MOHAMMED VI POLYTECHNIC UNIVERSITY

EMINES School of Industrial Management

UNIVERSITÉ MOHAMMED VI POLYTECHNIQUE

A Data Science approach to improving M&A decision making

By : Mohammed El Mehdi CHBIHI
Mohamed Amine CHAFIK
Ismail GHMIMAT

Table of contents

I.Acknowledgements :

I would like to express my special thanks of gratitude to our teacher Ms. GUESSOUUS as well as our principal M. CHEIMANOFF who gave us the golden opportunity to lead this wonderful project on the topic of data science applied to corporate finance which also helped me in doing a lot of research and we came to know about so many new things I am really thankful to them. Secondly We would also like to thank Ms ABOUYOUB, Ms CHAIRI, M.ALAOUI who by contributing their free time and gave us relevant insight on M&A logic, statistics and data science respectively.

II.Abstract:

In the last 20 years, mergers and acquisitions, commonly referred to as M&A, have become the most prominent practices of modern world's corporate finance. However, there are still many doubts over what parameters make an operation successful or not, and quantifying an operation's potential success rate has proven to be rather challenging. Integration has often been considered as key to an M&A operation's success, with a previous Singh and Zollo study referring to "tacit and codified" knowledge and previous experience with operations, while Timothy Galpin and Mark Herndon lean towards a more "culture-oriented" approach, with a special attention to how similar cultures of various parties are… All are parameters which could be interesting in small scale, intra-country deals, but not in an advanced, globalized context. We use data gathered from *mergr.com* ranging from 2000 to 2020 and used updating data such as inflation rates and yield curves data from SEC filings in addition to quantitative approaches cited in *The Complete Guide To Mergers And Acquisitions.* In contrary of some of the principles cited in that book, both macro and micro financial indicators of both entities, occurring country, occurring year, and more have a big impact in terms of both deal pricing and also the operation's success, which can be evaluated by key financial and economic indicators.

III.Introduction

Corporate finance is a field that gained traction throughout the second half of the 20th century. It has helped many companies get effectively bigger during the exponential market growth in the 'glorious thirty'. A lot of research has gone in optimizing its techniques and procedures, which has led to the establishment of a more concise domain of activity: mergers and acquisitions. Commonly referred to as M&A, the volume of transactions classified as 'M&A operation' has gone from 347 Billion dollars back in 1985 to 3,701 B$ in 2019[1]. The tendency is rather clear: M&A is the future of corporate finance, and every company aspiring to achieve considerable economic growth has to consider M&A at some point. This major shift has led to the appearance of various types of operations, which span all company sizes, interests & goals, and structures.

Mergers and Acquisitions refer to[i] the consolidation of companies or assets through various types of financial transactions, including mergers, acquisitions, consolidation etc. or as The Corporate Finance Institute[ii] defines it, M&A simply represents transactions between two companies combining in some form.

Although the two words are used interchangeably, they are slightly different and hold different legal meanings.

In a merger, two companies of similar size join forces to move forward as a single new entity. Both companies' stocks are surrendered and new company stock is issued in its place.

On the other hand, an acquisition is when a larger company takes over a smaller company and establishes itself as the new owner, thereby absorbing the business of the smaller company. M&A deals can be friendly or hostile, depending on the approval of the target company's board.

Our study was motivated by a very depressing statistic[iii]: According to most studies, up to 90% of M&A operations are considered failures, so we decided to take a data science approach to improve M&A decision making.

Our first idea was to predict M&A operations' success using anomaly detection via isolation forest, we also used propensity score matching and difference in differences

We chose to use quantitative quantifiers because most of the research we found has gone through qualitative estimators which can be misleading in the context of a globalized industry

IV.Dataset Constitution:

a.Data accessed through the Mergr Database:

Our first data was acquired from the Mergr[iv] website, a dead simple PE/M&A database that allows anyone to identify private equity firms, corporate acquirers, their M&A, and advisors.

After acquiring premium access to the website's databases, we downloaded approximately 10 000 lines of data, with the columns: Date, Target, Sector, Address, City, State/Province, Zip, Country, Phone, Url, Transaction Type, Deal Value, EV/Revenue, EV/EBITDA, Investor(s) ,Seller(s).

As it represents real data, we found a lot of missing values, especially in the EV/Revenue and EV/EBITDA, which lead us to search for a new way to collect more relevant financial data.

b.Data acquired through the API:

The website[v] proposes a large panoply of dictionaries containing different sets of financial statements results, metrics, ratios, and key scorings.

We used it to get the average stock price, earning per share, debt-equity ratio, net profit margin, inventory turnover, current ratio, total debt added and enterprise value for each acquiring or Merging entity, from the years 2000 to 2019. We also did the same for non M&A-engaged firms for comparison purposes.

c.Adopting the final dataset based on previous acquired data:

- The study focalized on 2400 deals from 2000 to 2015. (the number of deals was way bigger but we had to accomodate with the tickers we had and the deals value that seemed relevant)

- After this, we want to compare the performance of

the acquiring companies with other companies that didn't do M&A in that period. For this matter, we define a neighborhood list where we omit the companies that did M&A in the (+/- 5years radius). And then we randomly leave 8000 untreated observations for matching purposes.

- After this, we want to collect the 4 characteristics (MC,EV,Debt,Cash) + 5

ratios that investors judge to be relevant in cross-companies performance. The collected data will range from 5 years pre-merger to 5-year post-merger.

- The results after this get us 24 columns of financial data that contains its share of

Nan values. In order to clean it, we defined a threshold to drop deals that had more than 10 Nan values and we run KNN imputer on the rest of the data.

This decision was not easy as it made us loose 600 precious observations but was important to get relevant data and get rational tendencies throughout our study.

V.Exploratory Data Analysis

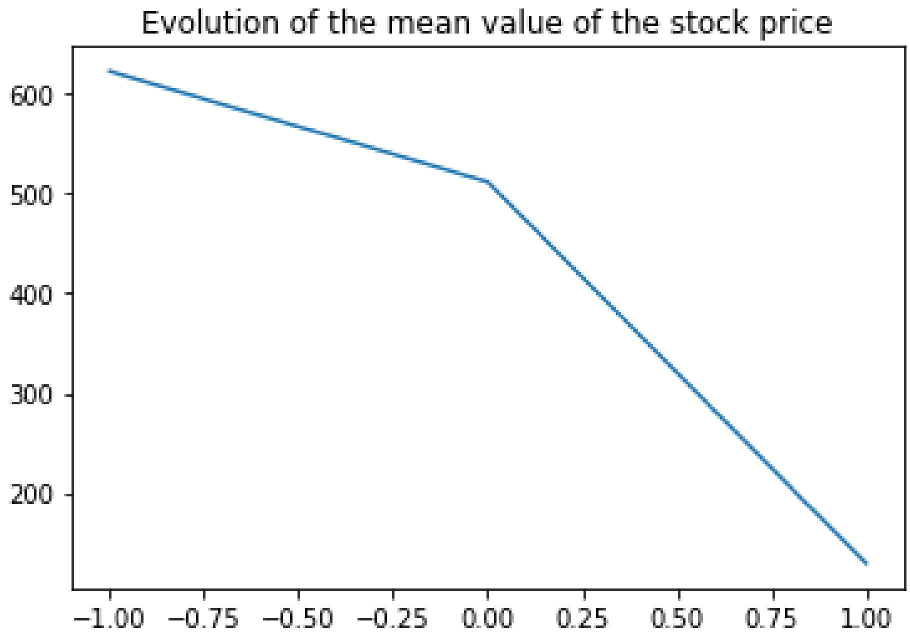
Our study is based on a dataset containing 3800 deals, with as main columns: date, target, sector, country, transaction type, deal value, and the evolution of the earnings per share, debt/equity ratio, net profit margin, inventory turnover, current ratio, total debt added in the year and enterprise value for the year before M&A, the year M&A took place and the year after that.

Our dataset average deal value is around 1 Billion dollars, with a standard deviation of 4 Billion dollars.

We can see by comparing averages of each year, that in general, companies that chose to do an M&A transaction do not perform well, and that shows in their financial statements:

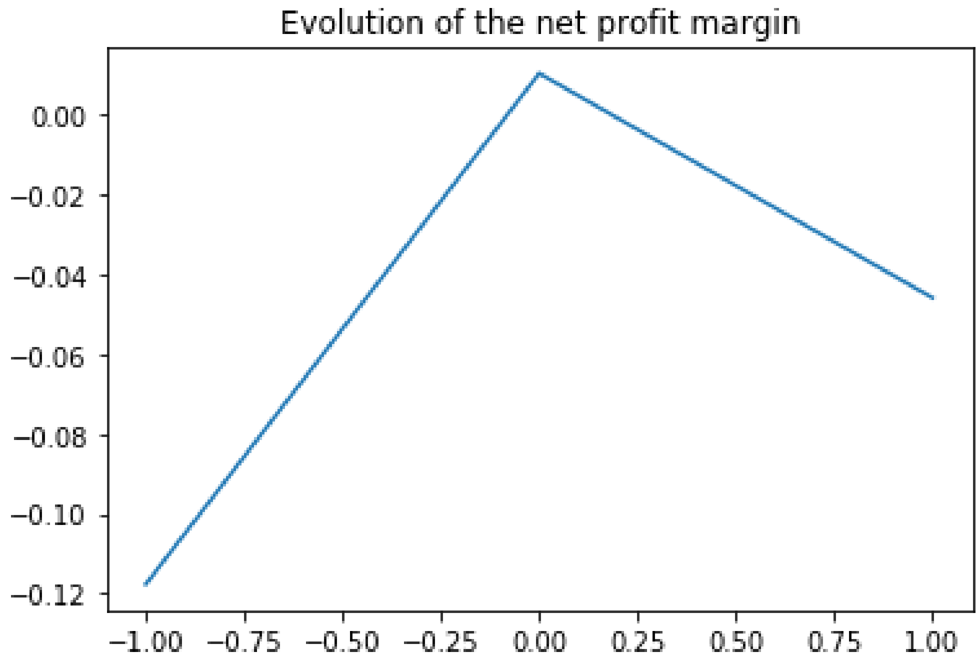
As we will see on the plots that will follow, the stock price and the earnings per share drop in general after an M&A operation:

The stock price drop can be explained by the fact that the company often pays a premium for the target company, or has to get into debt to finance the operation. Also, sometimes, investors think the deal value is too high.

