International Journal of Engineering & Technology, 7 (2.7) (2018) 1085-1090

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Probabilistic estimation of software development effort techniques using machine learning

Dr. P. Vidya Sagar [1] *, Dr. Nageswara Rao Moparthi [2], Venkata Naresh Mandhala [3]

[1] Associate Professor, Department of Information Technology,Velagapudi Ramakrishna Siddhartha Engineering CollegeVijayawada
[2] Department of Computer Science and EngineeringVelagapudi Ramakrishna Siddhartha Engineering CollegeVijayawada
[3] Department of Computer Science and EngineeringKoneru Lakshmaiah Education Foundation Vaddeswaram, Guntur
**Corresponding author E-mail: mvnaresh.mca@gmail.com*

Abstract

Precisely assessing programming exertion is likely the greatest test confronting for programming engineers. Assessments done at the prop-osition arrange has high level of incorrectness, where prerequisites for the degree are not characterized to the most reduced subtle elements, but rather as the venture advances and necessities are explained, exactness and certainty on appraise increments. It is vital to pick the correct programming exertion estimation systems for the forecast of programming exertion. Artificial Neural Network (ANN) and Support Vector Machine (SVM) have been utilized on guarantee dataset for forecast of programming exertion in this article.

Keywords: *Software Effort; SVM; Artificial Neural Networks; Support Vector Machine; Accuracy*

1. Introduction

A product venture is harder to outwardly screen and control than venture that makes a physical item like a building that you can see. For programming ventures, it ends up hard to meet the due dates and to convey the item includes "as guaranteed" and inside the financial plan. Before beginning a venture, programming venture arranging is a standout amongst the most basic exercises in current programming improvement. Without a practical and target design, a product improvement venture can't be overseen in a powerful way [3], [4].

Effort estimation is playing an effective role in managing and quality of a software development projects. The word Effort in software industries is used to refer as one of major software metric term where its play's a vital role in estimation of cost required by project and also in software duration prediction [2]. If the Effort estimation of under the required or of above the required may lead to an risky consequences and even sometimes it might likewise make issue venture improvement. It is vital to foresee the cost of the task with a precise level else it might prompt invade the financial plan. Exertion estimation in early times of programming improvement have been in movement of programming building research field for a long haul and it has prompted a development of various techniques for forecast of Effort [11]. Many Effort prediction methods are available for categorizing the estimation techniques. Most of them are listed below (i) Expert prediction: It's capability is based on acumen processes where the whole prediction is done. (ii) Formal Prediction: its capability is based on the automatic processes [9]. E.g., the use of Formal prediction is based on the historical data. (iii) Hybrid Prediction: This prediction model deals with the combination of acumen and of automatic processes on different sources.

2. Current applications for software prediction to predict the time series problems

2.1. Gaussian process

Gaussian Process in likelihood hypothesis, the Gaussian procedure (GP) is acknowledgment comprise of arbitrary esteems related with each point in a scope of times [6]. The irregular variable has an ordinary appropriation. It has been specified that "a Gaussian procedure is a stochastic procedure whose speculation of a Gaussian dissemination over a limited vector space to a capacity space of unbounded measurement". Gauss calculation is a strategy used to take care of non-direct slightest squares issues[7].Non-coordinate least squares issues develop for instance in non-straight backslide, where parameters in a model are searched for to such a degree, to the point that the model is in extraordinary simultaneousness with available observations.

2.2. Least median square

Least Median Square (LMS) technique is one of the factual strategies for unravelling the conditions which are more than obscure [5]. This strategy relatively utilized as a part of systematic relapse. Truth be told, Least Median Square is a technique for fitting the dataset [8]. The least Median Square should yield the littlest incentive for the middle of squared residuals figured for the whole informational collection [4]. It implies the residuals, the distinction between genuine information and anticipated information.

2.3. Function point analysis

It is a technique for evaluating the extent of a venture by considering the information and yield components that are in the task and

Copyright © 2018 Dr. P. Vidya Sagar et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

combine each kind of activity into information or exchange work [6]. The span of activities used to be registered utilizing the KLOC (Kilo Lines of Code), yet couldn't be connected before the venture was finished, as the expectation models were a long way from being precise, however the idea had similitude's to FPA as observed[8].The presentation of FP Counting practice finds out the extent of the undertaking by considering every one of the factors in the condition to convey the capacity point check, and gave preferable gauges over those registered esteems[7]. The Function Point Analysis estimation procedure approves the individual components and the related gatherings to touch base at an unpredictability level of high, medium and low and appoints a capacity point mean every subset [11]. The valuable multifaceted nature is figured as the total number of customer identifiable social events that exists inside DETs and is named as Record Element Type (RET) in Data Functions and all referenced archive makes are viewed as FTR in Transactions Functions. A comparing network holds the reference work point esteems for all capacity writes specifically the ILF, EIF, EI, EO and EQ, as for the scope of DET and RET/FTR in each capacity [10]. The aggregate whole of the high, medium and low tally of all tasks is the unadjusted capacity point check.

