Comparison of Reinforcement and Supervised Learning Algorithms on Startup Success Prediction

Yong Shi[1,2,3], Eremina Ekaterina[2,3], Wen Long[1,2,3,*]

[1] School of Economics & Management, University of Chinese Academy of Sciences, Beijing, P.R.China
[2] Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing, P.R.China
[3] Key Laboratory of Big Data Mining & Knowledge Management, Chinese Academy of Sciences, Beijing, P.R.China

Abstract
There has been an exponential growth in startups over the past few years. More than half of startups fail to gain funding. Predicting the success of a startup allows investors to find companies that have the potential for rapid growth, thereby allowing them to be one step ahead of competition. This paper proposes implementing a model to predict whether startup will be a failure or succeed based on many important factors like idea of the startup, place where the startup established, domain vertical to which the startup belongs, type of funding. On the preprocessed data we used several classification techniques along with data mining optimizations and validations. We provide our analysis using techniques such as Random Forest, KNN, Bayesian Networks, and so on. We evaluate the correctness of our models based on factors precision and recall. Our model can be used by startup to decide on what factors they should focus in order to succeed. Also this work aims to compare efficiency of supervised machine learning algorithms and reinforcement learning algorithms for multi-labeled classification task. Adaptations of successful multi-armed bandits policies to the online contextual bandits scenario with binary rewards using binary classification algorithms is also explored.

Key words:
CrunchBase, multi-class classification, contextual bandits, supervised machine learning

Declarations

Funding - This research was partly supported by National Natural Science Foundation of China (No.71771204, 71932008).
Conflicts of interest/Competing interests - not applicable.
Availability of data and material - available.
Code availability - available.
Author’s contributions - not applicable.

1. Introduction

Startups play a huge role in modern world economics, perhaps fast changing world leads many of them to failure. There can be several reasons like inefficient planning, inefficient way of using the funds, lack of good team to work, insufficient funds, etc. which leads to failure of startup. It is very important to increase the success rate of startups.

This work mainly aims to predict whether a startup which is currently operating turn into a success or a failure. The success of a company is defined as the event that gives the company's founders a large sum of money through the process of M&A (Merger and Acquisition) or an IPO (Initial Public Offering). A company would be considered as failed if it had to be shutdown.

This leads us to multi classification problem that usually is solved with classical supervised machine learning techniques, such as linear regression, k-nearest neighbors, naive bayes or random trees. Perhaps this techniques requires a lot of time and machine resources if implemented for big datasets thus being not efficient. Therefore in this work we redefine classification problem in terms of reinforcement learning and adopt some successful strategies and baselines that have been proposed for the multi-armed bandits setting and variations thereof to the contextual bandits setting by using supervised learning algorithms as black-box oracles, as well as exploration in the early phases in the absence of non-zero rewards.

Failures of startups have drawn massive attention and most of the companies are working on designing various kinds of prediction/futuristic models to successfully predict the fate of a new company. Few researchers have done some interesting work trying to find the success/failure patterns of a startup. One of the works discusses the success and risk factors involved in a pre-startup phase [22]. The authors focus on estimating the relative importance of a variety of approaches and variables in explaining pre-startup success. They created a framework, which suggests that startup efforts differ in terms of the characteristics of the individual(s) who start the venture, the organization that they create, the environment around this venture and the process, how it was started.

The work done in paper [5] closely addresses our problem. Research on personality characteristics relates dispositions such as risk-taking, locus of control, and need for achievement to the emergence and the success of entrepreneurship (for an overview, see [20]).

Greenewood et al. [16] have studied differences in motives as a success factor in nascent entrepreneur- ship. They find

Manuscript received July 7, 2020
Manuscript revised July 20, 2020

that women who start for internally oriented reasons, and men who start for externally oriented reasons (like perceiving a need in the market) have greater chances of successfully completing the pre-startup phase. Another work was on crowd sourcing which gets a mention in [15]. Authors focus on how successful organization can be created by crowd sourcing. Work of paper [21] focuses on developing a research program to investigate the major factors con- tributing to success in new technical ventures. Strategic alliances between companies is a good way to construct networks. Another work on new venture failure is done in the paper [19]. In this paper, the authors demonstrate two ways to investigate new venture failure - testing for moderating effects of new venture failure on the relationship between startup experience and got expertise with a sample of 220 entrepreneurs, secondly, by exploring the nature of these relationships.

Different research has been done trying to figure out several aspects of entrepreneurship and how some of them can lead to a successful company. Work done in paper [3] addresses similar issues. Another famous work is by R. Dickinson in his article [11] where he discusses the critical success factors and small businesses. Article [14] discusses a lot of problems faced by innovators. The market orientation for entrepreneurs is discussed in article [10], that also focuses on problems in terms of management. Research paper [18] discusses factors which can create successful companies.

Our work involves, the data mining analysis of more than 24000 companies, data (8361 companies with IPO, 4236 still in operation and 5600 closed/acquired companies). We modeled our data for top-10 countries (USA, Great Britain, Canada, China, India, France, Israel, Germany, Switzerland and Russia). We analyzed this data based on key factors like when the company was founded, how much seed funds it raised, how many months it took to raise the seed funds, factors which were affecting the growth of the company both positive and negative.

