The optimal investment strategy for Goodgrants foundation based on analytic hierarchy process

Contents

1 Introduction

1.1 Background

Post-secondary education attainment has been playing a more and more important role in effecting not only the earning power of individuals but also the quality of their lifetime nowadays. An investigation shows that a worker with a bachelor's degree earns 84% more than a worker without a degree - an average value of $2.8 million over the course of a lifetime[1].

Among the various financial sources of schools, donations from individuals and foundations have been approved to be much helpful to improve education quality and student performance. Famous charitable foundation such as the Gates foundation and the Lumina foundation provide projects covering many aspects in post-secondary education, with policies varying from one state to another.

However, with limited funding and limitless demanding, the investing plan should be as optimal as possible to get greatest returns on investment. In this way, the investment objects and the amount of economic support should be determined according to their capability of effectively using funding. And the time duration should be appropriate to get the strongest effect while the objects all get enough support. ROI also must be defined for foundations to make effect analyzing. Mathematical models can help in this procedure which would make sure every factor be well considered.

1.2 Planned approach

Since the information of candidate schools has been given including so many aspects such as costs, students' scores and graduates' financial situation, and the objects and amount should be determined. Analytic Hierarchy Process seems to conform to the demands, which can decompose a complex problem into a multilevel hierarchic structure of objectives, criteria, sub criteria and alternatives and provides a fundamental scale of relative magnitudes expressed in dominance units to represent judgments in the form of pairwise comparisons[13].

In this model, the criteria contains the situation of school, performance of undergraduates and graduates. Factors given in database that can reflect these three aspects are picked up and set to be sub criteria. The AHP model then help ranking the schools to get more effect with limited investment.

To establish this complicated model, several points should be achieved.

1. Given the related data of schools, return on investment is defined with several factors that can directly or indirectly show the effect of donation.

2. These 12 factors are divided into three aspects and the relative weights and performance scores are evaluated and compared with each other.

3. Through the AHP model, candidate schools are ranked by ROI expectation. Schools that will get donated and the amount of investment are determined with respect to reality. Then the initial invest plan comes out.

4. ROI is estimated according to the initial plan and the ROI definition, the time duration of each donation is revised to get a better ROI.

5. Sensitivity analysis is executed to get information about this model's response to data fluctuation. And the stability of this model is measured.

6. Strengths and weaknesses of our model are identified. And we get more to do in the future after further discussion.

7. By overviewing our approach, we get the conclusion of our work. Finally, we give a letter for the CFO of the Goodgrants Foundation.

2 General assumptions

With the data of candidate schools in America, we make the following assumptions to complete the model. And these assumptions will also be applied our report.

1. The data of 2015 can represent the average situation of schools and students.

2. Assume that the schools get donation will use all the investment into education and other aspects to improve students' performance.

3. The differences of different states that may actually influent schools and students are not taken into consideration.

The assumptions above are made to simplify the problem and fit it into the model.

3 Data cleaning

Data Cleaning is an essential part of data analysis and data mining. It plays an important role in Data science. For this problem, large amounts of data with imappropriate format make it hard to analysis and calculate in computer processing.

Data cleansing differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at entry time, rather than on batches of data[12].

Some data cleansing solutions will clean data by cross checking with a validated data set. Also data enhancement, where data is made more complete by adding related information, is a common data cleansing practice. For example, appending addresses with phone numbers related to that address[9].

What should be done first is to clean the whole data file. Data cleansing differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at entry time, rather than on batches of data.

3.1 Single imputation with linear regression

The data file contains too much 'NULL' and 'PrivacySuppressed', it's essential to choose a suitble measure to correct this data file or the missing data may draw tolerance into the result of analysis.

In general, the missing data could be processed in many ways, such as Single imputation and Multiple imputation[10]. In this problem, linear regression of Single imputation could meet the demand well. So a linear regression model is built to predict the missing data in 3-year repayment rate, Median earnings of students and Share of students earning over $25,000

3.2 The procession of linear regression

1. As there may be some nonlinear elements, the data sets need to be mapfeatured first. In this step, the feature mapping function map the features into polynomial terms, returns a new feature array with more features, comprising of $X_1, X_2, {X_1}^2, {X_2}^2, X_1{*}X_2, (X_1 * X_2)^2$, etc.

2. There are some features that contains more than 1000 missing data, which need to be done with FeatureNormalize. As features differ by order of magnitude, first performing feature scaling can make gradient descent converge much more quickly. And FeatureNormalize contains two steps:

   1. Subtract the mean value of each feature from the dataset.

   2. After subtracting the mean, additionally scale the feature values by their respective 'standard deviations'.

3. To combat the overfitting problem, the regularized linear regression model could be used. In this circumstance, the cost function and gradient is determined by the following formula:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(\sum_{j=0}^{n} x_{ij}\theta_j - y_i\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \tag{1}$$

$$\boldsymbol{grad} = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=0}^{n} x_{ij}\theta_j - y\right)\boldsymbol{x_i} + \frac{\lambda}{m}\boldsymbol{\theta}(\theta_0 = 0) \tag{2}$$

4. Learning parameters using $\boldsymbol{fminunc}$(Matlab Function:Find minimum of unconstrained multivariable functionexpand all in page)

The results of learning parameters could be expressed with learning curve. Combine the trained with the value of cost function in training set and cross validation set, we could get the reliable complete data set.

3.3 Assessment of rationality of the model

It's clear to learned that whether the result of this model is convergent or not from the cost function in training set. Looking into the cost of cross validation set can help on choosing more proper parameter λ. This section will be discussed specificly in Result analysis section, which will not repeat here.

