SEMESTER PROJECT REPORT

# Predict Trends in Startup Investment using Youtube and Machine Learning

Author:
Shengzhao Xia (shengzhao.xia@epfl.ch)

Supervisor:
Francisco Pinto

Professor:
Patrick Jermann

June 7, 2019

# Contents

# 1 Introduction

It is generally believed in the venture investment communities that the success probability of a start-up is about one-tenth, which conveys the view that the failure rate of the start-up is very high. Whether a start-up can obtain investment from investors is a key factor in its success and the prediction of investment is of great importance to learn more about this factor. Therefore, it is very valuable for companies to explore the information of some well-known venture capital investors and utilize it to do an analysis and prediction. In particular, it could tell start-ups how to deal with investors when they plan to make a round of financing.

The goal of this project is to use multiple facets of the data science stack to analyze and predict whether investors will invest in some specified research domains and the amount of investment via his remarks (interviews or speeches). The report mainly consists of three simple parts: data acquisition, feature extraction, and model analysis. In the first part, it describes the process of scraping and cleaning data of some prominent VC investors, like Marc Andreessen, Peter Thiel, from different data sources. The second part is about extracting features including natural language processing features, network features, and metadata features from data and setting labels for them. In the final part, the report states the performance of supervised machine learning model used in the project, namely linear regression, and random forest and do an analysis based on the results. The entire workflow of the project is as below in Figure 1:

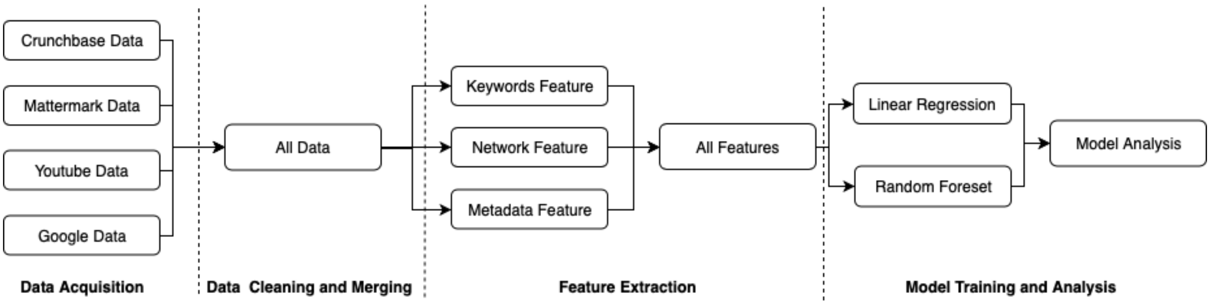
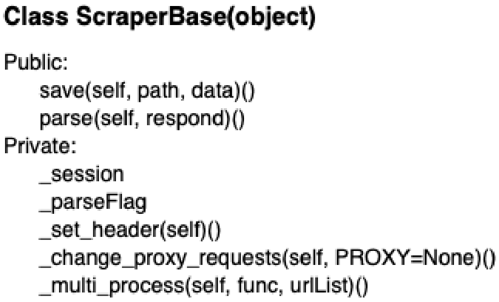


Figure 1: Workflow of the project

# 2 Data Acquisition

Since there is no readily available data provided by the project, data acquisition becomes a necessary and essential part of the project to gather the supportive data.

In order to scrape massive data and efficiently reuse the code, I construct a super (parent) crawler class name *ScraperBase*, which is implemented based on python and request packages. The structure and workflow of *ScraperBase* can be seen as below (Figure 2 and 3 respectively).



```
Class ScraperBase(object)

Public:
    save(self, path, data)()
    parse(self, respond)()
Private:
    _session
    _parseFlag
    _set_header(self)()
    _change_proxy_requests(self, PROXY=None)()
    _multi_process(self, func, urlList)()
```

Figure 2: Structure of ScraperBase

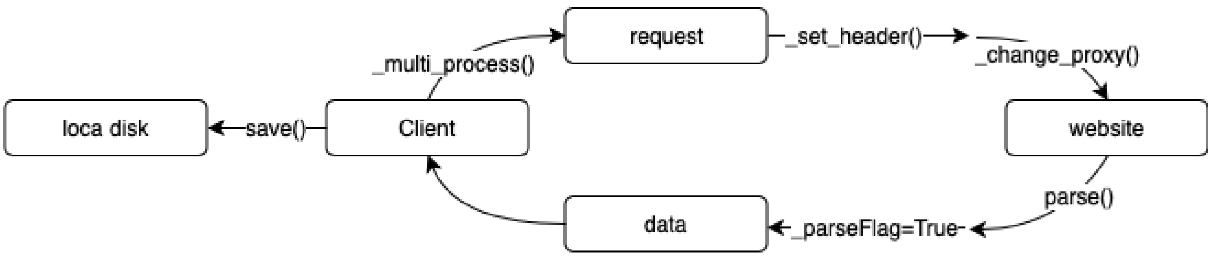


Figure 3: Workflow of ScraperBase

The workflow can be detailed described as:

