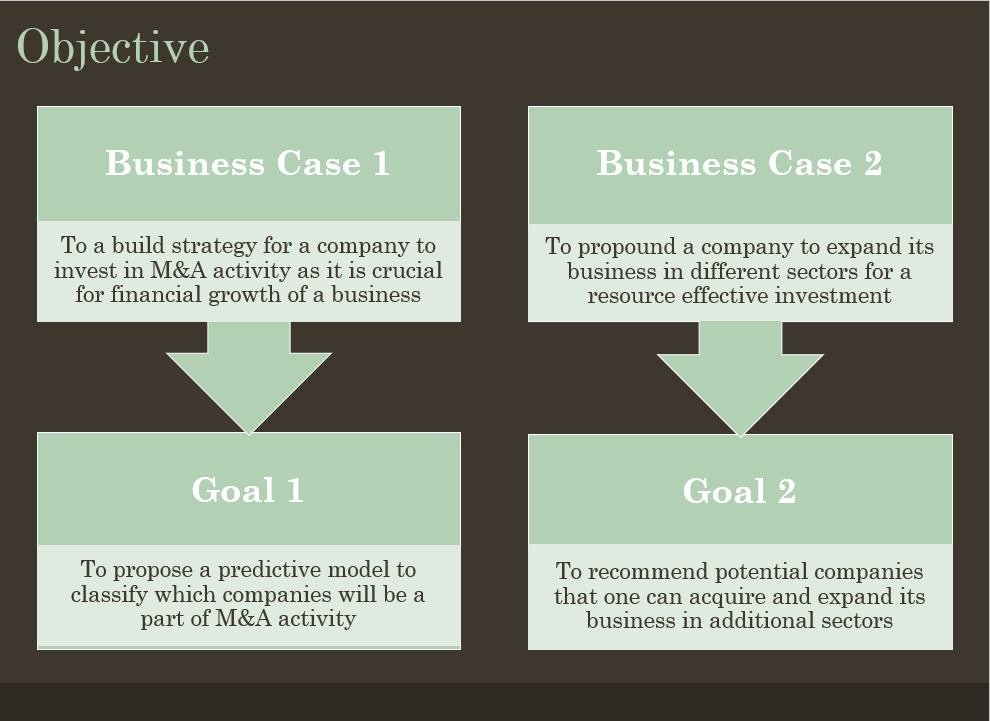


Merger and Acquisition (M&A) prediction has been an interesting and challenging research topic in the past the few decades but we are unmindful of how it happens. We remain untouched with the intricacies of this activity, the resources involved, risk factors, and output that comes out of this activity. To a common man, this is some rocket science in terms of business and economics which affects the market and the stocks, but we remain unenlightened of why the stock markets dances to the tunes of this activity and how investors go default or multiply their profits just in a day. Our motivation in this project was to read the lines between these activities and see how we can help all the company investors to be profited by M&A. In simple terms, we want to predict a being merged, acquired or not participating in that activity at all.

To an investor, a proper study of components of the company, how these components relate to each other and how it would help in deciding which company to invest in will be a magic wand that can-do wonders. We are giving that magic wand to all the investors by predicting

* Whether a company could take part in a merger and acquisition activity.
* Whether a company on the verge of the shutdown could be saved from this activity.
* If a company is going to take part in this activity, investing in which all sector would help the company to gain profits without starting the entire business from scratch.

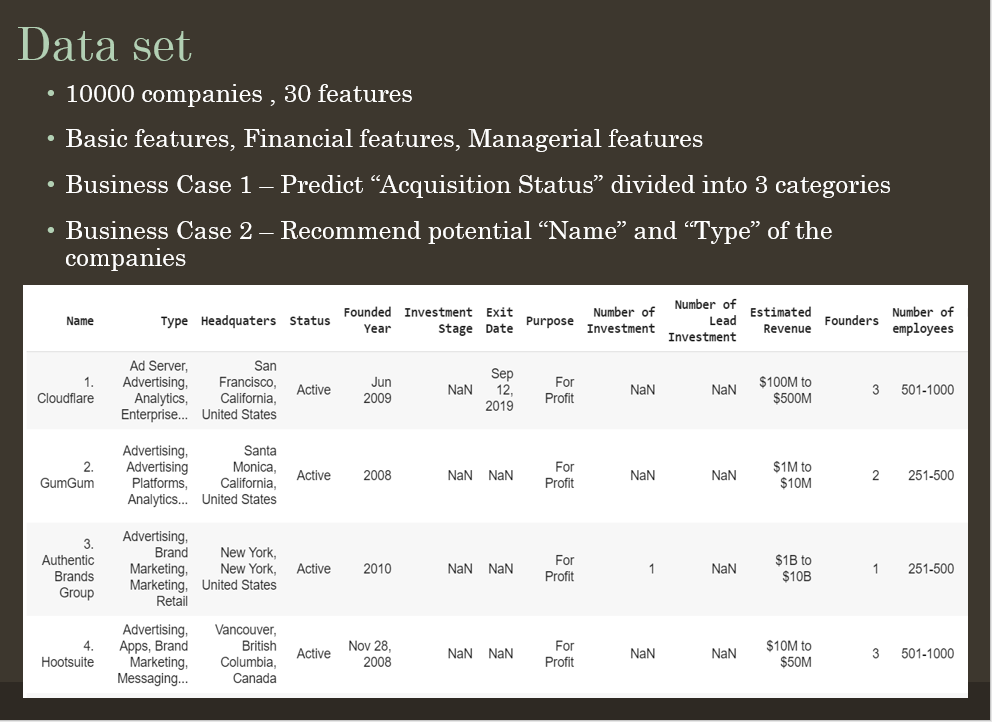
We aim to facilitate two different classes of people. First, the investors, who can study the merger and acquisition activity of the companies and decide their move based on these results. Second, the common man who invests in stocks which get largely influenced by these merger and acquisition activities. We are trying to suggest an easy way to both the classes by analyzing all the important factors of the companies and coming out with an understandable solution.



