For office use only
T1 _____
T2 _____
T3 _____
T4 _____

Team Control Number
44611

Problem Chosen
C

For office use only
F1 _____
F2 _____
F3 _____
F4 _____

2016
MCM/ICM
Summary Sheet

(Your team's summary should be included as the first page of your electronic submission.)

Type a summary of your results on this page. Do not include the name of your school, advisor, or team members on this page.

The paper bases on the data which the IPEDS (The Integrated Postsecondary Education Data System) provided to the public and the data from 7804 colleges and universities in the United States.

Our task is to help the Goodgrant Foundation to develop a model to determine an optimal investment strategy that identifies the schools, the investment amount per school, the return on that investment, and the time duration that the organization's money should be provided to have the highest likelihood of producing a strong positive effect on student performance.

The paper proposes one model, three algorithms and two formulas. The model set up under the machine learning technology including Support Vector Machines (SVM), Artificial Neural Network (ANN) and K-means algorithm. The two formulas are a way to calculate the weight is that we thought the better way to find the balance between the level of the salary and the better way to distribute the optimal investment strategy.

According to some data such as SAT scores, ACT scores and the salary level the graduates got, we made a formula to calculate the entire scores of the schools and aggregate them to 5 clusters which also means5 levels just like the QR Ranking Stars.

The rates of every cluster are based on a formula we created.

$$k = \frac{\log_e\left(\frac{score_{max} - score_{min}}{\pi^2}\right)}{7}$$

• $score_{max}$ and $score_{min}$ is the maximal score and the minimal score of the 5 cluster centers.

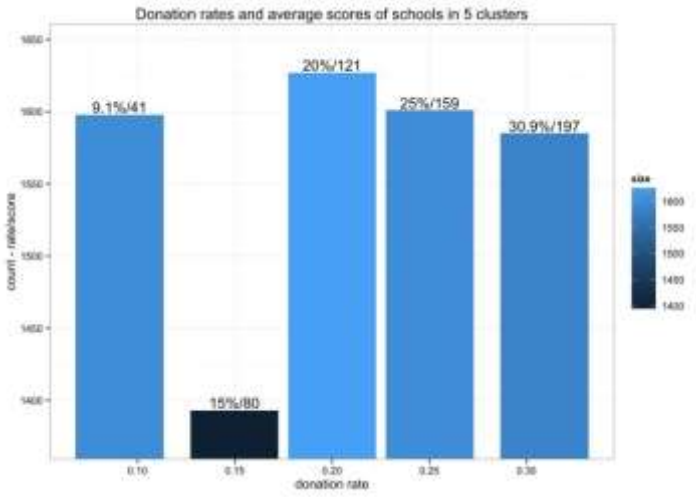
$$\Delta p_i = \frac{1}{5} \cdot k \cdot i \cdot \sin\left(\frac{\left(\frac{3-i}{2}\right)^2}{2}\right)$$

• i is the sorted index of the clusters, like 1, 2, 3.

$$r_i = \frac{1}{5} + \Delta p_i$$

• r_i is the i^{th} cluster's rate of investment.

The result of our model can be shown by the image as following.



Finally, the flow chart of algorithm in our mode, strengths and weaknesses and the improvement of model are given.

2016 MCM Problem C
The Goodgrant Challenge
Team # 44611

Abstract
We built a math model based on the data which the IPEDS (The Integrated Postsecondary Education Data System) provided to the public and including the data about the schools and the students. According to some data such as SAT scores, ACT scores and the salary level the graduates got, we made a formula to calculate the entire scores of the schools and aggregate them to 5 clusters which also means 5 levels just like the QR Ranking Stars. The first class is including 1184 schools such as Duke University, Rice University and Princeton University. This class can share 31% of the investment. There are 788 schools in the second class such as Blue Mountain College. The class can share 25% of the investment.

The way to calculate the weight is we thought the better way to find the balance between the level of the salary and the better way to distribute the investment.

Keywords: R, Machine Learning, SVM, ANN, K-Means, Optimal Investment Strategy

Introduction

Times are developing, the national competition becomes more and more reflected on competition for talent, as a fertile ground for training high-level personnel, colleges and universities to ensure that everyone enjoys equal right to learn will be key to guarantee qualities of personnel training. As a charity to help students complete their studies organization successfully, the Goodgrant Foundation hopes to donate one hundred million dollars annually since July 2016 for five years, for part of the students who did not receive funding.

Our team will provide the beat best investment decision model include selected investment school, the amount of investment in each school, the return on investment and time to access to these returns to Goodgrant Foundation base on data from The America national center for education statistics. In common sense, of course, the maximum returns on investment of college students is their growing up, which is the best embodiment of their the social recognition, and the form is their pay levels. In decision-making model, we will array out list of the best candidate school in accordance of the order of importance based on abilities each candidate school effective use of the fund's and expects to the return of funds.

Now R language is so popular around the world. As a new big data distributed processing framework, using spark will be extremely beneficial to dig-data processing and machine learning problems. We used SVM to classify the schools into Science and Art base on every school's percentages of degrees awarded of difference subjects. And we used the some entrance scores such as SAT scores and ACT scores, as the input values while we were using the average salaries of the graduates as the output values to using ANN for predicting the average salaries which were not provided. By the way, we used the range of the salaries to calculate the entire scores of the school but not the salary itself. This choice can make the model more flexible and avoid error from any single person. SAT scores, ACT scores, rate of the students receiving loan also play their role in the entire score calculating.

Basic Assumptions

According to data structure, data content and data value, which were showed in the topic, we combined the topic to analysis it and obtained the following assumption.

1. The NA's values set to be zero
2. Schools were only divided into two categories on the basis of the profession acceptance rate of each school, Art and Science. Because of the large differences in Professional type, employment situation between Art and Science, we divided all schools into these two categories in order to reduce errors

caused by mixed calculation. Which means that Art and Science will not be interfere with each other while calculating.

3. We found out some schools did not provide data for some reasons like some majors were not opened after reading and analyzing the data file carefully. So we regard these data as 0 to lower their weight and make sure the reasonability of results and the simple features of calculate which make it more smoothly while visualizing data.

4. Basic notations in the article are list as follow.

Symbol	Meaning
N, n	Number
w_i	Weight
x,y	Variables
d()	Distance
SVM	Support Vector Machines
ANN	Artificial Neural Network
K-Means	K-means algorithm

Model: School Investment Model

Our model set up under the machine learning technology including Support Vector Machines (SVM), Artificial Neural Network (ANN) and K-means algorithm. Machine learning, at its core, is concerned with algorithms that transform information into actionable intelligence.

