Predicting Online News popularity Using Neural Network

Thesis Submitted in Partial Fulfilment of the Requirement for the
Award Of The Degree of

MASTER OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

Under the Esteemed guidance of
Mr. MANOJ KUMAR
(Associate Professor)
Delhi Technological University

Submitted by
RAJJAT SODHI
2k16/cse/09

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

SESSION 2016-2018

TABLE OF CONTENTS

FIGURES AND TABLES

CERTIFICATE

This is to certify that Major Project-II Report entitled **“Predicting Online News popularity Using Neural Network”** submitted by **Rajjat Sodhi, Roll No. 2K16/CSE/09** for partial fulfilment of the requirement for the award of degree Master of Technology (Computer Science and Engineering) is a record of the candidate work carried out by her under my supervision.

Mr. Manoj Kumar
Department Of Computer Science & Engineering
Delhi Technological University

DECLARATION

I hereby declare that the major Project-II work entitled "**Predicting Online News popularity Using Neural Network**" which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of degree of Master Of Technology (Computer Science and Engineering) is a bonafide report of Major Project-II carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Rajjat Sodhi
2K16/CSE/09

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Mr. Manoj Kumar for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. Rajni Jindal, HOD, Computer Science & Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out. Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Rajjat Sodhi
University Roll no: 2K16/CSE/09
M.Tech (Computer Science & Engineering)
Department of Computer Science and Engineering
Delhi Technological University
Delhi – 110042

ABSTRACT

In this era with the rapid growth of internet, smartphone and other gadgets, everybody enjoys reading, sharing online news. We can direct relate online news popularity with the number of shares, number of comments and number of likes of that news that means popularity is directly proportional to number of share, comment and likes.

In this project, Our Goal is to find out the best suitable model to predict the popularity of online news, using artificial neural network. Artificial neural network have several advantage over machine learning algorithm, artificial neural networks have some interesting properties that made these family of machine learning algorithms very appealing when confronting difficult patter-discovery tasks. Artificial neural network has two kind, one is feed forward unidirectional ANN and other is feedback cycle ANN (Back propagation). ANN with Back propagation (BP) learning algorithm is widely used in solving various classification and forecasting problems our data comes from Mashable, a well-known online news website. Our dataset consist of 40K rows and 15 input parameter and 1 output parameter.

Artificial Neural Network turns out to be the best approach for prediction, goal of this thesis to predict the number of share of news article and it can achieve an accuracy with optimal parameters. Our work can help online news companies to predict news popularity before publication.

.

Keyword: Artificial neural network, Back Propagation.

CHAPTER 1- INTRODUCTION

1.1 Artificial Neural Network

Artificial Neural Networks or (ANNs) are software implementations of the neuronal structure of our brains. Without inferring deep into complex biology of our brain structures, it is adequate to say that the brain contains neurons which are kind of organic switches. These neurons can change their output state depending on the strength of their input which could be in electrical or chemical form. The neural network in a human brain is vast interconnected network of neurons, where the output signals of any given neuron may be the result of input signals to thousands of other neurons. Learning happens by repeatedly activating or preferring particular neural connections over others, and this emphasizes those connections. This will make them more likely to generate a desired outcome given some certain specified input. This learning involves feedback – when the desired outcome is produced, the neural connections causing that output becomes stronger.

Artificial neural networks attempt to interpret and imitate this particular behaviour of brain. These neural networks can be trained in supervised or unsupervised manner. In a Supervised ANN, matched input and output data samples are used to train the network, with the objective of producing the ANN to provide a desired output for some given input. For instance in an E-mail spam filter one could take the count of various words in the body of the email as the input training data, and taxonomy of whether the e-mail was exactly spam or not as the output training data. If a lot of samples of E-mails are analysed through the neural network this allows the network to determine what kind of input data produces it likely that certain e-mail is spam or not.

1.2 THE STRUCTURE OF AN ANN

1.2.1 The Artificial Neuron

An activation function simulates the biological neuron in an ANN .In classification tasks such as identifying spam in e-mails this activation function must have a “switch on” characteristic or in other words, once the input is greater than a particular value, the outcome should change its state i.e. from 0 to >0, from 0 to 1 or from -1 to 1. This produces the “turning on” of a biological neuron. A widely used activation function is the sigmoid function:

:

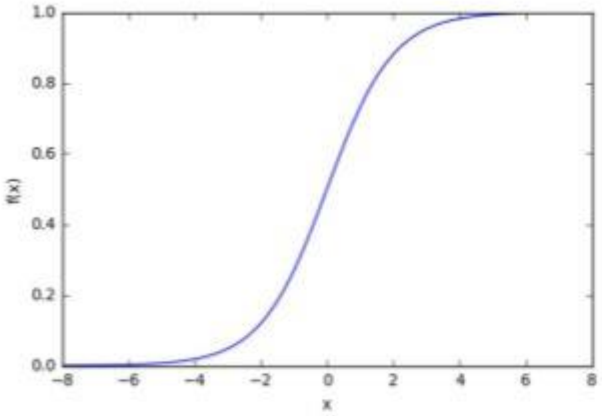


Fig.1.1

As shown in above diagram, the function is “activated” i.e. it changes from 0 to 1 when the input ‘x’ becomes greater than some certain value. The sigmoid function is however not a step function, the edge is not sharp and the output changes gradually. From this we can conclude that there should be some derivative of the function which is significant for the training algorithm

1.2.2 Nodes:

Biological neurons are connected hierarchical networks, where the outcome of some neurons being the inputs to others. These networks can be represented as connected layers of nodes, where each node takes more than one weighted inputs, applies the activation function to the summation of these inputs, and hence generating an output. This can be broken further, but to help things along let's consider the following diagram:

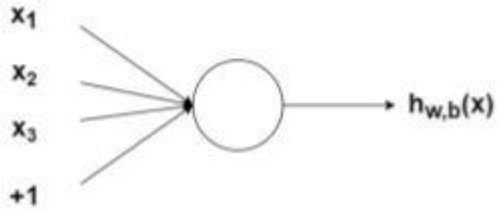


Fig.1.2

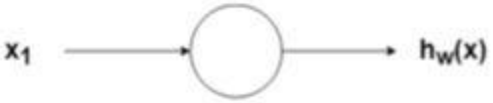
The circle in the above image shows the node. The node is considered as the "seat" of the activation function, and has the weighted inputs, adds them, then put them as input to the activation function. The outcome of the activation function is shown as 'h' in the figure. Note: A node as shown above is also called a perceptron in some compositions.

The weights mentioned above are real valued numbers (i.e. not binary 0s or 1s), and these are multiplied by the inputs and later added up in the node.

.

1.2.3 The Bias:

Let’s consider a simple node, with only one input and one output:



In above example the input to the activation function of the node is simple .Now what does changing do in this simple network?

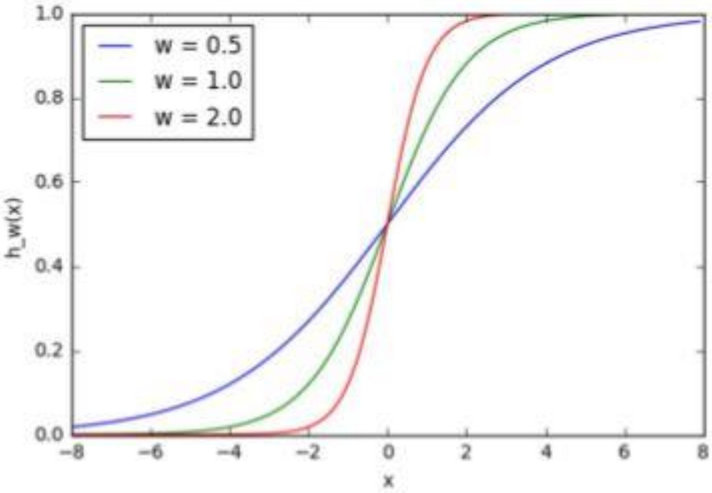


Fig.1.3

Here we can notice that changing the weightage also changes the slope of the outcome of the sigmoid activation function, this is significant if we want to model different strengths of relationships between the input and output variables. However, what if the output is required to change when x is greater than 1? This is where the bias comes in. Let us take the very same network with a bias input:

Here we can see that changing the weight changes the slope of the output of the sigmoid activation function, which is obviously useful if we want to model different strengths of relationships between the input and output variables. However, what if we only want the output to change when x is greater than 1? This is where the bias comes in – let’s consider the same network with a bias input:

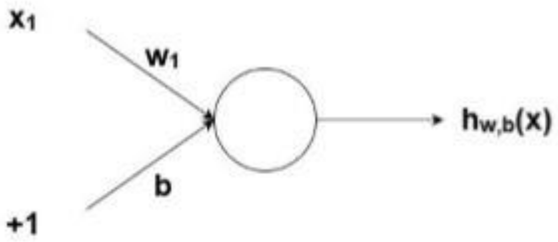


Fig.1.4

In the above scenario, the ‘w1’ has been increased to produce a more specified definition of “turn on” function. As can be seen, by changing the bias “weight” b, we can vary when the node activates. Hence, by adding a bias term, the node can be made to simulate a generic ‘if’ function, i.e. if (x > z) then 1 else 0. Without having a ‘bias term’, it is not possible to vary the z in that ‘if’ statement, it will be always held around zero. This is naturally very advantageous if you are expecting to produce conditional relationships

1.2.4 Putting together the structure

Expectantly the previous descriptions have given a good overview of how a certain perceptron /node/neuron in a neural network produce. However, as we are presumably aware, there are many such interrelated nodes in a fully-fledged neural network. These architectures can come in a myriad of different design, but the most common simple neural.

Network architecture contains an input layer, a hidden layer and an output layer. An instance of such a structure can be seen below:

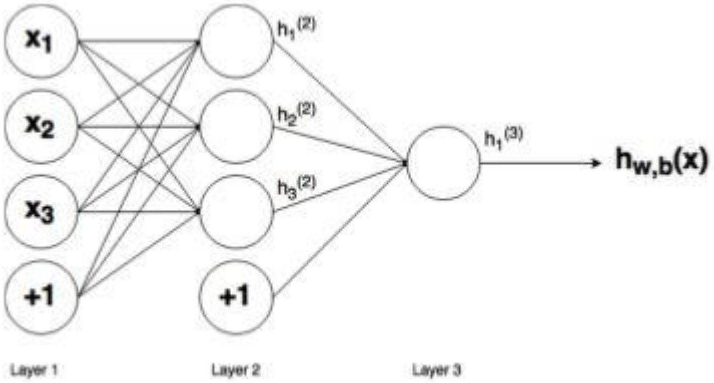


Fig.1.5

1.3 Using ANN with Regression:

Mostly, neural networks are used for the intent of clustering using unsupervised learning .classification using regression or supervised learning. That is, they assist group unlabelled data, classify labelled data or anticipate continuous values.

While classification typically uses a type of logistic regression in the net’s final layer to change continuous data into dummy variables like 0 and 1 – e.g. given someone’s height, weight and age, they can be classified as a heart-disease candidate or not .A true regression connects one set of continuous inputs to another set of continuous outcomes.

.

1.3.3 The Feed Forward Pass:

To determine how to calculate the outcome in neural networks from the input, let’s start with the particular case of the three layers neural network that was conferred above. Following it is presented in equation form.

$$
\begin{aligned}
h_1^{(2)} &= f(w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3 + b_1^{(1)}) \\
h_2^{(2)} &= f(w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2 + w_{23}^{(1)}x_3 + b_2^{(1)}) \\
h_3^{(2)} &= f(w_{31}^{(1)}x_1 + w_{32}^{(1)}x_2 + w_{33}^{(1)}x_3 + b_3^{(1)}) \\
h_{W,b}(x) &= h_1^{(3)} = f(w_{11}^{(2)}h_1^{(2)} + w_{12}^{(2)}h_2^{(2)} + w_{13}^{(2)}h_3^{(2)} + b_1^{(2)})
\end{aligned}
$$

Fig 1.6

In the above equation f(·) assigns to the node activation function, in this case the sigmoid function. The first line, h1 is the output of the first node in the second layer, and its input values are w11x1, w12x2, and w13x3. These inputs can be tracked in an above shown three-layer connection figure. They are added and then passed to the activation function to find the output of the first node. Same will be for another two nodes in the next layer.

The final row is the outcome of the single node in the third and final layer, which is final outcome of this neural network. Here this can be observed that instead of taking the weighted input variables such as (x1,x2,x3), the final node takes as input the weighted output of the nodes of the second layer (*h* 12, *h*22, *h*32), plus the weighted bias. Therefore, in equation form the hierarchical nature of artificial neural networks can be observed.

Deep Learning:

Deep Learning is a new section of Machine Learning research, which has been presented with the motive of enhancing Machine Learning closer to one of its pre-set real goals, Deep Learning is about learning multiple heights of presentation and abstraction that help to give meaning to data such as images, text, sound and data that consist some hidden pattern. Deep learning is very useful when data contains contamination or noise and very low inter-dependency. It's a very efficient model for regressive algorithm.

Deep learning A branch of Machine Learning Multiple levels of representation and abstraction One step nearer to true or actual "Artificial Intelligence" typically means Artificial Neural Networks Externally can hypothetically be imagined as a black box assigns inputs to outputs from assertions it learns from training .Training comes from predefined labelled input/output datasets.

Classification using deep learning:

Deep Learning uses ANN hidden layer technique to classify object with more clarity, it is very in terms of image recognition. Normally there are many concealed layers between input and outcome layer and they are totally connected by neuron transmission, which transfers learning to other neurons. The following diagram shows the classification structure:

:

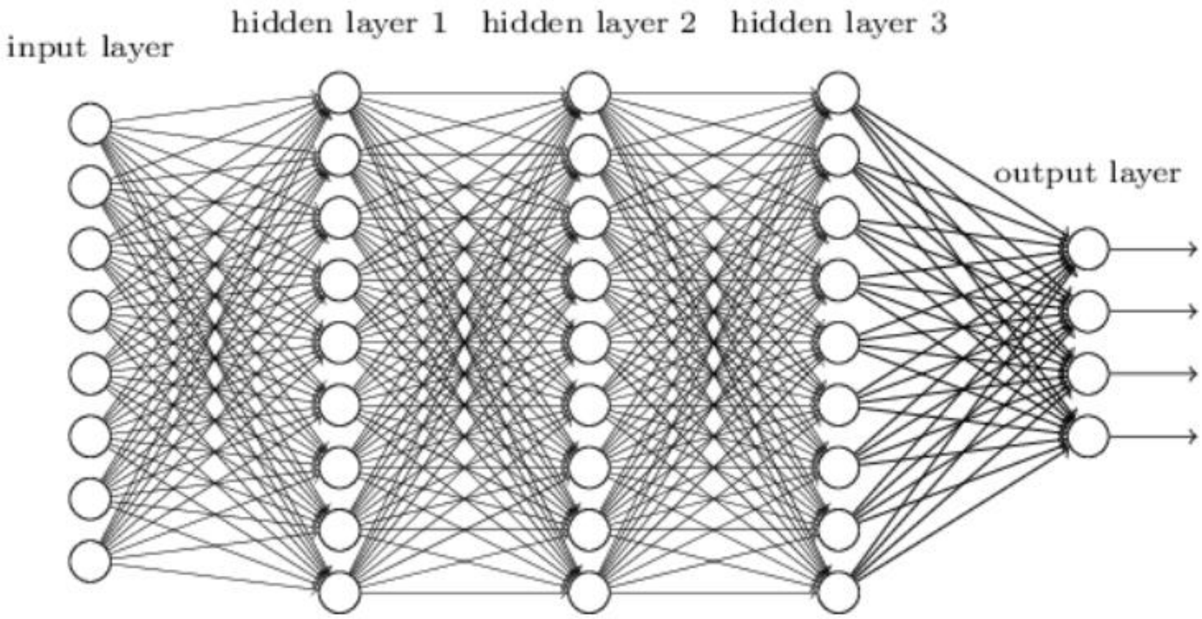


Fig.1.7

Regression:

Regression predictive modeling is the approximating a mapping function (f) from input variables (X) to a continuous outcome variants (y). A continuous outcome variable is a real-valued variant, such as an integer or floating point value. These are mostly quantities, like amounts and sizes.