1. The client uses function *_multi_process* to process scraping tasks in multi-cores of processor at the same time;
2. Multiple requests are sent at the same time and their headers will be set by function *_set_header*;
3. If the request is blocked by the website, function *_change_proxy_requests* will be called. It will ask user to input a new proxy or randomly pick an alternative proxy from the pre-constructed proxy pool if it exists;
4. After getting the response, function *parse* will be called to parse the web page, which would be overwritten in sub-classes and used for parsing data from different web pages;
5. The parsed data is pickled and saved to local.

The data used in the project is almost scraped from the Internet and can be divided into two kinds: investment data and text data. In the end, 541 investors that have all kinds of data from all different data sources are kept and their data is used. In the following sections, I will use the data of *Marc Andreessen* as an example to display the results.

## 2.1 Investment Data

The investment data in this project comes from online venture capital databases for start-up and is used for constructing network features and serving as labels for the following supervised machine learning task. It includes the investment information of the investors (e.g. investment time and amount of the investment) and the start-ups they invest in (e.g. category and location). Although there are a great number of venture capital databases online, none of them both contain complete data and query information for free (actually very expensive). After comparison and consideration, I choose to scrape data from Crunchbase and Mattermark, and merge them to get the final investment data which is relatively complete.

### 2.1.1 Crunchbase Data

Crunchbase Data is scraped from *www.crunchbase.com*, it only shows 10 most recent investment information of an investor or a start-up in the free trail, although its data is pretty complete and up-to-date. A class *CrunchbaseScraper* inheriting from *ScraperBase* is implemented for scraping data from Crunchbase. The investor and start-up data scraped from Crunchbase can be show as Table 1, Table 2 and Table 3 respectively. Each row of investor data represents one specific investment of the investor, while each row of company data shows one investment it got in the past.

| Announced Date | Organization Name | Funding Round | Money Raised |
|---|---|---|---|
| May 1, 2014 | Halo Neuroscience | Seed Round - Halo Neuroscience | $1.5M |
| Jun 18, 2013 | Muzy | Angel Round - Muzy | $4.4M |
| May 29, 2013 | PandoDaily | Seed Round - PandoDaily | $3M |
| Sep 21, 2009 | Fluther | Seed Round - Fluther | $600K |
| May 21, 2009 | Business Insider | Series B - Business Insider | $2.7M |

Table 1: Investment data of Marc Andreessen

| Announced Date | Money Raised | Lead Investors |
|---|---|---|
| Apr 30, 2018 | $1M | Future Planet Capital |
| Jan 30, 2018 | $13M | TPG |
| Feb 10, 2016 | $9M | Lux Capital |
| May 1, 2014 | $1.5M | Marc Andreessen |

Table 2: Investment data of Halo Neuroscience

| Categories | Consumer Electronics | Health Care | Medical Device |
|---|---|---|---|
| Headquarters Regions | San Francisco Bay Area | West Coast | Western US |

Table 3: Metadata of Halo Neuroscience

### 2.1.2 Mattermark Data

Mattermark Data is scraped form *www.mattermark.com*, the data from Mattermark is somehow outdated and the data records are incomplete, however it has a free trail and all the data in the database can be accessed at the trial. I used it to supplement the Crunchbase data and it can expand the field of company categories and add some old investment records. A class *MattermarkScraper* inheriting from *ScraperBase* is implemented for scraping it. Since it needs to login in to query the database, *selenium* package is used to automatically login in and maintain the cookies for the whole connection session.

### 2.1.3 Seedproof Data

A list of investors' names is scraped from *https://seedproof.com/investors/* and there are 937 investors in total. However, since some investors have no data in VC database or Youtube (described below), I finally only use 541 of them.

## 2.2 Text Data

Text data is the remarks of investors (e.g. interviews or speeches), which shows investors' attitudes or opinions about their investments and might be useful to explain their past investments or predict future investments.

### 2.2.1 Youtube Caption Data

The project was originally intended to use Twitter data, but many investors' Twitter has no investment-related contents. In order to obtain more informative data, the project chose Youtube subtitles as text data in the end. The subtitles of youtube were scraped using *Youtube Data Api* from *Google Api*, which has a quota for each day and finally limits the size of the text data. A class *MattermarkScraper* inheriting from *ScraperBase* is built to do the scraping.

