Final Project

Ryan Fischbach

5/13/2021

Contents

Abstract

The ability to predict how much funding a Kickstarter campaign will receive can help identify key characteristics that will allow for guidance to be given to entrepreneurs. A model to help them to predict the amount pledged to a campaign will allow them to set up their crowdfunding campaign to have the best chance to succeed. In this report, we discuss the process of estimating the amount of funding received by a Kickstarter campaign using data collected on Kaggle from the Kickstarter platform. The steps in the process and comparison of results from different models are discussed.

Section 1: Data and Motivation

The "Kickstarter Projects" dataset on Kaggle houses data on 378,661 Kickstarter Projects collected from the Kickstarter Platform. Kickstarter Projects are crowdfunding campaigns, giving a platform for entrepreneurs to put their idea on the platform and sell it to consumers. The consumers can then fund the project through to completion. Projects were collected until 01/2018, then the dataset was created. The raw data contains 15 features including:

- ID: Internal Kickstarter Unique Identifier
- name: Name of the Project
- category: Specific Category of the Project
- main_category: Broad Category of the Campaign
- currency: Currency Used to Support the Campaign
- deadline: Date Deadline for the Crowdfunding
- goal: The Amount of Money Needed to Successfully Fund the Project
- launched: Date the Project was Launched
- pledged: The Amount Pledged in Project's Currency
- state: The State of the Project ('Failed', 'Successful', 'Canceled', etc)
- backers: The Number of Backers for the Project
- country: The Country of Origin of the Project
- usd_pledged: Conversion to US dollars of pledged performed by Kickstarter
- usd_pledged_real: Conversion to US dollars of pledged column performed by API
- usd_goal_real: Conversion to US dollars of goal column performed by API

More information on the features and data can be found **here**.

The goal of this analysis is to predict the amount of funding a project would receive given the other attributes of the crowdfunding campaign. From this, the goal is to provide a recommendation for how best to create a campaign that will gather funding.

Section 2: Data Cleaning

To transform this initial dataset into data that can be used to generate a model, the following steps were taken to clean the data.

The uncleaned dataset was loaded, containing 15 variables and 378,661 observations.

1. First, columns that didn't encode any information to help in modeling were removed. The columns removed were the project's unique identifier, the name of the project, the project deadline, and the date launched.

2. Second, columns that encoded redundant information were removed to ensure information was only considered once later in the modeling phase. This removed category, pledged, state, goal, and USD pledged. Category was a more in-depth version of "main_category" so it was removed and pledged/USD pledged

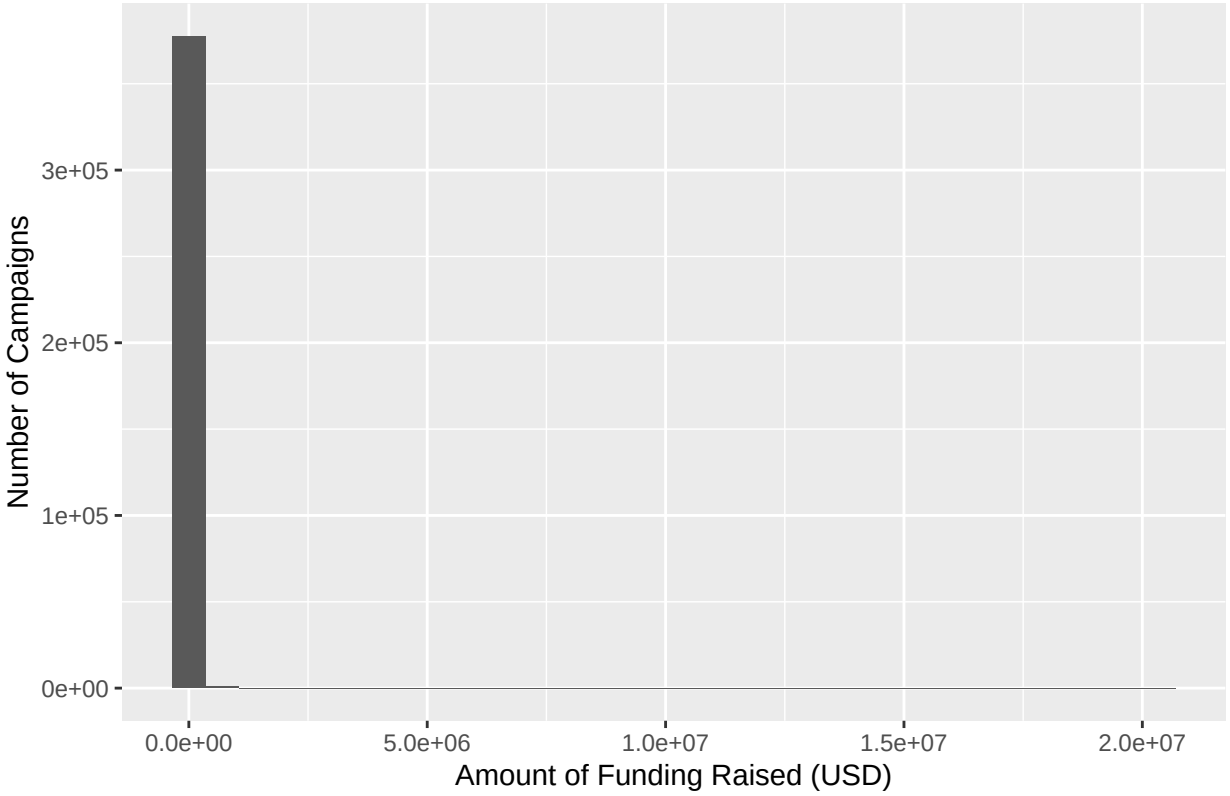
were redundant to usd_pledged_real. Additionally, columns like goal and pledged were in their native currency. The columns usd_pledged_real and usd_goal_real took these columns and converted them to USD, allowing for the metric to be standardized. Lastly, the state feature was removed because it would indicate whether a project met its goal and was successful.

