Advanced Data Science Capstone Project Report

Project: Predicting Power Output of a Combined Cycle Power Plant (From Regression to Deep Learning) Shahbaz Masih

1. Introduction

1.1 Business Understanding and Background

Combined cycle power plant (CCPP) is a kind of power plant which is composed of gas turbines, steam turbines and heat recovery steam generator (Figure 1 below). In a CCPP, the electricity is generated by gas and steam turbines, which are combined in one cycle. A CCPP can generate up to 50 percent more electricity from the same fuel than a traditional simple-cycle plant by routing the waste heat from the gas turbine to the nearby steam turbine, which generates extra power. A CCPP mechanism can be described as below.

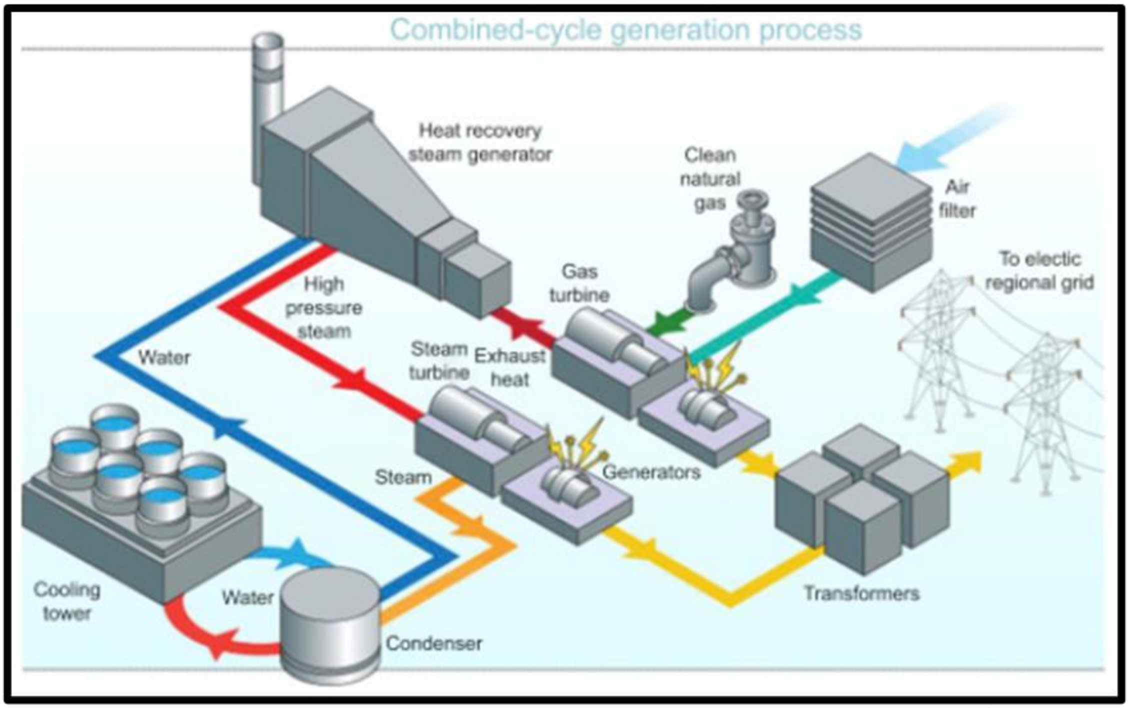


Figure 1 Combined cycle Power Plant Diagram (Source: Brun et al[1])

1) Fuel burns at the gas turbine, makes the turbine blades spinning and driving electricity generators.
2) Heat Recovery Steam Generator (HRSG) captures exhaust heat from the gas turbine. The HRSG creates steam from the gas turbine exhaust heat and delivers it to the steam turbine.
3) Steam turbine uses the steam delivered by the heat recovery system to generate additional electricity by driving an electricity generator.

Gas turbine load is sensitive to the ambient conditions e.g.

- Ambient temperature (AT)

- Atmospheric pressure (AP), and
- Relative humidity (RH).

However, steam turbine load is sensitive to the

- Exhaust steam pressure (or vacuum, V).

Here is an real-world example of demand forecast and actual demand for power on hourly time scale within a day (from available resources from the California power grid: http://www.caiso.com/Pages/TodaysOutlook.aspx), shown in Figure 1.

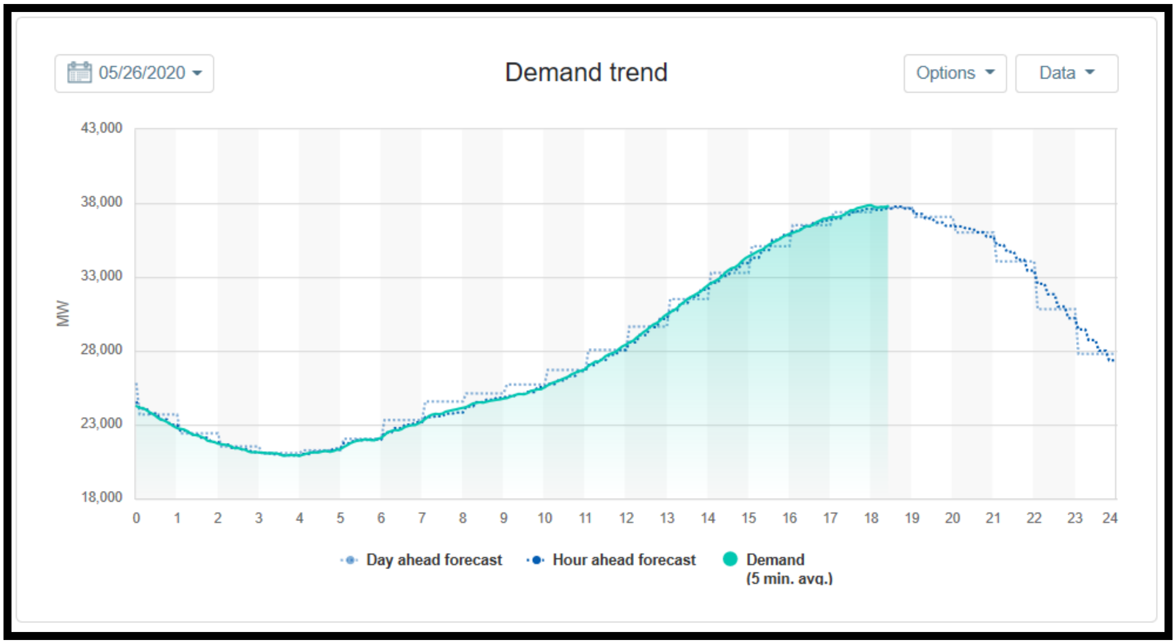


Figure 1 Power forecast and actual demand

Also, power is supplied by the grid from multiple resources to meet the demand as shown in Figure 2. And contribution from each source changes with time (also shown in figure 3).