2.4. Cocomo

The four fundamental components of the COCOMO II methodology are: Preserve the transparency of the COCOMO. The key concept structure of COCOMO II to future programming commercial center divisions depicted above; Key of data sources and yields of the COCOMOII sub models/methods to the level of data accessible. This enable sub models to be customized to a venture's specific procedure technique [5]. COCOMO II takes after the transparency standards utilized as a part of the first COCOMO. Consequently, the majority of its connections and calculations will be freely accessible. Likewise, the majority of its interfaces are intended to be open, all around characterized, and parameterized, so corresponding preprocessors post-processors (venture arranging and control apparatuses, venture flow methods, hazard estimation), and larger amount bundles (venture administration bundles, item transaction helps), can be joined direct with COCOMO II. To help the product commercial center divisions above, COCOMO II gives a group of progressively nitty gritty programming cost estimation models.

3. Methodologies

In this area, the strategies of the proposed framework display and the model plan in light of the Support Vector Machine(SVM) are clarified[1].In SVM& MLare overseen learning models with related learning computations that research data used for gathering and revert examination. Given a course of action of planning cases[16], SVM getting ready computation produces a model that handles new cases to one arrangement or the others, making it a non-probabilistic dual direct classifiers.SVM display is a portrayal of the belongings as centers in space, mapped with the objective that the instances of the diverse classes are isolated by a sensible opening that is as wide as would be judicious. New frameworks are then mapped into that same space and expected to have a place with an arrangement in light of which side of the opening they fall. Not with standing performing the direct portrayal, SVMs can only capably play out non straight request using what is known as the part trap, unquestionably mapping there to high dimensional component spaces. Right when data are not named, managed learning is not possible, and an unsupervised learning approach's are required, which tries to find expected grouping of the data to social events, and after that guide new data to these formed get-togethers. Bolster vector clustering [2]algorithm made applies the insights of help vectors, created in the help vector machines calculation.

SVM (SVM) is an idea in programming building for a course of action of related managed learning controlled learning systems that dismember data and see outlines, used for portrayal and backslide examination. The standard SVM takes an arrangement of information and its predicts, for the each given information, which of two conceivable classes the information is an individual from, which influences the help vector to a machine a non-probabilistic parallel straight classifier [9].

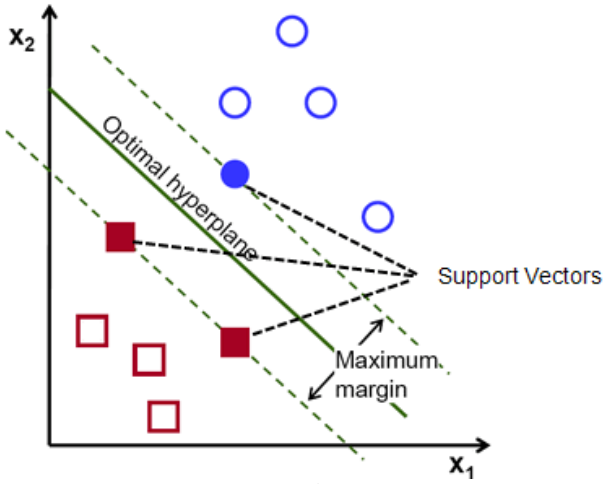

Fig. 1: SVM Classification Example.

3.1. SVM linear classification

Direct Classifier, An information point considered as a p-dimensional vectors(list of p-numbers) and we detach centers using (p-1) dimensional hyperplane. It will be numerous hyperplanes isolating information in a direct request, yet the best hyperplane is the one which boosts the edge i.e., the separation amongst hyperplane and nearest information purpose of different class.
The Maximum-edge hyperplane is dictated by the information indicates that lie closest it. Since we need to expand the separation amongst hyperplane and the information focuses [11]. Since we need to expand the separation amongst hyperplane and the information points. These information focuses on hyperplane are known as help vectors.

