ALICORN

Pre-IPO Unicorn Startups investment study

WITH MACHINE LEARNING

Gitesh Poudel **Mai Do**

Quoc Tuong (Lukas) Dong **Sravanthi Chava**

Linqian Shen

Analytics, College of Professional Studies, Northeastern University

ALY6080 Integrated Experiential Learning

Instructor: Doctor Sharad Shandilya

Proposal Summary

The main project objective is to help Alicorn find the most optimized prediction model to identify the most promising Late-stage startups in the successful DevOps industry and expand their investment portfolio. Late-stage startups are companies that have proven their self-sustainability but need additional funding to turn a profit. Thus, they are often sought after by the big Venture capitalist firms, and the hunting process is nothing more than a red ocean with those of significant financial resources poised to have the upper hands.

The dataset was sent to us from a third party through Alicorn. The data collection process is not disclosed, and there are many problems that we need to take care of before actually getting into the data modeling part. We need to understand whether these companies are performing, finding the strengths and weaknesses in a particular sector, and finding the earliest indicators that a previous company would successfully score a good deal like the CEO's background or the contract size. This data will provide us with better foretelling and find companies with similar indicators guaranteed to succeed in the future.

First, we perform the Cleaning process with either Big ML or Python and then perform exploratory data analysis with Tableau, Python, R Studio. These tools will help allocate the potential patterns and examples essential to Alicorn and the new startups. Then we use different feature engineering methods to extract the most relevant components for our dependent variables before building different Supervised models to predict investment metrics such as MOIC, Post Money Valuation, Pre Money-Valuation, etc. We make sure to optimize them for the best results. We also construct an Unsupervised model method to identify investment clusters using BigML. In the end, we summarize our findings, limitations, and recommendations for future research.

Literature Review

This part gives an overview of the Annotated bibliographies presented in the past weeks by our team members:

"Assessment of Machine Learning Performance for Decision Support in Venture Capital Investments". (Arroyo et al., 2019, p. 124234)

In the first week of research on Alicorn and venture capital fund's primary motive, utilizing Machine Learning in choosing the startups with high growth potential is discussed. For Alicorn, machine learning will be a good tool for assessing the right startups to invest in and prevent investment risk by analyzing and building predictive statistic models. The business question here is how Alicorn can use machine learning to make effective decisions when choosing startups to invest in or receive a good prediction of valuation and success. This research produces some helpful advice for VC investors. It will be a reasonable consideration for applying it to Alicorn when they want to understand and predict specific metrics for their next investment. Some quantitative variables to consider can be Exit Size & Total VC capital raised and categorical variables such as Exit Type & Industry Sector. Minimizing the risk and maximizing the return on investment will be the goal of any venture fund capital and Alicorn.

"How Do Venture Capitalists Make Decisions? (Gompers et al., 2016, p. 14)

Only implementing ML models in selecting the startups will not help in achieving success for Alicorn. Significant decision areas such as deal origination, choosing investment; valuation; deal structure; value-added post the acquisition; exits; internal firm organization; and relationships with limited partners in selecting venture capital investments are discussed in the second week.

Understanding the methods adopted by successful venture capitalists by analyzing the results of about nine hundred VCs and their decision-making process about their investments and portfolios and relating these methods to Alicorn Fund is done. VCs utilize the tools and assumptions in valuing companies, post-investment strategies, metrics to evaluate the performance of VCs such as cash-on-cash return, multiple of invested capital, and IRR.

From this Alicorn need to understand the importance of deal sourcing, deal selection, and value-added post-investment. This research helped to understand that Alicorn can rely on multiples of invested capital and internal return rates.

The strategies discussed in the previous weeks need to understand more about the objective and the data. In the third week, some helpful advice for Alicorn and VC investors to better view how they can get helpful information and techniques that they can use to support the decision-making process with potential startups was discussed.

“A Recommender System for Investing in Early-Stage Enterprises”. (Luef et al. 2020)

The research in this paper suggests more tools and techniques with which to make better investment choices. The research results show that the simple collaborative filtering recommender system is more effective than the knowledge-based, and trust-based recommendations have the best performance. The article is useful and related to my research topic because besides applying machine learning and other data processing techniques. When we create a suitable recommendation system that considers investor profiles, investment decision experiences in the past, and trust among investors, it will help the analysis process before funding.

Problem Definition

The investment committee at Alicorn Fund wishes to develop a data-driven process for a specific target company investment in the Software-DevOps sector with the following objectives.

- Given the investment objective, design the pertinent analytics around the software DevOps sector based on historical VC deals (investments) and Exits (investor returns) running from 2015 to 2020to facilitate the decision.
- Given specific characteristics (features) of the target investment company (to be provided), build applicable predictive models for key metrics (such as Valuation and MOIC/ Return on Investment) using the datasets provided.
- Validate outcomes (of #1 and #2) in comparisons or as complementary to the existing expert-driven due diligence and investment process.

However, before answering all the mentioned problem, it is best to address questions that are stated in the Analysis guide:

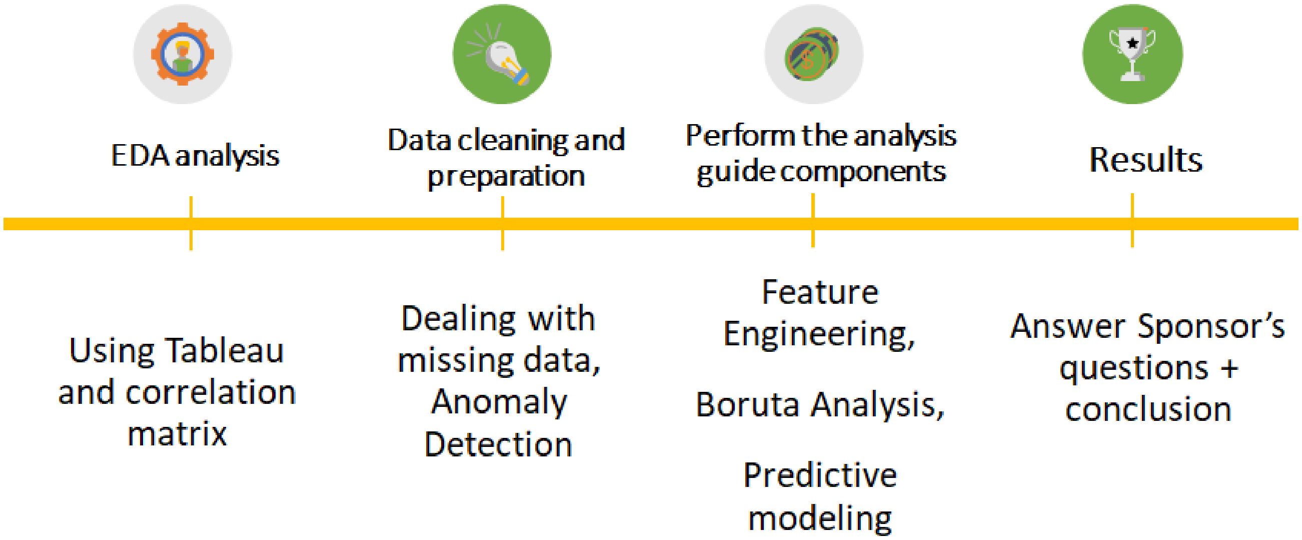
- What are the potential target variables? (ex. Acquisition Post Value / MOIC / Total Preferred Capital Raised)
- How are the potential target variables related to each other - both conceptually as well as quantitatively?
- Which time-series trends exist? How do we account for time-series trends in the models?
- Are your samples independent? If not, what measures should you take? Which algorithms should you use?

Scope of the project and Project Roadmap

The XN project is 12 weeks working project where we are assigned Alicorn as our sponsor. The 12 weeks long project includes studying the industry familiarization, data cleaning, data analysis, feature engineering, prediction models creating for the Venture capital industry in the DevOps field. In the initial weeks from week 1 to week 6, we rigorously worked on improving our knowledge in the domain since many team members are new. For example, we understand what impacts the startups with the funds raised and how AI has been beneficial to venture capital investors and DevOps pre and post COVID situations. Besides, we need to understand our sponsor domain, Alicorn, in terms of company culture, investment strategy, CEO's backgrounds, and the startup's data. For example, Alicorn focus on Tech in AI rather than AI combines with another industry, and 2 out of three companies in their portfolios are from Israel.

An important Source of evaluation that we assessed pertained to understanding the broader sociopolitical environment for Startups in the DevOps space and how we facilitated and utilized Alicorn's dataset appropriately to be relevant to Alicorn's investment strategy ultimately. To what effect can you understand the propositions of significant IPO'S DevOps startups are getting compared with other equally fascinating industries like Computer Hardware or IT services, for example, present in the dataset.

Later in this project, from the 6th-12th week, we work closely with the dataset that includes the data cleaning, data modeling, and creating visuals & dashboards. Here most parts had Alicorn's ability to evaluate startup's future Valuation/ MOIC/ Return on Investment and the potentials to introduce groundbreaking innovations to the public. To execute a successful analytical assessment, we follow this project road map as follows.



EDA analysis

Using Tableau and correlation matrix

Data cleaning and preparation

Dealing with missing data, Anomaly Detection

Perform the analysis guide components

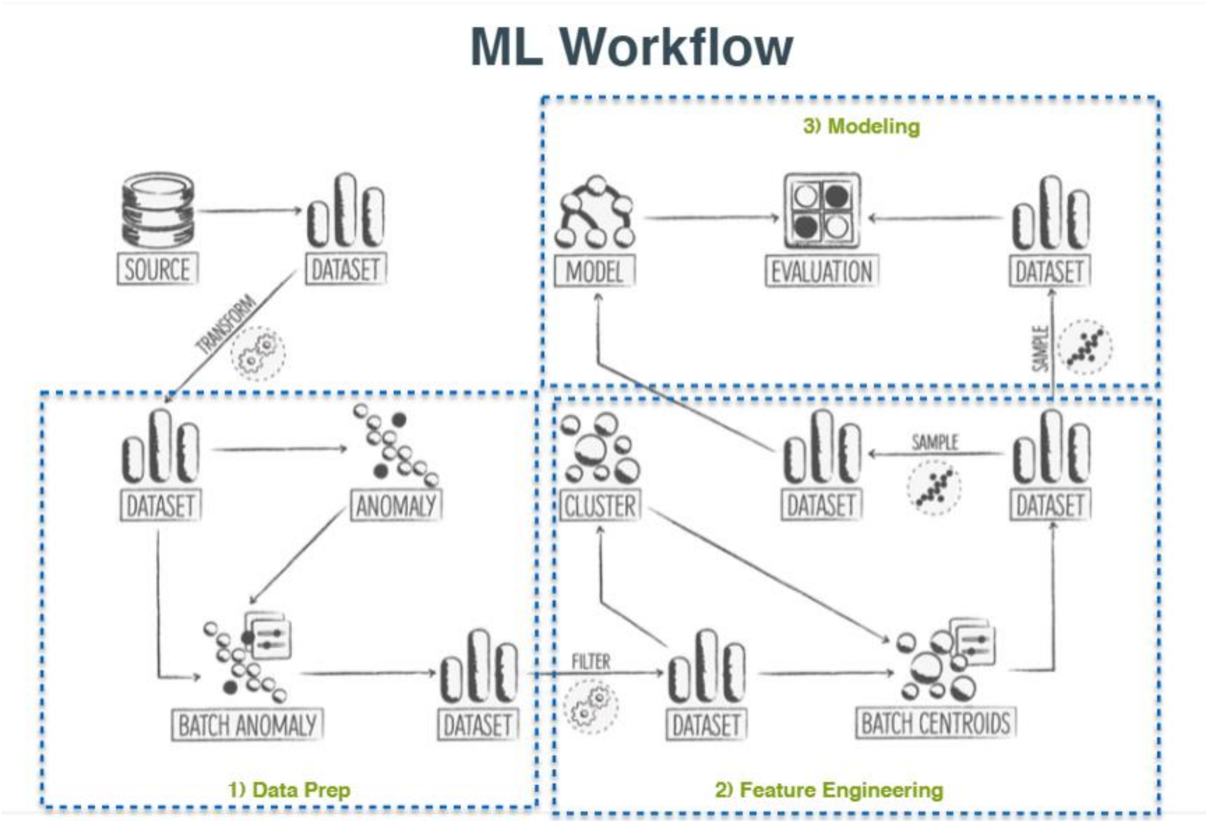
Feature Engineering,

Boruta Analysis,

Predictive modeling

Results

Answer Sponsor's questions + conclusion



ML Workflow

Limitations

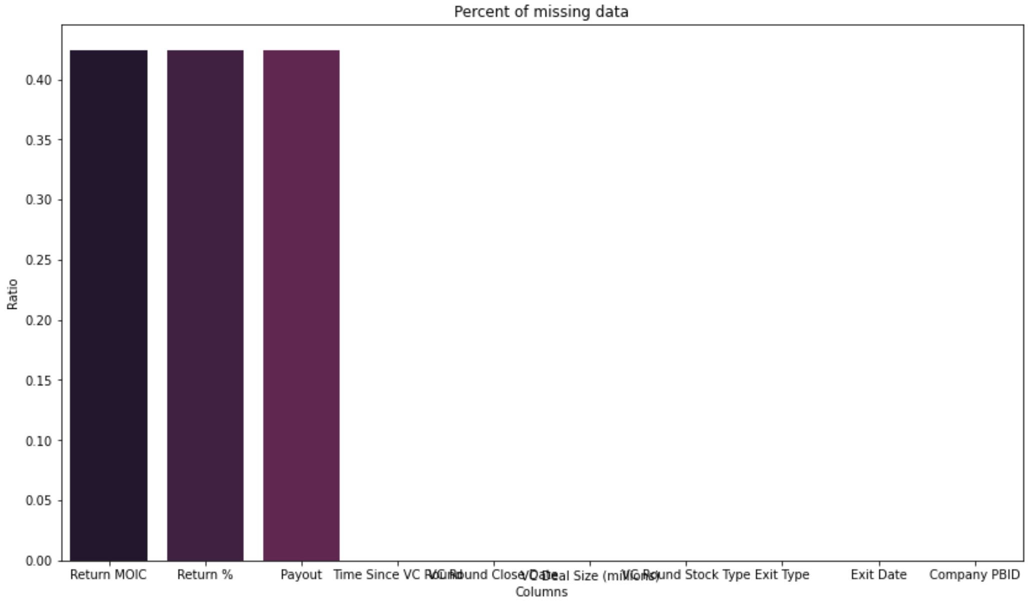
A crucial part of this analytical approach is preparing the data before starting the analysis. The data needs to be in the right shape and format for achieving better accuracy in predictions. The datasets provided by the sponsor have numerous missing values and unwanted data which might affect the accuracy of the predictive models; we might conclude an inaccurate inference from the data. In this part of the analysis, we have dealt with segregating the required and unnecessary features, dropping insignificant features, and scaling the features.

There are many challenges in the data preparation process in the project but from a respective point of view, two stand out the most:

- **Dealing with the missing data:** The datasets from the 3rd party that Alicorn Fund obtained from have many repetitive and meaningless attributes that they were not able to populate. Such a scenario created bias and variance in the beginning that we do not want. Additionally, some attributes have unstandardized categorizations like the "Last VC Deal Types" with Late-stage, Early-stage, etc. mixed with Seed A, Seed B, etc. And there are many variables/features in the datasets that will not be useful or are not even populated (deal type 2 and 3, Contingent payout, Deal status, implied EV, the debt amount, etc.) or with a negligible amount of values (EBIT, EBITDA, profit), etc.

- **Small Dataset:** The datasets are small. Usually, we will just use mean or median imputation to solve the missing numeric values or randomly assign them with categorical values. The datasets were given contain more than 40% of missing values. Although we have used various techniques to deal with missing data to deal with this problem such as imputation Mean or Median, or machine learning models to fill the missing values, the final quality of the models still gets effect because of the large amount of missing data in small datasets. When the imputation failed, it will lead to inaccurate predictive models and misleading which can affect the decision-making process of the sponsor.

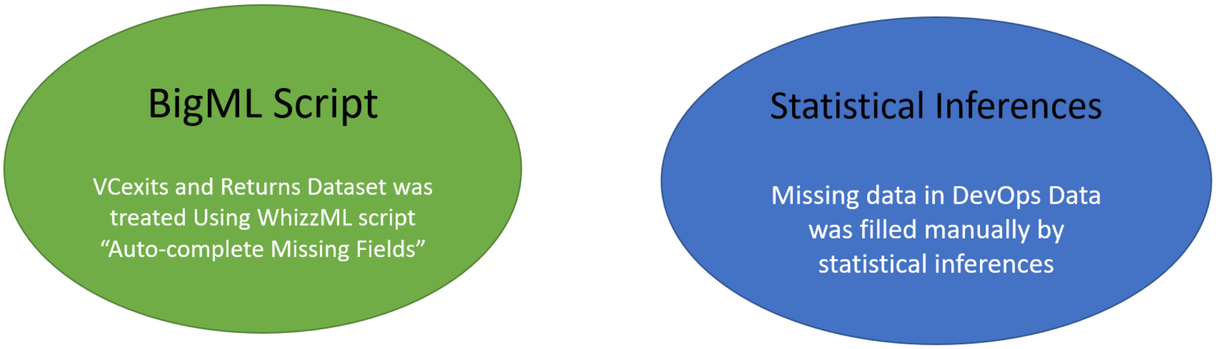
For example, in the Returns by series dataset, the amount of missing data is more than 40% of the dataset with 104 missing values for Return MOIC, Return %, and Payout.



Having an outlook of the percentage of values missing in each of these metrics, helps us have an overview of what we can do to deal with these missing for the best option outcomes.

Data Preparation process

VC Exits and Returns Dataset were treated using WhizzML from BigML script "Auto-complete Missing Fields". Three datasets contain sensitive financial information. Hence, we used a BigML native script which trains individual ensemble and predicts a missing value for each feature. However, the same cannot be said for the DevOps dataset. It has a huge dimension, many variables are missing a significantly huge portion of the dataset. BigML script failed to impute missing values due to the lack of data available. Hence, manual statistical inferences imputation methods were used.



BigML Script

VCexits and Returns Dataset was treated Using WhizzML script "Auto-complete Missing Fields"

Statistical Inferences

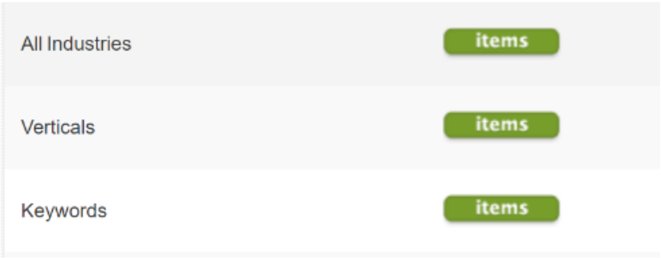
Missing data in DevOps Data was filled manually by statistical inferences

1. Big ML script (VC Exists and Returns by series)

- ***Step 1: Loading data and formatting***

Three datasets are added as data sources and configured. Dataset configuration includes formatting, selecting proper datatype, removing unnecessary fields missing most of the information, and understanding the data distribution of each field.

