



# Portfolio Risk Management using Big Data Technologies

DS-503  
Big Data Management

**Authors:**

Alexander Shoop  
Krushika Tapedia  
Mihin Sumaria  
Mukund Khandelwal

**Supervisor:**

Prof. Jitendra Singh

April 17, 2018

<b>1. Project Summary</b>	3
<b>2. Project Description</b>	3
What is Financial Risk Management?	3
<b>3. Background</b>	3
3.1 Financial features used in the data set	3
3.2 Annual Reports for Sentiment Analysis	4
3.3 Metrics being calculated	4
<b>4. Broader Impact and Intellectual Merit</b>	4
<b>5. Framework for investigation</b>	4
<b>6. Project Architecture</b>	5
6.1 For Text Mining & Sentiment Analysis	5
6.2 For Monte Carlo Simulation	5
6.3 For Twitter Feed Analysis	6
<b>7. Implementation</b>	6
7.1 Sentiment Analysis on 10-K Financial Reports	6
7.2 Calculation of Value at Risk (VaR) using Monte Carlo Simulations	8
7.2.1 Monte Carlo Simulation	8
7.2.2 The Data	9
7.2.3 Parallelizing the code	9
7.3 Twitter Feed Analysis	11
7.4 Project Application	13
<b>8. Justification</b>	13
<b>9. Technical Feasibility</b>	14
9.1 Sentiment Analysis of 10-K Reports	14
9.2 Monte Carlo	14
9.3 Twitter Feed Analysis	14
<b>10. Conclusion</b>	14

## 1. Project Summary

The proposed project, 'Portfolio Risk Management' aims to assist an investor in portfolio management by leveraging big data technologies to handle the sheer volume of data and the high computational costs required to process said data. When investing in the financial market, what matters is how an investor can achieve maximum returns with the lowest possible risk. Risk here means any kind of uncertainty which could hurt the investors, or make them lose an opportunity. There are various risk metrics that can be employed to provide a quantitative estimate of the variation of expected outcomes, and thus help mitigate risks in a given portfolio.

The metric that is of particular interest to us in order to assess the risk for a given portfolio is called Value at Risk (VaR). VaR calculations are complex, and require simulations that have high memory and processing requirements. Competent investors would also like to keep track of metrics associated to live social media, and reports released by the U.S. Securities and Exchange Commission (SEC) indicating the annual performance of relevant companies. The problem that arises when considering all of these factors, is that the volume and traffic of incoming data is huge, and it is difficult to track all of it simultaneously.

We propose a system, which provides a dashboard to the user, i.e. the investor in this case, which allows them to see all of these metrics in one place. The dashboard would display the following components:

1. Sentiment of the annual 10-K reports released by the SEC, indicating the overall annual performance of the relevant companies.
2. Monte-Carlo simulations to calculate the VaR of the portfolio.
3. Live Twitter text analysis of tweets associated to all the relevant companies.

The system makes use of Apache Spark to perform simulations to calculate Value at Risk, Apache Spark-Streaming to get the tweets for textual analysis, and Amazon Elastic Cloud Compute (EC2) to store the 10-K reports onto Amazon S3, and perform sentiment analysis on these reports.

## 2. Project Description

### What is Financial Risk Management?

In the financial world, risk management is the process of identification, analysis and acceptance or mitigation of uncertainty in investment decisions. Essentially, risk management occurs any time an investor or fund manager analyzes and attempts to quantify the potential for losses in an investment and then takes the appropriate action (or inaction) given his investment objectives and risk tolerance.<sup>1</sup>

## 3. Background

In order to calculate the risk metrics, we should have a basic understanding of the following:

### 3.1 Financial features used in the data set

- **Date/Timestamp:** Date and timestamp during which the open, high and close value of a particular stock are recorded.
- **Open:** 'Opening Price' is the price at which a security first trades upon the opening of an exchange on a given trading day.
- **Close:** 'Closing Price' is the price of the final trade before the close of the trading session.
- **High:** It is the highest price of the stock during the day of trading.

---

<sup>1</sup> Investopedia, Risk Management: <https://www.investopedia.com/terms/r/riskmanagement.asp>

- **Adj Close:** An ‘Adjusted Closing Price’ is a stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions (eg. splits and dividends) that occurred at any time prior to the next day's open. The adjusted closing price is often used when examining historical returns or performing a detailed analysis on historical returns.
- **Volume:** Volume is the number of shares or contracts traded in a security or an entire market during a given period of time. That is, when buyers and sellers agree to make a transaction at a certain price, it is considered one transaction. If only five transactions occur in a day, the volume for the day is five.

### 3.2 Annual Reports for Sentiment Analysis

“10-K” financial reports are annual company performance reports submitted to the US Securities and Exchange Commission (SEC). Companies operating in the US are legally obligated to submit these annual reports. As defined by the SEC, they provide a “comprehensive review of [a] company's business and financial condition and includes audited financial statements.”<sup>2</sup>

### 3.3 Metrics being calculated

- **Value at Risk (VaR):**  
Value at risk (VaR) is a statistical technique used to measure and quantify the level of financial risk within a firm or investment portfolio over a specific time frame.<sup>3</sup>
- **Daily Returns:**  
A return, also known as a financial return, in its simplest terms, is the money made or lost on an investment. Using daily returns, we compute the daily mean returns and volatility.
- **Sentiment Polarity:**  
Sentiment polarity analysis takes into account the amount of positive or negative terms that appear in a given sentence. It is useful to understand whether a given sentence, paragraph, or a document is positive or negative overall. The polarity score can range from [-1, 1], where -1 is completely negative, and 1 is completely positive.

