Kickstarter Funding Analysis

by Shilong Li

1. Overview of Project

Kickstarter is a very popular website where creators can put their ideas or products to raise funding for supporting their projects. The main goal of the project is to predict the probability of Kickstarter campaign to be funded by scraping each campaign web page, parsing text content, constructing features, and then building machine learning models. Accordingly, this project consists of four modules to fulfill these functions, and 30942 Kickstarter campaign projects are thoroughly analyzed and classified.

The first module, ***Kickstarter_project_scraping***, is designed to read csv files downloaded from Web Robots database, and to further scrape the content of each campaign page with help of Request library using corresponding URLs.

The second module, ***Construct_meta_features***, is devised to first parse the scraped contents using BeautifulSoup package. Sequentially, different meta features are constructed from the parse data by natural language processing, including, for instance, the number of sentences, number of words, number of images, number of all cap words, etc., which will help classify different campaign projects in machine learning stage.

The third module, ***Feature_analysis***, is built for data cleaning and feature correlation analysis. Unnecessary features are first cleaned, and missing values are filled. Different correlation analyses are conducted to deeply understand the effect of features on the project funding (successful or failed), which contain, for example, outliner detection, correlation map, kernel density estimation for successful and failed campaign projects. After features analysis, the useful features are kept as the input data to the machine learning models.

The fourth module, ***ML_modeling***, is to design machine learning models with input data capable of classifying the Kickstarter campaigns into successful or failed projects and predicting the probability of a new project to be funded in the future. Specifically, the gradient boosted tree model with XGBoost shows excellent performance in this classification task.

2. Feature analysis

Different feature analyses are performed to dig into the data distributions and correlations. There is not much class imbalance in the target labels as shown in the following Fig. 1.

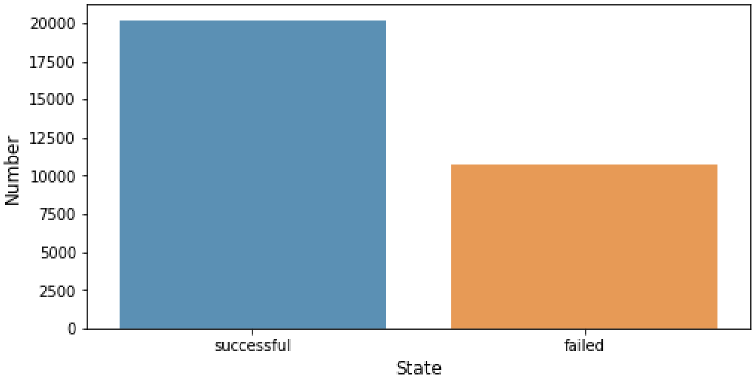


Fig. 1. Number of successful and failed projects

By plotting distributions of pledged feature as displayed in Fig. 2, the outliner is identified and deleted correspondingly in the data set.