3. Lastly, categorical variables were one-hot encoded and stored in a new dataset for the shrinkage regresion techniques.

After these efforts, the dataset without encoding has 6 features and 378,661 rows. The dataset using encoding had 47 columns and 378,661 rows.

Additionally, the dataset was scanned for missing values to uncover if any additional work needed to be done to deal with them. After the previous data cleaning steps, there are no missing values, requiring no action on our part.

Figure 2.1: Distribution of Pledged Dollars



Lastly, a distribution of the target variable was considered. Looking at the distribution of the response variable above (see Figure 2.1), it appears to be skewed-right, with its right tail being longer than its left tail. Additionally, it appears to be unimodal. The distribution has a median of 624.33 and of mean 9,058.92.

The distribution has a large spread with a standard deviation of 90,973.34. Additionally, there are multiple outliers that exist on the right tail with funding greater than 9,189 (1.5*IQR + Q3).

The largest thing that sticks out about this distribution is that the vast majority of projects get little to no funding at all. 50% of all projects receive less than 624 USD in pledged funding, but the average is significantly higher at 9,059 USD. The goal of this analysis is to find what makes a campaign successful. However, with so many campaigns not raising much money, the data is heavily skewed towards unsuccessful campaigns. This will then impact our model. Thus, we will perform undersampling of unsuccessful campaigns, with an arbitrary cutoff of 10,000 USD. Additionally, with these campaigns over 10,000 USD, we set an upper limit

of 100,000. Campaigns over 100,000 USD in pledged funding are possible "unicorns" and not reproducible or by a reputable company (among countless other reasons), so they do not add much in terms of insight for the average Kickstarter project.

There are 46,358 observations where funding is greater than 10,000 USD and less than 100,000 USD. Thus, we will create a new dataset with these 46,358 observations and also randomly sample 46,358 observations under or equal to 10,000 USD in funding. This will not modify the relationships between variables, but it will allow this analysis to get a better sense of how to predict successful campaigns.

After these cleaning steps, there are 6 features and 92,716 rows in the cleaned dataset. The one-hot encoded dataset has 55 features and 92,716 rows.