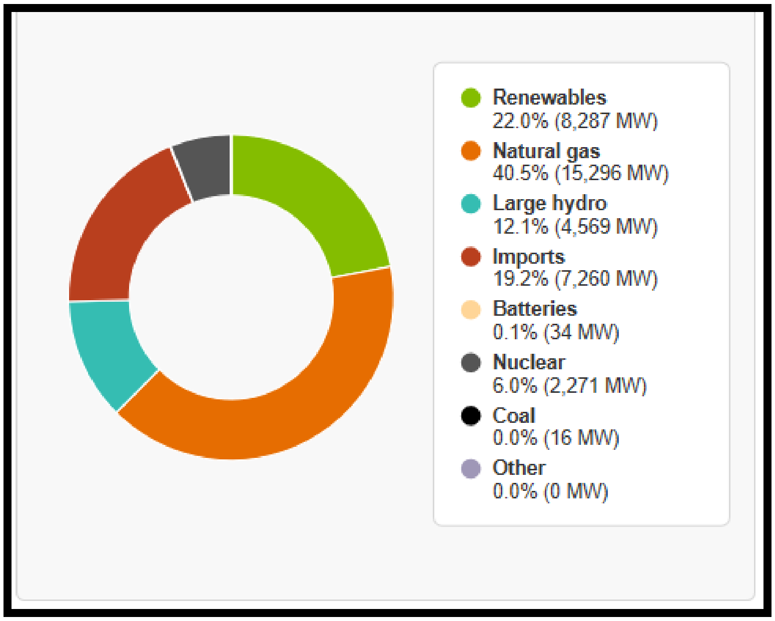


Figure 2 Multiple power resources

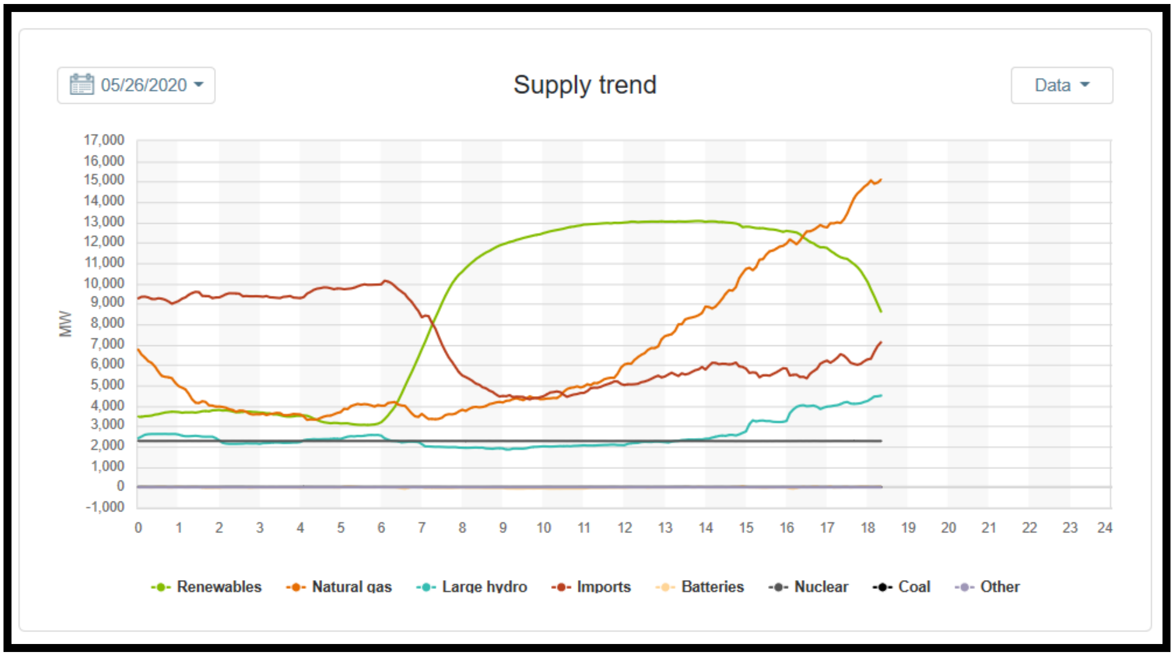


Figure 3 Contribution of Power Resources

Power generation is a complex process. Understanding and predicting power output is an important element in managing a plant and its contribution to the power grid. The operators of a regional power grid forecast power demand based on historical data. Then, they compare the forecast against power supply from available resources (e.g. coal, natural gas, nuclear, solar, wind, hydro power plants, CCPP etc.). Power generation technologies e.g. solar and wind are highly dependent on environmental conditions. Also, power generation technologies are subject to planned and unplanned maintenances.

1.2 Business Problem and Use Case

The challenge for a power grid operator is to manage a shortfall in available resources versus actual demand for power. Therefore, it is extremely vital to forecast power supply from available resources. Predicting the electricity generated hourly based on ambient variables enables to evaluate whether the generated power will be sufficient to meet the growing consumer demands. Proactive steps to address the demands can be taken if the forecasted power is found to be insufficient e.g. build more base load power plants (this process can take many years to decades of planning and construction), buy and import power from other regional power grids (this choice can be very expensive), or turn on small Peaker or Peaking Power Plants.

In this project, power generated by a CCPP will be forecasted and its accuracy will be determined based on data of ambient conditions e.g. ambient temperature (AT), atmospheric pressure (AP), relative humidity (RH) and exhaust steam pressure (or vacuum, V).

This forecast will help

1) Better management of power supply to meet the consumer demand
2) Take the most viable option economically to meet the consumer demand e.g. build more power plants, buy power from other grids or turn on peakers
3) Better management of power shortfalls from other sources of power supply due to planned and unplanned maintenances and inconsistent environmental conditions in case of solar and wind plants.

The ambient conditions data will be used to develop a correlation to forecast power output of CCPP by using regression models and neural network. The correlation will be used by the power grid to forecast power output from a CCPP. The forecasted power from a CCPP (with the help of regression machine learning models) will be used to manage power supply and power shortfalls from other sources and to make decision about the most viable option to meet consumer demand. The correlation will be handed over to the management of power plant in the form of this report.

2. Data

2.1 Required Data

As described in section 1.1, a CCPP is composed of gas turbines (GT), steam turbines (ST) and heat recovery steam generators (HRSG). Gas turbine load is sensitive to the ambient conditions e.g. ambient temperature (AT), atmospheric pressure (AP), and relative humidity (RH). However, steam turbine load is sensitive to the exhaust steam pressure (or vacuum, V). Therefore, ambient temperature (AT), atmospheric pressure (AP), and relative humidity (RH), exhaust steam pressure (or vacuum, V) and power output of the plant (PE) are required to forecast PE by using regression and neural network models.

2.2 Data Sources

The data was retrieved from UCI machine learning repository and loaded on IBM cloud object storage. The link is given below.
https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant
The schema definition from UCI repository is given below:
AT = Atmospheric Temperature in C
V = Exhaust Vacuum Pressure
AP = Atmospheric Pressure
RH = Relative Humidity
PE = Power Output. This is the value we are trying to forecast from the parameters given above

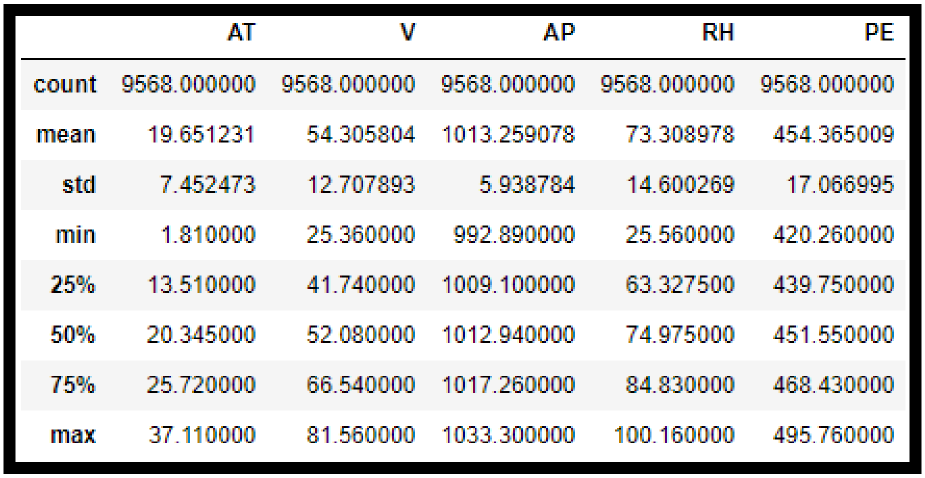
3. Methodology

3.1 Exploratory data analysis

Total of 9568 records were found. First to all data was looked for basic statistics functions like minimum, maximum, quartile, standard deviation, and mean values of each column. The results are shown in Table 1.

The result shows that relative humidity and plant power output have the highest standard deviation. Relative humidity depends upon ambient pressure and temperature and, therefore, has standard deviation (more widely spread out) than both temperature and pressure individually. The high standard deviation of power out also shows its high variability indicating the impact of variation of ambient conditions on power output. However, maximum value is only around 10% higher than the mean value unlike relative humidity, exhaust vacuum and ambient temperature. Atmospheric pressure has low standard deviation and indicates low variability because atmospheric pressure does not change much generally.

