EDUCATIONAL DATA MINING

Major Project Report

Submitted for Assessment for 6th semester.

Presented by,

SOHOM RANA

Registration Number: 151170510041 of 2015-16

Roll Number: 11701015040

Under the supervision of

Soumen Mukherjee

Asst. Professor at RCC Institute of Information Technology

At

RCC Institute of Information Technology

(Affiliated to Maulana Abul Kalam Azad University of Technology)

Canal South Road, Beliaghata, Kolkata-700015

May 2018

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Kolkata-700015, India

CERTIFICATE

This is to certify that the project titled ***Educational Data Mining*** submitted by **Sohom Rana** (Roll number **11701015040** of **MCA** Department) has been prepared under my/our supervision for the major project assessment, 6th semester.

[Signature of the Guide]

Soumen Mukherjee
Asst. Professor
Dept. of CA
RCC Institute of Information Technology

Countersigned by,

[Signature of the Head of the Department]

Arup Kumar Bhattacharya
Dept. of CA
RCC Institute of Information Technology

Declaration by Author

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

[Signature of Author]

SOHOM RANA
Roll No.: 11701015040
Registration No.: 151170510041 of 2015-16

Acknowledgement

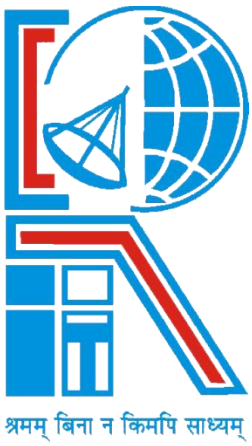
I express my sincere gratitude to Professor **Soumen Mukherjee** (Asst. Professor of Department of CA, RCCIIT) for extending his valuable time to guide me to take up this project and see it through.

[Signature of the student]

SOHOM RANA
Roll No.: 11701015040
Registration No.: 151170510041 of 2015-16

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Kolkata-700015, India



CERTIFICATE OF ACCEPTANCE

This is to certify that the project titled ***Educational Data Mining*** submitted by **Sohom Rana** (Roll number **11701015040** of **MCA** Department) is hereby recommended to be accepted for assessment for 6th Semester Examination in **Maulana Abul Kalam Azad University of Technology**.

Name of the Examiner(s):

Signature with Date:

_____

Date:

Abstract

Educational Data Mining (EDM) is the application of Data Mining (DM) techniques to educational data for the purpose of extracting useful information on the behaviours of students in the learning process. **Data Mining (DM)** processes large databases to extract useful informations. In this project, among the two broad categories of data mining tasks, descriptive and predictive, we focused more on the descriptive category.

The data set is taken from two Portuguese schools contains 33 features of 649 students. The main objective of the project is to analyse the data set to determine how the features are related to the target attribute, the final grade. For this, two different aspects were covered. Firstly, three different ranking methods, RelifF, CFS, and IICFS were applied to the data set. These different rankings are combined to get the final rank of each attribute. Secondly, to know how the features behave to get better final grade two different methods are applied, value wise average method and value wise number of occurrences method. Value wise average method computes the average of the students' final grade for distinct values of each feature to make a comparative result among the distinct values of features. In value wise number of occurrences method students' final grade is classified into four groups and checks for the number of occurrences for the groups for distinct values of each feature.

Table of Contents

1. Introduction

In the recent years the most important innovation that has changed the face of modern education is the Internet. Web based education i.e. E-learning is getting popular day by day among students, teachers and educational organizations. This e-learning systems deals with a lot of data. Although EDM is used in very few online education courses till now, it is going to be used in the offline education institutes, like schools, colleges very soon.

Through EDM the whole education system is moving to the next generation, in which a large amount of data will be collected from each and every student in time of the admission. That data may contains some sort of personal information like, parent's cohabitation status, quality of family relationships, students' alcohol consumption, and many more. All these data will be provided to the machine learning algorithms and a prediction result will be produced. Prediction result does not contain only a final grade but also some more significant qualities of the student, like which type of education method the student likes, which type of book the student prefers to read for knowledge etc. According to that prediction result the type of teaching for each and every student will be set, like students who prefer to study by hands on practical, their syllabus can be designed with more practical classes, low scoring students can be provided extra classes for better result, students will be provided study materials according to their merits and many more realistic cases like this.

EDM has to go a long way to meet the success by which it can be implemented in the educational systems globally. But EDM is one of the most popular research topic around the world. Hopefully it will meet the success very soon.

Some related works were done in various fields of EDM in the research papers [2], [3] and [4].

2. Data Set and IDE Details

The data set contains student achievement in secondary education of two Portuguese schools, *Gabriel Pereira* and *Mousinho da Silveira*. It was collected by using school reports and questionnaires. The data attributes include student grades, demographic, social and school related features [1].

Total no of instances: 649

Total no of attributes: 33

Missing Values: N/A

Date Donated: 2014-11-27

Attribute Characteristics: Integer

Data set copy write: https://archive.ics.uci.edu

Data set Source: [6]

Here the 33rd attribute, final grade is the output target. During the ranking of the attributes, 33rd attribute is used as the target variable to rank the other variables. During the analyzing relationship, we figure out the value of the 33rd variable based on the different values of other variables.

IDE Used in the project: Matlab R2016a

3. Approach Used

In order to analyze and describe the data in different aspects, two approaches are used.

3.1 Feature Selection

Feature selection technique is very helpful in machine learning and data mining fields. This technique is used to select the useful subset of input variables by terminating relatively less important features to provide effective result. The main objectives of feature selection are to avoid over fitting, improve model performance and to provide faster and more cost-effective models.

In this project, we used three feature selection methods to rank the attributes based on the importance of predicting the target attribute, final grade.

3.1.1Relieff Method

Relieff estimates rank and weight of predictors i.e., the attributes of index 1 to 32 using ReliefF or RReliefF algorithm.

Properties of Relieff Method

- *[ranks, weights] = relieff(X, y, k)* returns the ranks and weights of predictors where X is the input data matrix and y is the response vector. For example, if ranks(3) is 10, then the third most important predictor is the tenth column in X.
  weights indices the weights of the predictors. *weights* range varies from -1 to 1.
- k is the number of nearest neighbours [7].

Function to determine k:

```
X=stu_por(:,1:32); % predictor variables
Y=stu_por(:,33); % response variables
n=1;
rankRelieff= zeros(13,32);
weightRelieff= zeros(13,32);
for k=10:30:100
    [rankRelieff(n,:),weightRelieff(n,:)] =
relieff(X,Y,k);
```

```
    n=n+1;
end
for k=200:100:1000
    [rankRelieff(n,:),weightRelieff(n,:)] =
relieff(X,Y,k);
    n=n+1;
end
n=1;
rankRelieff(:,33)= zeros(13,1);
weightRelieff(:,33)= zeros(13,1);
for k=[10,40,70,100,200,300,400,500,600,700,800,900,1000]
    rankRelieff(n,33)=k;
    weightRelieff(n,33)=k;
    n=n+1;
end
rankRelieff = rankRelieff';
```

Reliff Function to get rank and weight:

```
X=stu_por(:,1:32); % predictor variables
Y=stu_por(:,33); % response variables

[ranks,weights] = relieff(X,Y,100); % the value of k is
taken as 100 because the result of the above function
shows ranks do not change after k reached to 100
```

3.1.2CFS Method

Correlation Feature Selection (CFS) method measures rank and weight of attributes based on the internal correlation between the attributes. There are two categories of CFS method: supervised and unsupervised.

Supervised CFS method takes the predictor variables and the response variables separately to rank the attributes.

Unsupervised CFS method takes only one set of variables to rank all the attributes.

Supervised CFS function:

```
function [weight,rank] = cfs_supervised(X,Y)

corrMat = abs( corr(X,Y) );
[weight,rank] = sort(corrMat,'descend');
```

```
end
```

Unsupervised CFS function:

```
function [weight,rank] = cfs_unsupervised(X)

 corrMat = abs( corr(X) );
 scoring = min(corrMat,[],2);
 [weight,rank] = sort(scoring,'descend');

end
```

[9]

3.1.3LLCFS Method

Local Learning based Clustering Feature Selection (LLCFS) involves two sub steps:

1. It creates the k-nearest neighbor graph in the weighted feature space.

2. It performs joint clustering to calculate weight of each feature [5].

3.1.4Combining Ranks

In the previous section three different ranking methods were discussed. These methods provide different ranks and weights. Now we need a single ranking of attributes to know which attributes are more important. In order to do this we used the above mentioned methods as discussed below

Preparing ranks and weights matrix

We prepare a matrix to keep all the methods' rankings and weights. The rows of the matrices contain:

- **Row 1**: The outcome of relieff function, i.e. the rank and weight output from the `relieff(X,Y,100)` is stored in the first row of *ranks* and *weights* matrix.
- **Row 2**: The outcome of supervised CFS method, i.e. the rank and weight output from the `cfs_supervised(X,Y)` is stored in the second row of *ranks* and *weights* matrix.
- **Row 3**: The outcome of unsupervised CFS method. But the output from `cfs_unsupervised(K)` contains the 33rd attribute as well. For this at

first we have excluded that value from both the rank and weight output. Then it is stored in the third row of *ranks* and *weights* matrix.

