BackerTracker

Backers Count Prediction System for Kickstarter

BUDT 768T – Data Mining & Predictive Analytics

Group: 4
Drishti Jain, Nishi Morbia, Nisarg Patel, Priyank Shah, Nishit Salot

Contents

Executive Summary

Kickstarter is an American public-benefit corporation based in Brooklyn, New York, that maintains a global crowdfunding platform focused on creativity The company's stated mission is to "help bring creative projects to life". Kickstarter has reportedly received more than $1.9 billion in pledges from 9.4 million backers to fund 257,000 creative projects, such as films, music, stage shows, comics, journalism, video games, technology, and food-related projects.

People who back Kickstarter projects are offered tangible rewards or experiences in exchange for their pledges. This model traces its roots to subscription model of arts patronage, where artists would go directly to their audiences to fund their work. We wanted to develop a solution where we will be going to look at the different features of Kickstarter campaigns and how they correspond to the likelihood that a campaign will succeed, getting prediction of how many people will back the project as well as classify if the project would be a big hit!

That is why we created a model that predicts the count of backers a project can expect depending upon the parameters like goal (USD), deadline, project category and other necessary variables. And we name it **BackerTracker!**. We wanted to give creators a way to estimate their final funding amount raised sooner so that they can start thinking about how they can ramp up production and fulfillment even earlier. Additionally, we also determine how social-identity factors like race, color, gender is statistically important for the projects listed on the Kickstarter.

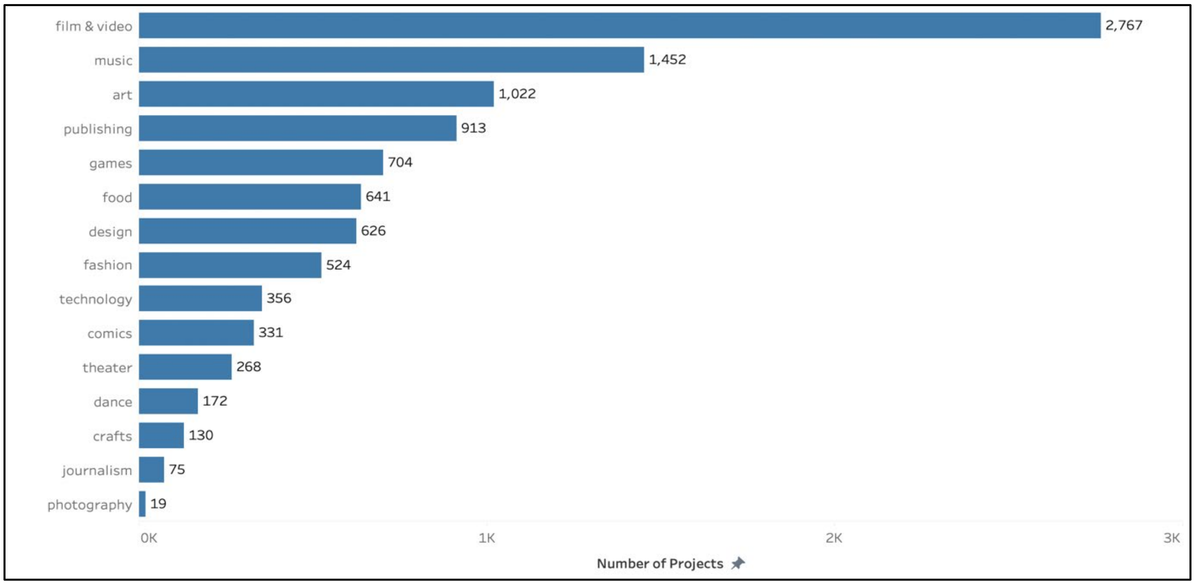
Predicting collective human behavior is a very difficult task because the causes at the individual level (reciprocal influences, groups of individuals with similar behavior) are often not directly recognizable from the systemic outcome.

