A REPORT
ON

MACHINE LEARNING IN PREDICTION OF CROPS YIELD IN RAJASTHAN

BY

Rajeev Singh Naruka ID No. 2017A7PS0010P

At

Directorate of Economics and Statistics, Rajasthan
Jaipur

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
(July, 2019)

A REPORT
ON

Machine Learning in Prediction of Crops Yield in Rajasthan

BY

Rajeev Singh Naruka ID No. 2017A7PS0010P B.E. Computer Science

Prepared in partial fulfilment of the

Practice School-1 Course No.
BITS F221

At
Directorate of Economics and Statistics, Rajasthan

Jaipur

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

(July, 2019)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)
Practice School Division

Station(s): Directorate of Economics and Statistics **Centre:** Jaipur
Duration: From: _21 May 2019 **To**: 13 July 2019
Date of Submission: 12 July 2019

Title of the Project: Machine Learning in Prediction of Crops Yield in Rajasthan

ID No.	Name(s) of Student(s)	Discipline
2017A7PS0010P	Rajeev Singh Naruka	Computer Science(A7)

Name(s) of expert	Designation
Mr. Mahesh Chandra Verma	Statistical Officer

Name of the PS Faculty
Mr. Hari Om Bansal

Key Words: Artificial Neural Network, ANN, Machine Learning, Data Mining, Deep Learning, Backpropagation, Analytics, Big data.

Project Area(s): Real life application of Machine Learning Algorithms, Artificial Neural Network, Impact of Technology in Agriculture sector, Precision Farming
‘

Abstract: Traditionally the Directorate of Economics and Statistics in Rajasthan predicts the crop yield immediately after the sowing season for planning purpose of the Government by using techniques like Crop-Cutting experiment and Statistical Methods. The use of Machine learning Algorithms like Artificial Neural Networks in predicting crop yield, which has achieved great success worldwide due to its precision, is still not a common practice. In this report, we try to discuss the potential of this technique in the Rajasthan Agriculture scenario and also develop an initial Artificial Neural Network Model based on the agriculture data provided by the department and other reliable sources like government portals. The Artificial Neural Network has the ability to self-improvise based on the results so that it can achieve more precision in predicting. In our model, we have used various weather and non-weather factors that come in to play when predicting crop yield. Data collected over many years are fed into the Model along with the crop yield as a special attribute so that the model can train and understand the non-linear inter-dependencies of various factors and the crop yield. Finally, the report concludes with suggestions for the improvement of the model in the future.

Signature(s) of the Student(s) **Signature of PS Faculty**

Date **Date**

Contents

Cover Page 1
Title Page 2
1. Introduction to Directorate of Economics and Statistics, Rajasthan 8
1.1 ABOUT THE DIRECTORATE 8
1.2 ABOUT THE AGRICULTURE AND TRS SECTION 9
1.2.1 Timely Reporting Scheme (TRS) 9
1.2.2 Crop Enumeration 9
2. Background and Motivation 10
2.1 Agriculture in India 10
2.2 Aim of the Project 11
3. About the Project 11
3.1 Technology and Algorithms used 11
3.2 What is an Artificial Neural Network? 12
3.3 How an ANN works? 13
3.3 Timeline and methodology 14
4. Analysis of Data with RapidMiner© Studio 15
4.1 Brief Introduction to the software 15
4.2 Mathematical Approach to ANN 15
4.2.1 Neurons 15
4.2.2 Connections, weights and biases: 16
4.3 Data Parameterization and Data Collection 16
4.3.1 Deciding Parameters 16
4.3.2 Data collection 17
4.4 Building the first predictive Model 18
4.4.1 Testing the Model A 21
5. Improving and refining Models 22
5.1 Improving upon existing model A 22
5.1.1 Deep Learning 23
5.1.2 Lazy Algorithm 27
5.1.3 Decision Tree 28
5.2 Adding more variables: Model C 31
5.2.1 Neural Net 32
5.2.2 Deep Learning on Model C 35
5.2.3 Lazy Algorithm on model C 38
5.2.4 Decision tree on Model C 39
6. Making predictions with models developed for next year 41

7. CONCLUSION AND WAY FORWARD 41
Appendix A 43
Bibliography 52

"The best way to predict the future is to invent it."

– Alan Kay

Acknowledgements

This project would not have successfully completed without a lot of guidance and assistance and I am extremely privileged to have got this all along the completion of my project from the munificent employees at the Directorate of Economics and Statistics, Rajasthan. All that I have done is only due to such supervision and assistance and I would not forget to thank them from the bottom of my heart.
I respect and thank Dr. Om Prakash Bairwa, Director and Joint Secretary, DES, for providing me an opportunity to do this project work here at DES and giving us all support and guidance which made me complete the project duly. I am extremely thankful to Mr. Ghanshyam Dutt Agrawal, Assistant Director, Agriculture and TRS section, DES for providing such a nice support and guidance, although he had a busy schedule managing the directorate affairs.

I owe my deep gratitude to PSD BITS Pilani for providing this opportunity and our PS-1 instructor Dr. Hari Om Bansal who guided me all along, till the completion of the project work. I would like to thank Mr. Mahesh Chandra Verma, Statistical Officer, DES, who took keen interest in my project work and helped immensely by providing all the necessary information for developing the ANN model. I would not forget to remember Mr. Bhudev Singh of training section, DES for his encouragement and more over for their timely support and guidance till the completion of our project work.
I heartily thank our co-instructor Ms. Pratibha Saini, BITS Pilani for her guidance and suggestions during this project work.
I must express my gratitude for being fortunate enough to get constant encouragement, support and steerage from all the employees at DES which helped me in successful completion the project work.

Rajeev Singh Naruka
2017A7PS0010P
BE (Hons.) Computer Science
BITS Pilani

1. Introduction to Directorate of Economics and Statistics, Rajasthan

1.1 ABOUT THE DIRECTORATE

The Directorate of Economics and Statistics (DES) was instituted as the Principal Statistical Organisation in the year 1956 to cater for the planning requirement of the State. There are also different statistical cells in the various department of the State Government which conduct statistical activities apart from DES as required by the department. The Directorate of Economics and Statistics also collaborates with the Central Government's statistical organizations in related statistical matters. [1]

The head of Directorate is Dr. Om Prakash Bairwa, Director, who conjointly act as Chief Registrar of Birth and Deaths, Statistical Authority for State Agricultural and Officer on Special Duty for Annual survey of industries.

the Directorate of Economics and Statistics, Rajasthan has been declared as Nodal agency for the State, on the recommendation of the Committee to review the National Statistical System, for coordinating the statistical activities in the State and developing an efficient statistical system. As a Nodal Agency the Directorate of Economics and Statistics, Rajasthan is made responsible and entrusted with the following functions:

1. Co-ordination of Statistical activities of various departments / organizations of the State Government.

2. Undertaking integration of data needed in numerous policy sectors and provide suggestions and means to improve the quality and coverage of data series

3. To avoid duplication in collection/assortment and compilation of data, and

4. To integrate accepted standards in collection of data, classification, processing and dissemination of data in its workflow.

1.2 ABOUT THE AGRICULTURE AND TRS SECTION

1.2.1 Timely Reporting Scheme (TRS)

INTRODUCTION:

The Timely Reporting Scheme (TRS) is being implemented in the State since 1972-73, sponsored by the Indian Government. The scheme intends to arrive at reliable, quick and accurate estimates of area immediately after the sowing of principal crops at the end of each agricultural season viz. Kharif, Rabi and Zaid Rabi within the State on the premise of actual area enumeration in the selected 20% villages through sample survey. These estimates help the Government to assess agricultural prospects during the current year and serve in taking quick policy measures relating to agricultural production, imports and exports of the agricultural commodities.

OBJECTS:

The main objectives of the scheme are to obtain:

1. reliable and statistically precise estimates of area under principal crops immediately after sowing of crops.
2. the estimates of area for irrigated and un-irrigated crops separately.
3. the estimates of area under high yielding varieties of crops.
4. the estimates of land utilization on the basis of nine-fold classification.

1.2.2 Crop Enumeration

The agricultural year in the State covering the period from 1st July to 30th June is divided into three seasons viz; Kharif, Rabi and Zaid Rabi. Following major crops are covered during each agricultural season under the scheme: -

SEASON	CROPS
(i) Kharif	Rice, Jowar, Bajra, Maize, Tur, Groundnut, Sesamum, Castor seed, Cotton, Sugarcane and Soya bean.
ii) Rabi	Wheat, Barley, Gram, Linseed, Rape & Mustard and Tara Mira.
iii)Zaid Rabi	Jowar, Bajra, Maize, Urad, Moong, Moth, Chaula, Fruits & Vegetables, Tobacco, Small millets, Spices, Fodder and chillies

The instructions were made to Tehsildars to hold the meeting of the Patwaris immediately after the completion of the field work and collect all the filled-in-schedules for onward transmission to the District/Block Economics and Statistics office. These schedules were scrutinized and data entry was performed on TRS portal at every District/Block office. Under this, estimates are worked out by both the ratio and unbiased methods of estimation. Annual Publication of Agricultural Statistics (APAS) is also released annually. This Publication contains district wise data concerning to classification of land use, source wise net and gross irrigation area, crop wise irrigated and total area, production of crops, average yield, price of farm harvest, quantity of irrigation sources and agricultural indices etc.

2. Background and Motivation

2.1 Agriculture in India

Agriculture is said to be the backbone of the Indian Economy and India ranks at place 2 in farm outputs worldwide. As of 2018, Agriculture sector provided employment to about half of the total workforce in India contributing to almost one-fifth (17-18%) to the nation's GDP.[2] Demographically it is the largest economic sector and plays a vital role in the overall socio-economic dynamics of our nation. Agriculture or crop production is dependent on many climate and economy factors. These factors are soil type and condition, climate, cultivation, irrigation, fertilizers, temperature, rainfall, harvesting, pesticide weeds and other factors. Historical data of crop yield play vital part in policy and decision making for supply chain operation of governments and companies engaged in agro-industries. An accurate estimate of crop production and risk helps these companies in planning supply chain decision like production scheduling, transportation decisions and other planning's like financial management as well. Business such as seed, fertilizer, agrochemical and agricultural machinery industries plan production and marketing activities based on crop production estimates. Following are some factors which are helpful for the farmers and the government in decision making namely:

a. It helps the government in making crop insurance policies and policies for supply chain operation.

b. It is the next step towards precision farming and helps in proper utilisation of resources.

c. It helps farmers in providing the historical crop yield record with a forecast reducing the risk management.

2.2 Aim of the Project

This project aims to construct a model that can help in accurately predicting crop yields in Rajasthan from climate and agricultural data made available by the Directorate. Although, some studies revealed statistical information about the agriculture in Rajasthan, not many studies have investigated crop yield prediction based on the historic climatic and production data. Artificial Neural Networks (ANNs) have been used widely for various purposes including classification, forecasting, vector quantization, clustering, function approximation, pattern association, control applications and optimization (Mehrotra, Mohan and Ranka, 1996). ANN predictions have also found use in financial industry and climate prediction. In this paper an ANN is used to predict crop yields based on the data provided for the state of Rajasthan in India.

3. About the Project

3.1 Technology and Algorithms used

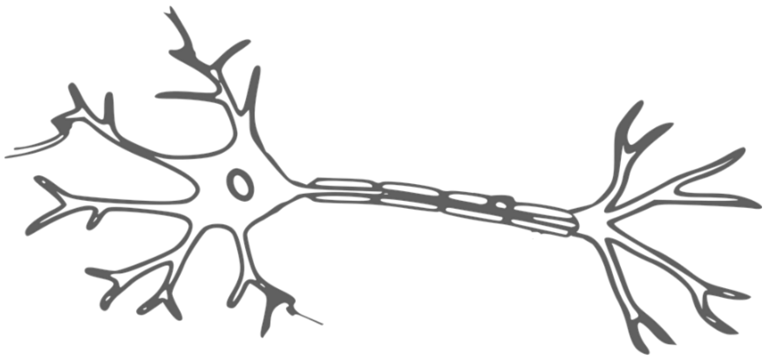
The project is based around use of different Machine Learning Algorithms and one of which is prominently focused in the project called Artificial Neural Network which is used to make the model to predict the crop yield. This ANN was built on a quasi- open source Software RapidMiner© Studio: Scholar Edition. Other than this the data was handled in Microsoft

Excel Program also. Various other plotting and visualization software were also used in the duration of the project.

3.2 What is an Artificial Neural Network?

To understand the project, one must have a working understanding of an Artificial Neural Network (ANN). This artificial software model is inspired from biological Neural Network or the Nervous System which consists of nerve cells called “Neurons”. A typical Neuron has 4 parts:

1. Dendrites: These nodes help in taking input electro-chemical signals from other cells.
2. Soma: This part help in processing the signal for output.
3. Axon: An axon helps to output the processed input.
4. Synapses: These act as the connection between different Neurons.



Similar to the Biological Neural Network, we have an Artificial Neural Network, also called a multilayer perceptron, which has same working parts: An input node layer, Processing node layers (called hidden layers), Output layer and Connections (called weights).