- **Row 4**: The outcome of LLCFS method. In this case we send the whole data set, including the target attribute as the argument. So, we get the output form `rank_llcfs(K)`, including the 33rd attribute. Again we exclude those values from both the rank and weight output. Then it is stored in the fourth row of *ranks* and *weights* matrix.
- **Row 5**: The outcome of LLCFS method. In this case we send the predictive variables only as the argument. So, we get the output form `rank_llcfs(X)`, excluding the 33rd attribute. So, it is stored in the fifth row of *ranks* and *weights* matrix directly.

Function to prepare the matrix:

```
X=stu_por(:,1:32); % predictor variables
Y=stu_por(:,33); % response variables
K = stu_por;

% ReliefF Method (Row 1)
[ranks(1,:),weights(1,:)] = relieff(X,Y,100);

% CFS Method supervised(Row 2)
[weights(2,:),ranks(2,:)] = cfs_supervised(X,Y);

% CFS Method unsupervised(Row 3)
[tmpWt,tmpRnk] = cfs_unsupervised(K);
% deleting 33rd attribute
toDel = tmpRnk ~= 33;
weights(3,:) = tmpWt(toDel);
ranks(3,:) = tmpRnk(toDel);

% LLCFS Method with target variable (Row 4)
[tmpWt,tmpRnk] = rank_llcfs(K);
% deleting 33rd attribute
toDel = tmpRnk ~= 33;
weights(4,:) = tmpWt(toDel);
ranks(4,:) = tmpRnk(toDel);

% LLCFS Method without target variable (Row 5)
[weights(5,:),ranks(5,:)] = rank_llcfs(X);
```

Extracting a single rank:

In order to get a single rank, we summed up all the weights we got from the functions, and then we sort them to get the final rank.

Function to get the single rank:

```
[m,n] = size(ranks);
comboIndex = zeros(m,n);
comboWeight = zeros(m,n);
for i=1:m
    [~,comboIndex(i,:)] = sort(ranks(i,:),'ascend');
    [comboWeight(i,:),~] = sort(weights(i,:),'ascend');
end

comboIndex(m+2,:) = sum(comboIndex,1);
comboWeight(m+2,:) = sum(comboWeight,1);
[~,combinedResult(1,:)] = sort(comboIndex(m+2,:),'ascend');
[combinedResult(2,:),~] = sort(comboWeight(m+2,:),'ascend');
```

3.2 Evaluating Relationship between Major Attributes and the Target Attribute

After knowing which features are mostly important for the target attribute, final grade, now we tried to figure out how they are related with the target variable. That is, for which values of each variable students get more average marks and for which values more students performed relatively well. In this context we used two methods:

3.2.1 Value wise average method

In this method, we have analysed the distinct values of each prediction variable and computed the average final grade for each of the distinct value. From the result, we can see for which values the average final grade is higher. For example, for the 15th variable, number of past class failures, we get the results like which students have zero failures they have the average final grade of 12.5, whereas which students have failures they have the average final grade of around 8. Thus we can say that students who did not failed in previous classes they have better opportunity to score better in final.

The function to compute value wise average:

```
Y = stu_por(:,33);
p = 1;
for k=1:32
    X = stu_por(:,k);
    unq = unique(X);
```

```
    n = numel(unq);
    allAvgResult(p:p+n-1,1) = k;
    for i=1:n
        allAvgResult(p+i-1,2) = unq(i);
        f = X == unq(i);
        allAvgResult(p+i-1,3) = sum(Y(f))/numel(Y(f));
    end
    p = p+n;
end
```

3.2.2Value wise number of occurrences method

This is another approach to determine the relationship of the prediction attributes with the target variable. In this method we have analysed the distinct values of each prediction variable and we divided the final grade into 4 groups. The final grade is marked within 0 and 20. Now the groups are divided as following:

- Group 1: 0 to 4
- Group 2: 5 to 9
- Group 3: 10 to 14
- Group 4: 15 to 20

Now we computed how many students has scored in each group of final grade. The output report also shows the percentage of occurrences. Thus, we can determine which values helps students for better final grade.

Function to compute value wise number of occurrences:

```
Y = stu_por(:,33);
[M,~] = size(Y);
c = 4; % no of classifications
m = 1;
for k=1:32
    X = stu_por(:,k);
    unq = unique(X);
    n = numel(unq);
    p = m+n*c-1;
    result(m:p,1) = k;
    q=m;
    for i=1:n
        clear cf;
        f = X == unq(i);
        sample = Y(f);
        a = 0;
```

```
        d = (20/c)-1;
        d = int16(d);
        for j=1:c-1
            cf(:,j) = ((sample >= a) & (sample <= a+d));
            a = a+d+1;
        end
        cf(:,c) = sample >= a;
        result(q:q+c-1,2) = unq(i);
        numsamp = numel(sample);
        result(q,3) = numsamp;
        for j = 1:c
            result(q+j-1,4) = j;
            numRes = numel(sample(cf(:,j)));
            result(q+j-1,5) = numRes;
            result(q+j-1,6) = numRes*100/numsamp;
            result(q+j-1,7) = numRes*100/M;
        end
        q=q+c;
    end
    m = p+1;
end
```

4. Results and Discussion

The outputs of each approach used is mentioned below:

4.1 Feature Selection

- **Function to determine k in relieff method:**

The function to determine the proper number of the nearest neighbors is very important for reliefF method in order to avoid noisy data. The output of this program is stated below:

rankRelieff:

K	10	40	70	100	200	300	400	500	600	700	800	900	1000
1	32	32	32	32	32	32	32	32	32	32	32	32	32
2	31	31	31	31	31	31	31	31	31	31	31	31	31
3	15	15	15	15	15	15	15	15	15	15	15	15	15
4	21	21	21	21	21	21	21	21	21	21	21	21	21
5	1	1	1	1	1	1	1	1	1	1	1	1	1
6	25	27	27	27	27	27	27	27	27	27	27	27	27
7	7	4	4	4	4	4	4	4	4	4	4	4	4
8	27	26	26	26	26	26	26	26	26	26	26	26	26
9	2	25	25	25	25	25	25	25	25	25	25	25	25
10	26	7	7	24	24	24	24	24	24	24	24	24	24
11	4	22	24	7	7	7	7	7	7	7	7	7	7
12	13	24	22	22	22	22	22	22	22	22	22	22	22
13	8	28	28	28	28	28	28	28	28	28	28	28	28
14	3	2	2	2	2	2	2	2	2	2	2	2	2
15	24	13	3	3	3	3	3	3	3	3	3	3	3
16	17	3	11	11	11	11	11	11	11	11	11	11	11
17	6	11	13	13	13	13	13	13	13	13	13	13	13
18	23	30	30	30	30	30	30	30	30	30	30	30	30
19	28	23	23	23	23	23	23	23	23	23	23	23	23
20	20	8	8	8	8	8	8	8	8	8	8	8	8
21	18	16	14	14	14	14	14	14	14	14	14	14	14
22	22	14	18	18	18	18	18	18	18	18	18	18	18
23	16	17	17	17	17	17	17	17	17	17	17	17	17
24	11	18	16	16	16	16	16	16	16	16	16	16	16
25	30	9	6	6	6	6	6	6	6	6	6	6	6
26	14	6	10	10	10	10	10	10	10	10	10	10	10
27	5	10	9	9	9	9	9	9	9	9	9	9	9
28	19	20	29	29	29	29	29	29	29	29	29	29	29
29	9	29	20	20	20	20	20	20	20	20	20	20	20

30	12	19	12	12	12	12	12	12	12	12	12	12	12
31	10	12	19	19	19	19	19	19	19	19	19	19	19
32	29	5	5	5	5	5	5	5	5	5	5	5	5