Next, box plots were plotted (Figure 5) to see data distribution and identify outliers and the range of values for outliers. Box plots show that both atmospheric pressure and relative humidity have outliers. Outliers in the relative humidity lie in 20-30% range approximately while outliers in atmospheric pressure less than 997 and greater than 1030. Atmospheric pressure has almost normal distribution (zero skewness) while other parameters have visible skewness.



	AT	V	AP	RH	PE
count	9568.000000	9568.000000	9568.000000	9568.000000	9568.000000
mean	19.651231	54.305804	1013.259078	73.308978	454.365009
std	7.452473	12.707893	5.938784	14.600269	17.066995
min	1.810000	25.360000	992.890000	25.560000	420.260000
25%	13.510000	41.740000	1009.100000	63.327500	439.750000
50%	20.345000	52.080000	1012.940000	74.975000	451.550000
75%	25.720000	66.540000	1017.260000	84.830000	468.430000
max	37.110000	81.560000	1033.300000	100.160000	495.760000

Table 1 Data statistics

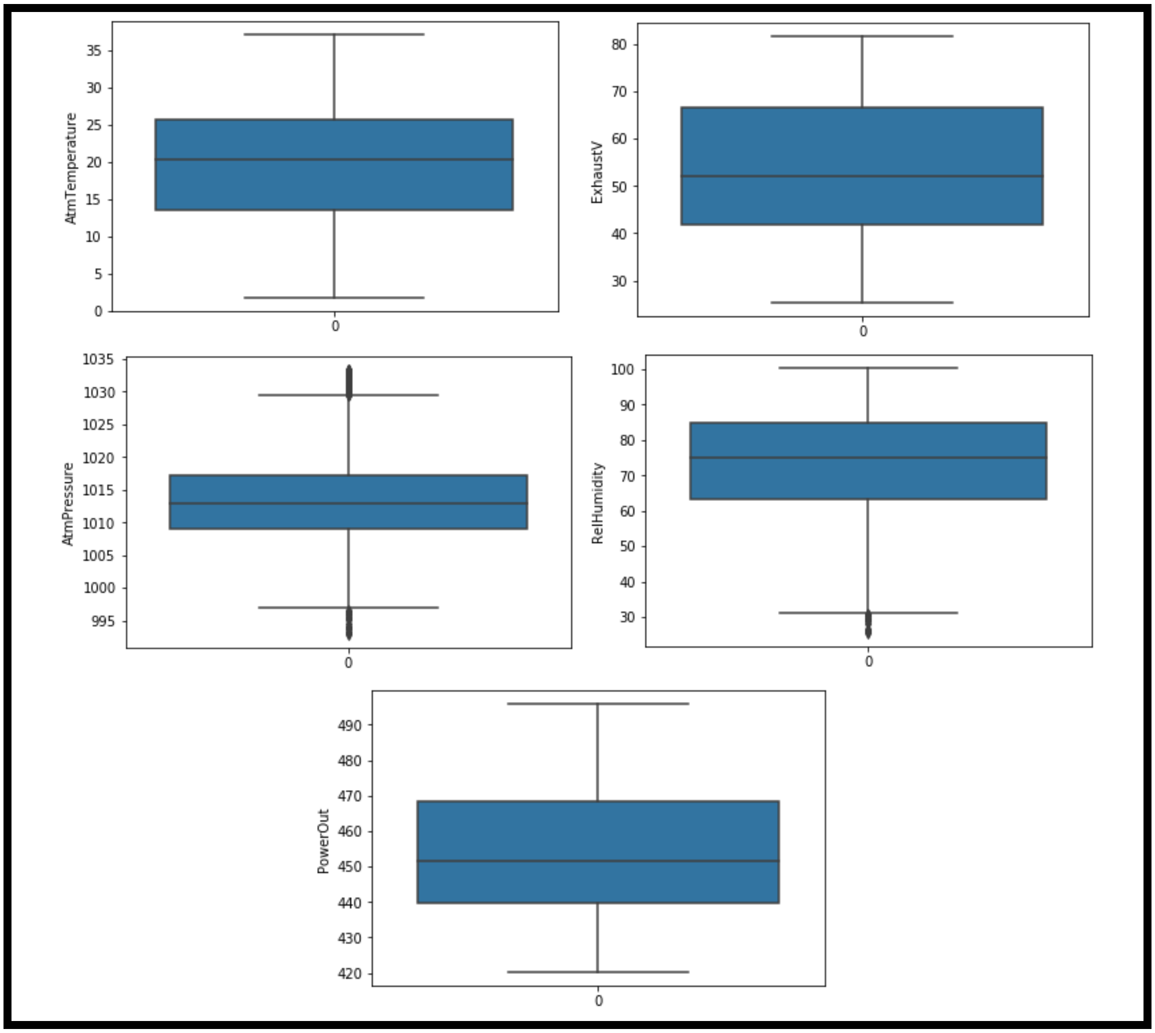


Figure 5 Box plots for all variables

Scatter plots helps to determine data scatter along different intervals and correlation type and strength between plotted variables. This strength helps determine features in feature engineering. In Figure 6, scatter plots were plotted to see relationship between dependent variable (Power output) and independent variables. Figure 6 shows that ambient temperature and exhaust vacuum have a linear and stronger relationship than atmospheric pressure and relative humidity. Also, atmospheric temperature affects exhaust vacuum pressure considerably.

In order to investigate correlation among different variables further, correlation coefficients (R) were calculated and correlation matrix was plotted (Figure 7). Figure 7 shows that Ambient temperature and exhaust vacuum are strongly linearly related to each other and to the power output, while Atmospheric pressure and relative humidity have weak linear relations to all other variables and output. Therefore, ambient temperature and exhaust vacuum pressure are considered governing variables. The effect of absence and presence of each of variable on regression modeling can be studied by feature engineering.

