Tutorial #4

The job of a data scientist is to develop a robust model to predict unseen events using the given data. With the rapid increase in computation power available, data scientists are not concerned about the complexity of the models. The state-of-the-art models available are almost impossible to be operated manually by any human. We discussed one such black box model in the previous tutorial called SVM.

Nature provides us inspiration for several advancements in science and technology. One such miracle is the nervous system of an organism. Neurons and their interconnections make up the nervous system. Scientists have beenstudyingthese neural networks for several centuries. All this study has come to fruition.

Data scientists have developed Artificial Neural Networks. To do this, the greatest problem to overcome is to translate a physical phenomenon to mathematics.

“With me everything turns into mathematics.” —Descartes

The neural networks developed by scientists are very complex equations and algorithms with several parameters and variables, hard to be interpreted manually. Fortunately, we do not have to understand them to use them. It is like driving a car. You do not have to be an expert mechanic to drive a car. You only have to know how the car will behave in various circumstances. Artificial Neural Networks (ANN) can also be used just as we use cars.

However, to get a basic understanding watchthisvideo.

ANNs are helping humanity achieve what was unimaginable before. To appreciate the development of ANNs, ACMawarded2018 ACM A.M.Turing Award, the highest prize awarded in the field of computer science, to three individuals “for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.” Several types ofANNsare developed for various purposes. However, for STLF, we will use the most straightforward type i.e., feedforward neural network.

Combined Cycle Power Plant

Power plants operate continuously to supply electrical power to massive loads. The power generation process should be efficient to increase profits and should not be interrupted as some essential loads might depend on the power plant such as hospitals andfactories with critical processes.

To make sure that power plants keep working correctly, engineers monitor various parameters of power plants to judge performance. By using these parameters, we can calculate the generation of a power plant. And by comparing the calculated and the actual generation of a power plant, we can detect if something is wrong.

Combined Cycle Power Plant (CCPP)uses both steam and gas to generate electricity. We will use four parameters described below to predict power output **EP**, of the power plant.

- Temperature (T) in °C,
- Ambient Pressure (AP) in milibar
- Relative Humidity (RH) in %
- Exhaust Vacuum (V) in cm Hg
- Net hourly electrical energy output (EP) in MW

The datset that we will use is availablehere.

The given data file is in .xlsx format. So we will use a package that specializes in handling such files.

The “CCPP.xlsx” file contains 5 sheets of the same data, shuffled to remove any order that might exist. Read more about ithere. However, we will only use data from sheet # 1.

```
# To help in reading .xlsx files
library(readxl)

# This function provides several parameters. Run ?read_excel
CCPP = read_excel("CCPP.xlsx",sheet =1)

str(CCPP)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':       9568 obs. of    5 variables: ##
    $ T : num    14.96 25.18 5.11 20.86 10.82 ...
##  $ V : num    41.8 63 39.4 57.3 37.5 ...
##  $ AP: num    1024 1020 1012 1010 1009 ...
##  $ RH: num    73.2 59.1 92.1 76.6 96.6 ...
##  $ EP: num    463 444 489 446 474 ...
```

The output above shows the structure of the data. There are 9568 instances, and all five variables are numeric.

The summary of the data shows that the maximum output power is 495.8 MW, which might also be the capacity of the power plant.

```
summary(CCPP)
```