To reach our goal, let's introduce such technologies as follow.

1. The Artificial Neural Network (ANN)

An ANN models the relationship between a set of input signals and an output signal using a model derived from our understanding of how a biological brain responds to stimuli from sensory inputs. Just as a brain uses a network of interconnected cells called neurons to create a massive parallel processor, the ANN uses a network of artificial neurons or nodes to solve learning problems.

The model of a single artificial neuron can be understood in terms very similar to the biological model. As depicted in the following figure (Fig.1), a directed network diagram defines a relationship between the input signals received by the dendrites(x variables) and the output signal (y variable). Just as with the biological neuron, each dendrite's signal is weighted (w values) according to its importance—ignore for now how these weights are determined. The input signals are summed by the cell body and the signal is passed on according to an activation function denoted by f.

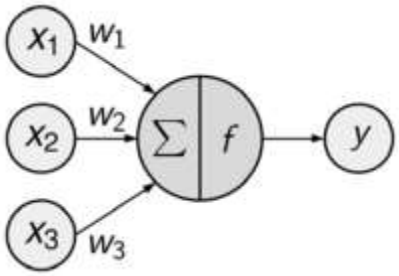


Fig.1The model of a single artificial neuron

A typical artificial neuron with n input dendrites can be represented by the formula that follows. The w weights allow each of the n inputs, (x), to contribute a greater or lesser amount to the sum of input

signals. The net total is used by the activation function f(x), and the resulting signal, y(x), is the output axon.

$$y(x) = f\left(\sum_{i=1}^{n} w_i x_i\right) \quad \text{(Eq.1)}$$

Although there are numerous variants of neural networks, each can be defined in terms of the following characteristics:

- An activation function
- A network topology
- The training algorithm

We also use a Network topology. As depicted here, a multilayer network adds one or more hidden layers that process the signals from the input nodes prior to reaching the output node in the following figure (Fig.2)

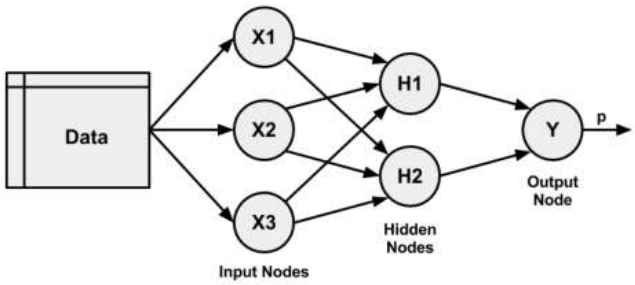


Fig.2 a Network topology

2. Support Vector Machines (SVM)

A Support Vector Machine (SVM) can be imagined as a surface that defines a boundary between various points of data which represent examples plotted in multidimensional space according to their feature values. The goal of an SVM is to create a flat boundary, called a hyper-plane, which leads to fairly homogeneous partitions of data on either side. The task of the SVM algorithm is to identify a line that separates the two classes.

The answer to that question involves a search for the Maximum Margin Hyper-plane that creates the greatest separation between the two classes. The support vectors (indicated by arrows in the Fig.3) are the points from each class that are the closest to the Maximum Margin Hyper-plane; each class must have at least one support vector, but it is possible to have more than one.

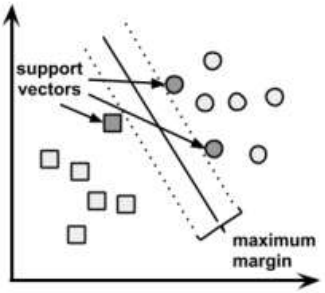


Fig.3 A Support Vector Machine (SVM)

It is easiest to understand how to find the maximum margin under the assumption that the classes are linearly separable. In this case, the Maximum Margin Hyper-plane is as far away as possible from the outer boundaries of the two groups of data points. These outer boundaries are known as the convex hull. The Maximum Margin Hyper-plane is then the perpendicular sector of the shortest line between the two convex hulls (Fig.4). Sophisticated computer algorithms that use a technique known as quadratic optimization are capable of finding the maximum margin in this way.

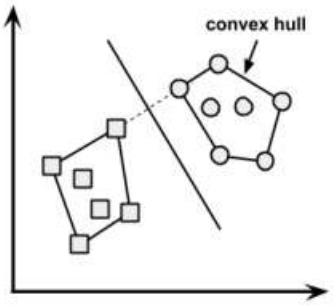


Fig.4 Convex Hulls

The SVMs can create a model highly complex relationship and its will be use in our model.

3. K-means algorithm

The k-means algorithm is perhaps the most often used clustering method. Having been studied for several decades, it serves as the foundation for many more sophisticated clustering techniques. If you understand the simple principles it uses, you will have the knowledge needed to understand nearly any clustering algorithm in use today. The k-means algorithm involves assigning each of the n examples to one of the k clusters, where k is a number that has been defined ahead of time. The goal is to minimize the differences within each cluster and maximize the differences between clusters.

Traditionally, k-means uses Euclidean distance, but Manhattan distance or Minkowski distance are also sometimes used. Recall that if n indicates the number of features, the formula for Euclidean distance between example x and example y is as follows (Eq.2):

$$\mathrm{dist}(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \qquad \text{(Eq.2)}$$

Using this distance function, we find the distance between each example and each cluster center. The figure (Fig.5) shows then assigned to the nearest cluster center.

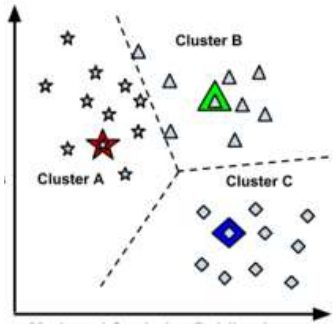


Fig.5 K-means algorithm

Our work is to help the Goodgrant Foundation to build a model to determine an optimal investment strategy that identifies the schools, the investment amount per school, the return on that investment, and the time duration that the organization's money should be provided to have the highest likelihood of producing a strong positive effect on student performance.

Let us start to build our model step by step as follows.