4 Return on investment(ROI)

Return on investment (ROI) is the benefit to the investor resulting from an investment of some resource. A high ROI means the investment gains compare favorably to investment cost. As a performance measure, ROI is used to evaluate the efficiency of an investment or to compare the efficiency of a number of different investments[11].

In this problem, ROI refers more to the latter- the improvement of students' performance, not the economic feedback for foundations, which means we could not accurately measure the ROI by money. So ROI need to be redefined with elements in the data sets.

Besides it is necessary to eliminate meaningless factors, which may mislead the results. Because too many details may reduce the consistency or even make the results unrealistic. Predominant factors therefore should be picked up from the information provided. And here are 13 factors need to be considered, and those elements can be divided into three aspects. And these three aspects composed of 12 elements together define the return on investment.

1. Fundamental conditions of school
2. Predominant degree awarded. Which can reflect the academic level in general.
3. Control of institution. Public universities and privately established schools really have differences in cost, scale, financial sources and so on.
4. Locale of institution. Location of school and the surroundings may have some connection with the scale of school and its education quality.
5. Enrollment of undergraduate degree-seeking students. The number of undergraduate students have something to do with the amount of economic support.
6. Average net price for Title IV institutions/public institutions. When considering the support for schools, the cost of students must be involved.
7. Achievements of undergraduates.
8. Percentage of undergraduates who receive a Pell Grant. The Pell Grant can reflect the performance of a student on a certain degree.
9. Percent of all federal undergraduate students receiving a federal student loan. This element shows the economic pressure of students in general.
10. Median debt of completers, suppressed for n=30. Debt of completers may reflect the time that has been spent on school life. Students that concentrate on school life tend to have more debts than those meanwhile working outside.
11. Median debt of completers expressed in 10-year monthly payments, suppressed for n=30.
12. Achievements of graduates
13. 3-year repayment rate, suppressed for n=30. The repayment can directly express the performance of graduates.
14. Median earnings of students working and not enrolled 10 years after entry. Earnings can represent the return in general.
15. Share of students earning over $25,000/year (threshold earnings) 6 years after entry. This factor cooperate with the other two above to reflect the economic situation of graduates.

One effective way to calculate the ROI overover these selected features is weighted sum approach. After pickingup 12 features, we will use the algorithm of the AHP to analyse data. Concretely, we will generate the weight vector using AHP Mode.

5 AHP model

5.1 Tradational AHP Model

the traditional analytic hierarchy process(AHP), which was originally developed by $Saaty$ in 1970s[4], is a structured technique for organizing and analyzing complex decisions, based on psychology. It is a decision-making procedure widely used in management science, business, healthcare shipbuilding[6] and operations research for establishing priorities within the context of multi-criteria decision making. It help the decision makers to solve the problem by decomposing a complex problem into a multilevel hierarchic structure of objects, criteria, sub-criteria and alternatives and provides a fundamental scale of relative magnitudes expressed in dominance units to represent judgements in the form of pairwise comparisons[13].

After deriving a ratio scale of relative magnitudes that are expressed in priority units from each set of comparisons, an overall ratio scale of priorities is synthesized to rank the alternatives[5]. Three principles can be used to summarize the procedure of an AHP, which are decomposition, comparative judgements, and synthesis of priorities[2].

The AHP help decision makers find one that best suits their goal and their understanding of the problem.

First, we decompose our decision problem into a hierarchy of more easily comprehended sub problems, every one of which can be analyzed independently. The elements of the hierarchy can relate to any aspect of the decision problem-tangible or intangible, carefully measured or roughly estimated, well or poorly understood- anything at all that applies to the decision at hand.

Once our hierarchy is built, we systematically evaluate its various elements by comparing them to each two at a time, with respect to their impact above them in the hierarchy. In making the comparisons, we can use concrete data about the elements, but the typically use our judgments about the elements' relative meaning and importance. It is the essence of the AHP that human judgments, and not just the underlying information, can be used in performing the evaluations.

The AHP converts these evaluations to numerical values that can be processed and compared over the entire range of the problem[8]. A numerical weight or priority is derived for each element of the hierarchy, allowing diverse and often incommensurable elements to be compared to one another in a rational and consistent way[13]. This capability distinguishes the AHP from other decision making techniques[8].

As can be seen in the material that follows, using the AHP involves the mathematical synthesis of numerous judgments about the decision problem at hand. It is not uncommon for these judgments about the decision problem at hand[4]. It is not uncommon for these judgments to number in the dozens or even the hundreds. While the math can be dozen by hand or with a calculator, it is far more common to use one of several computerized methods for entering and synthesizing the judgments[8]. The simplest of these involve standard spreadsheet software, while the most complex use custom software, often augmented by special devices for acquiring the judgments of decision makers gathered in a meeting room[8].

5.2 The typical procedure for using the AHP

1. Model the problem as a hierarchy containing the decision goal, the alternatives for reaching it, and the criteria for evaluating the alternatives.

2. Establish priorities among the elements of the hierarchy by making a series of judgments

based on pairwise comparisons of the elements.

3. Synthesize these judgments to yield a set of overall priorities for the hierarchy. This would combine the investors' judgments about location, price and timing for properties A, B, C and D into overall priorities for each property.

4. Check the consistency of the judgments.

5. Come to a final decision based on the results of this process.

5.3 Assessment of rationality of the model

Consistence checking is a proven way to assess the AHP model. This section will be discussed specificly in Result analysis section, which will not repeat here.

