INTERDISCIPLINARY PROJECT (IDP) 2017

Risk scoring of investment assets: Development of an algorithm based classification system to allocate risks

(November 2016 – April 2017)

Author : Ashwath M Narayan , Rajat Koner , Saurabh Nawalgaria

Advisor : Sarah Theinert, M.Sc.

Technische Universität München

Contents

List of Figures

List of Tables

Abstract

The purpose of this IDP (Interdisciplinary Project) is to implement a risk scoring metric which can be used by investors or researchers to decide whether the company they want to invest in is risky or safe. This documentation will help reader to understand the design of the algorithm and step by step process to achieve desired results. Structure of this project will enable further use of it for modification or extension and also it can be reused to clean and project different datasets.

After suffucient literature study, we have chosen eight parameters such that their overlap is minimal and ensuring the weight of each parameter does not compound to every other parameter in the list. This ensures independent analysis of the data based on our parameters. The data acquired from the parameter are then plotted over numerous iterations by generating pseudo random points of the parameter from its range. Then, using unsupervised learning, the risk factor of the company is calculated for each parameter and aggregated over all parameters to get the risk score for a company.

Project Scope

The project dataset is restricted to the CDAX companies but can be reused by changing the mapping. With no change in datatype and data format, the data cleaning tool can be used for other datasets as well. The data cleaning is the most code intensive aspect in the project. The data of these companies are provided to the team as part of the IDP task. The application is split into 4 parts, as follows:

- **Data cleaning** : The dataset being used is cleaned in terms of removing outliers and NULL/missing values. Our data cleaning algorithms cleans only the data that are required to calculate the parameters. Other fields are neglected. The algorithm used here can be used to clean any of the fields in the data, but it works on partially sparse data. It can be reused considering the format of the dataset with minor modification.
- **Parameter calculation** : We calculate the parameters from the data set required for our project. The parameters used will be discussed in further sections.
- **Monte-Carlo Simulation** : The parameters are calculated for a yearly basis. Then, they are put to the K-Means algorithm which picks the centroids for the given dataset. The project uses 3 centroids as the project works on small, medium and large scale companies. Also, pseudo-random numbers are generated within the dataset range to create a bigger dataset to achieve more accurate clusters.
- **Risk factor** : The parameter data is passed through a clustering algorithm and then given a risk rating based on their performance with repect to their cluster.
- **Cumulative Risk Calculation** : All the parameters are used to calculate the overall risk of the company. Each parameter is given a weightage based on its importance and also based on the ambiguity in the data.

The project works using unsupervised learning as there is no prior classification of the dataset available. The use of mutually independent parameters help make strong predictions of the risk associated with a company as error in one parameter is not propagated to the other parameters.

The approach is gives a base classification availble for the CDAX companies which can be used by other machine learning algorithms to classify and predict risk scores. Since there is no prior classification, the results obtained are skewed and provide a pessimistic risk scoring for a company. This can be refined with algorithms which use the classification done with new data.

Dataset Used and Description of Data

The dataset used in this project is from the Thompson Reuters data stream. The data set contains the daily time stamp of the companies from January 2006 till December 2017. Along with time stamp of stock prices. The data set contains a series of parameters of the company for the fiscal years. Not all of the data is used for the risk scoring. The ones used are chosen based on the number of original values available and how useful those parameters are to provide independent barometers for the risk scoring.

Selection of parameters was the major part of the research work in the project. The parameters should have significant impact on the companies performance and simultaneously be independent of external influence. The parameter selection was further driven by the availability of data. Most of the parameters were not available consistently over a 6 year period to be useful. Many of the parameters such as the greeks were removed due to the inconsistent data. The parameters in the fiscal years are as follows:

- Normalized EBITDA
- Normalized EBIT
- Normalized Income Before Taxes
- Normalized Income After Taxes
- Total Revenue
- Total Debt
- Accounts Receivable - Trade, Net
- Accounts Payable
- Total Inventory
- Cost Of Goods Sold – Actual
- Enterprise Value (Daily Time Series)
- Net Debt – Mean
- Total Debt
- Total Current Assets
- Total Current Liabilities
- Total Liabilities
- Total Equity
- Common Stock, Total
- Shares Out - Common Stock Primary Issue

- Cash and Equivalents
- Number of Employees

The dataset contains the above fields with certain outliers and NULL values, these are corrected with the data cleaning. Since there is no prior information about the limits of the values the parameters can take, removing outliers is based on the intrinsic of the data. Range, population of data and average values and extremities.
Certain aspects of the data such as the inherent floating precision error, information error is unknown. The data is accepted as is assuming that the values are accurate.
The dataset focuses on the CDAX companies. We use only this dataset for testing and results as we are restricted in dataset availability and classification.

Initial Parameters Used

We have selected 8 parameters to rate a company. The 8 parameters were chosen based on some criteria that the parameters should be independent of each other. This is to ensure that they weighing factor used later for the risk scoring does not get affected by using dependent parameters. The 8 parameters are:

- Volatility : Volatility is used to detect the rate of change of the share prices with time. The reason for choosing this parameter is because it shows the nature of the company. It is a useful barometer for long term investment. The equations and the method used to calculate volatility is mentioned in the Appendix

- Value at Risk : Value at Risk is a standard risk measure in the financial sector. A VAR statistic has three components. A time period, Confidence level and the loss amount. In this project, we have considered 95 % Confidence level of the rate of return to retrieve VAR. We have considered yearly investment time period of our calculations of VAR. We compute the daily rate of return of stock from 2006 to 2016. This daily return is our dataset to compute VAR. We Compute the VAR annually taking annual data at a time.

