**“STUDENT PERFORMANCE ANALYZING USING K NEAREST NEIGHBOR ALGORITHM”**



Department of Computer Science and Engineering(CSE)

**Bangladesh University of Business and Technology (BUBT)**

**DECEMBER , 2019**

**BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)**



**STUDENT PERFORMANCE ANALYZING USING**

**K NEAREST NEIGHBOR ALGORITHM**

*A Thesis*

*Submitted to the Department of Computer Science and*

*Engineering*

*in partial fulfillment of the requirements*

*for the degree of*

**Bachelor of Science in Computer Science and Engineering**

By

|  |  |  |
| --- | --- | --- |
| **Md Ashraful Islam** | **SK Tanvir Rahman** | **Shafiur Rahman Chowdhury** |
| 32th intake | 32th intake | 32th intake |
| ID: 15162103018 | ID: 15162103041 | ID: 15162103043 |
| Section: 01 | Section: 01 | Section: 01 |

**Supervised by**

**Md. Saifur Rahman**

Assistant Professor, Department of CSE, BUBT

Rupnagar, Mirpur-2, Dhaka-1216, Bangladesh

**ABSTRACT**

K nearest neighbor method is the use of a model to predict future values based on previously observed values. K nearest neighbor approach is widely used for non-stationary data, like economics, weather, stock price, student’s academic result, retail sales, etc. The achievement of this algorithm would bring a level of accuracy when a student can determine if someone might be able to get admitted to a particular university, based on his previous study performance. On another hand, the universities determine the admission rates for each discipline and program depending on the general high school exam results each year. A smart prediction model of exam results in next year will help the universities to determine the level from the beginning of the acceptance process. Not only these approaches but there are a lot of areas we can integrate this research and improve the quality of an organization.

**DECLARATION**

We hereby declare that the thesis entitled “**STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM**” submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering in the Faculty of Computer Science and Engineering of Bangladesh University of Business and Technology, is our own work and that it contains no material which has been accepted for the award to the candidate(s) of any other degree or diploma, except where due reference is made in the text of the thesis. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the thesis.

*Signature* *Signature* *Signature*

|  |  |  |
| --- | --- | --- |
| **-----------------------------** | **----------------------------** | **--------------------------------** |
| **Md Ashraful Islam** | **SK Tanvir Rahman** | **Shafiur Rahman Chowdhury** |
| ID: 15162103018 | ID: 15162103041 | ID: 15162103043 |



**CERTIFICATE**

This is to certify that Md Ashraful Islam, SK. Tanvir Rahman and Shafiur Rahman Chowdhury of B.Sc. in CSE has completed their thesis work titled **“STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM”** in partial fulfillment for the requirement of B.Sc. in Computer Science andEngineering from Bangladesh University of Business and Technology in the year 2020.

|  |  |  |
| --- | --- | --- |
| **-----------------------------** | **----------------------------** | **--------------------------------** |
| **Md Ashraful Islam** | **SK Tanvir Rahman** | **Shafiur Rahman Chowdhury** |
| ID: 15162103018 | ID: 15162103041 | ID: 15162103043 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Thesis Supervisor

**(Md. Saifur Rahman)**

Assistant Professor

Department of Computer Science and Engineering (CSE)

Bangladesh University of Business and Technology (BUBT)



**ACKNOWLEDGEMENTS**

We would like to pay our gratitude to the **Almighty Allah** who created us with all the abilities to understand analysis and develop the process with patience.

We are grateful to our beloved parents who nurtured us from the moment of our appearance in the world and utterly loving us.

We are thankful to our thesis supervisor Md. Saifur Rahman**,** Assistant Professor, Computer Science & Engineering Department, Bangladesh University of Business & Technology for his professional guidance and motivation during the work of this thesis which is a major part of it. Without his valuable support and guidance, this thesis could not reach this level of development from our point of view.

We would like to express our sincere and warm gratitude to all those who have encouraged us directly, provided mental encouragement and criticized our work in several phases during the development of this thesis and for preparing this thesis indirectly.

We would like to thank all the Faculty members, Department of CSE, Bangladesh University of Business & Technology for their valuable time spend in requirements analysis and evaluation of the thesis work.

**Md Ashraful Islam (**15162103018**)**

**SK. Tanvir Rahman (**15162103041**)**

**Shafiur Rahman Chowdhury (**15162103043)

**DEDICATION**

Dedicated to our parents, teachers, friends, relatives and all who loved us

for all their love and inspirations.

**APPROVAL**

This thesis is **“STUDENT PERFORMANCE ANAYZING USING K NEAREST NEIGHBOR ALGORITHM”** This report is submitted by **Md Ashraful Islam** (15162103018); **SK. Tanvir Rahman** (15162103041); **Shafiur Rahman Chowdhury** (15162103043), Department of Computer Science and Engineering, Bangladesh University of Business and Technology under the supervision of

**Md. Saifur Rahman**, Assistant Professor, Department of Computer Science and Engineering has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc. Engg.) in Computer Science and Engineering.

**Supervisor:**

**Md. Saifur Rahman**

Assistant Professor

Department of CSE, BUBT



**Abbreviations**

KNN

CNN

LSTM

LIFO

RMS

PDF

K-Nerest Neighbor

Convolutional Neural Network

Long Short-Term Memory

Last In First Out

Root Mean Square

Probability Density Function



*ix*

**Table of Contents**

|  |  |
| --- | --- |
| **ABSTRACT** | **ii** |
| **ACKNOWLEDGEMENT** | **iii** |
| **DECLARATION** | **iv** |
| **DEDICATION** | **vi** |
| **CERTIFICATE** | **vii** |
| **APPROVAL** | **viii** |
| **ABBRIVIATIONS** | **ix** |

|  |  |  |
| --- | --- | --- |
| **Chapter 01: Introduction** | | **1** |
| 1.1 | Introduction | 1 |
| 1.2 | Existing systems | 3 |
| 1.3 | Motivation | 5 |
| 1.4 | Aims and Objectives of the thesis | 7 |
| 1.5 | Contribution | 9 |
| 1.6 | Organization of thesis | 10 |
| 1.7 | Conclusions | 11 |

