**Advanced Data Science**

[COMP11068\_01](https://moodle.uws.ac.uk/course/view.php?id=769)

**Crowdfunding Success Prediction**

**Student: Banner ID:**

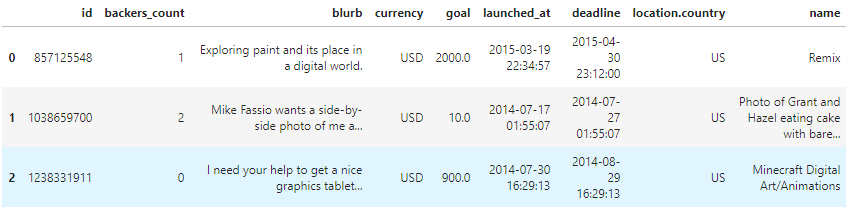
Cezar Anicai B00377631

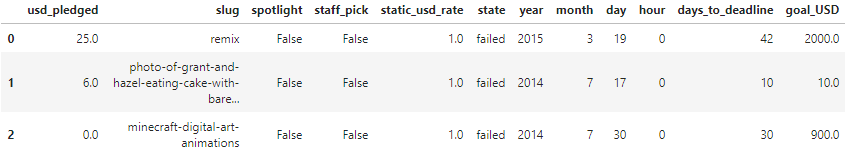
**Introduction**

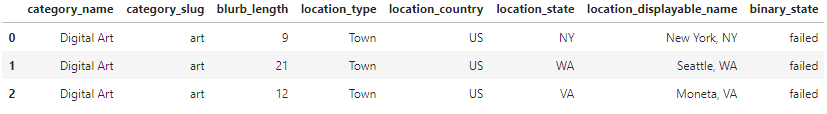
Crowdfunding is a method of gathering capital for starting a new business or scaling up an existing venture. It is done by collecting small amounts of money from a large number of persons. The degree of freedom regarding the business idea is much greater than traditional ways of funding offer, which makes it accessible to a wide range of entrepreneurs. According to Statista.com(2019), the global market size of crowdfunding in 2018 was 10.2 billion U.S. dollars and it is estimated to reach 28.8 billion by 2025, almost triple in less than a decade. With respect to the numbers stated above and the potential shown by the market, it is worthwhile to investigate what campaigns could be successful and which ones could be a failure. To do so, I accessed data from the largest crowdfunding platform and will perform a data science pipeline with the appropriate steps. The desired output of the pipeline is an accurate prediction whether a project will be successful or it will be a failure.

**Data Preparation**

**The dataset** was retrieved from Kaggle.com(2019) and it has over 310,000 unique raws containing information about campaigns ran on Kickstarter.com between 2009 and 2019. As for the features, there are 29 columns that can be classified as continuous and categorical variables, or in integer, float, bool, string and date.







**Data Cleaning**

In order to start working on the available data the most basic thing to do is to load it into the development environment. Since the dataset was stored in a comma separated value file, I made use of pandas built in function read\_csv().

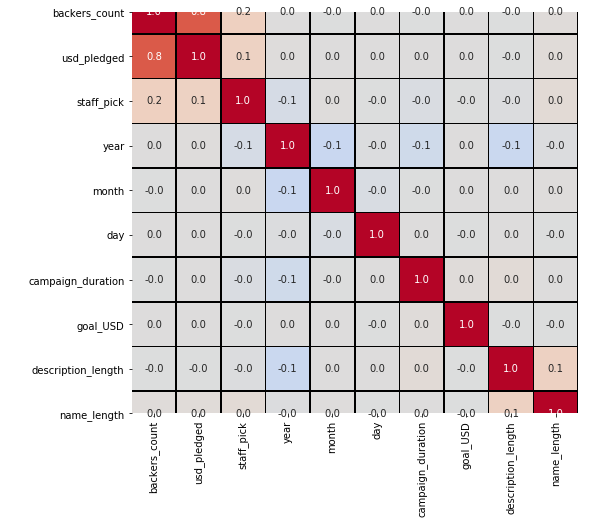
The cleaning phase is important because data can have many flaws and if they are not corrected it will be difficult to obtain valuable business decisions or it will lead to erroneous ones. The actions taken in this direction are listed below.

