TIM 245 - Data Mining: Final

Due: *June 15, 2017*

Instructor: Tyler Munger

Panos Karagiannis
ID: -

Contents

Problem 1: Classification Problem for Venture Capital Investment (30 points)

The firm has assembled a dataset of the historical investments that includes both the initial assessment of the company and final the investment rating (Green, Yellow, Red). You have been asked to perform the following tasks related to creating a classification model that be used to determine if a new company is likely to be a Green, Yellow, or Red investment based on their assessment.

1. Perform your EDA process, developed during your work at Xenefits, on the investment dataset and format the results into a well-structured report. Perform any data pre-processing steps (cleaning, transformation, etc.) necessary for addressing data quality issues that were discovered during the EDA process.

2. Before you start, the firm would like a written statement of your process for creating the classification model. (Hint: The process might include the steps such as experimenting with different learning algorithms or model evaluation).

3. Next, apply your process to create the classification model.

4. Based on your process, what is the best classification model for predicting an investment rating? What is your assessment of this classification model? Is the performance good enough for the firm to use when making investment decisions? What are some of the issues associated with equally weighting misclassification across the three classes (hint: is a Red misclassified as a Green the same as a Yellow misclassified as a Green?). Describe one possible solution for addressing this problem

Extra Credit:
Implement your solution for addressing the misclassification problem. Compare and contrast the results.

Answer:

1. Real world data are often **Incomplete**, **Noisy** as well as **Inconsistent**. Our *Exploratory Data Analysis* (EDA) consists of a combination of **visual** and **quantitative** tools to answer important questions for each selected attribute in our dataset. More precisely, since every attribute is categorical we will look into descriptive statistics such as the **mode** and the **frequency**. Moreover, we will employ visual tools in order to determine whether a particular distribution is **unimodal**, **bimodal** or **multimodal**.

   By looking at Table 5 we see that there is large variety of combinations of *missing, low, medium* and *high* values. More precisely, we observe attributes that have more *missing* than actual values (ex. `INO, EFF, PPL`), and we expect this fact to undermine the quality of our classification algorithms. Moreover, the majority of the distributions are *bimodal* or *multimodal* with very few *unimodal* ones. We expect to face difficulties in learning very skewed distributions correctly, since there will be insufficient data to train (i.e. `GRO` has only 1 *high* value). Finally, in the `rating` attribute, we see that the majority of values are `red`, followed by `green` and `yellow`.

In order to use the data in our classification models, we first encode all the categorical variables as numerical. More precisely, since each attribute has 3 values (ignoring missing), we create 3 new **binary** variables for each attribute.

Initially, we thought of deleting the **missing** values, in order to perform classification but that yielded a dataset with only 8 instances, clearly insufficient to perform learning. Instead, we replace the missing values of a given attribute with the **mode** value for this attribute.

Lastly, since our dataset consists of values between 0 and 1 we do not need to **normalize** the data or remove **outliers**. Also, the dataset does not contain any **inconsistencies**.

2. At this stage, we will experiment with various classification models in order to determine which one yields the best results. First we will create a baseline model, which will be very simple and interpretable and is going to provide us with a benchmark to compare the rest of our models. In order to evaluate our models we will consider two metrics, namely, **accuracy** and **F-measure**.

   The baseline model that we use, is the **ZeroR** model, whereas, the more elaborate models that we consider are: **SVM, Logistic Regression, KNN**.

   In terms of **complexity**, *SVM and Logistic Regression* models, behave very similarly in this problem. These models minimize a convex loss functions and hence any local minimum is also guaranteed to be a global minimum as well. As far as KNN is concerned, it does not minimize a loss function, yet, the computation of the "distance" between data points can be very expensive if many features are incorporated into the model.

   It has to be mentioned, that all these models have similar complexities for relatively small number of attributes, as it is in the case of $Xenefits$. **Nevertheless**, if the management wanted to incorporate more attributes into the model then the complexities of *Logistic Regression and KNN* would explode, whereas, *SVM* can tackle this problem. In *SVM*, we can proceed by solving the *Dual problem* instead of the *Primal* and using a *Kernel function* to retrieve the weights (a detailed explanation of this fact was submitted in HW2 when comparing Logistic Regression and SVM).

   As far as **performance** is concerned, KNN tend to overfit the data more easily than the other two methods. Therefore, in a KNN model we should carefully decide on the parameter K. Similarly in an SVM model as we increase the parameter C we decrease the tolerance for misclassification but if we increase too much there is the danger of overfitting the data ($C \gg 0$ is like a hard margin). Also, *Logistic Regression* models can overfit but we can control the size of the coefficient of the model by introducing

regularization.

Finally, in terms of **interpretability** the simplest model is KNN since no function minimization is involved. *SVM and Logistic Regression* are more complex models which try to maximize the *margin* and model the class conditional probabilities, respectively. *Logistic Regression* is always non-linear, whereas, in SVM a non-linear Kernel function is frequently used. In the following parts we use the **Normalized Polynomial Kernel** for our SVM model.

3. In Table 1 we see the performance of the aforementioned models:

Table 1: Performance of classification models for determining the `rating`. R., Y., G. stand for Red, Yellow, Green, respectively

	Accuracy(%)	F-measure R.	F-measure Y.	F-measure G.	Weighted F-measure
SVM (C=1)	89.21	0.94	0.73	0.87	**0.89**
Logistic Regression	76.47	0.86	0.55	0.71	0.77
KNN (K=1)	86.27	0.9	0.75	0.86	0.87
ZeroR	52.94	0.69	0	0	0.37

4. Simply by looking at the above table, we see that the more elaborate models outperform the baseline ($ZeroR$) both in terms of accuracy and F-measure. We see that the best performing model is **SVM** with **KNN** performing similarly.

Nevertheless, we need to be particularly careful with the assessment metrics we use. Since there are large class imbalances (53% red, 29% yellow, 18% green), the accuracy metric is not very useful because many models learn to predict the majority of class most of the times. Instead, to obtain a better evaluation, we look at the confusion matrix (Table 2) and the F-score computed in Table 1.

Table 2: Confusion matrix for SVM

		Classified as:		
		red	**yellow**	**green**
Actual Value	**red**	51	1	2
	yellow	2	11	5
	green	1	0	29

By observing the confusion matrix, we see that the SVM model has managed to partially

limit the effect of the unbalanced classes. Nevertheless, we observe that the model tends to misclassify *yellow* much more that it tends to misclassify *red*. More precisely, the **effect of the imbalances classes** is seen in the model, when it misclassifies 18%(= 5/18) of the yellow values whereas it only misclassifies 3% = 2/54 of the total red values.

Finally, since the company needs to make crucial investment decision, where many millions of dollars are involved, we would say that the results of the model are **still not very good** to allow the company to be profitable.

5. As explained in (4), one of the most important problems with class imbalances is that the model tends to predict the majority class more often than the other classes, and hence obtains a good accuracy score. For example, the accuracy of the SVM model is 89%, nevertheless, by looking at the confusion matrix the model misclassifies 18%(= 5/18) of the yellow values. A possible solution to that problem would be to **under-resample the dataset**, by deleting instances from the over-represented classes. This approach is easy to implement and fast to run, hence, it constitutes a good starting point.

Extra Credit:
In order to eliminate the problem created by the imbalanced classes, we randomly sample 40% and 70% of the initial red and green values, respectively. Thus, we are left with a total of 61 instances out of which 22 have red rating, 21 have green rating and 18 have yellow rating.

Applying an SVM model to our data we get:

Table 3: Performance of the SVM model when trained on balanced classes

	Accuracy(%)	F-measure R.	F-measure Y.	F-measure G.	Weighted F-measure
SVM (C=1)	82	0.87	0.79	0.80	0.82

Table 4: Confusion matrix for SVM

		Classified as:		
		red	**yellow**	**green**
Actual Value	**red**	17	1	5
	yellow	0	13	5
	green	0	1	20

By looking at Table 3 we see that the model performs worse than before in both accuracy and F-measure, but this can be attributed to the fact that we are now using 40% less data

to train our classifier.