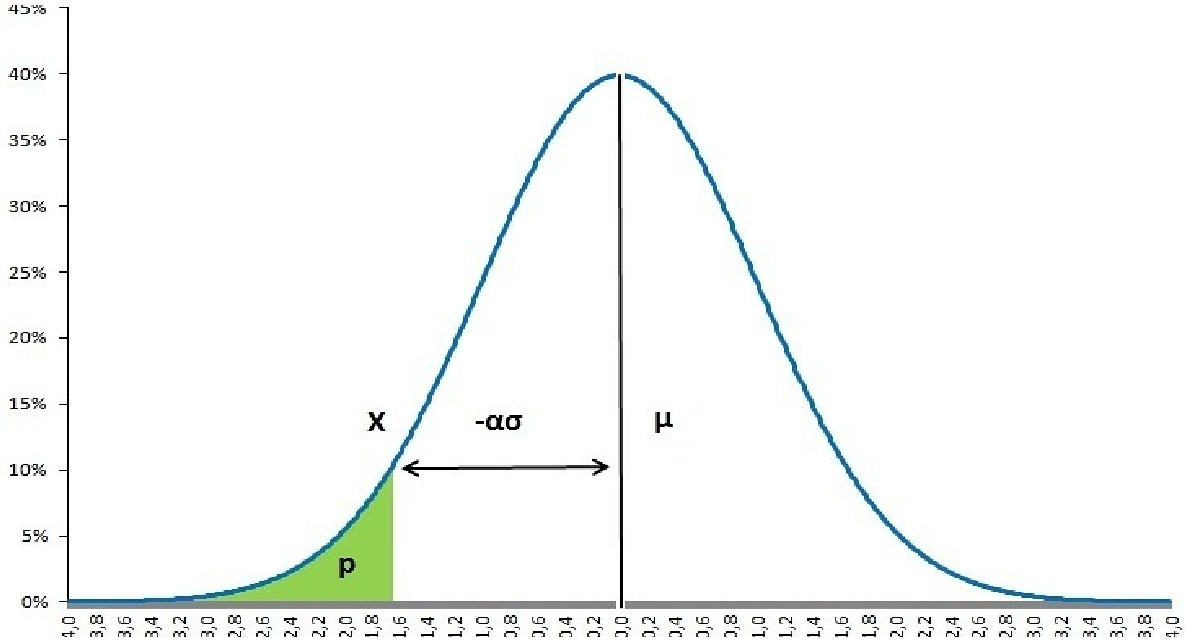


Figure 1: Calculating VAR

- Beta : a beta less than 1 indicates that the investment is less volatile than the market, while a beta more than 1 indicates that the investment is more volatile than the market.

$$Beta = \frac{Covariance(r_i, r_m)}{Variance(r_m)}$$

- Profitability : The profitability of a company is the ratio of the net income vs the net sales. The higher the profit margin, the better the company is using its resources efficiently to make as much profit from the existing resources.

- Return on Equity(Removed for risk scoring) : The ROE is the ratio of the net income to the average equity. The higher the ROE, the more money the company is making during that period. This shows how much the company has as income which it can use immediately for any investments/payments.

$$ROE = \frac{Net\ Income}{Average\ Equity}$$

- Gearing : The gearing ratio is the Total Debt to the Total Equity. This ratio is used to gauge the debt levels of the company and the lower the value, the better the company performance.

$$Gearing = \frac{Total\ Debt}{Total\ Equity}$$

- EBITDA : The EBITDA is the Earnings Before Interest Taxes Depreciation and Amortization. This gives the income the company makes during a calendar year before any of the tax interest etc. are applied. Higher the EBITDA, the better the company is performing.

- Return on Capital Employed : The ROCE is a parameter used to determine how well a company is investing and whether the investments the company makes returns a profit. The ROCE is the ratio of the EBIT to the average long term debt and the average equity. This gives an insight as to how the company is going to perform in the future based on the investments they make now.

$$ROCE = \frac{EBIT}{Avg.Long\ Term\ Debt + Avg.\ Equity}$$

Data cleaning for outliers and missing values

Data cleaning and outlier detection are perhaps the most important step for any data mining and data analytics project. In this project we are dealing with mainly two kind of data(aka: DataStream) one for daily statics of all listed equity for consecutive 11 years and second one is yearly metrics(like. EBITDA,Total Revenue) of listed equity for consecutive 6 years. As like other data analytics project this raw data also has various flaws which cant be used for any simulation algorithm. Some of the irregularities are listed below :

1) The most common type NULL value.

2) String literal in case of numeric value.

3) Cell contains corruptaed macro.

4) Cell contains outlier value.

We have to apply different data cleaning algorithms for these two types of data. For 1st one of daily statics null value or missing value can be replaced by the average value of earlier and next available data. But for detection of outlier we have checked if the if certain day closing value(as we need this value for most of out future calculation) varies to more than 60 %.

For the second sheet filling up the missing data is quite complicated. As different metrics dependency among each other is very different in nature, so filling up with average value is not so appropriate. For example we have Normalized EBITDA, Normalized EBIT, Normalized Income Before Taxes and Total Revenue are very closely related and their dependency are from left to right. So the basic algorithm works like below with an example

- Create list of column which are closely associated, and order them from left to right.

- If some value missing in any column(like EBITDA), then we need to check the next non empty consecutive column on same row scanning from left to right.

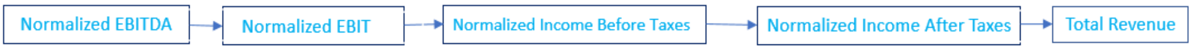


Figure 2: Association sequence of financial attribute

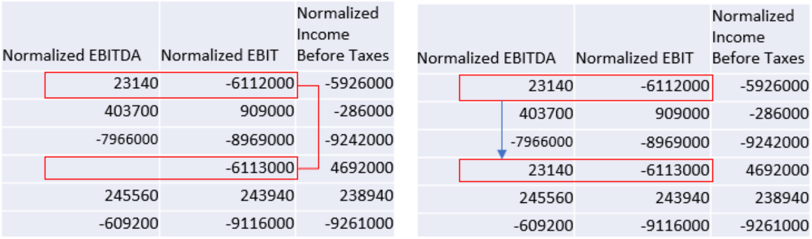
- If any value found in that row, then search for the closest value on that column and return the row number.

