ZKG INTERNATIONAL ISSN: 2366-1313

A machine learning method intended to predict a student's academic achievement

Prasadu Peddi, Dr. Akash Saxena

Research Scholar, Arni University, Kangra district, Himachal Pradesh

Research Guide, Arni University, Kangra district, Himachal Pradesh

Abstract: A component of artificial intelligence is machine learning. The core of machine learning is the development of mathematical models that facilitate the analysis and comprehension of data. The mathematical model contains a large number of variables that are changeable. This enables the software to adjust to variations in the data that was created using the mathematical model. Supervised learning and uncontrolled learning are the two primary categories of machine learning algorithms. In supervised learning, labels are applied to both the input and output data during the machine learning process. When supervised learning algorithms are applied, they can generate output independently, without human assistance, by making code modifications. Spam and non-spam texts can be distinguished using supervised learning techniques. Education is an important element of the society, every government and country in the world work so hard to improve this sector. With the corona-virus outbreak that has disrupted life around the globe in 2020, the educational systems have been affected in many ways; studies show that student's performance has decreased since then, which highlights the need to deal with this problem more seriously and try to find effective solutions, as well as the influencing factors.

Keywords: Machine Learning, supervised Learning, Mathematical Model

Motivation

The educational systems need, at this specific time, innovative ways to improve quality of education to achieve the best results and decrease the failure rate.

The method of forecasting student performance using machine learning techniques involves many crucial components. The process starts with collecting data from several sources, followed by preprocessing to eliminate impurities and ready the data. The objective of feature engineering is to find valuable features and, if necessary, to create novel ones. Subsequently, it is necessary to choose a suitable machine learning algorithm that considers the specific challenges and characteristics of the data. Appropriate metrics are used to evaluate the performance of the model once it has been trained using the training data. Performing predictions on new data is carried out once the model is deemed adequate. Model interpretability approaches are used to understand the factors that influence the

ISSN: 2366-1313

predictions. The model is iteratively improved based on input and assessment outcomes. Finally, stakeholders may implement the notion by deploying and integrating it into current educational systems or platforms. The project will use this method to develop a dependable and comprehensible model for forecasting students' performance.

As students of the IT department who studied in the last month a little about machine learning, we know that in order for an institute to provide quality education to learners, deep analysis of previous records of the learners can play a vital role, and wanted to work on this challenging task.

Problematic

As already mentioned, with the help of the old students records, we can came up with a model that can let us help students improve their performance in exams by predicting the student success. So, it is obvious it's a problem of classification , and we will classify a student based on his given informations, and we will also use diffrent classifiers such as KNN or SVM classifier and compare between them. Many factors affect a student performance in exams like famiily problems or alcohol consumption.

1) Predict whether a student will pass his final exam or not.

2) Came up with the best classifier that is more accurate and avoid overfitting and under fitting by using simple techniques.

3) Know what the most factors affect a student performance.

So, teachers and parents will be able to intervene before students reach the exam stage and solve the problems.

Dataset processing

Now before training our model we have to process our data:

1. We have to map each string to a numerical value so that it can be used for model training.

2. Let's take an example to make this clear : The mother job column contain five values : 'teacher',

3. 'health','services', 'at_home' and 'other'.

4. Our job then is to map each of these string to numerical values, and this is how it's done :

5. df['Mjob'] = df['Mjob'].map({'teacher': 0, 'health': 1, 'services': 2, 'at_home': 3, 'other': 4})

6. We have to perform feature scaling :

ISSN: 2366-1313

7. Feature scaling is a method used to normalize the range of independent variables

8. or features of data. In data processing, it is also known as data normalization

9. and is generally performed during the data preprocessing step.

11. This will allow our learning algorithms to converge very quickly. The operation requires

12. to take each column, let's say 'col', and replace it by :

13. But this is not the only scaling you can do, you can try also where std refers to standard deviation.

We will not perform feature scaling for binary columns that contains 0's and 1's .

Dataset visualization

After having our dataset processed, we move to the next step which is dataset visualization; so that patterns, trends and correlations that might not otherwise be detected can be now exposed.