DevOps and VC exist contains few companies' attributes like vertical and keywords which are comma-separated text, since these fields carry

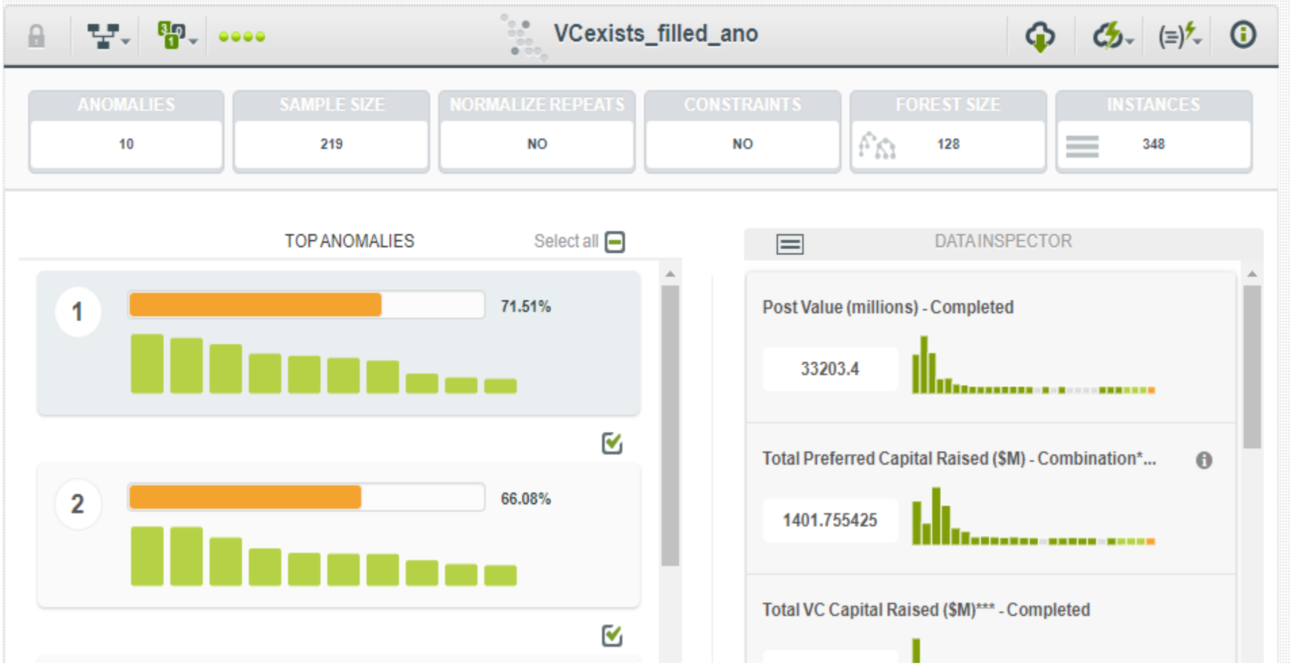


All Industries	items
Verticals	items
Keywords	items

valuable information regarding the domain and operating function of each industry, this can be both a good predictor and something we would be interested to analyze in the final model. By default, such a text field can be understood as a categorical field since the combination of strings would be different and the model would not be able to correlate individual words in the text field. To ensure we extract as much information as we can from these fields, we will format these fields as comma-separated 'Items', doing this each of the comma-separated words will be considered as a single class.

CEO education columns in the DevOps dataset are not standardized but a plain text field, to map it well in the end model we will change it to text format.

<u>Step 2: Anomaly Detection</u>



VCexists_filled_ano

ANOMALIES	SAMPLE SIZE	NORMALIZE REPEATS	CONSTRAINTS	FOREST SIZE	INSTANCES
10	219	NO	NO	128	348

TOP ANOMALIES

Select all

1 71.51%

2 66.08%

DATA INSPECTOR

Post Value (millions) - Completed

33203.4

Total Preferred Capital Raised ($M) - Combination*...

1401.755425

Total VC Capital Raised ($M)*** - Completed

Anomaly and outliers in our working dataset can negatively influence predicting power of the model rendering model less accurate. However, we would not be using the Interquartile range to remove outliers' instances because in most cases the financial distribution is skewed right. Since very few companies have a higher financial performance like MOIC and post valuation, they may seem like outlier relative to general distribution. However, we cannot remove them as outliers because the objective of the final model would be to identify companies that are expected to have good returns of investment. Having said that, we would still want to remove extreme anomalies which would be noise to the system. In VC exists dataset, we found two such anomalies where post value was $33.2 billion and total preferred capital raised was $1.4 billion. Two of the instances among 10 detected as anomalies by the unsupervised model were removed from data after proper inspection of all 10 instances.

No extreme anomalies were found on the Returns and DevOps dataset.

- ***Step 3: Filling Missing value***

Algorithm used: https://bigml.com/dashboard/script/603f90aa134bd326030000db

This script on the WhizzML platform of BigML is written in a platform-specific language, however, after running this script in our dataset, we can identify what models were used to fill the missing data. This script trains one ensemble model of each field that is missing data by considering the missing data field as target variable, it considers instances with complete data as the training set to train the model and uses that model to predict targets which are necessarily the missing fields.

VC exists and the Returns dataset was treated with Autocomplete script, the resultant dataset as expected does not have any missing value. However, we were not able to use the same process to impute DevOps dataset because many fields in it are missing a significantly huge portion of the dataset. Since some fields are missing 80-90% of data, it is not possible to train the model to predict the missing value.

2. Statistical Inferences (DevOps)

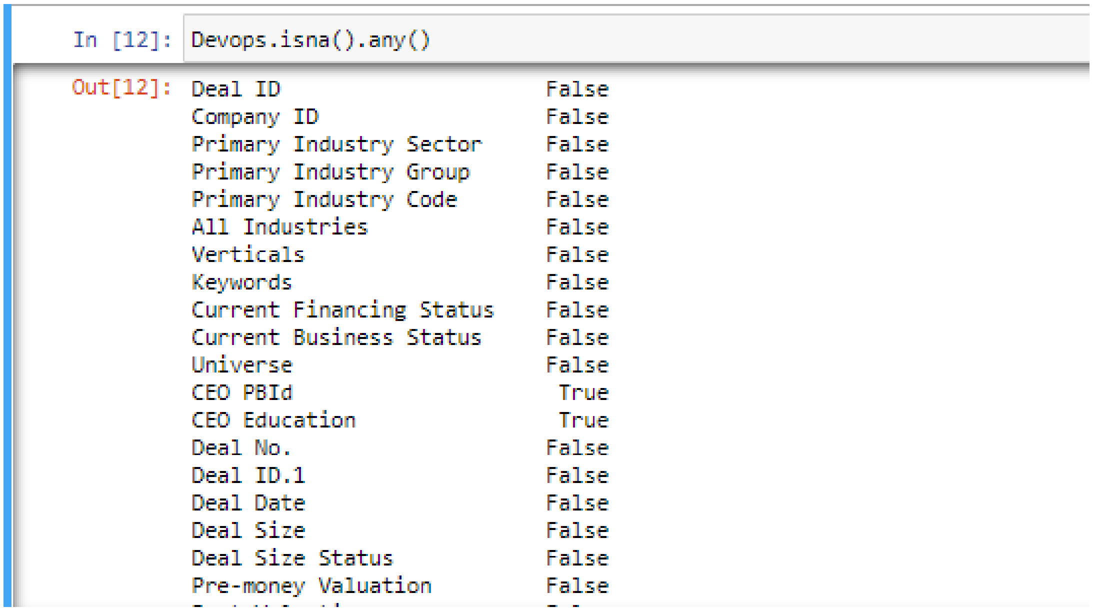
- ***Step 1: Dealing with the missing dataset.***

After eliminating some of the variables, thanks to the “Analysis guide” proposed by Doctor Shandilya Sharad, we continued to double-check and see how we can fill in the rest of the missing data. The actions were taken as followed:

- Deleted any variables with less than 20% of the entries and Deal Type 2”, “VC Round-Up/ Down/Flat”, “Employees.”
- “Current Employees”, “Raised to date”, “Deal Size”, “Total Invested Equity”: used Means of the variables,
- “Employees”: used Mode of the variables because they have significant scale and some outliners that do not represent the whole variable.
- “New Investor”: with “0” because they are not missing data but instead there is no one or no numeric values for that entry.
- “Financing status”: with “None” because they are not missing data but instead, there is no one or no categorical values for those entries.
- “VC Round”: with “No round” since they only have missing inputs when “Deal Type” have the value of “Grant”, “Accelerator”, “Incubator”, etc.
- “Year Founded”: with 2014 because that is the most common entries.
- “Series”: with “No Series” because some of the companies are in transition stages which is why they are missing data in this variable.
- “CEO PBlD”, “CEO Education”: got manually filled out.
- “Follow on Investor”: “Investor” minus “New Investor.”

- “Deal Size Status”: with the same “Deal Size Status” if they must have the same “Company PBID.”
- “PreMoney Value” and “Post value”: with -1. The reason why x=”-1” is to allow numeric operations to be performed on them after “?” caused them all to be factors. “-1” is also a good placeholder since it distinctively shows something with that input.

At the end of the process, we will have this cleaned dataset as below:



```
In [12]: Devops.isna().any()

Out[12]: Deal ID                     False
         Company ID                  False
         Primary Industry Sector     False
         Primary Industry Group      False
         Primary Industry Code       False
         All Industries              False
         Verticals                   False
         Keywords                    False
         Current Financing Status    False
         Current Business Status     False
         Universe                    False
         CEO PBId                     True
         CEO Education                True
         Deal No.                    False
         Deal ID.1                   False
         Deal Date                   False
         Deal Size                   False
         Deal Size Status            False
         Pre-money Valuation         False
```

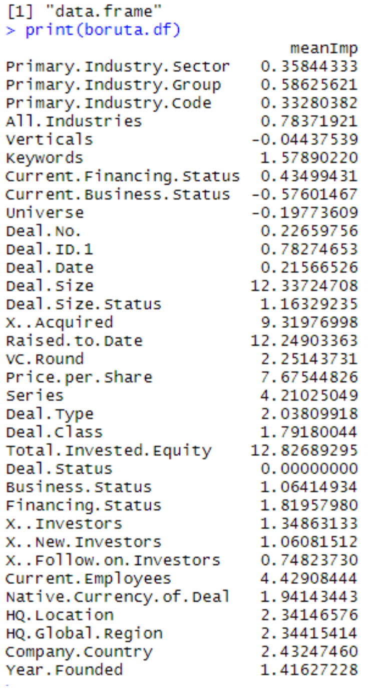
- ***<u>Step 2: Feature engineering with Boruta Analysis</u>***

We split the cleaned dataset into two; one includes the “Pre-money valuation”, the other “Post Valuation”. Because when the entry happens to have the “Pre-money valuation”, then it will have “Post “Valuation”, thus significantly altering the Boruta analysis and prediction models’ result at the end. Boruta is a feature engineering to assess variables' level of importance. It is done by joining them with the original predictions and constructs a random forest on the merged dataset. Then, we will compare the original dataset and the randomized variables to eliminate those with low importance ratio than the randomized variables. (Deepanshu Bhalla, 2017).

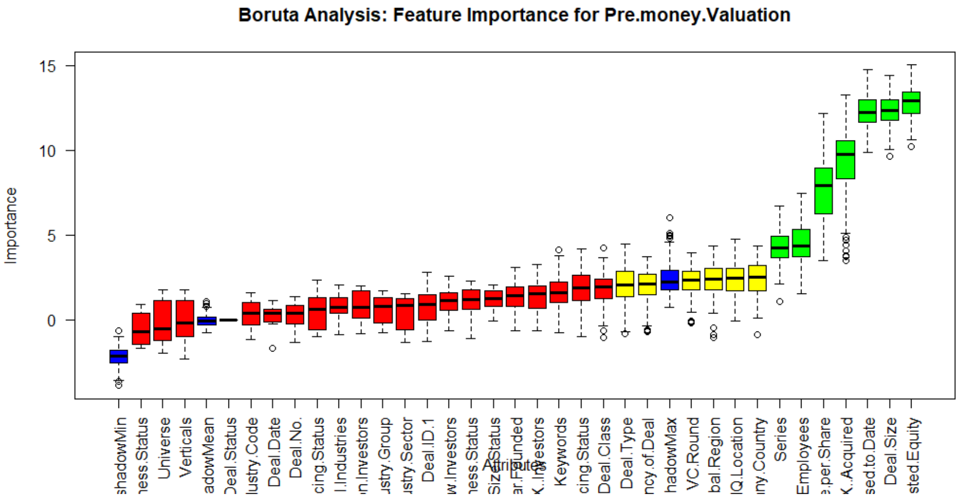
Boruta analysis is the cutting-edge approach to feature engineering, thanks to its' many advantages:

- Appropriate for both classification and regression projects because it considers multi-variable relationships.
- A betterment of random forest variable importance measure and can handle interplay between variables as well as changing the nature of the random forest importance measure.

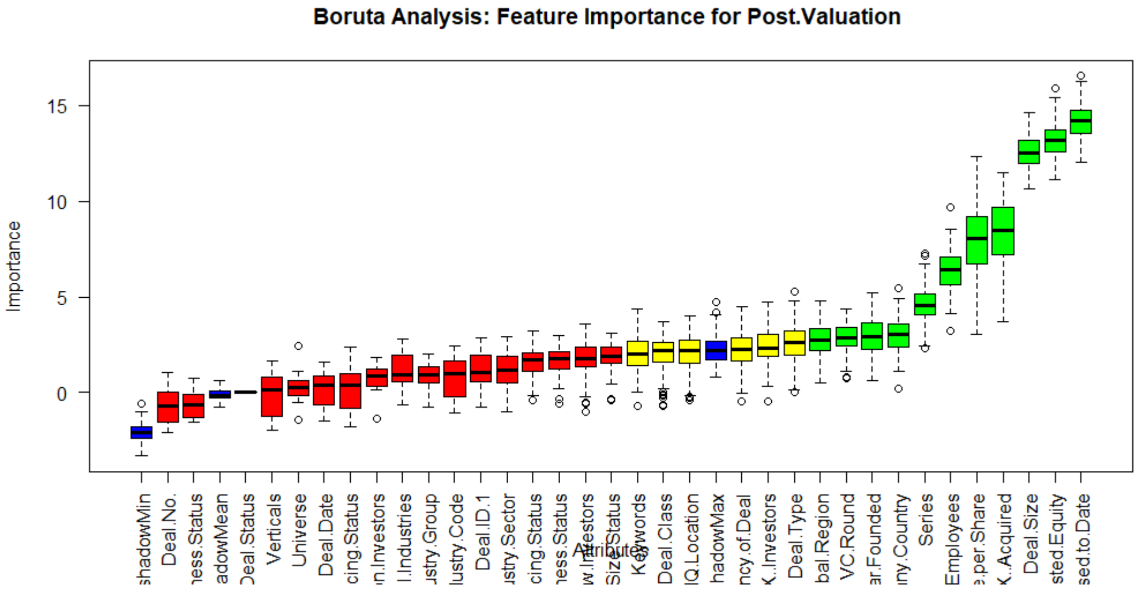
In this project, I will be using the Boruta package from R because it has better sets of illustration functions. The table and the graph below indicate that we have 11 variables that are confirmed to influence the "Pre-money valuation" variable strongly.

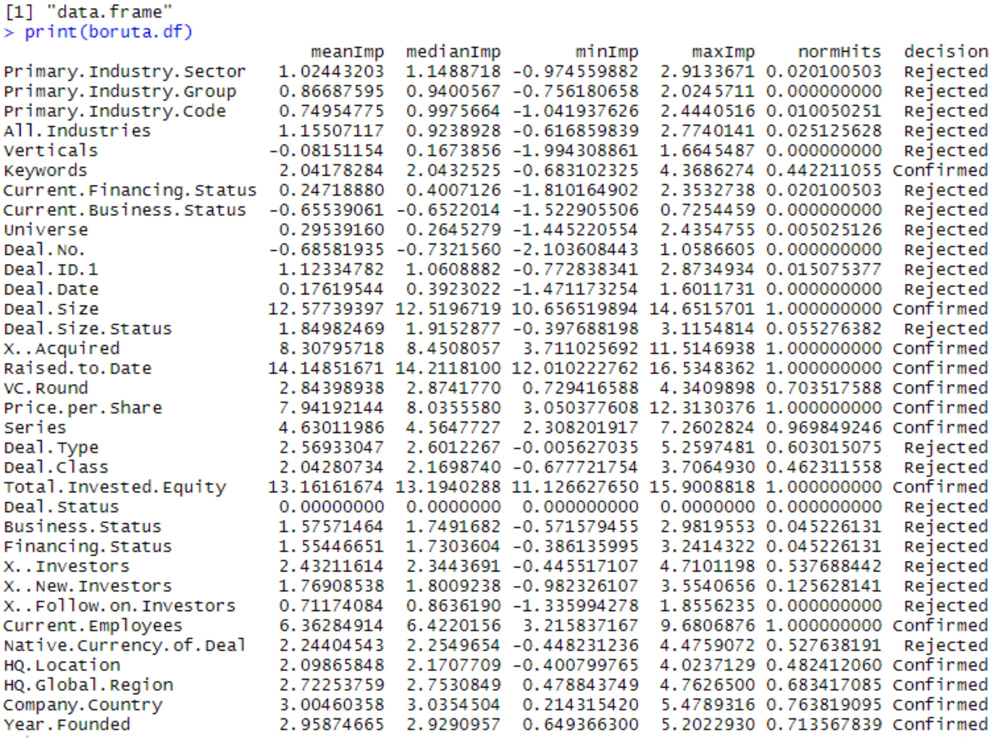


```
[1] "data.frame"
> print(boruta.df)
                                meanImp
Primary.Industry.Sector      0.35844333
Primary.Industry.Group       0.58625621
Primary.Industry.Code        0.33280382
All.Industries               0.78371921
Verticals                   -0.04437539
Keywords                     1.57890220
Current.Financing.Status     0.43499431
Current.Business.Status     -0.57601467
Universe                    -0.19773609
Deal.No.                     0.22659756
Deal.ID.1                    0.78274653
Deal.Date                    0.21566526
Deal.Size                   12.33724708
Deal.Size.Status             1.16329235
X..Acquired                  9.31976998
Raised.to.Date              12.24903363
VC.Round                     2.25143731
Price.per.Share              7.67544826
Series                       4.21025049
Deal.Type                    2.03809918
Deal.Class                   1.79180044
Total.Invested.Equity       12.82689295
Deal.Status                  0.00000000
Business.Status              1.06414934
Financing.Status             1.81957980
X..Investors                 1.34863133
X..New.Investors             1.06081512
X..Follow.on.Investors       0.74823730
Current.Employees            4.42908444
Native.Currency.of.Deal      1.94143443
HQ.Location                  2.34146576
HQ.Global.Region             2.34415414
Company.Country              2.43247460
Year.Founded                 1.41627228
```



The table and the graph below, on the other hand, show that we have 14 variables that are confirmed to influence the "Post Valuation" variable strongly. The critical variables between these two datasets are very similar except for "Post Valuation". "Post Valuation" set has 3 additional variables: “Year Founded”, “HQ Location, "Financing Status” (Venture Capital-Backed, Angle Backed, Corporations, etc.).

