PREDICTIVE ANALYSIS APPLIED ON KICKSTARTER

Niccolò Bencini, Emanuele Caiffa, Giovanni De Feudis, Eleonora Palomba, Nataliya Zayeva

Whether you like to define yourself as an innovation-passionate and you firmly believe in your entrepreneurial mind-set or not, you have surely heard at least once of "Kickstaster.com", one of the most important crowdfunding platforms aimed at helping people to obtain financial support for their creative ideas. In 2020 the company has reached a peak of 507.009 projects published, with a success rate of 38,28%. The analysis reported has been conducted with the aim of providing a prediction, using Machine Learning techniques, of whether a project will be successful or not.

Index

1. Introduction 1
2. Dataset overview 1
3. Data exploration 2
4. Pre-processing 3
5. Modelli di classificazione 4
6. Evaluation 5
7. Conclusions and future developments 6
References 6

1. Introduction

Kickstarter is an American corporation for the public benefit, and it's based in Brooklyn, NY. It was founded in 2009 by Perry Chen, Yancey Strickler and Charles Adler. The aim behind the platform is to help people – not necessarily potential entrepreneurs, but also musicians, designers, publishers, and many other creative minds – to reach a specific fundraising goal in order to bring their projects to life, which is also the company's slogan. Following this schema, Kickstarter is basically driven by creators and backers. Creators present their project details on the platform using text, videos and photos, setting a specific goal and a deadline for the fundraising campaign. Backers can be identified as people from a general public who decide to give their contribute by financing, through donations, ideas considered worthy or interesting: most of them, however, expects a reward in return if the project turns out to be successful. Creators may also set reward levels for backers who pledge specific amounts. Usually, the more a backer pledges, the bigger will be the reward.

2. Dataset overview

The dataset chosen for answering the research question – whether a project published on Kickstarter will be successful or not – is "Kickstarter Projects", available on Kaggle Platform, that collects the list of projects published until January 2018. It's made of 378661 records and of the following 15 attributes:

- *ID (Numeric)*: internal Kickstarter ID
- *NAME (String)*: name of the project - a project is a finite work with a clear goal that you'd like to bring to life. Think about albums, books, or films.

- *CATEGORY (String)*: category the project belongs to (music, restaurants, games, ...)
- *MAIN CATEGORY (String)*: main category of campaign
- *CURRENCY (String)*: currency used by backers to support the project
- *DEADLINE (Numeric – Interval)*: crowdfunding deadline
- *GOAL (Numeric)*: fundraising goal – the funding goal is the amount of money that creators need in order to complete their project
- *LAUNCHED (Numeric – Interval)*: date the project was launched
- *PLEDGED (Decimal)*: amount pledged by crowd
- *STATE (Categorical):* - current condition the project is in
- *BACKERS (Numeric):* number of backers that have decided to finance the project with an amount at least equal to the minimum settled
- *COUNTRY (String):* Country in which the project was launched
- *USD PLEDGED (Decimal):* pledged amount in USD (conversion made by KS)
- *USD_PLEDGED_REAL (Decimal)*: pledged amount in USD (conversion made by fixer.io api)
- *USD_GOAL_REAL (Decimal)*: goal amount in USD

The goal of our analysis is to predict whether the *state* of a project published on Kickstarter.com will be equal to 'successful' or 'failed', using machine learning tools, and evaluating the results of our prediction by using some validated parameters.

This report follows the organization described:
1. Data exploration: we used visual representations in order to better comprehend our data and their characteristics, with a particular focus on the variable '*state*'
2. Pre-processing: techniques used in order to make the dataset more suitable for the analysis
3. Models: we decided to experiment the use of different kind of predictive models for our analysis, which will be described in detail later
4. Evaluation: phase in which we compared the different models used, in order to choose those ones that reached a better performance.

3. Data exploration

The first step of the analysis consists in data exploration, here conducted with the intent to investigate the initial distribution of the variable '*state*'.
As it's possible to notice by observing the histogram given below, the original composition of the variable showed the presence of 6 attributes:

- *Failed:* contains all projects that failed collecting the fundraising goal before the deadline
- *Canceled:* contains all projects that have been stopped by their creators, because they wanted to make some changes such as in the funding goal or campaign duration, or because they wanted to rework their idea and start again, probably due to poor trust (and small pledges) by backers
- *Successful:* contains all projects that successfully reached the expected amount in time
- *Live:* contains all projects currently available on the platform when the dataset was created, whose deadline was not yet expired
- *Undefined:* projects that gone lost or could not be analysed when data were collected
- *Suspended:* projects that have been suspended by Trust & Safety team, due to evidence that they are in violation of Kickstarter's rules