Previous works uncovered a series of feedback mechanisms amplifying “microscopic”/individual inputs to the level of systemic transformations: multiplicative dynamics (Levy and Solomon, 1996), social percolation (Solomon et al., 2000), herding (Levy et al., 1994). Surprisingly, our findings for Kickstarter campaigns do not display such effects. We found that the success of a Kickstarter campaign depends on the arousal of project factors which can be inferred from the analysis of the statistical distribution of pledges.

Exploratory Data Analysis

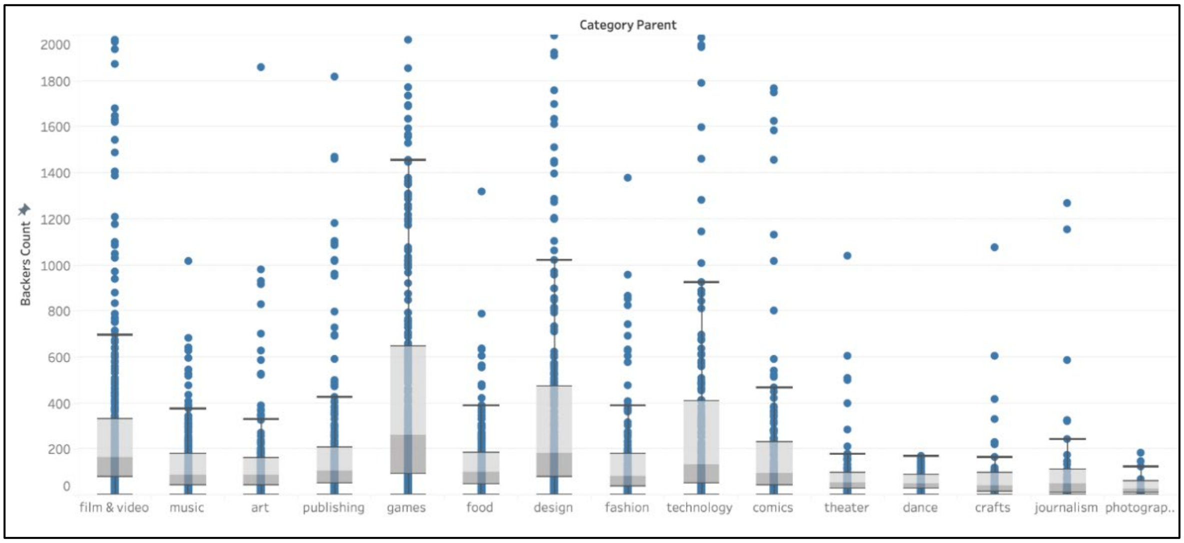
Our original dataset had 58 training variables and 3 target variables with 97420 data points for each variable. To significantly improve the work efficiency, we selected a small data set with 10000 data points for each variable. We will explore some visualizations of important parameters in our data set to see a clear picture before Data Processing

What types of projects are most popular?



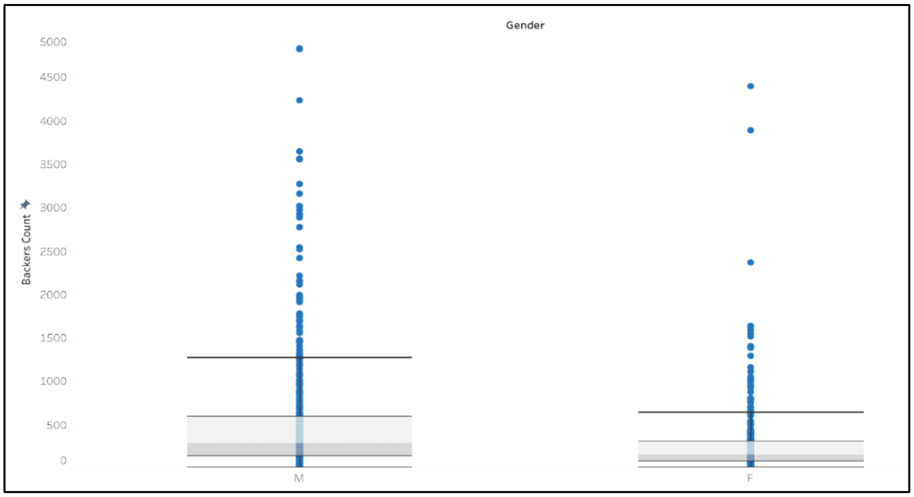
Film & Video appears to be the most popular project category with over 25% of all 10000 projects in the selected dataset and Photography the least popular.

