**1.1 Introduction**

Evaluating student performance is necessary for educators to retrieve early evaluation and take prompt action or early precautions if necessary, to improve the student’s evaluation. This prediction can be managed by locating the source of the problem. Should it be from extra activities that the student is participating in, family problems, or health problems. All these factors can have a major effect on student performance. By means of having a dataset for student’s performance can help us study such cases. The used dataset in this research paper is collected from internet, it has data of 500 students. We trained 80% of the dataset. In this research paper, we used a regression algorithm which is K-Nearest Neighbor (KNN) to predict the final grade of the student, which predicted the student’s performance. In a country's life inculcation plays a vital role to ascertain the survival of the state and the nation. In today's scenario scholastic technologies aide the process of learning and edifying (TL) as they are being utilized in scholastic domains including the traditional form of classrooms where it’s all about face to face and even the cognition platforms available online. Edifying actors have been benefited as they are provided with the germane information in which they have to act upon and thereby end up in promoting the quality predicated innovations in this domain. These days universities are run in a very puissant and dynamically viable manner. A substantial quantity of data is accumulated in the form of marks, records, documents, files, performance et cetetra all cognate to student performance.



*15*

* 1. **Overview of the method**

K-nearest neighbor technique is a machine learning algorithm that is considered as simple to implement (Aha etal. 1991).

K- Nearest Neighbors is a Supervised machine learning algorithm as target variable is known, Its Non parametric as it does not make an assumption about the underlying data distribution pattern. Lazy algorithm as KNN does not have a training step. All data points will be used only at the time of prediction. With no training step, prediction step is costly. An eager learner algorithm eagerly learns during the training step. It’s Used for both Classification and Regression. It Uses feature similarity to predict the cluster that the new point will fall into.

K is a number used to identify similar neighbors for the new data point. KNN takes K nearest neighbors to decide where the new data point with belong to. This decision is based on feature similarity. Instance-based algorithms are those algorithms that model the problem using data instances (or rows) in order to make predictive decisions. A kNN algorithm is an extreme form of instance-based methods because all training observations are retained as a part of the model. It is a competitive learning algorithm because it internally uses competition between model elements (data instances) to make a predictive decision. The objective similarity measure between data instances causes each data instance to compete to “win” or be most alike to a given unseen data instance and contribute to a prediction. Lazy learning refers to the fact that the algorithm does not build a model until the time a prediction is required. It is lazy because it only does work at the last second. This has the benefit of only including data relevant to the unseen data, called a localized model. A disadvantage is that it can be computationally expensive to repeat the same or similar searches over larger training datasets. Finally, kNN is powerful because it does not assume anything about the data, other than that the distance measure can be calculated consistently between any two instances. Thus, it is called non-parametric or non-linear as it does not assume a functional form.



*16*

**1.3 Motivation**

So why would someone use this classifier over another? Is this the best classifier? The answer to these questions are that it depends. There is no classifier that is best, it all depends on the data that a classifier is given. KNN might be the best for one dataset but not another. It’s good to know about other classifiers like Support Vector Machines, and then decide which one best classifies the a given dataset.. Ease of understanding and implementing are 2 of the key reasons to use kNN. Depending on the distance metric, kNN can be quite accurate.

But that’s just part of the story…

Here are five things to watch out for:

* kNN can get very computationally expensive when trying to determine the nearest neighbors on a large dataset.
* Noisy data can throw off kNN classifications.
* Features with a larger range of values can dominate the distance metric relative to features that have a smaller range, so feature scaling is important.
* Since data processing is deferred, kNN generally requires greater storage requirements than eager classifiers.
* Selecting a good distance metric is crucial to kNN’s accuracy.

**2.1 Is this classification technique appropriate?**

There is no such thing as the best classifier, it always depends on the context, what kind of data/problem is at hand. You can find a lot of machine learning systems for data classification. But it is worthy of consideration that we can categorize all of them into two main groups of local and global classification systems. Based on each dataset, features vector and size of dataset a local or global classifier can make a better model for final classification. For datasets with a huge number of cases if you work on ANN methods, it is possible for you to have feature selection of a long feature vector and classification simultaneously. Hence, you can make your final feature vector efficient enough and results are more reliable but not fast like KNN.

In a nutshell, for day to day research KNN is more conventional, for small datasets, it would be logical if you train and test one local and one global classifier on your data to find out which one makes better results. KNN is kind of state of the arts in this area.

**2.2 Every ML algorithm: three keys**

There are tens of thousands of machine learning algorithms and hundreds of new algorithms are developed every year.

Every machine learning algorithm has three components :

**2.2.1 Representation**

A classifier must be represented in some formal language that the computer can handle. Conversely, choosing a representation for a learner is tantamount to choosing the set of classifiers that it can possibly learn. This set is called the hypothesis space of the learner. If a classifier is not in the hypothesis space, it cannot be learned. A related question, which we will address in a later section, is how to represent the input, i.e., what features to use.

**2.2.2 Evaluation**

An objective function or a scoring function, to distinguish good models from a bad model. It can be possible that the evaluation function used by a particular algorithm may be different from the external one that we want the classifier to optimize. For a classification problem, we need this function to know if a a given classifier is good or bad. A typical function can be based on the number of errors made by the classifier on a test set, using precision and recall. For a regression problem, it could be the squared error, or likelihood.

**2.2.3 Optimization**



Estimate model parameters using optimization methods.

What is the optimization function that will find the minima or maxima of the objective function? What it does is to simply compute the model parameters and select those values that result in the lowest error(minima) or highest reward(maxima).

Optimization can be based on combinatorial like greedy search, on unconstrained continuous optimization like gradient descents, and on constrained continuous optimization like linear programming. It is actually the way candidate programs are generated known as the search process. For example, combinatorial optimization, convex optimization, constrained optimization.

**4.1 Introduction**

The KNN algorithm can compete with the most accurate models because it makes highly accurate predictions. Therefore, you can use the KNN algorithm for applications that require high accuracy but that do not require a human-readable model. The quality of the predictions depends on the distance measure. Therefore, the KNN algorithm is suitable for applications for which sufficient domain knowledge is available. This knowledge supports the selection of an appropriate measure. The KNN algorithm is a type of lazy learning, where the computation for the generation of the predictions is deferred until classification. Although this method increases the costs of computation compared to other algorithms, KNN is still the better choice for applications where predictions are not requested frequently but where accuracy is important.