Predicting Start-Up Acquisitions Using Venture Capital Data

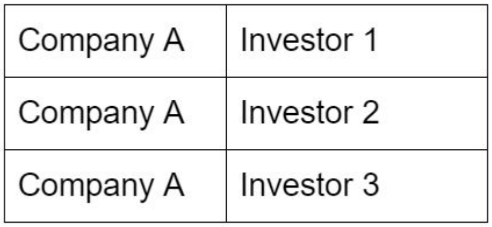
The Venture Capital (“VC”) industry continues to grow with a recording setting $132B invested in 2018, up from $48B just five years ago. There’s no denying that VC is its own asset class, situated as the wild west of investing in the latest and greatest innovations occurring on mother earth. Investors large and small are participating in the process of launching a company from angel investors providing seed capital to mature Series C+ rounds. Data about the industry continues to grow, but is still scant compared to public companies, even then, the companies themselves are rarely profitable and are mostly judged by growth metrics and the teams that a founder can assemble. Despite the opaque nature of these markets, much can be understood about human nature and the fear of missing out. As companies progress through rounds, more investors circle; signifying greater traction and larger investment rounds.

In this project, I focus on finding the quantitative determinants of what makes a company successful (i.e. acquired) versus companies that fail or don’t get acquired.

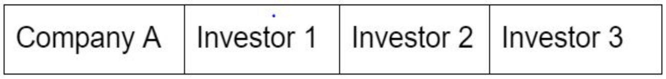
The Data

Crunchbase made data available as of year-end 2013, which contained four separate dataframes covering company level information, investors, investment rounds, and acquisitions. Wrangling the data required a number of steps, but the underlying goal was to extract and consolidate as much information as possible at the **company level.** This means moving the investment rounds and investor data, which contained multiple rows for each company, **from a long format to wide format at the company level.**

Long to wide example:



Company A	Investor 1
Company A	Investor 2
Company A	Investor 3



Company A	Investor 1	Investor 2	Investor 3

The idea behind this format is that we’d be able to consolidate all four dataframes and glean as much information as available about each company and prepare the data for a machine learning friendly format.

Company Dataframe

This dataframe served as the bedrock for the combined dataframe as it contained one unique record for over 17,000 companies. Primary variables included company location, company type, number of funding rounds, funding total, and founding date.

Acquisition Dataframe

This dataframe contained data on over 4,500 acquired companies, boosting our company dataframe to over 20,000 unique records. The only data we could use from this dataframe was the “acquired” status as showing the model other attributes would create false accuracy.

Investor Dataframe

The investor dataframe contained investor level data spanning over 51,000 records. This data ranged from investor location to when and how much a company received during a funding round. Variables used within this dataframe included number of investors per round, investor scores (see feature engineering), investor locations, funding dates, and funding amounts.

Rounds Dataframe

Funding round dates were provided on a company by company basis. Round amount, funding date, and day differences were all calculated and formatted for each round and transformed to wide format:

Company Name	*Founding date to round 1	*Days from Round 1 to Round 2	*Days from Round 2 to Round 3
Facebook	213 days ($0.5M raised)	242 days ($12.7M)	335 days ($27.5M)

*Two values in wide format: Days between rounds and round amount.

Notable issues

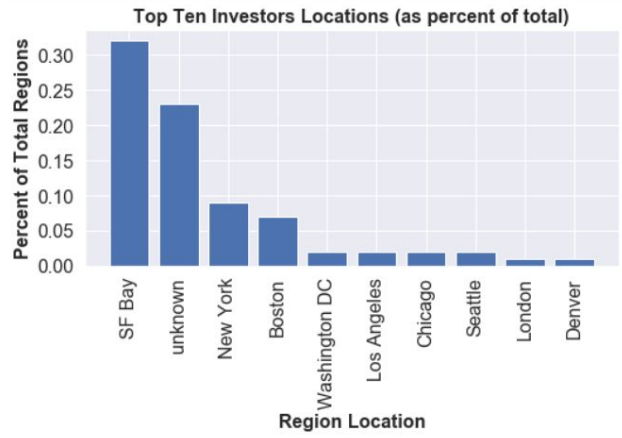
1) Individual dollar invested amounts weren’t given. For example, Facebook raised $27.5M in its 11th round from two investors. The data would show $27.5M for each investor, not each investors contribution.
2) Founding dates were inconsistent. Some founding dates were listed as being after the first funding round or a founding date would would be very early (1970’s). It wasn’t prevalent, but large enough to notice (~4% of data).
3) Some feature columns missing data.