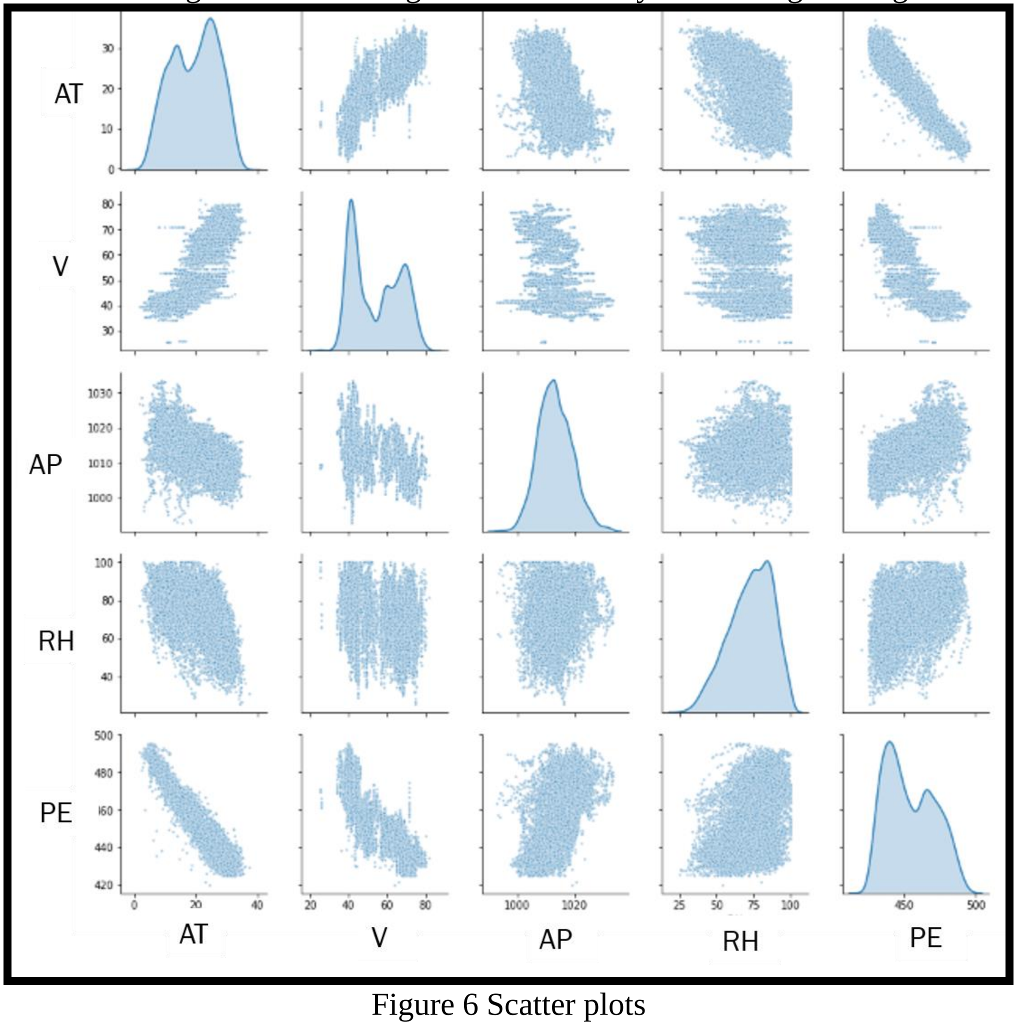


Figure 6 Scatter plots

3.2 Data Cleaning/Wrangling and Feature Engineering

The data was first loaded to IBM cloud storage object from online repository and a pandas data frame was created. Then column names were changed to be more representative of the data e.g. “AT” was changed to “AtmTemperature”, “V” to “ExhaustV”, “AP” to

“AtmPressure”, “RH” to “RelHumidity” and “PE” to “PowerOut”. To remove "nan" or missing/null values, first of "nan" values in each column. No null or missing values were found. Then the data set was checked for any “zero” values in all the columns and no zero values were found.

Since humidity is given as a percent relative humidity. A check was made if there are any values more than 100% present in the dataset. 55 values were found to be more than 100 %. Before the removal of the 55 records containing relative humidity greater than 100 percent, it was considered prudent to check if relative humidity can exceed 100 %. Relative humidity is defined as “the amount of water vapor present in air expressed as a percentage of the amount needed for saturation at the same temperature”. At any given temperature and air pressure, a specific maximum amount of water vapor in the air will produce a relative humidity of 100 percent. Supersaturated air literally contains more water vapor than is needed to cause saturation and can have relative humidity of more than 100%. Therefore, relative humidity values of greater than 100% were not removed.

Exploratory data analysis showed the presence of outliers in the data (Figure 5). Z-Score and IQR-Score methods were used to remove outliers from data.

The **Z-score** is the number of standard deviations away from the mean for a particular data point. The formula for calculating a z-score is z = (x-μ)/σ, where x is the data point, μ is the population mean, and σ is the population standard deviation. The points that are too far away from the mean are considered outliers. In most of the cases a threshold of 3 or -3 is used i.e. if the Z-score value is greater than or less than 3 or -3 respectively, that data point will be considered an outlier. After filtering data with Z-score threshold of between 3 and -3, the number of records were reduced to 9510 (58 records were filtered out). The number of outliers were reduced as shown by box plots for Atmospheric pressure and relative humidity in Figure 8.

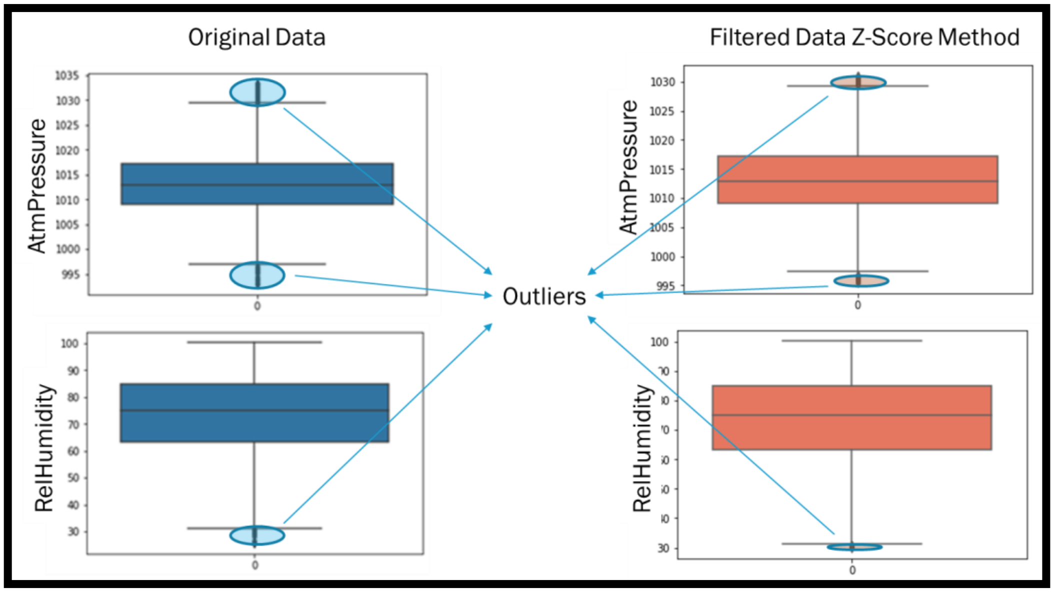


Figure 8 Reduced number of outliers after filtering data with Z-Score method

The interquartile range (IQR), sometimes also referred to as midspread or middle 50%, or technically H-spread, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, IQR = Q3 − Q1. The formula for IQR-Score is defined by the data range lying between (Q1 - 1.5 * IQR) and

(Q3 + 1.5 * IQR). The data points lying outside this range are considered outliers. After applying IQR-Score filter, the dataset was reduced to 9468 records and 100 records were filtered out. Figure 9 shows that IQR-score method filtered all outliers from Relative humidity data and more outliers from atmospheric pressure data than filtered by Z-method. Therefore, dataset filtered with IQR-Score method was chosen for modeling purposes.