This discipline will give us some insights into our data and help us understand the dataset by placing it in a visual context using python libraries such as: matplotlib and seaborn.

There are so many ways to visualize a dataset. In this project we chose to:

#1- Plot distribution histograms so that we can see the number of samples that occur in

each specific category.

E.g. Internet accessibility at home distribution shows that our dataset consists of

more than 300 students with home internet accessibility,

while there are about 50 students who have no access to the internet at home.

#2- Plot “Boxplots” to see to see how the students status is distributed according to each variable.

#3- Plot the correlation output that should list all the features and their correlations

to the target variable.

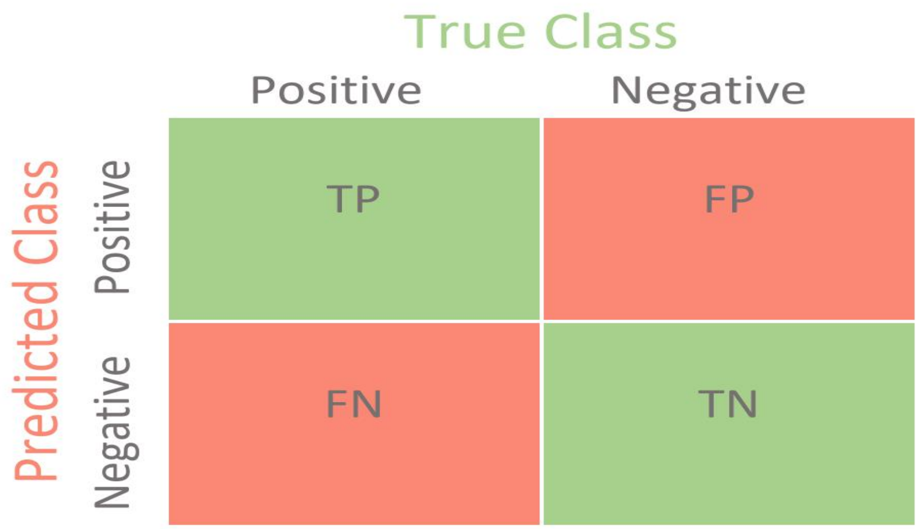
So that we have an idea about the most impactful elements on the students status.

Model evaluation (metrics)

ISSN: 2366-1313

Before starting training our classifers let us define the metrics that we will use to compare between our three classifiers :

1) Confusion matrix :



2) F1 score :

TP = number of true positives

FP = number of false positives

FN = number of false negatives

3) The ROC curve is a graphical depiction that illustrates the diagnostic performance of a binary classifier system as it modifies its discrimination threshold.

4) The ROC score corresponds to the total area enclosed by the ROC curve.

It might be contended that the optimal selection is the value of 1, as the area of a square with dimensions 1x1 is 1.

Logistic regression

KNN

1) Introduction to knn :

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.Its assumes the similarity between the new case data and available cases and put the new case into the category that is most similar to the available

ISSN: 2366-1313

categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

2) Advantages and Disadvantages of Knn algorithm:

The process of setting it up is straightforward.

-It functions even with noisy training data. -It may perform better if the training material is extensive.

-Constantly determining the value of K, which can be challenging at times.

-The expense of computation is substantial due to the necessity of distance-reducing all training sets.

3) What is the mechanism behind its functioning?

Another way to explain how K-NN works is with the following recipe:

First, pick the Kth neighbor.

Step 2: Find the Euclidean distance (or any other distance) between K neighbors.

Step 3: Find the neighborhood of k points and count how many data points are in each group.

For the fourth step, add the new points to the group with the most neighbors.

Now let's say we need to put some new information in the right group. Take a look at this picture:

-To start, we'll choose the number of friends, which is 5.

-The next thing we'll do is find the Euclidean distance between the records.

The Euclidean distance told us that category A had three close neighbors and category B had two.

Look at this picture:

This new data point must be from group A since it is close to three other data points that are also from that group.

