TradeCaster

FINAL REPORT

Leveraging Data Science to recommend high performing stocks

By Aditya Paliwal, Laurence McNally, and Sarthak Biyani
Guided By - Professor Vijaykumar Gandapodi

23rd JULY 2019

ABSTRACT

TradeCaster is a proof of concept which leverages different data science techniques to predict the trends of stock prices for a company, first phase focus was on the Microsoft. This uses 6 data sources such as historical stock prices, financial performance reports, and real time news information to predict the performance of the company's stock in the future. Neural Networks (Long Short-Term Memory), Sentiment Analysis (NLP) and basic regression models were used to predict stock prices.

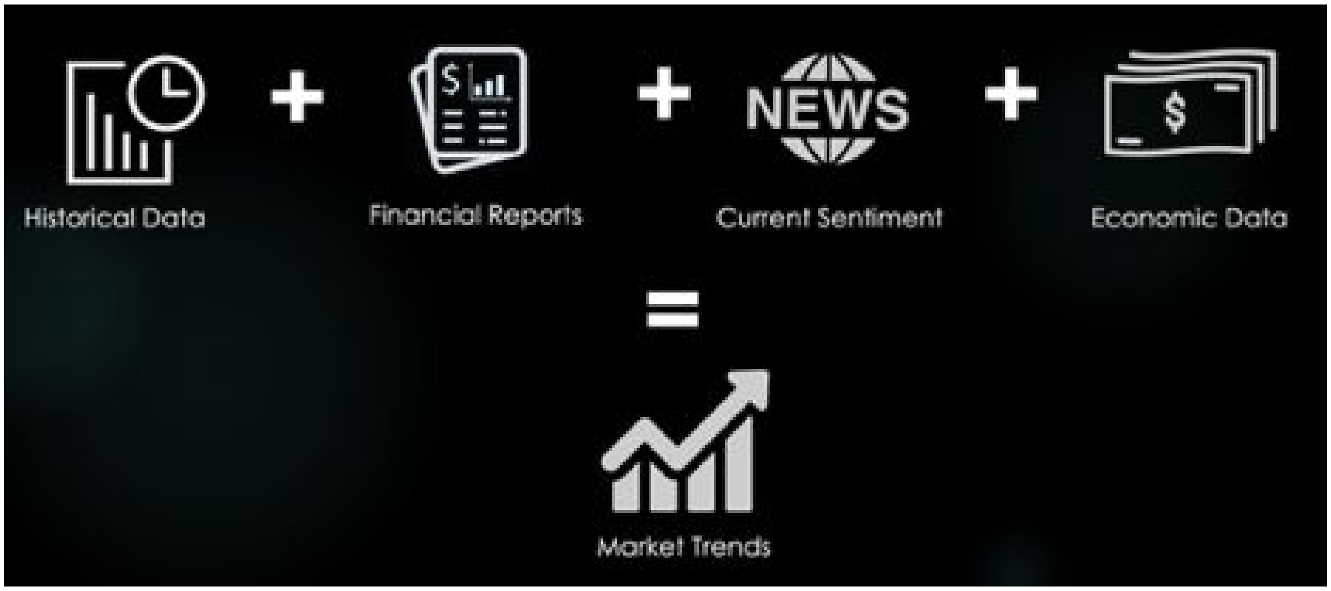
Why Microsoft?

Microsoft Corporation is an American multinational technology company with headquarters in Redmond, Washington. It develops, manufactures, licenses, supports and sells computer software, consumer electronics, personal computers, and related services. Its best known software products are the Microsoft Windows line of operating systems, the Microsoft Office suite, and the Internet Explorer and Edge Web browsers. Its flagship hardware products are the Xbox video game consoles and the Microsoft Surface lineup of touchscreen personal computers. As of 2016, it is the world's largest software maker by revenue.

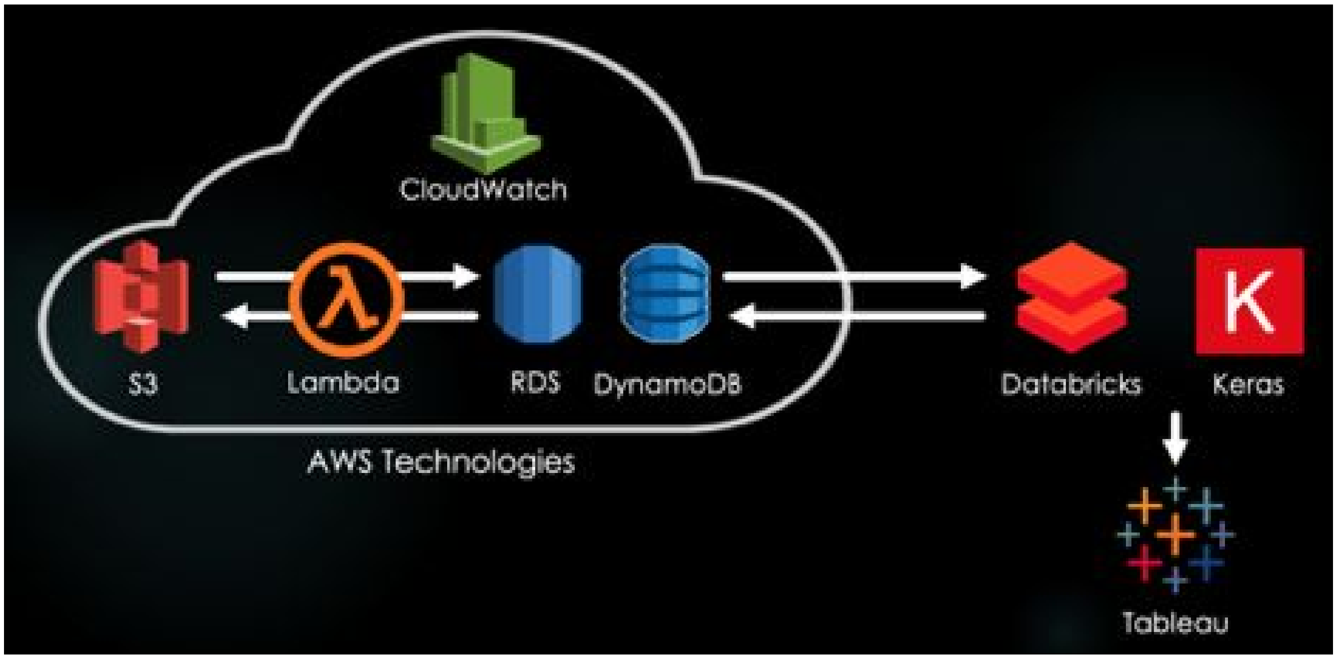
Microsoft has been part of Dow 30 companies consistently since last 10 years and has been consistent with particular domain and is only staunch towards technology. With Microsoft being a long player in the market had a stable share price over the years, it was easier to perform stock price prediction over the years.

When Microsoft went public and launched its Initial Public Offering (IPO) in 1986, the opening stock price was $21; after the trading day, the price closed at $27.75. As of July 2010, with the company's nine stock splits, any IPO shares would be multiplied by 288; if one were to buy the IPO today, given the splits and other factors, it would cost about 9 cents, The stock price peaked in 1999 at around $119 ($60.928, adjusting for splits). After certain splits and dividends its stock price is at $139.29 as of today(July, 23, 2019).

FUNCTIONAL OVERVIEW



ARCHITECTURAL DIAGRAM



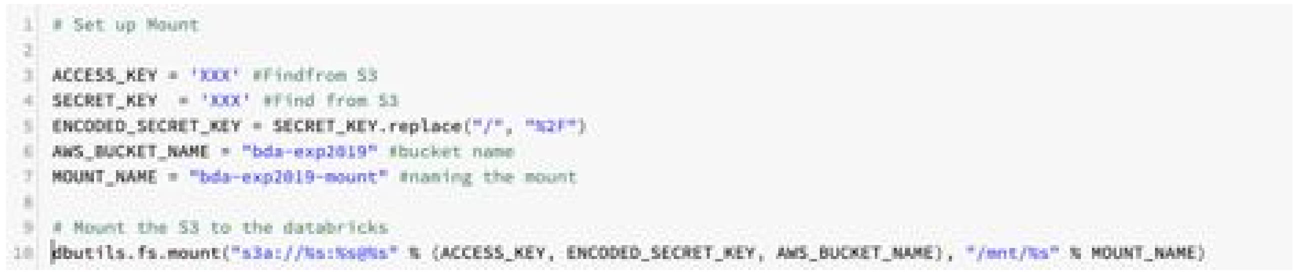
ETL

- The data will be extracted from 3 sources:
    1. Historical data from Yahoo Finance
    2. Live streamed news articles using the News API
    3. Financial data from WRDS
    4. Economic data for the country from federal reserve and bureau of labor website.
- Loading Data:
    1. Amazon S3 will store all the data that will be used in the project.
    2. Implementation of a script in Lambda in AWS to automate the process of populating data from files in S3 to DynamoDB and RDS.
    3. Pulling data from DynamoDB and RDS into Databricks
- Transforming / Cleaning the Data:
    1. Removing null values from the dataset.
    2. Merging stock data with financial data and economic data.
    3. Cleaning news article data (removing unwanted characters and words).
    4. Ensuring there is no redundant / duplicate data.

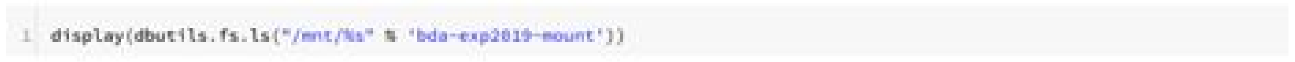
DATA STORAGE & SOURCING

The data will be sourced from 3 areas, News articles/blogs, Financial data from Wharton Research Data Services (WRDS) and historical stock data from Yahoo Finance.

1. Mounting the S3 Bucket to the Databricks Notebook



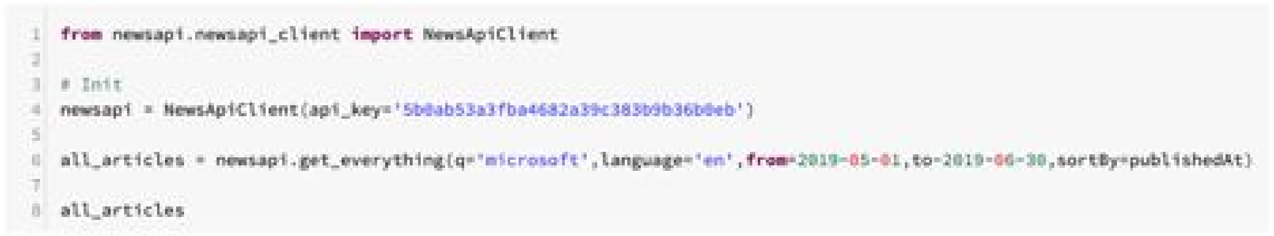
```
# Set up Mount

ACCESS_KEY = 'XXX' #Findfrom S3
SECRET_KEY  = 'XXX' #Find from S3
ENCODED_SECRET_KEY = SECRET_KEY.replace("/", "%2F")
AWS_BUCKET_NAME = "bda-exp2019" #bucket name
MOUNT_NAME = "bda-exp2019-mount" #naming the mount

# Mount the S3 to the databricks
dbutils.fs.mount("s3a://%s:%s@%s" % (ACCESS_KEY, ENCODED_SECRET_KEY, AWS_BUCKET_NAME), "/mnt/%s" % MOUNT_NAME)
```



```
display(dbutils.fs.ls("/mnt/%s" % 'bda-exp2019-mount'))
```

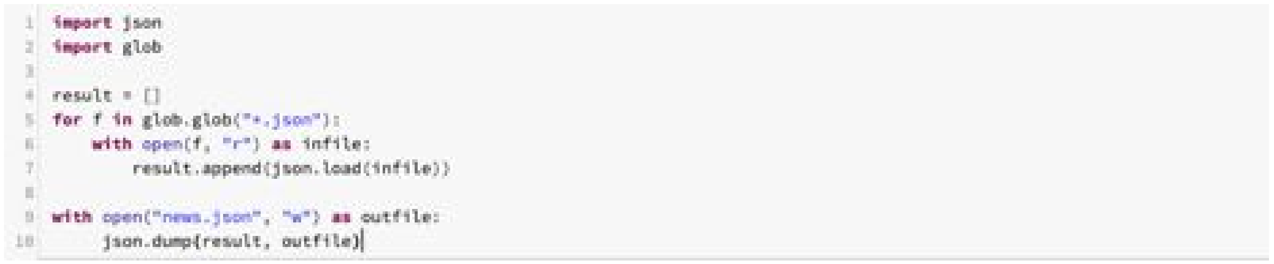
2. News Articles, blogs:

We are using the NewsAPI (https://newsapi.org/) to fetch news articles/blogs for a particular company for a given timeframe. We are pulling all articles/blogs for Microsoft from May 2019 to June 2019 to perform sentiment analysis.