|  |  |  |
| --- | --- | --- |
| **Chapter 2: Existing Systems** | | **13** |
| 2.1 | Is classification technique appropriate? | 13 |
| 2.2 | Every ML algorithm: three keys | 12 |
|  | 2.2.1 Representation | 14 |
|  | 2.2.2 Evaluation | 18 |
|  | 2.2.3 Optimization | 19 |
|  |  | 22 |
|  |  | 23 |

|  |  |  |  |
| --- | --- | --- | --- |
| .3 | Learning: three types | | 25 |
|  | 2.3.1 | Supervised learning | 25 |
|  | 2.3.2 | Unsupervised learning | 29 |
|  | 2.3.3 | Reinforcement learning | 32 |
| 2.4 The learning problem: an overview | | | 36 |
| 2.5 | Nearest Neighbor | |  |
| 2.6 | Importance of the factor K | |  |
| 2.7 | Conclusions | | 32 |

|  |  |  |
| --- | --- | --- |
| **Chapter 3: Proposed Model** | | **38** |
| 3.1 | Introduction | 38 |
| 3.2 | Procedure | 39 |
| 3.3 | Algorithm | 44 |
| 3.4 | Flowchart | 64 |
| 3.5 Advantages | | 65 |
| 3.6 | Conclusions | 65 |

|  |  |  |
| --- | --- | --- |
| **Chapter 4: Experimental Results and analysis** | |  |
|  | | **56** |
| 4.1 | Introduction | 56 |
| 4.2 | Result Analysis | 57 |
| 4.3 | Applications | 73 |
| 4.4 | Conclusions | 74 |



*xi*

|  |  |
| --- | --- |
| **List of Figures** | **xiv** |
| **List of Tables** | **xvi** |
| **References** | **78** |

|  |  |  |
| --- | --- | --- |
| **Chapter 5: Conclusions** | | **75** |
| 5.1 | Conclusions | 75 |
| 5.2 | Limitation | 76 |
| 5.3 | Future Works | 77 |



*xii*

**List of Figures**

|  |  |
| --- | --- |
| Figure 2.1: Traditional programming vs machine |  |
| learning | 15 |
| Figure 2.4: The Learning Problem | 26 |
| Figure 3.6: Flowchart of ID3 Algorithm | 30 |
| Figure 4.2: First Decision Tree | 35 |
| Figure 4.3: Final Decision Tree | 36 |



*xiii*

**List of Tables**

|  |  |
| --- | --- |
| Table 2.5: An unknown Boolean function with 7 examples, |  |
| each comprising of four Boolean inputs | 27 |
| Table 3.1: Observations of having diabetes | 29 |



*xiv*

**Chapter- 1**

**Introduction**

**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 data-set for student’s performance can help us study such cases. The used data-set in this research paper is collected from internet, it has data of 500 students. We trained 80% of the data-set. 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 puissance and dynamically viable manner. A substantial quantity of data is accumulated in the form of marks, records, documents, files, performance etc, all cognate to student performance.

**1.2 Existing Systems**

We have studied and worked with “**Student Performance prediction using algorithms of Data Mining**” [17]. Algorithms like LSTM, RNN, ANN act like Knowledge Discovery in Database (KDD) where there are some kind of memorization, prepossessing technique and pattern finding process involved.

We have also studied and worked with “**Student Performance Prediction Using Educational Data Mining Techniques**” [4]. In this process, Classification, Regression, Clustering . Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

Besides, regression is basically a statistical approach to find the relationship between variables. In machine learning, this is used to predict the outcome of an event based on the relationship between variables obtained from the data-set. Linear regression is one type regression used in Machine Learning.

As a primary goal of knowledge Discovery, predicting student performance, the term **Knowledge Discovery in Databases** (**KDD**), or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

KDD is an automatic, exploratory analysis andmodeling of large data repositories. **KDD** is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets.

A **Support Vector Machine** (**SVM**) is a discriminant classifier formally defined by a separating hyper-plane. In other words, given labeled training data (supervised learning), the **algorithm** outputs an optimal hyper-plane which categorizes new examples. It can solve linear and non-linear problems and work well for many practical problems.

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series,

**1.3 Motivation**

**K-nearest neighbors** (**KNN**) algorithm is a type of supervised ML algorithm whichcan be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −

* **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.
* **Lazy learning algorithm** − KNN is a lazy learning algorithm because it doesnot have a specialized training phase and uses all the data for training while classification.

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 data-set 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 data-set.. Ease of understanding and implementing are 2 of the key reasons to use KNN. Depending on the distance metric, KNN can be quite accurate.

Here are five things to watch out for:

* KNN can get very computationally expensive when trying to determine the nearest neighbors on a large data-set.
* 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.

**1.4 Aims and Objectives of the thesis**

Our aim to ascertain a result that defines a precise result of those 500 students, whose data has been trained by KNN Classifier method. The data is genuinely a combination of their CGPA's and results in their SSC, HSC & 1-12th semesters in their university period & evaluate the performance of each student predicated on the data they provided or predicated on their educational information.

It is a data mining technique for dividing data into predefined groups and if groups have label or name that is called supervised learning methods [3]. These kinds of method used to specify all data that have located to one of existence class, so it called supervised. If mining can be executed in educational environment that is educational data mining (EDM).

Data Classification can be viewed as a two-step process which consists of learning phase and actual classification. In the learning (training) phase, a general model is constructed to walk through to study patterns found available in a sample data-set to understand the relationships that exists in the data. Information gained from inter-class and intra-class relationships can then be used in the second phase to label and classify entities in test data [23]. Accuracy of classification models can be evaluated on a certain test data as the amount of instances that are correctly classified by the model.

In order to predict the student’s performance at university based on high school grades, and to predict the courses that mostly effect the performance of students in the first two years of university. Several data mining techniques were used for Classification. [8] Such as KNN, Logistic regression, SVM, Naive Bayes, Apriori, Adaboost etc.

**1.5 Contribution**

We tried to implement and experiment with existing ideas in our thesis work. In our system we proposed a way to gain more accuracy than previous works which can be said as the most important proposal of our work. For this we proposed a technique that we use for implementation known as KNN Classifier method.

It should be noted that it is not yet formally proven the correctness or falsehood of our proposed model but as we came to gain certain good outputs and by our calculation we can verbalize that this proposal is adequate for the next level of precision. Instead, this thesis is inhibited to contributing, hopefully strong, evidence for or against its validity.

