Tua **Wongsangaroonsri,** Elaine **Ren**, Kemjika **Ananaba**

Background and Motivation

The Kickstarter platform is a funding platform for creative projects. The projects are in different categories such as films, games, and music to art, design, and technology. The Kickstarter platform is full of ambitious, innovative, and imaginative ideas that are brought to life through the direct support of others through funding. Kickstarter, the hosting organization make its revenue by charging the project owners with successful projects 10% of the project goal.

Project backers make pledges to projects in order to help support the creative process. As a profitable organization, Kickstarter would like to predict what their yearly revenue would be. With millions of projects active in the Kickstarter platform at any given period, the business question is **"How can Kickstarter optimize their expected revenue earned from the projects?"**. In other words, our question needs to study "How many projects are predicted to be successfully completed?".

Data Description

Data was collected from the kick-starter website (https://www.kickstarter.com) about individual projects that have been completed and that were not successfully completed. The dataset contains 378,661 observations on all projects hosted on Kickstarter mainly between 2009 and 2017. The data-set has 15 columns with the following names:

Variable Name	Description
ID	Internal Kickstarter id
name	Name of the project
category	The subcategory of the project. (159 categories)
main_category	The main category of the project. (15 categories)
currency	Currency used to support the project. (14 currencies)
deadline	Deadline for crowdfunding.
goal	Goal amount in project currency.
launched	Date launched the project.
pledged	Amount pledged by "crowd".
state	Current condition of the project. (failed, canceled, successful, live, undefined, suspended)
backers	Number of backers.

country	Country pledged from. (23 countries)
usd pledged	Pledged amount in USD.
usd_pledged_real	Pledged amount in USD.
usd_goal_real	Goal amount in USD.

Basic Data Cleaning

To get the dataset into a format suitable for applying machine learning models, the data was cleaned based on potential issues identified as follows. Finally, the dataset contains 328,083 observations and 17 columns.

Issues with Data	Data Cleaning Solution
Some categories in column *currency* has very few observations	Eliminate currency with observations that are less than 40 (i.e. JPY)
There are columns with NA values.	Eliminate NA values using sklearn *SimpleImputer*. - Numerical values are replaced with column average. - Categorical values are replaced with the most frequent occurrence in the column.
Project *state* column has 6 project status, including Success and Failed.	Only keep rows with either Success or Failed in the project status column
Extreme values in the goal column (**Fig 1.1&1.2**).	Remove 1% observations of the extreme values
Fail to capture time effect on state prediction	Generate two new variables *year* and *daysavailable* - *year*: The project launched year - *daysavailable*: Length of the project starting from *launched* to *deadline*

Exploratory Data Analysis

Various features of the dataset were explored, in terms of project number, goal and success rates. **Fig 2** to **Fig 6** show the differences in some of the features between successful and failed projects.

Key Takeaways

1. Hong Kong, USA and the United Kingdom are home to a greater proportion of successful projects based on the number of successful projects.
2. The top three project categories launched on Kickstarter are technology, design and foods on the grounds of Average project goal.
3. The top project three categories launched on Kickstarter are theatre, dance and comics based on the number of successful projects. This is probably at least partly due to their relatively small funding goals as noted in **Fig 5**, as when compared based on the goal amount they are not as successful. Therefore, using the number of successful per category means that projects with smaller goals tend to be more successful.
4. The worst performing categories are craft when we consider the number of successful projects and the project goal amount.

Business Question and Statistical Question

Our business question is **'How can Kickstarter optimize expected revenue generation from projects?'**. To answer this problem, the first step would be to quantify the expected revenue that Kickstarter can generate on its currently on-going projects by predicting how many projects are predicted to be successfully completed. A huge variety of factors contribute to the success or failure of a project, such as the project category, external economic condition and the social economic impact brought by the project. The statistical questions that can be answered using the dataset obtained are as follows:

1. What will be the completion status of a kick-start project predicted by its features displayed on the kick-starter platform? And, what is its probability to be successful/failed?
2. What project features on the kick-starter platform contribute most to the success of a project?

First, as long as Kickstarter companies are able to predict the final state of a project based on its features, they can better manage different kinds of projects and identify profitable ones. For example, if projects in the theatre category are more likely to be successful, the platform could help the project and increase their online occurrence. Also, with success/failure probability, most contributable features of a kick-started project that are more inclined to be successful can be captured. With this information, Kickstarter can design some guidelines (i.e. how to set project goal) for potential poor performance projects to increase their success rate. Additionally, identifying these potentially successful projects would then allow Kick-starter to allocate visibility on the website to projects according to revenue generating potential.

However, due to data limitation, our questions only focus on features displayed on the kick-starter website and ignore other salient features that can be important in predicting the success status of the project such as global interest, popularity of the project. Besides, typically, projects with more backers are more likely to be successfully completed. Although variable *backers* have significant explaining power in final project status prediction, it actually penalizes the newly implemented projects to fail in the system because they usually have zero backer. Thus, our prediction model is a static one as the information seldom updates with a given period. This may lead to inaccurate prediction of the number of successful projects and the corresponding revenue.

