Predicting Decisions by Successful Investors Using Machine Learning

Hengyu Wang, *Department of Engineering Science, University of Oxford,*

***Abstract*—Predicting future outcomes in early-stage investing is more difficult than other investment fields. There are less numbers and more qualitative features. With the proliferation of data about private companies and advances in machine learning, we can build models to predict the future performance of a startup much better than relying on our intuitions.**

I. Introduction

PRIOR work at Vela proved that investors of a startup are the most important predictive feature of future success. My thesis is that this feature is a critical proxy signal that I should deep dive and understand the correlation of other attributes deeper. I am given two datasets:

- Companies invested by a successful investor: 2,875 companies.
- Companies invested by a failed investor: 8,152 companies.

In this project, my objective is to build a model to predict whether successful investors are going to invest in a start-up. This paper presents two main approaches for evaluating the likelihood of investment by investors who succeed beforehand. Method 1 provides a generative learning algorithm specific for text classification, called Naïve Bayes Classifier. It uses conditional independence assumptions to mathematically formulate the probability distribution of text data. Method 2 is motivated by previous work on feature engineering. I designed the input data matrix by artificial phrase-to-score mapping. Both results yield satisfactory level of accuracy and high true positive ratio.

II. Naïve Bayes

In this section, my goal is to provide Naïve Bayes (NB) framework that can be used to classify text documents. my main tool is Maximum Likelihood Estimation (MLE), which was developed in frequentists' view of statistical inference for unknown parameters. I start with the mathematical formalism of NB.

A. Naïve Bayes Model

Naïve Bayes is a very common algorithm for text classification. There are two main streams of modelling text by different probability distributions. The first one is Bernoulli event model. In this model, I assumed that the information of a start-up is generated by first determining whether it is invested by successful investors according to priors $P(A)$. Then, look through the dictionary, and decide whether to include each word j in the start-up information independently and according to conditional probabilities $P(X_j = 1 \mid y) = \Phi_{j|y}$.

The second model uses multinomial distribution and let us describe it using different notations. Now $x_j \in Z^+$ and $x_j \in \{1, ..., |V|\}$, where $|V|$ is the number of distinct vocabularies in the dataset. Let x_j denote the j-th word in the start-up information sentence. Similarly, we assume that a sentence of information is generated as before. Thus, the likelihood of one example is

$$\begin{aligned} P(x, y; \theta) &= P(x \mid y; \theta) \cdot P(y; \theta) \\ &= \prod_{j=1}^{d} P(x_j \mid y; \theta) P(y; \theta). \end{aligned} \tag{1}$$

where d is the number of words in that information sentence. Please note that although the likelihood looks like the same, but $P(X_j = 1 \mid y)$ is Bernoulli is the first model, while it represent multinomial distribution in the latter. Given by all the data and labels $\{(x^{(i)}, y^{(i)})$; i = 1,...,n$\}$, we can formulate the joint likelihood, given by

$$\begin{aligned} \ell(\theta) &= \prod_{i=1}^{n} P(x^{(i)}, y^{(i)}; \theta) \\ &= \prod_{i=1}^{n} (\prod_{j=1}^{d_i} P(x_j^{(i)} | y^{(i)}; \theta)) P(y^{(i)}; \theta). \end{aligned} \tag{2}$$

By Maximum Likelihood Estimation (MLE), we can estimate the parameters in the model. Warning! There is a subtle and crucial issue in implementation. At prediction time, if the classifier sees a new word that is absent in the dictionary (i,e: It has not seen it before.), the probability will result in $\frac{0}{0}$, which is undesired. To avoid this, I modified the estimator of parameters by Laplace Smoothing. See Andrew Ng's notes for more reference. Therefore, I obtained the following estimates of the parameters:

- Bernoulli event model:

$$\Phi_y = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}{n}$$

H. Wang is reading B.A. in Engineering Science at the University of Oxford, UK, OX1 3PN UK e-mail: (see hengyu.wang@wadham.ox.ac.uk).