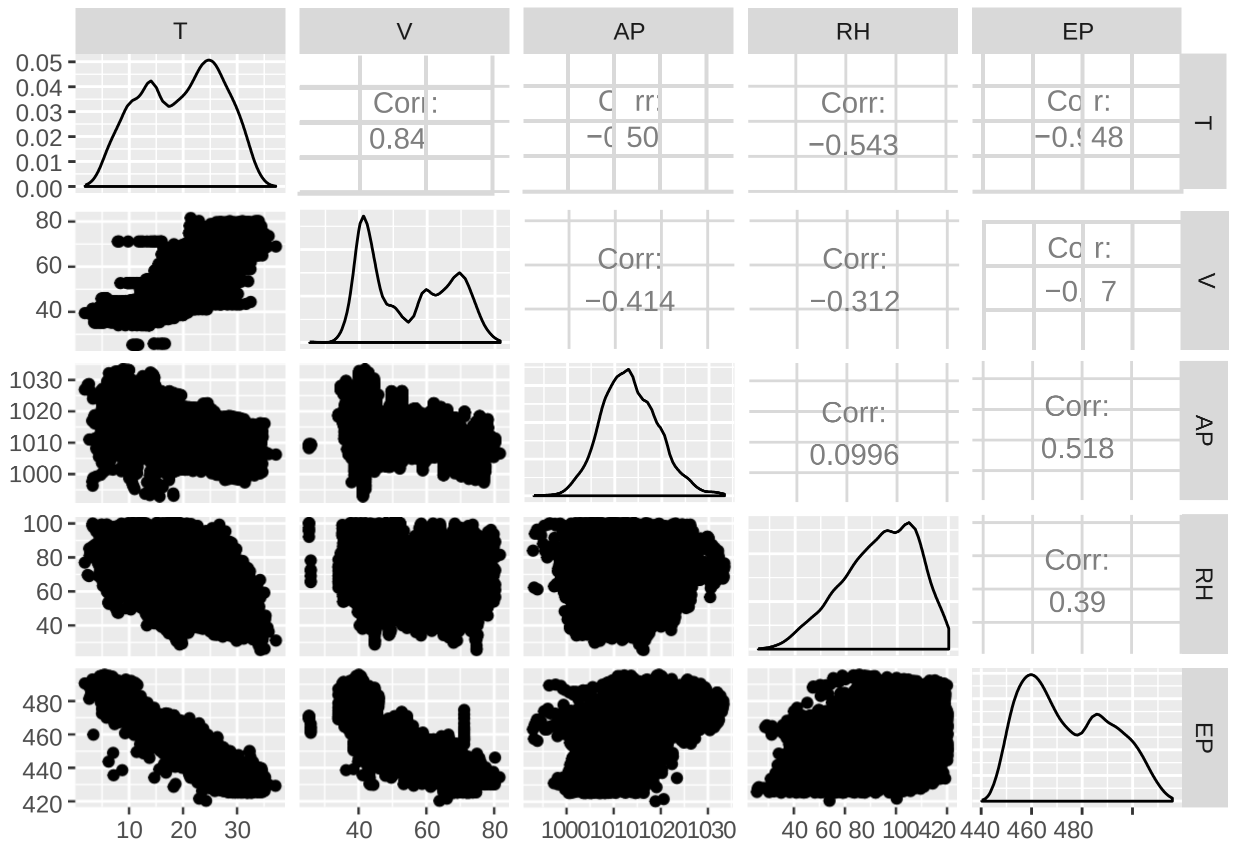
```
##        T               V               AP               RH
##  Min.   : 1.81   Min.   :25.36   Min.   : 992.9   Min.   : 25.56
##  1st Qu.:13.51   1st Qu.:41.74   1st Qu.:1009.1   1st Qu.:  63.33
##  Median :20.34   Median :52.08   Median :1012.9   Median :  74.97
##  Mean   :19.65   Mean   :54.31   Mean   :1013.3   Mean   :  73.31
##  3rd Qu.:25.72   3rd Qu.:66.54   3rd Qu.:1017.3   3rd Qu.:  84.83
##  Max.   :37.11   Max.   :81.56   Max.   :1033.3   Max.   :100.16
##        EP
##  Min.   :420.3
##  1st Qu.:439.8
##  Median :451.6
##  Mean   :454.4
##  3rd Qu.:468.4
##  Max.   :495.8
```

The plot below shows the data distribution and correlation between factors. Both **T** and **V** show a high correlation with **EP**, but both also have a high correlation with each other, so we will keep one. The summary above shows that **V** has a greater range (81.56 - 25.36 = 56.2) than **T** (37.11 - 1.81 = 35.3). So, we will use **V**. **AP** and **RH** both have marginal correlations with **EP**. We will use them both.

