building Model

a) Time Series
   i) ARIMA

Using ARIMA model, you can forecast a time series using the series past values .Depending on the frequency, a time series can be of yearly (ex: annual budget), quarterly (ex: expenses), monthly (ex: air traffic), weekly (ex: sales qty), daily (ex: weather), hourly (ex: stocks price), minutes (ex: inbound calls in a call canter) and even seconds wise (ex: web traffic).
https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

b) ML Models
   i) MLR

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and response (dependent) variable.
https://www.investopedia.com/terms/m/mlr.asp

   ii) SVM

A support vector machine (SVM) is machine learning algorithm that analyzes data for classification and regression analysis. SVM is a supervised learning method that looks at data and sorts it into one of two categories. An SVM outputs a map of the sorted data with the margins between the two as far apart as possible.
https://en.wikipedia.org/wiki/Support_vector_machine

   iii) Random Forest

The random forest is a supervised learning algorithm that randomly creates and merges multiple decision trees into one “forest.” The goal is not to rely on a single learning model, but rather a collection of decision models to improve accuracy. The primary difference between this approach and the standard decision tree algorithms is that the root nodes feature splitting nodes are generated randomly. https://deepai.org/machine-learning-glossary-and-terms/random-forest

c) Deep Learning Models
    i) Tensorflow and Keras
    Tensorflow and Keras are the framework and API that will be used for building deep learning models.
    ii) RNN (LSTM)& (GRU)

A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate** and a **forget gate**. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

GRU like LSTM is capable of learning long term dependencies.

GRU and LSTM both have a gating mechanism to regulate the flow of information like remembering the context over multiple time steps. They keep track of what information from past can be kept what can be forgotten. To achieve this GRU uses Update gate and Reset gate

https://en.wikipedia.org/wiki/Long_short-term_memory

https://medium.com/datadriveninvestor/multivariate-time-series-using-gated-recurrent-unit-gru-1039099e545a

    iii) MLP
    iv)

d) Ensamble

# 5) Building the UI

References - https://github.com/plotly/plotly.py
https://github.com/plotly/dash-sample-apps/tree/master/apps/dash-wind-streaming

The UI is basically a dashboard that will graphically present different factors affecting the power output and how they are changing and updating. As a main part of the UI, the power output prediction will also be visualized as line or scatter plot.

A. Designing the dashboard type UI
    a. Cloning the project repo

b. Understanding the working of Dash web server
c. Making a django project and designing of the application structure

B. Add visualizations for the independent factors of prediction
   a. Loading the data into the database
   b. Connecting the database with the web application
   c. Designing the graphs and pushing them on the dashboard
   d. Testing it out (unit testing)

C. Add plots for the prediction as dependent on different factors
   a. Predictions are saved in the database and regularly updating
   b. Make scatter plot / line plot for the predictions against different factors
   c. UI testing

D. Running application on localhost

# 6) Backend Part

Web app hosting - IBM Cloud Bluemix
Database for the web app - IBM Cloudant

A. Connecting the application with IBM Cloudant database
B. Deploy the application to IBM Cloud Bluemix
C. Test it working on assigned url eg. *myUrl.mybluemix.net*.

References - https://github.com/plotly/dash-on-bluemix

# 7) Ml Pipelines

REF - https://developer.ibm.com/technologies/artificial-intelligence/tutorials/deploy-a-python-machine-learning-model-as-a-web-service/

a) Preparing the model for deployment
b) Pickling the model to a file
c) Creating Training Scripts for new data