4) Python implementation of the KNN algorithm

In this step we will implement knn for our case study by following this step:

a) Data preprocessing step
b) Hyperparameters tuning
c) Fitting the K-NN algorithm to the Training set
d) Predicting the test result
e) Test accuracy of the result

ISSN: 2366-1313

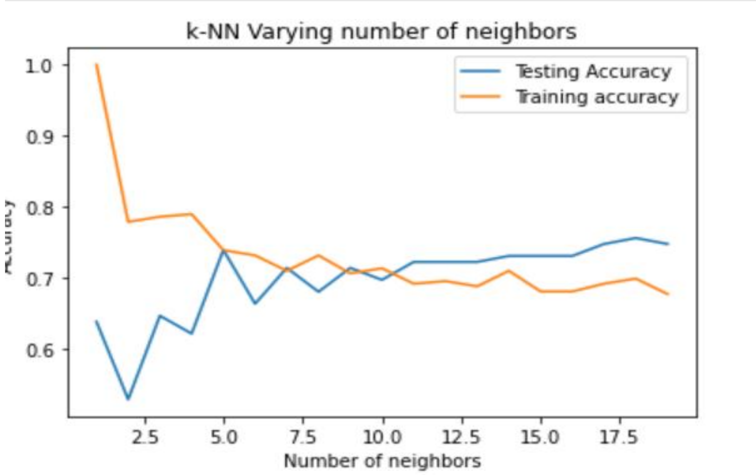
a) Data preprocessing step:

we should look into previous section

b) Hyperparameters tuning:

In this step we look after 2 methods to tune the best parameters for our model First method:

tuning the best k for better test_acquracy and training_acquracy using k-NN Varying number of neighbors plot



-Second method:

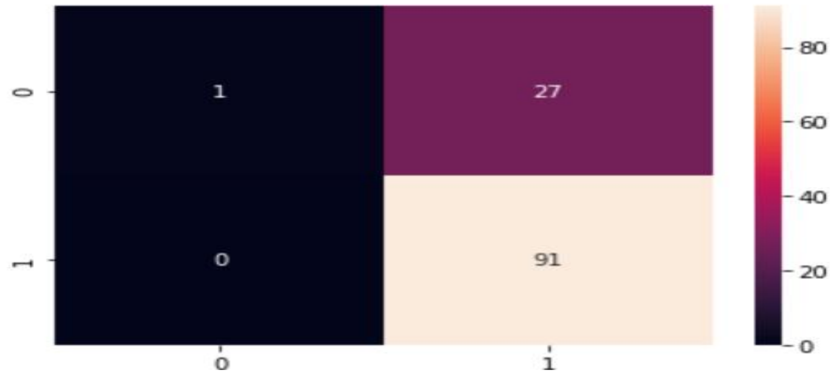
In this method we search for the Best parameters (K,metric=Distance) based on time and accuracy

```
Model_18 metric: euclidean, n_neighbors: 7
Time:  0.012062311172485352
Acc:  0.7394957983193278
Model_19 metric: manhattan, n_neighbors: 7
Time:  0.012945890426635742
Acc:  0.7394957983193278
Model_20 metric: chebyshev, n_neighbors: 7
Time:  0.010972738265991211
Acc:  0.7394957983193278
```

1) Confusion_matrix:

ISSN: 2366-1313

```
Accuracy is:  0.773109243697479
```



2) Classification_report :