Distribution of Backers for each project category?

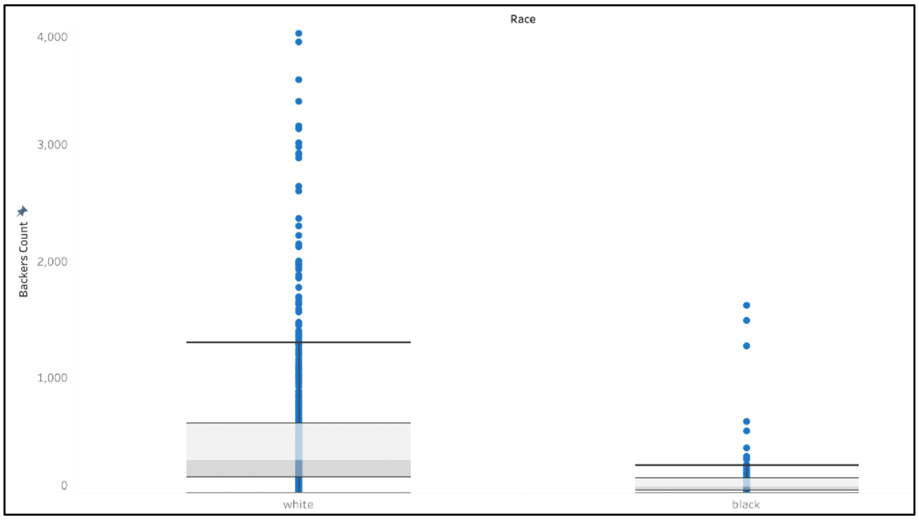


Games category has an incredibly high upper quartile and median. Although not nearly as high, Design and Technology also have relatively high upper quartile and median values as well. The average backers count for these two categories was lower than that of Comics and Film & Video, but they have higher median and upper quartile values.

How does social identity of creators affect the number of backers count?



It is evident from the above charts that projects that are created by Male creators tend to get more attention from funders and subsequently get more backers than that of females.



Similarly, comparing the backers count for white and black creators specifically, we can see a dramatic difference in the number of projects backed by public. Though, this shouldn't be the case, it is always important to understand how bias plays an important role in prediction models.

Data Processing

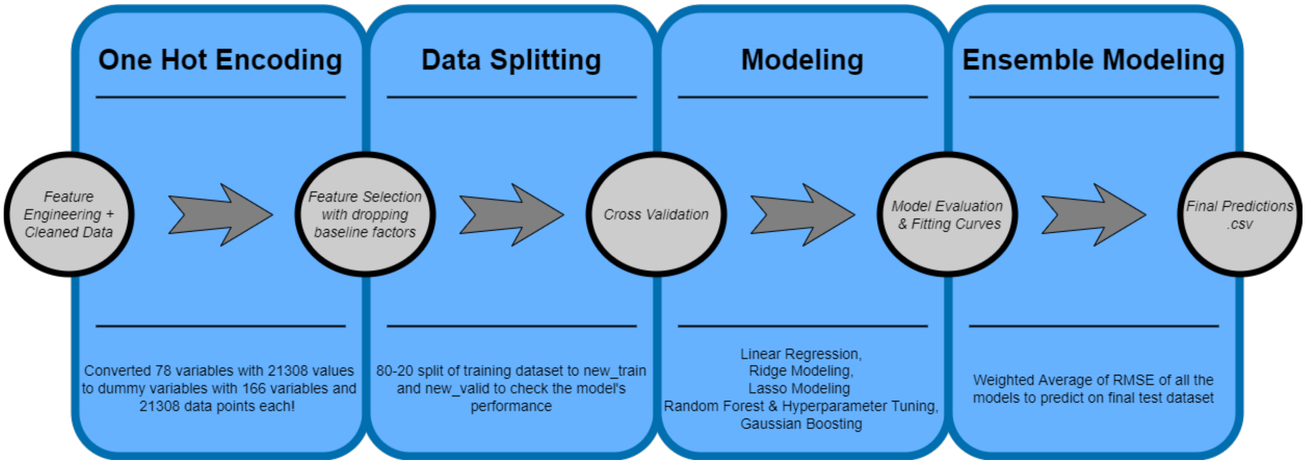
Getting known to our dataset, we can now further process the data with cleaning the data, text featurization, outlier treatment and engineering of features that we believe would be important parameters for the model to make better predictions. For more information, please refer to *Appendix: A*