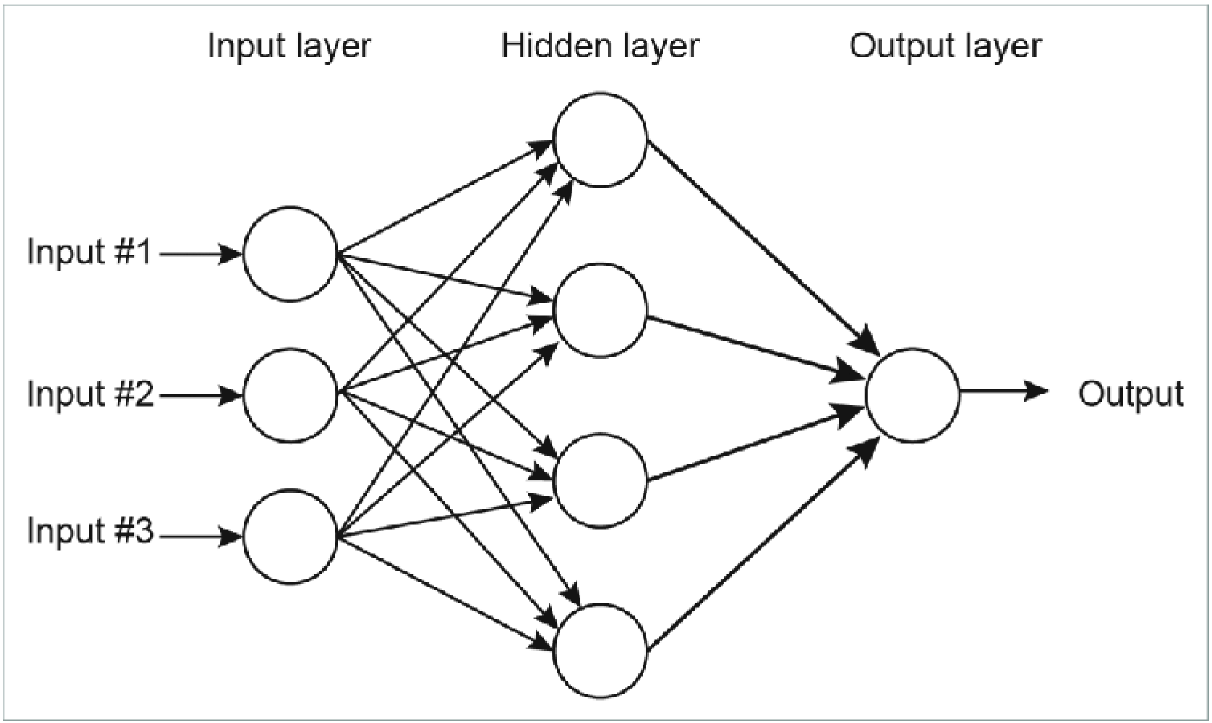


Figure 1. An Artificial Neural Network

3.3 How an ANN works?

An ANN, works similar to the information processed by the human brain (Hinton, 1992). A number of artificial units representing neurons are interconnected to form an ANN (Gershenson, 2003 and Yagnanarayana, 2006). The synapses, which are tips of an axon and act as connection between different nodes, are represented by a modifiable weight. Each unit receives an input which is integrated with the weight (say w1), a floating-point number, and transfers other units. Each input unit multiplies with its associated weight on the connection and all weighted inputs are added to get a quantity called the total input. The output which is produced by the network developed, is highly influenced by the weights associated with it and the input-output function of individual units (Yagnanarayana, 2006). This is how an ANN works in basically.

The determination of the connections and of the units with other units is necessary in order to perform a specific task using a neural network (Yegnanarayana, 2006 and Hassoun, 1995). The influence of one unit upon others is determine by their interconnections. Usually, ANN contains three layers for each unit. An input layer is connected to a hidden layer and the hidden layer is connected to an output layer. The input unit represents a vector of data provided to the network. The hidden layer performs tasks on the basis of input and weight. The output depends on the activity of the hidden layers associated with their respective weights (Hassoun, 1995).

3.3 Timeline and methodology

This project was completed in a pre-planned timeline with proper emphasis on each part of learning. First week was dedicated to Literature Survey in which I Learned about various Machine Learning Algorithms as well had a better grasp of ANN. I discussed with my Mentor Mr. Mahesh Chandra Verma (Statistical Officer, DES), about the various methods DES deploys to estimate the crop yield. I had discussions about crop-cutting techniques used by the Directorate in the Districts of Rajasthan. Next week was spent on the Data collection and its Parameterization in the RapidMiner© Studio. The data provided by the Agriculture department initially was crop-wise production data and was of little use for the model. I later requested them to provide source-wise data of previous years and they were kind enough to provide it soon. After that I visualized that data in the program and obtained valuable insights about the production in districts of Rajasthan. My main focus was Sawai Madhopur District of Rajasthan for this study of crop yield prediction. Next two weeks were spent on the Model construction and its refinement for more accuracy.

Objectives	Time Frame
1. Literature survey and study of machine learning algorithms. 2. Understanding crop yield prediction dependencies on weather variables and finding behaviour patterns.	27 May 2019 to 2 June 2019
3. Parameterization and Data Collection 4. Data visualization and labelling using visualization software (e.g. RapidMiner© Studio).	3 June 2019 to 9 June 2019
5. Data Pre-processing and transformation into model compatible format and splitting into training testing and validation sets. 6. Model Building	10 June 2019 to 16 June 2019
7. Model Training	17 June 2019 to 23 June 2019
8. Improving predictions by refining previous models	24 June 2019 to 30 June 2019
9. Conclusion of project	1 July 2019 to 12 July 2019

4. Analysis of Data with RapidMiner© Studio

4.1 Brief Introduction to the software

RapidMiner is a data science software platform researched and developed by the company of the RapidMiner, Inc. that provides an integrated environment for data preparation, text mining, deep learning, machine learning, and predictive analytics.[3] The software platform finds extensive use in business and commercial applications as well as for research, training, rapid prototyping of predictive models, education and application development and supports all steps of the machine learning process including data preparation, model validation, results visualization and optimization. RapidMiner is developed on an open core model where core code of software is opensource.

According to Bloor Research, RapidMiner helps in providing ninety-nine percentage of an advanced analytical solution through template-based frameworks that delivers speed along with reducing errors by nearly eliminating the need to write code. RapidMiner provides data mining and machine learning procedures including: transformation and data loading (Extract, transform, load (ETL)) , predictive analytics, data pre-processing and visualization and statistical modelling, evaluation, and deployment. RapidMiner is coded in the Java programming language. RapidMiner provides a GUI (Graphical User Interface) to design and execute analytical workflows. Those workflows are known as "Processes" in RapidMiner and they consist of multiple functions called "Operators". Each operator deals with a single task within the process, and the output of each operator act as the input of the next one. Alternatively, the engine can also be called from other programs or used as an API. Individual functions can be called from the command line. RapidMiner provides learning schemes, models and algorithms and can be extended using R and Python scripts.

4.2 Mathematical Approach to ANN

Followings are the components of an ANN

4.2.1 Neurons

A neuron with label j receiving an input $p_j(t)$ from predecessor neurons consists of the following components:

1. an *activation* $a_j(t)$, the neuron's state, depending on a discrete time parameter,
2. possibly a *threshold* Θ_j, which stays fixed unless changed by a learning function,
3. An activation function f that computes the new activation at a given time $t + 1$ from $a_j(t)$, Θ_j

   and the net input $p_j(t)$ giving rise to relation $a_j(t+1) = f(a_j(t), p_j(t), \Theta_j)$

and an *output function* f_{out} computing the output from the activation $o_j(t) = f_{out}(a_j(t))$.

An input neuron has no predecessor but serves as input interface for the whole network. Similarly, an output neuron has no successor and thus serves as output interface of the whole network.

4.2.2 Connections, weights and biases:

The *network* consists of connections, each connection transferring the output of a neuron *i* to the input of a neuron j_n this sense *i* is the predecessor of *j* and j is the successor of i. Each connection is assigned a weight $w_{ij.}$ Sometimes a bias term is added to the total weighted sum of inputs to serve as a threshold to shift the activation function.

4.3 Data Parameterization and Data Collection

4.3.1 Deciding Parameters

In order to model the predictive model, first thing we needed was to understand the dependencies of the crop production on various weather and non-weather variables so that we could use them as input data nodes for the model. After studying different research papers, we came to conclude that following data points might be useful:

(A) Total Precipitation (mm day-1)
(B) Temperature of 2 meters from ground(C)
(C) Surface Pressure (kPa)
(D) Earth Skin Temperature(C)
(E) Relative Humidity at 2 meters (%)
(F) Downward Thermal Infrared (Longwave) Radiative Flux (kW-hr/m^2/day)
(G) Dew/Frost Point at 2 Meters (C)
(H) Maximum Temperature at 2 Meters (C)
(I) Specific Humidity at 2 Meters (kg kg-1)

The Label or the special attribute was Total Wheat production in given Crop year.

For this study it was decided to take only a single district under consideration as the weather varies within the state to a large degree. We decided to focus on the District Sawai Madhopur for our study. Now we needed to decide if rainfall was important measure for the crop production in the district so that the predictive model could have a reliable input variable for the prediction.

The National Rainfed Area Authority (NRAA) was established in 2006 to give focussed attention to Rainfed areas of the country. This advisory body formulated some common guidelines for the Watershed Development Project and is in consultation with all the States for its implementation. According to NRAA districts having rainfall more than 750 mm per year are under rainfed farming.[5] Sawai Madhopur district receives 800 mm of rainfall on average.[6] Thus, we decided to include the rainfall data as a important input variable.

4.3.2 Data collection

The most vital data for the project was obtained from TRS section of the DES, Jaipur i.e. The annual production of the Wheat district-wise.

Majority of the other data was obtained from the NASA Langley Research Center (LaRC) POWER Project funded through the NASA Earth Science/Applied Science Program.[4] The Prediction of Worldwide Energy Resource (POWER) Project is funded through the NASA Applied Sciences Program within the Earth Science Division of the Science Mission Directorate. The Prediction of Worldwide Energy Resource (POWER) project was initiated to improve upon the current renewable energy data set and to create new data sets from new satellite systems. The POWER project targets three user communities: (1) Renewable Energy, (2) Sustainable Buildings, and (3) Agroclimatology.

The Agroclimatology Archive is designed to provide web-based access to industry-friendly parameters formatted for input to crop models contained within agricultural DSS.

Once the data was collected it was scrutinized and organized so that it can be useful for the project.

The data which was available to us was only from year 1997 to 2016 which is total of 20 years.

4.4 Building the first predictive Model

After the parameters were understood and required data was collected and scrutinized, we built the first predictive model A.

Since it was primitive model, we only used rainfall data and divided to convert it into weekly per day average rainfall/precipitation from the month of December to February when the rainfall was affecting the Wheat crop maximum.

Result History | ExampleSet (//DES project/Model A)

Data | Statistics | Visualizations | Annotations

Open in Turbo Prep | Auto Model | Filter (20 / 20 examples): all

Row No.	Year	Production	W1	W2	W3	W4	W5	W6	W7	W
1	1997	170400	0	0.006	0	0	0	0.047	0.136	0
2	1998	197926	1.874	1.953	0	0.021	0	0.120	0	0
3	1999	185096	0	0	0	0	0.367	0.001	0.001	0.
4	2000	111612	0	0	0	0	0	0.109	0	0.
5	2001	116887	0	0	0	0.049	0.073	0	0.003	0
6	2002	82843	0	0	0	0	0	0	0.663	0.
7	2003	130252	0.033	0.009	0	1.337	0	0	0	0.
8	2004	78218	0	0.017	0.986	0.233	0.006	0	0.561	0.
9	2005	90111	0	0	0	0.006	0	0	0.064	0.
10	2006	118521	0	0	0	0.013	0.001	0	0	0
11	2007	118179	0.009	0	0	0.015	0	0	0	0.
12	2008	164400	0.193	0.103	0	0	0	0.006	0	0
13	2009	213165	0.003	0	0.009	0	0	0.056	0.034	0
14	2010	291368	0	0	0.093	0.080	0.196	0.350	0	0
15	2011	230402	0	0	0	0.595	0	0	0	0
16	2012	229909	0	0	0	0.052	0.731	0	0.010	0
17	2013	218377	0	0.010	0	0	0	0	0.506	0
18	2014	271260	0	0	0	0.799	0.229	0	2.810	0

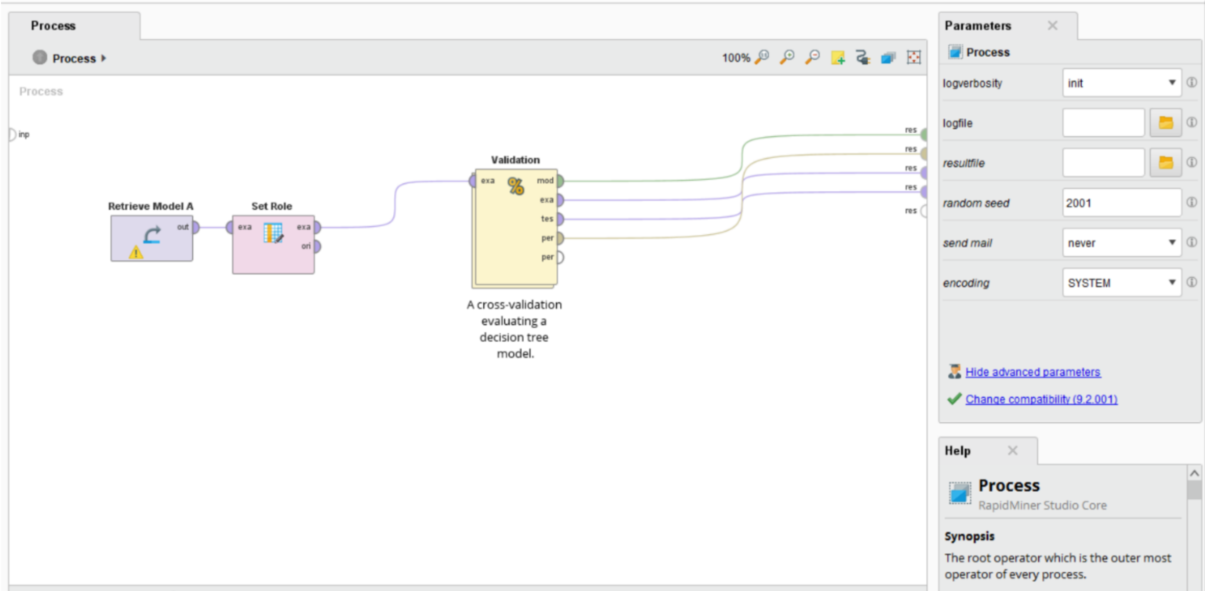
ExampleSet (20 examples, 2 special attributes, 13 regular attributes)

(Figure A)

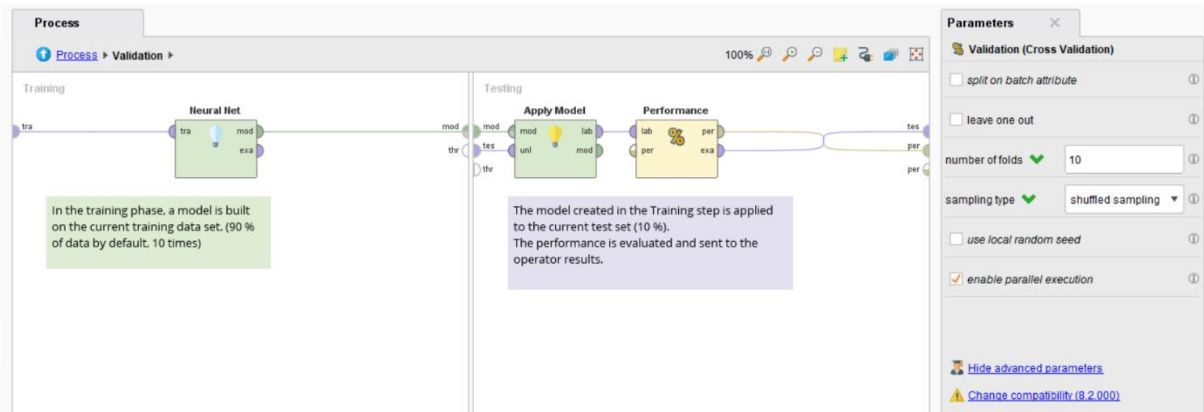
The figure A shows the inputs used in the model A with W1, W2, W3, etc representing weeks starting from November month. The Rainfall data is in mm per day.

