Kickstarter success prediction using Regression and NLP

ClickStarter
Csukás Tamás, Sillye Márk, Tóth Attila

People have always been obsessed with predicting future events, whether next week's lottery numbers or future health risks, knowing the future is always beneficial.
Kickstarter is the world's largest crowdfunding platform for projects from all around the globe. People who want to get funded create projects and set goals for them. They share these projects with the users of the website, who can then browse them and fund the ones they like. These projects have a deadline, if they reach their goal by the deadline, they are labelled successful, if they fail to reach their goal, they are labeled failed.
Creators want to know whether they can reach their goals in time and make their visions reality. Our Deep Learning project tries to predict the success of a project using the data that the user provides to Kickstarter. The model is trained on historical Kickstarter data and uses a mix of Dense Neural Networks and Convolutional Neural Networks to predict the success.

Az embereket mindig is érdekelte, hogy mit rejt a jövő. Ha tudhatnánk, mik lesznek a jövő heti lottószámok, vagy milyen betegségeink lesznek, fel tudnánk készülni a jövőre.
A Kickstarter a világ legnagyobb közösségi finanszírozó oldala, amely segíti a világ minden tájáról érkező az innovatív projektek anyagi támogatását. A projekt tulajdonosok megosztják az oldalon a projekt ötleteiket, illetve kitűznek egy anyagi célt, amivel a projekt megvalósítható. A weboldal felhasználói pedig böngészik a projekteket, és ha találnak olyan projektet, ami felkeltette az érdeklődésüket, pénzzel támogatják azt. Minden projektnek van egy határideje, ameddig össze kell gyűlnie a pénznek. Ha összegyűlik, akkor sikeresnek titulálják a projektet, ha nem, akkor pedig sikertelennek.
A projektek megalkotói tudni szeretnék, hogy el tudják-e érni a céljaikat és ezáltal meg tudják-e valósítani álmaikat. A Deep Learning projektünk megpróbálja megjósolni, hogy a projekt sikeres lesz-e a felhasználó által megadott adatok alapján. A modell a Kickstarterről lementett adatokból lett tanítva és sűrű, illetve konvolúciós neurális hálókat használ a siker megjóslására

Introduction

Kickstarter projects are fascinating, but putting a lot of effort into something that eventually fails, is painful. We were curious whether it is possible to predict a project's success, and if it is, with what certainty can a neural network predict it.

About Kickstarter prediction

We found a dataset on Kaggle and saw a couple of kernels using different regressions and statistical approaches with an accuracy of around 60-70%, which at first seemed pretty low. We were curious about how Kickstarter predictions can be improved.

Networks

We used a mix of fully-connected and convolutional networks to predict the success of projects.