Merger and Acquisition is a business activity that is very important to keep the dynamics of the company moving and thus with this objective companies choose to merge with or acquire companies in similar or different sectors. This activity is very crucial, and a lot of resources are invested in this activity. Thus, a successful merger or acquisition activity depends on various components of the company as well as the leadership.

In this project, we aim to solve 2 very important business cases. The first business case deals intending to build a strategy for a company to invest in M&A activity as it is crucial for the financial growth of a business. Thus, our goal is to build a model that could tell whether a company could participate in a Merger and Acquisition activity by studying its basic, financial and managerial features. As a result, if a new company is given as input to the model, it can tell whether a company would acquire, would be merged or would not participate in the activity. The application of this model could save thousands of dollars for a company and could help a company study its competitor's actions.

The Second Business case is to propound a company to expand its business in different sectors for a resource-effective investment to build a recommendation system to recommend potential companies that one can acquire and expand its business in additional sectors. After analyzing various factors and getting the result that a company could be a part of merger and acquisition activity, one crucial question arises which sector companies would be suitable to merge to or acquire if a company wants to maximize its profits. Maximizing profit not just depend on a single component revenue of the company but more components also. So keeping that in mind we targeted building recommendation systems which will consider all the important features such as the basic features, financial features and managerial features of companies and will recommend companies is based on them. This is a two-way treat as we can get directly which companies the target company can perform the merger and acquisition activity with and also other companies in the sectors of the recommended companies. This solution not just increases the possibility of expanding but the scope of expansion to maximize profits for a business.



The most crucial challenge that we faced in our project was getting the Data. After selecting the goal, we did extensive research on the factors that account for a Merger and Acquisition. Our primary reference was the research paper titled “A Supervised Approach to Predict Company Acquisition with Factual and Topic Features”. After getting the domain knowledge, we started with the hypothesis development process. The initial stages of the process involved deciding what features could help in the successful prediction of involvement of a company in a merger or acquisition process. We divided our feature selection process into 3 categories.

Abundant empirical evidence over the past decade has suggested that the size of training data eventually becomes more critical than the sophistication of the algorithms.

Basic features – The features which decide the basic statistics of the company. Basic features in our dataset include the name of the company along with its sectors, status, headquarters, founding year, IPO status and the number of employees.

