ESSEC BUSINESS SCHOOL

CentraleSupélec

MACHINE LEARNING
Final Project

KICK THE PREDICTION:
Predicting the success of Kickstarter fundraising campaigns

Jennifer VIAL
Sarah BOURIAL
Susan SAAL
David CHAMMA

CONTENTS

Introduction

Web mediated crowd funding is a talented paradigm and a potentially disruptive way to finance new ventures. Crowdfunding platforms offer promising opportunities for project founders to publish their project ideas and to collect money in order to be able to realize them. At the crossroad of micro-finance and social networking, crowdfunding is undergoing intense scrutiny from scholars and policymakers to understand where it positions in the chain of startup funding. Nonetheless, little is known about its effectiveness, nor the strategies that entrepreneurs should adopt in order to maximize their success rate. Given its notable prominence in driving innovation worldwide, we believe it makes an interesting topic to enquire.

We thus set our focus on one of the most-known reward-based crowdfunding platforms for creative projects – Kickstarter. Let us define Kickstarter: there are 2 different actors who play a part on the website. First, the "creators" are the heads of projects which are looking for money to grow. Each project is sorted in 15 categories such as Arts, Comic, Film, Games etc., and has a specific amount of money required. This amount is provided by people called 'backers' who are rewarded if the project goes through and is launched. Since its founding in 2009, Kickstarter has helped fund more than 40,000 projects (out of more than 100,000 created). As of writing, Kickstarter has overseen the transaction of over $569 million pledged funds.

Related Work

In this literature review, we surveyed the most relevant studies carried out in this field to date.
Chen et al. (2013) have developed a system to predict the success or failure of Kickstarter project before its completion. For this purpose, they have trained support vector machine (SVM) on campaigns' data. The dataset includes data retrieved from Kickstarter projects as well as social media sources. Final classifier of this model is able to predict campaign's final outcome with 90% accuracy. The finding of this research explores that project properties are important features in determining success of a project. Etter et. al. aimed at developing a method for predicting success of Kickstarter projects by using direct information and social media (2013). They have classified the campaigns as probable success or failure based on time series of money and information retrieved from tweets and Kickstarter's projects graph. Authors have shown the importance of social feature in predicting success of projects.

Researchers from Georgia Institute of Technology Atlanta (2014) have explored the features which lead to successfully funding Kickstarter projects. This study revealed that I Authors have explained the use of predictive phrases along with the control variables for backers and project creators to the best use of time and money.

Authors have studied and analyzed the factors affecting campaign results (Xu et.al., 2014). This literature targets the project page content and usage patterns of project updates. Semantic analysis is applied and they found discrepancies between intent of project updates and uses in practice. This analysis reveals that impact of updates rather than project details had stronger associations with campaign success.

Another paper by Rakesh et al. explains the features determining projects' success. They have expanded project features in to temporal behaviour, personal behaviour, geo-location behaviour, and social network behaviour. Using comprehensive dataset researchers have provided insights of these features and their effects on the success of campaigns. Authors have studied dynamics of Kickstarter and impact of social networks to this. Literature review reveals that, development of classification model for Kickstarter campaign has been an emerging area of research in the current decade. In regard to this, authors aimed at retrieving of the hidden knowledge from the Kickstarter campaigns and classification of these projects based on their features. To

accomplish this authors have designed a classifier for the analysis of Kickstarter campaigns by using direct information available online.

Research Goal

Clearly, crowd-sourced funding is an extremely effective and valuable approach to fundraising. However, not all the campaigns in Kickstarter attain their funding goal and are successful. Our paper aims at drawing an analysis of the relative importance of campaign details for its success in reaching funding goal, and thereby reducing moral hazard by predicting campaign success. This analysis is targeted to both campaign creators, and campaign backers. On one hand a creator would like to decrease the risk of their campaign failing. Indeed, time and money lost in preparing the fundraising is an important issue for those managers: they have to prepare a pitch, features to display on the page, making prototypes, without being sure the money will be given. On the other hand, investors are looking to minimize the risk of missing a successful campaign by erroneously predicting it will fail. Furthermore, they would rather not waste time funding a start-up doomed to fail.