* ensuring the rows are unique by eliminating the duplicates, “id” and “blurb”(campaign description) features were used as subset parameters for the drop\_duplicates() function.
* checking for impossible values such as, negative goal amount, projects starting in the future or too long in the past, if the values in the month column are between 1 and 12, if the values in the day column are between 1 and 31 and if the duration is negative or too long, using numpy’s where() function.
* checking for any missing values revealed that “location.country” column had 1079 empty entries. On this occasion, it was noticed that the feature is doubled by “location\_country” which did not have NaN values, so the first was simply dropped.
* detecting outliers was done through visualization techniques, using seaborn’s boxplot() and scatterplot(), I analyzed the features and found outliers at many standard deviations away from the mean. When handling outliers it is recommended to remove them if they are input mistakes, but in this dataset they are valid values. There was no imputation performed since it could introduce bias and negatively manipulate the output of an algorithm. In the cleaning phase there was no action taken regarding the outliers, yet being aware of their existence will influence the algorithm choice in the modeling phase.
* eliminating the irrelevant independent features, that do not impact the dependent variable, or that can be calculated from other columns. “id” was removed, since it is a value which is not controlled by the campaign owner, “launched\_at” and “deadline” can be calculated from “year”, “month”, “day” and “days\_to\_deadline” and so forth.

**Transformations** are a good way of improving the quality of the dataset, therefore I renamed some columns in order to express more clear what they represent and I introduced a new feature, “name\_length”, by counting the number of words in every campaign name, which increases the comprehension of the dataset.

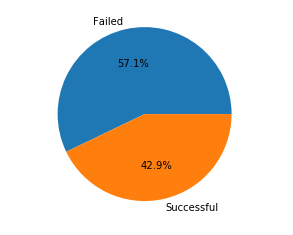
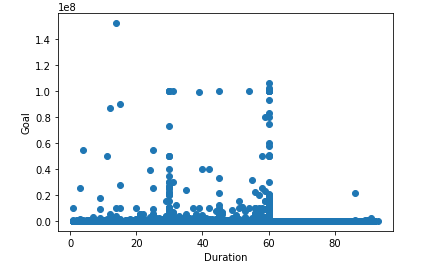
**Data Exploration**

In order to gain insight on the available data and to understand which features are the most appropriate to use in solving the classification task, an exploratory analysis was executed. If the dataset has perfectly positive or negative attribute, when a model will be trained it can be affected by multicollinearity. To avoid this issue a correlation analysis was executed by plotting the correlation matrix of the columns.



A high correlation can be observed between “usd\_pledged” and “backers\_count”, at this point the first feature was eliminated from the dataset.

Another means of visual exploration were piechart and scatterplot, using numpy and matplotlib. The figures show the “binary\_state” feature distribution and the relation between the goal amount and the duration.

A statistical exploration was also initiated, using pandas describe() function a few statistic measures were calculated such as mean and standard deviation. It was observed that the “hour” feature has mean, minimum and maximum values of 0 and was excluded.

**Data Modeling**

Before jumping into model selection we need to understand with what type of problem we are confronting. In the present dataset, the dependable variable that has to be predicted is categorical. It can take only two values, 0 or 1, successful or failure, therefore it is a classification problem.

There were four candidates algorithms used in the modeling phase. Logistic Regression, Decision Tree Classifier, Random Forest Classifier and [Bernoulli Naive Bayes Classifier. All the models were imported into the development environment from scikit-learn library.](https://chrisalbon.com/machine_learning/naive_bayes/bernoulli_naive_bayes_classifier/)

Logistic Regression was the first choice due to its simplicity in implementation, efficiency and ease of results interpretation. Because it is a commonly used algorithm in binary classification problems it seemed as a good starting point. Hence this algorithm cannot work with string values, a few more transformations were needed, specifically converting some features in dummy variables. Next I separated the features in independent and dependent:



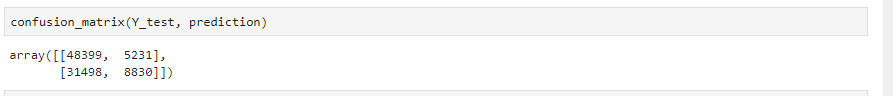
Afterwards the dataset was split in two parts, one for training the model and another one for testing. The split ratio was 70/30.

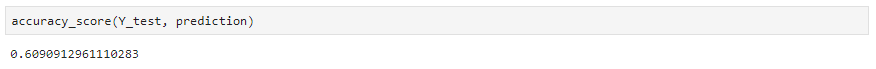


Then the model was created, fed with the training data and examined by the performances it has when making a prediction.

