AIDI 1002 - PROJECT REPORT

Kickstarter Campaign Success Predictor

—

Terence Yu

Shail Patel

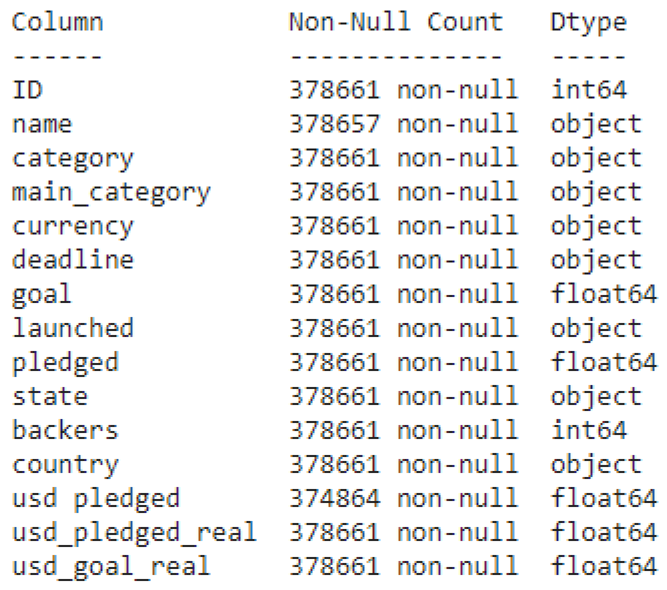
Sherap Gyaltsen

Jaspreet Singh Marwah

Github link: https://github.com/Zantorym/AIDI-1002-Project

I. EDA

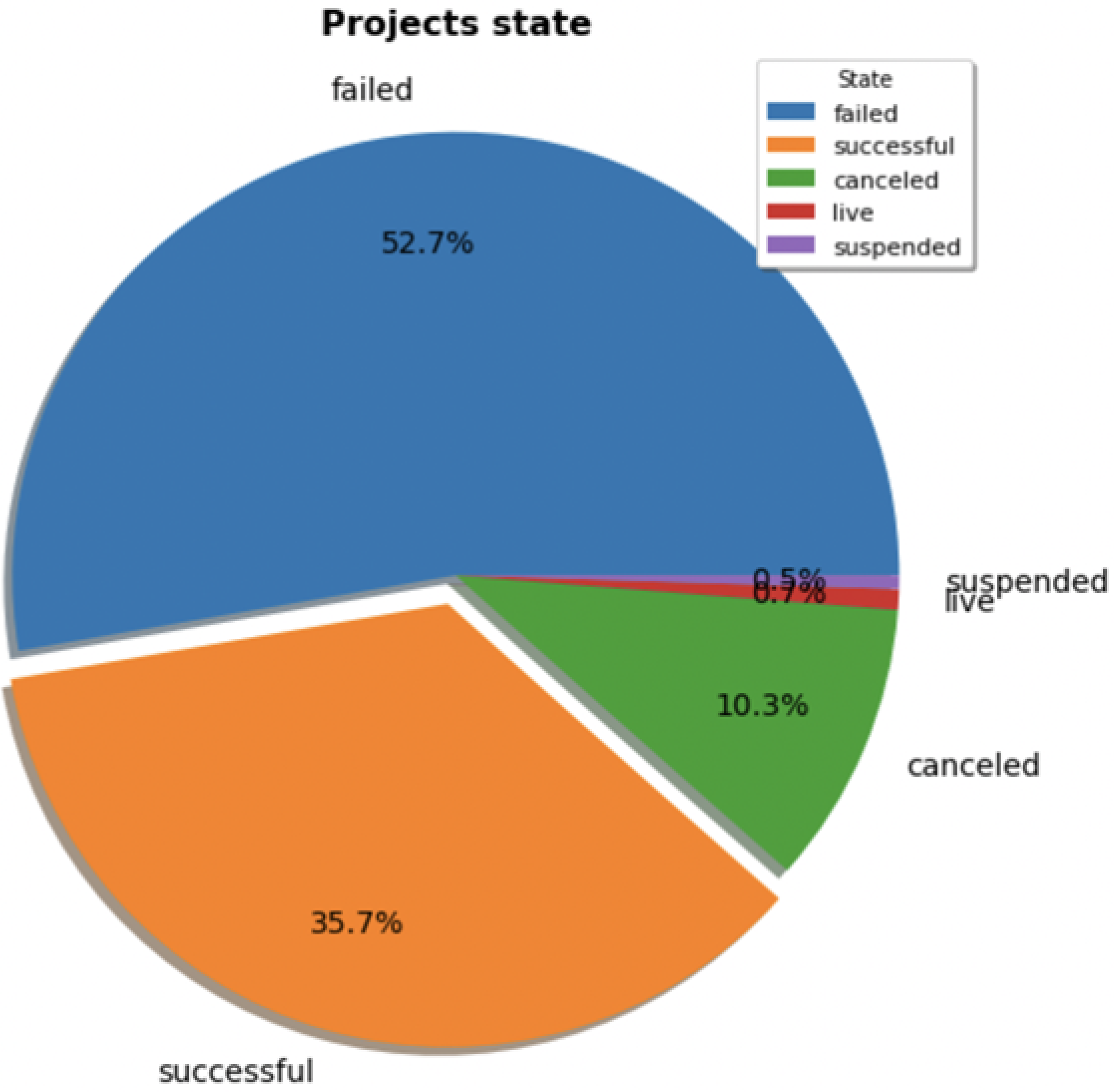
An initial glance at the dataset's features:



```
Column            Non-Null Count   Dtype
------            --------------   -----
ID                378661 non-null  int64
name              378657 non-null  object
category          378661 non-null  object
main_category     378661 non-null  object
currency          378661 non-null  object
deadline          378661 non-null  object
goal              378661 non-null  float64
launched          378661 non-null  object
pledged           378661 non-null  float64
state             378661 non-null  object
backers           378661 non-null  int64
country           378661 non-null  object
usd pledged       374864 non-null  float64
usd_pledged_real  378661 non-null  float64
usd_goal_real     378661 non-null  float64
```