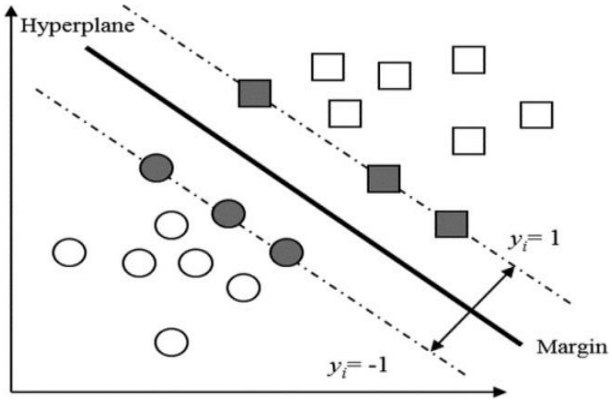

Fig. 2: SVM with Linear Data.

3.2. SVM non-linear classification

In reality, our dataset is for the most part scattered up to some degree. To tackle this issue partition of information into various classes based on a straight direct hyperplane can't be viewed as a decent decision. In Non-Linear SVM Classification, information focuses plotted in a higher dimensional space. It regularly happens that our information focuses are not straightly detachable in a p-dimensional (limited) space. To settle this, it was proposed to delineate dimensional space into a significantly higher dimension-

al space [12]. We can draw tweaked/non-straight hyperplane utilizing bit traps. Each portion holds a non-piece work.

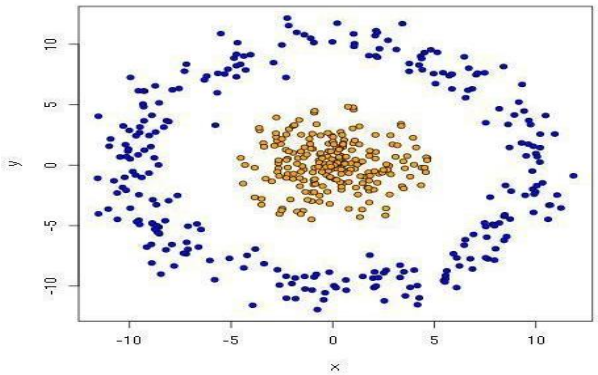

Fig. 3: SVM with Non-Linear Data.

3.3. Kernel functions

In machine learning, portion strategies are a class of calculations for design examination, whose best known part is the support vector machine (SVM). The general errand of example investigation is to discover and contemplate general kinds of relations (for instance groups, rankings, key parts, connections, arrangements) in datasets. For some calculations that explain these errands, the information in crude portrayal must be unequivocally changed into include vector portrayals by means of a client determined element outline: differentiate, part strategies require just a client indicated bit, i.e., a likeness work over sets of information focuses in crude portrayal.

3.4. Types of kernel function

Linear:-The Maximum-edge hyperplane is controlled by the information indicates that lie closest it. we have to expand the partition among hyperplane and the data centers. The information focuses which impacts our hyperplane are known as help vectors.

$$K(\vec{x_i}, \vec{x_j}) = (\vec{x_i} . \vec{x_j})$$

Polynomial (homogeneous) Kernel: The polynomial piece capacity is explained by the above articulation. Where k(xi, xj) is a section work, xi and xj are vectors of feature space and d is the level of polynomial limit.

$$k(x_i, x_j) = (1+x_i^T x_j)^p \quad (2)$$

Polynomial (non-homogeneous) Kernel: it is a steady term is additionally included. The enduring word"c" is generally called a free parameter. It impacts the blend of features. x and y are vectors of feature space.

$$k(x,y) = (x^\tau y+c)^d \quad (3)$$

Radial Basis Function Kernel: It is otherwise called RBF bit. It is a standout amongst the well known portions. For remove metric squared euclidean separation is utilized. It is useful to draw totally non-straight hyperplanes.

$$k(x, x') = \exp\left(-\frac{(||x-x'||)^2}{2\sigma^2}\right) \quad (4)$$

3.5. Motivation and information

Part strategies can be thought of as case based students: as opposed to adapting some settled arrangement of parameters relating to the highlights of their sources of info, they rather "recall the preparation case and learn for it a comparing weight. Prediction for unlabeled information sources i.e., those not in the preparation set, is dealt with by the utilization of a comparability work , called a bit, between the unlabeled information and every one of the preparation inputs. For case, a kernelized double classifier regularly registers a weighted whole of likenesses

$$\hat{y} = \operatorname{sgn} \Sigma_{i=1}^{n} w_i y_i (x_i x^i) \quad (5)$$

3.6. Algorithm of SVM (selecting the SVM hyperplanes)

For the data which can be secluded straightforwardly, we select two parallel hyperplanes that distinctive the two classes of data, with the objective that detachment among the lines is most noteworthy. The area b/w these two is known as "edge" and most extreme edge hyperplane is the one that lies amidst them.

