Kickstarter Predictor: whether or not a project will be successfully funded?

Neha Kumari
jaiswal.neha.05@gmail.com
September 11, 2017

1 INTRODUCTION

Kickstarter is a community driven funding platform commonly known as crowdfunding. It's a pool of creative, innovative projects seeking for its way to make it life. These has become main sources of initial funding for some of the small businesses and startups who wants to launch their innovative baby to world. Consisting of millions of creators, they have the opportunity to directly connect with the public in the first go and get direct response.

Their mission being bring creative projects to life and they are true to that. Let's look some of the stats as listed on their site. Having more than 3Billion dollars pledged to projects. Approx 130,000 already successfully funded with 4 Million backers and 40 Million repeated backers. This comes to as a big motivator for getting fund for the projects.

However, the way Kickstarter works its on all or nothing basis. Having a realistic goal for a project becomes important. If a project doesn't meet the goal then the owner gets nothing. Example: if goal is of 1000 dollars, and backers funding till 999 dollars won't be a success. It has to be met exact and above. So it becomes important to understand what are the factors that will make the possibility of getting the funding much higher.

In this project, I tried to study what all kind of projects can get funding. Predicting using available data, whether or not project will be successfully funded.

2 Problem Statement

Whether or not a project will be successfully funded?

3 Solution Steps

In order to solve this problem below are the few main steps I followed:

- Data Exploration
- Feature Engineering
- Predictive Modeling Algorithm and Results
- Result Analysis
- Text Analysis
- Conclusion

4 Data Exploration

I picked up an already labeled dataset on kaggle Dataset params

1. project-id: unique id of project
2. name: name of the project
3. desc: description of project
4. goal: the goal (amount) required for the project
5. keywords: keywords which describe project
6. disable communication: whether the project authors has disabled communication option with people donating to the project
7. country: country of project author
8. currency: currency in which goal (amount) is required
9. deadline: till this date the goal must be achieved (in Unix time-format)
10. state changed at: at this time the project status changed. Status could be successful, failed, suspended, canceled etc. (in Unix time-format)
11. created-at: at this time the project was posted on the website(in Unix time-format)
12. launched-at: at this time the project went live on the website(in Unix time-format)

13. backers-count: number of people who backed the project

14. final-status: whether the project got successfully funded (target variable âĂŞ 1,0)