With respect to the existing literature, this paper will focus on the crowdfounders' perspective. However, instead of concentrating on the motivations that lie beneath their decision to launch projects on crowdfunding platforms, it will rather attempt to discover what factors influence their ability to succeed in their fundraising efforts. From the analysis we will conduct, we hope to predict whether a Kickstarter campaign will be successful or not, and especially which features "creators" can play with in order to lead their campaign to success.

The implementation of our project was broken up into three major steps. First, we gathered data from Kaggle website as much data as we could on projects. Next, we performed some data wrangling in order to generate the most accurate predictors. Finally, we used various classifier algorithms to identify which features were the most important and to generate prediction models.

Data Pre-processing

Our dataset is from Kaggle.com, entitled 'Kickstarter Projects'. We selected the dataset spanning a timeline up to January 2018 as it was the most recent. It contains more than 379 000 observations and 14 different features, including the project name, start and end dates of the campaign, the category in which the project is sorted, number of backers, the amount requested, the final achieved amount once the campaign was over… etc.

ID	name	category	main_categc	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged	usd_pledged	usd_goal_real
1000002330	The Songs of	Poetry	Publishing	GBP	09/10/2015	1000.00	11/08/2015 12:12	0.00	failed	0	GB	0.00	0.00	1533.95
1000003930	Greeting Fro	Narrative Fil	Film & Videc	USD	01/11/2017	30000.00	02/09/2017 04:43	2421.00	failed	15	US	100.00	2421.00	30000.00
1000004038	Where is Hai	Narrative Fil	Film & Videc	USD	26/02/2013	45000.00	12/01/2013 00:20	220.00	failed	3	US	220.00	220.00	45000.00
1000007540	ToshiCapital	Music	Music	USD	16/04/2012	5000.00	17/03/2012 03:24	1.00	failed	1	US	1.00	1.00	5000.00
1000011046	Community F	Film & Videc	Film & Videc	USD	29/08/2015	19500.00	04/07/2015 08:35	1283.00	canceled	14	US	1283.00	1283.00	19500.00
1000014025	Monarch Esp	Restaurants	Food	USD	01/04/2016	50000.00	26/02/2016 13:38	52375.00	successful	224	US	52375.00	52375.00	50000.00
1000023410	Support Sola	Food	Food	USD	21/12/2014	1000.00	01/12/2014 18:30	1205.00	successful	16	US	1205.00	1205.00	1000.00
1000030581	Chaser Strips	Drinks	Food	USD	17/03/2016	25000.00	01/02/2016 20:05	453.00	failed	40	US	453.00	453.00	25000.00
1000034518	SPIN - Premi	Product Desi	Design	USD	29/05/2014	125000.00	24/04/2014 18:14	8233.00	canceled	58	US	8233.00	8233.00	125000.00
100004195	STUDIO IN T	Documentan	Film & Videc	USD	10/08/2014	65000.00	11/07/2014 21:55	6240.57	canceled	43	US	6240.57	6240.57	65000.00

Figure 1. *Snapshot of the initial dataset*

Data Wrangling

For reasons of computational easiness, but also accuracy in the fast-changing world of start-ups, we decide to restrict our dataset to Kickstarter campaigns started in 2017, which reduced our dataset to 52 200 observations. We started by dropping some obvious column. These included usd_pledged and usd_goal, both denominated in the project's local currency, which we assumed was introducing too much variation and correlation into the

model. Instead, we kept the variables usd_goal_real and usd_pleadged_real, both in USD dollars and computed using the fixer.io API.
Our dataset was relatively clean beside 4 and about 3800 missing values in features 'name' and 'usd_pledge', respectively. Moreover, we sometimes resorted to rounding value, as these presented too many decimals generating recurrent errors. We mapped the features currency, country and main category to numerical values by discretizing them.

Feature Engineering

For the purpose of our study, we decided to approach various features in different manners. Here are all the alterations that were made to the dataset.