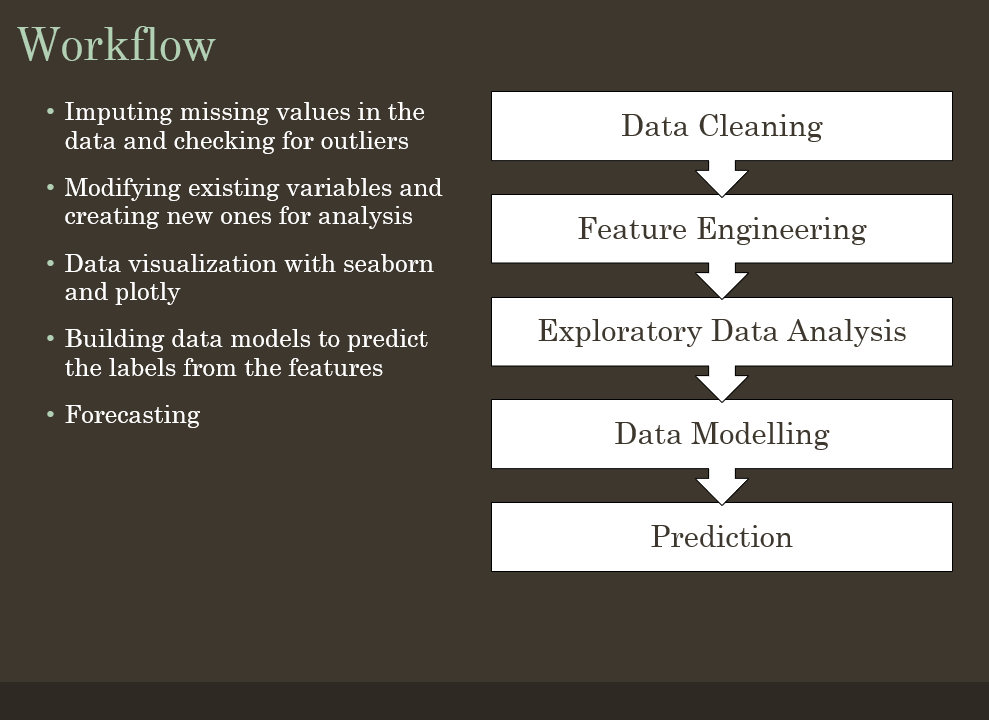
Financial features – The features which decide the financial health of a company. These include the net worth, funding amount, funding rounds and investment stage.

Managerial features – Conventional wisdom is that the experience and influence of founders is an invaluable impact on a company. These features decide the experience and influence of founders and managers in the company. These include the number of founders, job title, number of founded organizations by the founders and primary organizations of the founders.

Our label column consists of 3 classes namely “Made Acquisitions, “Was Acquired”, and “Did not participate”.

We did not have a public dataset readily available anywhere, so we had to web scrape the Crunchbase website for most of the features using Beautiful Soup. We also used Yahoo Finance for some financial features such as ‘Total Funding Amount’ and ‘Number of Lead Investments’.

The final Dataset that we uploaded on Kaggle had a total number of 10,000 companies with 30 feature columns and 1 label columns which is the ‘Acquisition Status’.

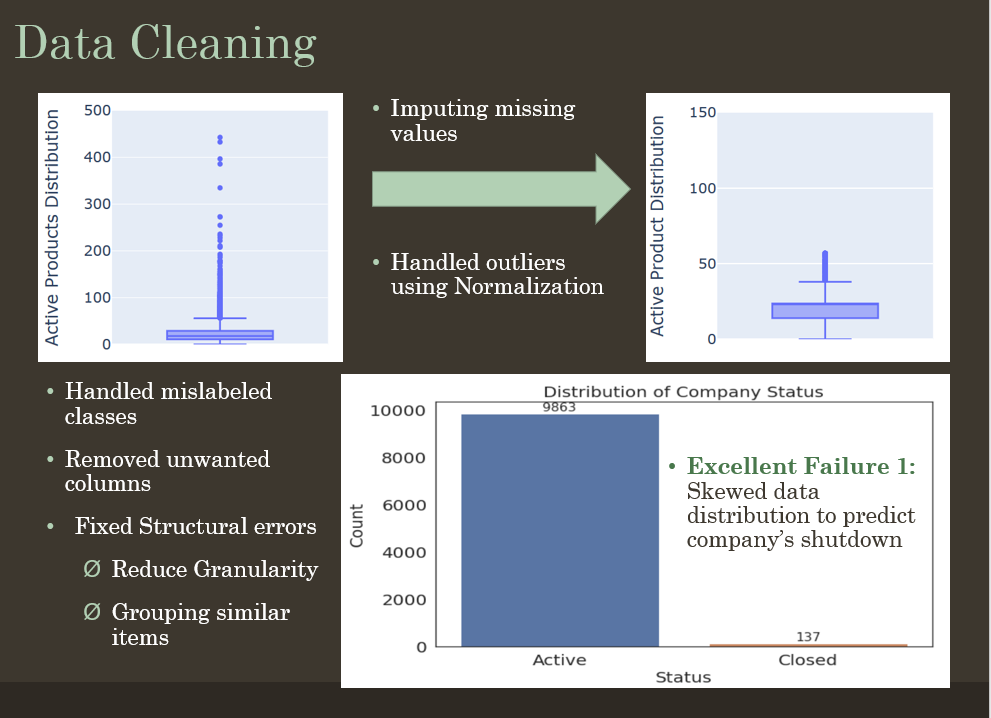


After generating the hypothesis and deciding all the necessary features, we were finally able to get the data. After getting the data, we planned to solve the problem in 5 stages.

