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

Team Control Number
74002

Problem Chosen
C

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

2017
MCM/ICM
Summary Sheet
A Data Driven Energy Profile Evaluation and Prediction System

On the cusp of an energy transformation, more attention than ever is focused on energy issues. As the dramatical benefits that new energy can bring to the society are admitted, it is necessary to learn the current situation of energy. Meanwhile, with the advent of the era of big data, Data Mining becomes increasingly efficient and practical. Our solutions consist of 2 models.

The first step is to process the data. After removing some garbage data, we find that there exists some missing values which are completed by K-means method. After that, the remaining 587 variables (ABICB etc.) are reconsidered and thus classified into 6 major features, i.e. we analyze them according to those features defined, on which our energy profile mainly focuses as well.

The second step is inspired by collaborative filtering algorithms in machine learning. Here we develop the first model: Collaborative-Filtering-Based Energy Profile Evaluation Model (CFEM) to evaluate the energy profile. In this model, 2 important evaluation matrixes are introduced which respectively reflect the characteristics of different years and different variables for the 6 features. The 2 two matrices who have a mutual filtering effect determine our energy profile. We end up by plotting radar charts so that the results are readily comprehensible.

Thirdly, the second model is projected to predict the future. This model formulated by time series method is named Time-Series-Based Exponential Smoothing Prediction Model (TSPM). Through data visualization, we find that the data of the four states before 2000 do not have significant influence to our prediction results which can be interpreted as the lack of awareness of the importance of new energy resources. Therefore we seek to minimize the effect of over-long data on our prediction using a time-smoothed approach. Thus inherent relation between energy profile and time is obtained, and this helps us predict the future energy profile.

In the fourth part, we discuss the possible impact of geography, industry, population and climate on our model. Besides, we give seasonable suggestions and set goals for the 4 states, among which an interstate compact is included.

Lastly, we perform Sensitivity Analysis on our model and check its validation. Then we analyze the strengths and weaknesses of our model.

To sum up, our model is a data driven model based on credible theories and methods which is considered reliable.

Key words: Data Mining, Collaborative Filtering, Time Series

Memo

To: Group of Governors

From: Team74002

Date: Feb 11, 2018

Subject: Profiles, predictions and advices for energy of 4 states

We are glad to hear that policy makers of the 4 states are paying close attention to new energy in an effort to form a realistic new energy compact focused on increased usage of cleaner, renewable energy sources. As is well-known, energy is vital to the development of a society, especially when currently ecological problems rise frequently. Here in this memo, we will summarize the state profiles as of 2009 along with our predictions as regards energy usage in absence of any policy changes, and our recommended targets for the energy compact to adopt.

Firstly, in 2009, we find with pleasure that California perform pretty well with a high score of 1.004 and Texas ranks second with a score of 1.002 in our model formulated. More precisely, both of them boost with a relatively high production of new energy. Moreover, their enormous potential for new energy development is a prove of a remarkable energy profile. Nevertheless the score of New Mexico in 2009 counts merely 0.267 due to its strong dependence on traditional energy, and this is unfavourable for sustainable development. As for Arizona, it has a score of 0.696, in spite of its dependence on traditional energy, we have found tremendous potential for new energy development in this state.

Secondly, by making predictions of the energy profile for the 4 states in 2025 and 2050 through quantifiable model, we find that all the states have an obvious improvement. The score of California and Texas reach 10 in 2050, which is what we sincerely look forward to.

In the end, we feel that a few of our central results can offer some insight for policy makers devoting to energy optimization. According to our analysis, we make feasible suggestions to the 4 states. First, except for maintaining the current status, California and Texas are supposed to further explore their advantages and enhance the potential of new energy. To be specific, build more solar and wind power centres. In addition, as for Arizona and New Mexico, they should accelerate energy transformation to get rid of their dependence on traditional energy. For example, replace conventional fossil-fuel power stations by nuclear power. What's more, an interstate cooperative contract should be concluded so as to achieve energy balancing distribution.

We hope this report riveting and that it will be helpful for administrations of the 4 states. We wish them the best in their endeavors.

Respectfully,
Team 74002

Contents

1 Introduction

1.1 Restatement of the Problem 1

1.2 Overview of Our Work 1

1.3 Assumptions and Justifications 2

2 Data Processing **2**

2.1 Overview of Data 2

2.2 Data Completion 3

2.2.1 K-means Clustering 3

2.2.2 Data Completion by K-means 4

2.2.3 Data Normalization by Z-score 4

2.2.4 Visualization of Data 4

3 Model 1: Collaborative-Filtering-Based Energy Profile Evaluation Model (CFEM)

3.1 Variable Mapping and Classification 5

3.2 CFEM Model 6

3.2.1 Introduction of the feature matrix 6

3.2.2 Basic Thinking 6

3.2.3 Collaborative Filtering Algorithm 6

3.2.4 Cost Function 7

3.2.5 Gradient Descent 7

3.2.6 Energy Profile 8

3.2.7 Energy Profile Criteria and Choice 9

3.3 Discussion on Potential Factors 10

3.4 Discussion of the potential factors 10

3.4.1 Geography [based on the assumption] 10

3.4.2 Climate [based on the assumption] 11

3.4.3 Population 11

3.4.4 Industry 11

3.5 Energy Usage Targets and Actions 13

3.5.1 Energy Usage Goals: 13

3.5.2 Actions 13
3.5.3 Interstate New Energy Compact 14

4 Model 2: A Time-Series-Based Exponential Smoothing Prediction Model (TSPM)
4.1 Data Collecting 14
4.2 Model TSPM 14
4.2.1 General Introduction 14
4.2.2 Basic Thinking 15
4.2.3 Model Specification 15
4.2.4 Determination of the Original Value 16
4.2.5 Prediction 17

5 Sensitivity Analysis and Validation
5.1 Learning Rate 17
5.2 weighting coefficient 18

6 Conclusion
6.1 Strengths 19
6.2 Weakness 19

References **20**

Appendices **21**

Appendix A Generation of Numpy matrixes **21**

Appendix B Data Selecting **22**

Appendix C K-means and Z-score **23**

Appendix D Collaborative Filtering Algorithm **24**

Appendix E Exponential Smoothing **25**

1 Introduction

1.1 Restatement of the Problem

Recent years, people are increasingly conscious of the importance of energy for human being as a whole, at the same time policy makers from all over the world take more consideration into the impact of energy on policies. It is natural to expect renewable energy to serve human better especially in terms of economic society. And one key point is that the cooperation between states helps achieve this goal.

We know that along the U.S. four states namely California(CA), Arizona(AZ), New Mexico(NM), and Texas(TX) are relatively advanced in this field, thus, given the practical situation, we try to propose a rational program for renewable energy distribution on the basis of data analysis.

In our paper, we investigate mainly energy usage and production of each states, with the goal of formulating a measurable model for corresponding assessment and make scientific measurable prediction of their future development. Here, we adopt the method of collaborative filtering and time series.

1.2 Overview of Our Work

First, we need to solve following problems.

