A Crowdfunding Campaign You Can't Refuse

Kuan-lin Liu
Courant Institute
New York University
New York, United States
kll482@nyu.edu

Hao-Ning Wu
Courant Institute
New York University
New York, United States
hnw244@nyu.edu

Andy Felix Monplaisir
Courant Institute
New York University
New York, United States
afm357@nyu.edu

***Abstract*—The rise of crowdfunding platforms allow us to turn our creative thoughts into real-world products. To facilitate this process and create benefits for both project owners and backers, we present a new methodology to analyze the success patterns of Kickstarter's campaigns. We dive deep into the historical data and extract insights from Tableau visualization. Furthermore, we relate Kickstarter to Amazon and Twitter datasets. Knowledge from the large corpus of Amazon can provide us for a better understanding of Kickstarter. For example, useful keywords can be generated by computing the similarity between documents. We further use machine learning and deep learning models to predict the outcome of a campaign and achieve 73.1% AUC score with random forest. Our analysis results serve as useful tips when ones try to launch a campaign.**

***Keywords*—Crowdfunding, Kickstarter, Amazon, Twitter, Big Data, Machine Learning, Deep Learning, Natural Language Processing, Recommendation**

I. Introduction

Collaboration leads to success. With the power of crowd, we could achieve anything. For example, crowdsourcing websites help people turn their novel ideas into reality by gathering resources from others. Kickstarter (https://www.kickstarter.com/) is a one of the pioneer of the crowdfunding website. Project owners can post their product ideas in the form of text, images and/or videos. In addition, they can design rewards for people pledging different amount of money. Next, they need to set a funding goal and campaign deadline. The objective of the owners is to get money exceeding the goal within the limited amount of the time. Since Kickstarter use a all-or-nothing model, the owners get nothing if they fail to achieve the goal.

In this work, we aim to increase the chance of success for the project owners. We collect three datasets: product profiles of Kickstarter, tweets on Twitter and product information from Amazon. Analytic and machine learning models are built upon these datasets to provide insights for project owners. Our summary of history data provides information like when should we launch the campaign. Moreover, our prediction model will tell the product owners how likely they will success in the end. So, they can decide how to improve their marketing strategies to more backers or investors. For example, a campaign with a lot of big words are more attractive to the customers. Finally, we build a keyword recommendation system to automatically find these keywords for the owners.

To provide better insights, the history data of Kickstarter itself may not be enough. First, its corpus is smaller comparing to other text datasets. Second, other sources may also serve as strong indicators of success, such as reactions on social media. That's why we choose to include the Amazon and Twitter datasets. We examine the synergies between these dataset carefully with Tableau visualization and conduct thorough experiments to substantiate our claims.

II. Motivation

The objective of a crowdsourcing campaign is to gather as much help as possible. However, an unsuccessful campaign can cost the project owner a lot of money, time and effort. If we can predict the success rate in the middle of or before the campaign, we can help the owner improve their marketing strategies without wasting too many resources.

Amazon Inc. sells millions of products. Just like Kickstarter, the product (project) owners try to persuade the customer to buy (invest) a product through text and graphic. Due to this similarity. It would be interesting to look into their relationship. Hopefully, we will have a better sense about what's a good campaign.

III. Related Work

Several factors have been used in previous paper to predict the success of crowdfunding projects, such as the number of Facebook friends of founders, and the absence of a campaign video. However, [1] only focused on text data and used two methods, including sequence deep neural network and hierarchical attention-based network. They chose the technology category which occupied around 24% of all the funding on Kickstarter. Surprisingly, only around 20% of the projects did meet the funding goals. In the beginning, they conducted NLP tasks on the sampled data, like mapping the text data into a vector representation. The authors found that the speech data were not large enough so that they weren't helping in prediction. Among all factors, the combination of Updates and backers' Comments help the model achieve 85-91% of accuracy. The baseline model with only Campaign section text only got to 76% accuracy.