* Data Cleaning – Our goal was to produce a dataset where a feature should be consistent dataset with other similar features in the dataset. The process involves detecting and correcting (or removing) corrupt or inaccurate records from the dataset and identified incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Features such as “Exit Date” and “Portfolio Companies” had a lot of missing data and features such as “Active Products” had some outliers. We then imputed the missing values and processed the outliers to produce a more meaningful structure
* Feature Engineering – This process involved modifying existing variables to create new ones for analysis. In our dataset, the “Founding year” and “Exit Date” variables were inconsistent as some values consisted of just the year while some had the entire date, we fixed them by just considering the year. Dependent features were dropped. Many features required grouping operation and binning. The continuous features were converted into categorical and we one hot encoded most of them.

* Exploratory Data Analysis - We performed the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypothesis and to check assumptions with the help of summary statistics and graphical representations. We plotted bar graphs for all the features to observe the balance. Along with observing the balances, we observed the mean, standard deviation along with the minimum and maximum values of the features using describe() function in python

* Data Modelling and Prediction – After the EDA, we then started solving business cases. For the first problem, i.e. classification of whether a company would be involved in merger or acquisition, we used ensemble methods. For the second business case i.e. recommending other companies and sectors to merge, we used cosine similarities as a parameter in the nearest neighbor algorithm.



Data cleaning is the process of ensuring that your data is correct, consistent and useable by identifying any errors or corruptions in the data, correcting or deleting them, or manually processing them as needed to prevent the errors and inconsistencies.

The “Type” column in our data set required a lot of work as each company has multiple sectors and some of the similar sectors were named differently. When we took a count, we observed the total sectors to be 675. These were reduced from 675 to 26 as similar sectors were merged under one umbrella.

The “Job Title” column of the data frame had the same problem as the “Type” column. The similar job titles were grouped, and the number of categories was reduced considerably.

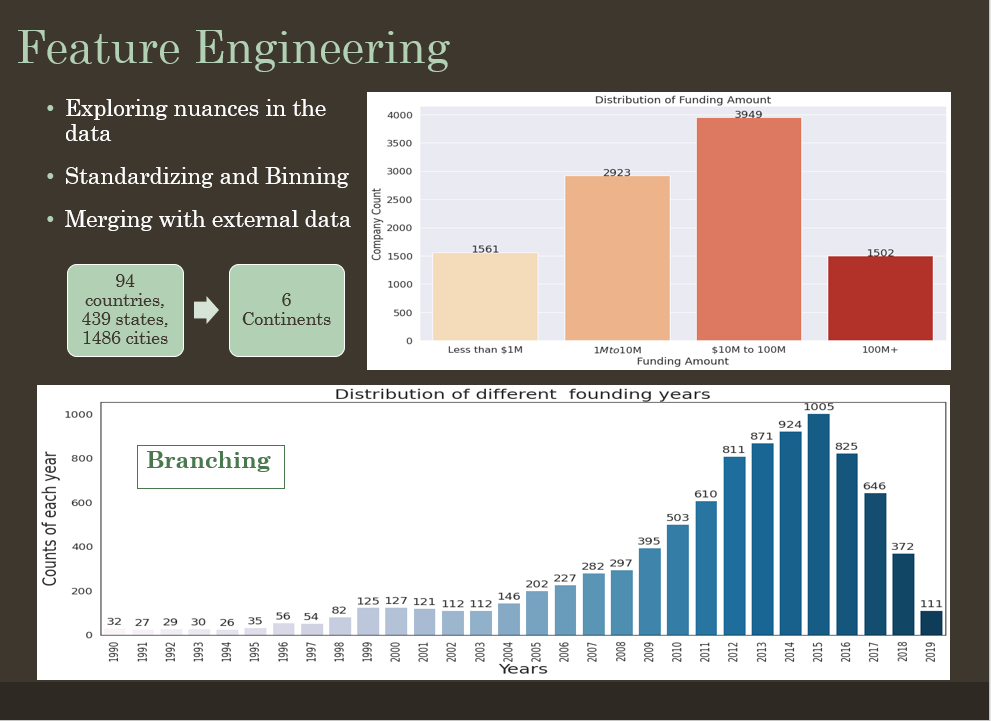
The missing values in the columns that were continuous initially, were replaced with the mean/ median. The missing values in the categorical columns were replaced with mode.

Columns such as “Investment Stage” have 95% missing values so it was dropped.

Outliers from the “Estimated Revenue” and “Active Products” column were handled using normalization.

One of the tasks in our Milestone 1 was to predict the success of the merger. We tried to predict which of the companies were in Active status after the merger. This was our Excellent Failure as more than 95% of the data was skewed towards Active status. Hence, we realized that this was not a possible task for the given dataset.

Unnecessary columns such as “Full Name of Lead Investor” and “Name of company” was dropped.



