Student Placement Predictor: Analysis Using Machine Learning Techniques

Bharti Tayal
Faculty of Engineering, Environment and
Computing, Coventry University
MSc Data Science and Computational
Intelligence
Coventry, United Kingdom
tayalb@uni.coventry.ac.uk

***Abstract*—Everyone dreams to get a job offer before leaving their college. In this study, the objective is to analyze data of previous batches' students and use it to forecast the placement probability of the current batch. In this paper, prediction has been done depending on certain features of a student who joins a college such as Degree Type, Stream chosen in HSC, MBA Specialization and many more. The system is based on SVM, KNN and Random Forest algorithms. The performance of the classification techniques is compared by providing and discussing their expected classification accuracy on testing data, which is visualized using confusion matrices. Techniques and algorithms were implemented using Python programing language utilizing machine learning repositories, prediction results and algorithm performance measures were obtained, and visualized for comparison and discussion.**

Keywords—Supervised Learning, KNN, SVM, Prediction, Machine Learning, Random Forest.

I. Introduction

Placement is one of the paramount factor for each and every institute. Major chunk of the students prefers to enroll themselves in the colleges by analyzing the placement rate of the college. Therefore, in this context the method is about the prediction and analyses for the placement necessity in the colleges which facilitates the institutes as well as students to improve their placements [1]. Using placement prediction system, students can have a rough estimation about their current level and what is required further to seek a satisfactory job. Consequently, students will work diligently taking steps towards profession of their choice. With the help of this system, mentors as well as placement team of institution will also be able to denote appropriate care towards the advancement of students during their course. This system has a remarkable place in educational system as every educational institution wants to have good reputation and admissions in which placement rate plays major role. This study uses Supervised Machine Learning Classification techniques: Support Vector Machine (SVM), Random Forest, and K-Nearest neighbors (KNN) to provide effective and precise results.

II. Dataset Description

This dataset has been extracted from Kaggle dataset and used in the study. This chosen dataset includes information on gender, SSC and HSC board and percentage, Stream in HSC, Degree type and percentage, work experience, MBA specialization and percentage and status of placement of previous students. There are 15 attributes in the dataset which consist a string, real and integer values. Moreover, it contains 215 number of records. The principal aim of the proposed system is to analyse if student will get employment or not; so that educational institutions can provide special attention and help students to get job. Table1 displays the summary of the data set describing features and feature types.

TABLE1. DATASET FEATURES

No	Description	Type	Categorical Value Range
1	Serial No	Numeric	
2	Gender	Categorical	Male='M',Female='F'
3	SSC Percentage	Numeric	
4	SSC Board (10th Grade)	Categorical	Central/ Others
5	HSC Percentage	Numeric	
6	HSC Board (12th Grade)	Categorical	Central/ Others
7	HSC Stream	Categorical	Commerce, Science, Arts
8	Degree Percentage	Numeric	
9	Degree Type	Categorical	Sci&Tech,Comm &Mgmt, Others
10	Work Experience	Categorical	Yes/No
11	Employability test percentage (conducted by college)	Numeric	
12	MBA Specialization	Categorical	Mkt&HR, Mkt&Fin
13	MBA Percentage	Numeric	
14	Status of Placement	Categorical	Placed/Not placed
15	Salary	Numeric	

III. Related Work

Ajay Shiv Sharma, Swaraj Prince, Shubham Kapoor, Keshav Kumar used Logistic regression technique on their college placement dataset which got 83.33% of accuracy [2].

Jai Ruby, Dr. K. David used ID3, J48, REP Tree, NB Tree, MLP, Decision Table Classification techniques on the placement dataset collected from their college. The results had shown that ID3 predicted well among them with an accuracy of 82.1% [3].

Ankita A Nichat, Dr. Anjali B Raut used C4.5 classification technique on the placement dataset which was collected from their college which got 80% of accuracy [4].

Oktariani Nurul Pratiwi used J48, Simple cart, kstar, SMO, NaiveBayes, OneR classification techniques on the data gathered from their high school. The results had shown that J48 and Simple Cart predicted well among them with an accuracy of 79.61% [5].

IV. Methodology

Machine Learning deals with the development, analysis and study of algorithms that can automatically detect patterns from data and use it to predict future data or perform decision making [6]. Machine Learning has various applications such as image recognition, Self-driving cars, Virtual Personal Assistance, Computational Finance, Search engine etc.