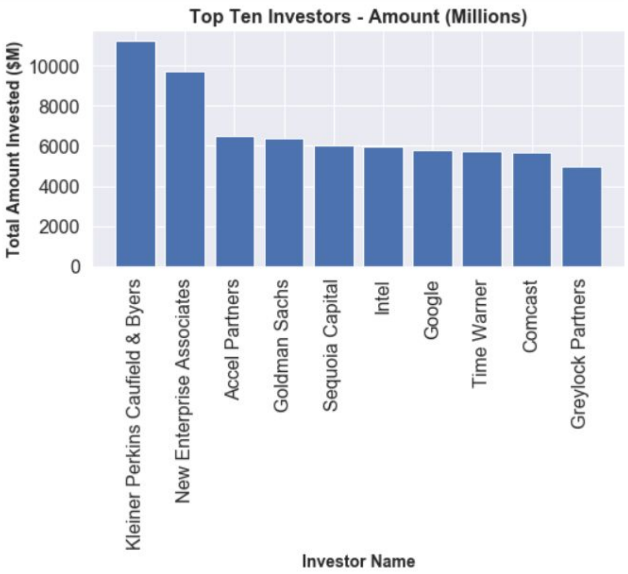
Columns	Pct. of total missing	Solution
Founded At and First Funding At (Used to calculate feature rddt_diff_1)	Founded = 6945 / 20621 = 34%	Replaced with median. If no funding, rounds = 0.
Investor Region	26% Unknown	Excluded from features
Funding Total	4399 / 20621 = 21%	Zeroed NaN’s. Most companies raised no

		money. Dropped rest (-7.5%).
Total Funding Rounds	2894 / 20621 = 14%	Zeroed NaN’s. Most companies raised no money.
Company Category	1063 / 20621 = 5%	Assigned own category code.