This process involved modifying existing variables to create new ones for analysis. In our dataset, the “headquarters” feature consisted of the city, state, and country names which was leading to a lot of skewness and inconsistency, so we categorized the headquarter location containing 94 countries, 439 states, and 1486 cities into 6 continents.

The columns such as “Number of Employees” and “Estimated Revenue” were continuous. We assigned certain bins to the columns to make them categorical. The bins were calculated based on the median to create balance.

The “funding amount” column had amount based on the currency of the respective country where the company was located. We first converted all the values to USD. Now the column had continuous values, we changed it to categorical using group conversion.

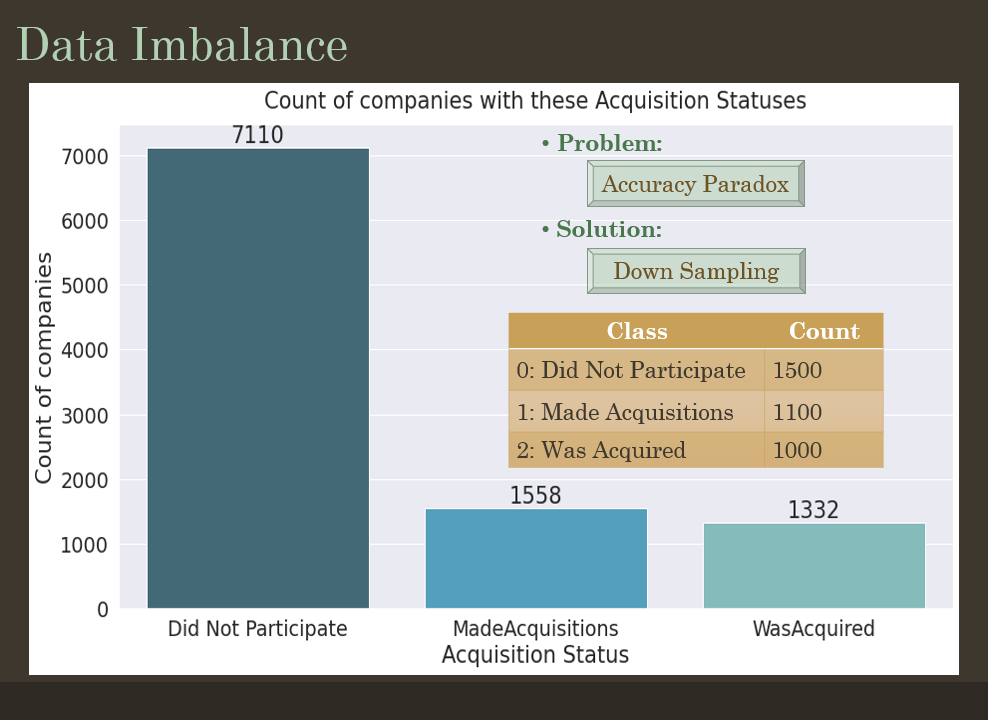
The “founding” and “exit” columns were inconsistent as some had just the year and some had the entire date. We created new columns considering just the founding and exit year.

The columns such as “Number of Investments” and “Number of Investors” were made categorical by assigning different ranges.

“Number of Employees” column was changed entirely by assigning uneven groups to first make it categorical and solving the problem of skewness

After making necessary changes to all the columns and making continuous columns categorical, we one hot encoded the entire data frame to make it numerical for the model to fit.

Now the model was ready for Exploratory Data Analysis and algorithms to be fit with.