For instance, a Flat may be predicted to be sold for a specific dollar value, perhaps in the range of $100,000 to $200,000**.**

- A regression problem needs the prediction of a amount.
- A regression can have real values or discrete input variables.
- A problem with more than one input variables is commonly known as multivariate regression problem.
- A regression problem where input variables are organized by time is called a time series forecasting problem

Because a regression predictive model forecast an amount, the competence of the model should be reported erroneous in those predictions. There are many ways to estimate the competency of a regression predictive model, but perhaps the most common is to find out the root mean squared error, abbreviated by the acronym RMSE.

.

1.4 MACHINE LE

Machine learning is a subfield of computer science which includes the study and constructing algorithms that can have knowledge from and make predictions on data. First,in order to make predictions to be data driven, a model is built from a training set of input observations and then machine learning algorithms operates on test data for generating the forecasted output. Here strict static program instructions are not followed. Instead historical data is involved for generating the prediction.

1.5 TYPES OF MACHINE LEARNING

Based on the nature of learning signal or feedback available to a learning system, machine learning is classified into three broad categories:

1. **Supervised learning**:

   The algorithm is presented with sample inputs and their required outcomes, given by a "supervisor", and the aim is to learn a general rule that maps inputs to outcomes.

2. **Unsupervised learning**:

   The algorithm does not require to be provided with labels. It is left on its own for finding architecture in its input. Unsupervised learning can be thought as an aim in itself which is to find existing hidden patterns/structures in data. Due to this reason it is also known to as feature learning.

3. **Reinforcement learning**:

   The algorithm connects with a dynamic environment in which it must perform a certain task for example driving a vehicle. This type of learning doesn’t need a supervisor. It does not need to be told whether it is inclined to its aim. Yet another example can be to learn playing a game by performing against another player.

There is one another type of machine learning that exists between supervised and unsupervised learning that is **Semi-supervised learning,** where an incomplete training signal with some of the target outcomes being missed is provided by the supervisor.

1.6 MACHINE LEARNING TECHNIQUES:

There are certain commonly used techniques in machine learning:

1. **Association Rule Learning**:

   Association rule learning is a technique of discovering interesting connections between variables in some large databases. It is expected to identify strong rules discovered in databases using some measures of eagerness or interestingness

2. **Decision Tree Learning**:

   It is a decision support tool that uses tree-shaped structures such as graph which represents the set of decisions and their possible outcomes. Rules are generated for the classification of a dataset through these decisions. It is one of the ways to represent an algorithm. Decision trees are specifically used in decision analysis where there is a need to identify a strategy which has a higher chances to reach a goal. For example, operations research. Machine learning is also a field where decision trees are very popular. Some of the decision tree methods are Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. Decision rules can be obtain from these techniques that can be applied to a new and unclassified dataset predict which records will belong to which category.

1.7 INTRODUCTION TO PYTHON

In this work, the high-level programming language, ‘python’ has been used for the implementation purpose. So, here is a brief about the language.

Python is a powerful, high-level programming language. Guido van Rossum created this language during 1985- 1990 at the National Research Institute for Mathematics and Computer Science in the Netherlands. It is derived for languages like ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is open source. Its source code is available under GNU General Public License.

Following are certain basic features of Python:

Python is Interpreted:

Like C and C++, Python need not to be compiled before execution. But it only has to be interpreted at runtime. This is similar to PERL and PHP.

Python is Interactive:

You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented:

Python follows the concept of Object-Oriented programming that encapsulates code within objects.

Python is a Beginner's Language:

Python is very easy to learn as compared to other languages. If one has a basic understanding of programming terminologies, there is nothing much left to do for him and it would be a huge plus point.

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read: Python is designed to be highly readable. It makes use of English keywords frequently and the code is clearly defined.

Easy-to-maintain: Python's source code is fairly easy-to-maintain.

A broad standard library: Python has a portable standard library which is compatible on different platforms like UNIX, Windows, and Macintosh.

Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable: New functions and data types implemented in C or C++ can be easily incorporated in the Python interpreter as extensions. Low-level modules can be added to the Python interpreter which enable programmers to customize their tools to be more efficient.

Databases: Python provides interfaces to all major commercial databases.

GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable: Python provides a better structure and support for large programs than shell scripting.

In addition to the above-mentioned features, Python has a long list of good features, few of them are listed below:

Python Have bunch of library for machine learning and deep learning, we are using tensor flow library for deep learning and scikitlearn library for machine learning.

Tensorflow: Tensorflow is a computational framework for building machine learning models. TensorFlow provides a variety of different toolkits that allow you to construct models at your preferred level of abstraction. You can use lower-level APIs to build models by defining a series of mathematical operations. Alternatively, you can use higher-level APIs to specify predefined architectures, such as linear repressors or neural network.

scikit-learn: is a free software machine learning library for the Python programming language.[3] It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy**.**

1.8 MOTIVATION

There is an elementary problem that often arises in a online news to predict its popularity Yet, popularity prediction is a challenging task due to various issues including difficulty to measure the quality of content and relevance of content to users; prediction difficulty of complex Online interactions and information cascades; inaccessibility of context outside the .Regression is an important part of exploratory Artificial Neural Network . Many algorithms exist to forecast news popularity. One of them is regression technique using random forest with good accuracy level. To get improved results we have used Deep learning in Artificial neural network..

1.9 RESEARCH OBJECTIVE

The objectives of this research work are as follows:

- To develop a model to predicting number of share for online news .
- To use the existing algorithms for constructing a new hybrid taking motivation from the previous work done in this area.
- To improve the model using parameter developed so that it can be applied to low dimensional as well as high dimensional data.

1.10 THESIS ORGANISATION

The dissertation starts with Chapter 1 that provides the Introduction to the work. Chapter 2 gives the literature survey of work done in parts. The works of scholars in fields of classification of datasets using the concerned algorithms using the medical data have been studied. Chapter 3 provides the research methodology used to reach the resultant hybrid algorithm. It emphasizes on the k-nearest neighbors and naïve bayes algorithms. Chapter 4 consists of the detailed explanation of the proposed algorithm. In the chapter 5, our proposed work has been evaluated on 5 medical related datasets. Finally, the conclusion and the future scope for this work is provide in Chapter 6

CHAPTER 2- LITERATURE REVIEW

In this era with the rapid growth of internet, smartphone and other gadgets, everybody enjoys reading, sharing news online; news may be in the form of documents, may be in graphical formats, may be the video or may be the records. Since we have a huge amount of data which comes in a different format. So the appropriate action needs to be taken not only to analyse the data but also to fetch important information and patterns from it. Most of time data is noisy so we need to clean data before analyse and fetch pattern, we called it is data cleaning. In the image classification case, speech recognition and many more cases machine learning not able to produce the correct result so for unseen pattern and equation we need to move on artificial neural network to achieve better accuracy

This paper [1] explains why we would choose deep learning instead of other machine learning model. This paper reveals the advantage of deep learning over the machine learning regression model. Fundamental measurements are the prediction of size, effort, resources, cost and time spent in the software development process. In this paper, predictive Artificial Neural Network (ANN) and Regression based models are investigated, aiming at establishing simple estimation methods alternatives. The results presented in this paper compare the performance of both methods and show that artificial neural networks are effective in effort estimation. So this paper gave me some brief idea about deep learning approach.