Feature Pre-selection

To solve the statistical questions proposed, 5 relevant features are chosen from 17 columns before the feature preprocessing and model selection. The column *state* is also kept as it is our target for prediction. Reasons are as follows:

1. **usd_goal_real:** For a project to succeed, the amount it pledges should be very close or equal to the project goal. As shown during EDA, projects with high goals are shown to have less probability to be successful. Thus, it turns out to be a useful indicator when predicting the project final state. In order to compare among projects supported by different currencies, we decide to use USD goal.
2. **daysavailable:** It can help measure the opportunity of its online occurrence and chance of being supported. Generally, the longer this length is, the more online appearance it will have. It means the project can have higher probabilities to be supported and successful.
3. **main_category***:* It stands for market interests to some extent. According to EDA, there exists some types of projects, such as theatre, that are popular all the time, and craft, that are just the opposite. As different category projects show a distinct difference between success rate, it is added to the features. Here, as *category* has too many kinds, we choose to use the corresponding *main_category*.
4. **currency**: From EDA, the project success rate in different countries is different. As some countries   in Europe that have low frequencies in our dataset share the same currency, we use *currency* to group them together as European Union countries instead of *country*.
5. **year**: As mentioned before, variable *year* helps capture the time effect on prediction of project status.

The remaining variables will be dropped. Among which, our reasoning to drop some of the potentially significant variables are as follow:

1. **backers:** *backers* correlates with the success of a project, we chose not to include this feature in the model.
2. **usd_pledged_real:** Similarly, *usd_pledged_real* features is in direct correlation with a project's success, especially in the case of our data set since having *pledged* less than *goal* would mean that the project failed.

To test whether our stipulation is valid, we will run a model which includes these two dropped features in the following section.

Learning workflow and model

Feature Preprocessing

The ultimate goal of this project was to create a machine learning model that could predict whether a project in the Kickstarter platform would succeed or fail. In order to prepare the data for the models, the following steps are taken:

1. One-hot encoding categorical variables *main_category*, *year* and *currency* columns.
2. Separating the data into the dependent target variable 'y' which is the *state* column (project success or failure) and the independent features 'X' which are variables *main_category*, *year*, *currency*, *usd_goal_real* and *daysavailable*.
3. Transforming the features in X so that they are all on the same scale.

Finally, the data was separated into a training (80%) and test (20%) set to learn the model performance later.

Model Selection

To start the model selection process, we utilize the Pipeline and GridsearchCV to select the best estimator with particular levels of hyperparameters. Due to the size of dataframe, the computation time is prohibitive of finely searching through models' parameter. Instead, we will first search through a wide interval of hyperparameters for each model. Once the best model is determined, we will fine-tune the level of hyperparameter in a later step. Best model is selected based on the lowest validation error from 5-fold cross validation.
The model and hyperparameters under-consideration are below:

Model	Parameter 1	Parameter 2
Logistic Regression	C: 0.1, 1, 10	NA
Random Forest Classifier	Max Depth: None, 10, 20	Number of Trees: 10, 50, 100
K-Nearest Neighbor	Number of Neighbors: 1, 10, 20	NA

Running through the GridsearchCV, we've arrived at our best estimator as Random Forest with Max Depth 20 and 100 Trees which yield the lowest validation error of 0.33531(**Fig 7**). However, we do note that most other models also produce errors within this region (**Fig 8**). Nevertheless, we would still proceed with choosing this random forest model as our best estimator under the current selection due to following reasons:

1. In addition to classification, Random Forest Classifier offers a probabilistic prediction which can be utilized to supplement business insight (Whereas KNN doesn't)
2. Random Forest can be tuned to be robust against overfitting using a high number of trees in the forest.

To test our earlier stipulation to not include *backers* and *usd_pledged_real*, we attempted to fit a random forest model with both dropped variables included. This leads to a significant drop in validation error rate to 0.00277 (**Fig 9**). However, as seen from (**Fig 10**), the model becomes almost entirely dependent on backers and *usd_pledged_real* for classification. Since our data set only consists of completed projects which are either successful or failed, having our model learning this feature means that we would be systematically under-predicting the success probability of all newly launched projects on Kick-starter. Hence, we are justified in our assumption to not include these two features in the mode.

Features Selection

To further fine-tune the model, we consider the best iteration of random forest and the check the importance of features (**Fig 11**). We noted that three features (*main_category, usd_goal_real, daysavailable*) already contribute 88.3858 percent of total importance in classification. We consider dropping two features (*year* and *currency*), which only contributes only marginally, which will reduce the dimensionality of our dataframe by more than half from (328083 rows × 36 columns) to (328083 rows × 16 columns). To test whether the tradeoff makes sense in terms of model accuracy, we utilize the best model (Random Forest with 20 max-depth and 100 trees) and refit the model with our *year* and *currency* features.

The new model has a validation error of 0.34978 which is 1.139% higher than the model with all features included. We consider this slight increase in error to be a good trade-off for the significant reduction in dimensionality and computation time of the simplified model. Additionally, there is a theoretical argument for dropping *years* from the feature. Since we are using dummy variables to control for year effect, it is possible for us to know the year effect of the current year and, hence, is not useful for forecasting. Otherwise, using the most recent year effect to conduct our forecast requires that the underlying trend from the previous year has to continue, which is a very strong assumption to make and is unlikely to be true in most scenarios. Hence, we would proceed with the model with 3 features (*main_category, usd_goal_real, daysavailable)* for the next part of the process.

Hyperparameter Tuning

Since we conducted GridSearchCV on a wide interval of data, we lastly tune the hyperparameter of the model, specifically the max-depth of random forest. As shown from the error associated with different types of model under consideration (**Fig 12**), varying the number of trees between 50 and 100 doesn't seem to have a large effect on the model error. Hence, we decide to utilize 50 trees in the final model to save computation time and would only be tuning the final model based on max-depth. As shown in **Fig 13** and **Fig 14**, validation error for random forest using final set of training and testing data for when max-depth is 17 corresponding validation error of 0.335422.
In the end, we choose the random forest classifier with the max_depth as 17 and number of trees as 50 to solve the questions proposed. After testing on our test data, the test error equals 0.343752, which is quite close to the training and cross-validation error. This indicates our model has a good performance for predicting the project state.