We run this program under R language and mainly use four program packages in R:

- e1071 (package for SVM)
- neuralnet (package for ANN)
- stats (package for K-means)
- ggplot2 (package for graphic)

Step one of the model

Based on the data which the IPEDS(The Integrated Postsecondary Education Data System) provided to the public and including the data about the schools and the students, we compile the Problem C - Most Recent Cohorts Data (Scorecard Elements).xlsx into a data.csv file which contains a total of 7804 records and 122 variables. We set schools as a database in R and replace NA's into 0. i.e. Read and clean the database in R. The results as in following figure (Fig.6)

```
> dim(schools)
[1] 7804  122
> str(schools)
'data.frame':   7804 obs. of  122 variables:
 $ UNITID                : int  100654 100663 100690 100706 100724 1007
 $ OPEID                 : int  100200 105200 2503400 105500 100500 105
 $ opeid6                : int  1002 1052 25034 1055 1005 1051 1007 100
 $ INSTNM                : Factor w/ 7634 levels "A & W Healthcare Educ
 $ CITY                  : Factor w/ 2545 levels "Aberdeen","Abilene",.
 $ STABBR                : Factor w/ 59 levels "AK","AL","AR",..: 2 2 2
 $ INSTURL               : Factor w/ 6237 levels "2ibs.org","abtu.edu",
 $ NPCURL                : Factor w/ 5429 levels "192.135.141.50/joomla
 $ HCM2                  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ PREDDEG               : int  3 3 3 3 3 3 2 3 3 3 ...
 $ CONTROL               : int  1 1 2 1 1 1 1 1 1 1 ...
 $ LOCALE                : num  12 12 12 12 12 13 32 31 12 13 ...
 $ HBCU                  : num  1 0 0 0 1 0 0 0 0 0 ...
 $ PBI                   : num  0 0 0 0 0 0 0 0 0 0 ...
```

Fig.6 Schools Database

Step two of the model

Due to the median earnings of students working and not enrolled 10 years after entry (MD_EARN_WNE_P10) is quite different, we divide7804 colleges and universities into two categories and name their as type.1.schools and type.2.schools. A figure (Fig.7) from both histogram and a figure (Fig.8) from both distributions show their difference.

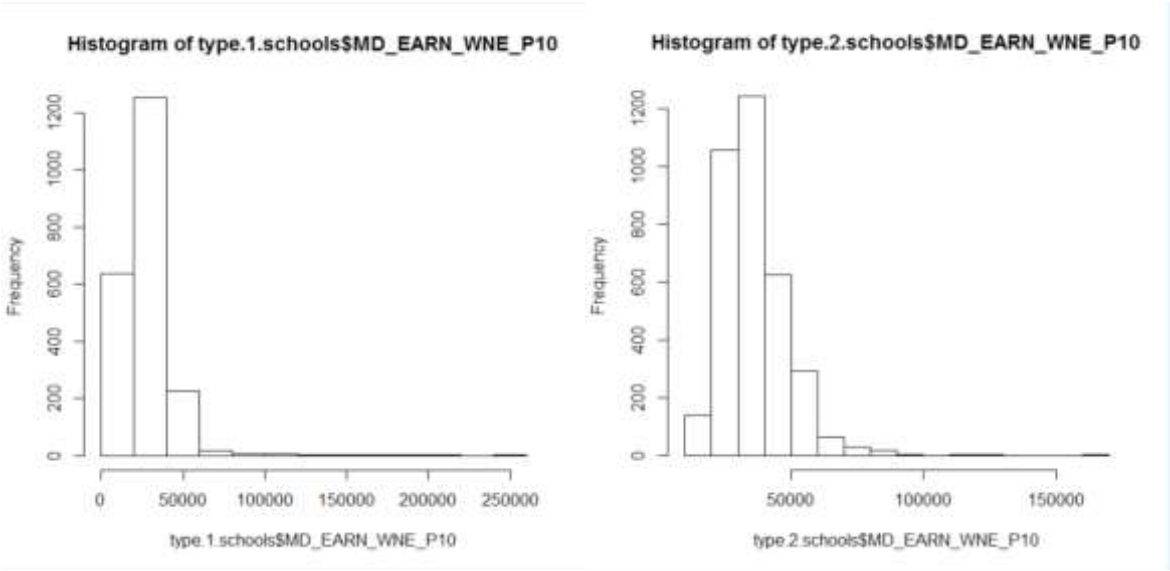


Fig.7 Histograms

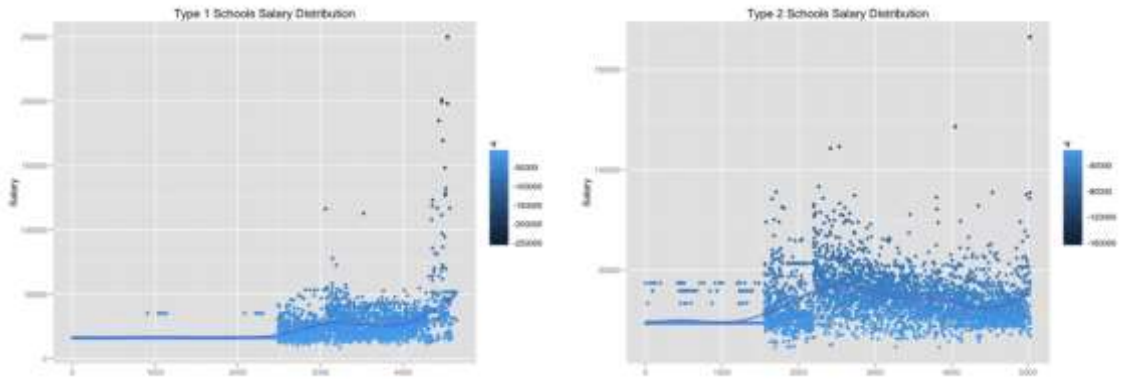


Fig.8 Distributions

Step three of the model

Using SVM to classify the schools with type of type.1.schools which includes 3483 colleges and universities, we set SAT scores and ACT scores as the input values and repayments as the output values. Training the ANN to calculate the repayment data that which the schools did not provide, we get one input node for each of the seven features, followed by two hidden nodes. Showed as follow figure (Fig.9)