Experiments with more than 30 classifiers were conducted to find that many meta classifiers used with decision trees can give impressive results, which can be further improved by combining the resulting prediction probabilities from several classifiers. For the first time both supervised machine learning (regression, random forest, knn, bayes, etc.) and reinforcement learning (adaptations of multi-armed bandits policies) were applied and compared for startup classification. Our results were represented in terms of parameters like Recall, Precision and F1-score values for supervised methods and with cumulative mean reward for multi-armed bandits.

his work proposes adaptations of some successful strategies and baselines that have been proposed for the multi-armed bandits setting and variations thereof to the contextual bandits setting by using supervised learning algorithms as oracles, also exploration in the early phases in the absence of non-zero rewards, benchmarking them in an empirical evaluation using Crunchbase dataset for multilabeled classification. In this work we use cumulative reward throughout the rounds instead of accumulated regret.While this has some chance of not being able to reflect asymptotic behaviors in an infinite-time scenario with all-fixed arms, it provides some insight on what happens during typical timelines of interest.

2. Related Work

A classification task has to predict the class for a given unlabeled item. The class must be selected among a finite set of predefined classes. Classification algorithms are used in many application domains, data associated to class label are available. In all these cases, a classification algorithm can build a classifier that is a model M that calculates the class label c for a given input item x, that is, c = M(x), where $c \in \{c1,c2, ...,cn\}$ and each ci is a class label. For building model algorithm requires a set of available items together with their correct class label. Set of classified items is called the training set. After generating the model M, the classifier can automatically predict the class for any new item that will be given as input. Several classification models have been introduced such as decision trees, k-nearest neighbors, neural networks and others.

The machine learning includes supervised, unsupervised and reinforcement learning. The supervised learning provides many different regression and classification techniques to implement a machine learning model based on the labeled data. The existing solutions for this problems include all the algorithms briefly explained below([1]).

The simple logistic regression is built by plotting the graph of the dataset and then forming the boundaries separating the different classes. This is inefficient when the data is linearly inseparable and very sensitive to the underfitting and overfitting problems. It uses a stage-wise fitting process.

Algorithm 1 Logistic Regression

1: compute ω^*, the classifier obtained by regularized logistic regression on the labeled examples $(x_1, y_1), \ldots, (x_n, y_n)$

2: pick a noise vector η according to the following density function: $h(\eta) \sim e^{-\frac{n \in \lambda}{2}\|\eta\|}$, to pick such a vector, choose the norm of η from the $\Gamma\left(d, \frac{2}{n \in \lambda}\right)$ distribution, and the direction of η uniformly at random

3: output $\omega^* + \eta$

Random Forest classifier is the collection of multiple independent decision trees. The disconnected decision trees are formed by taking different starting nodes. The initial nodes are selected based on the GINI index and many different criteria. the individual trees are built independent of the other trees. when an unknown data item is given to the model, the individual outputs of the decision tree is send to a optimiser which finds the maximum favorable class label and gives it as the output. As the output of multiple trees is considered the accuracy of the model is expected to be high and resist the underfitting and overfitting problems([7]). This algorithm is very robust and handles the highly imbalanced classes very effectively.

Naive Bayes classifier assume that the features of the data items are independent to each other. The hidden correlation between the features are not addressed effectively. Naive bayes classifier is trained on the supervised learning settings depending on the probability model. The accuracy of the output can be highly dependent on the supervised learning settings and fails to find the patterns and dependency of features.

K-Nearest Neighbors (KNN) is a standard method that has been extended to large-scale data mining efforts. The idea is that one uses a large amount of training data, where each data point is characterized by a set of variables. Each point is plotted in a high-dimensional space, where each axis in the space corresponds to an individual variable. KNN has the advantage of being nonparametric. That is, the method can be used even when the variables are categorical.

Reinforcement learning can be also applied for classification problem based on model settings. Multi-armed bandits with covariates are known as contextual bandits. The main difference is that contextual bandits have side information at each iteration and can be used for arm selection, rewards also depend on covariates.

Algorithm 2 Bayes classifier

Assign class label $\hat{y} = C_k$ for some k as

$$\hat{y} = \underset{k \in \{1, \ldots, K\}}{argmax}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

The problem is very similar to multi-class or multi-label classification (with the reward being whether the right label was chosen or not), but with the big difference that the right set of labels is not known for each observation, only whether the label that was chosen by the agent for each observation was correct or not.

Algorithm 3 Naive k-means

Inputs initial set of k means $m_1^{(1)}, \ldots, m_k^{(1)}$

1: assign each observation to the cluster with nearest mean $S_i^{(t)} = \left\{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\right\}$

2: recalculate means for observations assigned to each cluster $m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$

The simpler multi-armed bandits scenario has been extensively studied, and many good solutions have been proposed which enjoy theoretical limits on their regrets ([8]), as well as demonstrated performance in empirical tests ([23]). Upper confidence bounds is one of the best solutions with theoretical guarantees and good empirical performance (such bound gets closer to the observed mean as more observations are accumulated, thereby balancing exploration and exploitation), and Thompson sampling which takes a Bayesian perspective to choose an arm, according to its probability of being the best arm. Epsilon-Greedy algorithms with variations are typical comparison baselines. The idea is to select empirical best action or a random action with some probability.

The contextual bandits has been studied in different variations - as the bandits with ”expert advise”, with rewards assumed to be continuous (usually in the range [0,1]) and the reward-generating functions linear ([10], [17]).

Approaches taking a supervised learning algorithm as an oracle for a similar setting as presented here but with continuous rewards have been studied before ([10], [4]), in which these oracles are fit to the covariates and rewards from each arm separately, and the same strategies from multi-armed bandits have also resulted in good strategies in this setting. Other related problems such as building an optimal oracle or policy with data collected from a past policy have also been studied ([3], [9], [2]), but this work only focuses on online policies that start from scratch and continue ad-infinitum.