The second[2] paper explains Predicting Popularity of Online Articles using Random Forest Regression. Predictive analysis using machine learning has been gaining popularity in recent times. In this paper, the Random Forest regression model is used to predict popularity of articles

From the Online News Popularity data set. The performance the Random Forest model is investigated and compared with other models. Impact of standardization, regularization, correlation, high bias/high variance and feature selection on the learning models are also studied. This paper took machine learning regression technique to predict the popularity the online news.

Another paper [3] also describe online news popularity, this paper focuses on popularity prediction of online news by predicting whether users share an article or not, and how many users share the news adopting before publication approach. This paper proposes the gradient boosting machine for popularity prediction using features that are known before publication of articles. This approach produced quite satisfying result compare to previous approach. The proposed model shows around 1.8% improvement over previously applied techniques.

Our Prediction is also depend on time, so time is valuable parameter for our prediction , this paper[4] about considering time parameter In this paper, for the first time, an ensemble of deep learning belief networks (DBN) is proposed for regression and time series forecasting. Another novel contribution is to aggregate the outputs from various DBNs by a support vector regression (SVR) model. We show the advantage of the proposed method on three electricity load demand datasets, one artificial time series dataset and three regression datasets over other benchmark Methods.

Our dataset consist multiple parameter, so for achieving better accuracy we need to design multi instance deep learning technique, this paper [5] This paper presents a weakly supervised regression model for visual house appraisal problem, which aims to predict the value of a house from its photos and textual descriptions (e.g., number of bedrooms). The key idea of our approach is a multi-layer neural network, called multi-instance Deep Ranking and Regression (MiDRR) net, which jointly solves two coupled tasks: ranking and regression, in the multiple instance setting. The network is trained using weakly supervised data, which do not require intensive human annotations. We also design a set of human heuristics to promote deep features through imposing constraints over the solution space.

We have multivariate dataset which consist inter dependency among feature so we required multivariate analysis , this paper [6] explains An increasing number of online news triggers wide academic concern for the prediction of news popularity, which is affected by users' behaviors and not easy to predict. However, existing methods that predict the popularity of online news after publication are not timely enough, and predicting before publication lacks discriminatory features. This paper explores the variables which may affect news popularity and presents a novel methodology to predict the popularity of online news before publication. Through the observation of news, we first find that grammatical construction of titles can affect news popularity.

CHAPTER 3- PROBLEM STATEMENT

In present time with the rapid growth of internet, smartphone and other gadgets, everybody enjoys reading, sharing online news. Online news consist video, audio, document and etc.

Day by day growth of online news is a big concern for popularity of the news .So it is very hard to determine the end user behavior and very hard to predict. However, some existing methods that is not useful to predict because that’s predict the popularity of news after the publication, so that is unusual in real time. This thesis explores the most valuable parameter which affects more to news popularity and presents a novel way to predict the popularity of online news before publication. We have some test dataset consist of 15 input parameter and 1 target parameter In the 15 input feature we have time delta, URL ,number of tokens in title ,number of tokens in news article, number of tokens in content , number of self-link which is directed into self-page, number of links which is directed onto other page, number of images used in the article , number of videos used in the article, number of keywords, which type of channel it is , is it lifestyle , entrainment, it is socmed ,it is tech and last we have Target feature which is number of shares.

Our approach is based on predictive analysis.

In predictive analysis we analyse data based on the current and previous facts to make forecasting about future. Predictive analysis is widely used in financial services. In financial services the best known application is credit scoring and fraud detection for loan scheme. We are using same approach to detecting the number of share for online news popularity prediction.

In our dataset we have input and output feature, using these features, we finally predict news popularity in two ways: how much extent to news will be popular and number of views, link and comment and number of shares.

Online news Popularity is a very challenging task and risk factor is quite very respect to other regressive problems, because wrongly predicted news will go sometimes danger for the publication. Popularity prediction of online news goals to predict number of shares on social media, number of likes and number of comment of this news. So to do predict number of share we are using predictive analysis , In that we are using ANN(Artificial neural network) and machine learning approach , after implanting both we choose best approach among them . So in our dataset ANN is best approach for prediction. Artificial neural network have several advantage over machine learning algorithm, artificial neural networks have some interesting properties that made these family of machine learning algorithms very appealing when confronting difficult patter-discovery tasks. Artificial neural network has two kinds, one is feed forward unidirectional ANN and other is feedback cycle ANN (Back propagation). ANN with Back propagation (BP) learning algorithm is widely used in solving various classification and forecasting problems our data comes from Mash able, a well-known online news website. Our dataset consist of 40K rows and 15 input parameter and 1 output parameter. Artificial Neural Network turns out to be the best approach for prediction, goal of this thesis to predict the number of share of news article and it can achieve accuracy with optimal parameters. Our work can help online news companies to predict news popularity before publication.

3.1 Gap Analysis:

In Previous work, online new popularity estimated by machine learning techniques. All the pervious technique used machine learning regression technique. In machine learning approach they are used random forest, gradient boosting machine and linear model regression for predicting the news popularity on the basis of number of shares. Drawback of machine learning is, they cannot able to find hidden pattern because noise present in data and also lack sometime with the high value of error .So because of noisy data they sometime lack with the accuracy. Our motive is to improve accuracy find hidden pattern in data. The Artificial Neural Networks or (ANNs) are quite similar to brain structure or we can say neuron structure of our brain. Without inferring deep into complex theory of our brain, it is adequate to say that the brain consist of neurons which are type of organic input/output switches. The output state of neurons can change on the basis of their input strength mostly that’s chemical or some electrical form. So it is complex structure and very useful for finding the hidden pattern. While sometimes machine learning approach could not identify hidden pattern and noise which is present in data, Our problem is regression problem, Random Forest is very well known for regressive problem, In this thesis we also compare random forest to other model. We also mention some useful parameter such as impact factor, correlation, Impact and feature selection on the learning models are also studied. We are using deep learning model for this problem, we compare four model each other and after using best model for predicting our result. There are many library for implementing deep learning , example KERAS library is very well known and popular for implementing deep learning . But of other important aspects we are using H2O model for implementing deep learning in python.

3.2 Objective:

To analyse the data and find some pattern so that we can predict the number of share. Our Objective to improve accuracy for online news. The proposed scheme successfully implemented and predictive analysis of proposed and standard methodology has been done for our dataset which consist 40000 rows. For improving accuracy we are using deep learning approach with customized variable, you could see in implementation part that we are used epochs and hidden layer for achieving better accuracy compare to the previous machine learning model. We are also tried so many combination for achieving better result in terms of better RMSE value.

3.3 Methodology:

In this project, we are using artificial neural network to determine the number of share of online news. For this First step to pre-process the data remove the outlier and after that we find the important feature of our dataset. After the first, our dataset is ready for processing; Data pre-processing also refines our data and cleaning in our dataset. After data cleaning and pre-processing we did graph analysis of each input feature which is very useful for determining variable importance , we also did descriptive analysis for better understanding about the dataset. After all the pre-processing we build deep leaning model and also get the best combination of input feature to get better accuracy .

CHAPTER 4 - DESIGN AND IMPLEMENTATION

The basic modules of the proposed system are described in this chapter of system architecture.

Here we want to show how my proposed model will work and what different step of proposed architecture are. In the project we are working on an online news dataset which taken from a worldwide popular news site mash able .com. Our dataset consist 40000 rows and 16 features, in which 15 are input feature and 1 is output feature.

So our solution is broken down in four steps.

1. Data loading stage and library importing.
2. Data pre-processing and clearing stage.
3. Model training
4. Model testing
5. Model comparison.

The first part is data processing part, in this we prepare data for our model. First part consist of many sub part so we prepare data step by step.