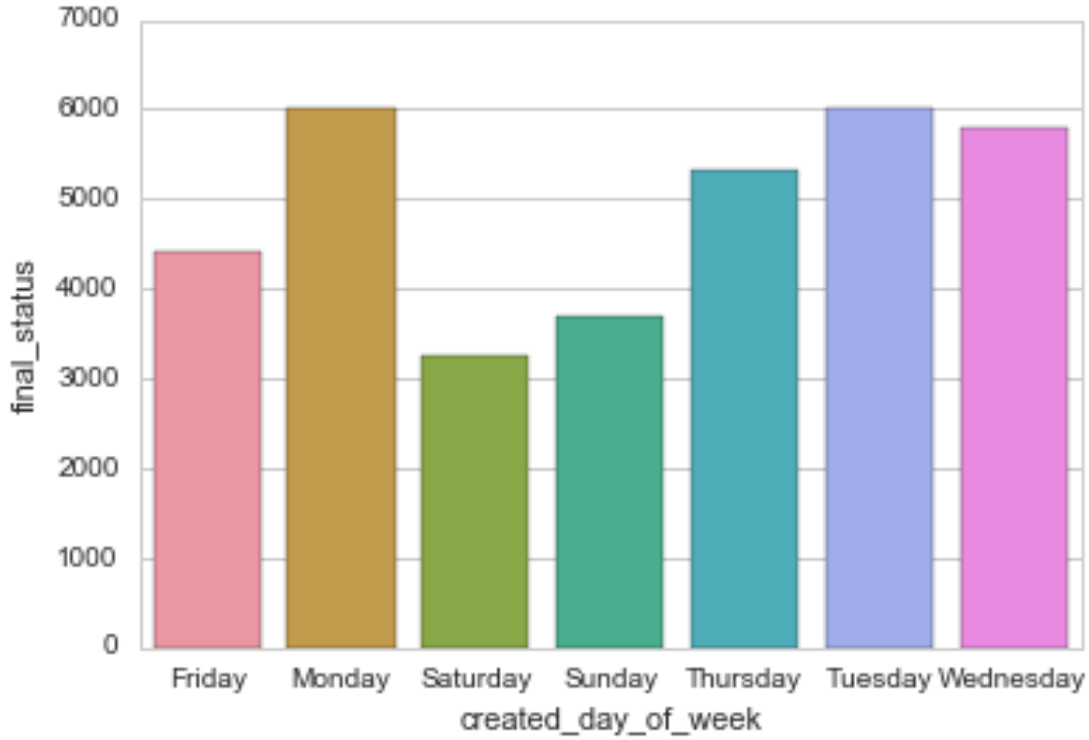
Dataset Source: https://www.kaggle.com/

4.1 Cleanup and Transformations (Data Exploration)

Data cleanup is a required step to generate proper results and transform it according to the need. First of all changing Unix time-format format in the data to a readable time-stamp format. Second transformation done is to fill non-available values with space.

4.2 Data Analysis: Weekdays vs Weekends (Data Exploration)

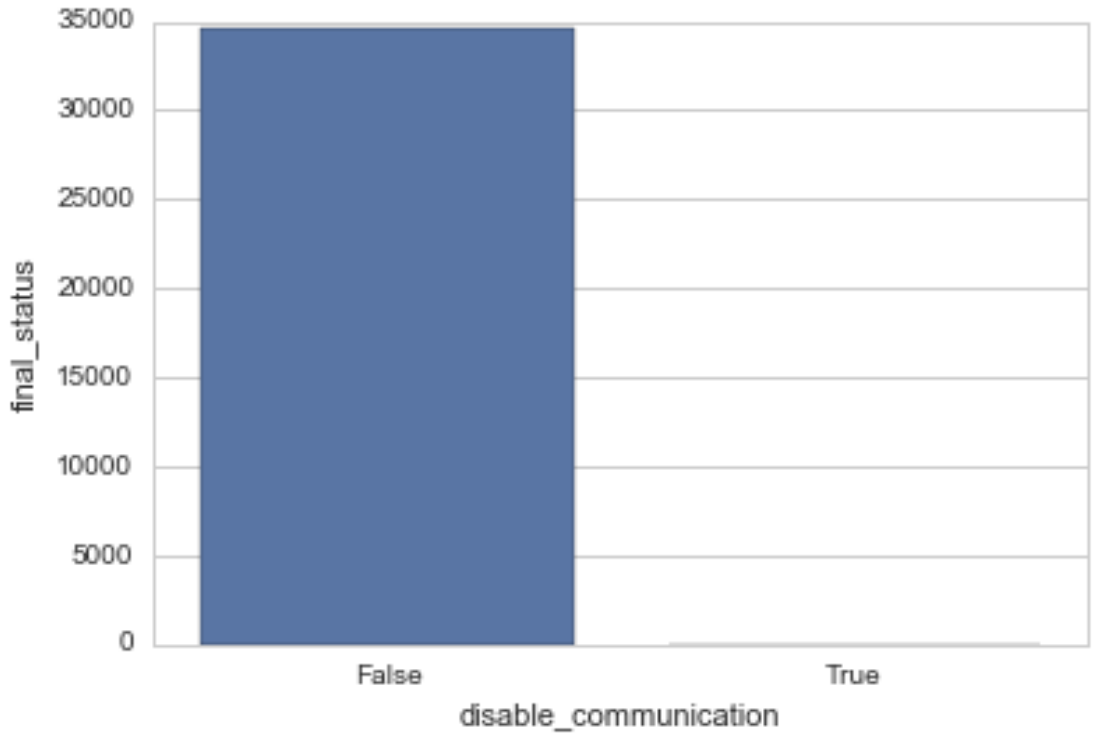
I wanted to analyze whether or not creating project during weekdays will be successful. Results are pretty interesting.



There is a clear indicator that most of the projects getting funded are launched during weekdays over weekend. Owners trying to post projects during weekdays could have getter probability getting their projects funded.