The year column is in Blue representing it as the ID and Production is highlighted in green representing it to be the Label or special attribute.

The modelling process created in the software was:



(Figure B: Model process 1 for Model A)



(Figure B: Model process 1 for Model A, validation using Neural Net)

The neural network (Neural Net) is defined as follows according to the RapidMiner Official platform:

This operator learns a model by means of a feed-forward neural network trained by a back propagation algorithm (multi-layer perceptron). This operator cannot handle polynominal attributes.

A feed-forward neural network is an artificial neural network where there is no direct cycle of links between the units. In this network, the data transmits from the input nodes to the output nodes in only one manner, forward, through the concealed nodes (if any). The network does not have cycles.

Back propagation algorithm is a monitored technique of learning that can be split into two stages: propagation and updating of weight. The two stages are repeated until the network performance is sufficiently great. In back propagation algorithms, to calculate the value of some predefined error function, the output values are contrasted with the right response. The error is then supplied back through the network by different methods. The algorithm adjusts the weights of each link using this data in order to decrease the error function value by a tiny quantity. Usually, after repeating this process for a sufficiently big amount of training cycles, the network converges to some state where the calculation mistake is low. In such cases, we would say that the network developed has learned a certain target function.

A multilayer perceptron (MLP) is an artificial neural network feed-forward model that maps input information sets to a suitable output set. In a directed graph, an MLP comprises of various layers of nodes with each layer fully linked to the next. Each node is a neuron (or processing component) with a nonlinear activation function except for the input nodes. MLP uses propagation of the back type to train the network. This network class comprises of various layers of computer units, generally interconnected in a feed-forward manner. The units of these networks use a sigmoid function as an activation function in many applications.

Usual sigmoid function is used in this operator as the activation function. Consequently, attribute values ranges should be scaled to-1 and+ 1. This can be achieved through the parameter of normalization. The output node sort is sigmoid when the learning information defines a task of classification and linear when the learning information defines a job of numerical regression.

In the model A:

In the training phase, the model is built on the current training data set. (90 % of data by default, 10 times). The model created in the Training step is applied to the current test set (10 %). The performance is evaluated and sent to the operator results. The number of the training set is 200 and learning rate is 0.01 with total momentum of 0.9 to avoid local maxima and minima.

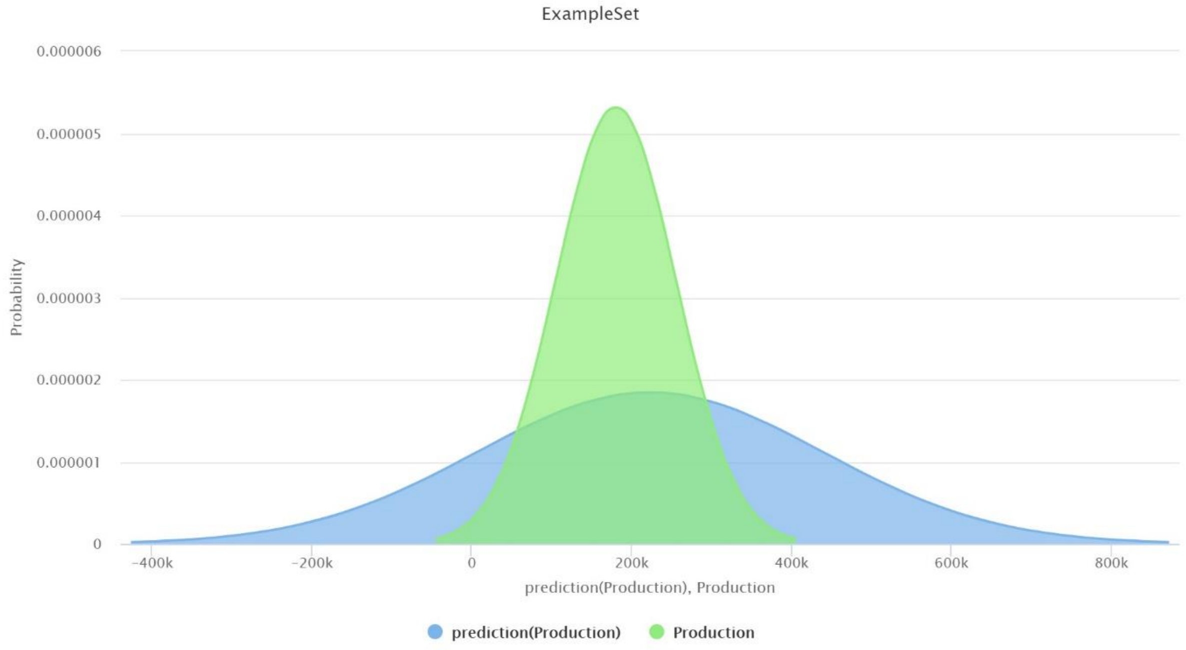
After the model was built and parameters were adjusted it was time to run the test model A.

4.4.1 Testing the Model A

When the model was tested following prediction result was obtained:

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	223280.009	0	0	0	0	0.367	0.001	0.
2	2012	229909	298272.753	0	0	0	0.052	0.731	0	0.
3	2005	90111	148675.787	0	0	0	0.006	0	0	0.
4	2016	350076	136178.714	0.359	0	0	0	0	0.001	0.
5	2003	130252	475900.082	0.033	0.009	0	1.337	0	0	0
6	2008	164400	186648.650	0.193	0.103	0	0	0	0.006	0
7	2009	213165	166032.417	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	93850.557	0	0	0	0.595	0	0	0
9	1998	197926	718430.027	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	132081.562	0	0	0	0.049	0.073	0	0.
11	2002	82843	139234.407	0	0	0	0	0	0	0.
12	2004	78218	652169.368	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	181722.924	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	-107279.473	0	0.010	0	0	0	0	0.
15	2014	271260	500839.784	0	0	0	0.799	0.229	0	2.
16	2015	230484	-140616.306	0	1.310	0	0.001	0.937	0.051	0.
17	2006	118521	158995.448	0	0	0	0.013	0.001	0	0

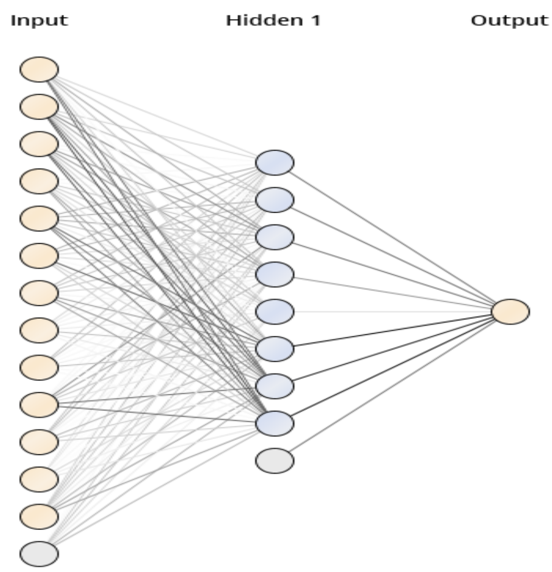
To better understand the difference in the prediction and the actual values we can analyse the bell curve between the production (actual) and production (prediction):



We can see that the prediction values are accurate when the production was in normal or average limits. Any high difference in production, be it too much lower or higher made it difficult for the model to predict accurately.

The root mean squared error for this model was:

```
Performance Vector:
root_mean_squared_error: 198353.786 +/- 135814.022 (micro average:
236527.225 +/- 0.000)
squared_error: 55945128231.700 +/- 59281957443.106 (micro average:
55945128231.700 +/- 92003049624.768)
```



(Figure A: Improved Neural Network for model A)

5. Improving and refining Models

Since we had developed the first model, we could now focus on improving the model and using other Machine Learning methods for better accuracy and quick results.

5.1 Improving upon existing model A

We modified the neural net operator to see if the accuracy of other operators were higher or lower. These operators were different machine learning algorithms with different abilities to learn and perform.

5.1.1 Deep Learning

Deep Learning is based on a multi-layer artificial neural feed-forward network that is trained using back-propagation with stochastic gradient descent. The network may comprise a big number of hidden layers of neurons with tanh, rectifier and activation functions for maxout. High predictive precision is enabled by advanced characteristics such as adaptive learning rate, rate annealing, momentum training, dropout and L1 or L2 regularization. Each compute node trains multi-threading (asynchronously) a copy of the worldwide model parameters on its local information and adds periodically to the worldwide model via network-wide model average.

The operator begins and operates the algorithm on a 1-node local H2O cluster. Even if one node is used, the execution is parallel. By altering the setting Settings / Preferences / General / Number of threads, you can set the amount of parallelism. It utilizes the system's suggested amount of threads by default. Only one cluster example is beginning and will continue to run until you close RapidMiner Studio.

The followings are the important parameters in the Deep Learning Model-

activation: The activation function (non-linearity) to be used by the neurons in the hidden layers.

Tanh: Hyperbolic tangent function (same as scaled and shifted sigmoid).

Rectifier: Rectifier Linear Unit: Chooses the maximum of (0, x) where x is the input value

Maxout: Choose the maximum coordinate of the input vector.

ExpRectifier: Exponential Rectifier Linear Unit function.

1. **hidden layer sizes**

The number and size of each hidden layer in the model. For example, if a user specifies "100,200,100" a model with 3 hidden layers will be produced, and the middle-hidden layer will have 200 neurons.

2. **epochs**

How many times the dataset should be iterated (streamed), can be fractional.

3. **epsilon**

Similar to learning rate annealing during initial training and momentum at later stages where it allows forward progress. Typical values are between 1e-10 and 1e-4. This parameter is only active if adaptive learning rate is enabled.

We can call model A which was updated with predictive model of deep learning as Model B with its first iteration Model B 1.0:

Model B 1.0 has following specifics:

Activation: Rectifier type

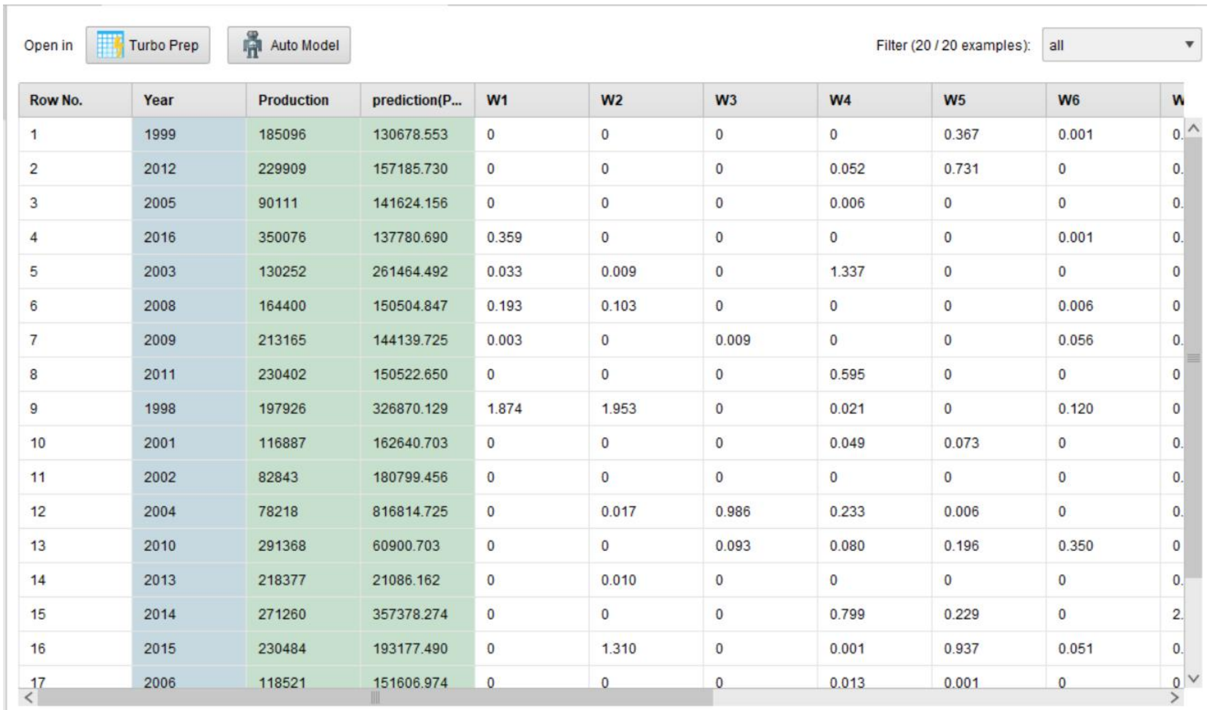
Epochs: 10.0

Epsilon: 1.0E-8

Hidden Layer Size: 50

The Data was split 90-10% between training and testing respectively.