Implementation

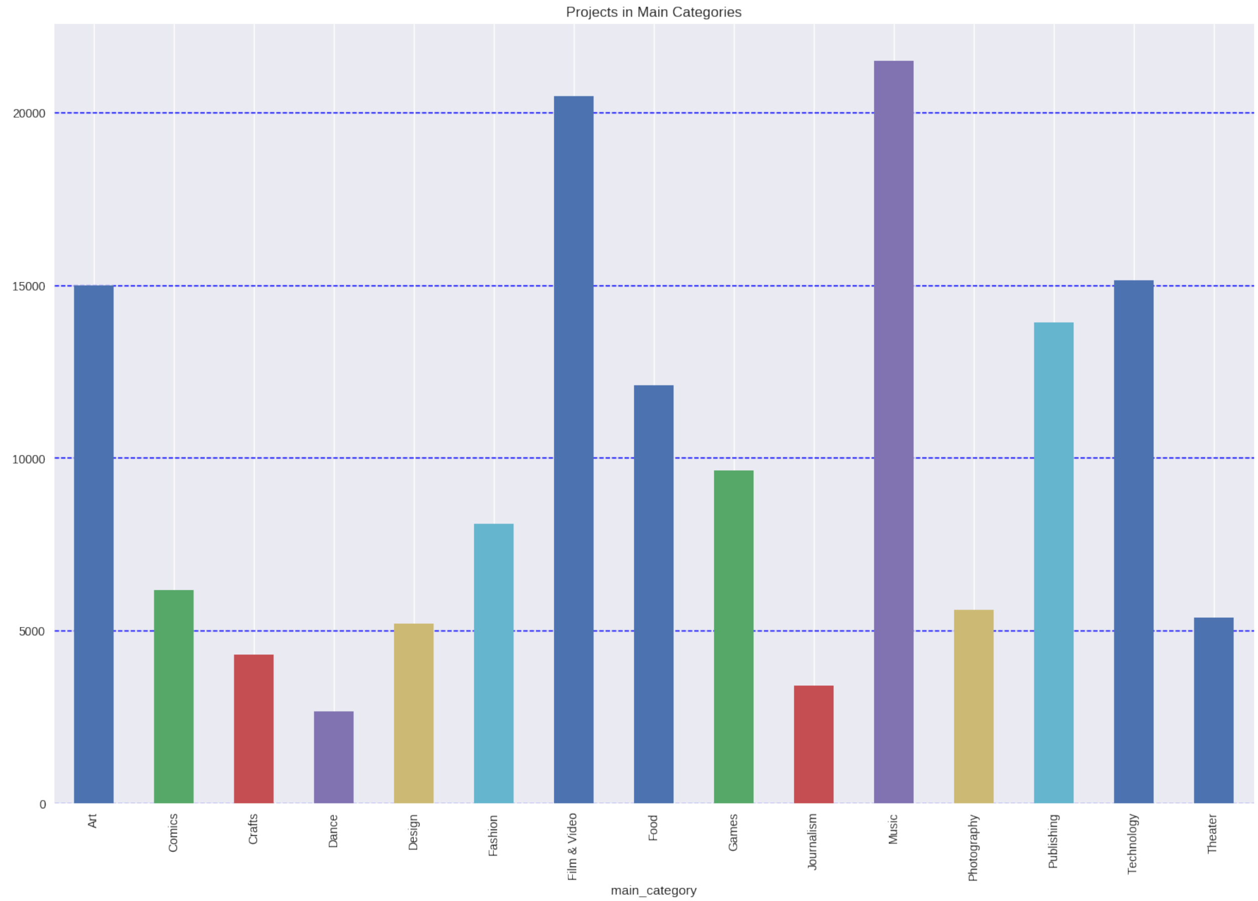
Our idea was to improve this accuracy using NLP. The Kaggle dataset however contained only the name of the project, which is not enough information.

Finding proper datasets

We found another dataset, mined by a web crawler, which was a rawer dataset consisting of 51 separate CSV files. We merged these files and extracted the necessary description field called blurb.

We pre-cleaned the datasets separately, removing any rows that contained incomplete rows.

We then joined the datasets on the project IDs and started cleaning them.



1. Figure Number of projects per main category.

Cleaning the datasets

First, we removed the whitespaces from the header names. Then, we removed columns that were unnecessary.

The spotlight and staff pick fields had to be removed, because project owners, who have been successfully funded will be awarded spotlight, so spotlight has a correlation of 1 with the success of the project. We tried to teach the network while having the spotlight field as input data, and it had an accuracy of 99+%, meaning it learned, that the spotlight fields correlates with the success.

The staff pick field is also awarded during the project, so it is not an input data either.

We also removed the different currency versions of the goal and pledged fields, and only kept the real US Dollar values.

We also calculated a new field called duration from the project launch and project deadline fields and encoded the categorical features with one-hot encoding.

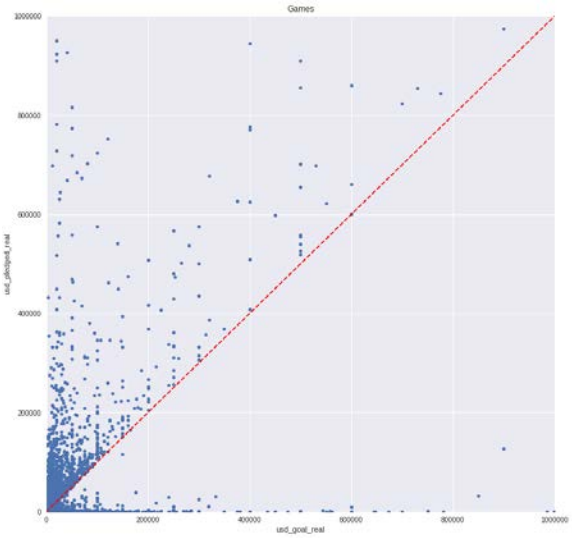
After that, we tried to process the name and description (blurb) fields with different methods. First, we tried to remove unnecessary characters with Regex, which didn't produce good results.

Then, we tried to clean it by removing stop words and non-alpha characters. This didn't produce good results either.

Finally, we decided to use a library called langdetect to detect the language of each project's text and throw away the non-English ones (which were about 3000 projects at that point), and then we used a library called Pattern to parse and tokenize the text and to find stem words, then removed words with non-alpha characters and the stop words.