4.3 Data Analysis: Communication (Data Exploration)

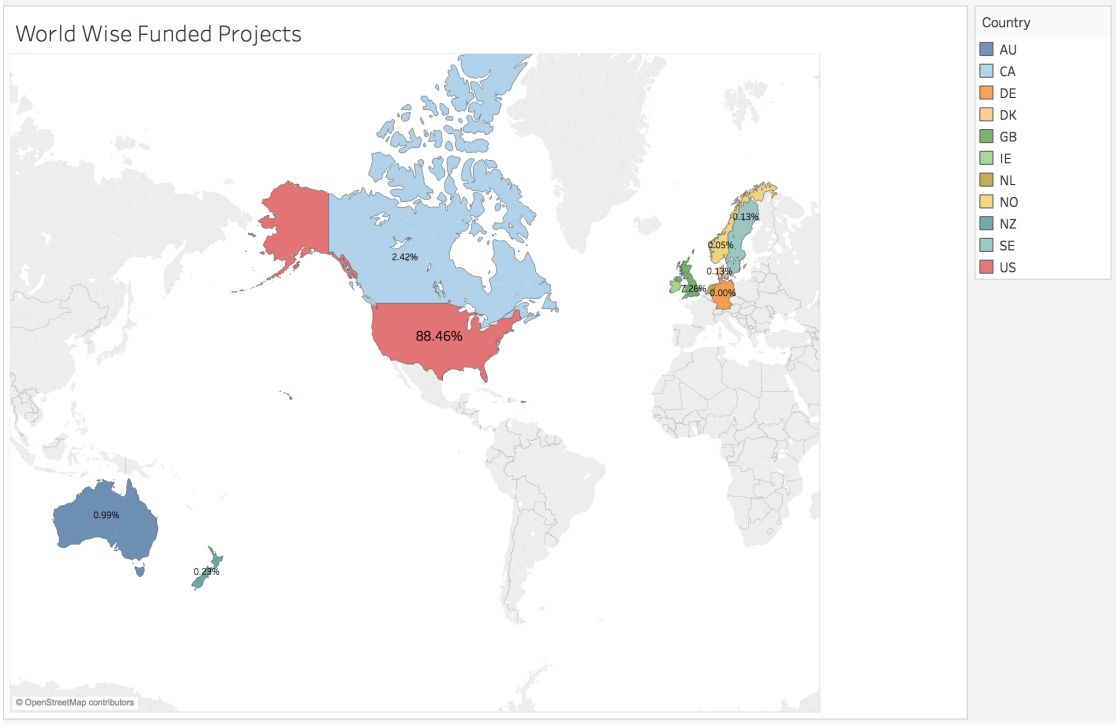
Whether or not communication enable or disabled matters.

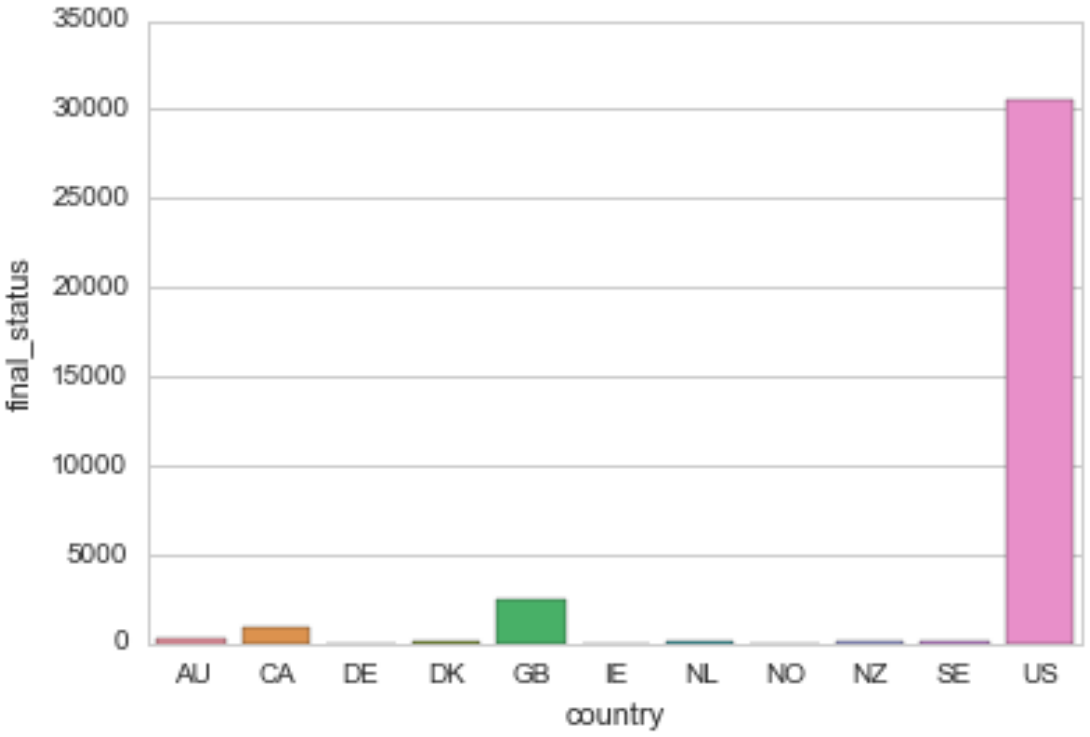


This indicates that when disable-communication is false, most of the projects gets the funding.