## 4. Broader Impact and Intellectual Merit

Risk analysis is part of every decision we make when faced with uncertainty, ambiguity, and variability. Indeed, even though we have unprecedented access to information, we can't accurately predict the future. In finance, there is a fair amount of uncertainty and risk involved with estimating the future value of financial products, due to the wide variety of potential outcomes. In the field of investing on the financial market, what matters is how an investor can achieve maximum returns with lowest possible risk. Therefore, we want to derive risk metrics than can help us determine risk involved in investing in the stock market.

## 5. Framework for investigation

We chose Amazon Web Services (AWS) as our primary framework to implement risk analysis. Our preliminary research discovered that Google Cloud Platform (GCP) also has many similar capabilities

---

<sup>2</sup> SEC Fast Answers, 10-K: SEC Fast Answers, 10-K:

<https://www.sec.gov/fast-answers/answers-form10khtm.html>

<sup>3</sup> Value at Risk (VaR), Investopedia <https://www.investopedia.com/terms/v/var.asp>

and features to AWS; for this project we will focus on AWS. The following are the components we choose for our implementation and also a brief summary of what made us choose this setting.

- AWS S3 for Storage
- AWS EMR Cluster framework for handling computation
- PySpark for computation in Zeppelin notebook
- AWS EC2 for handling sentiment analysis via TextBlob package

## 6. Project Architecture

### 6.1 For Text Mining & Sentiment Analysis

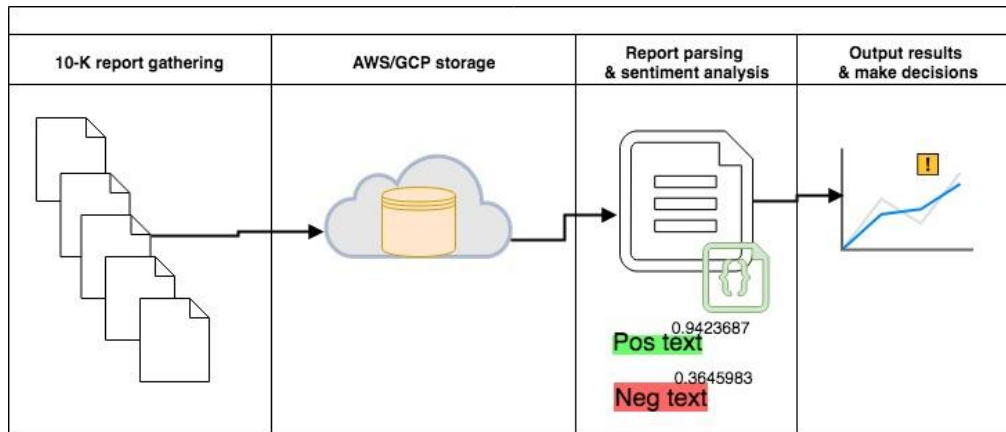


Figure A: Architecture of unstructured data analysis on company financial reports.

### 6.2 For Monte Carlo Simulation

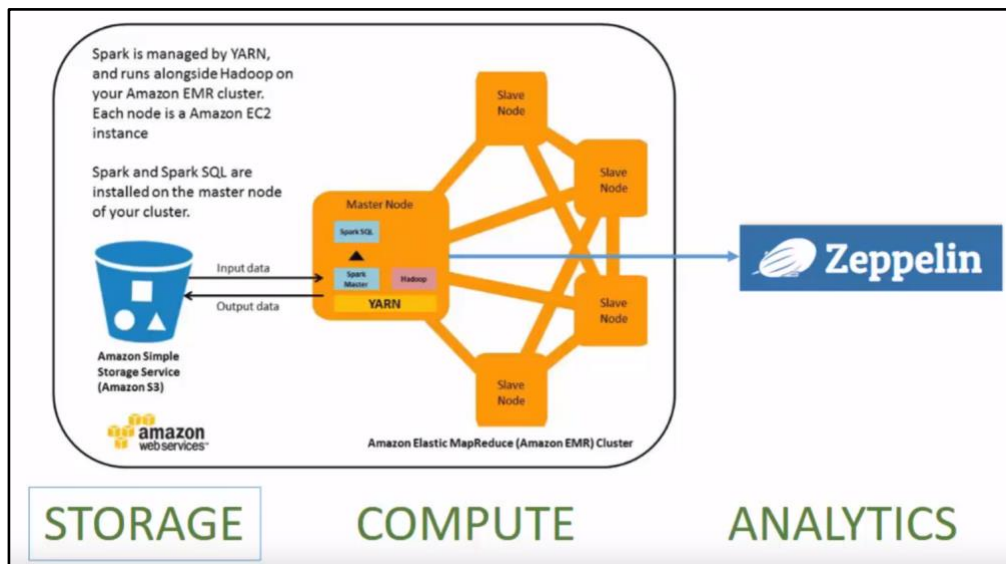


Figure B: Architecture of the project showing storage, computational power and analytics component used.<sup>4</sup>

<sup>4</sup> AWS EMR Spark Zeppelin: <https://www.youtube.com/watch?v=3VsNBf04A78>