Timing

The column 'launched' describes every campaign's start time in a very precise manner (to the second). We here assumed that all dates were aligned on the same timezone. We split this column into 6 different variables: date of launch (the date in dd/mm/yyyy format), Year, Month, Day, hours. Indeed, looking at the distribution of fundraising outcomes in the plots below, one can clearly see how the hour and day of launch seem to have an impact e.g. launching a campaign on a Wednesday or at 5pm increases chances of success.

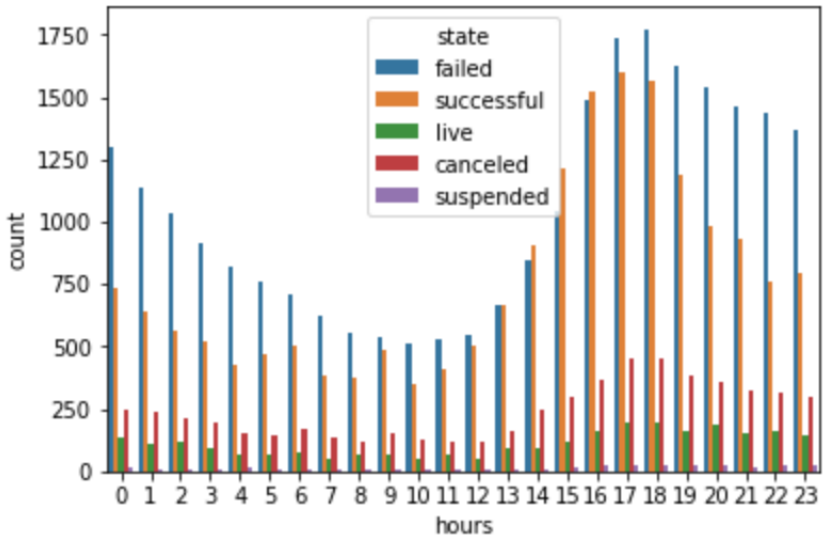


Figure 2. *Distribution of Campaign Outcomes based on the Hour of Launch*

We also computed the day of launch as a day of the week so as to account for time variations in website traffic throughout the week, as we noticed some days are more propitious to fundraising success. We subsequently transformed this feature into a categorical variable by ranking week days from 1 to 6, from Sunday to Saturday respectively. After dropping the column 'launched' all together, we computed the duration of the campaign using the date of launch and the deadline variables.

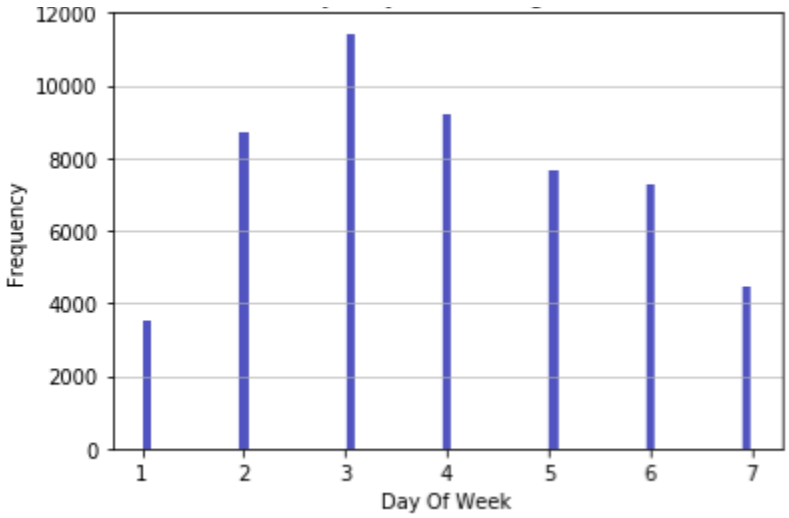


Figure 3. *Distribution of Campaign Outcomes based on the Day of Launch*

Competition

Our dataset contained both attributes of category and sub-category for every campaign. Crossing this information with other features as it seemed logical (see graph below), we created 4 new variables: average amount pledged by campaigners in every category, average goal of fundraising per category, the average backers per category, and aggregated those as features made to proxy competition, which accounts for the number of competitors every project as facing within its category and depending on the time period, which we set as being the launch month.

