PREDICTING SUCCESS OF KICKSTARTER CAMPAIGNS

Capstone Project Final Report

Pundir, Saurabh
saurabhpundir.data@gmail.com

Date: 05/29/2018

Table of Contents

1. Introduction 4

1.1. About: 4

1.1..1. Formal Definition - Crowdfunding: 4

1.2. Problem: 4

1.3. Clients and Audiences 4

1.4. What does success mean 5

1.5. Scope: 5

1.6. Question Client(s) really care about 5

2. Data Wrangling 5

2.1. Getting File 5

2.2. File Content: 6

2.3. Choosing Processing Interval: 6

2.4. 3 Step Approach 6

2.5. Extracting Data: 6

2.6. Cleaning Data: 6

2.7. Storing Data: 7

2.8. Merging Datasets: 7

2.9. Data Dictionary: 8

3. Exploratory Data Analysis 8

3.1. Data wrangling: 8

3.2. Exploring data relationship 9

3.2..1. What is project count as per project status? What is successful and failed ratio? 9

3.2..2. Is there relationship with the year, the month of a launch date for successful or failed? 11

3.2..3. Is there relationship with goal amount to the success or failure of the project? 13

3.2..4. Is there relationship with pledge amount in the success or failure of the project? 15

3.2..5. Is there relationship with a pledge to goal ration in the success or failure of the project? 18

3.2..6. Is there relationship with no of backers in the success or failure of the project? 20

3.2..7. Does staff pick mark more successful projects? 22

3.2..8. Is there relationship with a pledge to no of backers ration in the success or failure of the project? 22

3.2..9. Which country is having most and least successful or failed project? 24

3.2..10. What categories have most and least successful or failed project? 25

3.2..11. What location type is having most and least successful or failed project? 27
4. **Inferential Statistics** 28
4.1. Exploration of a number of backers 28
4.1..1. Successful project exploration 28
4.1..2. Fail project exploration 28
4.1..3. Hypothesis Testing 28
5. **Baseline Analysis** 29
5.1. Converting columns 29
5.1..1. To dummies 29
5.1..2. To number 29
5.2. Column selection 29
5.3. Logistic Regression 29
5.3..1. Setting hyperparameter 29
5.3..2. Splitting train and test size 29
5.3..3. Fitting data 30
5.3..4. Finding accuracy 30
5.3..5. Creating Classification report 30
5.3..5.1. Regularization parameter L2 30
5.3..5.2. Regularization parameter L1 30
5.3..6. Creating confusion matrix 31
5.3..6.1. Regularization parameter L2 31
5.3..6.2. Regularization parameter L1 31
5.3..7. Finding feature importance with regularization L1 31
6. **Further Analysis** 34
6.1. Random Forest Classifier 34
6.1..1. Estimator 15 -175 35
6.1..2. Estimator 100 -200 37
6.1..3. Estimator 150 -500 39
6.1..4. Estimator 300 -1000 41
6.2. Model Comparison &Conclusion 42
6.3. Recommendation for Clients 43
7. **Future Work** 43
1.1. Scope: 44

1.2. Launch Period: 44
1.3. Technology:... 44

1. Introduction

1.1. About:

It is no doubt that in current time humans are connected to each other more than ever before. This has opened the new opportunity for anyone with an idea to present it to the world.

The term "Startup" was coined in 1976 but become the hottest buzzword in last few years.

In the era of information and social media, it is sometimes easy and more worth to reach millions of people ready to give a dollar then one VC to invest million dollars. This is the basic concept of **crowdfunding** which again is not a new term but got muscles in its meaning in times of Internet.

Platforms like Kickstarter provide an opportunity to person with an idea to present it to everyone who has access to the internet. Also provides an opportunity for a person with belief in someone's idea to contribute, obviously with the expectation of expected rewards on investments.

1.1..1. Formal Definition - Crowdfunding:

Crowdfunding is the practice of funding a project or venture by raising monetary contributions from a large number of people in smaller amounts. Crowdfunding is a form of crowdsourcing and of alternative finance.

The company or a person comes up with a project proposal or idea and shares its details and defines the amount they are looking to raise.

1.2. Problem:

Crowdfunding has given the opportunity to various well known products like pebble watch, Oculus or innovative project like Baubax. As of April-2018 Kickstarter (crowdfunding platform) has raised **three billion** (3,647,430,067) US dollars from **fourteen million** people (14,547,043) for **four hundred thousand** projects using crowdfunding.

But are all projects worth investing? Do all fourteen million people have the expected rewards on investment? Were all three billion dollars given to projects worth?

In the investment world, it is obvious that there is no simple answer to the above questions. But there is an opportunity to provide all clients and audience involved an assistance to take better decision driven by the power of data.

The problem in this project to be addressed is that ***"Is it possible to estimate the probability of success for the recently launched or still running project?"***

1.3. Clients and Audiences

In problem section, we mentioned terms clients and audience. It is important to understand the definition of clients and audience to explore the above problem from their perspective.

There are 3 types of entities involved in any crowdfunding project.

1. **Project Owner**: Individual or company who creates a project to acquire funding from crowdsourcing
2. **Project backer**: Individual showing interest in funding the project.
3. **Crowdfunding facilitator (company hosting website)**: Company owning the platform.

All the entities mentioned above are clients and audiences because all of them are interested in one common thing **"Success"** of the project.

1.4. What does success mean

Before starting to address this question it is important to define ***<u>what is a success?</u>***

In crowdfunding ecosystem, success simply means that the project got the asked amount or more amount in the desired time frame. This amount is called **goal amount.**

It has a significant impact on clients mentioned above. If the project is a success.

1. **Project owner:** They will get money for the project.
2. **Project backer:** They will get an expected reward on their investment.
3. **Crowdfunding facilitator (company hosting website):** Company will get commission or fees. Also, based on prediction company can provide support by highlighting project.

The data has a label called **Status**, which has value Failed, Successful, Canceled and Live. The status successful is considered a success for this project.

1.5. Scope:

There can be various crowdfunding sources, however, this Project will be limited to projects funded using a website called **Kickstarte**r.

This project is about analyzing data available for crowdfunding platform and find out the best algorithm to predict the probability of success of newly launched crowdfunding project. The project should be launched in last twenty-four hours because the project which has passed more time might have received some pledge amount which is not covered in this model.

1.6. Question Client(s) really care about

All the above clients and audience have a concerning question...

"What is the probability of success of a recently launched Kickstarter project?"

2. Data Wrangling

The data wrangling section involved getting data and extracting useful columns or variable from the files for further processing. The individual section explains the approach applied for the cleaning and wrangling of data to extract data for all further analysis.

2.1. Getting File

Data is available as CSV file on webrobots.com. The data is available based on a monthly basis as a zip file. Each month file size is 140 MB.

2.2. File Content:

Each month's zip file contains nearly 40 CSV files of approximately 19 MB size each. Each CSV file contains information of around 4000 projects. There are 32 distinct fields for each project. The field details are mentioned in the data dictionary section.

2.3. Choosing Processing Interval:

The approach is to start with recent months file (Jan-2018) and keep adding previous months data. Reading and consolidation of multiple months of data showed that there are not many new projects added progressively in each month's files. The recent files contain most projects from previous months.

2.4. 3 Step Approach

Data wrangling was performed and included three stage process.

1. Extracting data for each month.
2. Data Cleaning and consolidation into a single file.
3. Storing data for further processing.

2.5. Extracting Data:

There are multiple files in each month's folder. For load sharing purposes, each month's folder was processed separately. Reading files inside the folder is performed using *"os.path"* module. Each data extracted from individual file gets appended to a data frame for a month.

For an initial understanding of data the information on rows, columns, and data type of the data frame are explored. Each month has 170K distinct projects.

The code related to this part is in Capstone_DataWrangling_I_ReadFiles.ipynb file.

2.6. Cleaning Data:

After reading monthly data file into a data frame. The fields mentioned above were extracted further by looping each data frame row. The JSON data is further extracted using JSON normalize. The date fields are converted using date time conversion from Unix time to UTC date ("%Y-%m-%d %H:%M:%S") format.

The new temporary data frame is built in process for above extracted and converted data. This temporary data frame is merged using join and project id as a common key into main data frame after all subfields value is extracted.

The list of subfields extracted from JSON columns

- **creator**
  - name
  - is_registered
  - user id
- **location**
  - city
  - state
  - type (district, town, city)

- **Category**
  - Project category type

Missing Data 1(Location):

Location information is not available for all rows in the data frame. So, this was handled by comparing location with the null & empty value before extracting the data. Every data frame around 2K records will not have any location value.

Missing Data 1(user info):

The field name is registered under user info and is not available from/before May 2017. This field is not relevant and hence filled with empty if not available.

Code related to this part is in Capstone_DataWrangling_I_ReadFiles.ipynb file.

2.7. Storing Data:

Challenge:

The challenge in cleaning and extracting data for each month in the data frame is the processing time. Processing each month's files take 8-10 hours on medium configuration machine (laptop) available. Processing 6 months data took almost three days to complete.

After cleaning data and creating final data frame for the monthly file. The file is stored on the hard disk to avoid re-running the time-consuming process. The pickle object sterilization is utilized to store each month data frame. After all monthly files are available, as individual files, a single pickle file is created with all data appended.

This file is unpickled and processed in further steps. The process of unpickling and making the file available is under 5 minutes.

This approach provided the benefit of processing the data once and then utilizing it further at a faster pace.

Further Work:

Hadoop and map -reduce technology can be utilized to make this process better. But currently, this was not the scope of our project.

2.8. Merging Datasets:

The pickle files created for each month mentioned in above process are utilized to read and create a final data frame. The two data frames are created from this data.

All unique projects:

Considering the latest months data frame (July) as the base data frame, the records from all previous months are appended to create a data frame considering **unique** projects. At the end of the process, we have unique **172892** projects available. This data frame will be utilized further stages of the project.

The code related to this part is in Capstone_DatWrangling_II_Consolidate_Files file.

2.9. Data Dictionary:

a) Kickstarter.com:
Data: https://webrobots.io/kickstarter-datasets/
The dataset is available for March 2016 for every month. This data is collected from web crawler created by the company. The latest dataset contains following fields.

1. Id: the unique id
2. Photo: Info for all photo associated with the project.
3. Name: name of the project
4. Blurb: intro & detail
5. Goal: Amount needed to be raised
6. Pledged: Actual amount raised
7. State: Current state of the project (canceled, failed, successful)
8. Slug: a brief description
9. Disable_communication: communication allowed to the creator
10. country: country of campaign origin
11. currency: the currency of campaign origin
12. currency symbol: currency symbol of campaign origin
13. currency_trailing_code: TRUE, if conversion needs to happen in a user currency
14. deadline: UNIX timestamp for a project deadline
15. state_changed_at: Unix timestamp
16. created_at: UNIX timestamp for the project created
17. launched at: UNIX timestamp for the project started
18. staff pick: TRUE, if staff picked
19. backers count: Total user backed the project
20. static_usd_rate: USD conversion rate from original currency
21. Usd_pledged: pledge amount in USD after conversion:
22. creator: details like username for the creator
23. location: location of the project
24. category: the category of the project
25. profile:
26. spotlight: feature spotlight available or not
27. URLs: Url info for a project
28. source URL: seems like URL for the category

3. Exploratory Data Analysis

The section involved exploring the various relationship between columns or independent variable in the context of the state of the project.

3.1. Data wrangling:

The data needs to be further modified to get some variables in format to provide it to data visualization library like Matplotlib and Seaborn.

The information extracted in this step

- Converted Staff Pick to category type
- Extracting the no of days between project creation and project ended
- The extracted ratio between pledged and goal amounts
- The extracted ratio between the pledged amount and no of people backing project.
- Camel casing the state column so it can be used in the axis label

The code related to this part is in Capstone_DataWrangling_III_BfrDataStory file.

3.2. Exploring data relationship

The various columns are visualized to get the data representation of their relationship with the successful and fail state.

The code related to this part is in Capstone_DataStory file.

3.2..1. What is project count as per project status? What is successful and failed ratio?

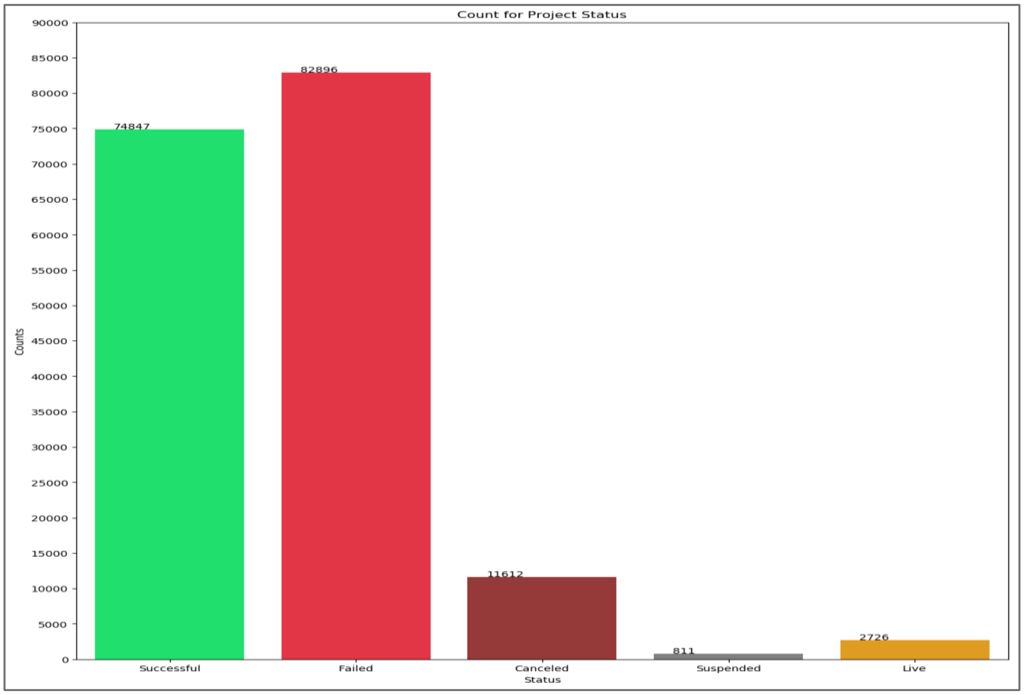


Figure 1: Project count as per project status

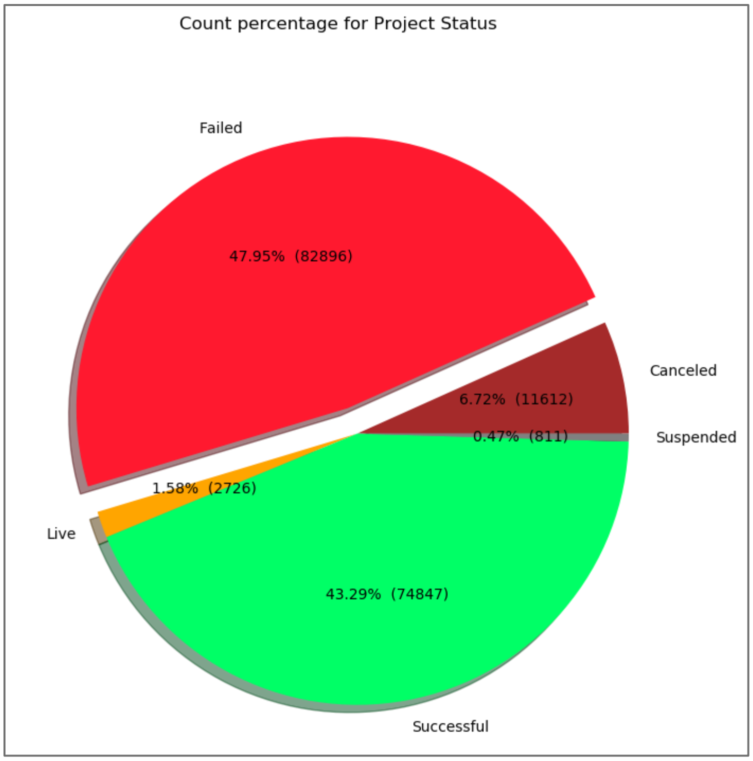


Figure 2: Project status ratio

The above displays that count of the project with each status available. Most of projects are either Successful or Failed. The other status are very small and not important for this project. So, they won't be mentioned much going forward.