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.6.3
```

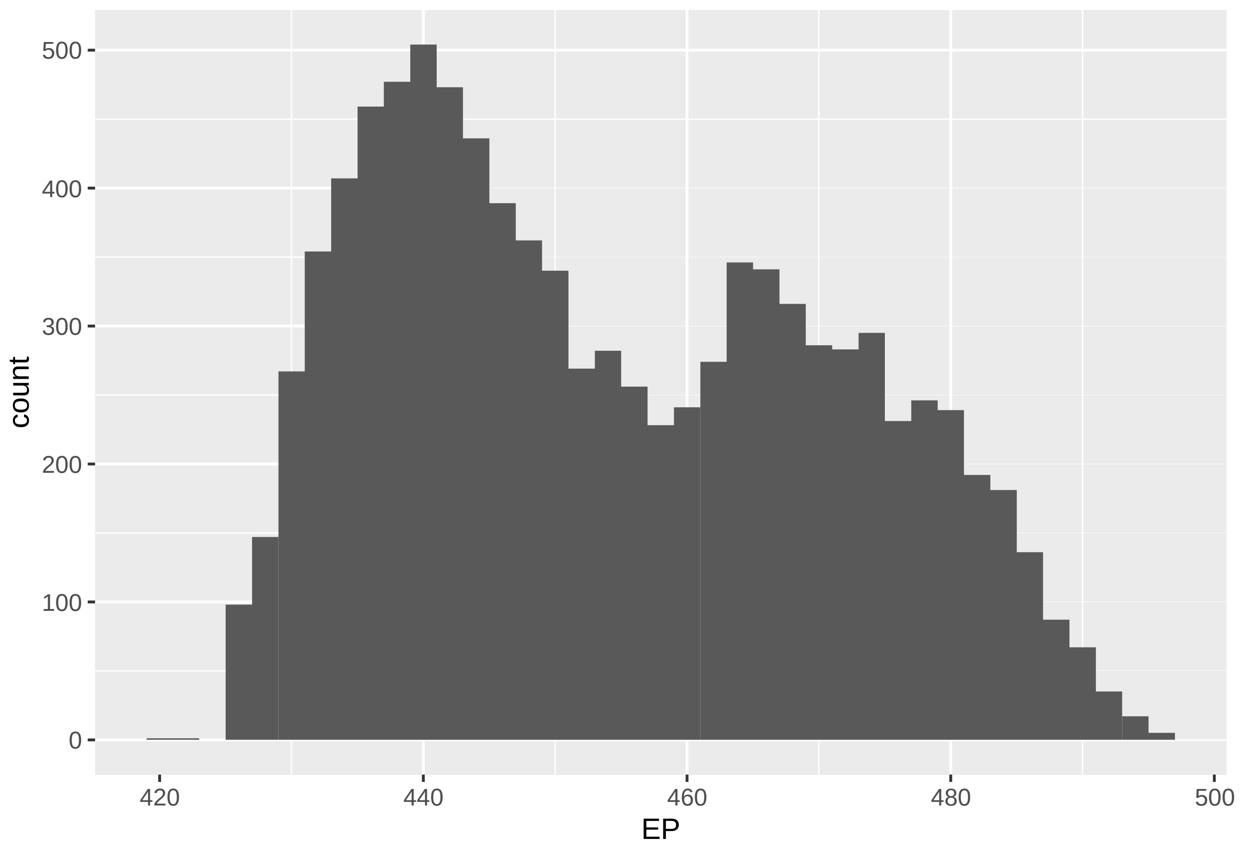
```
ggpairs(CCPP)
```



One of the most popular packages to create amazing plots and graphs is "ggplot2". Learning the art of data visualization will be fruitful for both research and development. We will use "ggplot2" to create the histogram for **EP**.

The plot below shows two peaks, one at approximately 440 MW and the other at about 465 MW. These two peaks are the two stages of CCPP. When more than 440 MW of power is required, the CCPP is shifted to the second stage to generate approximately 465 MW.

```
library(ggplot2)
# Basic histogram
ggplot(CCPP, aes(x=EP)) +geom_histogram (binwidth=2)
```



To prepare data for ANN training, first it is scaled to be between 0 and 1. This is called Normalization, it can be defined as follows:

$$X_N = \frac{X - min(X)}{max(X) - min(X)}$$

For more details about why scaling is important, readthis.

```
# Scale the Data
max = apply(CCPP[,-1] ,2, max)
min = apply(CCPP[,-1],2, min)
scaled = as.data.frame(scale(CCPP[,-1],center =min,scale =max                    - min))
```

Lets divide the data into test and train.