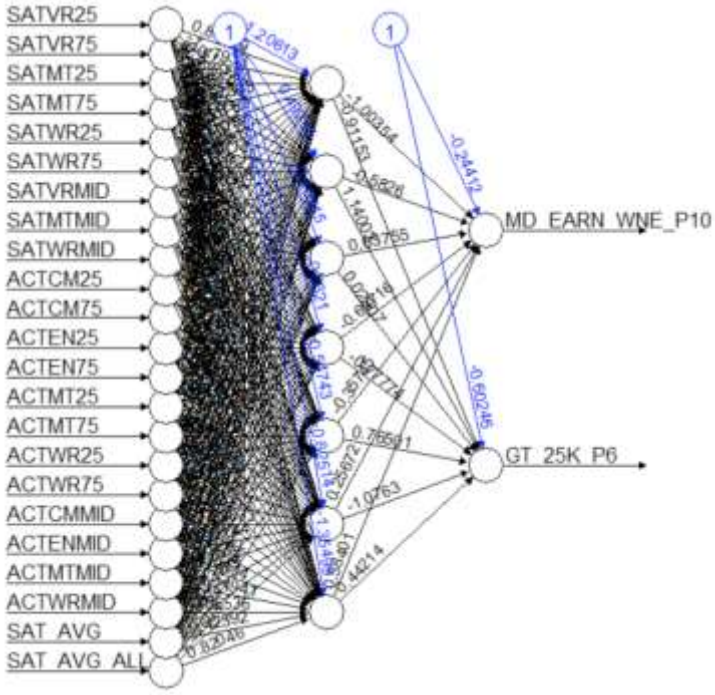


Fig.9 ANN of type.1.schools

For the type.2.schoolswhich includes 3483 colleges and universities, we have a similar ANN. (Fig.10)

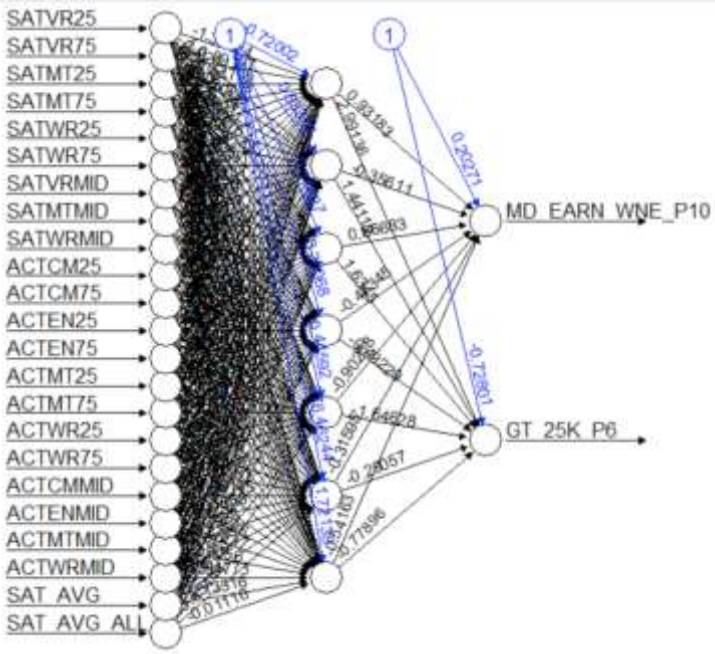


Fig.10 ANN of type.2.schools

Step four of the model

According to some data such as SAT scores, ACT scores and the salary level the graduates got, we made a formula to calculate the entire scores of the schools and aggregate them to 5 clusters which also means5 levels just like the QR Ranking Stars.

The rates of every cluster are base on a formula we created.

$$k = \frac{\log_e\left(\frac{score_{max} - score_{min}}{\pi^2}\right)}{7}$$

• $score_{max}$ and $score_{min}$ is the maximal score and the minimal score of the 5 cluster centers.

$$\Delta p_i = \frac{1}{5} \cdot k \cdot i \cdot \sin\left(\frac{\left(\frac{3-i}{2}\right)^2}{2}\right)$$

• i is the sorted index of the clusters, like 1, 2, 3.

$$r_i = \frac{1}{5} + \Delta p_i$$

• r_i is the i^{th} cluster's rate of investment.

Preparing for weights calculation and using K-Means to analyze the 7804 colleges and universities into 5 clusters, our final detail results as follows. (Fig.11)

ID OF THE CLASS	INVESTMENT RATE	# OF SCHOOLS	EXAMPLES
1	31%	1184	Duke University Rice University
2	25%	788	Blue Mountain College Lakes Region Community College
3	20%	152	Apex School of Theology
4	15%	478	Ecclesia College

5	9%	334	Selma University

Fig.11 The detail of the result of our work

Conclusion

There is an interesting feature in our model which could be describes as Mode is better, but not more. In the weight calculating formula, we analyzed the average salaries that the schools provided or we predicted to find the mode of salary and use the count of schools with the same salary to calculate the weights. The way to calculate the weight is we thought the better way to find the balance between the level of the salary and the better way to distribute the investment. The result of our model can be shown by table as following. (Tab.1)

5 classes	Support list in the top 5
1	Duke University
	Rice University
	Princeton University
	Pomona College
	Wellesley College
2	Blue Mountain College
	Lakes Region Community College
	Hobe Sound Bible College
	Wabash Valley College
	The University of Texas-Pan American
3	Apex School of Theology
	Mountwest Community and Technical College
	El Camino Community College District
	South Mountain Community College
	El Camino College-Compton Center
4	Ecclesia College
	Middle Georgia State College
	Mesabi Range Community and Technical College
	Heritage Christian University
	Chipola College
5	Selma University
	Central Georgia Technical College
	College America-Phoenix
	College America-Flagstaff
	Chicago ORT Technical Institute

Tab.1 Conclusion

Algorithm: The flow chart

The flow chart of algorithm in our model is listed as follow. (Fig.12)

