

Predicting success of crowdfunding campaigns on Kickstarter​

# BANA 7047-002 Group #3: Vallabh Reddy Utkarsh Singh Soumya Halder Rinisha M Saurabh Wani

# Executive Summary

* Text mining on the description field helps increase the overall accuracy significantly (by >5% with logistic regression)
* Goal amount is the most important determinant of success rate. Higher the goal, lower are the chances of success. Time taken to design the campaign, the category and subcategory of project, the ease of readability of the description and the day of launch are some other important factors
* Successful creators take longer to prepare their campaigns but run it for a shorter duration
* After the campaign is launched, the number of backers attracted so far becomes the primary determinant of success
* A Random Forest model provides the best overall accuracy – 75% and 92% accuracy for pre and post campaign launch predictions respectively

# About Kickstarter

Kickstarter, launched in April 2009,is an online crowd-funding platform for entrepreneurs to raise funds from the online community. Campaigns are intended to turn ideas into reality. It’s where creators share new visions for creative work with the communities that will come together to fund them. Project creators from across 22 countries can launch campaigns that are open to backers all over the world for funding. So far, Kickstarter has launched over 480,000 projects across 15 categories, collecting over USD 4.8 billion. The platform charges a 5% commission on successfully funded projects. Project backers receive rewards in the form of limited-edition product copies, exclusive user experiences etc. in return (It is not an equity-sharing model)

# Kickstarter Campaign Overview

Any person or team of people launching their kickstarter campaign or project have to go through the following five step process:

### Conceptualise

Pick a sub-category/theme that increases the chance of successful funding

### Curate content

Decide the project name, create description(“blurb”) and upload videos, photos, demo tools

### Set targets

Decide campaign duration, set optimum goals and create reward packages

### Initiate follower links

Build and grow online/offline networks and link social media channels and blog pages

### Plan launch

Plan week, day and time of launch to maximise initial views and notify followers

# Project Scope Identify factors that contribute to campaign success and predict success probability pre-launch and post-launch

# Problem Statement

Identify key attributes of the campaign that determine success, and quantify their impact

# Objective

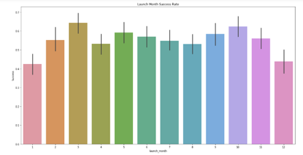
* Recommend creators on planning campaign content and launch to maximize the probability of success
* Identify projects that are below the threshold score for “success”, notify them about possible improvements

# Data Overview Our dataset has information on 211271 kickstarter campaigns (rows) and 38 columns. Of which, we have sub-categorized relevant columns as columns having data for dimension and columns not having data for dimension:

* Project Creation  
  Creator ID, Location, Creation date, Project ID, Category/Sub-category. All columns have data for dimension
* Content   
  Description, Multimedia, Network, Ad/Social media pages, Reward. Only the column on description has data for dimension
* Launch Plan  
  Launch date and launch time are the columns having data for dimension whereas the column ‘notification list’ does not have data for dimension.
* Leading indicators   
  Number of backers, Amount pledged and staff pick. All these columns have data for dimension

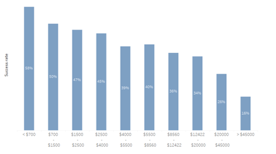
# Exploratory Data Analysis

Initial EDA presents scope of increasing probability of success through data-driven decision making