Table 1

weightRelieff

K	40	70	100	200	300	400	500
1	0.019046	0.021217	0.021899	0.022	0.022	0.022	0.022
2	0.001906	0.001744	0.001746	0.001796	0.001796	0.001796	0.001796
3	0.00115	0.001011	0.001064	0.001065	0.001065	0.001065	0.001065
4	0.00756	0.00838	0.008792	0.008841	0.008841	0.008841	0.008841
5	-0.01067	-0.01228	-0.01216	-0.01219	-0.01219	-0.01219	-0.01219
6	-0.00435	-0.00374	-0.0038	-0.00379	-0.00379	-0.00379	-0.00379
7	0.005634	0.004963	0.004817	0.004845	0.004845	0.004845	0.004845
8	-0.00134	-0.00146	-0.00138	-0.00136	-0.00136	-0.00136	-0.00136
9	-0.00388	-0.00416	-0.00407	-0.00406	-0.00406	-0.00406	-0.00406
10	-0.00541	-0.00403	-0.00387	-0.00383	-0.00383	-0.00383	-0.00383
11	0.000485	0.000763	0.000754	0.00078	0.00078	0.00078	0.00078
12	-0.0102	-0.00988	-0.00997	-0.00994	-0.00994	-0.00994	-0.00994
13	0.001175	0.000601	0.000553	0.00053	0.00053	0.00053	0.00053
14	-0.00305	-0.00193	-0.00172	-0.0017	-0.0017	-0.0017	-0.0017
15	0.027536	0.029145	0.02955	0.029569	0.029569	0.029569	0.029569
16	-0.00291	-0.00361	-0.0037	-0.00371	-0.00371	-0.00371	-0.00371
17	-0.00335	-0.00339	-0.0036	-0.00356	-0.00356	-0.00356	-0.00356
18	-0.00372	-0.00338	-0.00347	-0.00347	-0.00347	-0.00347	-0.00347
19	-0.00992	-0.01153	-0.01111	-0.01105	-0.01105	-0.01105	-0.01105
20	-0.00595	-0.00548	-0.00536	-0.00538	-0.00538	-0.00538	-0.00538
21	0.022624	0.024905	0.025437	0.025468	0.025468	0.025468	0.025468
22	0.004885	0.004418	0.004557	0.004599	0.004599	0.004599	0.004599
23	5.57E-05	0.000119	-0.00012	-9.5E-05	-9.5E-05	-9.5E-05	-9.5E-05
24	0.004373	0.004923	0.005044	0.005055	0.005055	0.005055	0.005055
25	0.006207	0.006326	0.006332	0.006342	0.006342	0.006342	0.006342
26	0.006499	0.006502	0.0067	0.006727	0.006727	0.006727	0.006727
27	0.011297	0.0122	0.012204	0.01222	0.01222	0.01222	0.01222
28	0.002174	0.003141	0.003245	0.003278	0.003278	0.003278	0.003278
29	-0.00727	-0.00504	-0.00465	-0.00459	-0.00459	-0.00459	-0.00459
30	0.000307	0.000238	0.000215	0.000226	0.000226	0.000226	0.000226
31	0.059563	0.060984	0.061344	0.061388	0.061388	0.061388	0.061388
32	0.094759	0.09613	0.096477	0.096511	0.096511	0.096511	0.096511

Table 2.1

K	600	700	800	900	1000
1	0.022	0.022	0.022	0.022	0.022
2	0.001796	0.001796	0.001796	0.001796	0.001796
3	0.001065	0.001065	0.001065	0.001065	0.001065

4	0.008841	0.008841	0.008841	0.008841	0.008841
5	-0.01219	-0.01219	-0.01219	-0.01219	-0.01219
6	-0.00379	-0.00379	-0.00379	-0.00379	-0.00379
7	0.004845	0.004845	0.004845	0.004845	0.004845
8	-0.00136	-0.00136	-0.00136	-0.00136	-0.00136
9	-0.00406	-0.00406	-0.00406	-0.00406	-0.00406
10	-0.00383	-0.00383	-0.00383	-0.00383	-0.00383
11	0.00078	0.00078	0.00078	0.00078	0.00078
12	-0.00994	-0.00994	-0.00994	-0.00994	-0.00994
13	0.00053	0.00053	0.00053	0.00053	0.00053
14	-0.0017	-0.0017	-0.0017	-0.0017	-0.0017
15	0.029569	0.029569	0.029569	0.029569	0.029569
16	-0.00371	-0.00371	-0.00371	-0.00371	-0.00371
17	-0.00356	-0.00356	-0.00356	-0.00356	-0.00356
18	-0.00347	-0.00347	-0.00347	-0.00347	-0.00347
19	-0.01105	-0.01105	-0.01105	-0.01105	-0.01105
20	-0.00538	-0.00538	-0.00538	-0.00538	-0.00538
21	0.025468	0.025468	0.025468	0.025468	0.025468
22	0.004599	0.004599	0.004599	0.004599	0.004599
23	-9.5E-05	-9.5E-05	-9.5E-05	-9.5E-05	-9.5E-05
24	0.005055	0.005055	0.005055	0.005055	0.005055
25	0.006342	0.006342	0.006342	0.006342	0.006342
26	0.006727	0.006727	0.006727	0.006727	0.006727
27	0.01222	0.01222	0.01222	0.01222	0.01222
28	0.003278	0.003278	0.003278	0.003278	0.003278
29	-0.00459	-0.00459	-0.00459	-0.00459	-0.00459
30	0.000226	0.000226	0.000226	0.000226	0.000226
31	0.061388	0.061388	0.061388	0.061388	0.061388
32	0.096511	0.096511	0.096511	0.096511	0.096511

Table 2.2

From the above tables (1 and 2) we can see that the ranks are changing for the value of k up to 70. After than the ranks are tested up to 1000, but they seem constant. So, we took the value of k as 100 for the further methods.

- **Function to prepare the ranks and weights matrix:**

Ranks:

Ranks	Relieff	CFS Supervised	CFS Unsupervised	LLCFS with target variable	LLCFS without target variable
1	32	32	2	30	30
2	31	31	32	31	31
3	15	15	31	32	32

4	21	21	22	29	28
5	1	1	1	28	29
6	27	14	30	3	3
7	4	7	7	9	9
8	26	8	21	11	26
9	25	27	28	8	10
10	24	28	4	26	8
11	7	4	17	10	11
12	22	9	10	7	25
13	28	11	12	25	14
14	2	22	24	27	7
15	3	2	9	24	24
16	11	13	3	13	27
17	13	25	5	14	13
18	30	3	14	12	12
19	23	29	18	19	23
20	8	30	29	23	17
21	14	23	13	2	2
22	18	26	25	5	19
23	17	16	11	15	1
24	16	10	20	17	15
25	6	24	6	4	5
26	10	19	16	1	4
27	9	17	15	20	20
28	29	18	19	22	22
29	20	5	23	6	6
30	12	12	26	21	21
31	19	20	8	16	16
32	5	6	27	18	18

Table 3

Weights:

Rank	Relieff	CFS Supervised	CFS Unsupervised	LLCFS with target variable	LLCFS without target variable
1	0.021899	0.918548	0.019301	0.62958	0.886208
2	0.001746	0.826387	0.018689	0.196073	0.074752
3	0.001064	0.393316	0.015251	0.039643	0.03395
4	0.008792	0.332172	0.007159	0.011682	0.002053
5	-0.01216	0.284294	0.004659	0.002854	0.001098
6	-0.0038	0.249789	0.004645	0.002811	0.00084
7	0.004817	0.240151	0.004614	0.002586	0.000583
8	-0.00138	0.2118	0.004523	0.001488	0.000231

9	-0.00407	0.204719	0.004215	0.000676	0.000117
10	-0.00387	0.176619	0.003787	0.000367	8.82E-05
11	0.000754	0.167637	0.003764	0.000361	5.71E-05
12	-0.00997	0.16679	0.003189	2.77E-05	7.78E-06
13	0.000553	0.157546	0.003177	2.53E-05	5.06E-06
14	-0.00172	0.150025	0.002603	1.00E-05	3.49E-06
15	0.02955	0.129077	0.002603	8.52E-06	2.39E-06
16	-0.0037	0.127173	0.00247	5.87E-06	1.74E-06
17	-0.0036	0.122705	0.002448	4.97E-06	4.58E-07
18	-0.00347	0.106505	0.002314	1.27E-08	3.12E-09
19	-0.01111	0.098851	0.002314	5.50E-10	1.46E-10
20	-0.00536	0.091379	0.001701	3.52E-10	6.48E-11
21	0.025437	0.090583	0.000937	3.16E-10	6.47E-11
22	0.004557	0.087641	0.000937	2.29E-10	4.48E-11
23	-0.00012	0.066405	0.000899	1.14E-10	6.86E-12
24	0.005044	0.065454	0.000899	5.55E-11	6.06E-12
25	0.006332	0.063361	0.000754	2.69E-11	3.72E-12
26	0.0067	0.059791	0.000745	1.53E-11	2.51E-12
27	0.012204	0.059206	0.000561	5.91E-12	1.91E-12
28	0.003245	0.054898	0.000561	4.24E-13	1.39E-12
29	-0.00465	0.045016	0.00052	1.22E-14	2.54E-15
30	0.000215	0.029474	0.00052	9.69E-17	2.26E-16
31	0.061344	0.028752	6.08E-05	6.68E-17	2.17E-17
32	0.096477	0.000754	6.08E-05	3.68E-17	1.59E-17

Table 4

In the above tables (3 and 4) the rank and weight outcomes of the different feature selection methods are stored. The most important attribute index comes in the rank 1, the second most important comes next and so on.