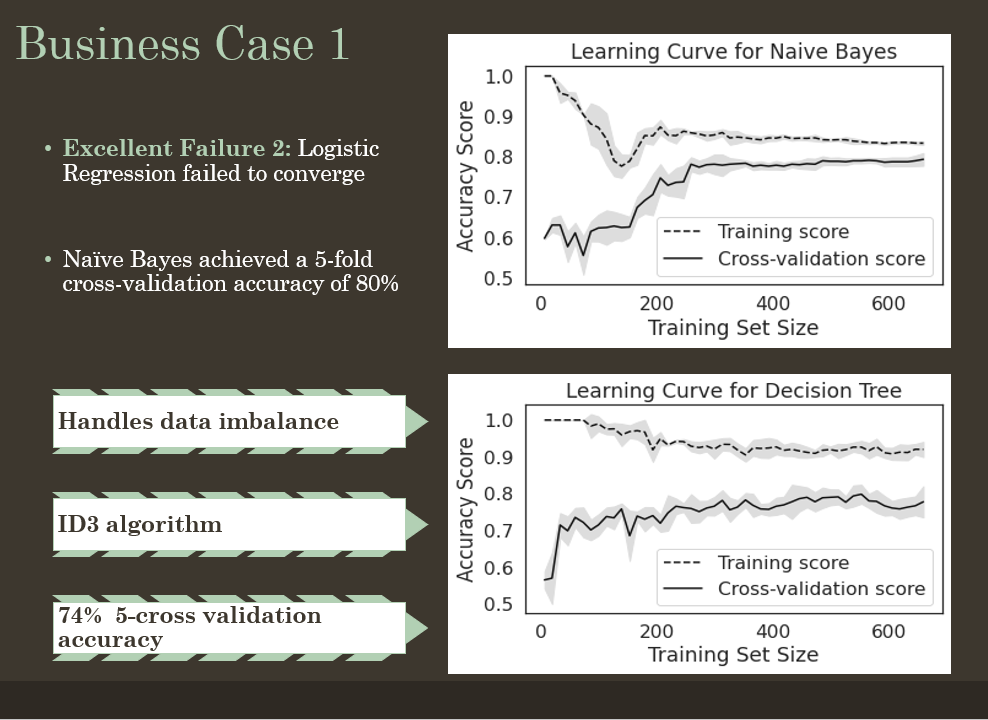


After the feature engineering process, the dataset was ready for Exploratory Data Analysis. While performing Exploratory Data Analysis on the target variable which is Acquisition Status that tells the status of a company with labels ‘Did not participate’, ‘was acquired’ and ‘made acquisition’, there was an observation that the target variable is imbalanced and skewed towards the ‘did not participate’ label.

This could result in an accuracy paradox problem which is the [paradoxical](https://en.wikipedia.org/wiki/Paradox) finding that [accuracy](https://en.wikipedia.org/wiki/Accuracy) is not a good metric for [predictive models](https://en.wikipedia.org/wiki/Predictive_model) when [classifying](https://en.wikipedia.org/wiki/Binary_classification) in [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics). This is because a simple model may have a high level of accuracy, but one class could be more dominant than the other classes. [Precision and recall](https://en.wikipedia.org/wiki/Precision_and_recall) are better measures in such cases but these too can be biased towards one class.

Sampling is one good solution to this problem. We performed ‘Down Sampling’ to handle class imbalance and took 1500 samples from class 0: did not participate class and kept the numbers the same from class 1: made acquisition and class 2: was acquired and class. When this data was modeled, a cross-validation accuracy of 76 % was observed which appeared to be quite good in a 3 – class problem.

We wanted to test this again on the newly acquired data to check whether it gives the similar type of results. The model gave a test accuracy of 78 % which is pretty good. Thus, we handled the accuracy paradox appropriately using down sampling with required results.



Our label was highly skewed as the count of companies that did not take part was over 7000 as opposed to those who were acquired and those who made acquisitions were near 1500. To overcome this problem, we tried bootstrapping.

Importantly, samples are constructed by drawing observations from a large data sample one at a time and returning them to the data sample after they have been chosen. This allows a given observation to be included in a given small sample more than once. This approach to sampling is called sampling with replacement.

After down sampling, the “Did not participate” label has 1500 companies, “Made Acquisition” label had 1558 companies and the “Was Acquired” label had 1332 companies which seemed fair as the skewness from the data was removed.

After the Data pre-processing and Feature engineering part, the next stage was the most fundamental part of our project. As our first question, we wanted to predict whether a company would make acquisitions, or will it be acquired, or it will simply not participate in the merger and acquisition process. This is a classic example of a multiclass classification problem.

We initially one hot encoded all the variables to make our features numerical

The data being categorical, we initially started with the baseline approach of fitting a Logistic Regression model on the sampled data. The model failed to converge. We made inferences that if the classes are well separated, the decision curve of Logistic Regression becomes unstable and hence, the model fails to converge. This was an Excellent Failure.

We then assumed conditional independence between our features and tried the Naïve Bayes algorithm. As this was not an ideal situation and there was some dependence between the features, the Naïve Bayes achieved 68% of 5-fold cross-validation accuracy.Though “Did not participate” and “Was Acquired” labels were predicted will high accuracy, the entire “Made Acquisition” class was predicted incorrectly. This was a shortcoming.

“Business Case 1 Data modeling slide 2”

After Naïve Bayes, we applied Decision Trees to overcome the problem of Conditional Dependence. The Decision Tree also performs well on imbalanced data. They work by learning a hierarchy of if/else questions and this can force both classes to be addressed. They use a layered splitting process, where at each layer they try to split the data into two or more groups so that data that fall into the same group are most similar to each other (homogeneity), and groups are as different as possible from each other (heterogeneity). Decision Trees apply a top-down approach

Here we used the ID3 algorithm which uses Entropy and Information Gain as metrics.