There are many types of learning available in Machine Learning, but generally, only two of them are taken into consideration; which are Supervised Machine Learning and Unsupervised Machine Learning. In Supervised Machine Learning, Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as supervised learning problems[7]. There are two main types of Supervised Learning problems: they are Classification which involves predicting a class label and regression which involves predicting a numerical value. In unsupervised learning, there is no instructor or teacher, and the algorithm must learn to make sense of the data without this guide. There are many types of Unsupervised learning, although there are two main problems: they are Clustering that involves finding groups in the data and density estimation that involves summarizing the distribution of data[8].

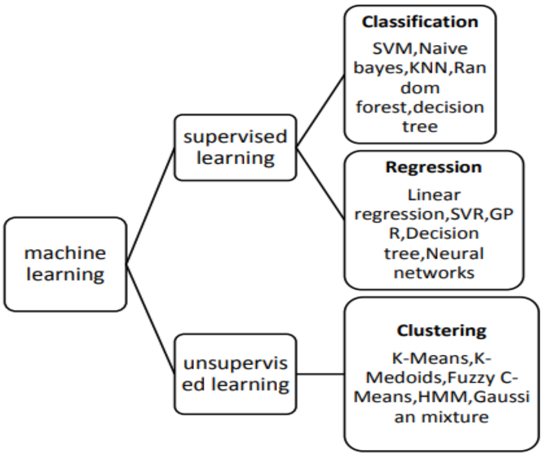

machine learning
supervised learning
unsupervised learning
Classification
SVM,Naive bayes,KNN,Random forest,decision tree
Regression
Linear regression,SVR,GPR,Decision tree,Neural networks
Clustering
K-Means,K-Medoids,Fuzzy C-Means,HMM,Gaussian mixture

Fig. 1. Some Important Machine Learning Algorithms

From the above mentioned machine learning models, I have used Classification Techniques: Support Vector Machines, KNN, Random Forest.

A. Support Vector Machines

SVM is used to classify instances by linearly separating them with the highest margin possible between the class instances. If the features of data are not linearly separable, an SVM algorithm can pre-process the data and represent the features in a higher-dimensional space in which they can become linearly separable [9]. Examples of linear separation and non-linear separation utilizing kernel displayed in Fig 2 Fig. 3. Figures are for example purposes and do not represent placement prediction dataset.

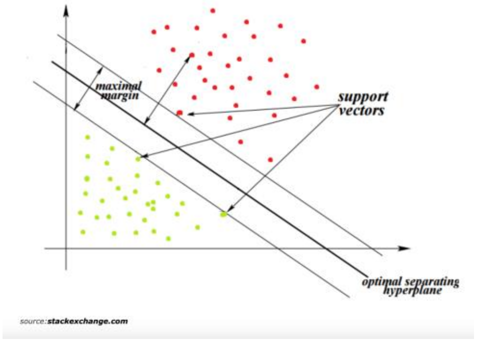


Fig. 2. SVM Linear Separation
(Source:www.stackexchange.com)

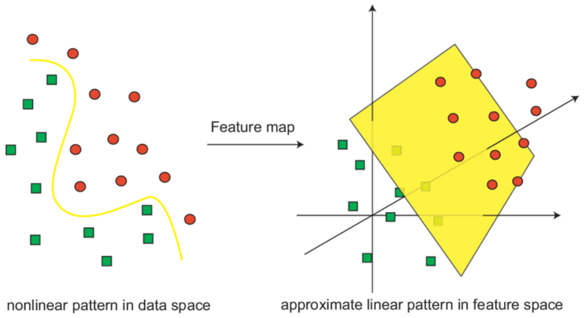

Feature map
nonlinear pattern in data space
approximate linear pattern in feature space

Fig. 3. SVM Non-Linear Separation Using Kernel
(Source:www.researchgate.net)

B. K-Nearest Neighbours (KNN)

The KNN algorithm is based on assumption that things which are alike reside in close proximity.

