SUMMARY REPORT

Kickstarter is an American crowdfunding platform which has the mission to 'help bring creative projects to life'. It was founded in 2009 and since then, has been on the rise with more and more projects being posted. For our project, we first focus on determining which factors can determine whether the project will be successful or not. Secondly, we develop a clustering model using predictor variables to group projects together and based on these variables, we explain the characteristics of each cluster.

Before we begin with our analysis, it is important for us to clean the data and bring it in an appropriate form for further analysis to be conducted on it. Hence, we first load the dataset. Then, we look for any missing values. Since 'launch_to_state_change_days' had a lot of missing values, we remove this column. For the rest of the missing values, we simply remove the values rather than columns because those columns are important for us. Next, since we must run classification models, it is important to dummify our categorical variables. After this, we consider only those rows that have the 'state' as successful or failed since that is what we will be using for further analysis.

Once data is cleaned, we specify the predictor and target variables and then we start off with feature selection process, which is basically used to identify the most important features that should be included in the classification models. Note that I excluded all variables that had either:

1. Unique values, or
2. Would be realized once the project was created

Because unique values cannot be classified directly and were not important predictors intuitively, and the instructions of the project mentioned to not include variables that would be realized after prediction starts. After data cleaning, I ran three different features selection techniques i.e., random forest, RFE, and LASSO. Out of all 3, I found LASSO as the most optimal technique to use because of its reliability and ease of interpretability. LASSO automatically identifies which features are not beneficial by reducing their coefficients to 0. Once the feature selection technique was finalized, it was important to do hyperparameter tuning to come up with the optimal hyperparameters for our LASSO model. After running a loop in python, I was able to deduce that a 0.01 value of alpha was the optimal value.

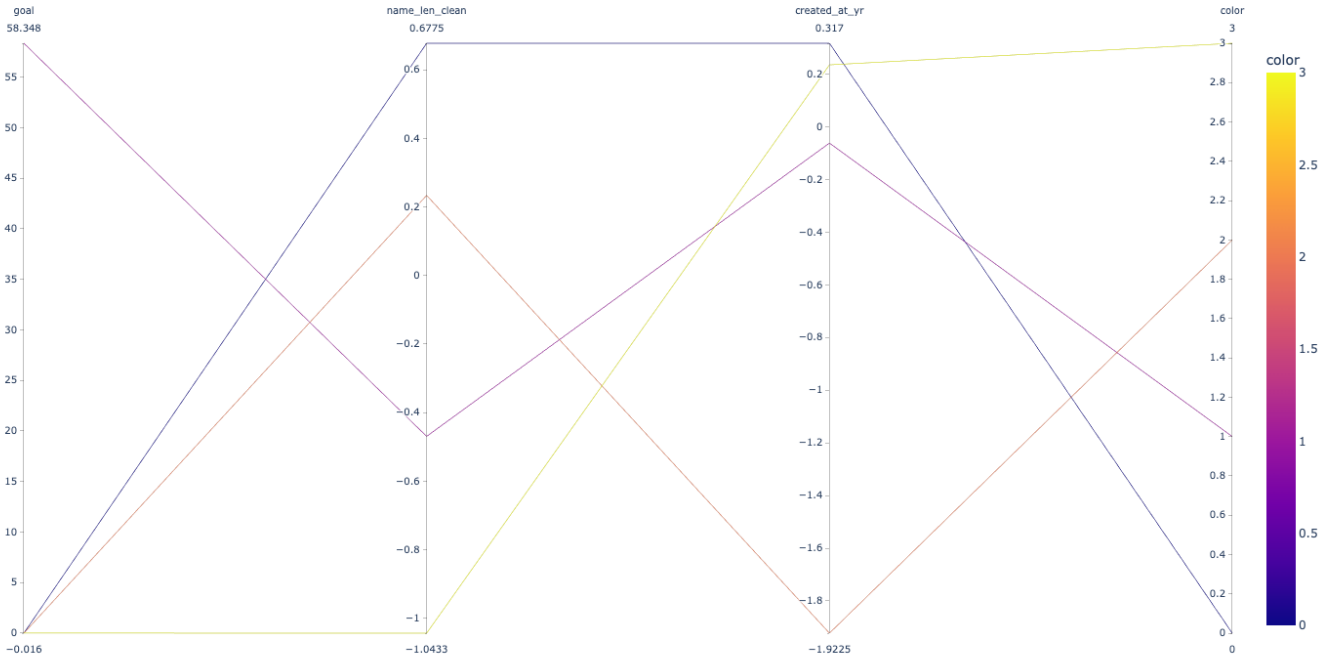
After this, the LASSO model was run again at alpha 0.01 to find out the predictors to use in our classification models. After getting a list of predictors, it was important to check if there was any correlation between those (this can be done on numerical predictors only). Initially, I was confused whether *pledged* and *usd_pledged* will be realized after project creation, hence, I included them in the correlation matrix too. I was able to deduce that pledged, usd_pledged, and backers_count were all highly correlated, hence I opted for backers_count only. Similarly, name_len and name_len_clean were also highly correlated so I chose name_len_clean because that was a more accurate predictor.

Finally, after choosing all the relevant predictors, I tried out different classification techniques with my base model being the one with logistic regression, since it was the easiest to execute out of all and gave me better insights and model accuracy. However, due to low accuracy and F1 score, I moved on to random forest because it reduces overfitting and improves accuracy. This was also evident within my results, because the accuracy with random forest improved by roughly 7%. However, the results were still not satisfying. Hence, I used gradient boosting technique to come up with another alternative solution and see how the results varied. Fortunately, the model accuracy increased by a further 3.5% and F1 score improved by roughly

2.5%, which are both indicators of model performance. Consequently, the accuracy and F1 score was satisfying enough for the current predictors if I used gradient boosting technique.

For part 2 in which we had to form clusters, I initially started off by choosing all the relevant numerical predictors since clustering does not work well on categorical variables. I found out the optimal number of clusters for Kmeans using the elbow method to pick the optimal number of clusters. A plot was created to interpret the data points and the cluster in which they would lie.

As per the below graph, if a datapoint has a low value of goal, high value of created_at_yr and high value of name_len_clean, then this datapoint will lie in cluster 0.


goal
name_len_clean
created_at_yr
color
color