In our project, the text from the projects' title, description, and comment, and also the words from the news will probably occupy 70% of all the information we used. Therefore, I think we can start from using the Updates and backers' Comments

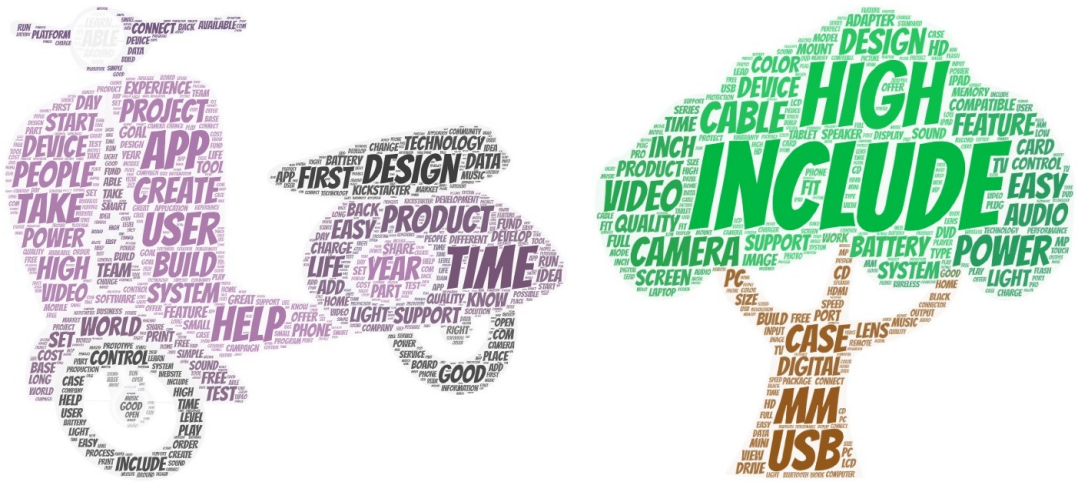


Fig. 1: Word Clouds of Kickstarter & Amazon

with the external news data, and see if we can improve the model accuracy.

In [2], the authors combined features extracted from text, image and metadata to predict whether a project will succeed or not. For text feature, BoW with TF-IDF is used; for image feature, VBoW and CNN are used. Eventually, all feature vectors are combined through a CNN model. They achieved a 7% increase in AUC by incorporating the image feature. In my opinion, their work strongly correlates with our proposed project. We might take their preprocessing approaches and evaluation metrics as references. On the other hand, there is still room for improvement. First, they didn't take into account information like animated images, comments, and social media trends. Second, the models they used are too simple. There are several more advanced NLP and CV models out there. Multimodal learning is also an active area of research. Third, they didn't compare their methods with the previous works. So, it's unclear whether deep learning is the best solution to this problem. For example, another work has achieved 98% accuracy using logistic regression. Fourth, they didn't provide insights on the common patterns of successful cases. Such information can be quite useful for creators looking for funds.

IV. Datasets

A. Kickstarter

The Kickstarter (https://webrobots.io/kickstarter-datasets/) dataset was collected from a scraper robot website, Web Robots. The website has set up a scraping machine since April 2014, and the latest data set we are using was updated on Oct 17 2019. The total data set size is about 1.1 gigabyte and totally contains 179782 unique campaigns according to the campaigns' ID. The data set was saved in the JSON format.

B. Twitter

We use the Twitter API (https://developer.twitter.com/en/docs) to query all tweets and retweets related to the campaigns we are interested in. The project titles and project urls were used to retrieved relevant data.