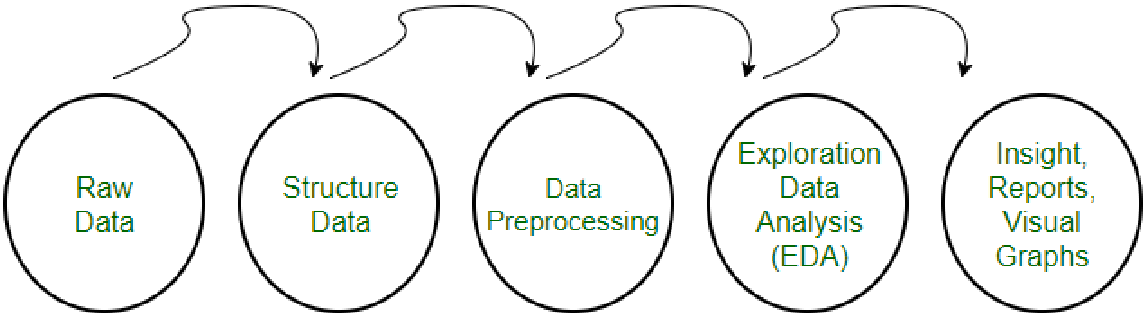

Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Fig 4.1

- **Inaccurate data (missing data) -** There are many reasons for missing data such as data is not continuously collected, a mistake in data entry, technical problems with biometrics and much more. So we eliminate the missing data.
- **The presence of noisy data (erroneous data and outliers) -** The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more. Our prediction will goes wrong if noise is present in our dataset. So for better accuracy we eliminate this feature.
- **Inconsistent data -** The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e., violation of data constraints and much more.

Dataset description:

We have 15 input parameter and 1 output parameter , 15 input parameter as shown below.

Input Variable:

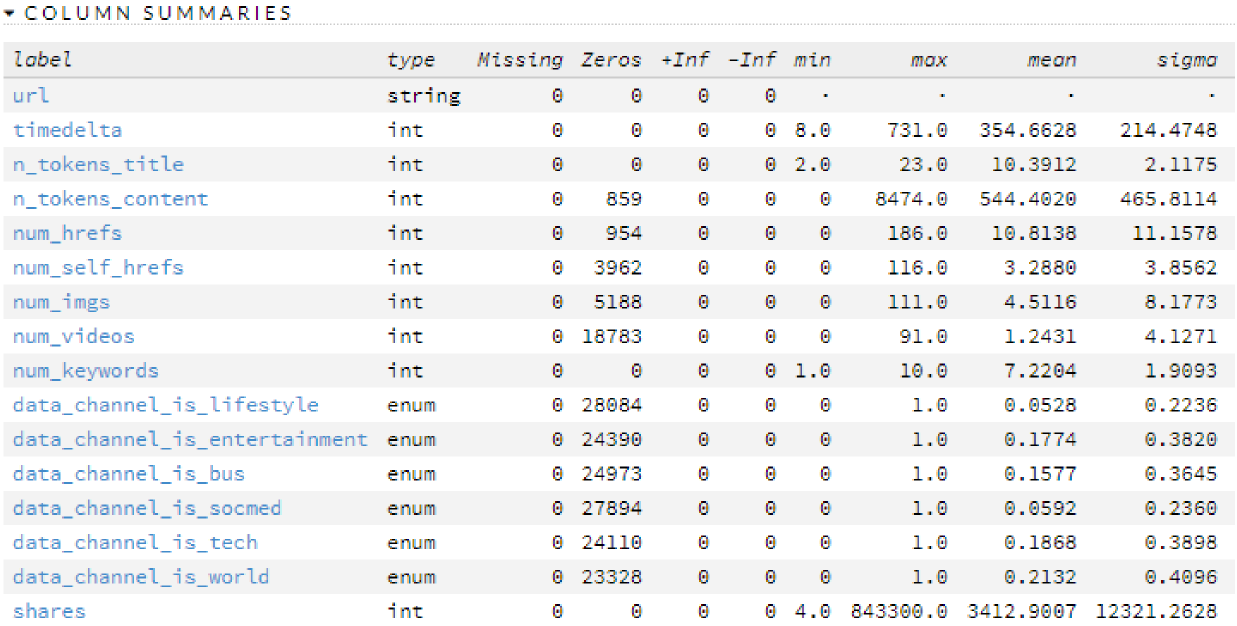
- Article – time delta of article
- Time delta - Days between the article publication and Data acquisition
- Number of tokens in article – Number of words in title
- Number of token in content – Number of words in content
- Number of redirect link – Number of link to other Article
- Number of self-redirect link – Number of link to self-Article
- Number of images – Number of Pictures used
- Number of videos – Number of Videos used
- Number of keywords used in content – Number of Keywords in metadata
- Data Channel is lifestyle - Is data channel 'Lifestyle'?
- Date channel is entrainment - Is data channel 'Entertainment'?
- Data channel is bus - Is data channel 'Business'?
- Data channel is tech – Is data channel ‘Tech’?
- Date channel is world - Is data channel 'World'?
- Data channel is socmed - Is data channel 'Social Media'?

Target Variable:

- Shares – Numbers of shares.

Variable classification:

Below table shows the variable classification and detailed value of every parameter. There are six parameter which are data channel is life style , data channel is entertainment , data channel is bus, data channel is socmed , data channel is tech and data channel is world is classify as enum means categorical variable and there are eight parameter which aretimedelta,n_tokens_title,n_tokens_content,num_hrefs,num_self_hrefs,num_imgs,num_videos,num_kewwords. Now we can see above dataset consist missing values zero. Max, mean and Min value also mention in the table. Variable classification part is very important for our problem because if we wrongly classify our variable then our accuracy lacks.



▾ COLUMN SUMMARIES

label	type	Missing	Zeros	+Inf	-Inf	min	max	mean	sigma
url	string	0	0	0	0	.	.	.	.
timedelta	int	0	0	0	0	8.0	731.0	354.6628	214.4748
n_tokens_title	int	0	0	0	0	2.0	23.0	10.3912	2.1175
n_tokens_content	int	0	859	0	0	0	8474.0	544.4020	465.8114
num_hrefs	int	0	954	0	0	0	186.0	10.8138	11.1578
num_self_hrefs	int	0	3962	0	0	0	116.0	3.2880	3.8562
num_imgs	int	0	5188	0	0	0	111.0	4.5116	8.1773
num_videos	int	0	18783	0	0	0	91.0	1.2431	4.1271
num_keywords	int	0	0	0	0	1.0	10.0	7.2204	1.9093
data_channel_is_lifestyle	enum	0	28084	0	0	0	1.0	0.0528	0.2236
data_channel_is_entertainment	enum	0	24390	0	0	0	1.0	0.1774	0.3820
data_channel_is_bus	enum	0	24973	0	0	0	1.0	0.1577	0.3645
data_channel_is_socmed	enum	0	27894	0	0	0	1.0	0.0592	0.2360
data_channel_is_tech	enum	0	24110	0	0	0	1.0	0.1868	0.3898
data_channel_is_world	enum	0	23328	0	0	0	1.0	0.2132	0.4096
shares	int	0	0	0	0	4.0	843300.0	3412.9007	12321.2628

Fig 4.2

Our response variable is number of shares which classify as a numeric category. Because Our it is regressive problem and our goal is to predict the number of shares which is integer . If our response variable is categorical then our response variable should be categorized into enum.

Variable Importance:

Below graph shown the variable importance, the graph tells about the parameter affected more to the number of shares (Target variable).