All algorithms were benchmarked and compared to the simpler baselines by simulating contextual bandits scenarios using multi-label classification datasets, where the arms become the classes and the rewards are whether the chosen label for a given observation was true or not. Observations were fed in rounds and each algorithm made his own choice, the same time context was presented to all, and whether the chosen label was correct or not was also revealed to each one.

Algorithm 4 MAB first

Inputs const. a, b, threshold m, contextual bandit policy π_k, **covariates x**

Output score for arm $\hat{r}_k$

1: if $\left|\left\{r \in R_k \mid r=0\right\}\right| < m \text{ or } \left|\left\{r \in R_k \mid r=1\right\}\right| < m$ **then**

2: sample $\hat{r}_k \sim Beta\left(a + |\{r \in R_k | r=1\}|, b + |\{r \in R_k | r=0\}|\right)$

3: else

4: set $\widehat{r_k} = \pi_k(x)$

return $\widehat{r_k}$

3. Algorithms

Most supervised learning algorithms used as oracles can not fit to data with one value or one label (e.g. only observations which had no reward), and typical domains of interest involve a scenario in which the non-zero reward rate for any arm is rather small regardless of the covariates (e.g. clicks). In some settings this problem can be solved by incorporating some shooting criterion, and it’s possible to think of a similar application for the proposed scenario in this work if the classifier is able to output probabilities. Thus a natural adaptation of the upper confidence bound strategy is as follows:

Adaptive-Greedy ([17]) use a random selection criterion, it doesn’t require multiple oracles per arm thus shows good performance:

Algorithm 5 Epsilon Greedy

Inputs probability p $\in$ (0, 1], decay d $\in$ (0, 1], oracles $\widehat{f}_{1:k}$
1: for each successful round t with context x^t do
2: with probability (1-p):
3: select action $a = argmax_k \widehat{f}_k(x^t)$
4: otherwise
5: select action a uniformly at random from 1 to k
6: update p:=p×d
7: obtain reward r_a^t, add observation $\{x^t, r_a^t\}$ to the history for arm a
8: update oracle $\widehat{f}_a$ with its new history

Algorithm 6 Softmax Explorer

Inputs oracles $\widehat{f}_{1:k}$, multiplier m, inflation rate i
1: for each successful round t with context x^t do
2: sample action $a \sim Mult\left(softmax\left(m \times sigmoid^{-1}\left(\widehat{f}_1(x_t), \ldots, \widehat{f}_k(x_t)\right)\right)\right)$
3: update m:=m×i
4: obtain reward r_a^t, add observation $\{x^t, r_a^t\}$ to the history for arm a
5: update oracle $\widehat{f}_a$ with its new history

The choice of threshold z is problematic though it might be a better idea to keep a moving average window from the last m highest estimated rewards of the best arm (Algorithm 9).

Algorithm 7 Bootstrapped UCB

Inputs number of resamples m, percentile p, oracles $\widehat{f}_{1:k}$
1: for each successful round t with context x^t do
2: for arm q in 1 to k do
3: set $\widehat{r_q^{ucb}} = Percentile_p\left\{\widehat{f}_{q,1}(x^t), \ldots, \widehat{f}_{q,m}(x^t)\right\}$
4: select action $a = arqmax_q \widehat{r_q^{ucb}}$
5: obtain reward r_a^t, add observation $\{x^t, r_a^t\}$ to the history for arm a
6: for resample s in 1 to m do
7: take bootstrapped resample $X_s, r_s f rom X_a, R_a$
8: refit $\widehat{f}_{a,s}$ to this resample

This moving window in turn might also be replaced with a non-moving window, i.e. compute the average for the first m observations, but don't update it until m more rounds, then at time 2m update only with the observations that were between m and 2m.

Algorithm 8 Contextual Adaptive Greedy

Inputs threshold z $\in$ (0, 1], decay rate d $\in$ (0, 1], oracles $\widehat{f_{1:k}}$

for each successful round t with context x^t do

  if $\widehat{f_k}(x^t) > z$ then

    select action $a = argmax_k \widehat{f_k}(x^t)$

  otherwise

    select action a uniformly at random from 1 to k

  update z:=z×d

  obtain reward r_a^t, add observation $\{x^t,\ r_a^t\}$ to the history for arm a

  update oracle $\widehat{f_a}$ with its new history

Instead of relying on choosing arms at random for exploration, active learning heuristics might be chosen for faster learning instead. Strategies such as Epsilon-Greedy are easy to convert into active learning - for example, assuming a differentiable and smooth model such as logistic regression or artificial neural networks (depending on the particular activation functions) - Algorithm 10.

It might also be a good idea to take the arm with the smallest/largest gradient for either label instead of a weighted average according to the estimated probabilities, but in practice a weighted average tends to give slightly better results. Contextual AdaptiveGreedy also can be enriched with this simple heuristic:

Algorithm 9 Contextual Adaptive Greedy2

Inputs window size m, initial threshold z_0, decay rate d $\in$ (0, 1], oracles $\widehat{f_{1:k}}$

for each successful round t with context x^t do

  if t=1 then

    set $z = z_0$

  set $\widehat{r_t} = max_k \widehat{f_k}(x^t)$

  if $\widehat{r_t} > z$ then

    select action $a = argmax_k \widehat{f_k}(x^t)$

  else

    select action a uniformly at random from 1 to k

  if $t \geq m$ then

    update $z := Percentile_p\{\widehat{r_t}, \widehat{r_{t-1}}, \ldots, \widehat{r_{t-m+1}}\}$

    update p:=p×d

  obtain reward r_a^t, add observation $\{x^t,\ r_a^t\}$ to the history for arm a

  update oracle $\widehat{f_a}$ with its new history

Algorithm 10 Active Explorer

Inputs probability p, oracles $\widehat{f_{1:k}}$, gradient functions for oracles $g_{1:k}(x, r)$

for each successful round t with context x^t do

  with probability p:

    select action $a = argmax_k \widehat{f_k}(x^t)$

  otherwise:

    for arm q in 1 to k do

      set $z_q = \left(1 - \widehat{f_q}(x^t)\right) \|g_q(x^t, 0)\| + \widehat{f_q}(x^t) \|g_q(x^t, 1)\|$

    select action $a = arqmax_k z_k$

  update z:=z×d

  obtain reward r_a^t, add observation $\{x^t,\ r_a^t\}$ to the history for arm a

  update oracle $\widehat{f_a}$ with its new history, along with $\widehat{g_a}$

4. Empirical Evaluation

Algorithm 11 Active Adaptive Greedy

Inputs window size m, initial threshold z_0, decay rate d $\in$ (0, 1], oracles $\widehat{f}_{1:k}$, gradient functions for oracles $g_{1:k}(x, r)$

for each successful round t with context x^t do
  if t=1 then
    set $z = z_0$
  set $\widehat{r_t} = max_k \widehat{f_k}(x^t)$
  if $\widehat{r_t} > z$ then
    select action $a = argmax_k \widehat{f_k}(x^t)$
  else
    for arm q in 1 to k do
      set $z_q = \left(1 - \widehat{f_q}(x^t)\right) \|g_q(x^t, 0)\| + \widehat{f_q}(x^t) \|g_q(x^t, 1)\|$
    select action $a = arqmax_k z_k$
  if $t \geq m$ then
    update $z := Percentile_p\left\{\widehat{r_t}, \widehat{r_{t-1}}, \ldots, \widehat{r_{t-m+1}}\right\}$
  obtain reward r_a^t, add observation $\left\{x^t, r_a^t\right\}$ to the history for arm a
  update oracle $\widehat{f_a}$ with its new history, along with $\widehat{g_a}$

Each model will be evaluated by recall, precision and f1-score metrics. In case of unbalanced data classification accuracy is not enough to decide whether the model is good or not. In order to avoid this we also use precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to the all observations in actual class. Weighted average of precision and recall is known as F1 score. Intuitively F1 is not as easy to understand as accuracy, but it is much better in case of uneven class distribution.

The algorithms above were benchmarked and compared to the simpler baselines by simulating contextual bandits scenarios using multi-label classification dataset, where the arms become the classes and the rewards can be for a given label correct or not. Each algorithm was fed by observations in rounds and made it own choice, perhaps the presented context was the same to all, and revealing to each one whether the label that it chose in that round was correct or not.

Oracles were refit every 50 rounds. Experiments were performed until iterating all observations in dataset, then data was shuffled and experiments were run again. Results were gained after 10 runs.Both full-refit and mini-batch-update versions were evaluated. The classifier algorithm used was logistic regression, with the same regularization parameter for every arm.

For all contextual bandits policies was plotted cumulative mean reward over time, where time is the number of rounds/observations and the reward being whether they choose a correct label/arm for an observation/context.

5. Data And Preprocessing

The CrunchBase dataset is a raw startup data which includes different attributes of different types. The repository contains 4 CSVs derived from the "CrunchBase 2013 Snapshot" as made available by CrunchBase under CC-BY. It encompasses roughly 208,000 organizations, 227,000 people, 400,000 relationships, and 53,000 fundraising events. Data was obtained as .csv files, that are presented on Figure 1. Working with such datasets demand for rigorous data preprocessing. The consolidated raw data may include some outliers, values that are out of range, few missing values, error values - such errors will lead to wrong results. The quality of data directly proportional to the accuracy of the model. While training the model the ambiguity raises due to redundant and unimportant data. This makes the necessity of data preprocessing before training the model. This process consist of few steps:

- Data cleaning: delete NaN-values, check for outliers and resolve if needed.
- Data transformation: normalize the attribute values, aggregate. In order to reduce number of categories was performed mapping into main domains: entertainment, health, manufacturing, news, social, etc. Number of categories was reduced from 687 to 9.
- Data reduction: delete duplicates. In this work we focus on startups from top-10 countries: USA, Great Britain, Canada, China, India, France, Israel, Germany, Switzerland and Russia - they held most of the fundings and companies. Distribution of startups can be seen on Figure 2.