## 6.3 For Twitter Feed Analysis

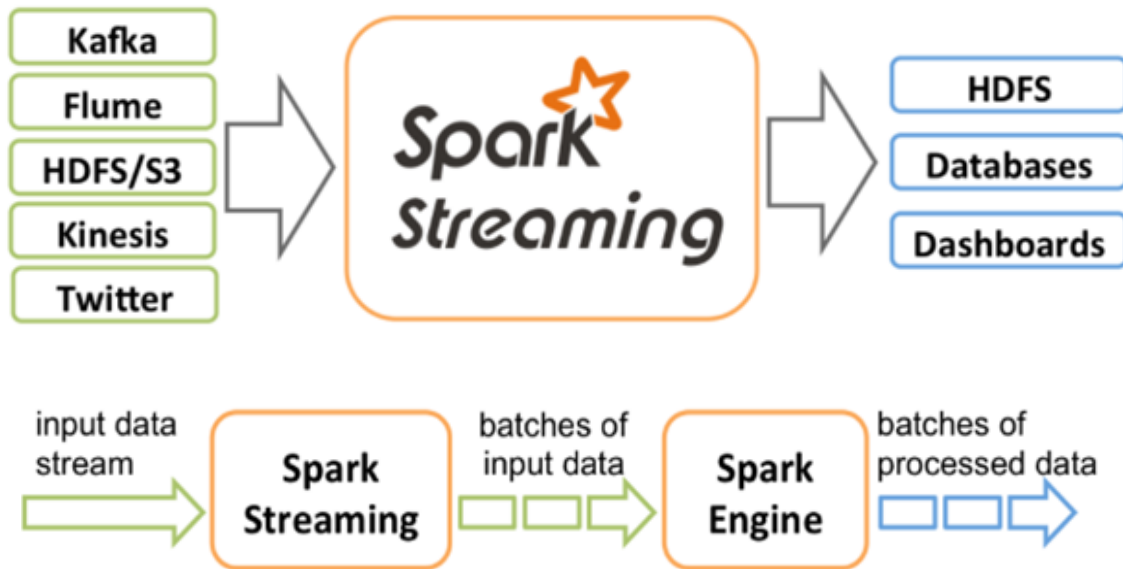


Figure C: Architecture of Spark Streaming for Twitter analysis.<sup>5</sup>

## 7. Implementation

### 7.1 Sentiment Analysis on 10-K Financial Reports

Part of our team project's unstructured data processing involved the analysis of company's 10-K financial filing reports. Our team sought to utilize the publicly available 10-K reports of a certain chosen group of companies to determine the efficacy of language processing and sentiment analysis on the textual data in the financial reports. Past studies such as Loughran & McDonald<sup>6</sup> and Pulliza<sup>7</sup> indicate that there are many trends and judgments that one can make purely from analyzing these firm performance reports.

In order to show that our methodology is a strong case, our team worked on a prototype representative model for the sentiment analysis on our five chosen Energy-related companies (Tesla, Vivint Solar, General Electric, Brookfield Renewable Partners, and Pattern Energy Group) on the past five years of filed 10-K financial reports. (NB: Vivint Solar did not file a 10-K in 2014.) The total size of all five companies 10-K reports (.txt files) totaled to 860MB; we stored the reports data on AWS S3. After cleaning up the textual reports (removing unnecessary "<html>" related tags), we were able to cut the size of some reports by about 70%. We used the built-in .polarity function from the Python package TextBlob, which uses the 'pattern' library and PatternAnalyzer functionality<sup>8</sup> in order to get the overall sentiment polarity a whole (cleaned-up) 10-K financial report. Our results of the overall sentiment performance from the past five years as well as on the 2018 filed reports can be found in the figures below.

<sup>5</sup> Spark Streaming Architecture: <https://spark.apache.org/docs/latest/streaming-programming-guide.html>

<sup>6</sup> Bodnaruk, Andriy and Loughran, Tim and McDonald, Bill. (2015). Using 10-K Text to Gauge Financial Constraints. Forthcoming in the Journal of Financial and Quantitative Analysis.

<sup>7</sup> Pulliza, J. (2017). Investigating the efficacy of sentence filtering in predicting analysts' ratings following a 10-K filing. Proceedings of the Association for Information Science and Technology, 54(1), 787-789.

<sup>8</sup> TextBlob Sentiment: [https://planspace.org/20150607-textblob\\_sentiment/](https://planspace.org/20150607-textblob_sentiment/)

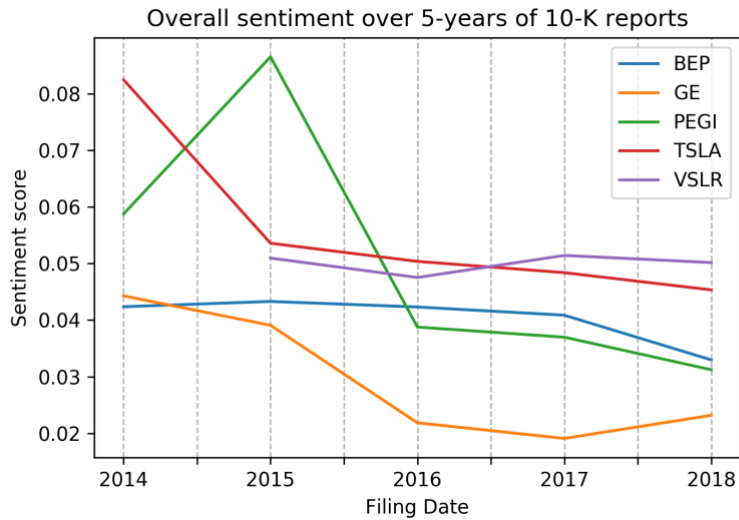


Figure D: Overall sentiment score from 10-K financial reports of the 5 Energy companies over 5-years.