- **Extracting a single rank:**

comboIndex:

Attribute Index	Relieff	CFS Supervised	CFS Unsupervised	LLCFS with target variable	LLCFS without target variable	Total
1	5	5	5	26	23	64
2	14	15	1	21	21	72
3	15	18	16	6	6	61
4	7	11	10	25	26	79
5	32	29	17	22	25	125
6	25	32	25	29	29	140
7	10	7	7	12	14	50

8	20	8	31	9	10	78
9	27	12	15	7	7	68
10	26	24	12	11	9	82
11	16	13	23	8	11	71
12	30	30	13	18	18	109
13	17	16	21	16	17	87
14	21	6	18	17	13	75
15	3	3	27	23	24	80
16	24	23	26	31	31	135
17	23	27	11	24	20	105
18	22	28	19	32	32	133
19	31	26	28	19	22	126
20	29	31	24	27	27	138
21	4	4	8	30	30	76
22	12	14	4	28	28	86
23	19	21	29	20	19	108
24	11	25	14	15	15	80
25	9	17	22	13	12	73
26	8	22	30	10	8	78
27	6	9	32	14	16	77
28	13	10	9	5	4	41
29	28	19	20	4	5	76
30	18	20	6	1	1	46
31	2	2	3	2	2	11
32	1	1	2	3	3	10

Table 5

comboWeight:

Attribute Index	Relieff	CFS Supervised	CFS Unsupervised	LLCFS with target variable	LLCFS without target variable	Total
1	-0.01228	0.000754	6.08E-05	3.68E-17	1.59E-17	-0.01147
2	-0.01153	0.028752	6.08E-05	6.68E-17	2.17E-17	0.017286
3	-0.00988	0.029474	0.00052	9.69E-17	2.26E-16	0.020111
4	-0.00548	0.045016	0.00052	1.22E-14	2.54E-15	0.040056
5	-0.00504	0.054898	0.000561	4.24E-13	1.39E-12	0.050418
6	-0.00416	0.059206	0.000561	5.91E-12	1.91E-12	0.055604
7	-0.00403	0.059791	0.000745	1.53E-11	2.51E-12	0.05651
8	-0.00374	0.063361	0.000754	2.69E-11	3.72E-12	0.060375
9	-0.00361	0.065454	0.000899	5.55E-11	6.06E-12	0.062745
10	-0.00339	0.066405	0.000899	1.14E-10	6.86E-12	0.063912
11	-0.00338	0.087641	0.000937	2.29E-10	4.48E-11	0.085198
12	-0.00193	0.090583	0.000937	3.16E-10	6.47E-11	0.089591
13	-0.00146	0.091379	0.001701	3.52E-10	6.48E-11	0.091624
14	0.000119	0.098851	0.002314	5.50E-10	1.46E-10	0.101284
15	0.000238	0.106505	0.002314	1.27E-08	3.12E-09	0.109058
16	0.000601	0.122705	0.002448	4.97E-06	4.58E-07	0.125759

17	0.000763	0.127173	0.00247	5.87E-06	1.74E-06	0.130413
18	0.001011	0.129077	0.002603	8.52E-06	2.39E-06	0.132702
19	0.001744	0.150025	0.002603	1.00E-05	3.49E-06	0.154385
20	0.003141	0.157546	0.003177	2.53E-05	5.06E-06	0.163895
21	0.004418	0.16679	0.003189	2.77E-05	7.78E-06	0.174433
22	0.004923	0.167637	0.003764	3.61E-04	5.71E-05	0.176743
23	0.004963	0.176619	0.003787	3.67E-04	8.82E-05	0.185825
24	0.006326	0.204719	0.004215	6.76E-04	1.17E-04	0.216054
25	0.006502	0.2118	0.004523	1.49E-03	2.31E-04	0.224544
26	0.00838	0.240151	0.004614	2.59E-03	5.83E-04	0.256313
27	0.0122	0.249789	0.004645	2.81E-03	8.40E-04	0.270285
28	0.021217	0.284294	0.004659	2.85E-03	1.10E-03	0.314123
29	0.024905	0.332172	0.007159	1.17E-02	2.05E-03	0.377971
30	0.029145	0.393316	0.015251	3.96E-02	3.40E-02	0.511306
31	0.060984	0.826387	1.87E-02	1.96E-01	7.48E-02	1.176885
32	0.09613	0.918548	1.93E-02	6.30E-01	8.86E-01	2.549767

Table 6

In the above tables (5 and 6), the rank and weight of each attribute is stored. That is, table 5 shows that attribute 1 has rank 3 for first three methods and has rank of 26 and 23 for fourth and fifth method.

combinedResult:

Rank	Attribte Index	Attribute Weight
1	32	-0.01147
2	31	0.017286
3	28	0.020111
4	30	0.040056
5	7	0.050418
6	3	0.055604
7	1	0.05651
8	9	0.060375
9	11	0.062745
10	2	0.063912
11	25	0.085198
12	14	0.089591
13	21	0.091624
14	29	0.101284
15	27	0.109058
16	8	0.125759
17	26	0.130413
18	4	0.132702

19	15	0.154385
20	24	0.163895
21	10	0.174433
22	22	0.176743
23	13	0.185825
24	17	0.216054
25	23	0.224544
26	12	0.256313
27	5	0.270285
28	19	0.314123
29	18	0.377971
30	16	0.511306
31	20	1.176885
32	6	2.549767

Table 7

In table 7, the final ranks and weights are stored after combining all the methods.

4.2 Evaluating Relationship between Major Attributes and the Target Attribute

4.2.1 Value wise average method

allAvgResult:

Attribute Index	Distinct Values	Average Final Grade
1	1	12.57683
	2	10.65044
2	1	11.40602
	2	12.25326
3	15	12.10714
	16	11.99435
	17	12.26816
	18	11.77143
	19	9.53125
	20	12
	21	11
	22	5
4	1	12.26327
	2	11.08629
5	1	12.13021
	2	11.81182
6	1	11.9051

	2	11.9125
7	0	11.66667
	1	10.7972
	2	11.66129
	3	11.92086
	4	13.06857
8	0	12.14286
	1	10.93678
	2	11.78469
	3	12.38168
	4	12.92188
9	1	13.13889
	2	13.0625
	3	12.14706
	4	11.04444
	5	11.67054
10	1	13.58333
	2	12.56522
	3	11.62983
	4	11.42857
	5	11.89101
11	1	12.18121
	2	12.94406
	3	11.54737
	4	10.69444
12	1	11.8967
	2	12.20261
	3	10.90244
13	1	12.25137
	2	11.57746
	3	11.16667
	4	10.875
14	1	10.84434
	2	12.0918
	3	13.2268
	4	13.05714
15	0	12.51002
	1	8.642857
	2	8.8125
	3	8.071429
16	0	11.97935
	1	11.27941
17	0	11.66534
	1	12.05779

18	0	11.95082
	1	11.20513
19	0	11.71856
	1	12.10476
20	0	11.71875
	1	11.95202
21	0	8.797101
	1	12.27586
22	0	11.02649
	1	12.17269
23	0	12.12927
	1	11.52301
24	1	10.63636
	2	10.86207
	3	11.59406
	4	12.34385
	5	11.63333
25	1	11.73333
	2	12.71028
	3	12.05976
	4	11.71348
	5	10.69118
26	1	10.72917
	2	12.66897
	3	12.15122
	4	11.97163
	5	10.87273
27	1	12.29933
	2	11.36364
	3	11.13953
	4	8.941176
	5	10.23529
28	1	12.36032
	2	12.26
	3	11.66667
	4	11.03448
	5	10.55556
29	1	12.47778
	2	12.19231
	3	11.83871
	4	12.30556
	5	11.46988
30	0	12.04098
	1	12.41667

	2	12.19091
	3	10.42857
	4	12.01075
	5	11.75
	6	12.12245
	7	13
	8	11.61905
	9	9.714286
	10	12.2381
	11	11.2
	12	10.08333
	13	14
	14	10.375
	15	11
	16	10.3
	18	12.33333
	21	11.5
	22	8
	24	9
	26	8
	30	16
	32	14
31	0	11
	4	4
	5	2.6
	6	8.222222
	7	7.090909
	8	8.190476
	9	9.784615
	10	10.68421
	11	11.50549
	12	12.53659
	13	13.29167
	14	14.57746
	15	15.48571
	16	16.45455
	17	17.3125
	18	18
	19	18
32	0	0
	5	5
	6	6.285714
	7	6.25
	8	8

	9	9.597222
	10	10.3253
	11	11.21359
	12	12.69767
	13	13.1875
	14	14.66667
	15	15.5
	16	16.2
	17	17.25
	18	17.71429
	19	19

Table 8

The output of value based average method is stored in the table 8. The first column keeps the attribute indexes, the second column keeps the distinct value of the corresponding attributes, and the third column keeps students' average final grade of the corresponding distinct value.

In the following part we showed bar plots of five best features to better identify the relationships between the key features and the target attribute.