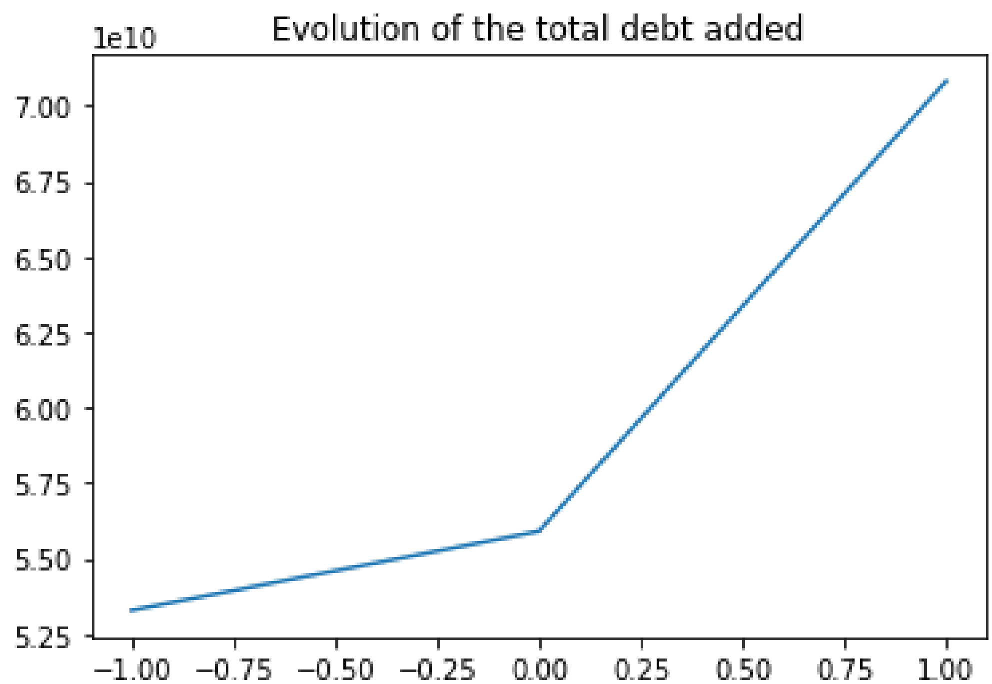


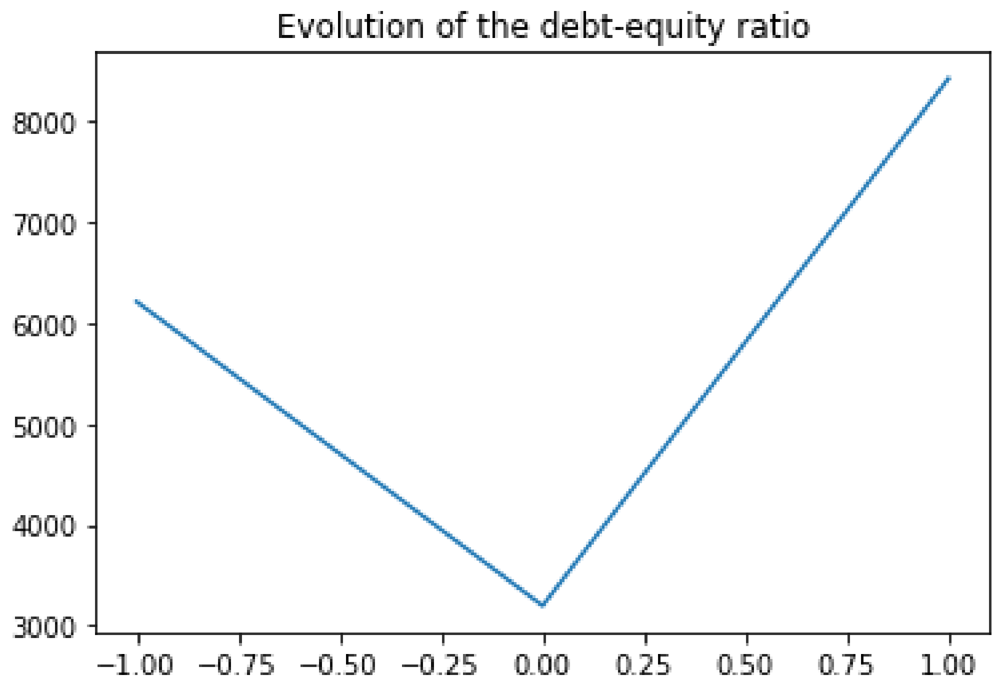
The drop of the earnings per share, can be explained by the fact that, as we said before, up to 90% of M&A deals are failures.

We can see from the plot that follows, that in general, companies that go through an M&A operation tend to lose money on the year after the operation.

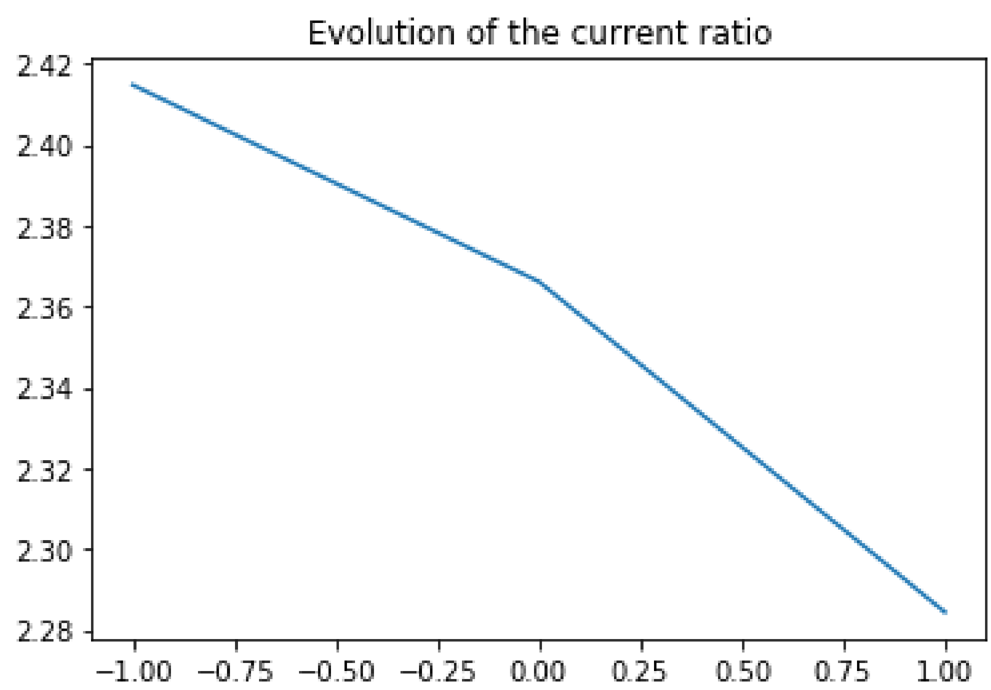


We can clearly see that the total debt added and the debt-equity ratio both have a big rise the year after an M&A operation, and that is mostly due to companies getting into debt to finance the M&A operation and unforeseen expenses.

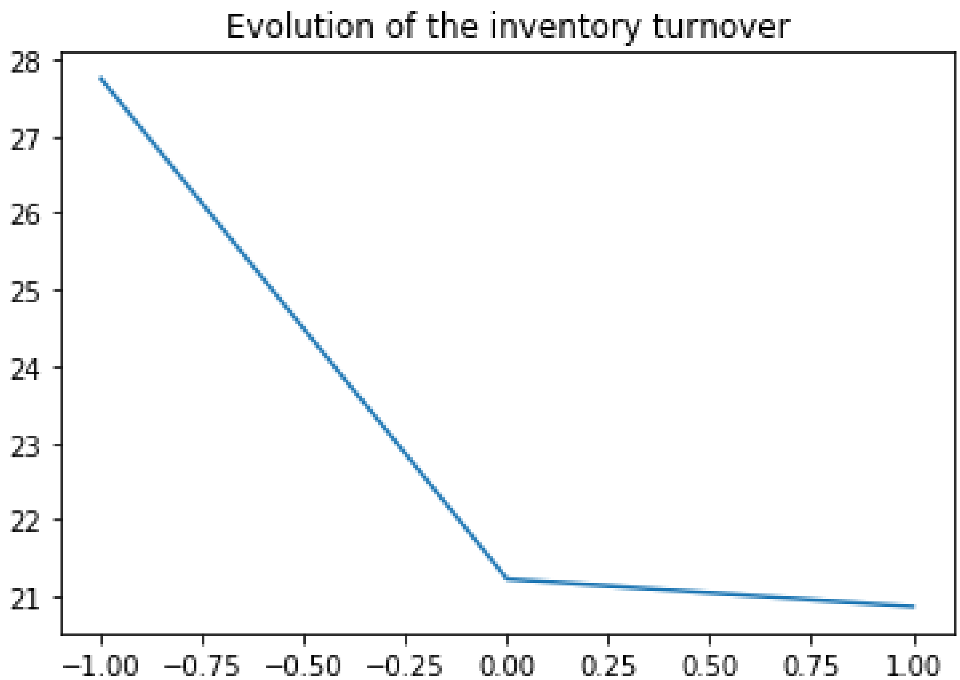




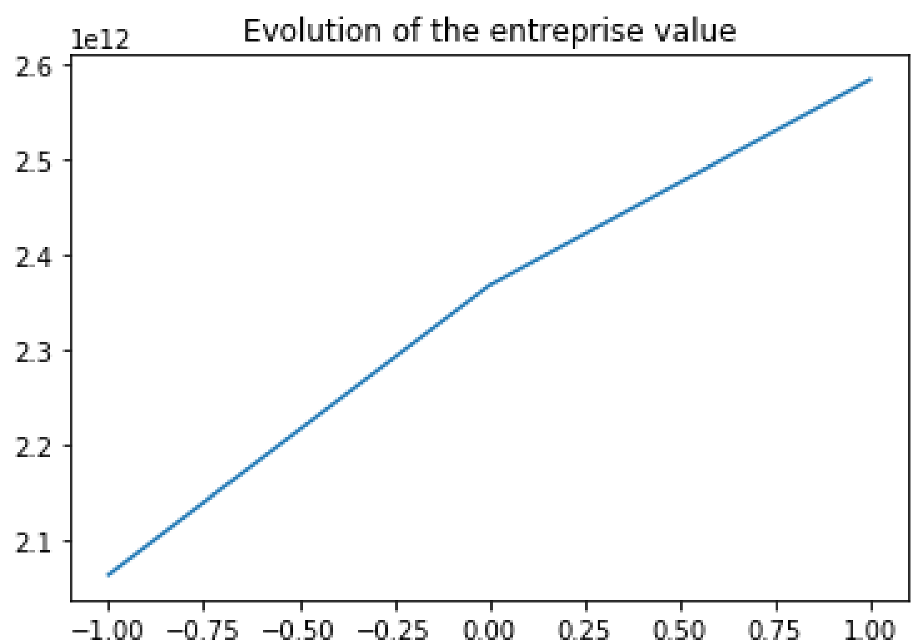
The current ratio also falls after M&A, as the companies spend a lot of liquidity on the deal value



The inventory turnover falls a little bit after M&A but stays quite stable.



The Enterprise value, which is Market Capitalization + Market Value of Debt – Cash and Equivalents, rises, logically, after an M&A operation.



After defining new columns, representing the evolution of each ratio a year after the M&A operation (the ratio after one year - the ratio the year of the operation): ‘SP’ being the stock price, ‘EPS’ the earning per share, ‘DER’ the debt-equity ratio, ‘NPM’ the net profit margin, 'IT' the inventory turnover, 'CR' the current ratio, 'Debt' the added debt and 'EV' the enterprise value. We calculate the correlation matrix:

	SP	EPS	DER	NPM	IT	CR	Debt	EV
SP	1.000000	0.991210	0.000142	0.007020	0.000926	0.005731	0.000419	0.000434
EPS	0.991210	1.000000	0.000136	0.016499	0.001048	0.007933	0.027034	0.002335
DER	0.000142	0.000136	1.000000	0.000770	0.002456	0.012214	0.000558	-0.000069
NPM	0.007020	0.016499	0.000770	1.000000	0.001404	-0.000741	0.000043	0.000840
IT	0.000926	0.001048	0.002456	0.001404	1.000000	0.000503	-0.002682	-0.002569
CR	0.005731	0.007933	0.012214	-0.000741	0.000503	1.000000	0.033929	0.004138
Debt	0.000419	0.027034	0.000558	0.000043	-0.002682	0.033929	1.000000	0.070139
EV	0.000434	0.002335	-0.000069	0.000840	-0.002569	0.004138	0.070139	1.000000

We can clearly see that the evolution of the stock price and earning per share after an M&A operation are highly correlated (99.12% correlation), and that all the other variables are not correlated one by one (less than 1% for every combination)

VI. M&A success rate

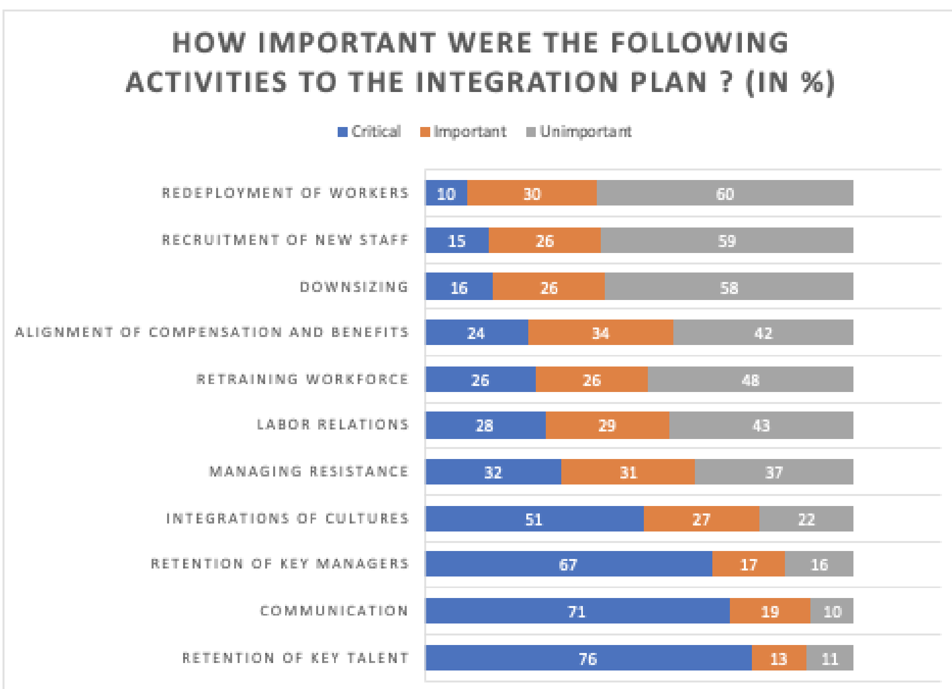
One of the most challenging aspects of M&A operations is how successful they are. A lot of research has gone through qualitative estimations of an operation's success, yet such approach can prove to be misleading in the context of a globalized industry, in addition to a transitioning landscape towards remote work.