Success rate vs month of launch

Insight: Campaign success rate varies across campaign launch months



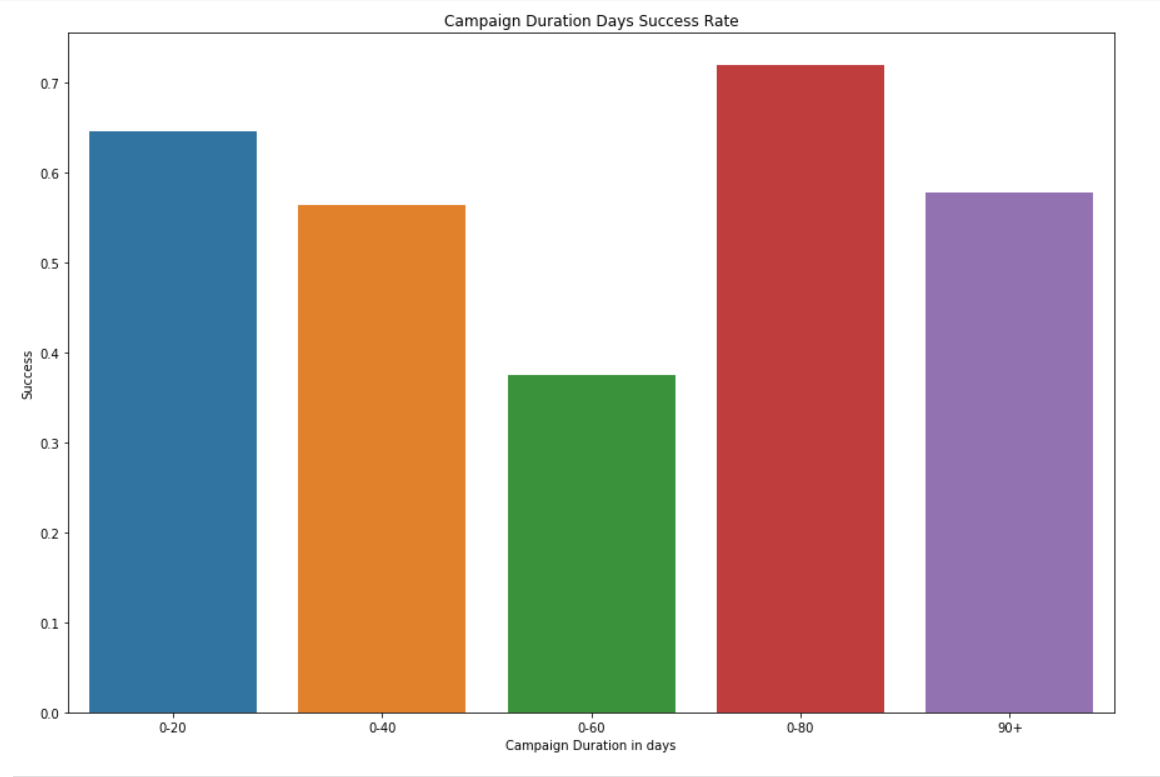
Success rate vs Goal amount

Insight: Goal amount is negatively correlated with success rate



Success rate across sub-categories

Insight: Success rate varies across categories



Success rate vs Campaign duration

Insight: Success rate varies with campaign duration

# Approach

Create a recommendation framework based on the success propensity:

* Estimate the probability of success before launch based on content uploaded
* Suggest improvement tips for a good launch
* Monitor response and refresh probabilities using a post-launch model with new predictors

# Detailed Approach:

* Text Mining:   
  Basically, we intend to use NLP algorithms for converting text fields viz. project name, blurb et al to have an output in the form of flesch readability score or keyword search/tokenization which then can be used as predictors for our models
* Model building:   
  Building two models, firstly a pre-launch success prediction model by using pre-launch campaign attributes like Project theme, content quality, campaign creation time, planned duration and goal amount to predict probability of success. Secondly, a post-launch success prediction model for tracking leading indicators such as number of backers, amount pledged, staff pick and additional predictors to refresh probabilities and revise recommendations
* Model Training:   
  Training ensemble tree-based models to obtain variable importance: by way of firstly creating feature set from all the text, numeric and categorical fields available. We’ll split the data into training and testing samples in the ratio 80:20. Post which, pass the feature matrix and response vector(train) into candidate models viz. Random Forest, AdaBoost and XGBoost for analysing variable importance results.
* Model Validation & Testing:   
  Using the identified important variables as the final set of predictors for predicting the probability of success before and after the launch of the campaign. Also, cross-validation on the training sample will be used for model selection. And, model performance will be tested on an out-of-sample test data.
* Performance Metrics  
  Since no real investment is made until the project succeeds, a symmetric misclassification rate will be primary metric to measure performance along with AUC, model lift percentage and K-statistic

# Modelling Phase

We’ve covered modelling for success of the campaign at two phases. First, before the launch of the campaign where we based our predictions purely on the content, goals and nature of the campaign. Second, post-launch where we included the performance