What is important to observe, is that now the proportion of red ratings misclassified as green is 18%(= 4/22), whereas the proportion of yellow ratings misclassified as green is 27%(= 5/18). The difference now is **much smaller** than previously, since our classifier has learned to "balance" the classes.

Appendix for Problem 1

EFF	count
Missing	61
low	40
medium	1

(a)

GRO	count
high	1
low	87
medium	14

(b)

MKT	count
low	83
medium	19

(c)

EXP	count
low	94
medium	8

(d)

CPL	count
high	20
low	22
medium	60

(e)

INO	count
Missing	59
low	36
medium	7

(f)

TCH	count
Missing	25
low	50
medium	27

(g)

JNT	count
low	96
medium	6

(h)

PTN	count
low	74
medium	28

(i)

MAC	count
high	35
low	43
medium	24

(j)

VPR	count
high	30
low	25
medium	47

(k)

PPL	count
Missing	31
high	8
low	34
medium	29

(l)

SCH	count
Missing	11
high	40
low	38
medium	13

(m)

PRT	count
high	59
low	24
medium	19

(n)

MGR	count
high	27
low	29
medium	46

(o)

SOM	count
Missing	6
low	96

(p)

MXS	count
high	84
low	7
medium	11

(q)

CMP	count
high	17
low	66
medium	19

(r)

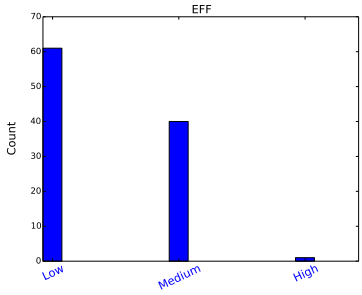
MDV	count
high	14
low	65
medium	23

(s)

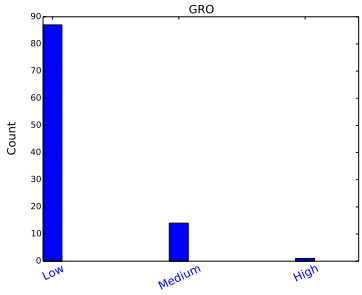
rating	count
green	30
red	54
yellow	18

(t)

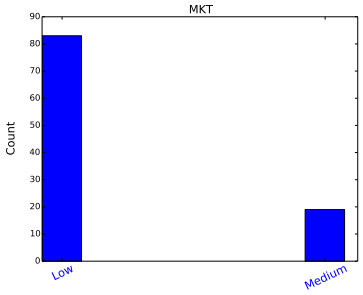
Table 5: Frequency Tables for all attributes in the dataset.



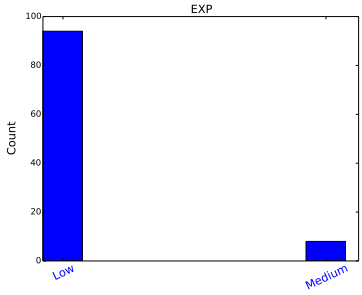
(a)



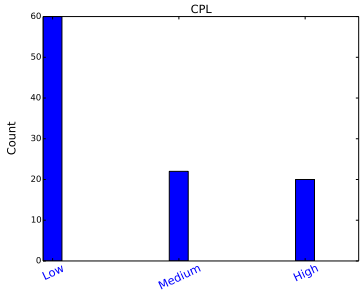
(b)



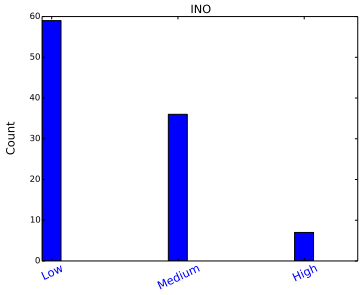
(c)



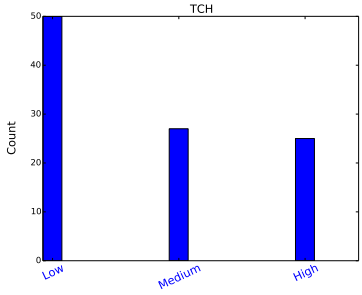
(d)



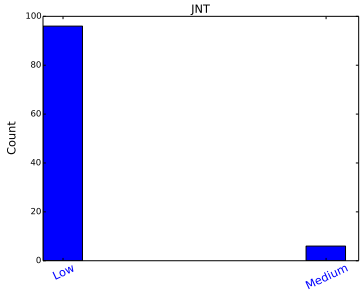
(e)



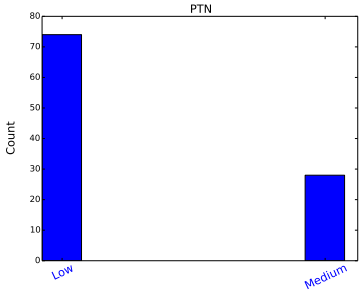
(f)



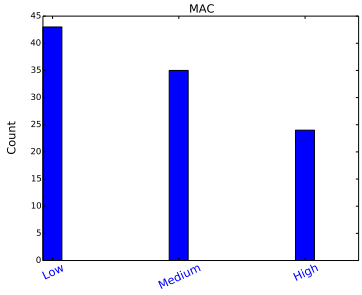
(g)



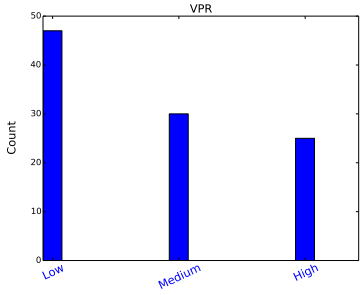
(h)



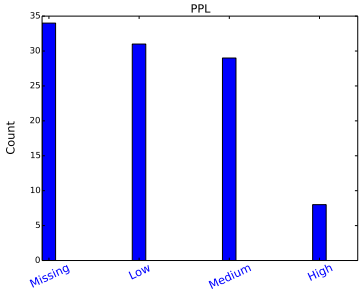
(i)



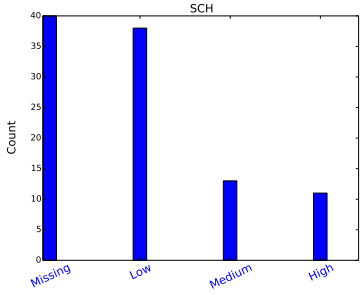
(j)



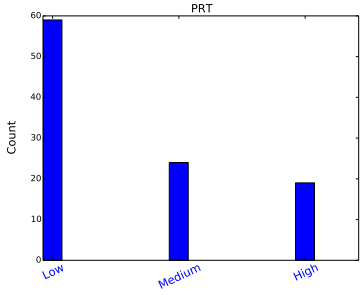
(k)



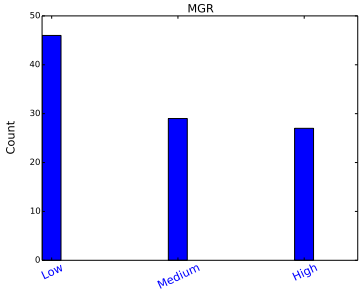
(l)



(m)



(n)



(o)