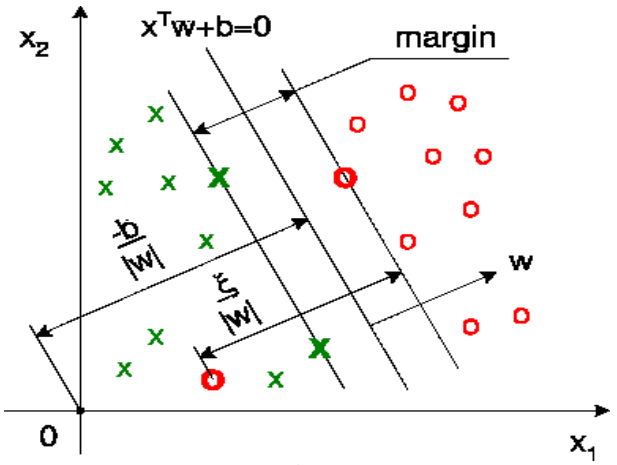

Fig. 4: Svm Classifier Linear Data Type.

$$\vec{w}x_i - b \geq 1 \text{ if } \theta_i = 1 \quad (6)$$

$$\vec{w}x_i - b \leq 1 \text{ if } \theta_i = -1 \quad (7)$$

Where $\vec{\ }$ is typical vector to the hyperplane, θi means classes and xi indicates highlights. The Distance between two hyperplanes is $\vec{\ }$, to augment this separation denominator esteem ought to be limited i. e, $\vec{\ }$ ought to be limited. For appropriate grouping, we can manufacture a joined condition:

$$\|\vec{w}\|_{min} \text{ for } \theta_i(\vec{w}x_i - b) \geq 1 \ \forall i = 1, 2 \ldots n \quad (8)$$

Non-Linearly Separable: To construct classifier for non-straight information, we attempt to limit

$$\left[\frac{1}{n}\Sigma_{i=n}^{n} \max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2 \quad (9)$$

3.7. SVM applications

SVM's are a consequence of Neural Network. They are extensively associated with configuration gathering and backslide issues. Here are some of its applications:

1) Facial expression classification: SVMs can be utilized to group outward appearances. It utilizes measurable models of shape and SVMs.
2) Speech recognition: SVMs are utilized to acknowledge watchwords and reject non-catchphrases them and manufacture a model to perceive discourse.
3) Written by hand digit acknowledgment: Bolster vector classifiers can be connected to the acknowledgment of disengaged transcribed digits optically filtered.
4) Content Categorization: In data recovery and afterward classification of information utilizing marks should be possible by SVM.

3.8. Multilayer perceptron

The multilayer perceptron (MLP) is standouts amongst the most broadly actualized neural system topologies MLPs are ordinarily prepared with the back proliferation calculation. The back propagation rule principle spreads the mistakes through the system and permits adjustment of the concealed PEs. The field of manufactured neural systems is regularly just called neural systems or multi-layer perceptron's after maybe the most helpful kind of neural system. A perceptron is a solitary neuron show that was an antecedent to bigger neural systems. It is a field that examines how basic models of organic brains can be utilized to tackle troublesome computational assignments like the prescient displaying undertakings we find in machine learning.

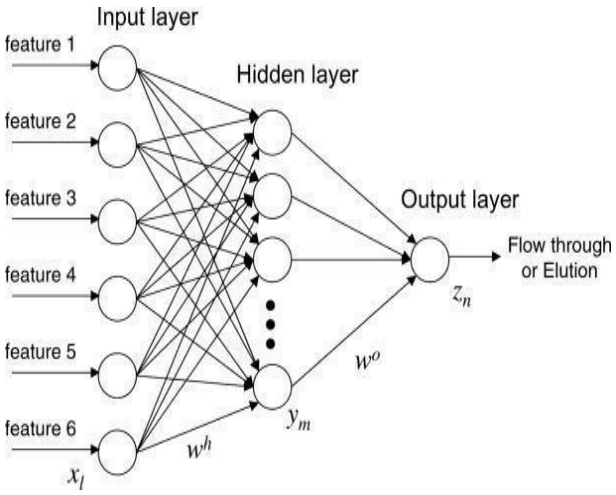

Fig. 5: Multilayer Perceptron Representation.