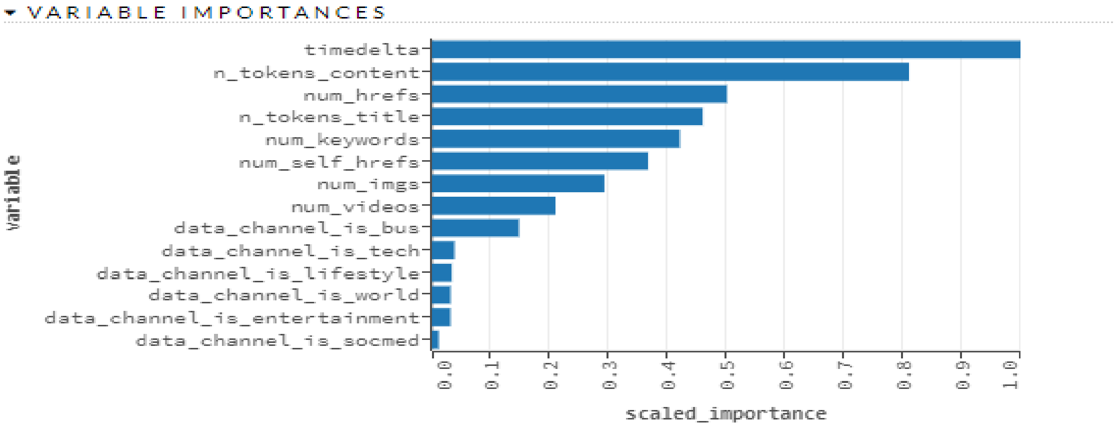


Fig 4.3

Above graph shows the time delta is most important parameter in our dataset and data channel is socmed is least importance parameter. N_tokens_content is second most important parameter and data channel is entertainment is second least important parameter. We can also remove outlier by python script using scipy and stats function. Now we have data and we also done pre-processing part and data cleaning is also done.

Now it’s time to build model on our dataset, we implement deep learning model on our dataset.

DEEP LEARNING:

An “apparent paradox” with Deep Learning is that it can generalize well in practice despite its large capacity, numerical instability, sharp minima, and no robustness. Another important issue with deep architectures is numerical instabilities. Numerical instabilities in derivative-based learning algorithms are commonly called exploding or vanishing gradients. Additional difficulties stem from instabilities of the underlying forward model. That is the output of some networks can be unstable with respect to small perturbations in the original features. In machine learning it is called non-robustness. The goal of generalization theory is to explain and justify why and how improving accuracy on a training set improves accuracy on a test set. The difference between these two accuracies is called generalization error or “generalization gap”. More rigorously generalization gap can be defined as a difference between the non-computable expected risk and the computable empirical risk of a function.

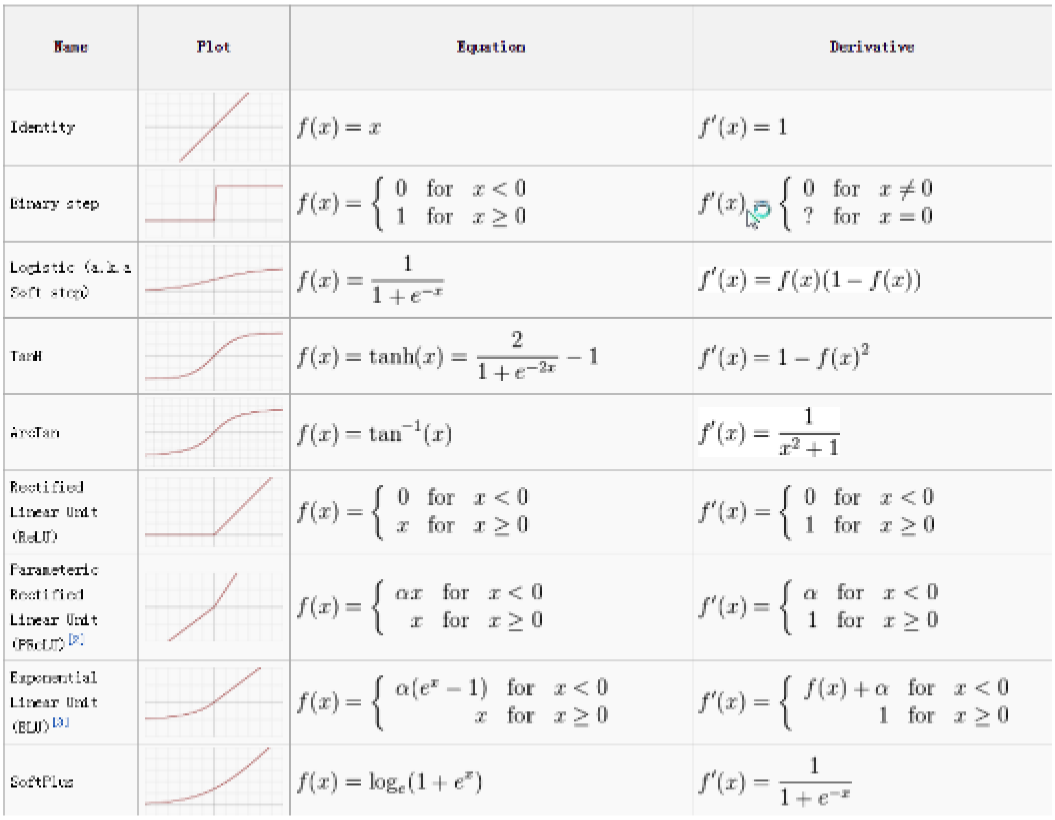
ACTIVATION FUNCTION:

An activation function is a mapping of summed weighted input to the output of the neuron. It is called an activation/ transfer function because it governs the inception at which the neuron is activated and the strength of the output signal**.**

$$Y \ = \ \Sigma(weight \ * \ input) + bias$$

We have many activation functions, out of which most used are relu, tanh, solfPlus.

Here list of activation shown below:



Name	Plot	Equation	Derivative
Identity		$f(x)=x$	$f'(x)=1$
Binary step		$f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x\ne 0\\ ? & \text{for } x=0\end{cases}$
Logistic (a.k.a Soft step)		$f(x)=\frac{1}{1+e^{-x}}$	$f'(x)=f(x)(1-f(x))$
TanH		$f(x)=\tanh(x)=\frac{2}{1+e^{-2x}}-1$	$f'(x)=1-f(x)^2$
ArcTan		$f(x)=\tan^{-1}(x)$	$f'(x)=\frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x)=\begin{cases}\alpha(e^x-1) & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}f(x)+\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$
SoftPlus		$f(x)=\log_e(1+e^x)$	$f'(x)=\frac{1}{1+e^{-x}}$

Fig 4.4

Here above some activation function with equations. We are using recified linear unit in our project with some custom modification in that.

COST FUNCTION AND GRADIENT DESCENT:

The cost function is the measure of “how good” a neural network did for its given training input and the expected output. It also may depend on attributes such as weights and biases. A cost function is single-valued, not a vector because it rates how well the neural network performed as a whole. Using the Gradient Descent optimization algorithm, the weights are updated incrementally after each epoch.

Compatible Cost Function:

Sum of squared errors (SSE)

$$J(w) = \frac{1}{2} \sum_{i} \left((target)^{i} - \ (output)^{i} \right) \ ^{2}$$

The magnitude and direction of the weight update is computed by taking a step in the opposite direction of the cost gradient.

$$\Delta w_{j} \ = \ - \eta \frac{\delta J}{\delta w_{j}}$$

where η is the learning rate.

Where ∆w is a vector that contains the weight updates of each weight coefficient w, which are computed as follows:

$$\Delta w_{j} = \frac{1}{2} \sum_{i} \left((target)^{i} - \ (output)^{i} \right) x_{j}^{(i)}$$

We calculate the gradient descent until the derivative reaches the minimum error, and each step is determined by the steepness of the slope (gradient). So our cost function defines accuracy level of our problem. Our final task to obtain minimum value w so that we can make better accuracy for our problem. Below graph shown relation of w and input parameter.

