AMIR MATZ, YOAV HALPERIN & EDAN ZWICK

Predicting success of a Kickstarter campaign – Data Science project

1. Background:

Kickstarter is an online crowdfunding platform, aimed to provide a means for small businesses or individual entrepreneurs to secure investments by addressing customers directly. Kickstarter has several competitors (including IndieGoGo and many local firms around the world) but is by far the largest and most popular platform in the world. Kickstarter campaigns are all or nothing, meaning that any investments in projects that failed to meet the fundraising goal by deadline, are returned to investors and the project is canceled.

2. Goal:

Predict the success of a Kickstarter campaign, based solely on its features and attributes and before any user interaction is made (allowing to estimate success before campaign is launched).

3. Real-world applications:

Having the ability to predict the outcome of a Kickstarter campaign would be valuable for many in the crowdfunding industry. For example, for Kickstarter (which charges a 5% commission out of the money raised by successful campaigns), this would allow it to focus on promoting success prone campaigns and increase the collected commissions. Additionally, this knowledge could help identify risky campaigns and suggest the creators means to improve their chances of success.

4. Building a hypothesis:

a. The initial interest in the project was triggered by a dataset found in Kaggle[1], containing records of about 300K projects. This dataset was nicely cleaned and edited, which allowed to almost immediately run a few basic models on it. Results yielded were on par with the data set's affiliated notebooks- success rate of about 60%.

b. At this point, we tried to understand why this low success rate was a common glass ceiling for so many notebooks. Turning our attention to the dataset, we saw that although it contains a large number of projects, there were relatively few features present for each. Suspecting that existing features were not enough to provide meaningful insight, we tried to hypothesize what separates the successful from the failed.

c. Following guidance from TA Somech, looking at several dozen campaigns with different outcomes, our hypothesis was that successful and failed campaigns differ in:

- Product or cause – We noticed that some type of products (categories) commonly appear in campaigns but are rarely successful (apps for instance are seldomly funded), while others (such as miscellaneous gadgets) are more prone to success. This might be due to real world demand or just that the platform is not the ideal way to market some products

[1] https://www.kaggle.com/kemical/kickstarter-projects

(users might not be excited about purchasing an app a few months before it is actually available).

- Attractiveness of campaign page – campaigns which had an attractive page that did a good job of marketing the product, were commonly successful. This was expressed in stylish cover photos, a professionally edited marketing video, and a clear motivating textual description.

5. Dataset:

a. Though our hypothesis might sound trivial, examining the original data set's columns, we could easily see that it was lacking: it contained mostly meta-data (such as dates, sums of money requested and raised etc.), but almost no data that represents the aspects which we hypothesized were the most significant.

b. At this point we decided to start looking for a richer dataset, that would allow us to extract features that correlate with the product and page attractiveness, and found one offered by a company promoting its web crawling services[2]. This dataset seemed much more promising and contained more features with good potential of representing the key aspects which we identified. The downsides to this dataset was that it was split across several generations (several iterations of scrapings over a few years), required intensive cleaning (detailed ahead) and still lacks important features such as campaign videos or rewards.

c. The dataset contains some 300K unique projects (no complete overlap with original project) scraped between December 2015 and November 2019. After studying this dataset, we helped the other Kickstarter team migrate to this dataset (which like us started with the Kaggle one).

d. The dataset columns:

- Name – campaign's name.
- ID – campaign unique ID.
- Created at, launched at, deadline, state changed at – Time fields containing the project page creation time, deadline, campaign launce and time when final status was given.
- Currency and currency symbol – currency which goal and rewards were stated in.
- Goal– Fundraising goal in local currency.
- Category – JSON including information about campaign's category and sub category.
- Photo – JSON field containing URLs to several versions of campaign cover photo.
- Country, country displayable name –fields containing campaign's origin country.
- Location – JSON field containing campaign's origin country, city etc.
- Creator – JSON field containing (amongst others) creator ID, name and profile pic URL.
- Blurb – short textual description of project.

