Data Science Project Protocol

Author(s):
Shay Boochboot

Great Thanks to Dr. Tomas Karpati

Introduction

Start-up Nation - Two items published recently point more than anything to the strength of the local high-tech industry:

1. While job vacancies declined sharply since the start of the Covid-19 crisis, the demand for employees in the high-tech sector returned very close to its pre-corona figure in November.
2. Foreign direct investments (FDI) reached a record level of USD 19 billion in the first three quarters of the year.

I believe that most of the investments focused on the high-tech industry.

Looking ahead, as long as the extremely low-interest-rate environment continues globally, and the NASDAQ remains at its record levels (evidence of the positive global sentiment towards tech companies), this trend will probably persist.

In recent years, the range of funding options for projects created by individuals and small companies has expanded considerably. In addition to savings, bank loans, friends and family funding, and other traditional options, crowdfunding has become a popular and readily available alternative.

Kickstarter is an internet service for people to raise money from crowdfunding. For a person to receive any resources, his or her campaign must meet its target by a deadline set at the time of publishing. To be successful & achieve full funding by the deadline, a campaign must effectively convey its mission to potential backers and persuade them to support it over other campaigns.

Previous literature on predicting the success of Kickstarter campaigns has focused on applying machine learning to investigate whether it is possible to use information present at the beginning of a campaign to predict its success without using time-dependent data or data external to the Kickstarter campaign itself. There have been several studies that leverage machine learning techniques to predict the success of a campaign. Etter et al. analyzed the social network by constructing a projects-backers graph and monitoring Twitter for tweets that mention the project. They also discuss predictions based on the time series of early funding obtained. Combining features with social information helped to improve the model substantially. A study by Ethan Mollick on Kickstarter dynamics found that higher funding goals and longer project duration lead to lower chances of success, while inclusion of a video in a project pitch and frequent updates on the campaign increase the likelihood of full funding

A huge variety of factors contribute to the success or failure of a project - in general, but also on Kickstarter. Some of these can be quantified or categorized, which allows for the construction of a model to attempt to predict whether a project will succeed or not.

The aim of this project is to construct such a model and also to analyse Kickstarter project data more generally, in order to help potential project creators to assess whether or not Kickstarter is a good funding option for them, and what their chances of success are.

Vincent Etter, Matthias Grossglauser, and Patrick Thiran. “**Launch Hard or Go Home!**” Ecole ´ Polytechnique Fed´ erale de Lausanne (EPFL). ´

Lausanne, Switzerland. Ethan Mollick. “The Dynamics of Crowdfunding: An Exploratory Study.” The Wharton School of the University of Pennsylvania. Philadelphia, Pennsylvania.

Methodology (Project design)

Data

This project analyzed the publicly available data on the Kickstarter campaigns using the database reported between the company's launch in April 2009, up until the date of the webscrape on 11 Nov 2020. The dataset used in this project was downloaded in .csv format from a webscrape conducted by a webscraping site called **Web Robots**. The time-series data covering 213,193 campaigns.

Cleaning and pre-processing

A fair amount of cleaning was required to get the dataset into a format suitable for applying machine learning models.
After duplicates and irrelevant rows were dropped because projects which were cancelled mid-campaign, or which were still live, I was left with a dataset of 168,979 projects.

The columns which were kept or calculated were:

- The project goal (in USD)
- Campaign length — number of days from launch to deadline
- Number of days from page creation to project launch
- Blurb word length
- Name word length
- Whether the project was highlighted as a staff pick (one-hot encoded)
- Category (one-hot encoded)
- Country (one-hot encoded)
- Month a project was launched in (one-hot encoded)
- Month of a project's deadline (one-hot encoded)
- Day of the week a project was launched on (one-hot encoded)
- Day of the week of a project's deadline (one-hot encoded)
- Two-hour time window a project was launched in (one-hot encoded)
- Two-hour time window of a project's deadline (one-hot encoded)