$$\Phi_{j|y=1} = \frac{1 + \sum_{i=1}^{n} 1\{y^{(i)} = 1 \cap x_j^{(i)} = 1\}}{2 + \sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$

$$\Phi_{j|y=0} = \frac{1 + \sum_{i=1}^{n} 1\{y^{(i)} = 0 \cap x_j^{(i)} = 1\}}{2 + \sum_{i=1}^{n} 1\{y^{(i)} = 0\}}$$

- Multinomial event model:

$$\Phi_y = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}{n}$$

$$\Phi_{k|y=1} = \frac{1 + \sum_{i=1}^{n} \sum_{j=1}^{d_i} 1\{y^{(i)} = 1 \cap x_j^{(i)} = k\}}{|V| + \sum_{i=1}^{n} d_i \cdot 1\{y^{(i)} = 1\}}$$

$$\Phi_{k|y=0} = \frac{1 + \sum_{i=1}^{n} \sum_{j=1}^{d_i} 1\{y^{(i)} = 0 \cap x_j^{(i)} = k\}}{|V| + \sum_{i=1}^{n} d_i \cdot 1\{y^{(i)} = 0\}}$$

At the prediction time, a new document will be classifies as positive if

$$\frac{P(y=1|x)}{P(y=0|x)} = \frac{P(y=1)P(x|y=1)}{P(y=0)P(x|y=0)} > 1$$

B. Data Processing

The rows of the excel file represent start-ups and the columns contain various information about that start-up, including city, product category and investors of that start-up, universities, degrees, subjects previous companies and previous titles of the founders, etc. My first guess is that those successful start-ups share some attributes, which appear more frequently than in failed start-ups. Specifically, 'San Francisco' combined with 'Computer' and 'Stanford' are quite strong indicator for a successful start-up. The Naïve Bayes will automatically place more weight on those indicative words.

I combined all columns into one, labelling by whether the start-up is invested by successful investors or not, and I have split the dataset into training and testing sets. The word dictionary for proceeding analysis was created by running through all the distinct words in the training dataset. A subset of my dictionary is shown below:

{"openinvest": 1, "san": 2, "francisco": 3, "financial": 4, "services": 5, "fintech": 6, "sustainability": 7, "is": 8, "a": 9, "public": 10, "benefit": 11, "corporation": 12, "and": 13, "asset": 14, "manager": 15, "that": 16, "uses": 17, "technology": 18, "to": 19, "mainstream": 20, "customized": 21, "investing": 22, "abstract": 23, "ventures": 24, "massachusetts": 25, "institute": 26, "of": 27, "nyu": 28, "stern": 29, "school": 30, "business": 31, "princeton": 32, "university": 33, "bse": 34, "bachelor": 35, "science": 36, "bs": 37, "master": 38, "administration": 39, "mba": 40, "computer": 41, "mathematics": 42, "male": 43, "berkeley": 44, "bridgewater": 45, "associates": 46, "cofounder": 47, "chief": 48, "strategy": 49, "officer": 50, ...}.

C. Numerical Results

My Multinomial event model outperformed the Bernoulli assumption since it is specifically for text classification.
Support Vector Machines (SVMs) are an alternative machine learning model for classification. To exploit high dimensional representation of features, I used radial basis function (RBF) kernel, a.k.a: Gaussian Kernel. One important ingredient of training RBF SVMs is to choose a optimal kernel parameter, σ in

$$K(x, z) = \exp(-\frac{\|x-z\|_2^2}{2\sigma^2}).$$

My approach is to use grid-search to find the optimal value, which was 10.

TABLE I
ACCURACY ON THE TEST SET

Models	Accuracy
Naïve Bayes Bernoulli event model	82.45 %
Naïve Bayes Multinomial event model	73.13 %
Gaussian Kernel Support Vector machines	73.78 %

As shown in Table 1, even though the Naïve Bayes classifier is not the best algorithm, it works satisfactorily well on the dataset. In business, it is often the initial method to try due to its simplicity and ease of code implementation.