- Check if there is any value on that missing column in that returned row number. If any value found then replace the missing value with this value.

- If there is no value in the return row of that missing column, then increase the column number and go to step 2.

- After we replaced all the value till the last column, the loop need to be reversed, as there may be some more missing value till left .

Here is a simple example, in below table 4th row of the 1st column has a missing value, and next element on same row is not null, so the algorithm finds its closest element in in 1st row, so missing Normalized EBITDA value of 4th row will be replaced by 1st row value.



Normalized EBITDA	Normalized EBIT	Normalized Income Before Taxes
23140	-6112000	-5926000
403700	909000	-286000
-7966000	-8969000	-9242000
	-6113000	4692000
245560	243940	238940
-609200	-9116000	-9261000

Normalized EBITDA	Normalized EBIT	Normalized Income Before Taxes
23140	-6112000	-5926000
403700	909000	-286000
-7966000	-8969000	-9242000
23140	-6113000	4692000
245560	243940	238940
-609200	-9116000	-9261000

Figure 3: NULL value replacement by finding nearest value of closest related attribute

Implementation of Monte Carlo

Definitions

Pseudo random numbers: Pseudo random numbers are defined as A pseudorandom number generator (PRNG), also known as a deterministic random bit generator (DRBG), is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by a relatively small set of initial values, called the PRNG's seed (which may include truly random values).

Clustering algorithm: A clustering algorithm is defined as Clustering is the process of partitioning a set of heterogeneous (different) objects into subsets of homogeneous (similar) objects. At the heart of cluster analysis is the assumption that given any two objects you can quantify the similarity or dissimilarity between those objects. In continuous search spaces distance measures similarity.

K-Means: k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

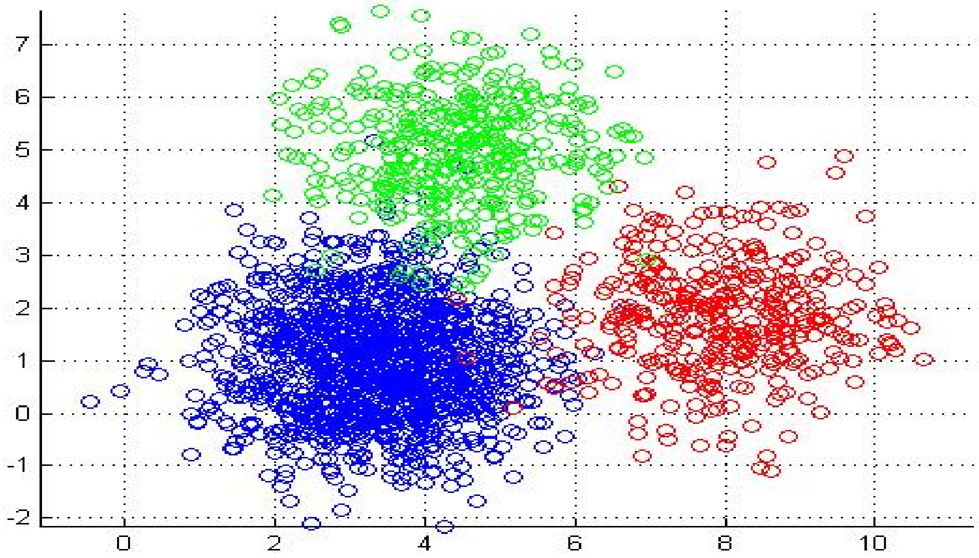


Figure 4: Kmeans clustering

Monte Carlo Finance: The Monte Carlo method encompasses any technique of statistical sampling employed to approximate solutions to quantitative problems. Essentially, the Monte Carlo method solves a problem by directly simulating the underlying (physical) process and then calculating the (average) result of the process. This very general approach is valid in areas such as physics, chemistry, computer science etc.

In finance, the Monte Carlo method is used to simulate the various sources of uncertainty that affect the value of the instrument, portfolio or investment in question, and to then calculate a representative value given these possible values of the underlying inputs. ("Covering all conceivable real world contingencies in proportion to their likelihood.") In terms of financial theory, this, essentially, is an application of risk neutral valuation. This very general approach is valid in areas such as physics, chemistry, computer science etc.

In finance, the Monte Carlo method is used to simulate the various sources of uncertainty that affect the value of the instrument, portfolio or investment in question, and to then calculate a representative value given these possible values of the underlying inputs.) In terms of financial theory, this, essentially, is an application of risk neutral valuation.

Monte Carlo Approach:

We have applied Monte Carlo approach to compute the risk scores of the above Companies. We adopt the K Means approach. K Means is a clustering algorithm.

We divide the Companies into 3 Categories/clusters based on their revenue. Large Scale, Medium Scale and Small Scale Industries. Using Python's kmeans package, we generate these 3 clusters from our dataset. We apply the formulae to each cluster to get the above mentioned 8 parameters. We perform the risk scoring of the companies in each cluster independently. For each parameter, we consider its minimum and maximum value for that cluster, then we generate pseudo random numbers between the minimum and maximum values.

This is done to enlarge our dataset so that we achieve more accurate scoring results. This is basically the Monte Carlo Approach applied by us to perform the risk ratings. After we have performed the risk ratings individually for all the 8 parameters, the next step is to incorporate weighing factor for each parameter. We find the weighted sum of the risk ratings taking all the parameters into consideration and then provide a risk rating for the company.