- We need to deal with a vast amount of data in a suitable way.
- Our research refers to 4 states, each with 605 variables of 50 years, altogether 121000 data. With only 105745 data given, obviously there exists missing data which requires to be taken into account.
- As the data given has a wide range and different units, we need to normalize our data.
- We need to consider how to determine the energy profile of a state in a certain year.
- Not all the factors involved in our research are reflected in the data given, hence we need to propose one method to manifest those factors.
- We need to build a time series model which permits a predicition of future energy profile.
- We need to consider how to set practical energy usage targets for the four states along with respective feasible actions.

Secondly, we mainly take following steps.

- First, we fetch information from data given according to different states, more precisely, we obtain a matrix sized $4 \times 50 \times 605$ ($states \times years \times variables$). Thereafter, we use K-means Algorithm to fill the missing data.

- Second, we define and build an assessment system of energy profile taking up with quantification of energy conditions of each state and fix different data in the same range. This is accomplished by Collaborative Filtering Algorithm and z-score normalization.
- Third, we find a way to account for factors of geography, industry, population and climate.
- Fourth, we build a time series model which serves to predict future energy profile.
- Last but not least, further analysis will be discussed in detail in our paper.

1.3 Assumptions and Justifications

Basic assumptions are listed as follows so as to simplify the problem without harming its scientificity. Beyond that, given additional relevant data, the model built could be more accurate and thus more reliable.

- Our target is the benefit maximization for those 4 states. For simplicity, we characterize our model in the light of the impact brought, i.e. that positive number indicates positive effect, while negative number indicates negative effect. More precisely, the score of energy profile has a positive correlation with new energy development degree.
- We seek to formulate a model conforming to all the years to the maximum extent, and therefore it is fair to make an assumption that our model may not the most suitable one for a certain year locally.
- We seek to formulate a model conforming to long-term benefit to the maximum extent, thus the model may not satisfy all the short-term interest.
- We neglect the impact of inflations and deflations in this period.
- We neglect the changes of climates, geography and other factors throughout the 50 years.
- Considering the future predictable, we predict the energy profile of each state in absence of any policy changes and other indeterminable factors.

Under the above and basic assumptions, we can set out to construct our model.

2 Data Processing

2.1 Overview of Data

Overview of States There are 4 states involved in the problem: California (CA), Arizona(AZ), New Mexico (NM), and Texas (TX).

Overview of Years We have data of 50 years of time, from 1960 to 2009.

Overview of Variables There are 605 variables in total which are all regarded as useful in our assessment system, hence we take all the data into consideration.

Nevertheless, we find that in 'ProblemCData.xlsx' worksheet ('seseds'), there are 2 sets of data of value 0 named GETXV and HYTXV which do not exist in worksheet ('msncodes'). Meanwhile, 24 sets of data in worksheet ('msncodes'), namely CCEXB, CCEXD, CCEXP, CCEXV, CCIMB, CCIMD, CCIMP, CCIMV, CCNIB, CCNIP, CCNIV, EMTCP, ENTCK, FFETK, LGTCK, MGTCK, NUETK, PAACK, PACCK, PAEIK, PAHCK, PAICK, PARCK, PATCK, do not belong to worksheet ('seseds'). These data are considered inutile and consequently culled.

In addition, there exist 3 variables who do not relate directly to energy, yet can be treated as potential influencing factors (economics, population, etc). They are ['GDPRV', 'GDPRX', 'TPOPP']. We consider them separately.

After the screening above, we reduced 605 features to 578.

2.2 Data Completion

In the process of observing the data, we find a large amount of missing values among which each state counts 3894. Therefore we take the method of k-means to fill missing values.

2.2.1 K-means Clustering

K-means is a clustering algorithm, whose basic idea is that: Given the training sample set, $D = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \\ x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}$, which has m samples and each sample has n variables.

We think that there exist k categories defined as $C = C_1, C_2, ..., C_k$, our objective is to divide m samples into k categories where k comes from k-means adopted here.

We seek to find k which minimizes the cost function defined as:

$$E = \sum_{i=1}^{k} \sum_{\lambda \in C_i} \|x - \mu_i\|^2 \tag{1}$$

where $\|\|$ indicates the length of vector, $\mu_i = \frac{1}{m_i} \sum_{x \in C_i} x$ indicates the mean vector of category C_i. μ_i describe to some extend the compact degree around the mean value of all the vectors belonging to C_i.

The specific algorithm is as follows.

Input: the training set $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, the number of cluster: k
1. Process: We pick k samples in the training set as the initial vectors $\{\mu_1, \mu_2, \ldots, \mu_k\}$
2. repeat:
3. Make $c_i = \phi$, $1 \leq i \leq k$
4. **for** : $j = 1, 2, \ldots, m$ do:
5. compute the distance between x_j and the average vector μ_i $(1 \leq i \leq k)$: $d_{ji} = \left\|x^{(j)} - \mu_i\right\|$
6. According to the average vector μ_i which is the closest to x_j, we determine its cluster label: $\lambda_j = i$ so that d_{ji} is minimum
7. We put the sample x_j into the associated cluster: $C_{\lambda_j} = C_{\lambda_j} \bigcup \{x_j\}$
8. **end for**
9. **for** : $i = 1, 2, \ldots, k$ do:
10. compute the new average vector : $\mu_{i'} = \dfrac{1}{|m_i|} \sum_{x \in C_i} x$
11. **if** $\mu_{i'} \neq \mu_i$, then:
12. $\mu_i := \mu_{i'}$
13. **else**
14. Keep as constant
15. **end if**
16. **end for**
17. until all the average vector keep constant, which means convergence.

2.2.2 Data Completion by K-means

Now back to the original question, without practical category labels, K-means offers an unsupervised learning algorithm to find automatically data of the same category. After running K-means, a data set well classified is obtained, and we fill those missing values according to known data.

It should be noted that multiple variables have missing parts,thus before running K-means we replace the missing data by the mean value of variable to which it belongs.

At length we acquire by calculation mean value of every variable presented in appendix 1.

2.2.3 Data Normalization by Z-score

The fundamental formula of Z-score is defined as:

$$Z = \frac{x_i - \mu}{\sigma} \tag{2}$$

Here μ is the mean value and σ is the standard deviation of all features in this data set. We normalize all our data so as to dispose of their unit and the impact of data range.

2.2.4 Visualization of Data

We show the data processed of California as our example in Figure 1.

