Kickstarter Data Modeling Project Report

Diwei Zhu

Part 1. Classification model

Data Cleaning - We only focus on classifying whether the project is successful/failed. Thus I dropped the rows of which the state is “live”, “canceled”, or “ suspended”. Then, I replaced “successful” with 1 and “failed” with 0 to make the target variable numerical. The same boolean/object-to-numerical transformation is applied to ‘staff_pick’, ‘spotlight’, and ’disable_communication’. By verifying the correlation between variables, I noticed that the correlation between ‘spotlight’ and the target variable is 1. Therefore, I excluded ‘spotlight’, or the accuracy score of my model would be abnormally close to 1.

Many columns with numerical contents in the imported data frame are actually in string datatype. To correct this, I transferred data in columns like ‘goal’, ‘pledged’, ‘backers_count’, etc. from string to float. Also, categorical columns like ‘category’, ‘deadline_weekday’,‘currency’, etc. were dummified.

The last part of data cleaning is dropping columns and NAs. I dropped ‘name’ and ‘project_id’ because I believed they are irrelevant; dropped ‘spotlight’ as suggested above; dropped ‘pledged’, ‘staff_pick’, ‘backers_count’, ‘static_usd_rate’, ‘usd_pledged’ because, when constructing this model, we need only focus on the status at the time when the projects are freshly launched; dropped timestamp columns, because the time gap information (e.g., ‘create_to_launch_days’) is already given in the dataset; dorpped the ‘launch_to_state_change_days’ for it contains too many Nas; dropped ‘name_len’ and ‘blurb_len’ because they can be replaced by ‘name_len/blurb_len**_clean**’. Lastly, I dropped 5 rows that contain NA values.

Feature Selection: With the cleaned dataset, I ran two feature selection methods with a 7:3 training-testing split. The first method is Lasso. With alpha=10000 decided by comparing

MSE, the Lasso model returned coefficients indicating that ‘goal’, being the only feature that has a non-zero coefficient, is the only viable feature for the classification model.

In order to discover more effective features, I ran the Random Forest feature selection, which returned the feature importance table on the right. Setting the importance threshold to 0.35, I have 9 features as the predictors (in the red square).

	predictor	feature importance
0	goal	0.088978
20	create_to_launch_days	0.053669
2	name_len_clean	0.043223
19	launched_at_hr	0.038082
13	created_at_day	0.037606
77	category_Web	0.036673
15	created_at_hr	0.036577
17	launched_at_day	0.036504
21	launch_to_deadline_days	0.036486
9	state_changed_at_day	0.034348

Model Selection: I ran the Random Forest model that returned an accuracy score = 0.7088823. I also ran the KNN model, where the n_neighbors = k, with k ∈ (1,50). All KNN models had worse performance than the Random Froest model (accuracy ~ 0.65).

Lastly, I ran the Gradient Boosting model (GBT) that returned an accuracy score = 0.7169571. Since the GBT had the best performance among these models, I chose this model as my final classification model.

```
# FINAL MODEL (GBT with 4 features) ____________________________
# GBT
gbt = GradientBoostingClassifier()
model_gbt = gbt.fit(X_train1, y_train1)
y_test_pred_gbt = model_gbt.predict(X_test1)

acc_gbt = accuracy_score(y_test1, y_test_pred_gbt)
print("Accuracy score of gradient boosting algorithmn:", acc_gbt)

# Accuracy score = 0.7169570760730982
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.
ed when a 1d array was expected. Please change the shape of y to (n_
  return f(*args, **kwargs)

Accuracy score of gradient boosting algorithmn: 0.7169570760730982
```

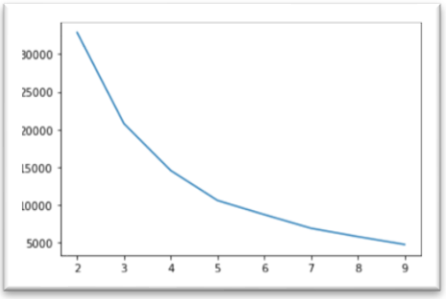
Part 2. Clustering model

Data Cleaning: The data cleaning part of the clustering model is similar to the previous part, except for the dropping of‘ pledged’, ‘staff_pick’, ‘backers_count’, ‘static_usd_rate’, and ‘usd_pledged’, because we are not focus only on the time where the projects are launched. The other steps (numerifying and dummifying variables, dropping irrelevant/collinearity columns and NA rows) are the same. Notably, I dropped ‘pledged’ and kept ‘usd_pledged’ because the former can be replaced by the later, and is less standardized.

Feature Selection: It is unwise to include all features. I again ran the Random Forest features selection based on the expanded set of features. As suggested by the result table to the right, setting the importance threshold to 0.05, 'usd_pledged', 'backers_count' and 'goal' are features with influential powers.

	predictor	feature importance
5	usd_pledged	0.258350
3	backers_count	0.237591
0	goal	0.119599
2	staff_pick	0.038128
24	create_to_launch_days	0.020435
25	launch_to_deadline_days	0.017432
6	name_len_clean	0.015654
81	category_Web	0.012966
17	created_at_day	0.012674
19	created_at_hr	0.012423

Determining Clustering Layers: I used Elbow method and Silhouette method to find the best number of clustering layers. The graph depicted by Elbow indicates that K=3 or K=5 can be ideal numbers of layers. According to the result from the Silhouette method, K=3 returned the highest average score, indicating that 3 cluster layers are optimal.




```
Silhouette result
K = 2 : 0.9686575652159152
K = 3 : 0.9697758666613713
K = 4 : 0.9401040081229042
K = 5 : 0.9397815817174344
K = 6 : 0.88804002448831
K = 7 : 0.8915101592600532
K = 8 : 0.8903941511299167
K = 9 : 0.867408931868203
```

Clusters: With KMeans model, I divided the observations into three clusters. The number of observations in each cluster is 15614, 66, and 5 respective, with differenent centres in terms of each selected feature (as shown in the below table). All clusters share the similar usd_pledged level, with Cluster 1 having a slightly lower level and Cluster 3 having a slightly higher level. As for goal, observations in Cluster 1 have the highest goal level, observations in Cluster 2 have lower goal level, while those in Cluster 3 have the lowest goals that are close to 0. Cluster 1 and 2 have similar backers_count levels, while Cluster 3 has very high backers_count level that distinguish the observations in Cluster 3 from those of the other two Clusters.

	Cluster	Number of observations	usd_pledged_centre	goal_centre	backers_count_centre
0	1	15614	-0.047698	11.298230	-0.184686
1	2	66	-0.039336	9.317316	-0.150923
2	3	5	-0.016015	0.070463	49.081106