Best Model	Max Depth	Number of Tree	Test Error
Random Forest	17	50	0.343752

Conclusion and Recommendations

With kick-starter's all-or-nothing funding system, the fund-givers (backers) are only charged for their contribution once a project is successfully funded. At which point, kick-starter's fee will be applied at 10% of the total funds raised for the project. Under this revenue model, kick-starter's business operation is dependent on the success of Kickstarter projects. Using the random forest classifier model, our best model can generate probability prediction of a project's likelihood to be successfully funded or not based on the project's Goal Amount, Project Duration, and Category of the Project.

Project Benefits and future implementation

The benefit of our model for Kick-starter business:

1. To Allow Accurate Forecast of Expected Future Revenue Stream from Newly Launched Projects
2. To Allow Search Ranking and Visibility Optimization for Project with High Completion Probability
3. To Recommend Project Owners Appropriate Project Duration and Goal to be Successful

To further illustrate the benefits of this model, we obtained the data of four newly launched projects to test their probability of success and failure. These four projects were launched on Feb 13, 2020, and they had less than 10 backers except for the last one. Results are shown in the following table:

Project	Usd_goal_real	Duration	Category	Failure Prob.	Success Prob.	Expected Revenue
All Terrain Electric Bike by Mountains To Sea Electric Bikes (https://www.kickstarter.com/projects/m2sbikes/all-terrain-electric-bike?ref=discovery_newest)	11985	11	design	0.69	0.31	374.05
DailyGrind - Ask for advice, anonymously (https://www.kickstarter.com/projects/dailygrind/dailygrind-ask-for-advice-anonymously?ref=discovery_newest)	16825	30	technology	0.65	0.35	587.91

Domnique's BBQ & Catering (https://www.kickstarter.com/projects/dominique-bbq1/domniques-bbq-and-catering?ref=discovery_newest)	14981	59	food	0.88	0.12	173.55
WÜDWARE: Magnetic Wall Hook (https://www.kickstarter.com/projects/wudware/wudware-magnetic-wall-hook?ref=discovery_newest)	214	29	design	0.28	0.72	15.48

From the table, it seems that the first three projects are more likely to fail as their related probability is about 70%-80%, while the last one has the opposite situation. With the probability obtained, we can also calculate the expected revenue of kick-starter, which is the product of project success probability rate and a percentage of project fund goal.

As the model forecasts expected revenue generated based on how likely the active projects in the platform are going to be successful, it not only solves the business question but also helps kick-starter to evaluate projects from initiation and take some action to increase their revenue further. For example, these projects could be ranked on an expected revenue basis, allowing Kick-starter to allocate website space and order search ranking by relative value of each project.

Within this context, our model should be bundled up as a part of computational pipeline in Kickstarter website to optimize the potential revenue generated from projects. Besides, based on the past experience, kickstarter can find out the most appropriate goal and project duration and set them as guidelines for those newcomers in each category to increase the probability of a project to be successful.

Project Limitations

One limitation of our model is that we have a limited feature that can indicate a project-specific quality and, hence, our probability forecast solely dependent on numerical quality of project. To improve the existing models, collecting more features regarding the project attributes such as number of stretch goals, whether there is video and demo of the products, and the comments of the backers will allow us to further fine-tune the model.

Another limitation is that our model can only generate one static forecast of the project success probability at its conception, but cannot update those probabilities as the projects are starting to receive backers and pledges. If we have access to a more real time updating data-set as a part of the pipeline in the Kick-starter website, we can consider incorporating a more dynamic aspect to the model. By adding updating features such as *backers, pledges remaining*, and *days remaining*, we can create an updating forecast model that reflects the changes that is happening in the website for a more accurate revenue forecast and finer optimization of search ranking for the projects.

Appendix

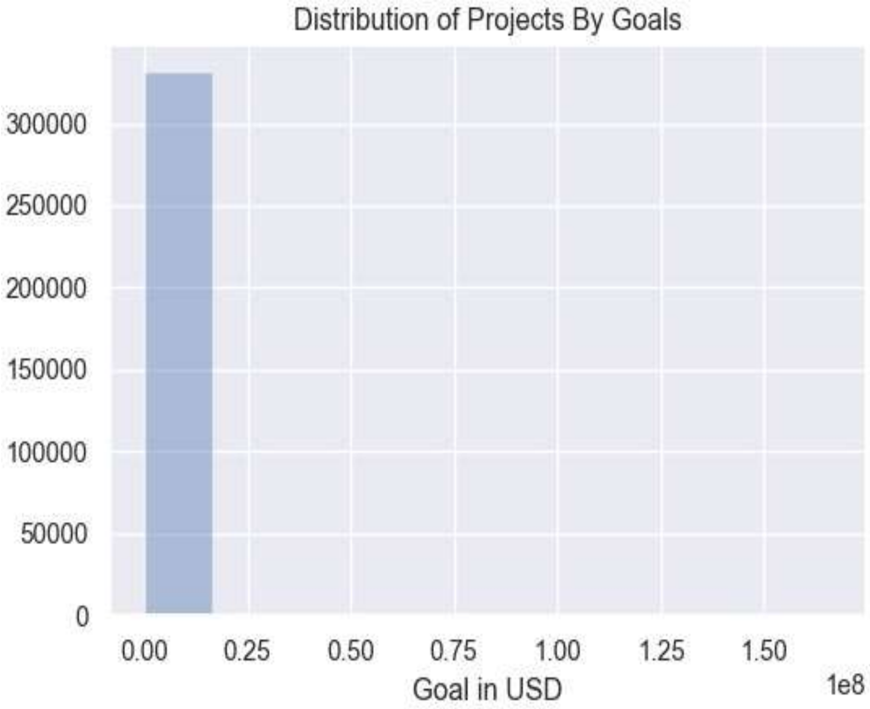


Fig 1.1 The Distribution of Projects By Goals

Fig 1.1 shows the distribution of projects by goals in USD, which is unimodal and right skewed. This indicates there exists some extreme values in the dataset. To solve this problem, the top 1% values in the goal column are dropped. After that, its distribution shown in **Fig 1.2** becomes much more normal and acceptable.