Bar Plotting for attribute 32:

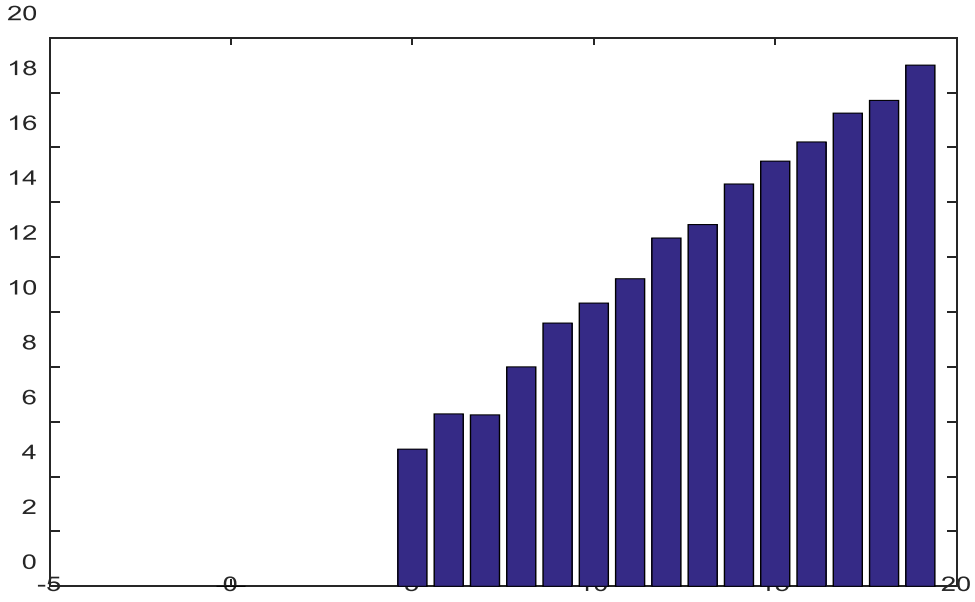


Fig. 1

In fig 1, we can see that the average marks increases gradually according to the second period grade. That is, the higher the second class grade is the higher probability of the student to perform better in the final.

Bar Plotting for attribute 31:

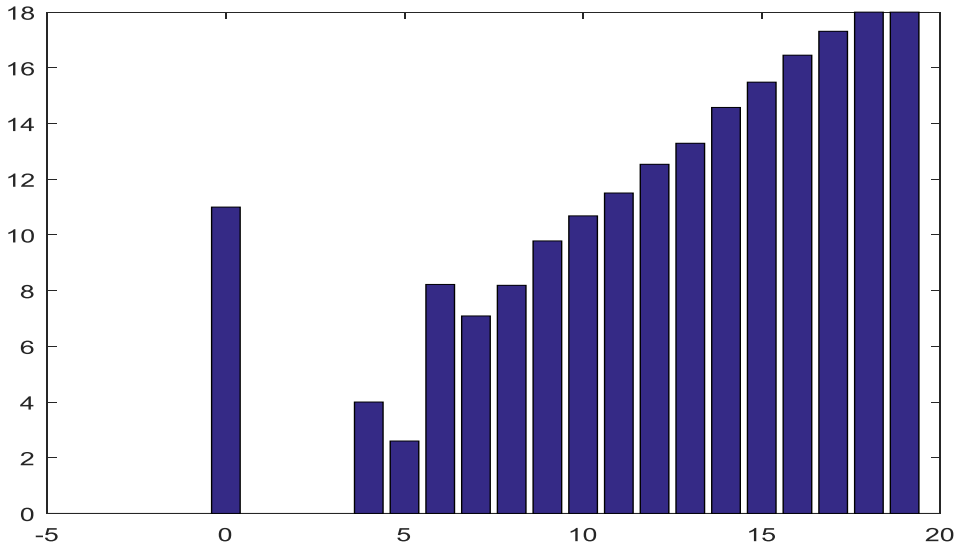


Fig. 2

The marks of first period grade also shows the same result as described above.

Bar Plotting for attribute 28:

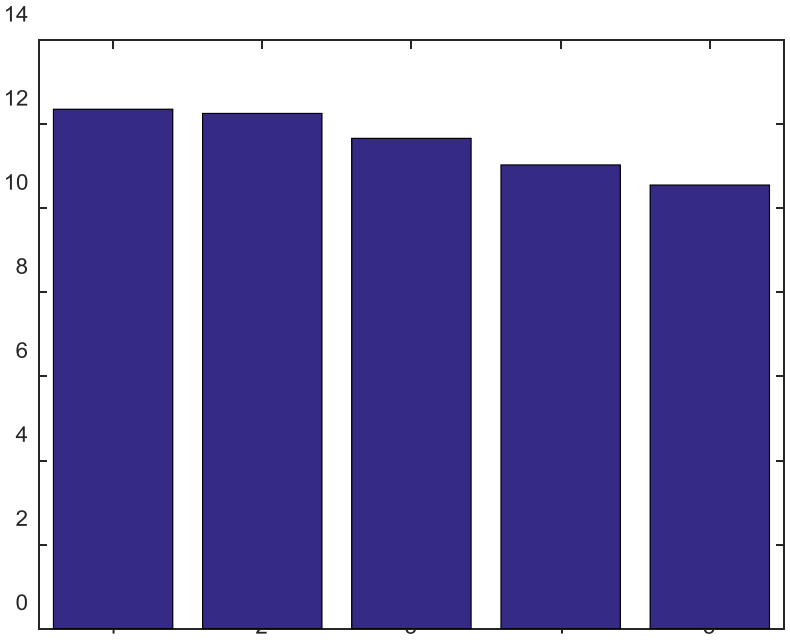


Fig. 3

This method shows that the student who takes less alcohol in the weekend has better performance in the final.

Bar Plotting for attribute 30:

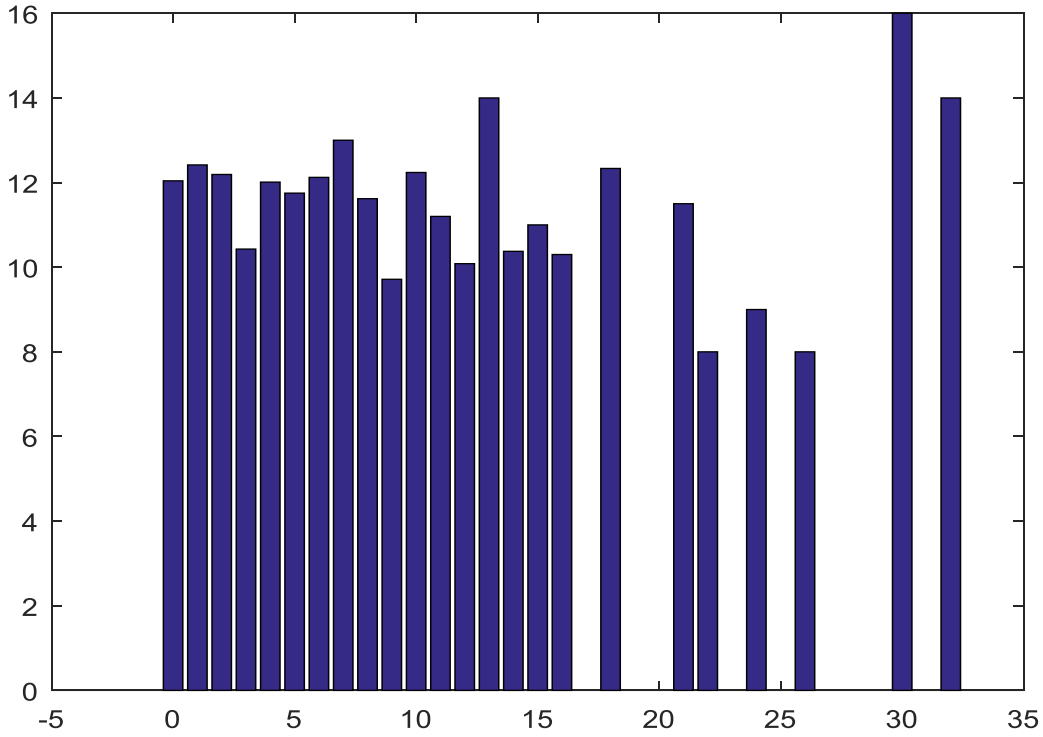


Fig. 4

Fig 4 shows a discrete result on absences. The highest average grade comes when the number of absences is 30. But no proper pattern can be noticed.

Bar Plotting for attribute 7:

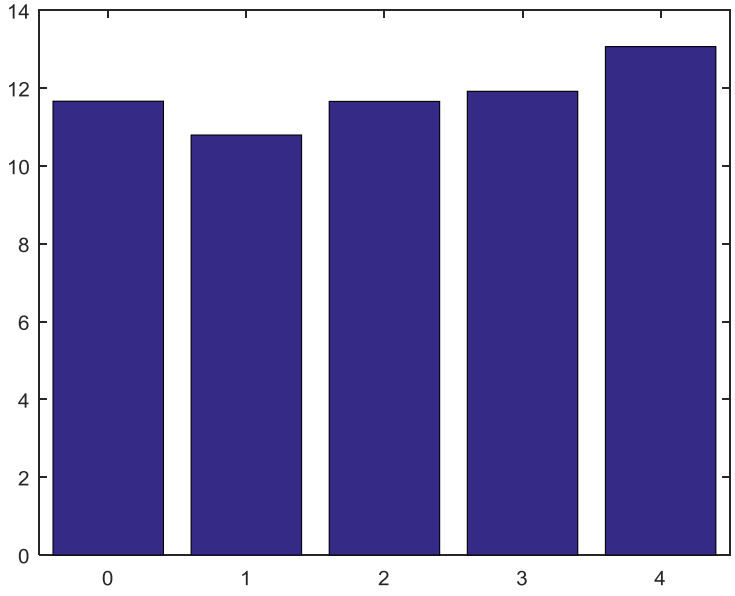


Fig. 5

Fig 5 shows that if mother is educated more, the students' performs better in the final.