The objective isn't to make practical models of the mind, however rather to create vigorous calculations and information structures that we can use to display troublesome issues. The energy of neural systems originate from their capacity to take in the portrayal in your preparation information and how to best relate it to the yield variable that you need to anticipate. In this sense neural systems take in a mapping. Numerically, they are equipped for adapting any mapping capacity and have been turned out to be an all-inclusive guess calculation.

The prescient ability of neural systems originates from the progressive or multi-layered structure of the systems. The information structure can choose (figure out how to speak to) highlights at various scales or resolutions and consolidate them into higher-arrange highlights. For instance from lines, to accumulations of lines to shapes.

3.8.1. Overview of neurons

The building obstruct for neural systems is counterfeit neurons. These are straightforward computational units that have weighted information flags and create a yield flag utilizing an enactment work. You might be comfortable with direct relapse, in which case the weights on the information sources are particularly similar to the coefficients utilized as a part of a relapse condition.

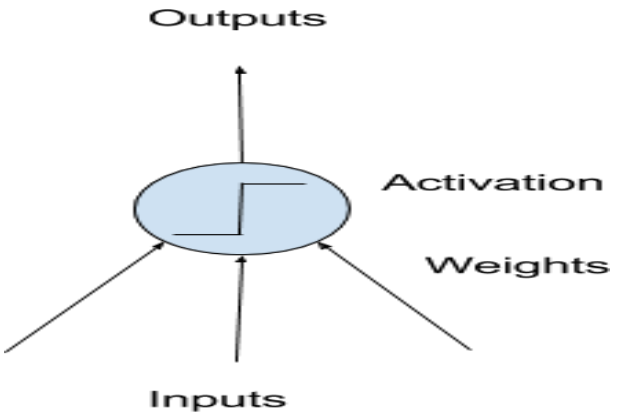

Fig. 6: Neuron Representation.

4. Proposed system algorithm with flow chart

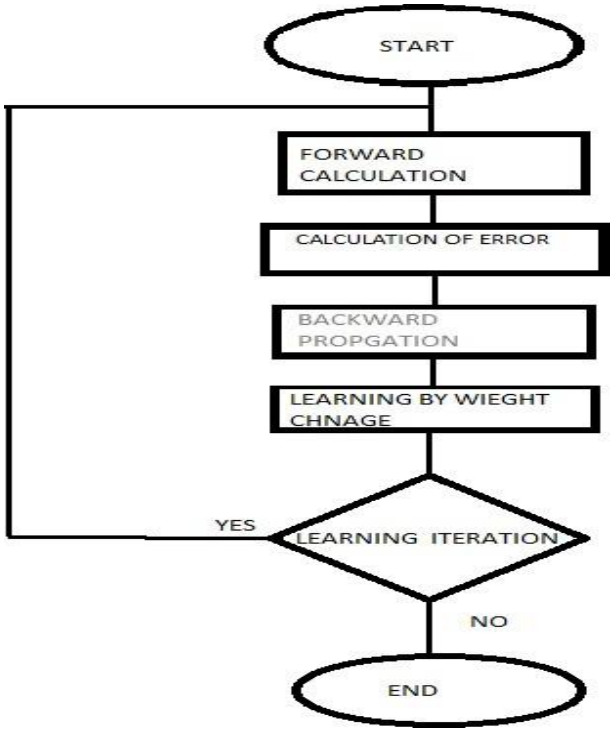

Fig. 7: Multilayer Perceptron Flow Chart.

Multilayer Perceptron works on Two Algorithms. They are

1) Feed forward Algorithm
2) Back Propagation Algorithm.

4.1. Feed forward algorithm Steps

1) Introduce association weights into little irregular esteems.
2) Pass the input Values Let say n values of x as x1, x2, x3.xn along with biased value w0.
3) The output is given as the Summation of weights with corresponding x values with bias values.
4) And it is multiplied with Activation Function sigmoid

1/ (1+e^ (-x))

$$y = \Psi(\Sigma_{i=1}^{n} w_i x_i + b) = \Psi(w^T x + b) \quad (10)$$

5) If the output does not match with predicted value the learning rule is applied to update the weighted values.
6) Error is given as

$$E(\vec{W}) \equiv \frac{1}{2}(t_d - o_d)^2 \quad (11)$$

5. Results and observations

As we discussed above we are classifying whole dataset into four categories by using SVM classifier. The arrangement depends on the cost of the undertaking and KLOC of the task later we foresee the required venture exertion in view of the classification. Let's classify the whole dataset using both kernel methods in SVM.