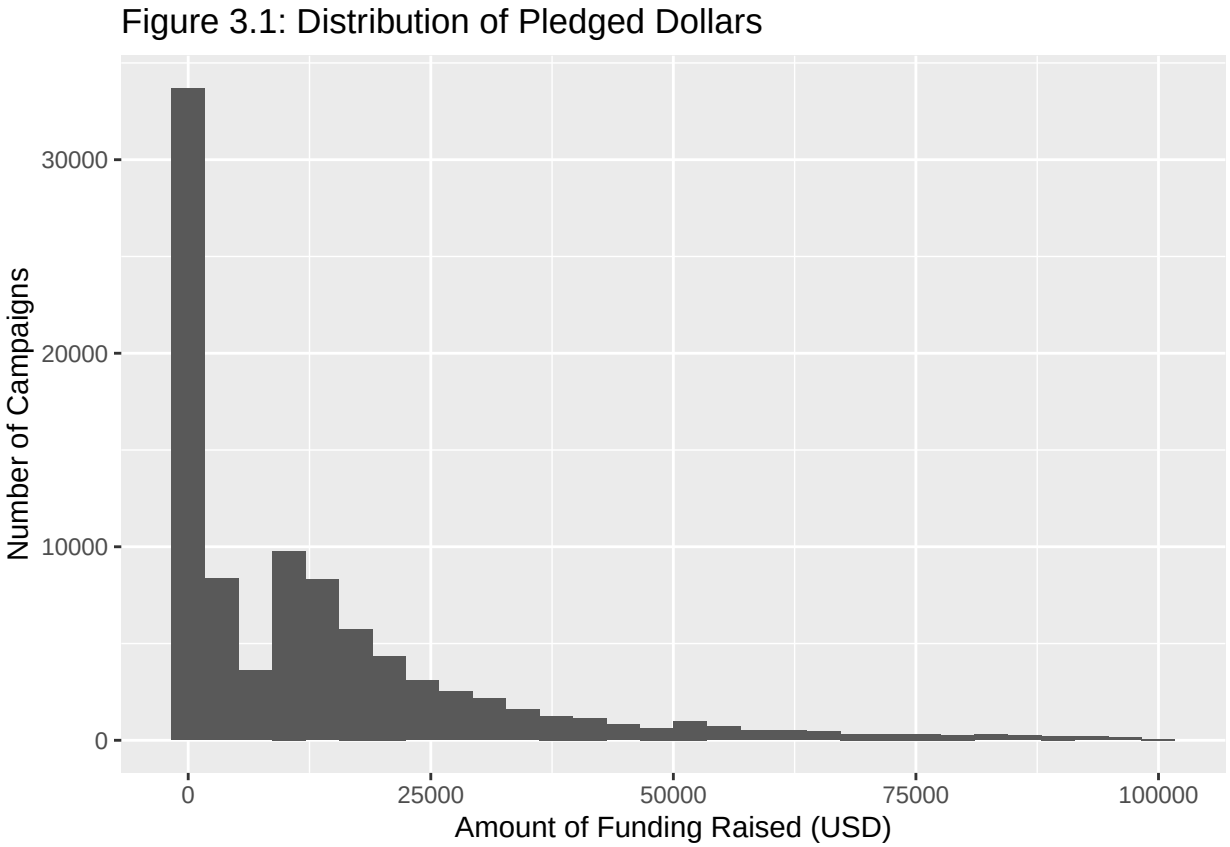
Section 3: Regression Tree

Section 3.1: Introduction

The first method chosen for this analysis was a regression tree. A regression tree is a non-parametric model that allows for numerical prediction to be given based on the features of the data. With the combination of categorical and numeric data, trees allow these to be handled without having to use any sort of encoding or other practices. Additionally, a tree can perform feature selection, allowing for the most important factors to be used when attempting to model calories. This will allow for an interpretable model given the flowchart-like structure of trees, which can also be visualized.

Section 3.2: Data Visualization

Before modeling the amount pledged to a campaign, we will first explore the response variable to better understand its distribution. A histogram will be used for this task.



Looking at the distribution of the response variable above (see Figure 3.1), it appears to be skewed-right, with its right tail being longer than its left tail. Additionally, it appears to be unimodal. The distribution has a median of 10,000.01 and of mean 13,765.81 USD.

The distribution has a large spread with a standard deviation of 18,073.35. Additionally, there are multiple outliers that exist on the right tail.

While this distribution is still very skewed, the downsampling of funding less than 10,000 USD has helped us identify projects that are deemed "successful" that we can analyze.