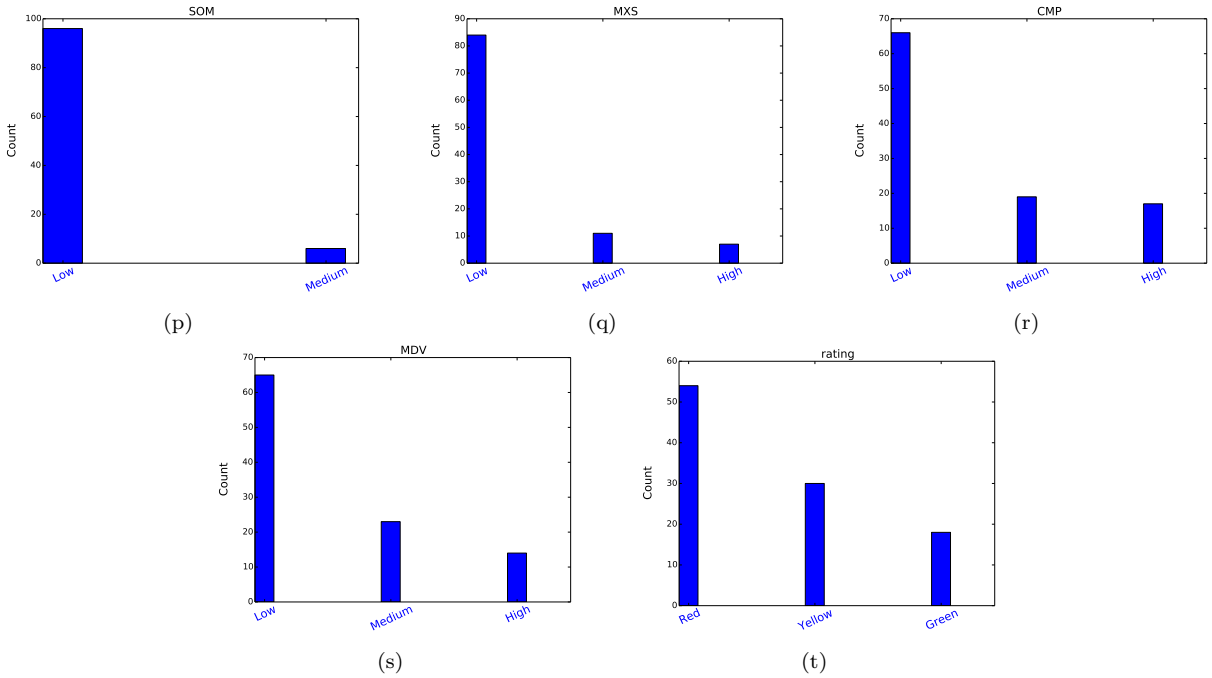


Figure 0: Frequency Tables for all attributes in the dataset.

Problem 2: Association Analysis of Successful Companies (20 points)

Based on the results of the investment classifier, the firm would like to understand if there are any association rules for the attributes of a successful company. For example, are successful companies that have a good leadership team more likely to have better market positioning? These rules would allow them to better understand patterns across their investments and make recommendations for improvement.
To generate these rules, you have been asked to perform the following tasks:

1. Create a subset of the original investment dataset that only contains the successful companies, i.e. green label.
2. Explain the difference between support, confidence, and lift. How will changing the support and confidence change the discovered patterns?
3. Experiment with different values for support and confidence. What threshold do you recommend using for support and confidence?
4. Identify 2-5 interesting rules generated using your selected support and confidence threshold. What is the interpretation of the rule? What is underlying rationale or reason for the rule, e.g. the diapers -¿ beer rule was because young fathers were sent to the store to buy diapers.
5. What are some other potential applications of the generated rules? For example, would these rules be useful to the founder of a start-up company?

Extra Credit:
Repeat the rule generation process using only the unsuccessful companies, i.e. Red label. Compare and contrast the results to the association rules for the successful companies.

Answer:

1. We can use Python and the `pandas` library in order to create a dataset that contains only the *green* labeled companies.
2. Let X, Y be two disjoint itemsets. Then we measure the strength of the association rule $X \to Y$, by using metrics such as **support, confidence** and **lift**. More precisely, the support is defined as:

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{N}$$

,and measures the **fraction of transactions that contain both X and Y**

We define the confidence of the rule to be:

$$c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

,which measures **how frequently items included in Y appear in rules that contain X.**

Finally, lift is defined as:

$$Lift = \frac{c(X \to Y)}{s(Y)} = \frac{Ns(X \cup Y)/\sigma(X)}{s(Y)} = \frac{s(X \cup Y)}{s(Y)\sigma(X)/N} = \frac{s(X \cup Y)}{s(Y)s(X)}$$

, which can be thought of as a ratio of **"actual"/"expected"**, thus we would like to pick rules for which $Lift \gg 1$. Also, if $Lift = 1 \implies s(X \cup Y) = s(Y)s(X)$ which implies statistical independence, therefore we would like to choose rules for which lift as large as possible.

Since we are using the **Apriori Algorithm** to find interesting association rules, setting the support to a small number (close to 0) will result in a much larger set of candidate rules to be searched. Moreover, as we decrease the support, the produced rules will tend to be more infrequent, thus we could come across rules that **occurred by chance**.

As we decrease the confidence (values close to 0), we will produce rules which are not very **reliable**. More precisely, the higher the confidence, the more likely it is for Y to be present in transactions that contain X.

3. We find the best threshold for the support and confidence heuristically, therefore we need to experiment with many different values. Using `R` we produce Figure 1 where we see that there exist no rules which have support less than 0.7 and greater than 0.13. Also, we observe that the highest lift value (≈ 15) is obtained for a small support and very large confidence. As a result, the threshold that we would recommend would be $s \approx 0.07$ and $c \approx 0.98$.

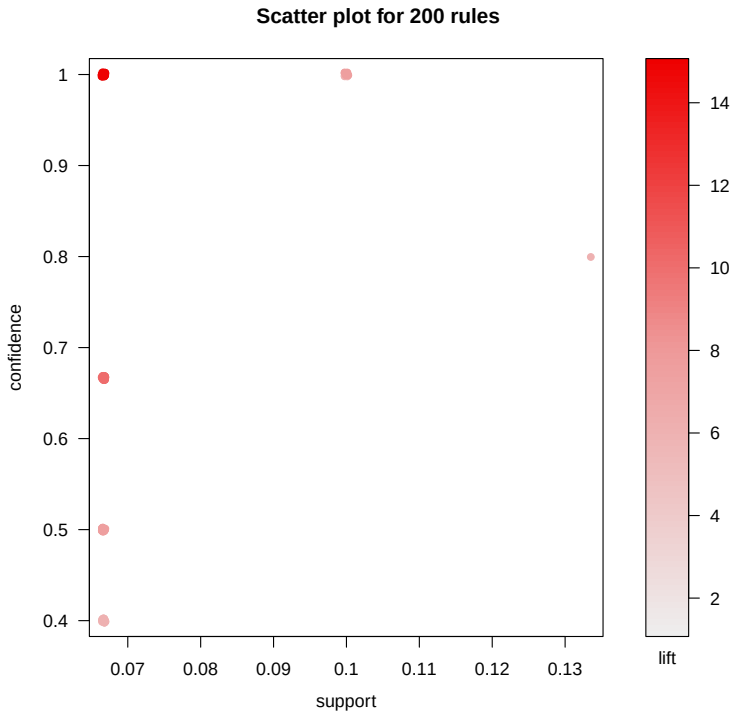


Figure 1: Scatter plot displaying the support, confidence and lift of 200 rules for the **green** labeled companies.

4. We present three interesting rules along with their interpretation in the following list:

   (a) $\{CMP = low, GRO = medium, SCH = high\} \Longrightarrow \{TCH = medium\}$:

   This rule achieves a support, confidence and lift equal to $s = 0.067, c = 1, lift = 15$. It implies that within the successful startup companies, it is observed that in a non competitive landscape (*low*), with *medium* ability to react to risk and a *high* quality of supply chain integration, a company often possesses of *medium* quality of technical team.

   (b) $\{SCH = low, CMP = medium\} \Longrightarrow \{VPR = high\}$:

   This rule achieves a support, confidence and lift equal to $s = 0.067, c = 1, lift = 7.5$. It implies that with a *low* quality of supply chain integration and *medium* level of competitive landscape, a successful company can obtain a *high* value position.

   (c) $\{EXP = low, JNT = low, CPL = medium, MAC = high, SCH = high\} \Longrightarrow \{INO = medium\}$:

   This is the lengthiest rule and it achieves a support, confidence and lift equal to $s = 0.1, c = 1, lift = 7.5$. It implies that with *low* leadership and joint team experience, *medium* levels of team completeness and *high* levels of market acceptance and supply chain integration, a succesful company often manages to have a *medium* ability to innovate.