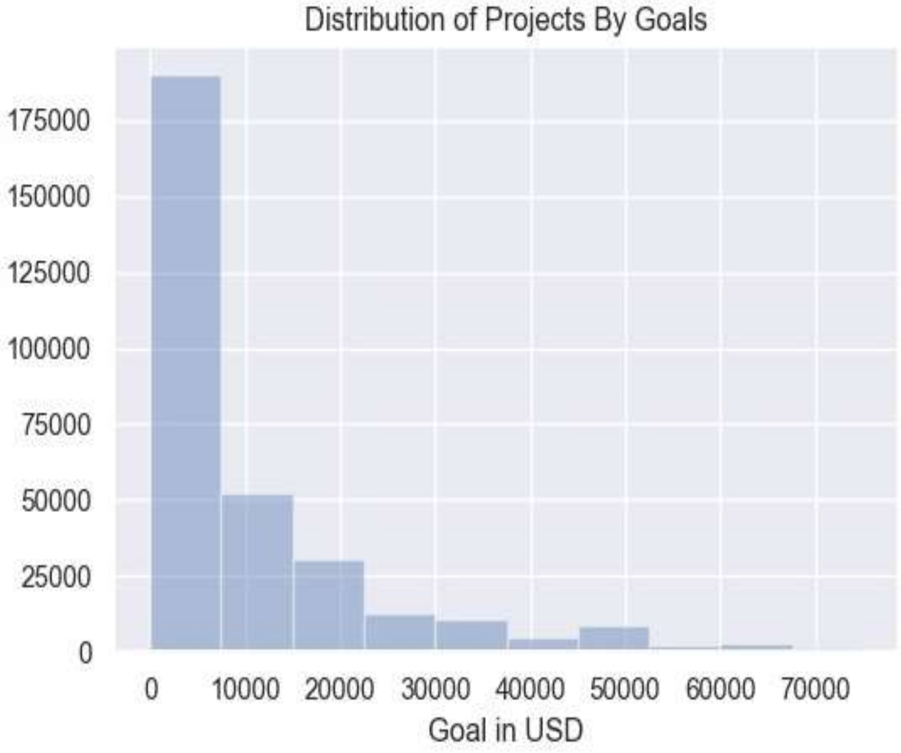


Fig 1.2 The Distribution of Projects By Goals After Dropping Top 1% Values

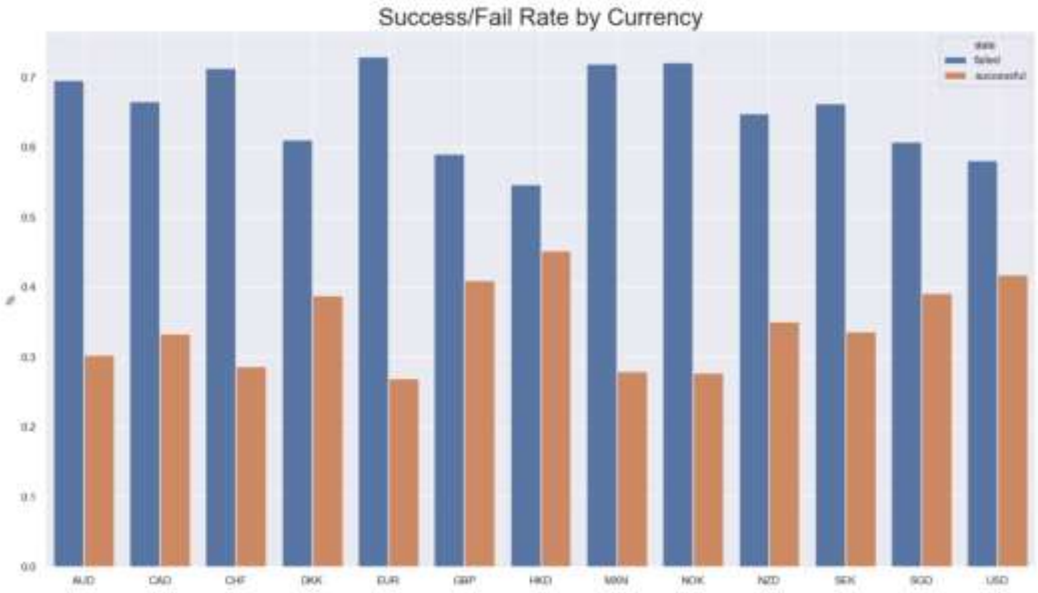


Fig 2: Success and Fail Rate by Currency