```
smp_size <- floor(0.70 *nrow (scaled))
train_ind <- sample(seq_len(nrow(scaled)),size =smp_size) train_CCPP_scaled <-
scaled[train_ind, ]
test_CCPP_scaled <-scaled[ -train_ind, ]
```

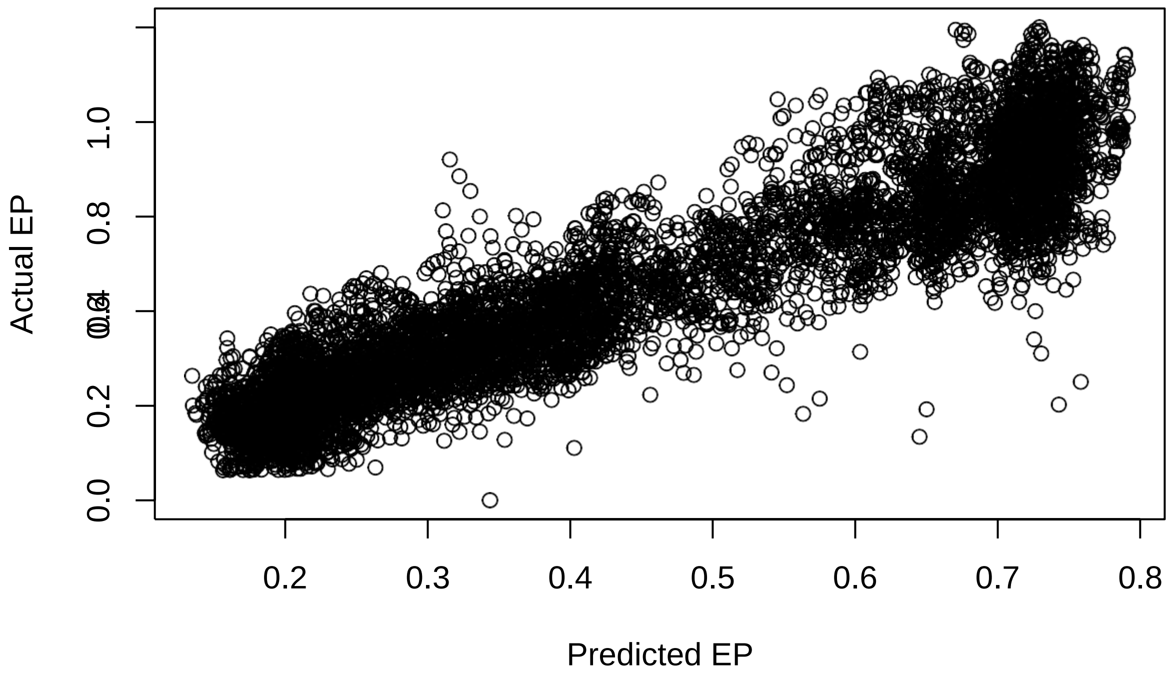
To implement the ANN, we will use “neuralnet” package. The function is also called *neuralnet()*. The dependent variable **EP** will be predicted using **V**, **AP**, and **RH**. The data used is *train_CCPP*. The *hidden* parameter is used to define the number of neurons in each hidden layer of the neural network. We are defining three neurons in the first and two neurons in the second layer. The *linear.output* should be true in case we want to perform regression.

After the model completes its training, the model predicts the training data and stores it in *net.result*. In the plot below, the predicted **EP** is plotted against the actual **EP**. As expected by a good model, the values lie on the y=x line.

```
library(neuralnet)

ANN_model = neuralnet(EP~V+AP+RH,data =train_CCPP_scaled,
                                        hidden = c(3,2),linear.output =TRUE)