5.1. SVM with linear and polynomial kernel

Now we take a keen look over data classification using the linear kernel method.

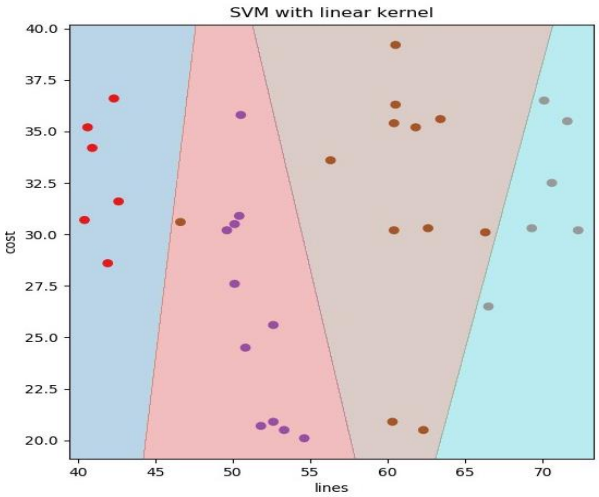

Fig. 8: SVM with Linear Kernel.

From the above fig 8, we can say that dataset has been not too accuracy in the classification. So, we move on to the non-linear classification kernel methods mainly Polynomial and Radial kernel methods.
Now we apply the polynomial kernel method over the dataset in order to know how the data has to be classified. Sometimes the polynomial kernel method also gives the wrong classifier data.

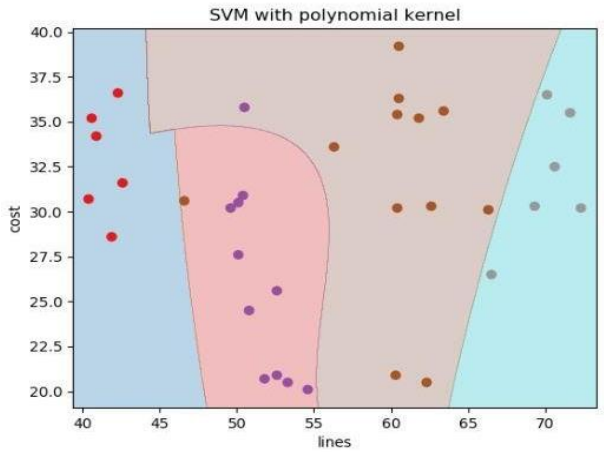

Fig. 9: SVM with Polynomial Kernel Method.

From the above fig 9, we can say that the polynomial kernel is not much accuracy in the classification of the data. So, we have moved to Radial Kernel which is one of the best classifier kernel method using in SVM over the non-linear dataset.

5.2. SVM with radial basic kernel

Now, we apply the Radial kernel method over the dataset and look over the classification accuracy on the dataset. But, in the Radial method the accuracy will be depend on the value of gamma and constant analyzer term C value.

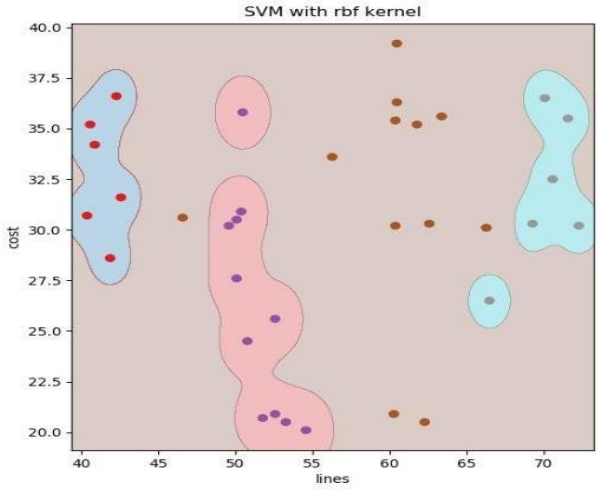

Fig. 10: SVM with RBF Kernel.

From the above fig 10, we can clearly see the classification of our dataset into four categories. We have applied with different gamma and C values into to classify the whole dataset with an high accuracy. At gamma values in range of 0.8 to 0.99 and the value of C ranges in between 10 to 13 gives the best classification accuracy.