[2] https://webrobots.io/kickstarter-datasets/

- Slug – Dash separated string, used as projects URL in Kickstarter.
- URLS – JSON with URLs to different segments of campaign page.
- State – campaign status when mined: successful, failed, canceled or suspended.
- Source URL – Page which led to campaign page (contained the mined URL).
- Pledged, converted, USD pledged - total money raised (in original currency and USD).
- Backers count – number of eventual investors in campaign.
- Spotlight, staff pick – Indicates if campaign was featured by Kickstarter on homepage.
- Fx rate, and static USD rate – foreign exchange rate from campaign currency to USD at page mining time and launch time respectively.
- Profile – JSON field containing data that exists other columns (Photo, ID etc).
- Disable communication, friends, Is backing, Is starrable, Is starred, permissions, US type – fields that were mostly null or all contained same value.

6. **Data cleaning and basic feature extraction**:
   a. The original dataset was split between some 57 scrapings, each containing about 20 large CSV files. In total, the unzipped files weighed almost 20 GB. This relatively large size was mostly due to several long JSON fields for each record and the fact that many projects were scraped several times (but most recent scraping did not contain all previously collected data).

   b. Our first challenge was creating a workable dataset, uniting all available data. Due to the magnitude of the dataset, we had to perform the combination sequentially file by file, removing duplicates on the go. An insight we gained later, was to first filter out live projects as some projects were scraped first when they were live (and when filtering duplicates we want to keep the finalized version which we can use).

   c. The unification was done by code that sequentially downloads the dataset versions, unzips and unites them into a single data frame. We used Pandas, and as this was our first time using them, we got a chance to learn their basics. As this process was quite lengthy and entailed downloading large files, the default option in the submitted notebook is to download the united dataset we created, instead of downloading and uniting the original files.

   d. Next step was to remove columns which were clearly not going to be useful. This includes fields which are redundant (such as the currency symbol of the projects local) empty fields (such as 'friends' or 'is_starrable').

   e. Other fields that we dropped were ones which had a potential to introduce leakage. As we wish to predict the outcome of a launched campaign, we can't use any data that was updated after that time. Furthermore, some of this data is strongly correlated with success (for example, being featured on Kickstarter's landing page, dramatically improves a project's prospects). Fields that were dropped were: Pledged, converted, USD pledged, Backers count, Spotlight, staff pick.

f. Time fields: All dataset time fields (launch time, deadline) were given in UNIX time stamps and had to be converted to readable date time format. This value was kept, but we also added the explicit fields of launched year, month and the number of days from launch to deadline (project duration).

g. Categories: Campaign category gives important insights as to what product or goal is being promoted. This data was encapsulated in a JSON format and was extracted to separated fields: parent category and subcategory. Plotting the success rates by categories showed that many product types are (as suspected) at a disadvantage.

h. Currency conversion: Campaign fundraising goal was given in the campaign's local currency. To gain standardization, all data had to be converted to USD using the exchange rate that was true during the time of the campaign launch.

i. Images: Dataset records contained URL's to the project's cover photos and creators profile picture. These had to be extracted from JSON strings to be useful.

j. Creator: Each record contained information about the campaign creator. This data had to be extracted from JSON strings. Most data did not prove to be useful (promising sounding fields such as whether the creator is experienced, turned out to be corrupted or empty). The creator ID was used to extract three extra features: number of past campaigns by same user, number of failed past campaigns by same user and number of successful past campaigns by same user. To avoid leakage, this data was only cross referenced with data in the training set. These features (which were suggested in the first presentation by prof. Deutch) turned out to be some of the most significant features used.

k. Campaign location: All projects contained two (sometimes conflicting) fields: location and country. We initially tried using the country field to represent the projects location, later to find out that this field isn't accurate (many worldwide projects appeared under the US). We switched to using the location field which again required parsing JSONs. This field was occasionally null, in which case data was completed from the country field. As the majority of projects originate in the US, we separated these projects by state to gain better granularity. As this led to an influx of countries, all countries with less than 450 projects were united as "Rest of the world".

l. State: Apart from successful or failed projects, the data set contained canceled or suspended projects. As there are a variety of reasons a project could be aborted, Meaningful insights can only be drawn from projects which were completed. All projects other that the ones which were labeled as successful or failed were dropped.