Here are our contributions of this thesis:

1.This thesis avails for generating new conception for higher precision of prognosticating the information and the proposed technique is much efficient than other technique that can be seen.

2.Our thesis will help to understand classification and recognition data in an easy way and we tried to implement the system in a simple manner so that anyone can use it for different purpose.

3.Supervised classification was used for getting different patterns on different data-sets for better results and accuracy and for analyzing the system performance in different situations.

4.Besides, gaining accuracy and correct information also testing it for the best possible outcomes.

**1.7 Conclusions**

In this research paper, we represented the KNN Classifier method. k-Nearest Neighborhood (K-NN) classification is a method adoptable for classifying entities based on closest training examples in a feature space [9]. K-NN is a lazy learning classifiers that adopts instance-based learning hence having prediction done in two stages. Firstly, it undergoes minimal operations of analyzing the attribute values of individual instances in training data-set [23]. This paper provides some basic fundamental ideas about the KNN method for mining a data set.

For future works, how to quickly and accurately adapt to more new samples in online classification systems should be researched, and choosing a more efficient assessment method that can reasonably assign the training set and the testing set is necessary. We have majorly focused on more the accuracy & gaining correct information. The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

**Chapter- 2**

**Machine Learning in a nutshell**

In this chapter we provide an overview to Machine Learning. First, we discuss how this ML technique is appropriate and different to the traditional form of every ML algorithm boils down into three components. We state the three major types of learning algorithms.

**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 combination search 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, combination optimization, convex optimization, constrained optimization.

**2.3 Learning: three types**

Next, we’ll briefly give an overview of the three major types of learning: supervised, unsupervised and reinforcement learning.

**2.3.1 Supervised learning**

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

It is called supervised learning because the process of algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers; the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

The basic principle is the training data includes desired outputs. Think of this as a

”learning with a teacher” - somebody has already labeled what the “right” answer is.

Eg: Somebody has already labeled which emails are spam and which aren’t;

Somebody labeled which x-rays show cancer and which don’t. Thus, we know what

to learn. Supervised or inductive learning is the most mature (widely studied and used

in practice) kind of learning, and forms the basis of decision trees learning.

Supervised learning problems can be further grouped into regression and classification problems.

* **Classification**: A classification problem is when the output variable is acategory, such as “red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value,such a as “dollars” or “weight”.

Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively. Some popular examples of supervised machine learning algorithms are:

* Linear regression for regression problems.
* Random forest for classification and regression problems.
* Support vector machines for classification problems.

**2.3.2 Unsupervised Learning**

Unsupervised machine learning cannot be directly applied to a regression because it is unknown what the output values could be, therefore making it impossible to train the algorithm how you normally would.

Unsupervised learning is where you only have input data (X) and no corresponding output variables.

The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.

In sharp contrast to inductive learning, the training data does not include desired

outputs within an unsupervised learning environment. This is a much harder but the most important kind of learning, in the long run. As a useful analogy, think of how babies learn how to walk on its own - this would be an instance of unsupervised learning. If a parent tells the baby, “that’s a chair” or “that’s a table” – that’s supervised learning.

Unsupervised learning problems can be further grouped into clustering and association problems.

**Clustering**: A clustering problem is where you want to discover the inherentgroupings in the data, such as grouping customers by purchasing behavior.

**Association**: An association rule learning problem is where you want to discoverrules that describe large portions of your data, such as people that buy X also tend to buy Y.

Some popular examples of unsupervised learning algorithms are:

* ***[K-Means Clustering](https://algorithmia.com/algorithms/pappacena/kmeans) –*** clustering your data points into a number (K) ofmutually exclusive clusters. A lot of the complexity surrounds how to pick the right number for K.
* ***[Hierarchical Clustering](https://algorithmia.com/algorithms/weka/WekaHierarchicalClusterer)* –**clustering your data points into parent and childclusters. You might split your customers between younger and older ages, and then split each of those groups into their own individual clusters as well.
* ***[Probabilistic Clustering](https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/)*** –clustering your data points into clusters on aprobabilistic scale.

**2.3.3 Reinforcement learning**

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, artificial intelligence gets either rewards or penalties for the actions it performs. Although the designer sets the reward policy–that is, the rules of the game–he gives the model no hints or suggestions for how to solve the game. It’s up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills. By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint the machine’s creativity. In contrast to human beings, artificial intelligence can gather experience from thousands of parallel game plays if a reinforcement learning algorithm is run on sufficiently powerful computer infrastructure. inspiring applications. Training the models that control autonomous cars is an excellent example of a potential application of reinforcement learning. In an ideal situation, the computer should get no instructions on driving the car. The programmer would avoid hard-wiring anything connected with the task and allow the machine to learn from its own errors. In a perfect situation, the only hard-wired element would be the reward function.

For example: in usual circumstances, we would require an autonomous vehicle to put safety first, minimize ride time, reduce pollution, offer passengers comfort, and obey the rules of law. With an autonomous race car, on the other hand, we would emphasize speed much more than the driver’s comfort. The programmer cannot predict everything that could happen on the road. Instead of building lengthy “if-then” instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent (another name for reinforcement learning algorithms performing the task) gets rewards for reaching specific goals.

Another example: deepsense.ai took part in the “Learning to run” project, which aimed to train a virtual runner from scratch. The runner is an advanced and precise musculoskeletal model designed by the Stanford Neuromuscular Biomechanics Laboratory. Learning the agent how to run is a first step in building a new generation

of prosthetic legs, ones that automatically recognize people’s walking patterns and tweak themselves to make moving easier and more effective. While it is possible and has been done in Stanford’s labs, hard-wiring all the commands and predicting all possible patterns of walking requires a lot of work from highly skilled programmers.

The main challenge in reinforcement learning lays in preparing the simulation environment, which is highly dependant on the task to be performed. When the model has to go superhuman in Chess, Go, or Atari games, preparing the simulation environment is relatively simple. When it comes to building a model capable of driving an autonomous car, building a realistic simulator is crucial before letting the car ride on the street. The model has to figure out how to brake or avoid a collision in a safe environment, where sacrificing even a thousand cars comes at a minimal cost. Transferring the model out of the training environment and into the real world. Scaling and tweaking the neural network controlling the agent is another challenge. There is no way to communicate with the network other than through the system of rewards and penalties. This in particular may lead to catastrophic forgetting, where acquiring new knowledge causes some of the old to be erased from the network Yet another challenge is reaching a local optimum – that is the agent performs the task as it is, but not in the optimal or required way. A “jumper” jumping like a kangaroo instead of doing the thing that was expected of it-walking-is a great example. Finally, there are agents that will optimize the prize without performing the task it was designed for. An interesting example can be found in the OpenAI video below, where the agent learned to gain rewards, but not to complete the race.