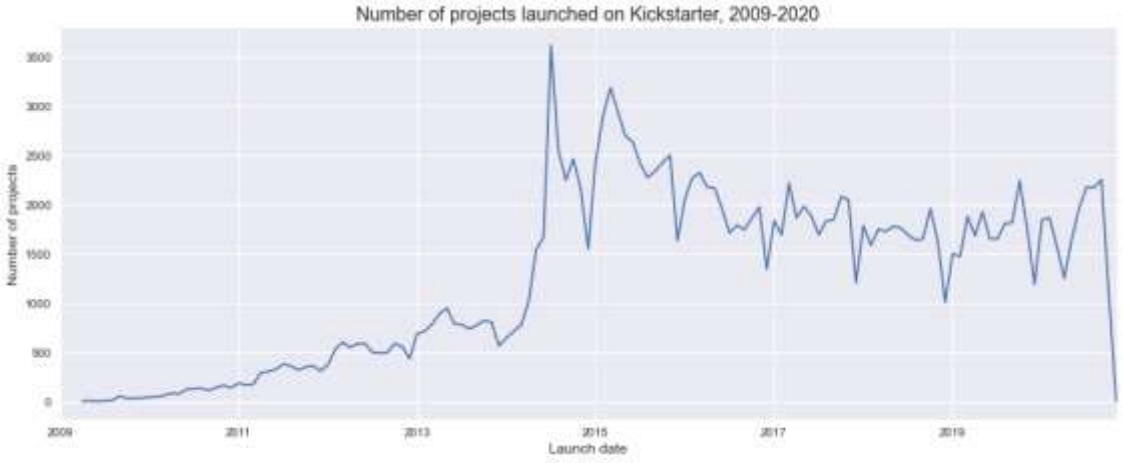
Some features were initially retained for exploratory data analysis (EDA) purposes, but were then dropped in order to use machine learning models. These included features that are related to outcomes (e.g. the amount pledged and the number of backers) rather than related to the properties of the project itself (e.g. category, goal, length of campaign)

Exploratory data analysis

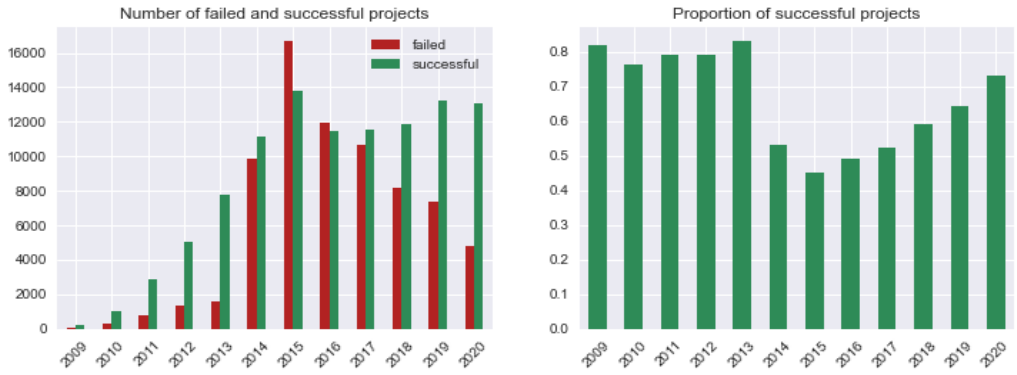
Crowdfunding at Kickstarter has become popular & grown massively since its launch in 2009, particularly during 2014 when entrepreneurs outside of the US joined along. The proportion of projects

that succeed decreased considerably at this point, however, the reason as the site was flooded with a much larger number of projects. The success rate has been on the increase in recent years.