5. These rules could have many different applications. First, the founder of a startup company would know the "recipe" for a successful company. For instance, by observing the aforementioned rules, a company would know that in order to obtain a high value position, it would need to first have a medium level of competitive landscape. Moreover, these results could also be of great value for *Redwood Capital*, since it can more easily identify successful companies by examining if they satisfy common rules that other successful companies satisfy.

Extra Credit:

In order to repeat the same procedure for the unsuccessful companies we first need to find the appropriate threshold values for the support and confidence. Using Figure 2 we see that the distribution of rules with large lift values is different from before. We get many rules with high lift values as we vary the support between $0.03 \leq s \leq 0.04$ and we keep the confidence fixed to approximately 1. As a result, we focus on rules that have $s \approx 0.035$ and $c \approx 1$.

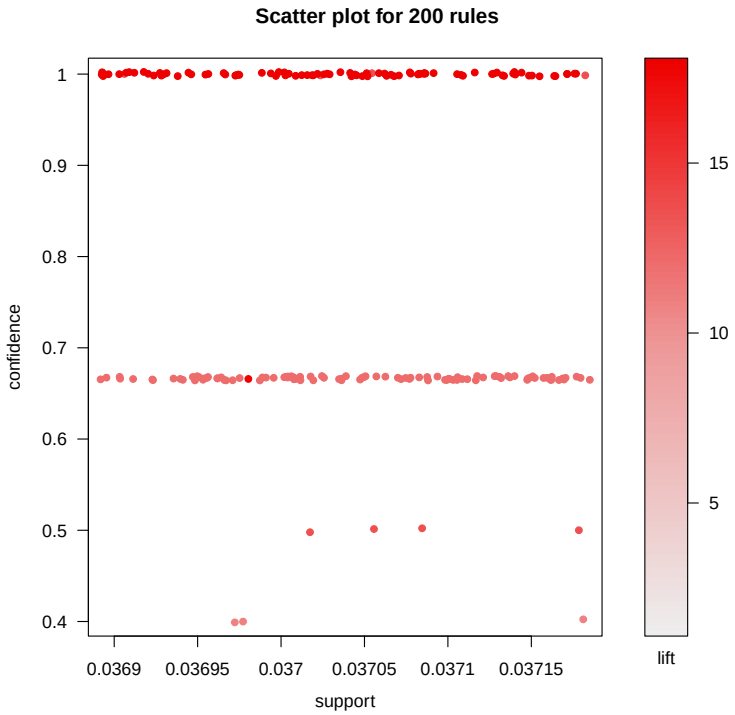


Figure 2: Scatter plot displaying the support, confidence and lift of 200 rules for the **red** labeled companies.

Some interesting rules that we see are the following:

1. $\{MDV = low, MKT = low, MAC = low\} \Longrightarrow \{CMP = high\}$:

   This rule achieves a lift score of 18 and demonstrates an common reason why many startup companies fail. If a company has minimal market experience, market acceptance as well as market diversity and functions in a very competitive landscape, it is also very likely to go bankrupt.

2. $\{MDV = low, MKT = low, GRO = medium\} \Longrightarrow \{EXP = medium\}$:

   This rule also achieves a lift score of 18. It shows that in unsuccessful companies, very often, low market experience in combination with low market diversity and inability to evaluate risk well, occur together with poor leadership experience. Without leadership and correct decision making, a company is very likely to go bankrupt, therefore it is labeled as "red".

3. $\{PTN = low, GRO = medium\} \Longrightarrow \{CMP = high\}$:

   This rule also achieves a lift score of 13 and is similar to the first rule. It describes that without intellectual product rights and a medium ability to evaluate risks in a competitive landscape, a start up company can often fail.

In general, comparing the rules generated for both datasets, we see **patterns** that successful (green) and unsuccessful (red) startup companies follow. We see that the ability to **innovate** and have a good **technical team** can often lead companies to success. Nevertheless, we observe that a **competitive landscape environment** can lead to failure for a newly formed company.

Problem 3: Clustering Crowdfunded Projects on Kickstarter (40 Points)

Crowdfunding is an alternative source of funding for start-up companies where small investments, e.g. 100$, are made by thousands of independent investors. These investments are typically managed through a crowd-funding portal, such as Kickstarter and Indie-Go-Go, where companies, or individuals, can post projects that they would like people to fund.

The crowdfunding market is potentially a new emerging investment opportunity for Redwood Capital. In addition to making several multi-million dollar investments every year, they could make small investments in hundreds of companies to diversify their portfolio and reduce risk. However, not all of the projects on crowd-funding sites are good candidates for commercialization. For example, projects often include humanitarian projects and art projects which would not be suitable for profit motivated investment firm like Redwood Capital.

The firm would like to further investigate these crowd-funding sites to see what kinds of projects are being funded and if they are a viable investment opportunity. To this end, they have collected a dataset of 4,000 projects from the popular crowd-funding site Kickstarter.
The dataset contains the following attributes:

- Project Title
- Description
- Amount Funded
- Project Goal
- Project Category
- Type of Currency
- Location
- Number of Backers

You have been asked to perform the following tasks related to understanding the different groups of projects that are being funded on Kickstarter:

1. First, remove the Project Title and Description fields from the dataset. Perform your EDA process on the remaining attributes and format the results into a well-structured report. Perform any data pre-processing steps (cleaning, transformation, etc.) necessary for addressing data quality issues that were discovered during the EDA process.

2. The firm has asked you to explain the differences between the following popular clustering algorithms: K-Means, DB-Scan, Hierarchical Clustering, and Mixture Models (EM). In particular, how do these algorithms with respect to computational complexity, types of clusters that they can find, and interpretability?

3. Before you start, the firm would like a written statement of your process for cluster analysis. (Hint: The process might include the steps such as visualizing the data or reviewing clusters with domain experts)

4. Then, they would like you to apply your cluster analysis process to the collected data-set and format the results into a well-structured report.

5. Identify 1-2 clusters of projects that would be good candidates for the firm to investigate further. Provide a brief summary description for each candidate cluster, e.g. video game projects developed in San Jose.

Extra Credit:
Create an additional set of attributes based on the project description and repeat your process for cluster analysis.

Answer:

1. We follow a similar approach to Problem 1 in order to analyze the categorical variables `location`, `category` and `currency` . As far as the numerical variables are concerned, our Exploratory Data Analysis will consist of a combination of **visual** and **quantitative** tools to answer important questions for each selected attribute in our dataset. More precisely, for each attribute we will look into descriptive statistics such as the **typical value**, the **spread** for a typical value and whether it affects other attributes (**correlation**). Moreover, we will employ visual tools in order to estimate a good distributional fit for the data but also determine the existence of outliers (**histogram, box plots**). Also, we will use a combination of visual tools and descriptive statistics (**scatter plot, correlation**) to determine the co-linearity of attributes.

Figure 3: Pearson Correlation Matrix for all numeric attributes

	amount_funded	project_goal	number_backers
amount_funded	1.000000	0.436182	0.717896
project_goal	0.436182	1.000000	0.444587
number_backers	0.717896	0.444587	1.000000

We first analyze the **numeric attributes** and for each attribute we discover the following information:
`amount_funded`:

(a) Typical Value: Calculating the mean for this attribute we get that the average score is approximately $289,992$

(b) Spread: Approximately, the standard deviation (sample) for this data is $711,936$. Since significant outliers exist in this attribute a better measure of dispersion is $mad = 67,354$

(c) Distributional fit: Drawing a histogram of the data we see that the distribution looks very skewed to the right (i.e. few data points have extremely large values, see Fig. 4)

(d) Correlation: Referring to Table 3 and Figure 4, we see that `amount_funded` is most strongly positively correlated with `number_backers`. This is logical since we would expect that increasing the number of backers would also increase the total amount funded.