**2.4 The learning problem: an overview**

Let us revisit Inductive learning and describe it more formally. Given examples of a function (X, F(X)) pair where X is the input of vector values that are either continuous or discrete. Eg: Symptoms of a patient (temp, blood pressure, glucose levels etc. And, F(X) is the value of the function for that element. Eg: Diagnosis of the patient -”yes, the patient has diabetes” or”no, the patient does not have diabetes”. The crux is how can we predict F(X) or”generalize” for new examples - data that we have not seen before?

There three types of supervised learning that needs to be mentioned:

**1. Discrete F(X):** classification. Eg: Computer vision system that wants to label the object that is seen (chair or table, say). As we are predicting the class of the object, hence this is known as a classification problem.

**2. Continuous F(X):** regression. Eg: Predicting the gas mileage of a car given its characteristics?

**3. F(X) = Probability(X):** probability estimation (if that is what we are predicting). Eg: Google might be interested in learning the probability of a user clicking on a particular ad? A harder problem would be predicting several things at the same time.

This brings us to the conclusion this chapter by stating the quintessence of the learning problem via the simple example below:

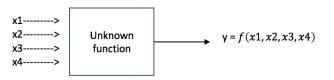


Figure 2.4: The Learning Problem

Suppose we are given a black box that represents a Boolean function. The input to this function are four Boolean inputs, x1, x2, x3, x4, each with a value of either 0 or 1. The output is y a y.

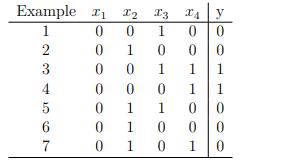


Table 2.5: An unknown Boolean function with 7 examples, each comprising of four Boolean inputs.

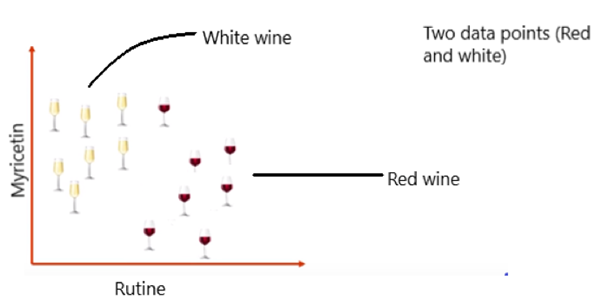
Each set of 4-values is known as an example. For these seven examples, we know what the outputs are - so what is y? If we are given an additional example set, what

would the corresponding value for y be? This, in essence, the inductive learning problem.

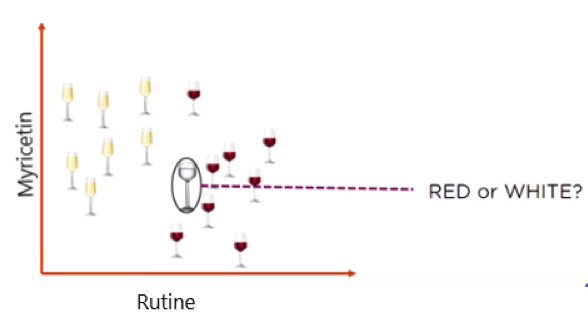
**2.5 Nearest Neighbor**

K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbors are classified.

Let’s take below wine glass example. Two chemical components called Rutime and Myricetin. Consider a measurement of Rutine vs Myricetin level with two data points, Red and White wines. They have tested and where then fall on that graph based on how much Rutine and how much Myricetin chemical content present in the wines.



‘k’ in KNN is a parameter that refers to the number of nearest neighbors to include in the majority of the voting process. Suppose, if we add a new glass of wine in the data-set. We would like to know whether the new wine is red or white?



So, we need to find out what the neighbors are in this case. Let’s say k = 5 and the new data point is classified by the majority of votes from its five neighbors and the new point would be classified as red since four out of five neighbors are red.

**2.6 Importance of the factor K**

Few ideas on picking a value for ‘K’

1. There is no structured method to find the best value for “K”. We need to find out with various values by trial and error and assuming that training data is unknown.
2. Choosing smaller values for K can be noisy and will have a higher influence on the result.
3. Larger values of K will have smoother decision boundaries which mean lower variance but increased bias. Also, computationally expensive.
4. Another way to choose K is though cross-validation. One way to select the cross-validation data-set from the training data-set. Take the small portion from the training data-set and call it a validation data-set, and then use the same to evaluate different possible values of K. This way we are going to predict the label for every instance in the validation set using with K equals to 1, K equals to 2, K equals to 3.. and then we look at what value of K gives us the best performance on the validation set and then we can take that value and use that as the final setting of our algorithm so we are minimizing the validation error .
5. In general, practice, choosing the value of k is k = Square-Root(N) where N stands for the number of samples in your training data-set.

6) Try and keep the value of k odd in order to avoid confusion between two classes of data

**2.7 Conclusions**

From this chapter we know the details about the related work of our thesis work. We have learned many things by studying all these systems about Learning Methods and classification. Moreover, we gather knowledge and also came to learn about much functionality related to Machine Learning (ML) and also about the modules from this existing systems we discussed in this chapter. These ML systems were very useful for developing our system and to get the concept of developing better idea on this work. Now a day KNN algorithm may not be very popular all over the world. But we hope that it will be in the upcoming future as this system and method is easy to understand than the existing system and also the performance is higher.

**Chapter 3**

**Classification With K Nearest Neighbor**

**3.1** **Problem description and goals**

The problem we are trying to solve is to identify the students’ upcoming semester’s probable grades and overall score. To solve this problem, we’ve takes a step in this thesis and trying to minimizing the error rate of result.

There are several data columns we are considering, these are:

1. Student gender
2. Faculty info
3. Past institutional result
4. Previous semester’s results

These information are gathered first and we run them using our model. While running the algorithm, we consider which column is useful and which is not to get the most and best result from the data.