III. More Advanced Approaches

In this section, I engineered a more elaborate feature representation by using statistics. Then, I trained multiple advanced machine learning models to validate the high accuracy achieved by that representation. Finally, I evaluated the feature importance and decided which attribute of a start-up that determines whether it will be invested by a successful investor or not.

A. Feature Engineering

Inspired by the previous work, I seek for mapping those text data into numerical values. This is done by assigning each attribute a score, which depends on how frequent that instance of attribute appears in successful dataset. Take the attribute 'universities of founders' as an example. I obtained the rank of counts shown in Table II.

The more frequent a university appears in the successful dataset, the attribute is rewarded higher score.This process is

TABLE II
UNIVERSITIES OF FOUNDERS

Universities	Counts
Stanford University	1421
Massachusetts Institute of Technology	795
University of California Berkeley	698
University of California	589
Harvard University	481
Harvard Business School	399
Carnegie Mellon University	356
University of Pennsylvania	589
Cornell University	481
Wharton School of the UPenn	399
.....	.....

subjective. Automatic score assignment can be a future direction of research, but in this paper, I stick to my convention. The score dictionary of universities of founders is shown in Table III.

TABLE III
SCORES OF UNIVERSITIES OF FOUNDERS

Universities	Scores
Stanford University	5
Massachusetts Institute of Technology	3
University of California Berkeley	3
.....	.....
Yale University	2
Columbia University	2
Duke University	1
Princeton University	1
University of Michigan	1
The University of Texas at Austin	1
.....	.....

Other universities not on the list are scores as 0. For each start-up, there are usually multiple founders. To take this into account, I calculated the maximum, minimum, average and total scores of one attribute. By the similar manner, I created score mappings for the remaining attributes, and obtained a data matrix $X \in R^{11189\times28}$.

B. Numerical Results from Advanced Models

To begin with, I tried to visualize the high dimensional data to inspect whether my feature representation yields a clear decision boundary so that I could clearly distinguish positive and negative examples. This was confirmed by t-distributed stochastic neighbor embedding (t-SNE) plot. T-SNE is a probabilistic approach to place objects from high-dimensional space into low-dimensional space so as to preserve the identity of neighbors. It tries to keep close-by points close, so I could infer that the positive examples share lots of similarities as shown in Fig. 1.

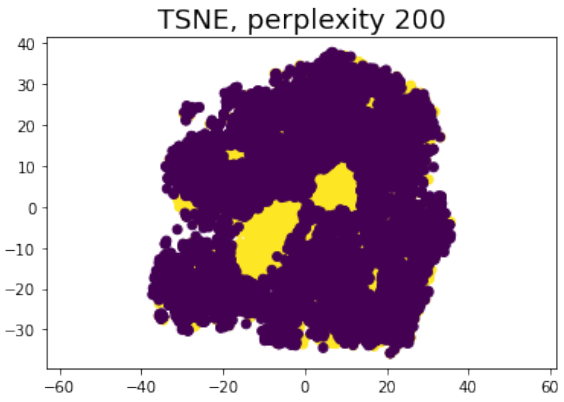

Fig. 1. t-distributed stochastic on-linear embedding plot of data matrix.

Then, I trained logistic regression, Gaussian kernel support vector machines, random forest classifier, gradient boosting classifier and deep neural network. The accuracy achieved on the test set is very encouraging, which further strengthened my argument in feature engineering as shown in Table IV.

TABLE IV
ACCURACY OF VARIOUS MODELS

Models	Accuracy
Logistic Regression	96.07 %
RBF SVMs	94.80 %
Random Forest	96.18 %
Gradient Boosting	99.19 %
Neural Network	98.94 %
Logistic Regression without all$_seed_investor$	73.38 %

A system with high recall but low precision returns many positive results, but most are incorrect. A system with high precision but low recall is the opposite, returning very few positive results, but most of its predicted labels are correct. Machine learning engineers should make trade-off between between those two conflicting objectives. Therefore I am most interested in generating models with both high precision and recall, but precision should take priority.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Positives (FP)}}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negatives (FN)}}$$

The confusion matrices for these models are attached in the Appendix A.