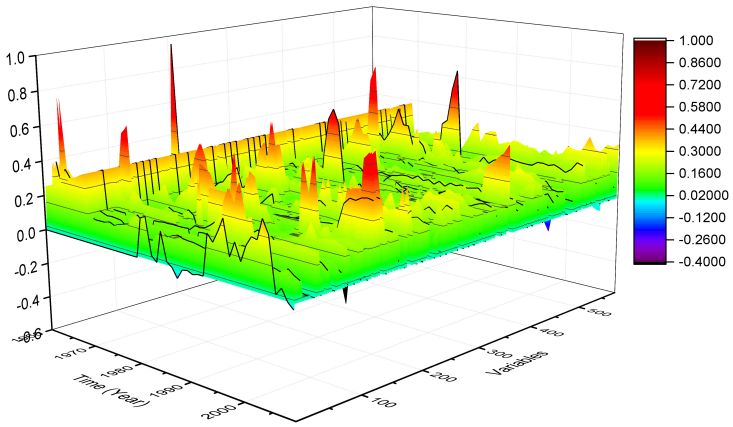

Figure 1: Variation of 578 Variables in 50 Years of California

3 Model 1: Collaborative-Filtering-Based Energy Profile Evaluation Model (CFEM)

3.1 Variable Mapping and Classification

For simplicity, throughout this part, those 578 variables screened through the above steps are mapped onto a set of integers from 1 to 578 in a one to one way. We choose 'Var' to signify all our variables, for instance:

Variable name	Serial number
ARTCP	Var[10]
CLTCD	Var[79]
NGCCV	Var[338]

Table 1: Example of variable names

Through observing all the variables in worksheet ('msncodes'), we analyze these variables considering the following types of features denoted as Cls:

1.Traditional Energy Consumption, TEC

2.Traditional Energy Production, TEP

3.New Energy Consumption, NEC

4.New Energy Production, NEP

5.Traditional Energy Potentiality, TEPT

6.New Energy Potentiality, NEPT

For example, CLPRB(Coal production)indicates a 100% feature of TEP and 0 of NEP.

That is to say, when estimating the energy profile of a state in a certain year, we actually build quantifiable model to rate each of these six aspects sorted above and eventually get the results.

3.2 CFEM Model

For convenience, we make the detailed explanation of California (CA), and models of other 3 states are learned similarly.

3.2.1 Introduction of the feature matrix

Here we introduce 2 distinct matrices. They are:

a) Classification weight matrix of variables: MatCls
This is a matrix of size 578×6, where 578 means the number of variables and six elements in each row of this matrix mean this variable's contributions (or classification weight) to every features in Cls. The greater the value of an element, the greater its contribution to this feature.

b) Energy profile evaluation matrix/Year feature distribution matrix: MatFea
This is a matrix of size 50×6, where 50 indicates 1960-2009 those 50 years and six elements in each row of this matrix indicates distribution of different characteristics in this year. That is to say, the greater the value of an element, the stronger the characteristics of this aspect of the year. Note as well that this distribution of characteristics is exact our evaluation towards energy profile. And our object is to train a characteristic distribution matrix through proper algorithm.

3.2.2 Basic Thinking

Given the data in worksheet ('seseds'), through methods of K-means and z-score, a matrix of size 578×50 is obtained. We here show a part of this matrix (Var[51]-Var[55],1985-1990):

$$D = \begin{bmatrix} 1.359338e-01 & 1.641231e-01 & 1.585406e-01 & 1.675592e-01 & 1.889679e-01 \\ 3.396972e-06 & 3.197920e-06 & 3.213152e-06 & 2.974862e-06 & 2.759626e-06 \\ 1.329887e-01 & 1.389319e-01 & 1.359791e-01 & 1.386315e-01 & 1.371833e-01 \\ 1.410054e-01 & 1.629637e-01 & 1.608389e-01 & 1.667360e-01 & 1.900246e-01 \end{bmatrix}$$

Since each element of this matrix represents the situation of a variable in a particular year, presuming that matrix MatCls and matrix MatFea are obtained, and then we should have the formula below:

$$MatCls \times MatFea' = MatRes \tag{3}$$

where MatFea' represent matrix transposition, and we set out to train MatCls according to matrix MatRes. Now that the matrix MatCls is unknown, we apply the Collaborative Filtering Algorithm in order to get the result needed.

3.2.3 Collaborative Filtering Algorithm

This algorithm initially focuses on the random assignment for element of matrix MatCls and matrix MatFea. For purpose of finding the most appropriate model, we fix

MatCls and find the most suitable matrix MatFea (when the Cost Function achieves a minimum) in this case. Then we use MatFea to find a satisfactory MatCls and repeat these steps until the Cost Function converges to a small value close to 0. It is because we screen one matrix through the other one that we name it collaborative filtering. Besides, an important characteristic of our model is that only after finishing training can we know in which column those 6 features locates and this offer the relationship between features and the matrix.

The specific algorithm is:

Through calculation,we discover 2 keys to solving our model:

1 Find a way to measure the error between our model predictions and actual results.

2 Find a way to making appropriate updates MatCls and Matfea to diminish the error. We will elaborate on our solution below.

3.2.4 Cost Function

We define our Cost Function as:

$$J(MatCls, MatFea) = \frac{1}{2}\sum_{i=1}^{50}\sum_{j=1}^{578}(MatCls(j) \times MatFea(i)' - y(j,i))^2 \tag{4}$$

where MatCls and MatFea are 2 row vectors of size 1×6 which represent respectively a particular line of MatCls and MatFea. And y(i, j) is the element determined by j_th row and i_th column. Therefore, we seek to minimize J by finding proper MatCls and MatFea

3.2.5 Gradient Descent

We utilize the Gradient Descent Algorithm to updates MatCls and MatFea. As we have got J(MatCls, MatFea), we can calculate

$$\frac{\partial J}{\partial MatCls(j)} \tag{5}$$

where $j \in [1, 578]$. This partial derivative embodies the changes of J with regard to MatCls(j). According to necessary mathematical knowledge, we know that $-\frac{\partial J}{\partial MatCls(j)}$ points out the direction of variable MatCls(j) in which J changes fastest, i.e. we will get smaller J by updating in this direction. In the same way, we update MatFea(i) in the direction of $\frac{\partial J}{\partial MatFea(j)}$. The foregoing ideas is the criterion of our gradient descent update rule.

for $i = 1$ to 50, repeat

$$\begin{aligned} & MatFea(i) \\ & = MatFea(i) - \alpha \times \frac{\partial J}{\partial MatFea(i)} \\ & = MatFea(i) - \alpha \sum_{j=1}^{587} (MatCls(j)' \times MatFea(i) - y(j,i) \times MatCls(j)) \end{aligned} \tag{6}$$

for $j = 1$ to 587, repeat

$$\begin{aligned} & MatCls(j) \\ & = MatCls(j) - \alpha \times \frac{\partial J}{\partial MatCls(j)} \\ & = MatFea(i) - \alpha \sum_{i=1}^{587} (MatCls(j)' \times MatFea(i) - y(j,i) \times MatFea(i)) \end{aligned} \tag{7}$$

Note that α is the learning rate which reflect the speed of update, mainly selected from $[0.1, 0.5]$

3.2.6 Energy Profile

Part of our results are shown as follows:

ABICB	-0.0438546	0.3265896	-0.1366538	-0.1592492	0.10043425	0.1851342
ABICP	0.0780874	0.0085927	0.0838937	0.0633286	0.01408579	0.0476320
ARICB	-0.3380710	-0.0657101	0.3145247	0.2387681	-0.11742753	0.0140671
ARICD	0.0780863	0.0085914	0.0838946	0.0633297	0.0140840	0.0476316
ARICP	-0.337268	-0.0652387	0.3142687	0.2386853	-0.1171448	0.0141474

Table 2: Results obtained through matrix MatCls

1960	0.14728515	0.7563367	0.04618046	0.2225499	0.32128638	0.5477865
1969	0.23987086	0.28205155	-0.19065396	0.57915307	0.78527014	0.53975402
1979	0.47652343	0.2549734	0.7059976	0.43250481	0.74219859	0.87762573
1989	0.15113982	0.49484118	0.70056128	0.14042994	0.72416359	0.5118759
1999	0.83625332	0.49140292	0.75678847	0.38387578	0.92173087	0.58873047
2009	0.01927959	0.65149672	0.57028136	0.63088829	0.03090221	0.63899715

Table 3: Results obtained through matrix MatFea

On lateral comparison of MatCls, we find that those 6 features of each row correspond to NEPT, TEPT, TEC, TEP, NEC and NEP.