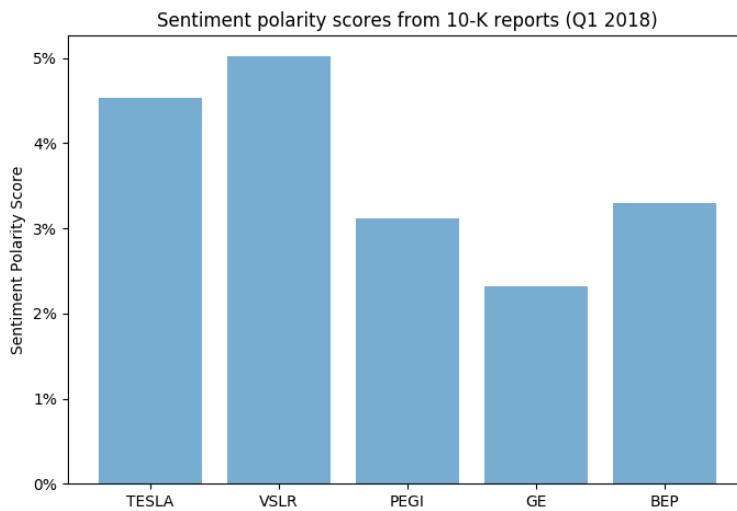


Figure E: Overall sentiment score on the 2018 10-K financial reports of the 5 Energy companies.

Even though our prototype implementation has plenty of room for improvement in regards to the accuracy of the polarity scoring, it is interesting to see the big picture of these companies from their publicly available 10-K reports. For example, for Tesla we performed a deep dive on the sentiment analysis results on its 2018 10-K financial report. We used the R package ‘edgar’ and positive/negative sentiment words list provided by Loughran & McDonald<sup>9</sup> to get our frequency count of sentiment words, seen in figure F.

<sup>9</sup> Loughran and McDonald Sentiment Word Lists: <https://sraf.nd.edu/textual-analysis/resources/>

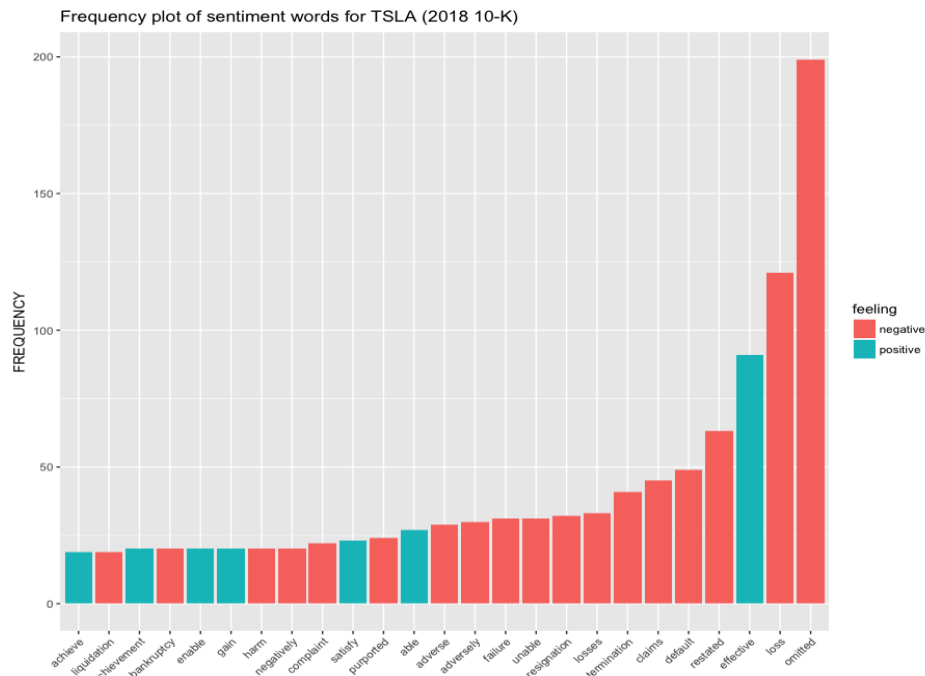


Figure F: Frequency bar-chart of most positive/negative words for Tesla from 2018 10-K filed report.

## 7.2 Calculation of Value at Risk (VaR) using Monte Carlo Simulations

### 7.2.1 Monte Carlo Simulation

In finance, there is a fair amount of uncertainty and risk involved with estimating the future value of financial products, due to the wide variety of potential outcomes. Monte Carlo simulation (MCS) (also known as the Monte Carlo method) allows inspecting many possible outcomes of the decision making process, and can be used to assess the impact of risk: this, in turn, allows for better decision-making under uncertainty. Monte Carlo simulation is a computerized mathematical technique that can be applied such that it is possible to account for risk in quantitative analysis and decision making. This technique is used in many different fields, such as R&D, risk management, portfolio management, pricing derivatives, strategic planning, project planning, cost modeling and many more.

<sup>10</sup>In general, MCS is a technique that "converts" uncertainty on input variables of a model into probability distributions. By combining the distributions and randomly selecting values from them, it recalculates the simulated model many times, to determine the probability of the output. The MCS method is also called a stochastic method because it uses random variables; more complex details can be researched in the field of Stochastic Processes. Note also that the general assumption is for input random variables to be independent from each other. When this is not the case, there are techniques to account for correlation between random variables. A MCS generates the output as a range instead of a fixed value and shows how likely the output value is to occur in that range. In other words, the model outputs a probability distribution.