Overall, 56% of completed projects (i.e. those that have finished and weren't cancelled or suspended) were successful.
Changes over time in the number of projects launched on Kickstarter

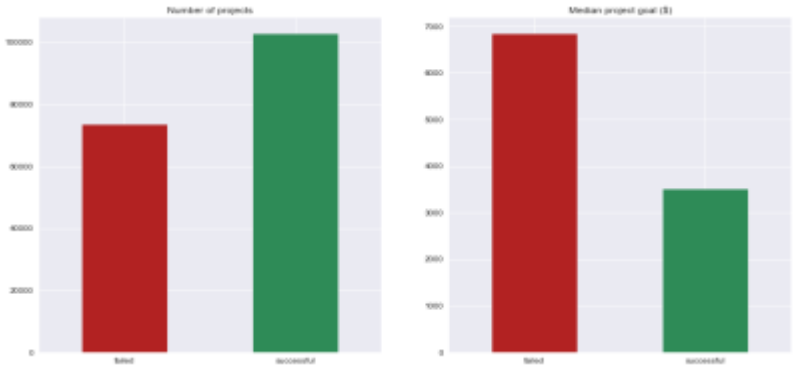


Changes over time in project successes and failures

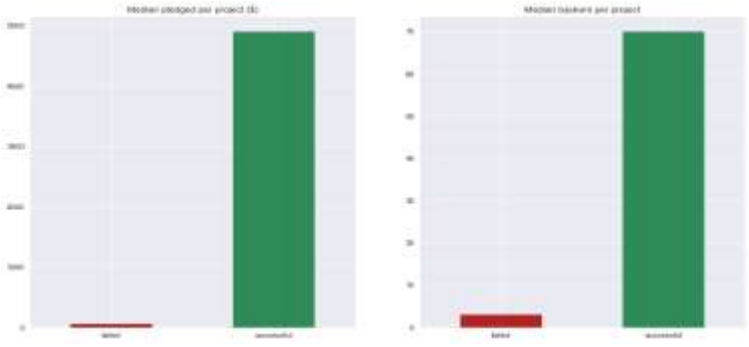


The graphs below show the differences in some of the features between successful and failed projects. The key takeaways from this are:

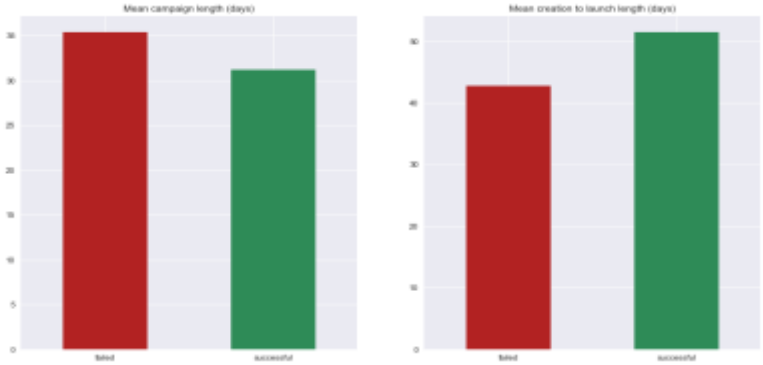
- Unsurprisingly, **successful projects tend to have smaller (and therefore more realistic) goals** — the median amount sought by successful projects is about half that of failed projects (medians are used due to high positive skew of funding and goal amounts).