2)**classification_report**

```
print(classification_report(y_test,y_pred))
```

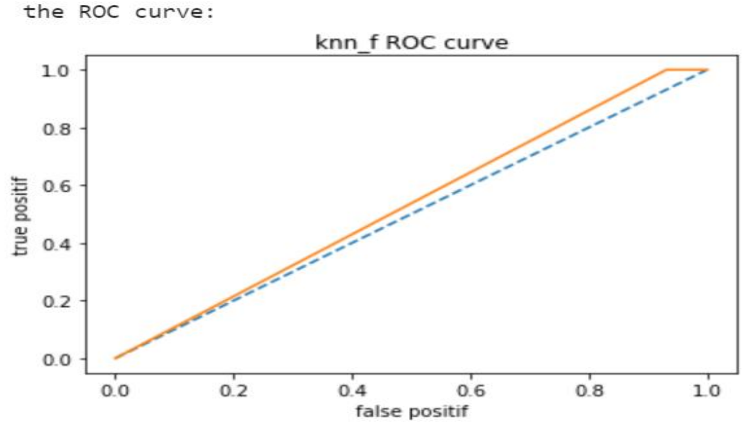
```
              precision    recall  f1-score   support

         0.0       1.00      0.04      0.07        28
         1.0       0.77      1.00      0.87        91

    accuracy                           0.77       119
   macro avg       0.89      0.52      0.47       119
weighted avg       0.83      0.77      0.68       119
```

3) Ploting Roc curv:

```
the ROC curve:
```



SVM

Now we will utilize the Support Vector Machine algorithm to assess its performance on our data. Now, let's explore the concept of the SVM algorithm.

Support-vector machines are a type of supervised learning model that use learning algorithms to analyze data for classification and regression analysis in the field of machine learning.

It utilizes a method known as the kernel trick to carry out data transformation. By employing these transformations, it can identify the optimal boundary among the potential outputs. We will utilize three distinct types of kernels: linear, polynomial, and Gaussian kernels.

1) Linear kernel: The linear kernel is ideal for situations where the data can be easily separated by a single line, making it a great option for linearly separable data. This kernel is widely used in a range of applications.

It is commonly employed when dealing with a significant amount of features in a specific Data Set.

The polynomial kernel is commonly employed in support vector machines and other kernelized models. It evaluates the similarity between vectors in a feature space by considering polynomials of the original variables, enabling the creation of non-linear models.

The Gaussian RBF (Radial Basis Function) is a widely used Kernel method in SVM models for various applications.

The RBF kernel is a mathematical function that calculates its value based on the distance from a specific point or the origin.

Presented below is the format of the Gaussian Kernel:

The similarity between X1 and X2 is determined by calculating their distance in the original space.

The measure of the angle between two points is what determines their similarity.

Here are the visual representations of the three kernels we have discussed:

Below, you will find the metrics and ROC charts for each SVM kernel after training our three models:

But before we proceed, let's delve into some of the parameters utilized in SVM:

If you're looking to enhance the precision of classifying each training example, you have the option to steer the optimization of SVM by specifying the desired weight of the C parameter.

The degree of the polynomial used to determine the hyperplane for data splitting remains unchanged.

The gamma parameter, which serves as a measure of distance, determines the influence of a single training example on the model. Higher values indicate a stronger correlation, while lower values indicate a weaker relationship. One way to grasp the gamma parameters is to consider them as the reciprocal of the radius of influence of the support vectors in the model.

Through a series of iterative tests, we can determine the optimal values for the SVM parameters that minimize the cost on the cross validation set.

ISSN: 2366-1313

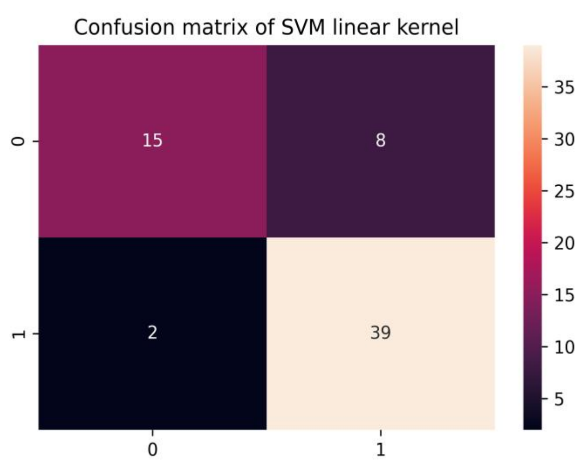
1) Our results :

1) Linear kernel : svm.svc(C)