According to a Singh and Zollo Study[1] on post M&A operations, the main aspect is knowledge and experience. They split the so-called knowledge into two distinct types: "Tacit" knowledge is more about the expertise of the executives and is mostly subjective as it is based on the estimation of a person's competence. "Codified" consists of written procedures that establish post-Merger routines, that guide integration and decision-making processes in the implementation phases of the deal. Key points cited by those key managers:

[1] *See Pandya, 1998*

- Faster integration
- Lower costs
- Surpassing projected synergies
- Protection of productivity and maintenance of customer focus
- Smoother Transition
- Faster and more effective response to workers' questions and concerns.

For Timothy Galpin and Mark Herndon[2] , the qualitative, people-oriented aspect is even more emphasized on. In their publications, they assure that the core aspect of a successful mergers and acquisitions operation is the cultural integration of employees of both firms. In addition to that, they rely on the expertise of leading managers in order to establish both tracking techniques that would provide an estimation of the cultural compatibility, such as surveys, task forces, and various initiatives that, in fine, are investigating within the company both pre-merge and post-merge.



[2] In their book : The Complete Guide To Mergers And Acquisitions

We can clearly see that for most surveyed CEOs and CFOs[3] surveyed (190 over the United States, Brazil, the Asia-Pacific region) from industries where the most M&A operations happen, which are:

- Manufacturing
- Consumer products
- Banking and financial services
- Chemical

That the top 7 most important activities when it comes to the integration plan are all related to employees and how they feel about the new entity. They consider the fact that keeping the adequate workforce, in addition to making sure such workforce is in symbiosis with the rest (hence, new) employees through communication and cultural integration processes, all of which can only be successfully conducted by experienced managers, who have the expertise necessary, i.e. the knowledge Singh and Zollo talked about.

When we take a glance at the percentages of issues cited as risk by surveyed companies, when can clearly see the same tendencies. The most curious part is the fact that conflicting expectations with target company is a considerable issue, because we believe that an M&A operation does not necessarily require a complete agreement. Indeed, in an acquisition scheme, or in any deal where one of the parties is dominant, healthy macro and micro economic metrics, in addition to experience, can easily overcome this sort of disagreement. As a matter of fact, lack of M&A experience is also cited as an issue, which is in a way directly linked to the approach of M&A acquisitions concerning two entities with different visions. Another manifesto of the limitations of the cultural, people-oriented approach towards mergers and acquisitions, is the fact that relative dominance of one partner, is also cited as an issue for the deal, whilst such dominance in most M&A deals is fundamental, as long as it is not a horizontal merger, which can also, in some cases, be needed.

Chart[4] for reference:

[3] Survey conducted by Watson Wyatt Worldwide : *Assessing and Managing Human Capital : A Key to Maximizing the M&A Deal Value*.
[4] Survey conducted by Watson Wyatt Worldwide : *Assessing and Managing Human Capital : A Key to Maximizing the M&A Deal Value*.

Certainly, the execution of an integration impacts a transaction’s success rate. However, it is more useful to consider whether the transaction was worth making in the first place, as a “suicidal” deal will always be destined to fail.

From this perspective, A.T. Kearney[5] has partnered with the UK Investor Relations Society (IR Society) to understand exactly what the most important metrics and analyses are in evaluating proposed merger and acquisition transactions. The solution to the risk issues lies in the metrics that would allow executives to determine the “healthiness” of a potential upcoming deal.

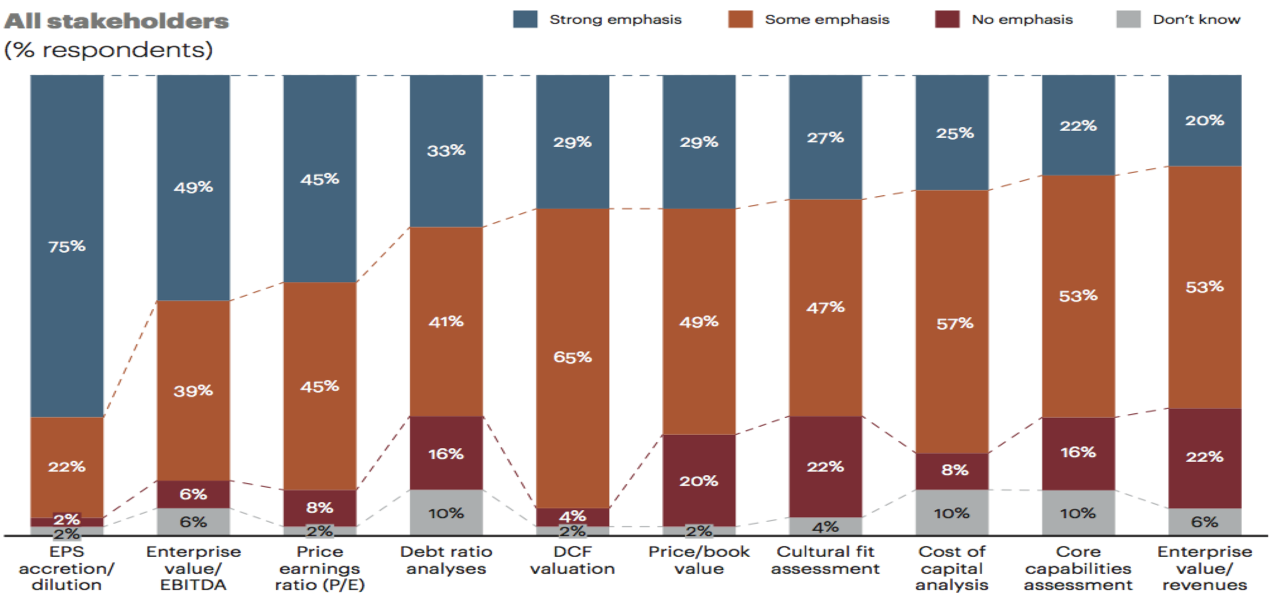
Investor relations professionals were asked about the views of key stakeholders - business executives, sales analysts and investors - on ten frequently used measures and analyses when elaborating a new M&A deal. We found that EPS analysis is by far the most widely used, with 75% of respondents ranking it in the "strong emphasis" category, 26 points above Business Value/EBITDA, the second most frequently assessed measure. Hence, our hypothesis, which states that macro and micro economic metrics of concerned parties of a deal are key to evaluating an upcoming deal’s success rate, can only be reinforced thanks to this study. In the following chart, we can clearly see that not only EPS is concerned, but even other economic metrics are

[5] See : https://imaa-institute.org/ma-deal-evaluation-challenging-metrics-myths/

prominent in the decision-making process of a deal, much more than the integration parameters mentioned before:

Figure 1

EPS accretion/dilution analysis is given the most emphasis in evaluating proposed M&A transactions



Note: Stakeholders include company executives, sell-side analysts, and investors; EPS is earnings-per-share; DCF is discounted cash flow.
Source: A.T. Kearney and Investor Relations Society, 2013

We can then clearly see that there are key economic parameters have a major interest in the decision making process of M&A operations. Unfortunately, some of these metrics are either confidential, or require major payments. To overcome this, we decided to work on metrics and ratios directly related to those cited before, and which have, in various occasions, have been proven to be strongly correlated to decision metrics. All in all, in order to predict M&A operations success rate, we will be working with :

- EPS (Earnings Per Share);
- EV (Enterprise Value);
- Debt Equity Ratio;
- Net Profit Margin;
- Inventory Turnover;
- AddTotalDebt;
- Current Ratio;

NB: All values will be inflation adjusted to the 2019 USD.

These ratios were not arbitrarily picked. Indeed, each one of them a certain performance that we want to monitor. As they respectively represent five main performance classes:

1. Liquidity Ratios
2. Solvency Ratios
3. Profitability Ratios
4. Efficiency Ratios
5. Coverage Ratios

However, when pushing our research further and trying to collect the mentioned ratios – Even through Mergr and our API– the conclusion we came with was simple: It is very rare to find, or even extract, all of them regarding one transaction. In addition to that, M&A analysts do not have the time to dig deep in the archives to look for a couple of ratios to make their decision, as other aspects come into play that are more of a qualitative and sustainability aspect.

a.*Determining most likely to be successful M&A operations via the Isolation forest algorithm:*

Einstein once said that we cannot expect different results by doing the same thing, and as mentioned before, all specialized statisticians concur on the fact that 70% to 90% of M&A transactions are considered a failure by executives. In other words, the "*norm*" for such transactions is then to be failure, and only outlier transactions are successful. This implies that, rather then finding patterns regarding successful M&A transactions, we should focus on detecting outlier transactions, as the diversity, the volume and the characteristics of such transactions highly differ and finding a relevant cluster cannot be reached through pattern analysis (as the numbers of similar "successful" transactions will be very few, and other successful transactions with other characteristics would be considered too afar to be included in the cluster), we decided to go for anomaly detection algorithms.

As it is rather difficult to find an accurate assessment of thousands of transactions' success, unsupervised learning was also a key approach to our work, as in addition to allowing us to overcome the necessity of providing such assessment for transactions' success, can be effectively used for anomaly detection methods.

We decided to go for the Isolation Forest algorithm, given its ability to work with a contamination rate (which is universally defined as 20%), and also because it will isolate each transaction in the data and split them into normal transactions or anomalies, so outliers and inliers. This separation is based on the time required to separate the transactions. Segregating a point which is obviously within the norm of our dataset will be really difficult to isolate, as it will have many points surrounding it. For anomalies however, Isolation Forest can detect them quite easily as much less points (if not none) are surrounding the transaction treated.

Another perk of this algorithm is that it handles huge datasets with a considerable number of dimensions. Given the fact that our dataset contains 2651 viable transactions, and that we will be trying to detect anomalies based upon the main ratios we discussed before, which reach a number of 6, it is clear than Isolation Forest (IF) is the go-to option for us.

In order come up with such detections, we first have to use this piece of code :

```
from sklearn.ensemble import IsolationForest
model=IsolationForest(n_estimators=555, contamination=float(0.2),max_features=1.0)
```

Here, the parameters of the model have been selected through a GridSearch, with 555 being the estimated optimal number of trees to be used, as the Isolation Forest method, as its name suggests, is a tree based method. The contamination parameter concerns the estimated percentage of anomalies within our data.

Isolation Forest separates each point out from other points randomly and constructs a tree based on its number of splits with each point representing a node in tree. Outliers appear closer to the root in the tree and inliers appear in higher depth. In case of Isolation Forest, a forest of trees are created based on the 555 estimators and the entirety of features (max_features=1.0) parameters and the score is derived from it.

One can use score_samples/decision_fuction to get the normalized anomaly_score of each point here more the negative it is anomalous based on sklearn computation.

Till this step contamination factor has no influence on the scores. From here when we apply predict in order to return anomaly binary results (1 for normal transactions/-1 for anomaly transactions, in other words 1 for inliers, -1 for outliers) contamination acts as a cutoff/percentile on scores and returns top x percentile negative scores as anomalies. (For eg: If contamination is set as 0.2 (20%) then points with top 5% negative scores are labeled anomalies)

We then proceed to the detection of such anomalies, with using this line of code :

```
dataexp2['Anomaly']=model.fit_predict(dataexp2[['stockPrice_year-1','stockPrice_year','earningsPerShare_year-1','earnin
```

This line allows us to detect anomalies within our dataset based on the ratios mentioned before. Also, it will generate a new column in our dataset, called "Anomaly" which will consist of the binary predictions we got through the Isolation Forest method. We can list the used metrics as such :

- Investing company *stock price* year-1
- Investing company *stock price* actual year
- Investing company *earnings per share* year-1
- Investing company *earnings per share* actual year
- Investing company *debt equity ratio* year-1
- Investing company *debt equity ratio* actual year
- Investing company *net profit margin* year-1
- Investing company *net profit margin* actual year
- Investing company *inventory turnover* year-1
- Investing company *inventory turnover* actual year
- Investing company *current ratio* year-1
- Investing company *current ratio* actual year
- Investing company *addTotalDebt* year-1
- Investing company *addTotalDebt* actual year
- Investing company *enterprise value* year-1
- Investing company *enterprise value* actual year

We decided to rely on metrics prior to the actual year where the transaction was held in order to make sure that the acquiring/investing company's situation prior to the deal is deeply taken into consideration, so as to be sure that "*we see whether the deal should have taken place initially*".