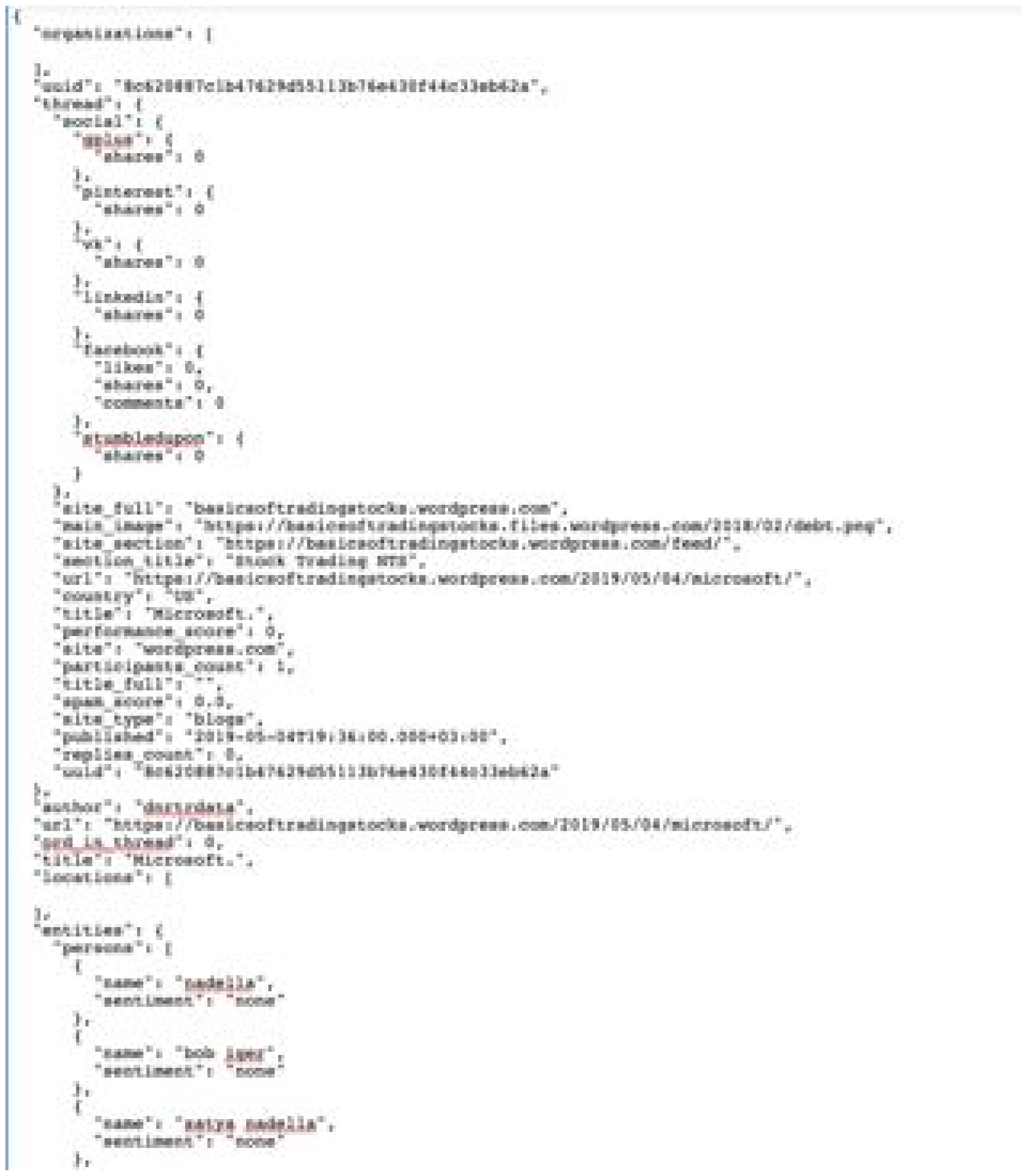
```python
from newsapi.newsapi_client import NewsApiClient

# Init
newsapi = NewsApiClient(api_key='5b0ab53a3fba4682a39c383b9b36b0eb')

all_articles = newsapi.get_everything(q='microsoft',language='en',from=2019-05-01,to=2019-06-30,sortBy=publishedAt)

all_articles
```

After fetching the articles and blog we merged them all into a single JSON file and uploaded it to S3.

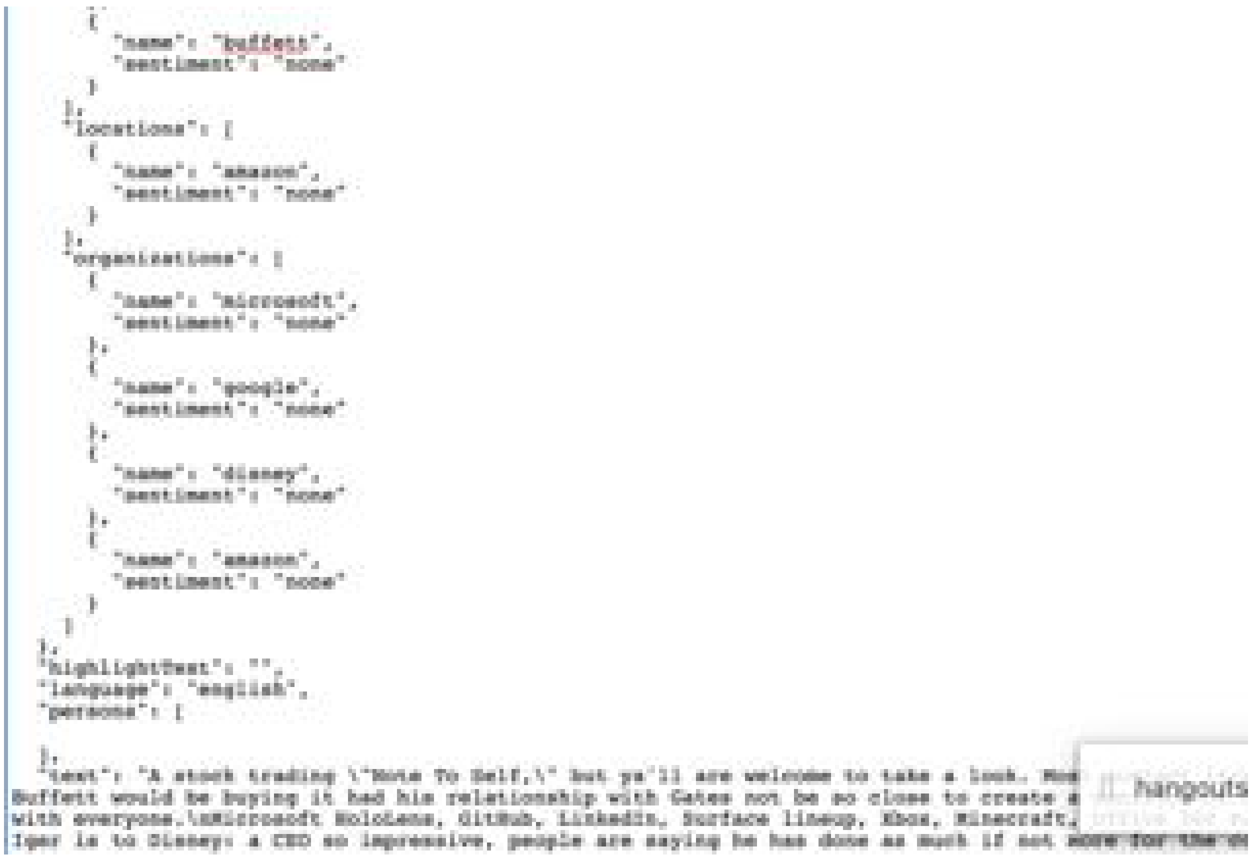


```python
import json
import glob

result = []
for f in glob.glob("*.json"):
    with open(f, "r") as infile:
        result.append(json.load(infile))

with open("news.json", "w") as outfile:
     json.dump(result, outfile)
```

Sample data:



```
{
  "organizations": [

  ],
  "uuid": "8c620887c1b47629d55113b76e430f44c33eb62a",
  "thread": {
    "social": {
      "gplus": {
        "shares": 0
      },
      "pinterest": {
        "shares": 0
      },
      "vk": {
        "shares": 0
      },
      "linkedin": {
        "shares": 0
      },
      "facebook": {
        "likes": 0,
        "shares": 0,
        "comments": 0
      },
      "stumbledupon": {
        "shares": 0
      }
    },
    "site_full": "basicsoftradingstocks.wordpress.com",
    "main_image": "https://basicsoftradingstocks.files.wordpress.com/2018/02/debt.png",
    "site_section": "https://basicsoftradingstocks.wordpress.com/feed/",
    "section_title": "Stock Trading NTS",
    "url": "https://basicsoftradingstocks.wordpress.com/2019/05/04/microsoft/",
    "country": "US",
    "title": "Microsoft.",
    "performance_score": 0,
    "site": "wordpress.com",
    "participants_count": 1,
    "title_full": "",
    "spam_score": 0.0,
    "site_type": "blogs",
    "published": "2019-05-04T19:36:00.000+03:00",
    "replies_count": 0,
    "uuid": "8c620887c1b47629d55113b76e430f44c33eb62a"
  },
  "author": "dsrtrdata",
  "url": "https://basicsoftradingstocks.wordpress.com/2019/05/04/microsoft/",
  "ord_in_thread": 0,
  "title": "Microsoft.",
  "locations": [

  ],
  "entities": {
    "persons": [
      {
        "name": "nadella",
        "sentiment": "none"
      },
      {
        "name": "bob iger",
        "sentiment": "none"
      },
      {
        "name": "satya nadella",
        "sentiment": "none"
      },
```



