**Find your crowd**

We have a data about crowdfunding campaigns. The data of Kickstarter crowdfunding platform provided in Kaggle and available in the web was used for analysis (<https://www.kaggle.com/kemical/kickstarter-projects>).

The goal of the project is to understand which factors has the most influence on campaigns’ success.

As the information is available in csv format, we’ve just downloaded the file and then imported to start the analysis. The first step was to observe the data variables.

Models used in the project include:

* logistic regression,
* decision tree,
* Random forest
* gradient boosting,
* neural networks with 3 hidden layers
* And Linear regression to predict pledged money for campaign

**Data cleaning**

We’ve found out that there are 15 columns and 378661 observations in original data.

After looking through columns we saw that there are missing values in the data, so we should drop them. Here are the variables that we’ve dropped for the following reasons:

* ID: we don't need this column, it is unique for each campaign
* name: like ID case it is unique for each row and does not contain information
  + It also has null values
* goal: this column has values that are from different currencies and real value is in usd\_goal\_real
* pledged: same situation as in goal column
* usd pledged: this one contains many missing values and in some cases the values are wrong calculated
  + The real correctly calculated information is in usd\_pledged\_real

For other variables we still need more information to make decision.

The next step is to check the existence of duplicates. We see that main\_category and category variables have duplicate values (378491 observations), that’s why we’ve decided to drop one of them (category).

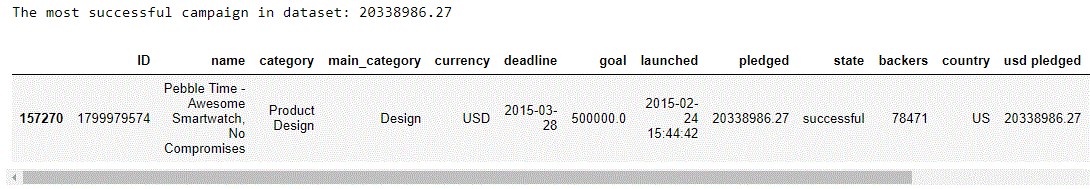
We also see that the top 5 countries of the number of crowdfunding campaigns are the USA, Great Britain, Canada, Australia and Denmark. The next plot shows the top 5 currencies of campaigns, which are USD, GBP, EUR, CAD, AUD. Most probably there’s a high correlation between these 2 variables. But we still need to check it.

For this purpose we’ve used crosstab to aggregate and jointly display the distribution of these two variables by tabulating the results one against the other in 2-dimensional grids. We see from the result that only one currency is used in each country, so there’s no need to keep the 2 variables.

By the next step pairplot was created with the variables "backers", "usd\_pledged\_real", "usd\_goal\_real". We see from the plot that

* there’s a positive correlation between "usd\_pledged\_real" and "backers"
* The relationship between "usd\_goal\_real" and "backers" is non linear and as the amount of the goal is getting higher, number of backers is getting lower.
* The relationship between "usd\_pledged\_real" and "backers" variables is linear, but there’s also noticeable sign of heteroscedasticity.

Afterwards, you may see that the most successful campaign in the dataset was “Pebble Time\_ Awesome Smartwatch” with the initial goal of 500 000 USD launched in USA in 2015.



The next calculation shows that the campaign exceeded its initial goal by 4068%.

Now let’s see the average performance of all campaigns:

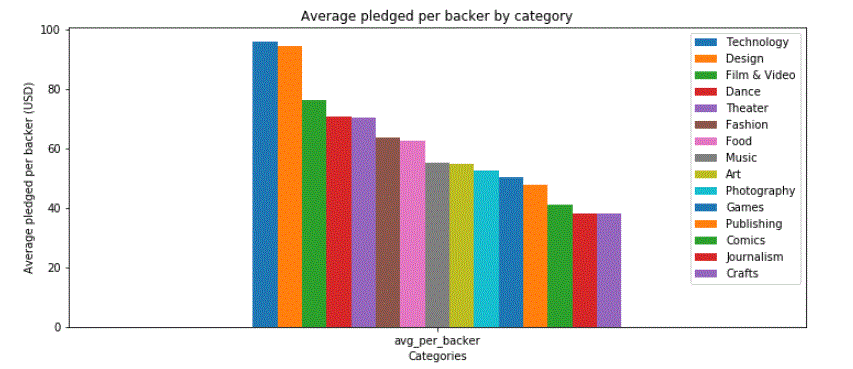
* Average pledged money in projects: 9058.924074119337
* Average goal in projects: 45454.40146545336
* Average backers in projects: 105.61747578969052
* Average duration of projects: 32.9549016356373
* Average pledged money per backer: 64.58076593075498

And here are the percentages of states of campaigns in dataset.

* failed 52.215306 %
* successful 35.376234 %
* canceled 10.241086 %
* undefined 0.940683 %
* live 0.739184 %
* suspended 0.487507 %

We need only successful and failed states in data to make classification model, so we should drop every other state.