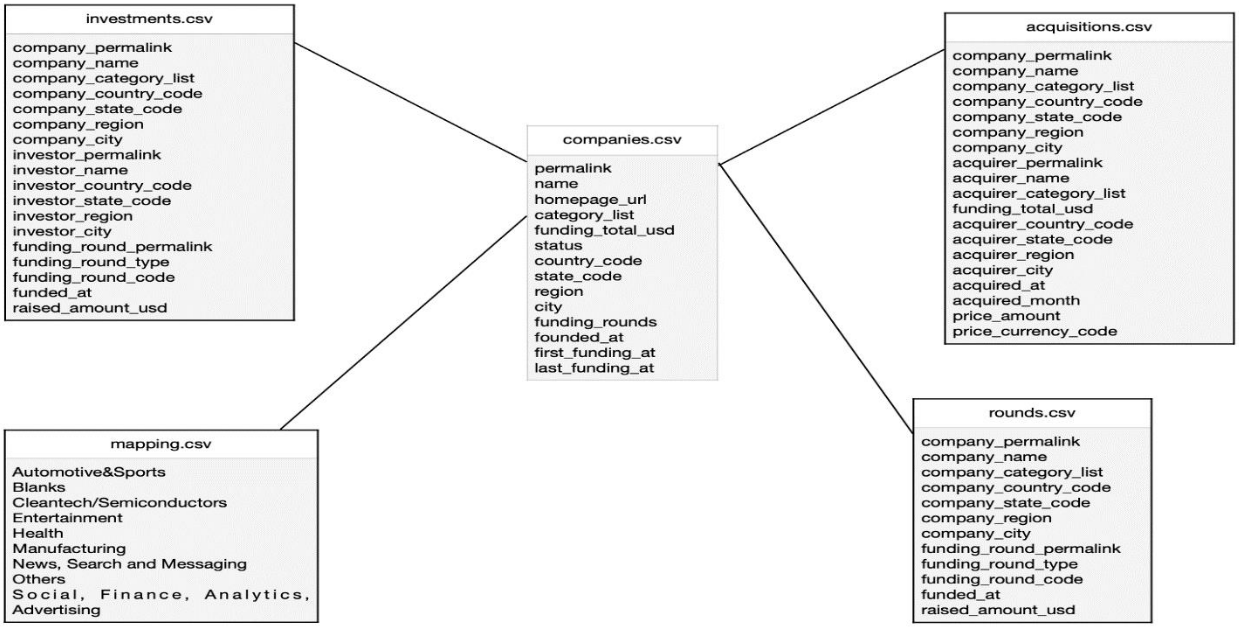
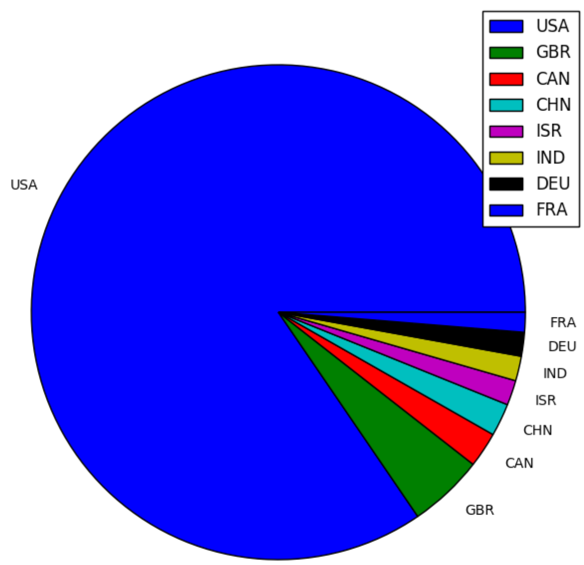


Fig. 1 Entity Relationship Diagram



	№ of startups
USA	18162
GBR	1889
CAN	923
CHN	917
ISR	555
IND	864
DEU	509
FRA	734
CHE	178
RUS	234

Fig. 2 Startup distribution by countries

For each startup, we retrieved its status (operating, ipo, acquired, closed), country code, funding rounds, funding round type (venture, angel, seed, private equity), category (from mapping.csv) and sum of total funding. Overall, our dataset consists of 24965 companies. As it can be seen from Figure 3 dataset is imbalanced, to deal with it was used algorithm which combines under-sampling the positive class with over-sampling the negative one: SMOTEENN that combines SMOTE (synthetic over sampling) with Edited Nearest Neighbours, which is used to pare down and centralise the negative cases. Startup distribution by status after preprocessing can be seen on Figure 4.