4.4 Data Analysis: Country Wise Funding (Data Exploration)

Analyzing country wise funded projects.

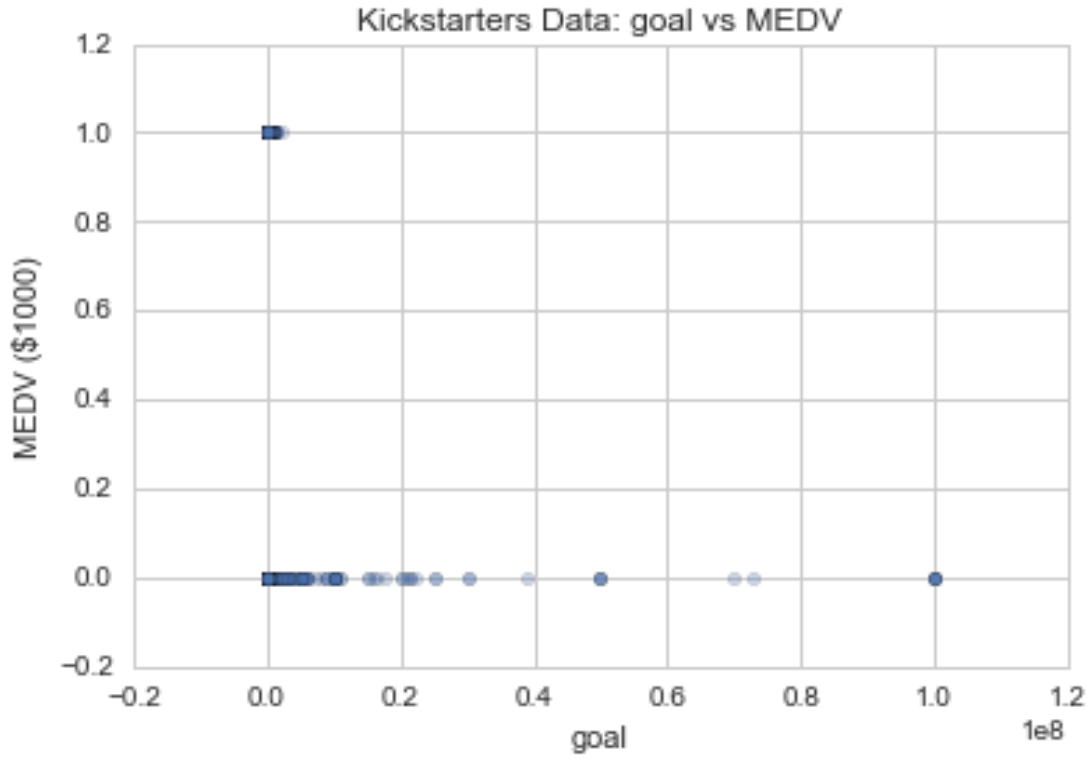


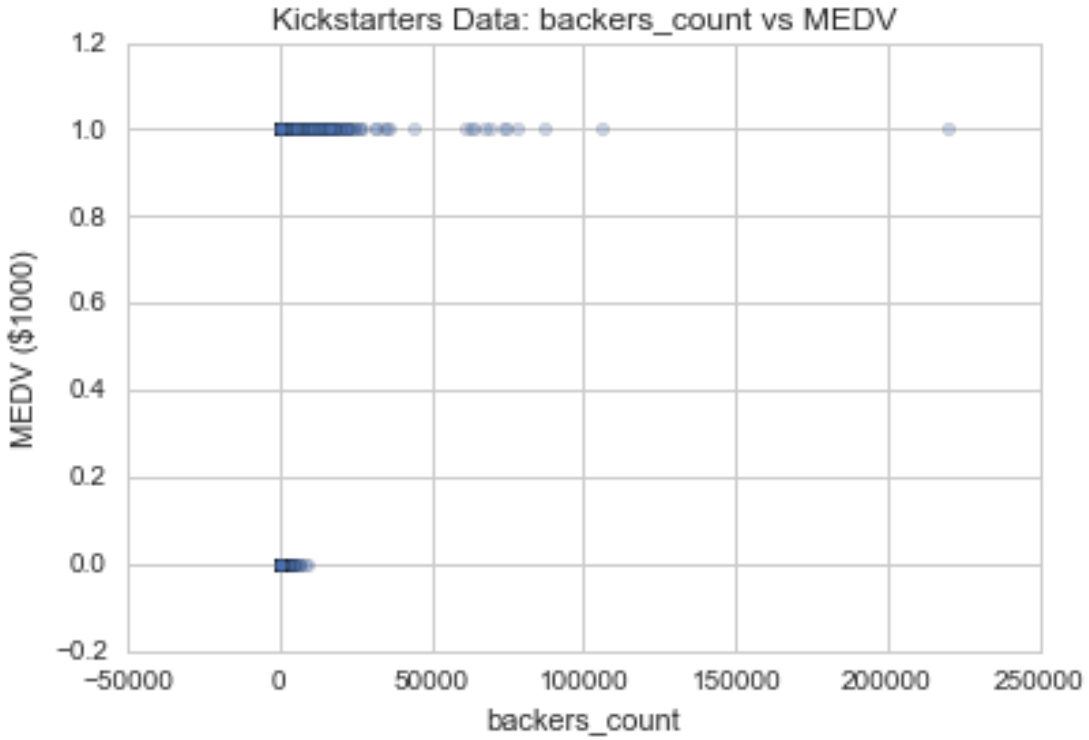


This graph shows US has highest no. of funded projects.

4.5 Data Analysis: Final status vs goal and backers-count (Data Exploration)

I tried to find the correlation between goal and final-status.