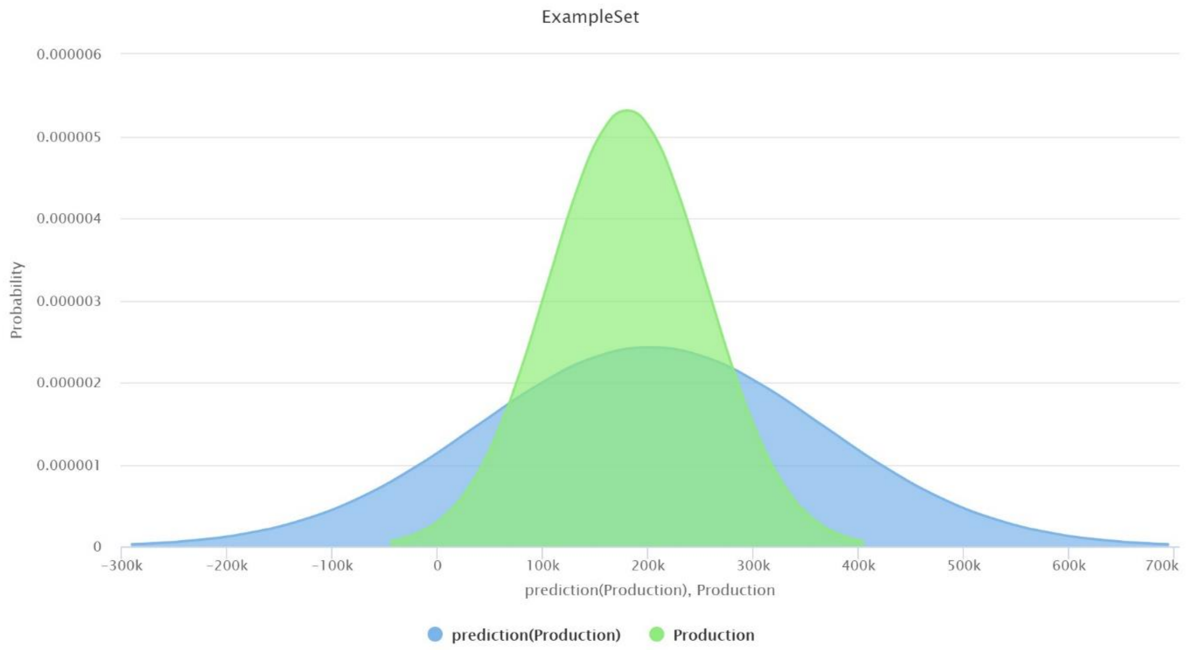
The prediction output was:



Open in Turbo Prep Auto Model Filter (20 / 20 examples): all

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	130678.553	0	0	0	0	0.367	0.001	0.
2	2012	229909	157185.730	0	0	0	0.052	0.731	0	0.
3	2005	90111	141624.156	0	0	0	0.006	0	0	0.
4	2016	350076	137780.690	0.359	0	0	0	0	0.001	0.
5	2003	130252	261464.492	0.033	0.009	0	1.337	0	0	0
6	2008	164400	150504.847	0.193	0.103	0	0	0	0.006	0
7	2009	213165	144139.725	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	150522.650	0	0	0	0.595	0	0	0
9	1998	197926	326870.129	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	162640.703	0	0	0	0.049	0.073	0	0.
11	2002	82843	180799.456	0	0	0	0	0	0	0.
12	2004	78218	816814.725	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	60900.703	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	21086.162	0	0.010	0	0	0	0	0.
15	2014	271260	357378.274	0	0	0	0.799	0.229	0	2.
16	2015	230484	193177.490	0	1.310	0	0.001	0.937	0.051	0.
17	2006	118521	151606.974	0	0	0	0.013	0.001	0	0

To have a better visualization we can see the bell curve between prediction and production:



The predictive capability of the model has been significantly increased in this model as the prediction values are more overlapping with actual production of wheat.

The performance of the model was:

```
Performance Vector:
root_mean_squared_error: 115200.938 +/- 103262.995 (micro average:
151222.279 +/- 0.000)
squared_error: 22868177613.217 +/- 37941759413.415 (micro average:
22868177613.217 +/- 50976909612.016)
```

For further improvement in the next iteration of the Model B that is Model B 1.2 we change the activation type to Tanh. This change in the activation type from Rectifier to Tanh which is hyperbolic tangent function resulted in higher accuracy in prediction.

The bell curve between prediction and production actually show positive results here.

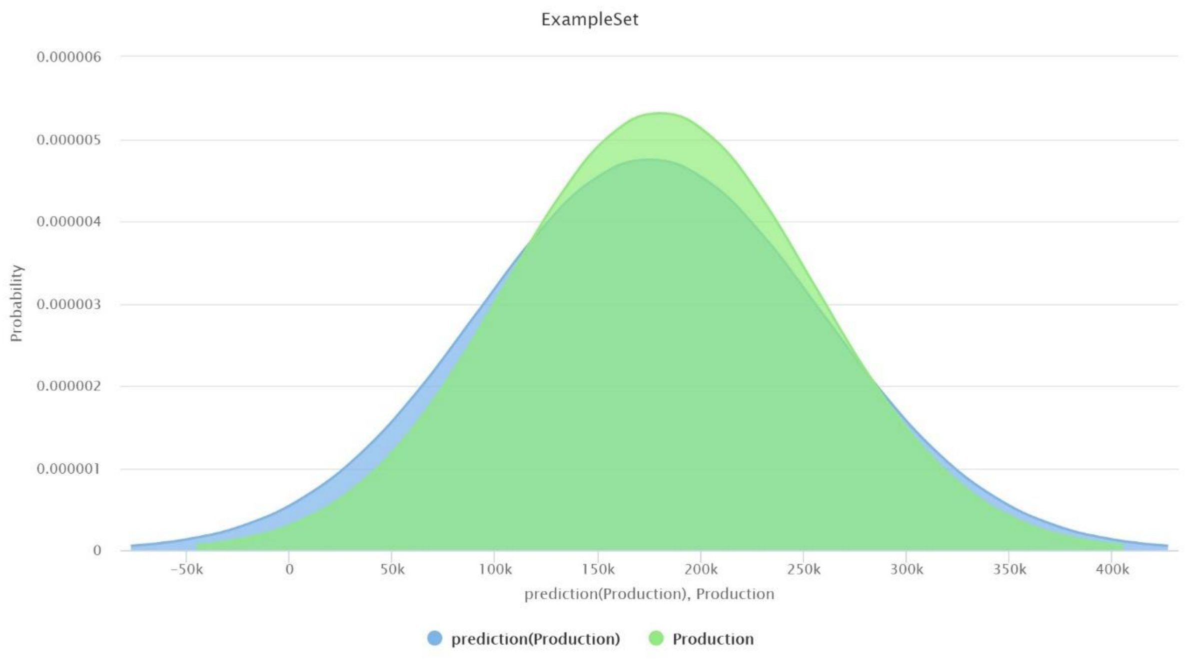
The following are the outputs from the process for model B 1.2:

Open in Turbo Prep Auto Model Filter (20 / 20 examples): all

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	134650.941	0	0	0	0	0.367	0.001	0.
2	2012	229909	213318.027	0	0	0	0.052	0.731	0	0.
3	2005	90111	137078.806	0	0	0	0.006	0	0	0.
4	2016	350076	133051.795	0.359	0	0	0	0	0.001	0.
5	2003	130252	299718.744	0.033	0.009	0	1.337	0	0	0
6	2008	164400	154902.471	0.193	0.103	0	0	0	0.006	0
7	2009	213165	150122.936	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	123066.471	0	0	0	0.595	0	0	0
9	1998	197926	416629.317	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	128964.162	0	0	0	0.049	0.073	0	0.
11	2002	82843	88761.440	0	0	0	0	0	0	0.
12	2004	78218	299891.680	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	189933.889	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	38344.974	0	0.010	0	0	0	0	0.
15	2014	271260	168140.936	0	0	0	0.799	0.229	0	2.
16	2015	230484	186937.069	0	1.310	0	0.001	0.937	0.051	0.
17	2006	118521	157230.239	0	0	0	0.013	0.001	0	0

ExampleSet (20 examples, 3 special attributes, 13 regular attributes)

(Fig A: Value table for prediction and production)



(Fig A: Bell curve for prediction and production)

The performance for this iteration was:

```
Performance Vector:
root_mean_squared_error: 103073.484 +/- 50400.674 (micro average:
113623.713 +/- 0.000)
squared_error: 12910348223.163 +/- 10005994634.543 (micro average:
12910348223.163 +/- 17184900988.554)
```

The micro average of the RMSE value is decreased in this model. This means this model of Deep Learning with use of the Tanh as the activation type can be useful and more accurate than the initial Rectifier method.

We also tried other methods of Maxout and ExpRectifier for the process but the accuracy was too low to make them of any use in the prediction process.

5.1.2 Lazy Algorithm

In all examples, the Default Model operator generates a model that predicts the default value specified for the label. You can select the technique to use to generate a default value through the parameter of the procedure. The default value can be median or average label values for a numeric label, or a continuous default value can be defined via the constant parameter. The label mode may be used for nominal values. An attribute's values can be used as projections; the attribute can be chosen via the parameter of the attribute. This operator should not be used to perform ' real ' forecast duties, but can be used to compare with guessing the outcomes of ' real ' teaching systems.

The outputs from this machine learning algorithms is also very close to actual values.

Open in Turbo Prep Auto Model Filter (20 / 20 examples): all

Row No.	Year	Production	W1	W2	W3	W4	W5	W6	W7	W
1	1997	170400	0	0.006	0	0	0	0.047	0.136	0
2	1998	197926	1.874	1.953	0	0.021	0	0.120	0	0
3	1999	185096	0	0	0	0	0.367	0.001	0.001	0.
4	2000	111612	0	0	0	0	0	0.109	0	0.
5	2001	116887	0	0	0	0.049	0.073	0	0.003	0
6	2002	82843	0	0	0	0	0	0	0.663	0.
7	2003	130252	0.033	0.009	0	1.337	0	0	0	0.
8	2004	78218	0	0.017	0.986	0.233	0.006	0	0.561	0.
9	2005	90111	0	0	0	0.006	0	0	0.064	0.
10	2006	118521	0	0	0	0.013	0.001	0	0	0
11	2007	118179	0.009	0	0	0.015	0	0	0	0.
12	2008	164400	0.193	0.103	0	0	0	0.006	0	0
13	2009	213165	0.003	0	0.009	0	0	0.056	0.034	0
14	2010	291368	0	0	0.093	0.080	0.196	0.350	0	0
15	2011	230402	0	0	0	0.595	0	0	0	0
16	2012	229909	0	0	0	0.052	0.731	0	0.010	0
17	2013	218377	0	0.010	0	0	0	0	0.506	0

The accuracy of the model is all time best with lowest RMS Error value up until all models:

```
Performance Vector:
root_mean_squared_error: 76862.586 +/- 30691.230 (micro average: 82192.540
+/- 0.000)
squared_error: 6755613627.600 +/- 5405837725.554 (micro average:
6755613627.600 +/- 6462425362.740)
```

5.1.3 Decision Tree

A choice tree is a tree-like set of nodes designed to produce a class-affiliated value judgment or a numerical target value estimate. For a particular attribute, each node reflects a dividing rule. This rule separates values belonging to distinct classes for classification, divides them for regression in order to optimally decrease the mistake for the chosen criterion of the parameter.

The construction of fresh nodes will be repeated until the requirements for stopping are met. A class label attribute prediction is determined based on the majority of examples that reached this leaf during generation, whereas an estimate for a numerical value is achieved by averaging the values in a leaf.

This Operator can process ExampleSets comprising attributes both nominal and numerical. The label attribute must be classification nominal and regression numerical.

Using the Apply Model Operator, the decision tree model can be implemented after generation to fresh examples. In accordance with the dividing rule, each Example follows the tree branches until a leaf is reached.

Differentiation:

CHAID

The CHAID Operator offers a pruned decision tree using criteria based on chi-squared rather than information gain or ratio gain. On ExampleSets with numerical attributes, this Operator cannot be implemented, but only nominal attributes.

ID3

The ID3 Operator offers a fundamental unpruned decision tree application. It only operates with nominal attributes for ExampleSets.

Random Forest

The Operator of Random Forests generates several random trees on various subsets of Examples. The resulting model is based on all of these trees voting. Because of this distinction, overtraining is less susceptible.

Bagging

Bootstrap aggregating (bagging) is a meta-algorithm for machine learning to enhance the stability and classification precision of classification and regression models. It also decreases variance and helps prevent' overfitting.' Although it is generally used for decision tree designs, it can be used with any model form.