<sup>10</sup> Introduction to Data Analysis using Spark  
[http://163.25.117.117/chmao/bigdata2015/Advanced\\_Analytics\\_with\\_Spark.pdf](http://163.25.117.117/chmao/bigdata2015/Advanced_Analytics_with_Spark.pdf)



## 7.2.2 The Data

The downloaded data was retrieved from Yahoo Finance. We chose a collection period from Apr 2013 to Apr 2018. We have such similar historical data for 5 stocks. We used the adjusted close prices to compute the weekly returns using R and PerformanceAnalytics package.

A	B	C	D	E	F	G
Date	Open	High	Low	Close	Adj Close	Volume
2013-01-02	35	35.450001	34.709999	35.360001	35.360001	1194800
2013-01-03	35.18	35.450001	34.75	34.77	34.77	742000
2013-01-04	34.799999	34.799999	33.919998	34.400002	34.400002	674000
2013-01-07	34.799999	34.799999	33.900002	34.34	34.34	442000
2013-01-08	34.5	34.5	33.110001	33.68	33.68	1284000
2013-01-09	34.009998	34.189999	33.400002	33.639999	33.639999	698000
2013-01-10	33.869999	33.990002	33.380001	33.529999	33.529999	922500
2013-01-11	34.040001	34.040001	32.110001	32.91	32.91	1563200
2013-01-14	33.080002	33.380001	32.849998	33.259998	33.259998	925100
2013-01-15	33.110001	34.25	33.080002	33.900002	33.900002	1624200
2013-01-16	33.849998	34.23	33.73	34.099998	34.099998	1378200
2013-01-17	34.16	34.849998	33.919998	34.380001	34.380001	1436700
2013-01-18	34.740002	34.779999	33.82	34.52	34.52	3555100

Figure G: Sample of Tesla's historical data from 2013

## 7.2.3 Parallelizing the code

As our main aim is to ease the computation and generate greater number of simulations to get a precise VaR value, we parallelize the code according to the Spark's architecture.

```
parallelism = 3
numTrials = 10000000
trial_indexes = list(range(0, parallelism))
seedRDD = sc.parallelize(trial_indexes, parallelism)
```

From the above part of code, we can observe that the numTrials is a very large number, which would have taken a long time to run if we do not parallelize these operations. Here, using the function *parallelize()*, where the driver node will break up the trials and assign these trials (ie., tasks) and also contain information regarding the split of the data it will operate on. Now, these trials where the simulations are being computed is being assigned to different worker nodes.

```
trials = seedRDD.flatMap(lambda idx: \
    simulateTrialReturns(max(int(numTrials/parallelism), 1), \
        factorMeans=0.000747222, factorCov=0.02434665, T=7,S=335.12))
```

Once the workers have been assigned different starting points, we apply a transformation function *flatMap()*. We specifically chose *flatMap()* because later on we can then combine all the output (ie., returns of the stock) resulting from the function *simulateTrialReturns()*.

Now, *trials* is a transformed *seedRDD* with all the transformations being done on it. In order to plot these simulations, we later collect the results, for which we use the *collect()* function.

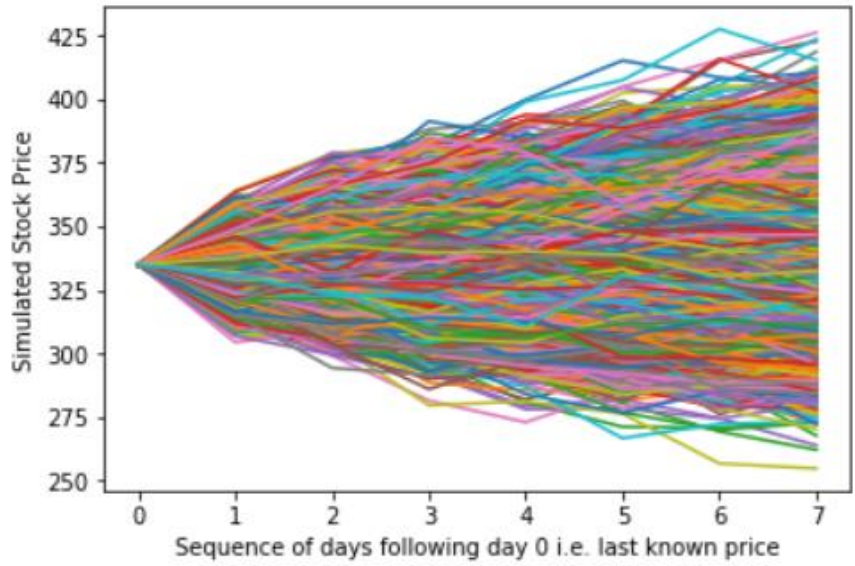


Figure H: Day 0 represents the last known closing price of the stock, and the figure shows what all possible values of the closing price could be for the next 7 days.