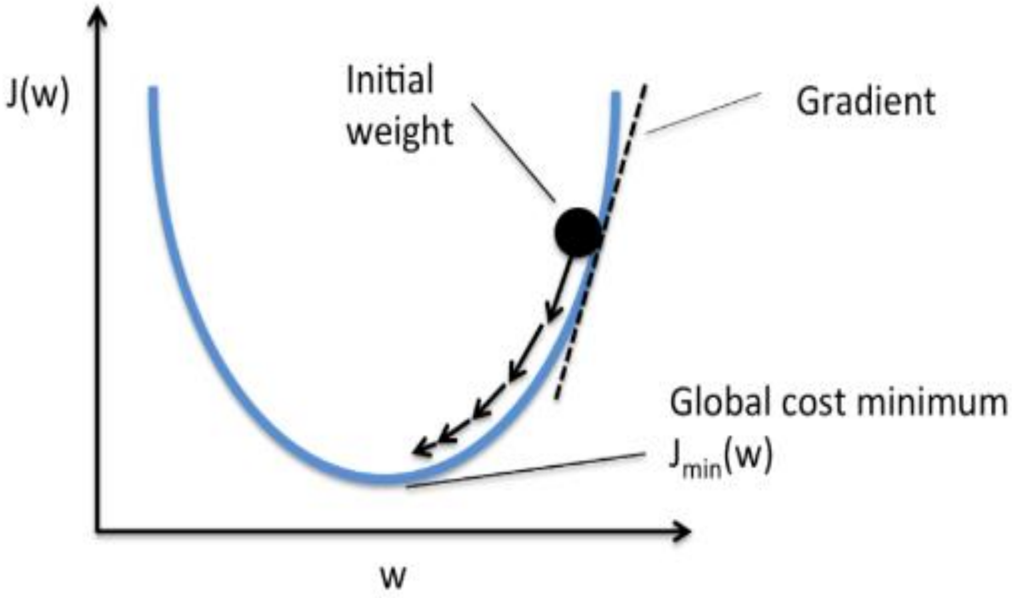


Fig 4.5

In the above its very clear cost function value decrease with the value of initial weights.So with the intial weights its very high , on reducing the initial weights cost function also decrease , but at some time when cost function reches its thershold after thast by decresing input weight cost function value increase . So finally we find jmin value whrer we find global minimun cost function.

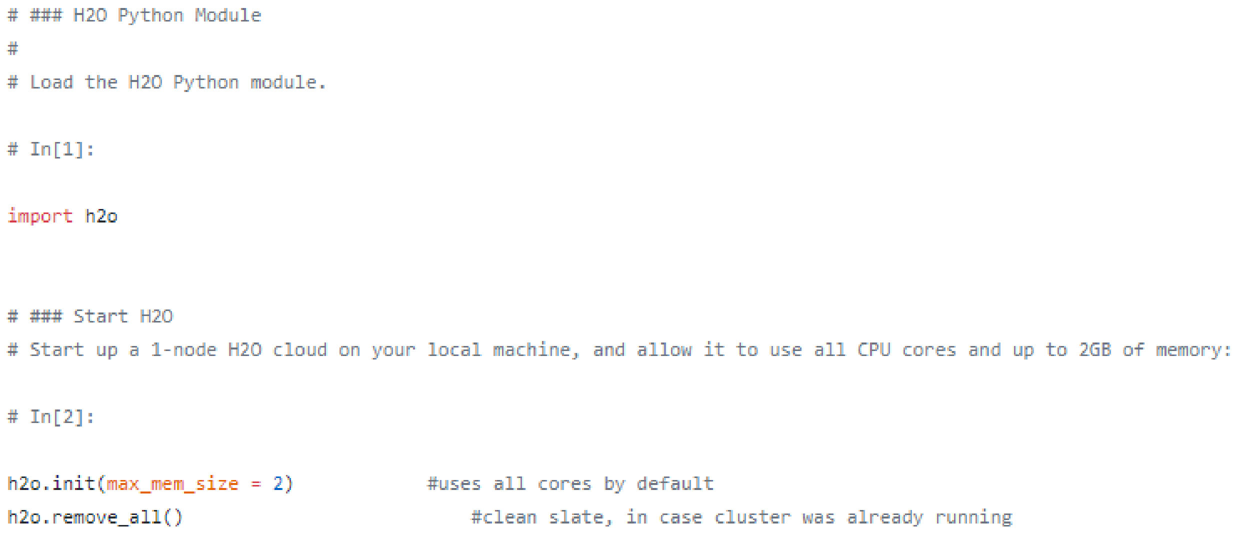
Now we have idea about cost functiona and gradient decesnt , so its time build model using some python library. We have implemented deep learning model using python and h2o combinely . H2o ai is open source plateform for machine learning and neural network. So we are going to demostrate our code and also trying to explain , how is going to construct the model.

So below we are mentioning some step how to build our model step by step.

- Making environment for H20 in python
- Load data
- Define model
- Compile model
- Define model
- Fit model
- Evaluate model

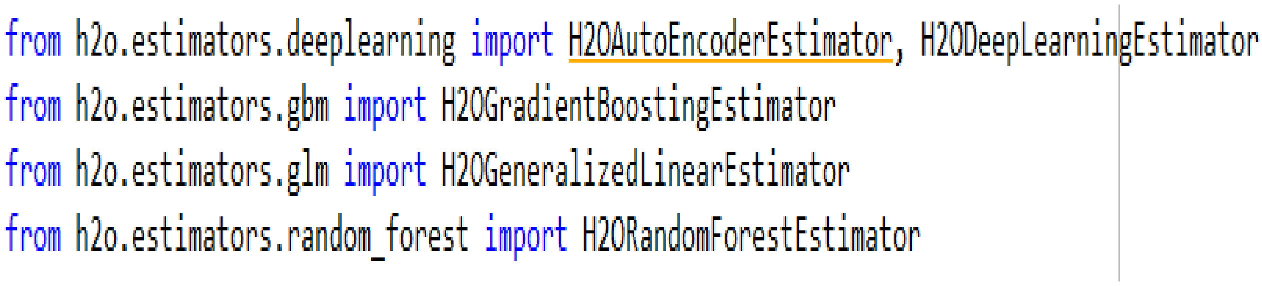
H2O environment in python:

First step to install h2o module in python. We are using Anaconda navigator for development of this project. Before going to use h2o in our project we downloaded h2i library .



```
# ### H2O Python Module
#
# Load the H2O Python module.

# In[1]:

import h2o


# ### Start H2O
# Start up a 1-node H2O cloud on your local machine, and allow it to use all CPU cores and up to 2GB of memory:

# In[2]:

h2o.init(max_mem_size = 2)            #uses all cores by default
h2o.remove_all()                          #clean slate, in case cluster was already running
```

Above screenshot shows import command to use h2o module in your script and init command for initialization a single instance of h2o. Remove all command for cluster purpose in case any cluster is exist and any h2o running in the project.

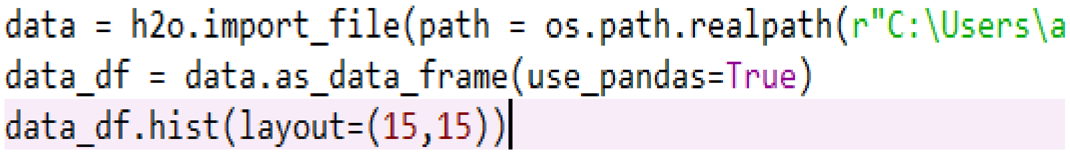


```
from h2o.estimators.deeplearning import H2OAutoEncoderEstimator, H2ODeepLearningEstimator
from h2o.estimators.gbm import H2OGradientBoostingEstimator
from h2o.estimators.glm import H2OGeneralizedLinearEstimator
from h2o.estimators.random_forest import H2ORandomForestEstimator
```

Fig 4.6

Above screenshot shows the import model from h2o library, the motive behind multiple modelling because we will compare the error rate of multiple mode and at last finalize the model on the basis of low error rate. In the above screenshot, you can see we imported random forest, deep learning , gradient boosting machine and general linear estimator. We have some other model but these are very famous having richest functionality for regression type of problem. Since we have regression type of problem so we preferred these machine learning model , Now our goal is to compare machine learning model and ANN model , and finally we will use the best model among them . As mention on above part we have dataset consisting 40000 rows and 16 column, in which 15 feature treat as input feature and 1 is target feature. So our model is going to predict the value according to target value. So this is a regression type problem so our measurement of accuracy in terms of