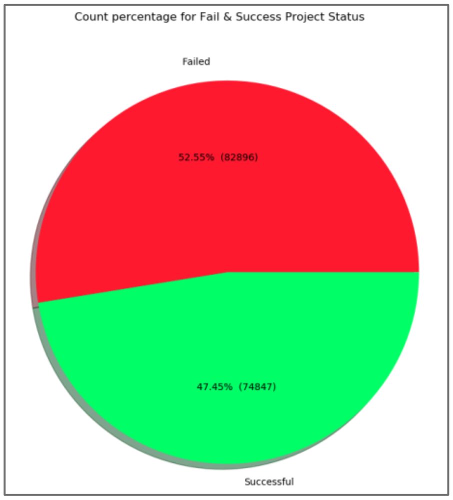


Figure 3: Ratio of Failed and Successful project

In the above diagram, it is very obvious that there are more failed projects than successful projects but still, the ratio is very close, and we have data for both the states in almost equal proportion.

3.2..2. Is there relationship with the year, the month of a launch date for successful or failed?

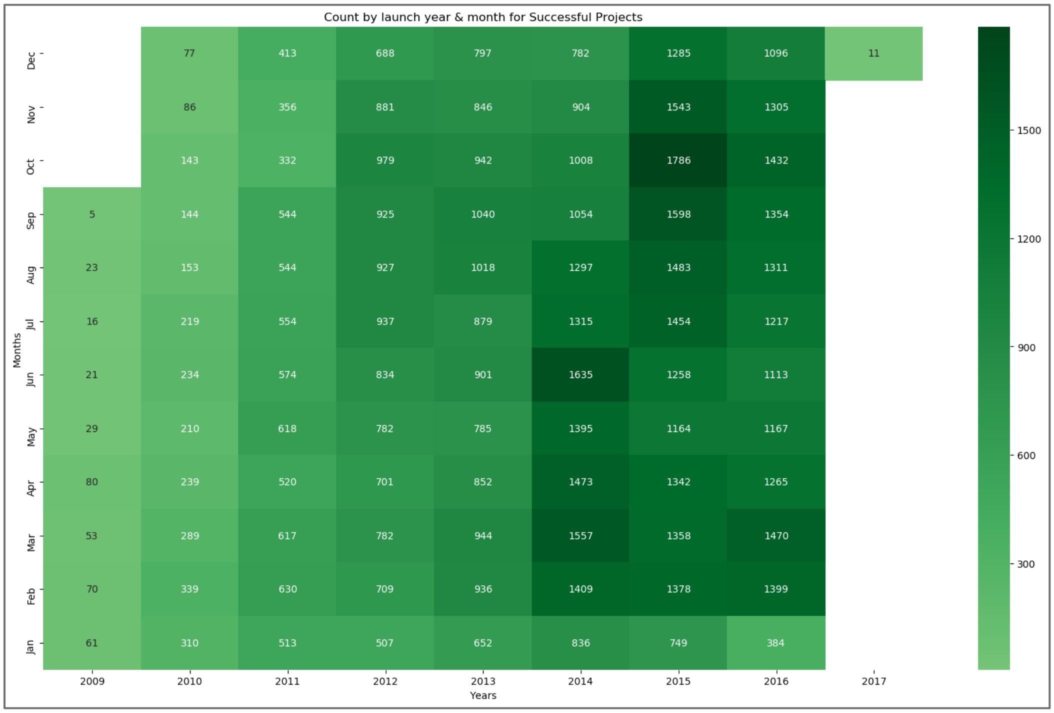


Figure 4: Heat map for year and month for successful

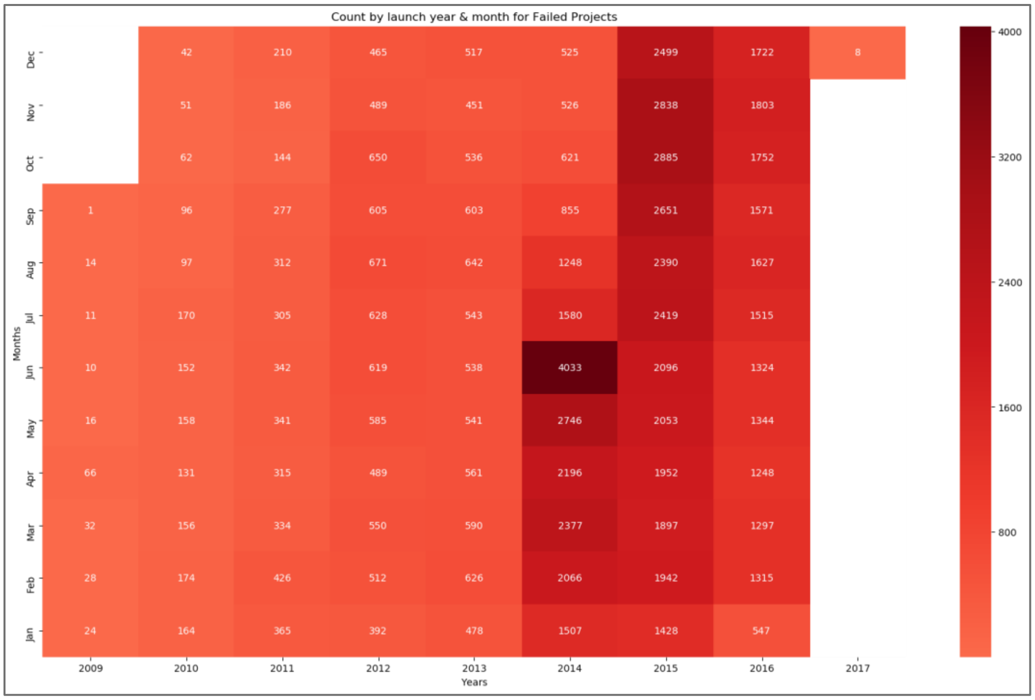


Figure 5: Heat map for year and month for failed

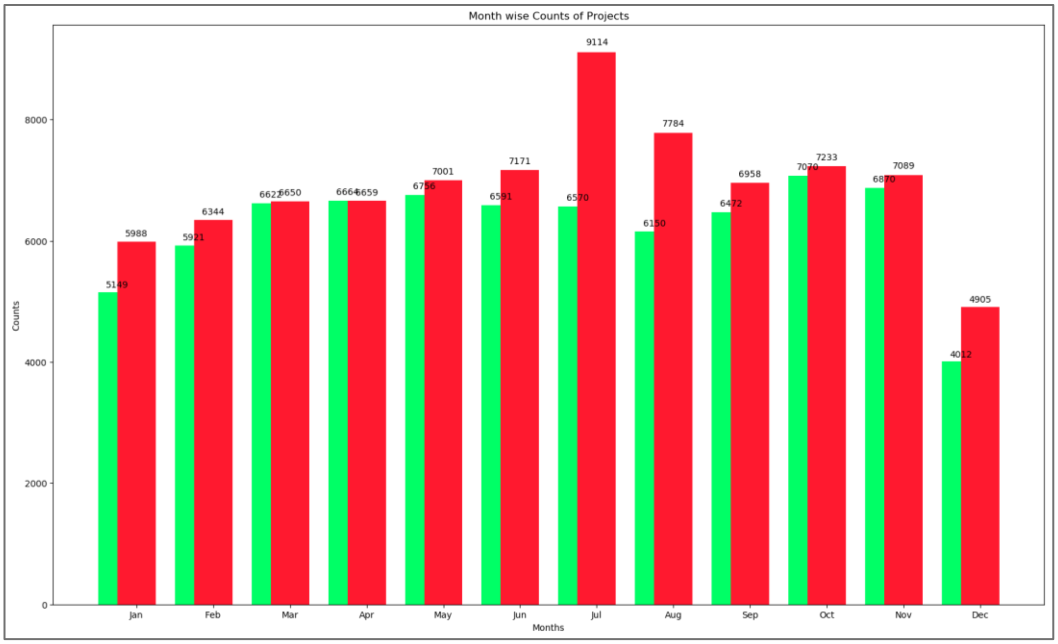


Figure 6: Month-wise successful and fail project

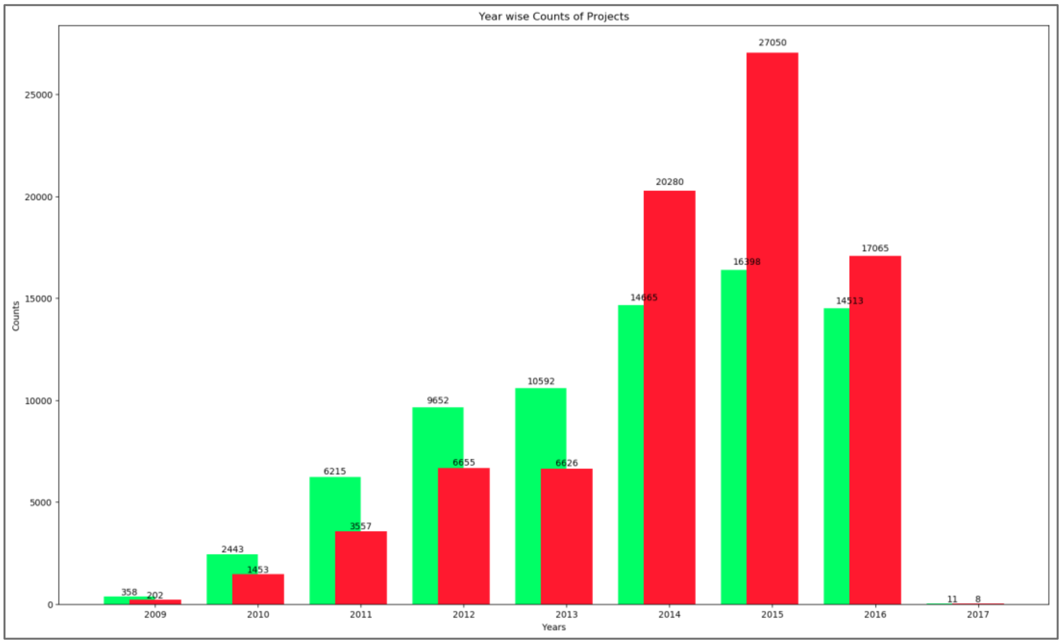


Figure 7: Year-wise successful and failed project

The above diagram shows a number of successful and failed projects in a relationship with year and month. The influence of year and month on the state is explored and following observation can be concluded.

- Projects are launched evenly all months around with few months showing spike in numbers
- Every month more projects failed but still, the counts are close
- Kickstarter was launched in 2009. There was steady growth in the number of projects till 2015.

- Insufficient data available from 2017 and onward
- There is an indication that Kickstarter is getting less popular and interest is on the decline
- There is an indication that more project failed as year's progress. This may be due to decline in popularity or insufficient data. It should not be concluded.

3.2..3. Is there relationship between goal amount and the success or failure of the project?