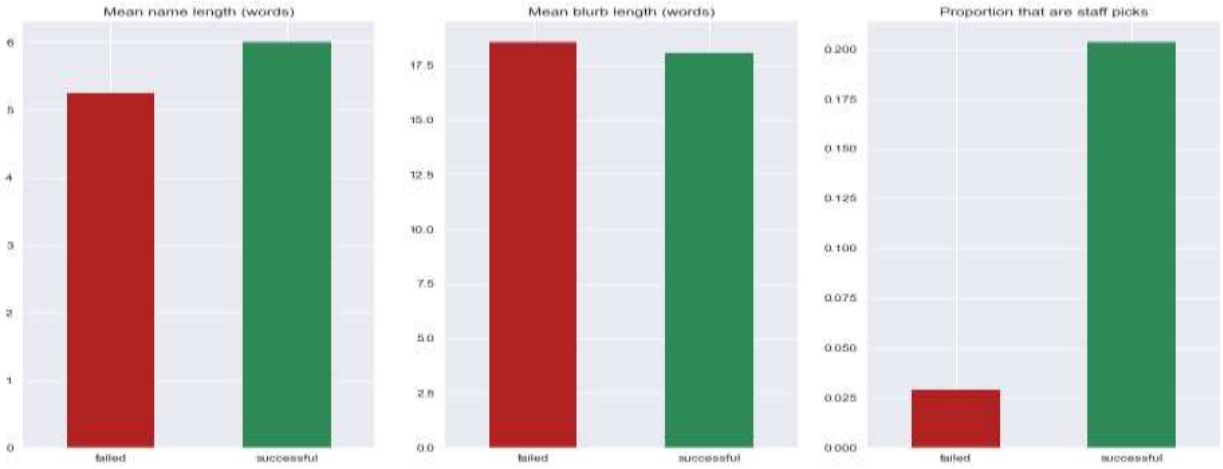


- The differences in the median amount pledged per project are more surprising. The median amount pledged per successful project is notably higher than the median amount requested, suggesting that **projects that meet their goal tend to go on to gain even more funding, and become 'over-funded'**.