5.3. Class prediction with linear and RBF kernel

Now we look over the class prediction the of project [with given parameters KLOC and cost of the project]. We have predict the class with both kernel methods Linear and Radial kernel and on the latter stage we have collected the historical effort values of those predict class values and given as an input to the neural networks.
Latter, we predict the class using radial kernel method by using the same parameters of the project and predict the correct class of the project. The Correct effort of the project will be depend on the accuracy of the prediction of the class. The More accuracy in the prediction of the class the more will be the accuracy in the effort estimation of the project.

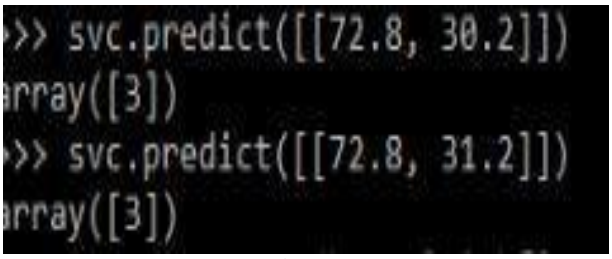

Fig. 11: Prediction of Class with Linear Kernel.

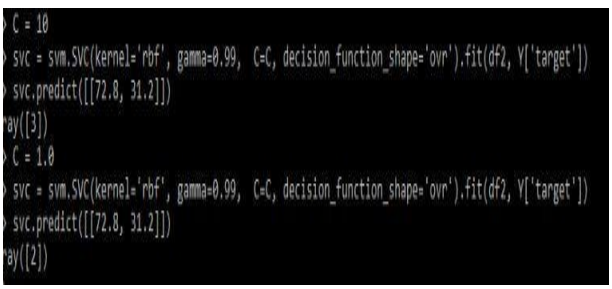

Fig. 12: Class Prediction Using the Radial Kernel Method.

From the above fig11 and fig 12, we have two types of predicted class. Therefore, in order to find out the correct method we have to move on to neural networks where the Effort value has to estimate.

The Linear Kernel method has predicted the class 3 and the radial Kernel Method have predicted the class 2. So, both class Historical effort values has to be given input to neural networks and find out the best kernel method by using accuracy level of effort prediction.

5.4. Effort estimation based on predicted class values

Now, we move on to neural networks with different hidden layers and with different class historical effort values. So, we move with linear kernel method where it has predicted the class 3.

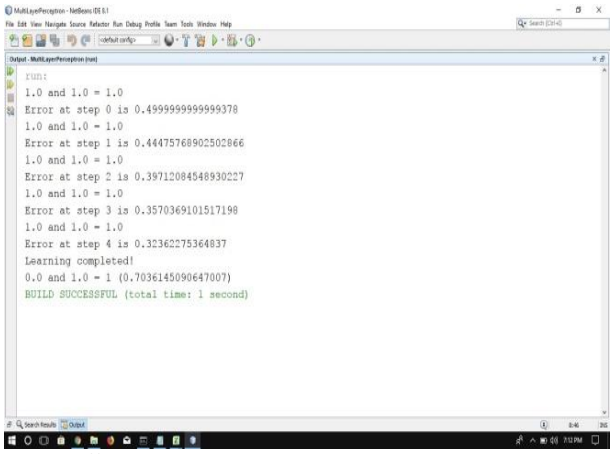

Fig. 13: Effort Prediction with Linear Class Data with 5 Hidden Layers.

From the above fig 13 the output effort value is of 7.03 where our predict value is of 7.1. So, we look over with different hidden layers. Now we arrange the data with 4 hidden layers and find the how much level it is matching over the predicted value with 5 hidden layers. So by applying different hidden layers we find out the best classifier among the linear and the radial Kernel method. The above fig.13, determines error values at every stage until it has reached the nearly zero error. We apply the same concept on 4 hidden layers and find out the accuracy matching over the effort values and compare with 5 hidden layers.

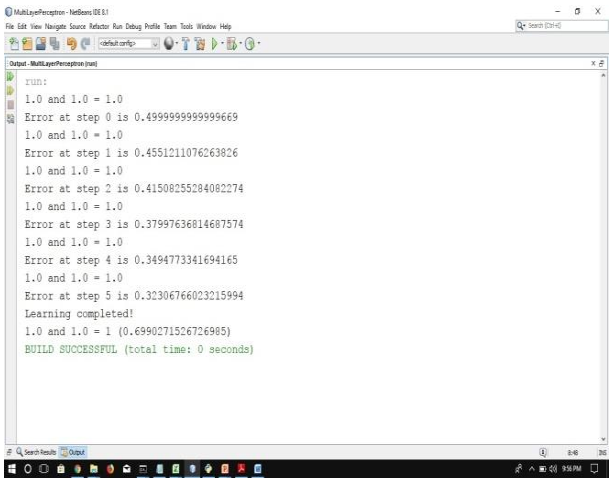

Fig. 14: Effort Prediction with Linear Kernel with 4 Hidden Layers.