Kickstarters Data: goal vs MEDV
MEDV ($1000)
goal
1e8
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

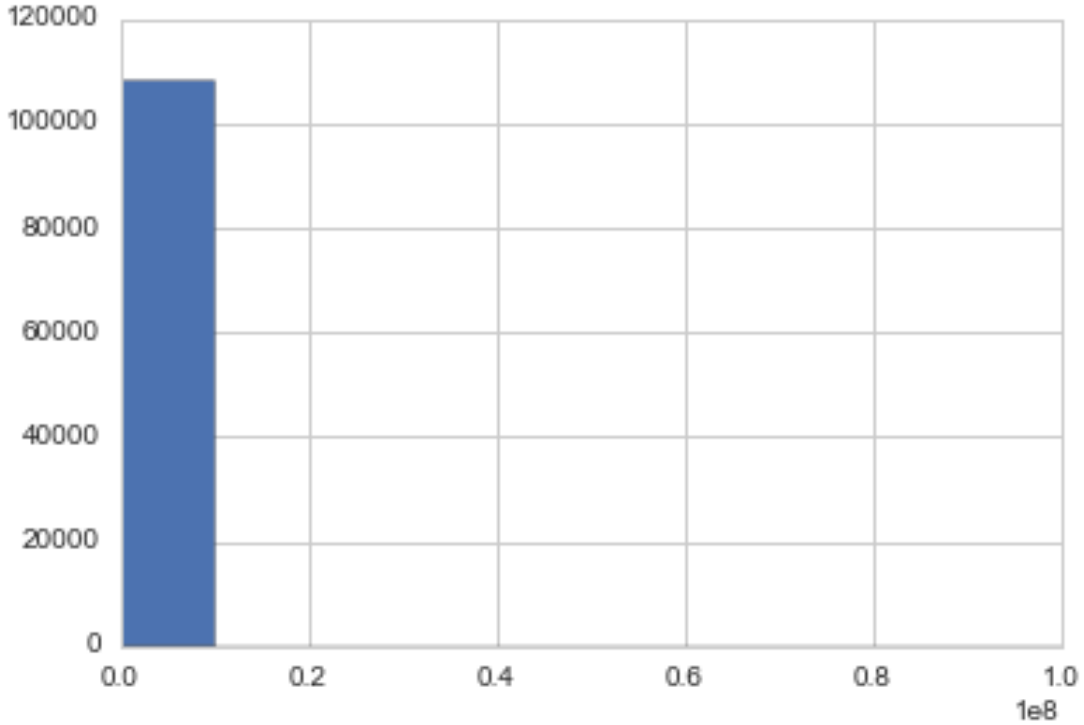

Kickstarters Data: backers_count vs MEDV
MEDV ($1000)
backers_count
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−50000
0
50000
100000
150000
200000
250000

This graph does not show much of correlation.