To figure out which feature is most indicative for a start-up to be invested by a successful investor, I evaluated the parameters in the logistic regression because it is the most interpretable model while others are 'black boxes'. As shown in Fig. 2, the 'all seed investor' ranked 1st amongst all attributes in terms of the feature importance. This is reasonable as failed dataset dose not include such attribute.

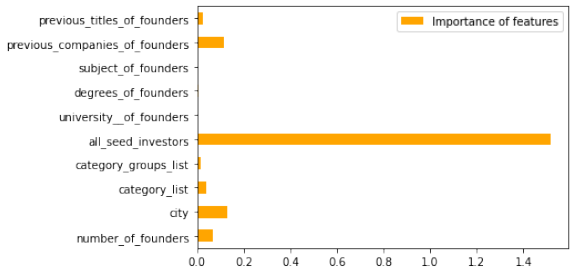

Fig. 2. Feature importance in logistic regression model.

IV. CONCLUSIONS & OUTLOOK

I proposed two frameworks for predicting successful investor's decision. The Naïve Bayes model is limited since words will not appear conditionally independently in a sentence. For instance, there is a high probability that 'Google' and 'computer' would exist simultaneity. Nevertheless, the model is ubiquitous in text classification problems and usually a good thing to start with. The models in Section III achieved surprisingly high accuracy on unseen dataset. One notable conclusion is that removing 'all seed investors' results in a drastic reduction in accuracy. This suggests that it is the most

significance indicator on deciding whether a start-up will be invested by a successful investor, which is the fundamental discovery in the project. Below several comments are provided on the future research directions.

- The dictionary contains some 'stop words'. Those words appear very frequently due to grammar but play no role in classification.
- The attribute 'short descriptions' was not encoded in the section III. Converting sentence into meaningful vectors can enrich the data matrix, and better utilise all the features.
- The hyper-parameters of models in section III were tuned automatically. From experience, they play a significant role in model evaluations.

APPENDIX A
CONFUSION MATRIX OF MODELS

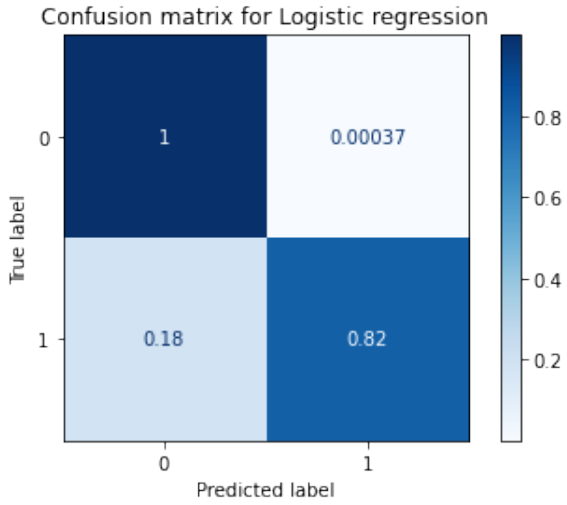

Fig. 3. Confusion matrix of logistic regression.

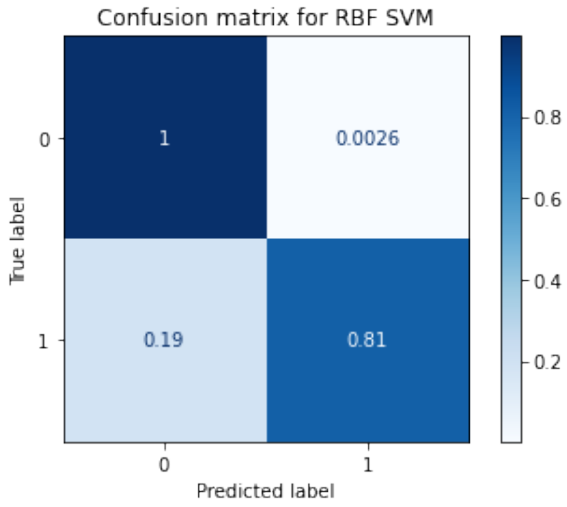

Fig. 4. Confusion matrix of Gaussian kernel Support Vector Machines.

