Shaan Kohli

INDIVIDUAL PROJECT

Introduction

Kickstarter is an American public-benefit corporation that maintains a global crowdfunding platform focused on creativity and merchandising to help bring creative projects to life. The goal of this project was to formulate regression and classification models to predict the amount of US dollars pledged towards a project (**usd_pledged**) and whether the project is successful or a failure (**state**), respectively.

Data-Preprocessing

The first step in building models for both the regression and classification tasks was choosing predictors among the 45 variables provided. Predictors were first filtered based on intuition and then filtered through further mathematical analysis. The table below summarizes the variables removed as well as the reason:

Predictors Removed	
Predictor Name	**Reason**
Project_id	No predictive power of usd_pledged or state. Mainly used for identification
Name	No predictive power of usd_pledged or state. Mainly used for identification
State	Removed for Regression Task as it is realized after time of launch
Disable Communication	Realized after the launch
Currency	High correlation with Country
deadline, state_changed_at, created_at, launched at	Removed to take the respective month, day, year, hour counterparts (for dummy variables)
staff_pick	Realized after the launch
backers_count	Realized after the launch
static_usd_rate	Removed, once columns values multiplied with goal to give all USD amounts
usd_pledged	Removed for Classification Task as it is realized after time of launch
pledged	High Correlation with usd_pledged and reflect same thing and thus removed
spotlight	Realized after the launch
name_len	name_len_clean was kept instead
blurb_len	blurb_len_clean was kept instead
state_changed_at_weekday	Realized after the launch
launched_at_weekday	Realized after the launch
state_changed_at_month	Realized after the launch

state_changed_at_day	Realized after the launch
state_changed_at_year	Realized after the launch
state_changed_at_hr	Realized after the launch
launch_to_state_change_days	Realized after the launch

The next step involved selecting only successful and failed states for both regression tasks. Of the 18568 observations, 15685 remained (loss of 2883 rows). To preserve the greatest amount of data for model training, launched_to_state_change_days was eliminated due to the great amount of NAN values. Afterwards, rows containing Na's were eliminated (14214 rows remained) thus preserving more rows. The next part of data preprocessing involved eliminating outliers within the dataset for numerical predictors. This process involved taking the z-score and eliminating those that more than 3 standard deviations away (12953 rows remained). To resolve the issue of categorical variables and code efficiency, potential discrete predictors were dummified in a particular way. Due to the large extent of unique categories for a specific predictor, categories were binned in specific ranges. The predictors deadline, created and launched were binned in terms of weeks (i.e. 0-7, 7-14, 14-21, and 21+ days) and hours (i.e. 0-8 hours, 8-16 hours, 16+ hours during a given day) for the day and hour variables. It was at this point a correlation test was performed to evaluate any collinearity between predictors. The variable **Currency** was correlated with **Country**. The latter was dropped.

Summary Statistics of Variables

The following table shows the summary statistics for variables that were not binned and are numerical in nature:

Variable	**Count**	**Mean**	**Std**	**Min**	**Q1**	**Median**	**Q3**	**Max**
goal	12953	45701.39	149435.47	1	3689.73	12000	40000	3300000
name_len_clean	12953	5.12	2.42	1	3	5	7	12
blurb_len_clean	12953	13.14	3.01	4	12	13	15	22
deadline_month	12953	6.83	3.35	1	4	7	10	12
deadline_yr	12953	2014.86	1.02	2012	2014	2015	2016	2017
created_at_month	12953	6.46	3.26	1	4	7	9	12
created_at_yr	12953	2014.69	1.02	2011	2014	2015	2015	2017

launched_at_month	12953	6.62	3.29	1	4	7	9	12
launched_at_yr	12953	2014.78	1.01	2012	2014	2015	2016	2017
create_to_launch_days	12953	36.69	59.11	0	4	13	41	373
launched_to_deadline_days	12953	34.10	11.49	1	30	30	37	64
usd_pledged	12953	12783.24	37854.76	0	26	744	5863.61	396299

Feature Selection

For the *regression task*, feature selection was used to find the best subset of predictors that minimize mean squared error (mse). Random Forest was applied for feature selection since it provided an initial low mse. The number of predictors is 139. LASSO in particular was not performed due to uncertainty around the correct alpha value. Linear Regression, KNN and Random Forest were applied to look for look for mse for the given set of predictors. The mse for different models given the list of predictors from Random Forest is provided in table below.