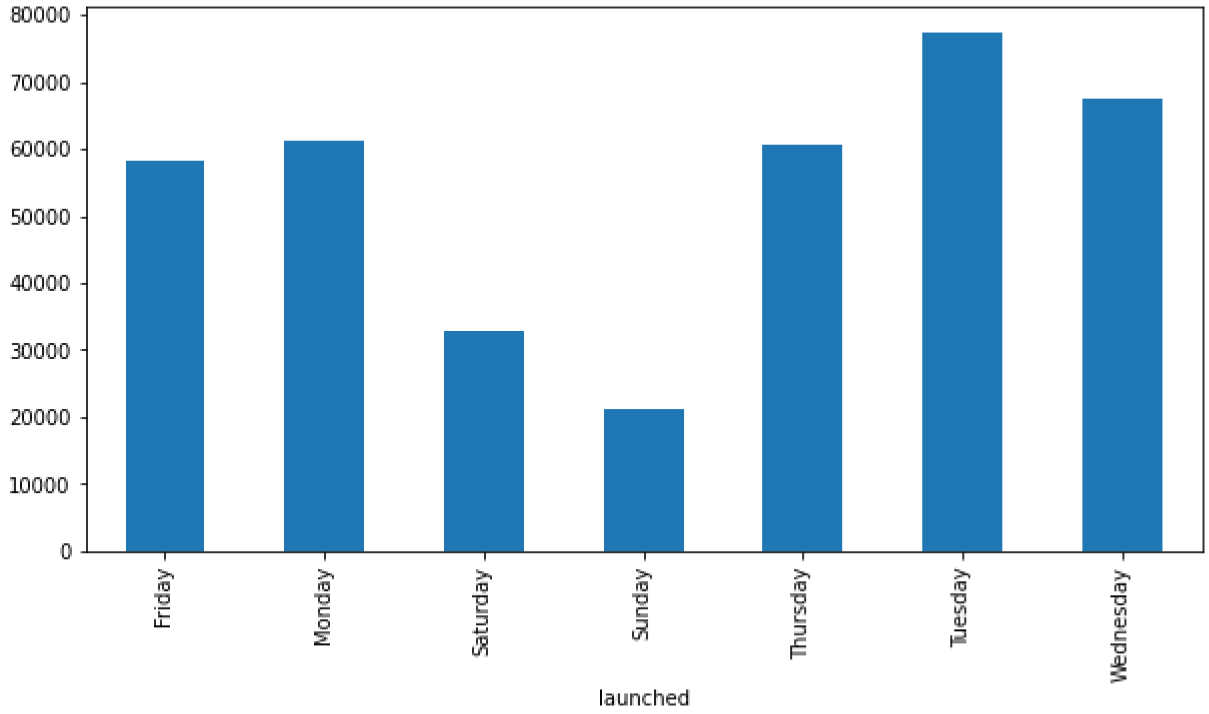
From this, we can see that only name and usd_pledged contain null values. Overall, the dataset is healthy.