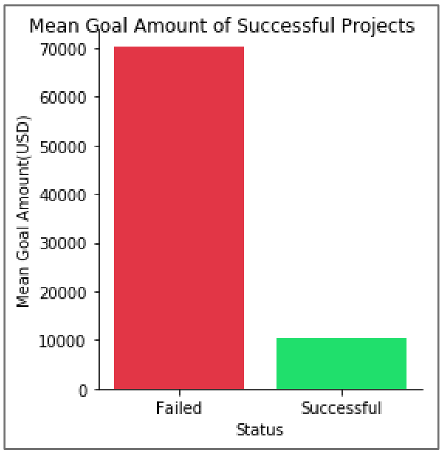


Figure 8: Mean goal amount of Successful and failed projects

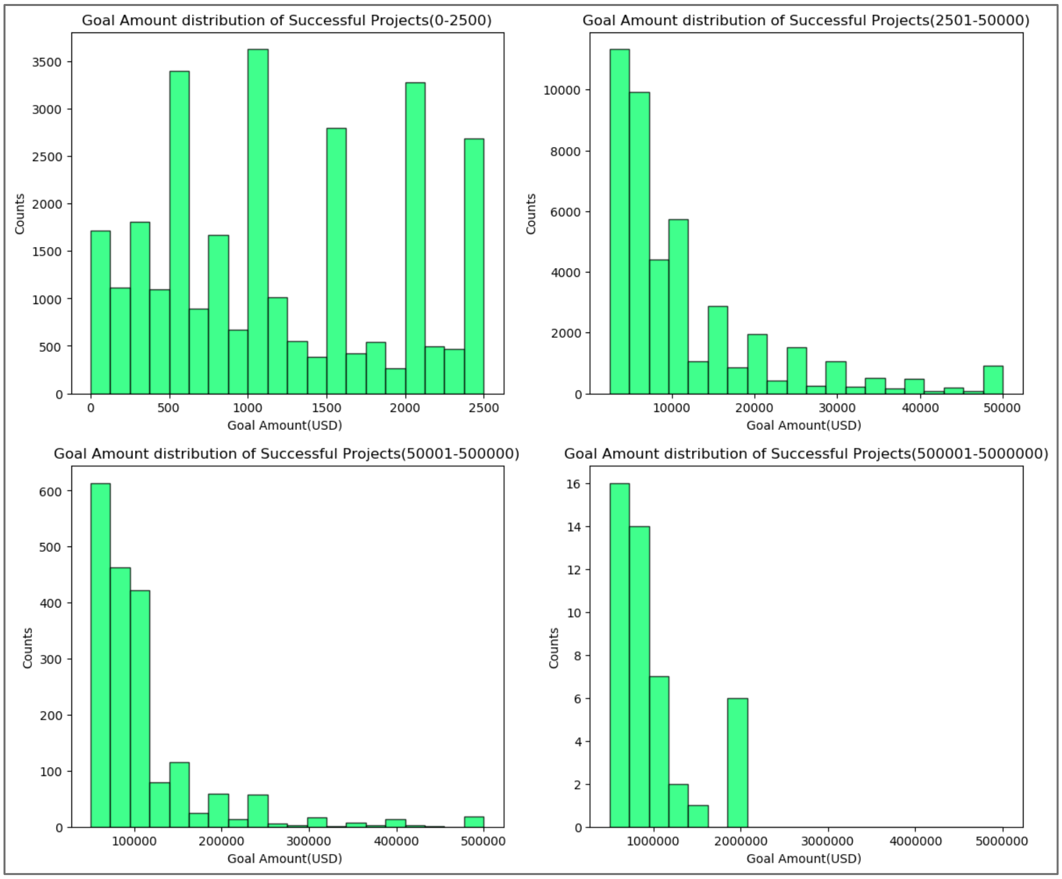


Figure 9: Goal amount distribution of Successful projects

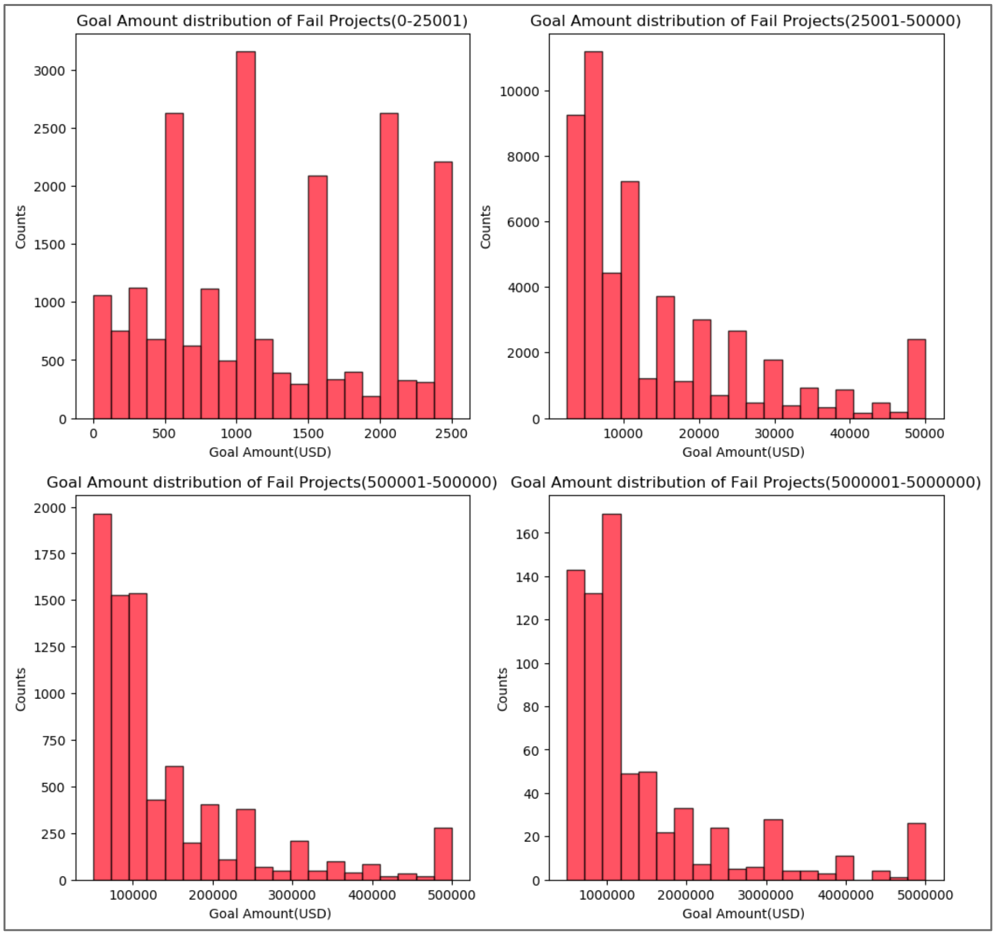


Figure 10: Goal amount distribution of failed projects

The above diagram shows relationship between goal amount distribution of successful and failed projects. It is an important variable in our analysis.

- The average goal amount of failed project is seven times higher. It indicates project with higher ask amount fail more.
- Most successful projects have goal amount in the range of 2500-5000.
- Most failed projects have goal amount in the range of 5000-7000
- The interesting observation is that goal amount of 0 - 4000 USD the difference between the number of failed and successful project is very similar. It is in higher amounts the difference display impact.

3.2..4. Is there relationship with pledge amount in the success or failure of the project?

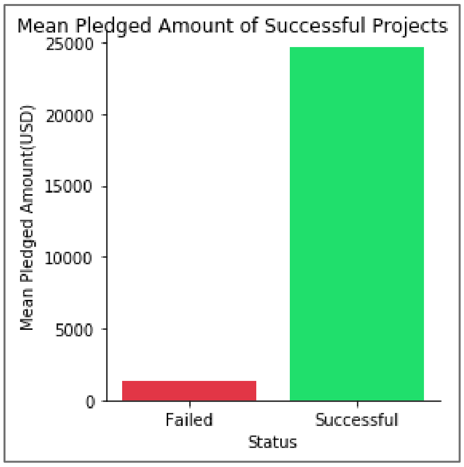


Figure 11: Mean pledged amount of Successful and failed

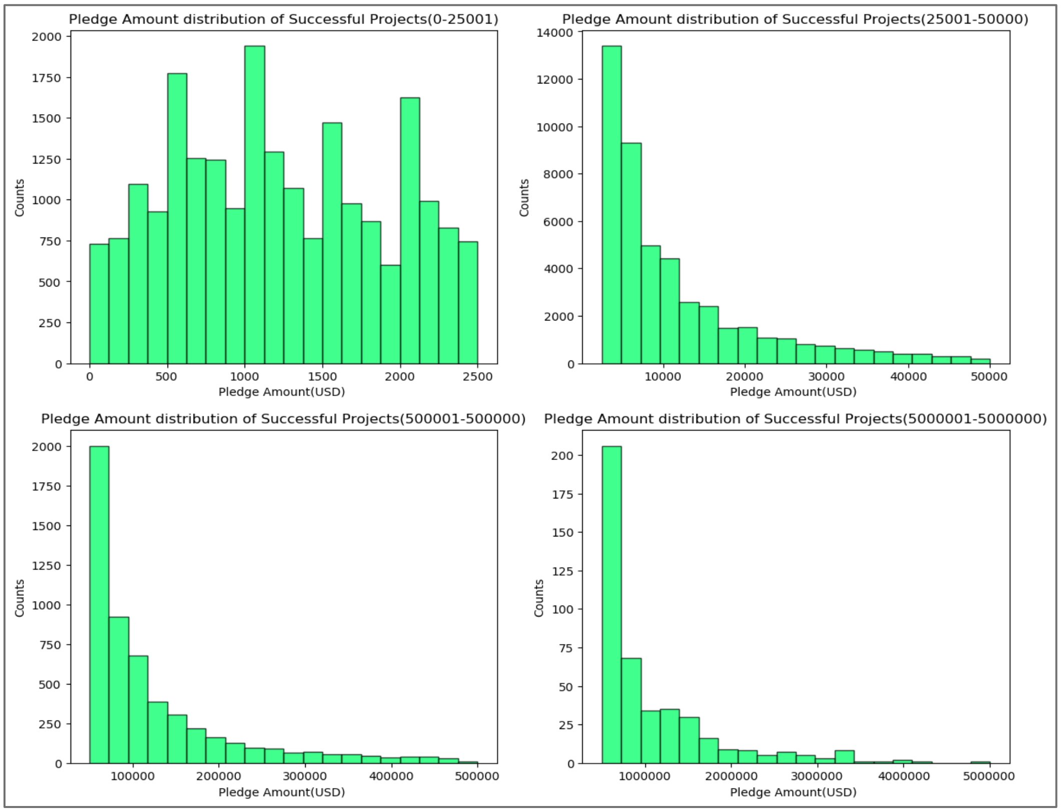
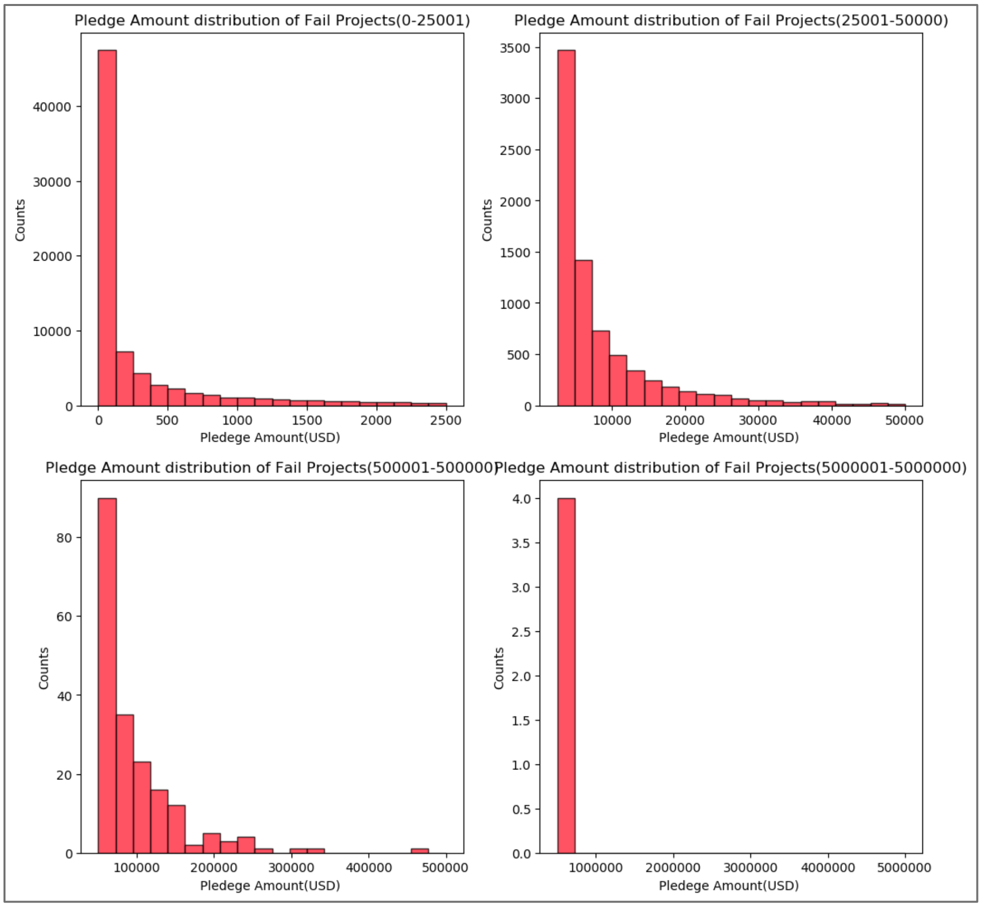


Figure 12: Pledged amount distribution of successful projects



Pledge Amount distribution of Fail Projects(0-25001)

Pledge Amount distribution of Fail Projects(25001-50000)

Pledge Amount distribution of Fail Projects(500001-500000)

Pledge Amount distribution of Fail Projects(5000001-5000000)

Figure 13: Pledged amount distribution of failed projects

The above diagram analyses if there is a relationship between pledge amount and success/failure of the project. The pledged amount is variable not included for modeling, but it is necessary to understand the pattern based on pledged amount.

- The average goal amount of successful project is twenty-five times higher. It indicates project with higher amount asked fails more. No surprise but still interesting.
- The pledge amount and number of failed projects also have a strong negative exponential relationship.

3.2..5. Is there relationship with a pledge to goal ratio in the success or failure of the project?