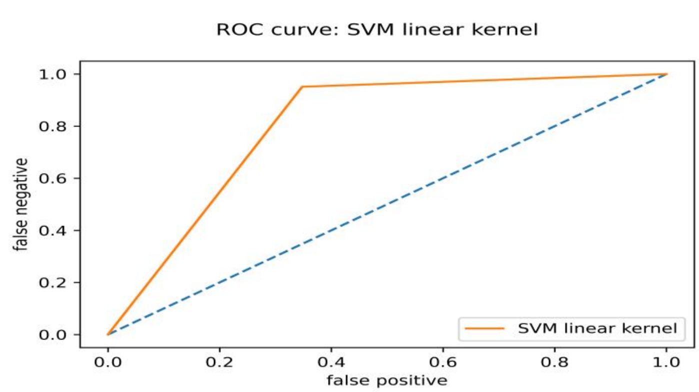
After training our first svm model, here are our results:

training time	10ms
accuracy	84.0 %
f1 score	0.82
The roc_auc_score	0.8

Confusion matrix:



ROC Curve :



ISSN: 2366-1313

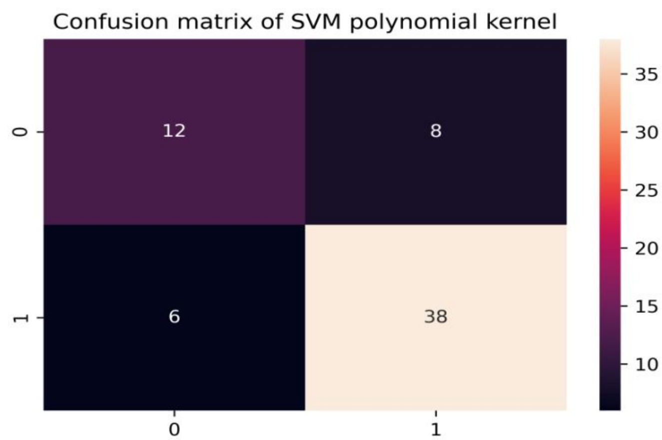
2) Polynomial kernel : svm.svc(C,d)

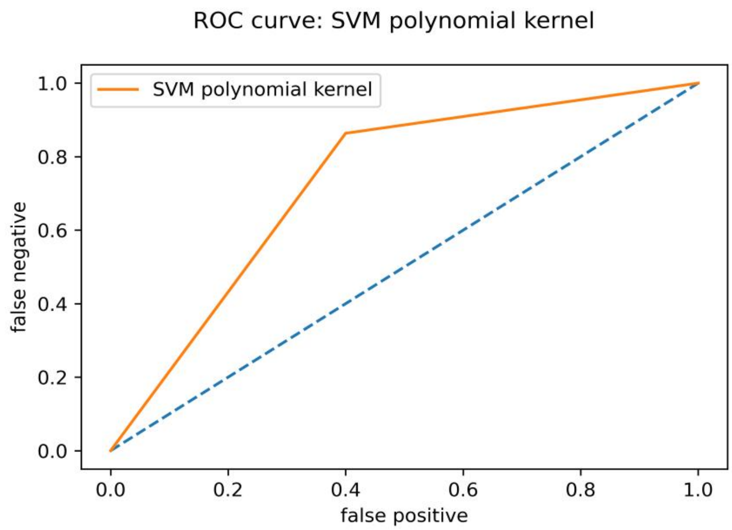
Here we will use diffrent values for both C and d. For example if we have 2 values for C and 3 values for d then we have 2x3 possibilities.

After training our second svm model, here are our results :

Metric	Value
training time	7ms
Accuracy	78.00 %
f1 score	0.74
The roc_auc_score	0.73

Confusion matrix :





ROC Curve :

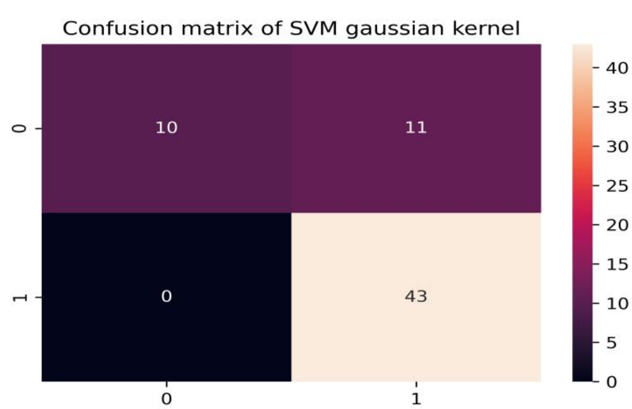
3) Gaussian kernel : svm.svc(C,gamma)

Here we will use diffrent values for both C and gamma.