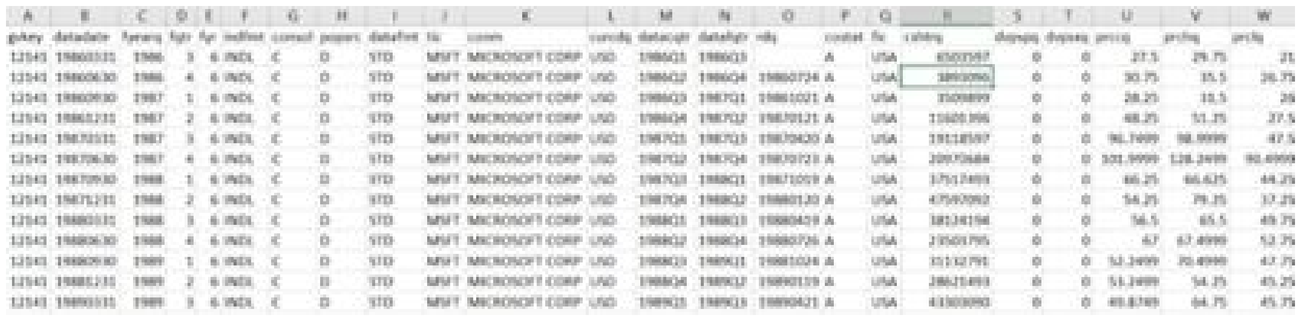
```
      {
        "name": "buffett",
        "sentiment": "none"
      }
    ],
    "locations": [
      {
        "name": "amazon",
        "sentiment": "none"
      }
    ],
    "organizations": [
      {
        "name": "microsoft",
        "sentiment": "none"
      },
      {
        "name": "google",
        "sentiment": "none"
      },
      {
        "name": "disney",
        "sentiment": "none"
      },
      {
        "name": "amazon",
        "sentiment": "none"
      }
    ]
  },
  "highlightText": "",
  "language": "english",
  "persons": [

  ],
  "text": "A stock trading \"Note To Self,\" but ya'll are welcome to take a look. Mos
Buffett would be buying it had his relationship with Gates not be so close to create a
with everyone.\nMicrosoft HoloLens, GitHub, LinkedIn, Surface lineup, Xbox, Minecraft,
Iger is to Disney: a CEO so impressive, people are saying he has done as much if not more for the 
```

3. Financial data for a company:

We are querying the WRDS (https://wrds-web.wharton.upenn.edu/wrds/) database to obtain financial information of a company like net profits, dividends, sales, margins, total revenue, net income, dividend, total assets, total liabilities, net tangible assets, investments, etc.

Financial data is a necessity to take in consideration for predicting share prices as it gives an indication of how the company is performing currently in the past quarter and what is its roadmap ahead. With financial data being good or bad investors make their decision about investing which results in volatility of stock prices.

Wharton Research Data Services have company's financial data over the years from 1986 and it includes all Cash Flow Statements, Balance Sheets, Income Statements on a quarterly basis.

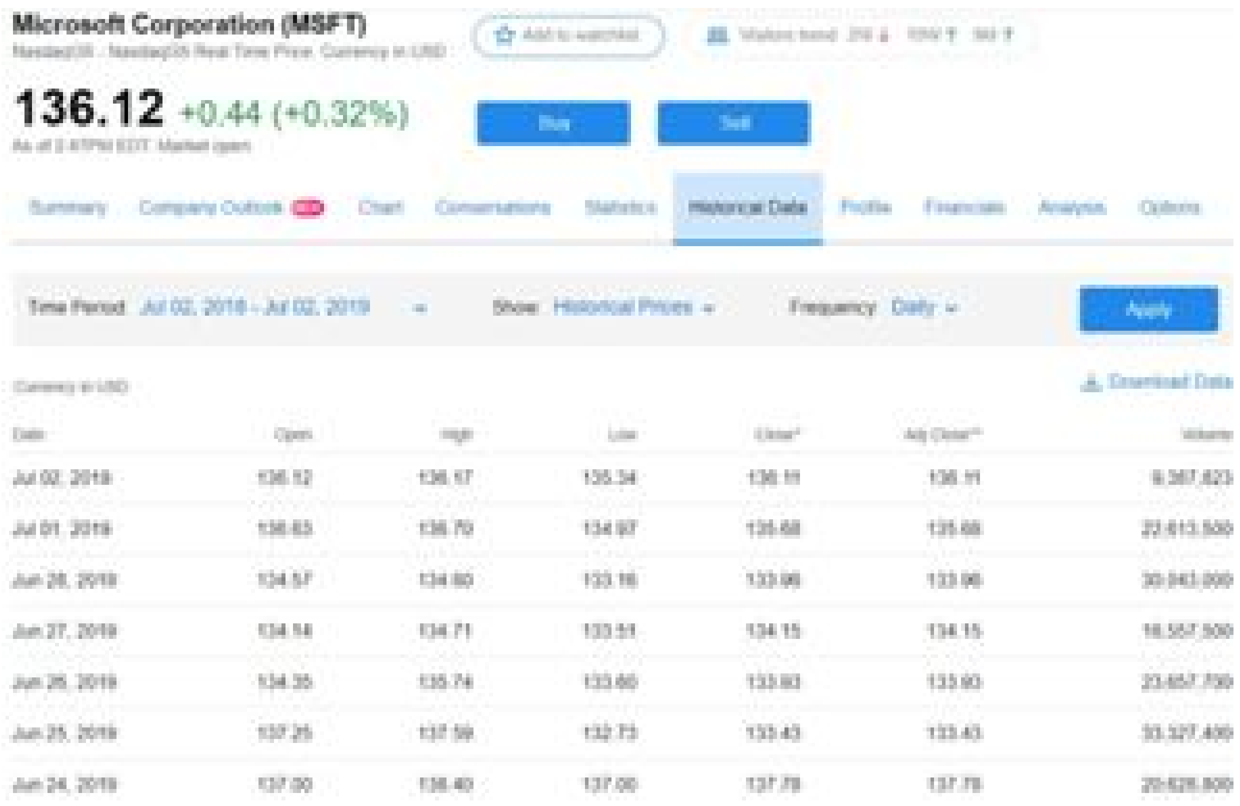


gvkey	datadate	fyearq	fqtr	fyr	indfmt	consol	popsrc	datafmt	tic	conm	curcdq	datacqtr	datafqtr	rdq	costat	fic	cshtrq	dvpspq	dvpsxq	prccq	prchq	prclq
12141	19860331	1986	3	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1986Q1	1986Q3		A	USA	6503597	0	0	27.5	29.75	21
12141	19860630	1986	4	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1986Q2	1986Q4	19860724	A	USA	3893096	0	0	30.75	35.5	26.75
12141	19860930	1987	1	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1986Q3	1987Q1	19861021	A	USA	3509899	0	0	28.25	33.5	26
12141	19861231	1987	2	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1986Q4	1987Q2	19870121	A	USA	11600396	0	0	48.25	51.25	27.5
12141	19870331	1987	3	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1987Q1	1987Q3	19870420	A	USA	19118597	0	0	96.7499	98.9999	47.5
12141	19870630	1987	4	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1987Q2	1987Q4	19870721	A	USA	20973684	0	0	101.9999	128.2499	90.4999
12141	19870930	1988	1	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1987Q3	1988Q1	19871019	A	USA	37517493	0	0	66.25	66.625	44.25
12141	19871231	1988	2	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1987Q4	1988Q2	19880120	A	USA	47597092	0	0	54.25	79.25	37.25
12141	19880331	1988	3	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1988Q1	1988Q3	19880419	A	USA	38124194	0	0	56.5	65.5	49.75
12141	19880630	1988	4	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1988Q2	1988Q4	19880726	A	USA	23501795	0	0	67	67.4999	52.75
12141	19880930	1989	1	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1988Q3	1989Q1	19881024	A	USA	31132791	0	0	52.2499	70.4999	47.75
12141	19881231	1989	2	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1988Q4	1989Q2	19890119	A	USA	28621493	0	0	51.2499	54.25	45.25
12141	19890331	1989	3	6	INDL	C	D	STD	MSFT	MICROSOFT CORP	USD	1989Q1	1989Q3	19890421	A	USA	43303090	0	0	49.8749	64.75	45.75

4. Historical stock data from Yahoo finance:

We are fetching historical stock data from yahoo finance for Microsoft starting from 1986 to June 30th 2019.
It is absolutely necessary to take historical data of the company from a long time period because it gives an indication of how the stock prices have sailed through over the years and how has the company's stock price moved over in correspondence to stock prices.

Microsoft launched it IPO at $21 and its current price is approximately $140 after 9 splits and providing multiple dividends over the years from 1986 to current time.



Microsoft Corporation (MSFT)

NasdaqGS - NasdaqGS Real Time Price. Currency in USD

136.12 +0.44 (+0.32%)

As of 2:47PM EDT. Market open.

Time Period: Jul 02, 2018 - Jul 02, 2019 Show: Historical Prices Frequency: Daily

Currency in USD

Date	Open	High	Low	Close*	Adj Close**	Volume
Jul 02, 2019	136.12	136.17	135.34	136.11	136.11	8,387,823
Jul 01, 2019	136.63	136.70	134.97	135.68	135.68	22,613,500
Jun 28, 2019	134.57	134.60	133.16	133.96	133.96	30,043,000
Jun 27, 2019	134.14	134.71	133.51	134.15	134.15	16,557,500
Jun 26, 2019	134.35	135.74	133.60	133.93	133.93	23,657,700
Jun 25, 2019	137.25	137.59	132.73	133.43	133.43	33,327,400
Jun 24, 2019	137.00	138.40	137.00	137.78	137.78	20,628,800

5. Economic data from Federal Reserve and Labour Bureau Website:

a) We took Inflation rate for U.S.A from 1986-2019 available on Federal Reserve Website.

Inflation refers to an overall increase in the Consumer Price Index (CPI), which is a weighted average of prices for different goods. The set of goods that make up the index depends on which are considered representative of a common consumption basket. Therefore, depending on the country and the consumption habits of the majority of the population, the index will comprise different goods. With change in inflation rate it highly affects the sales of company so it was necessary to incorporate this doing stock price prediction.