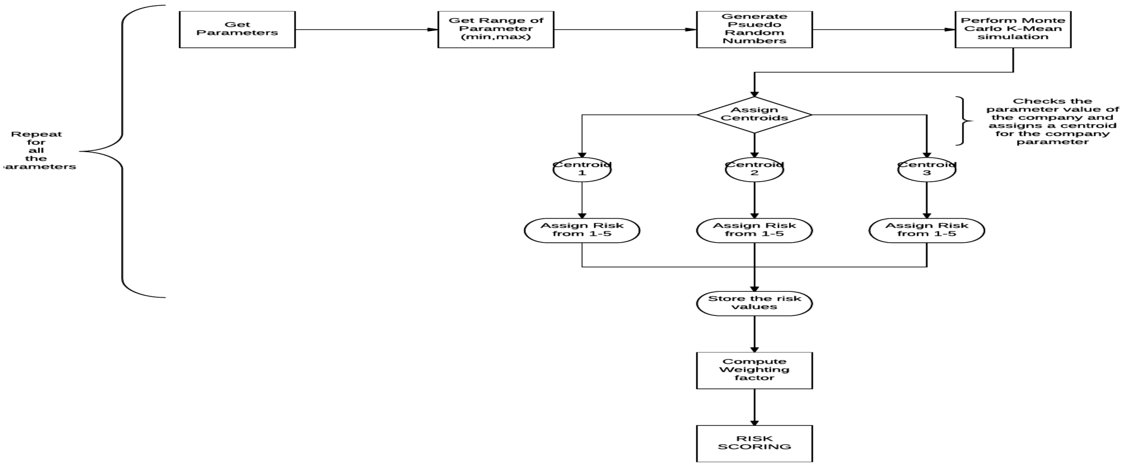

Get Parameters
Get Range of Parameter (min,max)
Generate Psuedo Random Numbers
Perform Monte Carlo K-Mean simulation
Assign Centroids
Checks the parameter value of the company and assigns a centroid for the company parameter
Repeat for all the -arameters
entroid 1
entroid 2
entroid 3
Assign Risk from 1-5
Assign Risk from 1-5
Assign Risk from 1-5
Store the risk values
Compute Weighting factor
RISK SCORING

Figure 5: Basic overview of risk scoring procedure

Get Parameters:

We use the datasheet containing the parameters as input. Using the XLRD and XLWT packages from Python, the data is read and stored as arrays that can be used for computation. The package was chosen primarily for the ease of use and that the parameter datatype can be type-casted at input.

Get Range of Parameters:

Once the parameters have been read from the dataset. There is a buffer added to the minimum and maximum value of the data to accommodate for outliers and to improve the confidence of border cases. This buffer is adding 6.25 % more to the minimum and maximum data points. This buffering also allows deploying other datasets which have different maximum and minimum values

Generate Pseudo Random Numbers:

We require pseudo random numbers (PRN) to populate the dataset required for Monte Carlo simulation. For this, we use the Python Random package and generate 10000 PRN's chosen normally from the range of min and max from the dataset. This increases the size of the dataset for the Monte Carlo from 425 points to 10425 data points.

Perform Monte Carlo K Mean Simulation:

With the dataset sufficiently populated. The Monte Carlo simulation can be performed. For this, we use the K Means method and use the dataset to get 3 clusters. The choice for having 3 clusters is to separate the datasets into 3 parts and returns the clustering analysis accurately for the points. We resorted to using just 3 centroids because we wanted to avoid overfitting the data and also since we didn't have any priors, calculating appropriate centroid and classifying the results would not be possible.

Assign Risk 1-5:

Once we have the classification from the kmeans. We use the classification and the centroid to mark the distance of the company from the centroid. This range is then split into 5 parts. Based on where a company falls on the splits we provide a risk rating between 1-5 for the parameter associated with the company.

Compute Weighting Factor:

The weighting factor provides a scaling for the overall parameter set to get the final risk rating for a company. To do this, we use the number of duplicates as the barometer.
The higher the duplicates in the dataset (which occurs due to Data Cleaning and removing NULL values) reduces the impact of the parameter on the overall risk. So, we perform an inverse scaling. This provides a high weighing factor for parameters with the least duplicates.

Risk Scoring:

Once we have the weighing factor for each parameter for a given fiscal year. We sum up the risk rating for the entire dataset and average the risk to get the definitive answer. This dual iteration with respect to weighing and kmeans approach makes sure that any outlier in the dataset does not affect our results.
The results are stored in an excel file for viewing.

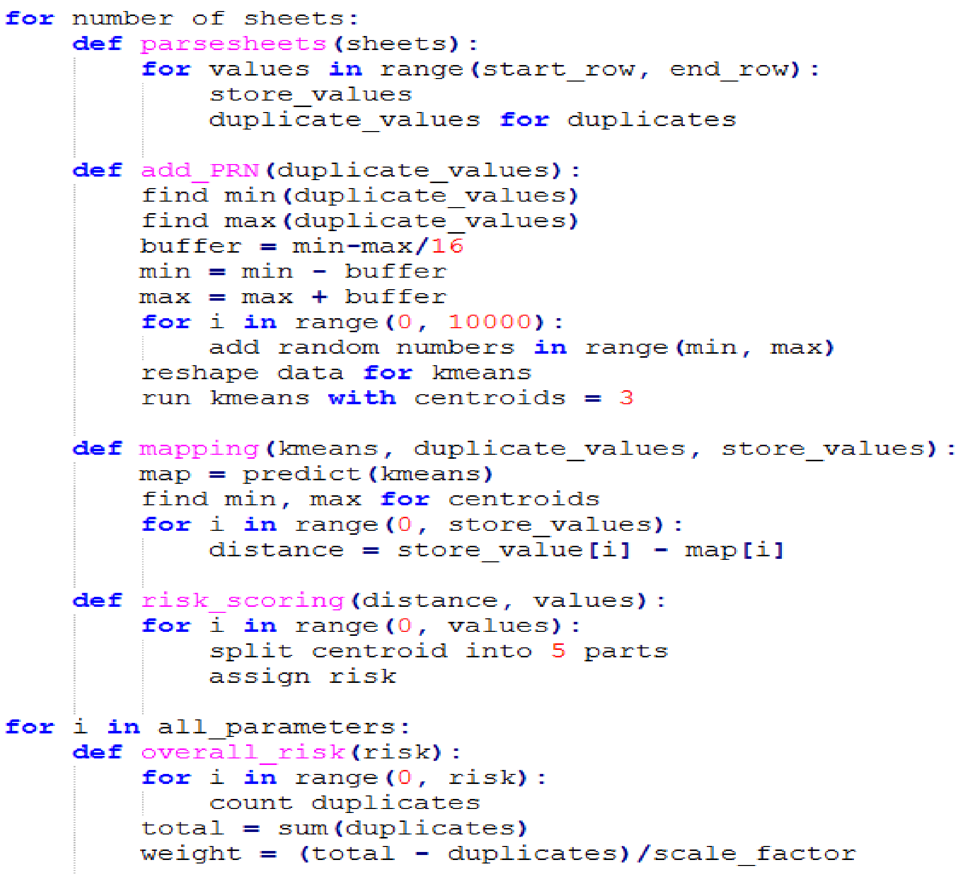
Design Choices and Algorithm:

Based on the confidence of the data received and the ratio of available data and duplicated values, a neural network based supervised learning was not possible as there was no prior classification of the data was available.

Hence an unsupervised learning approach was taken to classify the parameters independently. The choice for using Monte Carlo was decided because the dataset was sparse and required random inputs to populate the dataset to perform a normal distribution to perform a confident clustering of the parameters.

The unsupervised learning approach chosen was the kmeans clustering algorithm. The kmeans uses the data points and calculates the distance between the points to find the centroids of a cluster of points. This approach directly suits the dataset and is exactly what the scope of the project needed.

The following pseudocode provides an explanation of the risk scoring.



```
for number of sheets:
    def parsesheets(sheets):
        for values in range(start_row, end_row):
            store_values
            duplicate_values for duplicates

    def add_PRN(duplicate_values):
        find min(duplicate_values)
        find max(duplicate_values)
        buffer = min-max/16
        min = min - buffer
        max = max + buffer
        for i in range(0, 10000):
            add random numbers in range(min, max)
        reshape data for kmeans
        run kmeans with centroids = 3

    def mapping(kmeans, duplicate_values, store_values):
        map = predict(kmeans)
        find min, max for centroids
        for i in range(0, store_values):
            distance = store_value[i] - map[i]

    def risk_scoring(distance, values):
        for i in range(0, values):
            split centroid into 5 parts
            assign risk

for i in all_parameters:
    def overall_risk(risk):
        for i in range(0, risk):
            count duplicates
        total = sum(duplicates)
        weight = (total - duplicates)/scale_factor
```

Test results

To generate a sufficient sample space to run the monte-carlo simulation, we created a sample data distribution for each of the parameter based on the range of values we receive from the data set. We chose 10,000 additional pseudo random samples and fitted these samples around the data. This is required to make the dataset sufficiently large and random for the Monte-Carlo simulation to return an unbiased result.

The test results were accurate in the test data pointing out expected market leading companies such as BMW, Allianz as stable companies. The results also gave expected results with respect to small and medium scale companies such as start-ups which are just entering the market.

We gave a high penalty of 5 for every company whose data was not available or we were not able to calculate. This is because we had to accommodate uncertainty of not having the parameters due to lack of data.

Without classification of the data and using just a dataset, the Monte-Carlo algorithm with K-Means classified the companies and accommodated for outliers after adding pseudo random numbers.

Since there was no prior data provided to train a network for the weighting parameters. The number of duplicates in the parameters were used as a barometer to get the confidence and the uniqueness of the data set. This was used to get the weighting factor for the parameters to calculate the risk for a company.