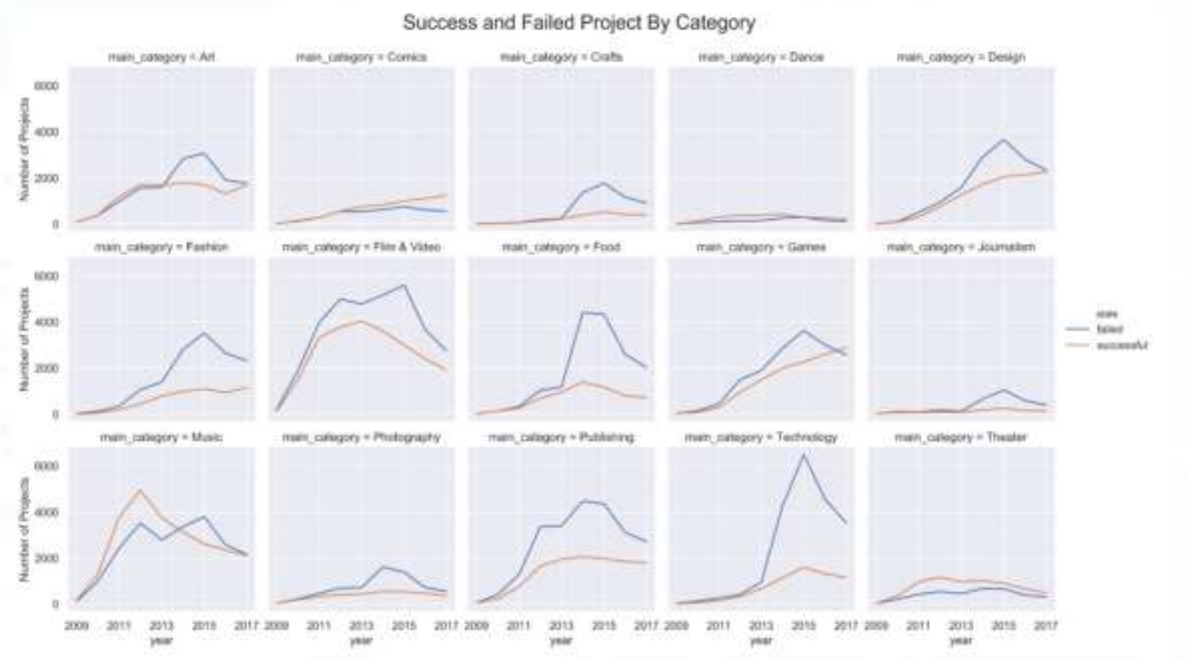


Fig 3: Success and Failed projects by Category

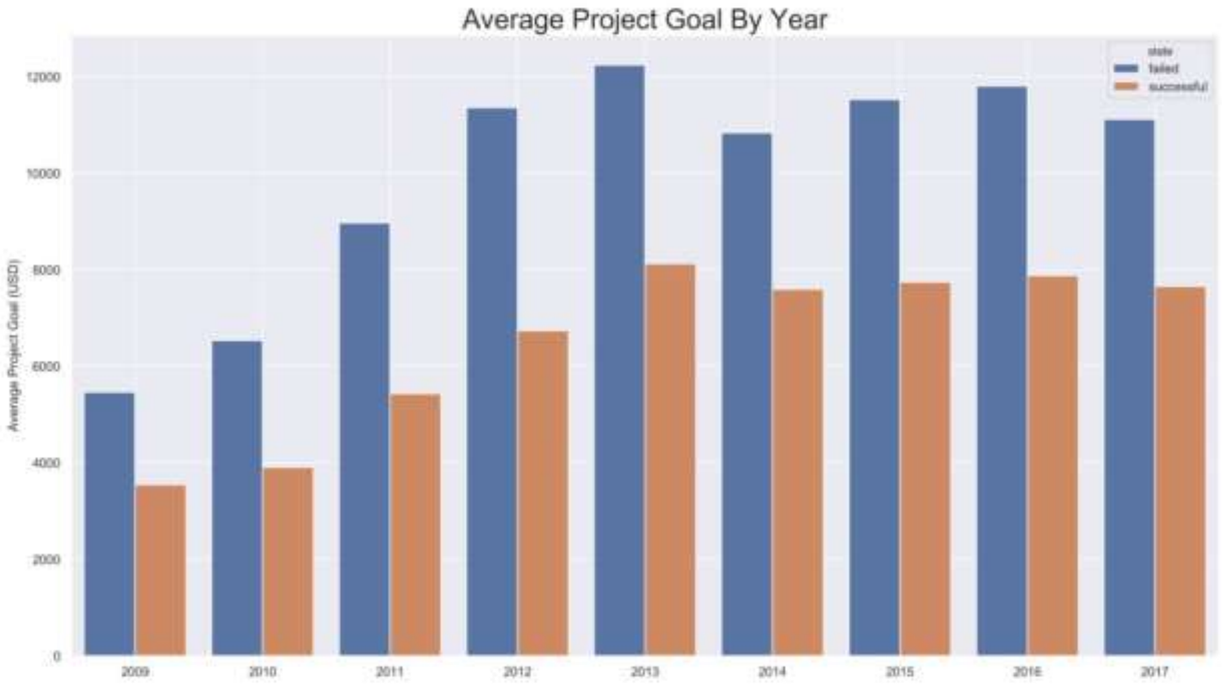


Fig 4: Mean Goal Amount for Success/Fail Projects by year