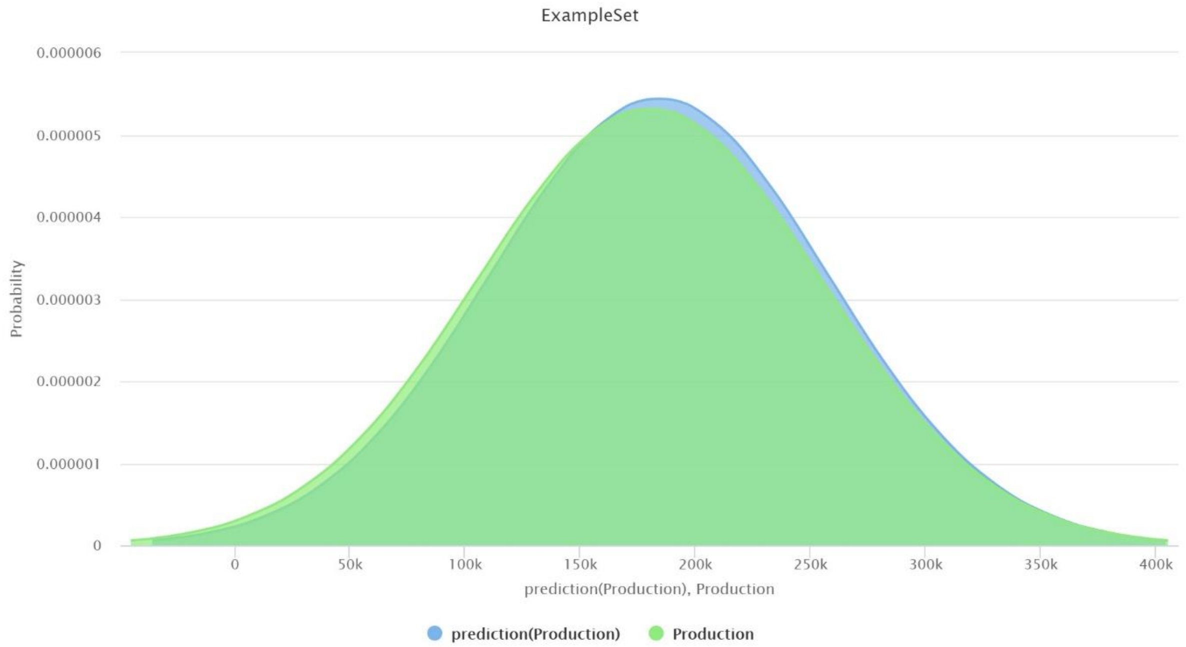
We decided to use Decision Tree for the further iteration of the model and we had great success in accurate prediction by bringing the RMSE down.

The accuracy of this model can be assessed by the RMSE value which was:

```
Performance Vector:
root_mean_squared_error: 108341.068 +/- 40248.377 (micro average:
114872.649 +/- 0.000)
squared_error: 13195725600.631 +/- 9050986609.529 (micro average:
13195725600.631 +/- 11975890515.289)
```

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6
1	1999	185096	310668	0	0	0	0	0.367	0.001
2	2012	229909	117862.333	0	0	0	0.052	0.731	0
3	2005	90111	120014.333	0	0	0	0.006	0	0
4	2016	350076	177575.333	0.359	0	0	0	0	0.001
5	2003	130252	251042.200	0.033	0.009	0	1.337	0	0
6	2008	164400	117862.333	0.193	0.103	0	0	0	0.006
7	2009	213165	100861.500	0.003	0	0.009	0	0	0.056
8	2011	230402	117862.333	0	0	0	0.595	0	0
9	1998	197926	247553.333	1.874	1.953	0	0.021	0	0.120
10	2001	116887	247553.333	0	0	0	0.049	0.073	0
11	2002	82843	310668	0	0	0	0	0	0
12	2004	78218	206420.333	0	0.017	0.986	0.233	0.006	0
13	2010	291368	141006	0	0	0.093	0.080	0.196	0.350
14	2013	218377	310668	0	0.010	0	0	0	0
15	2014	271260	157674	0	0	0	0.799	0.229	0
16	2015	230484	83724	0	1.310	0	0.001	0.937	0.051
17	2006	118521	192357.667	0	0	0	0.013	0.001	0

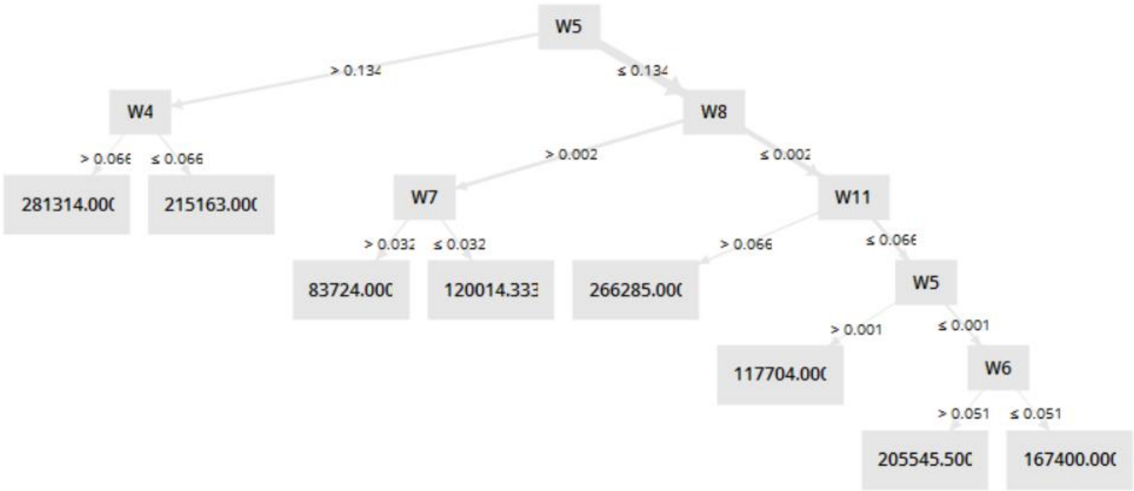
(Fig A: Value table for prediction and production)



(Fig A: Bell curve for prediction and production)

One can clearly see the accuracy of the model by overlapping of prediction and production values in the model (green and blue in the graph).

In this decision tree we can clearly see that the most important factor that affects the prediction is the value of average rainfall / precipitation in the week 5, i.e. first week of December.



(FIGURE A: decision tree)

5.2 Adding more variables: Model C

We had very assuring results with only the rainfall data. Now it was time to add other weather variables that could affect the accuracy of prediction as well. These variables were added and subtracted in various iterations and the final model was C, with following attributes:

1. rainfall data from week 1 of November to week 4 of February.
2. Specific humidity Monthly average.
3. Solar irradiation
4. Maximum temperature average
5. Minimum temperature average.

W10	Real	0	Min 0	Max 4.281	Average 0.655
W11	Real	0	Min 0	Max 4.459	Average 0.439
W12	Real	0	Min 0	Max 4.167	Average 0.323
Oct to Jan av Rainfall	Real	0	Min 0.140	Max 10.370	Average 2.144
Specific Humidity Mothly Avera...	Real	0	Min 0.003	Max 0.006	Average 0.005
solar irradiation	Real	0	Min 7.583	Max 8.158	Average 7.837
Max temp monthly av	Real	0	Min 28.288	Max 32.055	Average 30.659
min temp month av	Real	0	Min 12.353	Max 14.970	Average 13.733

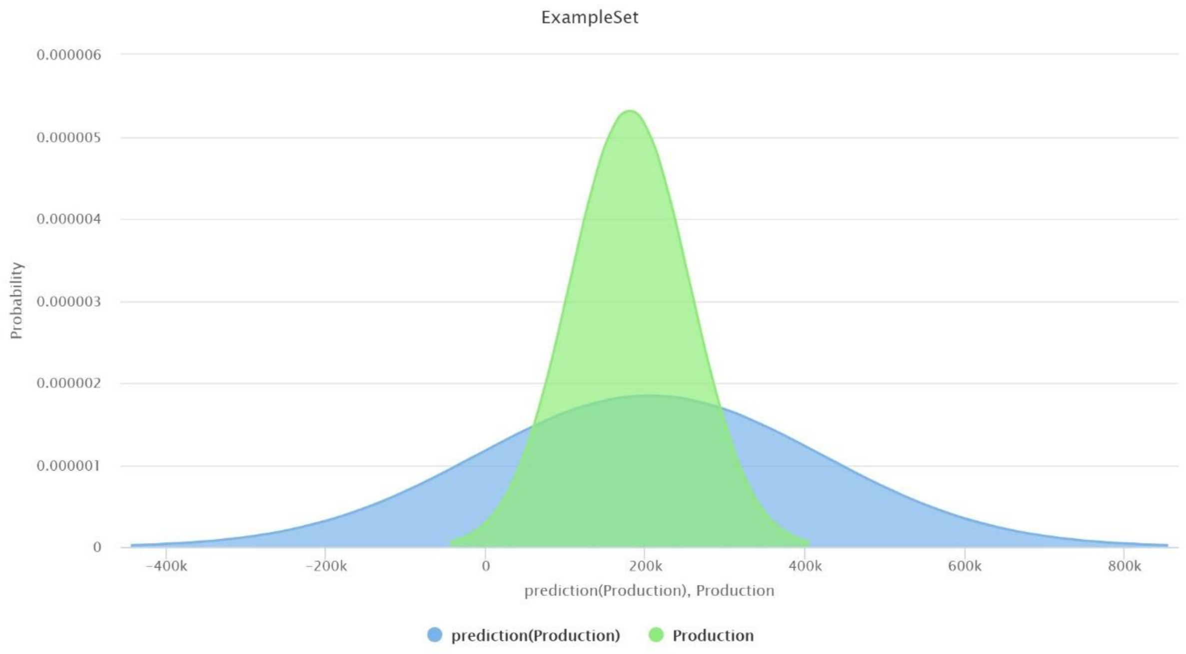
(Figure: Some of the attributes in the model C)

5.2.1 Neural Net

Firstly, we used the neural network operator for prediction in this model C with more variables.

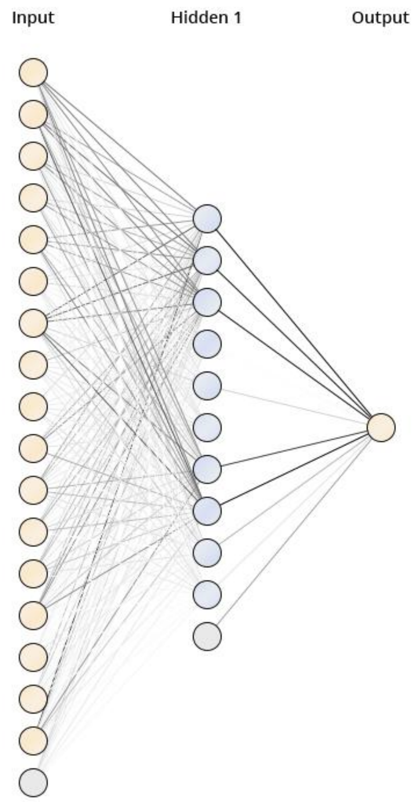
Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	170809.600	0	0	0	0	0.367	0.001	0.
2	2012	229909	298590.993	0	0	0	0.052	0.731	0	0.
3	2005	90111	193536.117	0	0	0	0.006	0	0	0.
4	2016	350076	167957.791	0.359	0	0	0	0	0.001	0.
5	2003	130252	584458.257	0.033	0.009	0	1.337	0	0	0
6	2008	164400	142779.972	0.193	0.103	0	0	0	0.006	0
7	2009	213165	209635.439	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	101247.961	0	0	0	0.595	0	0	0
9	1998	197926	649184.431	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	103650.502	0	0	0	0.049	0.073	0	0.
11	2002	82843	94661.116	0	0	0	0	0	0	0.
12	2004	78218	587702.207	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	377625.185	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	-252539.271	0	0.010	0	0	0	0	0.
15	2014	271260	245063.139	0	0	0	0.799	0.229	0	2.
16	2015	230484	-13538.777	0	1.310	0	0.001	0.937	0.051	0.

(Figure A: value table for Neural net)



(Figure B: Bell Curve for model C neural net)

This is the improved neural network with input nodes layer and hidden node layers with output nodes layer.



For the detailed information of this neural network please see appendix A.