Based on our results of calculations, we plot energy profiles for 1960, 1969, 1979, 1989, 1999, 2009 of each state in Figure 2,3,4,5.

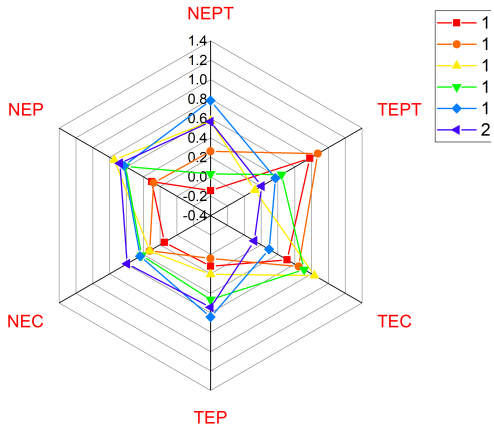

Figure 2: Energy Profile of AZ

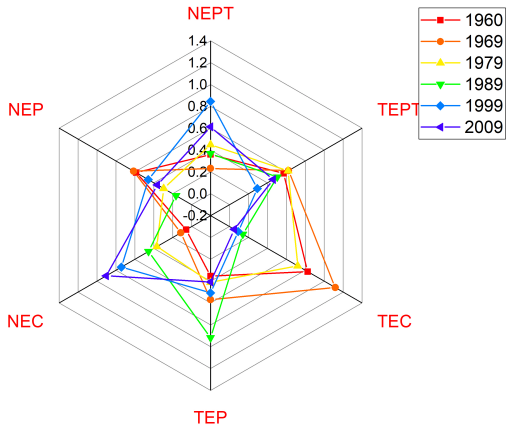

Figure 3: Energy Profile of CA

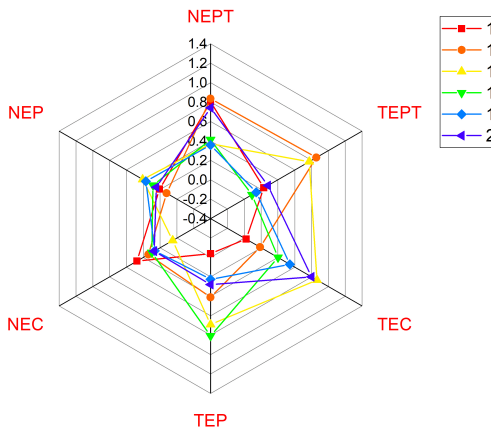

Figure 4: Energy Profile of NM

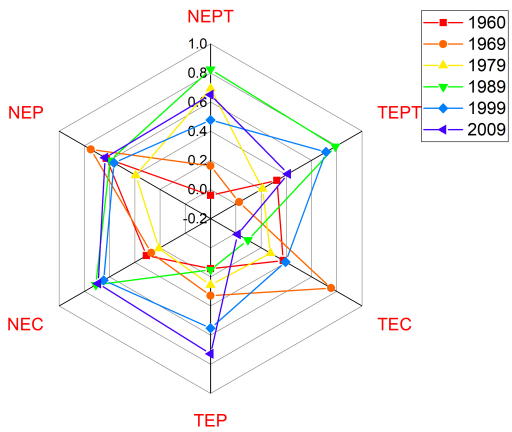

Figure 5: Energy Profile of TX

3.2.7 Energy Profile Criteria and Choice

Here we are using the assumption that new energy has a positive effect on the state's overall interests, for this reason we consider NEPT, NEC and NEP as positive factors for a good energy profile, while TEPT TEC and TEP as negative factors. We sum up the evaluation function as:

$$C = NEP + NEC + NEP - TEPT - TEC - TEP \tag{8}$$

On the basis of data in 2009, we obtain the score of each state:

State	NEPT	TEPT	TEC	NEP	NEC	TEP	C
CA	0.618	0.460	0.050	0.407	0.902	0.362	1.004
AZ	0.566	0.207	0.120	0.544	0.593	0.677	0.696
NM	0.740	0.281	0.799	0.679	0.27100	0.240	0.267
TX	0.649	0.414	0.015	0.728	0.691	0.636	1.002

Table 4: Scores of evaluation for each state

At last, we find that in 2009, CA has the best energy profile (1.004), followed by TX (1.002), AZ (0.696) and NM (0.267).

3.3 Discussion on Potential Factors

Beyond that, the images above indicates that the energy profile of Texas and California is better than the ones of the other two through comparison among their respective C-score. At the same time, we found that the energy profiles of these states have both similarities as well as distinctions. We list them along with some implications.

Similarities
1. The value of NEP generally has a tendence of decrease at first and then increase.
2. By and large, the score of N-energy increases while the score of T-energy decreases. This implies the general trend that the traditional energy is being replaced by the new energy.
3. In terms of numerical results, their profile score keeps increasing, which implies that the energy situation continues to improve.

Differences The differences mainly concentrate in New Mexico and Texas. We find that:
1. For New Mexico, on the contrary to the other three states, the value of NEC has a general downward trend, the value of NEPT is unexpectedly high in 1960, and the value of TEC has an upward trend. This proves that in New Mexico, there is more resistance to the development of new energy.
2. For Texas, unlike the other three states, the value of TEC increases at first and then decreases, and the value of NEP increases dramatically whereas the others changes within a narrow range. This proves that the development of new energy in Texas has presumably a relatively more promising future, which agrees with our research in Model 2.

3.4 Discussion of the potential factors

3.4.1 Geography [based on the assumption]

Geographic factors play an important role with regard to the energy profile. We analyse principally according to the distribution of the natural resources. In Figure 6, Based on the US energy distribution map, the geographic factors can be understood in the following way: the distribution of the traditional energy in Texas and California is rather deficient, while the other two states, especially New Mexico, have abundant natural resources. Traditional energy resources turns out to be a negative effect that the scarcity of resources in CA and TX motivates the development of new energy while developed traditional energy sources in inhibits the one of AZ and NM.