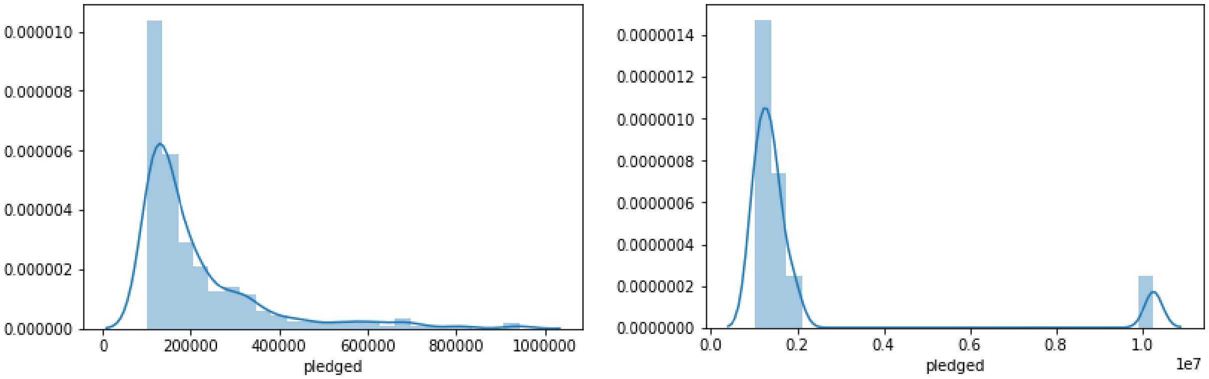


Fig. 2. Outliner detection

The correlation map is chosen to unveil the degrees of relationship among different features. From the map figure (Fig. 3), we can observe that these features, num_sents, num_words, num_exclms, num_images, num_bolded, may play vital roles in classifying the campaign projects. Furthermore, the features in terms of successful and failed projects are analyzed one by one to reveal those features that cannot function for the classification task. The correlation analysis results are illustrated in Fig. 4.