(e) Outliers: There exist some outliers which is evident from the magnitude of the measures of dispersion. Also outliers become apparent by looking at the histogram in Figure 4.

`project_goal`:

(a) Typical Value: Calculating the mean for this attribute we get that the average score is approximately $61,752$

(b) Spread: Approximately, the standard deviation (sample) for this data is $126,671$. Similarly to before, since outliers exist in this attribute a better measure of dispersion is $mad = 20,000$

(c) Distributional fit: Drawing a histogram of the data we see that the distribution looks very skewed to the right (i.e. few data points have extremely large values, see Fig. 4)

(d) Correlation: Referring to Table 3 and Figure 4, we see that `project_goal` is not significantly correlated with any of the other variables.

(e) Outliers: There exist some outliers which is evident from the magnitude of the measures of dispersion. Also outliers become apparent by looking at the histogram in Figure 4.

`number_backers`:

(a) Typical Value: Calculating the mean for this attribute we get that the average score is approximately $3,582$

(b) Spread: Approximately, the standard deviation (sample) for this data is $7,316$. Similarly to before, since outliers exist in this attribute a better measure of dispersion is $mad = 671$

(c) Distributional fit: Drawing a histogram of the data we see that the distribution looks very skewed to the right (i.e. few data points have extremely large values, see Fig. 4)

(d) Correlation: As explained, this variable is most strongly correlated with `amount_funded`.

(e) Outliers: There exist some outliers which is evident from the magnitude of the measures of dispersion. Also outliers become apparent by looking at the histogram in Figure 4.

Now we proceed to **clean** our dataset. First we need to explore whether our attributes contain **duplicates**. After querying the dataset, we find 7 duplicate instances, whose names are summarized in Table 6. We **remove the duplicates** from the dataset in order to perform clustering analysis.

Table 6: Titles of duplicate instances.

Title
Doug TenNapel Sketch Book Vol 2
Frame by Frame
The Infinite Loop Tablet and Smartphone Stand
Simple Bracket
THE BEST OF WDW - Volume 1
Aer Fit Pack: The Gym/Work Bag Designed for the City
XRAY.FM - The little station with big ideas.

Based on our analysis, we know that there exist **outliers** in our numerical attributes. Therefore for all of our attributes we define $Q_0 = 10\%$ quantile and $Q_1 = 80\%$ quantile and then take the attributes which fall inside these quantiles. Following this process, we only keep the following values:

$$81,136 \leq \texttt{amount\_funded} \leq 20,3491$$

$$18,000 \leq \texttt{project\_goal} \leq 50,000$$

$$1,537 \leq \texttt{number\_backers} :\leq 2,752$$

After discarding outliers we still have not merged the attributes together in order to create a single dataframe. We notice, that since we have deleted many values, "merging" the columns back together would result in having many **empty** cells. Therefore, we decide to delete all datapoints that contain at least 1 empty cell. That reduces significantly our datasets's size from $4,000$ instances to $1,013$.

The **reason behind** deleting all instances with at least one empty cell, is that we do not want any outliers in our dataset. More precisely, if an empty cell exists, then it means that its original value had been filtered out by the procedure we followed in the previous step (step 1), and hence, it was an outlier. An **alternative** approach would be to replace every missing value with the mean of that attribute, but that would inevitably distort the data and introduce some bias. This is not necessary here, since the resulting dataset size is adequate to allow our model to discover meaningful clusters.

We now analyze the **categorical** variables of our dataset. By looking at Table 7 we see that there is a large variety of combinations of values that the attributes take. More precisely, we observe all the attributes are multimodal, since there does not exist a value that occurs much more frequently than others. Moreover, by looking at Table 7 we see that the most frequent location (**mode**), category and currency are **San Francisco, Product Design** and **usd**, respectively.

In order to use the categorical information in our clustering models we need to add new binary attributes to the dataset, but we first use `OpenRefine` to ensure that there are no **spelling inconsistencies**. The number of discrete values that `location`, `category` and `currency` take are $806, 115, 9$, respectively. Evidently, if we encode all these variables as numerical the dimensionality of the problem would explode, leading to very sparse data (*curse of dimensionality*). Heuristically, we choose the top 30 most frequent items from the `location` and `category` attributes, and we name the rest of the items of each attribute as `other_location`, `other_category`, respectively. As a result, we create 73 new binary variables, out of which, 31 correspond to location, 31 correspond to category and 9 correspond to currency.

Appendix for Problem 3 Question 1

location	count
San Francisco, CA	280
Los Angeles, CA	271
New York, NY	212
Seattle, WA	147
London, UK	132
Chicago, IL	121
Brooklyn, NY	103
Portland, OR	84
San Diego, CA	62
Toronto, Canada	60

(a)

category	count
Product Design	773
Tabletop Games	742
Video Games	524
Hardware	182
Technology	152
Documentary	124
Gadgets	121
Design	99
Comics	79
Wearables	64

(b)

currency	count
usd	3438
gbp	252
cad	128
eur	96
aud	52
sek	14
nzd	10
dkk	7
chf	3

(c)

Table 7: Top 10 most frequent items for the **categorical** attributes.