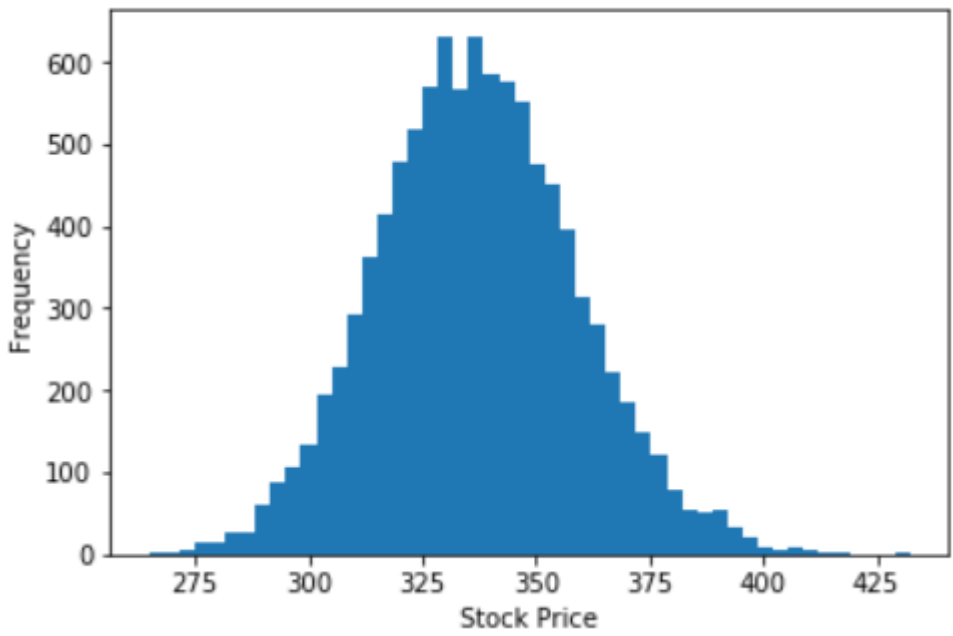


Figure I: This figure represents the worst possible case lies in the left corner of the distribution showing the 5th percentile, and the best case scenario is the right most corner of the distribution.

Stock	5 <sup>th</sup> Percentile	95 <sup>th</sup> Percentile
TESLA	283.0786504	404.2450693
VSLR	2.068533898	4.283002772
BEP	27.27485889	34.07272729
PEGI	14.80069974	19.93395855
GE	12.74691909	15.34155261

Figure J: Tabular summary of simulated values.

### 7.3 Twitter Feed Analysis

With the advancement of social media platforms such as Twitter and Facebook, industries are investing in resources to tap vast amount of customer sentiment information from these huge pools of unstructured data.

Amongst all different social media platforms, Twitter gives a candid snapshot of what the entire world is thinking about on any topic, at any point in time. <sup>11</sup>According to an article in the Financial Post, Twitter has 140 million active monthly users and is adding approximately 300,000 new users per day. Imagine how much information can be aggregated, analyzed, and deployed in systems from thousands and thousands of tweets. <sup>12</sup>

In the world of finance, real-time analysis and action can maximize the returns or minimize the losses well before the market has factored them into the stock price. Therefore, Twitter is an excellent tool for investors to use to stay on top of market events and screen potential investment opportunities on a real-time basis. However, it is impossible for any human to sift through thousands of tweets in even a day's time let alone real-time, gauge the sentiments, and make informed decisions.

To help asset managers analyze real-time demand of trending commodities, our team developed a product using Spark Streaming that fetches live tweets related to any given topic and creates a live dashboard of the most popular hashtags related to the specified word.

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like mapreduce, join, and window.

In the description below, we will take a glimpse of how this application works and how it could be used to make informed decisions. For this application, we made use of the following libraries: Tweepy - to connect python with twitter - and Matplotlib & Seaborn - for visualization.

We developed a python file "TweetRead.py" where we created a class which inherits from "StreamListener" class of "tweepy" library. A function listens to the twitter API and then uses JSON library to load and send text messages via a TCP-IP socket object created in this function.

<sup>11</sup> Tweak your Portfolio with Tweets, The Financial Post <http://business.financialpost.com/uncategorized/tweak-your-portfolio-with-tweets>

<sup>12</sup> Twitter Feed Analysis [https://blog.twitter.com/official/en\\_us/topics/insights/2017/Consumer-Insights-for-Finance-with-Twitter-Data.html](https://blog.twitter.com/official/en_us/topics/insights/2017/Consumer-Insights-for-Finance-with-Twitter-Data.html)

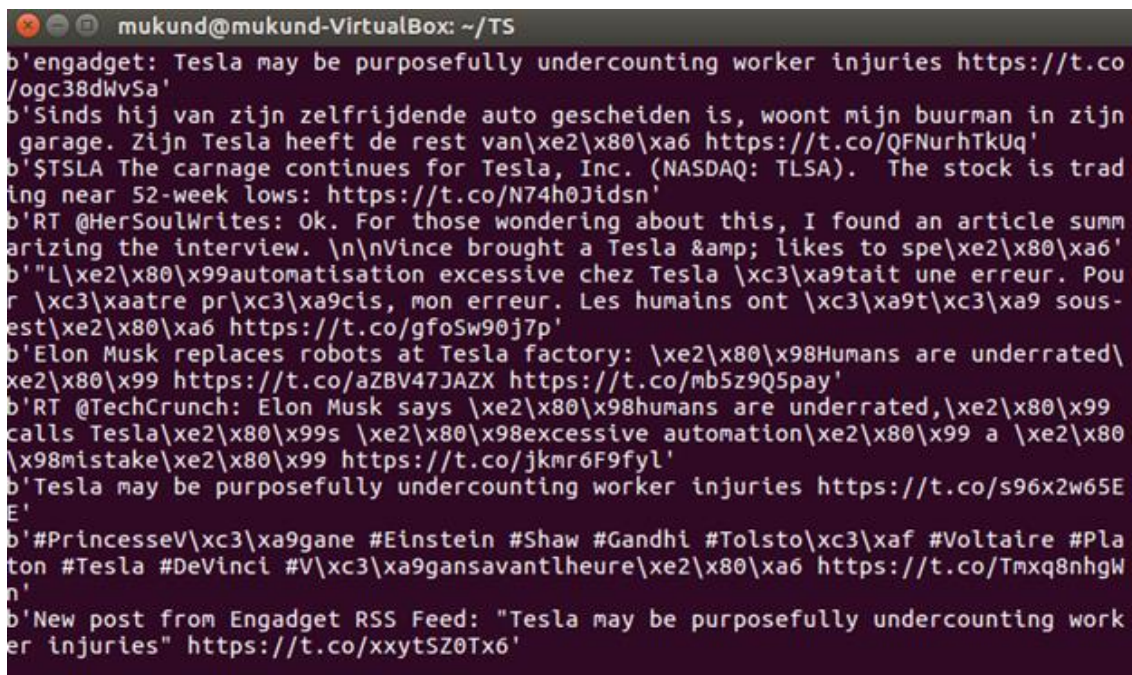
Subsequently, a second function uses the TCP socket object and Twitter API credentials (Consumer Key, Consumer Secret, Access Token, Access Secret) to stream and filter out data based on given keyword.