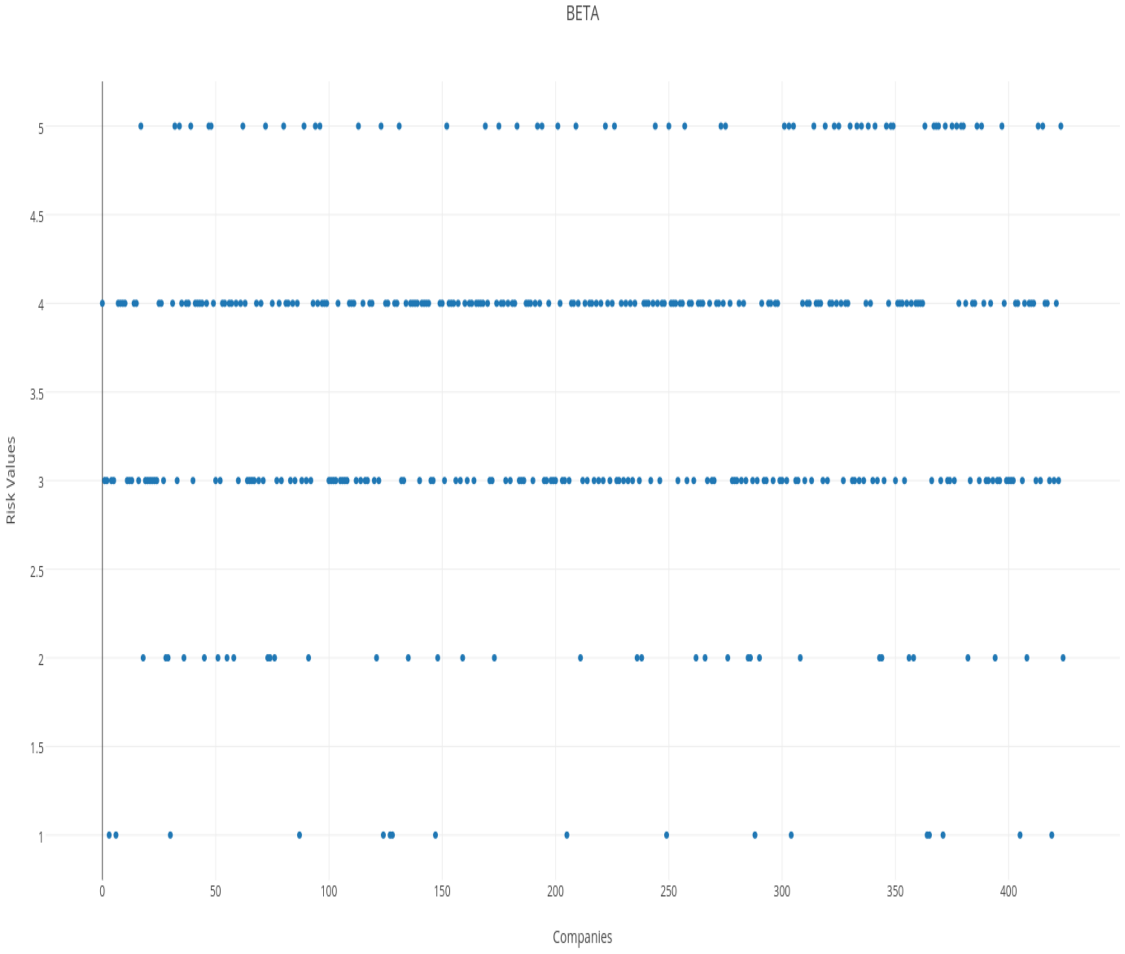
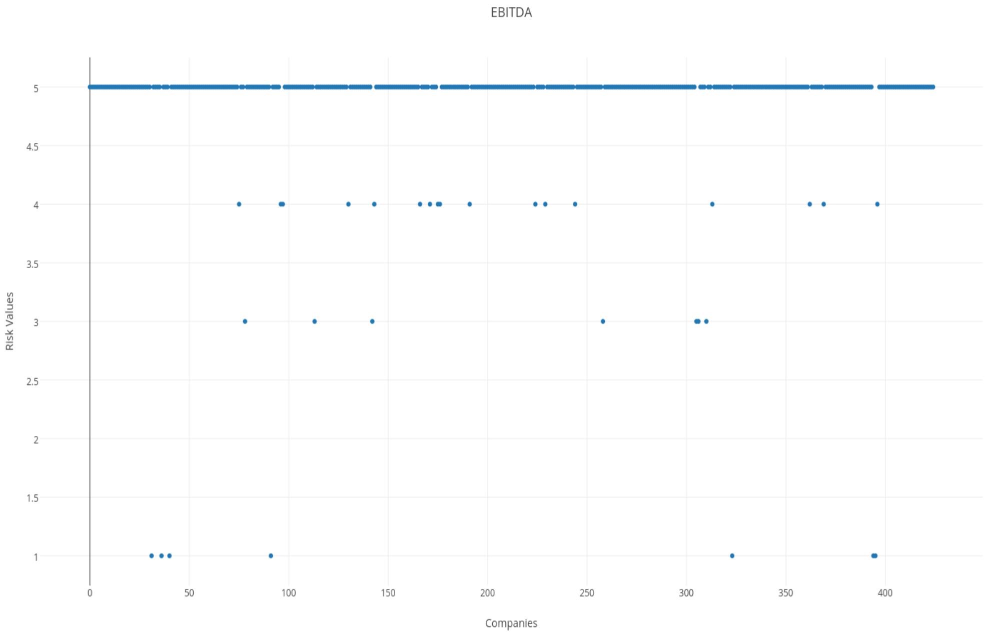
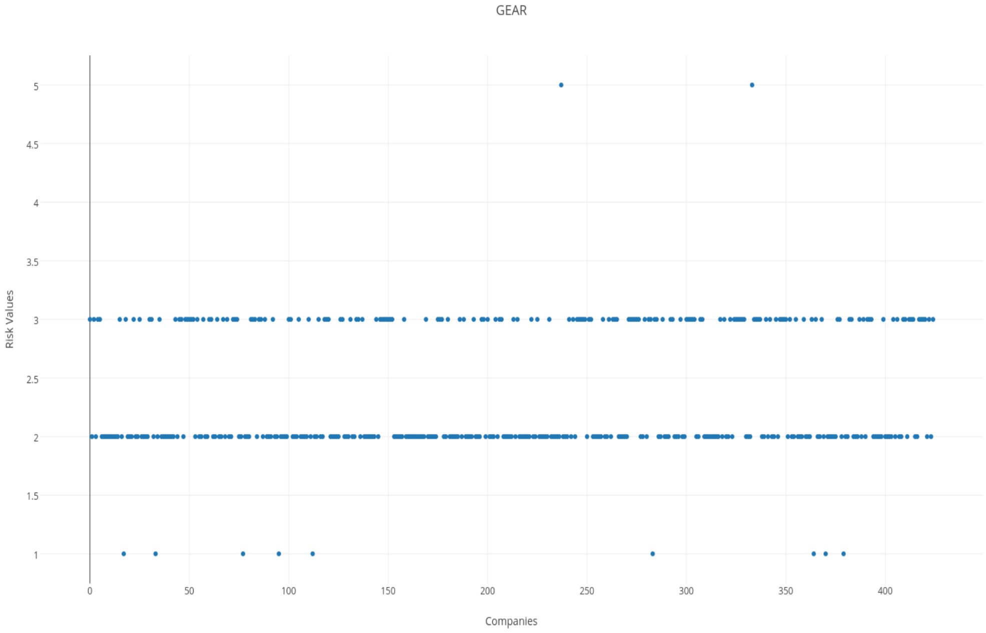
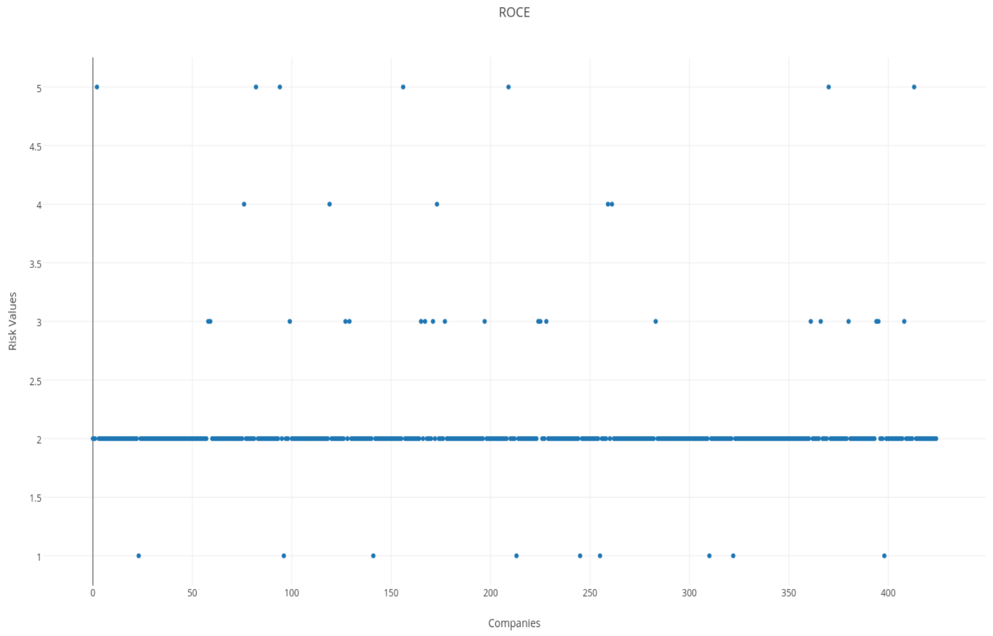
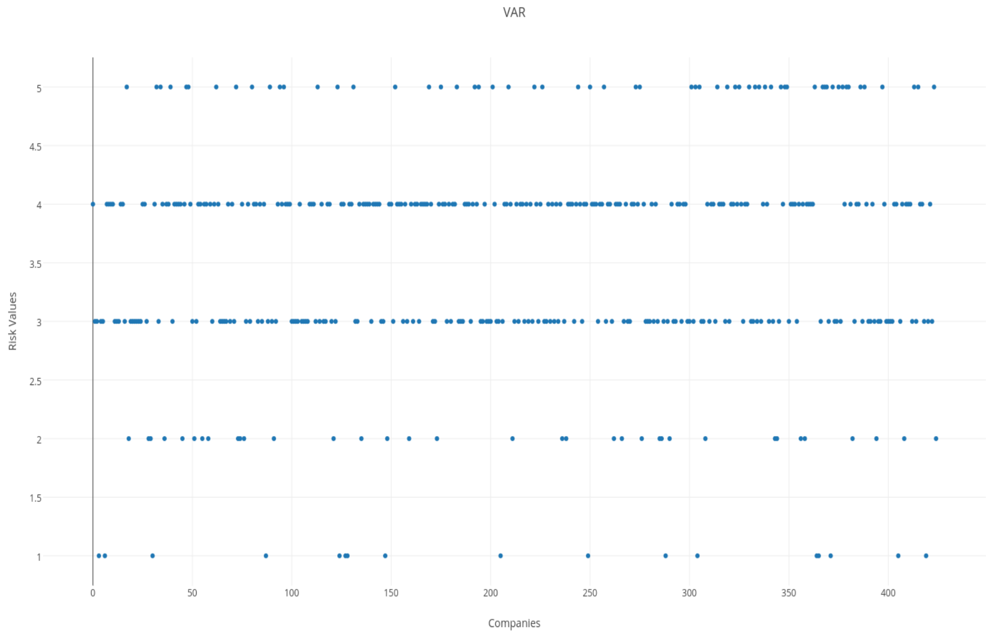