After we run the previous line, we obtain a binary prediction:

- If value returned is 1: The transaction presents no particular issue and is considered normal
- If value returned is –1: The transaction presents particularly abnormal patterns regarding the selected metrics, and is hence considered to be an ***anomaly.***

We then have the following output, which was directly assigned to our new "Anomaly" :

```
0        1
1       -1
2        1
3        1
4        1
        ..
2646     1
2647     1
2648    -1
2649     1
2650     1
Name: Anomaly, Length: 2651, dtype: int64
```

The model has detected some anomalies within the dataset, and to count how many, we simply use the value_counts() method:

```
dataexp2["Anomaly"].value_counts()
```

```
 1    2121
-1     530
Name: Anomaly, dtype: int64
```

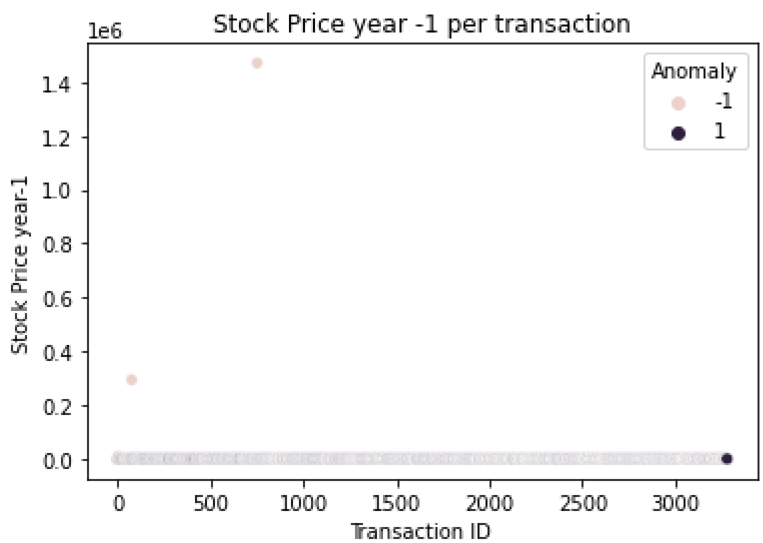
The algorithm has detected 530. Anomalies, which represent 20,7% of the dataset. Now, we have to see how these anomalies are distributed compared to the normal transactions.

Since we will be observing the behavior of all our transactions regarding the metrics that characterize the transactions, it is necessary to differentiate the transactions between anomalies and "normal" transactions. To do so, we decided to use the Seaborn library so as to rely on the hue parameter to make the distinction.

First, we made a PairPlot using all the metrics for our anomaly detection processes, so as to to have a general overview of the behavior. We will provide closer looks in the following, but if needed, the entirety of the PairPlot can be accessed in the notebook[6]. Also, we made sure to add the anomaly as hue parameter for better observation.

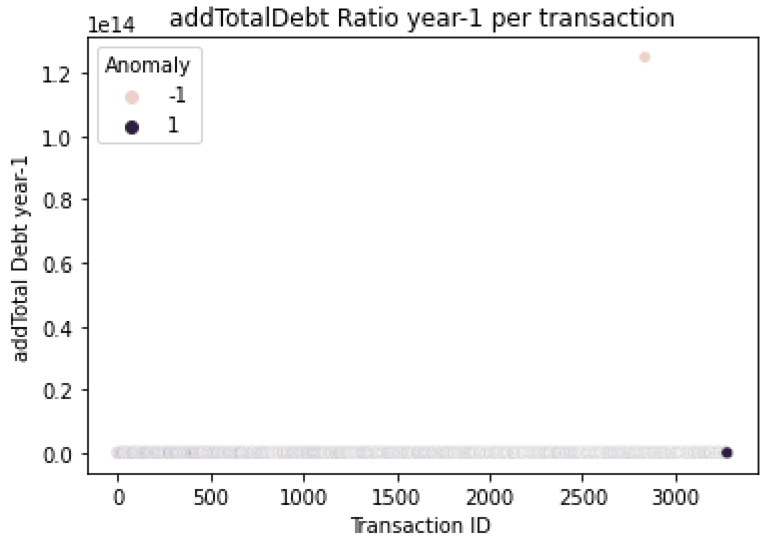
Here are some of the plots we made, where the patterns are clear regarding anomaly transactions :

Plot 1:

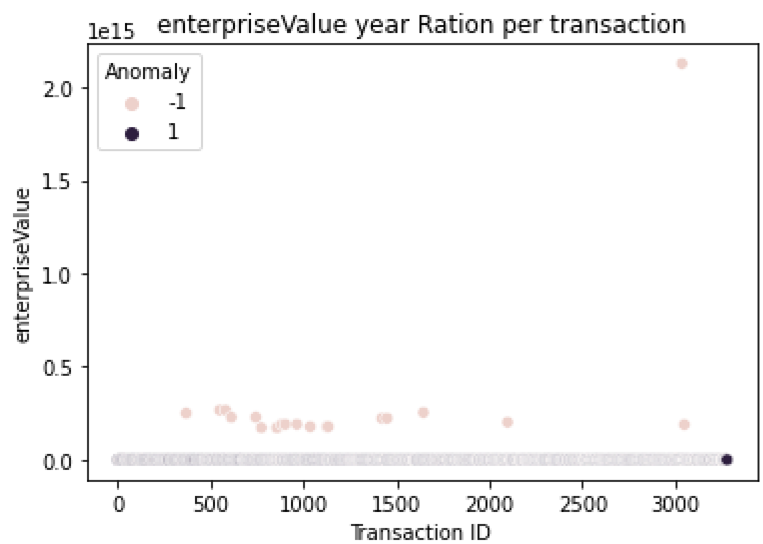


Plot 2:

[6] Colab Notebook in "Dossiers Étudiants", use the link : https://colab.research.google.com/drive/1tBEw9kTgDvg9WcrXNhgMpcV93vQ8QpBL?usp=sharing



Plot 3:



From here, we can clearly see that graphic interpretation regarding those plots is that there are sole outliers with very high values compared to the same ratios regarding other transactions, which make any sort of analysis almost impossible. Dropping these from our dataset will not affect the quality of our prediction and would allow us to come up with better visualization, so as to see whether anomalies detected are clearly anomalies.

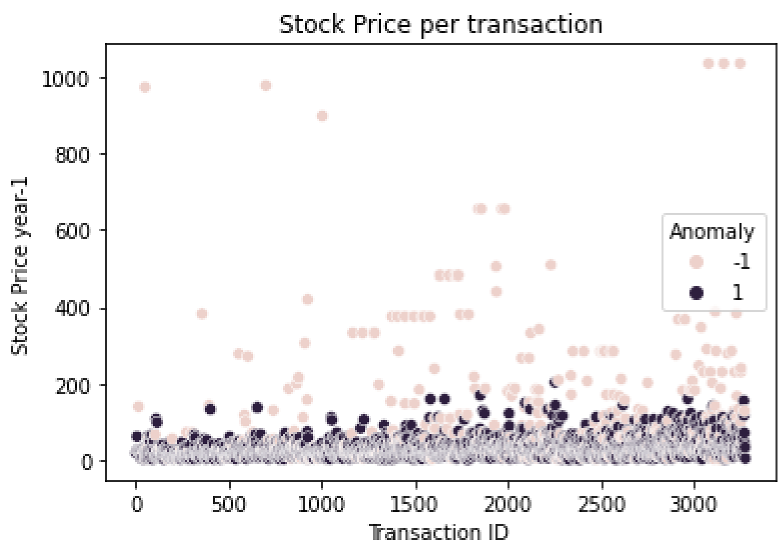
The logic behind these cuts is that such outliers are so isolated that any sort of data treatment could easily get rid of them. We want to see the patterns regarding the "anomaly" labelled points in a more general way.

To do so, we use thresholds using commands as such:

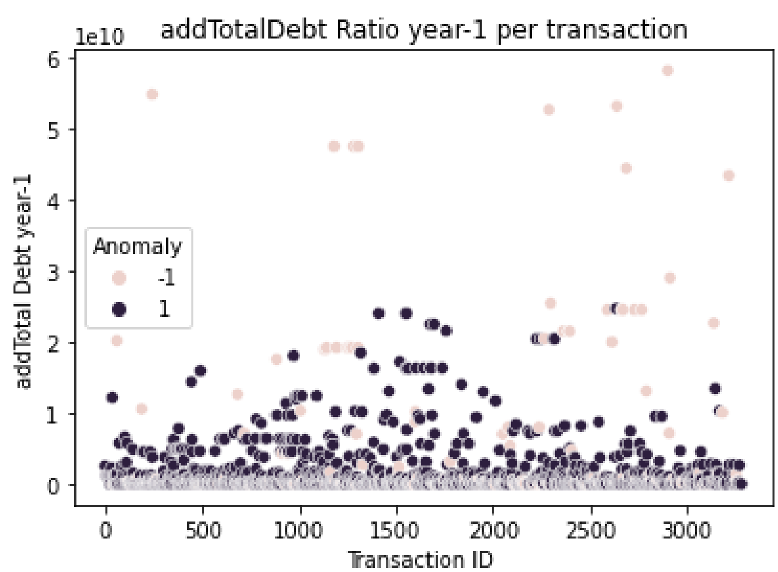
```
dataexp2=dataexp2[dataexp2['addTotalDebt_year-1']< 3e11]
dataexp2 = dataexp2[dataexp2['stockPrice_year-1']<5000]
```

NB : Similar lines of code will be applied to all of our metrics, had it been for year –1 or the actual transaction year.We then, by using Seaborn and the same code, have some plots of the sort :

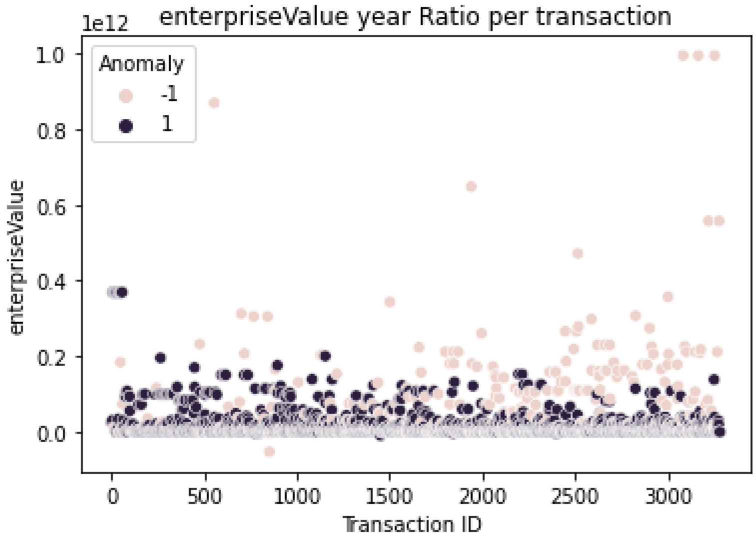
Plot 4:



Plot 5 :



<u>Plot 6:</u>



We presented here 3 metrics arbitrarily, other plots can be consulted on our notebook[7]. Still, the patterns observed are pretty much the same. We can clearly see that our anomalies present values towards our estimation metrics that are clearly differentiable.

In addition to that, we can also see that the algorithm didn't solely rely on 1 metric so as to detect anomalies: For example, in plot 4, there are multiple anomalies that are located amongst dense neighborhoods of "normal" points. That is explained by the fact that our cuts are multi-dimensional, it is reasonable to find that within "normal" transactions some anomalies depending on a certain metric.