5.4 Data processing and results

As has been explained before,12 elements have been picked up to reflect ROI. Now set the elements-Predominant degree awarded, Control of institution, Locale of institution, Enrollment of undergraduate degree-seeking students, Average net price for Title IV institutionsãĂĄPercentage of undergraduates who receive a Pell Grant, Percent of all federal undergraduate students receiving a federal student loan, Median debt of completers, Median debt of completers expressed in 10-year monthly payments, 3-year repayment rate, Median earnings of students working and not enrolled 10 years after entry, Share of students earning over $25,000/year (threshold earnings) 6 years after entry as sub-criteria C1-C12. And the 12 elements could be divided into three aspects: School(C1-C5), Undergraduates(C6-C9), graduates(C10-C12), which could be set as criteria B1-B3 Figure 1.

Through expert group judgement with rank policies from other professional ranking institutions, we get reciprocal matrix (judgement matrix) mat_1, mat_2, mat_3. Then through AHP, here comes out the eigenvectors (normalized) corresponding to max eigenvalues:

$$\boldsymbol{w_1'} = \begin{bmatrix} 0.4393 & 0.1045 & 0.0444 & 0.2245 & 0.1873 \end{bmatrix} \tag{3}$$

$$\boldsymbol{w_2'} = \begin{bmatrix} 0.4444 & 0.1111 & 0.2222 & 0.2222 \end{bmatrix} \tag{4}$$

$$\boldsymbol{w_3'} = \begin{bmatrix} 0.5396 & 0.1634 & 0.2970 \end{bmatrix} \tag{5}$$

Then merge C1-C5 into C_1, C6-C9 into C_2, C10-C12 into C_3, we get:

$$\begin{bmatrix} B_i \end{bmatrix} = \begin{bmatrix} C_i \end{bmatrix} \times \begin{bmatrix} w_i \end{bmatrix} \; i = 1, 2, 3 \tag{6}$$

Similarly, through expert group judgement, we get reciprocal matrix mat from C_1, C_2, C_3, and then we get the eigenvectors w′ (normalized) corresponding to max eigenvalues through AHP agiain:

$$\boldsymbol{w'} = \begin{bmatrix} 0.6000 & 0.3000 & 0.1000 \end{bmatrix} \tag{7}$$

The score of top N schools and the amount of donation are shown in next chapter Table 2:

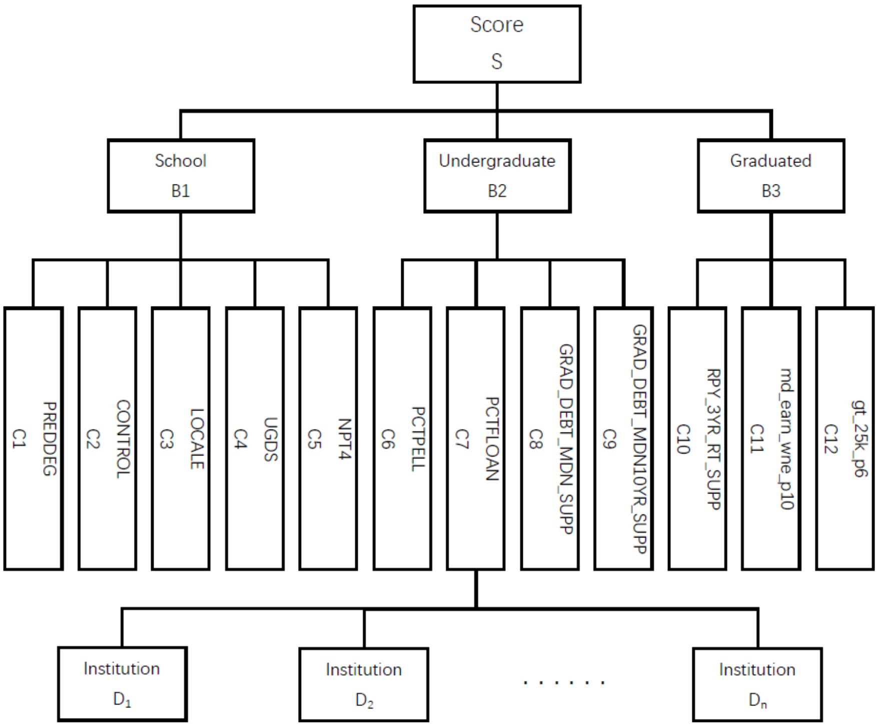

Figure 1: AHP model

Item	Weight
PREDDEG	0.26355
CONTROL	0.062706
LOCALE	0.02667
UGDS	0.134692
NPT4	0.112382
PCTPELL	0.133333
PCTFLOAN	0.033333
GRAD_DEBT_MDN_SUPP	0.066667
GRAD_DEBT_MDN10YR_SUPP	0.066667
RPY_3YR_RT_SUPP	0.053961
md_earn_wne_p10	0.016342
gt_25k_p6	0.029696

Table 1: The weight of items

6 Investment strategy

6.1 Determining the value of N

As we've get the candidate school ranked by ROI scores, the priority and the corresponding weight of each school to get funding has been determined. But the amount of donation for each school still depends on how much schools will get invested. An appropriate method should be found to get more schools benefitted while every benefitted school get enough support.