To summarize, we:

- ✓ Replaced NULL values with appropriate values or categories
- ✓ Converted all strings uniformly to lower case to easily do featurization and text mining on them
- ✓ Binned several counters for easy usage in modelling techniques
- ✓ Created new flag variables to divide certain values
- ✓ Created new variables based on our understanding by dividing current values
- ✓ Created new variables by extracting data from other variables
- ✓ Trimmed variables based on requirements
- ✓ Split strings to have concise understanding and usability of variables
- ✓ Rescaled variables to have a normalized range for simplified usage

Modeling Process & Specification Comparison

Modeling Process


One Hot Encoding
Data Splitting
Modeling
Ensemble Modeling
Feature Engineering + Cleaned Data
Feature Selection with dropping baseline factors
Cross Validation
Model Evaluation & Fitting Curves
Final Predictions .csv
Converted 78 variables with 21308 values to dummy variables with 166 variables and 21308 data points each!
80-20 split of training dataset to new_train and new_valid to check the model's performance
Linear Regression,
Ridge Modeling,
Lasso Modeling
Random Forest & Hyperparameter Tuning,
Gaussian Boosting
Weighted Average of RMSE of all the models to predict on final test dataset

Model Specification

Model	Cross Validation	Control	RMSE
Linear Regression	-	-	389.4611
Ridge Regression	10-fold	-	394.5368
Lasso Regression	10-fold	-	394.9642
Random Forest	5-fold	mtry = 2, ntree = 1000	391.2901
Gaussian Boosting	5-fold	n.trees = 2300	407.4053
Ensemble Modeling (Weighted Average)	-	(Linear Regression * 0.20) + (Boosting * 0.20) + (Random Forest * 0.45) + (Ridge * 0.10) + (Lasso * 0.05)	388.2901

Instead of fitting curves, we will be comparing the models based on their RMSE performance; since for all our tree-based model, we have different trees – in that case it is not advisable to evaluate on the basis of fitting curves.

Model evaluation and selection methodology

Ensemble Modeling having highest performance, we select this predictor as our final model to work with the test data. From our evaluation we found that Random Forest and Boosting show different *important features*, implying that those models are capturing different aspects of the data. To get the final model, we ensembled different predictors based on majority voting. Boosting, Linear Regression and Random Forest are given larger weights due to their better performance.

Takeaways & Conclusion

Coming to the end of this project, we could analyze and support our findings on Kickstarter data We worked well together as a team since we were aware of our individual strengths and divided the work accordingly. We focused our efforts on feature engineering during the first few weeks which resulted in a good baseline model. One of the challenges we faced was deciding whether to continue our efforts on improving the classification model we had already built or to switch to regression modeling. Changing the target variable close to the contest submission was a risk but we decided to take it up as a challenge. We believe that we made a very good regression model given the time constraints. If we had more time on the project we would have tried Decision Tree Regressor or spent more time to improve our final Ensemble model. Our advice for the future batch of students would be to focus on feature engineering even if it seems time consuming.