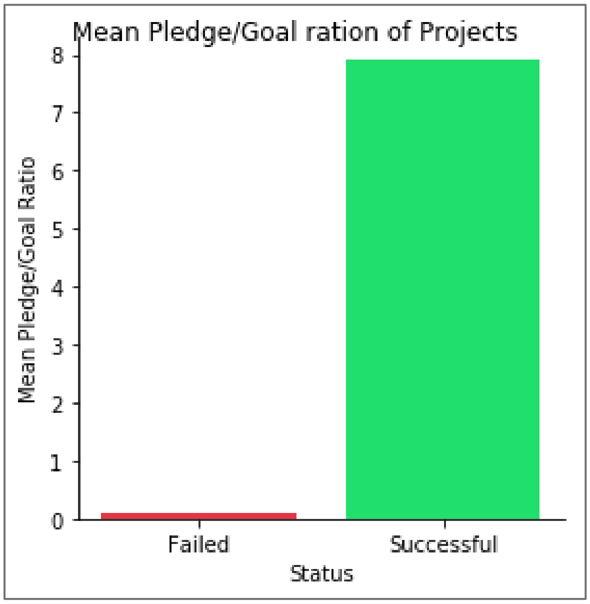


Figure 14: Mean of Pledge/Goal amount ratio

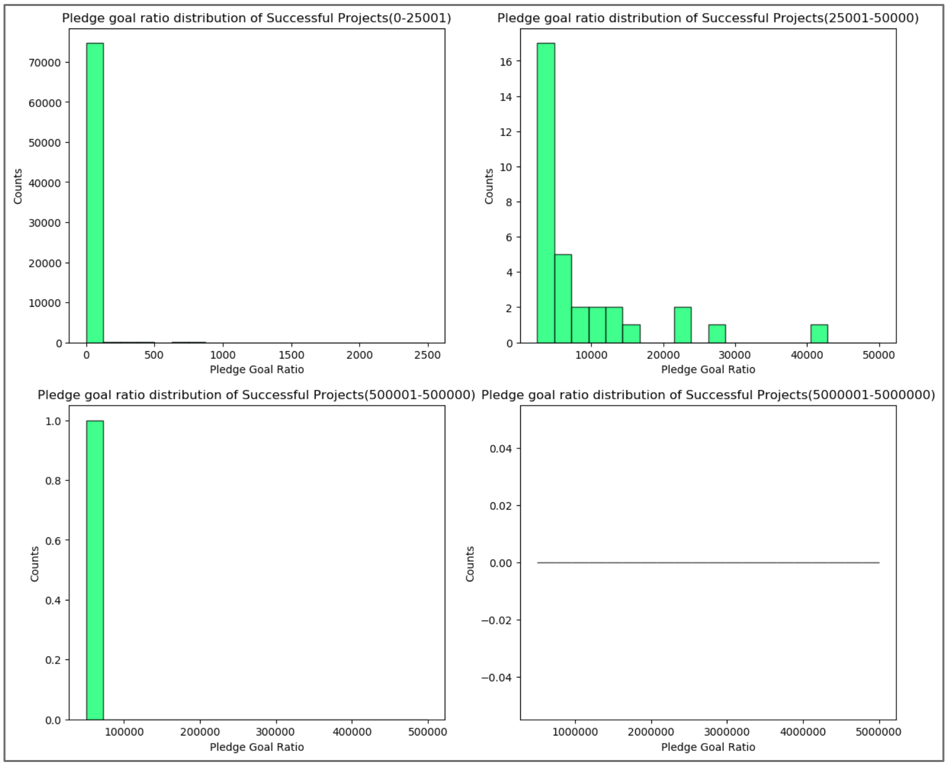


Figure 15: Pledge amount to goal amount ratio for successful

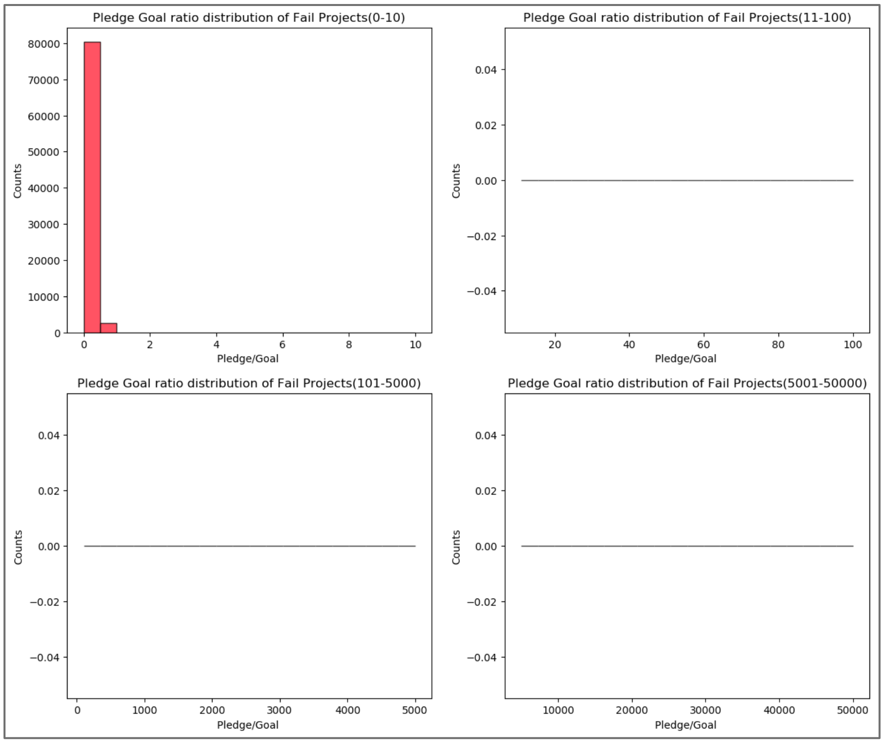


Figure 16: Pledge amount to goal amount ratio for failed

The above diagram shows the ratio of successful and failed projects in a relationship with the pledged amount by goal amount. This gives an indication of ratio the projects get than expected.

- The most successful project just achieves their goal amounts. Very rare projects get funding higher than goal amount.
- Almost all failed project miss their goal amount by more than 50%. Very rare they fail close to their targeted goal amount.

3.2..6. Is there relationship with number of backers in the success or failure of the project?

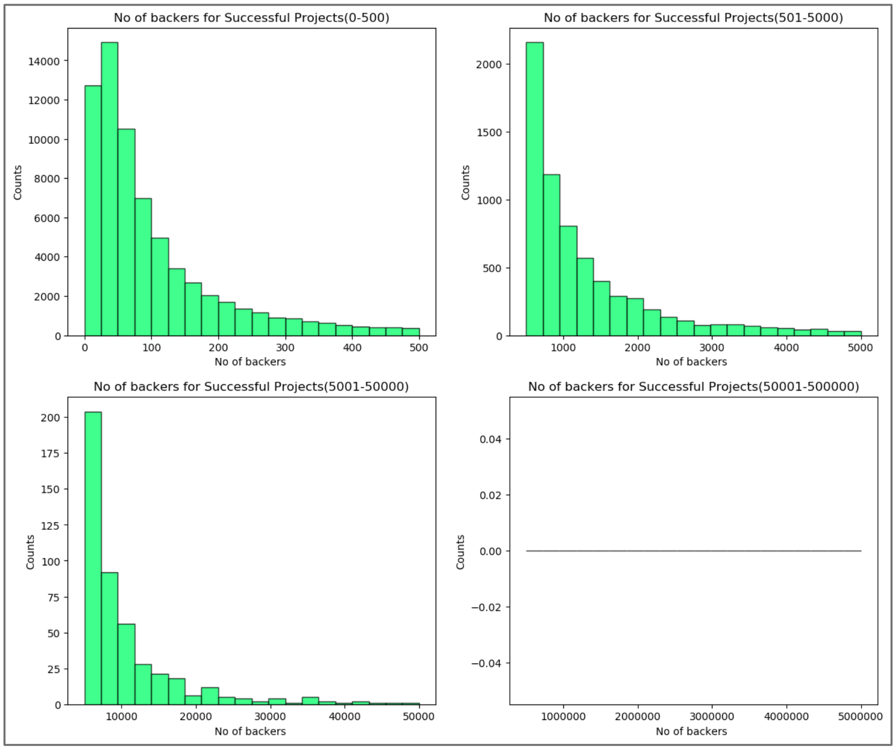


Figure 17: Number of backers impact success of projects

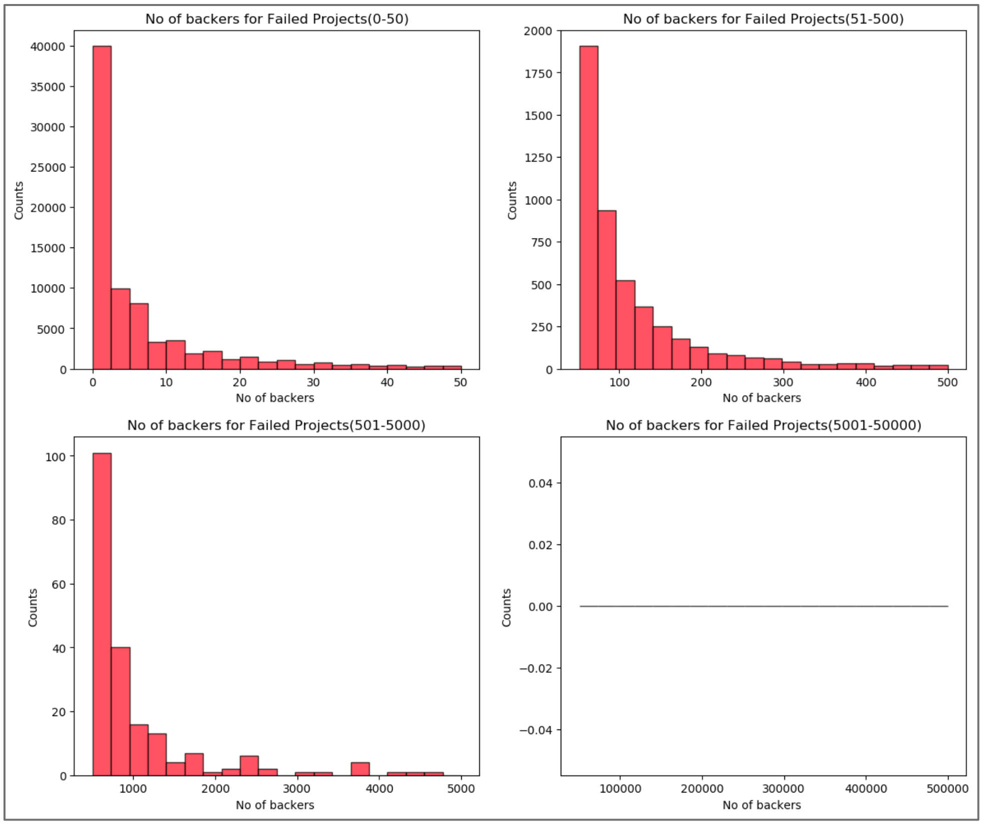


Figure 18: Number of backers' impact failure of projects

The above diagram shows the ratio of successful and failed projects in relation with the number of people backing the project.

- Most successful projects have 25-50 backers.
- Most successful projects have 0-25 backers.
- As the number of backers decline the count of successful and failed projects decline exponentially.

3.2..7. Does staff pick mark more successful projects?

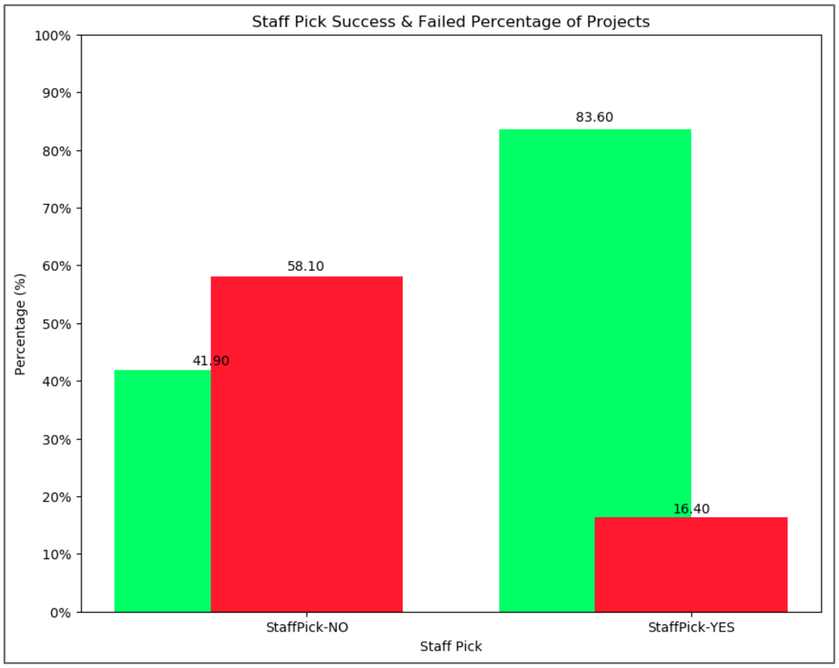


Figure 18: Staff pick for successful and failed project

The above diagram shows staff pick yes or no with successful and failed projects.

- When the project is not staff pick the ratio of success and failed project is close
- When the project is a staff pick the ratio of success is five times higher than a failed project

3.2..8. Is there relationship with a pledge to number of backer's ratio in the success or failure of the project?