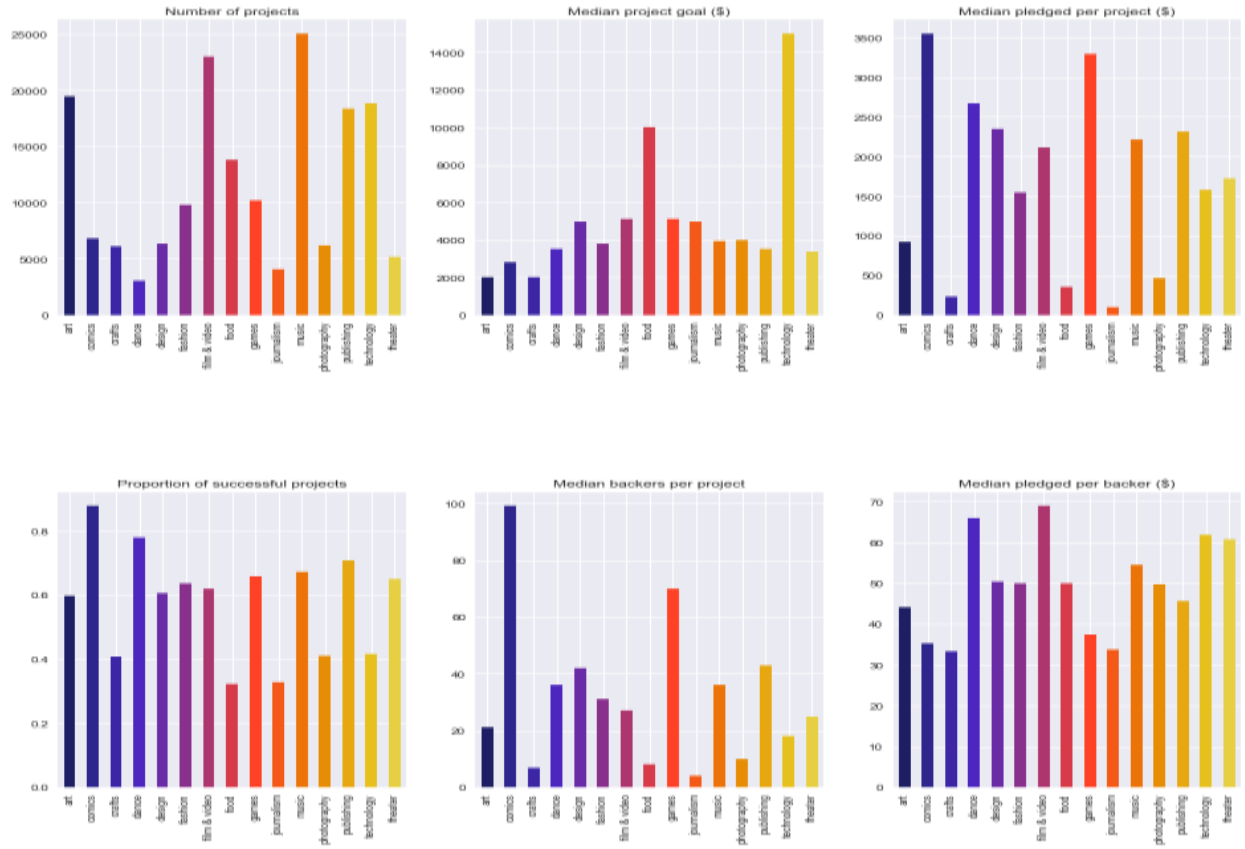


- On a related note, the difference between failed and successful companies is much larger in terms of amount pledged and the number of backers, compared to goal amount. Probably **once potential backers see that a project looks like it will be successful, they are much more likely to jump on the bandwagon** and fund it.
- Successful projects have slightly **shorter campaign lengths**, but take slightly longer to launch (measured from the time the project was first created on the site).
- Roughly 20% of successful projects were highlighted on the site as staff picks. It does not seem unreasonable to suggest a causative relationship here, i.e. that **projects that are chosen as staff picks are much more likely to go on to be successful**, and that only a few staff picks go on to fail.





Various other features were explored, in terms of project number, goal and funding amounts, backers and success rates. For example, the graphs below show the differences between different project categories.

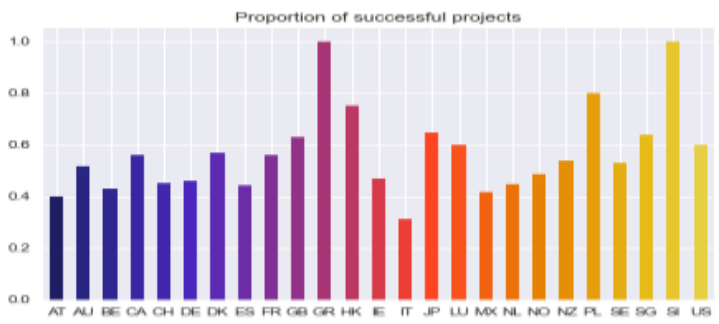


The key takeaways from this are:

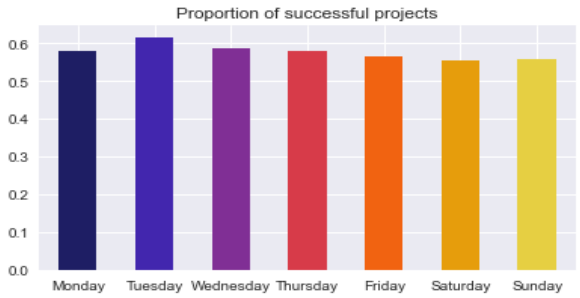
- The best project types to launch on Kickstarter are comics (on the grounds of success rate, number of backers and amount pledged), dance (success rate and amount pledged) and games (amount pledged and number of backers). This is probably at least partly due to their relatively small funding goals — as noted above, projects with smaller goals tend to be more successful.
- Although comics and games tend to attract the most backers, each backer tends to pledge relatively little. Dance and film & video tend to attract the most generous backers.
- Technology projects have the highest median goal size by far. However, they are towards the bottom of the leaderboard in terms of the median amount actually pledged.
- The worst performing categories are food, journalism and technology.

In the interests of space and retinas, only the 'success proportion' graphs will be shown below for additional features. The key takeaways from this are:

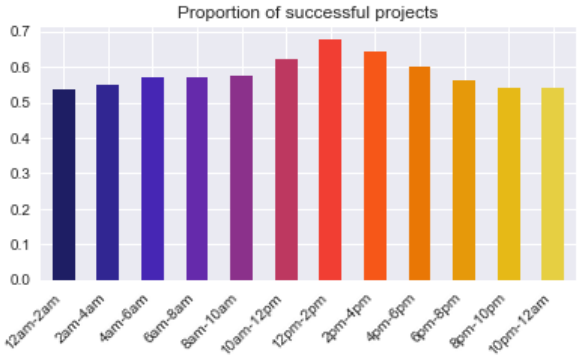
- **Projects from Greece, Slovenia & Poland are the most successful project.**