Appendix: A

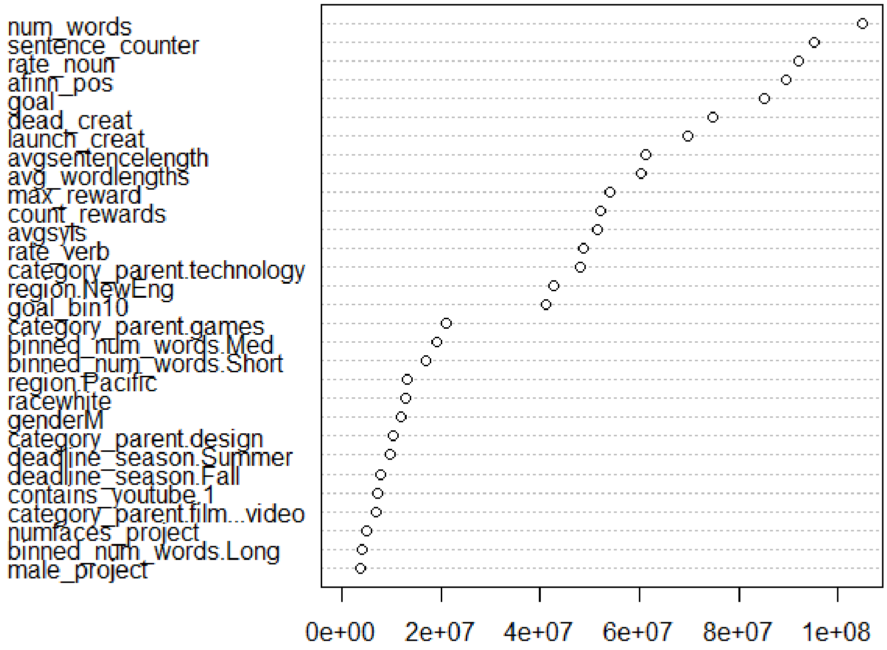
Sr. No.	Variables	Changes / New Variable Description
1	**Goal**	Normalize the Goal variable to a range from 0 to 1
2	**location_slug**	Extracted and cleaned the city names from the combination of city-state
3	**category_parent**	Changed the values to lower scale and Replaced NAs with "other" (factorized)
4	**category_name**	Changed the values to lower scale and Grouped to "other" if frequency less than 500 (factorized)
5	**smiling_project**	Replaced NA with mean and Normalize the values to a range from 0 to 1
6	**smiling_creator**	Replaced NA with mean and Normalize the values to a range from 0 to 1
7	**minage_project**	Replaced NA with mean and Normalize the values to a range from 0 to 1
8	**minage_creator**	Replaced NA with mean and Normalize the values to a range from 0 to 1
9	**maxage_project**	Replaced NA with mean and Normalize the values to a range from 0 to 1
10	**maxage_creator**	Replaced NA with mean and Normalize the values to a range from 0 to 1
11	**tag_names**	Replaced NA with "None" and separated the tagnames, taking only first image tag
12	**num_words**	Normalize the values to a range from 0 to 1
13	**sentence_counter**	Normalize the values to a range from 0 to 1
14	**Avgsentencelength**	Normalize the values to a range from 0 to 1
15	***first_quartile***	To get the atmost 25th percentile values from the data
16	***third_quartile***	To get the atleast 75th percentile values from the data
17	***Iqr***	To get the values between 25th & 75th percentile range to remove outliers

18	***binned_num_words***	Binned the values within 5 groups of "VShort", "Short", "Med", "Long", "VLong"
19	***flag_num_words_long***	Flag for records having number of words greater than the mean value.
20	***binned_sentence_counter***	Binned the values within 5 groups of "VShort", "Short", "Med", "Long", "VLong"
21	***category_name_freq***	To get the frequency of occuring category for further processing
22	***deadline_year***	Extracted "Year" from deadline variable
23	***flag_female***	Flag for records having either a female creator or not
24	***Gender***	Processed gender of creator using flag_female
25	***Race***	Race of the creator extracted with "predict_race" function. Replaced NAs with "Unknown"
26	***deadline_month***	Extracted "Month" from deadline variable
27	***deadline_season***	Extracted "Season" from deadline_month, using case...when
28	***State***	Extracted "State" from location_slug
29	***flag_color_foreground***	1 if foreground color of the image is either black, white or grey otherwise 0
30	***flag_color_background***	1 if background color of the image is either black, white or grey otherwise 0
31	***dead_creat***	Difference between deadline and created in DAYS
32	***launch_creat***	Difference between Launch and created in DAYS
33	***launch_deadline***	Difference between Launch and deadline in DAYS
34	***goal_bin4***	Groups the goal values into 4 equally distributed bins
35	***goal_bin10***	Groups the goal values into 10 equally distributed bins
36	***flag_overall_pos***	Flag for records who have Positive AFINN Sentiment score greater than Negative AFINN Sentiment score