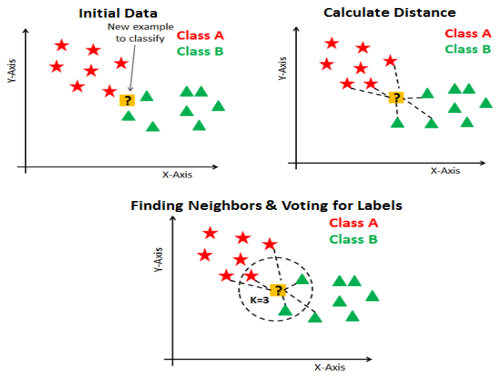


Fig. 4. KNN Illustration
(Source: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn)

KNN is recommended to be used on models where there are not too many attributes that define the class, so it seems to fit this model. A KNN algorithm classifies instances by locating the k-nearest instances in the training data. The most common value for the target value amongst those k instances is then chosen as the classification for any new instance.

However, there is no mathematical formula to determine the correct value of k and therefore when using a KNN algorithm to classify the given data set, k is chosen by experiment with different values and trying to maximize the correct classification rate over multiple values. KNN algorithm uses Euclidean distance to describe the distance between each instance of data. For two points p and q, the distance d can be represented by the following formula (1).

$$d(p, q) = d(q, p)$$
$$= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \qquad (1)$$

It is important that during data pre-processing, the data is scaled appropriately (normalized/standardized).

C. Random Forest

Random forest algorithm can also be considered as an ‘Ensemble Method’ in machine learning.

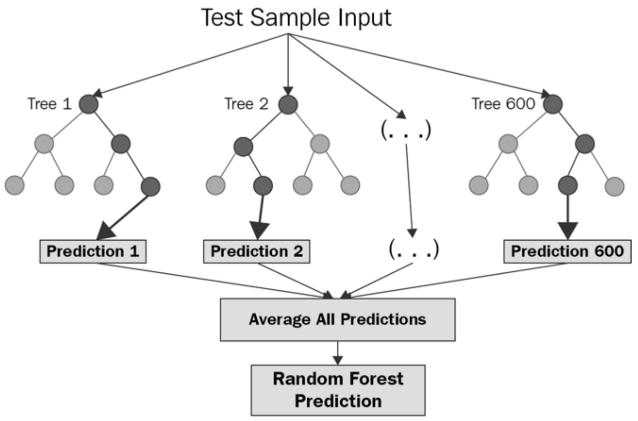


Fig. 5. Random Forest Illustration
(Source: https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f)

Random Forest is a popular Regression and Classification algorithm. It makes use of Decision Trees beneath and forms multiple trees and eventually takes majority vote out of it. Dataset consisting of records are given as an input to Random Forest algorithm. Afterwards, random subsets are formed from the input received. A Decision Tree will be constructed on each of the created random subset. This technique will choose the final class of test record which has the majority votes. Random forest algorithm makes use of the bagging and feature randomness.

V. Experimental Setup

To perform the experiments, this project uses Python programming language with the use of scientific libraries. The code is tested by using Jupyter Notebook (interactive computer environment) that supports IPython command shell for Python as well as other programming languages like R or Ruby[10].

The entire process can be categorised into following steps:

- Gathering Data
- Pre-processing
- Processing
- Interpretation

This entire technique is depicted below in a Fig .6:

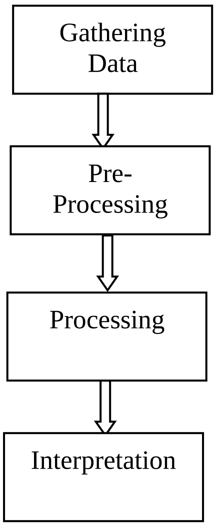


Fig. 6. Flowchart of Technique

A. Gathering Data

This data has been taken from Kaggle dataset and used in the study. This chosen dataset includes 15 attributes which consists a string, real and integer values. The collected dataset constitutes 215 records of students who have completed their study previously.

B. Pre processing

In this step, raw data is converted into a clean dataset. The data which is gathered is in raw format which is not suitable for analysis.

Pre processing involves following simple yet effective steps:

1) Attribute Selection

Some features that were not suitable to the experiment goal were ignored from the initial dataset. The attributes SSC Board, HSC Board,HSC Specialisation, Degree Type, Salary are dropped. The main attributes used for this study are gender, SSC Percentage, HSC Percentage, Degree Percentage, Work Experience, Employability Test Percentage, Specialization, MBA Percentage, Status.

2) Cleaning missing values

Sometimes, dataset consists of missing values. If they are less in number, rows containing missing values can be dropped. But in case of large number of missing values, it is recommended to deal with the missing data in the other way. The other options available for dealing with missing values are to:

- Impute missing values by replacing the blank fields with sensible values. One of the methods is to perform zero impute where the records are filled with zero value. The other way is to replace the records with the average value of the attribute.
- Use algorithms, which support missing values in the dataset.