## Predictions at time of launch

In this section, the “number of backers” feature is not included for prediction, as this will be available only in a few days past launch

#### Modelling without text features

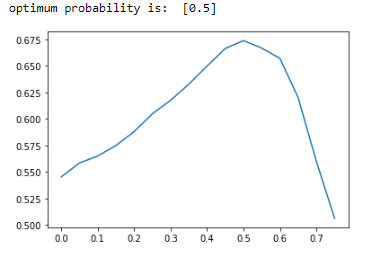
Reporting model performances without the Bag-Of-Words text features

The data was split into Train (70%), Validation (10%), Test (20%)

### **Model 1: Logistic regression**

* **Accuracy**: 66.6%

Below is the ROC curve for the model



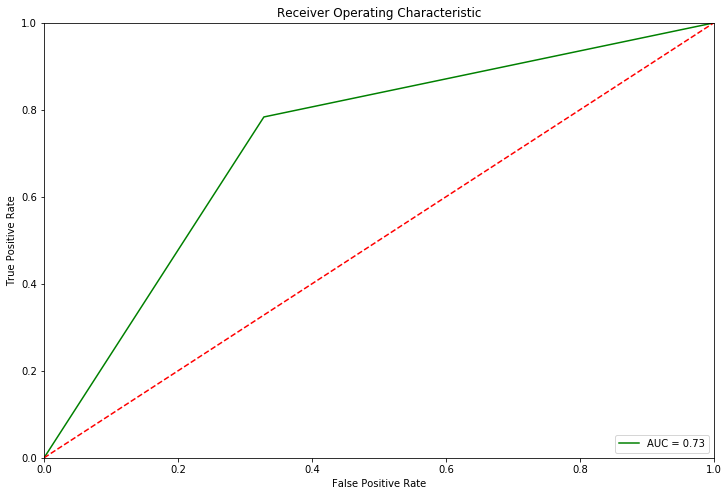
### **Model 2: Random Forest**

* **Accuracy**: 73.4%

#### Confusion Matrix

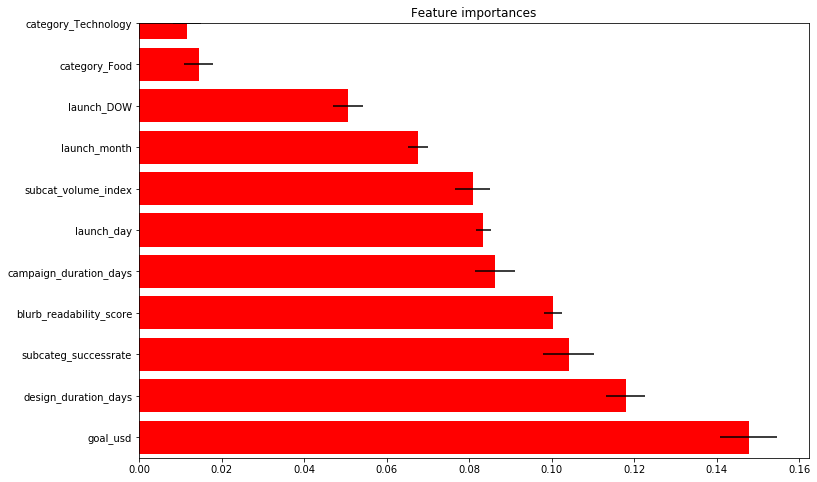
|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Prediction | |
| Failure | Success |
| Actual | Failure | 11451 | 5596 |
| Success | 4358 | 15795 |