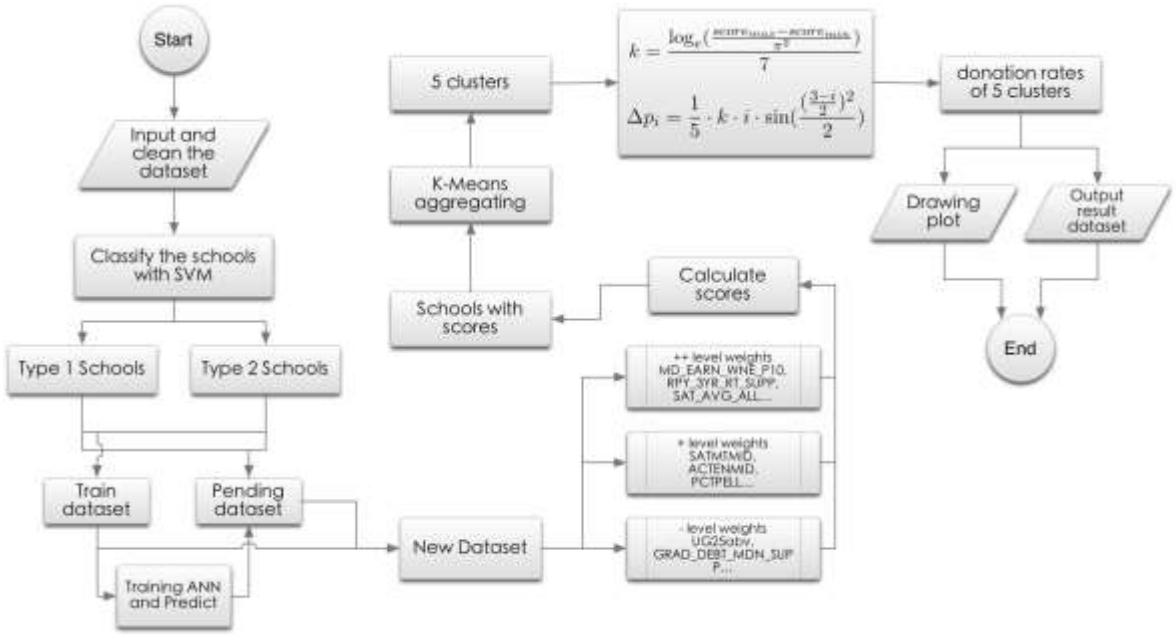


Fig.12 The flow chart of algorithm in our model

Strengths and Weaknesses

In this paper, we mainly use SVM，ANN，k-means to handle data and build a best investment decision model with weight. While using these ways to build up model and finding the best investment decision, we found they have both advantages and disadvantages in solving this problem as follow.

Advantage

1. Using SVM can simplified the classification of raw data. From the principle level, SVM is a novel learning method that has solid theoretical basis and requires only small amount of sample. It has differences with many other statistical methods for staying away from probability measures and Law of Large Numbers. In essence, it realize "transduction inference" from training sample to forecasting sample in a efficient way by avoiding the traditional process, from induction to deduction, which greatly simplified normal problem such as classification and regression.

2. Using SVM is benefit to solve multidimensional data the final decision function of SVM is only decided by a few support vectors, and calculation complexity depends on the number of support vector rather than the dimensions of the sample space. In a sense, these characteristics not only refrain from curse of dimension alit, but also reduce risks caused by overmuch index like schools and majors of this model

3. Using ANN makes it more efficient to find out optional solution We meet 4000-dimensional form while using subject data to synthetic Data. Finding out optional solution of a complex problem always needs large calculation, so using a feedback-type artificial neural network designed for a specific problem to give play to high-speed operation ability of computer can help us find it faster.

Disadvantage

1. The SVM code runs in a long time and SVM is hard to implementation large training sample.

2. ANN is not suitable for high precision computation. It will not have a good effect while large variable appear. It might be necessary to build up complex neural network structure if it is acquired for a reliable precision, which may cause long-time training.

Improvement

Due to our work base on the data in sight, we use the method of big data analysis, for its limited by time; we research it in a superficial level. If time allowed we think there are 3 places to be improved to make the model have better adaptability and more extensive using value.

1. Rewrite the code on Scala and run the model base on Apache Spark so that the model could be run with the real big data.
2. We will take a further study on the relations between the region where the schools located or rate of race and the development of the graduates.

Letter for the CFO

The Goodgrant Foundation
February 1st, 2016

Mr. Alpha Chiang:

Thank you for your attention on this letter. We are college students in China who also loves math and computer science too. We known that your foundation is looking forward to giving some investment to the universities in the U.S. while you want the investment can be more valuable and repayable. My team made some work to help you to solve this problem. So we are writing to you for describe our work, we really hope our work can let your investment more remarkable.

We built a math model based on the data which the IPEDS (The Integrated Postsecondary Education Data System) provided to the public and including the data about the schools and the students.

According to some data such as SAT scores, ACT scores and the salary level the graduates got, we made a formula to calculate the entire scores of the schools and aggregate them to 5 clusters which also means 5 levels just like the QR Ranking Stars.

The first class is including 1184 schools such as Duke University, Rice University and Princeton University. This class can share 31% of the investment.

There are 788 schools in the second class such as Blue Mountain College. The class can share 25% of the investment.

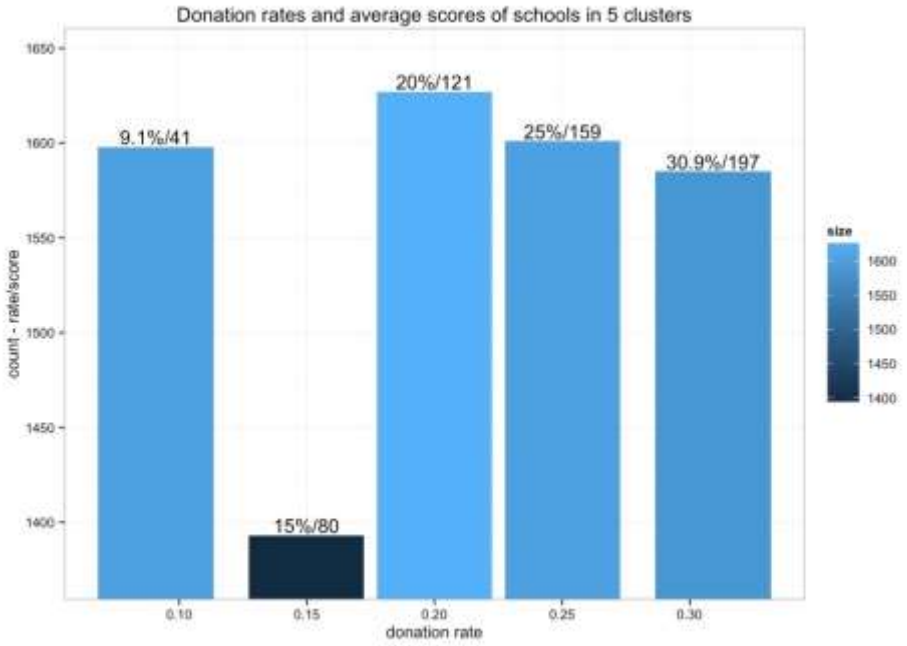
Here is a table show the detail of the result of our work.

ID OF THE CLASS	RATE	COUNT	EXAMPLES
1	31%	1184	Duke University Rice University
2	25%	788	Blue Mountain College Lakes Region Community College
3	20%	152	Apex School of Theology
4	15%	478	Ecclesia College
5	9%	334	Selma University

There is an interesting feature in our model which could be described as ***Mode is better, but not more***. In the weight calculating formula, we analyzed the average salaries that the schools provided or we predicted to find the mode (the most one) of salary and use the count of schools with the same salary to calculate the weights.

The way to calculate the weight is we thought the better way to find the balance between the level of the salary and the better way to distribute the investment.

The result of our model can be shown by the picture following.



The thing above is the result we worked out. We truly hope that you can take a further look on our model and hope it can help you to figure out more complex problems.