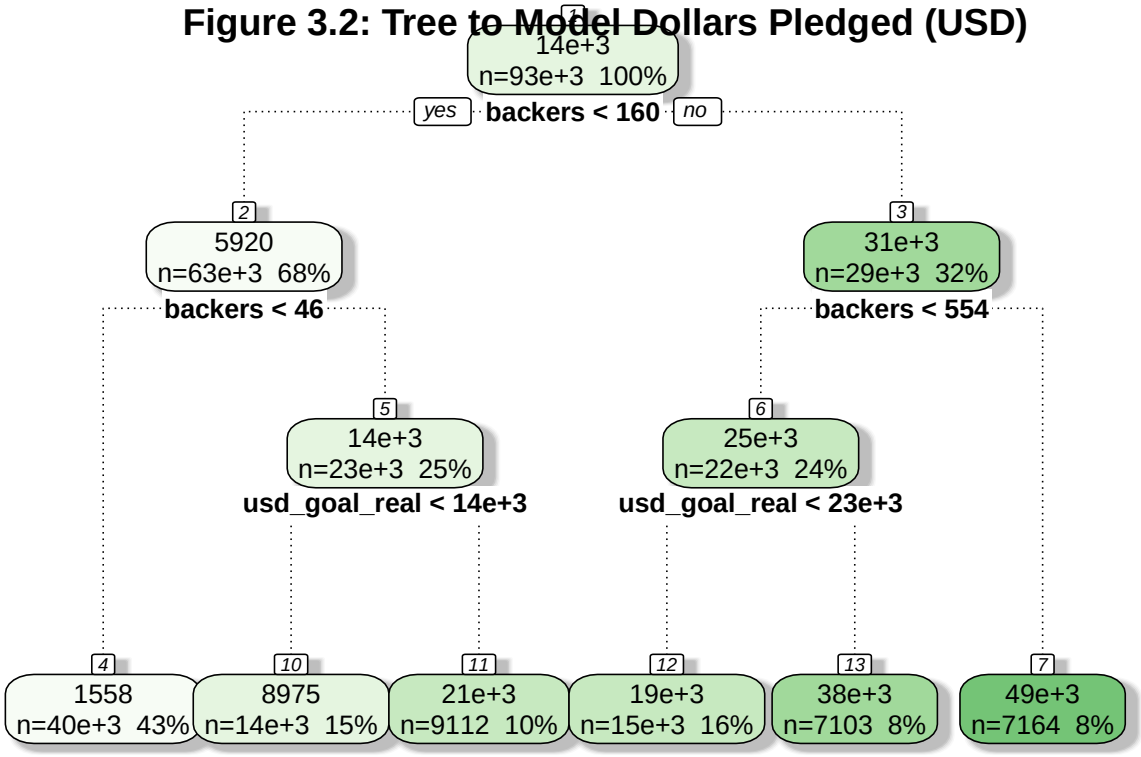
Section 3.3: Method

As mentioned before, Regression Trees are a flowchart-like model that take the characteristics of a new observation and guide it into a bucket, containing a prediction. To create a Regression Tree, you start by assigning all the data into one big bucket. The mean of the variable that is being predicted serves as the prediction for this bucket. This bucket is called the Root Node.

We then use the Residual Sum of Squares (the sum of squared errors on data aka RSS) as a metric to see how we can use our features to lower the RSS. For each feature, we find the split of the data (value in that feature) that minimizes the RSS. We then compare all features and see which one minimizes the error we get when creating our predictions. This process continues using a process called "Recursive Binary Splitting", where we can consider each new split on the last and continue to try to minimize the RSS. We eventually want this process to stop because new splits as we get farther will help less with predictive accuracy and just increase the complexity of the model we are creating. Thus, this is a balancing act between putting observations in the right "bucket" and keeping the tree simple and interpretable. Additionally, we consider a technique called pruning where we try to remove parts of the tree that rely too much on noise in the data to allow for this tree to perform better with new data and keep complexity low. We prune by reducing the

number of leaves that exist within the tree. All of this combined yields an interpretable flowchart that allows for prediction. We will now train a regression tree model, prune it, and visualize the result.

Figure 3.2: Tree to Model Dollars Pledged (USD)



After training this regression tree-based model and pruning it, the flowchart above was created. As you can see in Figure 3.2, the "splits" indicating a decision put the data into leaves at the bottom, corresponding with a prediction. These thresholds allow for classification of the amount pledged in USD along with keeping the tree understandable.

Section 3.4: Results

To validate this model, 10-fold cross-validation was chosen. To perform 10-fold cross-validation, we create 10 sets of the data, all roughly equal size. Each small dataset is a fold, derived from the original dataset. One fold is used as a test set to validate the model, with the rest being used to train the model. Each combination of training and test sets is used, yielding 10 combinations of cross-validation testing. This method was chosen because of its balance of computational complexity compared to a more rigorous method (like Leave One Out Cross-Validation) and the ability to eliminate randomness (unlike the Validation Approach). With so many observations, this method was appropriate for this problem context.

To pick the amount of pruning performed on the tree, a combination of different options for how strong the pruning is are tried. The option that minimizes both the average error on the test set and the complexity of the tree is chosen. In this case, 5 splits minimized the error on the test set and the complexity of the tree.

After pruning was performed, the average error of the pruned tree on the test set was 10,988.67 USD and the average error on the training dataset was 10,938.53 USD. Both were calculated by using the percentage reduction in the root node error multiplied by the root node error (see codebook for calculations).

From this tree, we can see that using the number of backers and the goal of the campaign in USD were the two important features that the tree picked. From this, we can learn that having a high number of backers

helps with the success of a campaign and a campaign with a high goal can indicate success. Additionally, when looking at variable importance metrics, backers and usd_goal_real were the only two features with any importance according to the tree, reaffirming this conclusion.