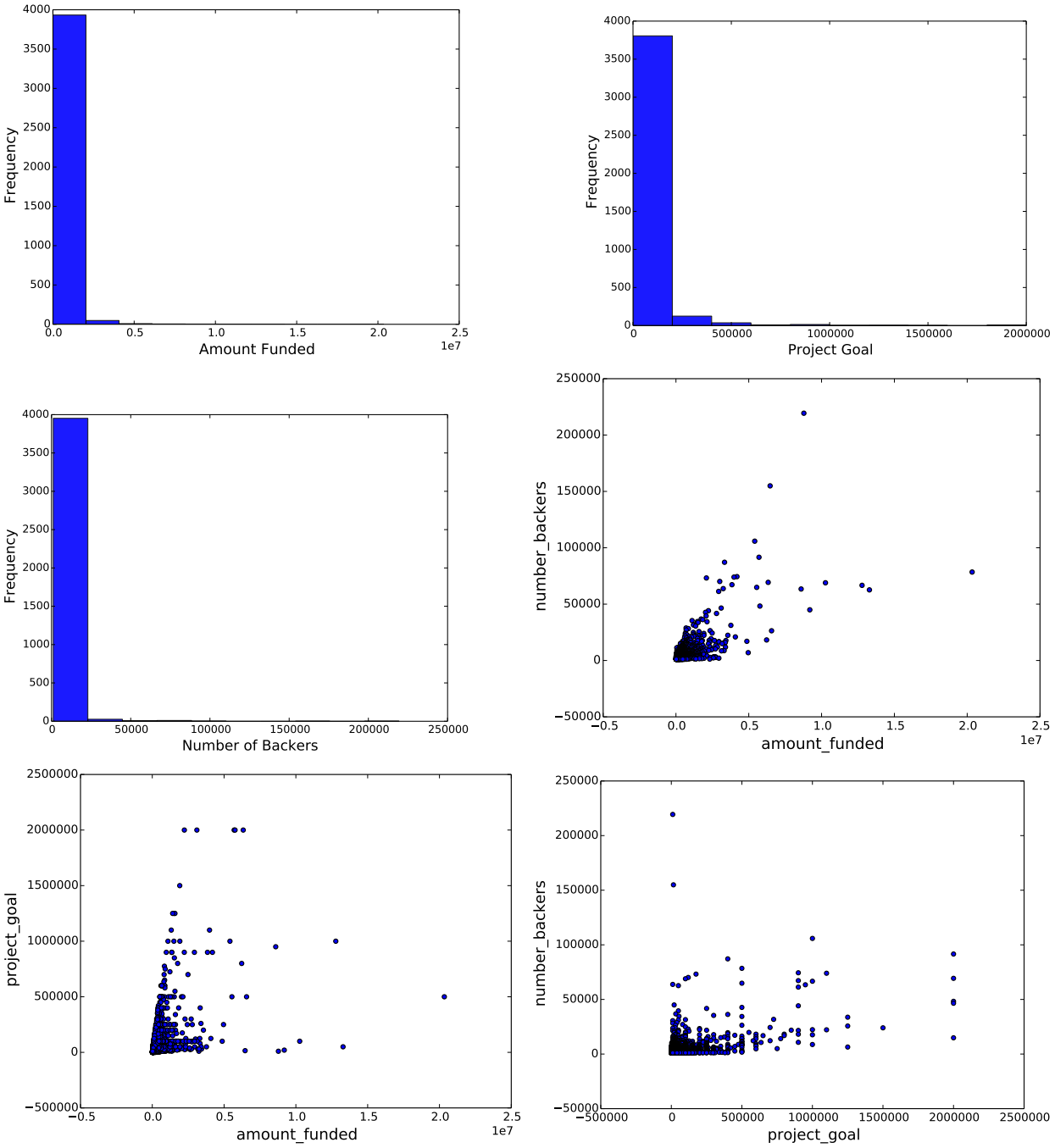


Figure 4: Histograms and Scatter plots for all **numeric** attributes

2. **K-Means** is one of the most **efficient** clustering algorithms. Since this algorithm relies on a distance metric it is is also very simple and hence intuitive to understand. A possible drawback of K-Means is that it only works well for **globular clusters** and that it is **sensitive to outliers**.

   DB-Scan is another very popular clustering algorithm that belongs in the family of

density based methods and performs well in many settings. The basic idea behind this method is that clusters are a set of density connected instances. **In contrast to K-Means**, when using DB-Scan the number of clusters is not fixed and we can naturally handle clusters of **any shapes and sizes**. Nevertheless, DB-Scan struggles with spareness of data and in many cases does not cluster all the data (i.e. noise instances). Moreover, one of the most important drawbacks of this method is that it is **inefficient** with large data.

Hierarchical clustering is a clustering method that functions on the idea that clusters can be represented as a hierarchy. More precisely, clusters can be **merged** (Agglomerative clustering) or **split** while attempting to minimize a loss function. One of the biggest advantages of this method is that it is **easy to interpret** and choose a level of resolution for the problem. Nevertheless, the drawback of this method is that it is very **slow** when we consider a large amount of data.

Finally, **Mixture Models (EM)** model clusters using statistical distributions, where each multivariate distribution corresponds to a cluster. This method can find **non-globular** clusters (i.e. ellipses) and it is **easy to add domain knowledge** by changing the parameters of the distributions. Nevertheless, this clustering method is **inefficient** and faces trouble with **noise and outliers**. Also, it struggles in accurately describing **small** clusters.

3. In order to find the best performing clustering algorithm, given our dataset, we experiment with various clustering models and values of the parameters and report the models for which we obtained the **highest silhouette score**. More precisely, the clustering methods that we explore are **K-Means, DB-Scan, Agglomerative Hierarchical clustering** and **Gaussian Mixture Models** and we vary the number of clusters from 0 to 15. We also produce plots by mapping high dimensional data into $2D$ data using *T-SNE*.

   After finding the algorithm (along with the parameters) that performs the best, we extract the corresponding clusters and attempt to interpret the results. Ideally, we would have human evaluators (domain experts) to assess the results of our clustering, but this is infeasible in the context of this project.

4. (a) The first method that we explore is *K-Means*.
   Using the "elbow method" we speculate that a useful clustering occurs for a number of clusters between 2 and 6 (Fig. 5). In order to assess our model we compute the *silhouette* score for all the values of $k = 1, 2, \ldots, 15$ and report the top 4 models in terms of silhouette score.

   As we see from Figure 6, the best silhouette score is achieved for $k = 2$ and is approximately equal to 0.1875. Nevertheless, this silhouette score is still not very

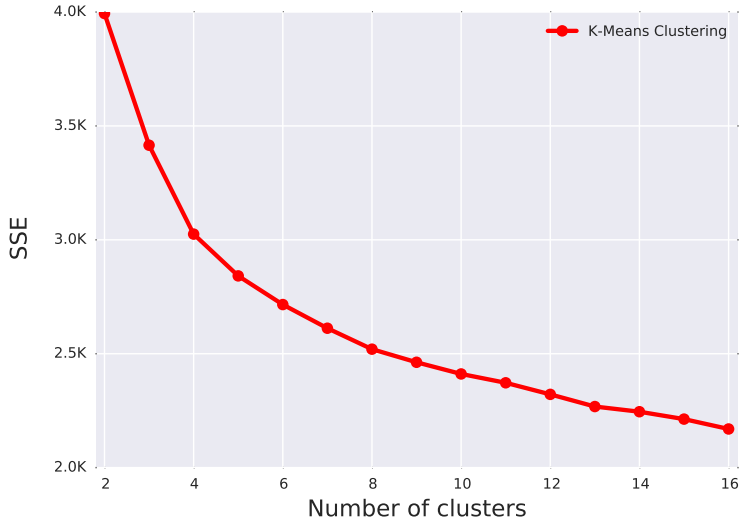


Figure 5: Sum of squared errors as we vary the number of clusters in the K-Means algorithm.

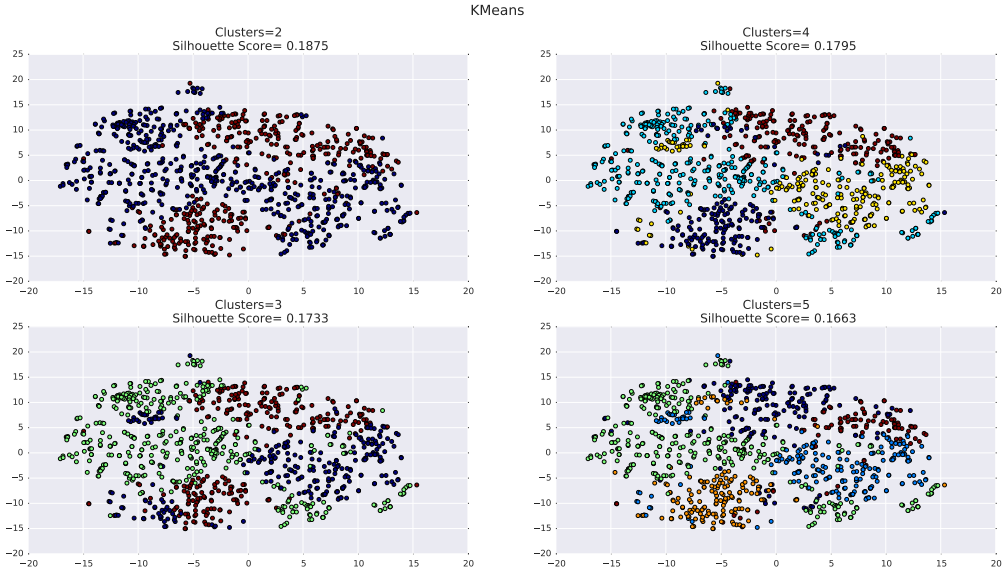


Figure 6: Four best K-Means models as we vary $1 \leq k \leq 15$.

close to 1, therefore, as we see next, K-Means is not the best performing clustering algorithm.

