**SYNOPSIS**

* The dataset contains 2 sections: Actuals and Forecast.
* In the actuals, there are 34 variables combined (weather\_actuals & power\_actual), out of which there is 1 pair of duplicate variable (datetime) on which we have merged the two tables.
* “Power” is the dependent variable (label) and the rest are independent variables (features).
* Since there is a dependent variable, this is a supervised learning problem.
* The dependent variable is continuous, hence we can use regression techniques to solve the problem.
* 10 variables have more than 50% missing values in it, out of which we have dropped 9 and imputed values in 1.
* The variables "sunrise" & "sunset" have dates and times, out of which only time seems to be relevant information for our model. Hence we’ll extract hours and minutes, and drop "sunrise" & "sunset".
* There seems to be a data entry error in most of the columns of the weather\_actual as "-9999", which does not seem to be a valid entry relative to the other entries of the columns. Hence we need to replace "-9999" with some relevant data.
* The method of imputation can be found out in the code.
* In the power\_actual table, drop "ghi" & "gti" as it contains 50% missing values and 50% "0", and hence we cannot extract relevant information from it.
* We'll use inner join to merge the tables (weather\_actual & power\_actual) because power has datetime values for every 15 minute and weather\_actual has values for every hour. By using inner join we'll get values for every hour.
* We'll divide the numerical & categorical columns, and apply scaling and label encoding on them respectively
* After completing all the data preprocessing, we’ll apply train test split on the “power\_actual” data to build our model.
* We’ll build 3 different regressors and check the cross validation score (R2 score) to select our model.
* XGBoost gives us the best results out of the 3 models, and hence we’ll use XGBoost.
* We'll use Bayesian optimization for hyperparameter tuning, to improve the performance of our model. We can also use GridSearach, but it requires larger computational power.
* Now, we’ll fit the optimized parameters to our model and fit the model to get the predictions.
* The predicted power is saved as a csv file named power\_forecast.