The ‘state’ feature (which defines the status of the kickstarter campaign) contains 6 unique values; failed, canceled, successful, live, undefined, and suspended. Their distribution is as follows:

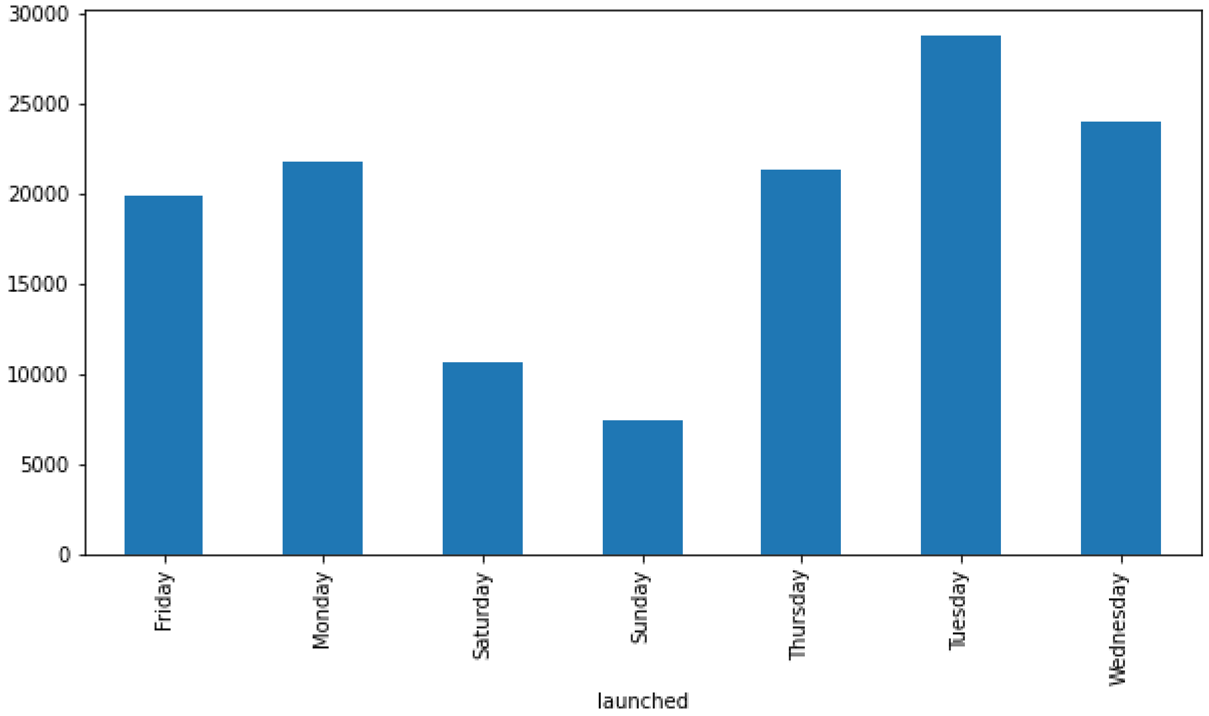


We see that 63% of campaigns are prone to failure. Only 35.7% of them achieve their funding goals.

Let us look at a distribution of what day kickstarter campaigns are launched on:

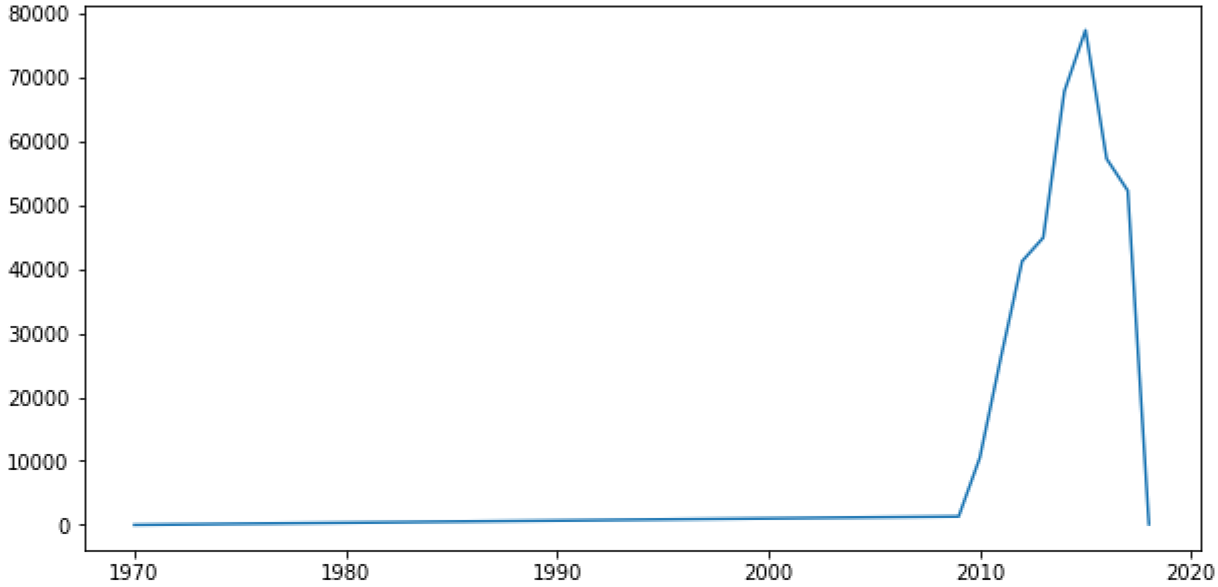


Now, let us look at a distribution of what day ***successful*** kickstarter campaigns are launched on:



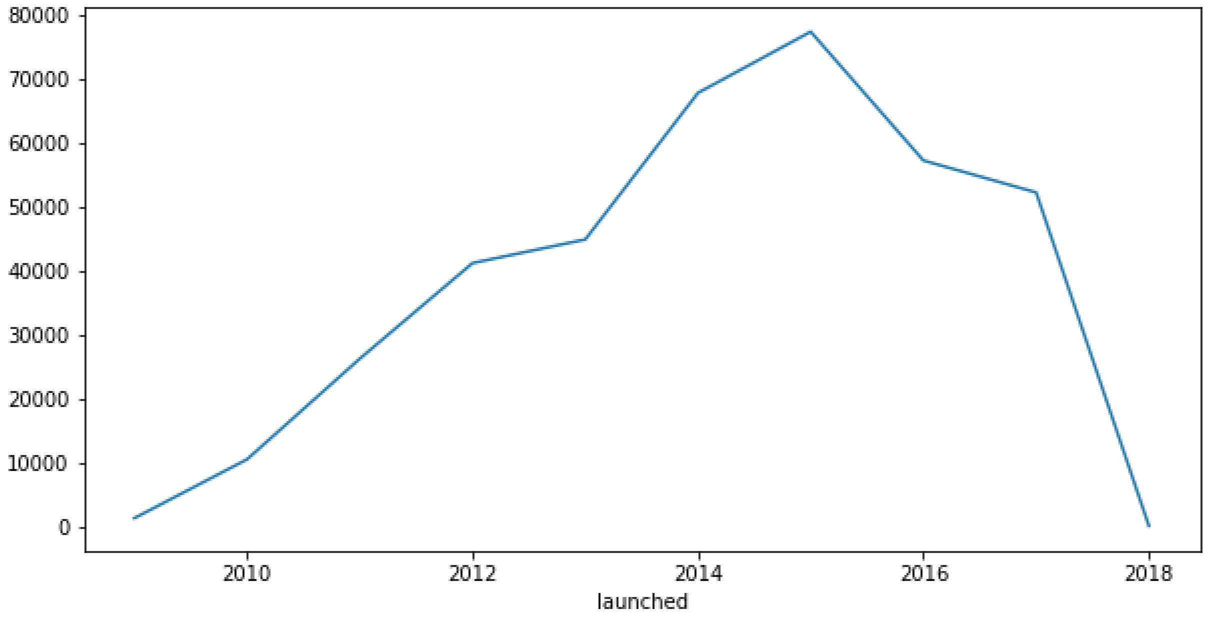
As we can see, the trends are exactly the same. This means that the day of launch has no bearing on the success of the campaign.

Let us observe the number of kickstarter campaigns launched per year:



Note: the Kickstarter platform launched in 2009, yet there are 7 entries for 1970. We can assume this is because January 1st, 1970 (a.k.a the Unix epoch) is considered to be (by convention) the default date for the start of time in computer systems. For entries where the start date was missing, this default date was likely added instead by the creators of the dataset.