In the scraping, investor's name is used as the keywords to query. Constraints are set that the video has to have caption in English and its duration should be longer than 20 minutes to make sure it is a long interview with adequate information. The results are sorted according to relevance as shown in Table 4, each row of the data includes the title, published-date, caption of the video. The result is quite convincing, the relevant videos about prominent investors are almost about investment or economy themes.

| Title | Published date | Caption |
|---|---|---|
| Marc Andreessen on Big Breakthrough Ideas and Courageous Entrepreneurs | 2014-03-08 | Thank you very much for taking the time to come in and speak to us. Many of us, are aspiring entrepreneurs, so we... |
| Fireside Chat: Marc Andreessen & Sebastian Thrun | 2018-03-29 | we'll be closing out intersect today and it is a meeting of brilliant minds our own Sebastian run is joined by entrepreneur and investor Marc Andreessen... |
| Marc Andreessen on Change, Constraints, and Curiosity | 2016-11-14 | well mark welcome back to Stanford and from one Midwestern to another we're honored to have you here great thank you thanks everybody... |

Table 4: Youtube data of Marc Andreessen

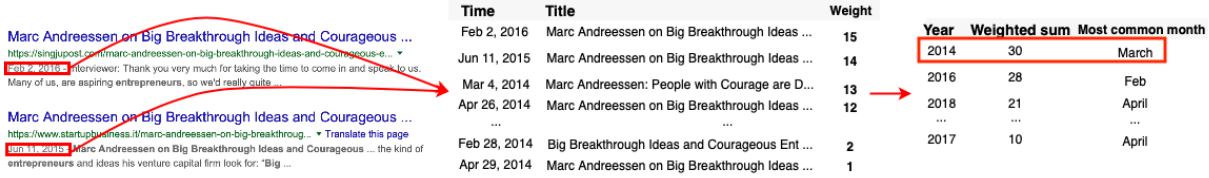
### 2.2.2 Google Search Data

How long will investors invest after giving an interview? Time interval between the interview and the investment is an important factor that worth consideration and analysis. However, the published-time of the video in Youtube is the time when the video was uploaded rather than the actual time when the interview was held. So in order to know the actual time of the interview, I use the searched results of Google to automatically extract it.

The idea is based on that most of the web pages about the event (the speech or interview) were created in a short period after it was held, which means the creation times of those web pages could approximately reflect the actual time of the event. I use the video name as the keyword, queried it and got results according to relevance, and then scraped the page's creation times of first 15 searched results.

From those creation times, I assigned a weight to the year of each creation time according to the relevance, which means the year of most relevant result will have the largest weight. Then I choose the year having the largest sum of weights as the year of actual year and the most common month of this year as the actual month. Using weights to decide actual time can weaken the influence of the results that are not that relevant but appear frequently in these 15 results. The workflow is shown below as Figure 4, the actual data obtained in this example is March, 2014.

A Class *GoogleScraper* inheriting from *ScraperBase* is built to achieve the workflow above. The results obtained is shown below in Table 5. The column auto-extract time is the time extracted above and the column real time is ground truth time of event found manually. From the results of Marc Andreessen, we can see that at the level of year and month, most auto-extract times are consistent with the real published time (for Marc Andreessen, 80% is correct).



Marc Andreessen on Big Breakthrough Ideas and Courageous ...
https://singjupost.com/marc-andreessen-on-big-breakthrough-ideas-and-courageous-e...
Feb 2, 2016 - Interviewer: Thank you very much for taking the time to come in and speak to us. Many of us, are aspiring entrepreneurs, so we'd really quite ...

Marc Andreessen on Big Breakthrough Ideas and Courageous ...
https://www.startupbusiness.it/marc-andreessen-on-big-breakthroug... Translate this page
Jun 11, 2015 - Marc Andreessen on Big Breakthrough Ideas and Courageous ... the kind of entrepreneurs and ideas his venture capital firm look for: "Big ...

| Time | Title | Weight |
|---|---|---|
| Feb 2, 2016 | Marc Andreessen on Big Breakthrough Ideas ... | 15 |
| Jun 11, 2015 | Marc Andreessen on Big Breakthrough Ideas ... | 14 |
| Mar 4, 2014 | Marc Andreessen: People with Courage are D... | 13 |
| Apr 26, 2014 | Marc Andreessen on Big Breakthrough Ideas ... | 12 |
| ... | ... | |
| Feb 28, 2014 | Big Breakthrough Ideas and Courageous Ent ... | 2 |
| Apr 29, 2014 | Marc Andreessen on Big Breakthrough Ideas ... | 1 |

| Year | Weighted sum | Most common month |
|---|---|---|
| 2014 | 30 | March |
| 2016 | 28 | Feb |
| 2018 | 21 | April |
| ... | ... | ... |
| 2017 | 10 | April |