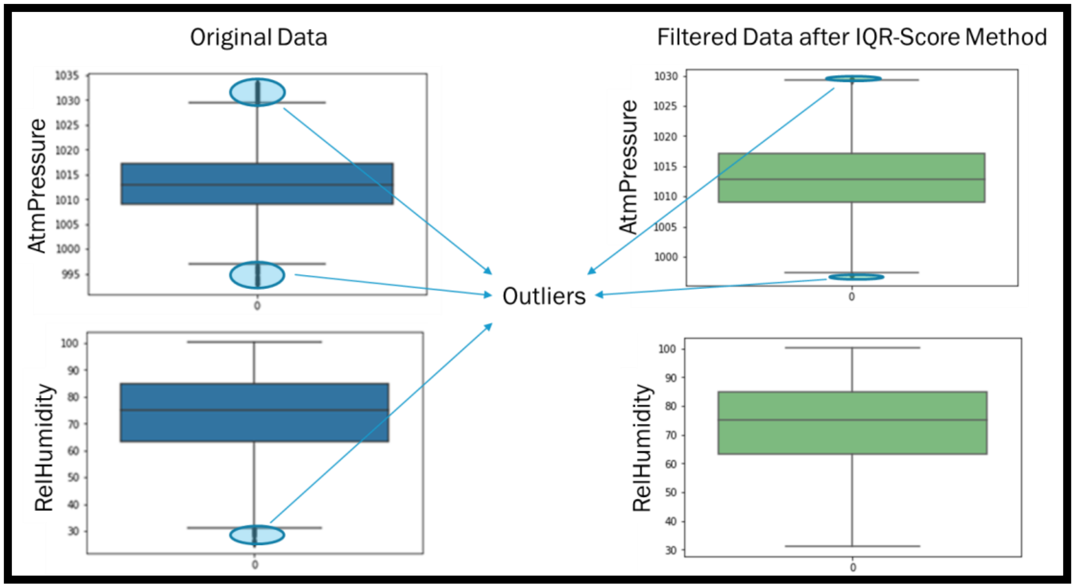


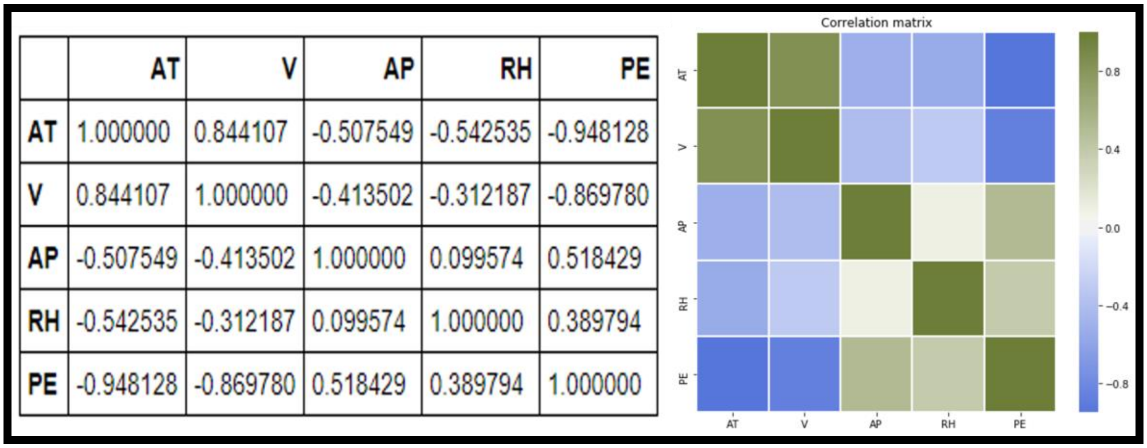
Figure 9 Reduced number of outliers after filtering data with IQR-Score method

The data filtered by IQR-Score method was split into training and testing data and loaded onto Apache spark.

3.3 Feature Engineering

In machine learning, a feature is an individual measurable property or characteristic of a phenomenon being observed. Choosing informative, discriminating, and independent features is a key step for writing effective algorithms in classification and regression. As explained in section 3.2, In Figure 6, scatter plots were plotted to see relationship between dependent variable (Power output) and independent variables. Figure 6 shows that ambient temperature and exhaust vacuum have a linear and stronger relationship than atmospheric pressure and relative humidity. Also, atmospheric temperature affects exhaust vacuum pressure considerably.

To investigate correlation among different variables further, correlation coefficients (R) were calculated and correlation matrix was plotted (Figure 7). Figure 7 shows that Ambient temperature and exhaust vacuum are strongly linearly related to each other and to the power output, while Atmospheric pressure and relative humidity have weak linear relations to all other variables and output. Therefore, ambient temperature and exhaust vacuum pressure are considered governing variables. The atmospheric pressure correlates stronger with the plant power output that also defines an features variation. The effect of absence and presence of each of variable on regression modeling can be studied by feature engineering. Therefore models (or pipelines for models) were created with all four independent variables as input features, 3 independent variables (ambient temperature, exhaust vacuum pressure, and atmospheric pressure) as input features and two independent variables (ambient temperature and exhaust vacuum pressure) as input features. The process of building pipelines will be discussed in more detail in the modeling section.



	AT	V	AP	RH	PE
AT	1.000000	0.844107	-0.507549	-0.542535	-0.948128
V	0.844107	1.000000	-0.413502	-0.312187	-0.869780
AP	-0.507549	-0.413502	1.000000	0.099574	0.518429
RH	-0.542535	-0.312187	0.099574	1.000000	0.389794
PE	-0.948128	-0.869780	0.518429	0.389794	1.000000

Figure 7 Correlation coefficients

3.4 Modeling and Evaluation

3.4.1 Linear Regression Model

Pyspark ML (machine learning library built by collaboration between apache spark and python) was used to build linear regression models and tune hyperparameters.

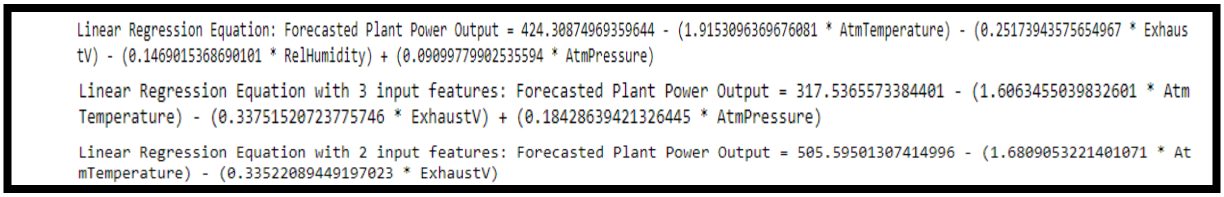
The process of building linear regression models consisted of three main steps.

a) Building a pipeline

A Pipeline is used to facilitate the creation, tuning, and inspection of practical machine learning (ML) workflows. After extracting data from cloud storage object and transforming it, it was loaded onto apache spark as mentioned in the previous steps. For building a pipeline, first of all, Vector Assembler, a transformer that combines a given list of columns into a single vector column, was used to create a features column by combining atmospheric temperature, exhaust vacuum pressure, atmospheric pressure and relative humidity. It is useful for combining raw features and features generated by different feature transformers into a single feature vector, to train ML models like logistic regression and decision trees. Then, Linear Regression model was built by passing label column (i.e. power output), features column (column created by vector assembler) and prediction column (forecast for power output) and setting other regression model parameters e.g. maximum number of iterations, Regularization parameter etc. The fixed regularization parameter defines the trade-off between the two goals of minimizing the loss (i.e., training error) and minimizing model complexity (i.e., to avoid overfitting). Then vectorizer and the model were passed to the pipeline.

b) Model training and evaluation

In this step, model was trained by using spark training data frame. Then, the trained model was passed testing data set and new data set was created including forecasted power output parameter. The number of iterations and regularization parameters were optimized. And R^2 was calculated to evaluate the model. Also, in order to hand in client the correlation to forecast future power output of the CCPP, weights/coefficients and intercept from the model were exported and a linear regression equation was built (with 4, 3 and 2 input features respectively) as shown below.



```
Linear Regression Equation: Forecasted Plant Power Output = 424.30874969359644 - (1.9153096369676081 * AtmTemperature) - (0.25173943575654967 * Exhaus
tV) - (0.1469015368690101 * RelHumidity) + (0.09099779902535594 * AtmPressure)

Linear Regression Equation with 3 input features: Forecasted Plant Power Output = 317.5365573384401 - (1.6063455039832601 * Atm
Temperature) - (0.33751520723775746 * ExhaustV) + (0.18428639421326445 * AtmPressure)

Linear Regression Equation with 2 input features: Forecasted Plant Power Output = 505.59501307414996 - (1.6809053221401071 * At
mTemperature) - (0.33522089449197023 * ExhaustV)
```

Similarly, correlations were built after each set of models were run after hyper parameter tuning and cross validation.

c) Hyper Parameter Tuning

A model hyperparameter is defined as a configuration parameter that is external to the model and whose value cannot be estimated from data. Or any parameter that is specified manually is a model hyperparameter e.g. learning rate is a hyper parameter.

ML library supports model selection or hyperparameter tuning by using tools such cross validator. These tools require the following items:

1. Estimator: algorithm or Pipeline to tune
2. Set of ParamMaps: parameters to choose from, sometimes called a “parameter grid” to search over
3. Evaluator (a metric to measure how well a fitted Model does on held-out test data)

Cross Validator begins by splitting the dataset into a set of folds which are used as separate training and test datasets. For example, with 3 number of folds, Cross Validator generates 3 (training, test) dataset pairs, each of which uses 2/3 of the data for training and 1/3 for testing. In order to evaluate a particular ParamMap, Cross Validator computes the average evaluation metric for the 3 Models produced by fitting the Estimator on the 3 different (training, test) dataset pairs.

After hyperparameter tuning, the correlation or equation for estimating plant power output was derived again. Then, as mentioned in the feature engineering section, input features were reduced to 3 and 2, linear regression models were built, hyperparameter tuning done and new correlations to forecast plant power output were derived based on learned models. The results of linear regression models before and after hyperparameter tuning by using grid search method have been shown in Table 1.

