STUDYING DATA MINING TOOLS AND TECHNIQUES FOR PREDICTING STUDENT PERFORMANCE

[1]Prasadu Peddi, [2]Dr. Akash Saxena

[1]Research Scholar, Arni University, Kangra district, Himachal Pradesh

[2]Research Guide, Arni University, Kangra district, Himachal Pradesh

ABSTRACT: The goal of data mining's prediction process is to foretell potential outcomes by identifying patterns and correlations in the collected data. Exam scores, dropout rates, and the probability of success in a specific course are just a few of the student performance metrics that may be predicted using prediction algorithms in the context of student data. It should be remembered that prediction models are far from flawless, the accuracy of these models relies on the appropriateness of the algorithm used and the quality and representativeness of the training data. To keep the prediction models relevant and successful in predicting student achievement, they need to be monitored and updated often. The goal of data mining is to extract useful information from massive data sets by identifying patterns and correlations. Data mining is the process of discovering useful information in datasets via the use of different algorithms and approaches. Data mining seeks to unearth latent patterns, foretell forthcoming trends, and enable data-driven decision-making. It helps businesses in many different sectors, including retail, healthcare, marketing, and finance, to stay ahead of the competition.

Keywords: Data Mining, Prediction, Data mining Algorithms.

I. INTRODUCTION TO DATA MINING

Data mining is the process of discovering patterns, correlations, and anomalies within large datasets to predict outcomes. Using a combination of statistics, machine learning, and database systems, data mining transforms raw data into useful information. The primary steps involved in data mining are:

1. **Data Cleaning**
2. **Data Integration**
3. **Data Selection**:
4. **Data Transformation**
5. **Data Mining**
6. **Pattern Evaluation**
7. **Knowledge Presentation**

Data Cleaning

Data cleaning, also known as data cleansing or data scrubbing, is the process of identifying and correcting (or removing) inaccuracies, inconsistencies, and errors in a dataset. It is a crucial step in data preparation, as the quality of data significantly impacts the outcomes of data analysis and mining. Here are the key aspects and techniques involved in data cleaning:

Key Aspects of Data Cleaning

1. **Handling Missing Data**:
   - **Imputation**: Filling in missing values with mean, median, mode, or other estimated values.
   - **Deletion**: Removing records or fields with missing values.
   - **Substitution**: Using placeholder values to indicate missing data.
2. **Removing Duplicates**:
   - Identifying and removing duplicate records to ensure each entry is unique.
   - Deduplication techniques often involve matching fields like names, addresses, and IDs.

3. **Correcting Inconsistencies**:
    - Standardizing data formats (e.g., date formats, units of measurement).
    - Ensuring consistency in naming conventions, abbreviations, and categorization.
4. **Addressing Outliers**:
    - Identifying outliers that deviate significantly from other data points.
    - Deciding whether to correct, transform, or remove outliers based on context and impact.
5. **Validation and Verification**:
    - Ensuring data conforms to predefined rules and constraints (e.g., data types, ranges).
    - Cross-checking data against reliable sources to verify accuracy.
6. **Handling Errors**:
    - Correcting typographical errors, misspellings, and syntax errors.
    - Ensuring logical coherence (e.g., age values should be non-negative).
7. **Normalization and Standardization**:
    - Transforming data to a common scale without distorting differences in ranges (normalization).
    - Converting data into a standard format (standardization).

Techniques and Tools for Data Cleaning

1. **Manual Cleaning**:
    - Reviewing and correcting data manually, often necessary for small datasets or complex issues.
    - Time-consuming and prone to human error.
2. **Automated Tools and Scripts**:
    - Using software tools and custom scripts to automate repetitive cleaning tasks.
    - Examples include OpenRefine, Trifacta, and Talend.
3. **Data Integration Tools**:
    - Tools that combine data from multiple sources and ensure consistency.
    - Examples include Apache Nifi, Informatica, and Microsoft SQL Server Integration Services (SSIS).
4. **Machine Learning Techniques**:
    - Using machine learning algorithms to predict and correct missing or inconsistent data.
    - Examples include clustering for anomaly detection and regression for imputation.
5. **Data Quality Assessment Frameworks**:
    - Implementing frameworks to continuously monitor and assess data quality.
    - Examples include data profiling, data quality rules, and metrics.