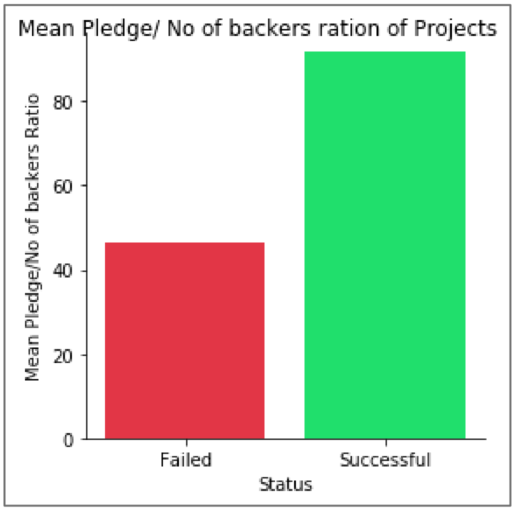


Figure 19: Pledge amount to number of backer ratio

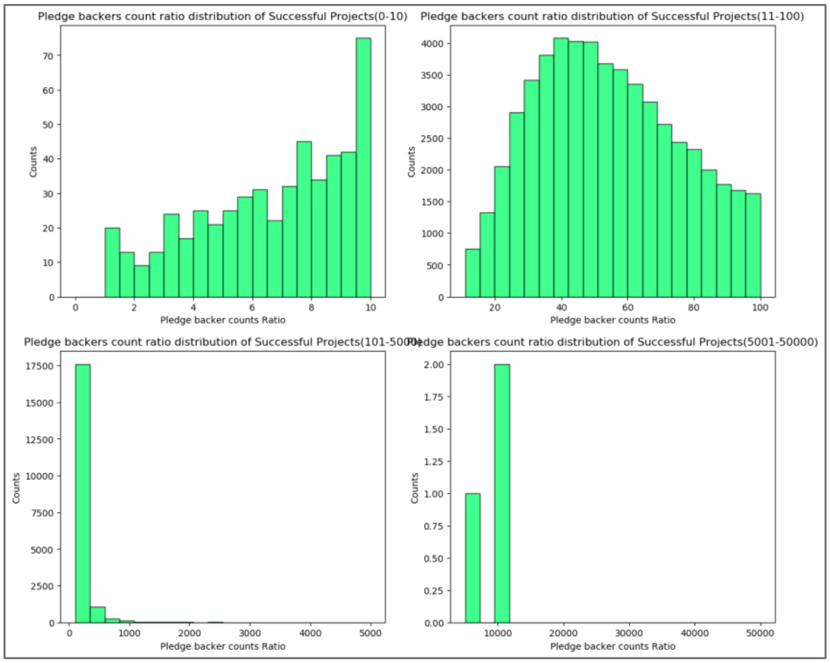


Figure 20: Pledge amount to no of the backer ratio for successful projects

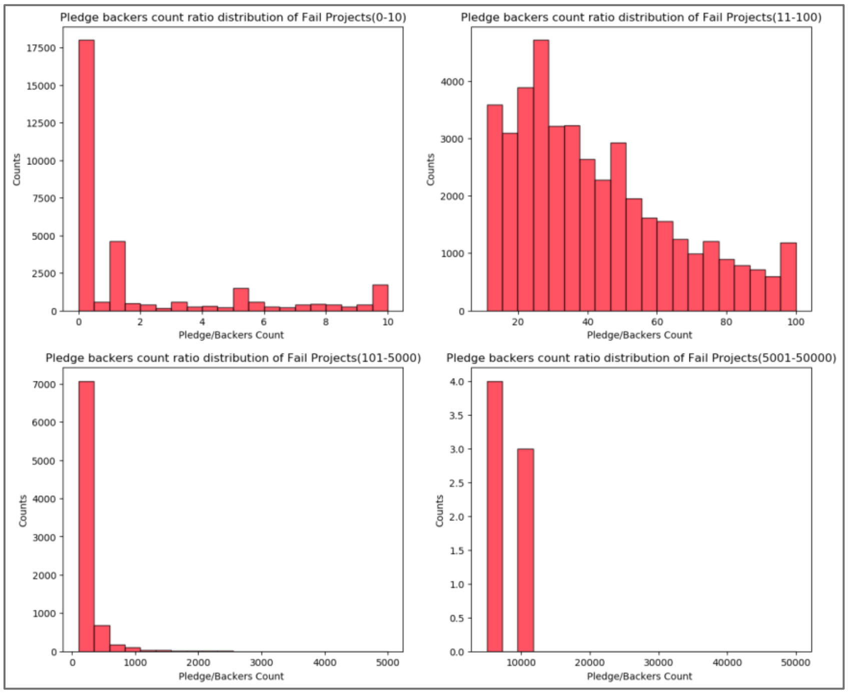


Figure 21: Pledge amount to number of backer ratio for failed projects

The above diagram shows ratio of pledge amount by number of backers for successful and failed projects.

- The average failed project get less than one USD per person
- The average successful project get 35-55 USD per person

3.2..9. Which country is having most and least successful or failed project?

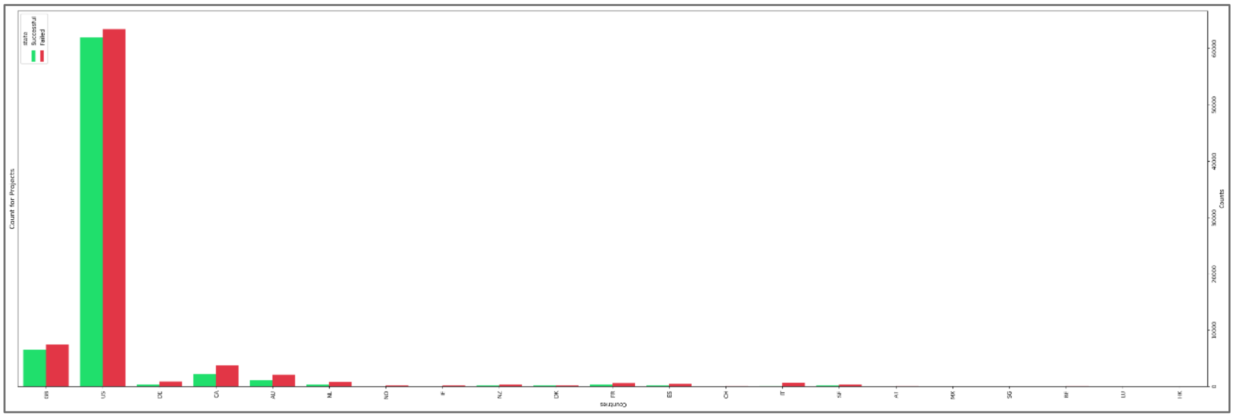


Figure 22: Country-wise distribution of projects

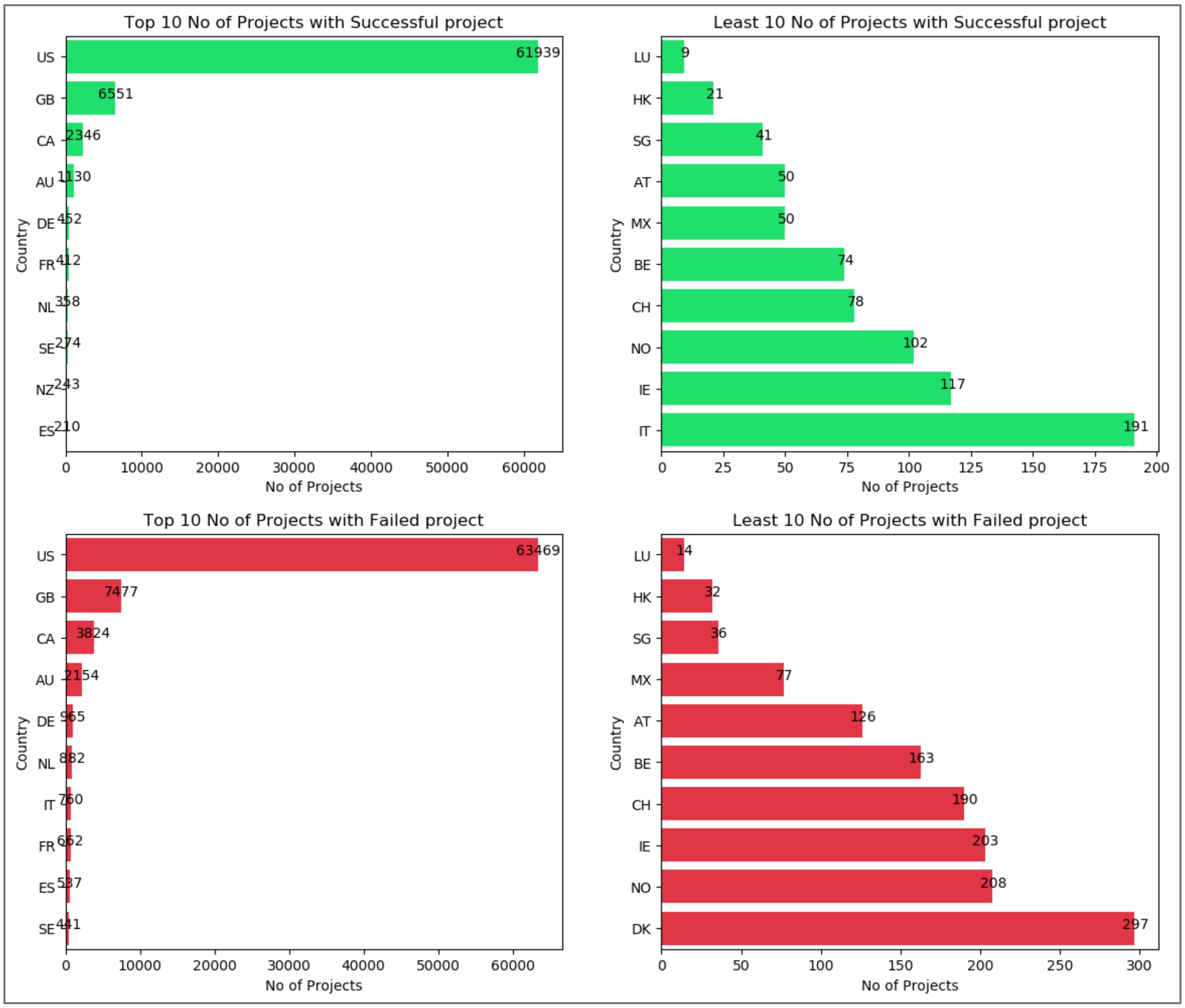


Figure 23: Country-wise Top 5 successful and fail

The above diagram shows the country-wise distribution of successful and failed projects.

- The most project belongs to the United States in both Success and Failed Status.
- Most countries in Top 10 and bottom 10 are same. It means these are most and least participating countries
- The ratio of successful and failed project for a particular country is approximately similar

3.2..10. What categories have most and least successful or failed project?

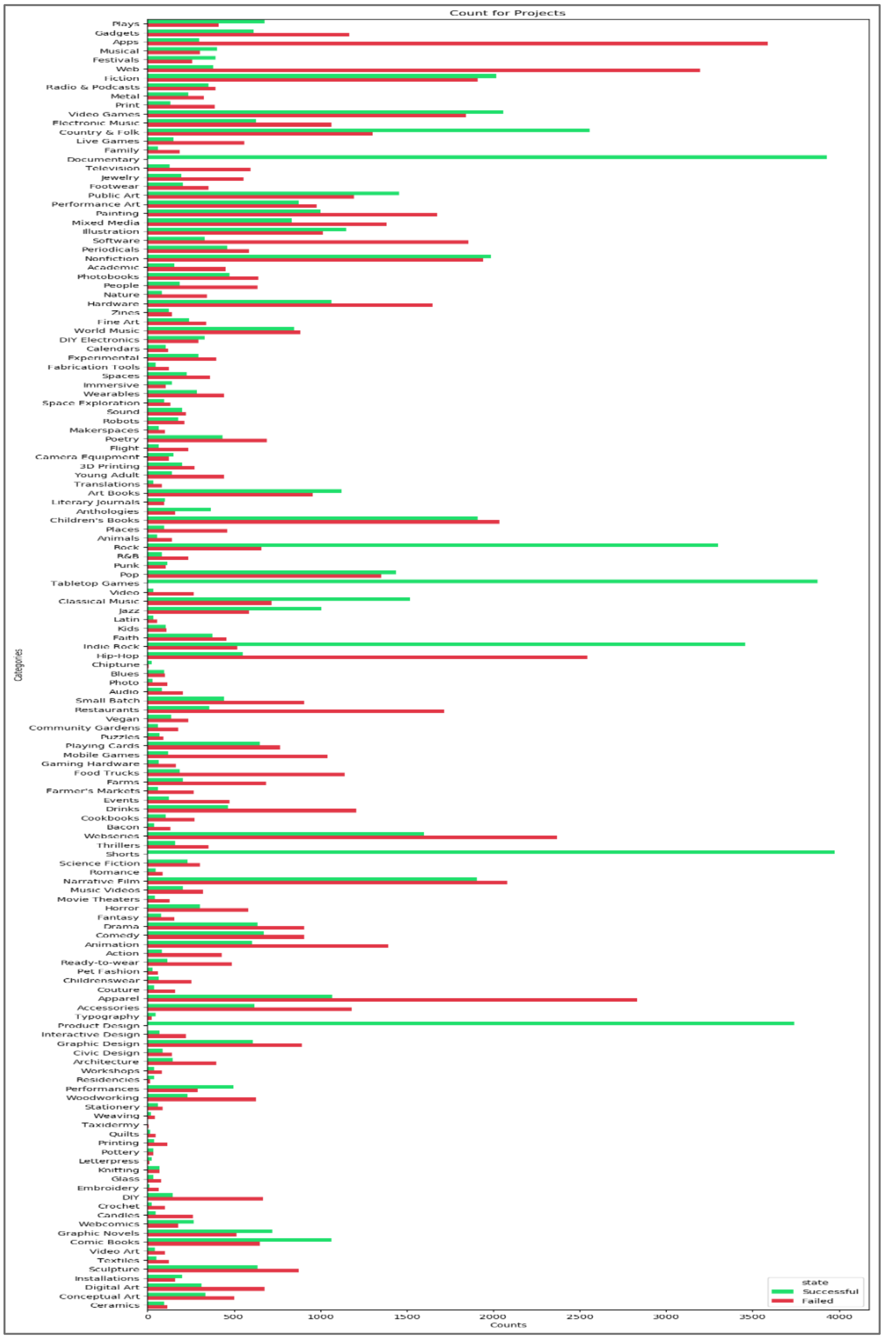


Figure 24: Category wise successful and failed projects

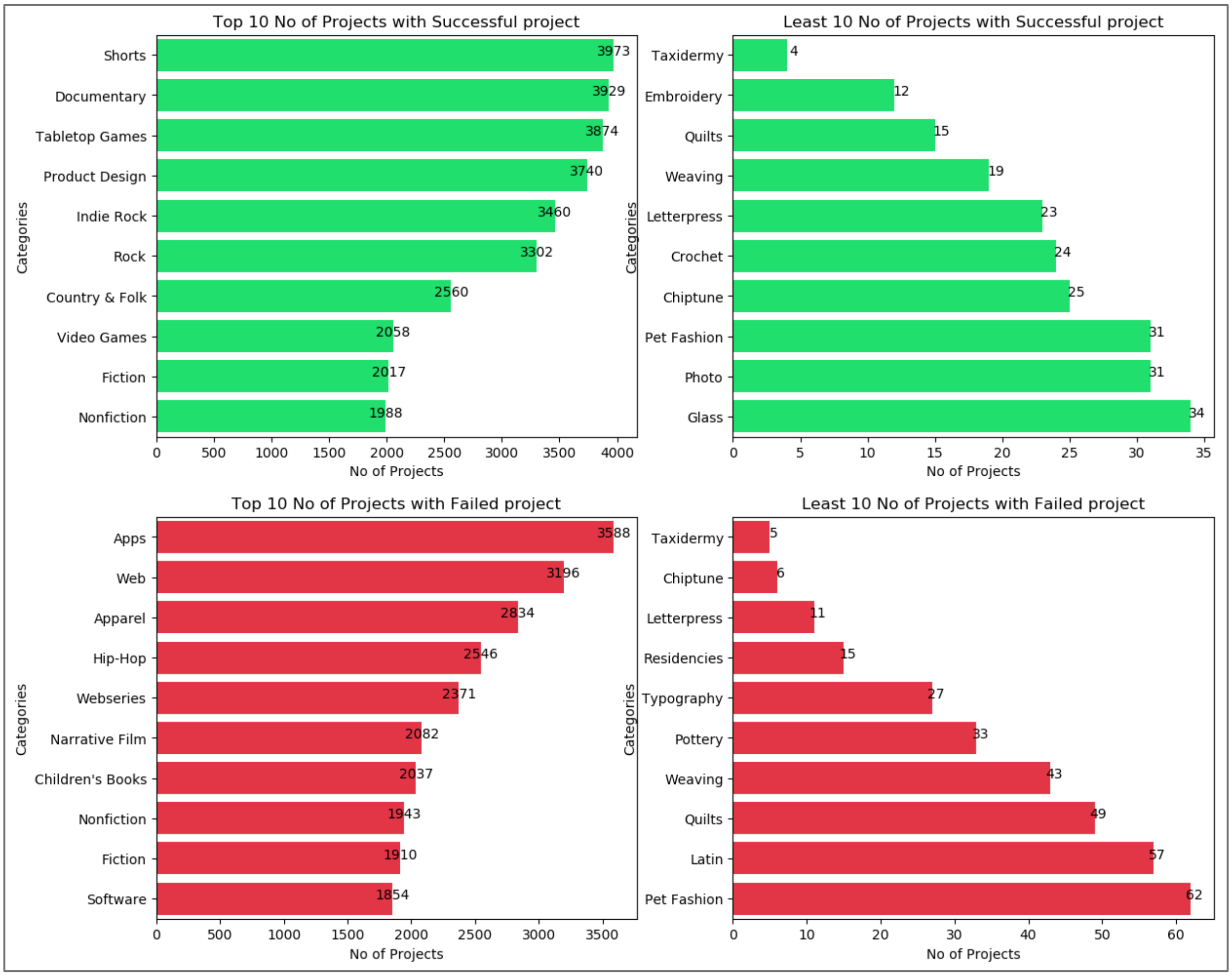


Figure 25: Top 5 categories for successful and failed projects

The above diagram shows the category-wise distribution of successful and failed projects.