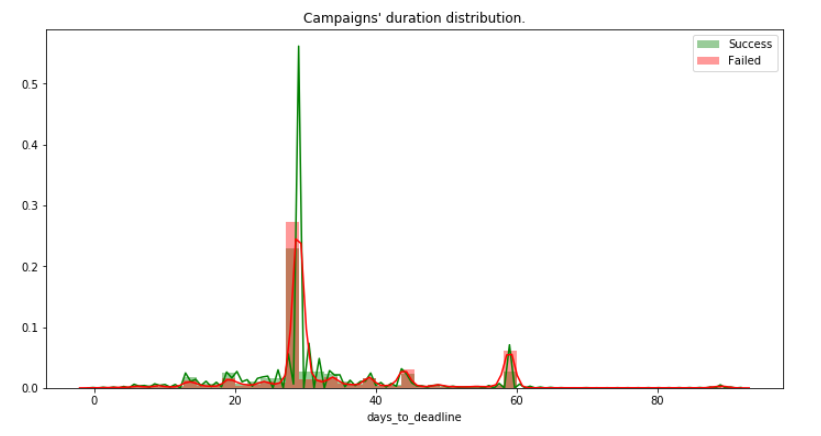
The biggest amount of money pledged in campaigns per backer in average are from Technology and Design categories.



In order to have the whole duration of campaign, we’ve created a new variable days\_to\_deadline, which is constructed from the <deadline-launched> and then we dropped “deadline” and “launched” variables from the dataset.

On average successful campaign have duration of 31 days. On the other hand, failed campaigns on average have duration of almost 34 days.

Here is the distribution of probabilities of either success or failure of campaign by its duration.



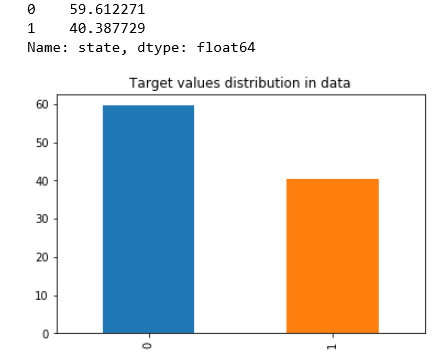
Variable “category” was dropped, because:

* It has 159 unique values (One Hot Encoding will make min. 158 columns)
* Dropping will make our models faster
* not to have much noise in the data
* more general info is contained in main\_category

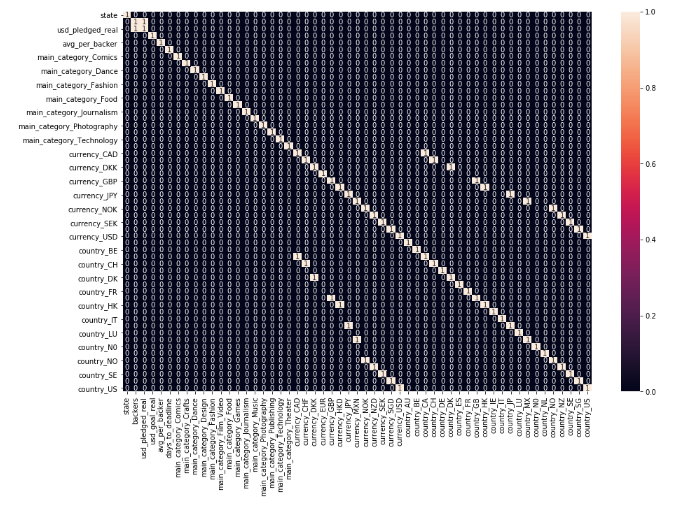
It’s time to make dummies for categorical variables. Final data has 331675 observations and 54 variables.



Almost 60% of campaigns failed and only 40% reached their goal. ( See the distribution attached below)



There is correlation between currencies and countries, which was assumed before from the bar plots as well.

****

**Classification**

In order to have faster computations we have to try to sample or implement dimension reduction techniques.

From logit summary we revealed the significant variables of the model. Those variables are the ones, that have p-values less than 0.05.

* backers
* usd\_goal\_real
* days\_to\_deadline
* main\_category\_Comics
* main\_category\_Crafts
* main\_category\_Dance
* main\_category\_Design
* main\_category\_Fashion
* main\_category\_Film\_Video
* main\_category\_Food
* main\_category\_Games
* main\_category\_Journalizm
* main\_category\_Photography
* main\_category\_Music
* main\_category\_Publishing
* main\_category\_Technology
* main\_category\_Theatre
* country\_CH
* country\_HK
* counry\_NO

We have done cross validation and grid search hyperparameter tuning for logistic regression and decision tree classification models by sampling data to 10,000 observations as our devices’ computational power could not handle more. Then we took best parameters and trained with them our models on whole data. We also kept some part of data for final testing.

**Best parameters for logistic regression:** ‘C’ 0.45, class-weight=balanced, penalty=l1.

**Best parameters for decision tree:** class-weight=None, criterion= gini, max-depth=5, min\_simple\_leaf=170.

The average score for ROC-AUC are:

* Logistic regression: 95,96%
* Decision tree: 90,92%

Then we split our data to train, test sets, and fitted it to models.