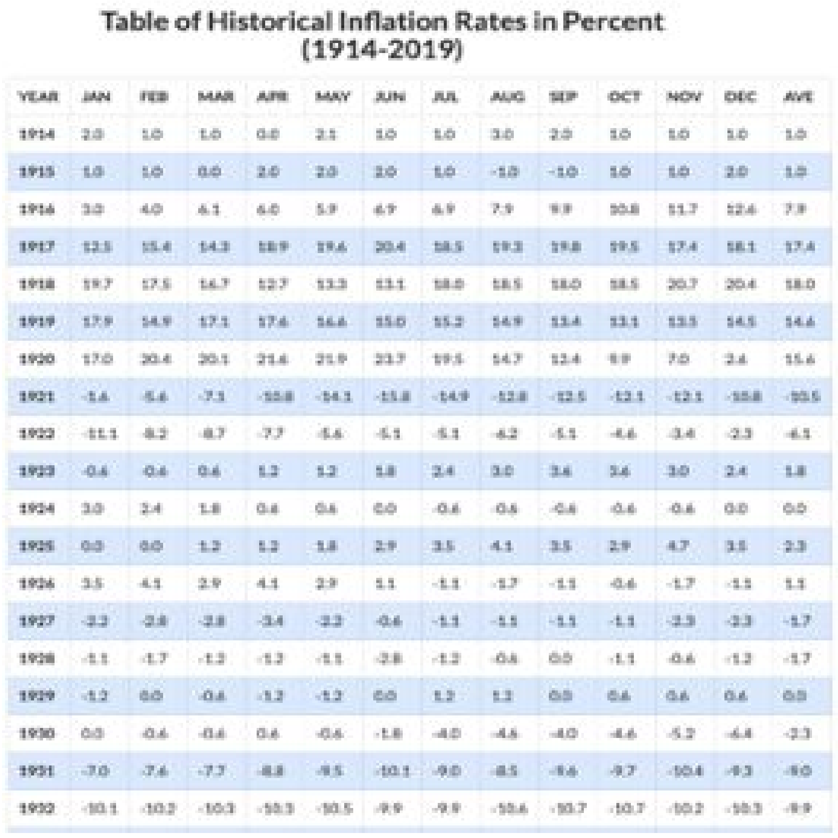


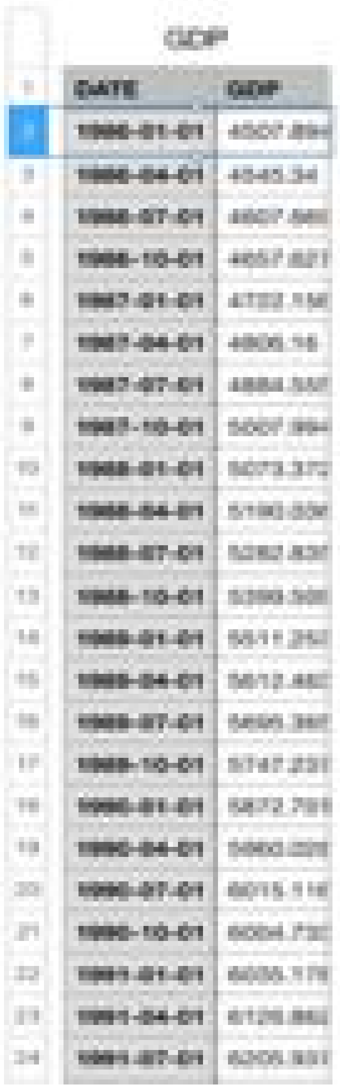
Table of Historical Inflation Rates in Percent (1914-2019)

YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	AVE
1914	2.0	1.0	1.0	0.0	2.1	1.0	1.0	3.0	2.0	1.0	1.0	1.0	1.0
1915	1.0	1.0	0.0	2.0	2.0	2.0	1.0	-1.0	-1.0	1.0	1.0	2.0	1.0
1916	3.0	4.0	6.1	6.0	5.9	6.9	6.9	7.9	9.9	10.8	11.7	12.6	7.9
1917	12.5	15.4	14.3	18.9	19.6	20.4	18.5	19.3	19.8	19.5	17.4	18.1	17.4
1918	19.7	17.5	16.7	12.7	13.3	13.1	18.0	18.5	18.0	18.5	20.7	20.4	18.0
1919	17.9	14.9	17.1	17.6	16.6	15.0	15.2	14.9	13.4	13.1	13.5	14.5	14.6
1920	17.0	20.4	20.1	21.6	21.9	23.7	19.5	14.7	12.4	9.9	7.0	2.6	15.6
1921	-1.6	-5.6	-7.1	-10.8	-14.1	-15.8	-14.9	-12.8	-12.5	-12.1	-12.1	-10.8	-10.5
1922	-11.1	-8.2	-8.7	-7.7	-5.6	-5.1	-5.1	-6.2	-5.1	-4.6	-3.4	-2.3	-6.1
1923	-0.6	-0.6	0.6	1.2	1.2	1.8	2.4	3.0	3.6	3.6	3.0	2.4	1.8
1924	3.0	2.4	1.8	0.6	0.6	0.0	-0.6	-0.6	-0.6	-0.6	-0.6	0.0	0.0
1925	0.0	0.0	1.2	1.2	1.8	2.9	3.5	4.1	3.5	2.9	4.7	3.5	2.3
1926	3.5	4.1	2.9	4.1	2.9	1.1	-1.1	-1.7	-1.1	-0.6	-1.7	-1.1	1.1
1927	-2.3	-2.8	-2.8	-3.4	-2.2	-0.6	-1.1	-1.1	-1.1	-1.1	-2.3	-2.3	-1.7
1928	-1.1	-1.7	-1.2	-1.2	-1.1	-2.8	-1.2	-0.6	0.0	-1.1	-0.6	-1.2	-1.7
1929	-1.2	0.0	-0.6	-1.2	-1.2	0.0	1.2	1.2	0.0	0.6	0.6	0.6	0.0
1930	0.0	-0.6	-0.6	0.6	-0.6	-1.8	-4.0	-4.6	-4.0	-4.6	-5.2	-6.4	-2.3
1931	-7.0	-7.6	-7.7	-8.8	-9.5	-10.1	-9.0	-8.5	-9.6	-9.7	-10.4	-9.3	-9.0
1932	-10.1	-10.2	-10.3	-10.3	-10.5	-9.9	-9.9	-10.6	-10.7	-10.7	-10.2	-10.3	-9.9

b) We are fetching GDP for U.S.A from 1986-2019 available on Federal Reserve.

Gross Domestic Product (GDP) is the total monetary or market value of all the finished goods and services produced within a country's borders in a specific time period. As a broad measure of overall domestic production, it functions as a comprehensive scorecard of the country’s economic health. GDP includes all private and public consumption, government outlays, investments, additions to private inventories, paid-in construction costs, and the foreign balance of trade.

GDP plays a significant role in identifying where is the economy of the country heading and economical situation of the country plays a key role in company's performance.



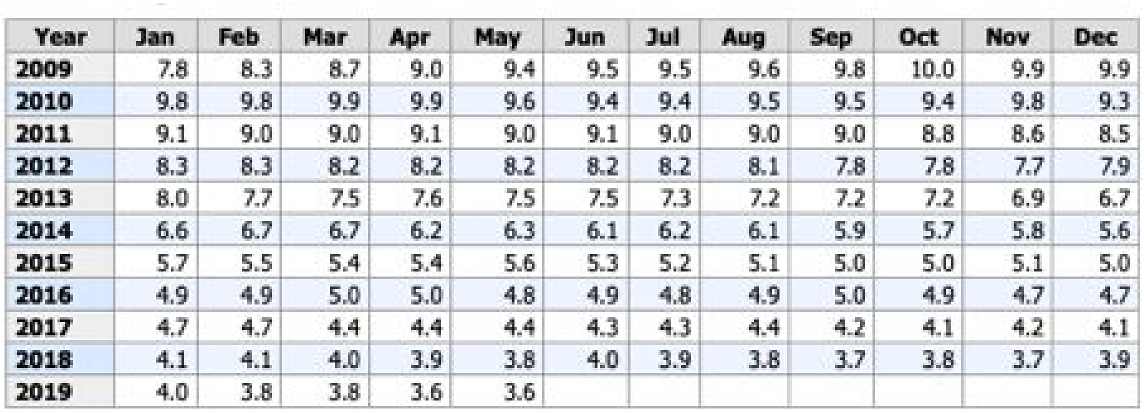
GDP

DATE	GDP
1986-01-01	4507.89
1986-04-01	4545.34
1986-07-01	4607.66
1986-10-01	4657.62
1987-01-01	4722.15
1987-04-01	4806.16
1987-07-01	4884.55
1987-10-01	5007.99
1988-01-01	5073.37
1988-04-01	5190.03
1988-07-01	5282.83
1988-10-01	5399.50
1989-01-01	5511.25
1989-04-01	5612.46
1989-07-01	5695.36
1989-10-01	5747.23
1990-01-01	5872.70
1990-04-01	5960.02
1990-07-01	6015.11
1990-10-01	6004.73
1991-01-01	6035.17
1991-04-01	6126.86
1991-07-01	6205.93

c) We are fetching Unemployment Rate for U.S.A from 1986-2019 available on Federal Reserve.

The unemployment rate is the share of the labor force that is jobless, expressed as a percentage. It is a lagging indicator, meaning that it generally rises or falls in the wake of changing economic conditions, rather than anticipating them. When the economy is in poor shape and jobs are scarce, the unemployment rate can be expected to rise. When the economy is growing at a healthy rate and jobs are relatively plentiful, it can be expected to fall.

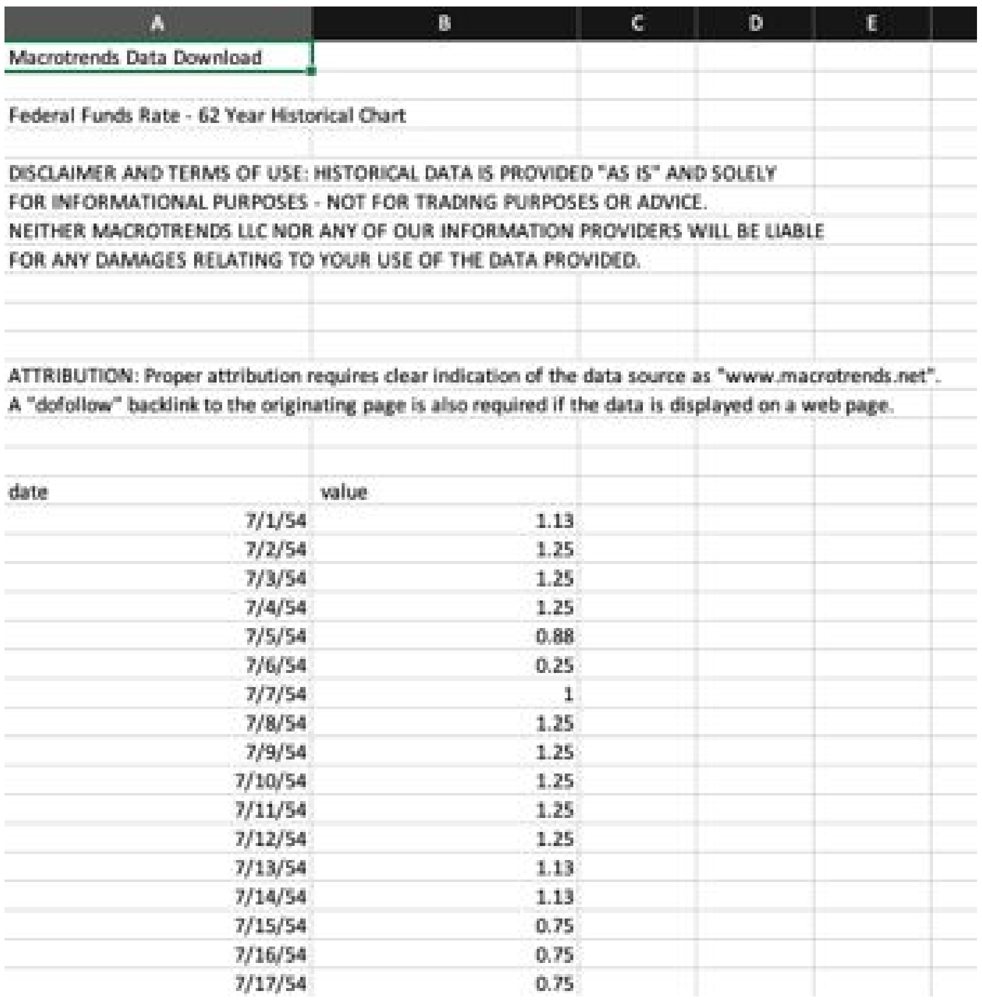
It gives an idea of the job market which is significant as in a growing market companies tend to employ more people.



Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2009	7.8	8.3	8.7	9.0	9.4	9.5	9.5	9.6	9.8	10.0	9.9	9.9
2010	9.8	9.8	9.9	9.9	9.6	9.4	9.4	9.5	9.5	9.4	9.8	9.3
2011	9.1	9.0	9.0	9.1	9.0	9.1	9.0	9.0	9.0	8.8	8.6	8.5
2012	8.3	8.3	8.2	8.2	8.2	8.2	8.2	8.1	7.8	7.8	7.7	7.9
2013	8.0	7.7	7.5	7.6	7.5	7.5	7.3	7.2	7.2	7.2	6.9	6.7
2014	6.6	6.7	6.7	6.2	6.3	6.1	6.2	6.1	5.9	5.7	5.8	5.6
2015	5.7	5.5	5.4	5.4	5.6	5.3	5.2	5.1	5.0	5.0	5.1	5.0
2016	4.9	4.9	5.0	5.0	4.8	4.9	4.8	4.9	5.0	4.9	4.7	4.7
2017	4.7	4.7	4.4	4.4	4.4	4.3	4.3	4.4	4.2	4.1	4.2	4.1
2018	4.1	4.1	4.0	3.9	3.8	4.0	3.9	3.8	3.7	3.8	3.7	3.9
2019	4.0	3.8	3.8	3.6	3.6							

d) We are fetching Federal Fund Rate (Interest Rate) for U.S.A from 1986-2019 available on Federal Reserve.

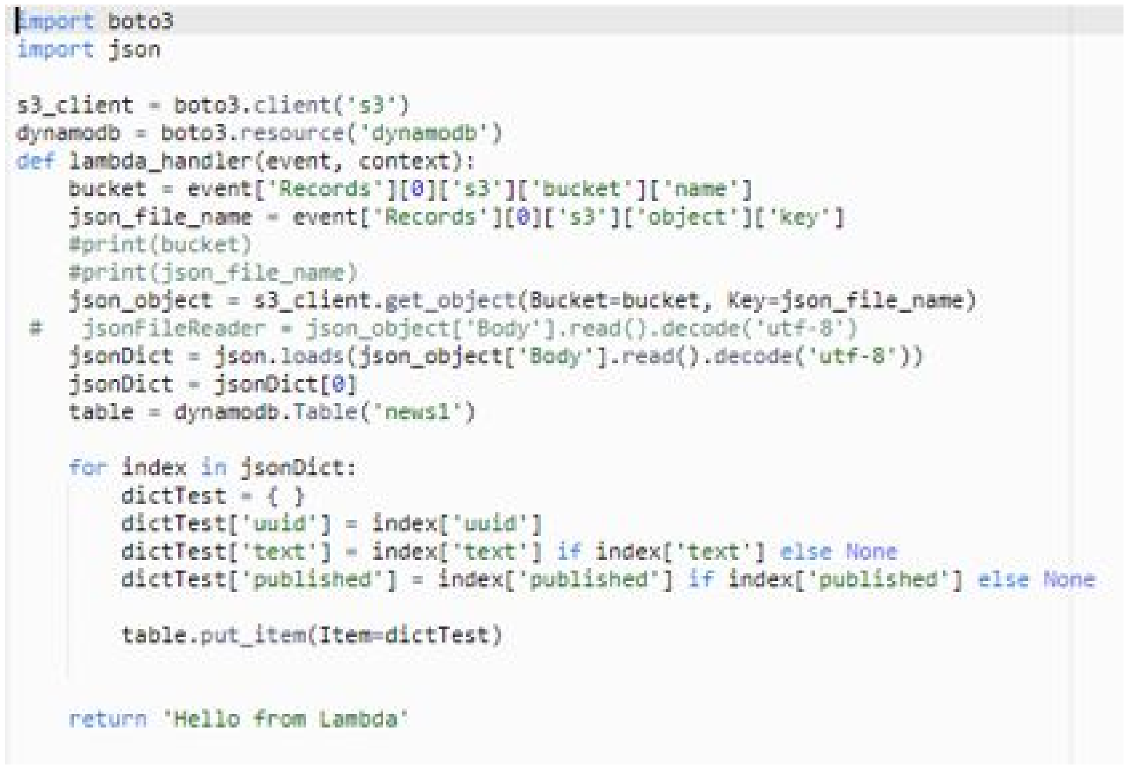
The federal funds rate refers to the interest rate that banks charge other banks for lending them money from their reserve balances on an overnight basis. By law, banks must maintain a reserve equal to a certain percentage of their deposits in an account at a Federal Reserve bank. Any money in their reserve that exceeds the required level is available for lending to other banks that might have a shortfall.



A	B	C	D	E
Macrotrends Data Download				
Federal Funds Rate - 62 Year Historical Chart				
DISCLAIMER AND TERMS OF USE: HISTORICAL DATA IS PROVIDED "AS IS" AND SOLELY				
FOR INFORMATIONAL PURPOSES - NOT FOR TRADING PURPOSES OR ADVICE.				
NEITHER MACROTRENDS LLC NOR ANY OF OUR INFORMATION PROVIDERS WILL BE LIABLE				
FOR ANY DAMAGES RELATING TO YOUR USE OF THE DATA PROVIDED.				
ATTRIBUTION: Proper attribution requires clear indication of the data source as "www.macrotrends.net".				
A "dofollow" backlink to the originating page is also required if the data is displayed on a web page.				
date	value			
7/1/54	1.13			
7/2/54	1.25			
7/3/54	1.25			
7/4/54	1.25			
7/5/54	0.88			
7/6/54	0.25			
7/7/54	1			
7/8/54	1.25			
7/9/54	1.25			
7/10/54	1.25			
7/11/54	1.25			
7/12/54	1.25			
7/13/54	1.13			
7/14/54	1.13			
7/15/54	0.75			
7/16/54	0.75			
7/17/54	0.75			

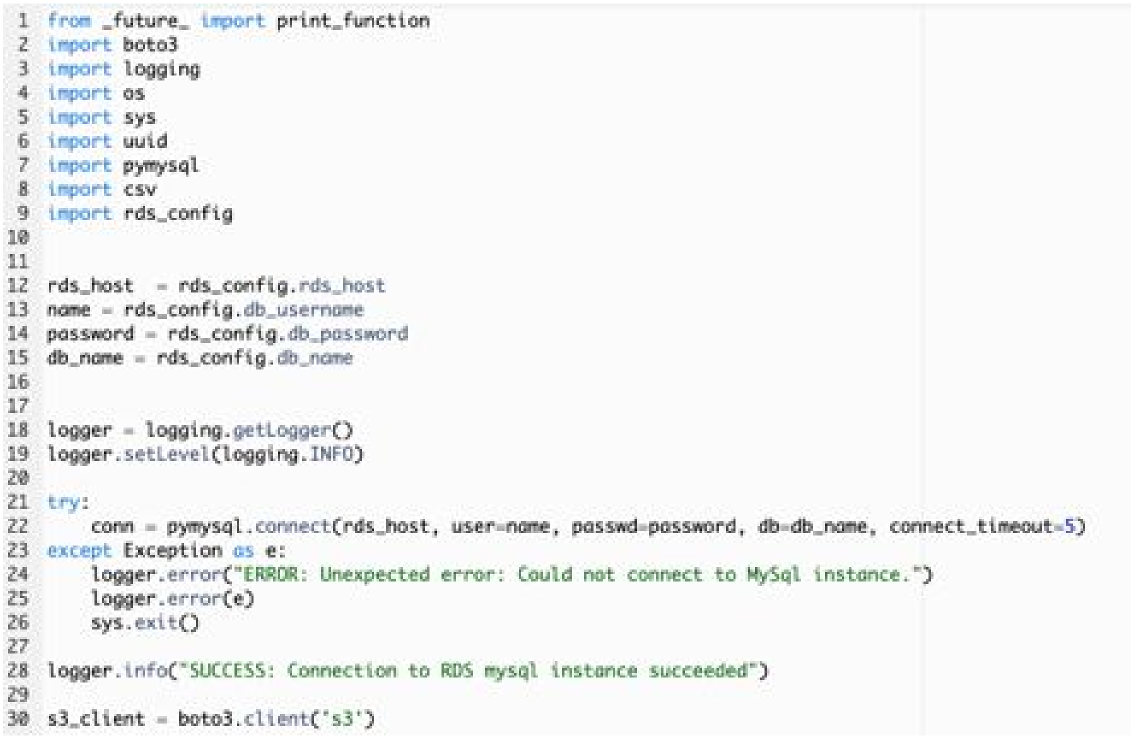
LAMBDA FUNCTION: Pull Data from S3 & Push to RDS/DynamoDB

DynamoDB

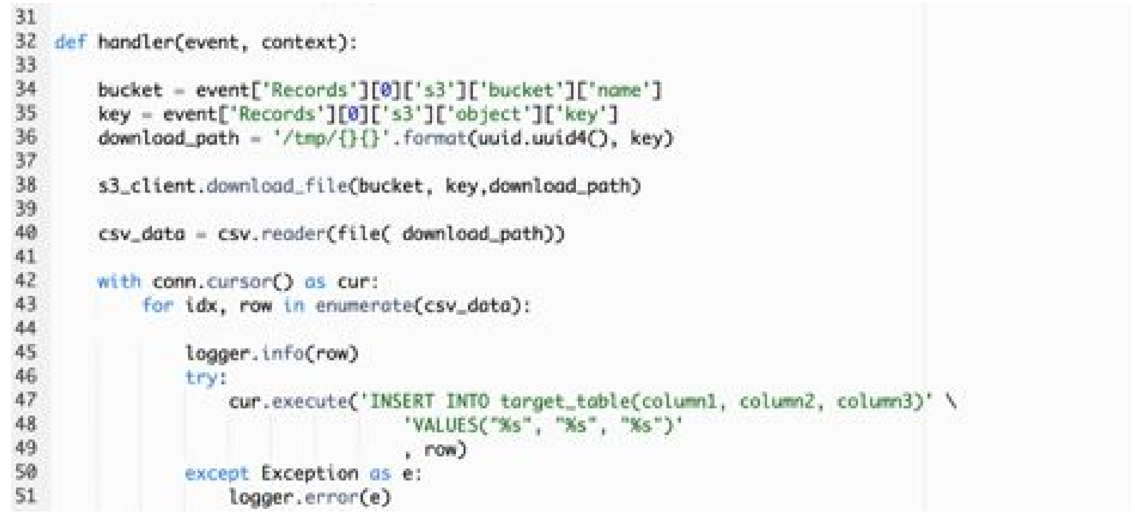


```python
import boto3
import json

s3_client = boto3.client('s3')
dynamodb = boto3.resource('dynamodb')
def lambda_handler(event, context):
    bucket = event['Records'][0]['s3']['bucket']['name']
    json_file_name = event['Records'][0]['s3']['object']['key']
    #print(bucket)
    #print(json_file_name)
    json_object = s3_client.get_object(Bucket=bucket, Key=json_file_name)
 #   jsonFileReader = json_object['Body'].read().decode('utf-8')
    jsonDict = json.loads(json_object['Body'].read().decode('utf-8'))
    jsonDict = jsonDict[0]
    table = dynamodb.Table('news1')

    for index in jsonDict:
        dictTest = { }
        dictTest['uuid'] = index['uuid']
        dictTest['text'] = index['text'] if index['text'] else None
        dictTest['published'] = index['published'] if index['published'] else None

        table.put_item(Item=dictTest)


    return 'Hello from Lambda'
```