Benefits of Data Cleaning

1. **Improved Data Quality**: Ensures data is accurate, complete, and reliable.
2. **Enhanced Decision-Making**: High-quality data leads to better insights and informed decisions.
3. **Increased Efficiency**: Reduces the need for rework and corrections in later stages.
4. **Compliance and Trust**: Helps in maintaining compliance with regulations and builds trust in data-driven processes.
5. **Optimized Performance**: Clean data enhances the performance of data mining algorithms and analytical models.

II. DATA MINING TOOLS

There are several types of data mining tools available, each serving a specific purpose. Some common types include:

1. Data exploration tools: These tools are used to explore and visualize data to gain a better understanding of its structure and characteristics. Examples include Tableau, Power BI, and Google Data Studio.

2. Predictive analytics tools: These tools focus on predicting future trends and outcomes based on historical data. They use statistical modeling and machine learning algorithms to make predictions. Examples include IBM SPSS Modeler, SAS Enterprise Miner, and RapidMiner.

3. Text mining tools [4]: These tools are designed for analyzing and extracting insights from unstructured text data, such as social media posts, customer reviews, and news articles. Examples include RapidMiner, Knime, and GATE.

4. Cluster analysis tools: These tools are used to classify and group similar data points based on their similarity or dissimilarity. They are helpful in identifying patterns and segments within a dataset. Examples include WEKA, Orange, and SPSS.

5. Association rule mining tools [5]: These tools are used to discover patterns and relationships between variables, such as frequent itemsets and association rules. They are commonly used in market basket analysis and recommendation systems. Examples include WEKA, RapidMiner, and Orange.

6. Decision tree and rule induction tools: These tools generate classification and regression models in the form of decision trees or sets of rules. They are useful for making predictions or understanding the factors that influence a particular outcome. Examples include WEKA, RapidMiner, and Orange.

These are just a few examples, and there are many other data mining tools available in the market, each with its own set of features and capabilities. The choice of tool depends on the specific requirements and objectives of a data mining project.

III. DATA MINING ALGORITHMS

There are several data mining algorithms available, each designed to solve different types of problems and extract valuable insights from data. Some common data mining algorithms include:

Association rule mining algorithms [9]: These algorithms are used to discover relationships and associations between variables in a dataset. The most famous algorithm in this category is the Apriori algorithm, which identifies frequent itemsets and association rules.

Association rule mining algorithms are used to identify interesting patterns or relationships between items in a dataset. They are commonly used in market basket analysis, where the goal is to uncover associations between products that are frequently purchased together. However, association rule mining algorithms can also be applied to various other domains, including student performance prediction.

In the context of student performance prediction, association rule mining algorithms can be used to discover relationships between different factors and academic performance. For example, by mining a dataset containing information about students' demographic data, socio-economic background, study habits, and academic outcomes, one can identify patterns or associations that can help predict a student's performance.

Here's a general process for applying association rule mining algorithms for student performance prediction:

1. Data preprocessing: Start by collecting the necessary data related to student performance, such as demographic information, educational background, attendance records, test scores, and other relevant factors. Clean the data and remove any irrelevant or incomplete records.

2. Data transformation: Convert the data into a suitable format for association rule mining. This typically involves representing the data as transactions, where each transaction represents a student's attributes or characteristics.

3. Choosing suitable measures: Define suitable measures for determining interestingness of association rules, such as support (the proportion of transactions containing the itemset), confidence (the probability that an item B is purchased given that item A is purchased), and lift (the ratio of the observed support to the expected support).

4. Mining association rules: Apply an association rule mining algorithm, such as the Apriori algorithm, FP-growth algorithm, or Eclat algorithm, to discover association rules between different factors and student performance. These algorithms analyze the data to identify frequent itemsets and generate rules based on predefined support and confidence thresholds.

5. Rule evaluation and interpretation: Analyze the generated association rules to identify patterns or relationships that are meaningful and relevant to student performance. Evaluate the rules based on their interestingness measures and interpret the results.

6. Prediction and evaluation: Use the discovered association rules to predict student performance for new or unseen data instances. Evaluate the predictive accuracy of the association rule model using appropriate performance metrics, such as accuracy, precision, recall, or F1 score.

It's important to note that association rule mining algorithms may not capture causal relationships or provide explanations for why certain associations exist. Therefore, the discovered patterns should be further analyzed and validated using other techniques or domain knowledge.

Overall, association rule mining algorithms can be a valuable tool in predicting student performance by identifying interesting relationships or factors that contribute to academic outcomes.