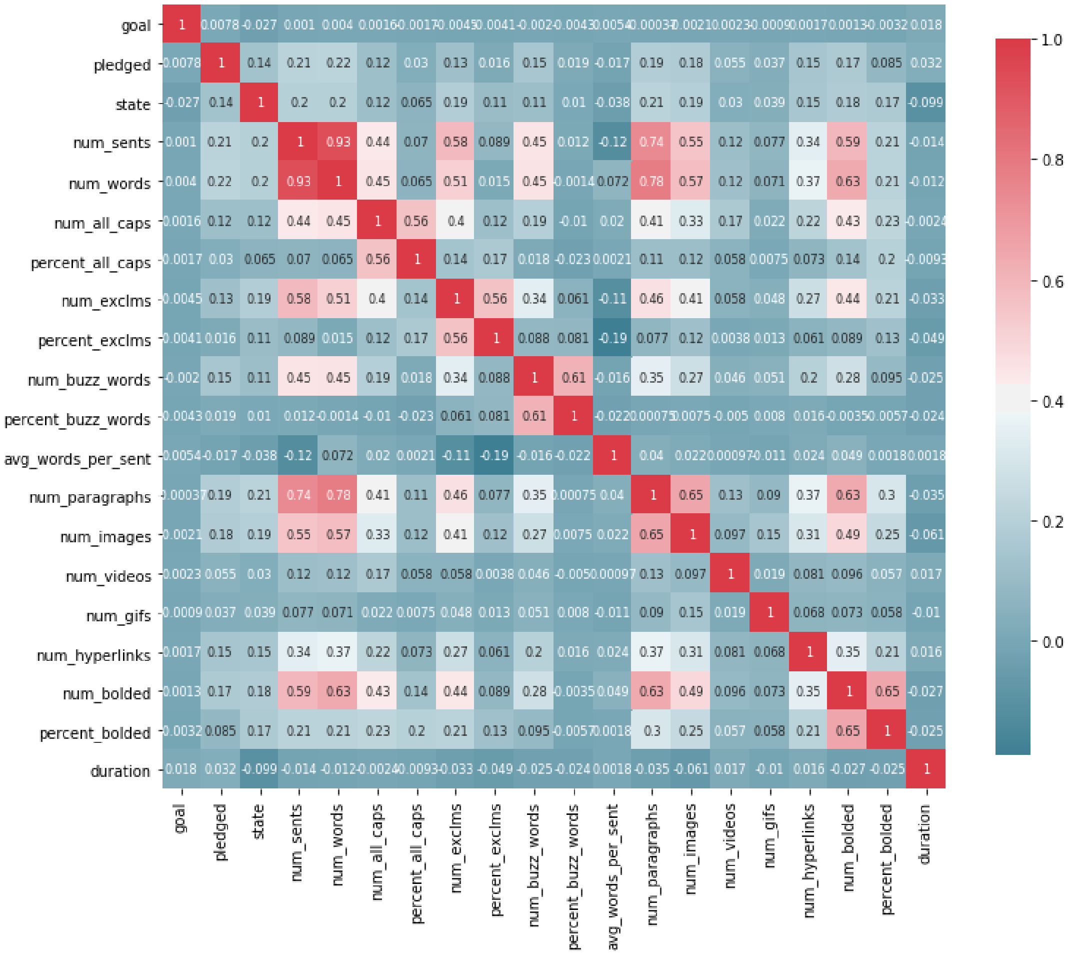
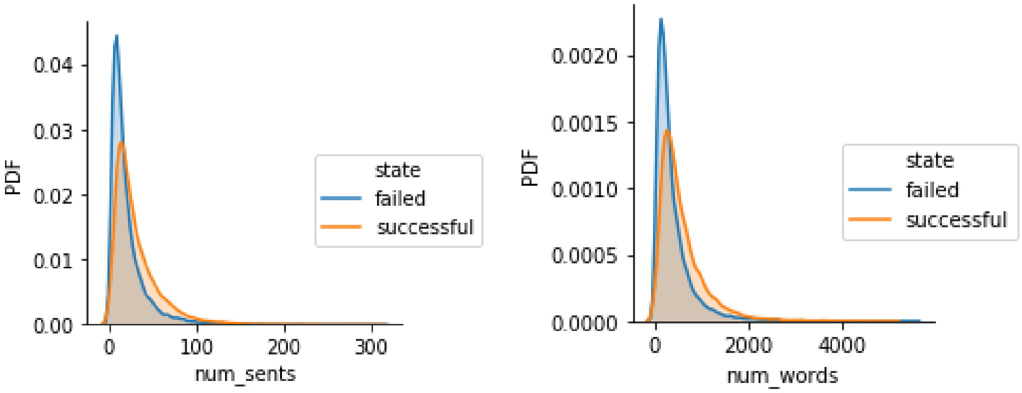
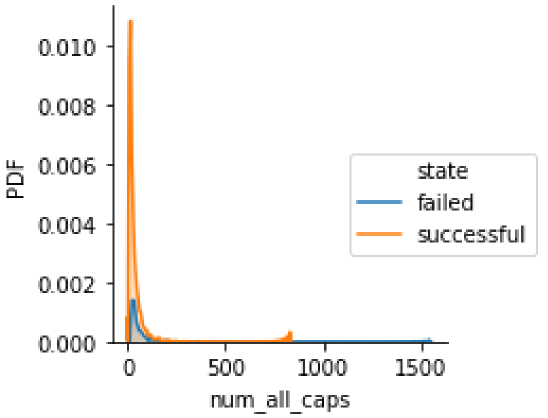
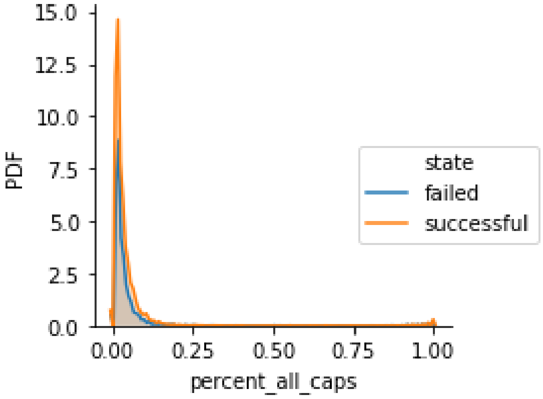
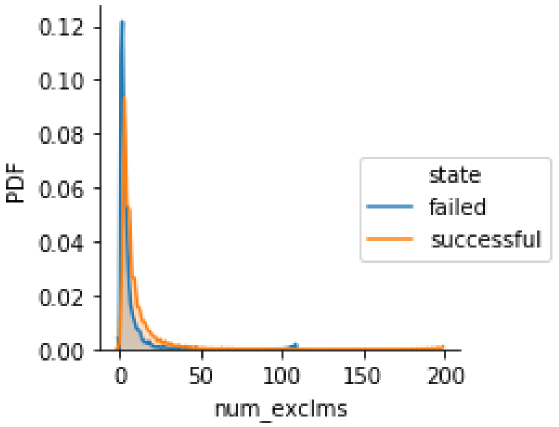


