COMPARATIVE ANALYSIS OF SVM & *k*NN FOR ACADEMIC PREDICTION OF STUDENTS

Raj Kumar[1] , Ms. Akshita Sharma[2]

[1]Assistant Professor, Department of CSE Jind Institute of Engineering & Technology, Jind(Haryana)

[2]M.Tech student, Department of CSE Jind Institute of Engineering & Technology, Jind(Haryana)

Abstract: - The various data mining techniques could be effectively implemented on educational data. From results it is clear that classification techniques could be applied on educational data for predicting student's outcome & improve their results. Efficiency of various classification algorithms could be analyzed based on their accuracy & time taken to drive result. Support vector machine is a model for statistics & computer science, to perform supervised learning, methods that are used to make analysis of data & recognize patterns. SVM is mostly used for classification & regression analysis. In same way k-nearest neighbor algorithm is a classification algorithm used to classify data using training examples. In this paper we use SVM & KNN algorithm to classify data & get prediction (find hidden patterns) for target [5]. Here we use educational nominal data to classify & discover data pattern to predict future courses, Uses education mining which is use to classify text analysis in future.

Keywords- SVM, kNN, Patterns, Analysis, Classification, knowledge discovery.

1 INTRODUCTION

Data mining (sometimes called data or knowledge discovery) is process of analyzing data from different perspectives & summarizing it into useful information. Data mining software is a number of analytical tools for analyzing data [1]. It allows users to analyze data from many different dimensions or angles, categorize & summarize relationships identified. Technically, data mining is process of finding correlations or patterns among dozens of fields in large relational databases. Data mining tools predict future trends & behaviors, allowing businesses to make proactive, knowledge-driven decisions. Automated, prospective analyses offered by data mining move beyond analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools could answer business questions that traditionally were too time consuming to resolve [2].

2 EDUCATION DATA MINING

Educational Data Mining refers to techniques, tools & research designed for automatically extracting meaning from large repositories of data generated by or related to people learning activities in educational system settings. Quite often, this data is extensive, fine-grained, & precise [3]. For example, several learning management systems (LMSs) track information such as when every student accessed each learning object, how many times they accessed it, & how many minutes learning object was displayed on user's computer screen. As another example, Intelligent tutoring systems record in the data every time a learner submits a solution to a problem; they may collect time of submission, whether or not solution matches expected solution, amount of time that has passed since last submission, order in which solution components were entered into interface, etc. Precision of this data is such that even a fairly short session within a computer-based learning environment may produce a large amount of process data for analysis.

In other cases, data is less fine-grained. For example, a student's university transcript may contain a temporally ordered list of courses taken by student, grade that student earned in each course & when student selected or changed his or her academic major [4]. EDM leverages both types of data to discover meaningful information about different types of learners & how they learn, structure of domain knowledge, & effect of instructional strategies embedded within many learning environments.

3 RESEARCH METHODOLOGIES

Within explosive growth of data on different domains like education, industries & others required to extract knowledge from data in such manner to explore much knowledge from that data. For that purpose we identify most frequently used data mining algorithm & we found that for classification purpose researchers are go through SVM & K-NN [5].

To work within SVM & K-NN we decide to perform complete task fewer than three steps.

Experimental data selection:

Data selection, data transformation

Different type of data selected as experimental data set. To get performance is varies or not according to data. Here we collect data of different size & different types, like we use data nominal data & numerical data both to evaluate results.

Data analysis using selected data models: Here implementations of algorithms are includes. Data analysis used to different algorithm includes data analysis or model building using both data models.

Result analysis: different system generated resultant parameters are generated. Result analysis includes performance analysis of system on different parameters like accuracy, time taken to build model etc.

4 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine is a supervised machine learning algorithm which could be used for both classification & regression challenges in educational sector.