TABLE I: Kickstarter Dataset

Field	Type	Description
id	Int	Campaign ID
name	String	Campaign Title
state	String	Successful State
backers_count	Int	Amount of Backers
blurb	String	Description
usd_pledged	String	Funding in USD
country	String	Launch Country
launched_at	Date	Launch Date
deadline	Date	Close Date
Created_at	Date	Campaign Created Date
goal	Int	Goal of Funding
staff_pick	Boolean	Picked by Staff
mainCategory	string	Main Category
subCategory	String	SubCategory
creatorID	Int	Creator ID
url	String	Link of Campaign

C. Amazon

The Amazon dataset (http://deepyeti.ucsd.edu/jianmo/amazon/index.html) is collected by UCSD. It contains product information and user reviews span across 29 categories. In our final analysis, we only use the Electronics category, which includes 3.1G user reviews and 0.7G product information.

TABLE II: Amazon Review Dataset

Field	Type	Description
asin	String	Product ID
overall	Int	User Rating
reviewText	String	Review Text
reviewTime	Long	Unix Time
reviewerID	String	User ID
verified	Boolean	Verified Purchase or Not
vote	Int	Number of Vote

TABLE III: Amazon Product Metadata Dataset

Field	Type	Description
asin	String	Product ID
brand	String	Brand Name
description	String	Product Description
rank	Int	Product Ranking
title	String	Product Name
similar_item	Array[String]	List of IDs
also_buy	Int	List of IDs
also_view	Int	List of IDs

V. Description of Analytic

According to the Fig. 2, we found that Kickstarter started to boom dramatically in 2014. The number of the launched projects grew from 9416 to 24430 in respectively 2013 and 2014. It reached to the peak in 2015 and became stable in the following four years. Also, the failure rate increased significantly in both 2014 and 2015, and it looks like a crowdfunding bubble with a oversupply market. Therefore, we decided to analyze the project from January 2016 to October 2019 with a relatively stationary trend.

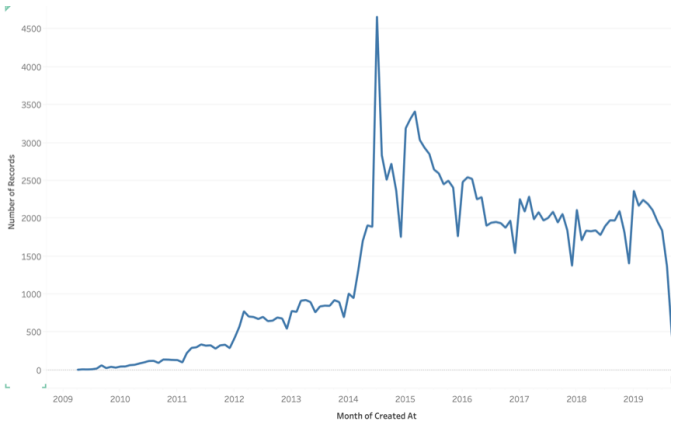


Fig. 2: Number of Kickstarter Projects Every Year

The backers can find 15 main categories on the Kickstarter website, and from the previous year, the technology projects have been generally popular. Since we are considering the relation between Kickstarter's projects and the products in the Amazon dataset, we think focusing only on the technology category could be a good start.

After filtering out the projects not under the technology category, we compute descriptive statistics and conduct exploratory data analysis on the features. It is found that the project launched from the United States occupies 56%, so we combine the project in the other countries as a new category. Kickstarter separates the technology type into more subcategories so that backers are able to look for an interesting project easier as the Fig. 3 shows. However, there are only around 1% to 2% in some subcategories. We also integrate the subcategory feature.

Closely looking at the original features in the Kickstarter dataset, a few independent variables are useless and inappropriate to be fit into the prediction models. Nevertheless, there are still some insights can be extracted, such as the month from the launched date and deadline, and the length of the short description under the title.