4.2.2Value wise number of occurrences method

Result:

Attribute Index	Distinct values	Number of occurrences in values	Groups	Number of occurrences in groups	Percentage within values	Overall percentage
1	1	423	1	2	0.472813	0.308166
			2	30	7.092199	4.622496
			3	291	68.79433	44.83821
			4	100	23.64066	15.40832
	2	226	1	14	6.19469	2.157165
			2	54	23.89381	8.320493
			3	127	56.19469	19.56857
			4	31	13.71681	4.776579
2	1	266	1	9	3.383459	1.386749
			2	41	15.41353	6.317411
			3	173	65.03759	26.65639
			4	43	16.16541	6.625578
	2	383	1	7	1.827676	1.078582
			2	43	11.22715	6.625578
			3	245	63.96867	37.75039
			4	88	22.9765	13.55932
3	15	112	1	0	0	0
			2	12	10.71429	1.848998
			3	85	75.89286	13.09707
			4	15	13.39286	2.311248
	16	177	1	3	1.694915	0.46225
			2	19	10.73446	2.927581
			3	124	70.0565	19.10632
			4	31	17.51412	4.776579
	17	179	1	2	1.117318	0.308166
			2	26	14.52514	4.006163
			3	105	58.65922	16.17874
			4	46	25.69832	7.087827
	18	140	1	8	5.714286	1.232666
			2	19	13.57143	2.927581
			3	76	54.28571	11.71032
			4	37	26.42857	5.701079
	19	32	1	3	9.375	0.46225
			2	7	21.875	1.078582
			3	22	68.75	3.389831

			4	0	0	0
	20	6	1	0	0	0
			2	0	0	0
			3	4	66.66667	0.616333
			4	2	33.33333	0.308166
	21	2	1	0	0	0
			2	0	0	0
			3	2	100	0.308166
			4	0	0	0
	22	1	1	0	0	0
			2	1	100	0.154083
			3	0	0	0
			4	0	0	0
4	1	452	1	6	1.327434	0.924499
			2	50	11.06195	7.70416
			3	296	65.48673	45.60863
			4	100	22.12389	15.40832
	2	197	1	10	5.076142	1.540832
			2	34	17.25888	5.238829
			3	122	61.92893	18.79815
			4	31	15.73604	4.776579
5	1	192	1	2	1.041667	0.308166
			2	22	11.45833	3.389831
			3	125	65.10417	19.2604
			4	43	22.39583	6.625578
	2	457	1	14	3.063457	2.157165
			2	62	13.56674	9.553159
			3	293	64.11379	45.14638
			4	88	19.25602	13.55932
6	1	569	1	14	2.460457	2.157165
			2	74	13.00527	11.40216
			3	367	64.49912	56.54854
			4	114	20.03515	17.56549
	2	80	1	2	2.5	0.308166
			2	10	12.5	1.540832
			3	51	63.75	7.858243
			4	17	21.25	2.619414
7	0	6	1	0	0	0
			2	0	0	0
			3	5	83.33333	0.770416
			4	1	16.66667	0.154083
	1	143	1	5	3.496503	0.770416
			2	32	22.37762	4.930663
			3	93	65.03497	14.32974

			4	13	9.090909	2.003082
	2	186	1	5	2.688172	0.770416
			2	22	11.82796	3.389831
			3	129	69.35484	19.87673
			4	30	16.12903	4.622496
	3	139	1	2	1.438849	0.308166
			2	20	14.38849	3.081664
			3	87	62.58993	13.40524
			4	30	21.58273	4.622496
	4	175	1	4	2.285714	0.616333
			2	10	5.714286	1.540832
			3	104	59.42857	16.02465
			4	57	32.57143	8.782743
8	0	7	1	0	0	0
			2	0	0	0
			3	6	85.71429	0.924499
			4	1	14.28571	0.154083
	1	174	1	7	4.022989	1.078582
			2	38	21.83908	5.855162
			3	106	60.91954	16.33282
			4	23	13.21839	3.543914
	2	209	1	7	3.349282	1.078582
			2	24	11.48325	3.697997
			3	136	65.07177	20.95532
			4	42	20.09569	6.471495
	3	131	1	1	0.763359	0.154083
			2	9	6.870229	1.386749
			3	95	72.51908	14.6379
			4	26	19.84733	4.006163
	4	128	1	1	0.78125	0.154083
			2	13	10.15625	2.003082
			3	75	58.59375	11.55624
			4	39	30.46875	6.009245
9	1	72	1	2	2.777778	0.308166
			2	2	2.777778	0.308166
			3	46	63.88889	7.087827
			4	22	30.55556	3.389831
	2	48	1	0	0	0
			2	6	12.5	0.924499
			3	25	52.08333	3.85208
			4	17	35.41667	2.619414
	3	136	1	1	0.735294	0.154083
			2	18	13.23529	2.773498
			3	86	63.23529	13.25116

			4	31	22.79412	4.776579
	4	135	1	4	2.962963	0.616333
			2	25	18.51852	3.85208
			3	90	66.66667	13.86749
			4	16	11.85185	2.465331
	5	258	1	9	3.488372	1.386749
			2	33	12.7907	5.084746
			3	171	66.27907	26.34823
			4	45	17.44186	6.933744
10	1	36	1	1	2.777778	0.154083
			2	2	5.555556	0.308166
			3	19	52.77778	2.927581
			4	14	38.88889	2.157165
	2	23	1	0	0	0
			2	3	13.04348	0.46225
			3	14	60.86957	2.157165
			4	6	26.08696	0.924499
	3	181	1	6	3.314917	0.924499
			2	27	14.91713	4.160247
			3	114	62.98343	17.56549
			4	34	18.78453	5.238829
	4	42	1	1	2.380952	0.154083
			2	6	14.28571	0.924499
			3	29	69.04762	4.468413
			4	6	14.28571	0.924499
	5	367	1	8	2.179837	1.232666
			2	46	12.53406	7.087827
			3	242	65.94005	37.28814
			4	71	19.34605	10.93991
11	1	149	1	3	2.013423	0.46225
			2	14	9.395973	2.157165
			3	103	69.12752	15.87057
			4	29	19.46309	4.468413
	2	143	1	2	1.398601	0.308166
			2	8	5.594406	1.232666
			3	89	62.23776	13.71341
			4	44	30.76923	6.779661
	3	285	1	6	2.105263	0.924499
			2	49	17.19298	7.550077
			3	181	63.50877	27.88906
			4	49	17.19298	7.550077
	4	72	1	5	6.944444	0.770416
			2	13	18.05556	2.003082
			3	45	62.5	6.933744

			4	9	12.5	1.386749
12	1	455	1	11	2.417582	1.694915
			2	64	14.06593	9.861325
			3	285	62.63736	43.91371
			4	95	20.87912	14.6379
	2	153	1	4	2.614379	0.616333
			2	12	7.843137	1.848998
			3	105	68.62745	16.17874
			4	32	20.91503	4.930663
	3	41	1	1	2.439024	0.154083
			2	8	19.5122	1.232666
			3	28	68.29268	4.31433
			4	4	9.756098	0.616333
13	1	366	1	6	1.639344	0.924499
			2	45	12.29508	6.933744
			3	233	63.6612	35.90139
			4	82	22.40437	12.63482
	2	213	1	8	3.755869	1.232666
			2	26	12.20657	4.006163
			3	138	64.78873	21.26348
			4	41	19.24883	6.317411
	3	54	1	2	3.703704	0.308166
			2	10	18.51852	1.540832
			3	35	64.81481	5.392912
			4	7	12.96296	1.078582
	4	16	1	0	0	0
			2	3	18.75	0.46225
			3	12	75	1.848998
			4	1	6.25	0.154083
14	1	212	1	8	3.773585	1.232666
			2	42	19.81132	6.471495
			3	140	66.03774	21.57165
			4	22	10.37736	3.389831
	2	305	1	8	2.622951	1.232666
			2	33	10.81967	5.084746
			3	197	64.59016	30.35439
			4	67	21.96721	10.32357
	3	97	1	0	0	0
			2	7	7.216495	1.078582
			3	58	59.79381	8.936826
			4	32	32.98969	4.930663
	4	35	1	0	0	0
			2	2	5.714286	0.308166
			3	23	65.71429	3.543914