4.6 Data Analysis: Statistics around goal and backers-count (Data Exploration)

Statistics for Goals in Dollars:

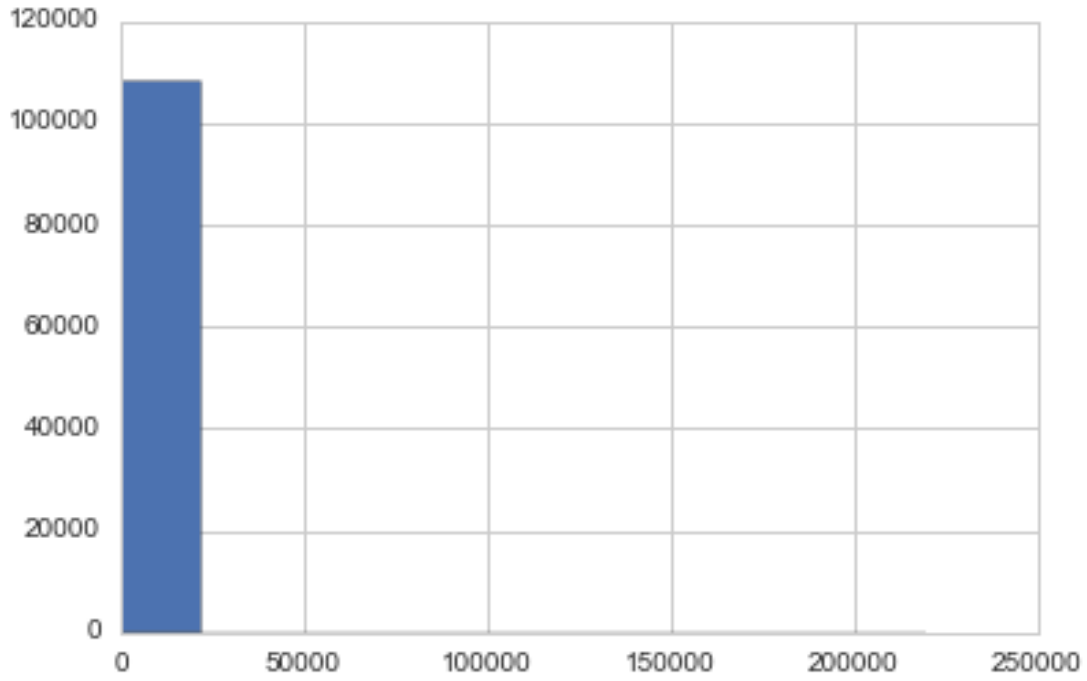
- Minimum goal amount: 0.01
- Maximum goal amount: 100,000,000.00
- Mean goal amount: 36,726.23
- Median goal amount 5,000.00
- Standard deviation of goal amount: 971,898.21
-



Statistics for backers-count:

- Minimum backers-count: 0.00
- Maximum backers-count: 219,382.00
- Mean backers-count: 123.52

- Median backers-count 17.00
- Standard deviation of backers-count: 1,176.74



5 Feature Engineering

Feature engineering is a process of extracting features to run the algorithms. In order to use data for a machine learning model, data needs to prepare in the usable format. This takes quite an engineering effort to prepare the right dataset for the problem they are solving. Often large volume of data from multiple sources needs to be combined. Big data platforms like Hadoop is used as the backbone of such pipeline.

It's very important to understand data by analyzing it first by using some of the statistical measures. Early data analysis accounts for measuring data distribution, detecting anomalies etc. In general, feature engineering is the process of using domain knowledge of the data to create features that machine learning algorithm consumes and uses to predict.

I performed below steps in order to create the feature set.

1. Finding no of days before the deadline from the date of launch.
2. Finding word length of desc field.
3. Creating list of important features

4. Convert categorical data into numerical

5. Normalizing the features

Selected normalized features for the predictive algorithm are: ***['goal', 'country', 'currency', 'backers-count', 'created-day-of-week', 'days-diff-launch-deadline', 'length-of-desc']***

6 PREDICTIVE ALGORITHMS

There are few well-adapted machine learning methods like supervised learning models where training set is already available (data is labeled with target answers which model can directly consume), non-supervised learning where labeled data is unavailable, reinforcement learning where each next action is learned from the previous action for the given state.

Supervised Machine Learning: is to build a model that makes predictions based on evidence in the presence of uncertainty.