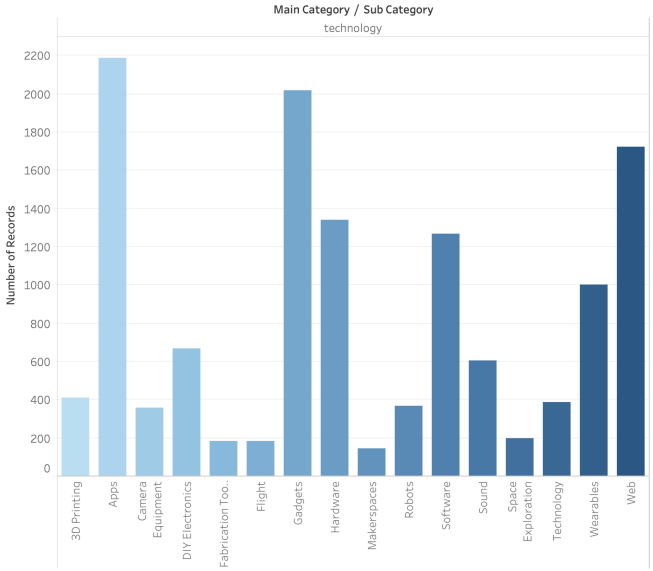


Fig. 3: Technology Subcategory

VI. Application Design

A. Application Flow

1) Get the dataset by either directly downloading, scraping by Python or using API.
2) Perform data profiling and extract, transform, load (ETL) with Spark. The processed data are stored as JSON files or hive tables for future use.
3) Analyze the processed data by its text content (description of a product) or other features (such as lauch data, funding goal, ...). We employ Spark NLP (https://nlp.johnsnowlabs.com/) to perform standard NLP pipeline includes stop word removal and lemmatization.
4) Our actuation is threefold. First, we find the most similar documents of a campaign and summarize the useful keywords that lead to a success campaign. Second, we predict the crowdfunding outcome by deep learning models with NLP feature and machine learning models with other features. Our models are built with Spark MLlib and BigDL [3] (https://bigdl-project.github.io/). Last, we construct a tableau dashboard to visualize our analytical results.
5) Overall analysis.

B. Visualization

We build a Tableau dashboard for visualization as shown in Fig. 5.

VII. Actuation or Remediation

A. NLP-based Prediction

According to Fig. 1, we can see the similarity between Kickstarter and Amazon datasets. This allows us to leverage