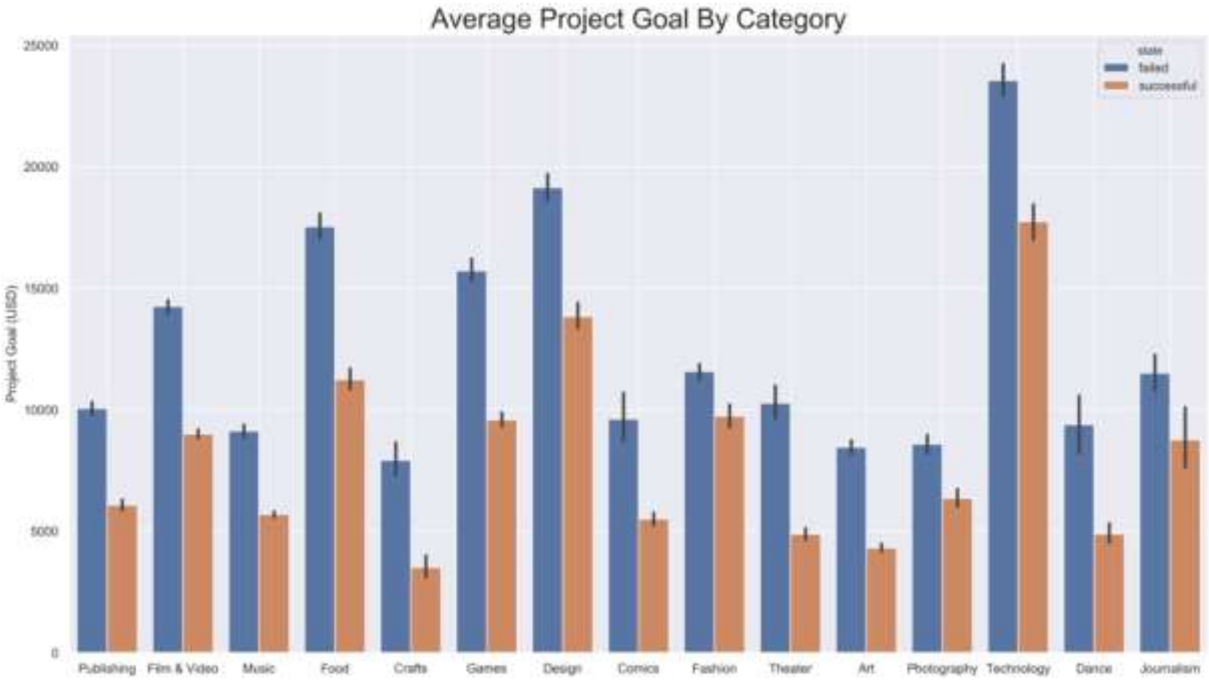


Fig 5: Average project goal per Category

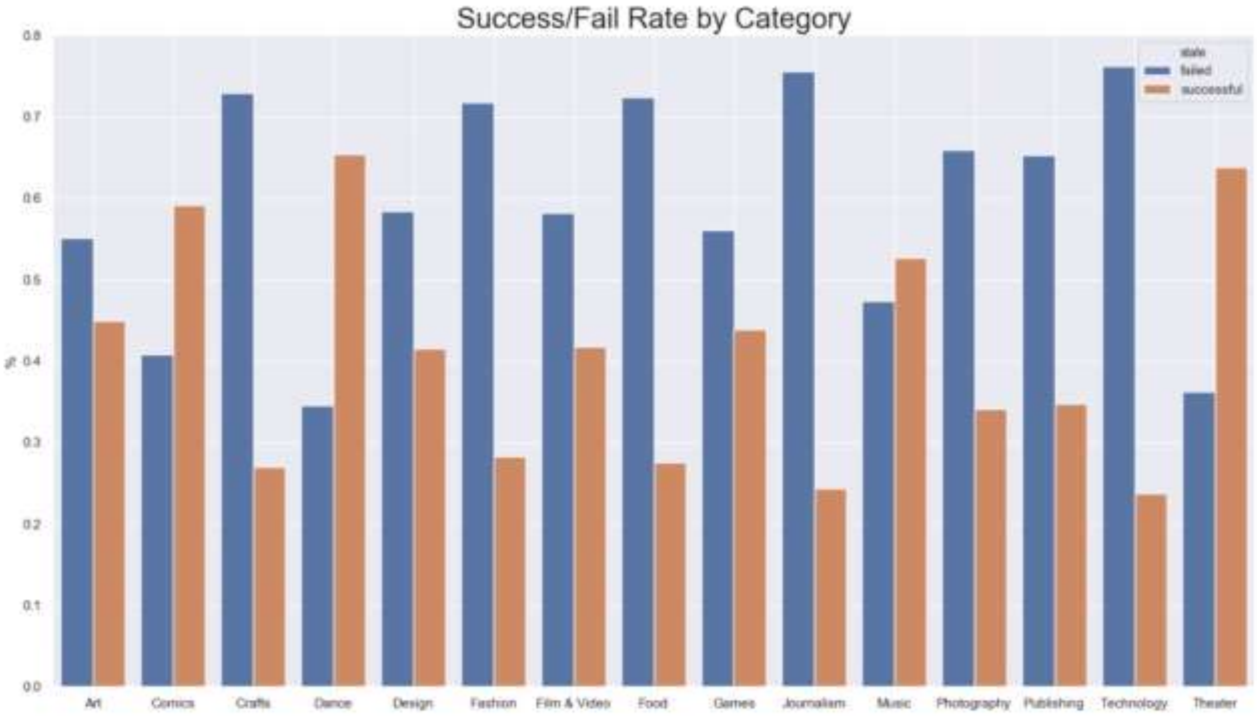
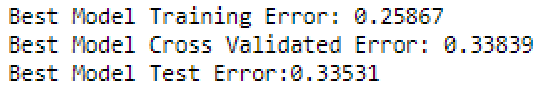
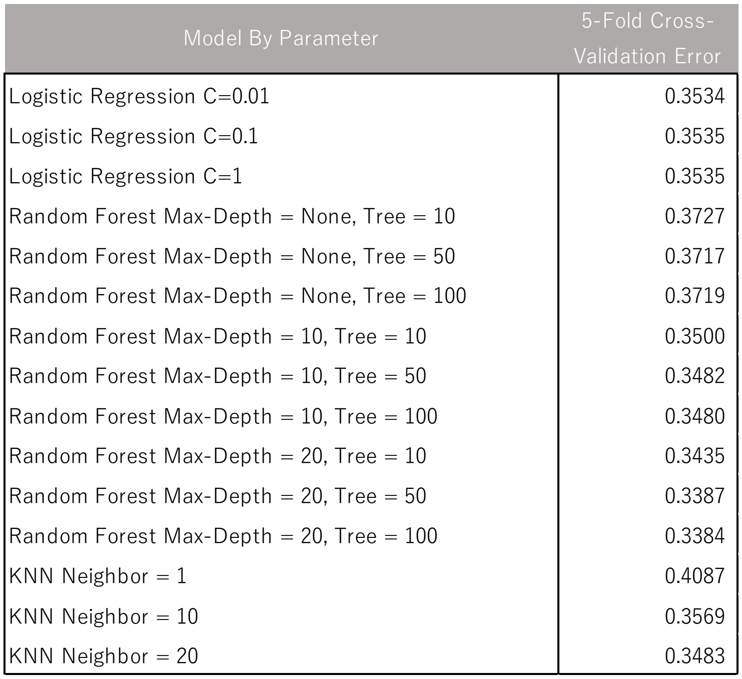