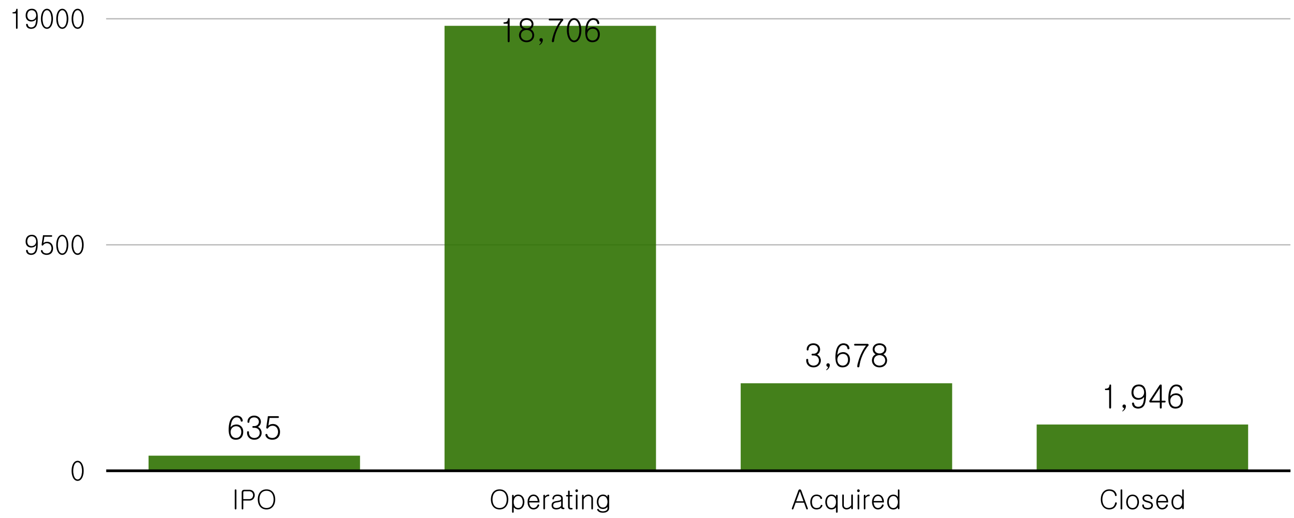


Fig. 3 Startup distribution by status in raw dataset

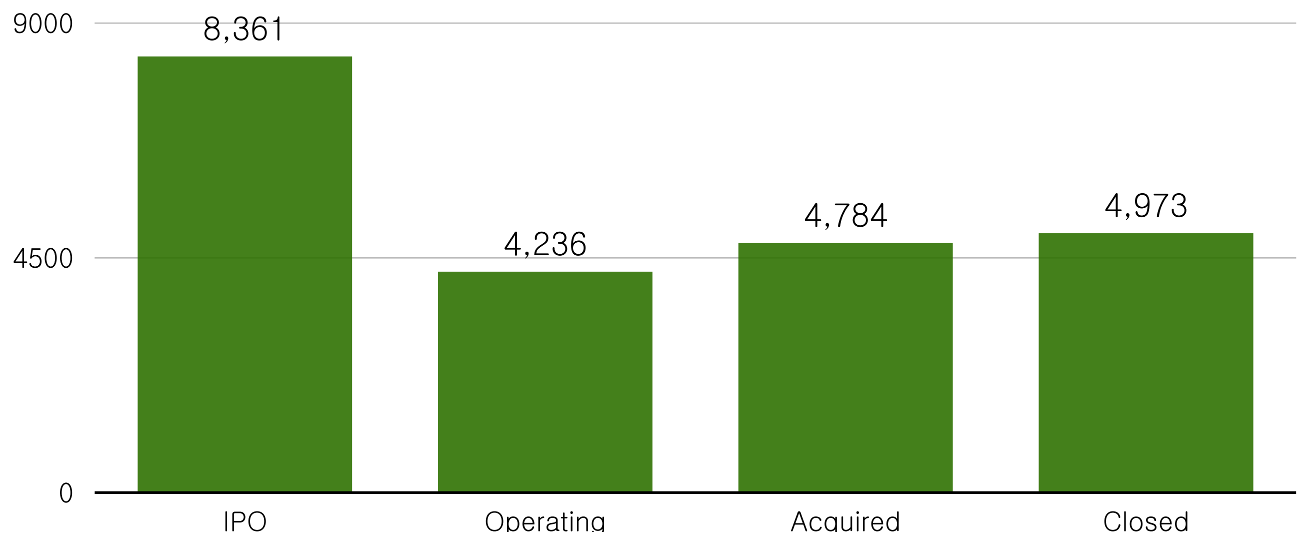


Fig. 4 Startup distribution by status in final dataset

6. Results

This work explored potential advantages of using multi-armed bandits for classification task compared to classical supervised machine learning algorithms. An empirical evaluation of supervised algorithms proved higher efficiency of kNN and Random Forest even compared to classical logistic regression. Precision and recall can be seen on Figures 5 and 6 respectively.

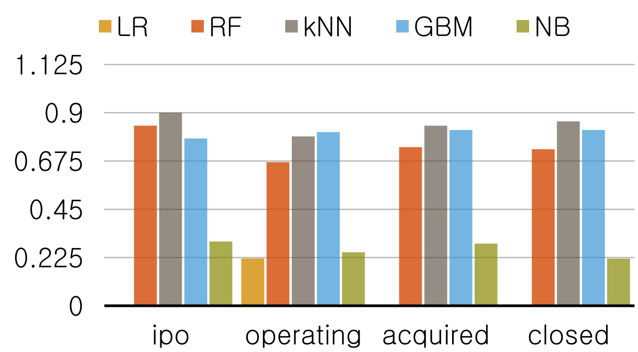


Fig. 5 Precision for supervised startup predictions

The variance in prediction precision indicates that we have unbalanced dataset. While kNN gives 0,90 precision for «ipo», that is presented much less than other classes, logistic regression is not able to define this class at all.

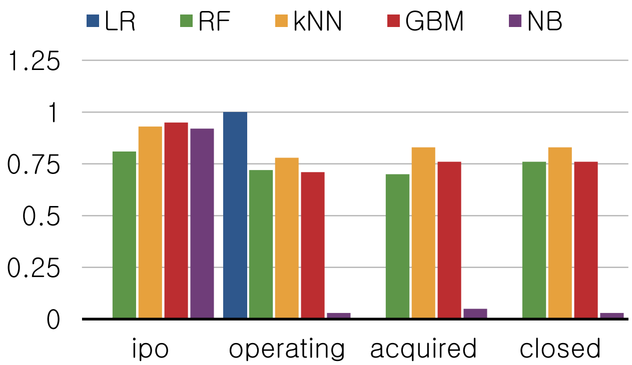


Fig. 6 Recall for supervised startup predictions

From Figures 5-7 can be seen that evaluation metrics of feature selection methods, on average, among all the classification methods that use them. It can be seen that gradient boosting and kNN are the best performers, when regression and naive bayes fail.