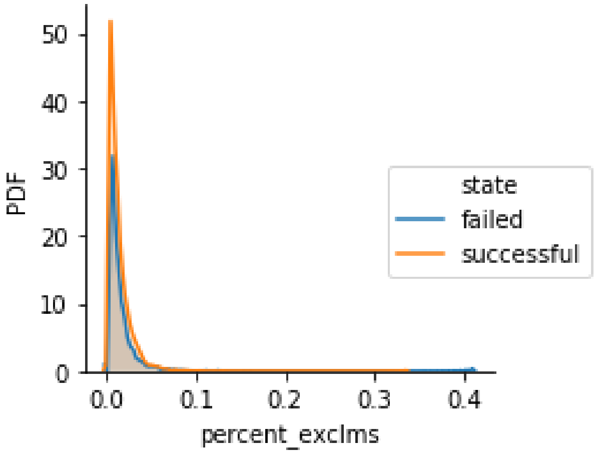
Fig. 3. Correlation map of different features

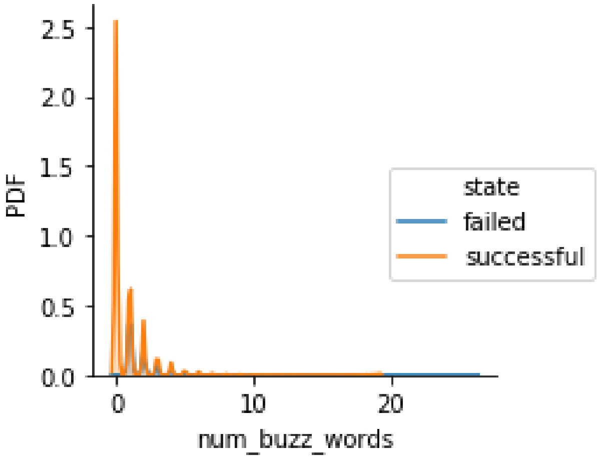


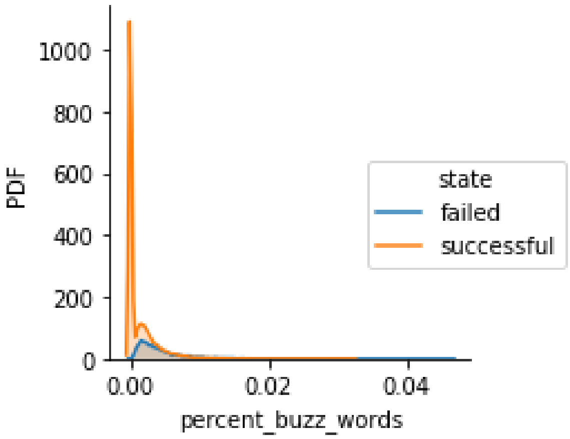

PDF
0.010
0.008
0.006
0.004
0.002
0.000
0
500
1000
1500
num_all_caps
state
failed
successful