(b) The second clustering algorithm that we explore is *Hierarchical Agglomerative Clustering*. In this type of clustering we are given more "freedom" than K-Means, in the sense that we can also choose the *linkage* type. More precisely, for Agglomerative Clustering the cost function (linkage) is the distance between clusters. In this project we experiment with all three common linkage methods, namely:

- Single Linkage: The distance between two clusters is the *minimum* distance of their respective points
- Complete Linkage: The distance between two clusters is the *maximum* distance of their respective points
- Average Linkage: The distance between two clusters is the *average* distance of their respective points

More precisely, for each linkage type we vary the number of clusters k such that $k = 1, 2 \ldots, 15$ **but only report the four values of k for which we calculated the largest silhouette score.**

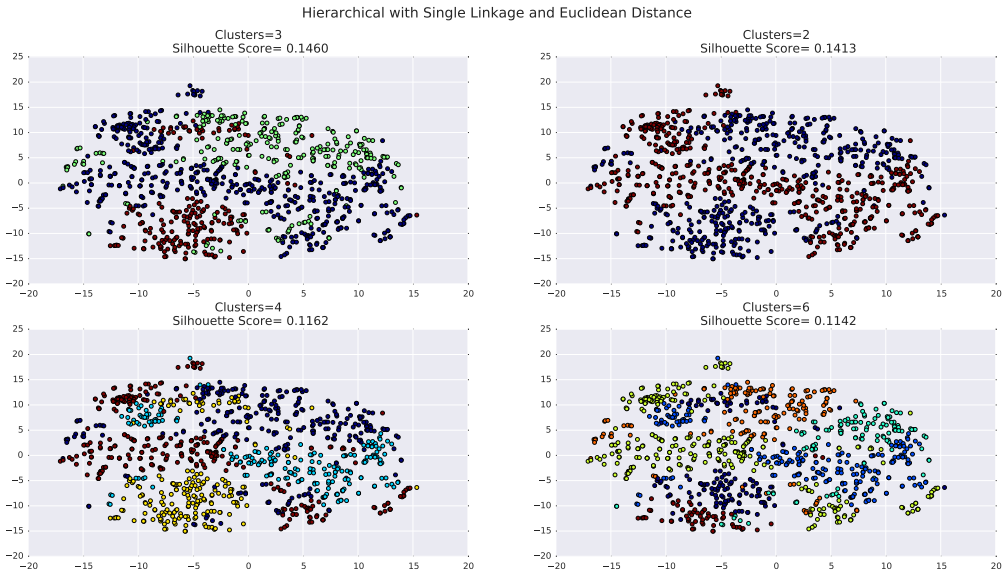


Figure 7: Four best Single Linkage models as we vary $1 \leq k \leq 40$.

As we see from Figures 7, 8, 9, single and complete linkage perform poorly achieving a silhouette score of 0.1460 and 0.1417, respectively. Average linkage performs the best out of the three methods, achieving a score of 0.2408 for $k = 2$.

Also, in Figure 10 we see the that if we let the euclidean distance be greater than 29 we get two clusters.

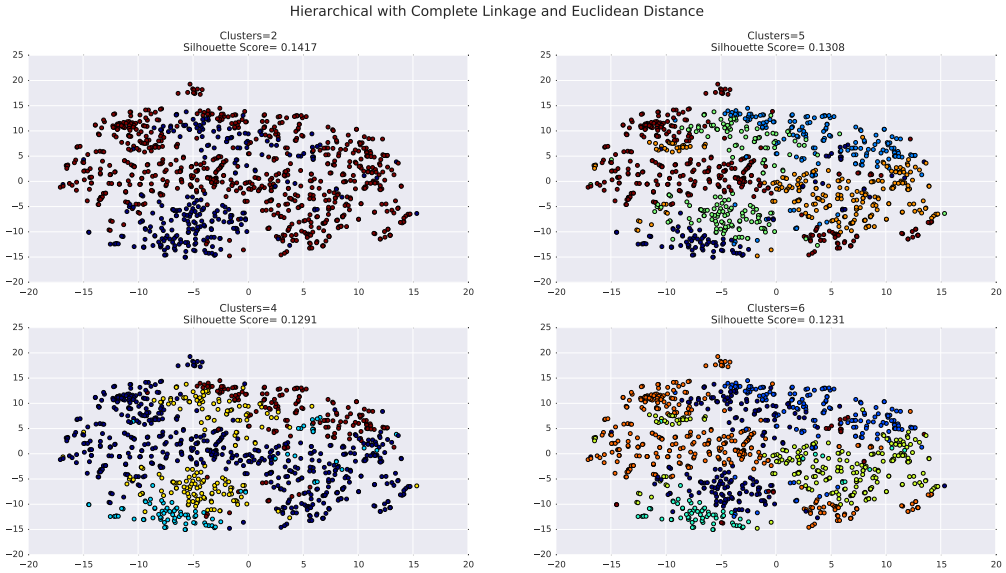


Figure 8: Four best Single Linkage models as we vary $1 \leq k \leq 40$.

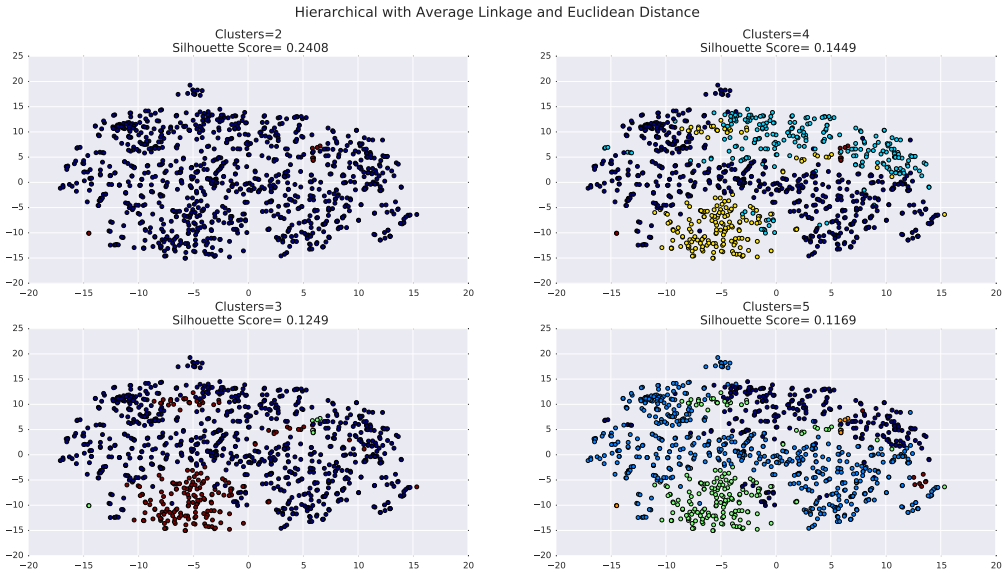


Figure 9: Four best Single Linkage models as we vary $1 \leq k \leq 40$.