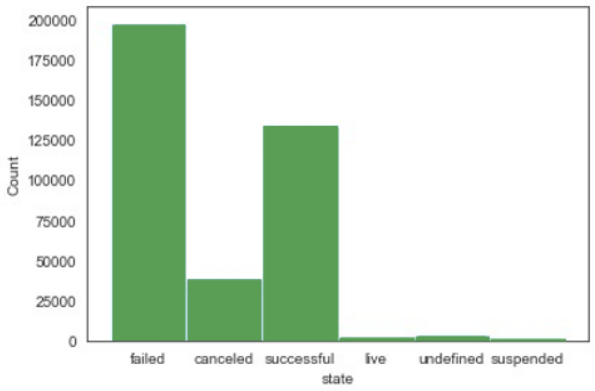

Figure 1: Original distribution of 'state' variable

However, since the purpose of our analysis is to predict successful and failing projects, we made the decision to rework our dataset in order to discard 'live', 'undefined' and 'suspended' projects, and to merge 'canceled' with 'failed' projects. After completing this pre-processing procedure, the situation of the variable '*state*' is as follows:

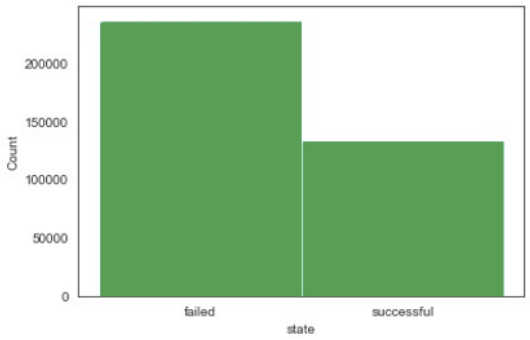

Figure 2: Distribution of 'state' variable after pre-processing

It's possible to deduct by observing the histogram that the target class features 64% of failing projects, and 36% of successful ones. This means that the class is imbalanced and we cannot use accuracy as an evaluation measure.
Consequently, we decided not to consider as features the values of 'usd_pledged' and the number of 'backers', since they are highly correlated with the value of 'state'. This result might be caused to the fact that, by observing the number of backers as well as the amount collected, it is possible to deduce if a project will be successful or failing.

4. Pre-processing

The subsequent step of the analysis includes the application of data cleaning and data pre-processing techniques. The dataset initially showed some anomalies. Starting from the original dataset, consisting of 378661 rows, 3797 rows were removed, due to the presence of an incorrect value of the *country* column. Furthermore, in order to reach the purpose to build a model that can predict whether a project will be successful or not, it was decided to discard the records that had a state value equal to '*undefined*', '*live*' or '*suspended*' (4642 rows), but also to consider those with state '*canceled*' as belonging to the same class as the records with state '*failed*'. Overall, 8439 rows were removed, corresponding to approximately 2% of the dataset.
Columns *ID* and *name* were then discarded, as they were considered irrelevant for the creation of the model; also, in order to avoid problems arising from currency differences, it was then decided to use *usd_pledged* and *usd_goal*, instead of *pledged* and *goal*, since they contained the same values converted into US dollars.
In addition, it was decided to discard the *country* column as it is highly correlated with the *currency* column. Subsequently, the variable *days* was created with the intent of describe the difference in days between the expiration date of the project (deadline) and the launch date (launched).
Thus, we applied "one-hot encoding" procedure to create four new features, considering the month and the day of the week, of both launch and deadline dates.
Other categorical variables have been transformed with the same procedure, creating a dummy variable for each value of each variable. The reason behind the use of this procedure is that only certain algorithms can successfully be applied and directly work with categorical data (for example, decision trees can be learned directly from categorical data with no data transformation required). Indeed, many machine

learning algorithms cannot directly operate on label data, as they require all input variables and output variables to be numeric.

At the end of the pre-processing phase, the following features have been obtained (and will also be used for the development of the models):

- Main_category (15 dummy variables)
- Currency (14 dummy variables)
- Days
- USD pledged
- USD goal
- Month of launch date
- Month of deadline
- Day of the week of launch date
- Day of the week of deadline

Afterwards, we considered a second dataset containing a time series of circa 14 currencies, filtering the rows of interest and removing those currencies that were absent in the Kickstarter dataset. We managed missing values with a mean interpolation, dividing all the currencies values by an external currency value that we considered as an index: subsequently, we calculated the difference from the intertemporal mean. We joined the two datasets, combining each project with the correspondent currency value available at the launch date of the project (or the deadline if the launch date was missing). It was not possible to associate a specific currency value to 12980 projects, so that 357241 projects were considered overall.