From the above Fig 14 we can conclude that by using Radial kernel in SVM and with 5 hidden layers in neural networks we can reach our estimated value nearly.

6. Conclusion and future scope

The project Effort Estimation is a key term in determines the project success. So, Accuracy in measuring effort is difficult in this field. So, Different approaches have been developed where SVM and ANN are the two methods widely using in determine the Effort of software project. Function Point Analysis is another method where it is mainly used to determine the effort at each and every function of the project. So, it involves in high risk over effort prediction.

In this article we have drawn SVM and Artificial Neural Networks over the promise dataset to anticipate the exertion of a venture. The Multilayer perceptron and Radial Kernel function in SVM gives the best effort with different hidden layers and different kernel methods. Other classification methods can be also used in order to determine effort values example fuzzy decision tree is one among the best classifier.

References

[1] G. R. Finnie and G.E. Wittig, “AI Tools for Software Development Effort Estimation,” Proceedings of the International Conference on Software Engineering: Education and Practice (SEEP’ 96).

[2] G. R. Finnie and G.E. Wittig, “A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case Based Reasoning and Regression Models,” Journal of Systems and Software, vol.39, pp.281289, 1997.

[3] A.Tosun, B. Turhan and A.B. Bener, “Feature Weighting Heuristics for Analogy- based Effort Estimation Models,” Expert Systems with Applications, vol. 36, pp.10325-10333, 2009.

[4] M. O. Elish, “Improved Estimation of Software Project Effort using Multiple Additive Regression Tree,” Expert Systems with Applications, vol.36, pp. 10774-10778, 2009.

[5] C. L. Martin, J. L. Pasquier and Cornelio Y M and Agustin G. T., “Software Development Effort Estimation using Fuzzy Logic: A Case Study,” Proceedings of the Sixth Mexican International Conference on Computer Science (ENC’05), IEEE Software, 2005.

[6] P.VidyaSagar,Dr. N.Geethanjali―Multi-objective based Event Based Project Scheduling Model Using EMBACO Algorithm ‖ International Journal of advanced research in computer science and software engineeringVol 5,PP 752-756,Sep2015.

[7] C.J. Burgess and M.Lefley, “Can Genetics Programming improves Software Effort Estimation? A Comparative Evaluation,” Information and Software Technology, vol.43, pp.863-873, 2001.

[8] Parag C. Pendharkar, “Probabilistic estimation of software size and effort,” An International Journal of Expert Systems with Applications, vol. 37, pp.4435-4440, 2010.

[9] L. Radlinki and W. Hoffmann, “On Predicting Software Development Effort Using Machine Learning Techniques and Local Data,” International Journal of Software Engineering and Computing, vol. 2, pp.123- 136, 2010.

[10] I. Attarzadeh and Siew Hock Ow, “Software Development Effort Estimation Based on a New Fuzzy Logic Model,” International Journal of Computer Theory and Engineering, Vol. 1, No. 4, pp.1793-8201, October 2009.

[11] Nageswara Rao Moparthi,Dr. N.Geethanjali―A Novel Privacy Preserving Based Ensemble Cross Defect Prediction Model for Decision Making‖ Perspectives in Science-Elsevier, Vol 8,PP 76-78,April 2016.

[12] K. Srinivasan and D. Fisher, “Machine Learning Approaches to Estimating Software Development Effort,” IEEE Transactions on Software Engineering, vol.21, Feb.1995.

[13] G. Boetticher, T. Menzies and T. Ostrand , PROMISE Repository of Empirical Software Engineering data http://promisedata.org/repository, West Virginia University, Department of Computer Science, 2007.

[14] R. Malhotra, A. Jain, “Software Effort Prediction using Statistical and Machine Learning Methods,” International Journal of Advanced Computer Science and Applications, vol.2, No.1, January 2011.

[15] C. Mair, G.Kadoda, M. Lefley, K.P.C.Schofield, M. Shepperd and Steve Webster, “An Investigation of Machine Learning Based Prediction Systems,” Empirical Software Engineering Research Group, Bournemouth University, U.K. 09 July 1999.

[16] Nageswara Rao Moparthi, Dr. N. Geethanjili“Design and implementation of hybrid phase based ensemble technique for defect discovery using SDLC software metrics”, An International Conference by IEEE, PP. 268 – 274, 2016.