In this dataset, only salary attribute consists of some null values which was dropped off while removing irrelevant attributes.

3) *Categorical Feature Encoding*

Machine Learning models work very well for dataset having only numeric values, but categorical data can be represented as text. Categorical data must be encoded into binary values. In this paper, four categorical features have been converted to numeric one.

TABLE 2. ENCODING CATEGORICAL FEATURES

Gender	**Numeric**
Male	0
Female	1

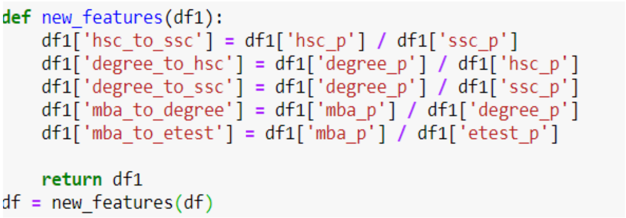
Work Experience	**Numeric**
No	0
Yes	1

Status	**Numeric**
Not Placed	0
Placed	1

Specialisation	**Numeric**
Mkt&HR	0
Mkt&Fin	1

4) *Additional Feature Generation*

Feature generation is the process of creating new features from one or multiple existing features, potentially for using in statistical analysis. This process adds new information to be accessible during the model construction and therefore hopefully result in more accurate model. Feature generation can improve model accuracy when there is a feature interaction. By adding new features that encapsulate the feature interaction, new information becomes more accessible for the prediction model (PM) [10].



```
def new_features(df1):
    df1['hsc_to_ssc'] = df1['hsc_p'] / df1['ssc_p']
    df1['degree_to_hsc'] = df1['degree_p'] / df1['hsc_p']
    df1['degree_to_ssc'] = df1['degree_p'] / df1['ssc_p']
    df1['mba_to_degree'] = df1['mba_p'] / df1['degree_p']
    df1['mba_to_etest'] = df1['mba_p'] / df1['etest_p']

    return df1
df = new_features(df)
```

Fig. 7. Additional Features

5) *Feature Scaling*

It is a method used to standardize the range of independent variables or features of data. In order to evaluate each feature equally, Scaling must be performed to ensure all the features are on the same scale and none of the features are given higher weighting by the algorithms. This could happen as 100 is not the same as 1 and in the formula 100 would have a higher weighting for prediction. Standardization is done by subtracting the mean from each feature and divide by the standard deviation. Standardization is applied on every observation of the selected column; resulting in fitting it to a scale.

In this paper, standardization was completed utilizing python Standard Scaler class from preprocessing module in scikit-learn library.

6) *Dimensionality Reduction and Feature Extraction*

Dimensionality reduction refers to techniques in which the number of input variables in training data are reduced. Due to reasonable dataset size and performance, it was decided that there were no requirements to perform feature engineering such as PCA or regression feature elimination.

7) *Training and Test data*

This procedure is used to calculate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. In this, dataset is divided into two subsets viz, training subset; used to fit the model. The second subset is test dataset. Some inputs from test set are given to model and predictions are made which is then compared to the expected values.

Scikit-learn function train_test_split has been used; which shuffles the dataset and separates it into variables for attributes and labels being trained and tested. The function allows to determine the size of training (test_size) and testing set. In this project, 70% of data is used for training, and the rest (30%) is used for testing. The training data sample had a size of 150 tuples and the testing dataset had a size of 65.

C. *Processing*

In Processing, some classification algorithms are applied using predefined functions from scikit-library, which allows performing the classification quickly and precisely, by creating a classifier and training it. This project implements three classification methods for the supervised learning process:

- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Random Forests

The training dataset containing the placement data of previous students is loaded to the python code, fit to aforementioned classifier model using Scikit libraries. Once the modelling is completed, the test data is uploaded to the python to a predict function available in Scikit learn.

VI. EXPERIMENTAL RESULTS

Models were trained using parameters that deliver the best results. Confusion Matrix heatmap along with Classification Report containing Accuracy, Precision, Recall, F1-score, Support are presented for each model. Precision evaluates the measure of correct positive predictions made by the model.

Precision=TruePositives/(TruePositives+FalsePositives) (2)

It gives value between 0.0, when there is no precision, and 1.0, in case of full or perfect precision.