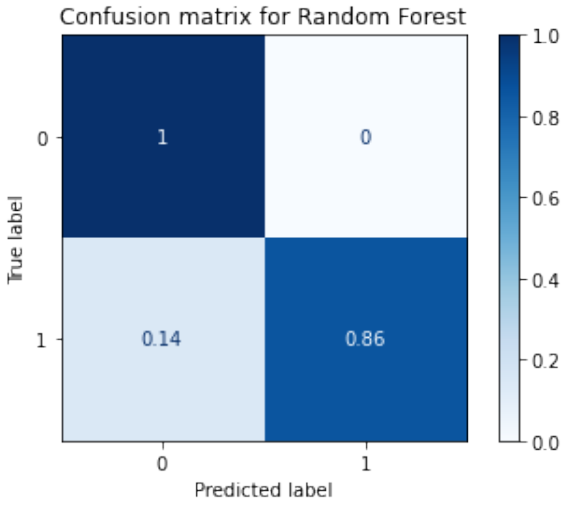

Fig. 5. Confusion matrix of random forest.

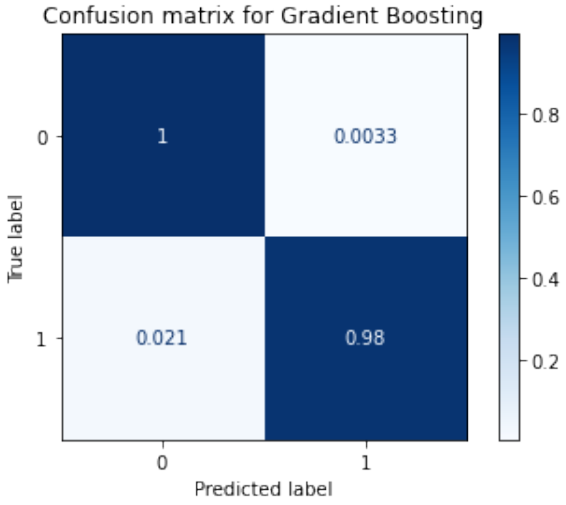

Fig. 6. Confusion matrix of gradient boosting.

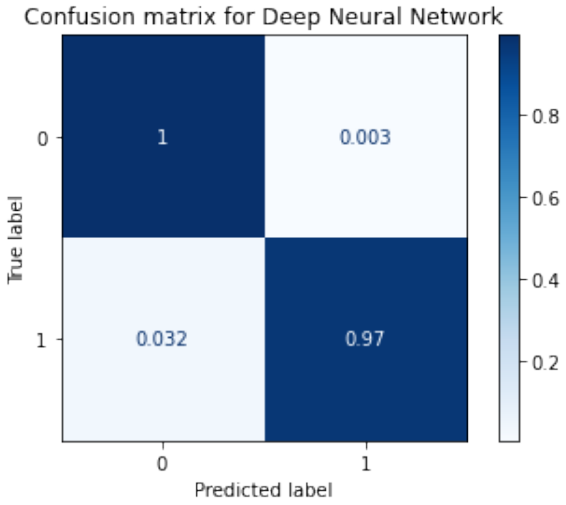

Fig. 7. Confusion matrix of a 8-layer neural network.

ACKNOWLEDGMENT

The author would like to thank Yigit Ihlamur, who obtained his M.Sc in Computer Science at the University of Oxford and co-founded Vela Partner, an early-stage focused venture capital firm, pioneering a novel use of data and machine learning, in San Francisco, US. Thanks so much for his internship offer, and invaluable guidance and feedback during my work.

REFERENCES

[1] Andrew Ng and Tengyu Ma, *Stanford University CS229 Machine Learning Notes.*

[2] Caroline Uhler, *Massachusetts Institute of Technology 6.419x Data Analysis: Statistical Modeling and Computation in Applications.*