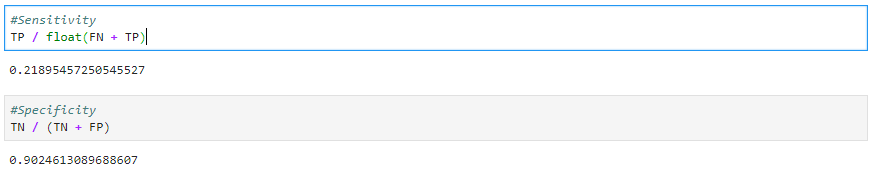


The model was evaluated by calculating multiple metrics from the confusion matrix. The accuracy score is the first number we look at and it is quite low, ~ 61%.

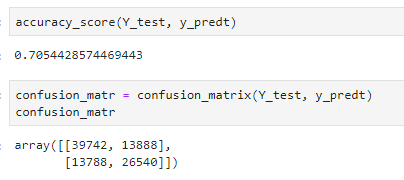
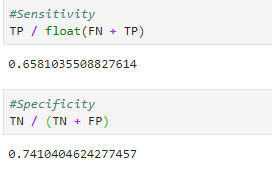




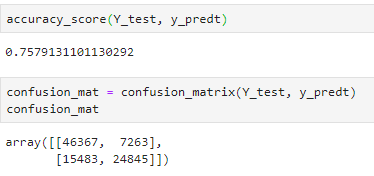
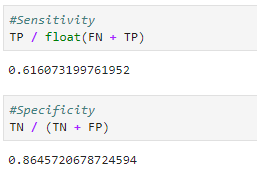
The sensitivity score represent the percentage of correct predictions when the actual value is positive (successful campaign) while specificity score represent the percentage of correct predictions when the actual value is negative (failed campaign) . In an ideal situation both of these scores should be 100%. For Logistic Regression the specificity is very good ~90%, but the sensitivity is very low ~22%.



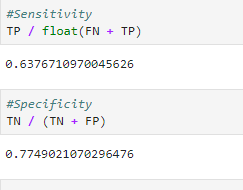
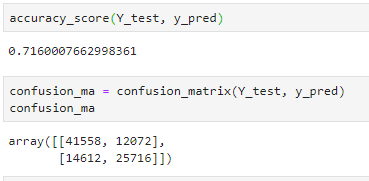
Since creating the model, training it and testing is similar for all the algorithms, further we will discuss only the results. The Decision Tree Classifier algorithm can work with both categorical and numerical values and it has good performance even on less tidy dataset. The cleaning process may overlook some mistakes and an algorithm that is not heavily affected by this is desirable. The accuracy, sensitivity and specificity score for this model are displayed below.

Random Forest Classifier was a logical next step, it has a set of decision trees and it should be more precise, it can also handle overfitting by averaging the predictions from all the trees eliminating thus the biases. The results obtained are:

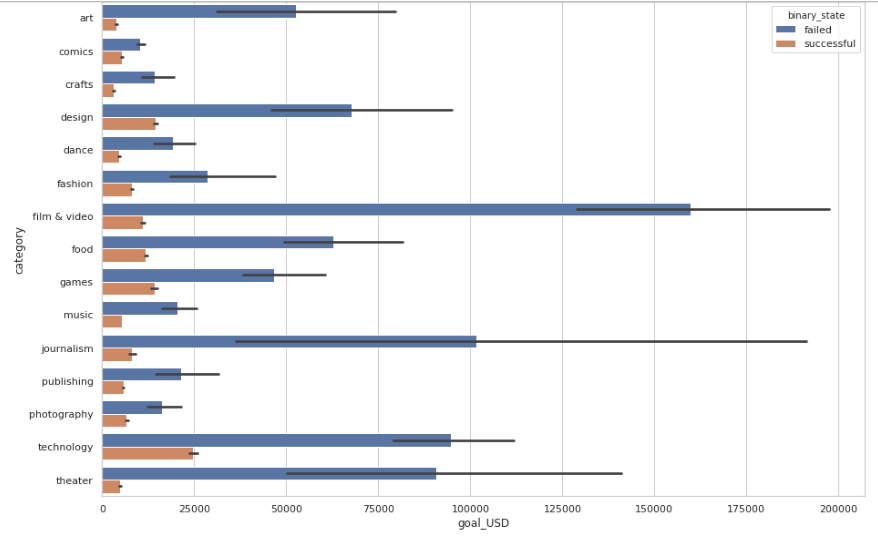
After creating the dummy variables in the beginning of the modeling phase, the dataset was transformed in a large collection of binary variable. The Bernoulli Naive Bayes Classifier requires exactly this kind of input and because it’s computationally faster than Random Forest Classifier it was selected as a candidate. The results yielded are:



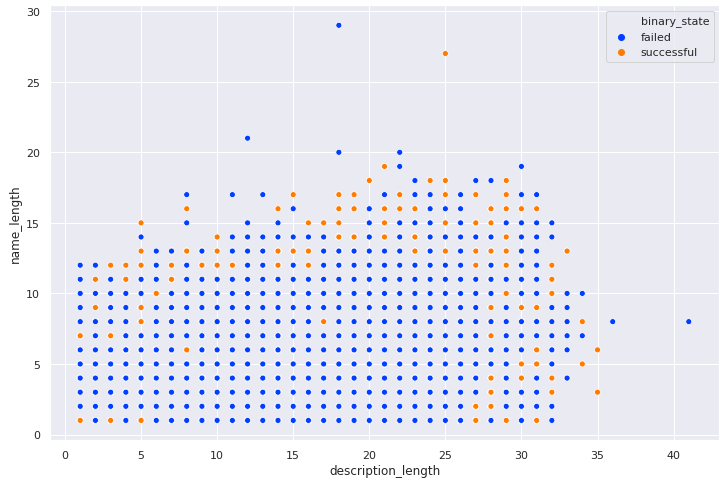
**Data Analysis**

The data collected for this project comprises of descriptive information regarding crowdfunding campaigns. In the data analysis phase the aim is on revealing and discussing the findings.

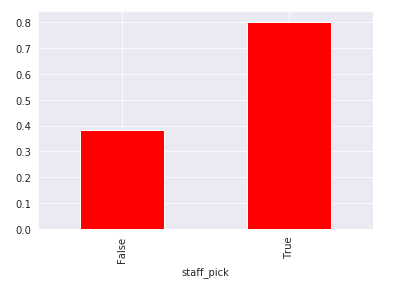
Examining the bar chart below we observe that the most ambitious goals are set for “film&video” projects, yet in every single category the successful campaigns have smaller goals then the failed ones. And overall to access the funding is recommended to ask for up to 25,000 U.S. dollars.



The next figure is a scatter chart displaying the success or failed campaigns based on the number of words in their description and title. It is easily remarkable that projects with titles that contains around 15 words are more likely to succeed, although if the description reaches approximately 30 words, the name length seems to be irrelevant.



An interesting feature of the dataset is “staff\_picked”, which states if the campaign was selected by the crowdfunding platform and promoted. Although probably “staff\_picked” is not influencing only by itself the success rate, it is noticeable that the campaigns promoted by the crowdfunding platform have a 80% chance of accessing the funding.



The results from the models applied to the dataset are controversial, the first algorithm, Logistic Regression has unacceptable accuracy but very high specificity, meaning that will predict correctly when a campaign is not successful 90 times out of 100. The Decision Tree Classifier performed better overall, with an accuracy score of approximately 70% and much higher sensitivity. On the down side there was a decrease in specificity and it requires more computation power. The Bernoulli Naive Bayes Classifier has behaved similarly to the previously mention model, there was an insignificant increase in accuracy and perhaps a faster period of training. The best candidate for this classification problem proved to be the Random Forest Classifier algorithm, despite the fact that it has the longest training period it obtained the best scores. An accuracy of almost 76% and a specificity of 86%. I believe that even better results could be reached, by tuning the Random Forest Classifier and executing a more accurate data cleaning and more comprehensive exploratory analysis on the dataset.

**Justification for tools used**

JupyterLab – was used as a development environment, it has a modular structure and you can create projects containing multiple source files any additional data files. It offers an attractive user interface, easy to operate, the code is executed line by line which gives a better learning curve and it makes it easy to fix.

Pandas – was used for data manipulation and statistical analysis. Reading the dataset file into a dataframe, renaming features inside the dataframe, identifying unique values inside the columns, accessing subset of the dataframe and describing the dataset.

NumPy – was used for mathematical operations and array processing, indexing and slicing.

MatPlotLib – was used for plotting, editing the plot size, the axes names and size, title, legend

Seaborn – was used for data visualization, displaying statistical relationships with scatterplot, barplot, piechart, and managing plot aesthetics like style, colorpallete.

SciKit-Learn – was used for data modeling, it has a very wide range of algorithms for all of the types of data science problems. The models are already implemented and ready for import and utilization. It also provide metrics for evaluating the models, such as confusion matrix, accuracy score or ROC curve.

Beside the functionality provided by all the libraries listed above, the community support is fantastic, the documentations are accessible for beginners and have examples and there are quite a lot of people engaged in using and developing this tools.

**References**

<https://www.statista.com/statistics/1078273/global-crowdfunding-market-size/>

<https://www.kaggle.com/toshimelonhead/400000-kickstarter-projects>