RDS



```python
from _future_ import print_function
import boto3
import logging
import os
import sys
import uuid
import pymysql
import csv
import rds_config


rds_host  = rds_config.rds_host
name = rds_config.db_username
password = rds_config.db_password
db_name = rds_config.db_name


logger = logging.getLogger()
logger.setLevel(logging.INFO)

try:
    conn = pymysql.connect(rds_host, user=name, passwd=password, db=db_name, connect_timeout=5)
except Exception as e:
    logger.error("ERROR: Unexpected error: Could not connect to MySql instance.")
    logger.error(e)
    sys.exit()

logger.info("SUCCESS: Connection to RDS mysql instance succeeded")

s3_client = boto3.client('s3')
```

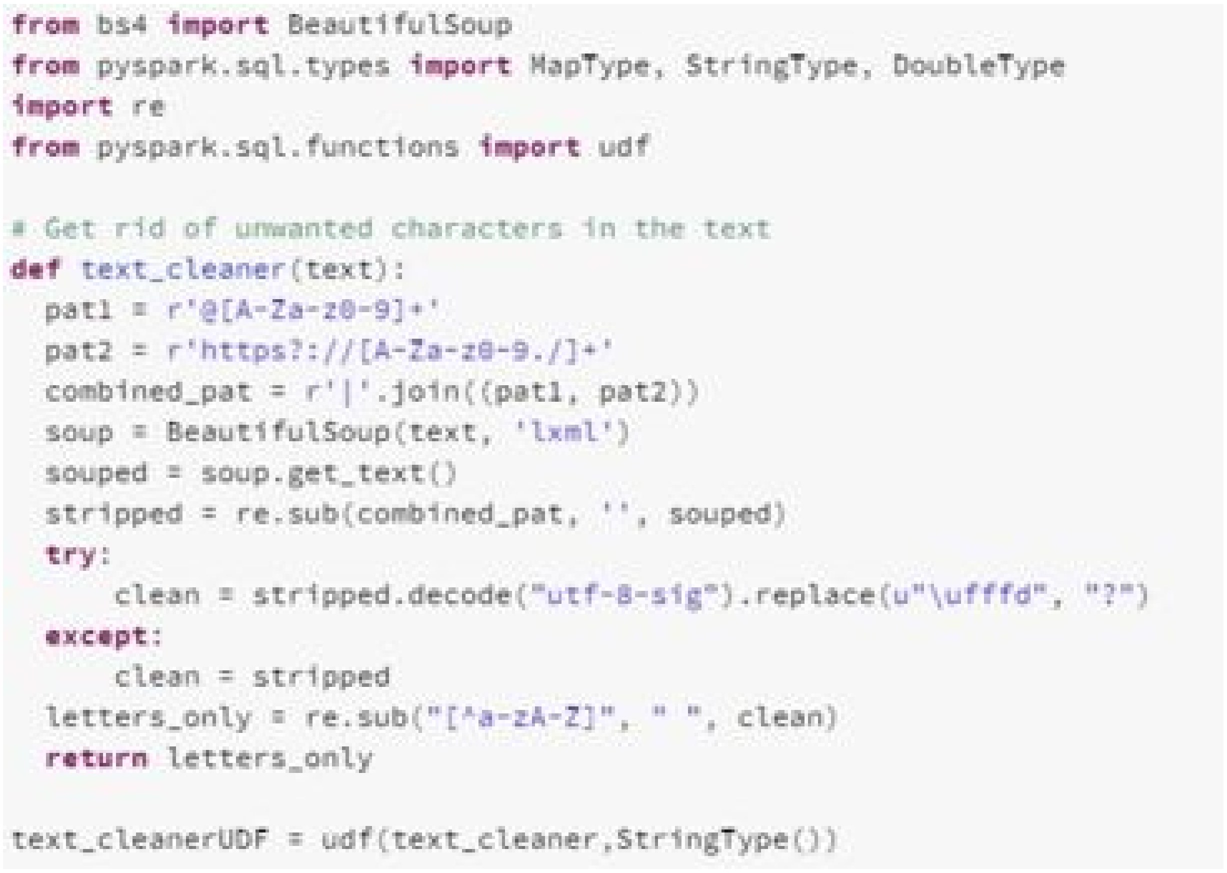


```python

def handler(event, context):

    bucket = event['Records'][0]['s3']['bucket']['name']
    key = event['Records'][0]['s3']['object']['key']
    download_path = '/tmp/{}{}'.format(uuid.uuid4(), key)

    s3_client.download_file(bucket, key,download_path)

    csv_data = csv.reader(file( download_path))

    with conn.cursor() as cur:
        for idx, row in enumerate(csv_data):

            logger.info(row)
            try:
                cur.execute('INSERT INTO target_table(column1, column2, column3)' \
                                'VALUES("%s", "%s", "%s")'
                                , row)
            except Exception as e:
                logger.error(e)
```

CLEANING

There are going to be two main data cleansing processes in this project, 1, Natural language processing, 2. financial reports.

- For the News data, NLP will be performed. The cleaning process will be as follows: removing the HTML tags, removing punctuation, removing stopwords, stemming, convert everything to lowercase, and finally text to vector conversion.
- The financial data will have to be in the form that mathematical calculations can be performed on it, this may include converting strings to floats, removing null values, and special characters.



```
from bs4 import BeautifulSoup
from pyspark.sql.types import MapType, StringType, DoubleType
import re
from pyspark.sql.functions import udf

# Get rid of unwanted characters in the text
def text_cleaner(text):
  pat1 = r'@[A-Za-z0-9]+'
  pat2 = r'https?://[A-Za-z0-9./]+'
  combined_pat = r'|'.join((pat1, pat2))
  soup = BeautifulSoup(text, 'lxml')
  souped = soup.get_text()
  stripped = re.sub(combined_pat, '', souped)
  try:
      clean = stripped.decode("utf-8-sig").replace(u"\ufffd", "?")
  except:
      clean = stripped
  letters_only = re.sub("[^a-zA-Z]", " ", clean)
  return letters_only

text_cleanerUDF = udf(text_cleaner,StringType())
```

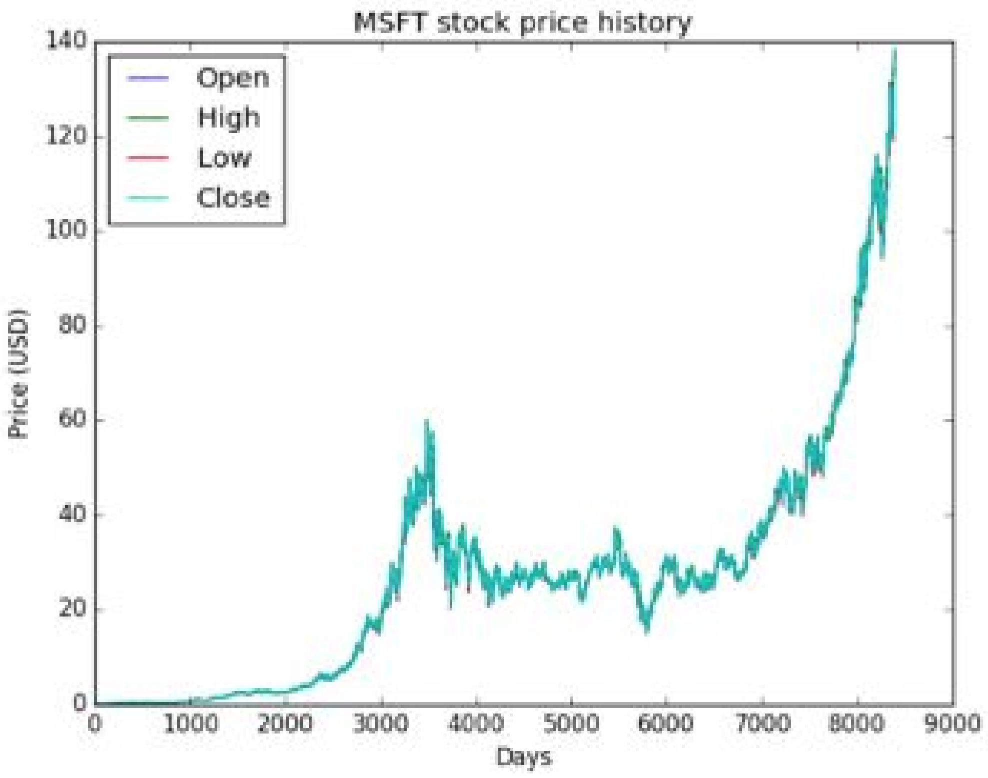
Data Mapping

It is essential for us to map all the data sources to each other and use it effectively, Some of the financial data is available on a quarterly basis and some economic data is available on a yearly basis so it's essential for us to map it to stock price historical company data which we have with us on a daily basis. We also had to do some transformations to map dates as each data source had different formats of data.

Initial Data Exploration –

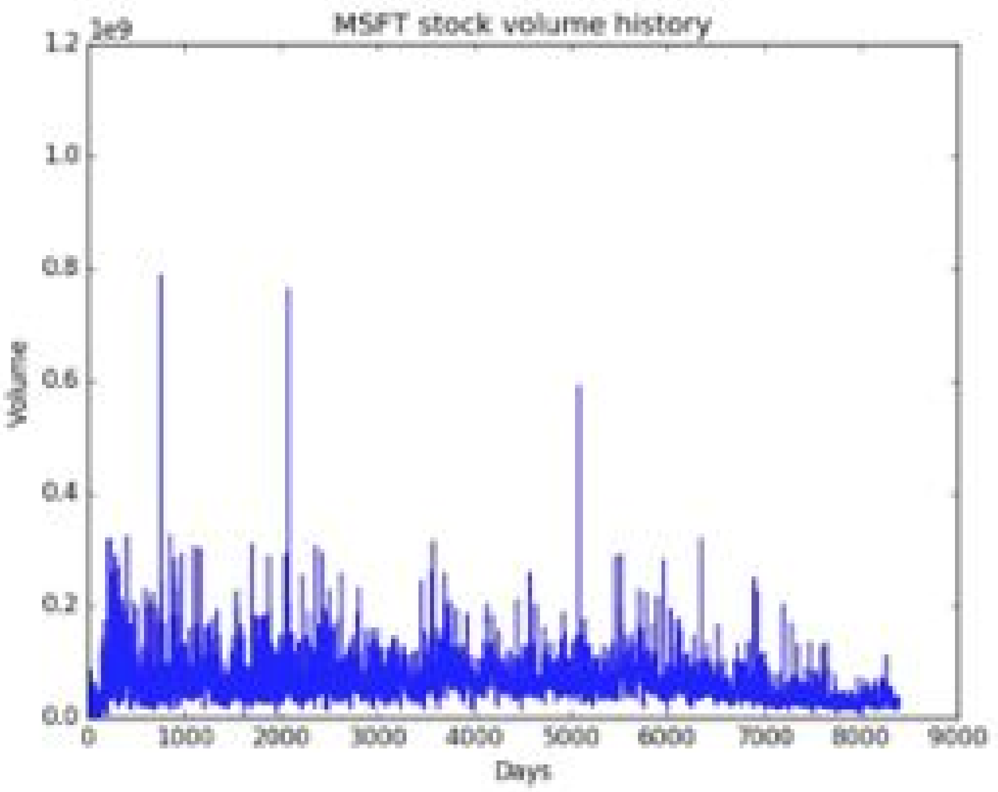
1 Stock Price History for Microsoft since 1986. (Price vs Days)

This indicates the price fluctuation of Microsoft over the years.