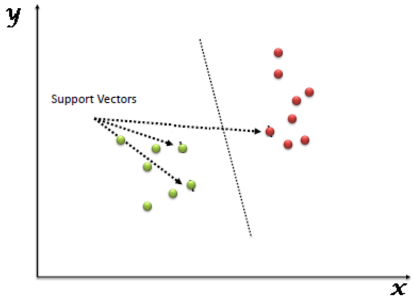


Fig 1 SVM

However, it is mostly used to classification problems. In this algorithm, we plot each Educational mining data item as a point in n-dimensional space (where n is number of features you have) within value of each feature being value of a particular coordinate [6]. Then, we perform classification by finding hyper-plane that differentiates two classes very well. Support Vectors are simply co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates two classes (hyper-plane/ line).

5 KNN

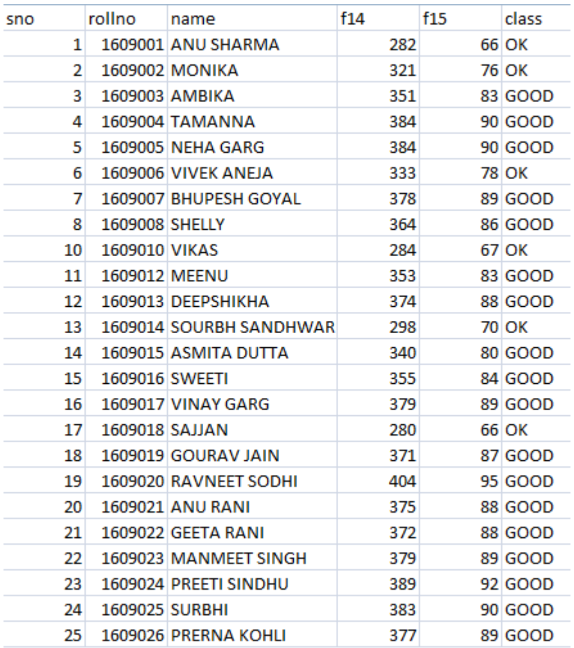
In pattern recognition, *k*-Nearest Neighbors algorithm (or *k*-NN for short) is a non-parametric method used for classification & regression. In both cases, input consists of k closest training examples in feature space. Output depends on whether *k*-NN is used for classification or regression [7]:

In *k-NN classification*, output is a class membership. An object is classified by a majority vote of its neighbors, within object being assigned to class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then object is simply assigned to class of that single nearest neighbor.

In *k-NN regression*, output is property value for object. This value is average of values of its *k* nearest neighbors. *K-nearest* neighbor algorithm is a method for classifying objects based on closest training examples in the feature space. *K-nearest* neighbor algorithm is among the simplest of all machine learning algorithms. Training process for this algorithm only consists of storing feature vectors and labels of the training images.

6 STUDENT PREDICTION USING SVM & KNN

Previous section describes the used algorithms for implementation. Implementation of both algorithms is performed & results are described in this section. We have applied the SVM and KNN classification algorithm to find the classification accuracy. WEKA tool is used for the implementation of the given dataset. We select data for experiment purpose in CSV format. Performance evaluation of both algorithms is obtained by using N cross validation process. And performance analysis is conduct under accuracy, model build time etc. For the prediction we have used student academic record of B.Tech CSE from JIET college batch 2013. Data given in below table represents our Data set which is used to train model and predict them. There attributes are roll no., student's name, total marks, percentage, target class. In machine learning there are two main things first training. The below given data is used to train algorithm actually training is a model building process where we design a data model to negotiate with it. After that we use some data samples to test data. At the time of training required to supply all attributes with Target value which is required to be predicted. After training or model building process we supply the test values to the model and negotiate with model. After negotiation we get the predicted values from the model. In our data set target is some values which is to be predicted. We can say target is our class value.