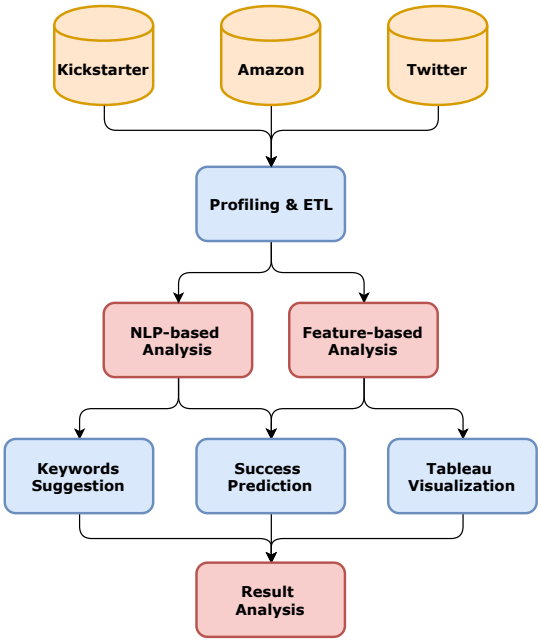


Fig. 4: Application Diagram

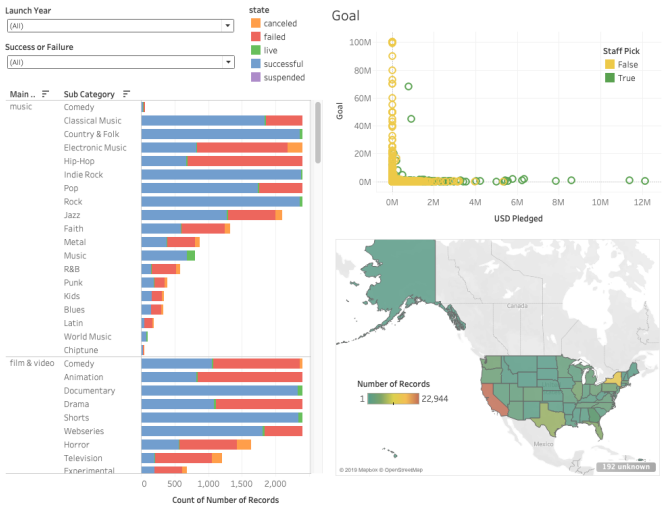
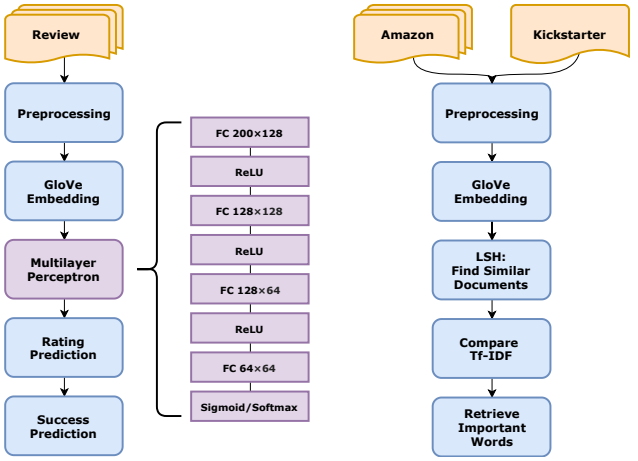


Fig. 5: Tableau Interactive Visualization

Amazon's large corpus. For example, the review data of Amazon can help us understand the comments on Kickstarter. These comments can in turn help us gain insights about the quality of a campaign. However, the comment sections on Kickstarter's website are restricted to the backers. Naturally, most of them are positive feedbacks or neutral inquiries about the product details. As a solution, we decide to treat the responses of Kickstarter's campaigns on Twitter as the review data.

The prediction flow can be summarized in Fig. 6a. Our goal is to assign a score to each tweet to reflect users' preference toward a campaign. Therefore, the Amazon dataset becomes our training set and the Twitter dataset becomes our test set. The preprocessing part includes a standard NLP pipeline. We remove all special characters and white spaces, lower each character, remove stopwords and lemmatize. Afterwards, we choose pretrained GloVe embedding (https://nlp.stanford.edu/projects/glove/) on Twitter to encode each product description as a 200d vector. Then, we feed a batch with 512 vectors into a multilayer perceptron (MLP). The final layer could be either a sigmoid or softmax layer. In the former case, the output is score within the range of $[0, 1]$. We should scale it back to $[1, 5]$ as the original Amazon ratings. In the latter case, the output is the probability for each of the 5 rating classes.



(a) NLP Prediction (b) Keyword Recommendation

B. *Feature-based Prediction*

Before fitting the feature extracted from feature engineering to the machine learning models, we still need to transform the features so that they can be understood by the machine and are more likely to generate a robust result. With the Spark's machine learning package, MLlib, we are able to build a pipeline, which automatically sets an index to a string-type variable, implements one-hot encoding, and standardizes all the independent variables to the same scale.

In order to make prediction, we randomly split the whole dataset into a training set and a testing set. 10-fold cross validation has been executed on the training set to tune the parameters. MLlib provides several traditional machine learning API. Among all, we used logistic regression with a L2 regularizer as the baseline model. Besides, two tree-based model, decision tree and random forest, are built with a combination of hyperparameters. However, MLlib only has the option of linear support vector machine without the choice of a common kernel technique, radial basis function.

C. *Keyword Recommendation*

Word usage plays an important role of getting others to back up a project. [4] shows that words can be categorized into several strategies and each have different persuasive effects on the readers. We build a recommendation system to suggest most suitable keywords for project owners. These keywords have shown effective in those high-rated products that is most closely-related to the Kickstarter projects.