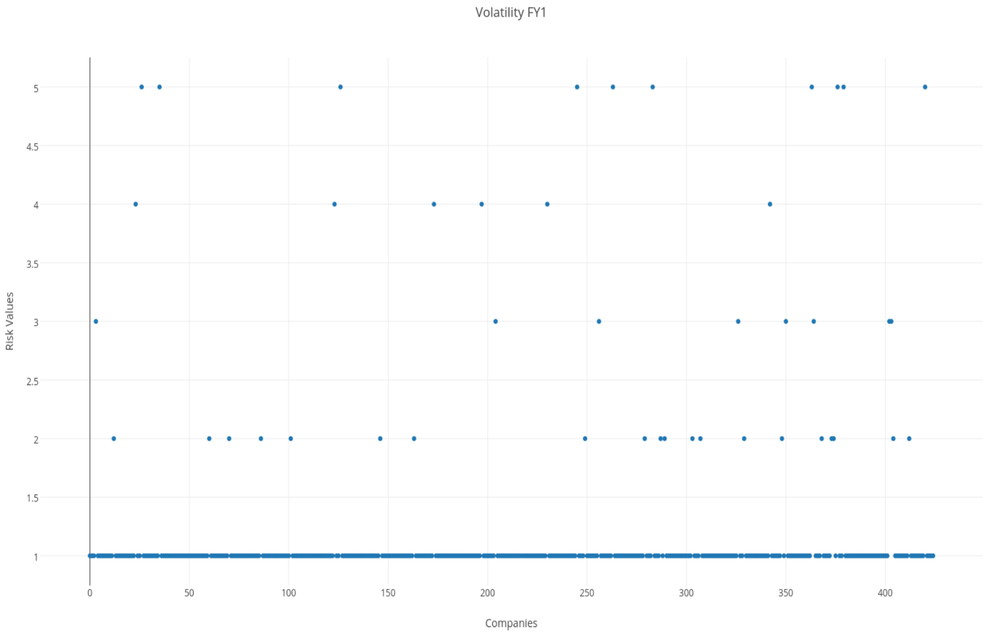
Figure 6: Association sequence of financial attribute


EBITDA
Risk Values
5
4.5
4
3.5
3
2.5
2
1.5
1
0
50
100
150
200
250
300
350
400
Companies


GEAR
Risk Values
5
4.5
4
3.5
3
2.5
2
1.5
1
0
50
100
150
200
250
300
350
400
Companies