Tri-dimensional plots were attempted; however, no clear sight could be achieved as given the considerable number of transactions and the density of the plot, we couldn't come up with a reliable observation.

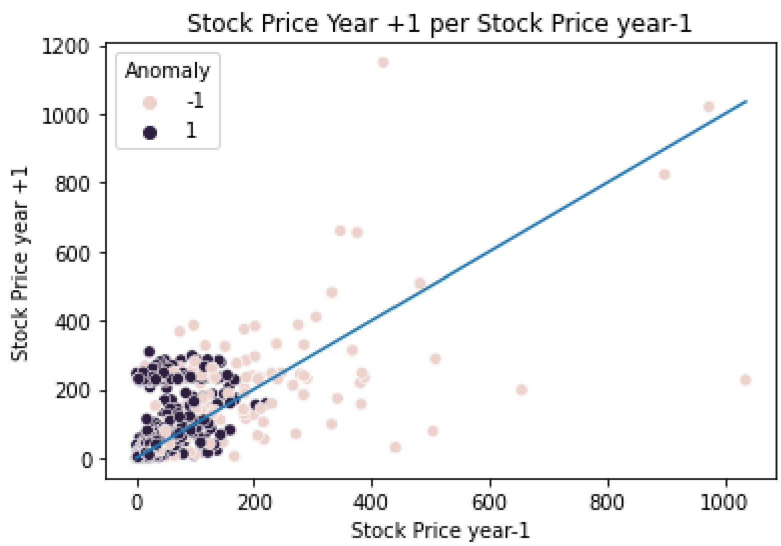
Another limit to our graphic anomaly observation is that we couldn't clearly see the distribution of anomalies who present negative values for our desired metrics. From plot 6, there is proof that some values are negative regarding our metrics, and also that anomalies are detected within that range too. However, their <u>amplitude is too small</u> to be fully appreciated, but we believe such phenomenon is also happening for other metrics. But you should keep in mind

[7] Colab Notebook in "Dossiers Étudiants", use the link : https://colab.research.google.com/drive/1tBEw9kTgDvg9WcrXNhgMpcV93vQ8QpBL?usp=sharing

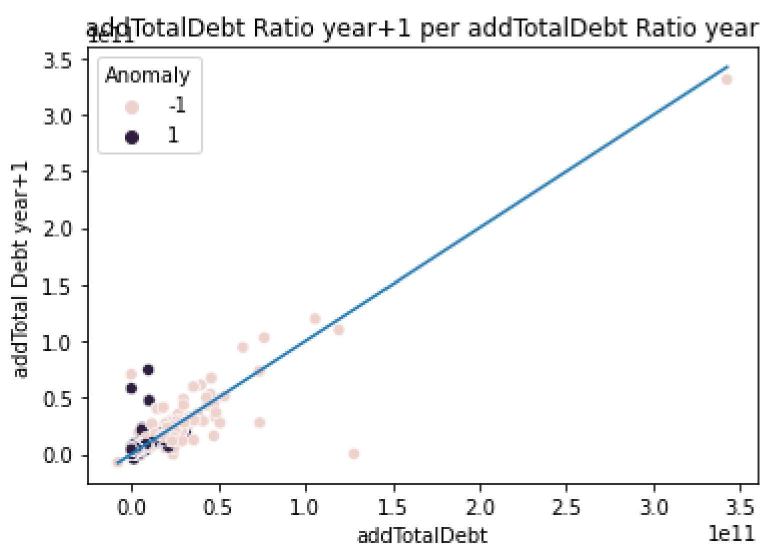
that having a negative value does not necessarily mean that an M&A transaction is destined to fail at all, as multiple investors use this type of situation to either enhance their results or even as a strategic maneuver. We can then conclude that our classification is reliable.

For further assurance, we also used the same metrics, but during the year *after* the transaction (year +1) and saw how our 'anomalies' fare once again compared to the rest of transactions, depending on the previous year's same metric value.

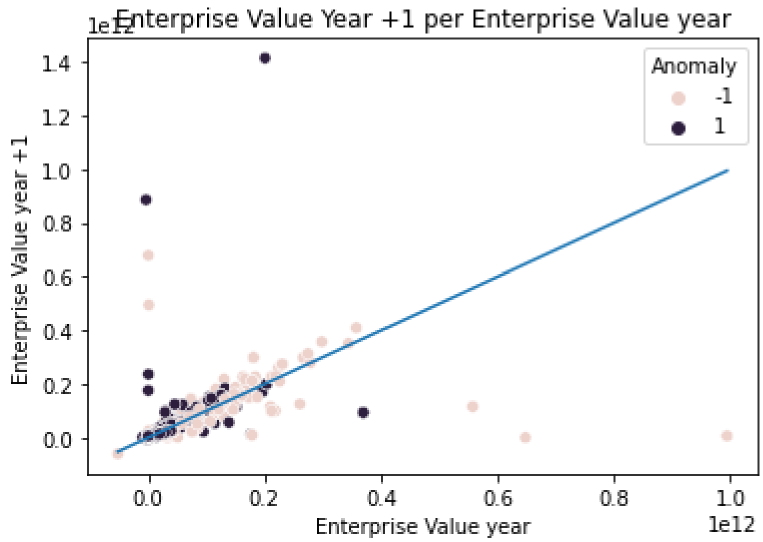
Plot 7:



Plot 8:



Plot 9:



Here, the investing company's stock price, added total debt ratio enterprise value in our anomaly transactions seem to remain generally close to the y=x line too, which shows that there isn't a big difference in regard to the previous year's ratios. However, there seems to be a little more anomaly on the upper side of the equality line (in blue), which shows slight improvement of the investing companies' situation (which are here labelled as anomalies). The reason for no major change (small improvements or underperformances) is the fact that the ratios can be altered with external factors: a company's stock price growth can be seriously altered due to a failed product, an accident, a communication mishap... And same goes for ratios presented here, or in the notebook for further observation concerning the rest of metrics.

We will present a more detailed approach to study the impact our anomaly labelled transactions on involved companies, which will go beyond the simple *year+1 metric vs current year metric,* towards a more detailed study of the impact of an M&A transaction on the investing company's key metrics and how its situation fares in regard to similar companies in terms of those metrics.

Providing a tangible success rate:

Classifying M&A transactions into two categories of successful and unsuccessful transactions requires a little more precision towards this so-called success.

There is little doubt that transactions labelled as anomalies, even though classified within the same category, are not the same. Obviously, this implies that the post-deal effect they will have on investing entities will be different, and sometimes, even though a transaction can be labelled by the algorithm as “anomaly” i.e. potentially successful, such estimation can be considered more or less robust depending on its characteristics.

We use this piece of code based on the same metrics as before so as to get an “Anomaly score” that will allow us to determine how potentially successful a transaction can be :

```
dataexp2['scores']=model.decision_function(dataexp2[['stockPrice_year-1','stockPrice_year','earningsPerShare_year-1','e
```

This will return for each transaction a certain score. In this case, here is our output :

```
0        0.034835
1       -0.201724
2        0.000478
3        0.005962
4        0.005962
           ...
2646     0.033591
2647     0.012021
2648    -0.124423
2649     0.024965
2650     0.030252
Name: scores, Length: 2651, dtype: float64
```

According to the Isolation Forest documentation[8] the lower the score, the more it is considered as being different from the data fitted within the model.

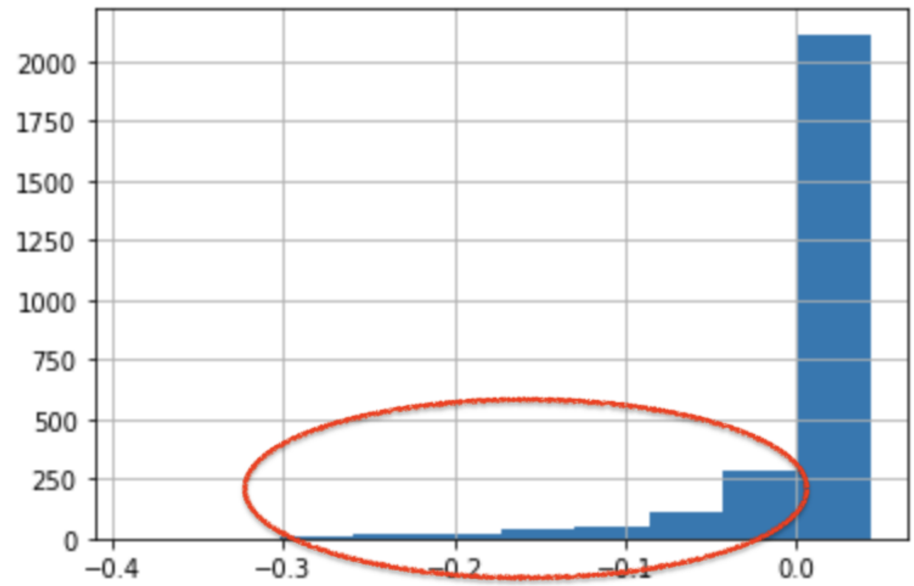
The anomaly score of an input sample is computed as the mean anomaly score of the trees in the forest, and the measure of normality of an observation given a tree is the depth of the leaf containing this observation, which is equivalent to the number of splittings required to isolate this point. In case of several observations n_left in the leaf, the average path length of a n_left samples isolation tree is added.

[8] Documentation obtained through the scikit learn website : https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

Now, let's observe how these scores are distributed :

```
dataexp2['scores'].hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa
```



Knowing that the lower, the more abnormal, and that negative scores generally represent outliers, and positive scores represent inliers, we will subsequently be focusing on the circled in red area, which concerns outliers, so, *in extenso* anomaly labelled transactions.

Let's isolate the anomaly labelled transactions in our dataset into a different one:

```
dft2=dataexp2.loc[dataexp2['Anomaly']==-1]
```

We can then directly proceed to generating a new column dedicated to success rates (consisting of percentages), using this formula :

```
dft2['Rates']=50+abs(dft2['scores'])*100
```

Here, we added the 50% constant because since a transaction is classified as an anomaly, it necessarily presents a success rate that would be higher than 50%, as "normal" labelled transaction can –as nothing is absolute– prove to be successful in much lower rates, that obviously go much lower than 50% : same formula would be used, but the 100*score (positive this time) shall be *subtracted (-)* from the 50% constant).

```
dft2['Rates']
```

```
1       71.966399
6       58.559995
19      53.108447
21      74.375244
25      57.001636
          ...
2633    65.854642
2635    53.312221
2637    59.733699
2639    51.289939
2648    61.718889
Name: Rates, Length: 530, dtype: float64
```

We then have defined success rates for each “anomaly” labelled transaction after regrouping them in one dataset, and this sort of display would give a more ergonomic, easy-to-use information regarding potential new predictions.

<u>Limitations of these methods:</u>

- M&A success, as mentioned before, does not *only* rely on economic and financial metrics, but also on desired synergies and qualitative parameters, such as expertise, experience...
- Some “normal” and “anomaly” scores are very close to 0.00, which means that some transactions may either be considered anomalies/potentially successful in an exaggerated way, or that one of these transactions’ might be underestimated.
- Since estimations for M&A transactions tangible success rate are pretty much inexistant, evaluating our method’s prediction regarding previous data was impossible. Regardless of us doing a double evaluation –Through what has been presented and what’s to come– of our results in order to confirm our theories and methods, a tangible precision rate would have also been an interesting assessment of our work.

- Further segmentation through three essential categories: Sector and Transaction Type. However, for data reasons, such approach was not doable as with the amount of “good” exploitable data, we would have ended up with only around a couple of hundred of observations, which could affect the quality of our predictions.

b.*Determining most likely to be successful M&A operations via the PSM method and difference in differences*

In our quest to formulate a proper success rate, we are invited to investigate the relevance of the Isolation forest method and its results. Besides validating its predecessor in this paper, propensity score matching and difference in difference are methods that we will use to quantify the EPS (Earnings Per Share), EV (Enterprise Value), Debt Equity Ratio, Net Profit Margin, Inventory Turnover and AddTotalDebt and Current Ratio metrics variation posterior to the acquisition deal, where it will be assessed in a 5-year post M&A deal radius.

What is propensity score matching?