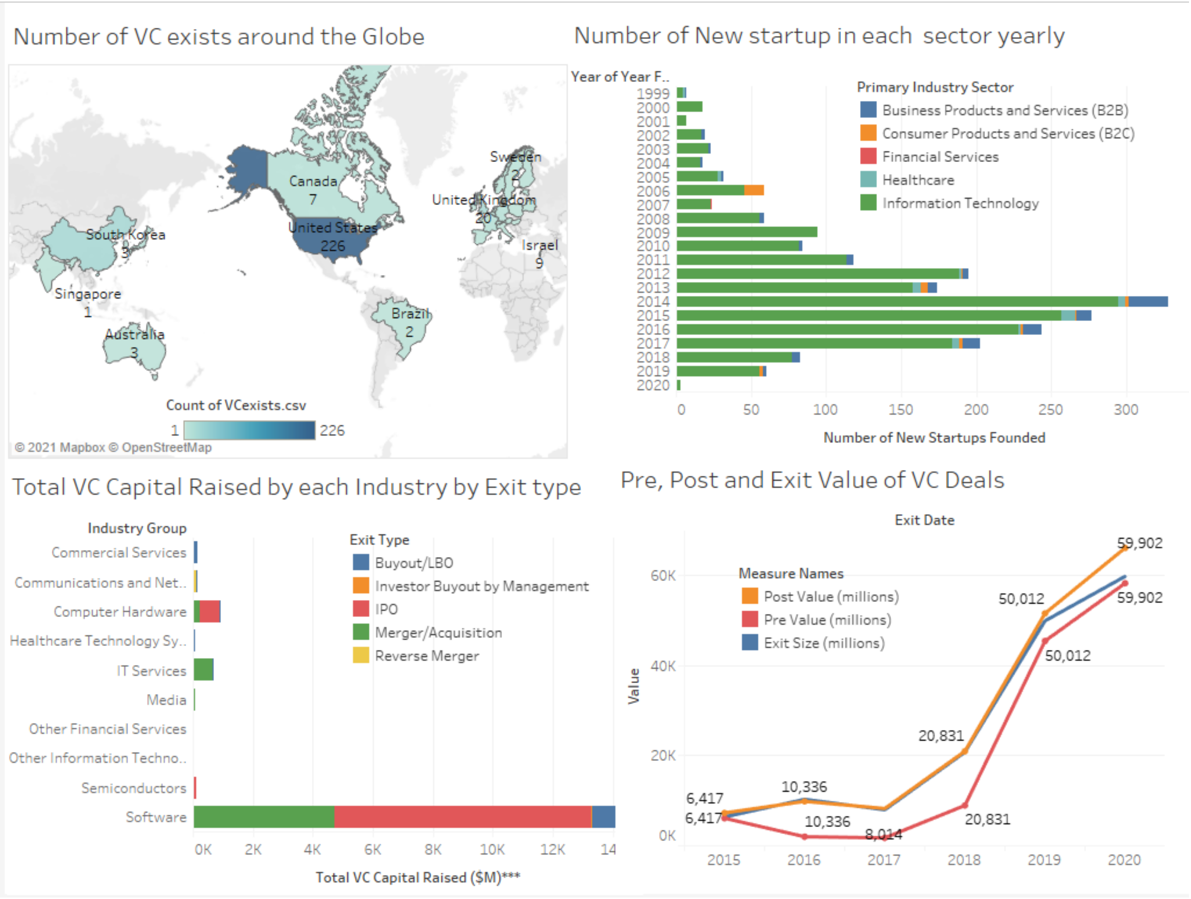




```
[1] "data.frame"
> print(boruta.df)
                            meanImp  medianImp      minImp     maxImp    normHits  decision
Primary.Industry.Sector   1.02443203  1.1488718 -0.974559882  2.9133671 0.020100503  Rejected
Primary.Industry.Group    0.86687595  0.9400567 -0.756180658  2.0245711 0.000000000  Rejected
Primary.Industry.Code     0.74954775  0.9975664 -1.041937626  2.4440516 0.010050251  Rejected
All.Industries            1.15507117  0.9238928 -0.616859839  2.7740141 0.025125628  Rejected
Verticals                -0.08151154  0.1673856 -1.994308861  1.6645487 0.000000000  Rejected
Keywords                  2.04178284  2.0432525 -0.683102325  4.3686274 0.442211055 Confirmed
Current.Financing.Status  0.24718880  0.4007126 -1.810164902  2.3532738 0.020100503  Rejected
Current.Business.Status  -0.65539061 -0.6522014 -1.522905506  0.7254459 0.000000000  Rejected
Universe                  0.29539160  0.2645279 -1.445220554  2.4354755 0.005025126  Rejected
Deal.No.                 -0.68581935 -0.7321560 -2.103608443  1.0586605 0.000000000  Rejected
Deal.ID.1                 1.12334782  1.0608882 -0.772838341  2.8734934 0.015075377  Rejected
Deal.Date                 0.17619544  0.3923022 -1.471173254  1.6011731 0.000000000  Rejected
Deal.Size                12.57739397 12.5196719 10.656519894 14.6515701 1.000000000 Confirmed
Deal.Size.Status          1.84982469  1.9152877 -0.397688198  3.1154814 0.055276382  Rejected
X..Acquired               8.30795718  8.4508057  3.711025692 11.5146938 1.000000000 Confirmed
Raised.to.Date           14.14851671 14.2118100 12.010222762 16.5348362 1.000000000 Confirmed
VC.Round                  2.84398938  2.8741770  0.729416588  4.3409898 0.703517588 Confirmed
Price.per.Share           7.94192144  8.0355580  3.050377608 12.3130376 1.000000000 Confirmed
Series                    4.63011986  4.5647727  2.308201917  7.2602824 0.969849246 Confirmed
Deal.Type                 2.56933047  2.6012267 -0.005627035  5.2597481 0.603015075  Rejected
Deal.Class                2.04280734  2.1698740 -0.677721754  3.7064930 0.462311558  Rejected
Total.Invested.Equity    13.16161674 13.1940288 11.126627650 15.9008818 1.000000000 Confirmed
Deal.Status               0.00000000  0.0000000  0.000000000  0.0000000 0.000000000  Rejected
Business.Status           1.57571464  1.7491682 -0.571579455  2.9819553 0.045226131  Rejected
Financing.Status          1.55446651  1.7303604 -0.386135995  3.2414322 0.045226131  Rejected
X..Investors              2.43211614  2.3443691 -0.445517107  4.7101198 0.537688442  Rejected
X..New.Investors          1.76908538  1.8009238 -0.982326107  3.5540656 0.125628141  Rejected
X..Follow.on.Investors    0.71174084  0.8636190 -1.335994278  1.8556235 0.000000000  Rejected
Current.Employees         6.36284914  6.4220156  3.215837167  9.6806876 1.000000000 Confirmed
Native.Currency.of.Deal   2.24404543  2.2549654 -0.448231236  4.4759072 0.527638191  Rejected
HQ.Location               2.09865848  2.1707709 -0.400799765  4.0237129 0.482412060 Confirmed
HQ.Global.Region          2.72253759  2.7530849  0.478843749  4.7626500 0.683417085 Confirmed
Company.Country           3.00460358  3.0354504  0.214315420  5.4789316 0.763819095 Confirmed
Year.Founded              2.95874665  2.9290957  0.649366300  5.2022930 0.713567839 Confirmed
```

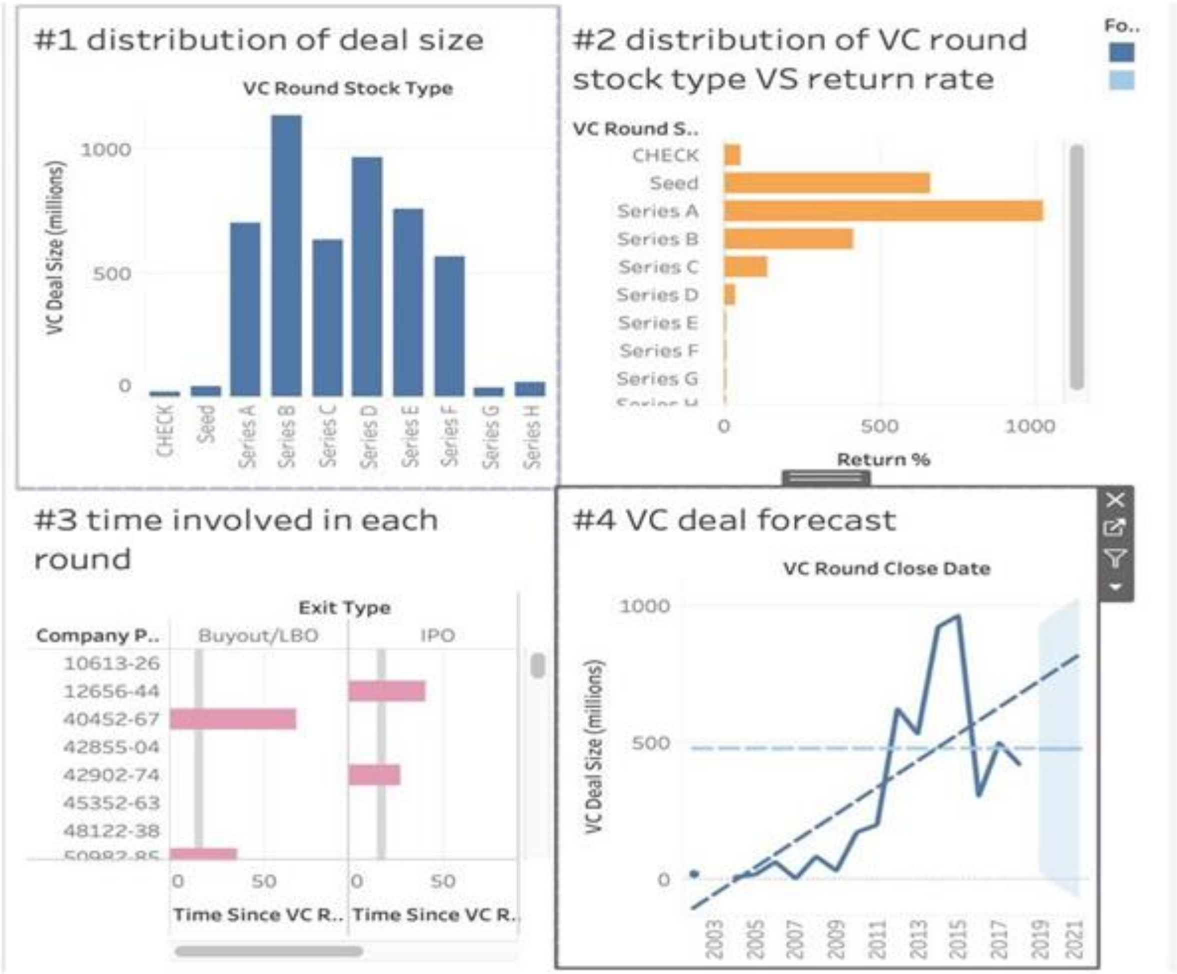
Exploratory Data Analysis and Data Visualization

To develop an effective business proposal and/or a deliverable for Alicorn, we have taken an overall perspective in understanding the factors affecting funding of many startups in different stages. Using Tableau to get some overview of the datasets as below:



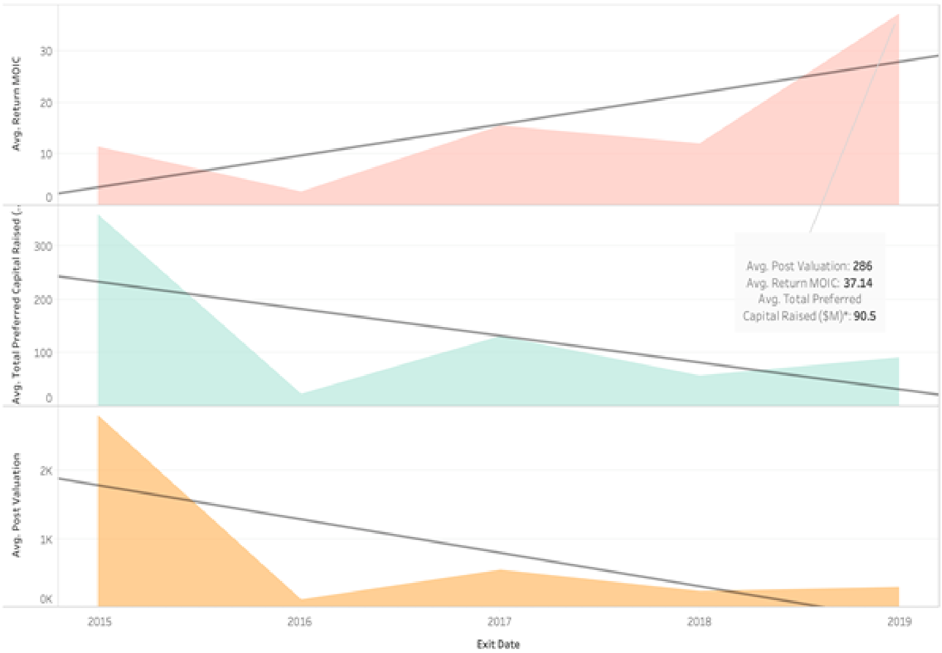
Overview of the VC Exit dataset

In this dashboard, the upper left geographic plot shows that the start-ups are most seen in North American followed by Europe (including the UK) and Asia. The bar ploy on the top right side shows interesting insights: the number of new start-ups founded in 2015 is approximately 6 times more than in 2005. Though a decreasing trend after 2015, the overall trend shows an increase of B2B start-ups. The lower left plot showcases that software start-ups are the dominant player, and that merge and acquisition is most preferred in raising funding in the IT service industry with the largest amount in IPO for exit type. The lower right line plot shows the trend of post value, pre-value, and exit size from 2015 to 2020. The value of VC deals starts to increase from 2018 to 2020.



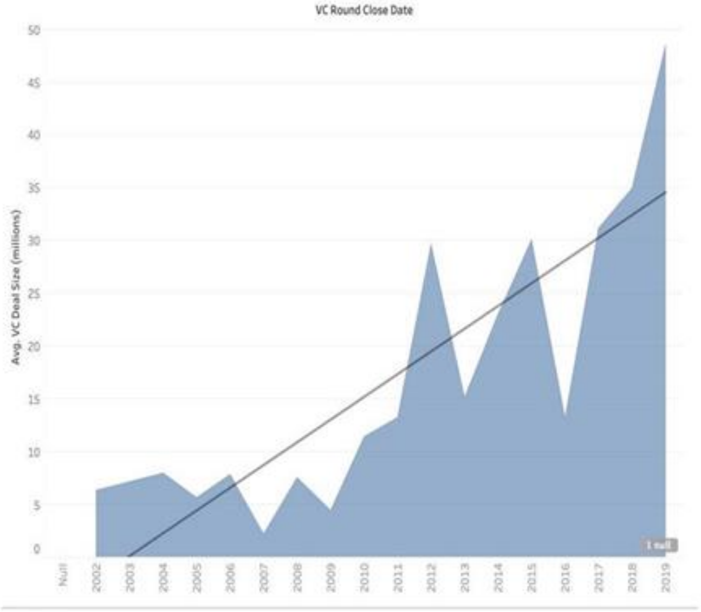
Profit versus the funding round.

You see that when VC funding always comes in at round B and round D. About 95 quantiles of return rate happen between seed and round D, which means that when start-ups IPO after seed but before series D, the return rate is comparably high. The right-down plot shows the growing trend of deal size. We forecast a 48 million size in 2021.



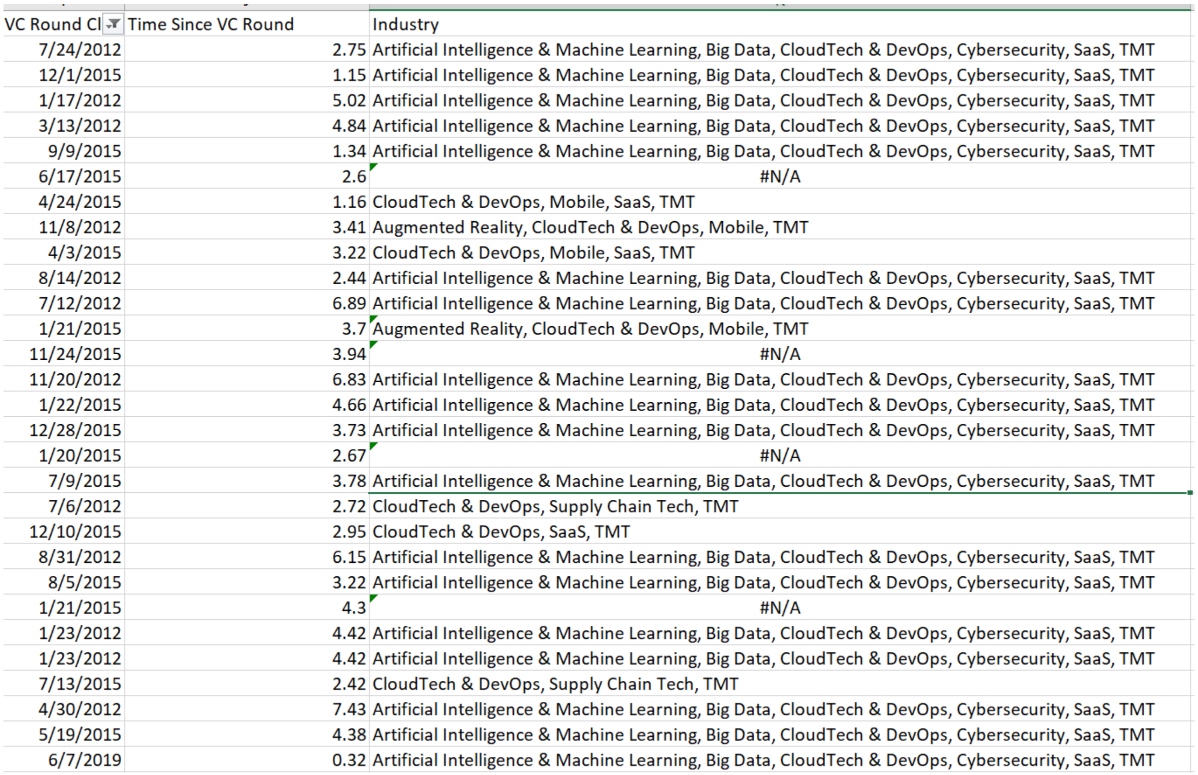
Return MOIC, Total Capital Raised, and Post valuation trends from 2015 to 2019.