2. Stock Volume History for Microsoft since 1986 (Volume vs Days)

This indicates the number of shares been traded each day for Microsoft.

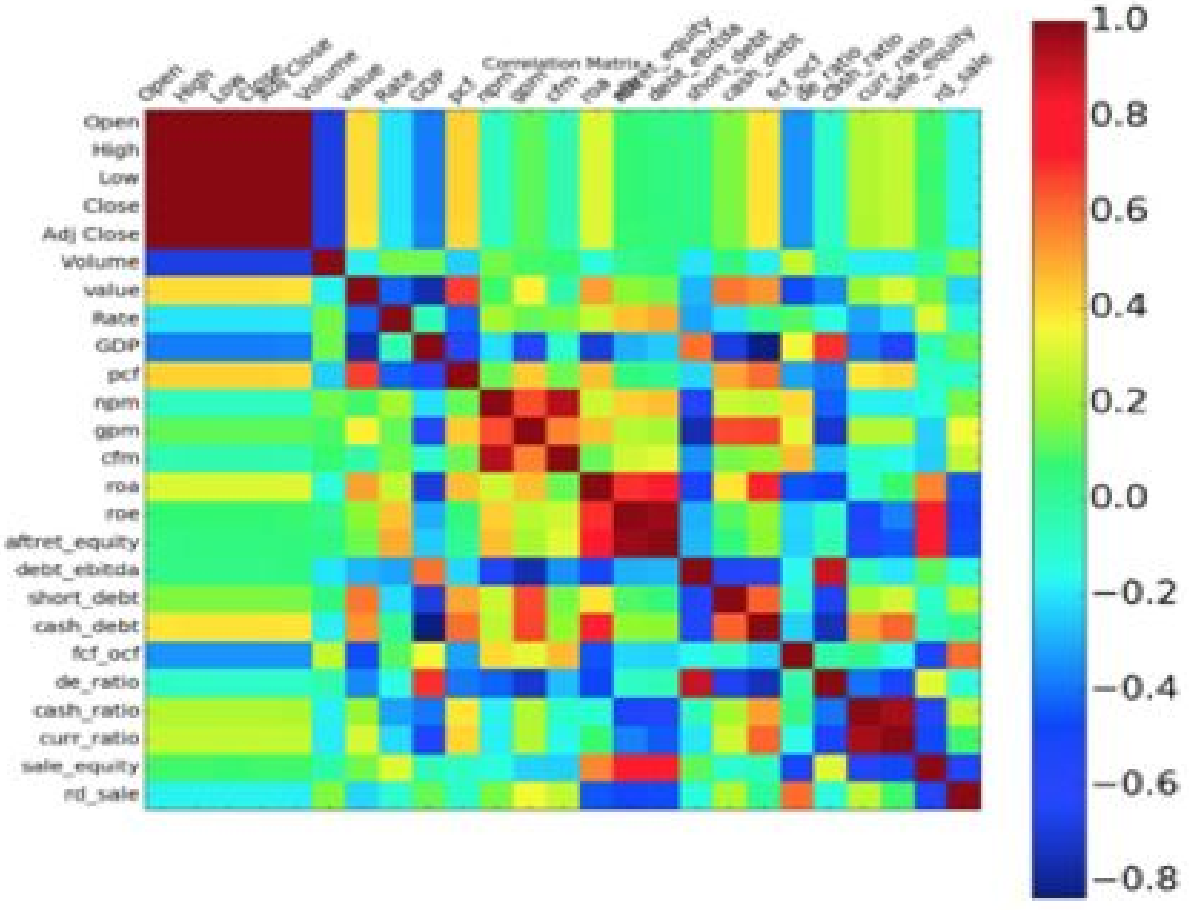


Features Initially – Open, Close, High, Low, Volume, GDP, Inflation Rate, Bench Mark Interest Rate, Dividend Payout Ratio, Dividend Yield, Price/Cash flow, Price/Sales, After-tax Return on Total Stockholders Equity, Gross Profit Margin, Net Profit Margin, Current Ratio, Payables Turnover, Receivables Turnover, Sales/Stockholders, Equity, Return on Assets, Return on Equity
Free Cash Flow/Operating Cash Flow, Cash Flow/Total Debt, Cash Flow Margin, Short-Term Debt/Total Debt, Total Debt/EBITDA, Total Debt/Equity, Cash Ratio, Research and Development/Sales

Feature Selection - Used to keep only variables that are useful for prediction. With combination of Historical data of Microsoft, Financial data of Microsoft and Economic Data of country over the years we had 32 features. Using Random Forest Model, we eliminated some features which weren't useful in our predicting factor "Open". These factors were making least impact to the R^2 value and were removed 'public_date','dpr','ps','pay_turn','rect_turn'.

Correlation Plot

Indicating some factors such as Cash Debt, Short Debt, Dividend Yield, ROA, Interest Rate were closely attached to the Open price of share.



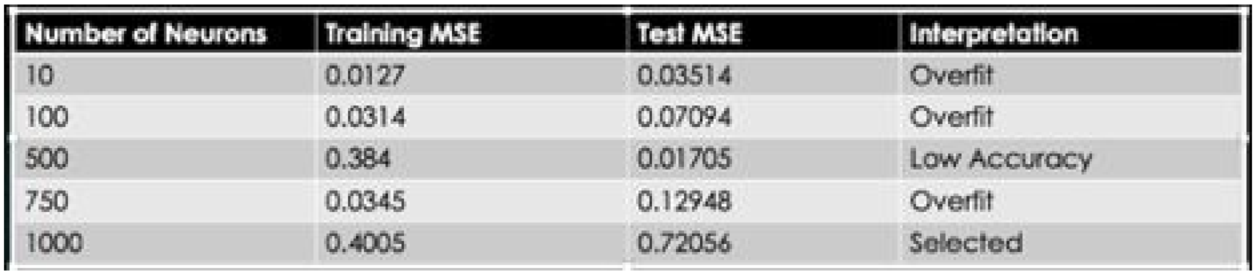
Machine Learning Models

LSTM (Long Short Term Memory) - "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. LSTMs were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. This was the reason to choose LSTM for stock price prediction.

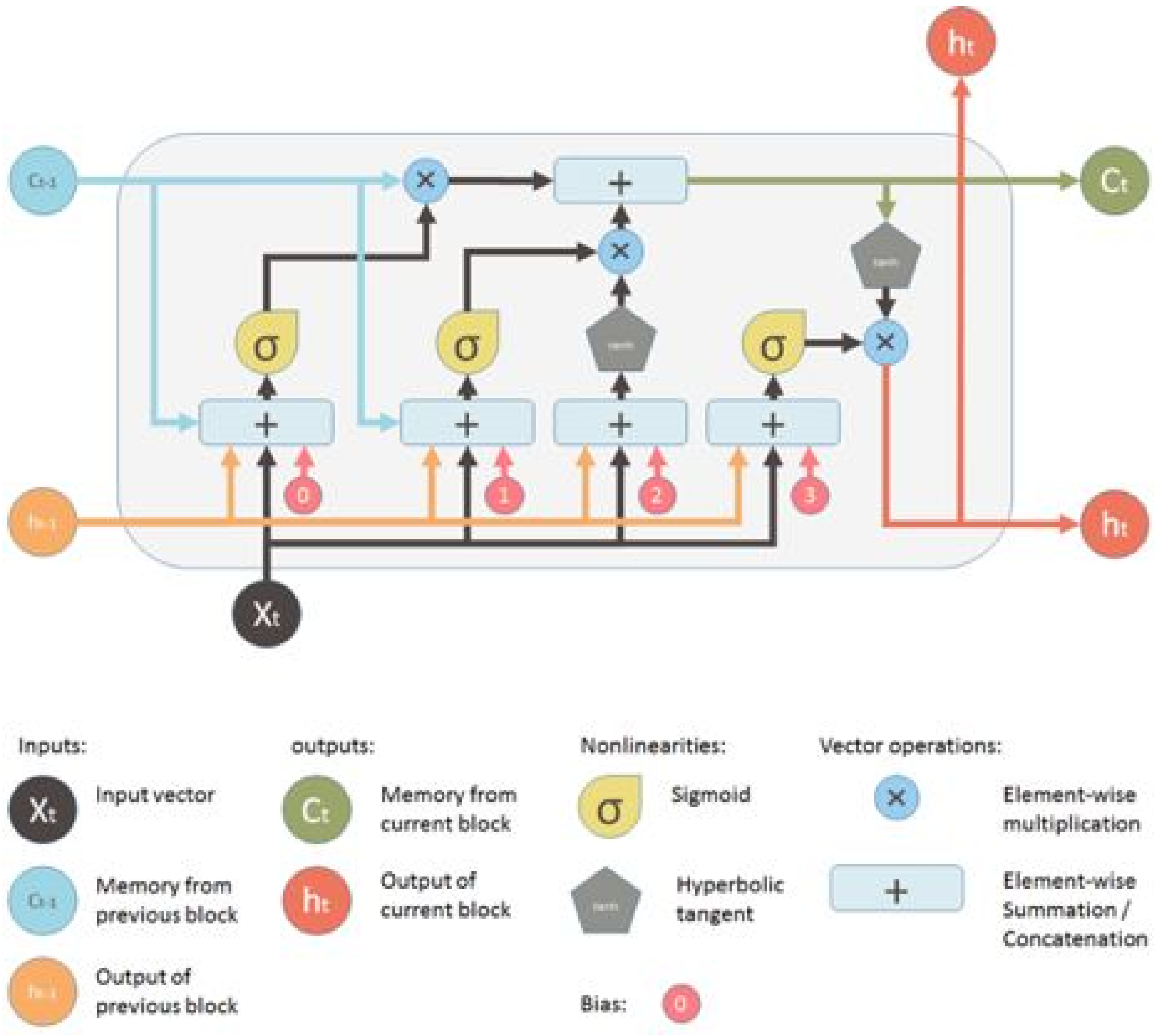
All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

Results of LSTM Model



Number of Neurons	Training MSE	Test MSE	Interpretation
10	0.0127	0.03514	Overfit
100	0.0314	0.07094	Overfit
500	0.384	0.01705	Low Accuracy
750	0.0345	0.12948	Overfit
1000	0.4005	0.72056	Selected

We executed our LSTM model with a range of number of neurons(10-1000) where we observed various Mean Squared Errors for training and test datasets. We observed that majority of the times our model was overfitted because of the really negligible MSE. We chose a model which we thought wasn't a overfit or a underfit.