- There are distinct 150 + categories of projects.
- Categories like Apps, Web have high failing ratio
- Categories like Product design and Shorts have high success ratio
- No Top 10 or bottom 10 categories are common, which means categories have the impact on the success or failure of the project

3.2..11. What location type is having most and least successful or failed project?

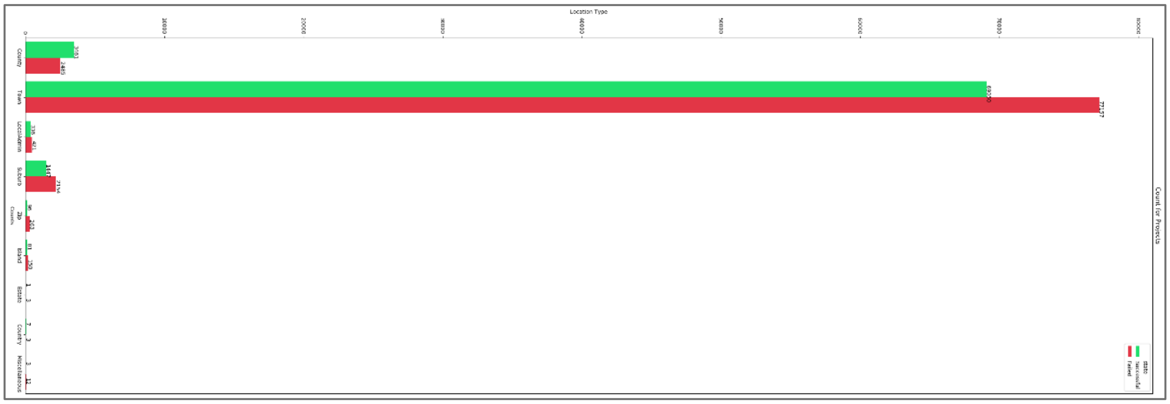


Figure 26: Location wise successful and failed projects

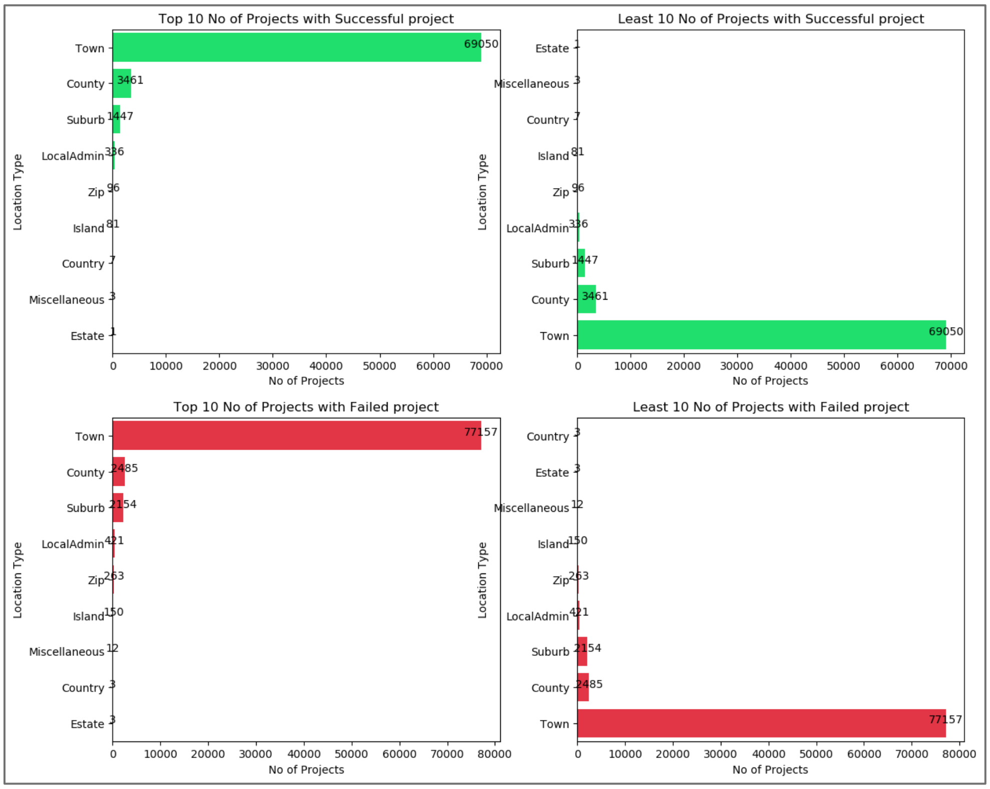


Figure 27: Top 5 location type for successful and failed

The above diagram shows location wise distribution of successful and failed projects.

- The most project belongs location type Town for both Success and Failed Status.
- The ratio of successful and failed project for a particular location is approximately similar.

4. Inferential Statistics

The inferential statistics section involved exploring columns or variables from the data to perform statistical analysis. This section applies some inferential statistical concept on the data.

4.1. Exploration of a number of backers

The column or variable number of backers for the project is a very interesting column and can be explored with inferential statistic tools.

4.1..1. Successful project exploration

The describe from pandas data fame provides some information

```
count    74847.000000
mean     291.705680
std      1844.760094
min      1.000000
25%      34.000000
50%      72.000000
75%      173.000000
max      219382.000000
```

4.1..2. Fail project exploration

The describe from pandas data fame provides some information

```
count    82896.000000
mean     15.552765
std      82.981983
min      0.000000
25%      1.000000
50%      3.000000
75%      10.000000
max      6550.000000
```

4.1..3. Hypothesis Testing

There is a difference between the mean of backer counts between successful and failed project. We can further analyze this hypothesis. Answer: The hypothesis is as follows

Null Hypothesis: There is no difference in a number of backers in successful or failed project. Which means for a number of backers for successful - means for number of backers for failed equals Zero.

Alternate Hypothesis: There is a significant difference in successful or failed project. Which means for successful - means for failed not equals Zero.

Calculate Z score and p score for the null hypothesis

Calculate Z stat using ztest method in a weightstat module with significance level 0.005
The calculated values are as follow for two-sided & larger
t-statistic: 40.9166979163
p-value: 0.0

Calculate T score and p value to test the same hypothesis

Calculate T score using the ttest_ind method in stats module with significance level 0.005
The calculated values are as follow
t-statistic: 40.9166979163
p-value: 0.0

The above p value is less than our significance value and hence there is enough evidence to reject Null hypotheses.

The code related to this part is in Capstone_Inferential_Statistic file.

5. Baseline Analysis

The baseline analysis is performed by applying logistic regression machine learning model. The model obtained is considered as a baseline for further analysis.

5.1. Converting columns

5.1..1. To dummies

The following category type columns are converted into dummy column using the get_dummies method in panda's module.

1. Category
2. Country
3. Location type

5.1..2. To number

The column staff pick which has value Yes or No is converted to 0 and 1

5.2. Column selection

There are few columns which are filtered out from part of the dependent variable or y in the algorithm. The idea is to include as many as column possible but remove any dummy to a variable or not required column.

1. Goal amount
2. Project id

The 177 columns used for the model including the one converted into dummies.

5.3. Logistic Regression

The logistic regression algorithm is used to create model [0.01, 0.1, 1, 10, 100]

5.3..1. Setting hyperparameter

The [0.01, 0.1, 1, 10, 100] are used for the hyperparameter to find the best estimator using gridsearch from the model selection.

5.3..2. Splitting train and test size

The train and test split are 80-20. The train_test_split is used to get the two sets.

5.3..3. Fitting data

The training data obtain by above split and the model return by grid search is used to fit the data and obtain the model.

5.3..4. Finding accuracy

The accuracy score is calculated on training and test data using score method of the model for L1 and L2 regularization parameter. The L2(default) provides more accuracy on test data as compared to L1.

Regularization parameter	accuracy of training data	accuracy of test data
L2(Default)	0.82	0.83
L1	0.78	0.78

Table 1: Logistic Regression Accuracy comparison

5.3..5. Creating Classification report

The classification report provides the information about various parameter like precision, recall f1-score and support. These figures tell about model performance based on how good the real fit is between test and train data based on true expected vs predicted.

The report for L1 and L2 regularization parameter is shown below. The L2 (default) is having better recall and precision as compared to L1.

Model features

yTrain length	y_predict_training	ytestlr length	y_predict_test
126194	126194	31549	31549

Table 2: Logistic Regression Model features

5.3..5.1. Regularization parameter L2

Classification Report (Training)

	Precision	recall	f1-score	support
0	0.78	0.92	0.85	66280
1	0.89	0.71	0.79	59914
Avg/total	0.83	0.82	0.82	126194

Table 3: Logistic Regression (L2) Training Classification Report

Classification Report (Test)

	Precision	recall	f1-score	support
0	0.78	0.92	0.85	16616
1	0.89	0.2	0.80	14933
Avg/total	0.84	0.72	0.80	31549

Table 4: Logistic Regression (L2) Test Classification Report

5.3..5.2. Regularization parameter L1

Classification Report (Training)

	Precision	recall	f1-score	support
0	0.74	0.89	0.81	66280

1	0.84	0.65	0.73	59914
Avg/total	0.79	0.78	0.77	126194

Table 5: Logistic Regression (L1) Training Classification Report

Classification Report (Test)

	Precision	recall	f1-score	support
0	0.74	0.89	0.81	16616
1	0.84	0.66	0.74	14933
Avg/total	0.79	0.78	0.78	31549

Table 6: Logistic Regression (L1) Test Classification Report

5.3..6. Creating confusion matrix

The confusion matrix is another representation of model performance by displaying in a column the expected vs actual in matrix form.

5.3..6.1. Regularization parameter L2

	Predicted False	Predicted True
Actual False	15336	1280
Actual True	4236	10697

Table 7: Logistic Regression (L2) Confusion Matrix

5.3..6.2. Regularization parameter L1

	Predicted False	Predicted True
Actual False	14811	1805
Actual True	5123	9810

Table 8: Logistic Regression (L1) Confusion Matrix

5.3..7. Finding feature importance with regularization L1

The regression model is used to find the important feature using standardize parameter and fitting model with Xtrain /standard deviation(Xtrain,0) and ytrain . The coefficient of this model provides with high influencing independent variable for both success and failure. The top five values with absolute values provide the important features in both success and failure of projects.