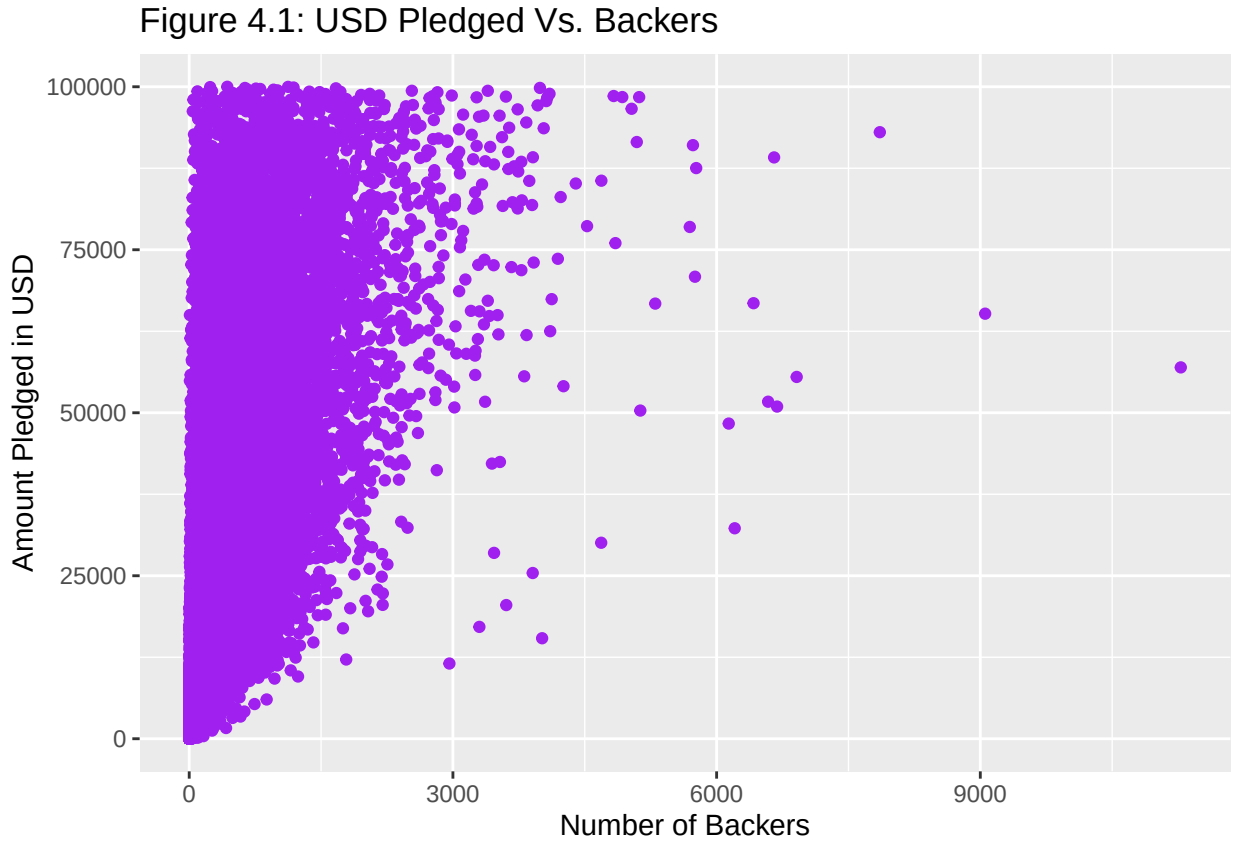
Section 4: Lasso Regression

Section 4.1: Introduction

The second method chosen for this analysis is called Lasso Regression. Lasso is a regression technique that allows for variable selection and shrinkage (lowering the weights of features). Lasso regression will allow for the important features to be selected for the model. The downside of this model is that the categorical variables will have to be encoded and that lasso tends to prefer fewer predictors. Thus, Lasso can create situations where it was better to just reduce the coefficient of a feature but instead it was removed.