#### ROC Curve



* **AUC**: 73%

#### Variable Importance Chart

****

### **Model 3: Neural Network**

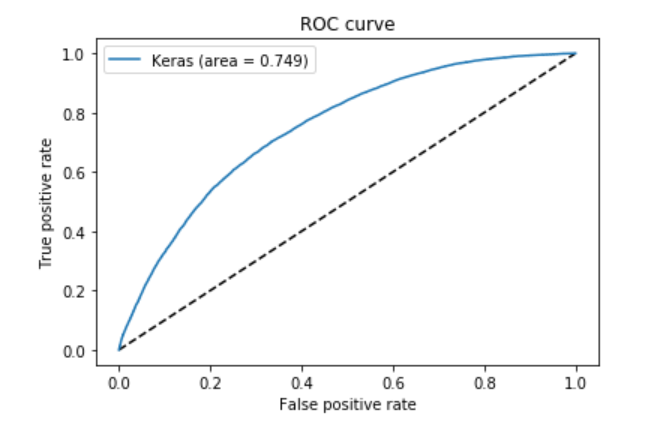
## Model Properties

* **Model Architecture**
  + 18 layers in total
    - Input layer is the size of the number of features, i.e., 44
    - 16 hidden layers each with 300 neurons.
    - 1 output layer with 2 neurons, one for each possible outcome.
* **Activation function** for hidden layers: ReLu
* **Activation function** for output layer: Softmax
* **Optimizer used**: Adam
  + Tried the learning rates [0.000001, 0.0001, 0.001, 0.01 and 0.1] and found that **'0.001'** had the best performance.
* **Loss**: Categorical Crossentropy
* Fitting
  + Validation split: 30%
  + Number of epochs: 30
  + EarlyStopping callback patience: 4

## Model Performance

* **Accuracy**: 68.34%
* **Sensitivity**: 72.5%
* **Specificity**: 60%

#### ROC Curve



**AUC**: 74.9%

#### Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Prediction | |
| Failure | Success |
| Actual | Failure | 14521 | 6865 |
| Success | 5517 | 10297 |

#### Modelling with text features

Steps followed:

* Text cleaning​
* Remove punctuations​
* Remove stop words​
* Remove words shorter than length of 3​
* Word stemming using Porter Stemmer​
* Count vectorization​
* Transformation into a sparse array

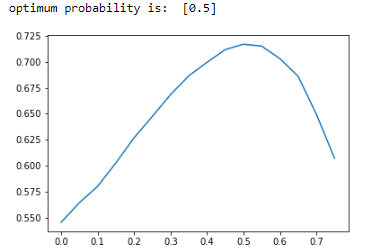
Reporting model performances with the Bag-Of-Words text features included

**Input matrix dimensions**

Split into Train (70%), Validation (10%), Test (20%)

### **Model 4: Logistic regression**

* **Accuracy: 71.4%**



* **Lift of 5% obtained with text mining**

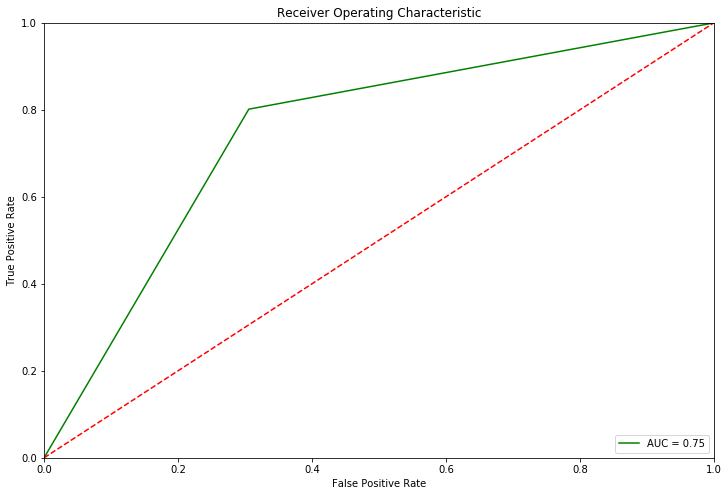
### **Model 5: Random Forest**

* **Accuracy**: 75.22%

#### Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Prediction | |
| Failure | Success |
| Actual | Failure | 11835 | 5212 |
| Success | 3997 | 16156 |

