**ReNew Power Hiring Hackathon**

The project’s objective was to predict the rotor-bearing temperature of wind turbines from the yearly data. Following are the steps with a brief explanation that were followed in the submission notebook,

* Imported training dataset
* Created new features relevant to wind energy systems:
  + Apparent power: hypotenuses to active and reactive power
  + Power difference between active and average power: useful to detect surges
  + Power from wind speed:
  + Frictional factor: ∝
  + Difference between raw and convertor calculated power
  + Temperature difference between inside and outside nacelle temperature
* Anomaly detection and removal was performed using PCA transformation:
  + Divided df based on ‘turbine\_id’ and separately saved the dependent/independent variables
  + PCA transformation of independent variables with same number of features as original data
  + Reconstructed original data using transformed features from the previous step
  + Used z-score to eliminate data with an extreme reconstruction error
* Visualized features separately for each turbine:
  + Turbines behave differently and scaling all as same data will undermine values from the turbine with smaller values. Scaled each turbine independently and later concatenated all together.
  + Many features depend on season/weather (different values in different months). **As timestamps can’t be used to extract months, clustering was used to separate these different data points**
* Clustering based on climate/weather features to segregate data into two clusters (most likely hot and cold seasons):
  + Scaled the data before clustering to avoid feature dominance
  + k-mean and hdbscan clustering were used based on seasonal features from the visualization
  + For each turbine, data were clustered into two labels, which were later used as a new feature for modelling
* Other performed data transformations include:
  + Skewness correction with PowerTransformer (yeo-johnson transformation)
  + Concatenation of different turbines data into single df
  + Outlier removal based on visualization wrt. target column
  + OHE for turbine\_id
  + Feature selection:
    - Many columns were highly correlated (> 0.9). Dropped columns with multi-collinearity
    - Dropped columns which don’t impact the model by hit & trail
* Split data into train-test with stratification on turbine\_ids (train-test should have proportioned data of each turbine)
* Model selection:
  + Different models such as KNN, LinearRegression, RandomForest, XGBoost, Catboost were trained on train split of the data with default parameters
  + A simplistic vanilla neural network was also trained with a fixed learning rate and SGD optimizer
  + Models evaluated based on MAPE and R2 scores on test split of the data
  + Neural network outperformed other models and was chosen for optimization
  + Developed Tensorflow sequential neural network
    - Regularization techniques such as dropout and batch-normalization were employed to avoid overfitting
* Defined learning rate schedule, optimizer and loss function:
  + For continuation of the learning process, decaying learning rate schedule was fed instead of a constant value
  + Different optimizers such as RMSProp, Adam and SGD were tried; Adam outperformed the others
  + Error functions such as MSE, MAE, MAPE and MSLE were used; Optimization on MAPE gave the best score for the competition
* The model was trained on complete data with 600 batch size and 1000 epochs
* Losses and evaluation metrics wrt. epochs were plotted for the inference
* Trained model and weights saved as JSON and h5 files, respectively
* Loaded saved model for prediction on the test dataset
* Submission:
  + Repeated the same transformations on the test df
  + Prediction from the loaded model
  + Saved the submission file
  + Verified the predicted data