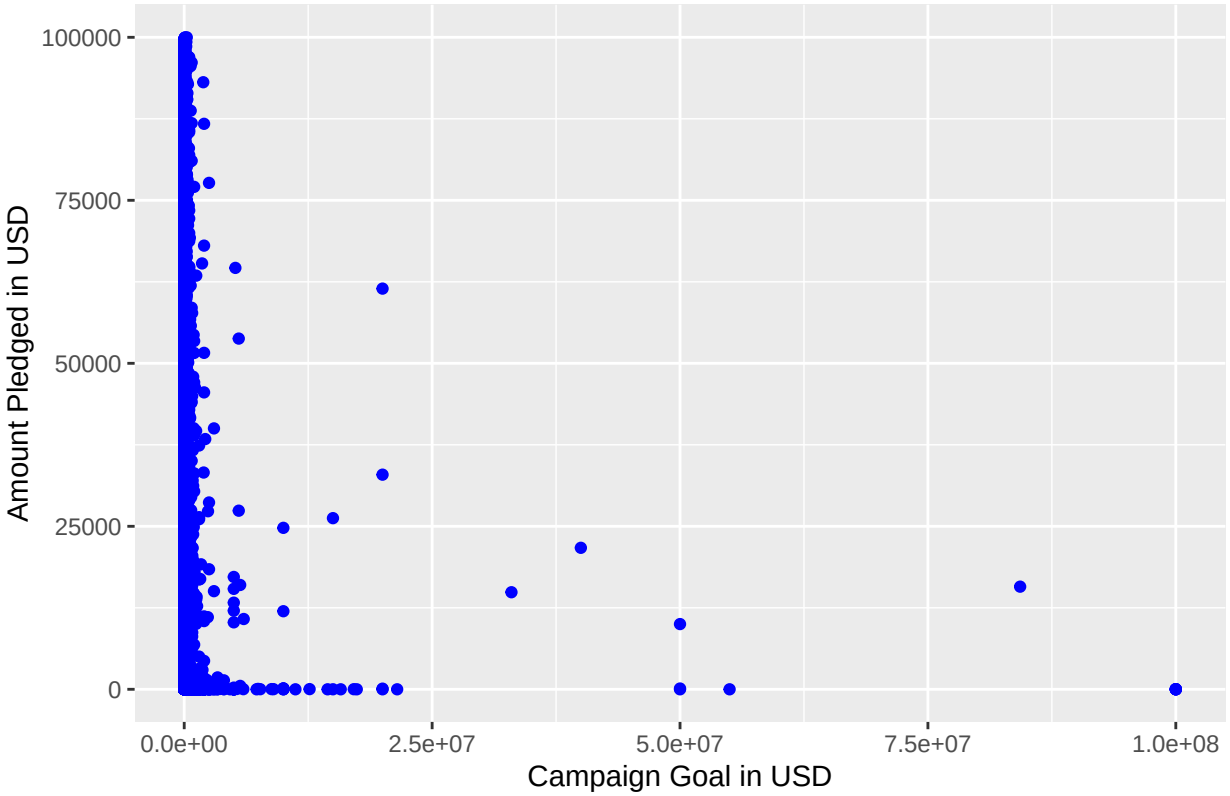
Section 4.2: Data Visualization

To perform Lasso regression, we need to ensure that our numeric predictors have a linear relationship with our target variable. After previously analyzing our target's distribution, this is the next step to ensure that a linear model should be used.



The Number of Backers seems to have no discernable relationship with the amount pledged (Figure 4.1). While there could exist a linear relationship, the number of observations and their concentration on the left side of the graph makes it hard to tell. Despite the fact that this condition might not be satisfied, this variable will be arbitrarily kept in the model because of its general importance and logical relationship to the amount of funding.

Figure 4.2: USD Pledged Vs. USD Goal



Looking at the relationship between the campaign goal in USD and the amount pledged in USD (Figure 4.2), it appears that the campaign goal has no relationship with the amount pledged. Once again, a large number of observations are concentrated on the left side of the graph, with low campaign goals. There does not appear to be a possibility of a relationship, so the campaign goal will be dropped as a predictor because the assumption of a linear relationship is not satisfied.