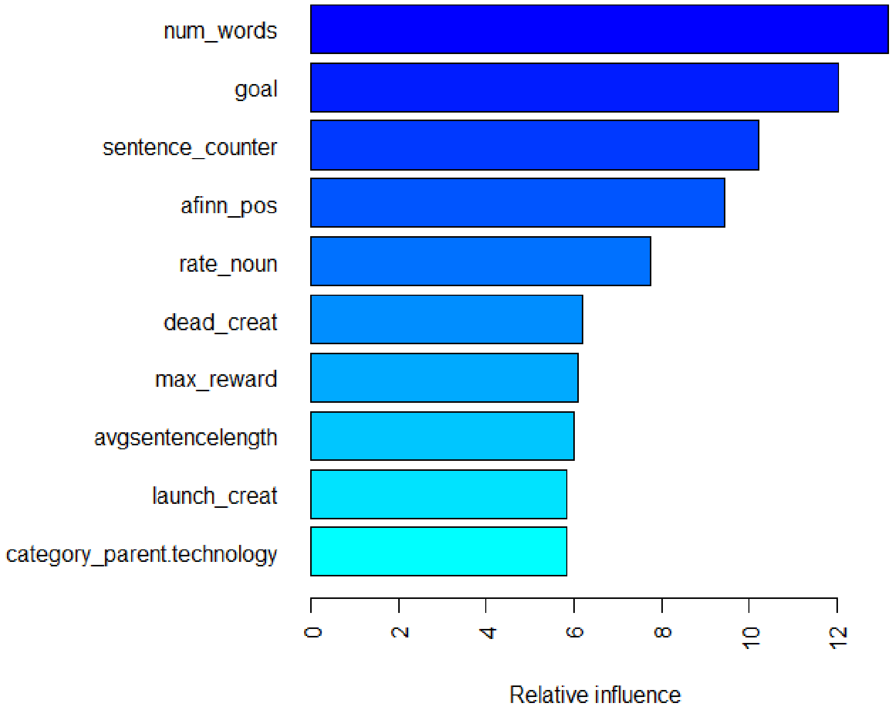
37	***rate_adv***	Rate of Adverb from total number of words for each record
38	***rate_noun***	Rate of Noun from total number of words for each record
39	***rate_adp***	Rate of Adpositions from total number of words for each record
40	***rate_prt***	Rate of particles from total number of words for each record
41	***rate_det***	Rate of determiners from total number of words for each record
42	***rate_pron***	Rate of pronouns from total number of words for each record
43	***rate_verb***	Rate of verbs from total number of words for each record
44	***rate_num***	Rate of numerals from total number of words for each record
45	***rate_conj***	Rate of coordinating conjunctions from total number of words for each record
46	***rate_adj***	Rate of adjectives from total number of words for each record
47	***count_creator_id***	Count of all creators in dataset
48	***quartile_rank_creator_id***	Groups the creators in 4 bins depending upon the number of projects they submitted
49	***ntile_rank_creator_id***	Groups the creators in 10 bins depending upon the number of projects they submitted
50	***top_creator_ids***	Top Creator Flag if the creator is in 10th bin of "ntile_rank_creator_id"
51	***all_rewards***	Splitted list of all rewards
52	***count_rewards***	Count of rewards from "all_rewards"
53	***min_reward***	Minimum collaboration amount (USD) to get a lowest reward enlisted
54	***max_reward***	Minimum collaboration amount (USD) to get a highest reward enlisted

Appendix : Variable Importance

Random Forest



Gaussian Boosting


num_words
goal
sentence_counter
afinn_pos
rate_noun
dead_creat
max_reward
avgsentencelength
launch_creat
category_parent.technology
Relative influence