ISyE 6416 – Computational Statistics – Spring 2020
Final Report

Team: Paula Gluss, Nathan Rich, Jared Babcock
Title: Predicting the Value of Startups and Related Driving Factors

Problem Statement:
For an investor, deciding which startups to invest in and how much to invest are difficult questions to answer. Additional insight into how much money a company could potentially raise would help inform investment decisions. Here, we will look at a dataset containing financial information about startups across multiple industries and their final amount of funding raised. Our goal is to predict the final amount raised based on early financial landmarks and identify which landmarks are the most meaningful.

This problem is important because an investor needs to optimize how they allocate their funds to different startups. Having an insight into how much a startup could be worth based on current available information would help the investors better allocate their available resources.

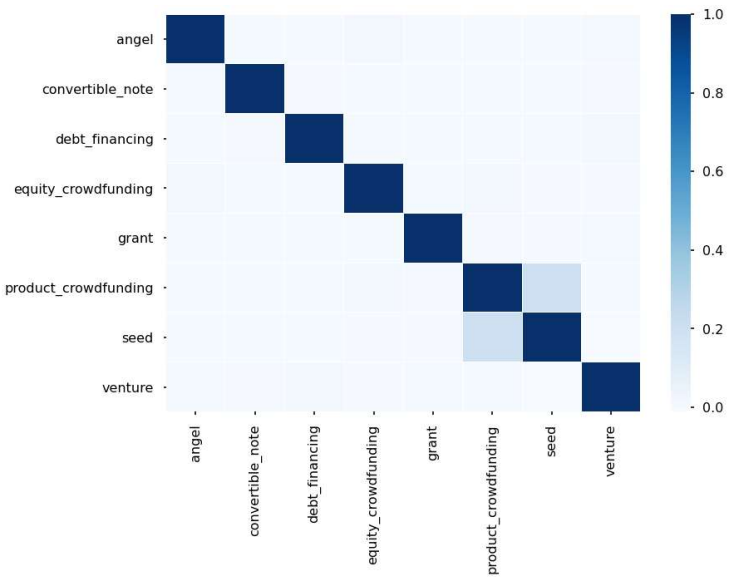
In summary, we have two main tasks that we want to accomplish:
1. Can we predict the amount of money a startup will raise within an acceptable level of accuracy?
2. What are the features that drive the amount of money a startup will raise? In other words, which features might make a startup more valuable to an investor?

Data Source:
We used the "Investments – Venture Capitalist" dataset from Kaggle. This dataset has approximately 49,500 rows and 40 columns. The data types are mixed and include strings, dates, and floats. Each row represents an individual startup company. The dependent variable is the total amount of funding a company has raised after all fundraising rounds are complete.

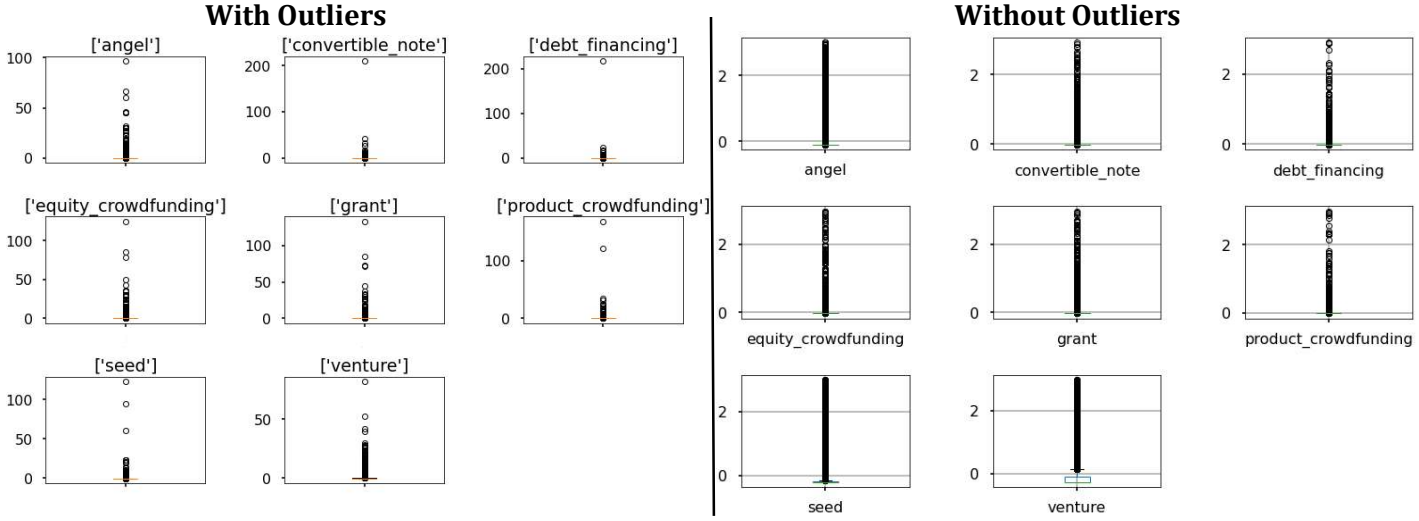
Methodology:
Data Wrangling
The initial dataset had multiple columns that were not pertinent to building a predictive model and had columns with mismatched data types. Our first step was to remove columns that contained company specific details such as the permalink, name, and website URL. We then checked the data types of the resulting columns and adjusted the strings and floats. One error we found included floats in the format " 17,50,000 ", which includes extra commas and whitespace.