7. **Initial results and models**:
   a. In order to get a sanity check and assess viability, we ran the simple models used in Kaggle notebooks on the new dataset's features which overlapped with the original data set: launched_at_month, launched_at_year, category, parent_category, destination_delta_in_days, goal. These features mostly did not require complex extraction, apart from the category data.

   b. The two chosen models were:
      - Logistic regression: does not have any hyper parameters, yielded results similar to the one's we got for the original dataset with 64% accuracy.

      - K nearest neighbors: used an additional holdout set and cross validation to choose the only hyper parameter – number of neighbors. Yielded about 67% accuracy, identical to the ones we achieved on Kaggle dataset.

   c. KNN is a relatively simple model which is not expected to perform well on complicated data, and regression is more adequate when working with continuous data. Seeing that our data contained many categorical fields we thought that using models more suited working with such data will show improvement:
      - Random forest: Number of trees set by cross validation. 69%
      - Gradient boosting: Number of estimators set by C.V. 72%

8. **Campaign images**:
   a. When evaluating projects by hand, the attractiveness of project's page is one of the factors we hypothesized were most significant in predicting success. A major part of the page is the cover photo meant to showcase the product or cause. We hypothesized that having a high-quality photo that is also aesthetically pleasing would encourage users to invest and might also suggest thought and effort were put into the campaign.

   b. In order to extract insights from the photos, we first had to have them. We wrote a short script to download all 300K photos linked from our dataset. This was run on a cloud VM we setup with enough storage and GPU that assisted later stages.

   c. Having little experience in image processing we started a self-learn process, to get some basic understanding in this field. While we found several methods of assessing technical quality, methods of extracting semantic insights from images were more challenging. State-of-the-art in image processing are convolutional neural networks, but these mostly have categorical output (classification etc.) and we were looking for a continuous metric – a score rating the image's aesthetic appeal.

d. Further reading got us to a paper by google research titled: “NIMA: Neural Image Assessment”[3] suggesting harnessing the power of CNN’s to predict a continuous aesthetic score for images. This is done by using transfer learning from proven image recognition networks which are retrained to output a 10-dimensional vector representing different scores for the image. The data this model was trained on was a database of images submitted to review by a panel of judges, paired with a 10d vector containing the number of reviews the image received of each score. Loss function was set as earth mover’s distance[4], a metric of statistical distance between the predicted vector (representing a distribution) and the real scores. Essentially what this model is trying to predict is the verdict of a panel of judges on a given image. This has very interesting properties as it can also identify controversial images (where scores vary drastically).

e. Disappointingly, the authors of the NIMA paper did not release their model to be viewed publicly, and we had to search for an implementation. After trying several models which did not yield satisfactory results[5], we found a model (mobile-net as base network) built by a German e-commerce firm to rate attractiveness of online hotel photos[6]. Using this model required changing the preprocessing pipeline to fit our photos, writing a new data loader, and re-parsing the output. We used this model with two weight sets to evaluate the technical quality and the aesthetical quality of all the campaign cover photos and all of the user’s profile pictures. The user’s cover photos did not yield any promising results so was eventually omitted from the notebook.

f. The NIMA model’s results showed that successful projects tended to have a higher rating both in the technical and aesthetical models. As the aesthetical model had a more distinct difference, this is the only score we used. The rating gap between successful and failed campaigns was not as big as we hoped, but as can be seen in the notebook, it seems as though the rating itself was credible (beautiful pictures did get a high rating and vice versa). Examining the data manually showed that the distribution similarity was caused mostly by failed projects with pictures that scored well. These were typically the worst projects, featuring beautiful and generic stock photos which usually didn’t even relate to the campaign. These campaigns are usually so lacking, that they were adequately separated using the other features. An option we considered but did not have time to implement was to classify the image content and its relevance to the campaign.