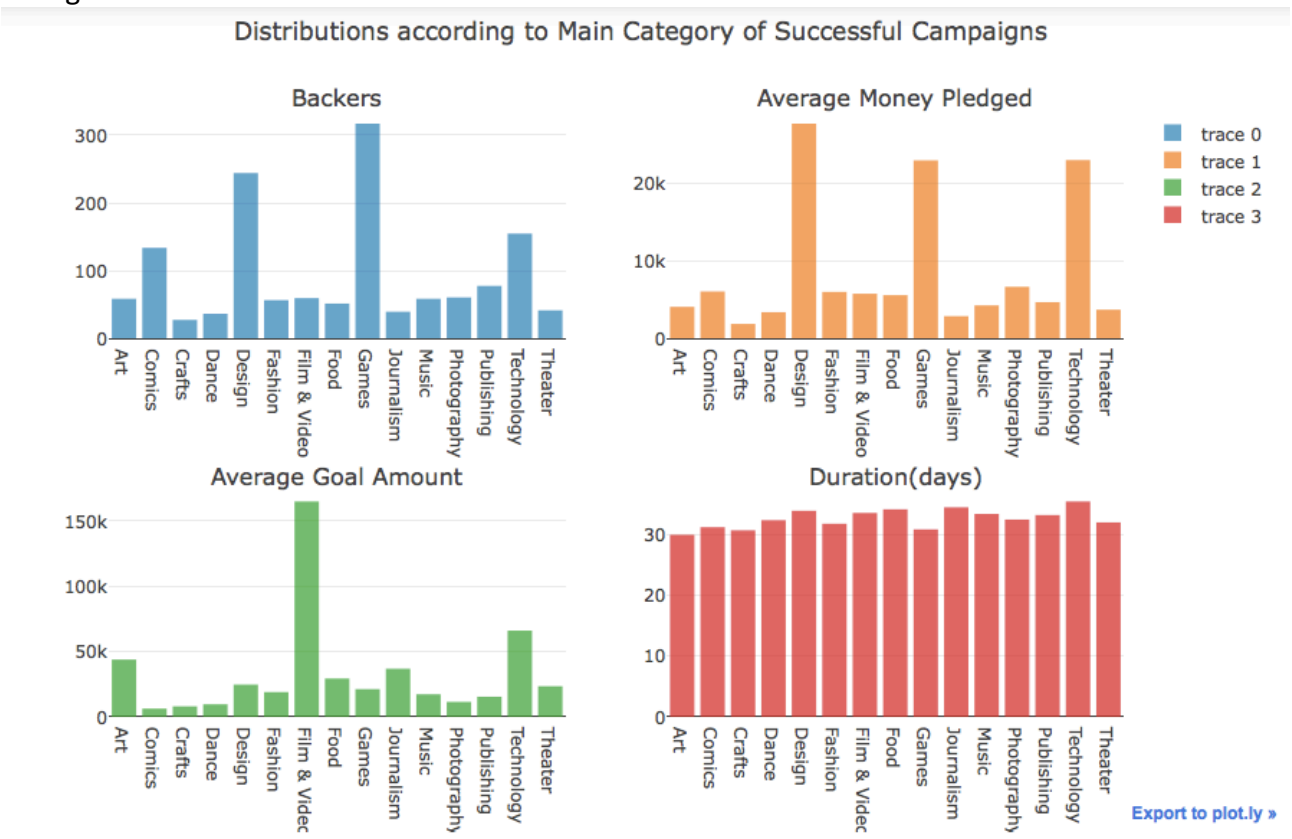


Figure 3 – *Distribution of Main Attributes depending on the Category*

Money Matters

In terms of campaign finances, we tried to go beyond the two variables describing goal amount and pledged amount. Firstly, we created a metric based on the difference between the goal amount of a given project and the average goal amount of its category. This metric was aimed at showing whether the project was above or below its category average. Finally, we binned the goal amount feature so a to account for its high variability.