Top 5 feature with towards success (1): positive values

Sr.No	Feature	Coefficient
1	usd_pledged	3.270424
2	backers_count	2.68740
3	Shorts	0.678813
4	Documentary	0.607895
5	Product_Design	0.527864

Table 9: Logistic Regression Top 5 feature importance positive

Top 5 feature with towards fail (0): negative values

Sr.No	Feature	Coefficient
1	Apps	0.329265

2	Web	0.270048
3	Software	0.183272
4	deadline_days	0.182803
5	Restaurants	0.168901

Table 10 Logistic Regression Top 5 feature importance negative

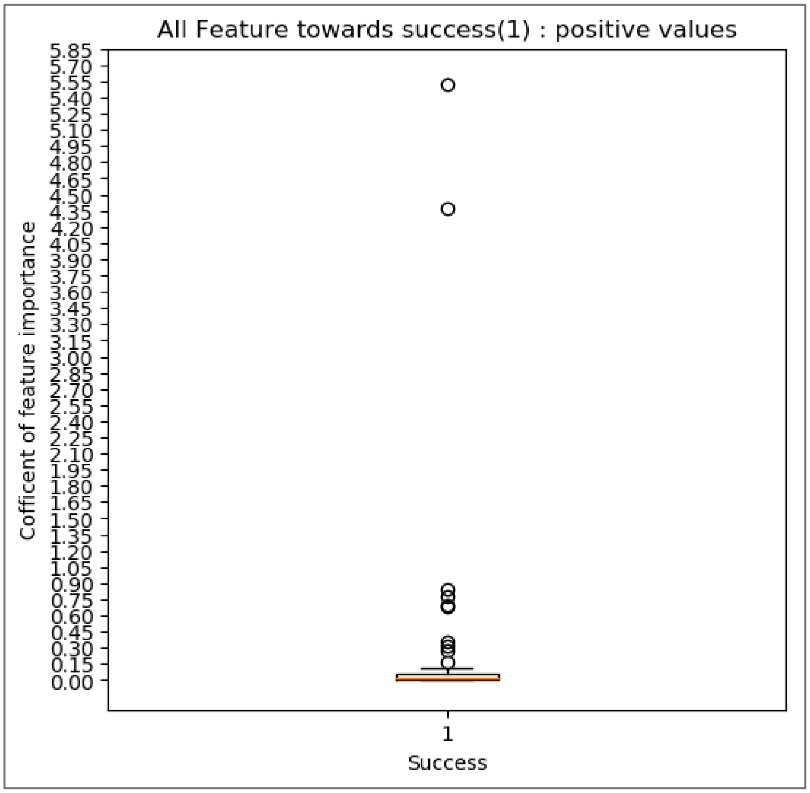


Figure 28: All Success feature spread

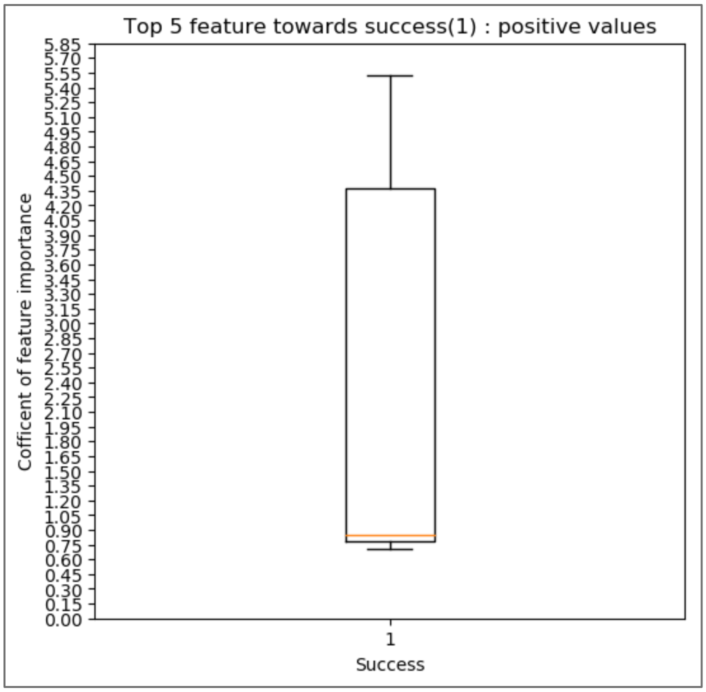


Figure 29: Top 5 Success feature spread

Top features influencing success of project/campaign (1): positive values

- The spread of top positive feature is large due to top 2 value(5.55 & 4.50) being relatively very high numbers
- Most value resides in lower range providing nearly same feature importance coefficient

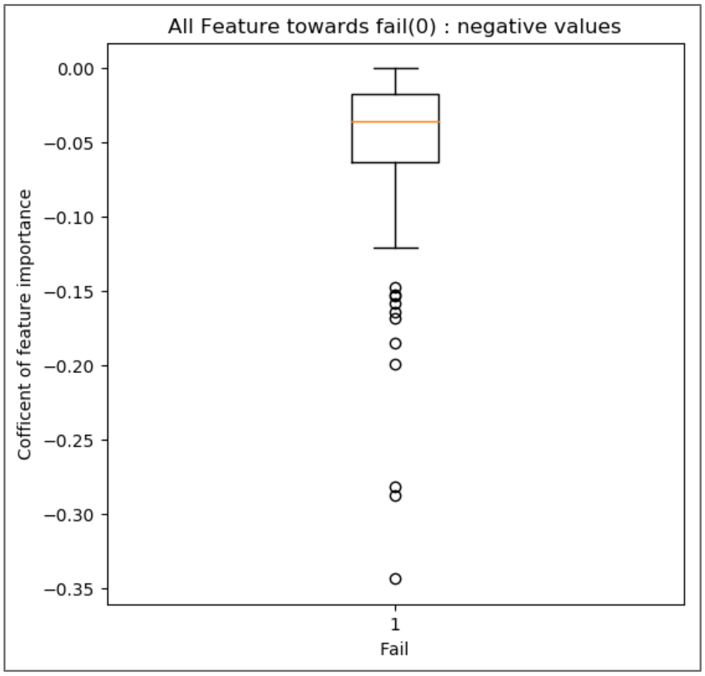


Figure 30: All fail feature spread

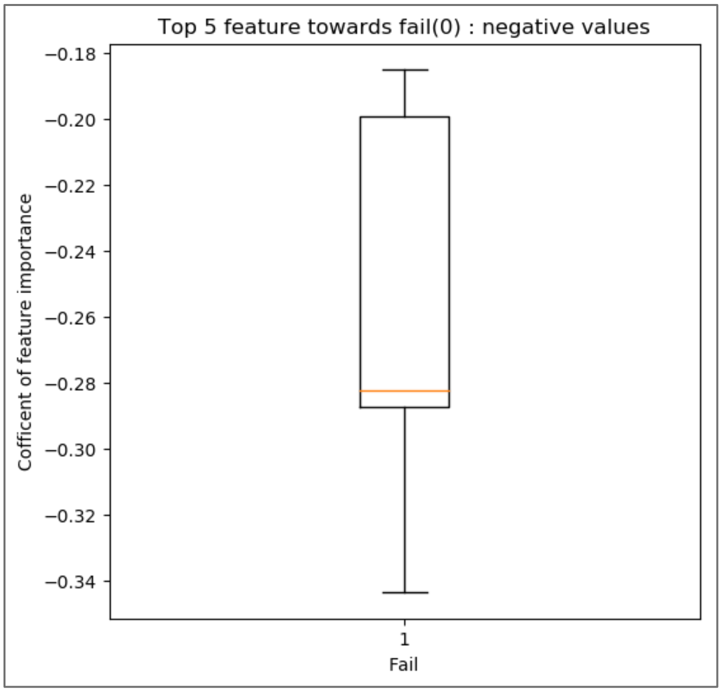


Figure 31: Top 5 fail feature spread

Top feature influencing failure of project/campaign (0): negative values

- The spread of top negative feature is large due to top 2 value (0.35 & 0.30) being relatively very high numbers.
- Most value resides in higher (absolute number) range providing, unlike success coefficient.
- The absolute value of fail is comparatively very small to success top most coefficient.

6. Further Analysis

The further analysis is performed by applying random forest classifier machine learning model. The model will be compared with a baseline to conclude final analysis.

6.1. Random Forest Classifier

The random forest classifier is applied to same test and training data to analyze and compare performance. The random classifier is run for three max features sqrt, log2 and none to compare the least oob_error (1-oob_score).

Initially, the classifier run for a range of n_estimator to get the oob_eror(1-oob_score) to find and compare the values. For better understanding, the effect of the n_estimator the multiple runs was experimented to see the effect on oob_error for higher values of c_estimator

6.1..1. Estimator 15 -175

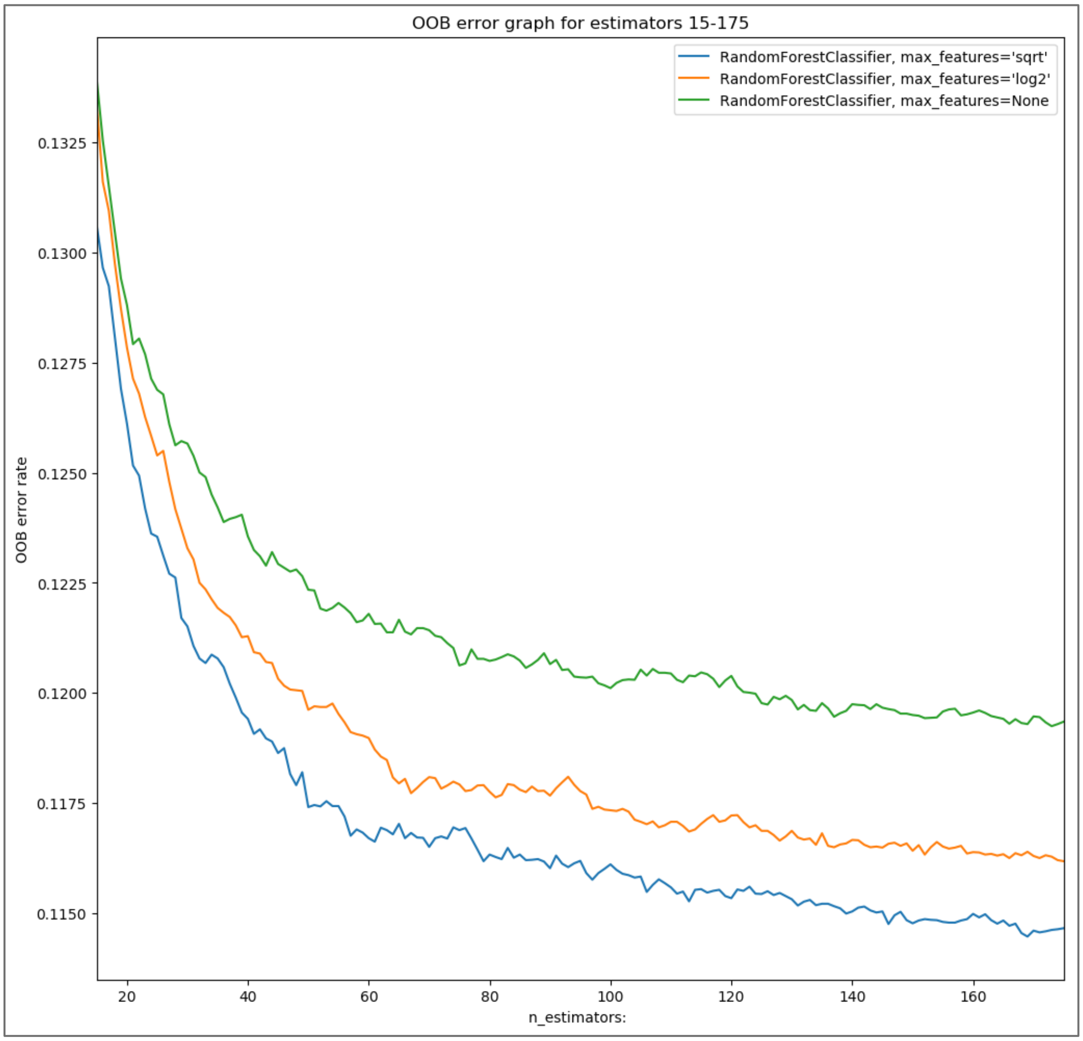


Figure 32: Random Forest Classifier (15-175)

OOB Error rate	Estimator	Max Feature
0.11446661489452747	169	'sqrt'
0.11617826521070729	175	'log2'
0.1192449720271962	173	None

The model created with the highlighted least OOB error rate with estimator **169.**

Model features

Accuracy of test data	oob error on model	yTrain length	y_predict_ training	ytestlr length	y_predict_test
0.89	0.11	126194	126194	31549	31549

Classification Report

	Precision	recall	f1-score	support
0	0.91	0.87	0.89	16616

1	0.87	0.90	0.88	14933
Avg/total	0.89	0.89	0.89	31549

Confusion Matrix

	Predicted False	**Predicted True**
Actual False	14533	2083
Actual True	1498	13435

Top 5 feature with towards success (1): positive values

Sr. No	Feature	Coefficient
1	usd_pledged	0.363263
2	backers_count	0.342628
3	deadline_days	0.058642
4	staff_pick	0.029299
5	Shorts	0.015465

Top 5 feature with towards fail (0): negative values

Sr. No	Feature	Coefficient
1	Taxidermy	1.944475e-05
2	Letterpress	1.605715e-05
3	Miscellaneous	1.065033e-05
4	Country	8.956387e-07
5	Estate	5.096736e-07

6.1..2. Estimator 100 -200

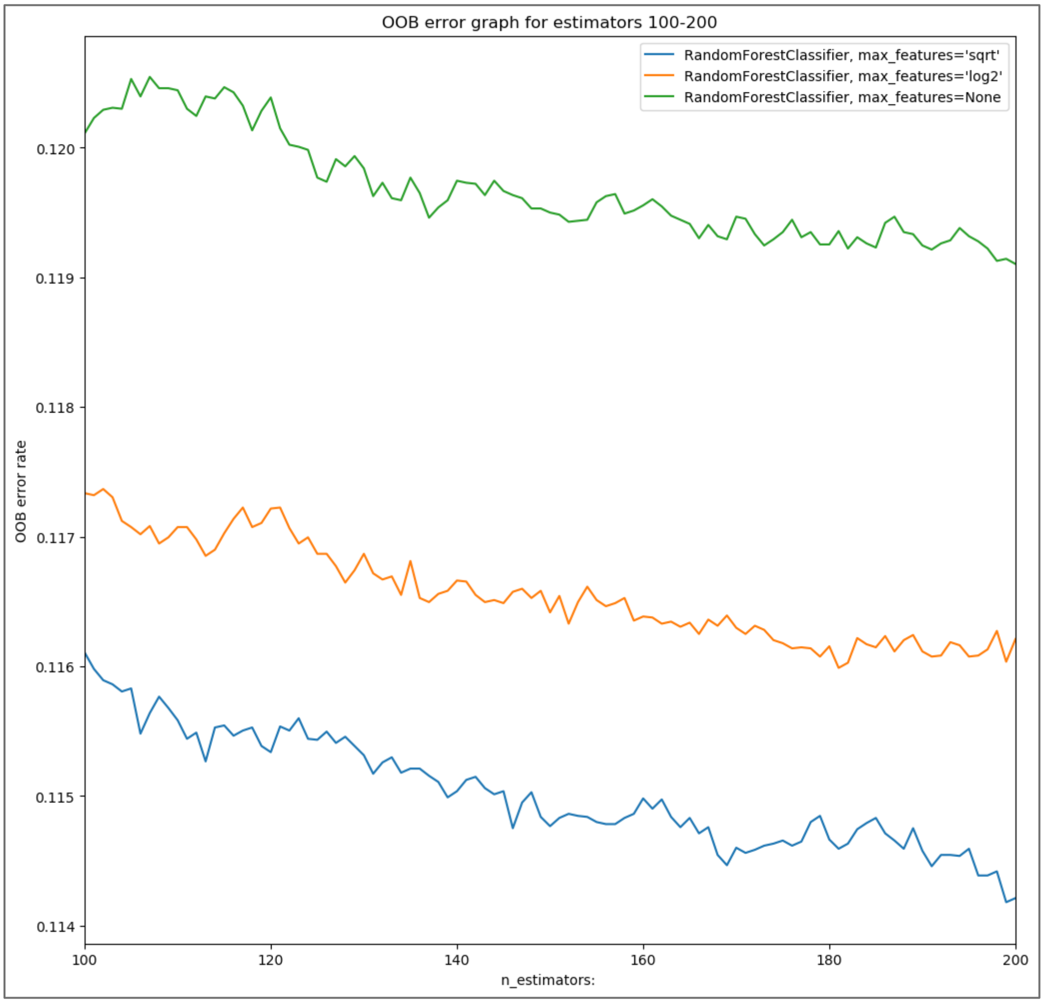


Figure 33: Random Forest Classifier (100-200)

OOB Error rate	Estimator	Max Feature
0.11418133984183088	199	'sqrt'
0.11598808184224285	181	'log2'
0.11910233450084795	200	None

The model created with the highlighted least OOB error rate with estimator **199**.