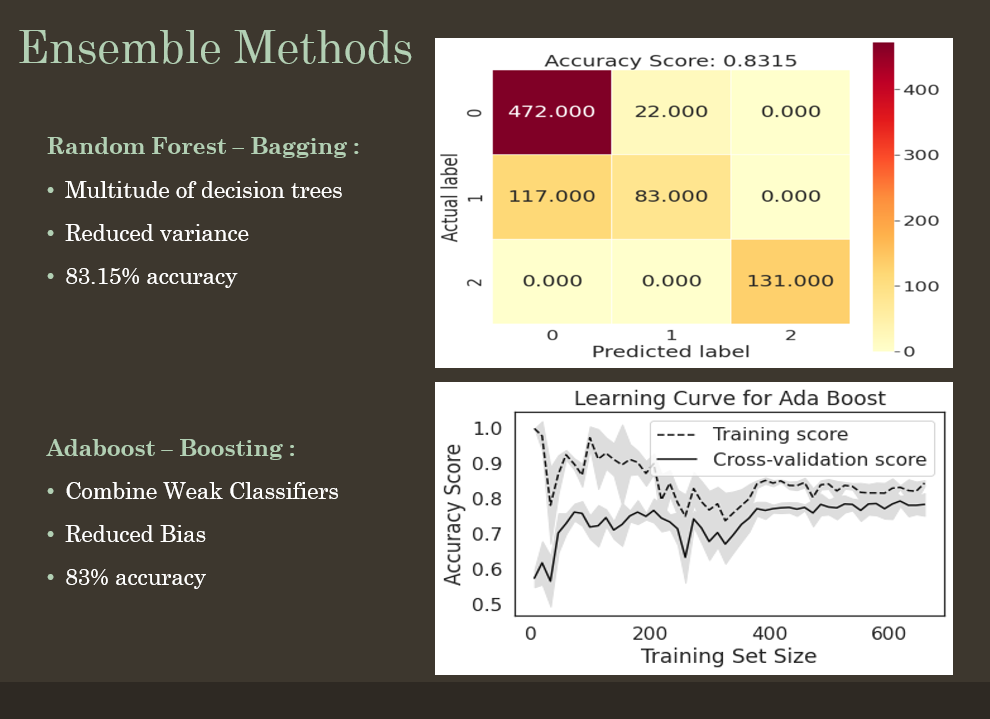
After using the Decision Tree, we observed 84% of 5 fold cross-validation accuracy which was a significant increase from what we observed in Naïve Bayes.

Out of all the classes we predicted, the “Was Acquired” class was predicted correctly all the time while “Made Acquisition” and “Did not participate” were predicted correctly 82% and 81% of the time respectively.

After plotting the ROC curve, we observed the micro and macro average area under the curve to be 0.88.After getting the accuracy metrics, we tested our model on unobserved data and it predicted the correct class with an accuracy of 77%.

True positives are data point classified as positive by the model that actually are positive (meaning they are correct), and false negatives are data points the model identifies as negative that actually are positive (incorrect). We now wanted to go beyond accuracy, hence, we calculated precision and recall score. We observed Average Precision Area to be 0.69

The training curve and the model results on completely unobserved data inferred that the model was not overfitting. The one problem that we observed in our data was also the high variance of a single estimate so we now wanted an approach to decrease the variance of that single estimate for higher stability.



After applying a tree-based model to the Data, we wanted to try other Ensemble methods in which multiple models are trained using the same learning algorithm.

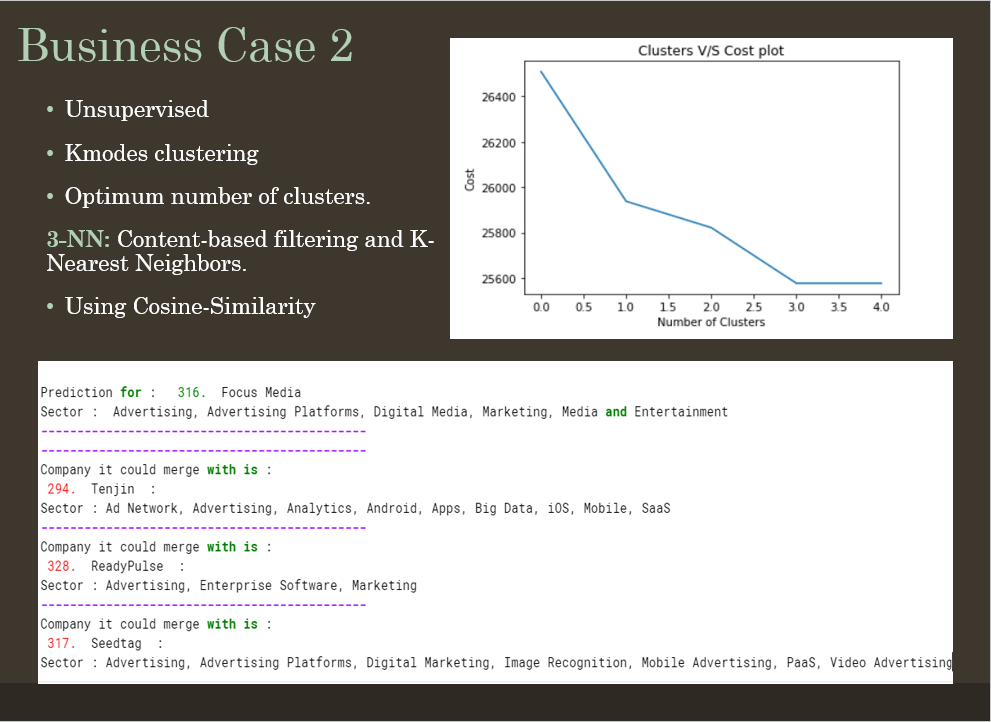
We tried our approach with bagging initially. The objective was to create several subsets of data from the training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees. As a result, we get an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