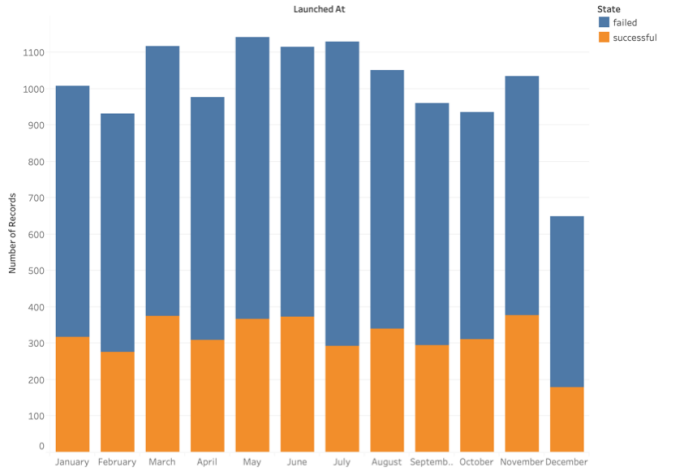


Fig. 7: Successful Rate in Each Month for Technology Campaigns

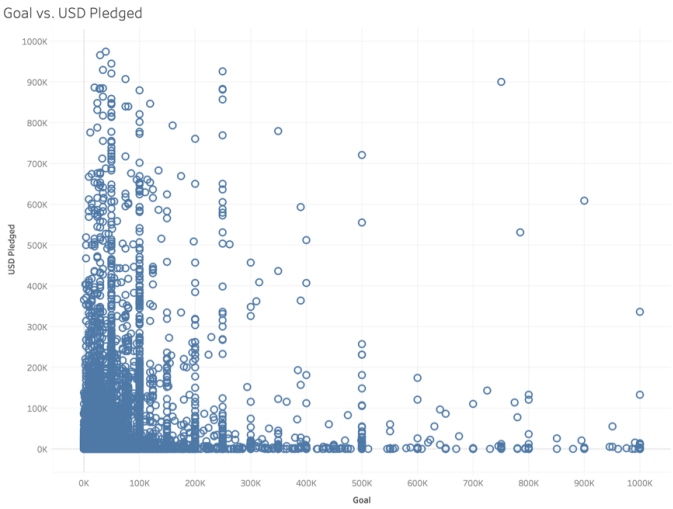


Fig. 8: Scatter Plot of Goal and USD Pledged

Given the description of a Kickstarter campaign as a query document, we would like to the retrieve the most similar documents from the Amazon dataset. The most straightforward idea is to compare its cosine similarity to each document one by one. However, this is compute-intensive. Therefore, we adopt locality sensitive hashing (LSH) to map each 200d GloVe embedding to a lower dimensional binary vector. The close documents in the original vector space will very likely be close in the new space. And we can reduce a great amount of search time.

Finally, we select the high-rated documents within top-k similar documents and compute the mean of their TF-IDF vector. If a word is more important, it will have a higher value in the resulting mean vector. Subtract the TF-IDF vector of the query document from the mean vector, we can find out which important words are not yet in the query documents. Note that a vocabulary is needed to map TF-IDF index back to the original word.

VIII. Analysis

We run all our experiments on Dumbo, a 48-node Hadoop cluster of NYU.

A. NLP-based Prediction

We observed that the raw Amazon dataset is quite messy. Its text contains a lot of typos, words composed of digits and alphabets, even redundant html code. Furthermore, there are many documents with identical product descriptions. These are the same products from different retailers, or products in the same series with different model number. It took us a lot of effort to make the data clean.

Also, the Amazon dataset is imbalanced. First, each product have different number of reviews. Second, the number of positive reviews are way more than negative reviews. Therefore, we only sample a fixed number of samples from each of the ratings in $[1, 5]$. Our final dataset contains 1M reviews. We split it into training set and validation set with the ratio 7:3.