#### ROC Curve



* **AUC**: 75%

### **Model 6: Neural Network**

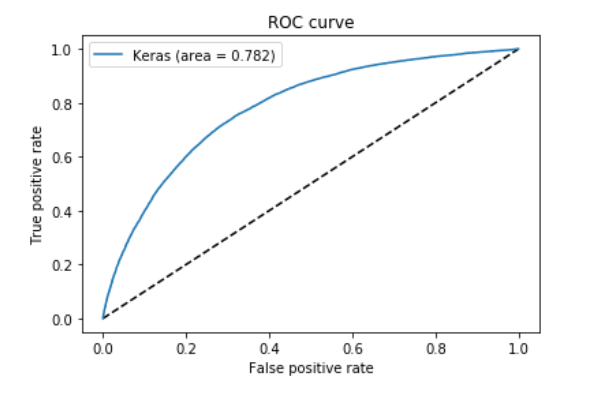
## Model Properties

* **Model Architecture**
  + 18 layers in total
    - 1 Input layer size equal to number of features.
    - 1hidden layer of size 2000 neurons.
    - 1 hidden layer of size 700 neurons.
    - 1 hidden layer of size 500 neurons.
    - 13 hidden layers of size 300 neurons each.
    - 1 output layer with 2 neurons, one for each possible outcome (1 or 0).
* **Activation function** for hidden layers: ReLu
* **Activation function** for output layer: Softmax
* **Optimizer used**: Adam
  + Tried the learning rates [0.000001, 0.0001, 0.001, 0.01 and 0.1] and found that **'0.001'** had the best performance.
* **Loss**: Categorical Crossentropy
* Fitting
  + Validation split: 30%
  + Number of epochs: 30
  + EarlyStopping callback patience: 4

## Model Performance

* **Accuracy**: 68.93%
* **Sensitivity**: 77%
* **Specificity**: 60%

#### ROC Curve



**AUC**: 78.2%

#### Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Prediction | |
| Failure | Success |
| Actual | Failure | 10297 | 6865 |
| Success | 5517 | 14521 |

# Predictions post-launch

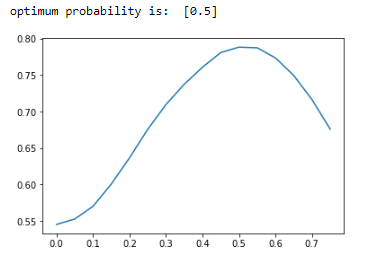
* In this section we will include the no\_of\_backers feature along with all other features
* We have already demonstrated the lift obtained by text mining, so we will include text features in this final phase

The data was split into Train (70%), Validation (10%), Test (20%)

## **Model 7: Logistic regression**

* **Test Accuracy:** 78.5%

#### ROC Curve



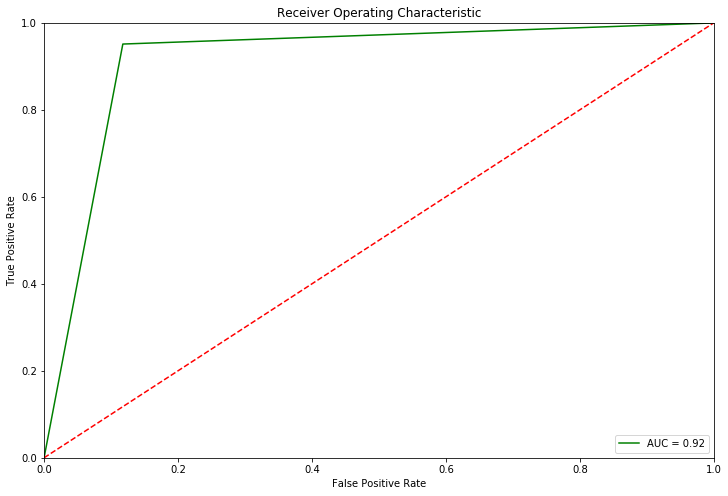
## **Model 8: Random Forest**

* **Test Accuracy:** 91.97%

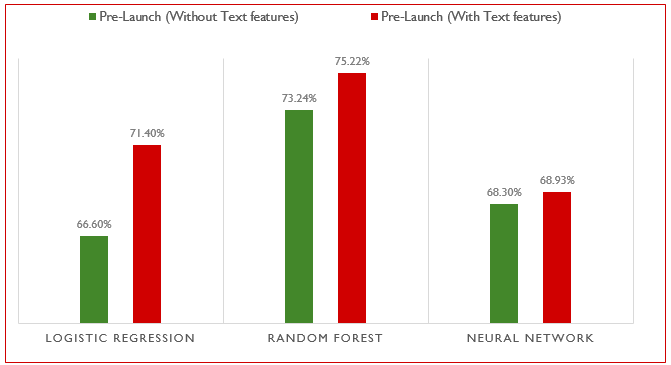
#### Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Prediction | |
| Failure | Success |
| Actual | Failure | 15042 | 2005 |
| Success | 979 | 19174 |