Yours sincerely,

Team # 44611

References

[1]Brett, Lantz. Mashine Learning with R[M]. Bermingham UK, 2013
[2]Richard, Ekman; uo, Lin, Lee&Jianwen, wu. Application of artificial neural network in food industry[J]. Guangdong China, 2003.
[3]Qi,Wang,Yufeng,Chen. Traffic incident detection based on artificial neural network [J]. Sichuan China,2006

Appendix and Supporting Data

R language program for our model as following:

```
library(e1071)
library(neuralnet)
library(stats)
library(ggplot2)

# Read and clean the database
setwd("/Users/wen/Desktop")
schools <-read.csv("data.csv", na.strings= c("NULL", "PrivacySuppressed"), header =TRUE)
potential.candidate.schools<- read.csv("potential_candidate_schools.csv", header =TRUE)

schools[is.na(schools)] <-0
names(schools)[121:122] <- c("MD_EARN_WNE_P10", "GT_25K_P6")

schools$MD_EARN_WNE_P10 <-as.numeric(as.character(schools$MD_EARN_WNE_P10))
schools$GT_25K_P6 <-as.numeric(as.character(schools$GT_25K_P6))

# Using SVM to classify the schools with type *Science* and *Art*
# according to the degrees awarded of different subjects
```

```
academics.table<- schools[46:83]
academics.svm.model<-svm(academics.table)
academics.svm.pred<- predict(academics.svm.model, academics.table)

type.1.schools<- schools[as.numeric(names(academics.svm.pred[academics.svm.pred])), ]
type.2.schools<- schools[as.numeric(names(academics.svm.pred[!academics.svm.pred])), ]

# Type 1 Schools
#  SAT scores and ACT scores as the input values and repayments as the output values
type.1.schools.train <- type.1.schools[!is.na(type.1.schools$MD_EARN_WNE_P10), ]
type.1.schools.train <- type.1.schools.train[!is.na(type.1.schools.train$GT_25K_P6),  ]
type.1.schools.train <- subset(type.1.schools.train, select =
c(SATVR25,SATVR75,SATMT25,SATMT75,SATWR25,SATWR75,SATVRMID,SATMTMID,SA
TWRMID,ACTCM25,ACTCM75,ACTEN25,ACTEN75,ACTMT25,ACTMT75,ACTWR25,ACTW
R75,ACTCMMID,ACTENMID,ACTMTMID,ACTWRMID,SAT_AVG,SAT_AVG_ALL,MD_EAR
N_WNE_P10,GT_25K_P6))
type.1.schools.train$MD_EARN_WNE_P10 <- type.1.schools.train$MD_EARN_WNE_P10  /1e5
attach(type.1.schools.train)
type.1.schools.train.matrix  <-
model.matrix(~MD_EARN_WNE_P10+GT_25K_P6+SATVR25,SATVR75,SATMT25,SATMT75,
SATWR25,SATWR75,SATVRMID,SATMTMID,SATWRMID,ACTCM25,ACTCM75,ACTEN25,
ACTEN75,ACTMT25,ACTMT75,ACTWR25,ACTWR75,ACTCMMID,ACTENMID,ACTMTMID,
ACTWRMID,SAT_AVG,SAT_AVG_ALL, data = type.1.schools.train)
#  Training the ANN to calculate the repayment data that which the schools did not provide
type.1.schools.net <-
neuralnet(MD_EARN_WNE_P10+GT_25K_P6~SATVR25+SATVR75+SATMT25+SATMT75+SA
TWR25+SATWR75+SATVRMID+SATMTMID+SATWRMID+ACTCM25+ACTCM75+ACTEN2
5+ACTEN75+ACTMT25+ACTMT75+ACTWR25+ACTWR75+ACTCMMID+ACTENMID+ACT
MTMID+ACTWRMID+SAT_AVG+SAT_AVG_ALL, type.1.schools.train.matrix,
                    hidden =7, rep =10, linear.output=FALSE, algorithm ="rprop+")

#  Merge the database
type.1.schools.compute <-rbind(type.1.schools[type.1.schools$MD_EARN_WNE_P10 ==0, ],
type.1.schools[type.1.schools$GT_25K_P6 ==0, ])
UNITIDs <-subset(type.1.schools.compute, select = c(UNITID))
type.1.schools.compute <- subset(type.1.schools.compute, select =
c(SATVR25,SATVR75,SATMT25,SATMT75,SATWR25,SATWR75,SATVRMID,SATMTMID,SA
TWRMID,ACTCM25,ACTCM75,ACTEN25,ACTEN75,ACTMT25,ACTMT75,ACTWR25,ACTW
R75,ACTCMMID,ACTENMID,ACTMTMID,ACTWRMID,SAT_AVG,SAT_AVG_ALL))
type.1.schools.compute[is.na(type.1.schools.compute)]  <-0

type.1.schools.compute.res <-as.data.frame(compute(type.1.schools.net,
type.1.schools.compute)$net.result)
names(type.1.schools.compute.res) <-c("MD_EARN_WNE_P10", "GT_25K_P6")

type.1.schools.compute[type.1.schools.compute  ==0] <-NA

type.1.schools.compute.res <-cbind(UNITIDs, type.1.schools.compute, type.1.schools.compute.res)
type.1.schools.compute.res$MD_EARN_WNE_P10 <-
type.1.schools.compute.res$MD_EARN_WNE_P10 *1e5
type.1.schools.compute.res$MD_EARN_WNE_P10 <-
round(type.1.schools.compute.res$MD_EARN_WNE_P10)

# Type 2 Schools
#  The same processing way as type 1
type.2.schools.train <- type.2.schools[!is.na(type.2.schools$MD_EARN_WNE_P10), ]
type.2.schools.train <- type.2.schools.train[!is.na(type.2.schools.train$GT_25K_P6),  ]
type.2.schools.train <- subset(type.2.schools.train, select =
c(SATVR25,SATVR75,SATMT25,SATMT75,SATWR25,SATWR75,SATVRMID,SATMTMID,SA
```

```
TWRMID,ACTCM25,ACTCM75,ACTEN25,ACTEN75,ACTMT25,ACTMT75,ACTWR25,ACTW
R75,ACTCMMID,ACTENMID,ACTMTMID,ACTWRMID,SAT_AVG,SAT_AVG_ALL,MD_EAR
N_WNE_P10,GT_25K_P6))
type.2.schools.train$MD_EARN_WNE_P10 <- type.2.schools.train$MD_EARN_WNE_P10 /1e5
attach(type.2.schools.train)
type.2.schools.train.matrix <-