5. Classification Models

After completing the pre-processing phase, the dataset is ready to be used for the machine learning models we chose, that are here described in brief:

- **Decision Tree J48**: is a technique based on decision tree concept, that is also able to classify nominal data. When implementing the model, we decided not to modify default parameters, leaving the confidence factor equal to 0.25 and the number of folds equal to 3.
- **Decision tree**: implements an algorithm based on decision trees
- **Random forest**: is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees
- **Logistic regression**: is a regression-based classification technique in which the posterior probability that the target variable returns the input value is calculated
- **XGBoost**: is a popular tree-based algorithm, that implements gradient boosting (method that differs from the normal loss function because the average split value of the trees is calculated including the loss function and the tree function). The concept of decision tree is included in the loss function:

$$\hat{\theta}_m = argmin_{\theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \theta_m))$$

where m represents boosting algorithm iteration[1]. (Hastie, et al., 2017)
- **MLP**: Multi-Layer Perceptron (MLP) is a neural network application. In particular, it was implemented a network with one hidden layer, composed by 5 neurons.

Each model was evaluated using cross validation. In particular 10-fold cross validation was used: the dataset was split in ten subsets using stratified sampling, and the models were

[1] $T(x;\theta) = \sum_{j=1}^{J} \gamma_j I(x \in R_j) \qquad x \in R_j \Rightarrow f(x) = \gamma_j$

trained ten times, using, at each iteration, nine parts as training set and one part as test set.
This was done in order to have a clearer understanding of the model performances and to better compare the performances of all models.

6. Evaluation

The metrics that we have chosen to evaluate our classification models' performances are: *F-measure*, *precision*, *recall* and *AUC* (Area Under Curve). We did not consider *accuracy* measure, since we noticed the presence of a class-imbalance problem (number of projects with state equal to 'failed' was significantly bigger than 'successful' ones).
Recall measure (also called *sensitivity*) represents the portion of positive records that are correctly classified by the models:

$$Recall = \frac{TP}{TP+FN}.$$

Precision measure describes the fraction of records that are actually positive above all the records classified as such:

$$Precision = \frac{TP}{TP + FP}$$

A valid classification model should show a high value of both measures. For this reason, F-measure (F_1 score) is considered too, that can be defined as the harmonic mean between precision and recall, and it allows us to interpret these two metrics at the same time:

$$F - measure = 2\frac{Precision \times Recall}{Precision + Recall}$$

Executing the validation of the training and test sets, the following results emerge, shown in the table below:

Classification Model	F-measure	Precision	Recall
J48	0.61	0.62	0.60
Decision Tree	0.58	0.59	0.58
Random Forest	0.58	0.59	0.58
Logistic Regression	0.56	0.62	0.57
XGBoost	0.62	0.66	0.62
MLP	0.60	0.64	0.60

It's possible to observe that the classifier that obtains the best performance in terms of F-measure, precision and recall is XGBoost.

Finally, ROC curve was calculated, in order to make a comparison between all the models used. The ROC curve (Receiver Operative Characteristic) is created by plotting the True Positive Rate (TPR), also known as *Sensitivity* or *Recall* against the False Positive Rate (FPR) at various threshold settings. The value of the Area Under the Curve (AUC) is extrapolated from this graph: the most performing classifier is the one that reaches the highest AUC value. In this case, as a confirm of what was previously observed trough F-measure, XGBoost can be considered superior to other classifiers. Multilayer Perceptron occupies the second position.

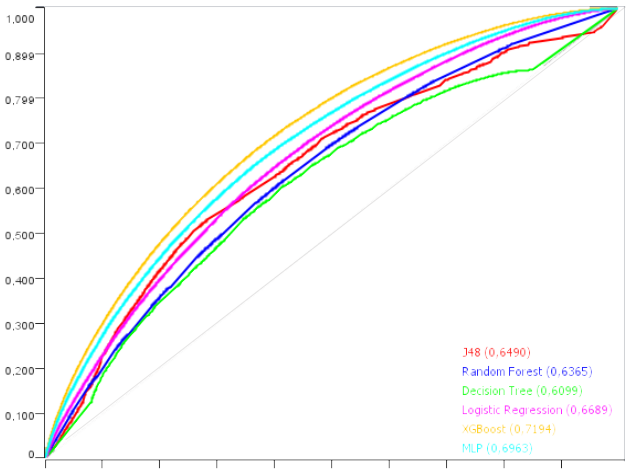

Figure 3: ROC Curve

7. Conclusions and future developments

Even though the results are encouraging, the adoption of more complex models, like bigger neural networks, or the addiction of more features, could be useful to improve the analysis.
For example, it could be interesting to apply text mining tools in order to analyse project titles and descriptions.
Furthermore, a stochastic process could be applied in order to identify models' optimal parameters, since we decided to maintain the default values in the models we implemented.

References

Hastie, Trevor , Tibshirani, Robert and Friedman, Jerome. 2017. *The Elements of Statistical Learning*. New York : Springer , 2017. pp. 353-360.

Kickstarter Projects . *Kaggle*. [Online] [Cited: 11 15, 2020.] https://www.kaggle.com/kemical/kickstarter-projects.

Currency Exchange Rates. *Kaggle*. [Online] [Cited: 11 15, 2020.] https://www.kaggle.com/thebasss/currency-exchange-rates.