Overall, the Return MOIC has the opposite trend when it increased slightly while the value of Total Preferred Capital Raised and Post valuation decreased from 2015 to 2019. We can safely say that these features are negatively correlated to each other.



VC Deal size and VC Round Close Date trend from 2002 to 2019

VC deal size trend increases from 2002 to 2019. We see a spark growth of VC deal size in 2012, 2015, and 2019. The sudden spikes and followed declines could be explained by industry:

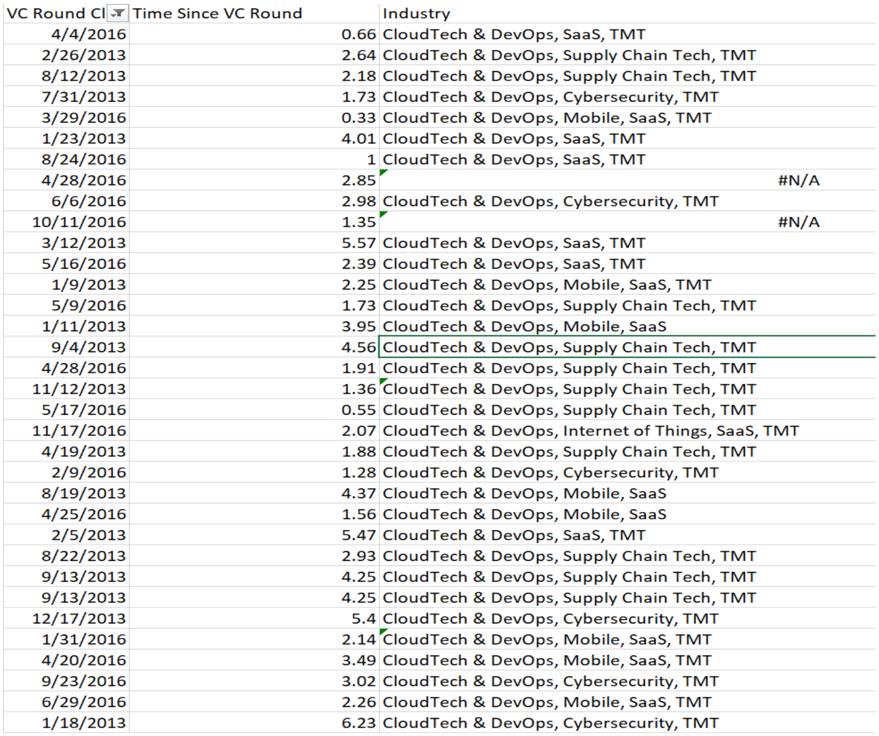


VC Round Cl	Time Since VC Round	Industry
7/24/2012	2.75	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
12/1/2015	1.15	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
1/17/2012	5.02	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
3/13/2012	4.84	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
9/9/2015	1.34	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
6/17/2015	2.6	#N/A
4/24/2015	1.16	CloudTech & DevOps, Mobile, SaaS, TMT
11/8/2012	3.41	Augmented Reality, CloudTech & DevOps, Mobile, TMT
4/3/2015	3.22	CloudTech & DevOps, Mobile, SaaS, TMT
8/14/2012	2.44	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
7/12/2012	6.89	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
1/21/2015	3.7	Augmented Reality, CloudTech & DevOps, Mobile, TMT
11/24/2015	3.94	#N/A
11/20/2012	6.83	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
1/22/2015	4.66	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
12/28/2015	3.73	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
1/20/2015	2.67	#N/A
7/9/2015	3.78	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
7/6/2012	2.72	CloudTech & DevOps, Supply Chain Tech, TMT
12/10/2015	2.95	CloudTech & DevOps, SaaS, TMT
8/31/2012	6.15	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
8/5/2015	3.22	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
1/21/2015	4.3	#N/A
1/23/2012	4.42	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
1/23/2012	4.42	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
7/13/2015	2.42	CloudTech & DevOps, Supply Chain Tech, TMT
4/30/2012	7.43	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
5/19/2015	4.38	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT
6/7/2019	0.32	Artificial Intelligence & Machine Learning, Big Data, CloudTech & DevOps, Cybersecurity, SaaS, TMT

VC Deal size and industries

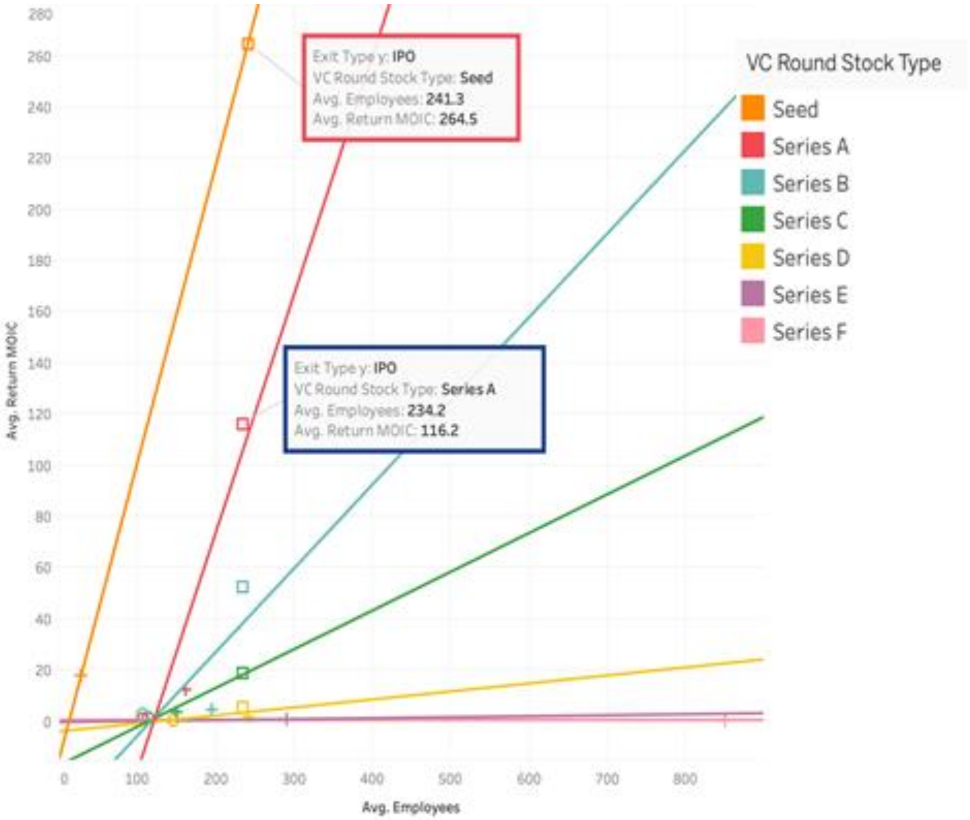
When we look at the relationship between the VC Deal size and industries, we can see that companies who have invested in years 2012,2015 and 2019 on AI and ML i.e., in the sector of Artificial Intelligence & Machine Learning, Big Data, Cloud Tech & DevOps, Cybersecurity, SaaS, TMT have exhibited greater average VC Deal size.

Whereas companies are majorly investing in the fields of Cloud Tech & DevOps, Supply Chain Tech, TMT in 2013 and 2016 have seen a lesser average VC deal size. This suggests that Alicorns investment in AI and ML fields startups can be preferred when compared to other sectors while considering the average VC deal size.



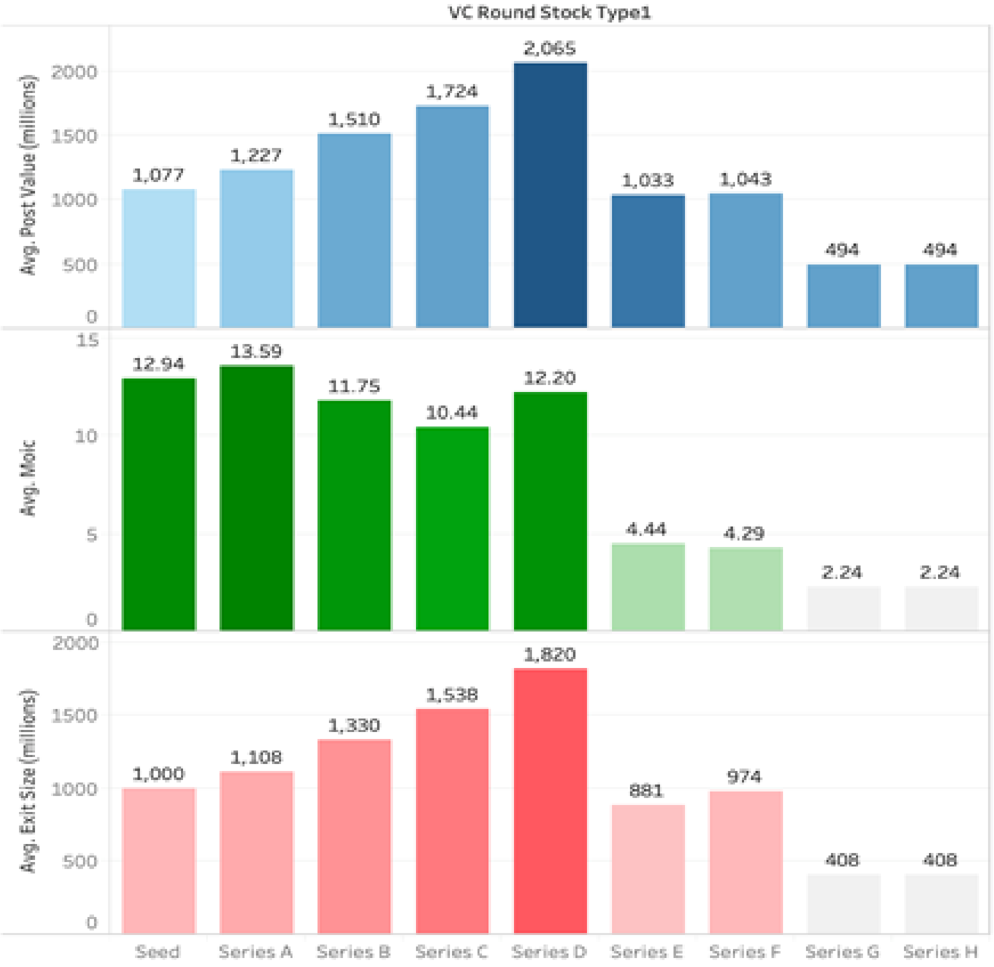
VC Round Cl	Time Since VC Round	Industry
4/4/2016	0.66	CloudTech & DevOps, SaaS, TMT
2/26/2013	2.64	CloudTech & DevOps, Supply Chain Tech, TMT
8/12/2013	2.18	CloudTech & DevOps, Supply Chain Tech, TMT
7/31/2013	1.73	CloudTech & DevOps, Cybersecurity, TMT
3/29/2016	0.33	CloudTech & DevOps, Mobile, SaaS, TMT
1/23/2013	4.01	CloudTech & DevOps, SaaS, TMT
8/24/2016	1	CloudTech & DevOps, SaaS, TMT
4/28/2016	2.85	#N/A
6/6/2016	2.98	CloudTech & DevOps, Cybersecurity, TMT
10/11/2016	1.35	#N/A
3/12/2013	5.57	CloudTech & DevOps, SaaS, TMT
5/16/2016	2.39	CloudTech & DevOps, SaaS, TMT
1/9/2013	2.25	CloudTech & DevOps, Mobile, SaaS, TMT
5/9/2016	1.73	CloudTech & DevOps, Supply Chain Tech, TMT
1/11/2013	3.95	CloudTech & DevOps, Mobile, SaaS
9/4/2013	4.56	CloudTech & DevOps, Supply Chain Tech, TMT
4/28/2016	1.91	CloudTech & DevOps, Supply Chain Tech, TMT
11/12/2013	1.36	CloudTech & DevOps, Supply Chain Tech, TMT
5/17/2016	0.55	CloudTech & DevOps, Supply Chain Tech, TMT
11/17/2016	2.07	CloudTech & DevOps, Internet of Things, SaaS, TMT
4/19/2013	1.88	CloudTech & DevOps, Supply Chain Tech, TMT
2/9/2016	1.28	CloudTech & DevOps, Cybersecurity, TMT
8/19/2013	4.37	CloudTech & DevOps, Mobile, SaaS
4/25/2016	1.56	CloudTech & DevOps, Mobile, SaaS
2/5/2013	5.47	CloudTech & DevOps, SaaS, TMT
8/22/2013	2.93	CloudTech & DevOps, Supply Chain Tech, TMT
9/13/2013	4.25	CloudTech & DevOps, Supply Chain Tech, TMT
9/13/2013	4.25	CloudTech & DevOps, Supply Chain Tech, TMT
12/17/2013	5.4	CloudTech & DevOps, Cybersecurity, TMT
1/31/2016	2.14	CloudTech & DevOps, Mobile, SaaS, TMT
4/20/2016	3.49	CloudTech & DevOps, Mobile, SaaS, TMT
9/23/2016	3.02	CloudTech & DevOps, Cybersecurity, TMT
6/29/2016	2.26	CloudTech & DevOps, Mobile, SaaS, TMT
1/18/2013	6.23	CloudTech & DevOps, Cybersecurity, TMT

But what about the relationship between the VC stock round and other factors?



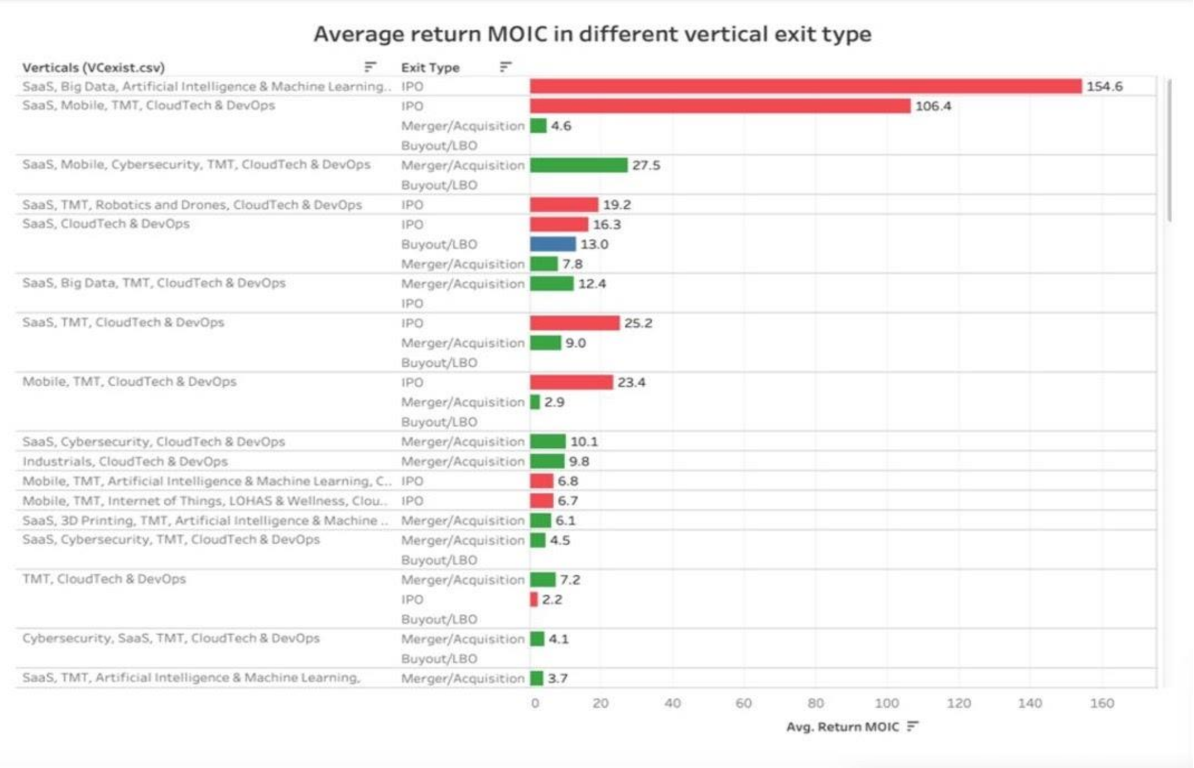
Relationship between Return MOIC, Employees, VC Round Stock Type, and Exit Type

The highest return MOIC belongs to Seed and Series A, followed that by series B and C (early-stage startups). Additionally, the relationship between Return MOIC with the number of employees is positive, which means the companies that have more employees will have higher return MOIC. The highest Return is 264.5 with 241 employees in the seed round.