Figure 4: Workflow of the project

| Title | Video pub-time | Auto-extract time | Real time |
|---|---|---|---|
| Marc Andreessen on Big Breakthrough Ideas... | 2014-3-8 | 2014-3 | 2014-2-28 |
| Fireside Chat: Marc Andreessen... | 2018-3-29 | 2018-3 | 2018-3-29 |
| Marc Andreessen on Change, Constraints... | 2016-11-14 | 2016-11 | 2016-11-8 |
| ... | ... | ... | ... |
| a16z Podcast: Ben and Marc Explain... | 2019-1-1 | 2014-8 | 2014-8-25 |
| Marc Andreessen - Startup School 2011 | 2013-4-24 | 2011-10 | 2011-10-16 |

Table 5: Comparasion between video time, auto-extract time and real time

## 2.3 Data Merge

In order to get a complete data-frame for the feature extraction, I have to merge text data and investment data above. It is done by computing the cross product of them and filtering the row where its interview published-time is later than the investment time, which means this interview has no contribute to the investment. A class *InvestorDataFrame* will run the entire pipeline to use different scrapers to gather different kinds of data, clean and merge them into a final data-frame which can shown below in Table 6.

| Investor | Start-up | Categories | Region | Announced Time | Money Raised | Title | Caption | Published time |
|---|---|---|---|---|---|---|---|---|
| Marc Andreessen | Halo Neuroscience | Consumer Electronics | US | 2014-5-1 | $1.5M | Marc Andreessen on Big Breakthrough Ideas and Courageous Entrepreneurs | Thank you very much for taking the time tocome in and speak to us. Many of us, areaspiring entrepreneurs, so we... | 2014-03 |

Table 6: Merge all data into a data-frame

# 3 Feature Extraction

After obtaining the data, I extracted meaningful numerical features form it and those features can roughly divided into three kinds: text feature, network feature and Metadata feature.

## 3.1 Text Feature

### 3.1.1 Sentence segmentation

Since most of the subtitles of YouTube video are generated by an automatic speech recognition system which outputs an unpunctuated sequence of words. In order to increase the effectiveness of subsequent processing (both keyword and sentiment feature extraction), I did punctuation restoration, namely sentence segmentation at first.

Inspired by using sequence to sequence model to do translation [1], I try to adapt it to restore punctuation, that is to "translate" words to punctuations. Here I implemented a bidirectional recurrent neural network[2] model with attention mechanism[3] for punctuation restoration in unsegmented text. At time step $t$ the model outputs probabilities for punctuations $y_t$ between the previous word $x_{t-1}$ and current input word $x_t$. The process of the model from input to output can be described as below and the graphical illustration is shown as Figure 5.:

1. The sequence of one-hot encoded input words $X = (x_1,..., x_T)$ is first processed by a bidirectional layer consisting of two recurrent layers with GRU units in opposite dirrection. The bidirectional state $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t}]$ where $\overrightarrow{\mathbf{h}_t} = GRU(\mathbf{x}_t\mathbf{W}_e, \overrightarrow{\mathbf{h}}_{t-1})$ and $\overleftarrow{\mathbf{h}_t} = GRU(\mathbf{x}_t\mathbf{W}_e, \overleftarrow{\mathbf{h}}_{t-1})$ represent forward and backward recurrent layer respectively.

2. The bidirectional layer is followed by a decoder which output the probability $p(\mathbf{y}_i|\mathbf{y}_1, ..., \mathbf{y}_{i-1}, \mathbf{x})$ $= g(\mathbf{y}_{i-1}, \mathbf{s}_i, \mathbf{c}_i)$ where $s_i$ is an RNN hidden state for time i, computed by $\mathbf{s}_i = f(\mathbf{y}_{i-1}, \mathbf{s}_i, \mathbf{c}_i)$.