In the statistical analysis of observational data, propensity score matching (PSM) is a statistical matching technique that attempts to estimate the effect of a treatment, policy, or other intervention by accounting for the covariates that predict receiving the treatment.[9]

A dear statistics and machine learning PhD professor of ours proposed this analysis even if it is conventionally used for public policy assessment. The idea is clear: in order to evaluate the impact of a certain M&A deal on the acquiring/investing company A, we want another company B (other companies) that did not engage in any deal, but that had the same chance/probability of making the deal in that time, to be matched with the company A. We therefore create subsets of the treated and untreated(control) groups to further investigate their financial data without loss of generality and getting rid of bias in our panel data (dropping some observations) which is a precious outcome that PSM delivers us.

[9] https://en.wikipedia.org/wiki/Propensity_score_matching

It is worth noting that our panel data was subdivided by deal year to mitigate any macroeconomically related market conditions.

For this matter, we use the 'MatchIt' package in R:

```
install.packages('MatchIt')
library(MatchIt)
matcheddata <- matchit(formula = b, method="nearest",data=Data,ratio = 3)
```

We opted for the 'nearest matching algorithm' which performs greedy nearest neighbor matching. A distance is computed based one the propensity score between each treated unit and each control unit, and, one by one, each treated unit is assigned 3 control units as a match.

The 'formula' we integrated in the matchit function contains the 7 financial elements dating one year short from the deal in order to not include the changes in the matches, nut to leave them for variation analysis.

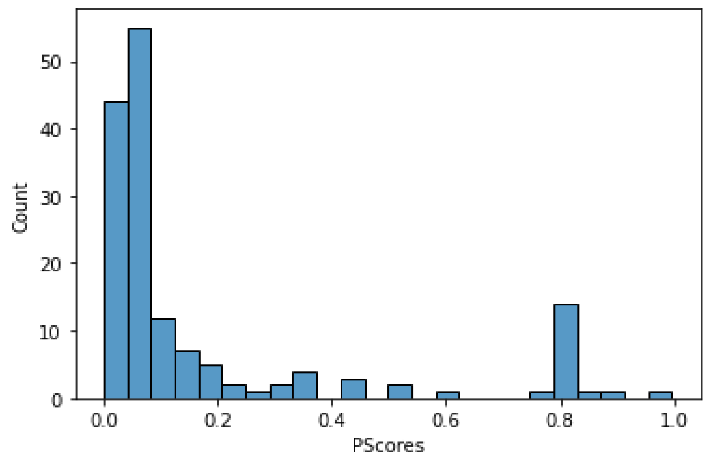
For notation purposes:

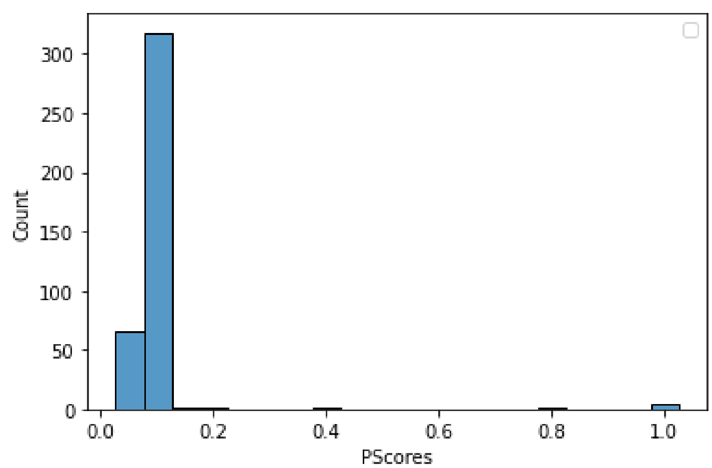
- **(TREAT = 1):** Treated companies are the ones that engaged in M&A activity within the 2000-2015 period
- **(TREAT = 0):** Untreated (companies that didn't engage in M&A activity within the 2000-2015 period)

Thus, creating subsets that contain one treated unit and 3 untreated ones, like shown in the following table representing the 10th, 11th and 12th subsets in the 2000 panel data.

Index	Treated's Index / Untreated Company's ticker	TREAT	Propensity Score	Subsets
18	42.0	1	0.0697977055250074	10
95	TRIB	0	0.0708826382694295	10
104	SMED	0	0.0680411589038821	10
148	BUG.AX	0	0.0693786273509266	10
19	44.0	1	0.0748894324692917	11
45	BDEV.L	0	0.0748746185811408	11
118	NWL	0	0.0748207439088523	11
134	JSG.L	0	0.074906358316404	11
2	4.0	1	0.0997970370287764	12
133	FRPH	0	0.0986687817138579	12
139	LWAY	0	0.100718656662502	12
149	RRC	0	0.100796860738729	12

An arbitrary look into Propensity Score histograms of the year 2000 and 2008 gives us the following plots:



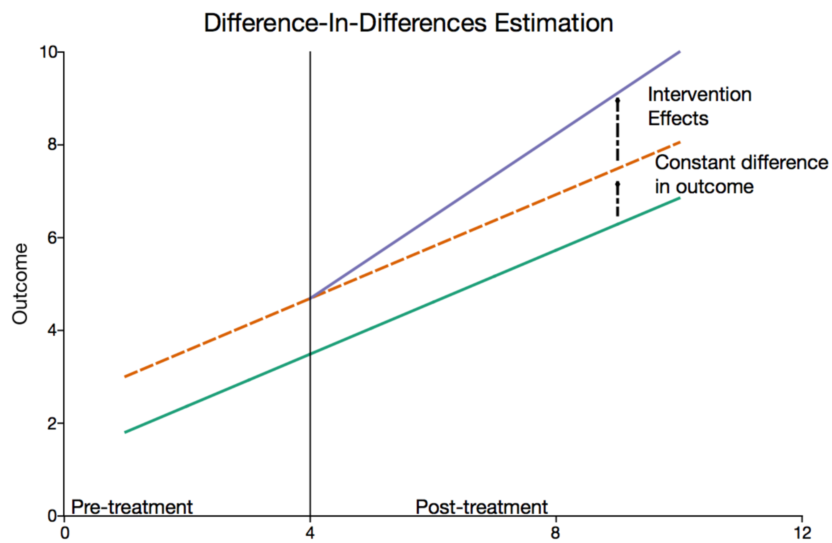


We are dealing now with quite different distributions as the first one ranges all the way to 1 Pscore (the 2000 panel data) and the second plot is clearly evolving around the 0.1 PScore (the 2008 panel data). Let us not forget that the more “PScores” is close to 1, the more the company having that PScore is susceptible of making an M&A deal in that respective period and the contrary goes for PScores close to 0. This may suggest several interpretations that lead some macro-economic considerations tied with the 2008 economic crisis. The study of these histograms will not be furtherly developed in our paper but invites all interested readers to discussion.

All in all, once the treated and untreated groups were matched (using PSM), we can immediately see the effect of bias reduction in the chosen 7 variables. For example, in the 2003 panel data, means and standard deviations were reduced by 72.6% in most of the variables intra-subclass. We now require a rigorous method of analysis to study the treatment (= M&A deal) and evaluate its impact on the studied variables. Therefore, we picked the difference in difference method (DID) as a reliable approach in causal inference subjects.

<u>What is the difference in difference method?</u>

The following illustration sums up DID and its intuitiveness.



Our study defines intervention effects as the average treatment effect on the treated (commonly known as ATET). We will apply the DID method on all studied variables represented as growth percentages to avoid some intra-subclass scale issues. Thus, and in the light of investigating our convergence to the Isolation forest findings, we will follow this path.

◊ Let us first create a dataframe of Intervention effect lists as that we are going to organize in a proper dataframe. It is worth noting that the varaible 'i'ranges from 0 to 5 years post acquisition.

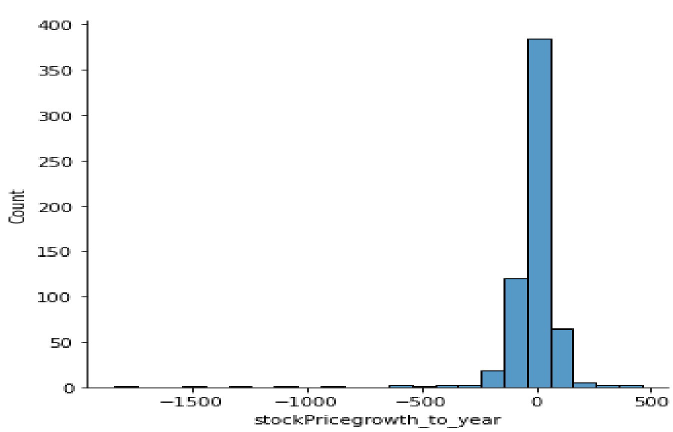
- **stockPricegrowth_to_year_i**
- **revenuePerSharegrowth_to_year_i**
- **debtEquityRatiogrowth_to_year_i**
- **netProfitMargingrowth_to_year_i**
- **inventoryTurnovergrowth_to_year_i**
- **currentRatiogrowth_to_year_i**
- **enterpriseValuegrowth_to_year_i**
- **addTotalDebtgrowth_to_year_I**

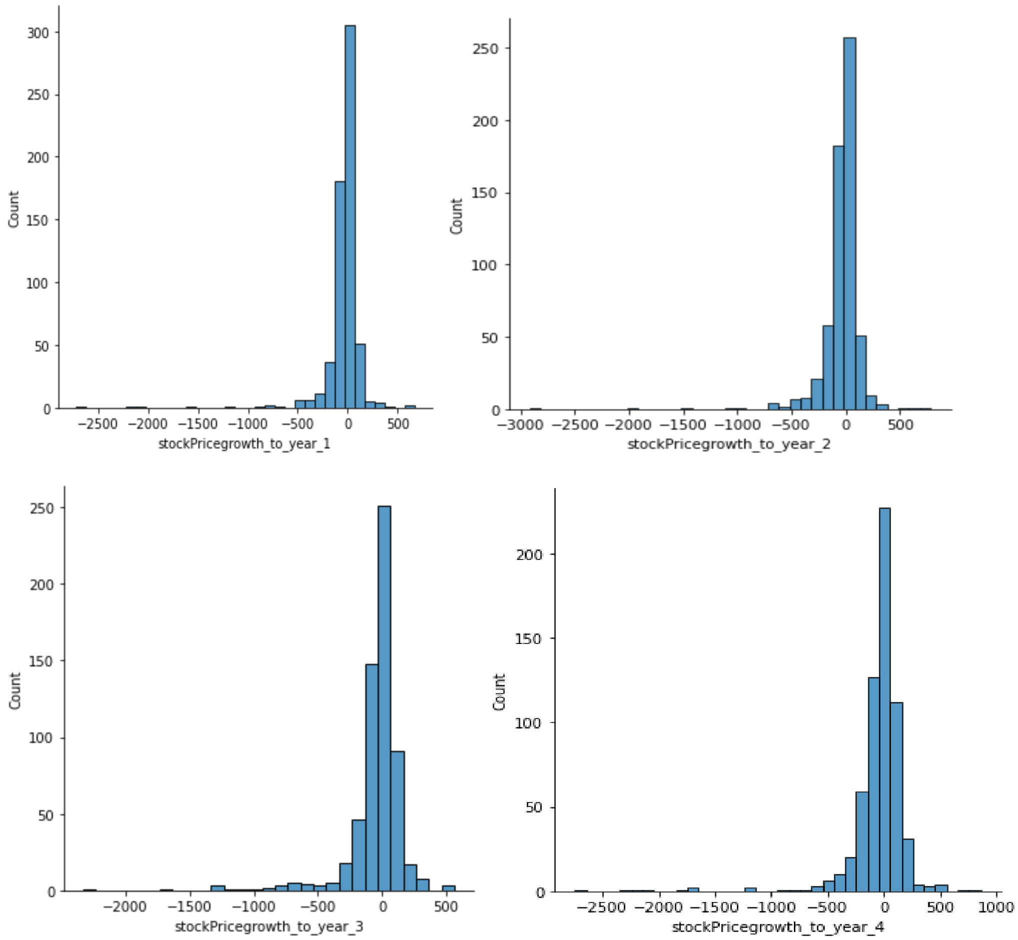
The growth formula will include both treated and untreated companies' growth as follows:

$$j\_growth\_year\_i = \left(\frac{j_{Year+i} - j_{Year-1}}{j_{Year-1}}\right)_{Treated} - \left(\frac{j_{Year+i} - j_{Year-1}}{j_{Year-1}}\right)_{Untreated}$$

$$j = stockPrice, revenuePerShare, debtEquityRatio, netProfitMargin, inventoryTurnover, currentRatio, enterpriseValue, addTotalDebt$$
$$Year = year\ of\ the\ deal$$

◊ All the histograms clearly converge to one statement: there is no substantial difference between companies who did M&A activity and other matched companies in all growth estimates in exception of the TotalDebt growth column where acquiring companies tend to have more trailing debt.