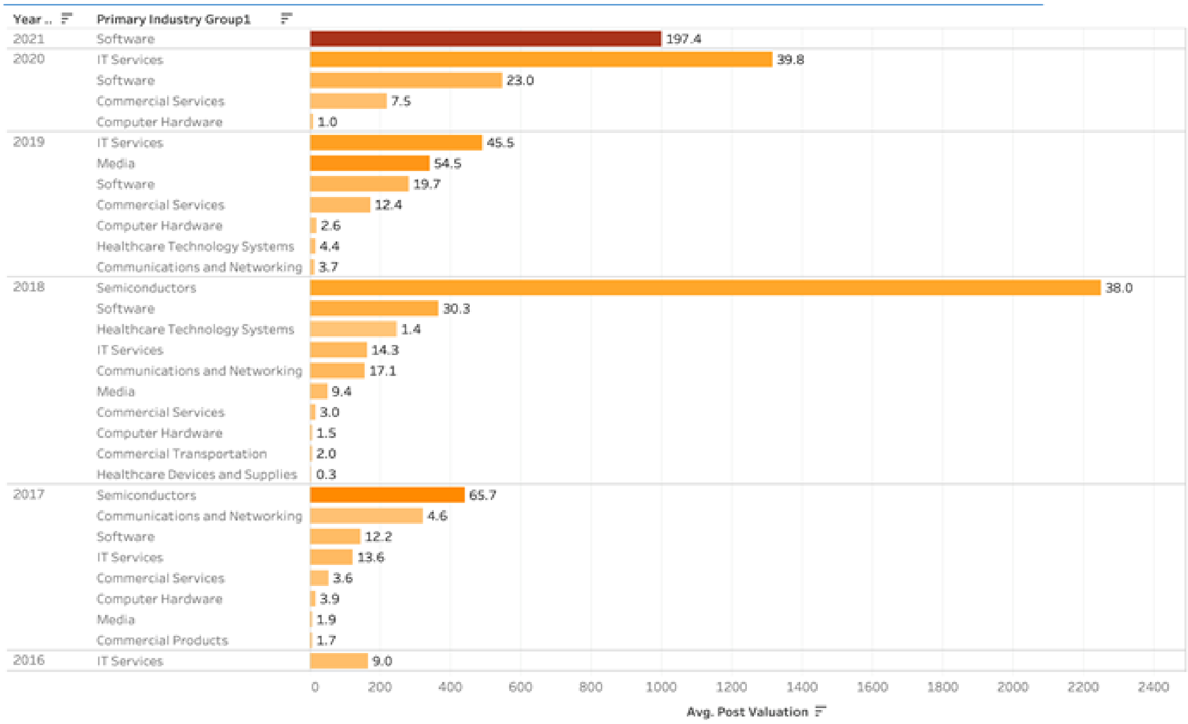
MOIC, Post Value, and Exit Size by different VC Round Stock Type

Overall, Post Value, MOIC, and Exit Size are high value from Seed to series D (Early-stage investment) and then lower from series E.



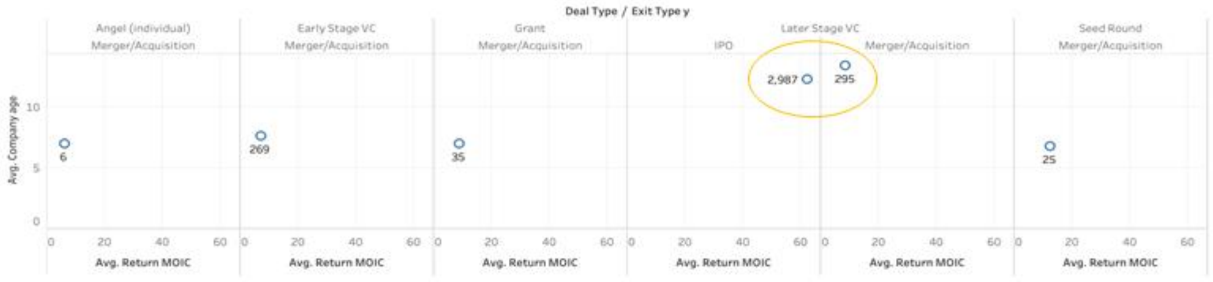
Relationship between average Return MOIC in different vertical exit types.

From the plot above, we can see the verticals such as SaaS, Artificial Intelligence & Machine Learning, TMT, Big Data, Cloud Tech & DevOps have the highest return with more than 154.6 return MOIC with exit type is IPO.



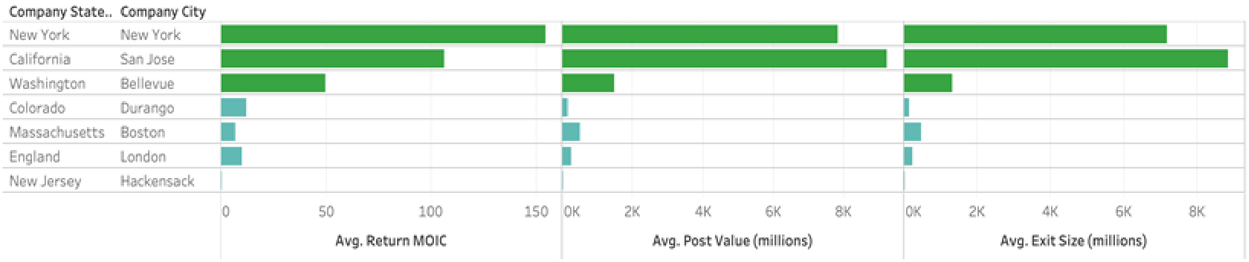
Post Valuation from primary industries from 2015- 2021

Post valuation values focus on the Software industry and IT services in 2020 and 2021. In 2018, the highest post valuation belongs to Semiconductor.



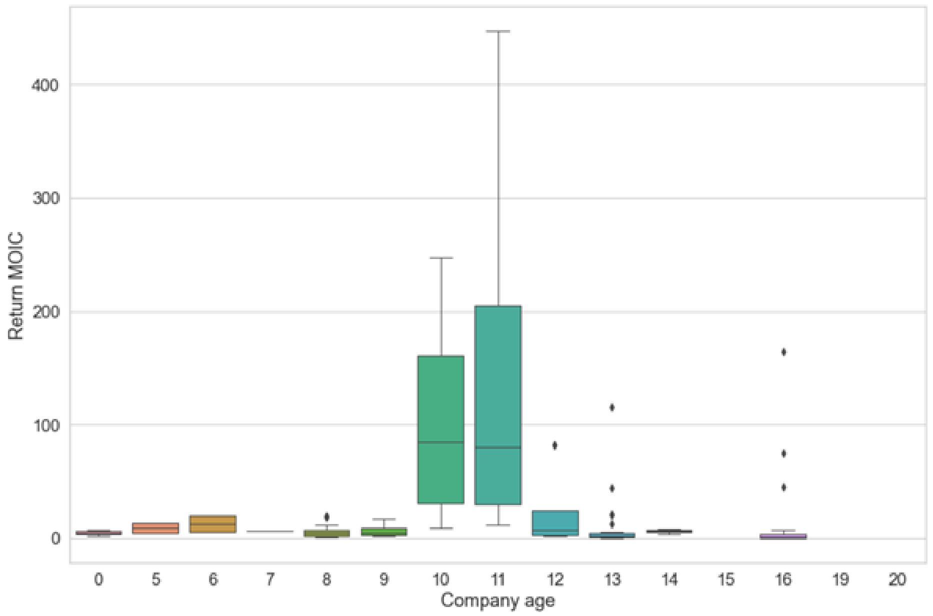
Relationship between Return MOIC, Company age, Deal Type, and Exit Type

Checking the relationship between Return MOIC, Company age, which was calculated from the year found of the companies, based on Deal type (early stage, lager stage) with Exit Type (Merge/acquisition or IPO). We can see the companies over 10 years old with exit type IPO, following is Merger/Acquisition got the highest return MOIC.



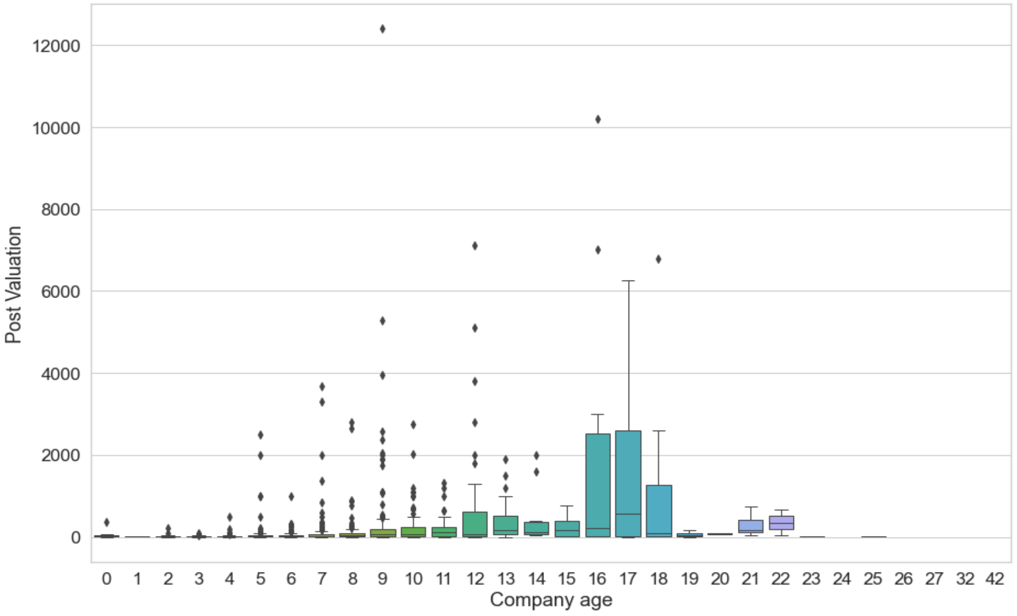
Top locations of startups that have high return MOIC.

California, New York, and Washington State include many startups with high return MOIC.



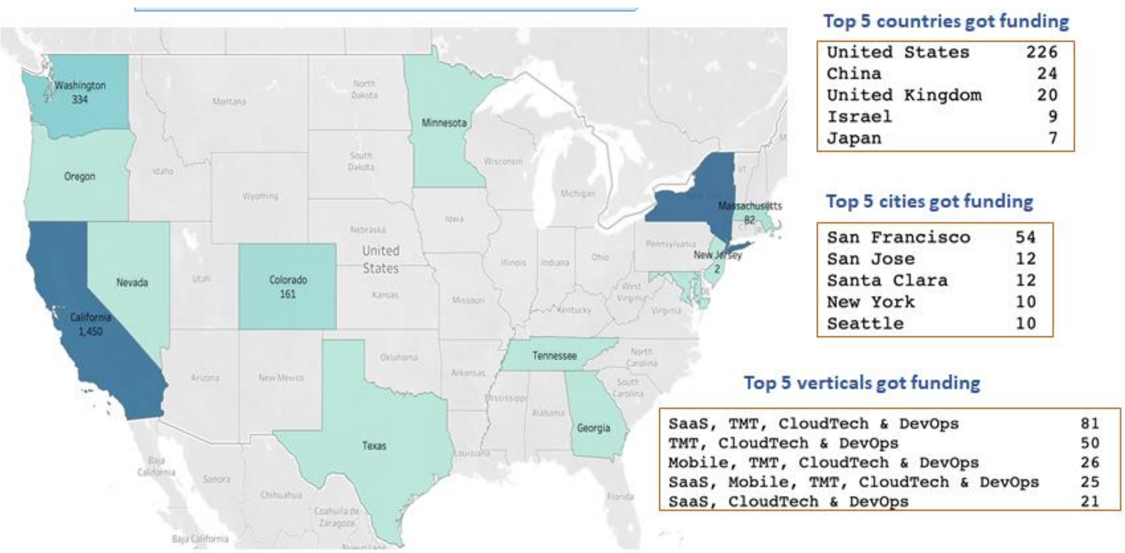
Relationship of Company Age & Return MOIC

Focus more on company age and return MOIC, when we merged 3 datasets and visualize the relationship between those features, we can see that the company with 10 and 11 years old got the highest return MOIC with more than 100M to 200M.



Relationship of Company Age & Post Valuation

Post valuation has high values with companies have 16 to 18 years old, mostly belong to later-stage VC.



Top 5 countries got funding

United States	226
China	24
United Kingdom	20
Israel	9
Japan	7

Top 5 cities got funding

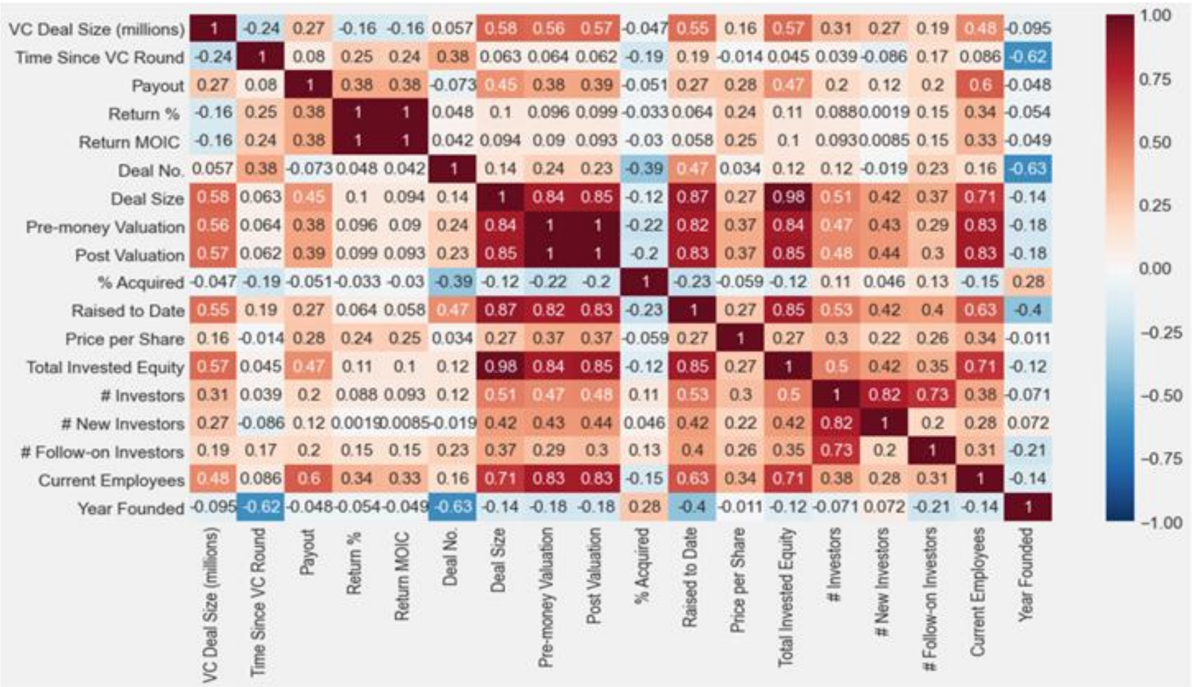
San Francisco	54
San Jose	12
Santa Clara	12
New York	10
Seattle	10

Top 5 verticals got funding

SaaS, TMT, CloudTech & DevOps	81
TMT, CloudTech & DevOps	50
Mobile, TMT, CloudTech & DevOps	26
SaaS, Mobile, TMT, CloudTech & DevOps	25
SaaS, CloudTech & DevOps	21

Return MOIC distribution in the USA.

High return MOIC **startups are mainly located on the east coast and west coa**st of the USA, especially in New York and California.



	VC Deal Size (millions)	Time Since VC Round	Payout	Return %	Return MOIC	Deal No.	Deal Size	Pre-money Valuation	Post Valuation	% Acquired	Raised to Date	Price per Share	Total Invested Equity	# Investors	# New Investors	# Follow-on Investors	Current Employees	Year Founded
VC Deal Size (millions)	1	-0.24	0.27	-0.16	-0.16	0.057	0.58	0.56	0.57	-0.047	0.55	0.16	0.57	0.31	0.27	0.19	0.48	-0.095
Time Since VC Round	-0.24	1	0.08	0.25	0.24	0.38	0.063	0.064	0.062	-0.19	0.19	-0.014	0.045	0.039	-0.086	0.17	0.086	-0.62
Payout	0.27	0.08	1	0.38	0.38	-0.073	0.45	0.38	0.39	-0.051	0.27	0.28	0.47	0.2	0.12	0.2	0.6	-0.048
Return %	-0.16	0.25	0.38	1	1	0.048	0.1	0.096	0.099	-0.033	0.064	0.24	0.11	0.088	0.0019	0.15	0.34	-0.054
Return MOIC	-0.16	0.24	0.38	1	1	0.042	0.094	0.09	0.093	-0.03	0.058	0.25	0.1	0.093	0.0085	0.15	0.33	-0.049
Deal No.	0.057	0.38	-0.073	0.048	0.042	1	0.14	0.24	0.23	-0.39	0.47	0.034	0.12	0.12	-0.019	0.23	0.16	-0.63
Deal Size	0.58	0.063	0.45	0.1	0.094	0.14	1	0.84	0.85	-0.12	0.87	0.27	0.98	0.51	0.42	0.37	0.71	-0.14
Pre-money Valuation	0.56	0.064	0.38	0.096	0.09	0.24	0.84	1	1	-0.22	0.82	0.37	0.84	0.47	0.43	0.29	0.83	-0.18
Post Valuation	0.57	0.062	0.39	0.099	0.093	0.23	0.85	1	1	-0.2	0.83	0.37	0.85	0.48	0.44	0.3	0.83	-0.18
% Acquired	-0.047	-0.19	-0.051	-0.033	-0.03	-0.39	-0.12	-0.22	-0.2	1	-0.23	-0.059	-0.12	0.11	0.046	0.13	-0.15	0.28
Raised to Date	0.55	0.19	0.27	0.064	0.058	0.47	0.87	0.82	0.83	-0.23	1	0.27	0.85	0.53	0.42	0.4	0.63	-0.4
Price per Share	0.16	-0.014	0.28	0.24	0.25	0.034	0.27	0.37	0.37	-0.059	0.27	1	0.27	0.3	0.22	0.26	0.34	-0.011
Total Invested Equity	0.57	0.045	0.47	0.11	0.1	0.12	0.98	0.84	0.85	-0.12	0.85	0.27	1	0.5	0.42	0.35	0.71	-0.12
# Investors	0.31	0.039	0.2	0.088	0.093	0.12	0.51	0.47	0.48	0.11	0.53	0.3	0.5	1	0.82	0.73	0.38	-0.071
# New Investors	0.27	-0.086	0.12	0.0019	0.0085	-0.019	0.42	0.43	0.44	0.046	0.42	0.22	0.42	0.82	1	0.2	0.28	0.072
# Follow-on Investors	0.19	0.17	0.2	0.15	0.15	0.23	0.37	0.29	0.3	0.13	0.4	0.26	0.35	0.73	0.2	1	0.31	-0.21
Current Employees	0.48	0.086	0.6	0.34	0.33	0.16	0.71	0.83	0.83	-0.15	0.63	0.34	0.71	0.38	0.28	0.31	1	-0.14
Year Founded	-0.095	-0.62	-0.048	-0.054	-0.049	-0.63	-0.14	-0.18	-0.18	0.28	-0.4	-0.011	-0.12	-0.071	0.072	-0.21	-0.14	1

Correlation between features

Finally, to check the collinearity between different features and facilitate feature selection, we create a heat map of the correlation between Returns by series and VC exists datasets. We can see that some features have collinearity. The score shows that when near 1.0, the correlation is strong and implies collinearity. However, many features have a high correlation with each other such as Pre-valuation and Post-valuation, Return % and Return MOIC, we need to notice those features when we create predictive modeling. Also, we need to identify important features that we will spend more time on for further analysis.