We also trained **random forest** and **gradient boosting** classifiers.

For decision tree based models the most significant features are backers and usd\_goal\_real variables and in the third place days\_to\_deadline, the results are the same for all.

**Neural network**

For neural network we used keras deep learning library, we designed network with three hidden layers, with rectified linear activation function, which takes a number as an input and gives the number if it is greater than or equal to zero, and zero otherwise. In output layer we’ve used sigmoid activation function to calculate probabilities of two possible outputs.

In compiler we’ve used adam optimizer (the most used/suggested), binary cross-entropy as loss function and accuracy as metric.

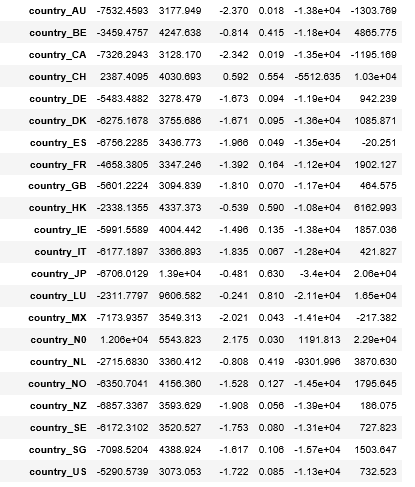
Results on train/test sets with 10 epochs (patience 3) stopped at 92,58 % accuracy.

**Linear Regression**

While applying linear regression we’ve found out that only 56% of variation in Y is explained by target variables, which shows that the model is not good and there predictor that we should take into account(for instance: if campaign owner send gifts to backers for support etc.).

We have implemented linear regression by statsmodels OLS and here is the interpretation of results:

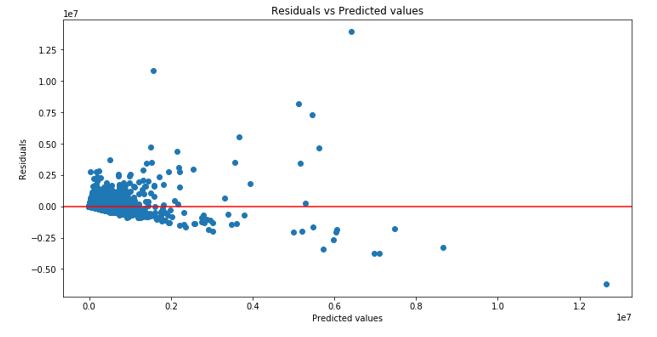




The variables that has p-values less than 0.05 are significant in terms of the effect on the success of the campaign, such as number of backers and duration of campaign. It is important to mention that amount of goal of campaign has effect on failure or success of the campaign, but is not significant in describing how much will raise it.

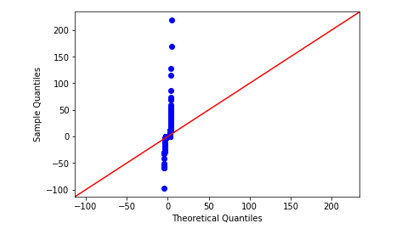
Now let’s check 4 assumptions:

1. Linearity of residuals
2. Independence of residuals
3. Normal distribution of residuals
4. Equal variance of residuals



Plot shows that the linearity assumption is not violated, but we see heteroscedasticity here.

Afterwards, we implemented quantile to quantile plot and shapiro-Wilk test to check normality assumption.



QQplot shows that the normality is violated. Same result is given by Shapiro test.

The null-hypothesis of Shapiro-Wilk normality test is that the data was drawn from a normal distribution. We reject the Null hypothesis which means that we can state with 95% confidence that our data is normally distributed. Concluding, normality assumption is violated.



The Null hypothesis of Rainbow test for linearity is that the regression is correctly modelled as linear. P-value of the test is more than 0.05 almost 100, which means that we fail to reject the null hypothesis, so by 99% we can claim that the model is linear.

Final results are the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Logistic Regression** | **Decision Tree** | **Random Forest** | **Gradient Boosting** |
| **ROC\_AUC** | **74** | **92** | **89** | **93** |
| **Accuracy** | **75** | **92** | **89** | **93** |
| **Recall** | **65** | **93** | **88** | **93** |
| **Train** | **75** | **92** | **89** | **94** |
| **Test** | **75** | **92** | **89** | **93** |

We can see from train and test scores that we don’t have overfitting in the mentioned table for non of the models, as the scores are too close to each other. From recall score we can see that for Logistic regression each new campaign has almost 65% chance to succeed. Same refers to the other models as well but with different values. We need ROC\_AUC score as bigger as possible, because the higher ROC\_AUC score leads to the higher recall, so we will have more confidence (probability will be bigger) to claim that each new campaign will have success.

In total 75% of logistic regression, 92% of decision tree, 89% of random forest and 93% of gradient boosting were predicted correctly. However from confusion matrices and classification reports we can see that our models predict failures of campaigns better than success.