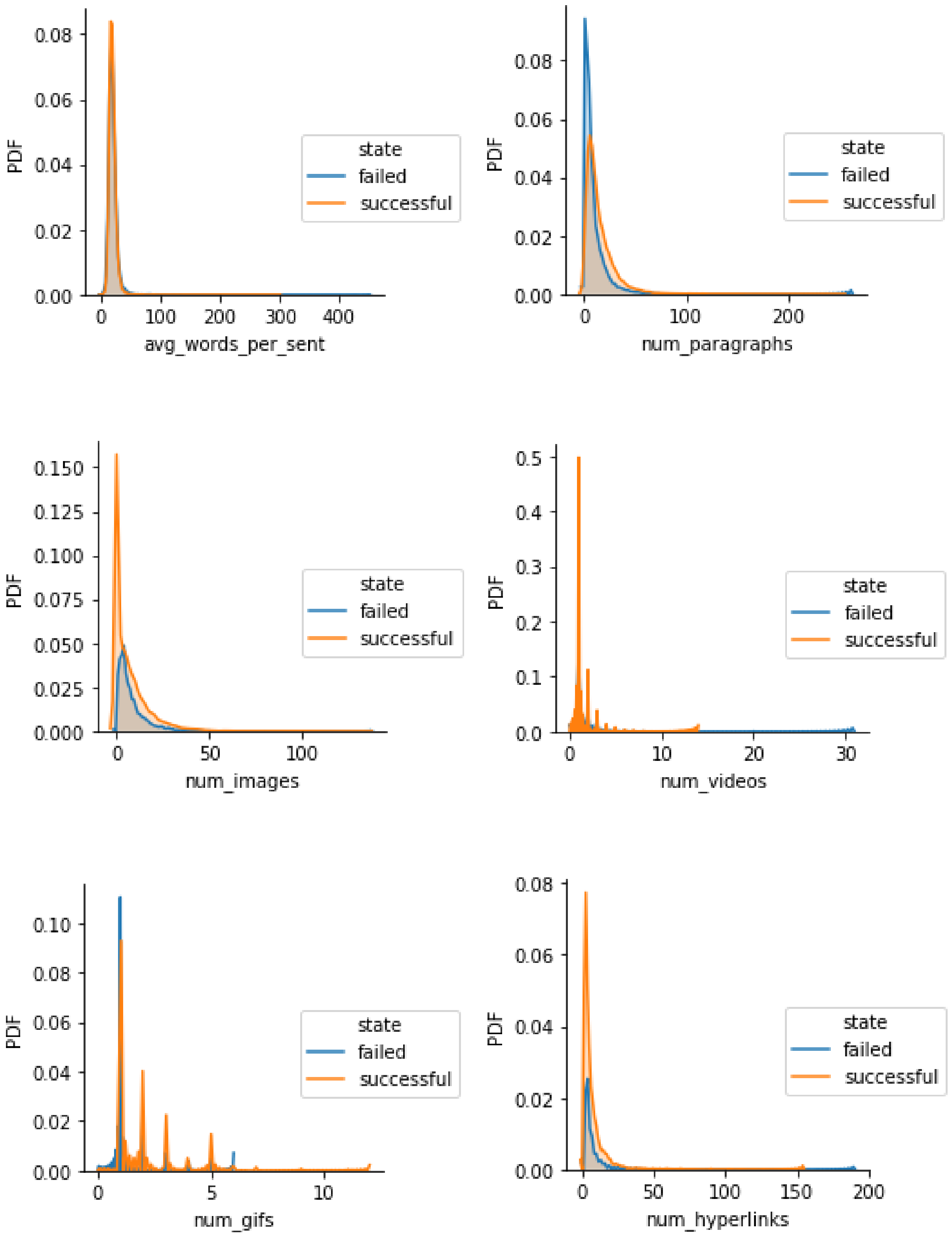

PDF
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.00
0.25
0.50
0.75
1.00
percent_all_caps
state
failed
successful


PDF
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
50
100
150
200
num_exclms
state
failed
successful


PDF
50
40
30
20
10
0
0.0
0.1
0.2
0.3
0.4
percent_exclms
state
failed
successful


PDF
2.5
2.0
1.5
1.0
0.5
0.0
0
10
20
num_buzz_words
state
failed
successful


PDF
1000
800
600
400
200
0
0.00
0.02
0.04
percent_buzz_words
state
failed
successful


PDF
0.00
0.02
0.04
0.06
0.08
0
100
200
300
400
avg_words_per_sent
state
failed
successful
PDF
0.00
0.02
0.04
0.06
0.08
0
100
200
num_paragraphs
state
failed
successful
PDF
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0
50
100
num_images
state
failed
successful
PDF
0.0
0.1
0.2
0.3
0.4
0.5
0
10
20
30
num_videos
state
failed
successful
PDF
0.00
0.02
0.04
0.06
0.08
0.10
0
5
10
num_gifs
state
failed
successful
PDF
0.00
0.02
0.04
0.06
0.08
0
50
100
150
200
num_hyperlinks
state
failed
successful