Decision tree algorithms [1]: Decision tree algorithms are used for classification and regression tasks. They create a tree-like model that represents decisions and their possible consequences. Some popular decision tree algorithms include ID3, C4.5, and CART.

Decision tree algorithms are commonly used for student performance prediction because of their ability to handle categorical and numeric data, handle missing values, and handle interactions between variables. Decision trees are graphical models that represent decisions and their possible consequences as a tree-like structure. In the context of student performance prediction, decision tree algorithms can be trained on historical data to learn patterns or rules that can predict a student's performance.

Here are some popular decision tree algorithms used for student performance prediction:

1. C4.5: C4.5 is one of the most widely used decision tree algorithms. It uses information gain or gain ratio as a measure to evaluate the features and split the data at each node. C4.5 supports both categorical and numeric data and handles missing values.

2. ID3: ID3 (Iterative Dichotomiser 3) is an older decision tree algorithm that was developed prior to C4.5. It uses information gain as the measure to split the data at each node. However, ID3 only supports categorical features and does not handle missing values.

3. CART (Classification and Regression Trees): CART is a versatile decision tree algorithm that can be used for both classification and regression tasks. It uses either Gini impurity or entropy as a measure to evaluate feature splits, depending on whether the task is classification or regression. CART supports both categorical and numeric features and handles missing values.

4. Random Forest: Random Forest is an ensemble method that combines multiple decision trees to improve prediction accuracy and reduce overfitting. It creates a large number of decision trees on different subsets of the data and aggregates their predictions to make the final prediction. Random Forest is robust to noisy data and handles missing values.

5. Gradient Boosted Trees: Gradient Boosted Trees is another ensemble method that combines multiple decision trees. It builds decision trees iteratively, where each new tree is trained to correct the mistakes made by the previous trees. Gradient Boosted Trees can provide high predictive accuracy but may require more computational resources.

To apply decision tree algorithms for student performance prediction, the general process involves:

Data preprocessing

Choosing suitable features

Training the decision tree

Evaluation

Visualization and interpretation

Prediction

It's important to note that decision tree models may suffer from overfitting if not properly regularized or pruned. Regularization techniques such as pruning or using random forests can alleviate overfitting issues and improve the model's generalization performance.

Overall, decision tree algorithms are powerful tools for student performance prediction due to their interpretability, flexibility, and ability to handle both categorical and numeric data.

3. Clustering algorithms [1,2]: Clustering algorithms group similar data points into clusters based on their similarity or dissimilarity. Examples of clustering algorithms include K-means, Hierarchical Clustering, and DBSCAN.

Clustering algorithms can also be used for student performance prediction by grouping students based on similar characteristics or performance patterns. Here are some common clustering algorithms used in this context:

1. K-means [3]: K-means is a popular clustering algorithm that partitions data into K clusters. Each cluster is represented by its centroid, and data points are assigned to the cluster with the closest centroid. In the context of student performance prediction, features such as demographics, study habits, and academic outcomes can be used to cluster students into groups based on their similarities.

2. Hierarchical clustering: Hierarchical clustering builds a hierarchical structure of clusters by recursively merging or splitting clusters based on a defined similarity measure. It can be agglomerative (bottom-up) or divisive (top-down). Agglomerative hierarchical clustering is commonly used, where each data point starts as its own cluster and is successively merged based on the similarity measure. Hierarchical clustering can provide a dendrogram that visualizes the clustering structure.

3. DBSCAN: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that groups together data points that are close to each other, while also identifying noise as outliers. It does not require specifying the number of clusters in advance and is robust to noise and outliers. DBSCAN is suitable for student performance prediction when there are natural clusters in the data.

4. Gaussian Mixture Models (GMM): GMM assumes that the data points are generated from a mixture of Gaussian distributions. It identifies the parameters (means and covariances) of these distributions and assigns each data point to the most likely cluster. GMM clustering is beneficial when the underlying data distribution is not well separated and can handle overlapping clusters.

5. Spectral clustering: Spectral clustering projects the data into a lower-dimensional space and then applies a traditional clustering algorithm (often K-means) to the transformed data. It is useful when the data has complex structures and cannot be well separated linearly.

Clustering algorithms can be useful for identifying student subgroups or patterns that may not be immediately apparent and can help in personalizing educational interventions and support strategies based on students' specific needs.