We trained the models with Adam optimizer. For regression task, the loss function should be binary cross entropy; for classification task, the loss function should be cross entropy. After trying several combinations of hyperparameters. We found out that a 4-layer MLP with ReLU and a learning rate of 5e-4 outperforms other models. Interestingly, using techniques like lemmatization or dropout don't help us achieve better performance. Our best regression model have the accuracy of 54.8% after 20 epochs of training. And 91.5% of the predictions have score differences less or equal to 1. Our best classifcation model have the accuracy of 57.9% after 20 epochs of training. And 89.5% of the predictions have score differences less or equal to 1.

One of the issues we encounter is the use of BigDL, a distributed deep learning framework built upon Apache Spark. Compared to MLlib, it supports a wider range of neural network architectures. However, it's not as mature as other well-developed GPU frameworks. Therefore, some of the APIs don't behave as described in the document. It also have some weird constraint, e.g., the batch size should be divisible by the product of number of executors and number of cores. Furthermore, it takes up a lot of memory and time to complete the course of training. For example, executor memory and driver memory needed to be set to 10G to prevent out-of-memory error. Some of the Spark settings may also make BigDL applications extremely slower. All in all, it takes us a lot of effort to make the whole training pipeline work properly.

B. Feature-based Prediction

The failure rate is higher than the successful rate, so we have to solve the unbalanced problem on the target variable before building the machine learning models. We conduct both oversampling and undersampling at the same time on the training set. By randomly picking more successful projects and filtering out a subset of the failure campaigns, we are able to obtain a balanced training dataset. Then, we tune

the hyperparameters of each model with the grid search method. From the ten-fold cross validation, the best parameter combination with the highest AUC score is selected. Finally, we compare the performance of machine learning model with metrics, such as AUC score, f1 score, precision rate, and recall rate. According to the Table IV, we can see that our baseline model, logistic regression, has a relative good result on AUC. Except for the decision tree, all the other machine learning models have AUC score at around 73%. We summarise that using campaigns' profile information as features has its limit. That could be the reason why most of the related paper tried hard to analyze text data from the introduction section of the campaigns.

TABLE IV: Prediction Results

	AUC	Precision Rate	Recall Rate	F1 Score
Logistic Regression	73.1%	52.7%	73.6	61.4%
Decision Tree	68.9%	48.6%	58.4%	53.1%
Random Forest	73.7%	57.3%	68.5%	62.4%
Support Vector Machine	72.9%	48.6%	81.6%	60.9%

C. Keyword Recommendation

Take a Kickstarter project "ESPRESSOBin" for example, our recommendation tool will return a list of keywords sorted by their importance. "ESPRESSOBin" is a SOC board. And the returned keywords are [raid, bio, quadra, motherboard, lacie, athlon, firewire, sli, express, adaptec, readynas, esata, controller, bus, fanless, amd, quiet, overclocking, nvx, demand, nforce, bandwidth, architecture]. It's clear that we do capture the topics of the Kickstarter campaign and generate a reasonable keyword list. However, it's very hard to quantify our result. Since some of the return keywords are simply relevant technical terms. One can argue that it doesn't provide much help in terms of getting more backers. We also observe that name entities show up frequently in our list. Remove these word in advance might also be helpful.

IX. Conclusion

Using Spark SQL on Tableau, we are able to implement exploratory data analysis to extract interesting patterns from raw data. Taking the technology category as an example from the Fig. 8, we know that most of the successful technological campaigns have the goal at under 100 thousand US dollar. After building machine learning models with MLlib, we claim that the profile features on Kickstarter look like having reached their limit on prediction with around 73% AUC score. However, this is just a baseline prediction of our goal. In the near future, we will focus on the Natural Language Processing method with the tweets and retweets related to a specific Kickstarter campaign. From the creators' perspective, our analysis and prediction results can offer some detailed insights on the Kickstarter campaigns which were launched from 2009 to 2019. Especially in the technological category, we can provide the prediction of successful probability. From the backers' perspective, if they are interested in a technological product launched on Kickstarter, we are able to recommend similar products on Amazon based on the description. By this way, it helps the backers' to decide if a campaign is deserved to be invested.