model.matrix(~MD_EARN_WNE_P10+GT_25K_P6+SATVR25,SATVR75,SATMT25,SATMT75,
SATWR25,SATWR75,SATVRMID,SATMTMID,SATWRMID,ACTCM25,ACTCM75,ACTEN25,
ACTEN75,ACTMT25,ACTMT75,ACTWR25,ACTWR75,ACTCMMID,ACTENMID,ACTMTMID,
ACTWRMID,SAT_AVG,SAT_AVG_ALL, data = type.2.schools.train)
type.2.schools.net <-
neuralnet(MD_EARN_WNE_P10+GT_25K_P6~SATVR25+SATVR75+SATMT25+SATMT75+SA
TWR25+SATWR75+SATVRMID+SATMTMID+SATWRMID+ACTCM25+ACTCM75+ACTEN2
5+ACTEN75+ACTMT25+ACTMT75+ACTWR25+ACTWR75+ACTCMMID+ACTENMID+ACT
MTMID+ACTWRMID+SAT_AVG+SAT_AVG_ALL, type.2.schools.train.matrix,
                    hidden =7, rep =10, linear.output=FALSE, algorithm ="rprop+")

type.2.schools.compute <-rbind(type.2.schools[type.2.schools$MD_EARN_WNE_P10 ==0, ],
type.2.schools[type.2.schools$GT_25K_P6 ==0, ])
UNITIDs <-subset(type.2.schools.compute, select = c(UNITID))
type.2.schools.compute <- subset(type.2.schools.compute, select =
c(SATVR25,SATVR75,SATMT25,SATMT75,SATWR25,SATWR75,SATVRMID,SATMTMID,SA
TWRMID,ACTCM25,ACTCM75,ACTEN25,ACTEN75,ACTMT25,ACTMT75,ACTWR25,ACTW
R75,ACTCMMID,ACTENMID,ACTMTMID,ACTWRMID,SAT_AVG,SAT_AVG_ALL))
type.2.schools.compute[is.na(type.2.schools.compute)] <-0

type.2.schools.compute.res <-as.data.frame(compute(type.2.schools.net,
type.2.schools.compute)$net.result)
names(type.2.schools.compute.res) <-c("MD_EARN_WNE_P10", "GT_25K_P6")

type.2.schools.compute[type.2.schools.compute ==0] <-NA

type.2.schools.compute.res <-cbind(UNITIDs, type.2.schools.compute, type.2.schools.compute.res)
type.2.schools.compute.res$MD_EARN_WNE_P10 <-
type.2.schools.compute.res$MD_EARN_WNE_P10 *1e5
type.2.schools.compute.res$MD_EARN_WNE_P10 <-
round(type.2.schools.compute.res$MD_EARN_WNE_P10)

# Combining Type 1 and Type 2
for (iin1:length(type.1.schools.compute.res$UNITID)) {
  id <-as.integer(type.1.schools.compute.res$UNITID[i])
schools[schools$UNITID== id, ]$MD_EARN_WNE_P10 <-
type.1.schools.compute.res[i, ]$MD_EARN_WNE_P10
schools[schools$UNITID== id, ]$GT_25K_P6 <- type.1.schools.compute.res[i, ]$GT_25K_P6
}
for (iin1:length(type.2.schools.compute.res$UNITID)) {
  id <-as.integer(type.2.schools.compute.res$UNITID[i])
schools[schools$UNITID== id, ]$MD_EARN_WNE_P10 <-
type.2.schools.compute.res[i, ]$MD_EARN_WNE_P10
schools[schools$UNITID== id, ]$GT_25K_P6 <- type.2.schools.compute.res[i, ]$GT_25K_P6
}

# Preparing for weights calculation
expections.salaries<- schools$MD_EARN_WNE_P10
expections.salaries.counts<-as.data.frame(table(expections.salaries))
expections.salaries.counts<-expections.salaries.counts[order(-expections.salaries.counts$Freq), ]

total.count<- sum(expections.salaries.counts$Freq)
total.SAT_AVG_ALL<- sum(schools$SAT_AVG_ALL)
```

```
total.ACTCMMID<- sum(schools$ACTCMMID)
total.SATVRMID<- sum(schools$SATVRMID)
total.SATMTMID<- sum(schools$SATMTMID)
total.SATWRMID<- sum(schools$SATWRMID)
total.ACTENMID<- sum(schools$ACTENMID)
total.ACTMTMID<- sum(schools$ACTMTMID)
total.ACTWRMID<- sum(schools$ACTWRMID)
total.PCTPELL<- sum(schools$PCTPELL)
total.PCTFLOAN<- sum(schools$PCTFLOAN)

schools.fields<-  names(schools)

weights.data.frame<-as.data.frame(t(apply(schools, 1, function(school.row) {
# MD_EARN_WNE_P10: Median earnings of students working and not enrolled 10 years after entry
  MD_EARN_WNE_P10.weight <-
expections.salaries.counts[expections.salaries.counts$expections.salaries==as.numeric(as.character(sc
hool.row[match("MD_EARN_WNE_P10", schools.fields)])),  ]$Freq/total.count*200

# RPY_3YR_RT_SUPP: 3-year repayment rate
  RPY_3YR_RT_SUPP.weight <-as.numeric(as.character(school.row[match("RPY_3YR_RT_SUPP",
schools.fields)]))  *200

# SAT_AVG_ALL: Average SAT equivalent score of students admitted for all campuses rolled up to
the 6-digit OPE ID
SAT_AVG_ALL.weight<-log(as.numeric(as.character(school.row[match("SAT_AVG_ALL",
schools.fields)]))  /total.SAT_AVG_ALL*1e4) /1e2
if (SAT_AVG_ALL.weight<0) SAT_AVG_ALL.weight<-0
SAT_AVG_ALL.weight<-SAT_AVG_ALL.weight*200

# SATVRMID: Midpoint of SAT scores at the institution
SATVRMID.weight<-log(as.numeric(as.character(school.row[match("SAT_AVG_ALL",
schools.fields)]))  /total.SATVRMID*1e4) /1e2
if (SATVRMID.weight<0) SATVRMID.weight<-0
SATVRMID.weight<-SATVRMID.weight*200

# ACTCMMID: Midpoint of the ACT cumulative score
ACTCMMID.weight<-log(as.numeric(as.character(school.row[match("ACTCMMID",
schools.fields)]))  /total.ACTCMMID*1e4) /1e2
if (ACTCMMID.weight<0) ACTCMMID.weight<-0
ACTCMMID.weight<-ACTCMMID.weight*200

# SATMTMID: Midpoint of SAT scores at the institution (math)
SATMTMID.weight<-log(as.numeric(as.character(school.row[match("SATMTMID",
schools.fields)]))  /total.SATMTMID*1e4) /1e2
if (SATMTMID.weight<0) SATMTMID.weight<-0