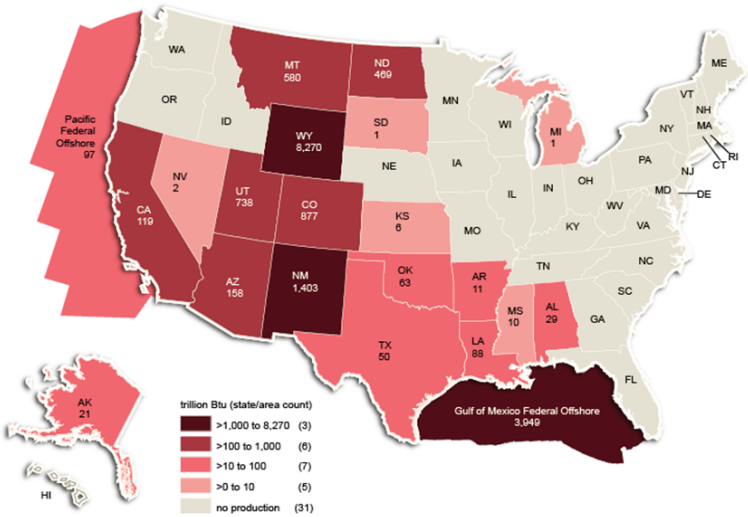

Figure 6: Distribution of National Natural Resources

3.4.2 Climate [based on the assumption]

CA: Mediterranean climate with abundant solar energy in summer.
TX: Temperate climate and summer rain warm climate with adequate wind energy resource.
AZ: Varied climates while mainly desert climate.
NM: Varied climates while mainly desert climate.

Boasting with an abundant resource of wind energy, Texas is the largest wind power state across the United States. In addition, close to the sea, California belongs to the Mediterranean climate characterized by abundant sunshine, which provides suitable conditions for photovoltaic energy development. In contrast, in AZ and NM the varied climates in which the desert climate forms the majority renders the development of new energy sources more challenging.

3.4.3 Population

Based on the "TPOPP" (Resident population including Armed Forces) data in "ProblemCData.xlsx", we plot in the Figure 7:

We find that, Texas and California have a larger population, which implies a larger need of energy. However, due to the rising cost and limited quantity of traditional energy, this has prompted people in both states to seek actively alternative energy solutions, thus triggered the development of new energy sources.

3.4.4 Industry

Based on "GDPRX" (Real gross domestic product) and "GDPRV" (Current-dollar gross domestic product) data in "ProblemCData.xlsx", we plot in the Figure 8, 9:

Based on the figures we find that California and Texas have well-developed economies and generally greater talent and technology advantages which reflect a stronger industrial capabilities. These factors facilitate the energy transformation, which accounts for

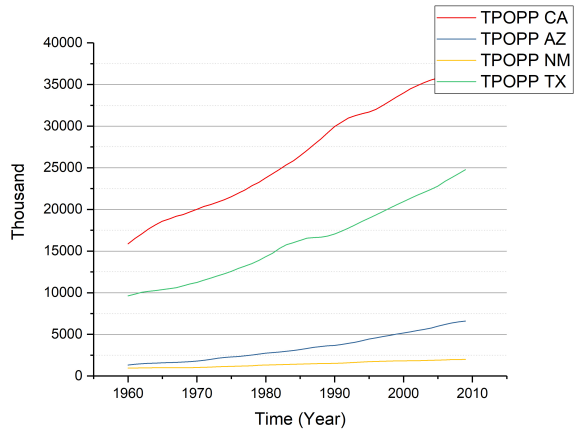

Figure 7: Resident population including Armed Forces of four States

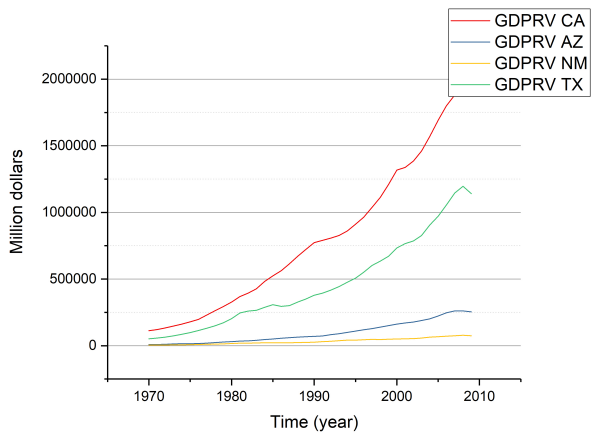

Figure 8: Current-dollar gross domestic product of four States

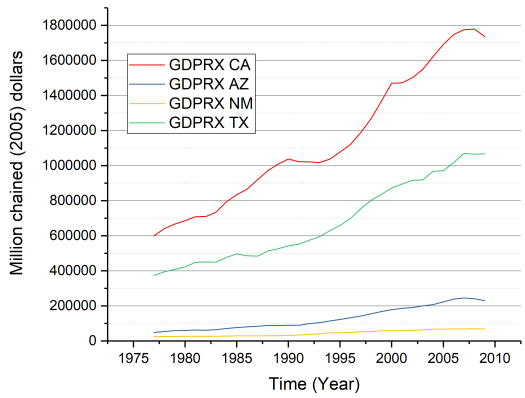

Figure 9: Real gross domestic product of four States

the better energy profiles of the 2 states.

3.5 Energy Usage Targets and Actions

3.5.1 Energy Usage Goals:

For California: California has the highest energy profile s core. H owever, i t doesn't have the best performance in the new energy field, the reason of which is that California has the best performance in eliminating the traditional energy. As a result, the goal of California in 2025 and 2050 is to improve all NEP, NEC, NEPT three features.

For Texas: Texas also receives a fairly high energy profile score, however it could further improve its energy profile by increasing NEPT and reducing TEP.

For Arizona: Arizona has obvious defect in terms of elimination of traditional energy, as a consequent it should set the goal to reduce the values of all TEP, TEC and TEPT.

For New Mexico : For New Mexico: New Mexico gets the lowest energy profile score and only the NEPT score is comparatively high, so its endeavors should focus on the other three features Based on our observation, we think it should increase NEP first, at the same time reduce TEC and TEP.

3.5.2 Actions

For California:
1. Improve more new energy production to increase NEC, which can be achieved for instance, by increasing the value of WYTCB (Electricity produced from wind energy).
2. Increase the values of features like NUETD (Nuclear fuel expenditures in the electric power sector) to increase NEC. For example, we can build more nuclear power plant.
3. Make the new energy more advantageous to increase NEPT. This can be achieved by lowering the price of new energy and increasing the price of traditional energy, NUETD (Nuclear fuel average price, all sectors), for example.

For Arizona:
1. Reduce values of some features like CLICP (coal consumption by the industrial sector) for decreasing TEC which can be achieved by shutting down more coal power plants.
2. Reduce values of some features like CLPRP (Coal production) for reducing TEP which can be achieved by shutting down more coal mines.
3. Increase values of some features like FSICD (Petrochemical feedstocks, still gas, price in the industrial sector) for reducing TEPT.