After correcting datatype errors, we then looked at correlation between our numeric columns. Having correlated columns is inefficient during modeling, as it results in a larger feature space, higher training and prediction times, and can reduce the interpretability of the features. Given that positively or negatively correlated variables give the same information to our model, we want to remove correlation.

We created a correlation matrix and found that the seed funding has some positive correlation to the product crowd funding. The correlation coefficient is approximately 0.3 and as a rule of thumb, a coefficient above 0.5 means that one of the features can be removed. We decided to keep all the columns and see if, during variable selection, one of the columns we found here would be removed.



After checking for correlation between variables, we then did outlier analysis. Outliers cause the best fit line in linear regression models to shift, resulting in incorrect results. Thus, it is important to identify and remove them. We looked at boxplots of each of the numeric columns to get a feel for the distribution of the data. We then used a z-score test to identify any point outside of three standard deviations from the mean and removed them.

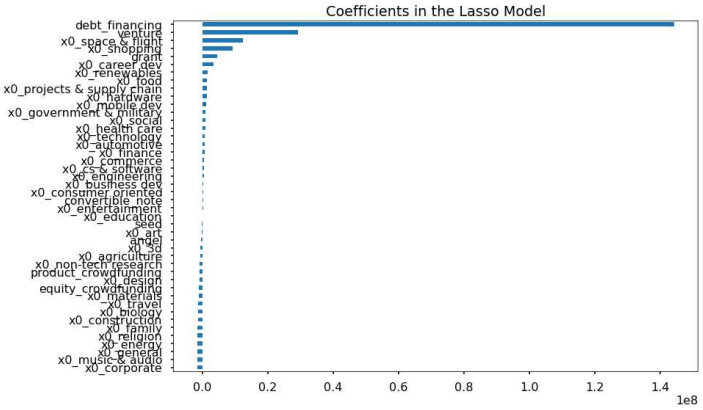


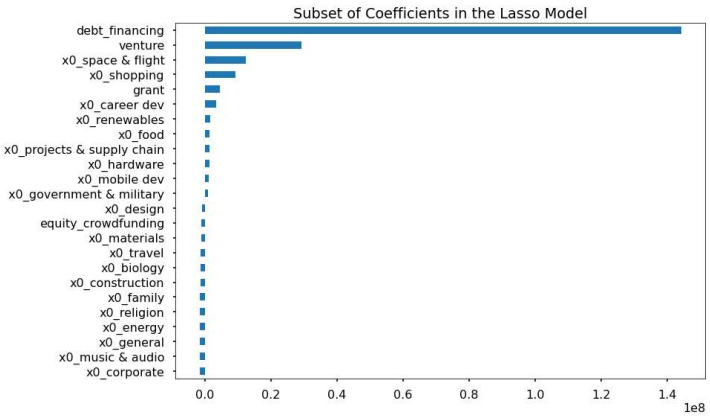
Above, on the left are the original boxplots of the numerical columns. On the right are the boxplots after outlier removal using the z-score test. Removing outlying values brought our row count from 49,438 to 47,860.