3.4.2 Random Forest Regressor

Random Forests are ensembles of decision trees. Random forests combine many decision trees to reduce the risk of overfitting. Random forests train a set of decision trees separately, so the training can be done in parallel. The process of building random forest regression models consisted of three main steps.

a) Building a pipeline

The same vector assemblers were used as in linear regression with 4, 3 and 2 input features. Then, random forest regression model was built by passing label column (i.e. power output), features column (column created by vector assembler) and prediction column (forecast for power output) and setting other regression model parameters e.g. randomness, number of trees, maximum depth etc.

The algorithm injects randomness into the training process so that each decision tree is a bit different. The two most import parameters whose tuning them can often improve performance are

1. numTrees: Number of trees in the forest. Increasing the number of trees will decrease the variance in predictions, improving the model’s test-time accuracy. Training time increases approximately linearly in the number of trees.

2. maxDepth: Maximum depth of each tree in the forest. Increasing the depth makes the model more expressive and powerful. However, deep trees take longer to train and are also more prone to overfitting.

Then vectorizer and the model were passed to the pipelines.

b) Hyper Parameter Tuning

PySpark ML library supports model selection or hyperparameter tuning by using cross validator. It was passed pipeline, parameter grid and the model as explained in section 3.4.1.

c) Model training and evaluation

In this step, model was trained by using spark training data frame. Then, the trained model was passed testing data set and new data set was created including forecasted power output parameter. The number of trees and maximum depth were optimized. And R^2 was calculated to evaluate the model. Models were run 4, 3 and 2 input features as in case of previous model.

3.4.3 Decision Tree Regressor

The implementation of decision trees partitions data by rows, allowing distributed training with millions of instances. A decision tree predicts the same label for each bottommost (leaf) partition. Each partition is chosen in an aggressive manner by selecting the best split from a set of possible splits, in order to maximize the information gain at a tree node. The process of building decision tree regression models consisted of three main steps.

a) Building a pipeline

The same vector assemblers were used as in linear regression with 4, 3 and 2 input features. Then, decision tree regression model was built by passing label column (i.e. power output), features column (column created by vector assembler) and prediction column (forecast for power output) and setting other regression model parameters e.g. randomness, number of bins etc. The parameter of maxBins was tuned.

- Increasing maxBins lets the model to consider more split candidates and make fine-grained split decisions. However, it increases computation and communication.

b) Hyper Parameter Tuning

Cross validator was passed pipeline, parameter grid and the model as explained in section 3.4.1 and 3.4.2.

c) Model training and evaluation

In this step, model was trained by using spark training data frame. Then, the trained model was passed testing data set and new data set was created including forecasted power output parameter. The number of maximum bins was tuned or optimized. And R^2 was calculated to evaluate the model. Models were run 4, 3 and 2 input features as in case of previous models.

3.4.4 Deep Learning Model-Neural Network

Neural network has been chosen as a deep learning model. A neural network or artificial neural network (ANN) is the basic building block of deep learning. It consists of layers of sigmoid neurons stacked together to form a bigger architecture. The network consists of 3 types of layers, an input layer, an output layer, and hidden layers. Each neuron has its own weight values. The first layer(input) takes the independent variable of data as input. Output layer predicts the class. The number of hidden layers and number of neurons in hidden layers is not fixed and are selected to get the best results. Neural networks are the weighted sum of the inputs and the learning of neural networks is based on updating these weights. A loss function (MSE, Mean absolute Percentage Error etc.) measures the performance of the network. Once the loss is calculated, a method (e.g. gradient descent) is required to change the weights of the neural network with respect to the calculated loss.

Neural network has advantages over regression due to automated feature engineering (neural net work itself comes up with the best combination of weights and features) and non-linear features to take care of non-linearity of the input data.

Figure 10 shows Keras regression architecture. The input to the network is a datapoint including atmospheric temperature, exhaust vacuum pressure, atmospheric pressure and relative humidity. It has three hidden layers with 8 neurons, 6 neurons and 4 neurons respectively. The output of the network is a single neuron with a linear activation function. The linear activation allows the neuron to output the predicted value of energy output of CCPP.

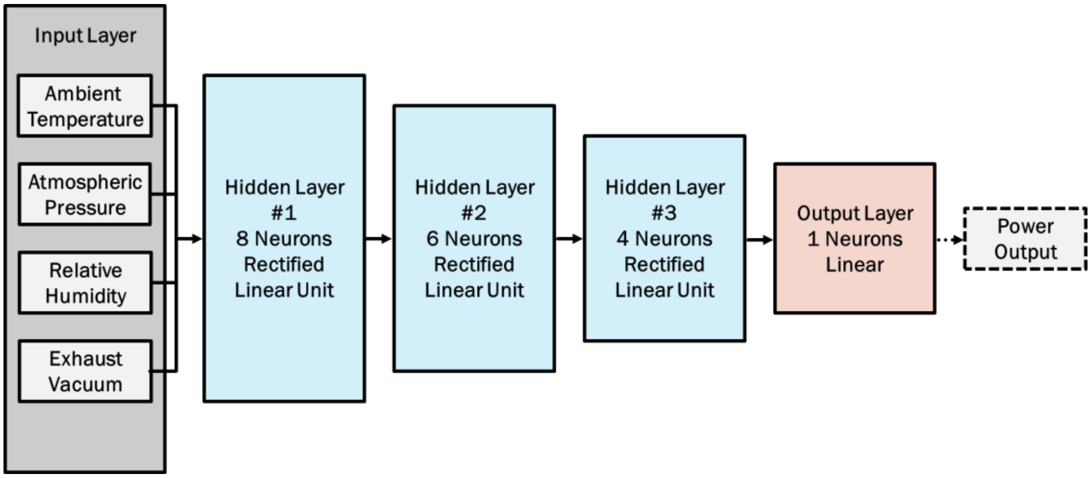

Input Layer
Ambient Temperature
Atmospheric Pressure
Relative Humidity
Exhaust Vacuum
Hidden Layer #1 8 Neurons Rectified Linear Unit
Hidden Layer #2 6 Neurons Rectified Linear Unit
Hidden Layer #3 4 Neurons Rectified Linear Unit
Output Layer 1 Neurons Linear
Power Output

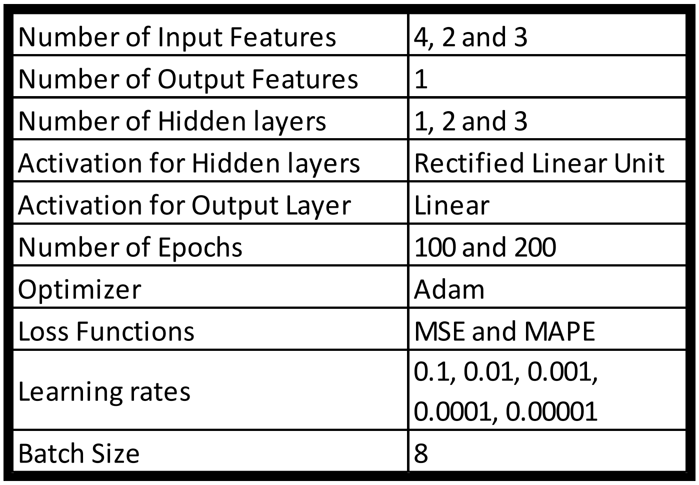
Figure 10 Architecture of Neural Network for Regression

The number of input features was set at 4 (i.e. all input features were taken) but the results (shown in Table 2) were not as promising as by linear regression. As already explained in feature engineering section, the number of input parameters for neural network was also set to 2 and 3. Since only plant power output is being predicted by the model, the number of output features is only one. The learning rate was set at 0.001 in adam optimizer. The batch size divides our data into batches of equal size. The batch size number of samples are loaded into memory and processed. Once one batch is done, it is flushed from memory and the next batch is processed. The batch size was set equal to 8. In terms of artificial neural networks, an epoch refers to one cycle through the full training dataset. Usually, training a neural network takes more than a few epochs. The number of epochs was set to 100. Since desired results were not achieved by neural network, different sensitivities were on e.g.