sno	rollno	name	f14	f15	class
1	1609001	ANU SHARMA	282	66	OK
2	1609002	MONIKA	321	76	OK
3	1609003	AMBIKA	351	83	GOOD
4	1609004	TAMANNA	384	90	GOOD
5	1609005	NEHA GARG	384	90	GOOD
6	1609006	VIVEK ANEJA	333	78	OK
7	1609007	BHUPESH GOYAL	378	89	GOOD
8	1609008	SHELLY	364	86	GOOD
10	1609010	VIKAS	284	67	OK
11	1609012	MEENU	353	83	GOOD
12	1609013	DEEPSHIKHA	374	88	GOOD
13	1609014	SOURBH SANDHWAR	298	70	OK
14	1609015	ASMITA DUTTA	340	80	GOOD
15	1609016	SWEETI	355	84	GOOD
16	1609017	VINAY GARG	379	89	GOOD
17	1609018	SAJJAN	280	66	OK
18	1609019	GOURAV JAIN	371	87	GOOD
19	1609020	RAVNEET SODHI	404	95	GOOD
20	1609021	ANU RANI	375	88	GOOD
21	1609022	GEETA RANI	372	88	GOOD
22	1609023	MANMEET SINGH	379	89	GOOD
23	1609024	PREETI SINDHU	389	92	GOOD
24	1609025	SURBHI	383	90	GOOD
25	1609026	PRERNA KOHLI	377	89	GOOD

Table 1 Shows Total marks & percentage with Target class.

Here, we have taken two classes i.e. GOOD and OK. The students having percentage greater than equal to 80 are classified in the class GOOD and other is classified in OK class. According to this dataset there are 84 instances, 16 Attributes. Where 23 instances are OK and 61 instances are GOOD. We have used a confusion matrix to get the proper result. A confusion matrix is a table that is often used to describe performance of a classification model on a set of test data for which true values are known. In field of machine learning & specifically problem of statistical classification, a confusion matrix also known as an error matrix. It is a specific table layout that allows visualization of performance of an algorithm. The most basic terms of confusion matrix are:

1) TP (True Positives): These are cases in which we predicted GOOD (they have good grades i.e., greater than equal to 80), and they also do have good grades.

2) TN (True Negatives): We predicted OK and they don't have good grades.

3) FP (False Positives): We predicted GOOD, but they don't actually have good grades. Also known as *Type I error.*

4) FN (False Negatives): We predicted OK but they actually do have good grades. Also known as *Type II error.*

5) Positive Predictive Value: This is very similar to precision, except that it takes prevalence into account.

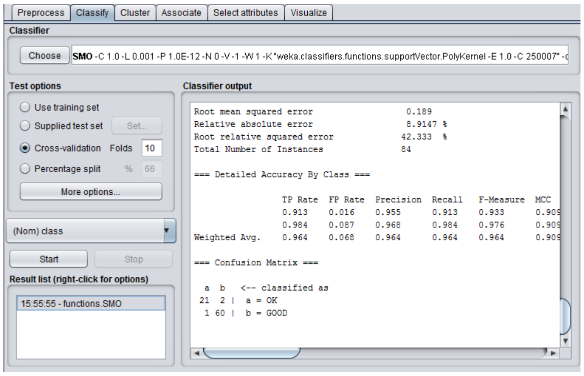
6) Null Error Rate: This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be 23/84=0.27 because if you always predicted GOOD, you would only be wrong for the 23 OK cases).

7) Cohen's Kappa: This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance.

8) F Score: This is a weighted average of the true positive rate (recall) and precision.

9) ROC Curve: This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.

To obtain results we have chosen cross validation process and set the number of folds 10 to find accuracy and other performance parameters of SVM & KNN and also represent confusion matrix using both algorithm. Experimental results & there attributes are shown in below figures.



SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -c

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

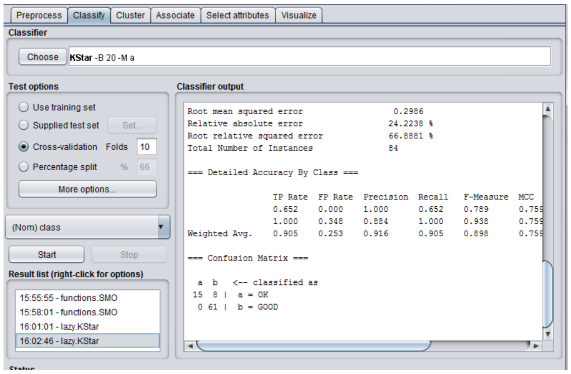
(Nom) class

15:55:55 - functions.SMO

```
Root mean squared error                  0.189
Relative absolute error                  8.9147 %
Root relative squared error             42.333  %
Total Number of Instances               84

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.913    0.016    0.955      0.913    0.933      0.909
                 0.984    0.087    0.968      0.984    0.976      0.909
Weighted Avg.    0.964    0.068    0.964      0.964    0.964      0.909

=== Confusion Matrix ===

  a  b   <-- classified as
 21  2 |  a = OK
  1 60 |  b = GOOD
```

Fig 7 Shows Confusion Matrix using SVM.

Similarly we have selected cross validation to find accuracy of KNN and also represent confusion matrix of KNN.



KStar -B 20 -M a

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

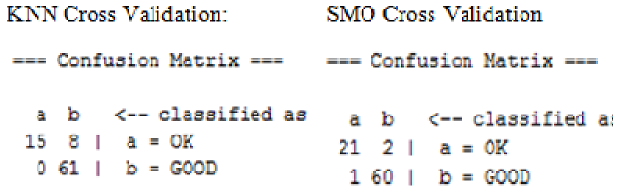
(Nom) class

15:55:55 - functions.SMO
15:58:01 - functions.SMO
16:01:01 - lazy.KStar
16:02:46 - lazy.KStar

```
Root mean squared error                  0.2986
Relative absolute error                 24.2238 %
Root relative squared error             66.8881 %
Total Number of Instances               84

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.652    0.000    1.000      0.652    0.789      0.759
                 1.000    0.348    0.884      1.000    0.938      0.759
Weighted Avg.    0.905    0.253    0.916      0.905    0.898      0.759

=== Confusion Matrix ===

  a  b   <-- classified as
 15  8 |  a = OK
  0 61 |  b = GOOD
```

Fig 9 Shows Confusion Matrix of KNN using WEKA.

As shown in above figures Time taken by SMO to build model is 0.15 sec & Time taken by KNN is 0 sec. In SMO Correct classified instances are 81 whereas in KNN correct classified instances are 76 and Incorrect classified instances of SMO is 3 whereas in KNN are 8. We can see the comparative Analysis result of KNN and SVM in confusion Matrix form.



```
KNN Cross Validation:              SMO Cross Validation

=== Confusion Matrix ===           === Confusion Matrix ===

  a  b   <-- classified as           a  b   <-- classified a
 15  8 |  a = OK                    21  2 |  a = OK
  0 61 |  b = GOOD                   1 60 |  b = GOOD
```

Fig 10 Comparative Analysis result of KNN and SVM in confusion Matrix.

KNN Confusion Matrix:

N=84	Predicted: OK	Predicted: GOOD	
Actual: OK	TN 15	FP 8	23
Actual: GOOD	FN 0	TP 61	61
	15	69	

SVM Confusion Matrix:

N=84	Predicted: OK	Predicted: GOOD	
Actual: OK	TN 21	FP 2	23
Actual: GOOD	FN 1	TP 60	61
	22	62	

Above each column of matrix represents instances in a predicted class while each row represents instances in an actual class. There are two predicted classes i.e., GOOD & OK. Here, GOOD would mean they are good (they have obtained good grades which is greater than 80). And OK would mean they don't have good grades which are below than 80. After evaluation we found the following results of KNN and SVM. The classifier made a total of 84 predictions (84 instances were being tested).