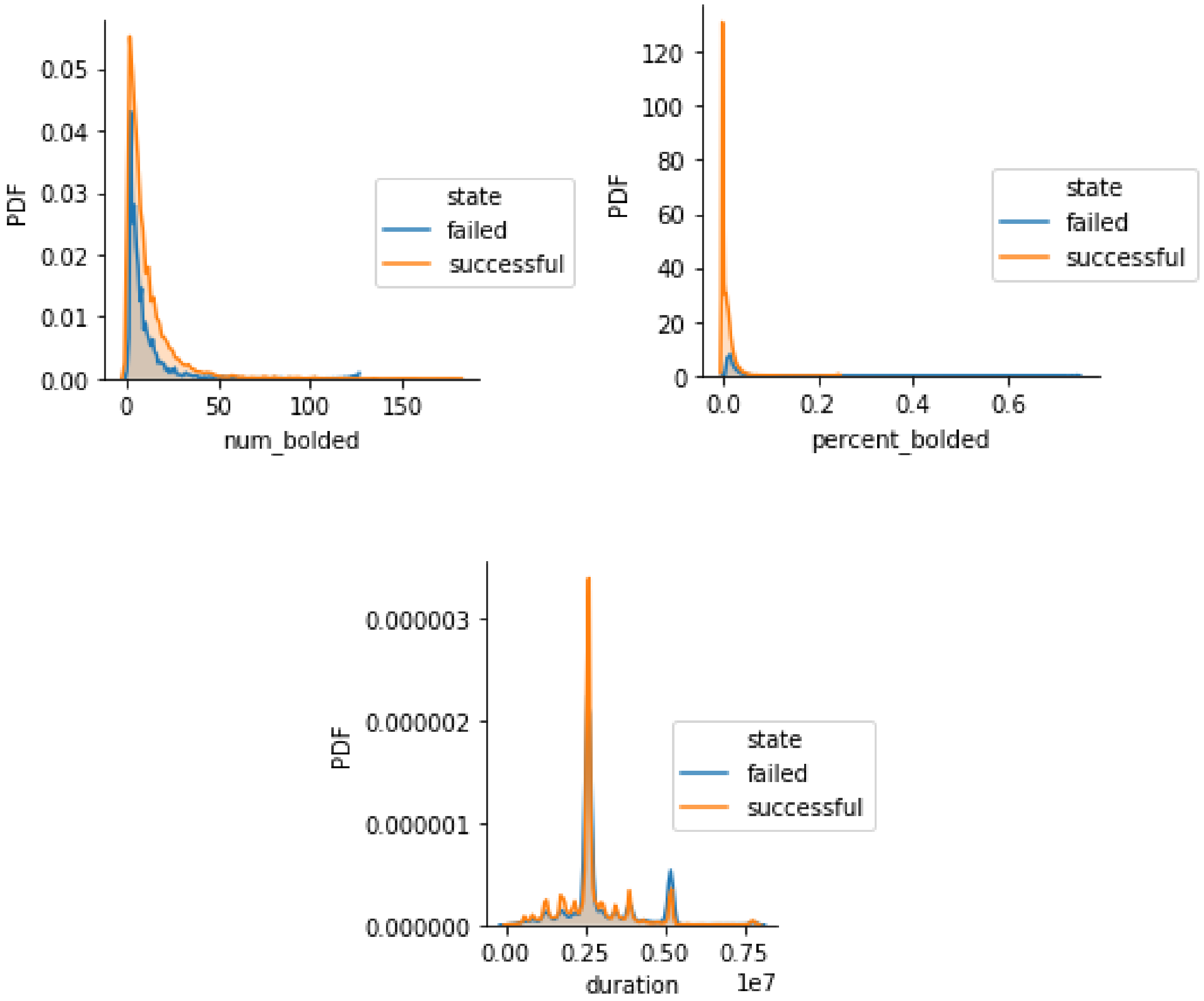


Fig. 4. Correlation analysis of features for successful and failed projects

From the analysis results, we can conclude that four features, avg_words_per_sent, num_videos, num_gifs, and duration, do not show any differences between successful and failed projects which will not facilitate the classification. Thus, they are safely deleted from feature data base.