We are using KNN to solve this problem and current approach is a part of the problem that we are solving.

We implemented the model in five steps,

1. Step-1: Selecting the number K of the neighbors
2. Step-2: Calculating the Euclidean distance of K number of neighbors
3. Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
4. Step-4: Among these k neighbors, count the number of the data points in each category.
5. Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
6. Step-6: Our model is ready.

**3.2 In Depth Explanation of KNN Algorithm**

KNN is used for classifying the data into clusters in unsupervised learning.

It is also used in detecting the multiple outliers easily which is not achievable with other techniques. The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems.

Let’s take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS) :

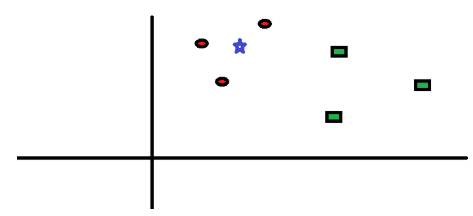


Fig: a spread of red circles (RC) and green squares (GS)

We intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The “K” is KNN algorithm is the nearest neighbor we wish to take the vote from. Let’s say K = 3. Hence, we will now make a circle with BS as the center just as big as to enclose only three data points on the plane.

Refer to the following diagram for more details:

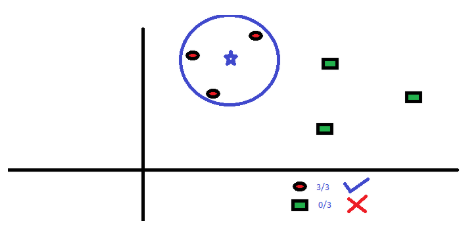


Fig: K = 3, a circle with BS as the center just as big as to enclose only three data points on the plane.

The three closest points to BS is all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm. Next, we will understand what are the factors to be considered to conclude the best K.

**How do we choose the factor K?**

K in KNN is the number of nearest neighbors considered for assigning a label to the current point. K is an extremely important parameter and choosing the value of K is the most critical problem when working with the KNN algorithm. The process of choosing the right value of K is referred to as parameter tuning and is of great significance in achieving better accuracy. If the value of K is too small then there is a probability of over fitting the model and if it is too large then the algorithm becomes computationally expensive. Most data scientists usually choose an odd number value for K when the number of classes is 2. Another formula that works well for choosing K is, k- sqrt(n) where n is the total number of data points.

Selecting the value of K depends on individual cases and sometimes the best method of choosing K is to run through different values of K and verify the outcomes. Using cross-validation, the KNN algorithm can be tested for different values of K and the value of K that results in good accuracy can be considered as an optimal value for K.

First let us try to understand what exactly does K influence in the algorithm. If we see the last example, given that all the 6 training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. In the same way, let’s try to see the effect of value “K” on the class boundaries. The following are the different boundaries separating the two classes with different values of K.

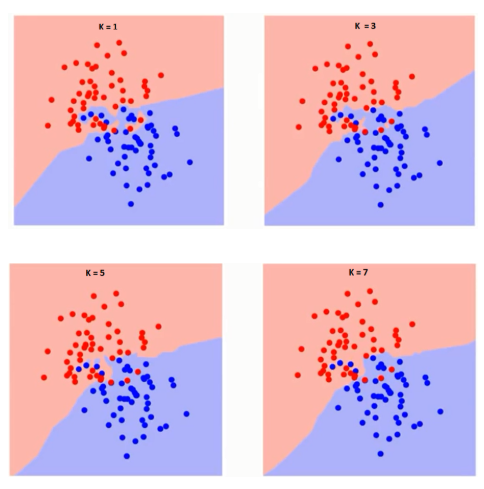
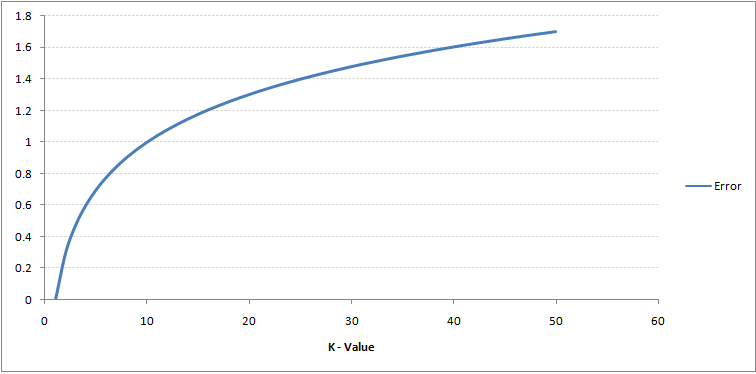


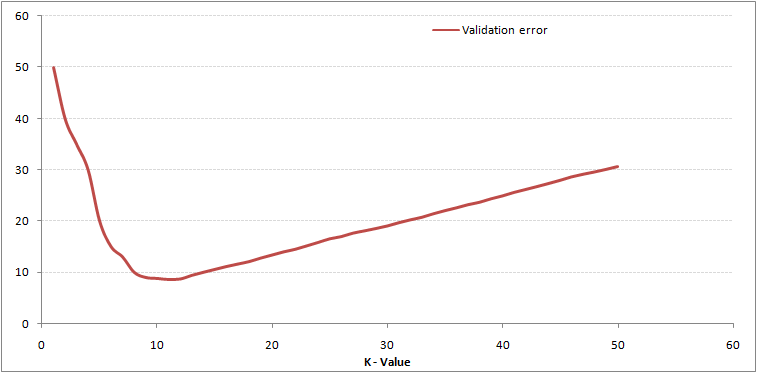
Fig: Different boundaries separating the two classes with different values of K

Here, we can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority.  The training error rate and the validation error rate are two parameters we need to access different K-value.