Name

Without digging too far into Natural Language Processing, we attempted at taking into account influences of project's names. We computed the number of words in every project title, along with the number of letters. The rationale behind this was that a very long name would put off potential investors. Using name tagging, we also searched for the most recurrent words in successful campaigns, and similarly in failed ones. We further computed attributes about whether the name contained (1) an exclamation point, (2) a question mark, (3) was upper case. An example of our name tagging findings is displayed below:

Figure 4. *Successful versus Unsuccessful words in Campaign's names*

State

The 'state' feature described the campaign outcome of every project. As it was presenting some inconsistencies e.g. state is 'undefined', we decided to restrict to a binary classification problem: success or failure. Indeed, these two states made up about 88% of our total dataset (all years combined_, while only 11.64% was distributed across the three possibilities cancelled, live or undefined. We deleted all campaigns that were still live, and similarly that were 'undefined'. We attached the remaining alternative, 'suspended', to the 'cancelled' state as we could not find more information about a 'suspended' state. Finally, we mapped the three outcomes to a score ranking system: success is +1, failure is -1.

Eventually, we do not seem to face any correlation issue in our data as shown in the correlation matrix below. Thereby, we can move to implementing our models. Our dataset is now made of 52 200 observations, focused on 2017 with 23 features.

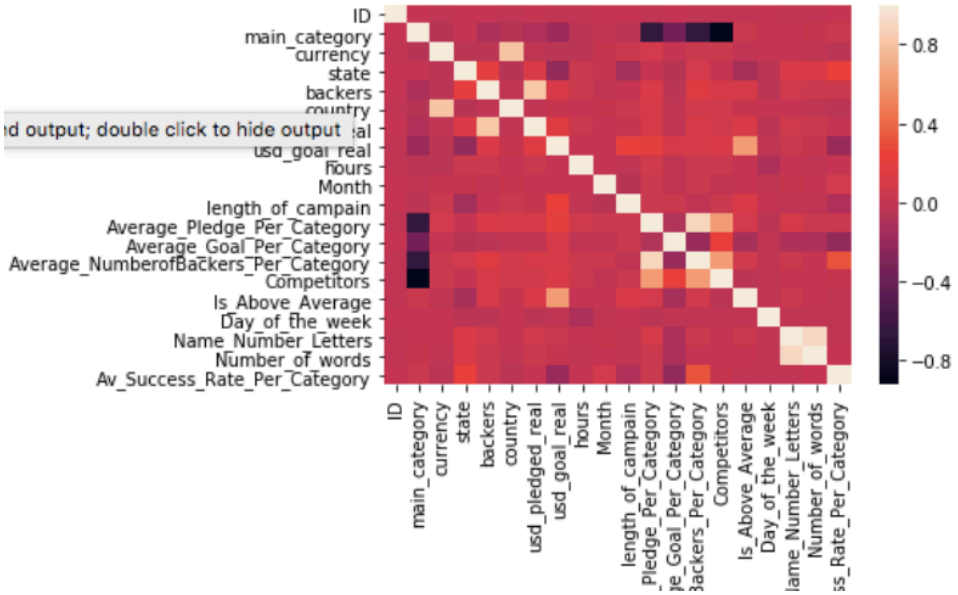


Figure 5. *Correlation Matrix*

Implementation

I. Key Findings on Influential Variables

Given our primary goal of identifying decisive features in campaign success, we also made an attempt at extracting from our features the most relevant ones by running a features selection.

Recursive Features Selection (RFE)

This method was used to select features by recursively considering smaller and smaller sets (10 sets) given an external estimator – we used Logistic Regression - that assigns weights to variables.

SelectKBest

This method was used to score the features using a Chi2 function. It computes the Chi2 statistic between each feature of X and y (assumed to be class labels) and then removes all but the k highest scoring features.

Ultimately, the features that appeared to be most relevant are the following: **Length of Name, Launch Hour, Campaign Duration, Goal Amount, Competition** (all the features that related to competition i.e. all computed averages across categories).