Here is a cleaner look at the same plot, with the invalid entries removed:

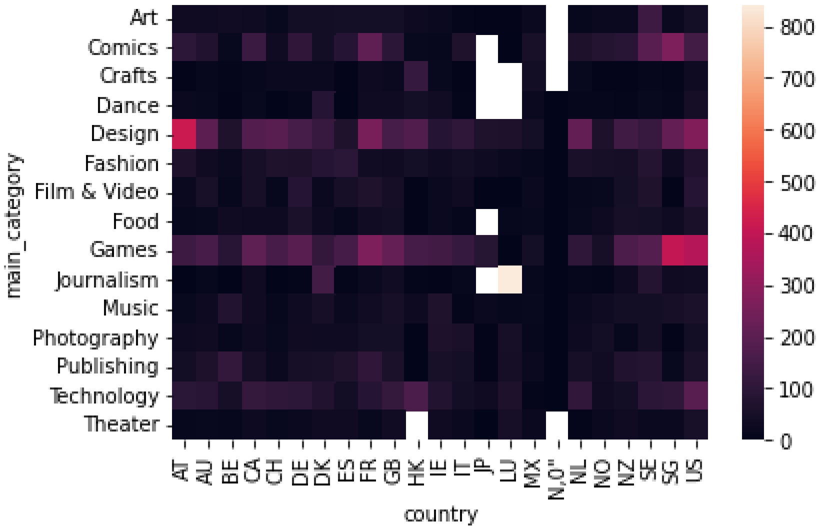


We see that 2015 was the most popular year for kickstarter campaigns, and their popularity has been reducing ever since. Note that the number of entries for 2018 is low because this dataset was released in the beginning of 2018.

On further exploring the dataset, we've uncovered some interesting facts:

- The average amount pledged to Kickstarters is 9,926 USD
- The average number of backers for a kickstarter is 112
- The average funding goal for a kickstarter is 53,155 USD

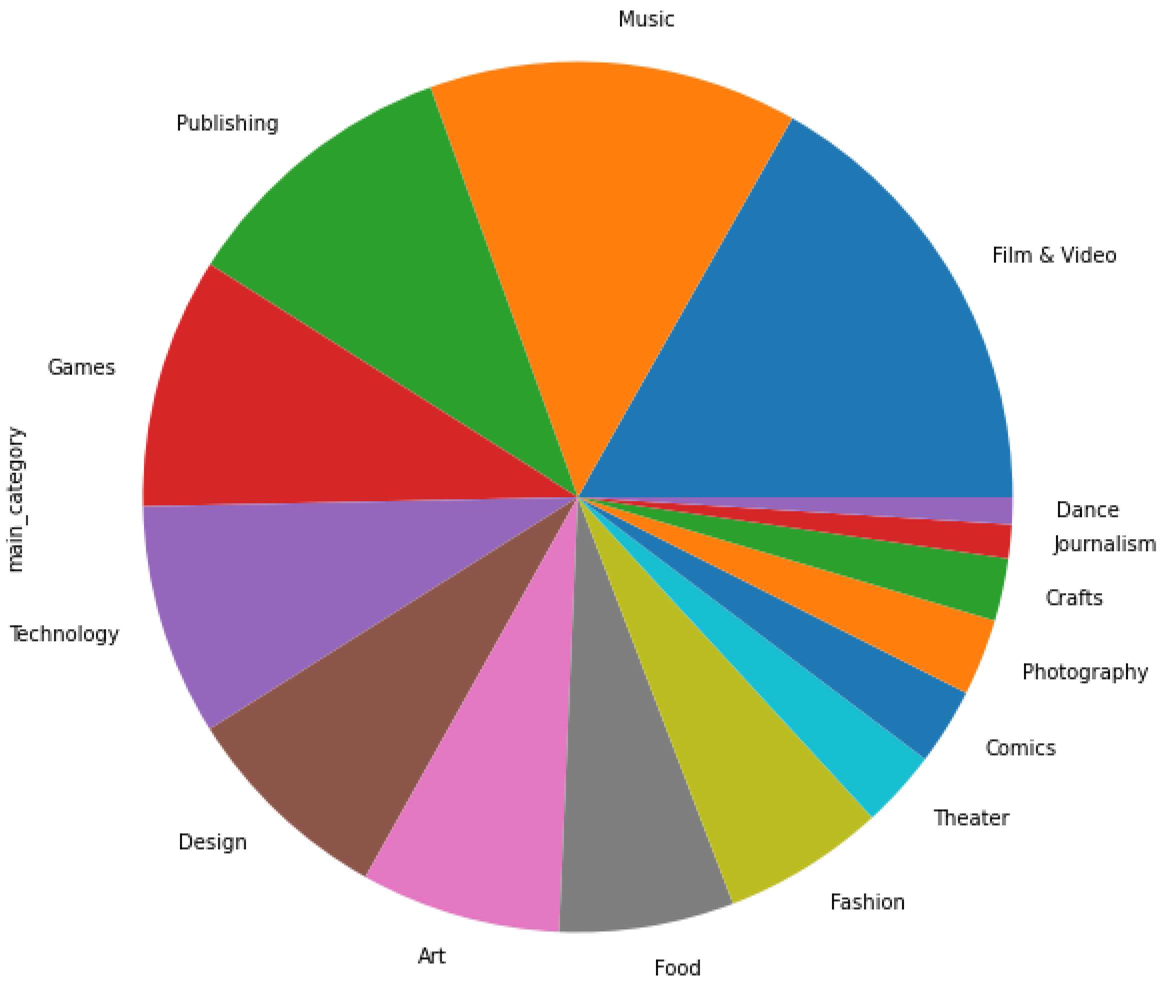
Let us observe the following heatmap:



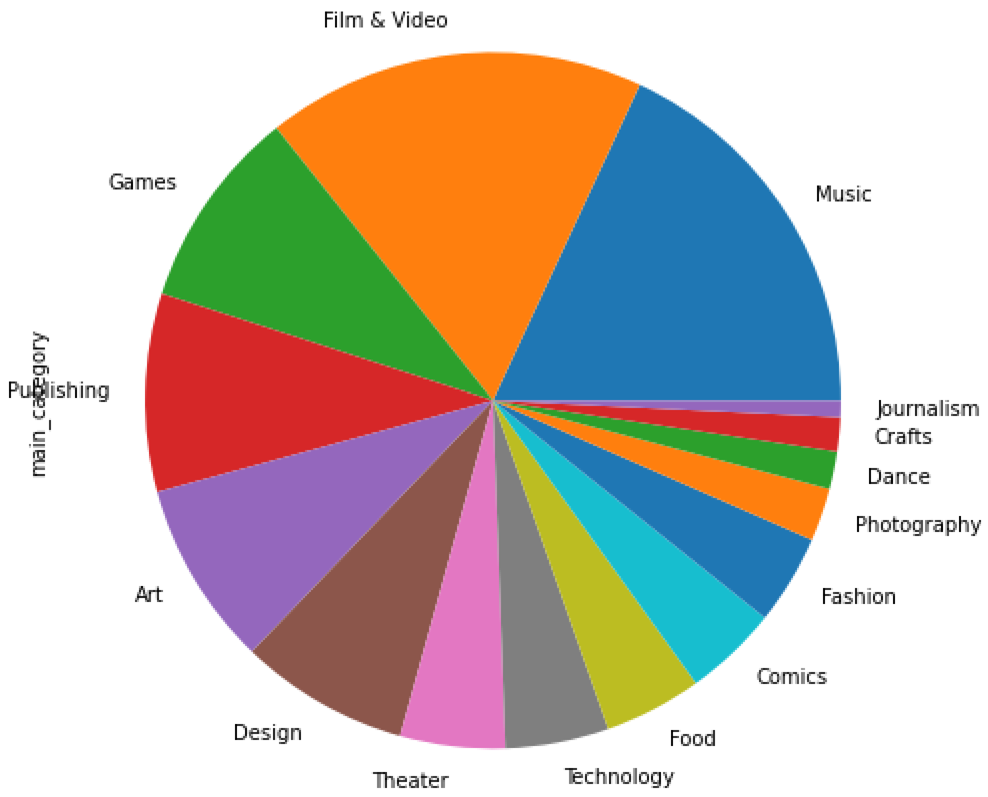
This heat map depicts which categories of kickstarters are most popular in each country (based on average number of backers per campaign in each category).

We see that in Austria design-based kickstarters are somewhat popular. Games-based kickstarters are mildly popular in Singapore and the US. In Hong Kong, theatre-based kickstarters are extremely popular. Journalism, dance, and crafts-based kickstarters are very popular in Japan and Luxembourg. Additionally, Food and comics-based kickstarters are also very popular in Japan.

The following is a distribution of kickstarters by category:

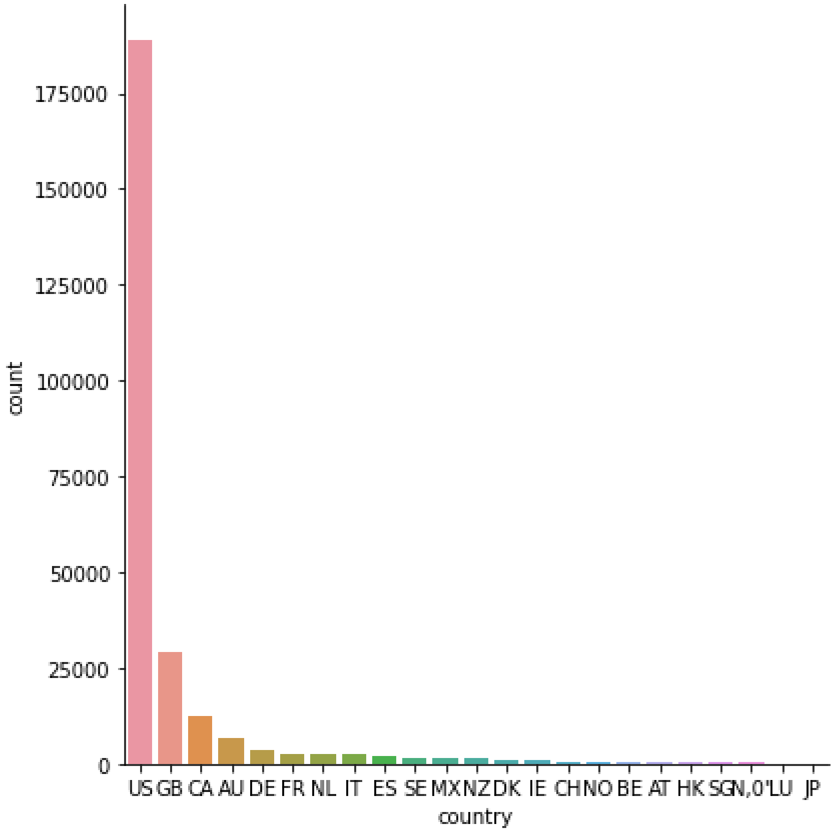


The following is a distribution of ***successful*** kickstarters by category:



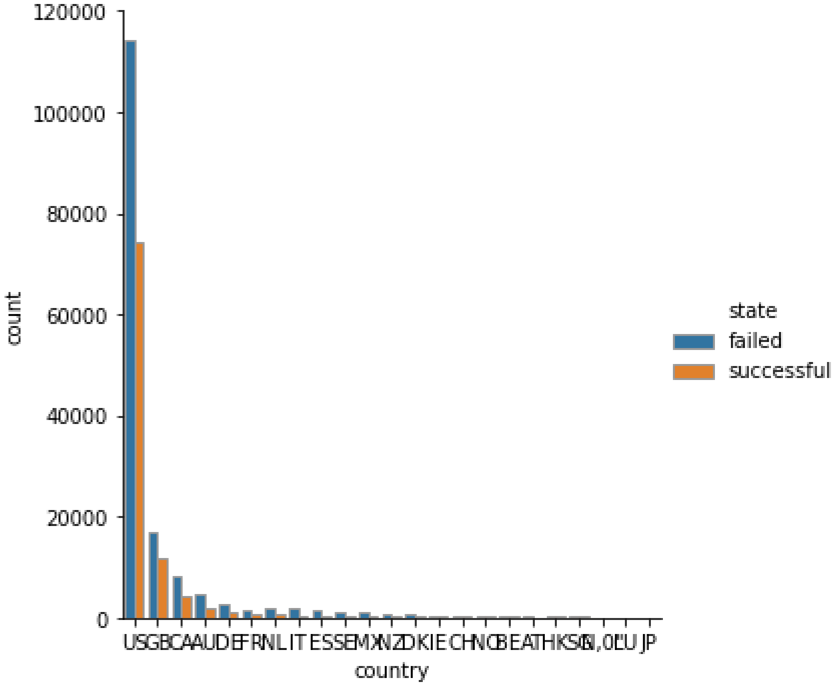
As we can see, category does not make a large difference in the success or failure of a project.

Now, let us look at which countries the kickstarter campaigns are published from:



As we can see, the US is the main source of kickstarter campaigns. This is followed by Great Britain, Canada and Australia.

Let us now see how many of these campaigns are successful.



All countries have a similar success to failure ratio. This indicates that the country in which the campaign is launched does not not affect the success of the campaign.

II. Modelling

Pre-Processing and Feature Engineering

The dataset contains features which are categorical data, that is, they contain distinct text values to describe the record. Some of these categorical features include country, category, main category, and currency. Since machine learning algorithms work best with numerical data, these "text" fields need to be encoding into a numerical vector in order to be processed. Converting these fields to numeric values is done using Scikit-Learn's LabelEncoder which assigns a distinct numerical value for each distinct text value.

Similarly, there are two fields that represent date and time information - launched and deadline. These fields need to be converted to proper datetime objects. Using these two fields, another field is added into the dataset which represents the number of days that the Kickstarter project will be running. This new field, named "project length" is obtained by subtracting the launch date from the deadline date. Upon generation of the project length field, the two original date fields are now redundant and are dropped for better performance.