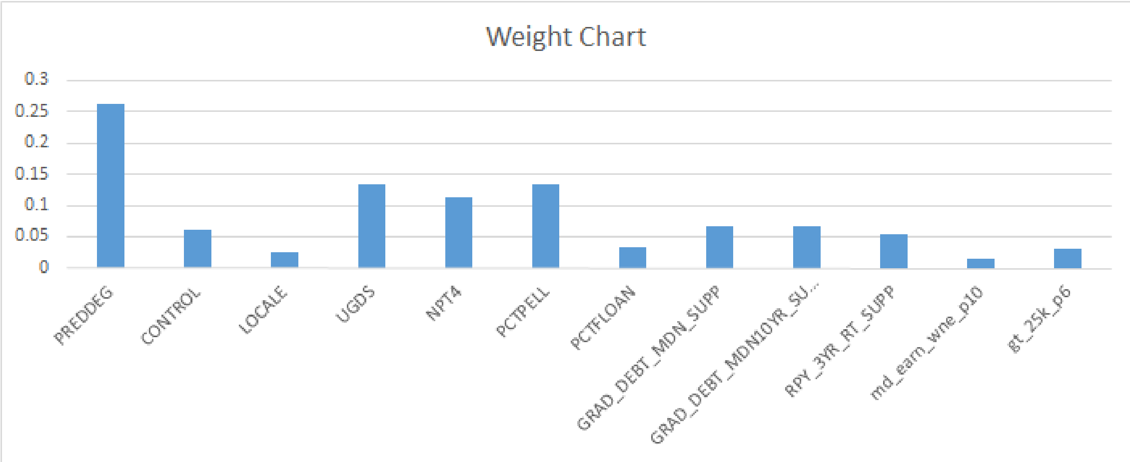

Figure 2: The weight chart

Assume that top N schools in the rank table will get investment, obviously the 1^{st} school get the most money and the Nst school get the least money. By analyzing the donation records of several schools, we get an conclusion that the least amount of money that will give a college enough help is about $2,000,000, and $5,000,000 million is the top limit of efficient investment. And we get the relationship between the most/least money and the number N chosen, which could be shown as following figures:

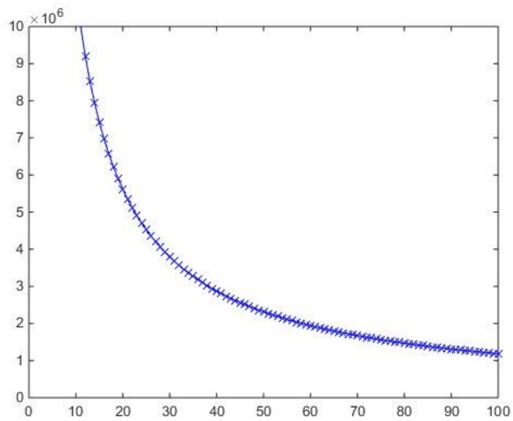

Figure 3: Most money recieved

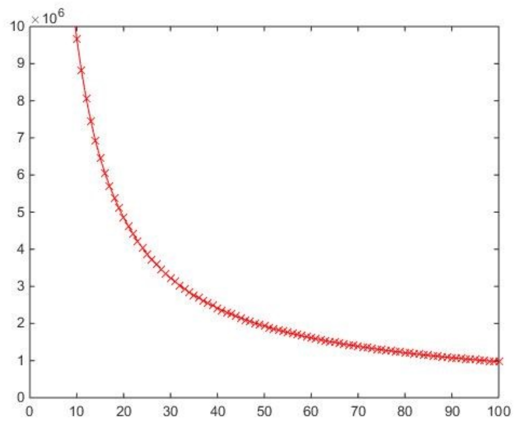

Figure 4: Least money recieved

From these figures, we can see that the most/least money will decrease if N gets larger. According to the result above, the most money should not be more than $5,000,000, and the least money should not be less than $2,000,000. Combine with the figure, we can get the conclusion that 22 schools be invested should be the most effective and efficient way. With N = 22, the amount of money for every school is also determined.

6.2 Schools ranked by investment amount

As the data sets will change as time goes by, the rank result will also get updated. The first year's invest plan is shown below Table 2:

UNITID	INSTNM	Score	Money
232557	Liberty University	4.055349646	5120877.083
150987	Ivy Tech Community College	3.819812778	4823453.813
132903	University of Central Florida	3.735811479	4717381.498
214777	Pennsylvania State University-Main Campus	3.700081951	4672264.175
164872	Boston Architectural College	3.636278663	4591696.819
109651	Art Center College of Design	3.605513625	4552848.387
193900	New York University	3.603229283	4549963.843
133979	Florida Memorial University	3.596437466	4541387.502
133951	Florida International University	3.579236866	4519667.511
123952	Southern California Institute of Architecture	3.5691246	4506898.286
385619	Everglades University	3.566176079	4503175.053
204796	Ohio State University-Main Campus	3.552167052	4485485.208
171100	Michigan State University	3.549879846	4482597.047
102377	Tuskegee University	3.530919899	4458655.448
228778	The University of Texas at Austin	3.530853676	4458571.826
212054	Drexel University	3.521109863	4446267.862
186380	Rutgers University-New Brunswick	3.51945571	4444179.088
433387	Western Governors University	3.514403689	4437799.667
164988	Boston University	3.505537472	4426603.886
183026	Southern New Hampshire University	3.502775741	4423116.521
166656	MCPHS University	3.500158862	4419812.067
138947	Clark Atlanta University	3.498167442	4417297.41

Table 2: Schools ranked by investment amount

7 Analysis of the results

7.1 Stability analysis

Stability analysis of linear regression In order to analyze the stability, we plot a learning curve and cross validation curve of error v.s. m and λ as belowFigure 5:

Learning curves is useful in debugging learning algorithms. It is observed that both the train error and cross validation error are high when the number of training examples is increased. This reflects a high bias problem in the model - the linear regression model istoo simple and is unable to fit our dataset well. In the next picture, after the implement of polynomial regression to fit a better model for this dataset, there will not be a gap between the training and cross validation errors, indicating a high variance problem.

the value of λ can significantly affect the results of regularized polynomial regression on the training and cross validation set. In particular, a model without regularization ($\lambda = 0$) fits the training set well, but does not generalize. Conversely, a model with too much regularization ($\lambda = 100$) does not fit the training set and testing set well. A good choice of λ can provide a good fit to the data.