#### ROC Curve

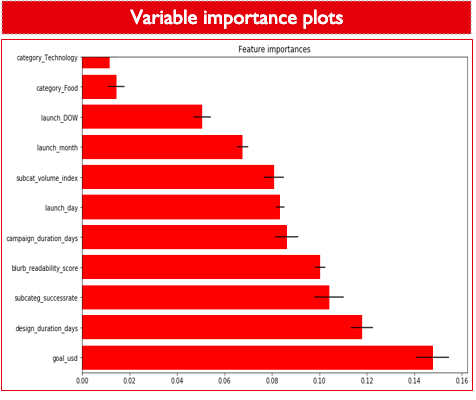


* **AUC**: 92%

****

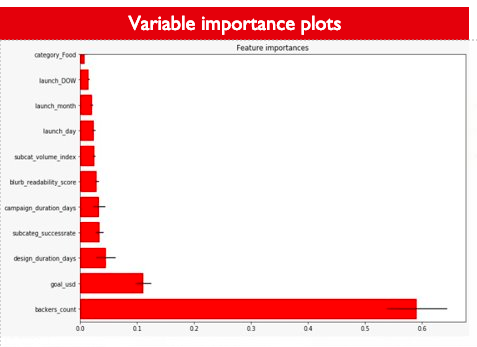
* **Text mining features improved the model performance significantly.** Lift in overall accuracy obtained by including text features:
* Logistic Model -4.8%
* Random Forest- 2%
* Neural Network- 0.6%

It can be concluded that Logistic Regression utilizes the text features most efficiently

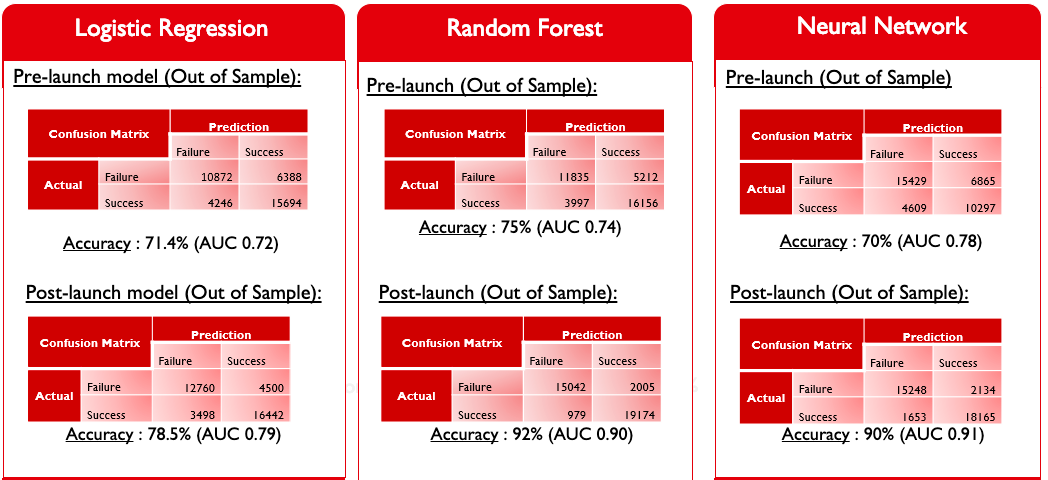
* Random Forest variable importance result (Pre-launch model without Bag-of-words text feature matrix)  
  

The goal amount is the most important predictor followed by the campaign design duration (days), subcategory success rate and the blurb readability score

* Random Forest variable importance result (Post-launch model)​



Once the campaign is launched, the number of backers attracted so far becomes the primary determinant of success​

****

# Bibliography:

* <https://www.kickstarter.com/>