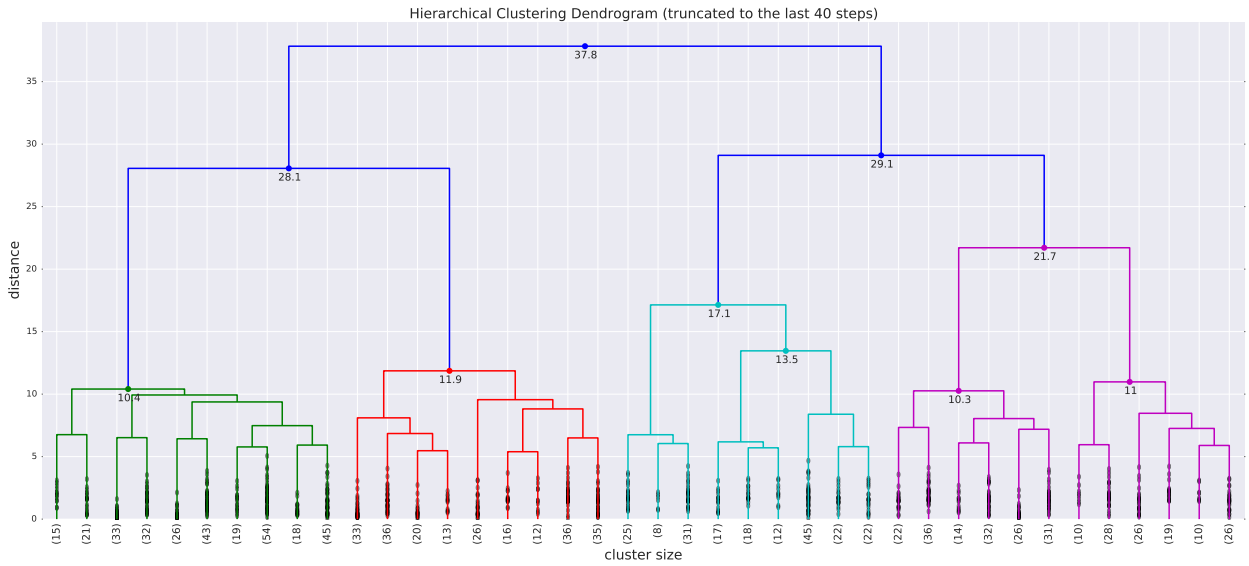


Figure 10: Dendrogram for Average Linkage

(c) The next clustering method that we explore, is *DB-Scan*. In contrast to the previously seen clustering algorithms, in this case we do not need to supply the number of clusters, as the algorithm finds the number of clusters based on ϵ and $MinPoints$. Heuristically, we found that letting $MinPoints > 4$ always resulted in the same clustering where only one cluster was present. Therefore, we vary $1 \leq MinPoints \leq 4$ and we let $\epsilon = 0.1 + 0.1t$ for $t = 0, 1, \ldots, 34$. As we varied $MinPoints$ between 1 and 5 we observed no significant changes in the value of the silhouette score. Nevertheless, the value of ϵ played a crucial role in getting a good clustering model. In fact, by looking at Figure 11 we see that we achieve the highest silhouette score for $k = 2$ and $0.1 \leq \epsilon \leq 0.4$.

(d) Finally, we try the *Expected Maximization* algorithm where we use Gaussian Priors. We see that this model performs very poorly compared to other clustering techniques (Fig. 12). According to this clustering model, the number of clusters that achieve the highest silhouette score (-0.0385) is 14, which is very different than the number of clusters suggested by our previous analysis.

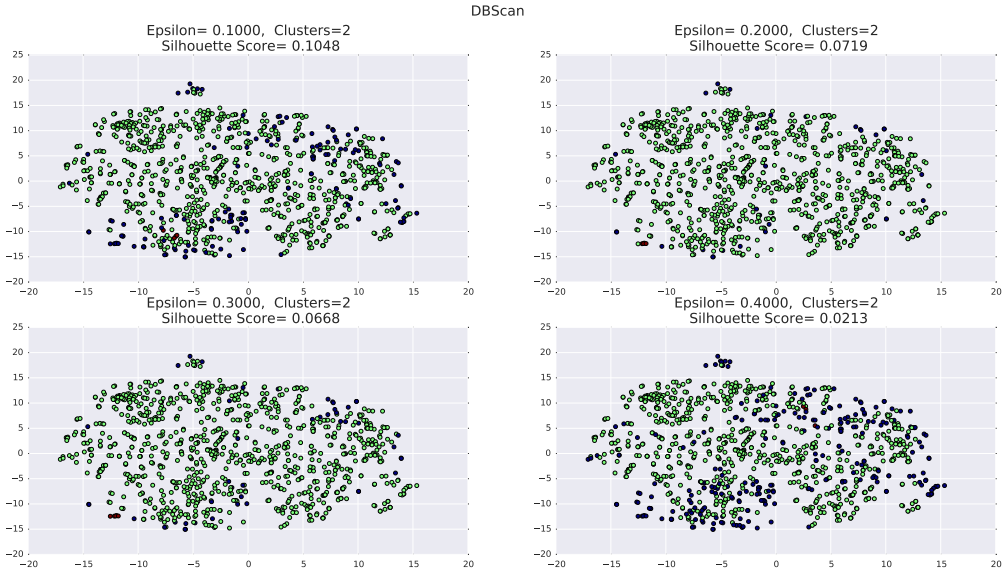


Figure 11: Four best Single Linkage models as we vary $1 \leq k \leq 40$ and fix $MinPoints = 4$.

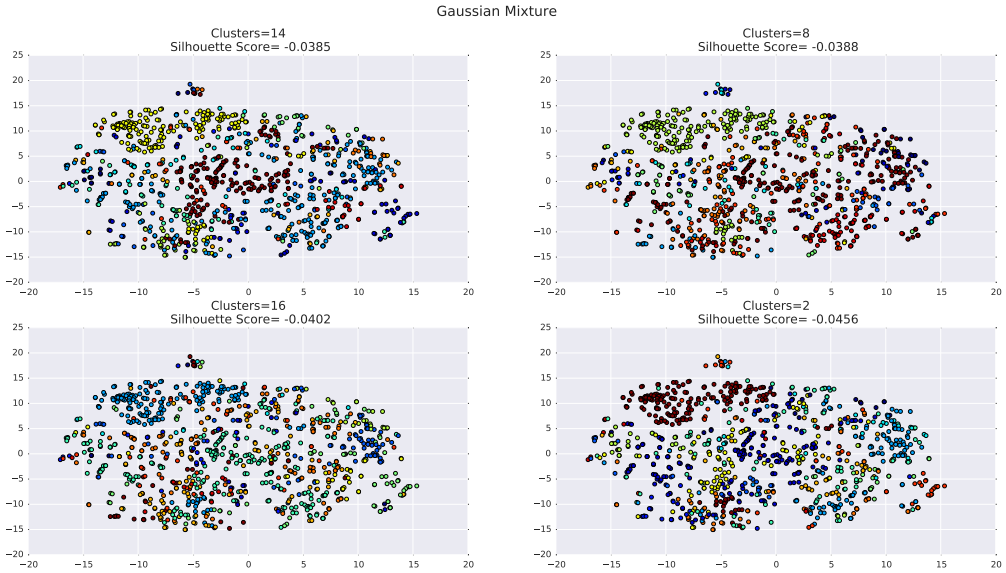


Figure 12: Four best Gaussian Mixture models as we vary $1 \leq k \leq 15$.

After analyzing various models we conclude that the best performing model is **Agglomerative Hierarchical Clustering with Average linkage and the Euclidean distance metric**.

5. Looking further into our clustering results, we see that there exist 2 clusters of unequal sizes. Namely, the first clusters consists of 270 instances, whereas, the second cluster consists of 743 instances.

A good candidate that the company could investigate further are the points that lie in cluster 1. In this cluster we see that the majority of projects are in the **Video Game** category and are based in **New York**. These are good opportunities that *Redwood Capital* could explore since technology investments are usually profit motivated.

Problem 4: Brainstorming for Financial Investment Data Mining (10 points)

The firm has asked you for new ideas about how data mining can be used to improve their investment strategy. Apply a structured brainstorming process to generate 3-5 possible ideas for using data mining to improve financial investments.

For each generated idea, provide a brief description of:

1. What type of investment problem is being addressed, e.g. identifying promising new start-up companies or determining how much money to invest in a particular company.
2. What type of data mining task is involved: classification, prediction, cluster analysis, or association analysis.
3. What data-sets would need to be collected for the data mining task.
4. How could Redwood Capital use the resulting model (supervised learning) or patterns (unsupervised learning)?

Answer:

1. Improve Investment Strategies: A possible idea would be to use classification to further expand and advance the models we have explored in Problem 1, in order to classify start up companies as promising or not. In order to perform such a task more accurately, the company could enhance the existing dataset to include more instances thus covering a broader number of cases that would allow the model to generalize with higher accuracy.

2. Stock Market Prediction: Another possibility where data mining could be used in this sector, is in attempting to predict the stock market behavior. This task falls under the *time series analysis* approach and is certainly very complex. Nevertheless, the company could vastly benefit from a such a model since it would know the appropriate times to invest. The dataset to be used for such a task, could be the behavior of the stock market during the past decade.

3. Patterns of Successful Companies: Using data mining we could explore further what kind of patterns occur in successful companies. Finding these patterns would allow the company to optimize their investment strategies by looking for patterns that make start up companies successful. We could use association analysis (unsupervised learning) for this task and use a similar dataset as the one provided in Problem 1.