For New Mexico:
1. Increase values of some features like REPRB (Renewable energy production) in or-

der to increase NEP which can be achieved by building more solar power plant.
2. Reduce values of some features like PCCCB (Petroleum coke consumed by the commercial sector) in order to reduce TEC which can be achieved by vigorously promoting electric vehicles.
3. Reduce values of some features like PAPRB (Crude oil production including lease condensate) in order to reduce TEP which can be achieved by shutting down more oilfield.
4. Increase gasoline tax as it has the lowest tax rate among the 4 states.

For Texas:
1. Reduce values of some features like NUEGD (Nuclear fuel price in the electric power sector) in order to increase NEPT.
2. Reduce NGMPB (Natural gas marketed production) in order to reduce TEP, which can be achieved by shutting down more gas field.

3.5.3 Interstate New Energy Compact

For the sake of convenient transportation and resource balancing distribution, we pro-pose concluding a one-to-one contract where California shares energy with Arizona and Texas shares energy with New Mexico.

4 Model 2: A Time-Series-Based Exponential Smoothing Prediction Model (TSPM)

We model the energy profile for predicting the future.

4.1 Data Collecting

Here, we formulate the energy profile based on model 1. As before, we discuss simply the case in CA and other 3 states are predicted by the same idea.

4.2 Model TSPM

4.2.1 General Introduction

1. Time series refer to a data sequence, especially a sequence of data collected only from a period of time in which data in front indicates an earlier collection time.

2. Smoothing methods refer to a method to eliminate random fluctuations in time series which allows the series to be more smooth, thus reflect its basic trace and make predictions according to the model.

3. Weighting coefficient is used to measure how much previous data affects future data.

4. Exponential smoothing is a method where weighting coefficient is consistent with the exponential rule.

4.2.2 Basic Thinking

We think that our data obtained has a certain interdependence in the time domain and we invoke our 'predictable future assumption'. If so, we feel it is fair to think that in our time series = Rules (dependent) + Noise (ruleless, independent). We aim to find inherent rules and for this purpose to eliminate noise.

Back to our problem, since the energy profile of CA is predicted in accordance with 6 features: NEPT, TEPT, TEC, TEP, NEC and NEP, we build 6 different time series to find inherent rules between features and time.

4.2.3 Model Specification

Here we model the prediction of NEPT and the others are modelled the same way. Basic settings are:

- The known time series is y_{1960}, ..., y_{2009}
- $S_t^{(1)}$ and $S_t^{(2)}$ represents respectively the single and double exponential prediction of $(t+1)$ year.
- α is the weighting coefficient, $0 < \alpha < 1$

We have:

$$S_t^{(1)} = ay_t + (1-\alpha)S_{t-1} \tag{9}$$

This equation indicates that the future prediction is determined by both the nearest y_t to the data and all the previous data S_{t-1}. Besides, our choice is reflected by α.

Given its expansion equation:

$$S_t = a \times y_t + (1-\alpha) \times [\alpha y_{t-1} + (1-\alpha)S_{t-2}] = ...\alpha \sum_{j=0}^{\infty}(1-\alpha^j)y_{t-j} \tag{10}$$

we find that S_t (y_{t+1}) is the equal-Weighted historical data, with weighting coefficient = α, $\alpha(1-\alpha)$, $\alpha(1-\alpha)^2$... Clearly, $\sum_{j=0}^{\infty}\alpha(1-\alpha)^j = \frac{\alpha}{1-(1-\alpha)}$=1. It shows that our weighting coefficient meets the regulation of decreasing in a smooth way. Moreover, the fact that the older data has less impact on present fits our intuition exactly.

Whereas, one defect of single exponential smoothing is the significant lag when time series changes appear to be linear (i.e. the weighted coefficient prevents the linear trend from appearing in time). Therefore the double exponential smoothing is introduced to establish a linear trend model based on the law of deviation lag.

$$\begin{cases} S_t^{(1)} = & ay_t + (1-\alpha)S_{t-1}^{(1)} \\ S_t^{(2)} = & \alpha S_t^{(1)} + (1-\alpha)S_{t-1}^{(2)} \end{cases} \tag{11}$$

When y_t begins to appear a linear trend, we denote $N = \frac{1}{\alpha}$, then we predict:

$$\widehat{y}_{(t+T)} = a_t + b_t \times T,\ where\ T = 1, 2 \tag{12}$$

$$\begin{cases} a_t = & 2S_t^{(1)} - S_t^{(2)} \\ b_t = & \dfrac{2\alpha}{1-\alpha}(S_t^{(1)} - S_t^{(2)}) \end{cases} \tag{13}$$

$$a_t = y_t$$

$$y_{(t-q)} = y_t - b_t$$

$$\vdots$$

$$y_{(t-N+1)} = y_t - (N-1)b_t$$

$$S_t^{(1)} = \frac{y_t + y_{(t-1)} + ... + y_{(N-1)}}{N} = \frac{y_t + (y_t - b_t) + ... + [y_t - (N-1)b_t]}{N}$$

$$= \frac{Ny_t - [1 + 2 + ... + (N-1)]b_t}{N} = y_t - \frac{N-1}{2}b_t$$

$$y_t - S_t^{(1)} = \frac{N-1}{2}b_t$$

$$y_{t-1} - S_{t-1}^{(1)} = \frac{N-1}{2}b_t$$

$$y_t - y_{t-1} = S_t^{(1)} - S_{(t-1)}(1) = b_t$$

This way, $S_t^{(1)} - S_t^{(2)} = \frac{N-1}{2}b_t$

$$\begin{cases} a_t = & 2S_t^{(1)} - S_t^{(2)} \\ b_t = & \dfrac{2}{N-1}(S_t^{(1)} - S_t^{(2)}) \end{cases} \tag{14}$$

4.2.4 Determination of the Original Value

Given equation , we discover that it is important to select α and the original value noted S_0.

Selection of S_0 Considering that data in 1960 is far enough and thus has the minimal impact, we choose NEPT in 1960 as the original value.

Selection of alpha Here α=0.5. Other values of α will be discussed in sensitivity analysis.

α	2025	2050
0.5	1.457	2.829

Table 5: Prediction of CA when $\alpha = 0.5$

4.2.5 Prediction

We feed our data to our model and get predictions:

As for the other 3 states, results acquired by this model are shown in Figure 10, 11.

Now we can score all the 4 states in 2025 and 2050.

	CA	AZ	NM	TX
2025	5.1919	3.8302	0.5090	5.2355
2050	11.8904	8.7657	1.8976	11.7464

Table 6: Prediction of 4 states in 2025 and 2050

5 Sensitivity Analysis and Validation

5.1 Learning Rate

In the first model CFES, we have used gradient descent to seek the optimal solution of cost function. We have taken learning rate $\alpha_{learning} = 0.1$. Now, we will discuss the influence of selection of different learning rate on our cost function and result of the model.

We find that, when the learning rate is smaller, our cost function descends more slowly. it's more computationally expensive to achieve the optimal model. When the learning rate is larger, our cost function descends more quickly. However, when the learning reach a certain level, we will face the situation where our model is no more convergent.

This phenomenon is shown in Figure 12.