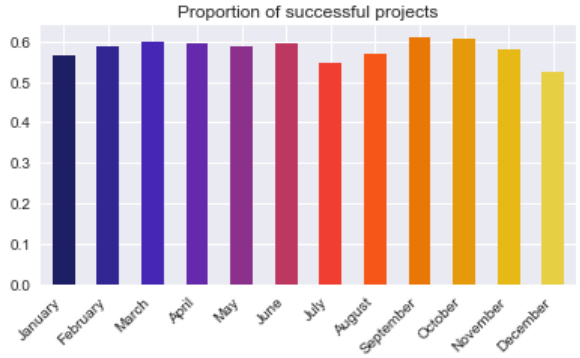
- **Tuesday is the best day to launch a project**, and **weekends are the worst** (the same pattern holds for the amount raised and the number of backers).



- **12pm to 2pm UTC** is the best time to launch a project — it also has the greatest median number of backers and amount of funding. **6pm to 4am UTC** is the worst time to launch.



- **September is the best month to launch a project** — it also has the greatest median number of backers and amount of funding. **July and December are the worst months.**



Preparing the data for machine learning

The ultimate goal of this project was to create a model that could predict, with a good level of accuracy, whether a project was likely to succeed or fail.
In order to prepare the data for machine learning, the following steps were taken (code below):

1. One-hot encoding categorical variables.
2. Separating the data into the dependent target variable ‘y’ (in this case ‘state’, i.e. project success or failure) and the independent features ‘X’.
3. Transforming the features in X so that they are all on the same scale. For this project, StandardScaler from Scikit-learn was used to transform each feature to a mean of 0 and a standard deviation of 1.
4. The data was separated into a training and test set, for robust evaluation of the models.

It is good practice to choose an evaluation method before running machine learning models — not after. The weighted average F1 score was chosen. The F1 score calculates the harmonic mean

between precision and recall, and is a suitable measure because there is no preference for false positives or false negatives in this case (both are equally bad). The weighted average will be used because the classes are of slightly different sizes, and we want to be able to predict both successes and failures.

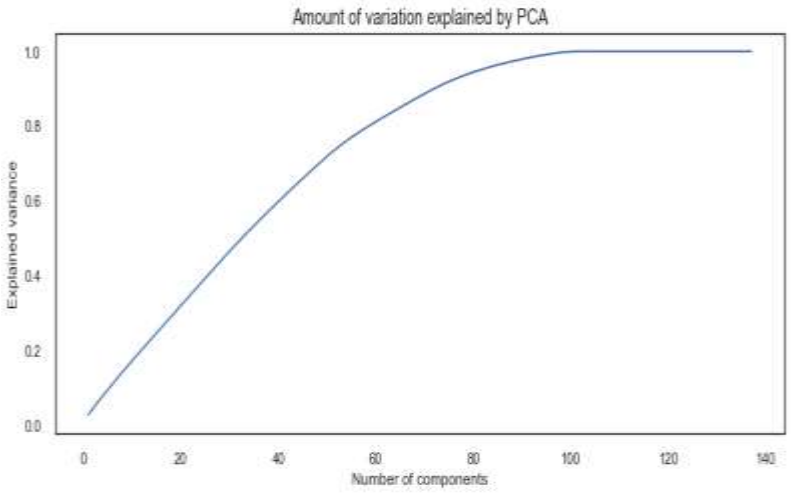
Model 1: vanilla logistic regression
Logistic regression can be used as a binary classifier in order to predict which of two categories a data point falls in to. To create a baseline model to improve upon, a logistic regression model was fitted to the data, with default parameters.

```
Logistic regression score for training set: 0.72047
Logistic regression score for test set: 0.71856

Classification report:
              Precision    recall  f1-score   support

           0       0.71      0.56      0.63     22164
           1       0.73      0.83      0.77     30696

    Accuracy                           0.72     52860
   Macro avg       0.72      0.70      0.70     52860
Weighted avg       0.72      0.72      0.71     52860
```

The model has a weighted average F1 score of 0.71. The aim is now to improve upon this score.