1. Number of input features (4, 2 and 3)
2. Type of loss function (MSE and MAPE) and
3. Number of hidden layers (1, 2 and 3)
4. Learning rate

If a model had only one hidden layer, first layer with 8 neurons was used. If a model had two layers, 1st layer (with 8 neurons) and 2nd layer (with 6 neurons) were used. And If a model had three hidden layers, 1st layer (with 8 neurons), 2nd layer (with 6 neurons) and 3rd hidden layer (with 4 neurons) were used as shown in Figure 10. The results after running these models have shown in Table 2. R^2 was chosen as the model evaluation parameter because it is a regression problem and R^2 is considered the ultimate parameter to evaluate any regression problem. The closer the value of R^2 to one, the better the model performance.

Table 1 Neural Network Details

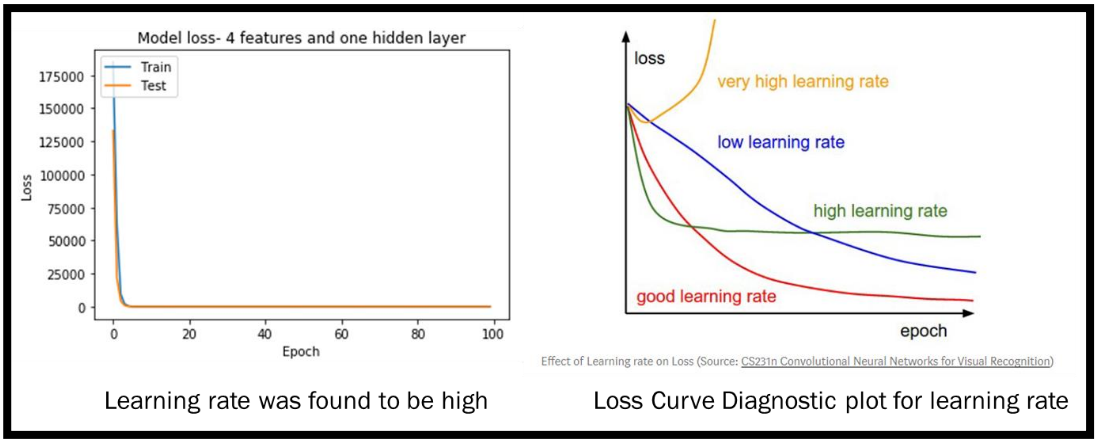


Number of Input Features	4, 2 and 3
Number of Output Features	1
Number of Hidden layers	1, 2 and 3
Activation for Hidden layers	Rectified Linear Unit
Activation for Output Layer	Linear
Number of Epochs	100 and 200
Optimizer	Adam
Loss Functions	MSE and MAPE
Learning rates	0.1, 0.01, 0.001, 0.0001, 0.00001
Batch Size	8

3.4.4.1 Diagnostics of Neural Networks during training

Training Neural Networks is a challenging task and can produce results that are sometimes far better than an expectation or perform far worse and produce just noise. A deep learning model consists of a lot of learnable parameters. Analysing how each parameter changes during training and how one parameter affects others is an impossible task. Fortunately, there are tools to make diagnosis if model training is moving in the right direction. For example, loss curve, uncertainty curve, accuracy curve etc. Loss curve will be discussed because it gives us an insight into learning rate, one of the most important parameters of a neural net work.

The learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function or higher accuracy. In simple words, the learning rate controls how quickly the model is adapted to the problem. If a learning rate is too high, optimum loss or accuracy might never be achieved. If learning rate is too low, model will take a long time to run and it makes it difficult to converge at times as shown in Figure 12. The loss curve is one of the most used plots to debug a neural network during model training. It gives us a snapshot of the training process and the direction in which the network learns. Figure 12 (Right hand side) explains how loss plotted against epoch can give an indication of learning rate. Also, Figure 12(LHS shows that learning rate of 0.001 (default for Adam optimizer used in the model) is high, but not too high. Therefore, training models were set up with different values (0.1, 0.01, 0.001, 0.0001 and 0.00001) of learning rate by custom setting. Figure 13 shows that for models set up with 4 all four features and single hidden layer, learning rates of 0.1 and 0.01 were too high due to the sharp decline shape of the loss curve. Also, learning rate of 0.00001 was found to be too low due to the time it is taking for convergence. Learning rates of 0.001 and 0.0001 both showed similar convergence. Also, Table 2 shows the best R^2 with learning rate of 0.0001. The learning rate of 0.001 is set up as a default for Adam optimizer in Kera. Therefore, training models were set up with learning rate of 0.0001 to compare the results with cases run with learning rate of 0.001. Figure 14 (enlarged view) shows that model trained with learning rates of 0.0001 for 2 and 3 hidden layers ended up at better loss (MSE) which is evident by R^2 numbers in Table 2.



Effect of Learning rate on Loss (Source: CS231n Convolutional Neural Networks for Visual Recognition)

Learning rate was found to be high

Loss Curve Diagnostic plot for learning rate

Figure 12 Learning rate diagnosis from the loss curve

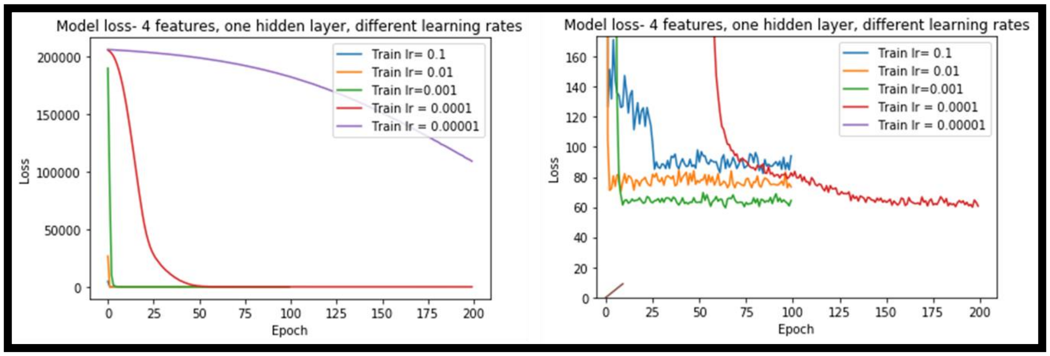


Figure 13 Model loss (MSE) at different learning rates (Enlarged view on the right)

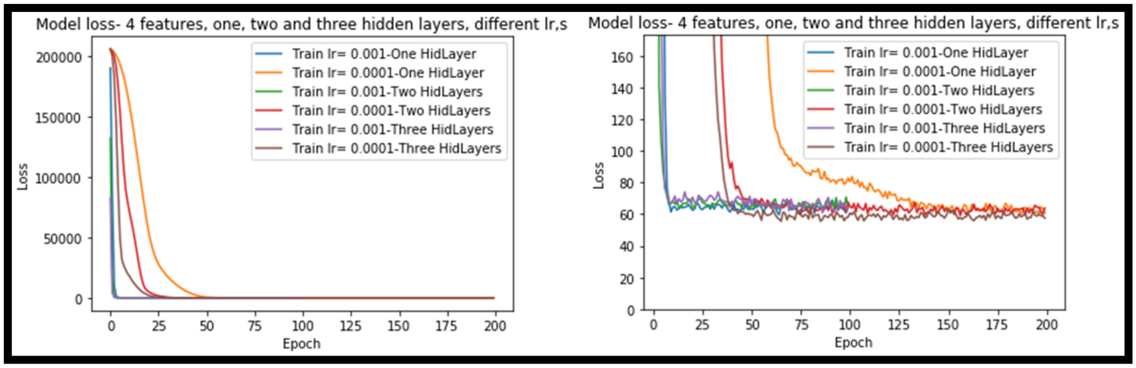


Figure 14 Comparison of learning rates of 0.001 and 0.0001(Enlarged view on the right)