Visualizing the data

We visualized, how many entries the dataset contained per categorical features, displayed how much USD projects put as goals and then received per main category. We also displayed the top 50 most common words that the project names and descriptions contained.



2. Figure Scatter diagram of game projects goal and pledge amounts on axis X and Y. Any dot above the red dotted line is successful.

Teaching

First, we tried to teach the network using a weak hashing vectorized representation of the project name and project description. This did not improve the accuracy at all, so we removed these fields and tried to teach the network using only the numerical values and the one-hot encoded categorical features.

We reached a validation accuracy of around 0.75 this way.

Hyper parameter optimization

We tried to optimize the network using Talos.

We set different learning rates, neuron sizes, hidden layers, dropout values, activation functions, batch sizes, and ran Talos to optimize and find the best parameters to learn.

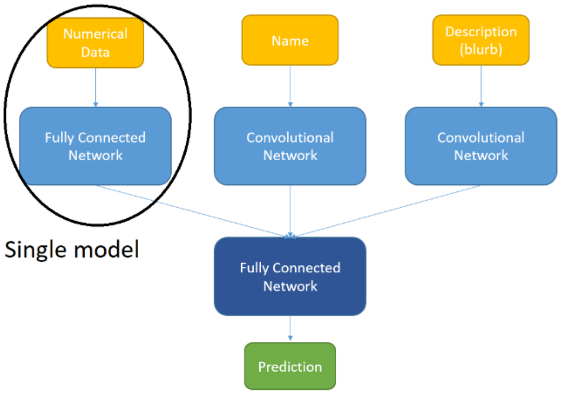
After Talos ran and tried out 300 different combinations, it found that the best parameters are:

Hidden layers	0
Batch size	64
Activation	ReLU
Learning rate	0,01
Dropout	0,4
Optimizer	Adam
First Neuron size	2048

This means, that it does not need any hidden layers, which means that it's technically a linear regression.

After this, we built a convolution network to predict the success. We did this by first building an embedding matrix from the name and blurb fields using GloVe and then feeding it into the convolutional network using an Embedding layer as its first layer. The first runs showed a validation accuracy of 0.64375 for the name CNN with a mean absolute error of 0.4349. This is a terrible result; it means that the network wasn't able to learn much from the name alone. The blurb CNN was able to reach a validation accuracy of 0.68255 and an absolute error of 0.4028, which is not great either, but it shows some promise.

After this, we created a fourth model, that had the output data from the three different models as its input and then produced a single prediction as its output. The fourth model tried to learn, which network to trust and which to avoid.



3. Figure The Single model, consisting of a fully-connected network and the combined model, consisting of 2 fully-connected and 2 convolutional networks.

Validation

We ran the predictions on the test data.

The combined model we built provided decent, but not great accuracy on the data. The best validation accuracy on the single model was 0.75211 with a mean absolute error of 0.296 on the test data, and the best validation accuracy on the combined model was 0.76874 with a mean absolute error of 0.3068, which means that the combined model performed slightly worse than the single model.

Another thing to note is that since we intentionally avoided non-English texts, this model cannot be used reliably with text from other languages.

Future

We would like to improve model accuracy on the name and blurb networks, because that would probably improve the accuracy of the combined network as well.

If the accuracy is decent, we would like to improve this model by attempting to predict the amount of funding a Kickstarter project will possibly receive. This would give much more information to a project owner than a simple binary answer and would allow the project owner to fine-tune his or her description, goal amount, funding duration, categories until it reaches an acceptable sum.

References

- Kaggle dataset - https://www.kaggle.com/kemical/kickstarter-projects#ks-projects-201801.csv
- Web crawler dataset - https://webrobots.io/kickstarter-datasets/
- Word embeddings - https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html
- Regression and neural networks - https://www.sciencedirect.com/science/article/pii/S1532046403000340
- Convolutional Best Practices - http://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Fall.2016/pdfs/Simard.pdf
- GloVe Global Vectors for Word Representation - http://www.aclweb.org/anthology/D14-1162
- Natural Language Processing Framework - http://www.thespermwhale.com/jaseweston/papers/unified_nlp.pdf
- Natural Language Processing - http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf
- Kickstarter prediction from 2013 - https://infoscience.epfl.ch/record/189675/files/etter2013cosn.pdf
- Kickstarter prediction no2 from 2013 – https://pdfs.semanticscholar.org/fcfc/059870dea7f70f3acc86735dc03601d302b6.pdf
- Kickstarter prediction from 2015 - https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=5486&context=etd
- Word tokenization - http://aircconline.com/acii/V3N1/3116acii04.pdf