Principal Component Analysis

There were a large number of features (106) in the dataset used for the initial logistic regression model. PCA (Principal Component Analysis) was used to reduce this into a smaller number of components which still explain as much variation in the data as possible. This can help improve model fitting and accuracy.
The graph below shows that there was no obvious cut-off for the number of components to use in PCA. The following results were found:




```
Number of components explaining 80% of variance: 58
Number of components explaining 90% of variance: 71
Number of components explaining 99% of variance: 94
```

To choose the number of components to use in the machine learning models, each of these values was plugged into a pipeline for a logistic regression model using the default parameters: The results showed that the score is highest for 90 components, although the difference is small (c. 3% improvement from 58 components):

```
Number of components: 58
Score: 0.66897

Number of components: 71
Score: 0.69591

Number of components: 94
Score: 0.7182
```

The above results show that the score is highest for 94 components, although the difference is small

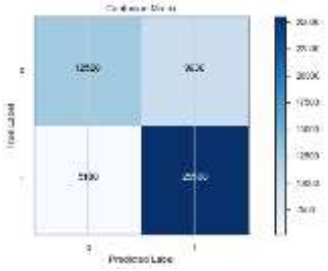
Model 2: logistic regression with PCA and parameter optimization

The logistic regression model can potentially be further improved by optimizing its parameters. GridSearchCV was used to test multiple different regularization parameters (values of C), penalties (l1 or l2) and models with and without an intercept.

```
Results from the logistic regression parameter optimization:
# Time taken to run: 4.7 minutes
# Best accuracy: 0.72
# Best parameters: {'clf__C': 10, 'clf__fit_intercept': True, 'clf__penalt
y': 'l2'}
```

A classification report and a confusion matrix were then produced for the logistic regression model using the best parameters (according to the accuracy score).

```
Logistic regression score for training set: 0.72047
Logistic regression score for test set: 0.71956

Classification report:
              precision    recall  f1-score   support

           0       0.71      0.57      0.63     22164
           1       0.73      0.83      0.77     30696

    accuracy                           0.72     52860
   macro avg       0.72      0.70      0.70     52860
weighted avg       0.72      0.72      0.71     52860
```

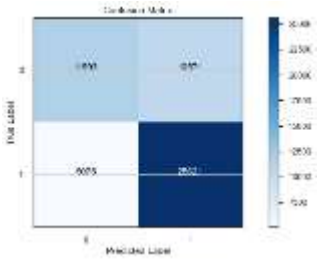


After hyperparameter tuning, the model's accuracy score is the same as the logistic regression model using default parameters (0.71 weighted average F1 score).

Model 3: Random Forests

Random Forest classifier was used. The Random Forest algorithm is a supervised learning algorithm that can be used for classification. It works by building multiple different decision trees to predict which category a data point belongs to. Again, GridSearchCV was used to test multiple different hyperparameters, in order to optimise the model.

```
Results from the Random Forest parameter optimisation:
Time taken to run: 49.3 minutes
Best accuracy: 0.7
Best parameters: {'clf__max_depth': 40, 'clf__min_samp
les_split': 0.001, 'clf__n_estimators': 100}
```



Best Random Forest model

```
Random Forest score for training set: 0.78204
Random Forest score for test set: 0.70401

Classification report:
              precision    recall  f1-score   support

           0       0.70      0.52      0.60     22164
           1       0.71      0.83      0.77     30696

    accuracy                           0.70     52860
   macro avg       0.70      0.68      0.68     52860
weighted avg       0.70      0.70      0.70     52860
```

After hyperparameter tuning, the model's weighted average F1 score increased to 0.70 for a model with default settings. This is similar to, although slightly worse than, the logistic regression model. Also, the difference between the score for the training set and the test set suggests there might be some over-fitting. There may well be more scope for hyperparameter tuning here to further improve the model, but time precluded it.