SATMTMID.weight<-SATMTMID.weight*100

# SATWRMID: Midpoint of SAT scores at the institution (writing)
SATWRMID.weight<-log(as.numeric(as.character(school.row[match("SATWRMID",
schools.fields)]))  /total.SATWRMID*1e4) /1e2
if (SATWRMID.weight<0) SATWRMID.weight<-0
SATWRMID.weight<-SATWRMID.weight*100

# ACTENMID: Midpoint of the ACT English score
ACTENMID.weight<-log(as.numeric(as.character(school.row[match("ACTENMID",
schools.fields)]))  /total.ACTENMID*1e4) /1e2
if (ACTENMID.weight<0) ACTENMID.weight<-0
ACTENMID.weight<-ACTENMID.weight*100
```

```
# ACTMTMID: Midpoint of the ACT math score
ACTMTMID.weight<-log(as.numeric(as.character(school.row[match("ACTMTMID",
schools.fields)])) /total.ACTMTMID*1e4) /1e2
if (ACTMTMID.weight<0) ACTMTMID.weight<-0
ACTMTMID.weight<-ACTMTMID.weight*100

# ACTWRMID: Midpoint of the ACT writing score
ACTWRMID.weight<-log(as.numeric(as.character(school.row[match("ACTWRMID",
schools.fields)])) /total.ACTWRMID*1e4) /1e2
if (ACTWRMID.weight<0) ACTWRMID.weight<-0
ACTWRMID.weight<-ACTWRMID.weight*100

# PCTPELL: Percentage of undergraduates who receive a Pell Grant
PCTPELL.weight<-log(as.numeric(as.character(school.row[match("PCTPELL", schools.fields)]))
/total.PCTPELL*1e4) /1e2
if (PCTPELL.weight<0) PCTPELL.weight<-0
PCTPELL.weight<-PCTPELL.weight*100

# PCTFLOAN: Percentage of undergraduates who receive a Pell Grant
PCTFLOAN.weight<-log(as.numeric(as.character(school.row[match("PCTFLOAN", schools.fields)]))
/total.PCTFLOAN*1e4) /1e2
if (PCTFLOAN.weight<0) PCTFLOAN.weight<-0
PCTFLOAN.weight<-PCTFLOAN.weight*100

# UG25abv: Percentage of undergraduates aged 25 and above - Negative Weight
 UG25abv.weight <- (1-as.numeric(as.character(school.row[match("UG25abv", schools.fields)])))
/10*100

# GRAD_DEBT_MDN_SUPP: Median debt of completers
GRAD_DEBT_MDN_SUPP.weight<- (10-
log(as.numeric(as.character(school.row[match("GRAD_DEBT_MDN_SUPP", schools.fields)]))))
/5*100
if (is.infinite(GRAD_DEBT_MDN_SUPP.weight)) GRAD_DEBT_MDN_SUPP.weight<-0

# GRAD_DEBT_MDN10YR_SUPP: Median debt of completers expressed in 10-year monthly
payments
 GRAD_DEBT_MDN10YR_SUPP.weight <- ((-
log(as.numeric(as.character(school.row[match("GRAD_DEBT_MDN10YR_SUPP", schools.fields)]))
+1/1e6)) /1e2) *100

 weights <-c(MD_EARN_WNE_P10.weight, RPY_3YR_RT_SUPP.weight,
SAT_AVG_ALL.weight, SATVRMID.weight, ACTCMMID.weight,
SATMTMID.weight, SATWRMID.weight, ACTENMID.weight, ACTMTMID.weight,
ACTWRMID.weight, PCTPELL.weight, PCTFLOAN.weight,
          UG25abv.weight, GRAD_DEBT_MDN_SUPP.weight,
GRAD_DEBT_MDN10YR_SUPP.weight)

return(c(as.numeric(as.character(school.row[match("UNITID", schools.fields)])), weights,
sum(weights)))
}))

# Using K-Means to analyze the schools into 5 clusters
schools.cl <-kmeans(weights.data.frame[, 2:17], 5)
schools.clusters<-data.frame(id=1:5, schools.cl$center, size=schools.cl$size, rate =1/5)
schools.clusters<-schools.clusters[order(-schools.clusters$V17), ]
schools.clusters<-cbind(schools.clusters, order=1:5)

# Calculating the rate offsets
k <- log((max(schools.clusters$V17) - min(schools.clusters$V17)) / pi^2) /5
```

```
schools.clusters$rate<-schools.clusters$rate+ (1/5) * sin((3-schools.clusters$order)^2/0.5)  * k * ((3-
schools.clusters$order) /2)

weights.data.frame<-cbind(weights.data.frame, cluster =schools.cl$cluster)
potential.candidate.schools.weight<-weights.data.frame[weights.data.frame$V1
%in%potential.candidate.schools$UNITID, ]

school.ids<-apply(schools.clusters,  1, function(r)
potential.candidate.schools.weight[potential.candidate.schools.weight$cluster==as.numeric(r[1]),  ]$U
NITID)

# Drawing plot
ggplot(schools.clusters,  aes(x=rate, y=size, label=paste(round(rate*1000)  /10, "%/", round(V17),
sep=""), fill=size))  +
geom_bar(stat ="identity")  +
geom_text(vjust=-0.2, nudge_y=0.5) +
scale_y_continuous(limits=c(min(schools.clusters$size)  -20, max(schools.clusters$size)
+20),oob=rescale_none) +
xlab("donation  rate") +
ylab("count  - rate/score") +
ggtitle("Donation  rates and average scores of schools in 5 clusters") +
theme_bw() +
ggsave("problem_c_output.pdf",  width =10, height =7)

# Output the result data
df<-as.data.frame(t(apply(schools.clusters, 1, function(cluster) {
return(c(cluster[1], cluster[19],
length(potential.candidate.schools.weight[potential.candidate.schools.weight$cluster==as.numeric(clu
ster[1]),]$V1),
paste(potential.candidate.schools.weight[potential.candidate.schools.weight$cluster==as.numeric(clus
ter[1]), ]$V1, collapse="|")))
})))
names(df) <-c("id",  "rate",  "size",  "UNITIDs")
write.csv(df, "problem_c_output.csv",  row.names=FALSE, quote =FALSE)
```