The performance of the model was:

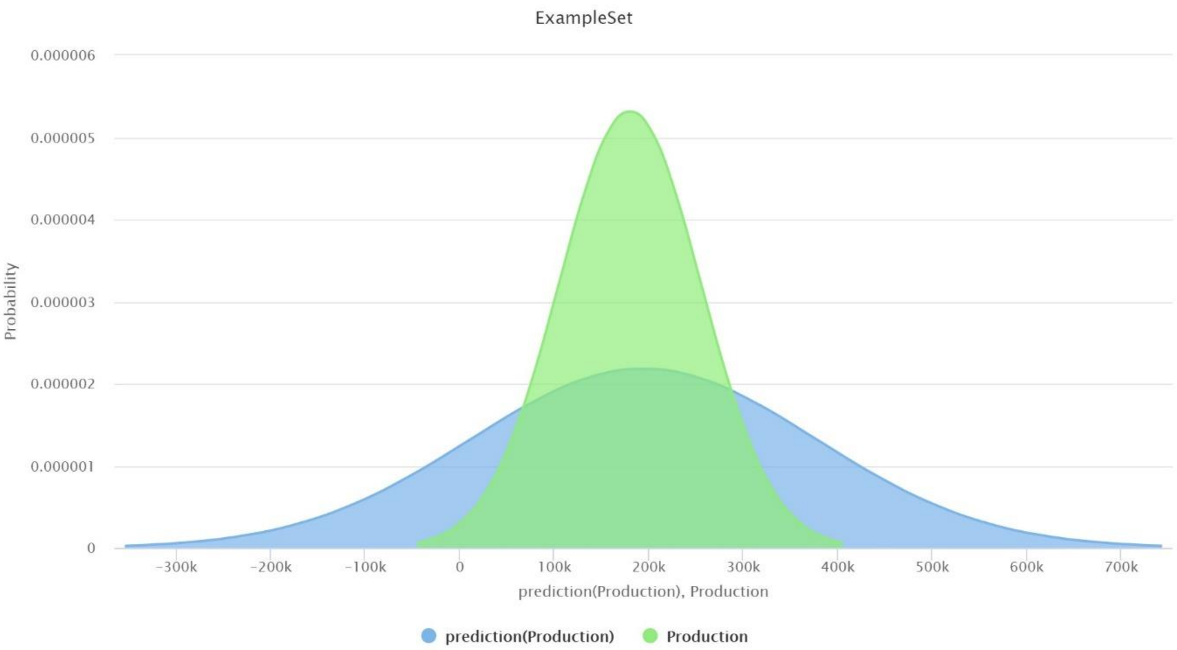
```
Performance Vector:
root_mean_squared_error: 200435.058 +/- 121570.733 (micro average:
231248.160 +/- 0.000)
squared_error: 53475711339.175 +/- 51910851524.078 (micro average:
53475711339.175 +/- 86476531768.963
```

We decide to decrease the number of the Cycles in Neural network and had following results:

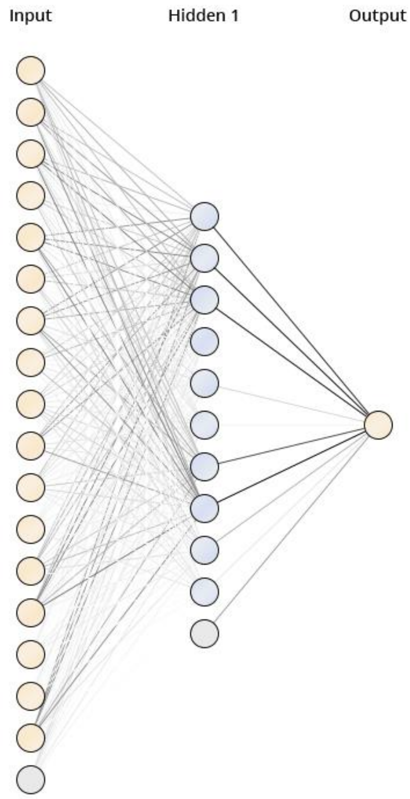
Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	152063.032	0	0	0	0	0.367	0.001	0.
2	2012	229909	255535.311	0	0	0	0.052	0.731	0	0.
3	2005	90111	187322.811	0	0	0	0.006	0	0	0.
4	2016	350076	160408.051	0.359	0	0	0	0	0.001	0.
5	2003	130252	391250.267	0.033	0.009	0	1.337	0	0	0
6	2008	164400	143174.488	0.193	0.103	0	0	0	0.006	0
7	2009	213165	210771.053	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	163000.781	0	0	0	0.595	0	0	0
9	1998	197926	624835.571	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	106585.421	0	0	0	0.049	0.073	0	0.
11	2002	82843	77473.031	0	0	0	0	0	0	0.
12	2004	78218	550934.495	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	364994.433	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	-215203.044	0	0.010	0	0	0	0	0.
15	2014	271260	49451.108	0	0	0	0.799	0.229	0	2.
16	2015	230484	74786.601	0	1.310	0	0.001	0.937	0.051	0.

ExampleSet (20 examples, 3 special attributes, 17 regular attributes)

(Value table model C neural network with training cycles 100)



(Bell curve for model C neural network with training cycles 100)



Improved Neural network with training cycles 100

Performance of the model:

```
PerformanceVector:
root_mean_squared_error: 169969.697 +/- 114799.844 (micro average:
201868.278 +/- 0.000)
squared_error: 40750801642.214 +/- 43017800882.218 (micro average:
40750801642.214 +/- 68748909145.798)
```

For detailed information please see Appendix A (ModC:NN2)

5.2.2 Deep Learning on Model C

When we applied deep learning operator in Model C on data, we acquired following results:

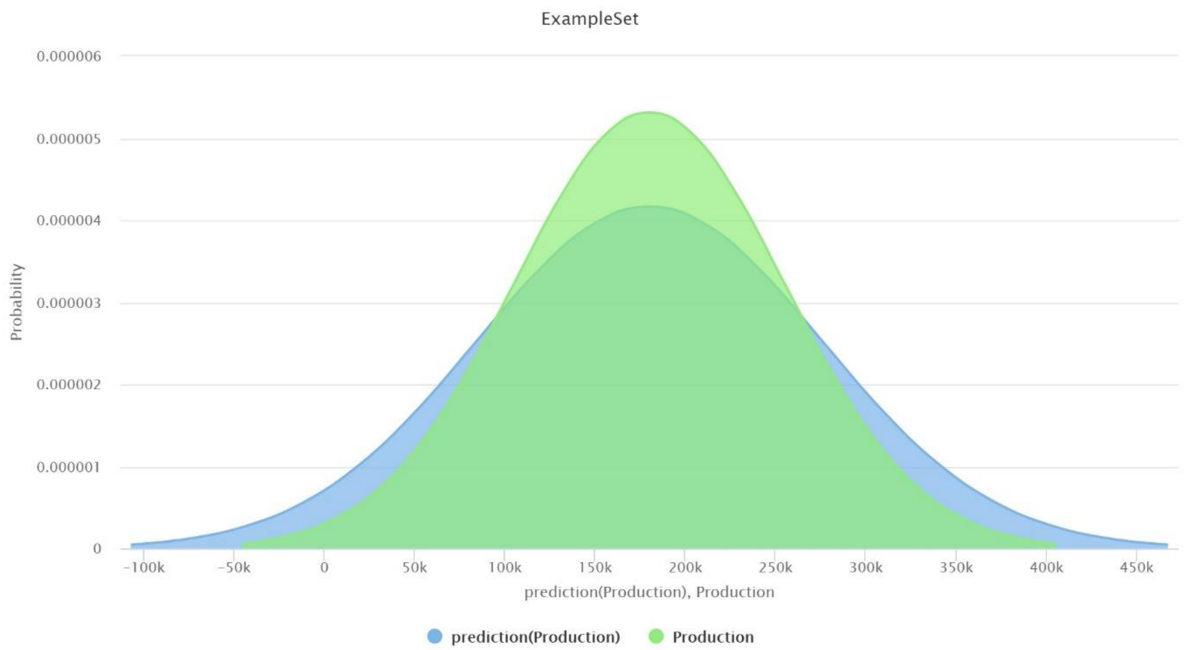
Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	177970.385	0	0	0	0	0.367	0.001	0.
2	2012	229909	181698.354	0	0	0	0.052	0.731	0	0.
3	2005	90111	161954.480	0	0	0	0.006	0	0	0.
4	2016	350076	158224.910	0.359	0	0	0	0	0.001	0.
5	2003	130252	120449.595	0.033	0.009	0	1.337	0	0	0
6	2008	164400	146030.732	0.193	0.103	0	0	0	0.006	0
7	2009	213165	186245.576	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	198026.253	0	0	0	0.595	0	0	0
9	1998	197926	474508.602	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	119229.362	0	0	0	0.049	0.073	0	0.
11	2002	82843	167371.930	0	0	0	0	0	0	0.
12	2004	78218	377224.808	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	247945.185	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	41838.290	0	0.010	0	0	0	0	0.
15	2014	271260	120062.379	0	0	0	0.799	0.229	0	2.
16	2015	230484	86354.725	0	1.310	0	0.001	0.937	0.051	0.

(Figure A: value table for Deep Learning model)

We can see that the values are aligned with the actual values but we can still make it further accurate and to create a more reliable model we have to change the activation function. We will have to change the rectifier activation function with Tanh activation function.

More information can be found in the appendix (ModC:DL1)

You can see in the following bell curve that the predictions are in accordance with the actual production and quite accurate.



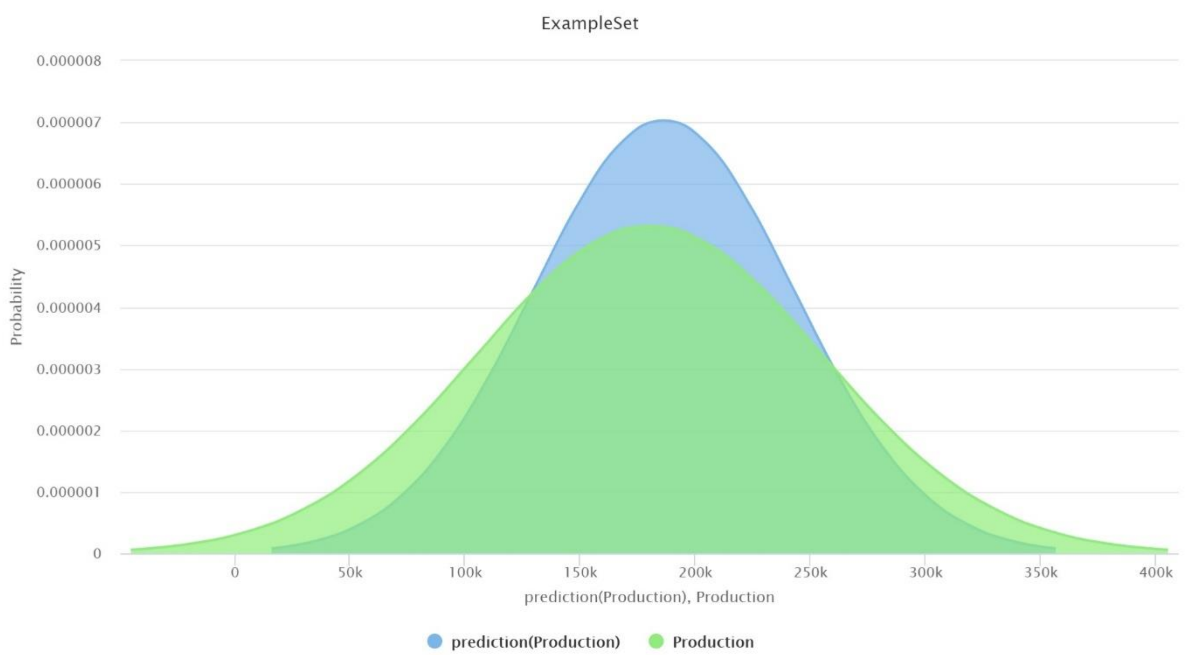
```
The performance was:

PerformanceVector:
root_mean_squared_error: 99089.102 +/- 76984.190 (micro average: 123095.732 +/- 0.000)
squared_error: 15152559127.887 +/- 17314327101.568 (micro average: 15152559127.887 +/- 25142870913.153
```

Let us change the activation function to tanh:

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	151367.144	0	0	0	0	0.367	0.001	0
2	2012	229909	247344.708	0	0	0	0.052	0.731	0	0
3	2005	90111	198885.638	0	0	0	0.006	0	0	0
4	2016	350076	167958.925	0.359	0	0	0	0	0.001	0
5	2003	130252	234710.460	0.033	0.009	0	1.337	0	0	0
6	2008	164400	134434.327	0.193	0.103	0	0	0	0.006	0
7	2009	213165	193163.953	0.003	0	0.009	0	0	0.056	0
8	2011	230402	206895.782	0	0	0	0.595	0	0	0
9	1998	197926	366856.618	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	116666.099	0	0	0	0.049	0.073	0	0
11	2002	82843	123686.122	0	0	0	0	0	0	0
12	2004	78218	188173.828	0	0.017	0.986	0.233	0.006	0	0
13	2010	291368	200485.616	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	103523.523	0	0.010	0	0	0	0	0
15	2014	271260	213491.258	0	0	0	0.799	0.229	0	2
16	2015	230484	158572.480	0	1.310	0	0.001	0.937	0.051	0

The performance of the model was:

```
PerformanceVector:
root_mean_squared_error: 75253.877 +/- 40482.253 (micro average: 84487.144 +/- 0.000)
squared_error: 7138077549.791 +/- 6935422081.633 (micro average: 7138077549.791 +/- 9021666980.616)
```



This is the bell curve for model C with deep learning applying Tanh as the activation function.

For more information, go to appendix A[ModC:DL2].

5.2.3 Lazy Algorithm on model C

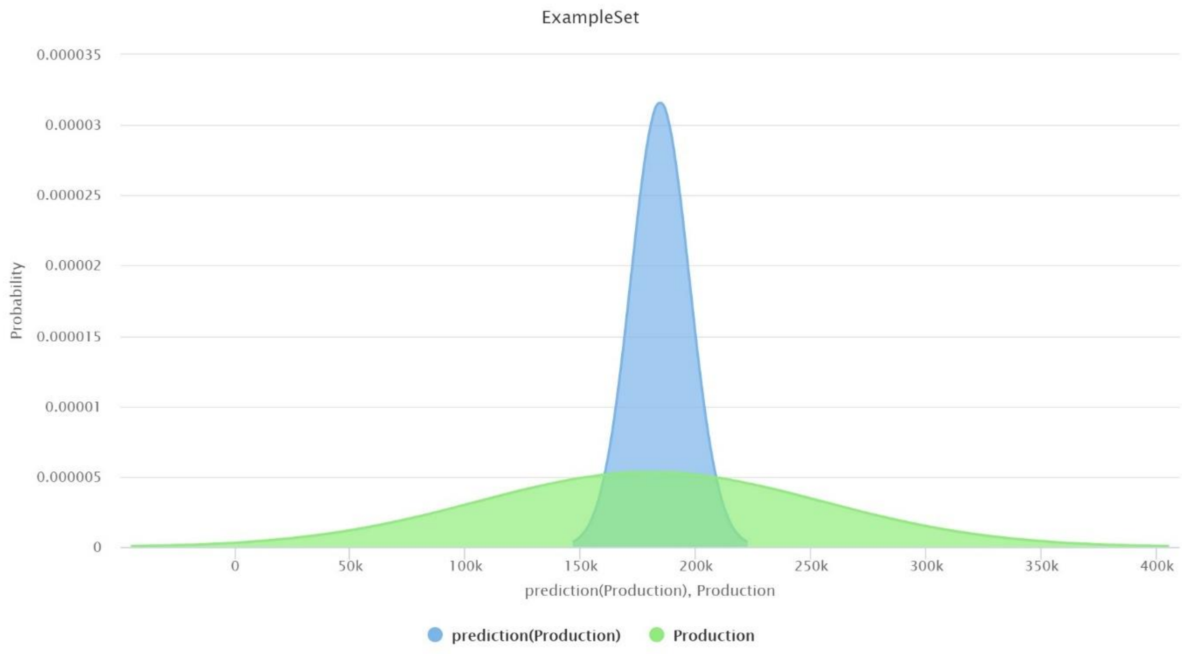
Now we apply default Lazy method operator on Model C and get the following results.