Fig 6: Success Rate by Category Number



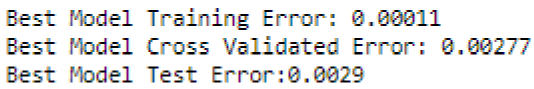
```
Best Model Training Error: 0.25867
Best Model Cross Validated Error: 0.33839
Best Model Test Error:0.33531
```

Fig 7: Error Rate for Best Model (Random Forest 20 max-depth and 100 trees) with 5 Features



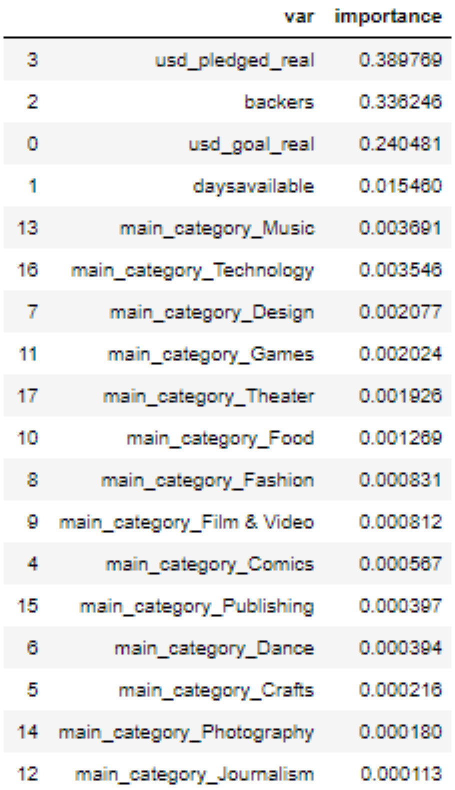
Model By Parameter	5-Fold Cross-Validation Error
Logistic Regression C=0.01	0.3534
Logistic Regression C=0.1	0.3535
Logistic Regression C=1	0.3535
Random Forest Max-Depth = None, Tree = 10	0.3727
Random Forest Max-Depth = None, Tree = 50	0.3717
Random Forest Max-Depth = None, Tree = 100	0.3719
Random Forest Max-Depth = 10, Tree = 10	0.3500
Random Forest Max-Depth = 10, Tree = 50	0.3482
Random Forest Max-Depth = 10, Tree = 100	0.3480
Random Forest Max-Depth = 20, Tree = 10	0.3435
Random Forest Max-Depth = 20, Tree = 50	0.3387
Random Forest Max-Depth = 20, Tree = 100	0.3384
KNN Neighbor = 1	0.4087
KNN Neighbor = 10	0.3569
KNN Neighbor = 20	0.3483

Fig 8: Validation Error by Model Searched



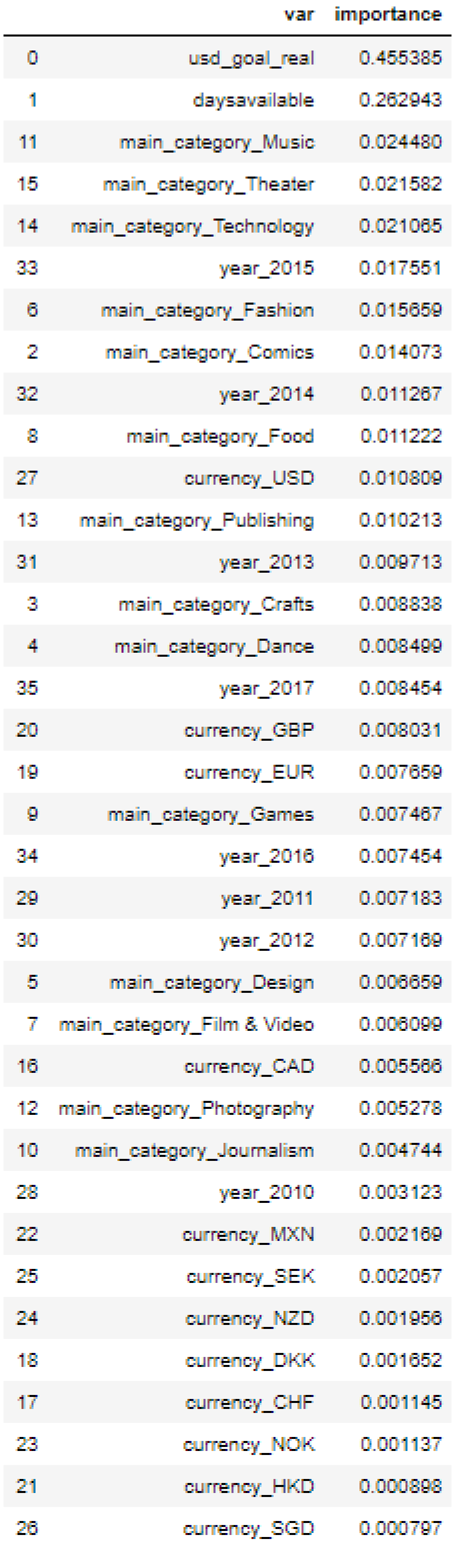
```
Best Model Training Error: 0.00011
Best Model Cross Validated Error: 0.00277
Best Model Test Error:0.0029
```