II. Model Selection and Performance

Kickstarter runs an "All-or-Nothing" model, meaning that each Kickstarter project consists of a target amount of funding required and a fixed time period for gathering the given amount. If the target amount is reached within the specified time, the project is considered a success, and the owner receives all of the pledged funds. Otherwise, the project is unsuccessful and the owner does not receive any of the pledged amount. For our purposes, this gives a clear cut definition of success or failure; this allows us to apply standard binary classification algorithms. We trained five different classifiers using the Scikit-learn python library.

SVM

Based on related literature from Chen et al., we decided to try Support Vector Machines on our dataset. SVM is based on principle of structural risk minimization that exhibits good generalization performance. With SVM, finding an optimal separating hyperplane between classes by focusing on the support vectors is proposed. This hyperplane separates the training data by a maximal margin. SVM solves nonlinear problems by mapping the data points into a high-dimensional space. Unfortunately, despite its accurate complexity, this algorithm presented disappointing results.

Logistic Regression

We thus switched to a more interpretable algorithm being Logistic Regression (LR). LR is one of the most popular methods used to classify binary data, based on the assumption that the value of dependent variable is predicted by using independent variables. In the model, Y is the state of the campaign we are trying to predict by observing features X which is the input or set of the independent variables. The value of Y that corresponds to the projects as either successful (Y=1) or failed (Y=-1) and is summarized by (X=x). From this definition, the conditional probability follows a logistic distribution given by (P(Y= 1 | X = xi). This function called as regression function we need to predict Y.

XgBoost

We further turned to more advanced machine learning algorithms like XgBoost (eXtreme Gradient Boosting package). It is capable of performing the three main forms of gradient boosting (Gradient Boosting (GB), Stochastic GB and Regularized GB) and it is robust enough to support fine tuning and addition of regularization parameters. Based on boosting methods, it however instead of assigning different weights to the classifiers after every iteration, fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. As it was built and developed for the sole purpose of model performance and computational speed, we figured it was a suitable model for us to try in this study.

Catboost

Alternatively, CatBoost (Cat-egory Boost-ing) is a machine learning algorithm that uses gradient boosting on decision trees. It has the flexibility of giving indices of categorical columns so that it can be encoded as one-hot encoding using one_hot_max_size (Use one-hot encoding for all features with a number of different values less than or equal to the given parameter value). Alternatively, for remaining categorical columns which have unique number of categories greater than one_hot_max_size, CatBoost uses an encoding which is similar to mean encoding but reduces overfitting. Overall, we chose this model for (1) its competitive performance, (2) because it handles categorical features automatically (can be used without any explicit pre-processing to convert categories into numbers); (3) because it reduces the need for extensive hyper-parameter tuning and lower the chances of overfitting.