After training our last svm model, here are our results :

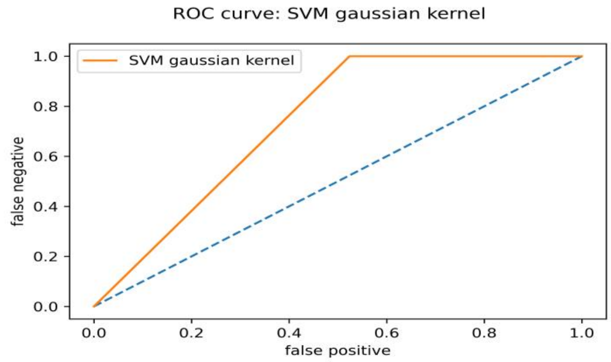
Metric	Value
training time	16ms
accuracy	83.0 %
f1 score	0.77
The roc_auc_score	0.74

Confusion matrix:



ROC Curve:

ISSN: 2366-1313



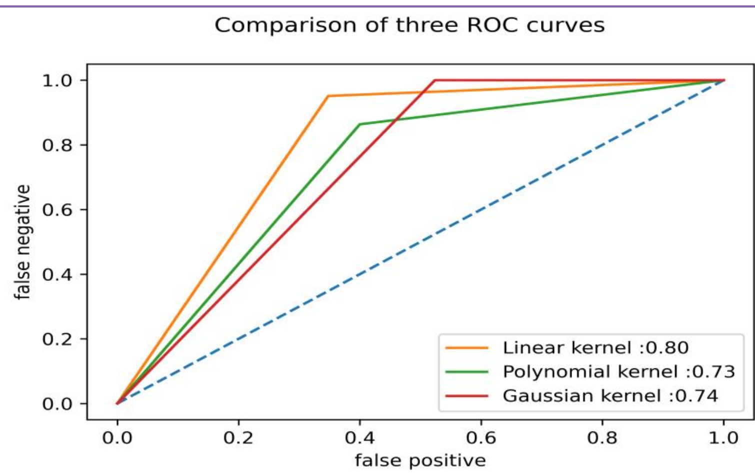
2) Comparison:

Now, after training our three svm models, it is better to group all the results so it can be easy to compare between them :

Now after we train our three models, here are the metrics and the ROC plots :

This table contains all metrics:

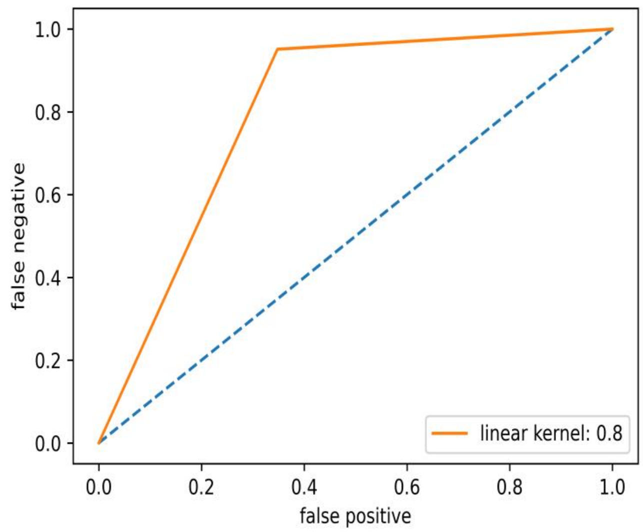
Metric	Linear kernel	polynomial kernel	gaussian kernel
training time	11ms	7ms	3ms
accuracy %	84.375	78.125	82.8125
confusion matrix	[15 8] [2 39	[12 8] [6 38]	[10 11] [0 43]
f1 score	0.82	0.74	0.77
roc_auc_score	0.80	0.73	0.74



3) The more accurate svm classifier :

As you can see the training times are soo small thanks to feature scaling. As you can also notice, the best svm model the one that used the linear kernel.