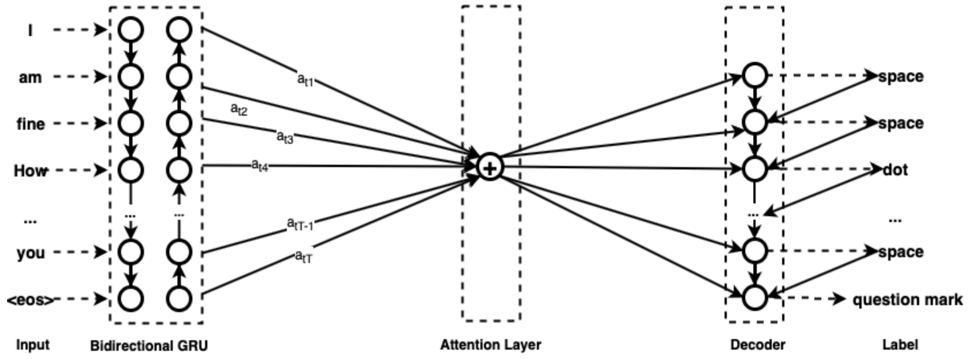


Figure 5: Graphical illustration of the model

The data I use is the well-known Europarl English corpus (6M words). The input sequence is a 50 words long slice and each word is one-hot encoded. The slice begins with the first word of a sentence and ends with an unfinished sentence and the unfinished part would be copied to begin the next slice. The output sequence is one element shorter punctuation slices since there is no punctuation before the first word. In the project, only 4 kinds of punctuations are considered including space, comma, period, question mark and they are also one-hot encoded as labels. In the training, the slices are shuffled and arranged into mini-batches of 128 slices. I set all hidden layers to 256 dimensions and start to train the model using AdaGrad with a learning rate of 0.01. Precision, Recall and F1-socre are used to simply evaluate of the model (on 10% of corpus data) are shown below in Table 7:

| Punctuation | Precision (%) | Recall(%) | F1-socre(%) |
|---|---|---|---|
| Comma | 63.9 | 58.1 | 63.2 |
| Period | 74.2 | 73.2 | 73.5 |
| Question Mark | 64.8 | 59.5 | 63.3 |
| Overall | 68.9 | 61.5 | 65.3 |

Table 7: Performace of the model

An example of sentence segmentation can be shown as in Table 8.

| Original Caption | Punctuated Caption |
|---|---|
| we'll be closing out intersect today and it is a meeting of brilliant minds our own Sebastian run is joined by entrepreneur and investor Marc Andreessen co-founder and general partner at Andreessen Horowitz | We'll be closing out intersect today, and it is a meeting of brilliant minds. Our own Sebastian run is joined by entrepreneur and investor Marc Andreessen, co-founder and general partner at Andreessen Horowitz. |

Table 8: Example of sentence segmentation

### 3.1.2 Keywords Feature

After restoring the punctuation for the subtitles, keywords can be extracted more accurately and effectively. The keywords are extracted using RAKE[4] (Rapid Automatic Keyword Extraction algorithm) which uses stopwords and delimiters to partition the document text into candidate keywords. Co-occurrences of words within these candidate keywords can be scored according to word frequency, word degree and ratio of degree to frequency and keywords can be picked out by those metrics. I extracted keywords with a maximum length of two, an example result of keywords extraction is shown as Table 9.

Then I utilize *word2vec* model[5] (pre-trained by Google) to embed the keywords and compute the cosine similarity between each keyword and the corresponding investment domain. All those similarities will be accumulated as the similarity between the interview and investment. Furthermore, I also compute the number of the words whose similarity with the investment domain are larger than a threshold (50% largest similarity among all words) as the number of relevant words. Both of them are normalized by dividing by the length of the caption, which can intuitively reflect the contribution from the interview to the investment. The features could also be shown as Table 9.

| Original Caption | Punctuated Caption | Similarity | RelevantWordsFreq |
|---|---|---|---|
| We'll be closing out intersect today, and it is a meeting of brilliant minds. Our own Sebastian run is joined by entrepreneur and investor Marc Andreessen, cofounder and general partner at Andreessen Horowitz. | ['sebastian run', 'intersect today', 'general partner', 'brilliant minds', 'andreessen horowitz', 'meeting', 'joined', 'founder', 'entrepreneur', 'co', 'closing'] | 0.195 | 0.0156 |

Table 9: Keywords feature extraction

### 3.1.3 Sentiment Feature

The sentiment contained in the interview expresses the attitudes, evaluations, and emotions of the investor to a specific investment domain and sentiment analysis enables me to make more sense out of the subtitle data. The tools I use to extract sentiment feature is VADER (Valence Aware Dictionary and Sentiment Reasoner)[6], which is a lexicon and rule-based sentiment analysis tool. It uses a combination of sentiment lexicon (a list of lexical features) which are generally labelled according to their semantic orientation as either positive or negative.