4. Classification algorithms [6]: Classification algorithms are used to assign labels or categories to data instances based on their features. Some common classification algorithms include Naive Bayes, Logistic Regression, Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN).

Classification algorithms can be used for student performance prediction by training models to classify students into different performance categories based on their features. Here are some common classification algorithms used in this context:

1. Logistic Regression: Logistic regression is a popular algorithm for binary classification tasks. It models the probability of an event occurring (in this case, student performance category) based on a set of predictor variables (features). Logistic regression can handle both continuous and categorical features.

2. Decision Trees: Decision trees are flowchart-like structures where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents a class label (performance category). Decision trees are easy to interpret and can handle both numeric and categorical features.

3. Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees and combines their predictions. It reduces overfitting and improves prediction accuracy by averaging the predictions of individual trees. Random forest is suitable when dealing with high-dimensional feature spaces.

4. Support Vector Machines (SVM) [10]: SVM is a supervised learning algorithm that separates instances into different classes by finding an optimally separating hyperplane. SVM can handle both linear and nonlinear classification problems by using different kernel functions. SVM is effective when the data is not well separable or when there are many irrelevant features.

5. Naive Bayes: Naive Bayes is a probabilistic algorithm that applies Bayes' theorem with an assumption of independence among the features. It is fast, simple, and efficient, making it suitable for large datasets. However, the naive assumption of feature independence may limit its accuracy.

6. K-Nearest Neighbors (KNN) [11]: KNN is a non-parametric algorithm that assigns the class label of a data point based on the labels of its k nearest neighbors in the feature space. KNN does not require training; instead, it

uses the entire dataset as its model. It is simple and intuitive but can be computationally expensive for large datasets.

7. Gradient Boosting: Gradient boosting is an ensemble learning method that combines multiple weak learners (e.g., decision trees) to create a strong learner. It builds the model in a stage-wise manner, sequentially fitting new models to the residuals of the previous models. Gradient boosting is known for its high accuracy but can be computationally expensive.

Classification algorithms provide a framework for automating student performance prediction based on historical data and can help identify at-risk students, tailor interventions, and optimize educational outcomes.

5. Regression algorithms [3]: Regression algorithms are used to predict continuous numerical values based on input variables. Linear Regression, Polynomial Regression, and Support Vector Regression (SVR) are some popular regression algorithms.

Regression algorithms can be used for student performance prediction by training models to predict continuous values, such as the final grade or GPA, based on the input features. Here are some common regression algorithms used in this context:

1. Linear Regression: Linear regression is a basic algorithm that establishes a linear relationship between the input features and the output variable. It finds the best-fit line that minimizes the sum of squared errors between the predicted and actual values. Linear regression is easy to interpret and implement.

2. Polynomial Regression: Polynomial regression extends linear regression by introducing nonlinear terms of the input features. It fits a polynomial curve to the data, allowing for more flexible modeling of the relationship between the features and the output variable. Polynomial regression can capture more complex patterns in the data.

3. Ridge Regression: Ridge regression is a regularized variant of linear regression that adds a penalty term to the coefficient estimates. The penalty term helps control the model's complexity and prevents overfitting. Ridge regression is suitable when there are multicollinearity issues among the input features.

4. Lasso Regression: Lasso regression is another regularized linear regression algorithm that shrinks some of the coefficient estimates to zero, effectively performing feature selection. This can be useful when dealing with high-dimensional feature spaces and when there are many irrelevant features.

5. Support Vector Regression (SVR): SVR is a regression counterpart of SVM. It aims to find a hyperplane that maximizes the margin while allowing some deviation from the exact prediction. SVR can handle both linear and nonlinear regression tasks by using different kernel functions.

6. Decision Tree Regression: Decision tree regression builds a regression model by partitioning the feature space into regions and assigning a constant value to each region. It splits the data based on the features and their thresholds, aiming to minimize the variance within each terminal node. Decision tree regression can handle both numeric and categorical features.

7. Random Forest Regression: Random forest regression is an ensemble learning method that combines multiple decision trees to reduce overfitting and improve prediction accuracy. It aggregates the predictions of individual trees to obtain the final prediction. Random forest regression is suitable when dealing with high-dimensional feature spaces and when there may be interactions among the features.

8. Gradient Boosting Regression: Gradient boosting regression is an ensemble learning method that builds a strong regression model by sequentially adding weak learners (usually decision trees) to the ensemble. Each new model is trained to reduce the errors of the previous models. Gradient boosting regression is known for its high accuracy but can be computationally expensive.