The primary objective of bagging is to reduce variance to avoid overfitting. The higher dimension data is handled very well in bagging and the accuracy of missing data is maintained.

The algorithm that we used was Random Forest. After applying the algorithm, we observed the 5 fold cross-validation accuracy of 82%. After plotting the confusion matrix, we observed that the class “Was Acquired” was predicted correctly all the time which was similar to the results in Decision Tree.

After bagging, we wanted to apply the Boosting method as the learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree. When an input is misclassified by a hypothesis, its weight is increased so that the next hypothesis is more likely to classify it correctly. This process converts weak learners into a better performing model.

The bias is reduced and the model becomes more generalized.We used Ada Boost for our data and got a 5 fold cross-validation accuracy to be 82% which was similar to the bagging algorithm we applied previously.We observed that Random Forest is immune to high variance and Ada Boost is prone to high variance and immune to high bias.



In business case 2 we wanted to predict the sector of the prospective company a company could merge to or acquire. We planned to do it by supervised machine learning approach when we devised the problem statement but while implementation after data preparation step, we came to the realization that we do have the labels in the sector as there are several sectors and no supervised machine learning algorithm would work in this case.

We decide to solve this problem using an unsupervised machine learning approach before declaring it as an excellent failure. This problem was a tricky one because we wanted to predict sectors to which a company could merge to but the directly keeping sector as a target label felt meaningless because all the features are tied together using company names. Thus, we decided to predict the company name and from that, the prospective sectors could be accessed.

We wanted to see the what all companies are tied in one cluster and how many clusters are there in our data-keeping company name as the target, we chose Kmodes clustering as our data were categorical and kmeans deals with numerical data and if we get the dummy variables of our data we get 296 features and complexity will increase and Kmodes directly deals with categorical data. Even after giving 8 iterations, the cost was becoming constant after 3 clusters formation and thus, we achieved the results that the optimum number of cluster formations will be 3. As a result, we can say if we want to classify a new company to which cluster would it belong, a company could be part of any one of these clusters.

Furthermore, we wanted to recommend a new company to get the sectors of that company. We decided to solve this problem using content-based recommendation systems that use the K-nearest neighbors’ algorithm that calculates cosine similarity to get the nearest companies for the target company. We chose to predict the top 3 companies based on the cosine distance. We observed that the sector of these recommended top 3 companies has some sectors which were similar and others which were different could be prospective sectors in which the target company could perform Merger and Acquisition

In 2019, Morgan Stanley and JP Morgan Chase saved over 360,000 hours of lawyer work by using AI for contract analysis. This was possible because of Natural Language Processing and other Artificial Intelligence techniques.

This is the future scope of our project.

After predicting the Mergers and Acquisitions, we recommended companies and sectors to a company for a successful merger. We now aim to build an NLP model to analyze the contracts and the paperwork they do when they merge.

From scanning through a contract to check for abusive clauses, to searching for specific information like a price, a date, or amending information in a set of contracts, these are highly time-consuming tasks. Many of these tasks can be automated by extracting specific contract elements.

One approach to build the machine learning model is to make use of a bidirectional LSTM (BILSTM) operating on words, part-of-speech (POS) tag, and token-shape embeddings.

In a deployed system, we would apply the extractor for each corresponding contract element type separately. As such, the extractors can focus on identifying a single element that would improve the system’s accuracy.

NLP’s ability to transform unstructured data into actionable, quantitative, structured data provides insights to value generation that are otherwise overlooked or require significant time and effort to uncover. Text-mining identifies facts and relationships that would otherwise remain buried in a mass of text faster and cheaper than humans could achieve. By providing visualizations of the relationships between key concepts in unstructured texts, a value can be extracted by understanding the relationships between disparate variables from different sources.