plot(as.vector(ANN_model$net.result[[1]]), train_CCPP_scaled$EP, xlab ="Predicted
         EP",ylab ="Actual EP")
```



We can also visualize our model, using the following code. The plot shows the weights of each neuron, as we can see that there are two hidden layers with three and two neurons in each layer.

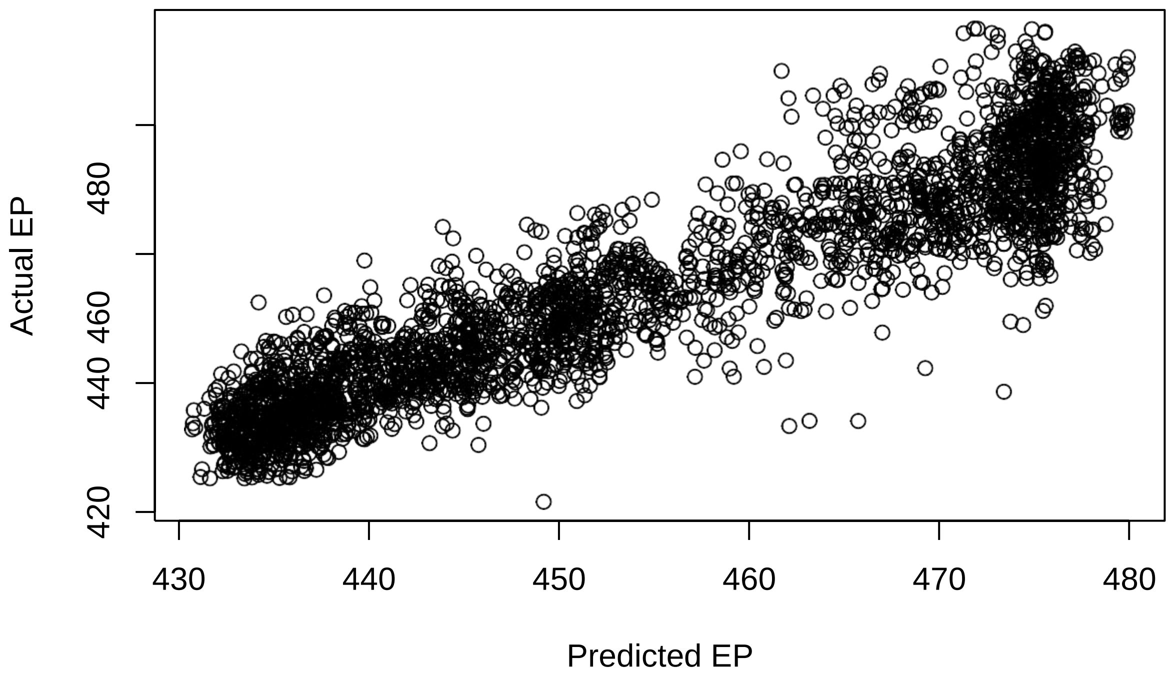
```
plot(ANN_model)
```

The following code will predict the test data and stores it in *test_CCPP_scaled$EP_Predicted*. The plot below shows the actual vs. predicted values for test data. The plot shows that the model is predicting unseen data quite nicely. The predictors had a high correlation with **EP**, which meant a strong linear relationship. Predicting **EP** is not very difficult for ANN.

```
CCPP_test_scaled_prediction = compute(ANN_model, test_CCPP_scaled[,-4]) test_CCPP_scaled$EP_Predicted
=(CCPP_test_scaled_prediction $net.result * (max(CCPP$EP,na.rm =T)                                      -
                                                  min(CCPP$EP,na.rm =T)))         +min (CCPP$EP,na.rm =T)

test_CCPP_scaled$EP_Actual =(test_CCPP_scaled $EP * (max(CCPP$EP,na.rm =T)                    -
                                                  min(CCPP$EP,na.rm =T)))         +min (CCPP$EP,na.rm =T)


plot(test_CCPP_scaled$EP_Predicted, test_CCPP_scaled$EP_Actual,
        xlab ="Predicted EP",ylab ="Actual EP")
```



The table below shows the error metrics for the test data predictions.

```
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##      method           from
##      as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##      method                      from
##      as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##      method                       from
##      fitted.fracdiff              fracdiff ##
        residuals.fracdiff fracdiff
```

```
accuracy(test_CCPP_scaled$EP_Actual, test_CCPP_scaled$EP_Predicted)
```

```
##                                    ME         RMSE           MAE                  MPE
                                   MAPE ## Test set -0.05357375 6.748359 5.159662 -
0.012543631.127849
```

Now using this model, we can predict the output power of CCPP. If for some reason, the CCPP malfunctions and the output power deviates from what is predicted by the model, alarms will be raised automatically to alert power plant engineers.

One thing to note is that we do not know the exact calculations of how the ANN model is calculating the unseen data, unlike the linear model we discussed in tutorial # 2, in which we manually devised the model equation.

Create a new ANN model, with three hidden layers, there should be three neurons in the first and the third layer and 8 in the second layer. The model would take some time to train. Be patient!

Compare your model to mine. Which one is better?

In tutorial # 2, you used linear regression to predict the cereal rating. Create an ANN model using multiple variables to predict the rating for the same data. Write a complete report comparing the two models.

Email me your detailed report.