Snapshot: Function accepting socket object and API credentials.

```
def sendData(c_socket):#accepts token
    auth = OAuthHandler(consumer_key, consumer_secret)#OAuthHandler is a part of tweepy library
    auth.set_access_token(access_token, access_secret)

    twitter_stream = Stream(auth, TweetsListener(c_socket))
    twitter_stream.filter(track=['Tesla'])#insert search keyword (Similarly for BEP, VSLR, GE, and PEGI)
```

Snapshot: Tweets getting fetched via Twitter API.



```
mukund@mukund-VirtualBox: ~/TS
b'engadget: Tesla may be purposefully undercounting worker injuries https://t.co/ogc38dWvSa'
b'Sinds hij van zijn zelfrijdende auto gescheiden is, woont mijn buurman in zijn garage. Zijn Tesla heeft de rest van\xe2\x80\xa6 https://t.co/QFNurhTkUq'
b'$TSLA The carnage continues for Tesla, Inc. (NASDAQ: TSLA). The stock is trading near 52-week lows: https://t.co/N74h0Jidsn'
b'RT @HerSoulWrites: Ok. For those wondering about this, I found an article summarizing the interview. \n\nVince brought a Tesla & likes to spe\xe2\x80\xa6'
b'"L\xe2\x80\x99automatisation excessive chez Tesla \xc3\xa9tait une erreur. Pour \xc3\xaatre pr\xe9cis, mon erreur. Les humains ont \xc3\xa9 sous-estim\xe2\x80\xa6 https://t.co/gfo5w90j7p'
b'Elon Musk replaces robots at Tesla factory: \xe2\x80\x98Humans are underrated\xe2\x80\x99 https://t.co/aZBV47JAZX https://t.co/mb5z9Q5pay'
b'RT @TechCrunch: Elon Musk says \xe2\x80\x98humans are underrated,\xe2\x80\x99 calls Tesla\xe2\x80\x99s \xe2\x80\x99excessive automation\xe2\x80\x99 a \xe2\x80\x98mistake\xe2\x80\x99 https://t.co/jkmr6F9fyl'
b'Tesla may be purposefully undercounting worker injuries https://t.co/s96x2w65EE'
b'#PrincesseV\xe3\xa9gane #Einstein #Shaw #Gandhi #Tolsto\xe3\xaf #Voltaire #Platon #Tesla #DeVinci #V\xe3\xa9gansavantlheure\xe2\x80\xa6 https://t.co/Tmxq8nhgWn'
b'New post from Engadget RSS Feed: "Tesla may be purposefully undercounting worker injuries" https://t.co/xxytSZ0Tx6'
```

Then, we developed another file, named “SparkStreaming.ipynb” that runs the “TweetRead.py” and connects it to Spark Streaming.

Snapshot: Code snippet gives an idea of how the live dashboard is plotted.

```
In [12]: count = 0
while count < 10: #10 tweets at a time.

    time.sleep(3) #plot dashboard with a 3 second time frame.
    #Asset Managers can increase or decrease frequency accordingly.
    top_10_tweets = sqlContext.sql('Select tag, count from tweets') #queries on temporary table
    top_10_df = top_10_tweets.toPandas()
    display.clear_output(wait=True)
    plt.figure(figsize = (10, 8))
    sns.barplot(x="count", y="tag", data=top_10_df)
    plt.show()
    count = count + 1
```