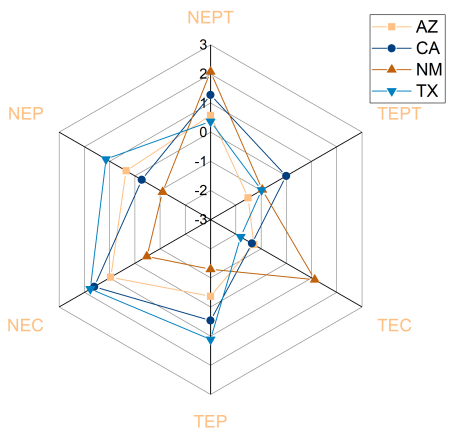

Figure 10: Prediction for 2025

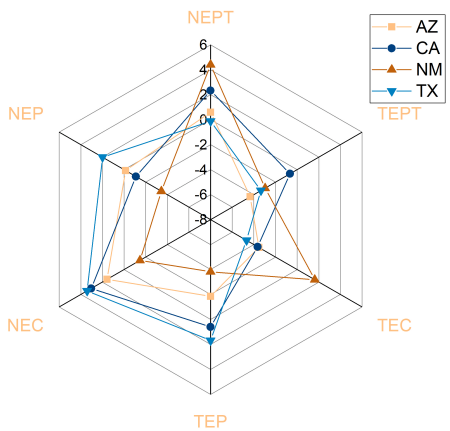

Figure 11: Prediction for 2050

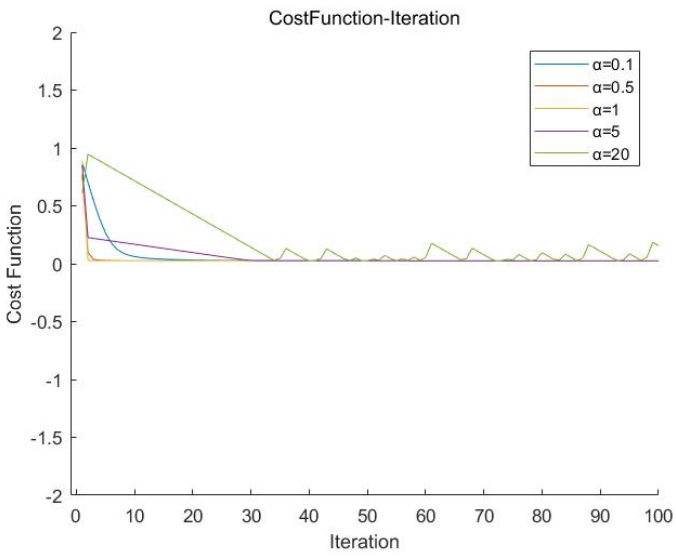

Figure 12: CostFunction-Iteration

Particularly, when choosing the learning rate as 0.1, our model could converge to a good level. So our model is valid.

5.2 weighting coefficient

In the second model TSPS, we choose the weighting coefficient $\alpha_{weight} = 0.5$. Now we discuss the influence of selection of different values of weighting coefficients on our model. The table below is the energy profile score of 2025 and 2050 with different α_{weight}

We find that the change of the energy profile score is less than 5%. In this way the validity of our model is proved.

Table 7: Prediction of Energy Profile in 2025

α_{weight}	CA	AZ	NM	TX
0.4	4.810	3.685	1.263	5.091
0.5	4.986	3.620	1.238	4.892
0.6	5.107	3.721	1.309	5.114

Table 8: Prediction of Energy Profile in 2050

α_{weight}	CA	AZ	NM	TX
0.4	10.810	7.686	3.659	11.094
0.5	11.372	8.344	3.887	11.017
0.6	11.402	8.272	3.921	11.215

6 Conclusion

6.1 Strengths

Accuracy of data: We have done a large-scale data processing, deleted data that we thought was useless, and completed the missing data using the K-means method.

Innovation of method: We learned from the collaborative filtering algorithm in machine learning and exponential smoothing in time series to construct mathematical models that can be quantified.

Feasibility of practice: We reviewed a large number of actual data (such as the US energy distribution map), and strived to better fit the actual situation of our model.

Visualization of result: We used methods like radar charts to visualize our data and result.

6.2 Weaknesses

Assumptions Simplification: We have made some assumption of simplification at the beginning of the essay, some of which cannot be realized in reality.

Risk of over-fitting: We only had data of 50 years. Our model has the risk of over-fitting. That is to say, our model fits well with the data of these 50 ye ars. However the data other than these years might not be accurately predicted.

Linearity of prediction: We have applied a time-series-based exponential smoothing model and finally the linear analysis. However, linear analysis may not fit well with the actual situation.

References

[1] Tom M.Mitchell, Machine Learning, 1st ed, 2003, 200:219.

[2] James Douglas Hamilton, Time Series Analysis, 1st ed, 1994, 155:180

[3] Fossil Fuel Production Map Us Fossil Fuels Map, http://cdoovision.com/fossil-fuel-production-map-us/fossil-fuel-production-map-us-fossil-fuels-map/, 2018.2.10

[4] California,https: //en.wikipedia.org/wiki/California, 2018.2.10

[5] Arizona, https: //en.wikipedia.org/wiki/Arizona, 2018.2.10

[6] New Mexico, https: //en.wikipedia.org/wiki/New_Mexico, 2018.2.10

[7] Texas, https://en.wikipedia.org/wiki/Texas, 2018.2.10

[8] Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan, Collaborative Filtering Recommender Systems, http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf, 2018.2.11

[9] DANIEL PREVE, Essays on Time Series Analysis With Applications to Financial Econometrics, https://www.diva-portal.org/smash/get/diva2:171806/FULLTEXT01.pdf, 2018.2.11

[10] T. E. Bearden, Rapid and Decisive Solution of the World Energy Crisis and Global Warming, http://siriusdisclosure.com/wp-content/uploads/2013/03/BeardenSolution.pdf, 2018.2.11

[11] Dr. Theodore C. Loder, Outside-the-Box Technologies, Their Critical Role Concerning Environmental Trends, and the Unnecessary Energy Crisis, http://siriusdisclosure.com/wp-content/uploads/2012/12/EPWoutofboxBrief.pdf, 2018.2.11

[12] USA Energy Facts, https://www.eia.gov/energyexplained/?page=us_energy_home, 2018.2.11

[13] Rankings: Total Energy Consumed per Capita in 2015, https://www.eia.gov/state/rankings/?sid=NY, 2018.2.12

Appendices

Here are all the programs we used in our paper.

Appendix A Generation of Numpy matrixes

Here is our python program for generating numpy matrixes of data.

