# MODULE 16

# **50 Start Up Dataset**

**Package used in R**

Neuralnet - Used to run the Neural Network

**Package used in python**

Pandas - Used for data manipulation

Numpy - Used for Mathematical calculation

Keras.models

Sequential- layer of Neurons

keras.layers

Dense - Connect all the layers

Matplotlib - Data Visualization

**Loading the data**

Loading the 50 Startup dataset in R and Python

**EDA**

No NaN Data Found in the dataset

Creating Dummy Variables on States Columns

Normalization the dataset with min-max Method

Rearranging the dataset

**Data Partitioning**

Creating Predictors and Target variables from the dataset

Splitting the data to x\_train,x\_test as Predictors with .75 data size; y\_train,y\_test as Target with .25 data size

**Modeling and Result**

Using Linear Activation Function like ReLu, SeLu, eLu.

**model.compile(loss="mean\_squared\_error",optimizer="adam",metrics = ["accuracy"])**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KNN Parameter | | | | Training | | Testing | |
| **Epochs** | **Activation 1** | **Activation 2** | **Layer** | **RMSE** | **Corr Coef** | **RMSE** | **Corr Coef** |
| 800 | relu | relu | 5,500,1 | 0.2386 | 0.9950 | 0.2885 | 0.9806 |
| 800 | selu | selu | 5,500,1 | 0.2241 | 0.9868 | 0.2561 | 0.9846 |
| 800 | elu | elu | 5,500,1 | 0.2174 | 0.9738 | 0.2683 | 0.9822 |

**Selecting the Model : Less the RMSE Source and Higher the Corr Coef the Better the Model**

Model with **Selu** Activation Function has given a better result.

# **Forest Fire Dataset**

**Package used in R**

Neuralnet - Used to run the Neural Network

**Package used in python**

Pandas - Used for data manipulation

Numpy - Used for Mathematical calculation

Keras.models

Sequential- layer of Neurons

keras.layers

Dense - Connect all the layers

Matplotlib - Data Visualization

**Loading the data**

Loading the fireforests dataset in R and Python

**EDA**

No NaN Data Found in the dataset

Selected the best variables for the models.

Normalization the dataset with min-max Method

Rearranging the dataset

**Data Partitioning**

Creating Predictors and Target variables from the dataset

Splitting the data to x\_train,x\_test as Predictors with .75 data size; y\_train,y\_test as Target with .25 data size

**Modeling and Result**

Using Linear Activation Function like ReLu, SeLu, eLu.

**model.compile(loss="mean\_squared\_error",optimizer="adam",metrics = ["accuracy"])**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KNN Parameter | | | | Training | | Testing | |
| **Epochs** | **Activation 1** | **Activation 2** | **Layer** | **RMSE** | **CorrCoef** | **RMSE** | **CorrCoef** |
| 500 | elu | elu | 4,250,1 | 0.2692 | 0.9999 | 0.2653 | 0.9999 |
| 500 | selu | selu | 4,250,1 | 0.2695 | 0.9999 | 0.2652 | 0.9999 |
| 500 | relu | relu | 4,250,1 | 0.2695 | 0.9999 | 0.2653 | 0.9999 |

**Selecting the Model : Less the RMSE Source and Higher the Corr Coef the Better the Model**

Model with **elu** Activation Function has given a better result.

# **Concrete Dataset**

**Package used in R**

Neuralnet - Used to run the Neural Network

**Package used in python**

Pandas - Used for data manipulation

Numpy - Used for Mathematical calculation

Keras.models

Sequential- layer of Neurons

keras.layers

Dense - Connect all the layers

Matplotlib - Data Visualization

**Loading the data**

Loading the Concrete dataset in R and Python

**EDA**

No NaN Data Found in the dataset

Normalization the dataset with min-max Method

Rearranging the dataset

**Data Partitioning**

Creating Predictors and Target variables from the dataset

Splitting the data to x\_train,x\_test as Predictors with .75 data size; y\_train,y\_test as Target with .25 data size

**Modeling and Result**

Using Linear Activation Function like ReLu, SeLu, eLu.

**model.compile(loss="mean\_squared\_error",optimizer="adam",metrics = ["accuracy"])**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KNN Parameter | | | | Training | | Testing | |
| **Epochs** | **Activation 1** | **Activation 2** | **Layer** | **RMSE** | **CorrCoef** | **RMSE** | **CorrCoef** |
| 500 | relu | relu | 8,50,100,1 | 0.2862 | 0.9680 | 0.2894 | 0.9664 |
| 500 | relu | relu | 8,50,200,1 | 0.3052 | 0.9900 | 0.2446 | 0.9513 |
| 500 | elu | elu | 8,50,200,1 | 0.2895 | 0.9402 | 0.2413 | 0.9339 |
| 500 | selu | selu | 8,50,200,1 | 0.2968 | 0.9707 | 0.2440 | 0.9468 |

**Selecting the Model : Less the RMSE Source and Higher the Corr Coef the Better the Model**

Model with **Relu** Activation Function has given a better result.