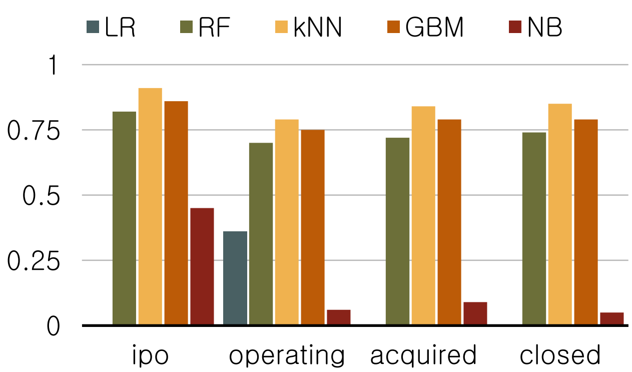


Fig. 7 F1-score for supervised startup predictions

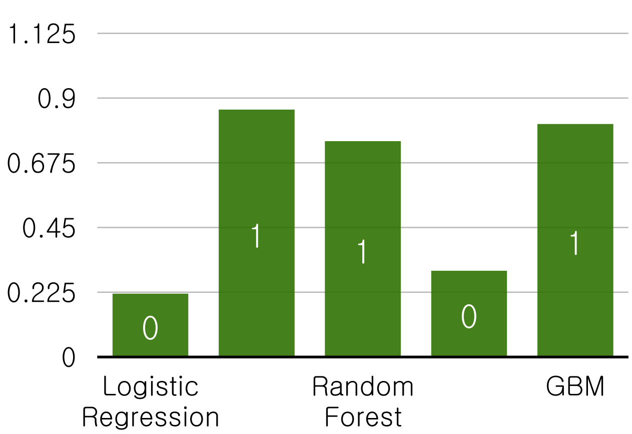


Fig. 8 Accuracy for supervised startup predictions

Figure 8 clearly shows, that logistic regression and naive bayes that are very successful with binary classification tasks fail to define classes in multi-labeled dataset. Gradient boosting is slightly worse compared to kNN, but shows higher speed performance.

In many cases empirical evaluation of adapted multi-armed bandits policies showed better results compared to simpler baselines.A further comparison (Figure 9) with similar works meant for the regression setting was not feasible due to the lack of scalability of other algorithms.

Just like in MAB, the upper confidence bound approach proved to be a reasonably good strategy throughout all datasets despite the small number of resamples used, having fewer hyperparameters to tune. Enhancing it by incorporating active learning heuristics did not seem to have much of an effect, and it seems that setting a given initial threshold provides better results compared to setting the threshold as a moving percentile of the predictions.