The influences of different categories on the outcomes of the campaign projects are also analyzed as illustrated in Fig. 5. We can see that the categorical features can contribute the classification task as the successful rates for different categories are distinctly characterized.

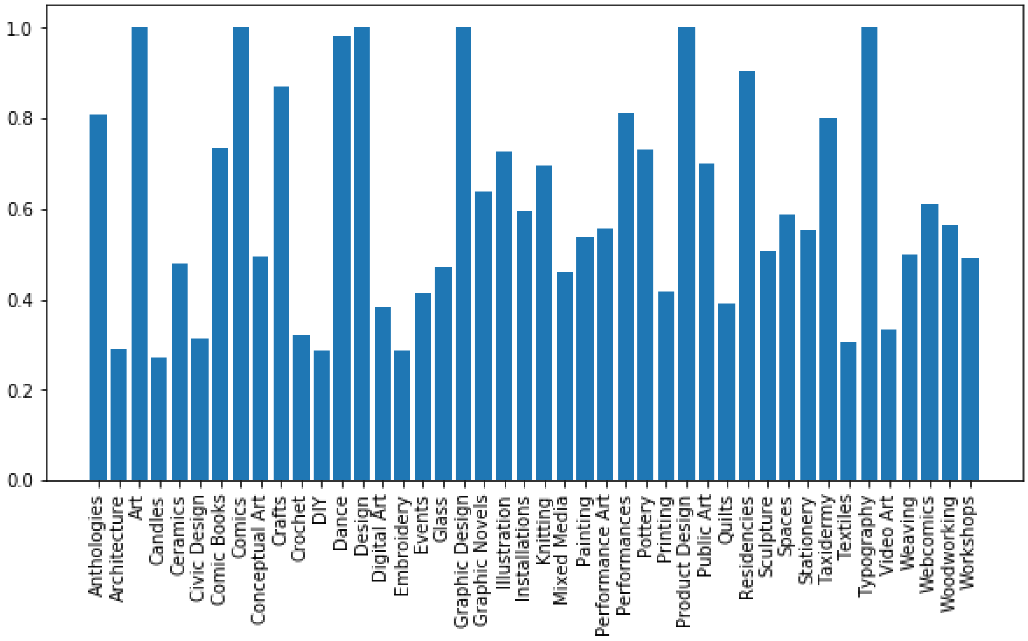


Fig. 5. Successful rates for different categories

3. Machine Learning Modeling

Since the xgboost algorithm is extensively utilized in classification problems and shows great performance in those tasks, this algorithm is first performed, and the logloss metric is used to measure the performance of this algorithm.

In the first model, we leave out the normalized text – not transfer it into vectors using like tfidf or counting frequency. Therefore, the final dimension of the train data set is (30941, 58), meaning there are 30941 instances and 58 features to train the xgboost model. There are two labels, successful and failed, to be classified. Consequently, this is a binary classification problem.

In the model train process, the cross-validation technique is utilized, specifically 5 folds being used to train the gradient boosted tree model. The learning curve of last round with respect to the log loss function is demonstrated in Fig. 6. We can see that the training and testing losses are 0.00921 and 0.009551, which proves the xgboost model is performing excellently even though the normalized texts are not transferred into vectors potentially helping improve accuracy. The confusion matrix is plotted in Fig. 7, showing the great accuracy of our modeling fed with our constructed features.