◊ The comparison with the Isolation Forrest suggests little to no convergence. Besides the indexes of the top performing firms are found at a rate of 22% in the anomalies presented in the Isolation Forrest method findings.

◊ It is worth noting that the general tendency of treated firms underperforming is validated throughout the whole intervention effect dataframe.

Limitations:

An old statistic saying goes: Correlation Does Not Imply Causation. This part's whole purpose was to deliver some evidence to some of our results and tried … yet causation can never be distinctively pointed at as with a high confidence level.

The results could eventually be more accurate when using the DID method if verifying assumptions like the parallel tendance before the treatment. The later adds more constraints towards external causation (not related to the M&A deal) and thus reinforce the unicity of treatment assumption.

VII.M&A operations pricing forecast

a.Data Exploration:

Mergers and acquisitions are increasingly emerging as the flagship of corporate finance. One of the key elements of the deal is the value of the merger or acquisition. It was proposed to embark on a Machine Learning approach instead of fundamental analysis in finance and to find a predictive model that can assess the value of mergers and acquisitions.

As mentioned before our dataset was carefully prepared through the Mergr platform and using an API to add more metrics.

One of the most important steps is the data exploration, which is what we started with in order to analyze our data properly.

b.Exploratory Data Analysis:

Firstly load the data and have a quick look at our dataset

```
df_train=pd.read_csv('/content/drive/MyDrive/Article M&A/Ultimate_Dataset.csv')
df_train.head()
```

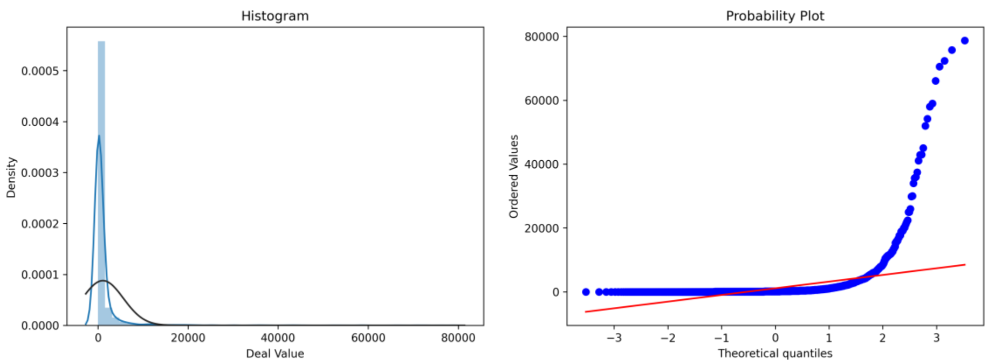
	Unnamed: 0	Index	Date	Target	Sector	Country	Transaction Type	Deal Value	Inv	stockPrice_year-1	stockPrice_year	stockPrice_year+1
0	0	0	2000.0	SmithKline Beecham Ltd.	Life Science	United Kingdom	Merger	75700.0	GlaxoSmithKline plc, 980 Great West Road, Bren...	NaN	NaN	NaN
1	1	1	2000.0	Etec Systems, Inc.	Semiconductors	United States	Add-on Acquisition	1800.0	Applied Materials, Inc., 3050 Bowers Avenue, P...	18.4289	15.5327	15.876
2	2	2	2000.0	Sirius XM Holdings, Inc.	Media	United States	Stake Purchase	1591.0	The Blackstone Group Inc. (PE), 345 Park Avenu...	NaN	NaN	NaN
3	3	3	2000.0	UST, Corp.	Financial Services	United States	Add-on Acquisition	1412.0	Citizens Financial Group, Inc., One Citizens P...	NaN	NaN	NaN

```
df_train.shape
```

```
(3282, 30)
```

c.Target Feature Analysis

We will plot **Histogram** and **QQPlot** to analyze the distribution and skewness type of the target feature.



d.Analysis:

- The Skewness co-efficient suggested the target feature is positive skewed.
- We will apply log transformation to the feature to make the distribution close to gaussian.
- We will apply log(1+x) transformation to avoid 0 values (if present).
- After transforming the variable, we will see both the above plots again.

First we identify the numerical and categorical variables and the missing values in order to impute them

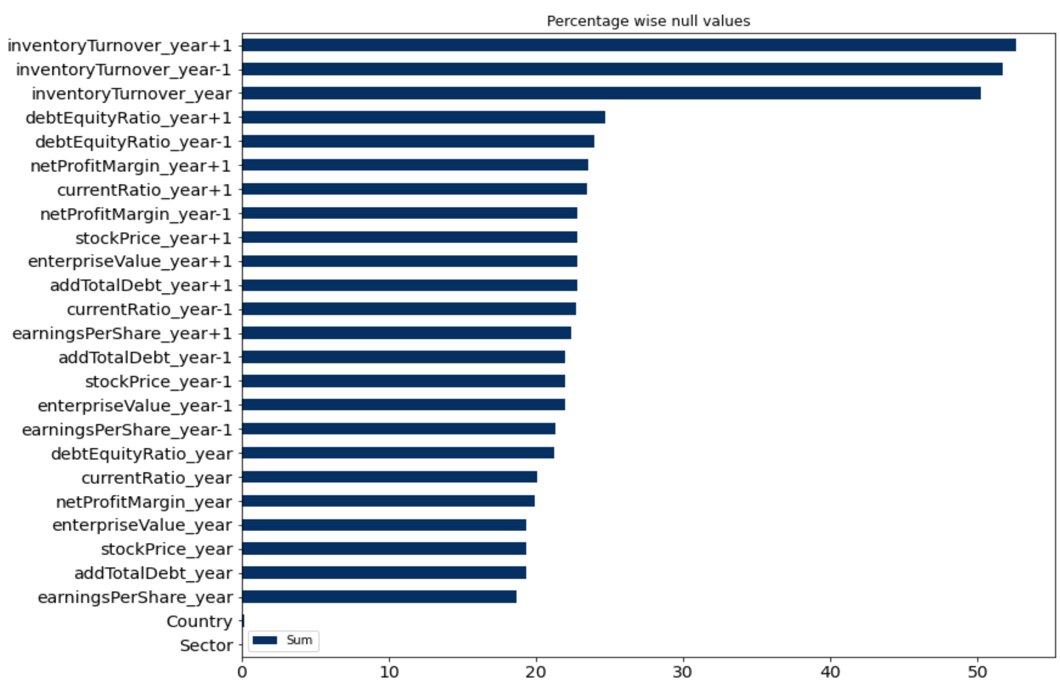
Categorical variables are : Target , Sector , Country , Transaction Type , Inv

Numerical variables are :

```
Unnamed: 0            int64
Index                 int64
Date                  Int64
Deal Value            float64
stockPrice_year-1     float64
```

stockPrice_year float64
stockPrice_year+1 float64
earningsPerShare_year-1 float64
earningsPerShare_year float64
earningsPerShare_year+1 float64
debtEquityRatio_year-1 float64
debtEquityRatio_year float64
debtEquityRatio_year+1 float64
netProfitMargin_year-1 float64
netProfitMargin_year float64
netProfitMargin_year+1 float64
inventoryTurnover_year-1 float64
inventoryTurnover_year float64
inventoryTurnover_year+1 float64
currentRatio_year-1 float64
currentRatio_year float64
currentRatio_year+1 float64
addTotalDebt_year-1 float64
addTotalDebt_year float64
addTotalDebt_year+1 float64
enterpriseValue_year-1 float64
enterpriseValue_year float64
enterpriseValue_year+1 float64

Regarding missing values we found the following result:

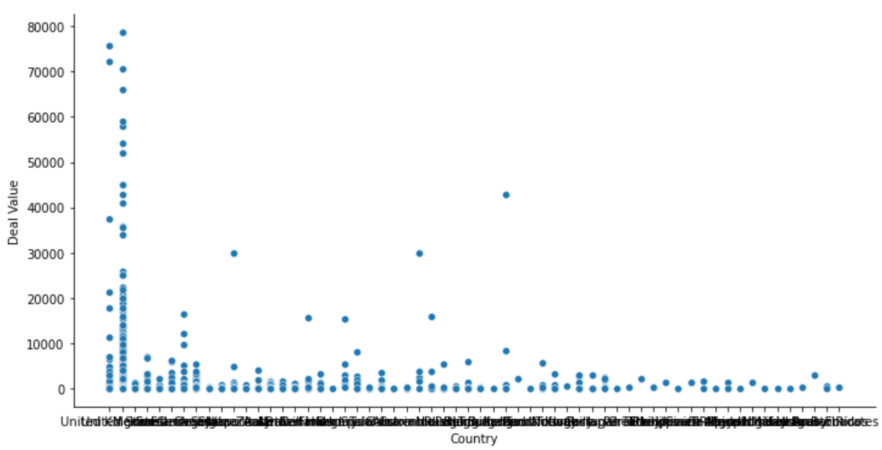


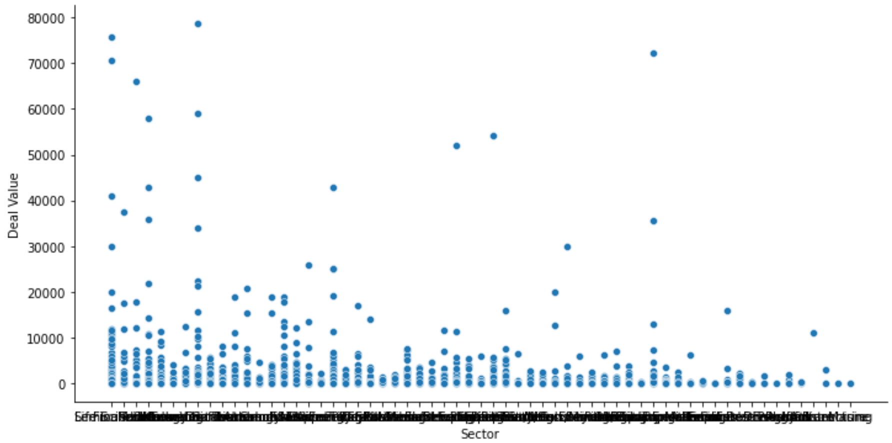
inventoryTurnover_year+1, inventoryTurnover_year, and inventoryTurnover_year-1 have more than 50% values as NaN. Our first impression will be to drop them but instead in our analysis we will take these features and see if they can increase our results.

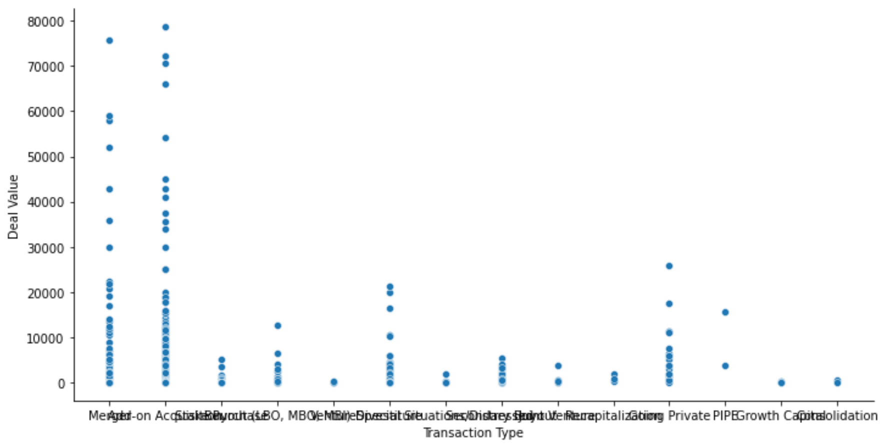
To deal with missing values we've chosen the K-Nearest Neighbors imputer , the idea in **KNN methods** is to identify 'k' samples in the dataset that are similar or close in the space. Then we use these 'k' samples to estimate the value of the missing data points. Each sample's missing values are **imputed** using the mean value of the 'k'-neighbors found in the dataset.

```
df_numeric = df_train.select_dtypes(include='number')
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=5)
df_numeric = pd.DataFrame(imputer.fit_transform(df_numeric),columns = df_numeric.columns)
```

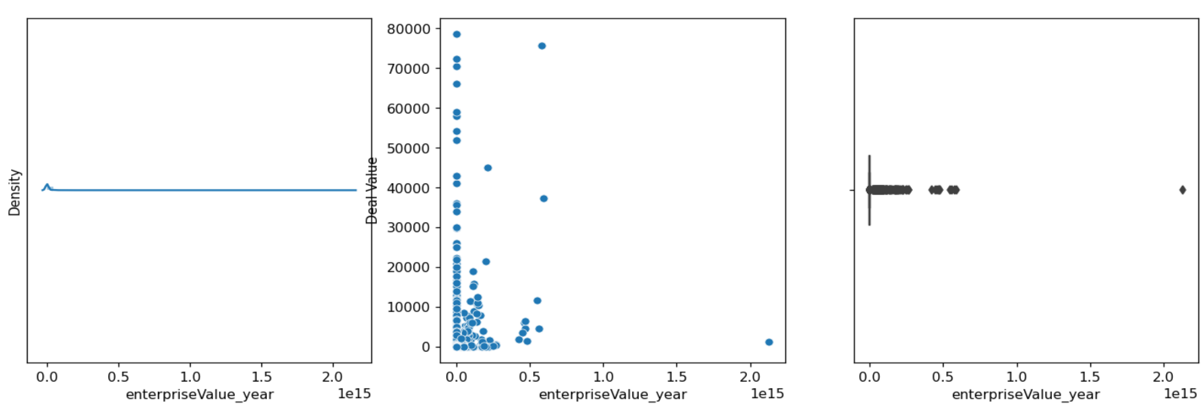
In order to assimilate the problem, we plotted our target variable according to the other variables, either categorical or numerical.