Regression algorithms provide a framework for predicting student performance based on historical data and can help identify factors that contribute to academic success or failure, as well as identify interventions or support that can improve student outcomes.

6. Neural network algorithms: The architecture and operation of the human brain serve as inspiration for neural networks, a class of machine learning algorithms. Among their many applications are pattern recognition, image/speech recognition, and NLP. Machine learning algorithms such as RNNs, Multilayer Perceptrons, and Convolutional Neural Networks are a few examples.

Predicting how well a pupil will do in school is another possible application of neural network algorithms, more especially deep learning models. Some popular neural network methods for this kind of application are as follows:

1. Feedforward Neural Networks (FNN): Feedforward neural networks consist of an input layer, one or more hidden layers, and an output layer. Each layer is composed of interconnected nodes, or neurons, that process and propagate information through the network. FNNs can learn complex, nonlinear relationships between input features and output variables. They can be used for regression tasks, predicting continuous values like final grades or GPA.

2. Recurrent Neural Networks (RNN): Recurrent neural networks are specialized for sequential data, such as time series or data with temporal dependencies. RNNs introduce recurrent connections that enable them to capture information from previous time steps. This makes them suitable for prediction tasks where the order of data is important, such as predicting student performance over time.

3. Long Short-Term Memory (LSTM): LSTM is a variant of RNN that addresses the issue of vanishing gradients and allows RNNs to capture long-range dependencies. LSTM networks have memory cells that can store and access information over longer intervals, giving them the ability to model complex temporal patterns. This makes them well-suited for predicting student performance over extended periods.

4. Gated Recurrent Unit (GRU): GRU is another variant of RNN that is similar to LSTM but has a simpler architecture. GRU networks are effective in handling sequential data and can capture short-term dependencies effectively. They are computationally less expensive than LSTM and can be used for student performance prediction tasks.

5. Convolutional Neural Networks (CNN): Convolutional neural networks are primarily used for image processing tasks, but they can also be used for feature extraction and prediction in student performance tasks that involve visual data. CNNs employ convolutional and pooling layers to capture spatial hierarchies in images and can be applied to tasks such as predicting handwriting or visual performance.

6. Hybrid Models: Hybrid models combine multiple neural network architectures to exploit their respective strengths. For example, a hybrid model may use convolutional layers for feature extraction from visual data and feed the extracted features into an LSTM or FNN for final prediction. This approach can capture both spatial and temporal patterns simultaneously.

Neural network algorithms, with their ability to learn complex patterns and handle sequential data, can be effective for student performance prediction tasks. However, they also require larger amounts of data and longer training times compared to traditional regression algorithms. Additionally, interpretation of neural network models can be more challenging due to their black-box nature.

7. Anomaly detection algorithms: Anomaly detection techniques are specifically developed to find patterns in data that significantly differ from the normal or expected behaviour. These applications include fraud detection, network intrusion detection, and outlier identification. Several widely used methods for anomaly detection include One-class SVM, Isolation Forest, and Local Outlier Factor.

These are just a few examples of the wide range of data mining algorithms available. The choice of algorithm depends on the specific problem at hand, the type of data, and the desired outcome.

Anomaly detection algorithms can also be used for student performance prediction to identify unusual patterns or behaviors that deviate from normal or expected performance. Here are some common anomaly detection algorithms used in this context:

1. Statistical Approaches: Anomaly detection often use statistical methodologies. These approaches usually include computing summary statistics, such as the mean, standard deviation, or percentiles, for various aspects of student performance. Anomalies are identified as observations that deviate from a certain range or have a distinct distribution in comparison to the remaining data.

2. Clustering-based Approaches: Clustering algorithms, such as k-means or DBSCAN, can be used to group similar student performance instances together. Anomalies can then be detected as data points that do not belong to any cluster or belong to small, isolated clusters.

3. Classification-based Approaches: Classification algorithms, such as Support Vector Machines (SVM) or Random Forests, can be trained on a labeled dataset where anomalies are explicitly identified. These models can then be used to predict whether new instances of student performance are normal or anomalous.

4. Autoencoders: Broadly used neural network topologies for unsupervised anomaly detection include autoencoders. Two networks make up them: one encoder compresses the input data and one decoder reconstructs the original data. Anomalies are found when the reconstruction error rises over a certain level.