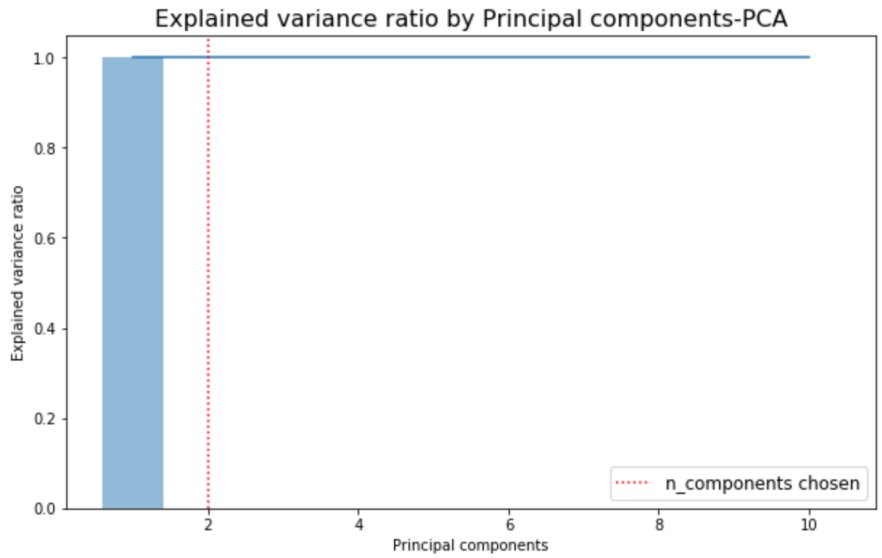
Random Forest

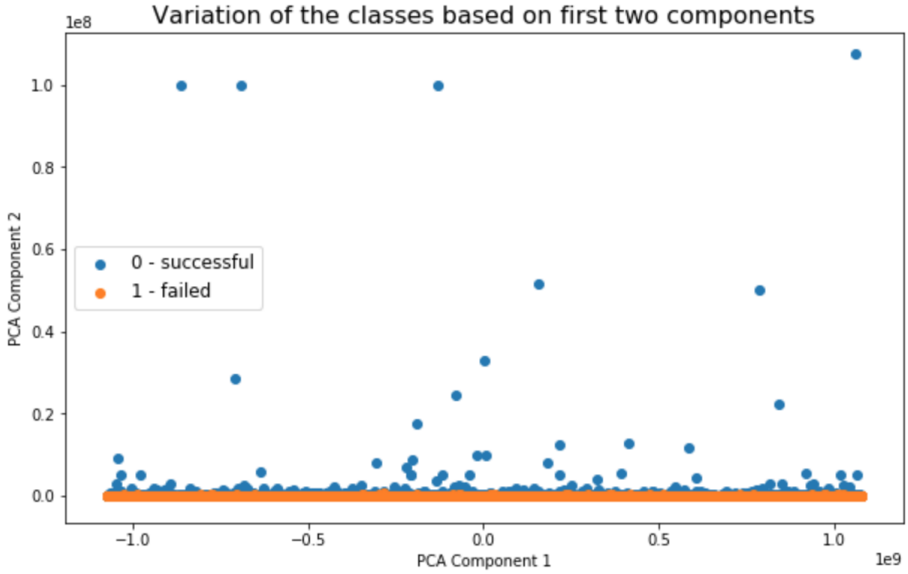
Finally, we tried a Random Forest (RF) algorithm which turned out to present the best results. RF is a classification algorithm developed by Breiman and Cutler that uses an ensemble of tree predictors. It is one of the most accurate learning algorithms and for many datasets; it achieves a highly accurate classifier. In RF, each tree is constructed by bootstrapping the training data and for each split randomly selected subset of features are used. Splitting is made based on purity measure. This classification method estimates missing data and large proportion of the data are missing it still maintains accuracy.

III. Evaluation

Parameters Tuning

We also decided to reduce the problem of over-fitting by tuning the different parameters through first PCA components for which only 2 components seemed to capture most the variation in our data.





We also attempted to tune our models by running the whole training job, looking at the aggregate accuracy, and adjusting our parameters accordingly, but with mixed results. We thus resorted to the StratifiedKFold function from Scikit-learn library, which returns stratified randomized folds method to automate our process of choosing the best combination of parameters. Similarly, to test the different parameters over the cross validation splits we used the function GridSearchCV from Scikitlearn.

Cross Validation

To get a good estimate of our out-of-sample performance, we then turned to model evaluation procedures. Our cross validation was made by splitting train_test_split functions. The idea was to partionate the dataset randomly to create sized-equal subsets, so that on top of our training and test sets, we generated complementary random validation sets on which the algorithm was further tested to evaluate its accuracy.

As a model evaluation metric, we used from the Python library Scikit-learn the function accuracy_score. This function computed the subset accuracy by defining whether the set of labels predicted for a sample matched the corresponding set of labels. We also plotted a confusion matrix to help us distinguish between the type of errors our model is making. As we can see, the rate of true predictions is 72%, and that of Type I (false positives) and Type II errors (false negatives) is 9.87% and 18.2% respectively.