			4	10	28.57143	1.540832
15	0	549	1	6	1.092896	0.924499
			2	45	8.196721	6.933744
			3	369	67.21311	56.8567
			4	129	23.49727	19.87673
	1	70	1	8	11.42857	1.232666
			2	24	34.28571	3.697997
			3	37	52.85714	5.701079
			4	1	1.428571	0.154083
	2	16	1	1	6.25	0.154083
			2	7	43.75	1.078582
			3	7	43.75	1.078582
			4	1	6.25	0.154083
	3	14	1	1	7.142857	0.154083
			2	8	57.14286	1.232666
			3	5	35.71429	0.770416
			4	0	0	0
16	0	581	1	15	2.581756	2.311248
			2	77	13.25301	11.86441
			3	360	61.96213	55.46995
			4	129	22.2031	19.87673
	1	68	1	1	1.470588	0.154083
			2	7	10.29412	1.078582
			3	58	85.29412	8.936826
			4	2	2.941176	0.308166
17	0	251	1	11	4.38247	1.694915
			2	32	12.749	4.930663
			3	158	62.94821	24.34515
			4	50	19.92032	7.70416
	1	398	1	5	1.256281	0.770416
			2	52	13.06533	8.012327
			3	260	65.32663	40.06163
			4	81	20.35176	12.48074
18	0	610	1	15	2.459016	2.311248
			2	76	12.45902	11.71032
			3	392	64.2623	60.40062
			4	127	20.81967	19.56857
	1	39	1	1	2.564103	0.154083
			2	8	20.51282	1.232666
			3	26	66.66667	4.006163
			4	4	10.25641	0.616333
19	0	334	1	8	2.39521	1.232666
			2	49	14.67066	7.550077
			3	213	63.77246	32.81972

			4	64	19.16168	9.861325
	1	315	1	8	2.539683	1.232666
			2	35	11.11111	5.392912
			3	205	65.07937	31.58706
			4	67	21.26984	10.32357
20	0	128	1	2	1.5625	0.308166
			2	17	13.28125	2.619414
			3	88	68.75	13.55932
			4	21	16.40625	3.235747
	1	521	1	14	2.68714	2.157165
			2	67	12.85988	10.32357
			3	330	63.33973	50.84746
			4	110	21.11324	16.94915
21	0	69	1	5	7.246377	0.770416
			2	28	40.57971	4.31433
			3	36	52.17391	5.546995
			4	0	0	0
	1	580	1	11	1.896552	1.694915
			2	56	9.655172	8.628659
			3	382	65.86207	58.85978
			4	131	22.58621	20.1849
22	0	151	1	7	4.635762	1.078582
			2	25	16.55629	3.85208
			3	100	66.22517	15.40832
			4	19	12.58278	2.927581
	1	498	1	9	1.807229	1.386749
			2	59	11.84739	9.090909
			3	318	63.85542	48.99846
			4	112	22.48996	17.25732
23	0	410	1	6	1.463415	0.924499
			2	48	11.70732	7.395994
			3	274	66.82927	42.2188
			4	82	20	12.63482
	1	239	1	10	4.1841	1.540832
			2	36	15.06276	5.546995
			3	144	60.25105	22.18798
			4	49	20.50209	7.550077
24	1	22	1	1	4.545455	0.154083
			2	8	36.36364	1.232666
			3	9	40.90909	1.386749
			4	4	18.18182	0.616333
	2	29	1	2	6.896552	0.308166
			2	3	10.34483	0.46225
			3	21	72.41379	3.235747

			4	3	10.34483	0.46225
	3	101	1	2	1.980198	0.308166
			2	14	13.86139	2.157165
			3	72	71.28713	11.09399
			4	13	12.87129	2.003082
	4	317	1	4	1.26183	0.616333
			2	30	9.463722	4.622496
			3	205	64.66877	31.58706
			4	78	24.60568	12.01849
	5	180	1	7	3.888889	1.078582
			2	29	16.11111	4.468413
			3	111	61.66667	17.10324
			4	33	18.33333	5.084746
25	1	45	1	0	0	0
			2	7	15.55556	1.078582
			3	33	73.33333	5.084746
			4	5	11.11111	0.770416
	2	107	1	3	2.803738	0.46225
			2	9	8.411215	1.386749
			3	63	58.8785	9.707242
			4	32	29.90654	4.930663
	3	251	1	5	1.992032	0.770416
			2	28	11.15538	4.31433
			3	168	66.93227	25.88598
			4	50	19.92032	7.70416
	4	178	1	3	1.685393	0.46225
			2	26	14.60674	4.006163
			3	116	65.16854	17.87365
			4	33	18.53933	5.084746
	5	68	1	5	7.352941	0.770416
			2	14	20.58824	2.157165
			3	38	55.88235	5.855162
			4	11	16.17647	1.694915
26	1	48	1	4	8.333333	0.616333
			2	8	16.66667	1.232666
			3	34	70.83333	5.238829
			4	2	4.166667	0.308166
	2	145	1	2	1.37931	0.308166
			2	14	9.655172	2.157165
			3	86	59.31034	13.25116
			4	43	29.65517	6.625578
	3	205	1	3	1.463415	0.46225
			2	20	9.756098	3.081664
			3	139	67.80488	21.41757

			4	43	20.97561	6.625578
	4	141	1	1	0.70922	0.154083
			2	20	14.1844	3.081664
			3	94	66.66667	14.48382
			4	26	18.43972	4.006163
	5	110	1	6	5.454545	0.924499
			2	22	20	3.389831
			3	65	59.09091	10.01541
			4	17	15.45455	2.619414
27	1	451	1	8	1.773836	1.232666
			2	50	11.08647	7.70416
			3	286	63.41463	44.0678
			4	107	23.72506	16.4869
	2	121	1	4	3.305785	0.616333
			2	20	16.52893	3.081664
			3	78	64.46281	12.01849
			4	19	15.70248	2.927581
	3	43	1	0	0	0
			2	7	16.27907	1.078582
			3	33	76.74419	5.084746
			4	3	6.976744	0.46225
	4	17	1	4	23.52941	0.616333
			2	1	5.882353	0.154083
			3	12	70.58824	1.848998
			4	0	0	0
	5	17	1	0	0	0
			2	6	35.29412	0.924499
			3	9	52.94118	1.386749
			4	2	11.76471	0.308166
28	1	247	1	4	1.619433	0.616333
			2	24	9.716599	3.697997
			3	160	64.77733	24.65331
			4	59	23.88664	9.090909
	2	150	1	2	1.333333	0.308166
			2	19	12.66667	2.927581
			3	97	64.66667	14.94607
			4	32	21.33333	4.930663
	3	120	1	6	5	0.924499
			2	17	14.16667	2.619414
			3	72	60	11.09399
			4	25	20.83333	3.85208
	4	87	1	2	2.298851	0.308166
			2	14	16.09195	2.157165
			3	62	71.26437	9.553159

			4	9	10.34483	1.386749
	5	45	1	2	4.444444	0.308166
			2	10	22.22222	1.540832
			3	27	60	4.160247
			4	6	13.33333	0.924499
29	1	90	1	1	1.111111	0.154083
		0	2	16	17.77778	2.465331
		0	3	44	48.88889	6.779661
		0	4	29	32.22222	4.468413
	2	78	1	2	2.564103	0.308166
			2	7	8.974359	1.078582
			3	50	64.10256	7.70416
			4	19	24.35897	2.927581
	3	124	1	3	2.419355	0.46225
			2	15	12.09677	2.311248
			3	87	70.16129	13.40524
			4	19	15.32258	2.927581
	4	108	1	2	1.851852	0.308166
			2	10	9.259259	1.540832
			3	73	67.59259	11.24807
			4	23	21.2963	3.543914
	5	249	1	8	3.212851	1.232666
			2	36	14.45783	5.546995
			3	164	65.86345	25.26965
			4	41	16.46586	6.317411
30	0	244	1	16	6.557377	2.465331
			2	18	7.377049	2.773498
			3	148	60.65574	22.80431
			4	62	25.40984	9.553159
	1	12	1	0	0	0
			2	0	0	0
			3	11	91.66667	1.694915
			4	1	8.333333	0.154083
	2	110	1	0	0	0
			2	16	14.54545	2.465331
			3	73	66.36364	11.24807
			4	21	19.09091	3.235747
	3	7	1	0	0	0
			2	3	42.85714	0.46225
			3	3	42.85714	0.46225
			4	1	14.28571	0.154083
	4	93	1	0	0	0
			2	11	11.82796	1.694915
			3	65	69.89247	10.01541

			4	17	18.27957	2.619414
	5	12	1	0	0	0
			2	2	16.66667	0.308166
			3	8	66.66667	1.232666
			4	2	16.66667	0.308166
	6	49	1	0	0	0
			2	6	12.2449	0.924499
			3	35	71.42857	5.392912
			4	8	16.32653	1.232666
	7	3	1	0	0	0
			2	1	33.33333	0.154083
			3	0	0	0
			4	2	66.66667	0.308166
	8	42	1	0	0	0
			2	9	21.42857	1.386749
			3	24	57.14286	3.697997
			4	9	21.42857	1.386749
	9	7	1	0	0	0
			2	3	42.85714	0.46225
			3	3	42.85714	0.46225
			4	1	14.28571	0.154083
	10	21	1	0	0	0
			2	3	14.28571	0.46225
			3	13	61.90476	2.003082
			4	5	23.80952	0.770416
	11	5	1	0	0	0
			2	0	0	0
			3	5	100	0.770416
			4	0	0	0
	12	12	1	0	0	0
			2	4	33.33333	0.616333
			3	8	66.66667	1.232666
			4	0	0	0
	13	1	1	0	0	0
			2	0	0	0
			3	1	100	0.154083
			4	0	0	0
	14	8	1	0	0	0
			2	3	37.5	0.46225
			3	4	50	0.616333
			4	1	12.5	0.154083
	15	2	1	0	0	0
			2	0	0	0
			3	2	100	0.308166