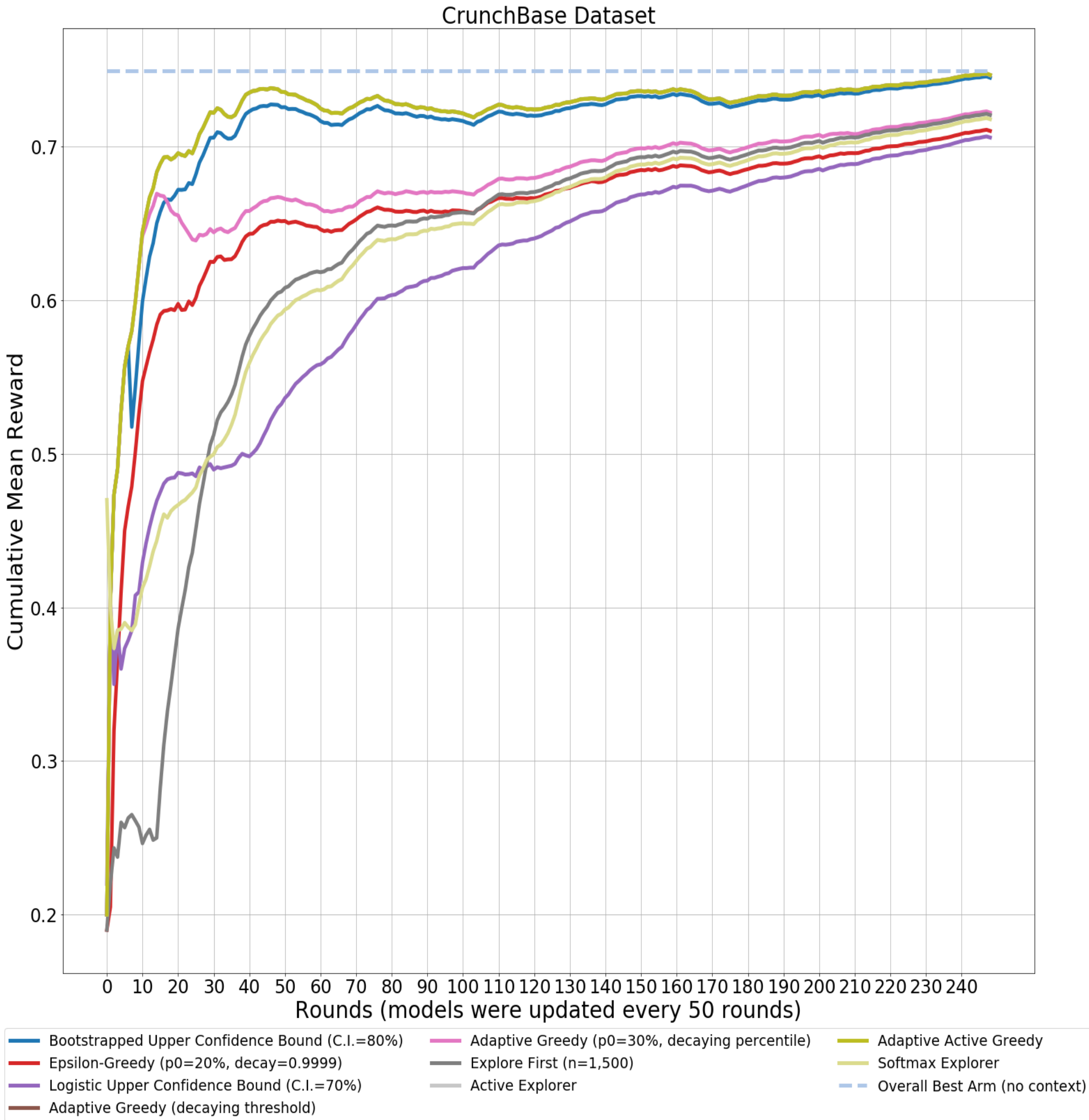


Fig. 9 Comparison of Contextual Bandit Policies

While theoretically sound, using stochastic optimization to update classifiers with small batches of data resulted in severely degraded performance compared to full refits across all metaheuristics, even in the later rounds of larger datasets, with no policy managing to outperform choosing the best arm without context, at least with the hyperparameters experimented with for the MAB-first trick. This might in practice not be much a problem consider the short time it takes to fit models to a small number of observations as done in the earlier phases of a policy.

It shall be noted that all arms were treated as being independent from each other, which in reality might not be the case and other models incorporating similarity information might result in improved performance.

CONCLUSION

In this paper was proposed and implemented model to predict future of startup and suggest improvements for future progress. Based on information about startup, such as location, industry, investment type models can predict possible expected funding range.This model gave 86% accuracy using kNN algorithm. It may alter the result if any other external factors that affect the funding the external factors can be like psychological reasons and emotional reasons of employee or candidate.

We have presented a case for an approach to RL that combines policy iteration and pure classification learning with rollouts.We believe that these initial successes will help to apply modern classification methods on other

reinforcement leaning domains. Of course, there are still many questions to be solved. This work proposed adaptations for the MAB setting, variations of contextual bandits setting by using supervised learning algorithms, benchmarking was performed using Crunchbase dataset.
Our empirical results suggest that more traditional methods such as GBM can be used successfully. However, contextual bandits have strong point in case of big datasets the algorithm itself allows division on threads and calculation on parallel kernels thus leading to time and processing costs.

References

[1] A. Agrawal, P. D. Deshpande, A. Cecen, G. P. Basavarsu, A. N. Choudhary, and S. R. Kalidindi. Exploration of data science techniques used to predict the strength of steel andIntegrating Materials and Manufacturing Innovation, 3(8):1–19, 2014.
[2] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In International Conference on Machine Learning, pages 1638–1646, 2014.
[3] T. M. Begley and W.-L. Tan. The socio-cultural environment for entrepreneurship: A comparison between east asian and anglosaxon countries. Journal of international business studies, pages 537–553, 2001.
[4] Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 129–138. ACM, 2009.
[5] J. Bru¨derl, P. Preisend¨orfer, and R. Ziegler. Survival chances of newly founded business organizations. American sociological review, pages 227–242, 1992.
[6] Alberto Bietti, Alekh Agarwal, and John Langford. A contextual bandit bake-off. 2018.
[7] Breiman. L. for Random forests. Mach. Learn., 45(1):5– 32, Oct. 2001.
[8] Giuseppe Burtini, Jason Loeppky, and Ramon Lawrence. A survey of on- line experiment design with the stochastic multi-armed bandit. arXiv preprint arXiv:1510.00757, 2015.
[9] Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi- armed bandits. In Advances in neural information processing systems, pages 273–280, 2009.
[10] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 208–214, 2011.
[11] R. Dickinson. Business failure rate. American Journal of Small Business, 6(2):17–25, 1981.
[12] Miroslav Dud´ık, Dumitru Erhan, John Langford, Lihong Li, et al. Doubly robust policy evaluation and optimization. Statistical Science, 29(4):485–511, 2014.
[13] Dylan J Foster, Alekh Agarwal, Miroslav Dud´ık, Haipeng Luo, and Robert E Schapire. Practical contextual bandits with regression oracles. arXiv preprint arXiv:1803.01088, 2018.
[14] W. B. Gartner. Who is an entrepreneur? is the wrong question. American journal of small business, 12(4):11–32, 1988.
[15] M. D. Greenberg, B. Pardo, K. Hariharan, and E. Gerber. Crowdfunding support tools: predicting success & failure. In CHI'13 Extended Abstracts on Human Factors in Computing Systems, pages 1815–1820. ACM, 2013.
[16] P. G. Greene, M. M. Hart, E. J. Gatewood, C. G. Brush, and N. M. Carter. Women entrepreneurs: Moving front and center: An overview of research and theory. Coleman White Paper Series, 3:1–47, 2003.
[17] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web, pages 661–670. ACM, 2010.
[18] D. C. McClelland. Characteristics of successful entrepreneurs*. The journal of creative behavior, 21(3):219–233, 1987.
[19] R. K. Mitchell, J. Mitchell, and J. B. Smith. Failing to succeed: new venture failure as a moderator of startup experience and startup expertise. Frontiers of entrepreneurship research, 2004.
[20] A. Rauch and M. Frese. Let's put the person back into entrepreneurship research: A meta-analysis on the relationship between business owners 'personality traits, business creation, and success. European Journal of Work and Organizational Psychology, 16(4):353–385, 2007.
[21] R. Stuart and P. A. Abetti. Start-up ventures: Towards the prediction of initial success. Journal of Business Venturing, 2(3):215–230, 1987.
[22] M. Van Gelderen, R. Thurik, and N. Bosma. Success and risk factors in the pre-startup phase. Small Business Economics, 24(4):365–380, 2005.
[23] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In European conference on machine learning, pages 437–448. Springer, 2005.