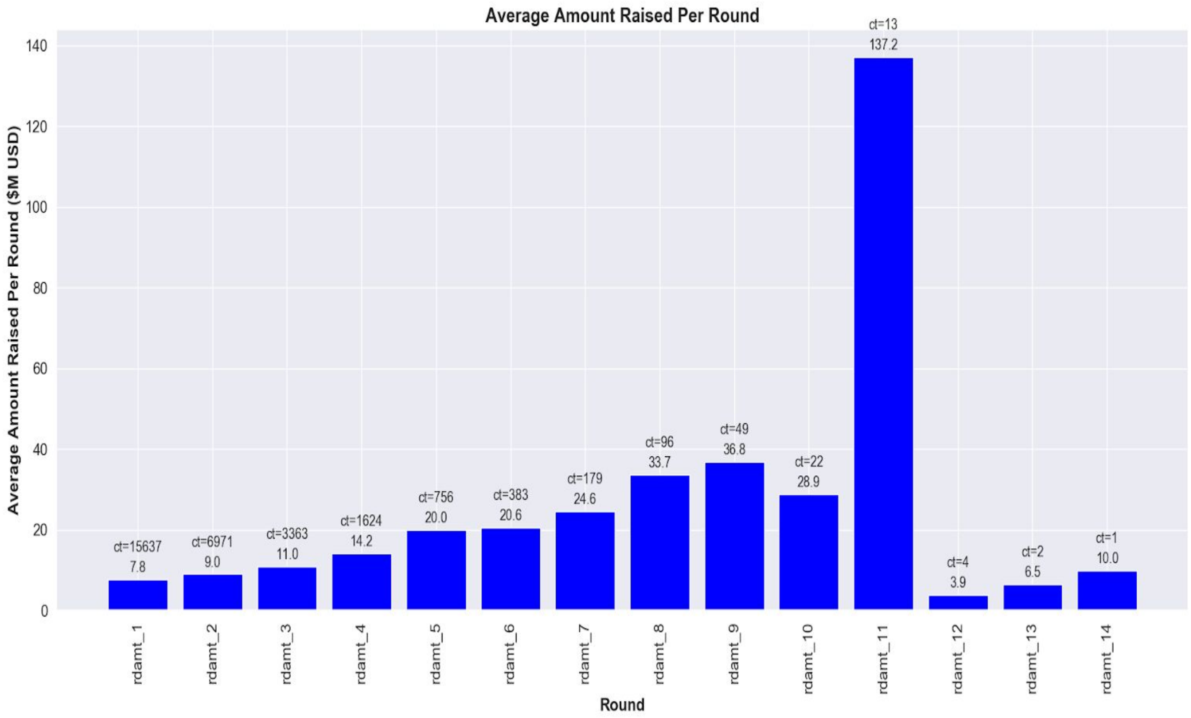
Exploring the data

My goal for EDA was to rapidly explore the different datasets available to me, determine completeness, and focus on the intuitive areas which are likely to be important data points. Funding rounds constituted the bulk of the dataset and I focused on finding clear trends, which could help me understand the quality of the data. The main area of focus within rounds is amount raised at each round, time between rounds, and investor score at each round.

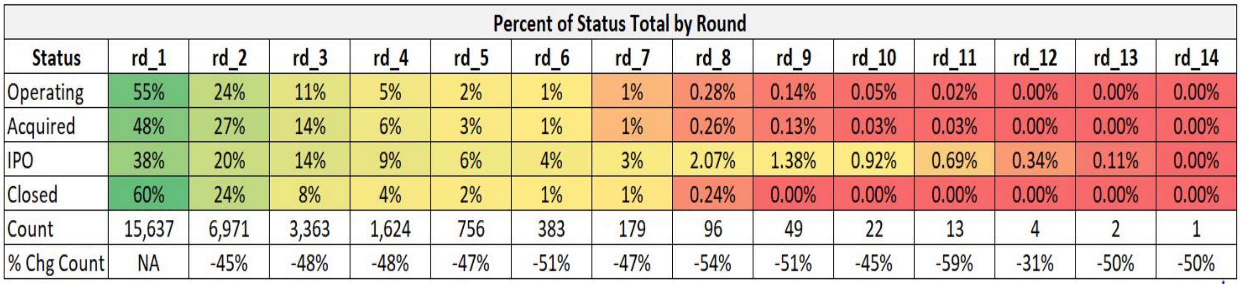




Not surprisingly, most investors were concentrated in the San Francisco Bay Area (~⅓). The unknown bucket is quite large and was reason enough for exclusion from the features list. I cross checked this data with company location and found a similar location trend as investors. There were 42 unique company categories and the top five categories were as follows: Software (17% of total), Biotech (11%), Web (10%), Enterprise (6%), and Mobile (6%).



I saw a fairly consistent trend of increasing average amounts raised as the rounds progressed to round 9. After round 9, we saw the number of instances begin to decrease, causing amounts to vary quite largely. Six companies went public during round 11 with the most notable being Facebook, Intrexon, and Tesla.