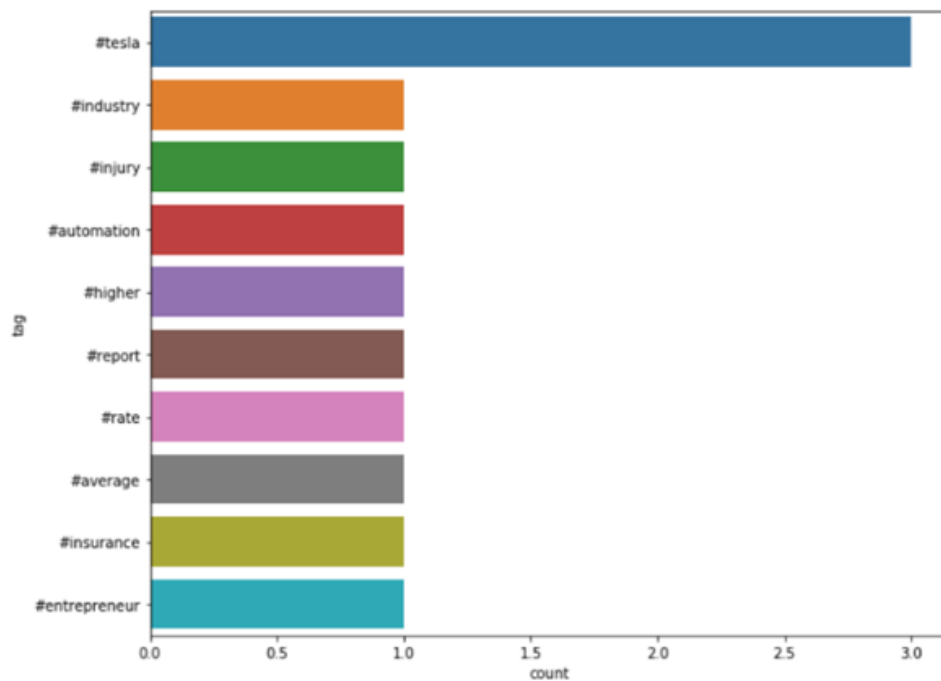


Figure K: Live dashboard static-view.

## 7.4 Project Application

Suppose we construct a single stock only portfolio of “Tesla” and invest \$100,000. Based on the reference adjusted closed price used (i.e. \$335.12), total number of stocks purchased will be equal to 298. Now, using 5% simulated price (i.e. \$283.07), our portfolio valuation would dip to approximately \$84,357. This would imply that the Value at Risk (VaR) after a week of market movement would be  $\$100,000 - 84,357 = \$15,642$ . Simultaneously, the asset manager can have a glance on how “Tesla” is faring in the market using sentiment and polarity analysis on text documents and how popular is the commodity in consideration via twitter analysis. Note: The idea demonstrated above is to simply explain how we can calculate VaR. In practice, an asset manager would be managing a basket of securities, including and not limited to stocks. And therefore, VaR would be calculated on the portfolio’s overall value.

## 8. Justification

In the current generation of many advanced technologies and big data tools, the amount of financial information out there is both valuable and overwhelming. One option an investor has is to adjust their financial portfolio of assets to maximize returns based on the historical data simulation work for VaR calculation. But this process, especially for large baskets of securities, is computationally intensive. Regarding the more traditional risk managers, investors also have the secondary option to carefully read and analyze a company's mentions on Twitter as well as published financial reports available on the SEC website. From this reading, they can determine the well-performing or disappointing companies. However, as one can imagine, this process is extremely tedious.

Our proposed platform of a dashboard view containing three components (VaR calculation simulations, live twitter analysis feed, and financial report analysis feed) aims to concentrate only what is necessary in order to make smart risk management decisions for an investor. One immediate



advantage that our platform offers is the fact that all our components are centralized; rather than having to manually download historical data to perform simulations or to even download the many financial reports via a script, our proposed platform does everything you would want to do in a single area. Our application also makes great use of the current big data technologies such as storage, cloud computation for simulation, and automatic parsing for sentiment analysis.

This project is most justified for portfolio managers and risk takers whose main interest is to maximize returns. And in order to provide verifiable and interpretable quantitative results, we needed to make use of the cloud storage and computing technologies in order to handle the high volume and high velocity<sup>13</sup> of financial data. Furthermore, in a similar vein to the importance of diversification for portfolio management, we show a variety of structured (historical time series) and unstructured (financial filings, live tweets) big data.

## **9. Technical Feasibility**

### **9.1 Sentiment Analysis of 10-K Reports**

Analysis of 10-K reports requires scraping it off from the Official SEC website, cleaning the HTML file, and then performing sentiment analysis on it. If you have multiple companies for which you want an analysis done, local machines cannot provide enough memory for the storage of these reports, and the cleaned version of it too. One can circumvent this problem by making use of AWS S3 storage to store the reports once it is scraped off, and use an EC2 instance to do the processing by pulling the cleaned reports from S3. Just 4 companies required more than 2GB of memory. Since 10-K reports and other reports are not released very often, i.e. rather than daily it is usually a monthly or yearly manner, we can also schedule to scrape and clean jobs to run only after a specific period of time.

### **9.2 Monte Carlo**

While simulations can certainly be implemented on local machines, one would require distributed systems to do heavy lifting and run massive computations. Hence, to run large number of simulations, we would require distributed systems. This is a technical limitation to run large number of Monte Carlo Simulations.

### **9.3 Twitter Feed Analysis**

#### **1. Port Availability :**

In order to extract twitter feeds, for streaming, we need a socket port available. Generally, the public ports which are open can dynamically be assigned for other use, and in this case, on which the twitter feeds are streaming will raise an error saying “Port is not available”. Therefore, we should make sure to assign an available and open port for streaming.

#### **2. Spark Context Assignment :**

In PySpark, SparkContext is already defined as ‘sc’, and is initialized when we start the interpreter. Hence, in order to create multiple and new jobs, we should stop the earlier context initialized using *sc.stop()* or restart the kernel.

## **10. Conclusion**

Our project’s proposed dashboard application with the major components in Monte Carlo Simulation, Twitter feed, and sentiment analysis aims to provide portfolio and risk managers with a centralized platform to make an informed decision. By harnessing more advanced Big Data technologies, we can scale the product to store and analyze terabytes of structured as

---

<sup>13</sup> Ahmed K., Bouhorma M., Ahmed M. (Oct 2014). Age of Big Data and Smart Cities: Privacy Trade-Off.

well as unstructured data and derive insights at an unprecedented speed which is not possible using traditional technologies and methods.