Recall evaluates the measure of correct positive predictions made out of all positive predictions that could have been made by the model. It provides an indication of missed positive predictions.

Recall = TruePositives/(TruePositives + FalseNegatives) (3)

It gives value between 0.0, when there is no recall, and 1.0 , in case of full recall.

A. K-Nearest Neighbor Classifier

Using python libraries K-Nearest Neighbor classifier was trained and results presented in Fig. 8 and Fig. 9.

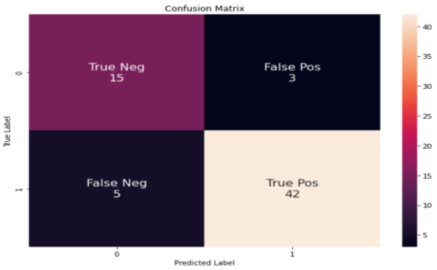
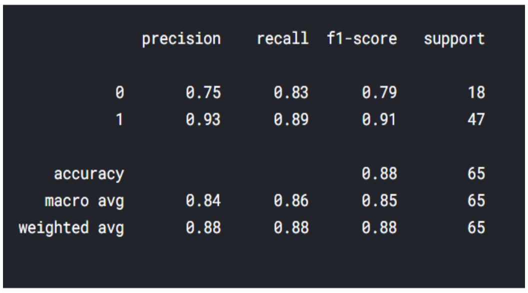


Fig. 8. Confusion Matrix K-Nearest Neighbor Classifier



```
              precision    recall  f1-score   support

           0       0.75      0.83      0.79        18
           1       0.93      0.89      0.91        47

    accuracy                           0.88        65
   macro avg       0.84      0.86      0.85        65
weighted avg       0.88      0.88      0.88        65
```

Fig. 9. Classification Report K-Nearest Neighbor Classifier

The accuracy achieved was 87% with number of neighbors as 3 due to small size of dataset.

B. Support Vector Machine (SVM)

Using python libraries SVC classifier, support vector machine model was trained and results presented in Fig. 10 and Fig. 11.

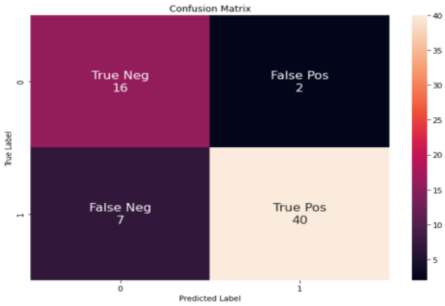
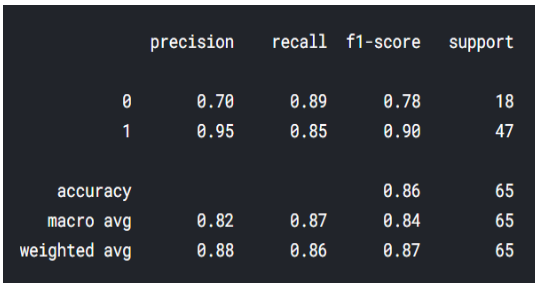


Fig. 10. Confusion Matrix Support Vector Machine Classifier



```
              precision    recall  f1-score   support

           0       0.70      0.89      0.78        18
           1       0.95      0.85      0.90        47

    accuracy                           0.86        65
   macro avg       0.82      0.87      0.84        65
weighted avg       0.88      0.86      0.87        65
```

Fig. 11. Classification Report Support Vector Machine Classifier

The accuracy achieved was 86%. From above two Classification Reports, it can be concluded that K-nearest model results were slightly more accurate in comparison to SVM.

C. Random Forest

Random forest algorithm from the scikit-learn repository in python was trained, tested and results obtained. Confusion matrices along with Classification Report are presented in Fig. 12, and Fig. 13 respectively.

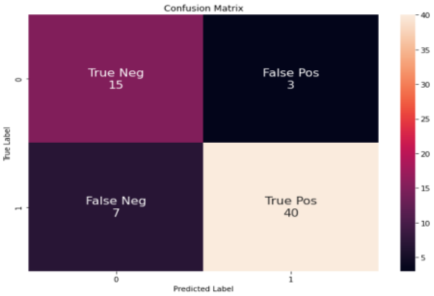
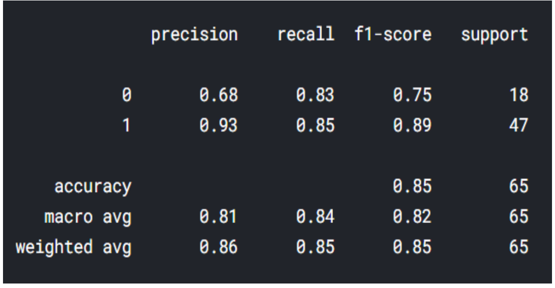