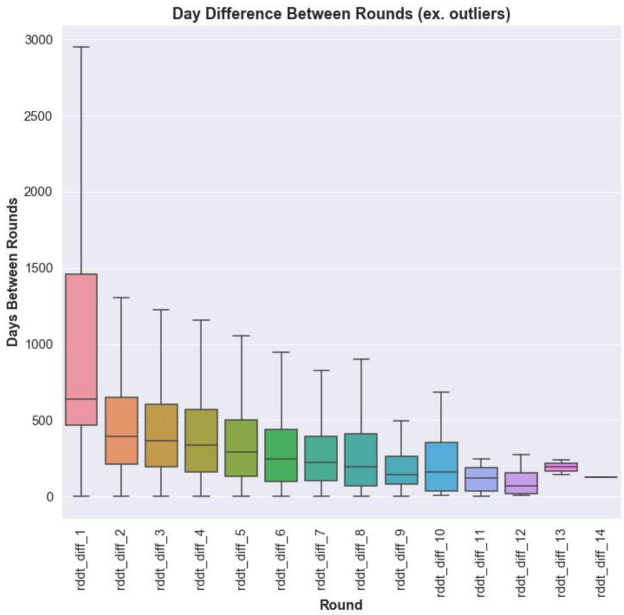


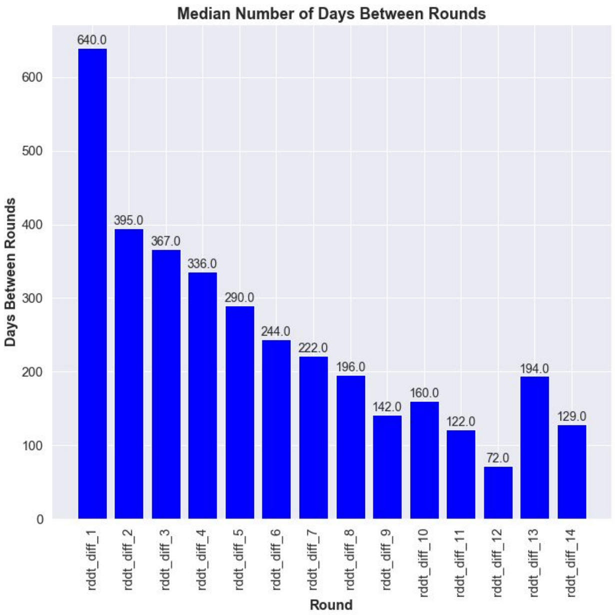
Percent of Status Total by Round

Status	rd_1	rd_2	rd_3	rd_4	rd_5	rd_6	rd_7	rd_8	rd_9	rd_10	rd_11	rd_12	rd_13	rd_14
Operating	55%	24%	11%	5%	2%	1%	1%	0.28%	0.14%	0.05%	0.02%	0.00%	0.00%	0.00%
Acquired	48%	27%	14%	6%	3%	1%	1%	0.26%	0.13%	0.03%	0.03%	0.00%	0.00%	0.00%
IPO	38%	20%	14%	9%	6%	4%	3%	2.07%	1.38%	0.92%	0.69%	0.34%	0.11%	0.00%
Closed	60%	24%	8%	4%	2%	1%	1%	0.24%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Count	15,637	6,971	3,363	1,624	756	383	179	96	49	22	13	4	2	1
% Chg Count	NA	-45%	-48%	-48%	-47%	-51%	-47%	-54%	-51%	-45%	-59%	-31%	-50%	-50%

We can intuitively verify where most of the action occurs...round 1, which makes sense as this is often the largest hurdle for a young company. It becomes increasingly rare to progress to the next round of funding as we see company counts halved fairly consistently in the dataset. As the Company progresses, so to do this round amount and subsequently we see increases in IPO's as a percent of total companies in each round.

One important element to this dataset is that we used the founding date provided to the first funding date to calculate the first round. The first round contained many outliers, which included age-old companies like Xerox, Raytheon, and Barnes and Noble that were started in the early 1900's. Excluding outliers, the median number of days from founding to first funding was approximately 640 days versus a full average of over 1300 days; triple that of round 2. To be clear, the outliers were valid and remained in the dataset, but the visuals were more helpful when excluding the outliers.

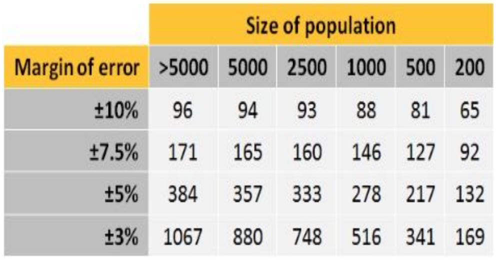




<u>**Modeling**</u>

After experimenting with several classification models, It appeared my best results were going to come from a **RandomForestClassifier**. Furthermore, this type of model seemed like an intuitive approach to this type of problem.

First, let’s look at the target labels:

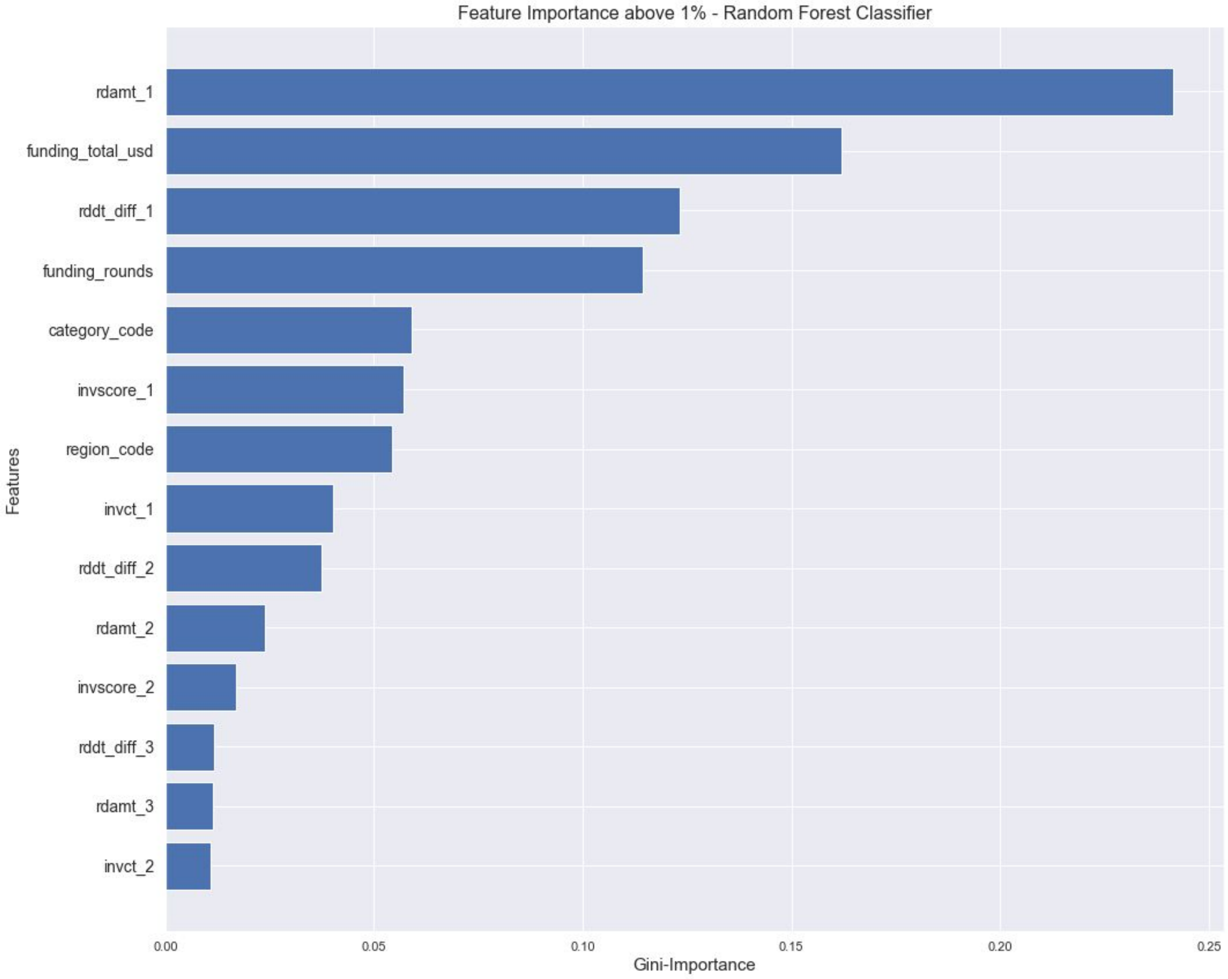
Target Label	One Hot Encoding	Count (% of Total)	 Source: tools4dev.org
Operating	3	13,806 (72%)	
Acquired	0	4,432 (23%)	
Closed	1	500 (3%)	
IPO	2	337 (2%)	

	Size of population					
Margin of error	>5000	5000	2500	1000	500	200
±10%	96	94	93	88	81	65
±7.5%	171	165	160	146	127	92
±5%	384	357	333	278	217	132
±3%	1067	880	748	516	341	169

While class imbalanced, each label does meet minimum statistical thresholds (see population chart). *I chose to maintain the multi-label structure, but an alternative route would be to remove closed and combine IPO into acquired.*

Feature Engineering

All in, we compared 19,075 companies across 58 features. Only eight of those features were unique in nature as many companies had up to 14 funding rounds where each round constituted a feature for four different fields (Round Amount, Days Between Rounds, Investor Score, and Investor Count). After instantiating and training the default RandomForestClassifier, I was able to narrow my feature count to four at a 10% variance threshold. I also tested a 5% variance and saw minimal difference in scoring. From the chart below you can tell that considering features beyond round 1 added little value to the model. It’s likely that funding rounds considered later round information versus needing to break out each round in granularity, however, breaking out each round proved worthwhile as four features that made the cut considered first round information. This also isn’t surprising given a large proportion of companies in the dataset weren’t funded at all (~14%).