Plotting the cross validation curve implement an automated method to select the λ parameter (Figure 6). Concretely, when using a cross validation set to evaluate how good each λ value is, a proper λ could be chosen. After selecting the best λ value using the cross validation set, we can then evaluate the model on the test set to estimate how well the model will perform on actual unseen data.

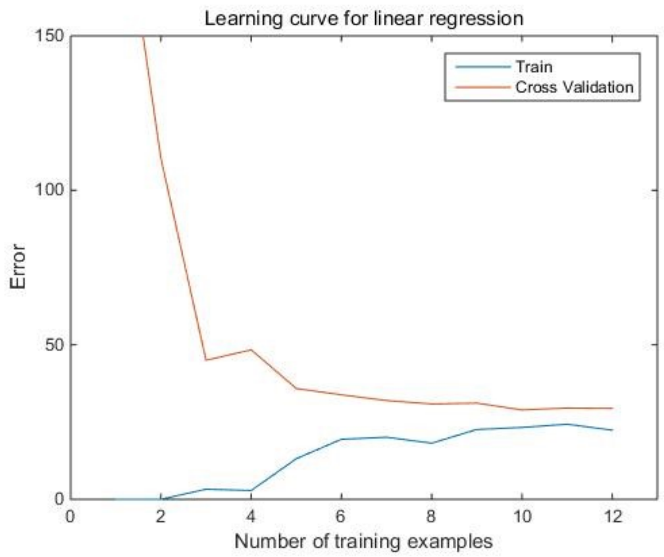

Figure 5: Learning curve for linear regression

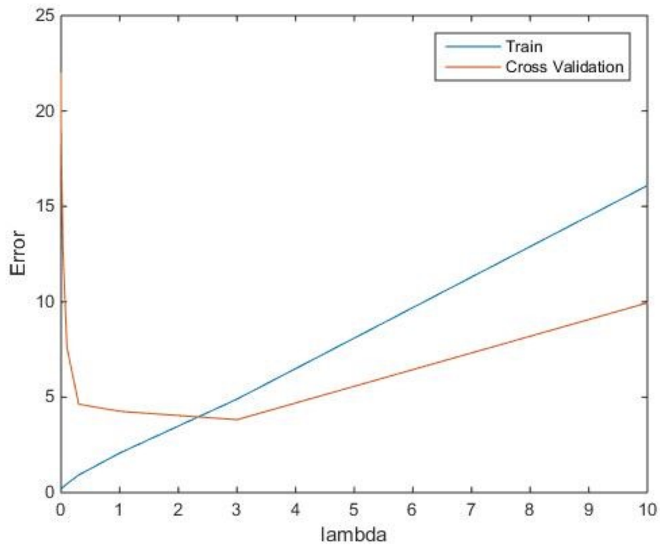

Figure 6: The cross validation curve

In this figure (Figure 6), we can see that the best value of λ is around 3. Due to randomness in the training and validation splits of the dataset, the cross validation error can sometimes be lower than the training error.

Stability analysis of AHP In general, the reciprocal matrix is not consistent. But in order to use the eigenvectors corresponding to max eigenvalues as the weight vector of compared elements, the inconsistency should be in the allowable range.

But the larger that λ is than n, the more severe that the inconsistency of matrix $\boldsymbol{A}$ will be, and the error will be larger when regarding the eigenvectors as weight vectors. So the consistency index was defined by $Saaty$ as:

$$\boldsymbol{CI} = \frac{(\lambda - n)}{(n-1)} \tag{8}$$

As the standard of $\boldsymbol{CI}$, Saaty then came up with the concept of random index, $\boldsymbol{RI}$ for short. $\boldsymbol{RI}$ comes from the average of $\boldsymbol{RI}$ of the reciprocal matrixes constructed randomly. The empirical value of RI is given in the table below (Table 3):

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51

Table 3: The empirical value of RI

Define $\boldsymbol{CR}$ as the ratio of $\boldsymbol{CI}$ and $\boldsymbol{RI}$, when

$$\boldsymbol{CR} = \frac{\boldsymbol{CI}}{\boldsymbol{RI}} < 0.1 \tag{9}$$

We can say that the inconsistency is in the allowable range, which means that it is reasonable to regard the eigenvectors as weight vectors.

CR	mat_1	mat_2	mat_3	mat_4
Value	0.0445	-0.0000	0.0079	-0.0000

Table 4: The CR of 4 matrix

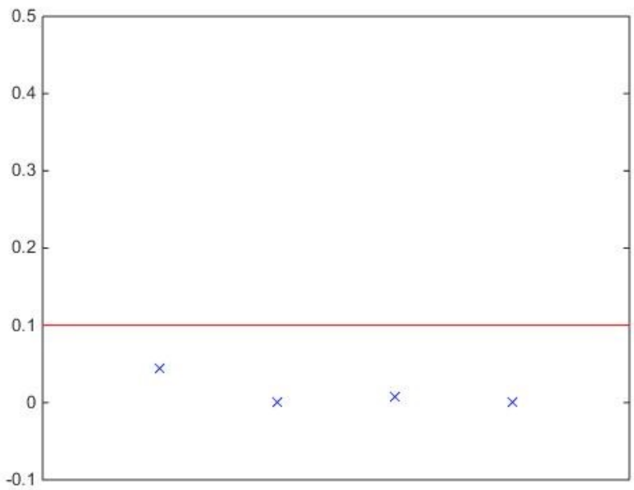

Figure 7: The CR of 4 matrix

According to the theory above, we can conclude that the inconsistency is in the allowable range, and it's reasonable to regard the eigenvectors as weight vectors.