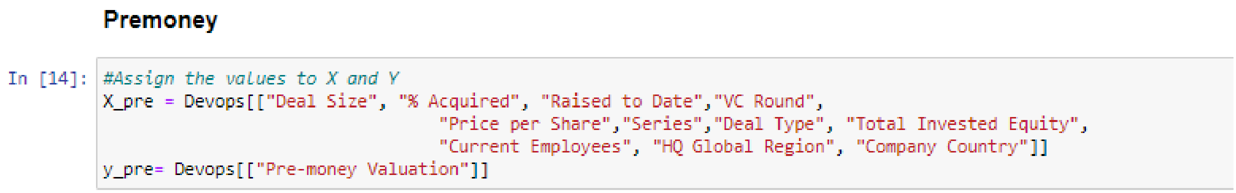
Method of Analysis/ Prediction Analysis

We applied different models to choose the final models that have the best performance to help the sponsor predict the best startups for their next investment.

1. DevOps

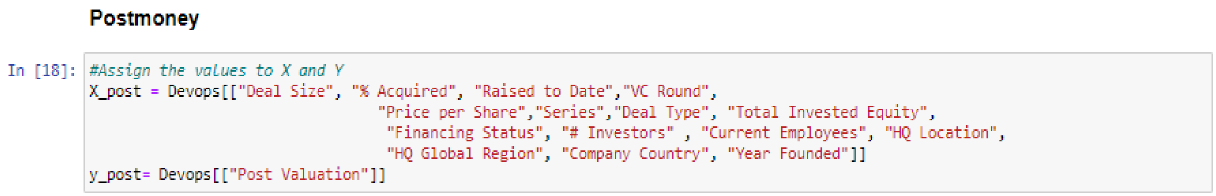
- ***Step 1: Building the Regression Analysis models***

Our two dependent variables are perfect for Regression Analysis. The first step is to put assign them correctly to independent X and dependent y. Keeping in mind that "Post Valuation" has a slightly more independent variable as stated above (14 and 11). Geographical elements ("HQ Global Region", "HQ Locations", "Company Country") determine the target variables' values for both but only contribute less than 5% to the final prediction models. Therefore, we believe that if the sponsor decides to invest somewhere else in the future, it would not affect the "PreMoney Valuation" or the "PostValuation"



Premoney

```
In [14]: #Assign the values to X and Y
         X_pre = Devops[["Deal Size", "% Acquired", "Raised to Date","VC Round",
                                           "Price per Share","Series","Deal Type", "Total Invested Equity",
                                           "Current Employees", "HQ Global Region", "Company Country"]]
         y_pre= Devops[["Pre-money Valuation"]]
```

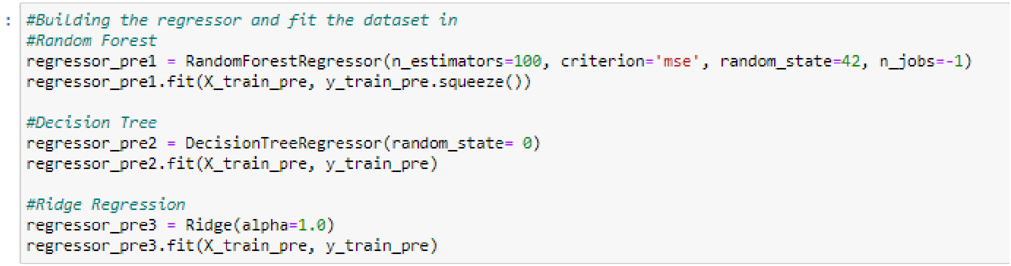


Postmoney

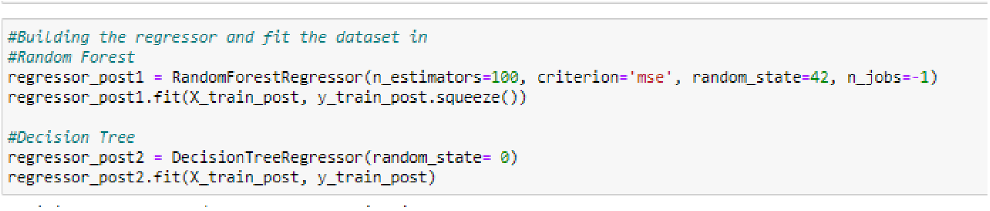
```
In [18]: #Assign the values to X and Y
         X_post = Devops[["Deal Size", "% Acquired", "Raised to Date","VC Round",
                                        "Price per Share","Series","Deal Type", "Total Invested Equity",
                                         "Financing Status", "# Investors" , "Current Employees", "HQ Location",
                                         "HQ Global Region", "Company Country", "Year Founded"]]
         y_post= Devops[["Post Valuation"]]
```

Then we move on to the models training step, where we decide to go with Random Forest (100 trees, criterion= MSE), Decision Tree, and Ridge Regression. However, we will skip the Ridge

Regression for the “Post valuation” set due to poor metric performance in the “Pre Money-Valuation.”



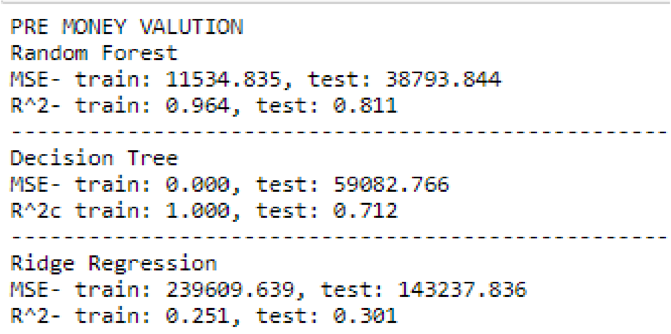
```
#Building the regressor and fit the dataset in
#Random Forest
regressor_pre1 = RandomForestRegressor(n_estimators=100, criterion='mse', random_state=42, n_jobs=-1)
regressor_pre1.fit(X_train_pre, y_train_pre.squeeze())

#Decision Tree
regressor_pre2 = DecisionTreeRegressor(random_state= 0)
regressor_pre2.fit(X_train_pre, y_train_pre)

#Ridge Regression
regressor_pre3 = Ridge(alpha=1.0)
regressor_pre3.fit(X_train_pre, y_train_pre)
```



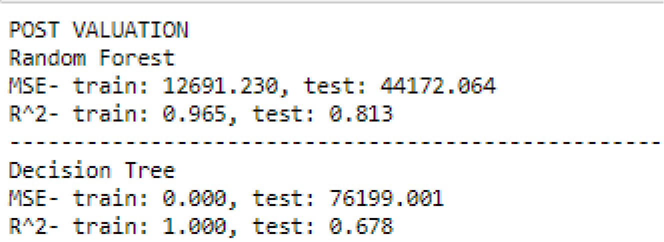
```
#Building the regressor and fit the dataset in
#Random Forest
regressor_post1 = RandomForestRegressor(n_estimators=100, criterion='mse', random_state=42, n_jobs=-1)
regressor_post1.fit(X_train_post, y_train_post.squeeze())

#Decision Tree
regressor_post2 = DecisionTreeRegressor(random_state= 0)
regressor_post2.fit(X_train_post, y_train_post)
```

Next, we will go with Mean-Square error (MSE) and R-Squared (R^2) for the performance metrics valuation step. The similarity between these two is that they are good for Regression tasks, especially statistical models like Linear Regression. MSE gets pronounced based on whether the data is scaled or not. That is where R^2 comes into place because it is the standardized version of MSE. R^2 illustrates the fraction of variance of response variable captured by the Regression model rather than the MSE which only gets the residual error. Hence, R^2 (or adjusted R^2) is more beloved since it gives a clearer picture of the regression models' quality. If you want to use MSE, it is advised to use the Rooted version (RMSE), which ignored the fact that the values of the response variable are called. (Kumar, 2020)

From a respective point of view, it seems that Random Forest performed best in both “Post Valuation” and “Pre-money valuation” with R^2 of the dataset around 0.811. The difference between Random Forest’s R^2 and Decision Tree's is what you would normally see. It is noteworthy that Ridge Regression has a significantly low R^2 in "Pre-money Valuation", so we decided to skip such method as stated above.



```
PRE MONEY VALUTION
Random Forest
MSE- train: 11534.835, test: 38793.844
R^2- train: 0.964, test: 0.811
--------------------------------------------------
Decision Tree
MSE- train: 0.000, test: 59082.766
R^2c train: 1.000, test: 0.712
--------------------------------------------------
Ridge Regression
MSE- train: 239609.639, test: 143237.836
R^2- train: 0.251, test: 0.301
```

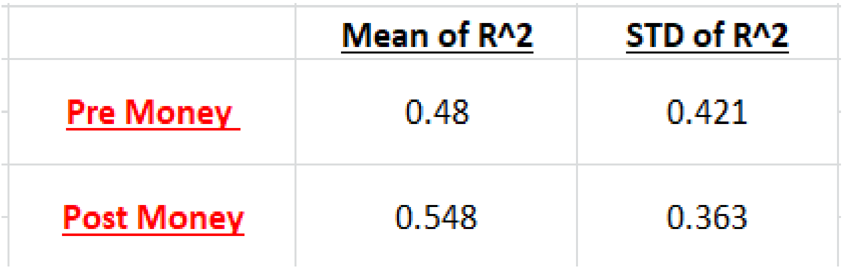


```
POST VALUATION
Random Forest
MSE- train: 12691.230, test: 44172.064
R^2- train: 0.965, test: 0.813
--------------------------------------------------
Decision Tree
MSE- train: 0.000, test: 76199.001
R^2- train: 1.000, test: 0.678
```

As seen above, both Decision Tree and Random Forest overfit the dataset but the difference between Train/Test results from the former is worse compared to the latter despite having higher train results. Thus, we will choose Random Forest for model optimization.

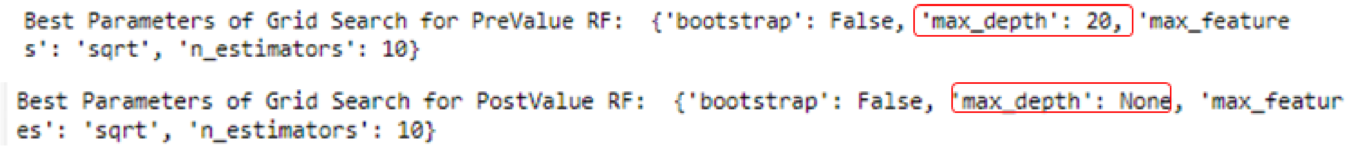
- ***Step 2: Model's optimization***

We perform the K folds Cross-Validation for 10 folds with R^2 as the main metric to see what the threshold is of training; we have for the two models. The table below indicates that both Random Forest Models achieve approximately 0.5 mean accuracies, with "Post Valuation" s performance is slightly better. However, "Post Valuation" s standard deviation is 0.1 points lower than that of "PreMoney Valuation". We can conclude that "Pre Money-Valuation" is more prone to overfitting compared to "Post Money" valuation". It could be the fact that we have fewer independent factors on "PreMoney Valuation" compared to the "Post Valuation", small dataset or that it is hard to predict something with no concrete supporting factors such as "PreMoney Valuation".



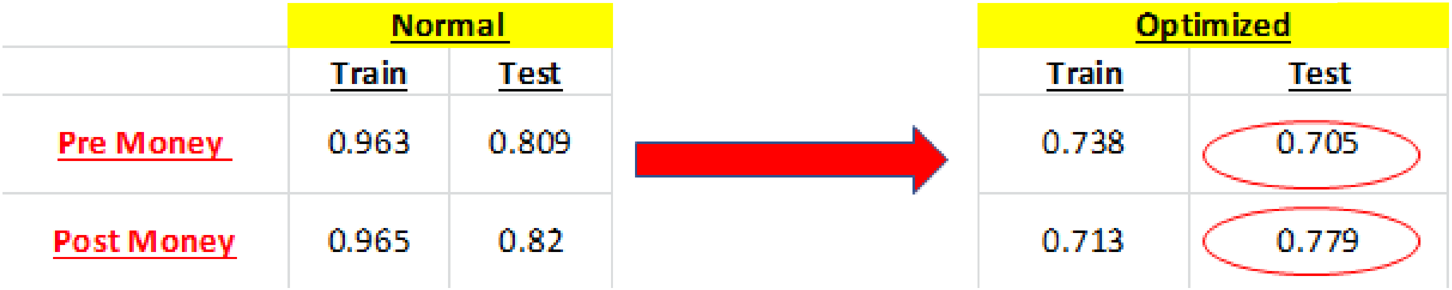
	Mean of R^2	STD of R^2
Pre Money	0.48	0.421
Post Money	0.548	0.363

After that, we perform the Grid Search function to find the best parameters for the most appropriate R^2 (reducing the mentioned R^2 of Train/Test). From a respective point of view, all the parameters are like each other except the "max_depth" as "20" for Pre and "None" for Post



```
Best Parameters of Grid Search for PreValue RF:  {'bootstrap': False, 'max_depth': 20, 'max_feature
s': 'sqrt', 'n_estimators': 10}
Best Parameters of Grid Search for PostValue RF:  {'bootstrap': False, 'max_depth': None, 'max_featur
es': 'sqrt', 'n_estimators': 10}
```

Lastly, we apply the models to the datasets and compare the outcomes. After optimization, the train and test set results are significantly smaller than before optimization, from 0.9 and 0.8 to around 0.7. Also, there is significant stability in the Optimized Results, and they do not show any great sign of overfitting, hence, we strongly recommend future research to replicate these models on better datasets.



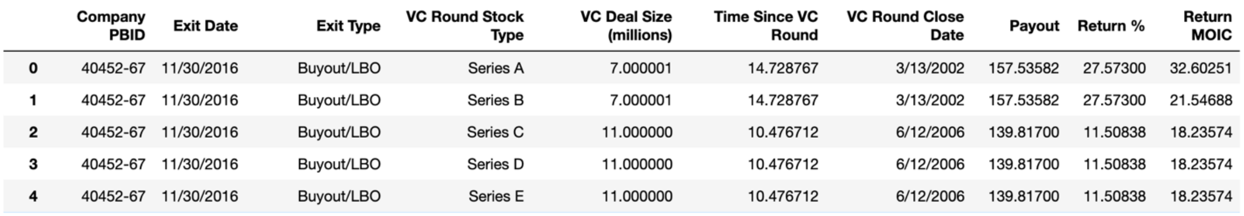
	Normal	
	Train	Test
Pre Money	0.963	0.809
Post Money	0.965	0.82

Optimized	
Train	Test
0.738	0.705
0.713	0.779

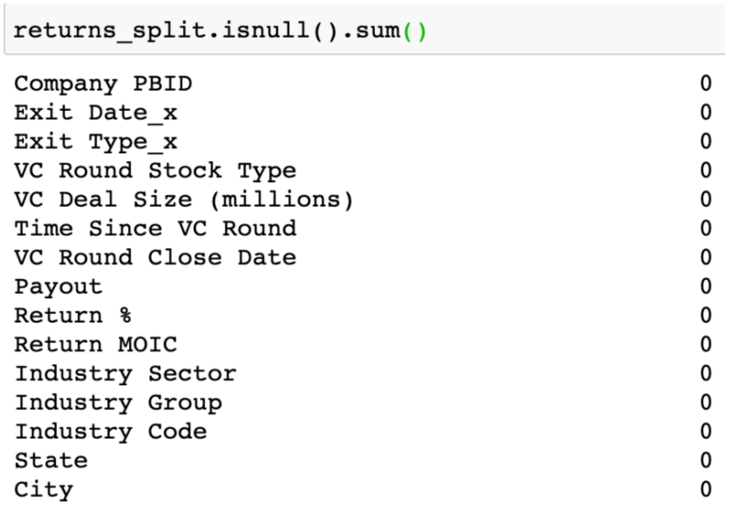
2. Returns by series

- ***Step 1: Loading and get overall information about the dataset.***

We used the Returns by series dataset which is cleaned by using the big ML platform which Including details about Investor returns by series of 245 companies along with financial and company details. We have the information as below:



	Company PBID	Exit Date	Exit Type	VC Round Stock Type	VC Deal Size (millions)	Time Since VC Round	VC Round Close Date	Payout	Return %	Return MOIC
0	40452-67	11/30/2016	Buyout/LBO	Series A	7.000001	14.728767	3/13/2002	157.53582	27.57300	32.60251
1	40452-67	11/30/2016	Buyout/LBO	Series B	7.000001	14.728767	3/13/2002	157.53582	27.57300	21.54688
2	40452-67	11/30/2016	Buyout/LBO	Series C	11.000000	10.476712	6/12/2006	139.81700	11.50838	18.23574
3	40452-67	11/30/2016	Buyout/LBO	Series D	11.000000	10.476712	6/12/2006	139.81700	11.50838	18.23574
4	40452-67	11/30/2016	Buyout/LBO	Series E	11.000000	10.476712	6/12/2006	139.81700	11.50838	18.23574

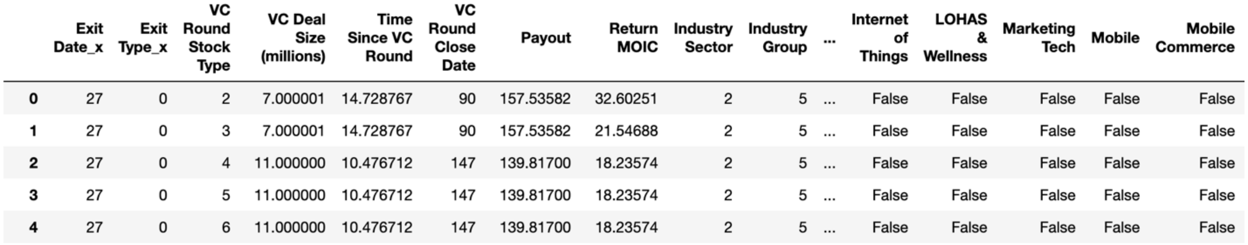


```
returns_split.isnull().sum()
```

```
Company PBID                                   0
Exit Date_x                                    0
Exit Type_x                                    0
VC Round Stock Type                            0
VC Deal Size (millions)                        0
Time Since VC Round                            0
VC Round Close Date                            0
Payout                                         0
Return %                                       0
Return MOIC                                    0
Industry Sector                                0
Industry Group                                 0
Industry Code                                  0
State                                          0
City                                           0
```

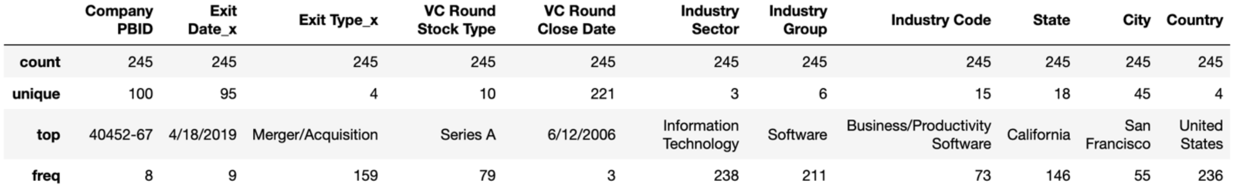
In this dataset, Company PBID contains many records that are related to one company in many different rows. Because Return by Series dataset is a small dataset without having enough features to support the performance of the predictive model, for improving the predictive result, we connect this dataset with some variables in VC exists dataset such as Industry Sector, Verticals, State, City. Those Categorical features carry a lot of information about the company. Some of them are encoded using labels and some are treated with one-hot encoding.