			4	0	0	0
	16	10	1	0	0	0
			2	2	20	0.308166
			3	8	80	1.232666
			4	0	0	0
	18	3	1	0	0	0
			2	0	0	0
			3	3	100	0.46225
			4	0	0	0
	21	2	1	0	0	0
			2	0	0	0
			3	2	100	0.308166
			4	0	0	0
	22	2	1	0	0	0
			2	1	50	0.154083
			3	1	50	0.154083
			4	0	0	0
	24	1	1	0	0	0
			2	1	100	0.154083
			3	0	0	0
			4	0	0	0
	26	1	1	0	0	0
			2	1	100	0.154083
			3	0	0	0
			4	0	0	0
	30	1	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	1	100	0.154083
	32	1	1	0	0	0
			2	0	0	0
			3	1	100	0.154083
			4	0	0	0
31	0	1	1	0	0	0
			2	0	0	0
			3	1	100	0.154083
			4	0	0	0
	4	2	1	1	50	0.154083
			2	1	50	0.154083
			3	0	0	0
			4	0	0	0
	5	5	1	3	60	0.46225
			2	2	40	0.308166
			3	0	0	0

			4	0	0	0
	6	9	1	0	0	0
			2	8	88.88889	1.232666
			3	1	11.11111	0.154083
			4	0	0	0
	7	33	1	5	15.15152	0.770416
			2	22	66.66667	3.389831
			3	6	18.18182	0.924499
			4	0	0	0
	8	42	1	4	9.52381	0.616333
			2	24	57.14286	3.697997
			3	14	33.33333	2.157165
			4	0	0	0
	9	65	1	1	1.538462	0.154083
			2	17	26.15385	2.619414
			3	47	72.30769	7.241911
			4	0	0	0
	10	95	1	1	1.052632	0.154083
			2	9	9.473684	1.386749
			3	84	88.42105	12.94299
			4	1	1.052632	0.154083
	11	91	1	1	1.098901	0.154083
			2	1	1.098901	0.154083
			3	88	96.7033	13.55932
			4	1	1.098901	0.154083
	12	82	1	0	0	0
			2	0	0	0
			3	76	92.68293	11.71032
			4	6	7.317073	0.924499
	13	72	1	0	0	0
			2	0	0	0
			3	63	87.5	9.707242
			4	9	12.5	1.386749
	14	71	1	0	0	0
			2	0	0	0
			3	32	45.07042	4.930663
			4	39	54.92958	6.009245
	15	35	1	0	0	0
			2	0	0	0
			3	5	14.28571	0.770416
			4	30	85.71429	4.622496
	16	22	1	0	0	0
			2	0	0	0
			3	1	4.545455	0.154083

			4	21	95.45455	3.235747
	17	16	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	16	100	2.465331
	18	7	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	7	100	1.078582
	19	1	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	1	100	0.154083
32	0	7	1	7	100	1.078582
			2	0	0	0
			3	0	0	0
			4	0	0	0
	5	3	1	1	33.33333	0.154083
			2	2	66.66667	0.308166
			3	0	0	0
			4	0	0	0
	6	7	1	1	14.28571	0.154083
			2	6	85.71429	0.924499
			3	0	0	0
			4	0	0	0
	7	16	1	3	18.75	0.46225
			2	13	81.25	2.003082
			3	0	0	0
			4	0	0	0
	8	40	1	2	5	0.308166
			2	31	77.5	4.776579
			3	7	17.5	1.078582
			4	0	0	0
	9	72	1	1	1.388889	0.154083
			2	22	30.55556	3.389831
			3	49	68.05556	7.550077
			4	0	0	0
	10	83	1	1	1.204819	0.154083
			2	8	9.638554	1.232666
			3	73	87.95181	11.24807
			4	1	1.204819	0.154083
	11	103	1	0	0	0
			2	1	0.970874	0.154083
			3	102	99.02913	15.71649

			4	0	0	0
	12	86	1	0	0	0
			2	1	1.162791	0.154083
			3	83	96.51163	12.78891
			4	2	2.325581	0.308166
	13	80	1	0	0	0
			2	0	0	0
			3	79	98.75	12.17257
			4	1	1.25	0.154083
	14	54	1	0	0	0
			2	0	0	0
			3	24	44.44444	3.697997
			4	30	55.55556	4.622496
	15	38	1	0	0	0
			2	0	0	0
			3	1	2.631579	0.154083
			4	37	97.36842	5.701079
	16	25	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	25	100	3.85208
	17	20	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	20	100	3.081664
	18	14	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	14	100	2.157165
	19	1	1	0	0	0
			2	0	0	0
			3	0	0	0
			4	1	100	0.154083

Table 9

In table 9, the output of value wise number of occurrences is stored. The first column keeps the attribute indexes, the second column keeps the distinct values of each attribute, the third column keeps the number of occurrences for each distinct value of the attributes, the fourth column keeps the final grade group numbers, the fifth column keeps the number of occurrences in each group for the corresponding distinct value, the sixth column keeps the percentage of occurrences in each group within distinct values, and the seventh column keeps the overall percentage of occurrence.

In this table, key features are shaded. We can describe this in several ways.

Firstly, for a particular distinct value based on the number of occurrences we can predict which type of final score the student can get. For example, for the attribute index 7, when the value is 4 maximum percentage lies on the third group, 62.59. So we can say that if a student's 7^{th} feature value is 4 then there are higher chances for the student to score within 10 to 14.

Secondly, when the students do better in the final exam, i.e. scoring in the 4^{th} group. For example, if we notice the table for attribute 7, we get that there are the highest percentage of 4^{th} group occurrences among all the distinct values is 32.57% for the value of 4. So we can say that the value which leads the students to perform better in the finals is higher education of mother.

5. Conclusion

In this project, we have tried to describe the key features on basis of better final grade. The results show that the best five features related to the final grade is final grade, second period grade, weekend alcohol consumption, number of school absences, mother's education.

Our research shows that, students perform outstanding when they score more than 14 in the second and first period exam. The final grade comes well when the students' weekend alcohol consumption is less and mother has higher education. Although the number of absences comes as a key feature but our research results were unable to meet any pattern in this feature. Exploring more features may provide us more interesting facts about student performance.

6. Future Scope of Work

EDM is a very large field as a research subject. So, it always contains some non-discovered paths which can lead to a great innovation. We faced some problem during the development of this project which can be solved in further progress of this project. They are mentioned below:

- This data set size, both instances and attributes are not enough to describe or predict this type of complex data mining projects. For further improvement, we need a larger data set.
- Our recorded highest accuracy for predicting the target feature, final grade is 50.5% by linear SVM method. This must be increased a lot to make it work in real world.
- We can see that the value based average method does not show a large discrete in result between the distinct values in maximum variables. For this, value based average method cannot describe the features properly.
- Although the value wise number of occurrences describes some features really well, it needs further improvement to make it work better.

We are aware of the shortcomings and seek to improve those in future.

7. References

[1] P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7

[2] M. Ramaswami and R. Bhaskaran, "A Study on Feature Selection Techniques in Educational Data Mining", Journal Of Computing, Volume 1, Issue 1, December, 2009, ISSN: 2151-9617

[3] Behrouz Minaei-Bidgoli, Pang-Ning Tan, and William F. Punch, "Mining Interesting Contrast Rules for a Web-based Educational System"

[4] Cristóbal Romero, "Educational Data Mining: A Review of the State-of-the-Art, IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews"

[5] Liang Du, Yi-Dong Shen, Unsupervised Feature Selection with Adaptive Structure Learning, April, 2015

[6] https://archive.ics.uci.edu/ml/datasets/Student+Performance

[7] Robnik-Sikonja, M., and I. Kononenko. (1997). "An adaptation of Relief for attribute estimation in regression."

[8] https://in.mathworks.com

[9] https://it.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/56937/versions/28/previews/FSLib_v6.0_2018/methods/cfs.m/index.html

[10] https://it.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/56937/versions/28/previews/FSLib_v6.0_2018/methods/llcfs.m/index.html

[11] Gibbons, J.D. Nonparametric Statistical Inference. 2nd ed. M. Dekker, 1985

[12] Kendall, M.G. Rank Correlation Methods. Griffin, 1970

[13] Giorgio Roffo and Simone Melzi and Umberto Castellani and Alessandro Vinciarelli, “Infinite Latent Feature Selection: A Probabilistic Latent Graph-Based Ranking Approach”, October, 2017

[14] Roffo, Giorgio and Melzi, Simone, “Ranking to Learn”, 2017