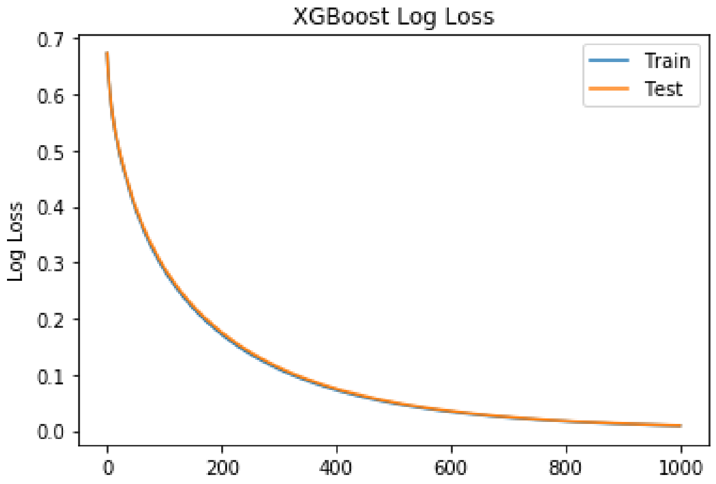


Fig. 6. Learning curve for training and testing data

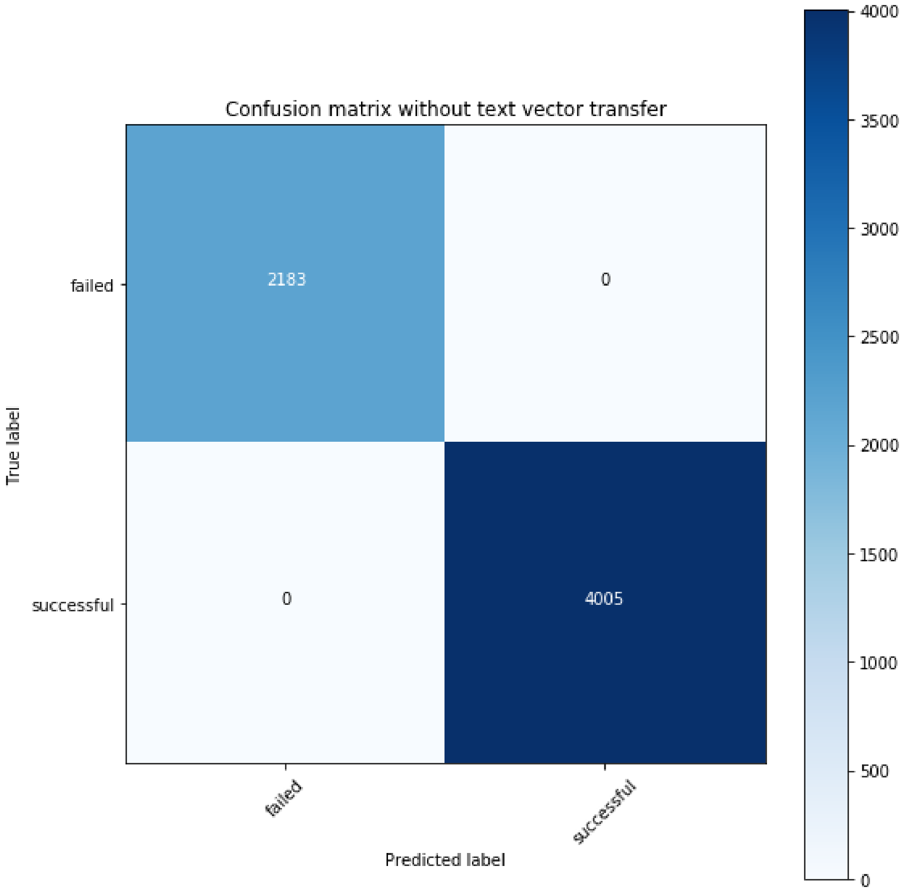


Fig. 7. Confusion matrix for testing data set