7.2 Strength and weaknesses

Strength Through the theory of AHP, it's not hard to say that our model has the following strength:

1. Structed. Our model regard the problem as a system, which could be divided and solved by comparison.

2. Practicable. The AHP model combine the methods of quantitative and qualitative. Thus it can deal with realistic problem which may not be solved by traditional theory.

3. Concise. Even people with medium degree of education could understand how this model works and get used of its steps, and the calculation won't be complex.

Weaknesses However, there are still some weaknesses. Firstly , deriving fuzzy weights from fuzzy pairwise comparison matrices may be too sophisticated and rare to be applied, while deriving crisp weights from fuzzy pairwise comparison matrices proves to be either invalid or subject to significant drawbacks such as producing multiple even conflict priority vectors for a fuzzy pairwise comparison matrix, leading to distinct conclusions[7].

Besides the methods of eigenvector method or LLSM are both computationally simple but they does not reach acceptable values of the optimization criteria as maximum deviation, sum of deviations, etc[3].

8 Conclusion

The model is based on the real condition and the data given, and we combined them to determine the optimized investment strategy.

- We formulate a Single imputation with linear regression model to processing the missing data. This step could make sure that our results come from credible data.
- The ROI for charitable organizations is redefined by elements given in data sets. Through AHP model, the ROI score of each school is figured out. Then we get the schools ranked by ROI score.
- Combining the ranking result with reality, we determine the concrete investment strategy including the investment list, the amount of each donation and the time duration.
- The stability of this model is analyzed to see how does this model respond to small changes.

References

[1] Lumina Foundation. Lumina foundation. `https://www.luminafoundation.org/`.

[2] Patrick T Harker and Luis G Vargas. The theory of ratio scale estimation: Saaty's analytic hierarchy process. Management science, 33(11):1383–1403, 1987.

[3] J Jablonsky. Analysis of selected prioritization methods in the analytic hierarchy process. In Journal of Physics: Conference Series, volume 622, page 012033. IOP Publishing, 2015.

[4] Thomas L Saaty. A scaling method for priorities in hierarchical structures. Journal of mathematical psychology, 15(3):234–281, 1977.

[5] Thomas L Saaty. An exposition of the ahp in reply to the paper "remarks on the analytic hierarchy process". Management science, 36(3):259–268, 1990.

[6] Burak Omer Saracoglu. Selecting industrial investment locations in master plans of countries. European Journal of Industrial Engineering, 7(4):416–441, 2013.

[7] Ying-Ming Wang and Kwai-Sang Chin. Fuzzy analytic hierarchy process: A logarithmic fuzzy preference programming methodology. International Journal of Approximate Reasoning, 52(4):541–553, 2011.

[8] Wikipedia. Analytic hierarchy process. `https://en.wikipedia.org/wiki/Analytic_hierarchy_process`.

[9] Wikipedia. Data cleaning. `https://en.wikipedia.org/wiki/Data_cleansing`.

[10] Wikipedia. Imputation (statistics). `https://en.wikipedia.org/wiki/Imputation_(statistics)`.

[11] Wikipedia. Return on investment. `https://en.wikipedia.org/wiki/Return_on_investment`.

[12] Shaomin Wu. A review on coarse warranty data and analysis. Reliability Engineering & System Safety, 114:1–11, 2013.

[13] Zeshui Xu and Huchang Liao. Intuitionistic fuzzy analytic hierarchy process. Fuzzy Systems, IEEE Transactions on, 22(4):749–761, 2014.

Appendices

Appendix A Letter to the CFO of the Goodgrant Foundation

Dear Mr. Alpha Chiang:

Firstly please let us show the greatest respect to all the charitable organizations just like you that working hard for people's happiness and dreams. Education can not only improve students appearance but also change their whole life sometimes. And we both have the same aim now-make sure every dollar invested counts.

In order to achieve this goal, we analyzed the data of thousands of schools in America, including the fundamental information of schools themselves, the performance of under-graduates and the economic situation of graduates. The ROI, is redefined by these three aspects including 12 elements in the data set. And with the ROI defined, we could judge which school have potential to make good use of the investment.

Besides, in order to avoid duplicating the investments and focus of other large grant organizations, we built our own model rather than using the model that has been used by other organizations. May this model provide feasible suggestions for your investment. As it comes to choose schools from candidates, we finally choose to simulate the AHP model, which can transform the complex problem-distributing weight among elements into comparing every two elements. And we set high ROI to be the aim point, then get the weight allocation among the 12 elements. Then schools are ranked by ROI scores.

Here comes the top 10 schools in the table.