The roc curve :



As you can see accuracy is pretty good for our problem, but we need to see also that f1 score has a high value so the model performed very well on the test set. Our model is able to generlize for new data and get good prediction. The value of the area under roc curve is approximatly 0.80 which is good, this tells us that model is capable of distinguishing between the two classes.

Conclusion

Improving the education system is a big problem, as an engineering student we can help achieve this goal by using technologies and study resources like machine learning materials, to come up with an innovative solution to help the student in need, especially students who live in difficult conditions. conditions (demographic, social and educational issues). In this project, we came up with the idea of creating a model that predicts the status of students based on different functionalities. Our main challenges were to define the best classification algorithm and identify the most influential factors for the academic status of students to provide them with a summary or valedictorian of the best conditions for students to achieve high academic status and avoid failures. For this project entitled "Prediction of student performance and difficulty" ", we used several classification methods such as logistic regression, KNN and SVM and we evaluate this model using different metrics like f1 score, roc curve and the confusion matrix and finally we got a winner with SVM with a precision of

ISSN: 2366-1313

80%compared to other algorithm . Before taking up our main challenges, there were several steps to take: -data processing -data visualization -Implementation of models -comparison of 3 algorithms.

REFERENCES

A. M. Shahiri, W. Husain, and N. A. Rashid, "A Review on Predicting Student's Performance Using Data Mining Techniques," in Procedia Computer Science, 2015.

P. Guleria, N. Thakur, and M. Sood, "Predicting student performance using decision tree classifiers and information gain," Proc. 2014 3rd Int. Conf. Parallel, Distrib. Grid Comput. PDGC 2014, pp. 126–129, 2015

Sudjana, N. (2010). Penilaian Hasil Proses Belajar Mengajar.(Cet. XV). Bandung: PT. Remaja Rosdakarya

Sukmadinata, N. S. (2009). Landasan Psikologi Proses Pendidikan: Bandung. Remaja Rosdakarya.

P. M. Arsad, N. Buniyamin, and J. L. A. Manan, "A neural network students' performance prediction model (NNSPPM)," 2013 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2013, no. July 2006, pp. 26–27, 2013

K. F. Li, D. Rusk, and F. Song, "Predicting student academic performance," Proc. – 2013 7th Int. Conf. Complex, Intelligent, Softw. Intensive Syst. CISIS 2013, pp. 27–33, 2013.

G. Gray, C. McGuinness, and P. Owende, "An application of classification models to predict learner progression in tertiary education," in Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014, 2014.

Harlen, W. (2005). Trusting teachers' judgement: Research evidence of the reliability and validity of teachers' assessment used for summative purposes. Research papers in education, 20(3), 245- 270. doi: 10.1080/02671520500193744.

K. R. Wentzel and A. Wigfield, "Academic and Social Motivational Influences on Students Academic Performance", Educational Psychology Review, *vol. 10, no. 2, pp. 155-175, 1998.*

L. Rokach and O. Maimon, Data Mining with Decision Trees: Theory and Applications, World Scientifc Publishing, 2007.

ISSN: 2366-1313

Prasadu Peddi (2015) "A review of the academic achievement of students utilisinglarge-scale data analysis", ISSN: 2057-5688, Vol 7, Issue 1, pp: 28-35.

I. Rish, "An empirical study of the naive Bayes classifier", IJCAI 2001 workshop on empirical methods in artificial intelligence, *vol. 3, no. 22, pp. 41-46, 2001.*

T. Hastie, R. Tibshirani, J. Friedman and J. Franklin, "The elements of statistical learning: data mining inference and prediction", The Mathematical Intelligencer, *vol. 27, no. 2, pp. 83-85, 2005.*