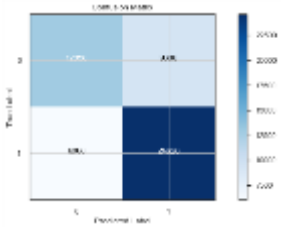
Model 4: XGBoost

XGBoost is a form of gradient boosting algorithm. Similar to Random Forests, it is an ensemble method that produces multiple decision trees to improve classification of data points, but it uses gradient descent to improve the performance of the model for the data points that are particularly difficult to classify. GridSearchCV was used to hyperparameter testing. Results from the XGBoost parameter optimization:

```
Time taken to run: 172.5 minutes (3 hours)
Best accuracy: 0.71
Best parameters: {'clf__learning_rate': 0.1, 'clf__max_depth': 35, 'clf__m
in_child_weight': 100, 'clf__n_estimators': 100, - 'clf__subsample': 1}
```

Although 172 Minutes, it was still only able to achieve the same accuracy as the initial regression model (this was also only a 0.01 increase in accuracy from an XGBoost model that was run with default parameters). The full results for the best XGBoost model are below:

```
XGBoost score for training set: 0.78123
XGBoost score for test set: 0.70916

Classification report:
              precision    recall  f1-score   support

           0       0.68      0.58      0.63     22164
           1       0.73      0.80      0.76     30696

    accuracy                           0.71     52860
   macro avg       0.70      0.69      0.69     52860
weighted avg       0.71      0.71      0.70     52860
```



As with the Random Forest model, the difference between the accuracy score for the training set and the test set suggests there might be some over-fitting. Again, there may well be more scope for hyperparameter tuning here to further improve the model — but I didn't have another 3 and a half hours to spare.

Model evaluation

Each model was able to achieve an accuracy of about **70%**, after parameter tuning. Although it was relatively easy to reach roughly this level of accuracy, parameter tuning was only able to increase accuracy levels by a small amount. Possibly the reasonably large amount of data for each of only two categories meant that there was enough data for even a relatively simple model (e.g. logistic regression with default settings) to achieve a good level of validation accuracy.

The best Random Forest and XGBoost models created still showed some degree of over-fitting. Further parameter tuning would be required to attempt to reduce this.

The final chosen model is the tuned **logistic regression model**. This is because, although each model was able to achieve a similar level of accuracy for the test set, this is the only model that did not exhibit overfitting.

Interestingly, **each model performed worse at predicting failures compared to successes**, with a lower true negative rate than true positive rate. I.e. it classified quite a few failed projects as successes, but relatively few successful projects as failures. Possibly the factors that might cause a project to fail

are more likely to be beyond the scope of the data, e.g. poor marketing, insufficient updates, or not replying to messages from potential backers.
The false positive and false negative rates mean that, if the data about a new project is fed through the model to make a prediction about its success or failure:

- if the project is going to end up being a success, the model will correctly predict this as a success about 80% of the time
- if the project is going to end up being a failure, the model will only correctly predict this as a failure about 60% of the time (and the rest of the time will incorrectly predict it as a success).

Conclusion

Hear some recommendations for project creators considering Kickstarter. Some of the factors that had a **positive effect** on success rate and/or the amount of money received are:
Most important:

- Smaller project goals
- Being chosen as a staff pick (a measure of quality)
- Comics, dance and games projects
- Projects from Hong Kong

Less important:

- Shorter campaigns
- Taking longer between creation and launch
- Film & video and music projects (popular categories on the site, and fairly successful)
- Launching on a Tuesday (although this is also the most common day to launch a project, so beware the competition)
- Launching in October
- Launching between 12pm and 2pm UTC (this is of course related to the country a project is launched from, but remember that backers can come from all over the world)

Factors which had a **negative effect** on success rate and/or the amount of money received are:
Most negative:

- Large goals
- Food and journalism projects
- Projects from Italy

Less negative:

- Longer campaigns
- Launching on a weekend
- Launching in July or December
- Launching between 6pm and 4am UTC

Overall, Kickstarter is well suited to small, high-quality projects, particularly comics, dance and games. It is less suited to larger projects, particularly food (e.g. restaurants) and journalism projects.