UNITID	INSTNM	Score	Money
232557	Liberty University	4.055349646	5120877.083
150987	Ivy Tech Community College	3.819812778	4823453.813
132903	University of Central Florida	3.735811479	4717381.498
214777	Pennsylvania State University-Main Campus	3.700081951	4672264.175
164872	Boston Architectural College	3.636278663	4591696.819
109651	Art Center College of Design	3.605513625	4552848.387
193900	New York University	3.603229283	4549963.843
133979	Florida Memorial University	3.596437466	4541387.502
133951	Florida International University	3.579236866	4519667.511
123952	Southern California Institute of Architecture	3.5691246	4506898.286

Table 5: Top 10 Schools

As investment is limited to 100 million, we want more schools get benefitted while it's also necessary to make sure every school get enough support. Finally top 22 schools in the table are picked up to get donation distributed in accordance with ROI scores. The investment plan is shown in the table attached.

In conclusion, schools are picked up by ROI potential and the invest amount depends on the scores also. All these are to make sure the investment get more effect on the improvement of students' performance.

Sincerely

MCM team

Appendix B Source Code

Algorithm source code and simulation programm used in our model as follow.

Input matlab source:

```
function [w,CR] = AHP(A)
n = size(A,1);
m = size(A,2);
if n ~= m
    fprintf('Error, A must be a square matric.\n');
    return
end
[V,D] = eig(A);
D_colonm = diag(D);
lambda = max(D_colonm);
w = V(:,find(D_colonm==lambda));

CI = (lambda-n)/(n-1);
RI = [0 0 0.58 0.90 1.12 1.24 1.32 1.41 1.45 1.49 1.51];%data only for n = 1~11
CR = CI/RI(n);%Consistency condition checking
if CR >= 0.1
    fprintf('Consistency condition check: Unsatisfied\n');
end
end
```

```
function [error_train, error_val] = ...
    learningCurve(X, y, Xval, yval, lambda)
%LEARNINGCURVE Generates the train and cross validation set errors needed
%to plot a learning curve
%   [error_train, error_val] = ...
%       LEARNINGCURVE(X, y, Xval, yval, lambda) returns the train and
%       cross validation set errors for a learning curve. In particular,
%       it returns two vectors of the same length - error_train and
%       error_val. Then, error_train(i) contains the training error for
%       i examples (and similarly for error_val(i)).
%
%   In this function, you will compute the train and test errors for
%   dataset sizes from 1 up to m. In practice, when working with larger
%   datasets, you might want to do this in larger intervals.
%

% Number of training examples
m = size(X, 1);

% You need to return these values correctly
error_train = zeros(m, 1);
error_val   = zeros(m, 1);

% ====================== YOUR CODE HERE ======================
% Instructions: Fill in this function to return training errors in
%               error_train and the cross validation errors in error_val.
%               i.e., error_train(i) and
%               error_val(i) should give you the errors
%               obtained after training on i examples.
%
% Note: You should evaluate the training error on the first i training
%       examples (i.e., X(1:i, :) and y(1:i)).
%
%       For the cross-validation error, you should instead evaluate on
%       the _entire_ cross validation set (Xval and yval).
%
```

```
% Note: If you are using your cost function (linearRegCostFunction)
%       to compute the training and cross validation error, you should
%       call the function with the lambda argument set to 0.
%       Do note that you will still need to use lambda when running
%       the training to obtain the theta parameters.
%
% Hint: You can loop over the examples with the following:
%
%       for i = 1:m
%           % Compute train/cross validation errors using training examples
%           % X(1:i, :) and y(1:i), storing the result in
%           % error_train(i) and error_val(i)
%           ....
%
%       end
%

% ---------------------- Sample Solution ----------------------
initial_theta = zeros(size(X, 2), 1);
for i=1:m
     % Set Options
     options = optimset('GradObj', 'on', 'MaxIter', 400);

     % Optimize
     [theta, ~, ~] = ...
            fminunc(@(t)(costFunctionReg(t, X, y, lambda)), initial_theta, options);
     [error_train(i),~] = costFunctionReg(theta, X(1:i,:), y(1:i), 0);
     [error_val(i),~] = costFunctionReg(theta, Xval, yval, 0);
end



% -------------------------------------------------------------

% =========================================================================

end
```

```
clear;clc;

% Training data loading
X = load('X_Gates.txt');
Y = load('Y_Gates.txt');

% Feature Normalize
[X_norm, ~, ~] = featureNormalize(X);
Y = Y/sum(Y);

X_norm = [ones(size(X_norm,1),1) X_norm];

X_train=X_norm(1:30,:);
X_val=X_norm(31:end,:);

Y_train=Y(1:30,:);
Y_val=Y(31:end,:);

initial_theta = zeros(size(X_norm, 2), 1);

% Set regularization parameter lambda to 1
lambda = 1;

% Compute and display initial cost and gradient for regularized logistic regression
[cost, grad] = costFunctionReg(initial_theta, X_train, Y_train, lambda);
```

```
fprintf('Cost at initial theta (zeros): %f\n', cost);

% Set Options
options = optimset('GradObj', 'on', 'MaxIter', 400);

% Optimize
[theta, J, exit_flag] = ...