X. Future Work

1) In this work, we treat the text feature and other features independently. If they can be combined in an efficient way, we can potentially get a higher accuracy.
2) The prediction models we use still have room for improvement, it's possible to use a more complex model, e.g., RNN-based models to improve the accuracy.
3) If we scrape the data day by day, we can get the trend of total funding. It would be interesting to examine how the crowd's reactions affect the growth of funding.
4) Some Kickstarter campaign information required to be scraped from the website can be important features. For example, whether the campaigns have images or videos in the introduction section can be influential since backers could have a better understanding of the projects.

References

[1] S. Lee, K. Lee, and H.-c. Kim, "Content-based success prediction of crowdfunding campaigns: A deep learning approach," in *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW '18. New York, NY, USA: ACM, 2018, pp. 193–196. [Online]. Available: http://doi.acm.org/10.1145/3272973.3274053

[2] C. Cheng, F. Tan, X. Hou, and Z. Wei, "Success prediction on crowdfunding with multimodal deep learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 2158–2164. [Online]. Available: https://doi.org/10.24963/ijcai.2019/299

[3] J. J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang, Y. Wang, X. Jia, C. L. Zhang, Y. Wan, Z. Li, J. Wang, S. Huang, Z. Wu, Y. Wang, Y. Yang, B. She, D. Shi, Q. Lu, K. Huang, and G. Song, "Bigdl: A distributed deep learning framework for big data," *CoRR*, vol. abs/1804.05839, 2018. [Online]. Available: http://arxiv.org/abs/1804.05839

[4] D. Yang, J. Chen, Z. Yang, D. Jurafsky, and E. Hovy, "Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3620–3630. [Online]. Available: https://www.aclweb.org/anthology/N19-1364

[5] C.-T. Lu, S. Xie, X. Kong, and P. S. Yu, "Inferring the impacts of social media on crowdfunding," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, ser. WSDM '14. New York, NY, USA: ACM, 2014, pp. 573–582. [Online]. Available: http://doi.acm.org/10.1145/2556195.2556251

[6] T. Mitra and E. Gilbert, "The language that gets people to give: Phrases that predict success on kickstarter," in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ser. CSCW '14. New York, NY, USA: ACM, 2014, pp. 49–61. [Online]. Available: http://doi.acm.org/10.1145/2531602.2531656

[7] Y. J. Kim, Y. G. Cheong, and J. H. Lee, "Prediction of a movie's success from plot summaries using deep learning models," in *Proceedings of the Second Workshop on Storytelling*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 127–135. [Online]. Available: https://www.aclweb.org/anthology/W19-3414

[8] J. Chung and K. Lee, "A long-term study of a crowdfunding platform: Predicting project success and fundraising amount," in *Proceedings of the 26th ACM Conference on Hypertext and Social Media*, ser. HT '15. New York, NY, USA: ACM, 2015, pp. 211–220. [Online]. Available: http://doi.acm.org/10.1145/2700171.2791045

[9] H. Yuan, R. Y. Lau, and W. Xu, "The determinants of crowdfunding success: A semantic text analytics approach," *Decision Support Systems*, vol. 91, pp. 67 – 76, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923616301373

[10] J. C. Kaminski and C. Hopp, "Predicting outcomes in crowdfunding campaigns with textual, visual, and linguistic signals," *Small Business Economics*, Jul 2019. [Online]. Available: https://doi.org/10.1007/s11187-019-00218-w

[11] J. Gera and H. Kaur, "A novel framework to improve the performance of crowdfunding platforms," *ICT Express*, vol. 4, no. 2, pp. 55 – 62, 2018, sI on Artificial Intelligence and Machine Learning. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959518301152