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	170400	0	0	0	0	0.367	0.001	0.
2	2012	229909	170400	0	0	0	0.052	0.731	0	0.
3	2005	90111	185096	0	0	0	0.006	0	0	0.
4	2016	350076	185096	0.359	0	0	0	0	0.001	0.
5	2003	130252	197926	0.033	0.009	0	1.337	0	0	0
6	2008	164400	197926	0.193	0.103	0	0	0	0.006	0
7	2009	213165	170400	0.003	0	0.009	0	0	0.056	0.
8	2011	230402	170400	0	0	0	0.595	0	0	0
9	1998	197926	185096	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	185096	0	0	0	0.049	0.073	0	0.
11	2002	82843	197926	0	0	0	0	0	0	0.
12	2004	78218	197926	0	0.017	0.986	0.233	0.006	0	0.
13	2010	291368	170400	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	170400	0	0.010	0	0	0	0	0.
15	2014	271260	170400	0	0	0	0.799	0.229	0	2.
16	2015	230484	170400	0	1.310	0	0.001	0.937	0.051	0

You can see that the values in this model of lazy algorithm are close to actual values and the predictive computation model is more reliable.

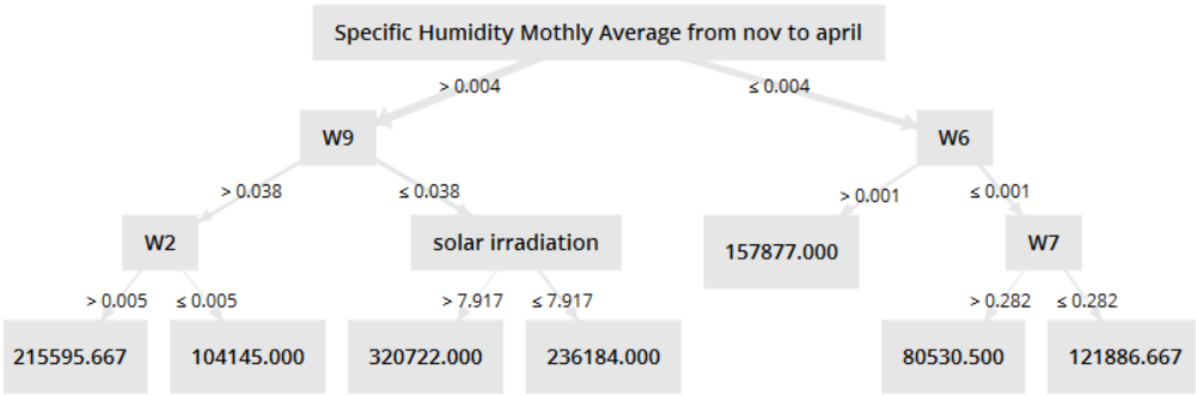
The performance was:

```
PerformanceVector:
root_mean_squared_error: 76862.586 +/- 30691.230 (micro average: 82192.540
+/- 0.000)
squared_error: 6755613627.600 +/- 5405837725.554 (micro average:
6755613627.600 +/- 6462425362.740)

Default (prediction model for label Production)
default value: 185096.0


Following is the bell curve for the model:
```



5.2.4 Decision tree on Model C

We applied decision tree operator on model C with following results:

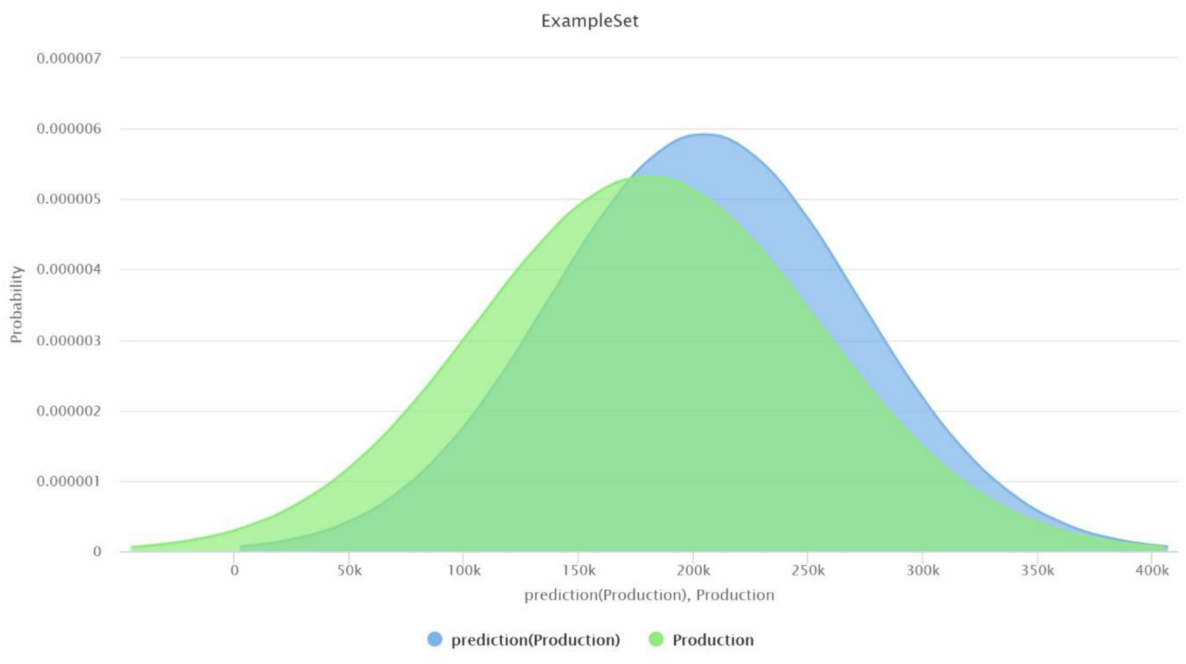


We can see that the most influencing factor for decision tree is the specific humidity for predicting the production of wheat.

Row No.	Year	Production	prediction(P...	W1	W2	W3	W4	W5	W6	W
1	1999	185096	119318	0	0	0	0	0.367	0.001	0
2	2012	229909	238275.667	0	0	0	0.052	0.731	0	0
3	2005	90111	230265	0	0	0	0.006	0	0	0
4	2016	350076	157877	0.359	0	0	0	0	0.001	0
5	2003	130252	320722	0.033	0.009	0	1.337	0	0	0
6	2008	164400	115529.500	0.193	0.103	0	0	0	0.006	0
7	2009	213165	320722	0.003	0	0.009	0	0	0.056	0
8	2011	230402	229591.200	0	0	0	0.595	0	0	0
9	1998	197926	224430.500	1.874	1.953	0	0.021	0	0.120	0
10	2001	116887	177748	0	0	0	0.049	0.073	0	0
11	2002	82843	114790	0	0	0	0	0	0	0
12	2004	78218	114790	0	0.017	0.986	0.233	0.006	0	0
13	2010	291368	213858.333	0	0	0.093	0.080	0.196	0.350	0
14	2013	218377	270411.750	0	0.010	0	0	0	0	0
15	2014	271260	226229.333	0	0	0	0.799	0.229	0	2
16	2015	230484	184163	0	1.310	0	0.001	0.937	0.051	0

The above is the value table and we can see the values are close to actual production. Hence this model is quite successful in predicting the production of wheat.

Following bell curve gives more accurate view:



For more information see appendix A [ModC:DT1]

The performance was:

```
PerformanceVector:
root_mean_squared_error: 81447.182 +/- 50090.671 (micro average: 94296.401
+/- 0.000)
squared_error: 8891811209.152 +/- 10093320601.362 (micro average:
8891811209.152 +/- 13067795689.730)
```

6. Making predictions with models developed for next year

In this section we will apply the different models developed and compare it with actual production in agricultural year 2017-2018 with the weather variable data available.

The production of the wheat crop in the year 2017-18 officially was:

251725 Metric Tone

We will now compare this value with prediction from different models we developed.

Machine Learning Algorithm Used	Value Predicted	Difference
Decision tree	236184.0	15,541
Lazy Algorithm (Default Operator)	185096.0	66,629
Deep learning (Rectifier)	240302.21	11,422.79
Deep learning (Tanh)	249496.67	2,228.33
Neural Net (200 training cycles)	270391.64	-18,666.64
Neural Net (100 training cycles)	253733.40	-2,008.4

We can see that Neural Net (100 training cycles) and deep learning (Tanh) gives most accurate prediction with only difference of 2008.4 and 2,228.33 Mt. ton.

Thus, out of all models Neural Network and Deep Learning method can be considered for future predictions in this regard.

7. CONCLUSION AND WAY FORWARD

Following are some conclusions we can draw from this study:

1. Specific humidity is a very vital data for prediction in wheat production as it has one of the highest correlations.
2. Artificial Neural Network with low training cycles (100) work best for low data-sets.
3. Deep Learning when used with Tanh Activation function gives fairly accurate results as compared to Rectifier method.
4. Increasing number of training cycles on low amount of data-set can increase outliers.

5. Week 5, that is the first month of January is also very important in prediction of crop yield and hence can be stated as most effective month for rainfall to mark a change in production of wheat in that year.
6. More years of data availability can help in better model development.

In this report, the performances of some machine learning algorithms are examined to predict crop yields in Rajasthan (Sawai Madhopur, in particular). In the training stage, there was only a mild mistake in the neural network and deep learning model showing the precision of this model. Furthermore, the comparison between the expected and the neural net result demonstrates the comparable result which further proves the precision of the forecast values using this model of the neural network. The predicted outcome can be utilized for enhancing the wheat yield in Sawai Madhopur district and can help in policy making. Future work includes the investigation of other Machine Learning (especially neural network) architectures including feedforward, acyclic and modular neural network and comparison of their performances. In addition, by using search algorithms such as the genetic algorithm, the appropriate input parameters, appropriate number of hidden layers and number of neurons can be selected to enhance prediction accuracy.

Appendix A

[ModC:NN1] Values for weights and biases in the model C: neural network.

```
Hidden 1
========

Node 1 (Sigmoid)
----------------
W1: 0.588
W2: -0.395
W3: -0.194
W4: -0.185
W5: 0.427
W6: 0.154
W7: 0.655
W8: 0.075
W9: 0.148
W10: -0.114
W11: 0.370
W12: -0.097
Oct to Jan av Rainfall: 0.149
Specific Humidity Mothly Average from nov to april: 0.230
solar irradiation: 0.051
Max temp monthly av: 0.264
min temp month av: 0.808
Bias: -0.024

Node 2 (Sigmoid)
----------------
W1: -0.571
W2: 0.645
W3: 0.420
W4: 0.356
W5: -0.442
W6: -0.111
W7: -0.665
W8: 0.064
W9: -0.008
W10: 0.339
W11: -0.277
W12: 0.244
Oct to Jan av Rainfall: -0.291
Specific Humidity Mothly Average from nov to april: -0.434
solar irradiation: 0.084
Max temp monthly av: -0.003
min temp month av: -0.283
Bias: -0.082

Node 3 (Sigmoid)
----------------
W1: -0.505
W2: 0.573
W3: 0.507
W4: 0.455
W5: -0.418
W6: -0.177
W7: -0.541
W8: 0.252
W9: 0.009
W10: 0.438
W11: -0.157
```

```
W12: 0.327
Oct to Jan av Rainfall: -0.262
Specific Humidity Mothly Average from nov to april: -0.471
solar irradiation: 0.022
Max temp monthly av: 0.221
min temp month av: -0.123
Bias: -0.191

Node 4 (Sigmoid)
----------------
W1: 0.092
W2: 0.046
W3: 0.135
W4: 0.065
W5: -0.020
W6: 0.047
W7: 0.049
W8: 0.072
W9: 0.085
W10: 0.083
W11: 0.096
W12: 0.080
Oct to Jan av Rainfall: 0.024
Specific Humidity Mothly Average from nov to april: -0.112
solar irradiation: -0.013
Max temp monthly av: -0.025
min temp month av: -0.065
Bias: -0.122

Node 5 (Sigmoid)
----------------
W1: 0.113
W2: -0.028
W3: 0.058
W4: 0.026
W5: 0.117
W6: 0.044
W7: 0.191
W8: 0.083
W9: 0.073
W10: 0.055
W11: 0.083
W12: -0.028
Oct to Jan av Rainfall: 0.098
Specific Humidity Mothly Average from nov to april: -0.043
solar irradiation: -0.019
Max temp monthly av: -0.031
min temp month av: -0.010
Bias: -0.088

Node 6 (Sigmoid)
----------------
W1: 0.069
W2: 0.092
W3: 0.106
W4: 0.122
W5: 0.007
W6: -0.013
W7: 0.060
W8: 0.105
W9: 0.101
W10: 0.119
W11: 0.084
W12: 0.066
```

```
Oct to Jan av Rainfall: -0.019
Specific Humidity Mothly Average from nov to april: -0.101
solar irradiation: -0.001
Max temp monthly av: -0.029
min temp month av: -0.077
Bias: -0.067

Node 7 (Sigmoid)
----------------
W1: 0.458
W2: -0.390
W3: -0.125
W4: -0.165
W5: 0.315
W6: 0.103
W7: 0.522
W8: 0.071
W9: 0.096
W10: -0.039
W11: 0.333
W12: -0.172
Oct to Jan av Rainfall: 0.201
Specific Humidity Mothly Average from nov to april: 0.250
solar irradiation: -0.034
Max temp monthly av: 0.097
min temp month av: 0.447
Bias: -0.003

Node 8 (Sigmoid)
----------------
W1: -0.538
W2: 0.727
W3: 0.468
W4: 0.381
W5: -0.483
W6: -0.141
W7: -0.682
W8: 0.134
W9: -0.018
W10: 0.350
W11: -0.287
W12: 0.299
Oct to Jan av Rainfall: -0.237
Specific Humidity Mothly Average from nov to april: -0.478
solar irradiation: 0.103
Max temp monthly av: 0.050
min temp month av: -0.228
Bias: -0.092

Node 9 (Sigmoid)
----------------
W1: 0.220
W2: -0.089
W3: 0.014
W4: -0.033
W5: 0.139
W6: 0.063
W7: 0.233
W8: 0.058
W9: 0.133
W10: -0.006
W11: 0.138
W12: 0.005
Oct to Jan av Rainfall: 0.124
```

```
Specific Humidity Mothly Average from nov to april: 0.057
solar irradiation: -0.071
Max temp monthly av: 0.040
min temp month av: 0.082
Bias: -0.057

Node 10 (Sigmoid)
-----------------
W1: 0.094
W2: 0.020
W3: 0.099
W4: 0.003
W5: 0.039
W6: 0.054
W7: 0.142
W8: 0.087
W9: 0.109
W10: 0.053
W11: 0.095
W12: 0.085
Oct to Jan av Rainfall: 0.040
Specific Humidity Mothly Average from nov to april: -0.009
solar irradiation: -0.016
Max temp monthly av: 0.017
min temp month av: -0.025
Bias: -0.034


Output
======

Regression (Linear)
-------------------
Node 1: 1.071
Node 2: -1.026
Node 3: -1.026
Node 4: 0.019
Node 5: 0.193
Node 6: -0.009
Node 7: 0.850
Node 8: -1.089
Node 9: 0.322
Node 10: 0.114
Threshold: 0.402
```