    fminunc(@(t)(costFunctionReg(t, X_train, Y_train, lambda)), initial_theta,
        options);

[cost, grad] = costFunctionReg(theta, X_train, Y_train, lambda);

fprintf('Cost at trained theta: %f\n', cost);
% Compute result of our own data
X_self = load('X_self.txt');

[X_self, ~, ~] = featureNormalize(X_self);
X_self = [ones(size(X_self,1),1) X_self];

Y_self = X_self*theta;

m = 30;
% Learning Curve
lambda = 0;
[error_train, error_val] = ...
    learningCurve(X_train, Y_train, ...
                  X_val, Y_val, ...
                  lambda);

plot(1:m, error_train, 1:m, error_val);
title('Learning curve for linear regression')
legend('Train', 'Cross Validation')
xlabel('Number of training examples')
ylabel('Error')
axis([0 13 0 1.50])

pause;
% lambda
[lambda_vec, error_train, error_val] = ...
    validationCurve(X_train, Y_train, X_val, Y_val);

close all;
plot(lambda_vec, error_train, lambda_vec, error_val);
legend('Train', 'Cross Validation');
xlabel('lambda');
ylabel('Error');
```

```
clear;clc;

mat=zeros(3,3);
mat_1=zeros(5,5);
mat_2=zeros(4,4);
mat_3=zeros(3,3);

% Set a_i,j for i<j
mat_1(1,2:end)=[5 7 2 3];
mat_1(2,3:end)=[2 1/3 1];
mat_1(3,4:end)=[1/5 1/7];
mat_1(4,5:end)=1;

mat_2(1,2:end)=[4 2 2];
mat_2(2,3:end)=[1/2 1/2];
```

```
mat_2(3,4:end)=1;

mat_3(1,2:end)=[3 2];
mat_3(2,3:end)=1/2;

mat(1,2:end)=[2 6];
mat(2,3:end)=3;

% Complete matrix
for i=1:3
    for j=1:i
        if i==j
            mat(i,j)=1;
        else
           mat(i,j)=1/mat(j,i);
        end
    end
end

for i=1:5
    for j=1:i
        if i==j
            mat_1(i,j)=1;
        else
           mat_1(i,j)=1/mat_1(j,i);
        end
    end
end

for i=1:4
    for j=1:i
        if i==j
            mat_2(i,j)=1;
        else
           mat_2(i,j)=1/mat_2(j,i);
        end
    end
end

for i=1:3
    for j=1:i
        if i==j
            mat_3(i,j)=1;
        else
           mat_3(i,j)=1/mat_3(j,i);
        end
    end
end


X = load('On candidate list.txt');

% Feature Normalize
[X_norm, ~, ~] = featureNormalize(X);

CR = zeros(4,1);
[w1,CR(1)] = AHP(mat_1);
[w2,CR(2)] = AHP(mat_2);
[w3,CR(3)] = AHP(mat_3);

% Normalization
w1=w1/sum(w1);
w2=w2/sum(w2);
```

```
w3=w3/sum(w3);

y_school = X_norm(:,1:5)*w1;
y_under = X_norm(:,6:9)*w2;
y_graduate = X_norm(:,10:12)*w3;

[w,CR(4)] = AHP(mat);

w=w/sum(w);

score = [y_school y_under y_graduate]*w;

[b,i]=sort(score,'descend');
N=1:100;
Money_max = zeros(length(N),1);
Money_min = zeros(length(N),1);
for i=1:100
    Money_max(i) = 100000000*b(1)/sum(b(1:i,:));
    Money_min(i) = 100000000*b(i)/sum(b(1:i,:));
end
plot(N, Money_max, 'bx',N, Money_max, 'b');
axis([0 100 0 10000000]);
figure;
plot(N, Money_min, 'rx', N, Money_min,  'r');
axis([0 100 0 10000000]);
for i=1:100
    if Money_max(i)<5000000
        idx1=i-1;
        break;
    end
end
for i=1:100
    if Money_min(i)<2000000
        idx2=i-1;
        break;
    end
end
money_final = zeros(idx1,1);
for i=1:idx1
    money_final(i) = 100000000*b(i)/sum(b(1:idx1,:));
end

figure;
lab=[1 2 3 4];
plot(lab,CR','bx',[0 5],[0.1 0.1],'r');
axis([0 5 -0.1 0.5]);
set(gca,'xtick',[])
```

```
function [lambda_vec, error_train, error_val] = ...
    validationCurve(X, y, Xval, yval)
%VALIDATIONCURVE Generate the train and validation errors needed to
%plot a validation curve that we can use to select lambda
%   [lambda_vec, error_train, error_val] = ...
%       VALIDATIONCURVE(X, y, Xval, yval) returns the train
%       and validation errors (in error_train, error_val)
%       for different values of lambda. You are given the training set (X,
%       y) and validation set (Xval, yval).
%

% Selected values of lambda (you should not change this)
lambda_vec = [0 0.001 0.003 0.01 0.03 0.1 0.3 1 3 10]';

% You need to return these variables correctly.
```

```
error_train = zeros(length(lambda_vec), 1);
error_val = zeros(length(lambda_vec), 1);

% ====================== YOUR CODE HERE ======================
% Instructions: Fill in this function to return training errors in
%               error_train and the validation errors in error_val. The
%               vector lambda_vec contains the different lambda parameters
%               to use for each calculation of the errors, i.e,
%               error_train(i), and error_val(i) should give
%               you the errors obtained after training with
%               lambda = lambda_vec(i)
%
% Note: You can loop over lambda_vec with the following:
%
%       for i = 1:length(lambda_vec)
%           lambda = lambda_vec(i);
%           % Compute train / val errors when training linear
%           % regression with regularization parameter lambda
%           % You should store the result in error_train(i)
%           % and error_val(i)
%           ....
%
%       end
%
%
initial_theta = zeros(size(X, 2), 1);
for i=1:length(lambda_vec)
    % Set Options
     options = optimset('GradObj', 'on', 'MaxIter', 400);

     % Optimize
     [theta, ~, ~] = ...
            fminunc(@(t)(costFunctionReg(t, X, y, lambda_vec(i))), initial_theta,
                options);
    [error_train(i),~] = costFunctionReg(theta, X, y, 0);
    [error_val(i),~] = costFunctionReg(theta, Xval, yval, 0);
end




% =========================================================================

end
```