3.4.5 Selection and justification of Model Performance Indicator

Since objective of the project is to predict or forecast plant power output based on environmental conditions, it is a regression problem. To solve regression problems, a fitted line is drawn or calculated by using algorithms. The better the data scatter around the fitted line, the better the regression model is. R-squared calculates the scatter of the data points around the fitted regression line. It is also called the coefficient of determination. For the same data set, higher R-squared (R^2) values represent smaller differences between the observed data and the fitted values as shown in Figure 11. Therefore, the higher (closer to 1) the value of R^2, the better the correlation or the model except a few cases where even low values of R^2 are considered good e.g. predicting human behaviour because human behaviour is very difficult to predict. R2 is calculated by the equation below.

$$(R)^2 = \frac{Variance\ Explained\ by\ the\ Model}{Total\ Variance}$$

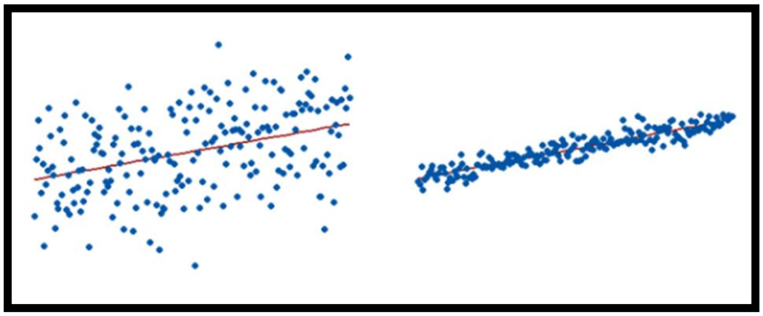


Figure 11 R^2 with values of 0.15 (left) and 0.85 (right)[4]

4. Results & Discussion

4.1 Regression

As discussed in feature engineering section, 4, 3 and 2 input features were decided to go ahead with based on correlation coefficients in Figure 7. Three types of regression models were used namely linear regression model, random forest regression model and decision tree regression model. The first set of models, run without hyperparameter tuning, were linear regression models. The number of iterations and regularization parameters were optimized manually. The best forecast based on R^2 value (0.926) was given by a model that was run with all four parameters as input parameters as shown in Table 1. The second set of models were run with hyper parameter tuning by using a grid search method and a considerable improvement in results was noticed as shown in Table 1. The best forecast based on R^2 value (0.930) was given by a model that was run with all four parameters as input parameters, again.

The third set of models were run by using random forest regression model with 4, 3 and 2 input parameters as features. The models were run with hyper parameter tuning by using cross validation and grid search by giving a grid map. The number of trees and maximum depth were optimized. The best forecast based on R^2 value (0.955) was given by a model that was run with all four parameters as input parameters as in previous cases.

The fourth set of models were run by using decision tree regression model with 4, 3 and 2 input parameters as features. Again, the models were run with hyper parameter tuning by using cross validation and grid search by giving a grid map. The maxBins parameter was optimized. The best forecast based on R^2 value (0.931) was given by a model that was run with all four parameters as input features as in all previous cases. Table 1 shows that random forest regressor gave the best regression model performance.

4.2 Neural Network

Since the results of a neural network are dependent upon number of hidden layers a great deal, the first set of models were run with all four input parameters e.g. ambient temperature, exhaust vacuum pressure, atmospheric pressure and relative humidity, with 1, 2 and 3 number of hidden layers. The best value of R^2 (0.921) with testing dataset was found to be with one hidden layer which is worse than the optimized linear regression R^2 value. It could be due to linear correlation of independent variables (Ambient temperature and exhaust vacuum) to the dependent variable which is considered one of reasons for poor performance of neural networks compared to regression models. Therefore, it was decided to change feature selection to reduced number of variables that have more direct correlation with predicted parameter i.e. power output.

Table 1 Regression Models Results

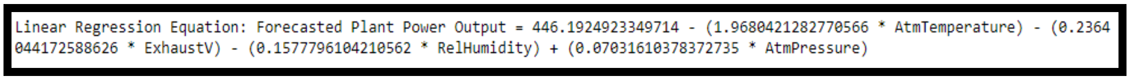
Model	No. Of Features	Hyper Parameter Tuning	R^2 of Training Dataset	R^2 of Testing Dataset
Linear Regression	4	✗	0.928	0.926
	3	✗	0.917	0.915
	2	✗	0.915	0.914
	4	✓	0.929	0.93
	3	✓	0.918	0.911
	2	✓	0.916	0.909
Random Forest Regressor	4	✓	0.973	0.955
	3	✓	0.964	0.949
	2	✓	0.951	0.938
Decision Tree Regressor	4	✓	0.936	0.931
	3	✓	0.935	0.929
	2	✓	0.933	0.928

Table 2 Neural Network Performance Results

No. Of Features	No. of Hidden Layers	Loss Function	Optimizer Learning Rate	No. of Epochs	R^2 of Training Dataset	R^2 of Testing Dataset
4	1	MSE	0.001	100	0.924	0.921
	2	MSE	0.001	100	0.859	0.861
	3	MSE	0.001	100	0.918	0.917
2	1	MSE	0.001	100	0.913	0.914
	2	MSE	0.001	100	0.92	0.921
	3	MSE	0.001	100	0.854	0.857
3	1	MSE	0.001	100	0.913	0.913
	2	MSE	0.001	100	0.91	0.91
	3	MSE	0.001	100	0.916	0.915
3	1	MAPE	0.001	100	0.914	0.914
	2	MAPE	0.001	100	0.916	0.913
	3	MAPE	0.001	100	0.884	0.887
4	1	MSE	0.0001	200	0.92	0.92
	2	MSE	0.0001	200	0.926	0.927
	3	MSE	0.0001	200	0.929	0.929
	1	MSE	0.1	100	0.897	0.895
	1	MSE	0.01	100	0.909	0.909
	1	MSE	0.00001	200	-367	-364

The number of input parameters or features was changed to 2 (ambient temperature and exhaust vacuum with better correlation coefficient than the other two as shown in Figure 7) by modifying feature engineering section. It did not improve the R^2 (0.92 with 2 hidden layers) with testing dataset after running the model with 1, 2 and 3 hidden layers but it still performed worse than the regression models. Then, the number of input parameters or features was changed to 3 (with ambient temperature, exhaust vacuum and atmospheric pressure by modifying feature engineering section. Atmospheric pressure has better correlation coefficient than the relative humidity as shown in Figure 7. It did not improve the R^2 (0.915 with 3 hidden layers) with testing dataset after running the model with 1, 2 and 3 hidden layers and it still performed worse than the regression models. Since all the previous models had MSE (Mean squared error) set as loss function, it was decided to run the neural net work model with different loss function. Therefore, models with 3 input parameters and 1, 2 and 3 hidden layers were modified to use MAPE (Mean absolute percentage error) as a loss function. The best value of R^2 (0.914) with newly created models was found to be one hidden layer.

The regression model with four input features showed highest score of R^2. The correlation, one of the main deliverables to the client, is given below.



```
Linear Regression Equation: Forecasted Plant Power Output = 446.1924923349714 - (1.9680421282770566 * AtmTemperature) - (0.2364
044172588626 * ExhaustV) - (0.1577796104210562 * RelHumidity) + (0.07031610378372735 * AtmPressure)
```

5. Conclusions

In this project, firstly, the business problem and use case were identified. Then, required data to solve the business problem was specified. The data was extracted from the identified source and prepared for analysis by data wrangling. Exploratory data analysis was performed to have an idea of tackling the business problem. Feature engineering was done based on exploratory data analysis and further correlation recognition. Lastly, Machine learning, hyperparameter tuning and deep learning were used to build and optimize models to solve business problem i.e. coming up with a correlation to estimate plant power output based on ambient conditions data. This report has been generated as the main deliverable for this business problem that also contains the correlation (section 4).

References

1. Brun, K., Friedman, P., and Dennis, R. (2017). “Supercritical Carbon Dioxide (SCO2) Based Power Cycles” 1st Edition. Woodhead Publishing. ISBN: 9780081008041
2. https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant
3. http://www.caiso.com/Pages/TodaysOutlook.aspx
4. https://statisticsbyjim.com/regression/interpret-r-squared-regression/