Following is the curve for the training error rate with a varying value of K :



As you can see, the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself.Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1. Following is the validation error curve with varying value of K:

****

This makes the story more clear. At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

**3.3 Pseudo-code of KNN**

1. Load the data

2. Initialize K to our chosen number of neighbors

3. For each example in the data

3.1 Calculate the distance between the query example and the current example from the data.

3.2 Add the distance and the index of the example to an ordered collection

4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances

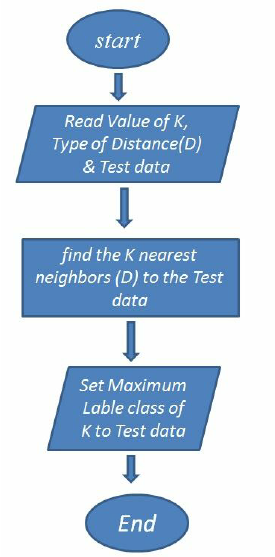
5. Pick the first K entries from the sorted collection

6. Get the labels of the selected K entries

7. If regression, return the mean of the K labels

8. If classification, return the mode of the K labels

**3.4 Flowchart**



**Figure 3.6: Flowchart of KNN Algorithm**

**3.5 Advantages**

KNN is a very simple algorithm used to solve classification problems. KNN stands for K-Nearest Neighbors. K is the number of neighbors in KNN. The main advantage of our system is that, we endeavored to develop a model that will result with better accuracy level. By this method we can gain a certain level of accuracy as we may need according to situation each time. For different situation it may be time efficient too while we may need less accuracy. It can also be faster in that case of operation.

The advantages of KNN is summarized by following points:

1. KNN is a perfect first step for machine learning beginners as it is very easy to explain, simple to understand, and extremely powerful. It yields highly competitive results, despite its simplicity. A fantastic application of this is the use of KNN in collaborative filtering algorithms for recommendation systems. This is the go-to technique behind the screens of Amazon’s Recommendation Systems.
2. KNN is a non-parametric algorithm and does not require any assumptions on the data distribution. This gives KNN an extra edge in specific settings where the data is highly unusual. This is the reason for KNN being the first choice when there is no prior knowledge or very little knowledge about the data distribution.
3. It is a versatile supervised machine learning algorithm that can be used for both regression and classification problems and also search.
4. This algorithm does not have an explicit training step as it is an instance-based learning algorithm. The training step of KNN is pretty fast as it involves only storing feature vectors and class labels of the training samples. Considering the minimal training time, KNN can be a perfect choice for off-the-bat analysis of a dataset on which you are planning to run complex algorithms.
5. Most of the classification algorithms are by default hard-coded for the binary setting. Using them for multi-class problems requires extension from binary or transformation to binary. KNN easily lends itself with multiclass datasets.
6. Flexible distance criteria to choose from when building a KNN model – Euclidean, Manhattan, and Hamming distance. Each of the distance functions has a different purpose based on the type of dataset. Based on the nature of features, it’s possible to choose the best option -Manhattan and Euclidean for numeric, and Hamming for categorical features.

**Disadvantages**

1. The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

**3.6 Conclusions**

In this chapter we have discussed about total procedure of our KNN classification system. We provided all the necessary discussions. We added the necessary diagrams for each steps of the system for explaining. We also tried to discuss all the details of the system as easy as it can be. For better understanding we added a flow chart of the proposed model. Moreover, the algorithm and a full example is discussed in this chapter. All the calculations were shown step by step with tables and values also. Finally we also tried to make understand why our system is better and the advantages of this proposed model

The algorithm can be summarized as:

1. A positive integer k is specified, along with a new sample
2. We select the k entries in our database which are closest to the new sample
3. We find the most common classification of these entries
4. This is the classification we give to the new sample

A few other features of KNN:

1. KNN stores the entire training dataset which it uses as its representation.
2. KNN does not learn any model.
3. KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance

**Chapter – 4**

**Experimental Results**

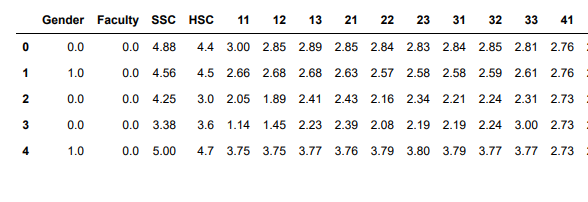
**4.1 Introduction of the Experimental Result Done by KNN**

The KNN algorithm can compete with the most accurate models because it makes highly accurate predictions. Therefore, we 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 and the amount of data is comparably smaller.

**4.2 Result Analysis**

At the iteration, we need to know which is best value of K to be chosen to do our analysis. To do that, KNN algorithm will find the *best attribute* which is has maximum accuracy. Given the information gain for each attribute.

Here are some of the data columns after normalizing the data classes:

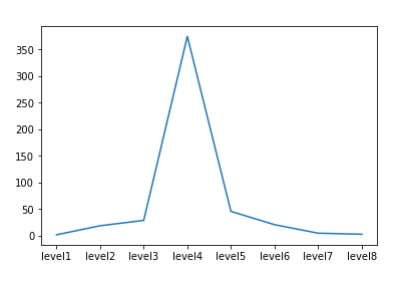


|  |  |  |
| --- | --- | --- |
| Level | StartValue | EndValue |
| 1 | 3.75 | 4 |
| 2 | 3.5 | 3.74 |
| 3 | 3.25 | 3.49 |
| 4 | 2.75 | 3.24 |
| 5 | 2.5 | 2.74 |
| 6 | 2.25 | 2.49 |
| 7 | 2 | 2.24 |
| 8 | 0 | 1.99 |

To get the density of the student results, we have used 8 types of category based on the average values. Here’s the list:

Here is t

The basic evaluation graph of the data density in 8 levels.



**4.2.1 Correlation**

The value of correlation helps us understand the relation between each 2 columns in the dataset. The value is found by using this equation:



Eq: Correlation

The correlation among the data columns after dropping these columns:

Gender, Faculty, SSC, HSC

Because those information seems not useful for the process.

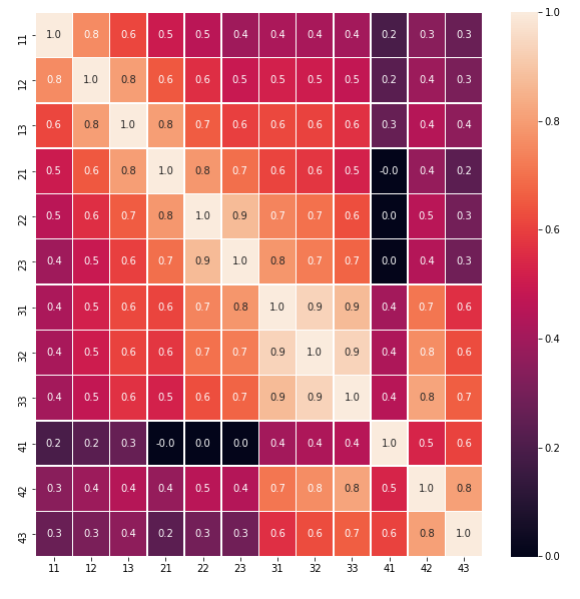


Fig: Correlation Heatmap of 12 columns of the Semester results.