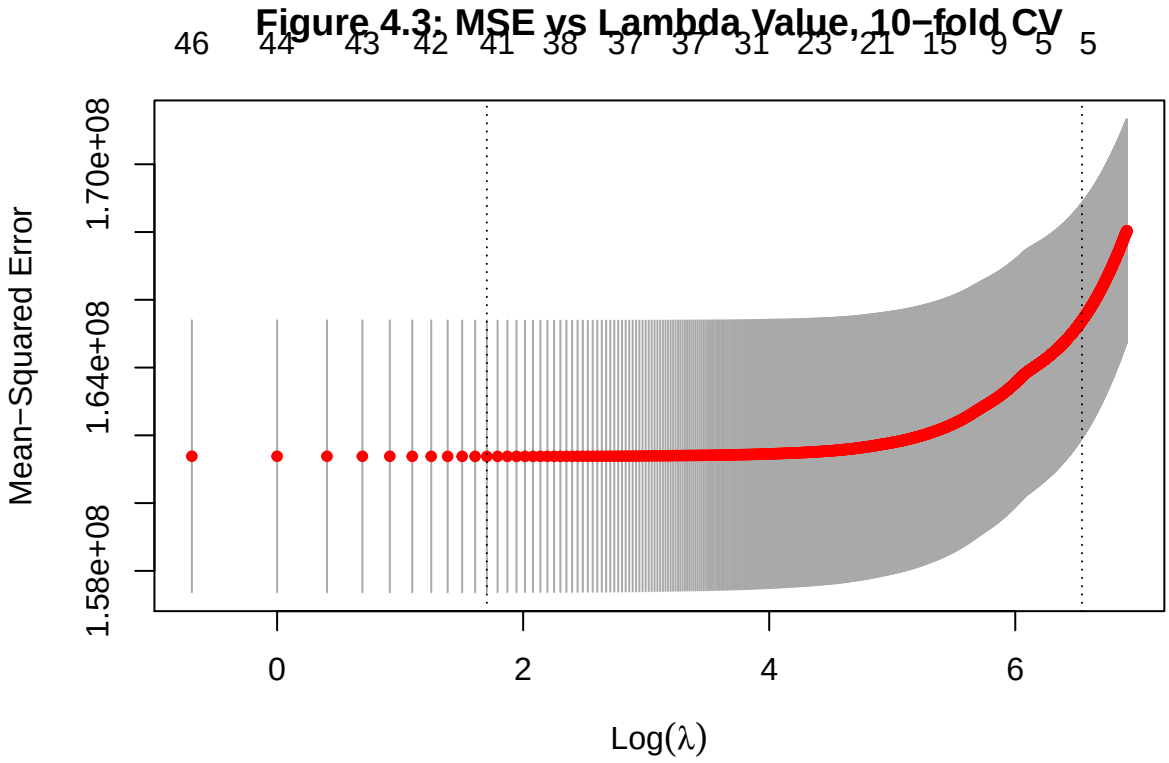
Section 4.3: Method

Lasso regression is nearly identical to linear regression but with a key difference. For Lasso regression, we look for coefficients of our variables that minimize the Sum of Squared Errors plus a penalty for our predictions. Lasso differs from other regression techniques because this penalty term is a scalar multiplied by the sum of the absolute value of the coefficients. In other words, we are trying to balance predictive accuracy with ensuring we don't rely too much on certain predictors.

In technical terms, we choose the estimates of $\hat{\beta}$ that minimize the term:

$$RSS + \lambda_{lasso} \left| |\hat{\beta}_j \right|_1 = (Y - X_D\hat{\beta})^T(Y - X_D\hat{\beta}) + \lambda_{lasso} \sum_{j=1}^{k} \left| \hat{\beta}_j \right|$$

where $\lambda_{lasso} \geq 0$ is a scalar.



To determine the optimal value of the tuning parameter λ for our Lasso model, we fit a large number of models using different *lambda* values and pick the model that minimizes a test metric (see Figure 4.3). We will use 10 fold cross-validation again for the balance of accuracy and speed, in addition to keeping the technique used for model validation the same across all models. After performing 10 fold cross-validation using the tuning parameter λ in the range 0 to 1,000 by increments of 0.5, we find that λ equal to 5.5 minimizes the Root Mean-Squared Error (as shown in Figure 4.3).

Our final lasso model obtained from 10-fold cross-validation takes the form:

$$\widehat{usdpledgedreal} = 4,478.14 - 1.255.44 maincategoryComics - 2,099 maincategoryCrafts + 116 maincategoryDance + 6,402.03 maincategoryDesign + 3,161.43 maincategoryFashion + 4,378.27 maincategoryFilmVideo + 3,047.38 maincategoryFood - 83.39 maincategoryGames - 336.15 maincategoryJournalism + 507.45 maincategoryMusic + 867.80 maincategoryPhotography - 766.22 maincategoryPublishing + 7,194.94 maincategoryTechnology + 1,578.49 maincategoryTheatre - 1,377.16 currencyCAD + 5,003.32 + currencyCHF + 1,068.99 currencyDKK + 501.77 currencyEUR - 88.49 currencyGBP + 2560.72 currencyHKD - 4,953.90 currencyJPY - 5,240.51 currencyMXN - 246.50 currencyNOK + 261.04 currencyNZD + 895.69 currencySEK + 387.28 currencySGD + 83.74 currencyUSD + 38.54 Backers - 1.492.96 countryAU + 198.32 countryBE - 154.22 countryCA + 1,433.65 countryCH - 739.09 countryDE + 457.39 countryDK - 1,342.92 countryES + 936.92 countryFR - 62.93 countryGB + 438.61 countryHK + 738.29 countryIE - 915.71 countryIT - 321.82 countryJP + 3,856,77 countryLU - 136.05 countryMX - 3,386.20 countryN, 0 + 239.96 countryNL - 304.04 countryNO + 250.37 countrySE$$

From this model, we can see that many of the explanatory variables remained, likely due to the low regularization performed with $\lambda = 5.5$.