Investor Score

Investor names where given for each round per company, but without further analysis I had no way of differentiating between the quality of each investor. As a proxy, I calculated investor scores which was made on the basis of how much that particular investor had invested to date. The idea is that investors who are raising and deploying more capital are likely more recognized, better resourced, and have access to more promising companies. The score itself is simply an investor percentile rank based on cumulative capital invested up until a given funding date. If multiple investors are in a round, then I take the average of all the scores. These scores were calculated for each round, which would glean additional information about traction (i.e. progression of investor quality by round). See example below:

Company Name	*Round 1 Investor Score	*Round 2	...*Round 11
Facebook	0.31 (2 investors)	0.45 (3 investors)	0.97 (2 investors)

*Values in wide format separately: investor score and investor count.

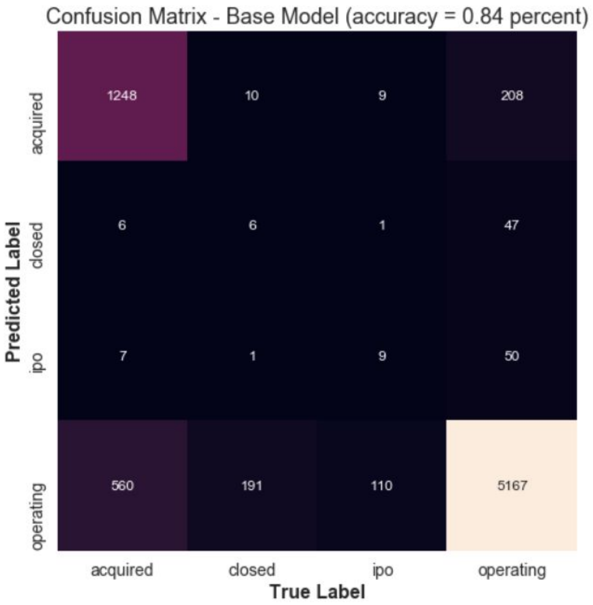
Parameter Tuning

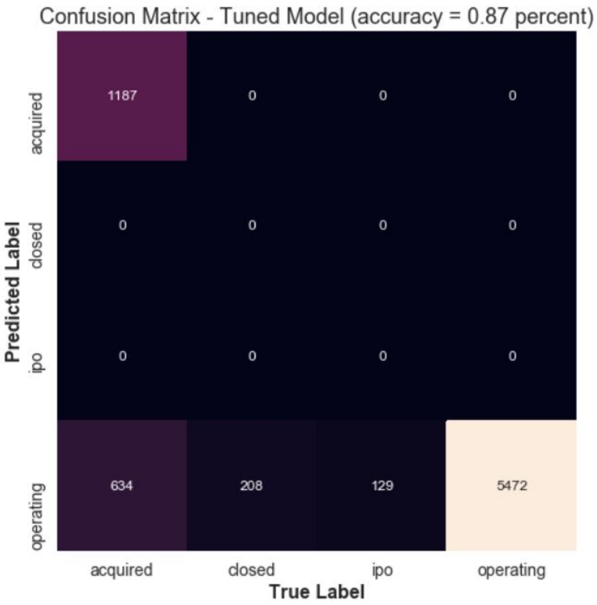
My chosen method of parameter tuning was RandomizedSearchCV to find some optimal inputs over a larger grid followed by a more exhaustive grid search once those outputs were identified. This approach is a tradeoff between compute time and thoroughness, but ultimately one that I felt presented a good balance between the two.