ROCE
Risk Values
5
4.5
4
3.5
3
2.5
2
1.5
1
0
50
100
150
200
250
300
350
400
Companies


VAR
Risk Values
5
4.5
4
3.5
3
2.5
2
1.5
1
0
50
100
150
200
250
300
350
400
Companies

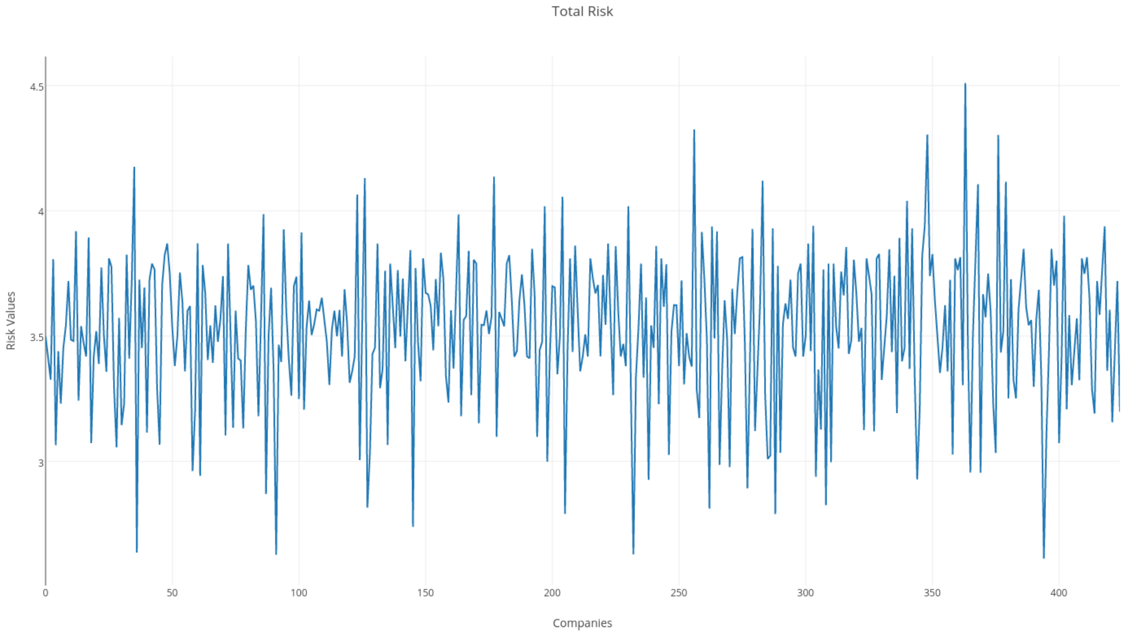

Volatility FY1
Risk Values
5
4.5
4
3.5
3
2.5
2
1.5
1
0
50
100
150
200
250
300
350
400
Companies

Findings

The parameters calculations were accurate to the dataset given. The formulae for the parameters were verified. Since the dataset has no training dataset or a benchmark to compare the results. Manual calculations were made to accommodate as priors to verify the accuracy of the parameters.

We found that the risk scoring of the companies was accurate based on the parameters we had selected and also pointed out to the performance of a company based on its classification on the kmeans. Since the parameter data was classified independent of each other, the results are not convoluted by relationships between the parameters. The risk scoring also verified that the companies in the CDAX are generally less volatile and the value at risk for most companies were in the no risk sections.

Regarding the fiscal year parameters used, i.e. EBITDA, ROCE and Gearing. It was noted that the companies performed in a more random fashion due to the underlying variables required to compute the parameter values. The weighting of the parameters based on the duplicates helped reduce the impact of a parameter with many duplicates.



Since the above results is on the safer side. Any Company below 3.6 is a good bet. This high number is due to the weighting factor giving EBITDA a high weight. We had to use EBITDA as a parameter because it is the most stable and least duplicated aspect of the data set provided.

As can be observed from the figure, the minimum value of risk was set at 2.6, this is because of the uncertainty in the data, duplicates, lack of classification. The results obtained are pessimistic values of the dataset, scaling the values obtained again will produce ambiguity as risky companies will be visualized as not risk due to scaling. Hence, it is avoided.

Conclusion

We could conclude that it is possible to classify CDAX Companies on the risk metric, Through our results and findings , It is evident that our obtained results complies with the expected findings. We were able to deploy unsupervised learning in the form of Kmeans Clustering and parameter weighing to assign risk scores to the financial firms.

Using the nearest neighbour approach to clean the dataset enabled us to handle null values. Parameters having most no of duplicates were given less weightage accordingly.

Appendix

1) https://www.value-at-risk.net/

2) https://en.wikipedia.org/wiki/Capital _asset _pricing _model

3) http://web.stanford.edu/class/msande444/2012/MS &E444 _2012 _Group2b.pdf

4) https://en.wikipedia.org/wiki/Arbitrage _pricing _theory

5) https://en.wikipedia.org/wiki/Multiple _factor _models

6) http://www.kellogg.northwestern.edu/faculty/papanikolaou/htm/finc460/ln/lecture6.pdf

7) http://people.stern.nyu.edu/ashapiro/courses/B01.231103/FFL09.pdf

8) http://www.kellogg.northwestern.edu/faculty/papanikolaou/htm/finc460/ln/lecture6.pdf

9) http://www.ingentaconnect.com/content/aea/jep/2004/00000018/00000003/art00002

10) http://web.abo.fi/fak/mnf/mate/tammerfors08/embrechts _tuesday.pdf

11) http://www.sciencedirect.com/science/article/pii/S0927539800000116

12) http://www.sciencedirect.com/science/article/pii/S0927539800000220

13) http://www.cfapubs.org/doi/abs/10.2469/faj.v56.n2.2343

14) http://link.springer.com/chapter/10.1007/978-3-540-71297-8 _33

15) http://onlinelibrary.wiley.com/doi/10.1111/1540-6261.00455/full

16) https://en.wikipedia.org/wiki/Monte _Carlo _methods _in _finance