Train and Test

In order to verify the accuracy of the machine learning algorithms used, the input dataset needs to be split into training and testing datasets. For this purpose, 80% of the input dataset is allocated to training and the remaining 20% is allocated for testing. Splitting is done using Scikit-Learn's train_test_split function which also has the added benefit of randomizing the order of the records.

The output classification labels for both training and testing is the state column. This column is separated from the training set and passed as the Y value (output label) for training. For testing, this field is separated and is only used to compare the predictions made by the algorithm to check if it is correct or not.

Modeling Algorithms

Five machine learning algorithms have been initially identified for the purpose of classifying the records. These are:

- Decision Tree Classifier
- Gradient Boosting Classifier
- K Nearest Neighbors Classifier
- Random Forest Classifier
- Gaussian Naive Bayes Classifier

All of these classifying algorithms are supported by Scikit-learn and follow it's fit-transform pattern which makes it easy to code and compare their results.

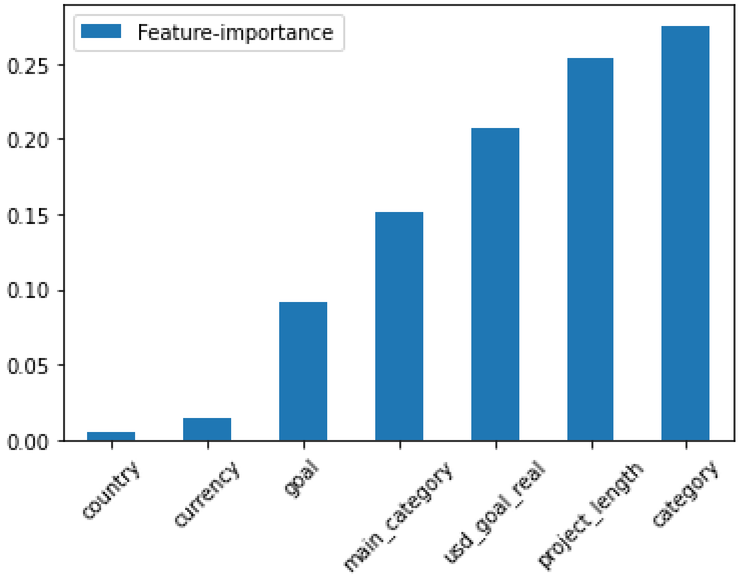
Initial Results

Training the different algorithms using the same allocated training dataset and using the test dataset to determine the output label provided a range of accuracy. The accuracy value is determined by getting the percentage of correct predictions against the total records being predicted. The following table summarizes the resulting accuracy values.

Classifier	Scikit-Learn Class	Resulting Accuracy
Decision Tree	DecisionTreeClassifier	61.57%
Gradient Boosting	GradientBoostingClassifier	68.72%
K Nearest Neighbors	KNeighborsClassifier	58.85%
Random Forest	RandomForestClassifier	65.88%
Gaussian Naive Bayes	GaussianNB	61.81%

As seen from the initial test results, it seems that classification using Gradient Boosting provides the best results, while K Nearest Neighbors performed the worst. However all algorithms provided results that are sub 70%, pushing for investigation and potential tuning.

Checking the features that contributed the most to the determination of the classification, it was found that the category and the length of the project are the most important. On the contrary, the project country provided the least weight in finding the classification. The following graph summarizes the findings.



Future Work for Improving Results

Some identified tasks that are planned to be done for improving the results are explained below.

First, instead of using LabelEncoder for converting text categories to numerical values, one hot encoding can be used instead. This will be done using Scikit-Learn's OneHotEncoder, which converts each distinct value as a new column which is either 0 or 1, with a 1 representing that value is used.

Another potential improvement is to scale the input dataset before training and testing. It has been widely acknowledged that some algorithms perform better when the input numerical range has been scaled.

Third, instead of directly using the state column as the output target classification, a composite of the field will be used instead. Since the objective is to determine whether a project will be successful or not, a state of "successful" will be labeled with a value of "1" and all other values will be labeled with a value of "0".

Closely related to the third point, note that some records are labeled with a state value of "live". These represent actual ongoing projects and are therefore not possible to be determined if they will be successfully funded or not. As such, it might be better to filter out those records before processing. Similarly, there are records with state value of "undefined", they require further investigation and are also not easily determined if they are successful or not.

Finally, advanced machine learning algorithms such as Deep Learning may be used to classify the records. Their flexibility and robustness may result in higher accuracy values.

III. Conclusion

Based on initial results, the best machine learning algorithm to use for Kickstarter project success classification is the Gradient Boosting algorithm. Among all the factors, it is the category of the project and its length which has the greatest bearing on its success.