In the project, I split sentences which are segmented above and compute the sentiment score for each sentence. A usually-used threshold 0.5 is set to distinguish positive and negative sentences, which means a sentence with sentiment score higher than 0.5 would be regarded as a positive sentence and a sentence with sentiment score lower than -0.5 would be regarded as negative. Then I count the number of positive sentences and sum all the sentiment scores of them, so as to negative sentences. Those values will be normalized and regarded as the positive and negative sentiment features, which can be seen as Table 10.

| Original Caption | PosSentNum | NegSentNum | PosSentScore | NegSentScore |
|---|---|---|---|---|
| Hello, all right, it works, welcome to startup school everybody. Thank you for getting up so early in the morning. | 0.423 | 0.110 | 0.446 | -0.371 |
| So I'm thrilled to be here today and when I was looking at my briefing sheet, and I saw that the title of the series is a view from the top I thought. | 0.411 | 0.204 | 0.425 | -0.386 |

Table 10: Sentiment feature

### 3.2 Network Feature

From the Crunchbase data, two kinds of network can be built: investor network and investment domain network.

- Investor network is built by linking investors who invest in the same domains. The investment domains of each investor are one-hot encoded to a vector and the weight of network edges (adjacency matrix) could be calculated by computing the cosine similarity of those domain vectors.
- Similarly, domain network are built by linking domains having the same investors and the weights is the cosine similarity of their one-hot encoded investor vectors.

The intuition is that an investor or investment domain with higher status or important within the investment network will, to some extent, affect the investment results, e.g. more prominent investors will probably invest more money or an investment domain invested by a plenty of investors would more likely to attract more investment. From each network, degree centrality, betweenness centrality and closeness centrality are computed which could reflect the importance of vertex in the network. The investor network features and domian network features can be shown as Table 11 and Table 12 respectively.

| Investor | Investor_degree | Investor_betweenness | Investor_closeness |
|---|---|---|---|
| Bill Gurley | 0.829 | 0.0058 | 0.368 |
| Jim Goetz | 0.948 | 0.0081 | 0.405 |
| Mary Meeker | 0.858 | 0.0025 | 0.375 |
| Peter Fenton | 0.846 | 0.00037 | 0.377 |
| Josh Kopelman | 0.144 | 0.000 | 0.276 |

Table 11: Investor network feature

| Domain | Domain_degree | Domain_betweenness | Domain_closeness |
|---|---|---|---|
| 3D Printing | 0.823 | 0.000009 | 0.365 |
| 3D Technology | 0.798 | 0.000011 | 0.355 |
| A/B Testing | 0.797 | 0.000012 | 0.355 |
| Accounting | 0.826 | 0.000009 | 0.365 |
| Ad Network | 0.791 | 0.000012 | 0.353 |

Table 12: Domain network feature

### 3.3 Metadata Feature

Metadata feature is the feature extracted from the metadata of the web page. Here I only compute the in months between the investment time and interview time (auto-extracted above). The feature is shown as Table 13.

| Investor | Domain | Title | Interval_Month |
|---|---|---|---|
| Marc Andreessen | Consumer Electronics | Marc Andreessen on Big Breakthrough Ideas... | 2 |
| Marc Andreessen | Consumer Electronics | Marc Andreessen - Startup School 2011 | 31 |
| Marc Andreessen | Consumer Electronics | Michelle Rhee: Lead from the Front | 2 |

Table 13: Time interval feature

### 3.4 Final Feature DataFrame

In conclusion, I extract 13 semantically meaningful features. Merging with the labels (investment amount and domain), the format of final data-frame for later supervised machine task has 39155 rows and one row as an exmaple can be seen in Table 14.

| Investor | Domain | Money Raised | PosSent Num | NegSent Num | PosSent Score | NegSent Score | Similarity | Relevant WordsFreq | Interval Month |
|---|---|---|---|---|---|---|---|---|---|
| Bill Gurley | Dental | $21M | 0.495 | 0.114 | 0.452 | -0.355 | 0.059 | 0.002 | 42 |

| Investor Degree | Investor Betweenness | Investor Closeness | Domain Degree | Domain Betweenness | Domain Closeness |
|---|---|---|---|---|---|
| 0.8288 | 0.0058 | 0.3675 | 0.7959 | 0.000014 | 0.3538 |

Table 14: Final feature data-frame

# 4 Model Analysis

The data will be shuffled and split into training set (80 %) and test set (20 %). Both of them and their labels will be standardized for training and test. 5-folds cross validation is used to tune the hyper-parameters.

## 4.1 Linear Regression

A rigid regression model (linear model with L2-regularization) is trained. The coefficient alpha of L2-regularization is regarded as an hyper-parameter and is tuned by cross validation.