FEATURE SELECTION FOR REGRESSION TASK			
	LINEAR REGRESSION	**KNN**	**RANDOM FOREST**
Random Forest	1147498899.43	1143772030.62	1112291467.54

For the *classification task*, to derive the best subset of predictors that yield the highest accuracy, RFE, LASSO and Random Forest were applied for feature selection. Logistic regression, random forest and SVC methods were applied to look for accuracy for the given set of predictors. The following table describes the number of predictors for each method and the accuracy score. Note that for RFE feature selection, the SVC and KNN methods did not run. For Lasso, the threshold was kept at 0.01 due to excessive computation time to test different possible alpha values. The threshold for random forest was kept at 0.05 for Gini-coefficients.

FEATURE SELECTION FOR CLASSIFICATION TASK						
	RFE		**LASSO**		**RANDOM FOREST**	
	Predictors	Accuracy	Predictors	Accuracy	Predictors	Accuracy
Logistic Regression	111	0.691	26	0.722	79	0.687
Random Forest	141	0.734	26	0.703	129	0.745

SVC (sigmoid, rbf, linear)	-	-	(26, 26, -)	(0.646, 0.723, -)	(30,60,60)	(0.676, 0.735, 0.735)

Model Building

For both **regression and classification**, the **optimal model** based on mse and accuracy was **Random Forest**. The optimal hyperparameter values were determined by looping through all possible hyperparameter values separately. Below is a table for the optimal hyperparameters for each model (while considering for computation time).

OPTIMAL HYPERPARAMETER VALUES FOR RANDOM FOREST		
Hyperparameter	**Regression**	**Classification**
max_features	23	123
max_depth	8	118
min_samples_split	10	32
min_samples_leaf	10	26
bootstrap	1	1

While some models could have proved to have better handled the dataset, models were tested based time efficiency. Model such as SVM did not perform optimally for the regression task and training a model with ANN proved to be computational heavy. For the *classification model*, KNN did not run and, once again, ANN proved to be to computational heavy. As per the table below, the accuracy for the classification model is roughly 75% and the MSE of the regression model is roughly 1 billion.

Model Interpretation and Business Context

MODEL PERFORMANCE FOR REGRESSION	
MSE	1051796093.548961
Error	32431.40597551948564
R^2	0.17723819650512462

For the *regression task*, the error of the model (square root of MSE) is roughly 32431.4. This says that, on average, the Random Forest Regression model can predict the correct amount of money pledged (USD) within ± 32431.40$. This range may be interpreted as very broad given that the mean for usd_pledged is

12783.24$. R^2 tells us that roughly 17.7% of the variation regarding the amount pledged (USD) can be explained using the model.

For the *classification task*, the odds that the model correctly predicts the state of the project is 75% - **Accuracy**. The model correctly predicted samples of the training data (sampled with replacement) 74.5% of the time - **OOB Score**. Kickstarter should minimize false positives to avoid cases where Kickstarter incur large promotion costs for projects that never reach their goal. In fact, the probability that the model incorrectly predicts the state of a project given the model identified a successful project is 13% - **False Positive (FP)**. Moreover, by minimizing false positives, Kickstarter should look to maximize precision for better return on investment on a project. The probability that the model is correct given that the model identifies a successful project is 65% - **Precision**. Furthermore, the probability that the model identifies a failed project among successful projects is 46% - **False Negative (FN)**. In fact, the probability that the model identifies successful projects among actual successful projects is 54% - **Recall**. Kickstarter should, within reason, want more false negatives (compared to false positives) and, hence, a lower Recall value. FN's can be considered as underdog projects that are not advertised but are successful and, thus, are better than FP's in that they do not incur cost.

MODEL PERFORMANCE FOR CLASSIFICATION	
Accuracy	0.7507920049950154
OOB Score	0.7451196647182089
Precision	65%
Recall	54%
False Positive	13%
False Negative	46%

To summarize, maximizing precision while minimizing recall will prevent poor investments into projects that are likely to fail and ensure that projects that are likely to be successful are well supported.

CONFUSION MATRIX FOR CLASSIFICATION		
	Predicted Failed	Predicted Successful
Actual Failed	2324	353
Actual Successful	557	652