Because the Verticals column contains many items, I split them into Boolean columns with True and False for further analysis. From there, we have the updated dataset as below:



	Exit Date_x	Exit Type_x	VC Round Stock Type	VC Deal Size (millions)	Time Since VC Round	VC Round Close Date	Payout	Return MOIC	Industry Sector	Industry Group	...	Internet of Things	LOHAS & Wellness	Marketing Tech	Mobile	Mobile Commerce
0	27	0	2	7.000001	14.728767	90	157.53582	32.60251	2	5	...	False	False	False	False	False
1	27	0	3	7.000001	14.728767	90	157.53582	21.54688	2	5	...	False	False	False	False	False
2	27	0	4	11.000000	10.476712	147	139.81700	18.23574	2	5	...	False	False	False	False	False
3	27	0	5	11.000000	10.476712	147	139.81700	18.23574	2	5	...	False	False	False	False	False
4	27	0	6	11.000000	10.476712	147	139.81700	18.23574	2	5	...	False	False	False	False	False

We get some overall information about category variables as below:



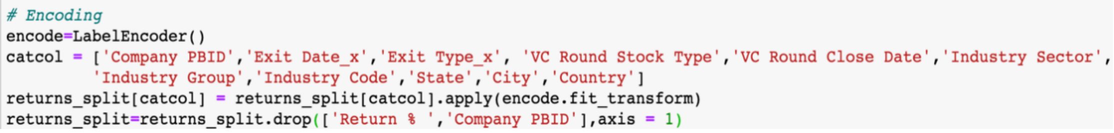
	Company PBID	Exit Date_x	Exit Type_x	VC Round Stock Type	VC Round Close Date	Industry Sector	Industry Group	Industry Code	State	City	Country
count	245	245	245	245	245	245	245	245	245	245	245
unique	100	95	4	10	221	3	6	15	18	45	4
top	40452-67	4/18/2019	Merger/Acquisition	Series A	6/12/2006	Information Technology	Software	Business/Productivity Software	California	San Francisco	United States
freq	8	9	159	79	3	238	211	73	146	55	236

- The top common Exit type for Returns by series is Merger/Acquisition.
- The most popular VC Round Stock Type is series A
- The top industry group is Software in San Francisco, California, USA

<u>Step 2: Testing the XGB Regressor model with Return MOIC as the target variable.</u>

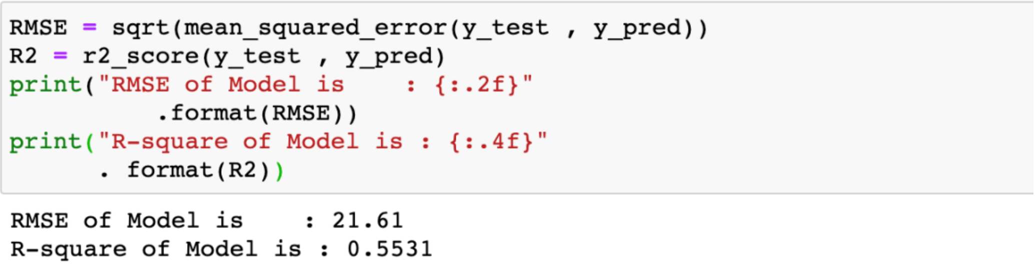
A gradient boosting model works by adding predictors to an ensemble in a sequential fashion, with the new predictor being fit to the residual errors made by the previous predictor (Michael Grogan, 2020). We decide to use a Tree-based algorithm because we have many numerical features, we don’t want to scale and the distribution of data to be a problem in modeling. Then applying XG boosting regressor model because we have implemented (L1) Lasso and (L2) Ridge regularization to minimize the effect of overfitting.

For feature engineering, we used One hot encoding to use this model with our categorical variables. For avoiding overfitting, the “Return %” is dropped because it has a strong correlation with the target variable “Return MOIC”, we also drop “Company PBID” when it will not help the model.



```
# Encoding
encode=LabelEncoder()
catcol = ['Company PBID','Exit Date_x','Exit Type_x', 'VC Round Stock Type','VC Round Close Date','Industry Sector',
         'Industry Group','Industry Code','State','City','Country']
returns_split[catcol] = returns_split[catcol].apply(encode.fit_transform)
returns_split=returns_split.drop(['Return % ','Company PBID'],axis = 1)
```

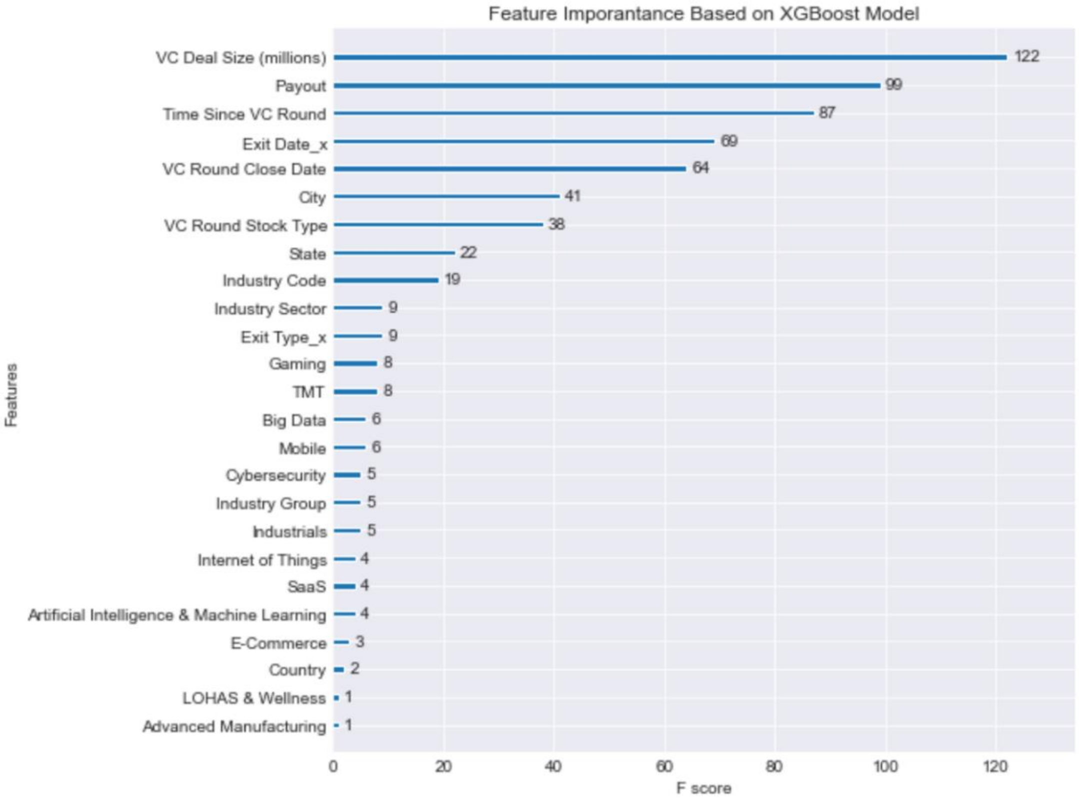
The next step is about splitting the dataset into the training set and testing set with a ratio of 7:3. We trained the model and predict with the test set. For XG Boosting Regressor, we have the RMSE and R- square of the model as below:



```
RMSE = sqrt(mean_squared_error(y_test , y_pred))
R2 = r2_score(y_test , y_pred)
print("RMSE of Model is     : {:.2f}"
          .format(RMSE))
print("R-square of Model is : {:.4f}"
      . format(R2))
```

```
RMSE of Model is     : 21.61
R-square of Model is : 0.5531
```

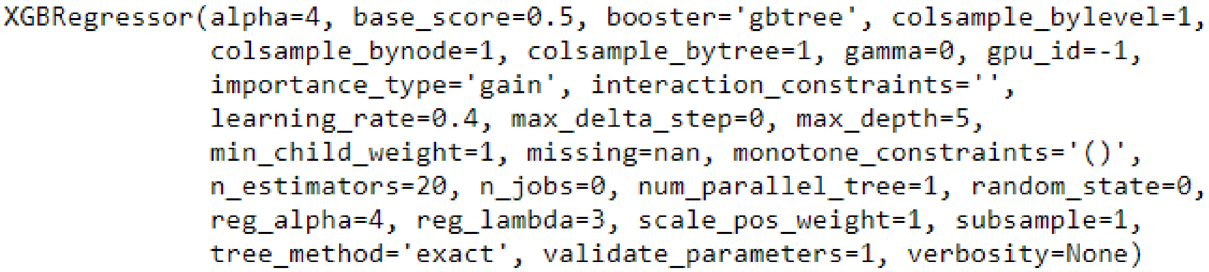
From the model, we check the most important feature on XG Boosting Regressor and we have the plot as below. We can see that VC Deal Size, Payout, Time Since VC Round, exit date are variables that have a strong relationship with the target variable Return MOIC.



3. VC exits:

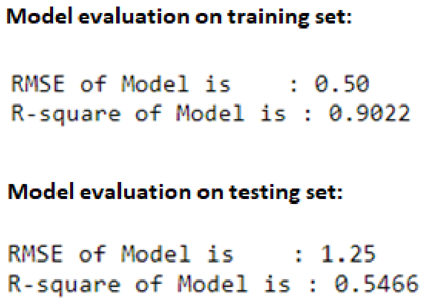
- ***Step 1: Post Value***

After splitting the dataset of 348 inputs into 7:3 ratio for training and testing, respectively, we tried different tree-based models. Since we are dealing with many numerical features which are quite correlated with each other, we have chosen to use boosted tree algorithm "XGBOOSTING', which allows L1 and L2 regularization parameters. Also, we change the target variable to its natural Log to mitigate the adverse effect of skewness of target distribution.



```
XGBRegressor(alpha=4, base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.4, max_delta_step=0, max_depth=5,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=20, n_jobs=0, num_parallel_tree=1, random_state=0,
             reg_alpha=4, reg_lambda=3, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

Before we train our model, we have removed few numerical features which are extremely correlated with others. During multiple trials of model evaluation and hyper-parameter tuning, we have noticed the correlated features were negatively influencing the model and in some cases, resultant models were overfitted with the training set.



Model evaluation on training set:

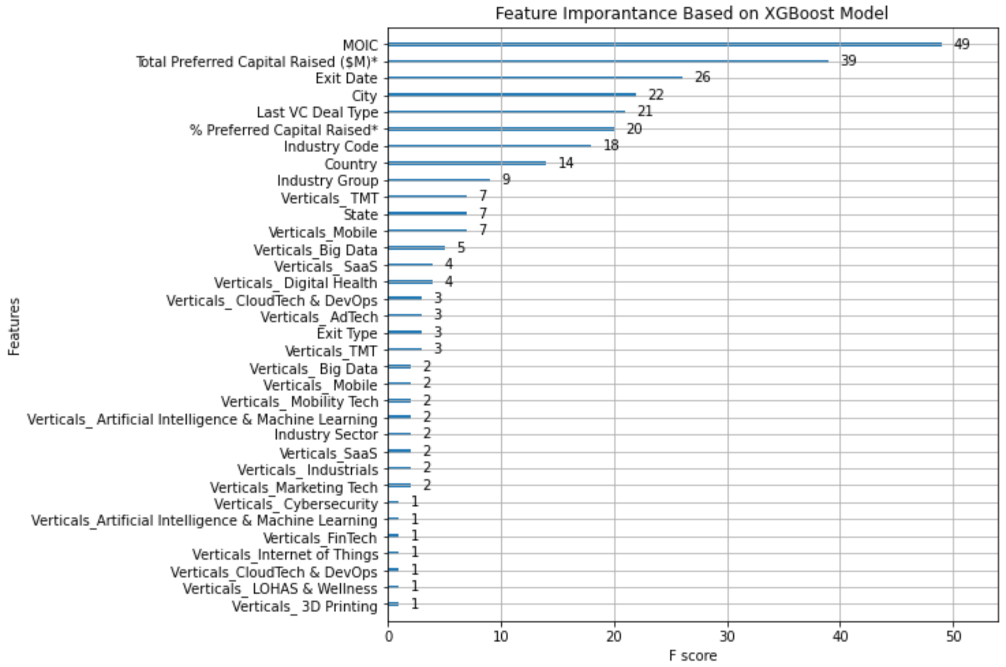
```
RMSE of Model is     : 0.50
R-square of Model is : 0.9022
```

Model evaluation on testing set:

```
RMSE of Model is     : 1.25
R-square of Model is : 0.5466
```

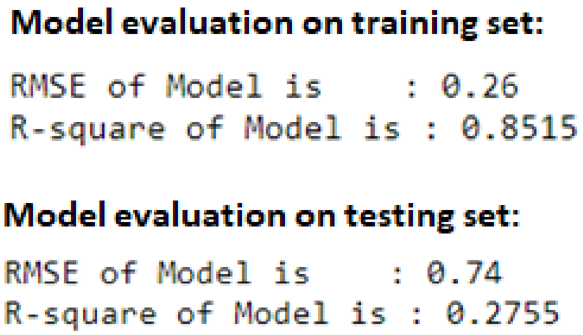
Fitted model when evaluated against training and testing set, we would see the difference in accuracy score (R-squared metric). This XGBoosting model trained to predict the “Post value” of a company yields the prediction score of 0.902 when evaluated with the training set and 0.54 against the testing set.

When we look at the feature importance based on XGBoost Model, as expected MOIC of a company is a major factor that influences the post valuation of the company followed by the Total preferred capital raised. It is interesting to see that the Exit date of the company heavily influences the valuation, and important vertical features give us an idea of which functional verticals are influencing the target both negatively and positively.



- ***Step 2: MOIC***

Similarly, we trained and tested the model to predict the MOIC of the company. Like in the previous model, the target variable was transformed to its natural Log and a few highly correlated features were excluded. As seen on the output below, the R-squared of the model is scored at 0.85 when evaluated with the training set and scored at 0.27 when evaluated with the testing set.



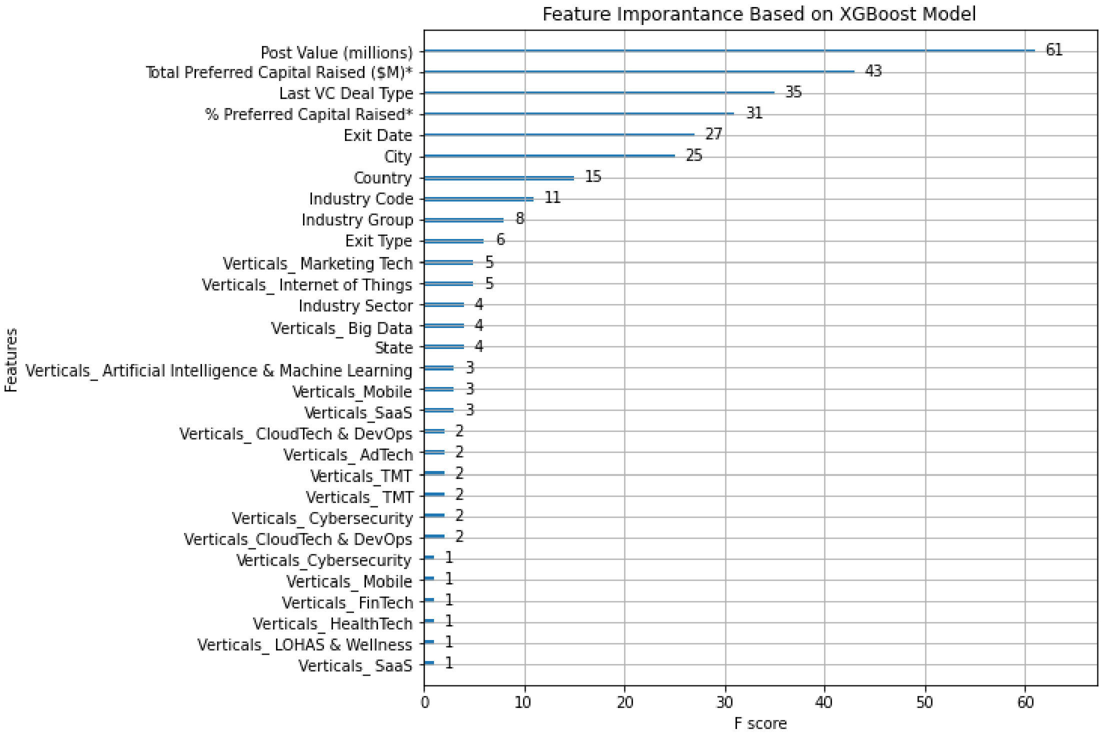
Model evaluation on training set:

```
RMSE of Model is     : 0.26
R-square of Model is : 0.8515
```

Model evaluation on testing set:

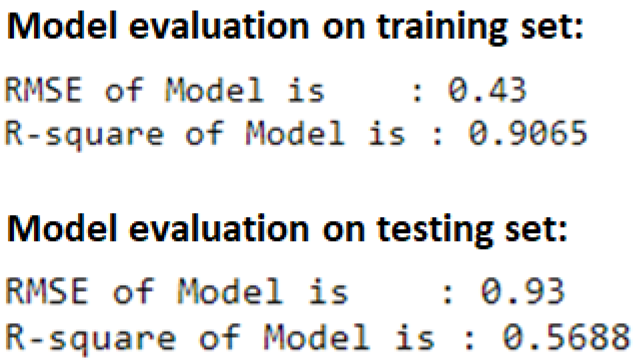
```
RMSE of Model is     : 0.74
R-square of Model is : 0.2755
```

As observed in the previous model, we see the post valuation and MOIC of a company directly complement each other. We also have VC deal type as one of the strong factors responsible to predict the MOIC of the company. Coming to vertical features, we see that functional verticals like marketing tech, Internet of Things, and Big Data are strong influencers of MOIC of the company.



- ***Step 3: Total Preferred Capital Raised***

A similar XGboosting model to predict The preferred capital Raised on a company was trained and evaluated. Concerning model evaluation output, we can see that the model saw the predictive score (R-squared) of 0.90 when tested with the training set and 0.56 when evaluated with the testing set which was not introduced to the model during training.

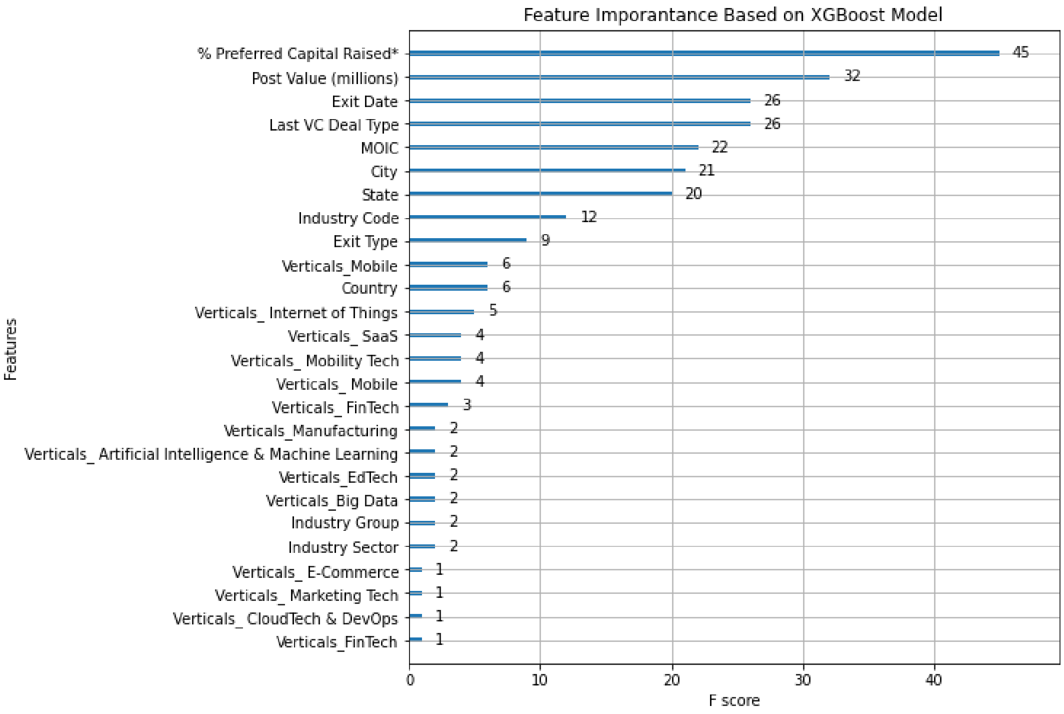


Model evaluation on training set:

```
RMSE of Model is     : 0.43
R-square of Model is : 0.9065
```

Model evaluation on testing set:

```
RMSE of Model is     : 0.93
R-square of Model is : 0.5688
```

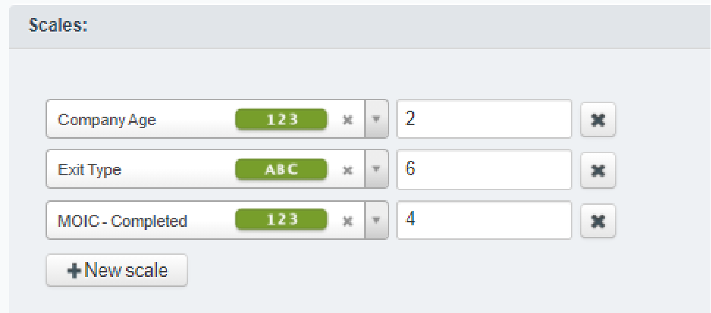


.

4. Unsupervised Model

Since we are specifically interested in pre-IPO companies which gives good returns, we will use the unsupervised model to divide data instances into the different group based on Exit type, MOIC (return of investment) and Age of company.

To ensure the output cluster are distinguished based on the above-discussed feature, we have scaled these features with an integer multiplier. This will increase their influences in distance computation, as a result, we will have a clear distinction of these features among each cluster. The rest of the numerical variables are scaled to have 1 standard deviation so that each feature has equal influence in the model.



Scales:

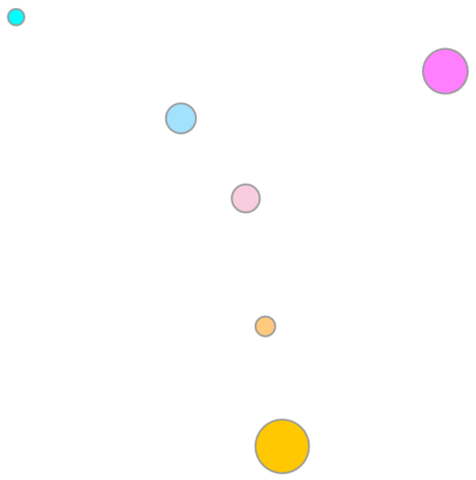
Company Age 123 – 2

Exit Type ABC – 6

MOIC - Completed 123 – 4

We have two major clustering algorithms in our hands, K-means, and G-means. In our case, we are not looking for the specific number of the cluster but are exploring how many clusters we can form based on returns, exits type, and company. Hence, we will be using G-means with the critical value of 4 which lets the algorithm generate a greater number of relevant clusters.

At this setting, the G-means algorithm-generated 6 clusters. Total 292 instances are distributed into 6 clusters in good proportion, the biggest clusters contain 28.42% (83 instances) of total data and the smallest cluster contains 3.08% (9 instances). Most importantly, the generated clusters are interpretable and distinguished based on selected three features.



Our next step is to name the clusters generated, however before naming, we will create a baseline or structure for the nomenclature of clusters which enhances the interpretability of each cluster without going into many details each time.

Cluster nomenclature baseline:

Exit Type		Company Age		MOIC/ROI	
Name Code	Attribute	Name Code	Attribute	Name Code	Attribute
IPO	IPO	**Young**	10 year old or younger	**Good**	Below 10
Acquisiton	Merger/Acquisition			**High**	Between 10 and 20
BuyOut	Buyout/LBO	**Old**	More than 10 year old	**Extreme**	Above 20

High-Level Summary of Clusters:

Cluster Name	Sample Size	Cluster Summary
IPO Old Good Returns	30	**Exit Type:** IPO **Avg. MOIC**: 45.03 **Avg. Company Age**: 13.89 Avg. No. of Employee: 711 Industry: Entertainment Verticals: VR, AR, Cloud Tech & DevOps, Gaming Avg. Exit/Pre/Post size ($): 8.5B, 8.8B, 9.5B Common Last VC Deal: Later Stage VC
IPO Old Extreme Returns	9	**Exit Type:** IPO **Avg. MOIC**: 8.54 **Avg. Company Age**: 12.37 Avg. No. of Employee: 2514 Industry: Business/Productivity Software Verticals: Big Data, AI, Cybersecurity, Mobility Tech Avg. Exit/Pre/Post size ($): 1.3B, 1.9B, 1.2B Common Last VC Deal: Later Stage VC
Buyout Young High Returns	35	**Exit Type:** Buyout/LBO **Avg. MOIC**: 11.01 **Avg. Company Age**: 7.04 Avg. No. of Employee: 103 Industry: Software Development Verticals: Avg. Exit/Pre/Post size ($): 0.72B, 2.3B, 0.54B Common Last VC Deal: Early-Stage VC

Acquisition Young High Returns	83	**Exit Type:** Merger/Acquisition **Avg. MOIC**: 11.93 **Avg. Company Age**: 7.04 Avg. No. of Employee: 38 Industry: Business/Productivity Software Verticals: AI, Big Data, SaaS, TMT Avg. Exit/Pre/Post size ($): 0.72B, 2.4B, 0.69B Common Last VC Deal: Series A
Acquisition Young Extreme Returns	14	**Exit Type:** Merger/Acquisition **Avg. MOIC**: 23.24 **Avg. Company Age**: 10.04 Avg. No. of Employee: 318 Industry: Automation/Workflow Software Verticals: Robotics and Drones, AI, 3D printing, Cloud Tech & DevOps, SaaS Avg. Exit/Pre/Post size: 1.1B, 1.1B, 1.1B Common Last VC Deal: Early-Stage VC
Acquisition Old Good Returns	121	**Exit Type:** Merger/Acquisition **Avg. MOIC**: 7.43 **Avg. Company Age**: 11.06 Avg. No. of Employee: 493 Industry: Software Development Verticals: Cloud Tech & DevOps, Mobile Avg. Exit/Pre/Post size: 0.56B, 2.1B, 0.45B Common Last VC Deal: Early-Stage VC

In the above table, we have a high-level summary of each cluster with Average Return of Investment (MOIC), Exit size, Pre value, Post value and Most recent VC deal type, Industry code, and functional Vertical, along with company age and the number of employees.

Link to Clustering Model: VC Exits Cluster - Alicorn VC Project

Based on the sample dataset, all the clusters have acceptable average MOIC, however since sponsors are looking to invest in a Pre-IPO unicorn startup, the focus would be specifically on "Acquisition Young Extreme returns" clusters as the companies in this cluster have an average post valuation of $1.1 billion. Similarly, other clusters with high returns can also be subject to the interest of sponsors, although companies might not achieve unicorn status.

Modeling and Predicting the Clusters

Now that we have classified our dataset into 6 clusters based on their attributes, we can use this cluster as a feature or a target to be predicted in a supervised predictive model. For that, we will first produce a batch centroid of the cluster in the dataset and add a new column to the original dataset with cluster labels.

Dataset with Cluster Labels: Dataset with Cluster Label

Before modeling, we will split our dataset into Training and Testing set in 4:1 ratio (Seed: 21). The model will only be trained with the training set and later evaluated/scored based on its ability to predict the cluster labels for exclusive data in testing which were not fed while training the model.

Training

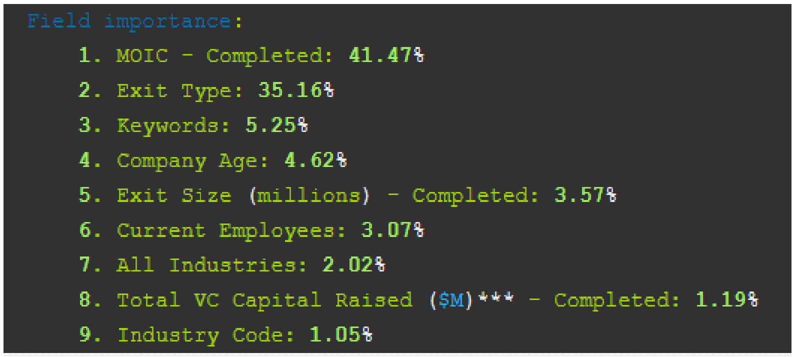
Before training the model, we will exclude few features which are extremely correlated with each other, including an independent feature with multicollinearity can negatively influence the predictive power of the model or sometimes might lead the model to overfit the training sample.

In our dataset, features "Exit Size", "Pre-Value" and "Post value" are extremely correlated, we will only feed "Exit Size" to the model. Similarly, we will choose to include "Total VC capital Raised" and exclude "Total Preferred Capital raised" as they are also extremely correlated with each other.

Considering the attributes and dimension of the dataset, we have chosen to use Boosted trees-based ensemble model to predict the cluster label of the company. Since, the tree-based model works great with a dataset with numerical features with improper distribution and add regularization component to tackle the possibility of overfitting due to small sample size, boosted trees ensemble is the best choice. Alternately, with the cost of extra time, we can choose to use the Automatic optimization feature in BigML.

Model: **Classification Model to Predict Cluster**

The following screenshot gives the feature importance of ensembles model trained to predict which cluster does each company falls under, this can be very useful in strategic investment planning. As expected, MOIC is the most important field as Clusters are segment based on Return rate, followed by exit type, keywords, and company age.

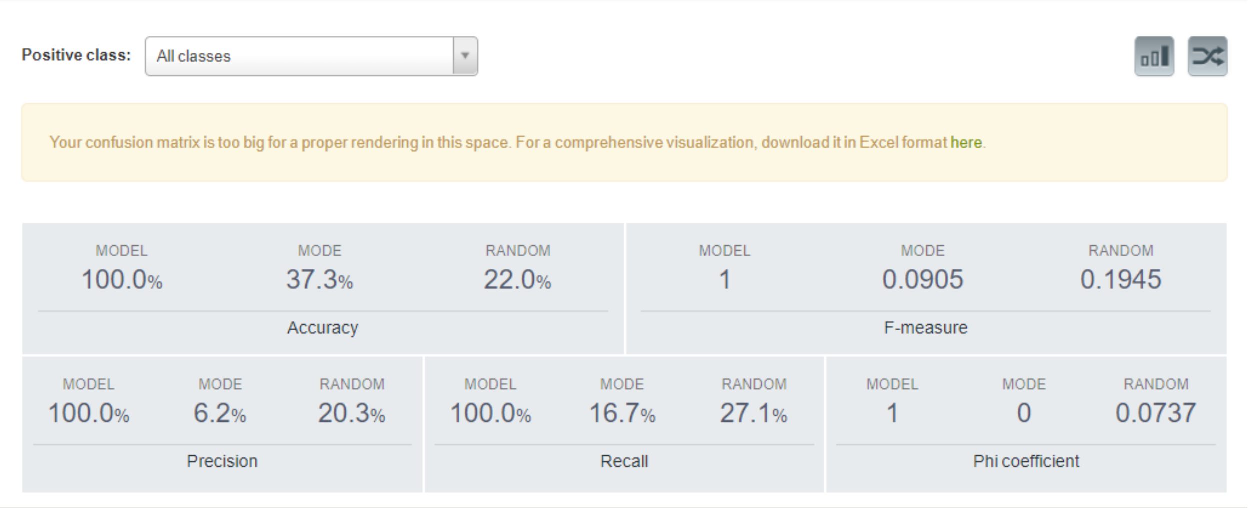


```
Field importance:
    1. MOIC - Completed: 41.47%
    2. Exit Type: 35.16%
    3. Keywords: 5.25%
    4. Company Age: 4.62%
    5. Exit Size (millions) - Completed: 3.57%
    6. Current Employees: 3.07%
    7. All Industries: 2.02%
    8. Total VC Capital Raised ($M)*** - Completed: 1.19%
    9. Industry Code: 1.05%
```

Evaluation

Evaluation Summary: **Classification Model Testing**

When evaluated with the testing set, instances that were not introduced to the model while training, the model was able to predict the cluster class with 100 % accuracy. This means each cluster has distinctive rules to define cluster class.



Positive class: All classes

Your confusion matrix is too big for a proper rendering in this space. For a comprehensive visualization, download it in Excel format here.

	MODEL	MODE	RANDOM
Accuracy	100.0%	37.3%	22.0%
F-measure	1	0.0905	0.1945
Precision	100.0%	6.2%	20.3%
Recall	100.0%	16.7%	27.1%
Phi coefficient	1	0	0.0737

Conclusions and Recommendations

In a nutshell, a tremendous amount of significant findings has been documented "Pre-IPO Unicorn Startups investment study with Machine Learning", ranging from EDA to Supervised and Unsupervised models building. First and foremost, most of the startups experience a high return of investment between the age of 9 to 12. We were able to divide the datasets into 6 different cluster groups based on Exit Type, MOIC, and Age of the company with interpretable results and definitive names that facilitate the investment strategy planning for Alicorn. The biggest cluster contains more than a quarter of the dataset (83 instances) while the smallest accounts for only 3% (9 instances). Secondly, it is observed that Return MOIC correlates positively with the number of employees, the higher the better and the best industry concerning the return of investments are Marketing Tech, Internet of things and Big Data (Software Technology). Average Return MOIC is high in IPO compare to other exit types such as buyout/ LBO, M&A. Third, geographical locations do play a somewhat important role in demining the models' successful prediction rate and other factors such as MOIC or Post Money valuation. East Coast and West Coast of the USA, especially New York and California, are the prime destinations. However, we must bear in mind that it is not the most essential factor, contributing by only 20-30% of the success rate and can easily be compensated by others. Israel, China, and the UK are some of the other hidden gems that most Venture capital giants often miss, evidently shown in the visualization above and given to Alicorn's investment philosophy of favoring Pre-IPO underdogs, we believe that it will not be the issue. To the best of our knowledge and ability, we have created the most applicable models that are suitable for our datasets, despite minor overfitting issues due to small datasets. We would advise future research to add other necessary features and variables to the dataset like the valuation changes throughout each round of raising capitals, what are the time intervals between each VC rounds, etc.

References

1. Arroyo, J., Corea, F., Jimenez-Diaz, G., & Recio-Garcia, J. A. (2019). Assessment of Machine Learning Performance for Decision Support in Venture Capital Investments. *IEEE Access*, *7*, 124233–124243. https://doi.org/10.1109/access.2019.2938659

2. Bhalla, D. (2017, August 5). *Feature Selection: Select Important Variables with Boruta Package*. ListenData. https://www.listendata.com/2017/05/feature-selection-boruta-package.html

3. *BigML is Machine Learning made easy*. (2019). BigML.Com.Au. https://bigml.com.au/accounts/login/?next=/dashboard/sources/new

4. Gompers, P. A., Gornall, W., Kaplan, S. N., & Strebulaev, I. A. (2016). How Do Venture Capitalists Make Decisions? *SSRN Electronic Journal*, 1–20. https://doi.org/10.2139/ssrn.2801385

5. Kasture, N. (2020, November 16). *6 stages to get success in Machine Learning project*. Analytics Vidhya. https://medium.com/analytics-vidhya/6-stages-to-get-success-in-machine-learning-project-2900555327bc

6. Kumar, A. (2020, September 30). *Mean Squared Error or R-Squared - Which one to use?* Data Analytics. https://vitalflux.com/mean-square-error-r-squared-which-one-to-use/

7. Luef, J., Ohrfandl, C., Sacharidis, D., & Werthner, H. (2020). A recommender system for investing in early-stage enterprises. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*.

8. Mazzanti, S. (2021, February 12). *Boruta Explained Exactly How You Wished Someone Explained to You*. Towards Data Science. https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a

9. Mwiti, D. (2021, February 25). *Random Forest Regression: When Does It Fail and Why?* Neptune.Ai. https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why

10. Problems of Small Data and How to Handle Them. (2020, January 30). EduPristine. https://www.edupristine.com/blog/managing-small-data

11. Villamor, F. (2019). *Predictive Analytics: How to build machine learning models in 4 steps*. Ducen. https://blog.ducenit.com/predictive-analytics-modeling-guide