Our last step in data wrangling was to convert the categorical column to a one hot encoding, where each unique category becomes a column containing ones and zeros. This resulted in a feature data frame containing 47,860 rows and 44 columns.

Variable Selection

Since we observed some correlation between our numeric columns and we want to reduce dimensionality, we decided to use lasso regression as our variable selection method. We did cross validation to find the optimal alpha value and tested alphas of 1, 0.1, 0.001, and 0.0005. Cross validation found the optimal alpha to be 1 and the algorithm converged in less than 170 iterations.





Above on the left is a graph of the coefficients from all the features in our dataset. On the right, we see the top and bottom 12 coefficients by magnitude. We decided to keep 24 of the 44 available features.

Linear Regression & Random Forest Modeling

We decided to build two models and compare their performance on the final feature set. We chose linear regression for its easy interpretability and random forests for its robustness. We also wanted to compare the performance on an ensemble model versus a non-ensemble model.

To train our models, we split the feature set into 80% train and 20% test. We used standard linear regression and tuned the random forest's depth and ending criteria. For each model, we calculated the mean squared error (MSE) of the output.

Evaluation:

Driving Features

We want to identify which features of a startup have a noticeable impact on the total funding the company will raise. Thus, to evaluate each feature, we will look at the magnitude of the coefficients from the lasso variable selection and from the linear regression model.

If a coefficient has a magnitude very close to 0, we know that variable doesn't carry a lot of weight. If the magnitude is very large, then the feature is important to determining the dependent variable.

Modeling

To evaluate our models, we considered multiple factors:

1. The resulting MSE on the test set. We want to minimize the MSE as much as possible.
2. Train time. Ideally, the selected model would be quick to train and test.
3. Interpretability. In some models, we can directly interpret the coefficients. Ideally, the selected model would be interpretable and easy to derive business insights from.

Results:

Driving Features

Based on the coefficients from our lasso and linear regression models, we found that the most impactful features with a high total funding amount is their debt financing, venture funding, and whether they are in the space/flight industry. Some of the least important features are whether the startup is in the education industry and how much they raised during the seed round of funding.

Modeling

From our models, we found that a linear regression model had a resulting MSE of 340 billion dollars and an R^2 score of 0.465. The random forest had a resulting MSE of 287 billion dollars and an R^2 score of 0.67. While linear

regression models have coefficients that we can directly interpret and, in this case, are faster to train and test, we opted to select the random forest for its increase in accuracy.

Conclusion:
In summary, we have identified features of startups that have a larger impact on their total amount of funding raised and have explored two models for predicting the final funding amount. We found that, while random forests achieved a lower MSE than linear regression, the MSE was still very high. In the process of building the models, we saw various aspects that indicate that the approach found here may not be ideal for identifying investment opportunities. For example, while we identified high debt as a feature of a future valuable startup, it may not make sense to invest in a company with too much debt at the get-go.

We can take additional steps to help improve the accuracy:

1. Try a different set of models, such as a Gaussian mixture model (GMM) or neural network (NN).
2. Verify the market/industry groupings. We're currently relying on data that has not been verified by us. Regrouping the startups may help us identify different important factors.
3. Gather more data. Predicting startup funding is a difficult task and the more data we have, the more representative it will be of the marketplace. This may help us find patterns we were previously missing.

Appendix:

Collaboration Plan

Name	Owner	Description
Proposal	Paula	Find dataset, prepare proposal with initial methodology and problem description
Data wrangling	Paula	Clean data, including data types, correcting string errors, grouping by market, and building one hot encodings
Outlier detection & handling	Paula	Fix any missing numerical values. Normalize data and check for outliers, removing them if needed.
Variable selection	Nathan & Jared	Implement variable selection method, do hyperparameter tuning, identify valuable features and explain results.
Linear regression modeling	Nathan & Jared	Implement linear regression model, calculate RMSE, report results. Identify meaningful features with justification.
Random Forest modeling	Nathan & Jared	Implement random forest, do parameter tuning with RMSE, report results. Identify meaningful features with justification.
Final report	Paula	Compile final methodology and results into final written report.

Technical Details

We used Python and Google collab, a cloud based Jupyter Notebook environment, to complete this project.