- identify patterns in data,
- a computer "learns" from the observations.
- takes a known set of input data
- known responses to the data (output),
- trains a model to generate reasonable predictions for the response to new data.

Classification, the goal is to assign a class (or label) from a finite set of classes to an observation. (Responses are categorical) In this case, predictor algorithm is a binary classifier. Applying a supervised learning algorithm to predict whether or not a project would be funded. I tried 2 algorithms here:

1. Random Forest
2. Decision Tree

6.1 SUPERVISED LEARNING: RANDOM FOREST (PREDICTIVE ALGORITHMS)

: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. Implementation Steps:

1. Split the dataset into training and test in 3:1 ratio with random state as 42
2. Initialize the classifier
3. Fit the model using the training split

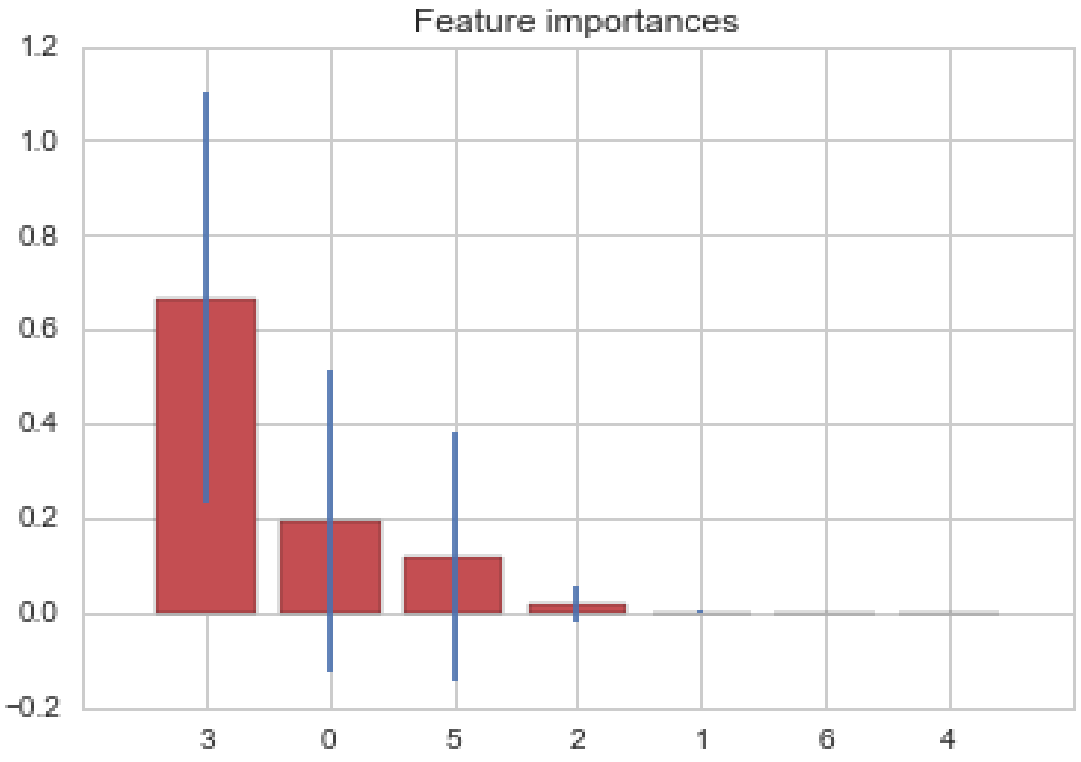
4. Predict using the test split

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(X_train,y_train)
y_pred=clf.predict(X_test)
```

Accuracy : 77

Features Importance :

1. 1. feature backers-count (0.666863)
2. 2. feature goal (0.192621)
3. 3. feature days-diff-launch-deadline (0.119009)
4. 4. feature currency (0.018811)
5. 5. feature country (0.002164)
6. 6. feature length-of-desc (0.000532)
7. 7. feature created-day-of-week (0.000000)



6.2 Supervised Learning: Decision Tree (Predictive Algorithms)

: Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Implementation Steps:

1. Split the dataset into training and test in 3:1 ratio with random state as 42
2. Initialize the classifier
3. Fit the model using the training split
4. Predict using the test split