Model features

Accuracy of test data	oob error on model	yTrain length	y_predict_ training	ytestlr length	y_predict_test
0.89	0.11	126194	126194	31549	31549

Classification Report (Test)

	Precision	recall	f1-score	support
0	0.91	0.87	0.89	16616
1	0.87	0.90	0.88	14933

Avg/total	0.89	0.89	0.89	31549

Confusion Matrix

	Predicted False	**Predicted True**
Actual False	14533	2083
Actual True	1498	13435

Top 5 feature with towards success (1): positive values

Sr. No	Feature	Coefficient
1	usd_pledged	0.363263
2	backers_count	0.342628
3	deadline_days	0.058642
4	staff_pick	0.029299
5	Shorts	0.015465

Top 5 feature with towards fail (0): negative values

Sr. No	Feature	Coefficient
1	Taxidermy	1.944475e-05
2	Letterpress	1.605715e-05
3	Miscellaneous	1.065033e-05
4	Country	8.956387e-07
5	Estate	5.096736e-07

6.1..3. Estimator 150 -500

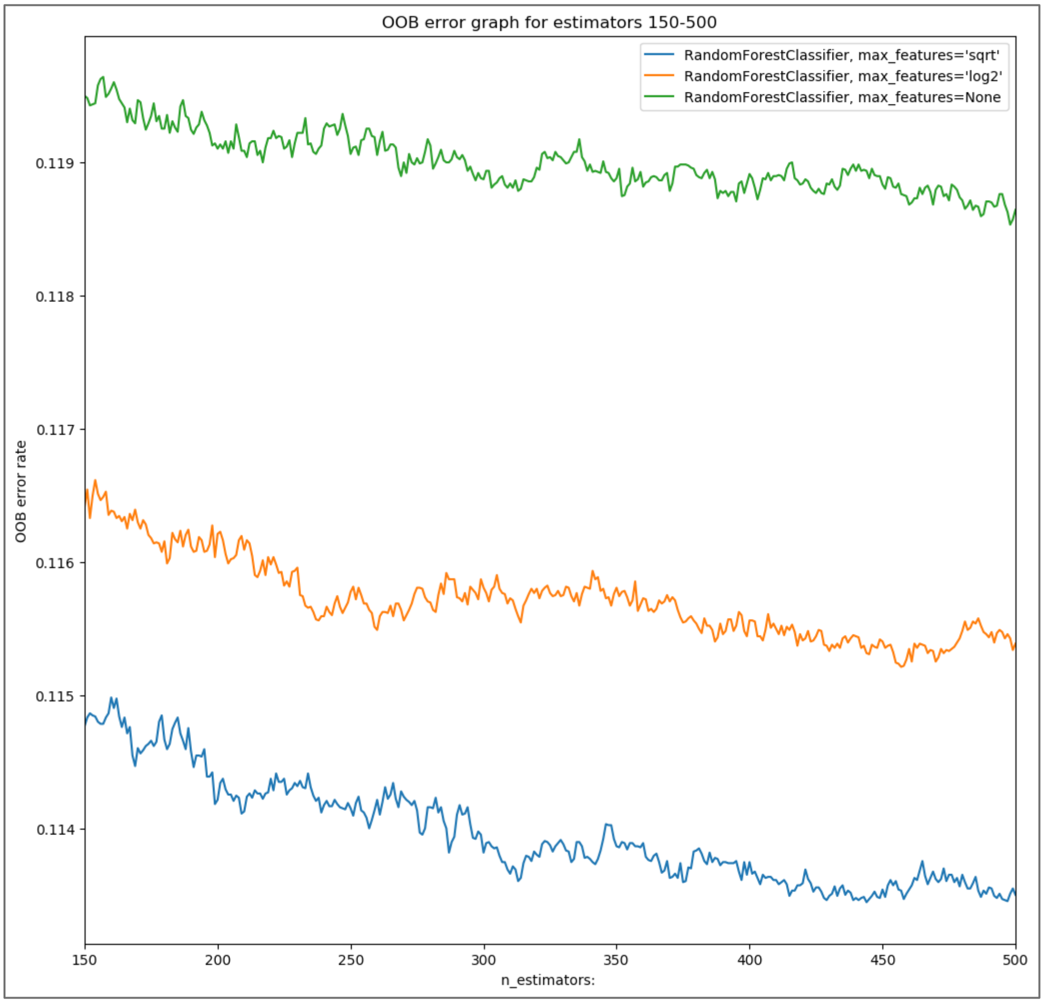


Figure 34: Random Forest Classifier (150-500)

OOB Error rate	Estimator	Max Feature
0.11344437928903117	444	'sqrt'
0.11521149975434652	457	'log2'
0.11853178439545464	498	None

The model created with the highlighted least OOB error rate with estimator **444**.

Model features

Accuracy of test data	oob error on model	yTrain length	y_predict_ training	ytestlr length	y_predict_test
0.89	0.11	126194	126194	31549	31549

Classification Report

	Precision	recall	f1-score	support
0	0.91	0.87	0.89	16616

1	0.87	0.90	0.88	14933
Avg/total	0.89	0.89	0.89	31549

Confusion Matrix

	Predicted False	**Predicted True**
Actual False	14533	2083
Actual True	1498	13435

Top 5 feature with towards success (1): positive values

Sr. No	**Feature**	**Coefficient**
1	usd_pledged	0.363263
2	backers_count	0.342628
3	deadline_days	0.058642
4	staff_pick	0.029299
5	Shorts	0.015465

Top 5 feature with towards fail (0): negative values

Sr. No	**Feature**	**Coefficient**
1	Taxidermy	1.944475e-05
2	Letterpress	1.605715e-05
3	Miscellaneous	1.065033e-05
4	Country	8.956387e-07
5	Estate	5.096736e-07

6.1..4. Estimator 300 -1000

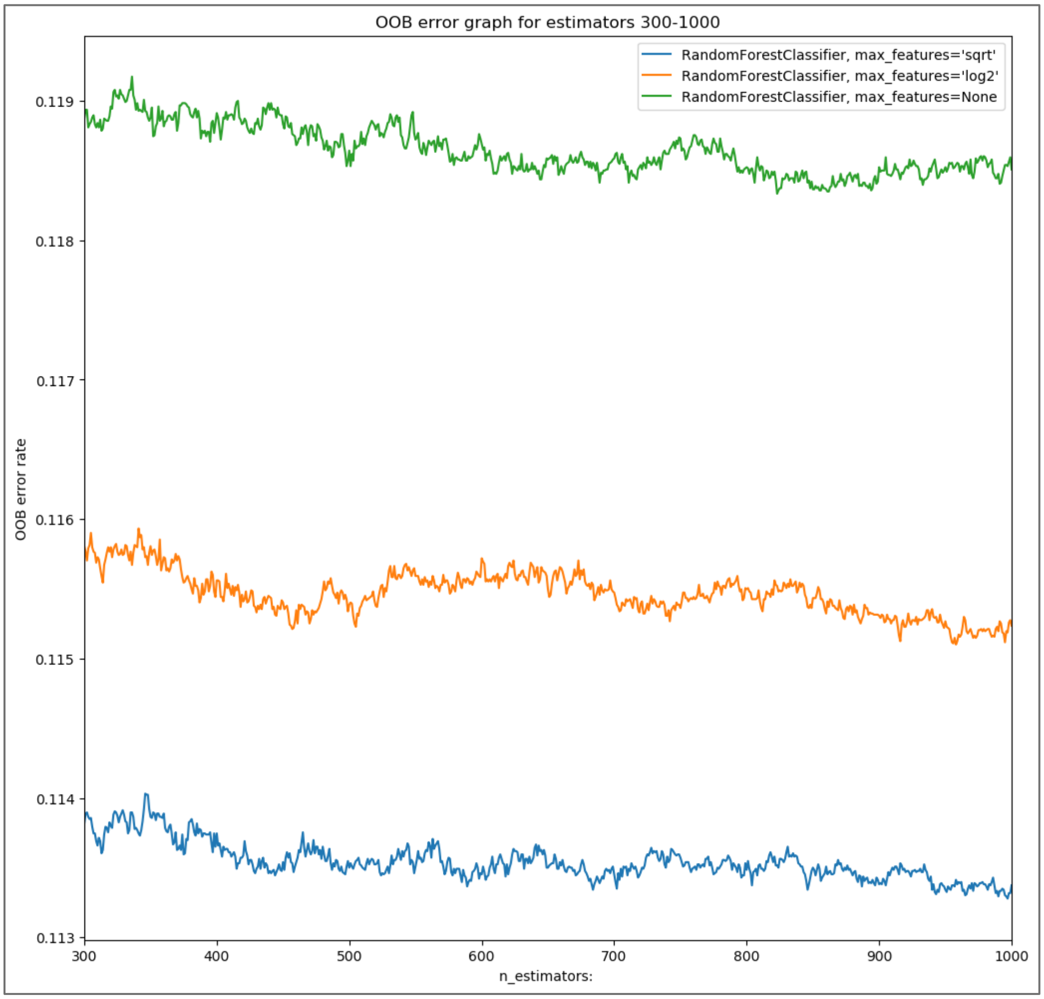


Figure 35: Random Forest Classifier (300-1000)

OOB Error rate	Estimator	Max Feature
0.11327796884162478	997	'sqrt'
0.11510055945607556	958	'log2'
0.11833367671997086	823	None

The model created with the highlighted least OOB error rate with estimator **997**.

Model features

Accuracy of test data	oob error on model	yTrain length	y_predict_ training	ytestlr length	y_predict_test
0.89	0.11	126194	126194	31549	31549

Classification Report

	Precision	recall	f1-score	support
0	0.91	0.87	0.89	16616

1	0.87	0.90	0.88	14933
Avg/total	0.89	0.89	0.89	31549

Confusion Matrix

	Predicted False	Predicted True
Actual False	14533	2083
Actual True	1498	13435

Top 5 feature with towards success (1): positive values

Sr. No	Feature	Coefficient
1	usd_pledged	0.363263
2	backers_count	0.342628
3	deadline_days	0.058642
4	staff_pick	0.029299
5	Shorts	0.015465

Top 5 feature with towards fail (0): negative values

Sr. No	Feature	Coefficient
1	Taxidermy	1.944475e-05
2	Letterpress	1.605715e-05
3	Miscellaneous	1.065033e-05
4	Country	8.956387e-07
5	Estate	5.096736e-07

The code related to this part is in Capstone_ML_LogisticRegression_RandomForestClassifier.ipynb file.

6.2. Model Comparison &Conclusion

Model features Test Data

	Random Forest Classifier 169	Random Forest Classifier -199	Random Forest Classifier - 444	Random Forest Classifier-997	Linear Regression -L2	Linear Regression -L1
Accuracy	0.89	0.89	0.89	0.89	0.83	0.78
Precision(1)	0.87	0.87	0.87	0.87	0.89	0.84
recall(1)	0.90	0.90	0.90	0.90	0.71	0.66
f1-score(1)	0.88	0.88	0.88	0.88	0.80	0.74
support(1)	14933	14933	14933	14933	14933	14933
Precision(0)	0.91	0.91	0.91	0.91	0.78	0.74
recall(0)	0.87	0.87	0.87	0.87	0.92	0.89

f1-score(0)	0.89	0.89	0.89	0.89	0.85	0.81
support(0)	16616	16616	16616	16616	16616	16616
False(Actual /Predicted)	14533	14533	14533	14533	15336	14811
True(Actual/ Predicted)	13435	13435	13435	13435	10697	9810

Conclusion on the Model performance:

The two machine learning algorithm random forest classifier and logistic regression were compared with each other and also with some variation of parameters within the same algorithm.

The Random forest with estimator with 169-997 provides insight that model work similar and hence increasing estimator does not provide any significant improvement on the model.
In Linear regression with L1 an L2 as regularization parameter, the L2 provides little better performance.

Between two machine algorithm, the random forest classifier performed well over logistic regression on many parameters. The accuracy improved with random forest classifier. The recall & f-1 score for positive(1) improved significantly with random forest but logistic regression did perform well for some false(0) data too.

6.3. Conclusion and Recommendation

We explored data and found out the best machine learning algorithm for predicting the success of Kickstarter campaigns/projects.

1. All sources of datasets contributed to the predictive power of the model
2. Out of supervised classification models, the Random Forest Classifier provided the best results
3. We performed feature importance analysis and identified features strongly contributing to the success and failure of the campaign
4. Kickstarter can add value to their current process of staff pick selection. They can use this model and understand if staff pick feature can take this project through the finish line
5. Through our model the Project Owner and Project backer can estimate the success or the failure of the campaign before launching or investing respectively
6. Project backer can study the graphs and trends analyzed in our project to learn more about categories and their success rates.

7. Future Work

The project has tried to cover many aspects of data science. But there is still scope to improve the methodology or explore more facets of the project. As for endnotes, this will be marked as future work in reference to this project

1.1. Scope:

There can be various crowdfunding sources, however this Project will be limited to campaigns funded using a website called Kickstarter.

In future we can include data from multiple companies like GoFundMe, Indiegogo etc.

1.2. Launch Period:

Study the restriction related to this model based on the age of the Kickstarter project

1.3. Technology:

Currently, the files are processed using code running in a Jupiter notebook on the local computer. The process like data wrangling, data cleaning and modeling took hours to finish on a local machine. Cloud technologies like **AWS, Azure, and Cloudera** etc. can be used to speed-up the processing.

We can use batch processing for getting new data every month and update the model.

Create a web application for visualization.