Before fitting the data to the linear regression model, I briefly have a look at the co-variance matrix of the feature data-frame and find that the co-variance between the feature *domain degree centrality* and feature *domain closeness centrality* is very high (0.99). I drop one of them and start training.

After training with cross-validation, the model get its minimum loss when alpha equals 100 while its $R^2$ score is only 0.047. $R^2$ score is the proportion of total variation which is explained by the model and a low $R^2$ score means the observations (features) are scattered quite randomly and the actual relationship between some features and label is fairly week. In order to explore and explain the result, I take a further step to do a statistic analysis on the linear regression. Some important statistics[7] can be shown below.

| | |
|---|---|
| R-squared | 0.050 |
| Adj. R-squared | 0.050 |
| F-statistic | 135.4 |
| Prob (F-statistic) | 0.00 |

Table 15: Summary of the regression

From the summary of the regression in Table 15, it can be seen that:

- The p-value associated with the F-test (Prob (F-statistic) in Table 15) is 0.00, which is much less than the level of significance (5% or even 1%).
- The F-statistic rejects the null hypothesis that all the coefficients are zero at the 5% level of significance and it's likely to say that at least one of those coefficients is none zero, namely at least one of these variables is significant.

We can get more in depth to look at the statistics about variables in Table 16, it shows that:

- Coefficient is the parameter (slope) before the variable in the linear regression expression. Minus coefficient means the variable might have a negative effect to the result (label). Feature *negSentNum* is the number of sentences with negative sentiment and it is reasonable that its coefficient is minus. However, it is out of expectation that *Similarity* and *RelevantWordsFreq* have the negative coefficients.
- The t-statistic is calculated by dividing the coefficient by standard error. The higher the t-statistic, the more significant the variable is. So *Investor_betweenness_centrality* is the most important feature in the regression, second important feature is *Investor_closeness_centrality* and followed by *posSentNum* and *Domain_betweenness_centrality.*

|  | Coefficient | Standard Error | $t$ | $P > \|t\|$ | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 7.173e-14 | 0.006 | 1.29e-11 | 1.000 | -0.011 | 0.011 |
| posSentNum | 0.0076 | 0.008 | -0.942 | 0.346 | -0.024 | 0.008 |
| negSentNum | -0.0288 | 0.008 | -3.525 | 0.000 | -0.045 | -0.013 |
| posSentScore | 0.0456 | 0.007 | 6.424 | 0.000 | 0.032 | 0.060 |
| negSentScore | 0.0022 | 0.009 | 0.250 | 0.802 | -0.015 | 0.019 |
| Similarity | -0.0212 | 0.006 | -3.435 | 0.001 | -0.033 | -0.009 |
| RelevantWordsFreq | -0.0064 | 0.006 | -1.101 | 0.271 | -0.018 | 0.005 |
| Time_Interval_Month | 0.0231 | 0.006 | 4.150 | 0.000 | 0.012 | 0.034 |
| Investor_degree_centrality | -0.1111 | 0.027 | -4.064 | 0.000 | -0.165 | -0.058 |
| Investor_betweenness_centrality | 0.1919 | 0.006 | 31.405 | 0.000 | 0.180 | 0.204 |
| Investor_closeness_centrality | 0.1942 | 0.027 | 7.190 | 0.000 | 0.141 | 0.247 |
| Domain_betweenness_centrality | 0.1586 | 0.028 | 5.669 | 0.000 | 0.104 | 0.213 |
| Domain_closeness_centrality | 0.1468 | 0.028 | 5.253 | 0.000 | 0.092 | 0.202 |

Table 16: Statistics about variables

- The p-value of t-statistic gives the probability of this coefficient occurring just due to the random chance. The probability larger than 5% can not reject null hypothesis, which means features *negSentScore*, *posSentNum* and *RelevantWordsFreq* might just fit the model by chance.

## 4.2 Random Forest

I also try the Random Forest model to do the regression, similarly, 5-folds cross validation is applied to choose the optimal parameter *n_estimators* (the number of trees in the forest) and *max_depth* (the maximum depth of the tree). As the result, the model with parameters $n\_estimators = 10$ and $max\_depth = 3$ is used.

The importance of features can analyzed by the model, I plot it in Figure 6.