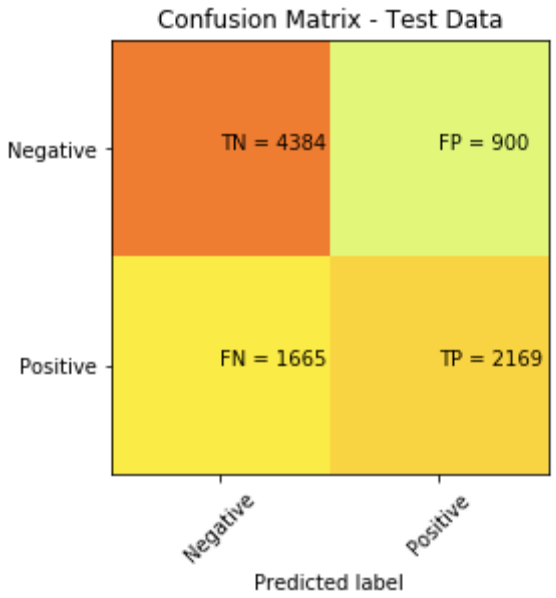


Figure 6. *Confusion matrix of our best performing model (Random Forest)*

The performances of our algorithms are described in the table below. Methods of evaluation revel that random forest is the most suitable model in this case is considered as classifier for Kickstarter campaigns.

	Support Vector Machines	Logistic Regression	XgBoost	CatBoost	Random Forest
Test Accuracy	0.64	0.67	0.68	0.69	0.67
Validation Accuracy	0.57	0.668	0.69	0.70	0.72

Figure 2. *Summary of model results*

Conclusion and Discussion

The end goal of this project was to predict the outcome of Kickstarter campaigns so as to make fruitful investments. This was broken down into three main branches: (i) identifying, generating and selecting the best features to be used for prediction; (ii) establishing the most appropriate algorithm for it and (iii) maximizing our prediction accuracy.

To find the best features, we ran a Recursive Feature Elimination and SelectKBest algorithms found around 10 features that were decisive.

Using a number of parameters tuning techniques (PCA, StratifiedKFold and GridSearchCV), we ran multiple models to find the most suitable one for our dataset and scope at hand. Comparing our algorithms on the basis of cross-validation accuracy results, our manually tuned Random Forest proved to be the most precise algorithm, leading to less than 28% error. Although being not as advanced as SVM or XgBoost, this algorithm bodes really well compared to other algorithms used, and competes with a fair amount of literature values.

Finally, our prediction accuracy proved to be satisfactory in real world setting, as it competes in precision with literature results. However, there are number of ways one could improve upon this result as outlined in the following section.

Limitations

There are a number of fronts we could explore to better our results given more flexibility and computational power. These are as follows:

• Applying machine learning approaches to the data set, particularly Neural Networks approaches for clustering and Natural Language Processing to analyze projects' names but also descriptions. We are confident these would optimize our prediction accuracy above our results.

• Harness features used in successful literature reviews, which were not included in our dataset, to get better accuracy levels. Examples here include more data about campaigners and backers personal information: fundraising history, social media sentiment analysis...

• Try crossing this data with some external datasets to account for exogenous variables that might affect campaigners and backers alike: demographics, foreign exchange, business cycle timing… etc.

References

Etter, V., Grossglauser, M., and Thiran, P. (2013) 'Predicting the Success of Kickstarter Campaigns', COSN'13, October 7–8, Boston, Massachusetts, USA.

Mitra, T. (2014) 'The Language that Gets People to Give: Phrases that Predict Success on Kickstarter', Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing, 49–61. ACM.

Rakesh, V., Choo, J., and Reddy, C. n.d. 'What Motivates People to Invest in Crowdfunding Projects? Recommendation using Heterogeneous Traits in Kickstarter', Association for the Advancement of Artificial Intelligence, [online] www.aaai.org (Accessed 17 November 2018)

Xu, A., Yang, X., Rao, H., Fu, W., Huang, S. and Bailey, B. (2014) 'Show Me the Money! An Analysis of Project Updates during Crowdfunding Campaigns', CHI'14, April 26 – May 1, Toronto, ON, Canada. ACM 978-1-4503-2473-1/14/04

Kevin Chen, Brock Jones, Isaac Kim, Brooklyn Schlamp. *KickPredict: Predicting Kickstarter Success* Dept. of Computing and Mathematical Sciences California Institute of Technology. Available at: https://pdfs.semanticscholar.org/fcfc/059870dea7f70f3acc86735dc03601d302b6.pdf (Accessed October 29 2018)