In case of KNN, out of these 84 cases the classifier predicted GOOD 69 times & OK 15 times and in reality, 61 students in the sample have GOOD grades & 23 Students don't. Similarly by applying SVM, out of these 84 cases the classifier predicted GOOD 62 times & OK 22 times and in reality, 61 students in the sample have GOOD grades & 23 students don't. The experimental results rates that are computed are given as:

1) Accuracy: Overall, how often is the classifier correct. Using confusion Matrix accuracy formula i.e., ACC= (TP+TN)/ (Total). In case of KNN, we get (61+15)/84= 0.904 whereas in case of SMO, we get (60+21)/84= 0.964. Which shows KNN gives 90% accuracy whereas SMO gives 96% accuracy. Thus, SVM is more accurate as compare to that of KNN. So from our experiments we conclude that SMO is better than KNN.

2) Misclassification Rate: Overall, how often is it wrong? Also known as **'Error Rate'**. It is calculated by (FP+FN)/ (Total). In case of KNN, we get (8+0)/84= 0.095 whereas in case of SMO, we get (2+1)/84= 0.035. As we can clearly see SMO is better than KNN.

3) True Positive Rate: When it is actually GOOD, how often does it predicted GOOD. It is calculated by TP/Actually GOOD. In case of KNN, we get (61/61) = 1 whereas in case of SMO, we get (60/61) = 0.938. It is also known as **'Sensitivity' or 'Recall'**.

4) False Positive Rate: When it's actually OK, how often does it predicted GOOD. It is calculated by FP/Actually OK. In case of KNN, we get (8/23) = 0.347 whereas in case of SMO, we get (2/23) = 0.086.

5) Specificity: When it's actually OK, how often does it predict Ok. It is calculated by TN/Actually OK. In case of KNN, we get (15/23) = 0.652 whereas in case of SMO, we get (21/23) = 0.913. It is also equivalent to 1 minus False Positive Rate.

6) Precision: When it predicted GOOD, how often is it correct? It is calculated by TP/ Predicted GOOD. In case of KNN, we get (61/69) = 0.884 whereas in case of SMO, we get (60/62) = 0.9677.

7) Prevalence: How often does the GOOD condition actually occur in our sample? It is calculated by Actual GOOD/Total. In case of KNN, we get (61/84) = 0.726 whereas in case of SMO, we get (61/84) = 0.726.

As we can clearly see from above experimental result rates SVM gives better results in most of the cases.

7 CONCLUSION & FUTURE SCOPE

After implementation we found that SVM is more accurate as compare to that of KNN. So from our experiments we conclude that SMO is better as compare to KNN. The various data mining techniques could be effectively implemented on educational data. From results it is clear that classification techniques could be applied on educational data for predicting student's outcome & improve their results. We have implemented the SVM & KNN on the dataset having the 84 records. The SVM and KNN can be implemented on the having the more records. Other machine learning technique can also be used.

REFERENCES

[1] Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", 2nd ed., *Morgan Kaufmann Publishers*, 2006.

[2] Raj Kumar, Akshita Sharma, 27 Jan 2017 "Data Mining in Education: A Review" IJMEIT// Vol.05 Issue 01//January//Page No: 1843-1845//ISSN-2348-196x.

[3] M. El-Halees, "Mining Student Data to Analyze Learning Behavior: A Case Study". In Proceedings of the 2008 *International Arab Conference of Information Technology (ACIT2008)*, University of Sfax , Tunisia, Dec 15- 18.

[4] Cecily Heiner, Ryan Bakery Kalina Yacef, -Proceedings of the Workshop on Educational Data Mining at the 8th *International Conference on Intelligent Tutoring Systems Jhongli* , Taiwan, 2006.

[5] J. S. Raikwal, 14 July 2012, "Performance Evaluation of SVM and K-Nearest Neighbor Algorithm over Medical Data set" *International Journal of Computer Applications (0975 – 8887)* Volume 50 – No.14, July 2012.

[6] Lloyd-Williams, M. —Case studies in the data mining approach to health information analysis, Knowledge Discovery and Data Mining (1998/434), *IEEE Colloquium* on, 8May1998.

[7] Christoper J.C. Burgers, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, 1998, pp.121–167.