**4.2.2 Distance Measurement**

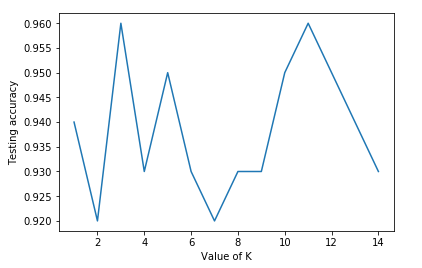
To measure the distance between 2 data points in the K area points, we’ve used the Manhattan distance formula.



Eq: Manhattan distance

To get the value of K for different iteration, here’s the list of different values of K with the accuracy result:

The result analysis chart is here:



As an example, these values of K gave us one of the most accuracy in predicting the class based on the training data.

K(2.5) = 94%

K(3) = 96%

K(11) = 95%

Compare all the information gain and we see that information , K=3 gave us the best result for this whole process.

**4.2.3 Accuracy calculation using Confusion matrix**

The performance of the model is described by a confusion matrix. There are 5 classes to consider and this is how the values are produced.

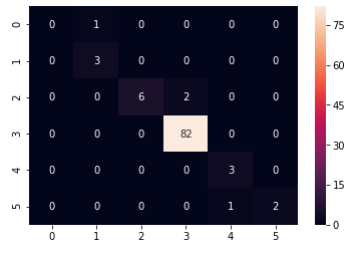
****

Fig: confusion matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cluster | 1 | 2 | 3 | 4 | 5 | 6 |
| Majority | 0 | 3 | 6 | 82 | 3 | 2 |
| Total | 0 | 4 | 6 | 84 | 4 | 2 |

Calculating accuracy and error rate:

****

****

****

The Error rate is

****

For any case, when there are same number of counts in the same y axis,

Hungarian Algorithm is widely used.

**4.3 Applications**

**KNN Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events**

In the present study k-Nearest Neighbor classification method, have been studied for economic forecasting. Due to the effects of companies’ financial distress on stakeholders, financial distress prediction models have been one of the most attractive areas in financial research. In recent years, after the global financial crisis, the number of bankrupt companies has risen. Since companies' financial distress is the first stage of bankruptcy, using financial ratios for predicting financial distress have attracted too much attention of the academics as well as

economic and financial institutions. Although in recent years studies on predicting companies’ financial distress in Iran have been increased, most efforts have exploited traditional statistical methods; and just a few studies have used nonparametric methods. Recent studies demonstrate this method is more capable than other methods. [24]

**KNN application on student result prediction**

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. k-Nearest Neighborhood (k-NN) classification is a method adoptable for classifying entities based on closest training examples in a feature space. k-NN is a lazy learning classifiers that adopts instance-base learning hence having prediction done in two stages.

Firstly, it undergoes minimal operations of analyzing the attribute values of individual instances in training dataset [23]. k-NN can be been trained for online and genuine time analysis of data to identify interestingness intrinsical to data stream, match particular utilizer group for relegation, or recommend exhaustive options that meet categorical users’ needs. Though the technique requires extravagant resources during computation, but it is transparent, consistent, straightforward, simple and facile to implement with high proclivity to possess desirable qualities than most other data mining techniques, categorically when there is diminutive or no prior erudition about data distribution. Popular techniques of K-NN classifiers are Euclidean distance or cosine homogeneous attribute between training and test datasets. In both techniques, the entity to be presaged is assigned a mundane class among its k-most proximate neighbors, weights are assigned to culled variables, and the strepitous data are pruned.



Fig : [Image showing how similar data points typically exist close to each other](https://commons.wikimedia.org/wiki/File:Map1NNReducedDataSet.png" \t "_blank)

Notice in the image above that most of the time, kindred data points are proximate to each other. The KNN algorithm hinges on this postulation being true enough for the algorithm to be utilizable.

KNN captures the conception of kindred attribute (sometimes called distance, proximity, or propinquity) with some mathematics we might have learned in our childhood—calculating the distance between points on a graph.

There are other ways of calculating distance, and one way might be preferable depending on the problem we are solving. However, the straight-line distance (also called the Euclidean distance) is a popular and familiar choice.

**Web Attack Detection**

Using KNN detection of web attacks can be pretty useful. As today’s smart technologies enable every operation to perform online, transactions has increased a lot. At the same time, the attacks on these online sites have also increased. Hence many organizations prefer an Intrusion Detection System.

For this purpose, we can classify the attack data and get the basic idea of the type of the attack and gather more meta-data on those.

Finally, required measures can be taken based on the found information.

**Application of KNN in Diabetes**

Diabetes is one among the challenging diseases that the human race is finding difficulties which is prevailing since years. Many of the analysts around the globe have been working on diabetes and making aware of the signs and effects of the disease on the various organs of the body.

KNN can be used to identify the type of diabetes and it also can be used to monitor daily activities. This is suitable for classification and regression process it can be easily used to identify and monitor a patient regularly.

**4.4 Conclusions**

KNN relies on majority voting based on class membership of K nearest samples for a given test data sample. The nearness of samples is typically based on Euclidean distance. Most of the times, data-set will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Eucledian or Manhattan distance between two data points in their computations, this is a problem. If left alone, these algorithms only take in the magnitude of features neglecting the units.

The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes. Any algorithm where distance play a vital role for prediction or classification, we should normalize the feature as we do the same process in KNN also.

**Chapter – 5**

**Conclusions**

**5.1 Conclusion**

We tried to gain a result that defines a precise result of those 500 students, whose data has been trained by KNN Classifier method. The data is genuinely a combination of their CGPA's and results in their SSC, HSC & 1-12th semesters in their university period & evaluate the performance of each student predicated on the data they provided or predicated on their educational information.

We have proposed an approach inspired by canonical correlation analysis for discovering interrelationship between learning resources of different types, only using student performance in them. This approach can also be used to predict students' performance. That is to say, we can predict students' performance in one type of learning resources, with the help of student activities in another resource type.