```
from sklearn.tree import DecisionTreeClassifier
clfDT = DecisionTreeClassifier()
clfDT = clfDT.fit(X_train,y_train)
y_pred=clfDT.predict(X_test)
```

Accuracy : 82.5

Features Importance :

1. 1. feature backers-count (0.521746)
2. 2. feature goal (0.210421)
3. 3. feature days-diff-launch-deadline (0.077133)
4. 4. feature currency (0.006645)
5. 5. feature country (0.008789)
6. 6. feature length-of-desc (0.121878)
7. 7. feature created-day-of-week (0.053389)

Decision tree classifier's accuracy is better

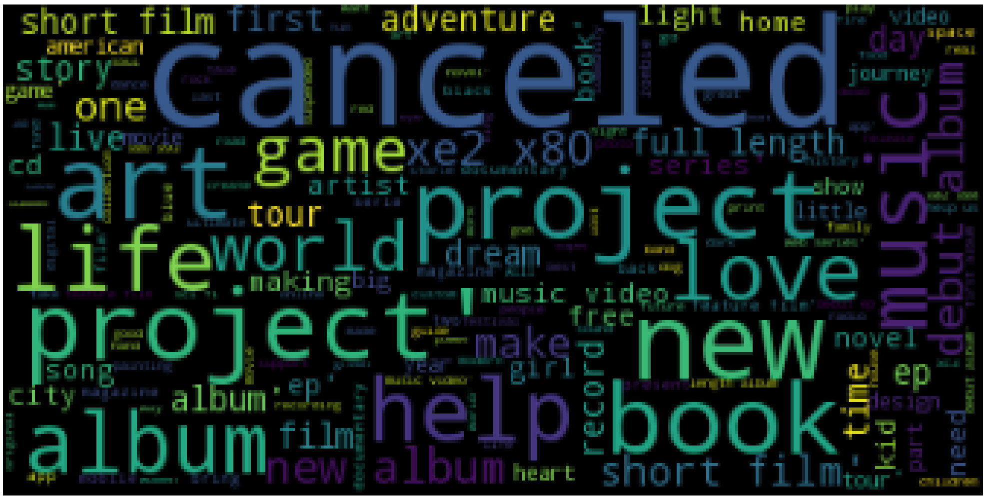
7 Text Analysis

The dataset contains 2 fields which contains text data. I tried to run clustering on titles.

7.1 Title Based Word Cloud(Text Analysis)

:

Top Words and Frequency based word cloud.



This dataset does not contain categories originally. However, looking at the this word cloud some categories can tagged example = album, film, video, music, artist, story, light, kid etc. Also using clustering analysis, we can group projects based on its similarity in the title or description.

7.2 Title Based Clustering (Text Analysis)

Grouping the titles into 10 clusters using KMeans algorithm.

```
tfidf_matrix = vectorizer.fit_transform(titles)
true_k = 10
model = KMeans(n_clusters=true_k, init='k-means++', max_iter=200, n_init=1)
model.fit(tfidf_matrix)

Top terms per cluster:
Cluster 0: new ep life love tour novel story cd magazine app record
movie documentary time man home season make cards city
Cluster 1: debut album ep cd record length recording release solo
help records band music lp video project studio novel making tour
Cluster 2: game music art series video web card new animated festival
board adventure life mobile tv comedy indie original season ios
Cluster 3: album project new length recording studio music release
cd art second solo band record tour love making ep records make
Cluster 4: help fund record make album new ep release finish needs
tour music support cd create bring length build need debut
```

Cluster 5: book children art series project poetry comic coloring photo canceled picture life new photography publishing coffee table publish story illustrated
Cluster 6: canceled project art cards playing new app music album magazine life game 3d com series mobile love card little man
Cluster 7: film short feature documentary horror project independent animated thesis festival student comedy sci fi length thriller production love new fan
Cluster 8: world music canceled premiere new end art war cup tour largest change game project **help** book real album save travel
Cluster 9: food truck canceled soul healthy mobile fast organic local cart farm fresh gourmet bbq trailer free street app market festival

8 Conclusion and Future work

I thoroughly enjoyed working on this project. Using predictive modeling techniques, I can predict the success of a project getting funded. However, as part of future extension to this project. I would like try following:

1. Find labeled categories for each projects in order to analyze how categories impact funding.
2. Try more algorithms, tuning the parameters to improve the accuracy > 82.5
3. Incorporate text based features for the model.

9 References:

1. http://scikit-learn.org/stable/modules/tree.html
2. http://seaborn.pydata.org/tutorial/categorical.html
3. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
4. https://www.kaggle.com/
5. https://www.kickstarter.com/help/stats
6. https://www.kickstarter.com/