5. One-Class Support Vector Machines (OC-SVM): OC-SVM is a variant of Support Vector Machines that is designed to identify anomalies in a one-class setting. It learns a boundary that encompasses the normal instances and identifies observations outside this boundary as anomalies.

6. Isolation Forest: Isolation Forest is an ensemble learning technique that separates anomalies by randomly partitioning the data into isolation trees. Anomalies are detected as instances with shorter average path lengths within the trees.

7. Density-based Approaches: Density-based algorithms, such as Local Outlier Factor (LOF) or Gaussian Mixture Models (GMM), identify anomalies based on the density or probability distribution of the data. Outliers are detected as instances with significantly lower densities or lower probabilities compared to the majority of the data.

Anomaly detection algorithms can be useful for detecting unusual or unexpected patterns in student performance data. They are particularly effective in scenarios where anomalies may be rare or where the definition of anomalies may be subjective or evolving. However, the success of anomaly detection algorithms depends on having a representative training dataset and selecting appropriate features for anomaly identification.

CONCLUSION

In this study, we investigated several data mining tools and algorithms for predicting student performance. There is significant promise for predicting student performance and facilitating more tailored and successful educational experiences by using data mining techniques. Nonetheless, it is critical to find a balance between technical improvements and ethical concerns to ensure that these tools are used fairly and equitably to promote and enhance student learning. Data mining methods for predicting student performance emphasise numerous significant factors based on the examination and implementation of different methodologies. These ideas are critical for educators, administrators, and researchers seeking to improve educational results and customise interventions more effectively. Every algorithm has strengths and disadvantages. For example, decision trees give interpretability, while neural networks may provide more accuracy but are often more complicated and opaquer. The creation of more robust, scalable, and interpretable models is a continuous task. Future research should concentrate on enhancing model generalizability and ensuring that forecasts are useful to students.

REFERENCES

1. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
2. Verma, Manish, et al. "A comparative study of various clustering algorithms in data mining." International Journal of Engineering Research and Applications (IJERA) 2.3 (2012): 1379-1384.
3. Varun Kumar, “An Empirical Study of Applications of Data Mining Techniques in Higher Education”, International Journal of Advanced Computer Science and Applications, Vol. 2(3), March 2011.
4. Karur Parminder and Qamar Parvez Rana. “Comparison of various data mining tools”, International Journal of Engineering Research & Technology (IJERT)ISSN: 2278-0181IJERTV3IS100246 Vol. 3 Issue 10, October- 2014.
5. Kumbhare, Trupti A., and Santosh V. Chobe. "An overview of association rule mining algorithms." International Journal of Computer Science and Information Technologies 5.1 (2014): 927-930.
6. Prasadi Peddi & Dr. Akash Saxena (2014), "EXPLORING THE IMPACT OF DATA MINING AND MACHINE LEARNING ON STUDENT PERFORMANCE", International Journal of Emerging Technologies and Innovative Research, ISSN:2349-5162, Vol.1, Issue 6, page no.314-318.
7. T. Silwattananusarn, “Data Mining & Its Applications for Knowledge Management: A Literature Review from 2007 to 2012,” Int. J. Data Min. Knowl Manag. Process, 2012, doi: 10.5121/ijdkp.2012.2502.
8. P. Guleria & M. Sood, “Data Mining in Education: A Review on the Knowledge Discovery Perspective,” Int. J. Data Min. Knowl. Manag. Process, 2014, doi: 10.5121/ijdkp.2014.4504.
9. G.Keseavaraj, S.Sukumaran,"Study on classification techniques on data mining," 4th ICCCNT ,IEEE, 2013.
10. Meneses, C., Categorization and Evaluation of Data Mining Techniques. Technical Report, Institute for Visualization and Perception Research, Department of Computer Science, University of Massachusetts Lowell, 1998.
11. Ronan Collobert, Samy Bengio, and C. Williamson. Svmtorch: Support vector machines for large-scale regression problems. Journal of Machine Learning Research, 1:143–160, 2001.

12. Peddi, P. (2015). The Adoption of a Big Data and Extensive Multi-Labled Gradient Boosting System for Student Activity Analysis. International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM).
13. Prasadu Peddi (2015) "A review of the academic achievement of students utilising large-scale data analysis", ISSN: 2057-5688, Vol 7, Issue 1, pp: 28-35.