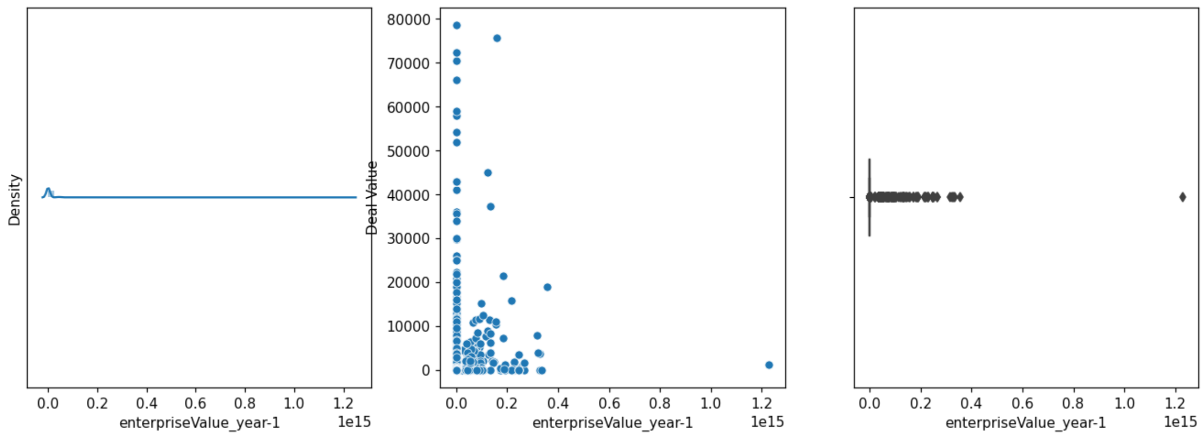






After visualizing the categorical variables , we've plotted the numerical variables with the scatterplot function and we've added also the density distribution and boxplot in order to detect some outliers





The relation between the variables seems to be linear for values of our variable target between 0 and 1000 (85% of the values).

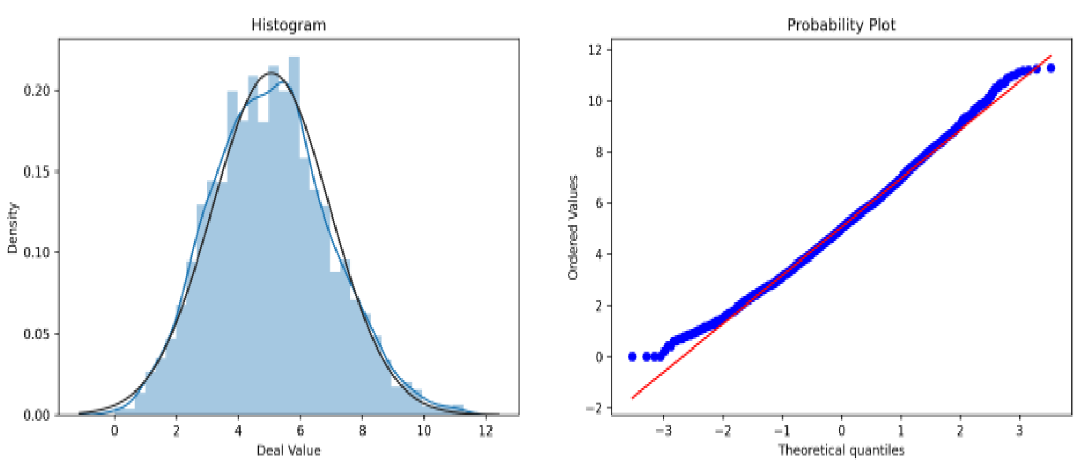
First we will evaluate the linear regression model and then we will evaluate other decision tree models.

As we have just mentioned in the analysis of the distribution of our target variable (Deal Value) we will use the log transformation (1+x) to make the distribution normal and reduce the skewness coefficient, we applied the same principle to all the other variables and therefore we had the following results:

```
numeric_features = df_train.skew().index

## Getting all the skewed features (skew > 0.5 or skew < -0.5)
skewed_features = df_train[numeric_features].skew()[np.abs(df_train[numeric_features].skew()) > 0.5].index

## Performing log(1+x) transformation
df_train[skewed_features] = np.log1p(df_train[skewed_features])
```



We can clearly notice that the distribution is normal as well as for all the other variables that have a remarkably large skewness coefficient, we will then discuss the effect of making the distribution Gaussian.

To start the linear regression we need to deal with categorical variables, we chose the get_dummies protocol which works with the factorization principle and adds columns to our dataset in advance with binary data (0 or 1 if the observation corresponds or not to the factorized column), but for these categorical variables they contain a large number of categories but not representative enough, so we just factorized that represent the majority as you can see below.

```
(df_train['Country'].value_counts()/df_train['Country'].count())*100
```

```
United States     74.520116
United Kingdom     5.521956
Canada             5.127531
Germany            1.788062
France             1.498817
                     ...
Romania            0.026295
Turkey             0.026295
Barbados           0.026295
Macedonia          0.026295
Uruguay            0.026295
Name: Country, Length: 61, dtype: float64
```

```
(df_train['Transaction Type'].value_counts()/df_train['Transaction Type'].count())*100
```

```
Add-on Acquisition               75.441804
Divestiture                      16.544790
Merger                            2.955515
Buyout (LBO, MBO, MBI)            1.432054
Secondary Buyout                  1.096892
Stake Purchase                    0.944546
Going Private                     0.792200
Special Situations/Distressed     0.274223
Joint Venture                     0.121877
Growth Capital                    0.091408
Recapitalization                  0.091408
Venture                           0.091408
Consolidation                     0.060938
PIPE                              0.060938
```

So after having a dataset containing only numeric variables, we proceed to split our data into training and testing sets and evaluate the model through the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics.

```
#Let's build our Linear Regressor
regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

```python
#Evaluate the performance of the algorithm. We'll do this by finding the values for MAE, MSE, and RMSE.
print('Mean Absolute Error:', metrics.mean_absolute_error(np.exp(y_test)-1, np.exp(y_pred)-1))
print('Mean Squared Error:', metrics.mean_squared_error(np.exp(y_test)-1, np.exp(y_pred)-1))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(np.exp(y_test)-1, np.exp(y_pred)-1)))
```

```
Mean Absolute Error: 1011.039345817981
Mean Squared Error: 13783928.414921572
Root Mean Squared Error: 3712.671331389512
```

Analysis:

By observing the results we obtained an error (RMSE) that represents 400% of the average of our variable target, which puts our model in difficulty and leads us to two hypotheses:

1-The relationship between our target variable and the other variables is not linear and it is better to test decision tree models.

2- The distribution of our variable target is articulated on a margin of values between 0 and 1000 (85% of the values) with a standard deviation of 4682 and a mean of 1000, so it is useful to dropping outliers that reach values of 78000 and that have botched our model.

Testing the first hypothesis :

Starting from the first hypothesis we tested the Random Forest model and XGBoost as two models known for their high precision in regression and especially in finance. we launched a GridSearch loop to determine the most performing parameters (Hyperparameters Tuning) of these two algorithms . and we obtained the following results

```python
from xgboost import XGBRegressor
xgb = XGBRegressor(objective ='reg:squarederror', colsample_bytree = 1, learning_rate = 0.1,subsample=0.8,
                          alpha = 10, n_estimators = 555,max_depth=7,booster= 'gbtree',nthread=4,seed=45)

xgb.fit(X_train, y_train)
```

```
from sklearn.ensemble import RandomForestRegressor
hyper_params1 = {'n_estimators': 555,
 'min_samples_split': 10,
 'min_samples_leaf': 4,
 'max_features': 'auto',
 'max_depth': 70,
 'bootstrap': True}
rf = RandomForestRegressor(**hyper_params1)
# Train the model on training data
rf=rf.fit(X_train, y_train)
```

```
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(np.exp(y_test)-1, xgb_preds)))
```

```
Root Mean Squared Error: 3531.100011537632
```

```
# Use the forest's predict method on the test data
predictions = rf.predict(X_test)
# Calculate the absolute errors
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(np.exp(y_test)-1, np.exp(predictions)-1)))
```

```
Root Mean Squared Error: 4723.549230103257
```

The results obtained lead us to the second hypothesis which calls into question the distribution of the variable target and the great influence of the outliers on the model, we will come back to these models in case of validation of the second hypothesis to hope for an improvement of the RMSE.

Testing the second hypothesis:

to better assimilate the problem, we came back to completely describe our variable target,

```
df_train['Deal Value'].describe()
```

```
count     3282.000000
mean      1095.690527
std       4541.124044
min          0.000000
25%         40.000000
50%        151.000000
75%        517.000000
max      78700.000000
```

It was proposed to keep the values between 0 and 1000 which represents 85% of all the values and which is a representative portion in case of model validation.

```
drop_index = df_train['Deal Value'][df_train['Deal Value'] > 1000].index
```

```
df_final = df_train.drop(drop_index).reset_index(drop=True)
df_final.shape
```

```
(2750, 30)
```

As before, we follow the same steps of splitting and fitting the model, and we run a linear regression which gives the following results:

```
#Evaluate the performance of the algorithm. We'll do this by finding the values for MAE, MSE, and RMSE.
print('Mean Absolute Error:', metrics.mean_absolute_error(np.exp(y_test)-1, np.exp(y_pred)-1))
print('Mean Squared Error:', metrics.mean_squared_error(np.exp(y_test)-1, np.exp(y_pred)-1))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(np.exp(y_test)-1, np.exp(y_pred)-1)))
```

```
Mean Absolute Error: 146.5238289328098
Mean Squared Error: 52377.249365506796
Root Mean Squared Error: 228.86076414603443
```

It turns out that the second hypothesis was the closest to reality since we obtain an error of 22% which leads us to validate the linear regression model, we add that even after having the model of Random Forest and XGBoost, we obtained an error of 24% which remains a good result, but the most efficient is that of the linear regression.

---

[i] https://www.investopedia.com
[ii] https://corporatefinanceinstitute.com/resources/knowledge/deals/mergers-acquisitions-ma/
[iii] https://imaa-institute.org/mergers-and-acquisitions-statistics
[iv] https://mergr.com
[v] https://financialmodelingprep.com

VIII.Conclusion:

During our pricing study, we focused on Exploratory Data Analysis and the performance of predictive models, the key word was the distribution of our target variable which had outliers which strongly botched our models and at the same time these values represented a very small portion of the values on which we trained our model, as a perspective for the future the Deal Value (Our Target Variable) can be treated with the principle of Time series Forecasting if we come back to a good dataset detailed with the dates of each merger and acquisition and their value (Similar to a Stock Price approach). This paper suggests evidence towards a under-performance of M&A engaged firms against their pairs.