[3] https://arxiv.org/pdf/1709.05424.pdf
[4] https://users.cs.duke.edu/~tomasi/papers/rubner/rubnerIccv98.pdf
[5] Amongst others: https://github.com/titu1994/neural-image-assessment
[6] https://github.com/idealo/image-quality-assessment

9. **NLP and language**:
    a. Following our hypothesis, attractiveness of a page could also be affected by the usage of different words in the language. Certain words are associated with a positive or negative connotation to a potential donor. Others could possibly symbolize a new and trending type of products, that is still not classified as a category in Kickstarter. The main field in the dataset that we focused on is blur, the short paragraph that describes the product itself, as it is the first thing the potential investor reads. However, text for itself is not a numerical value on which we can train on, and so, we needed to find numerical metrics that describe our textual data.

    b. To factor the connotation of a word we used semantic analysis. Our chosen library was vader[7]. Vader is a lexicon and rule-based sentiment module, which is specifically attuned to sentiments expressed in social media. Given a word or a sentence, it outputs fields: positive connotation, negative connotation and a metric called compound, that represents a single unidimensional measure of sentiment for a given sentence.

    c. To consider frequent keywords in our model and their contribution to the success of a project, we used BAG OF WORDS (using CountVectorizer in sklearn). We counted how many times each word appears in a successful and unsuccessful projects, and using the bad of words theorem predicted whether the project was going to succeed or fail.
    d. In order to consider the frequency of words, we used the metrics of tf-idf. We implemented this feature using the TfidsVectorizer from the sklearn library.

    e. To increase precision and describing capability of our BAG OF WORDS and tfidf, we ignored common words (stop words), which we acquired through the nltk module.

10. Results:

Having extracted features we believed directly related to a project's success, we were ready to re-evaluate our models. The features we ended up using are: launch month, launched year, category, parent category, project duration, goal, NIMA score, blurb positiveness score, blurb negativeness score, blurb compound score, tfidf and Bag of Words vectors, creator's count of successful and unsuccessful project and the count of previous projects. We also decided to represent category and parent category as one hot encoding instead of an ordinal, so that we would not confuse our model into thinking that different categories or subcategories have some kind of internal hierarchy or order when we clearly don't have it. The final results were: accuracy: 84% with recall: 81 %.

11. Nontrivial components:
    a. Dataset cleaning and preparation: Migrating from the original cleaned data set proved as a non-trivial task, as it required joining an enormous amount of data. Even after the unification

[7] https://github.com/cjhutto/vaderSentiment

process, many of the records were corrupt (especially the JSON fields which were often nonstandard and contained missing brackets, unclosed quotations etc.).

b. NLP: having a strong interest in this field (but little experience), the search for appropriate and useful tools for this project was enjoyable and educational. Examining several options and setting on sentiment analysis, introduced a novel approach not previously used for this problem.

c. Image: This task was nontrivial in a few aspects: the self-learn process required, the solution complexity and the amount of data we had to manipulate. This approach wasn't previously used a Kickstarter data set, and consisted a great learning opportunity for us.

12. Comparing to existing work:

a. Having our finished work, we can compare it to the existing notebooks on Kaggle. The most significant difference is the accuracy improvement. Most Kaggle notebooks had accuracy between 60% and 74%, while we achieved a prediction accuracy higher than 80%[8]. This is more significant considering that most notebooks (especially the ones with the higher accuracy) introduced leakage to their model by using fields which were updated at project's end (such as backers count etc.).

b. Further novelty introduced by our project was the transition to the new dataset and the more thorough feature extraction, most notably the creator history and improved location granularity.

c. The main nontrivial features that we extracted were the NLP assessment of the campaign's description and the image processing and scoring. On a personal note, these were also the areas where we felt the best learning experience, having gotten a foothold in new fields.

13. Working with friends

The final step was to include the work of the other Kickstarter team. After thorough consultation with them, we have decided to add the following features: category_count_7_days (number of products of the same category that were uploaded in the last 7 days), min_reward_price, max_reward_price, avg_reward_price (considering the value of rewards that were offered to potential donors for their contribution to the specified project), num_of_rewards_options, and photo_score (based on the aesthetic quality of the picture). Using the LGBM classifier, we received 90% accuracy with 81% recall. Hooray!

[8] https://www.kaggle.com/srishti280992/kickstarter-project-classification-lgbm-70-3
https://www.kaggle.com/kabure/kickstarter-projects-eda-stat-tests-pipeline