Model Score and Evaluation

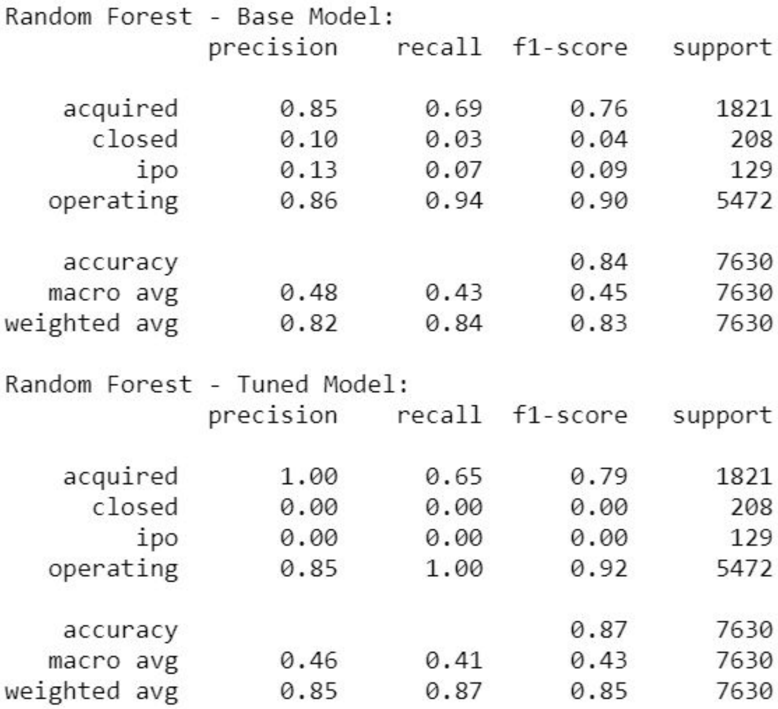
Model	**Accuracy Score**	**Parameters**
Base Model	0.84	Default
Tuned Model	0.87	n_estimators = 237 | min_samples_split = 120 | min_samples_leaf = 40 | max_features = ‘sqrt’ | max_depth = 74 | bootstrap = False
Best Grid	0.87	n_estimators = 200 | min_samples_split = 40| min_samples_leaf = 20 | max_features = ‘auto’| max_depth = 10 | bootstrap = True
Base Model - Full Dataset	0.86	Same as above
Tuned Model - Full Dataset	0.87	Same as above
Best Grid - Full Dataset	0.87	Same as above

We saw a modest performance uptick in the tuned versus base model (+3%) and negligible differences in performance versus the full dataset and only using the four selected features. The accuracy score is not the only metric which should be used and arguably not the best indicator using an imbalanced target class. The confusion matrix is one of my favorite views for seeing how accurate the model was, especially for a multi-class target.





Not surprisingly, the base model provided more diversity in predictions, but wasn't particularly accurate in those predictions (i.e. Closed and IPO). Given the lack of constraints, the base model mis-labeled over 350 operating companies as other labels. Conversely, the tuned model was binary in its predictions, which resulted in modestly higher accuracy and a cleaner confusion matrix. This outcome gives credence to the tuning process. Looking deeper into the scores, I examined the classification report.



```
Random Forest - Base Model:
              precision    recall  f1-score   support

    acquired       0.85      0.69      0.76      1821
      closed       0.10      0.03      0.04       208
         ipo       0.13      0.07      0.09       129
   operating       0.86      0.94      0.90      5472

    accuracy                           0.84      7630
   macro avg       0.48      0.43      0.45      7630
weighted avg       0.82      0.84      0.83      7630

Random Forest - Tuned Model:
              precision    recall  f1-score   support

    acquired       1.00      0.65      0.79      1821
      closed       0.00      0.00      0.00       208
         ipo       0.00      0.00      0.00       129
   operating       0.85      1.00      0.92      5472

    accuracy                           0.87      7630
   macro avg       0.46      0.41      0.43      7630
weighted avg       0.85      0.87      0.85      7630
```

Key Takeaways

1) The Random Forest Model predicted approximately 65% of the acquired companies correctly (recall). Considering acquisitions comprised 23% of the original dataset, the accuracy is reasonably good.

2) Predicted results were given multi-class labels, but predictions were mostly binary. We likely don't have enough datapoints on companies that did an Initial Public Offering (or it’s too rare) or Closed (very common, but not heavily reflected in the dataset). The tuned model distinguished between operating and acquired only, which was 96% of the original dataset. Interestingly, the base model picked up some of the IPO's and Closed companies, but not with any meaningful accuracy.

3) The average weighted F1 score was 87%, which is useful for class imbalanced data such as this.

4) The results were virtually identical regardless of if you used the four features or the full dataset.

5) Tuning helped the model improve by about 3% from the base. I tried tuning from different lenses by changing variance thresholds and parameter grids.

6) The calculated investor score didn’t heavily influence the model, but my intuition feels like it should. A percentile rank approach may not create enough separation and using a different metric may increase accuracy further as attracting capital from a big name VC firm is a validation of your company. Individual investor amounts would be very helpful, allowing us to see contributions by investors, which could be factored into the investor score.

Overall, this model could be of use, obtaining close to 90% accuracy on the given dataset. The key application would be when companies are in early rounds and VC investors are contemplating where to invest money...which would be in companies that are likely to be acquired and the VC firm exited at a nice premium!