Fig. 12. Confusion Matrix Random Forest Classifier



```
              precision    recall  f1-score   support

           0       0.68      0.83      0.75        18
           1       0.93      0.85      0.89        47

    accuracy                           0.85        65
   macro avg       0.81      0.84      0.82        65
weighted avg       0.86      0.85      0.85        65
```

Fig. 13. Classification Report Random Forest Classifier

The accuracy achieved was 85% with 1000 number of estimators.

VII. Discussion and Conclusions

Without any doubt, K-Nearest Neighbor outperformed Support Vector Machine and Random Forest algorithms with the accuracy of 87%. However, there is slight difference in performances of these models, especially between Support Vector Machine and Random Forest.
Comparison of used Algorithms is shown below in Table 3.

TABLE 3. COMPARISON OF PERFORMANCE OF VARIOUS ALGORITHMS

Algorithms	**Accuracy**	**Precision**	**Recall**
K-Nearest Neighbor	88%	0.93	0.89
Support Vector Machine	86%	0.95	0.85
Random Forest	85%	0.93	0.85

The campus placement activity is incredibly crucial for education providers as well as students. It is not feasible for TPO to predict placement status manually for each student. To overcome this issue, we can use data mining to help in predicting the student's placement. This proposed system implements a student placement prediction system which predicts particular student is placed or not with the help of these three algorithms – SVM, Random Forest and KNN. Many research papers based on educational sector have been published, which mainly emphasize on student's performance predictions. All these forecasts help the institute to improvise the student performance and can come up with best placement results. Many of the previous research papers concentrate on less number of parameters such as CGPA and Arrears for placement status prediction which leads to less accurate results, but proposed work contains many educational parameters to predict placement status which will be more accurate.

VIII. Future Enhancements

The further advancement which can be done in this project is to concentrate on adding some additional features such as, any internship done, communication proficiency, and many more to predict better employment results. Additionally, some solutions or suggestions for the user can also be predicted by the model which will enhance the project. Undoubtedly, trained model already performed well, however deep learning methods or neural networks could be applied to it in attempt to get better performance.

IX. Appendix

Original dataset and python programming files are available at this link:
https://github.com/bhartikapoor9634/ML_CW

References

[1]. Mangasuli Sheetal B, Prof. Savita Bakare "Prediction of Campus Placement Using Data Mining Algorithm Fuzzy logic and K nearest neighbour" International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 6, June 2016 .

[2]. Ajay Shiv Sharma, Swaraj Prince, Shubham Kapoor, Keshav Kumar "PPS-Placement prediction system using logistic regression" IEEE international conference on MOOC,innovation and Technology in Education(MITE), December 2014.

[3]. Jai Ruby, Dr. K. David "Predicting the Performance of Students in Higher Education Using Data Mining Classification Algorithms - A Case Study" International Journal for Research in Applied Science & Engineering Technology (IJRASET) Vol. 2,Issue 11,November 2014.

[4]. Ankita A Nichat, Dr.Anjali B Raut "Predicting and Analysis of Student Performance Using Decision Tree Technique" International Journal of Innovative Research in Computer and Communication Engineering V0l. 5, Issue 4, April 2017.

[5]. Oktariani Nurul Pratiwi "Predicting Student Placement Class using Data Mining" IEEE International Conference 2013.

[6]. Kohavi, R. and F. Provost(1998) Glossary of terms. Machine Learning 30:271-274.

[7]. Pattern Recognition and Machine Learning, 2006 (Page 3) by Christopher M. Bishop.

[8]. Deep Learning, 2016 (Page 105) by Lan Goodfellow, Yoshua Bengio, and Aaron Courville.

[9]. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," IEEE Intelligent Systems and their applications, vol. 13, no. 4, pp. 18–28, 1998.

[10]. Jupyter Team (2015) What Is The Jupyter Notebook? — Jupyter Notebook 5.2.2 Documentation [online] available from [8 December 2017]

[11]. Suzanne van den Bosch: "Automatic feature generation and selection in predictive analytics solutions", Master thesis Computing Science Faculty of Science, Radboud University