Input python source:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 13:55:24 2018
"""
from data import data
from check import get_data_usless,get_dict_usless
import pandas as pd
import numpy as np
class DataGen():


    def __init__(self):
        self.cdata=data()[0]
        self.cdict=data()[1]
        self.data_usless=get_data_usless()
        self.dict_usless=get_dict_usless()
        self.idx_data_list=[]
        self.idx_dict_list=[]
    def get_data_idx(self):
        self.idx_data_list=[]
        for usless in self.data_usless:
            for idx,name in enumerate(self.cdata['MSN']):
                if name==usless:
                    self.idx_data_list.append(idx)
    def get_dict_idx(self):
        self.idx_dict_list=[]
        for usless in self.dict_usless:
            for idx,name in enumerate(self.cdict['MSN']):
                if name==usless:
                    self.idx_dict_list.append(idx)
    def get_city_mat(self,city):
        mat=np.empty([50,self.cdict.shape[0]])
        for year_idx in range(50):
            for var_idx in range(self.cdict.shape[0]):
                for name_idx,name in enumerate(self.cdata['MSN']):
                    if name in np.array(self.cdict['MSN']):
                        for dict_idx,dic in enumerate(self.cdict['MSN']):
                            if dic==name and self.cdata['Year'][name_idx]
                                ==year_idx+1960

                                mat[year_idx,var_idx]=self.cdata['Data'][name_idx]

                    else:
                        mat[year_idx,var_idx]=np.nan
        return mat
    def main(self):
        self.get_data_idx()
```

```
        self.get_dict_idx()
        self.cdata=self.cdata.drop(self.idx_data_list)
        self.cdata=self.cdata.reset_index(drop=True)
        self.cdict=self.cdict.drop(self.idx_dict_list)
        self.cdict=self.cdict.reset_index(drop=True)
        pd.DataFrame(self.get_city_mat('AZ')).to_csv('az.csv')
        pd.DataFrame(self.get_city_mat('CA')).to_csv('ca.csv')
        pd.DataFrame(self.get_city_mat('NM')).to_csv('nm.csv')
        pd.DataFrame(self.get_city_mat('TX')).to_csv('tx.csv')

if __name__=='__main__':
    dg=DataGen()
    dg.main()
    #print(dg.cdata.shape)
    #print(len(dg.cdata['MSN']))
    #print(dg.cdict.info())
    #print(dg.cdict)
```

Appendix B Data Selecting

Here is our python program for selecting the garbage data.

Input python source:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 11:32:06 2018
"""
import numpy as np
import pandas as pd
def get_dict_usless():


    cdata=pd.read_csv('cdata.csv')
    cdict=pd.read_csv('cdict.csv')
    lis=[]
    ar_dict=np.array(cdata['MSN'])
    for msn in cdict['MSN']:
        if msn not in ar_dict :
            if msn not in lis:
                lis.append(msn)
    return lis
def get_data_usless():
    cdata=pd.read_csv('cdata.csv')
    cdict=pd.read_csv('cdict.csv')
    lis=[]
    ar_dict=np.array(cdict['MSN'])
    for msn in cdata['MSN']:
        if msn not in ar_dict :
            if msn not in lis:
                lis.append(msn)
    return lis


if __name__=='__main__':
```

```
    print(get_data_usless())
    print(get_dict_usless())


#print(cdict['MSN'])
#for msn in cdata['MSN']:
```

Appendix C K-means and Z-score

Here is our python program for performing k-means and z-score

Input python source:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 16:27:26 2018
"""
import pandas as pd
import numpy as np
import check
from data import data
from sklearn.cluster import KMeans
from sklearn.preprocessing import normalize
cdict=data()[1]


az=pd.read_csv('AZ.csv',header=None)
ca=pd.read_csv('CA.csv',header=None)
nm=pd.read_csv('NM.csv',header=None)
tx=pd.read_csv('TX.csv',header=None)
def kmeans(x):
    kmeans=KMeans(n_clusters=4).fit(x)
    label=kmeans.labels_
    return label
def to_norm(npary):
    nan_list=np.where(np.isnan(npary))
    temp=np.array(npary)
    for i in nan_list[0]:
        for j in nan_list[1]:
            temp[i,j]=np.mean(temp[i,:])
    labels={}
    for i in np.random.randint(500,size=5):
        label=[]
        label=kmeans(npary.drop(i))
        labels['%s'%i]=label
    return npary
idx_list=[]
dict_usless=check.get_dict_usless()
for usless in dict_usless:
    for idx,name in enumerate(cdict['MSN']):
        if name==usless:
            idx_list.append(idx)
idx_list.append(26)
az=az.drop(idx_list)
```

```
ca=ca.drop(idx_list)
nm=nm.drop(idx_list)
tx=tx.drop(idx_list)
az=az.reset_index(drop=True)
ca=ca.reset_index(drop=True)
nm=nm.reset_index(drop=True)
tx=tx.reset_index(drop=True)


az=to_norm(tx)
'''
to_norm(ca)
to_norm(nm)
to_norm(tx)
'''
norm=normalize(az)
print(norm.shape)
#pd.DataFrame(norm).to_csv('tx.csv')
```

Appendix D Collaborative Filtering Algorithm

Here is our python program for performing collaborative filtering algorithm.

Input python source:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 20:54:48 2018
"""
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
class Cf():


    def __init__(self,lbd=0.1,itr=100,cls=6,alpha=0.1):
        self.lbd=lbd
        self.itr=itr
        self.cls=cls
        self.alpha=alpha
        self.cost=np.zeros(itr)
        self.year=[]
        self.feature=[]
    def fit(self,x):
        self.year=np.random.rand(x.shape[1],self.cls)
        self.feature=np.random.rand(x.shape[0],self.cls)
        for i in range(self.itr):
            self.cost[i]=self.cost_fun(self.year,self.feature,x)/10000
            for ny in range(self.year.shape[0]):

                for nf in range(self.feature.shape[1]):

                    self.year[ny,:]-=self.alpha*np.dot(np.dot(self.feature[nf,:].transpos
                    self.feature[nf,:]-=self.alpha*np.dot(np.dot(self.feature[nf,:].trans
```

```
    def cost_fun(self,year,feature,x):
        cost=1/2*sum(sum((np.dot(feature,year.transpose())-x)**2))
        return cost

if __name__=='__main__':
    name='tx'
    ct=pd.read_csv('%s.csv'%name)
    ct_np=np.array(ct)[:,1:51]
    ct_np=np.delete(ct_np,[60,90,450],0)
    times=np.arange(100,1000,300)
    for time in times:
        cf=Cf(itr=time)
        cf.fit(ct_np)
        plt.plot(np.arange(time),cf.cost)
    plt.show()
```

Appendix E Exponential Smoothing

Here is our matlab program for performing exponential smoothing method.

Input matlab source:

```
function [A,B] = Fabian(m)
sizem = size(m);
A = zeros(1,sizem(1,1));
B = zeros(1,sizem(1,1));
for j = 1:sizem(1,1)
y=m(j ,1:end); n=50;

alpha=0.4; st1(1)=y(1); st2(1)=y(1);
for i = 2:n
st1(i)=alpha*y(i)+(1-alpha)*st1(i-1);
st2(i)=alpha*st1(i)+(1-alpha)*st2(i-1);
end

A(1,j)=2*st1(n)-st2(n);
B(1,j)=alpha/(1-alpha)*(st1(n)-st2(n));

end
end
```