Table 1: Table 4.1: Lasso Model Coefficients

	Lasso
(Intercept)	4478.14246
main_categoryComics	-1255.44733
main_categoryCrafts	-2099.02816
main_categoryDance	116.87193
main_categoryDesign	6402.02979
main_categoryFashion	3161.42861
main_categoryFilm & Video	4378.27520
main_categoryFood	3047.38230
main_categoryGames	-83.38755
main_categoryJournalism	-336.15752
main_categoryMusic	507.44962
main_categoryPhotography	867.79850
main_categoryPublishing	-766.22380
main_categoryTechnology	7194.94493
main_categoryTheater	1578.49993
currencyCAD	-1377.16530
currencyCHF	5003.31619
currencyDKK	1068.98551
currencyEUR	501.77360
currencyGBP	-88.49243
currencyHKD	2560.72085
currencyJPY	-4953.90043
currencyMXN	-5240.50883
currencyNOK	-246.49749
currencyNZD	261.04453
currencySEK	895.69389
currencySGD	387.27968
currencyUSD	83.73560
backers	38.53686
countryAU	-1492.95803
countryBE	198.32461
countryCA	-154.21515
countryCH	1433.65036
countryDE	-739.09491
countryDK	457.38967
countryES	-1342.91595
countryFR	936.91508
countryGB	-62.93336
countryHK	438.60640
countryIE	738.29050
countryIT	-915.71541
countryJP	-321.82420
countryLU	3856.77572
countryMX	-136.05002
countryN,0"	-3386.20205
countryNL	239.95892
countryNO	-304.03995
countryNZ	0.00000
countrySE	250.36867
countrySG	0.00000

	Lasso
countryUS	0.00000

Taking a look at the coefficients of the final Lasso model, the variable selection is apparent. A portion of the features' coefficients were shrunk to 0, allowing us to interpret that these are not important for the prediction task when it comes to Lasso.

Section 4.4: Results

Using the tuning parameter that minimized the test metric in the 10-fold cross-validation, an average error of 12,703.52 USD is achieved.

This model indicates that certain categories, countries, currencies, and a higher number of backers receive more money pledged towards their campaigns. The country and currency are not really allowed to be changed by entrepreneurs as most are locked in the country and currency where they own their business. Looking at backers, for 1 additional backer, the estimated additional pledged money is 38.54 USD. This would suggest that investing in marketing or other methods to achieve more backers is a strong indicator of a campaign's pledged amount. In terms of Kickstarter, this makes sense because a backer pays a set price and agrees to support the campaign, As the number goes up, more pledged funding is raised. Looking at the categories, it appears that technology, food, film and video, and design campaigns have the largest estimated coefficients while attempting to estimate the pledged funding in USD.

Conclusion

For this analysis, two models were considered to predict the amount pledged in USD to a Kickstarter campaign: a regression tree and Lasso regression. Looking at their results, the single regression tree achieved an average error of 10,988.67 USD with 10-fold cross-validation while the lasso regression model achieved an average error of 12,703.52 USD with 10-fold cross-validation. From these two metrics, it is clear that the single regression tree is the better performing model out of the two attempted during this analysis. Additionally, the ability to visualize the regression tree will better help an entrepreneur understand how the model works.

However, this metric put into perspective can indicate that this performance not ideal. The mean of our response variable is 13,765.81 and the median is 10,000.01. An average error of 10,988.67 USD for the tree might have beaten out the Lasso model, but that is still roughly equal to the mean and median value. In other words, our predictions are off on average by the mean of the variable we are trying to predict. This indicates that this "best" model has little predictive accuracy. This could potentially be due to the limited amount of features in the data and thus a limited ability of the data to create an informed prediction. Maybe what gives a campaign its edge is just not simply captured in the data.

I would recommend not using this regression tree model for a prediction task, but rather looking at what is identified as the key beneficial characteristics and weighing those while thinking about launching a Kickstarter campaign. In this case, having a large campaign goal and trying to increase the number of backers achieved. This could be done through marketing, or focusing on creating a project that solves a user need.

Works Cited Page

Kickstarter Projects. Version 7. Retrieved February 21st, 2021 from https://www.kaggle.com/kemical/kickstarter-projects.