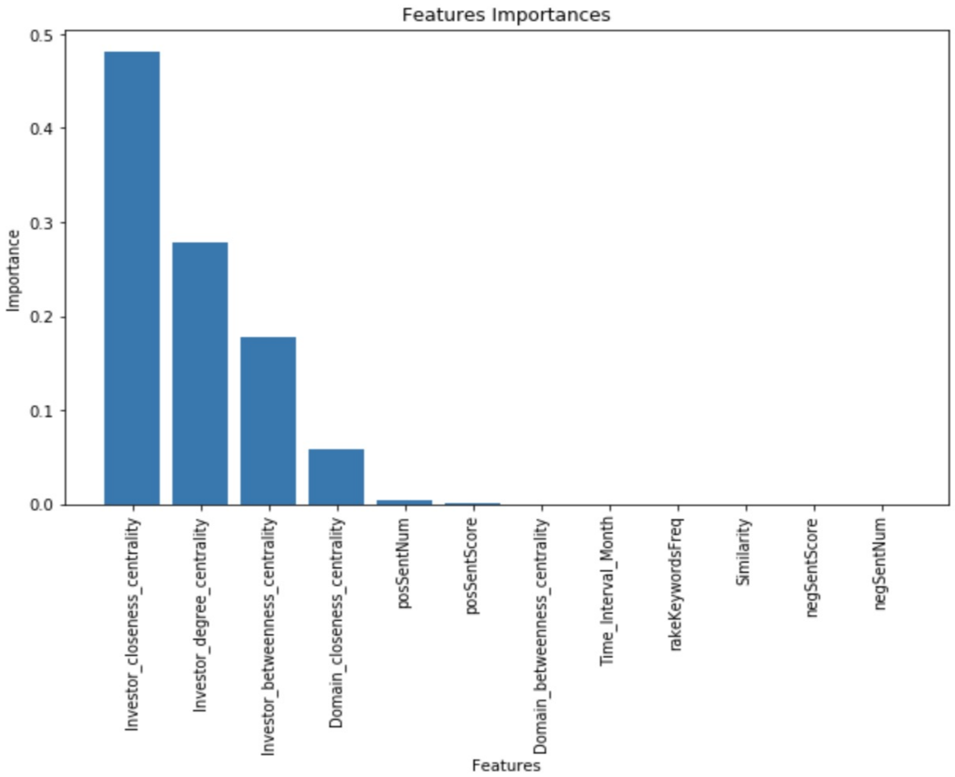


Figure 6: Features Importances

From Figure 6, it can be seen that the most important feature is *Investor_closeness_centrality*, and then followed by *Investor_degree_centrality*, *Investor_betweenness_centrality* and *Domain_closeness_centrality*. The importance of text features is nearly all 0, which means the text features almost have no influence to the result of regression in Random Forest model.

# 5 Conclusion and Feature Work

The project is to analyze investment by scrapping and analyzing investment information and remarks information (speeches and interviews) from prominent VC investors. The report:

- First introduce the methods of how to scrape data from different data sources including different venture capital database (Crunchbase and Mattermark), Youtube, Google and some other websites. Some crawler classes are built to achieve it.
- Secondly, some intuitive feature extraction ideas are applied on the scraped data. For the text data, sentence segmentation is done first to restore punctuation in Youtube subtitles and then sentiment in each sentence is analyzed to compute sentiment score. After that, it is followed by rake keyword extraction, similarity calculation with word2vec embeding and relevant word frequency count. For the investment data, the networks of investors and domains are constructed and centrality features of those networks are extracted.
- Finally, linear regression model and random forest model are trained to analyze the predictive power of the features. Statistics output of linear regression model is detailedly described to analyze the importance of each feature.

From the model analysis above, we can find that text features extracted from youtube subtitles do not have a really good performance to explain the investment behavior of investor. But the ideas of extracting those features are intuitive and should make some senses. From the analysis of linear regression, we can see that although those text features are not as strong as network features, they do have prediction power to some extent. I am convinced that those feature extraction methods can work well if the raw text data can be more carefully processed rather than just restore the punctuation. So I think the feature work should pay more attention to the pre-precessing of text data as described below:

- Some interviews on Youtube are debates between several investors and their opinions about investment might be different. It will work better if we can just extract the words of aimed investor.
- Although the Youtube data is queried according to the name of the investor, some other videos that are relevant to the topic (e.g. recommended by Youtube) but actually not relevant to the investors will be included sometimes. The result will be better if we could identify whether the aimed investor really appear in the videos.
- There might be some latent features that can not be described semantically and deep leaning might be applied to extract more effective and representative latent features.

## References

[1] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[2] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[4] S. Rose, D. Engel, N. Cramer, and W. Cowley, *Automatic Keyword Extraction from Individual Documents*, pp. 1 – 20. 03 2010.

[5] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *CoRR*, vol. abs/1402.3722, 2014.

[6] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," 2014.

[7] V. A. Barbur, D. C. Montgomery, and E. A. Peck, "Introduction to linear regression analysis.," 1981.