[ModC:NN2] Model c with Neural network with 100 cycles

```
Hidden 1
========

Node 1 (Sigmoid)
----------------
W1: 0.222
W2: -0.223
W3: -0.258
W4: -0.054
W5: 0.380
```

```
W6: 0.199
W7: 0.390
W8: -0.068
W9: 0.178
W10: -0.174
W11: 0.222
W12: 0.039
Oct to Jan av Rainfall: 0.170
Specific Humidity Mothly Average from nov to april: 0.236
solar irradiation: -0.060
Max temp monthly av: 0.037
min temp month av: 0.385
Bias: 0.025

Node 2 (Sigmoid)
----------------
W1: -0.286
W2: 0.360
W3: 0.435
W4: 0.198
W5: -0.432
W6: -0.141
W7: -0.377
W8: 0.162
W9: -0.117
W10: 0.346
W11: -0.110
W12: 0.022
Oct to Jan av Rainfall: -0.265
Specific Humidity Mothly Average from nov to april: -0.445
solar irradiation: 0.064
Max temp monthly av: -0.006
min temp month av: -0.403
Bias: -0.080

Node 3 (Sigmoid)
----------------
W1: -0.231
W2: 0.315
W3: 0.469
W4: 0.208
W5: -0.421
W6: -0.224
W7: -0.323
W8: 0.256
W9: -0.155
W10: 0.359
W11: -0.113
W12: 0.068
Oct to Jan av Rainfall: -0.237
Specific Humidity Mothly Average from nov to april: -0.440
solar irradiation: 0.027
Max temp monthly av: 0.099
min temp month av: -0.402
Bias: -0.134

Node 4 (Sigmoid)
----------------
W1: 0.071
W2: 0.025
W3: 0.111
W4: 0.041
W5: -0.040
W6: 0.026
```

```
W7: 0.027
W8: 0.050
W9: 0.065
W10: 0.060
W11: 0.071
W12: 0.058
Oct to Jan av Rainfall: 0.002
Specific Humidity Mothly Average from nov to april: -0.112
solar irradiation: -0.011
Max temp monthly av: -0.016
min temp month av: -0.056
Bias: -0.098

Node 5 (Sigmoid)
----------------
W1: 0.062
W2: 0.001
W3: 0.033
W4: 0.032
W5: 0.099
W6: 0.031
W7: 0.127
W8: 0.043
W9: 0.068
W10: 0.031
W11: 0.036
W12: -0.011
Oct to Jan av Rainfall: 0.079
Specific Humidity Mothly Average from nov to april: -0.043
solar irradiation: -0.013
Max temp monthly av: -0.021
min temp month av: 0.020
Bias: -0.067

Node 6 (Sigmoid)
----------------
W1: 0.053
W2: 0.065
W3: 0.083
W4: 0.093
W5: -0.013
W6: -0.036
W7: 0.043
W8: 0.085
W9: 0.079
W10: 0.096
W11: 0.062
W12: 0.038
Oct to Jan av Rainfall: -0.041
Specific Humidity Mothly Average from nov to april: -0.100
solar irradiation: 0.000
Max temp monthly av: -0.022
min temp month av: -0.073
Bias: -0.043

Node 7 (Sigmoid)
----------------
W1: 0.229
W2: -0.199
W3: -0.160
W4: -0.068
W5: 0.302
W6: 0.141
W7: 0.308
```

```
W8: -0.040
W9: 0.137
W10: -0.084
W11: 0.191
W12: -0.029
Oct to Jan av Rainfall: 0.196
Specific Humidity Mothly Average from nov to april: 0.259
solar irradiation: -0.055
Max temp monthly av: -0.022
min temp month av: 0.279
Bias: 0.018

Node 8 (Sigmoid)
----------------
W1: -0.263
W2: 0.404
W3: 0.472
W4: 0.192
W5: -0.456
W6: -0.182
W7: -0.388
W8: 0.220
W9: -0.156
W10: 0.336
W11: -0.141
W12: 0.053
Oct to Jan av Rainfall: -0.235
Specific Humidity Mothly Average from nov to april: -0.488
solar irradiation: 0.061
Max temp monthly av: 0.018
min temp month av: -0.411
Bias: -0.077

Node 9 (Sigmoid)
----------------
W1: 0.144
W2: -0.015
W3: -0.011
W4: -0.004
W5: 0.124
W6: 0.062
W7: 0.136
W8: 0.006
W9: 0.139
W10: -0.031
W11: 0.072
W12: 0.053
Oct to Jan av Rainfall: 0.116
Specific Humidity Mothly Average from nov to april: 0.057
solar irradiation: -0.066
Max temp monthly av: 0.036
min temp month av: 0.095
Bias: -0.039

Node 10 (Sigmoid)
-----------------
W1: 0.051
W2: 0.027
W3: 0.072
W4: -0.005
W5: 0.017
W6: 0.035
W7: 0.092
W8: 0.054
```

```
W9: 0.096
W10: 0.029
W11: 0.055
W12: 0.085
Oct to Jan av Rainfall: 0.019
Specific Humidity Mothly Average from nov to april: -0.008
solar irradiation: -0.011
Max temp monthly av: 0.027
min temp month av: -0.002
Bias: -0.010


Output
======

Regression (Linear)
-------------------
Node 1: 0.799
Node 2: -0.886
Node 3: -0.886
Node 4: -0.012
Node 5: 0.144
Node 6: -0.031
Node 7: 0.674
Node 8: -0.944
Node 9: 0.281
Node 10: 0.090
Threshold: 0.354
```

[ModC:DL1] Model c with deep learning with rectifier method of activation

```
Model Metrics Type: Regression
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning-198272
 frame id: rm-h2o-frame-deep_learning-527269
 MSE: 1.54007706E9
 R^2: 0.7124257
 mean residual deviance: 1.54007706E9
Status of Neuron Layers (predicting Production, regression, gaussian
distribution, Quadratic loss, 3,501 weights/biases, 47.1 KB, 200 training
samples, mini-batch size 1):
 Layer Units      Type Dropout       L1       L2 Mean Rate Rate RMS
Momentum Mean Weight Weight RMS Mean Bias Bias RMS
     1    17     Input  0.00 %
     2    50 Rectifier  0.00 % 0.000010 0.000000  0.010377 0.009228
0.000000   0.003796   0.177770  0.499215 0.009478
     3    50 Rectifier  0.00 % 0.000010 0.000000  0.009699 0.012970
0.000000  -0.004272   0.137888  0.999605 0.004826
     4     1    Linear          0.000010 0.000000  0.000984 0.000706
0.000000  -0.006276   0.211285  0.000236 0.000000
Scoring History:
           Timestamp   Duration Training Speed   Epochs Iterations
Samples     Training MSE Training Deviance Training R^2
 2019-07-12 10:48:50  0.000 sec                 0.00000          0
0.000000              NaN               NaN            NaN
 2019-07-12 10:48:50  0.007 sec 20000 rows/sec  1.00000          1
20.000000 4535427890.10247  4535427890.10247      0.15311
 2019-07-12 10:48:50  0.021 sec 15384 rows/sec 10.00000         10
200.000000 1540077048.96737  1540077048.96737      0.71243
```

```
H2O version: 3.8.2.6-rm9.0.0
```

[ModC:DL2]

```
Model Metrics Type: Regression
 Description: Metrics reported on full training frame
 model id: rm-h2o-model-deep_learning-961713
 frame id: rm-h2o-frame-deep_learning-503816
 MSE: 1.6788521E9
 R^2: 0.68651265
 mean residual deviance: 1.6788521E9
Status of Neuron Layers (predicting Production, regression, gaussian
distribution, Quadratic loss, 3,501 weights/biases, 47.0 KB, 200 training
samples, mini-batch size 1):
 Layer Units    Type Dropout       L1        L2 Mean Rate Rate RMS Momentum
Mean Weight Weight RMS Mean Bias Bias RMS
     1    17  Input  0.00 %
     2    50   Tanh  0.00 % 0.000010 0.000000  0.015886 0.015908 0.000000
-0.004068   0.181963  0.001540 0.010288
     3    50   Tanh  0.00 % 0.000010 0.000000  0.033636 0.054843 0.000000
0.006488   0.139103 -0.001926 0.009138
     4     1 Linear         0.000010 0.000000  0.001957 0.000486 0.000000
-0.056913   0.184811  0.005279 0.000000
Scoring History:
           Timestamp   Duration Training Speed   Epochs Iterations
Samples     Training MSE Training Deviance Training R^2
 2019-07-12 10:54:40  0.000 sec                 0.00000          0
0.000000              NaN               NaN           NaN
 2019-07-12 10:54:40  0.008 sec 10000 rows/sec  1.00000          1
20.000000 5518327456.48999  5518327456.48999     -0.03042
 2019-07-12 10:54:40  0.028 sec 10000 rows/sec 10.00000         10
200.000000 1678852146.90012  1678852146.90012      0.68651

H2O version: 3.8.2.6-rm9.0.0
```

[ModC:DT1]

```
 Specific Humidity Mothly Average from nov to april > 0.004
|   W9 > 0.038
|   |   W2 > 0.005: 215595.667 {count=3}
|   |   W2 ≤ 0.005: 104145.000 {count=2}
|   W9 ≤ 0.038
|   |   solar irradiation > 7.917: 320722.000 {count=2}
|   |   solar irradiation ≤ 7.917: 236184.000 {count=4}
Specific Humidity Mothly Average from nov to april ≤ 0.004
|   W6 > 0.001: 157877.000 {count=4}
|   W6 ≤ 0.001
|   |   W7 > 0.282: 80530.500 {count=2}
|   |   W7 ≤ 0.282: 121886.667 {count=3
```

Bibliography

1. “Departmental Information.” Department, plan.rajasthan.gov.in/content/planning-portal/en/des/about-us/profile-department-division/department.html.
2. Wikipedia contributors. (2019, July 8). Agriculture in India. In Wikipedia, The Free Encyclopedia. Retrieved 06:08, July 9, 2019, from https://en.wikipedia.org/w/index.php?title=Agriculture_in_India&oldid=905319433
3. Hinton, Geoffrey. “How Neural Networks Learn from Experience.” Scientific American, www.academia.edu/631731/How_neural_networks_learn_from_experience.
4. Gershenson, and Carlos. “Artificial Neural Networks for Beginners.” ArXiv.org, 20 Aug. 2003, arxiv.org/abs/cs/0308031.
5. Menon, A., Mehrotra, K., Mohan, C. K., & Ranka, S. (1996). Characterization of a class of sigmoid functions with applications to neural networks. Neural Networks, 9(5), 819-835.
6. Prasanna, S. M., Gupta, C. S., & Yegnanarayana, B. (2006). Extraction of speaker-specific excitation information from linear prediction residual of speech. Speech Communication, 48(10), 1243-1261.
7. Hassoun, M. H. (1995). Fundamentals of artificial neural networks. MIT press.
8. Wikipedia contributors. (2019, July 8). RapidMiner. In Wikipedia, The Free Encyclopedia. Retrieved 06:09, July 9, 2019, from https://en.wikipedia.org/w/index.php?title=RapidMiner&oldid=905327409
9. David Norris, “RapidMiner - a potential game changer,” Bloor Research, November 13, 2013.
10. (n.d.). Retrieved from https://power.larc.nasa.gov/data-access-viewer/
11. Rainfed Agriculture in India. (n.d.). Retrieved from https://www.gktoday.in/gk/rainfed-agriculture-in-india/
12. Sawai Madhopur. (2019, June 16). Retrieved from https://en.wikipedia.org/wiki/Sawai_Madhopur
13.