In the existing system we can evaluate the students result that is the main goal of our work and we successfully achieve the result of the goal without any hesitation. The prediction of student performance is getting difficult day by day. In this research we have developed a linear regression based model which will help students in knowing final grade in particular subject. To accomplish this research, internal semester result SGPA and CGPA are taken into consideration. Then the marks are converted into 100 (percentage) to have uniformity benchmark. These data is used to train the linear regression model to calculate the appropriate value. This model is a univariate i.e. it takes only one variable but it can be extended as multivariate model by adding more parameters to get more accurate results.

In future we want to work with bigger data sets with some new results like the student’s attendance whether the attend the class or absence, complete the semester properly with good result or bad result. For future works, how to quickly and accurately adapt to more new samples in online classification systems should be researched, and choosing a more efficient assessment method that can reasonably assign the training set and the testing set is necessary. We have majorly focused on more the accuracy & gaining correct information.

**5.2 Limitations**

We tried to develop a system so that this algorithm can be recognized in a easy, fast and more accurate way. We also tried it to be dynamic so that it can be used in different critical situations. For our work we used powerful tools and updated software and technologies. Still our proposed system may face some drawbacks and some are listed below:

1. Data may be over-fitted or over-classified, if a small sample is tested.
2. Only one attribute at a time is tested for making a decision.
3. Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.
4. To get the accuracy higher, the range of a net's weights and of the weight updates is very important for performance. We may need to normalize the input values in some cases when the features are different like in many of our given datasets and updates will all be on different systems.

**5.3 Future Works**

Our system can be used in various important different works and can be implemented easily. Although a lots of works are being done each and every day of this world This thesis presented a survey.. We thought about our work to be used in future works. Some are mentioned bellow.

1. We can achieve more accuracy can be achieved in many sectors based system through various problems using this system based on deep learning.
2. We can implement this system for automated traffic system from Traffic Signs Recognition and in this way we can develop an automated traffic system and can solve the traffic problems in many countries.
3. Space research field recognizes planetary star images and their movements and positions can be tracked by using this system as I will use unsupervised learning and deep learning so it will be easy apply in unknown situations
4. We can use this system to retrieve weather preview after experiencing weather image recognition and using the results we can develop systems for predicting weather conditions or can predict dangerous situations earlier to save lives and properties

**References**

1. Punlumjeak, Wattana, Nachirat Rachburee, and Jedsada Arunrerk. "Big data analytics: student performance prediction using feature selection and machine learning on microsoft azure platform." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.1-4 (2017): 113-117.
2. Sweeney, Mack, et al. "Next-term student performance prediction: A recommender systems approach." *arXiv preprint arXiv:1604.01840* (2016).
3. Mohammadi, Mehdi, et al. "Comparative study of supervised learning algorithms for student performance prediction." *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2019.
4. Kaur, Harleen, and Er Gourav Bathla. "Student performance prediction using educational data mining techniques." *International Journal on Future Revolution in Computer Science & Communication Engineering* 4.12 (2018): 93-7.
5. Parmar, Krina, Dineshkumar Vaghela, and Priyanka Sharma. "Performance prediction of students using distributed Data mining." *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 2015.
6. Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems*. 2015.
7. Ma, Yuling, et al. "Pre-course student performance prediction with multi-instance multi-label learning." *Science China Information Sciences* 62.2 (2019): 29101.
8. Al-Shehri, Huda, et al. "Student performance prediction using support vector machine and k-nearest neighbor." *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2017
9. Kabakchieva, Dorina. "Student performance prediction by using data mining classification algorithms." *International journal of computer science and management research* 1.4 (2012): 686-690.
10. Devasia, Tismy, T. P. Vinushree, and Vinayak Hegde. "Prediction of students performance using Educational Data Mining." *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. IEEE, 2016.
11. Hasan, Raza, et al. "Student Academic Performance Prediction by using Decision Tree Algorithm." *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, 2018.
12. Hasan, Raza, et al. "Student Academic Performance Prediction by using Decision Tree Algorithm." *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, 2018.
13. Luo, Ling, Irena Koprinska, and Wei Liu. "Discrimination-Aware Classifiers for Student Performance Prediction." *International Educational Data Mining Society* (2015).
14. Xu, Jie, et al. "Progressive prediction of student performance in college programs." *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.
15. Saa, Amjad Abu. "Educational data mining & students’ performance prediction." *International Journal of Advanced Computer Science and Applications* 7.5 (2016): 212-220.
16. Gadhavi, Mahesh, and Chirag Patel. "Student final grade prediction based on linear regression." *Indian J. Comput. Sci. Eng.* 8.3 (2017): 274-279.
17. Jamil A, Ahsan M, Farooq T, Hussain A, Ashraf R. "Student Performance Prediction Using Algorithms of Data Mining". In2018 International Conference on Computing, Engineering, and Design (ICCED) 2018 Sep 6 (pp. 244-249). IEEE.
18. Kumar, M., Singh, A.J. and Handa, D., 2017. "Literature survey on student’s performance prediction in education using data mining techniques." *International Journal of Education and Management Engineering*, *7*(6), pp.42-49.
19. Sahebi, Shaghayegh, and Peter Brusilovsky. "Student Performance Prediction by Discovering Inter-Activity Relations." *International Educational Data Mining Society* (2018).
20. Roy, S., Nag, S., Maitra, I.K. and Bandyopadhyay, S.K., 2013. "International Journal of Advanced Research in Computer Science and Software Engineering." *International Journal*, *3*(6).
21. Zhao, Z., Zhang, L. and Hu, D. 2016 August, "Student Performance Prediction Method Based on Fuzzy Cognitive Model." In *2016 International Conference on Education, E-learning and Management Technology*. Atlantis Press.
22. Bydžovská, H., 2015, June. " Student performance prediction using collaborative filtering methods." In *International Conference on Artificial Intelligence in Education* (pp. 550-553). Springer, Cham.
23. Omisore, O. M., and N. A. Azeez. 2015 "Predicting Academic Performance of Students in Higher Institutions with k-NN Classifier." *International Conference on Computer Science Research and Innovations.*
24. S B Imandoust et al. 2013, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: " Theoretical Background. *Int. Journal of Engineering Research and Application Vol. 3, Issue 5, Sep-Oct 2013, pp.605-610*