Fig 9: Error Rate for Random Forest 20 max-depth and 100 trees with *backer* and *usd_pledged_real*



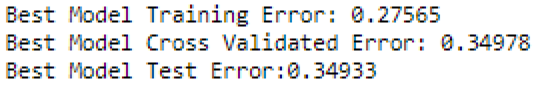
	var	importance
3	usd_pledged_real	0.389769
2	backers	0.336246
0	usd_goal_real	0.240481
1	daysavailable	0.015460
13	main_category_Music	0.003691
16	main_category_Technology	0.003546
7	main_category_Design	0.002077
11	main_category_Games	0.002024
17	main_category_Theater	0.001926
10	main_category_Food	0.001269
8	main_category_Fashion	0.000831
9	main_category_Film & Video	0.000812
4	main_category_Comics	0.000567
15	main_category_Publishing	0.000397
6	main_category_Dance	0.000394
5	main_category_Crafts	0.000216
14	main_category_Photography	0.000180
12	main_category_Journalism	0.000113

Fig 10: Feature Importance for Random Forest 20 max-depth and 100 trees with *backer* and *usd_pledged_real*



	var	importance
0	usd_goal_real	0.455385
1	daysavailable	0.262943
11	main_category_Music	0.024480
15	main_category_Theater	0.021582
14	main_category_Technology	0.021065
33	year_2015	0.017551
6	main_category_Fashion	0.015659
2	main_category_Comics	0.014073
32	year_2014	0.011267
8	main_category_Food	0.011222
27	currency_USD	0.010809
13	main_category_Publishing	0.010213
31	year_2013	0.009713
3	main_category_Crafts	0.008838
4	main_category_Dance	0.008499
35	year_2017	0.008454
20	currency_GBP	0.008031
19	currency_EUR	0.007659
9	main_category_Games	0.007467
34	year_2016	0.007454
29	year_2011	0.007183
30	year_2012	0.007169
5	main_category_Design	0.006659
7	main_category_Film & Video	0.006099
16	currency_CAD	0.005566
12	main_category_Photography	0.005278
10	main_category_Journalism	0.004744
28	year_2010	0.003123
22	currency_MXN	0.002169
25	currency_SEK	0.002057
24	currency_NZD	0.001956
18	currency_DKK	0.001652
17	currency_CHF	0.001145
23	currency_NOK	0.001137
21	currency_HKD	0.000898
26	currency_SGD	0.000797

Fig 11: Feature Importance from Random Forest Classifier with 20 Max-Depth and 100 Trees



```
Best Model Training Error: 0.27565
Best Model Cross Validated Error: 0.34978
Best Model Test Error:0.34933
```

Fig 12: Error Rate for Best Model (Random Forest 20 max-depth and 100 trees) with 3 Features

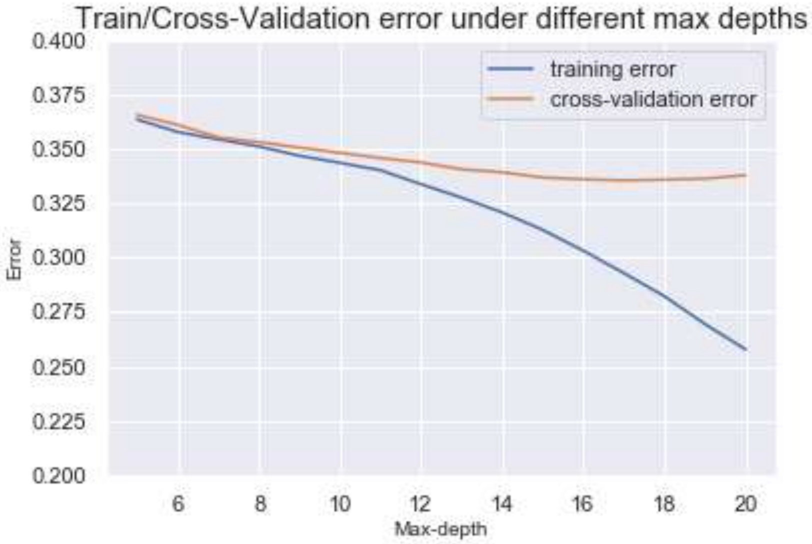
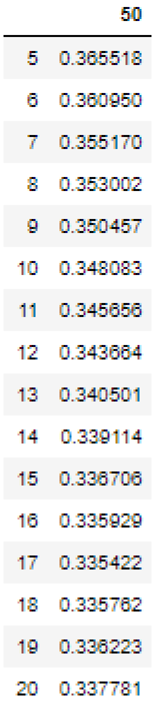


Fig 13: Training Error Vs Cross-Validation Error By Max-Depth for Random Forest with 50 Trees



	50
5	0.365518
6	0.360950
7	0.355170
8	0.353002
9	0.350457
10	0.348083
11	0.345656
12	0.343664
13	0.340501
14	0.339114
15	0.336706
16	0.335929
17	0.335422
18	0.335762
19	0.336223
20	0.337781

Fig 14: Validation Error By Max-Depth for Random Forest with 50 Trees