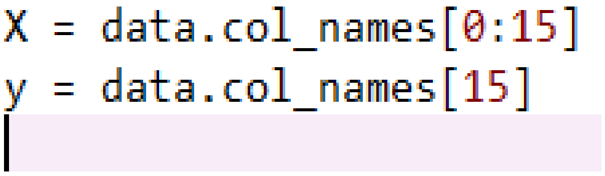
RMSE value, which mean low value of RMSE gives better accuracy , So our focus on low value of RMSE



```
data = h2o.import_file(path = os.path.realpath(r"C:\Users\a
data_df = data.as_data_frame(use_pandas=True)
data_df.hist(layout=(15,15))
```

Fig 4.7

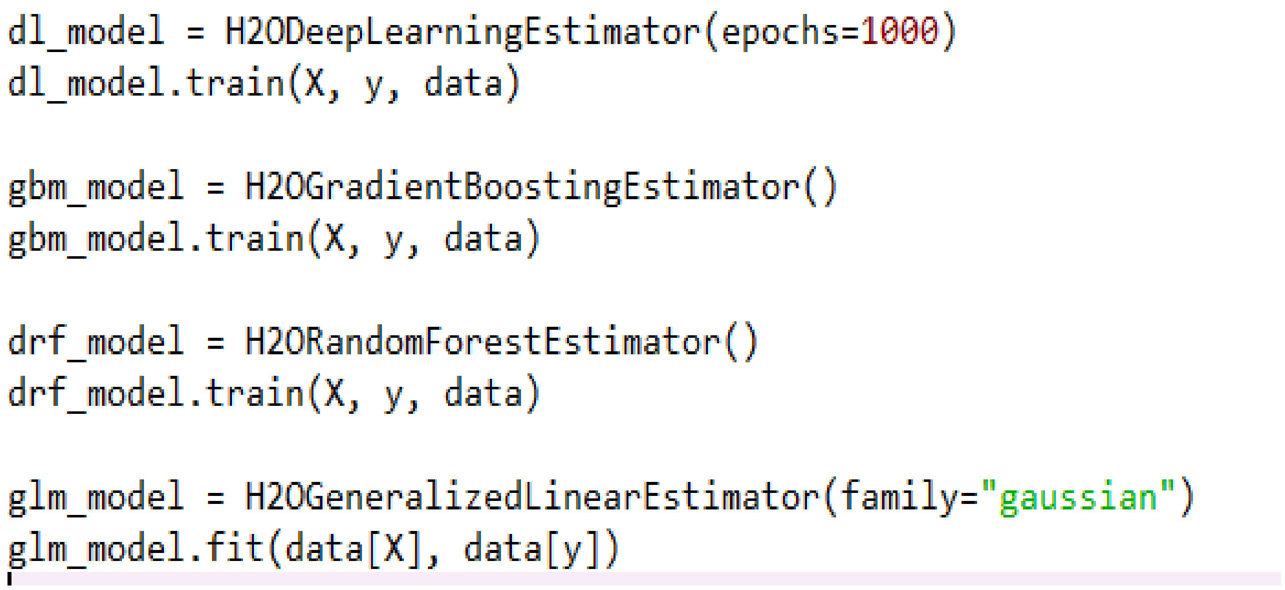
In the above screenshot we have h2o inport statement to import the datset into out kernel , after loading the datset with using import command (we imported CSV file into our program) .Then we import file into data variable , If someone want some graphical representation of the input feature then in the second line we convert into dataframe which is defined in pandas library .we are hist using in above screenshot for display the result in histogram.



```
X = data.col_names[0:15]
y = data.col_names[15]
```

Above sreeenshot shows , Two variable one is X and other is Y.

X contains all rows and columns 0 to 15 means contains only input feature and y contains all rows and single column having target feature . Input feature means those data   used as a trained data and Target feature is predicting variable .



```
dl_model = H2ODeepLearningEstimator(epochs=1000)
dl_model.train(X, y, data)

gbm_model = H2OGradientBoostingEstimator()
gbm_model.train(X, y, data)

drf_model = H2ORandomForestEstimator()
drf_model.train(X, y, data)

glm_model = H2OGeneralizedLinearEstimator(family="gaussian")
glm_model.fit(data[X], data[y])
```

Above screenhot is about the initliaze the models, So in above screenshot we first inialize deep leaning which is a part of artificial neural netwrok after that we train our deep leaning model on the basis if input and output feature. After running the command we have trained deep learinng model and it is store in the dl _model varibale . after that we initlaize the Gradient Boosting machine model which is based on supervised learning , and same as above we trained above model on the basis of input and output feature and it is trained and store in the gbm_model variable.Third line consist of initite random forest model which also a type of supervised leanring , and same as above we trained above model on the basis of input and output feature and it is trained and store in the drf_model. In last we initlaize the gernal linear model which is also a regressive approach , and same as above we trained above model on the basis of input and output feature and it is trained and store in glm model .

CHAPTER-5 RESULT AND ANALYSIS

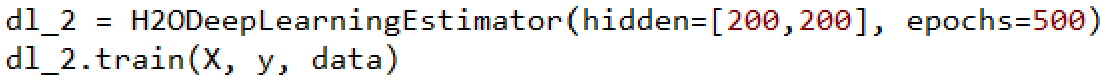
In result and analysis we are going to comare to each and evry trained model. Now we have four trained model, deep learing , random forest, gradient boosting machine and gernral linear model, Now we come to compare the accuracy of above four model Below graph shown the result of four model on the basis of RMSE value .

We have trained 4 model on the on line news popularity data set,

1. Random forest
2. Gradient Boosting Machine
3. General linear modeling
4. Deep learning

Random Forest , Gradient Boosting machine and General linear modeling is totally machine learning . Deep learning is totally based on Artificial Neural Network , So we used epochs=1000, and activation function is Rectifier. We take hidden layer size 100, 100. This is the combination of paremter which we take on to the deep learning model.. we have also partition our dataset on the basis of validation , test and training . we have 60% training dataset, 20% used for validation and 20% used for testing purpose.

As in the result graph we have seen the RMSE value of deep learning is quite low as compare to other model. So we finlizae the deep learning model for further process . Now we have so much custom paramter on deep learning so we can change according to our datsaset. Ex: we can change epochs, check_point , hidden layer size and many more so we can change this parameter.



```
dl_2 = H2ODeepLearningEstimator(hidden=[200,200], epochs=500)
dl_2.train(X, y, data)
```

Fig 5.1

From the above Foure model , Deep learning performs far better then other type of machine learning model. Our perfroms measure is on the basis of low RMSE value . We have a family of activation function which is gaussian type family. In model part we also did parameter tuning of deep learning model so that we could optmize the result and get the low RMSE value.

REFERENCES

1. Saha, R. K., Lease, M., Khurshid, S., & Perry, D. E. (2013, November). Improving bug localization using structured information retrieval. In Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on (pp. 345-355). IEEE.
2. Lewis, C., Lin, Z., Sadowski, C., Zhu, X., Ou, R., & Whitehead Jr, E. J. (2013, May). Does bug prediction support human developers? findings from a google case study. In Proceedings of the 2013 International Conference on Software Engineering (pp. 372-381). IEEE Press.
3. Harman, M., McMinn, P., De Souza, J. T., & Yoo, S. (2012, January). Search based software engineering: Techniques, taxonomy, tutorial. In Empirical software engineering and verification (pp. 1-59). Springer-Verlag.
4. Nam, J. (2014). Survey on software defect prediction. Department of Compter Science and Engineerning, The Hong Kong University of Science and Technology, Tech. Rep.
5. Kim, M., Hiroyasu, T., Miki, M., & Watanabe, S. (2004, September). SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. In PPSN (pp. 742-751).
6. Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm.
7. Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.
8. Gupta, V., Ganeshan, N., & Singhal, T. K. (2015). Developing Software Bug Prediction Models Using Various Software Metrics as the Bug Indicators. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 6(2), 60-65.

9. Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation
10. Predicting Popularity of Online Articles using Random Forest Regression
11. Predicting the Popularity of Online News from Content Metadata
12. Ensemble Deep Learning for Regression and Time Series Forecasting

13. Learning Multi-Instance Deep Ranking and Regression Network for Visual House Appraisal
14. Predicting the Popularity of Online News Based on Multivariate Analysis