Hisham Salem 261003138 12/05/2021

Introduction

Kickstarter is a crowd-funding site whose main purpose is to turn ideas become projects. Each project is built and crafted by the individual behind it, who may then launch it to the community once it is complete. Every project creator establishes a financial target and deadline, and if the project generates sufficient interest, individuals can give funds to help it succeed. This project is separated into two sections: classification and clustering, both of which will be discussed in depth.

Data Preprocessing

I started by taking a clean dataset of 18568 observation and 45 columns and kept only the observations that had a project state that was either successful or failed, this reduced the number of observations to 15685. The following tables shows the predictors that were dropped:

Predictor	**Reason Behind Removal**
Project_ID, name	Used for identification, have no effect on the model
Goal, static_usd_rate	A new variable was created that multiplies both for a more scalable comparison
Pledged	After the project was submitted
State	After the project was submitted
Disable_communication	After the project was submitted
Currency	Highly correlated to country
Deadline, State_changed_at, created_at, launched_at	Their respective dummies were used
Backers_count	After the project was submitted
Spotlight	After the project was submitted
Staff_pick	After the project was submitted
Name_len, blurb_len, Name_len_clean, blurb_len_clean	They were removed due to feature selection since they degraded the model's performance (they were kept for classification)

State_changed_at_weekday, month, day, year, hour	After the project was submitted
Launch_to_state_change_days	After the project was submitted
Usd_pledged	After the project was submitted (for classification task)

After dropping the aforementioned predictors, I made dummy variables that were created for the following categorical variables: created_at_yr,deadline_yr, deadline_month, created_at_month, deadline_weekday, created_at_weekday, country, category. I then involved the removal of anomalies using the Isolation Forest algorithm. The Isolation Forest models was built for the target variable and for the predictors respectively, reducing the total amount of observations to 13793. For the clustering portion. Anomaly detection was performed on my 3 predictors. This will then further reduce the number of input variables; feature selection was also performed for classification tasks by utilizing Lasso and the Logistic Regression Feature selection technique.

Model Selection

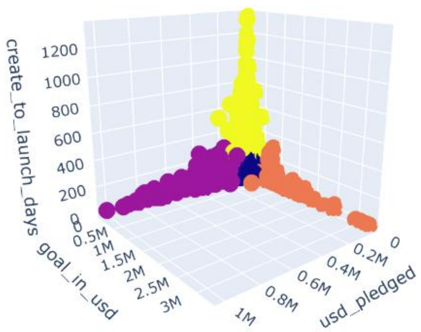
For classification tasks, the optimal model was selected based on the least MSE and highest accuracy using cross-validation. Different models were tested such as KNN, ANN, Random Forest, and Gradient Boosting. However, Gradient Boosting was selected as the best model despite the higher accuracy of Random Forest, the Gradient Boosting model is more robust to overfitted thus I believe that Gradient Boosting was champion model. The optimal hyperparameters were selected by using grid search for the two best models which are Random Forest and Gradient Boosting. In order to estimate GridSearch running time I have to calculate (Count of each list in parameters multiplied by each other ie A(1,2,3) and B(1,2) would equal 3*2) *Cross Validation/ Number of Processors*running time of model. To improve the speed of the grid search hyperparameter tuning I decided to use a binary selection approach to best find the optimal hyperparameters without having my grid search model run for 16+ hours. The tables below represent the results of my 4 models and 2 top classification model with extensive tuning:

Technique	KNN	ANN	Random Forest	Gradient Boosting
CV Score	0.7139	0.699	0.756	0.758

Top 2 Models with Extensive Grid Search tuning	Gradient Boosting Classification (Champion Model)	Random Forest Classification
Max features	Sqrt	Auto
Max depth	6	25
Minimum samples split	18	11
Min Samples leaf	11	1
N_estimators	117	108
Learning Rate	0.15	NA
Criterion	Friedman_MSE (Default)	Entropy

From a commercial standpoint, the prediction model requires more data, such as an industry study of each category, to produce a higher and more dependable model. The classification model may be suggested since it has a 75.6 % accuracy in classifying the condition of a project.

For my clustering model, it was performed on USD_pledged, Goal in USD and Create to lunch days. After dropping NA rows and removing anomalies using Isolation Forest, K-Means Clustering was selected, where using elbow method the optimal number of clusters was 4. As we can see from the 3D scatter using Plotly, it is easy to visualize our 4 clusters. Purple cluster represents creators who had low to intermediate goals and time to launch but exceeded their own expectations and ended up with a high USD_pledged. Yellow Cluster represents creators who took too long to launch with low goals and ended with low USD_pledged. Orange Cluster represents creators with intermediate-high goals, low-intermediate time to launch and ended with low USD_pledged. Finally, the blue cluster represents creators with low goals, launched quickly and ended with very low USD_pledged.



The model was then tested using silhouette score (0.775), Calinski-Harabasz score (8551.06) and p-value (1.11e-16), meaning that our data was well clustered.