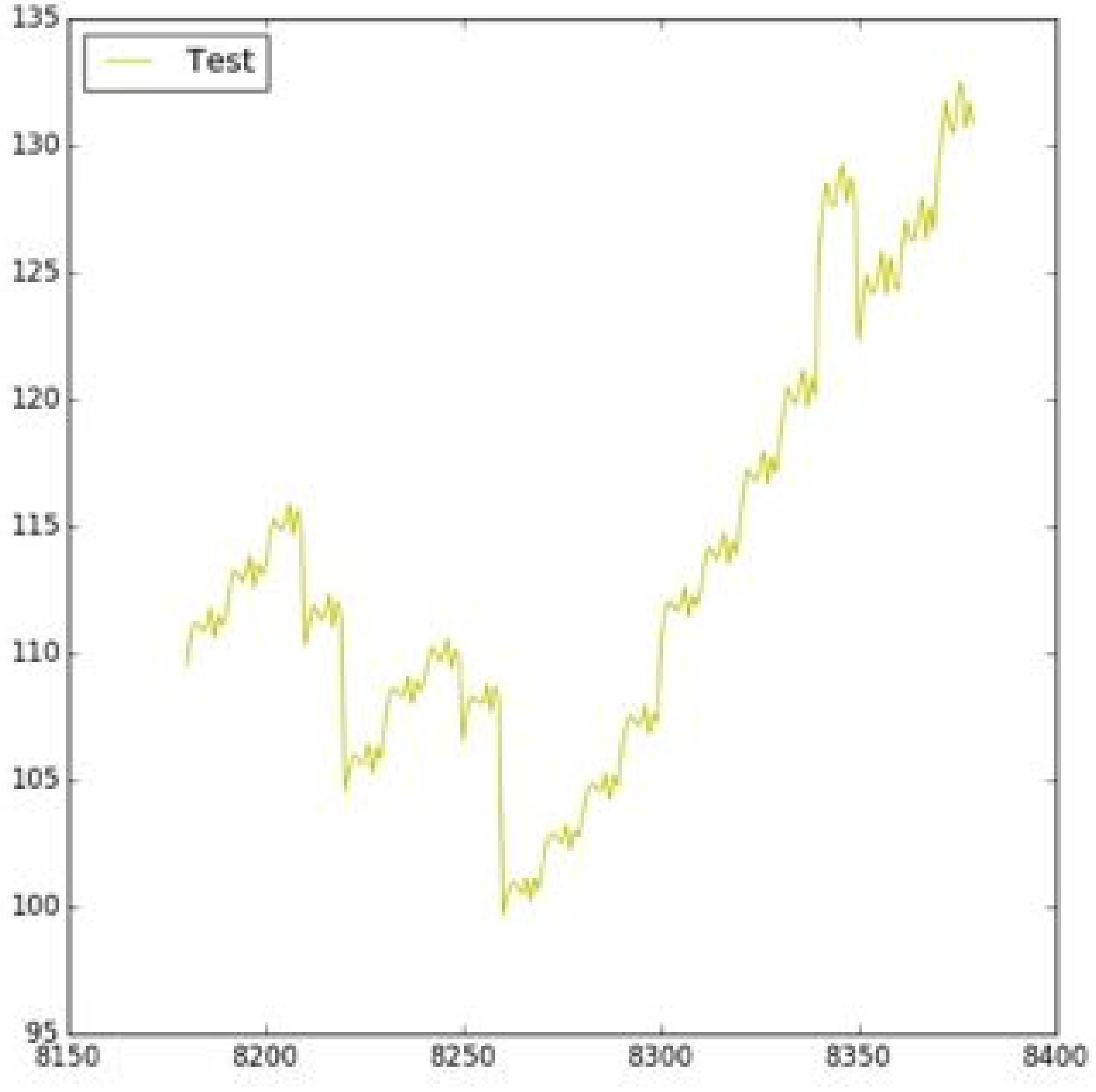


h_t
C_{t-1}
C_t
h_{t-1}
h_t
X_t
0
1
2
3
Inputs:
X_t Input vector
C_{t-1} Memory from previous block
h_{t-1} Output of previous block
outputs:
C_t Memory from current block
h_t Output of current block
Nonlinearities:
Sigmoid
Hyperbolic tangent
Bias:
Vector operations:
Element-wise multiplication
Element-wise Summation / Concatenation

(Architecture of LSTM Model)

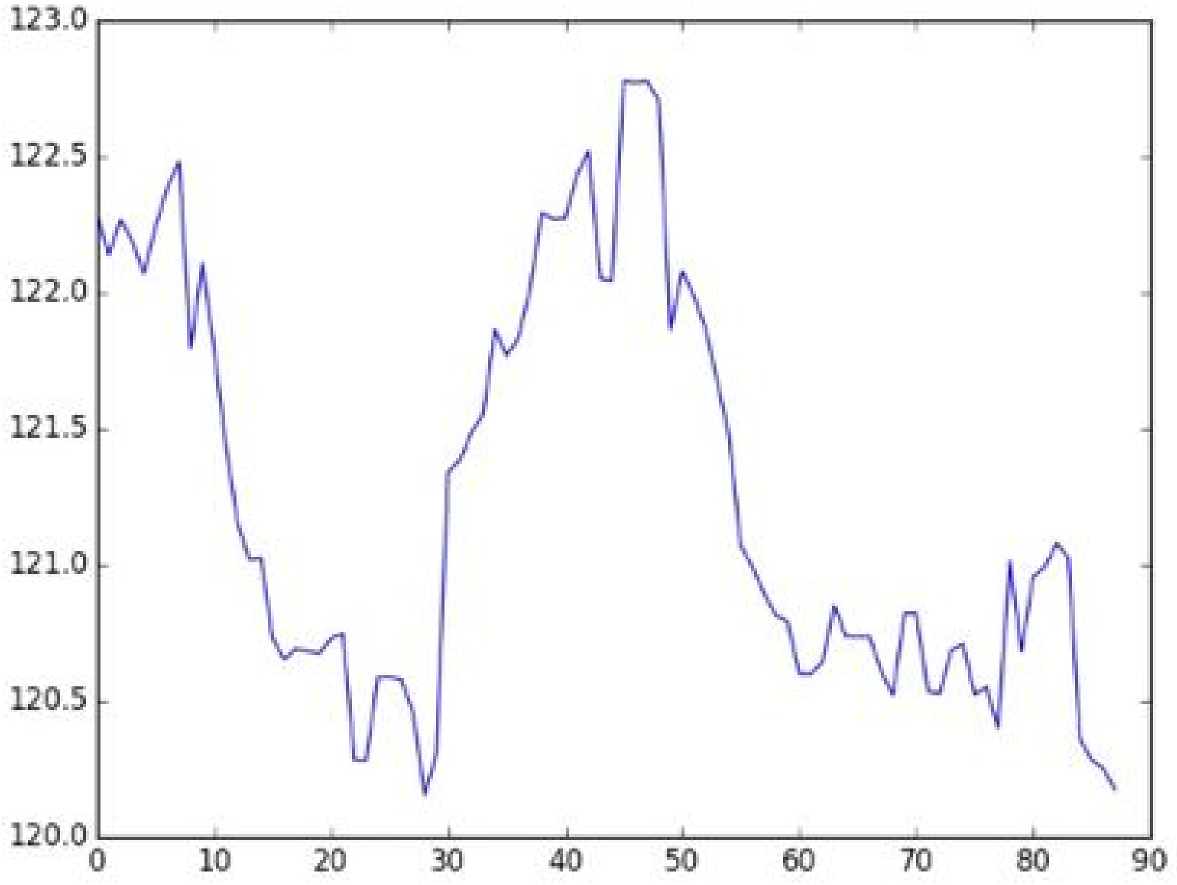
Visualizations for Predicted vs Actual Trends:

Predicted prices using only historical data:



Since overall the trends of prices have been on increasing side over a longer period of time, our algorithm predicts that the prices over the next 90 days are going to increase. Although this isn't sufficient so we predict using economic data and financial data along with historical data.

Predicted prices using historical + economic + financial data

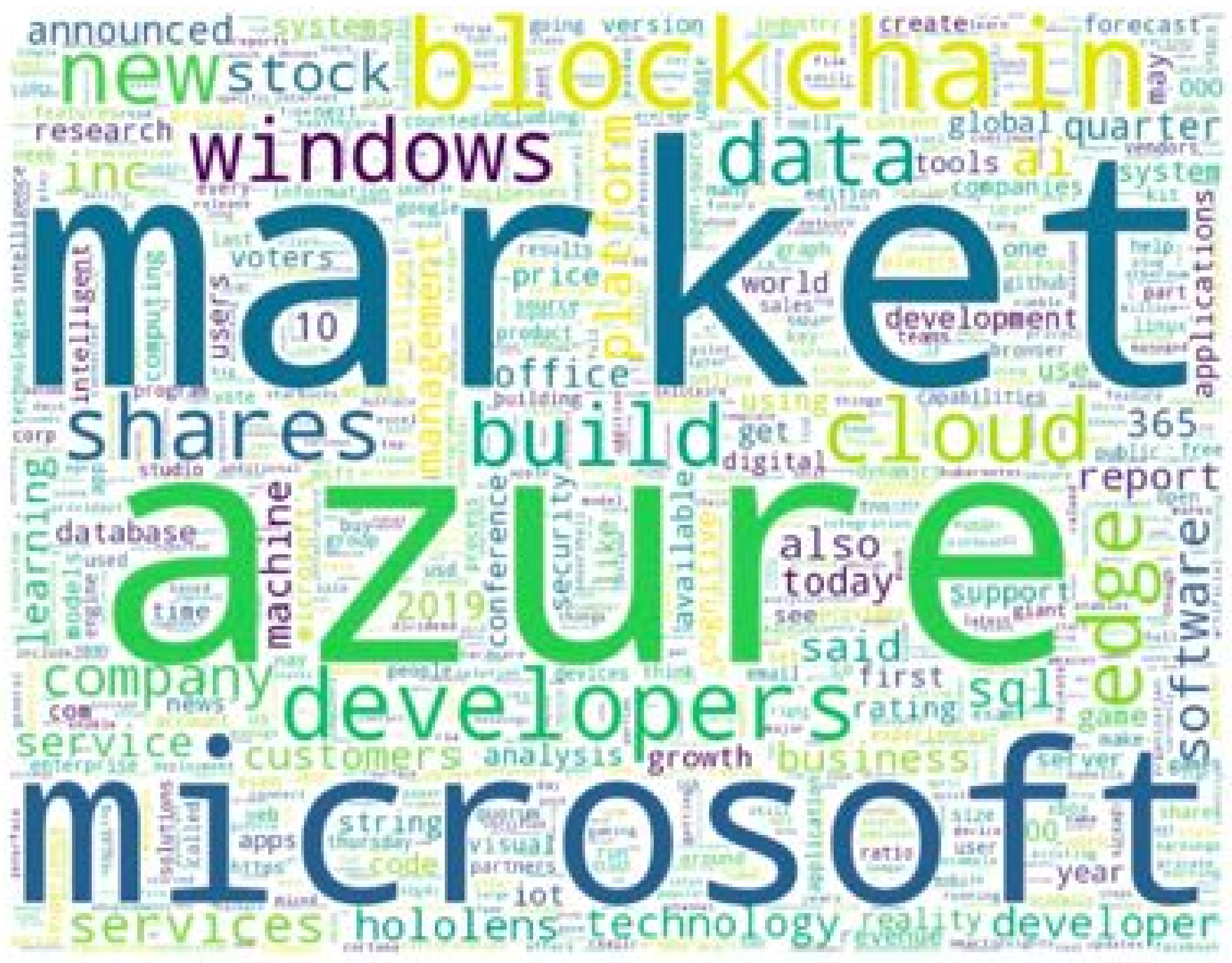


Although overall the trends of prices have been on increasing side over a longer period of time, but other economic data like GDP and Federal Reserve Rate are on lower side using this our algorithm predicts the prices over the next 90 days to be on the lower side and having a decline of about 3-4 dollars.

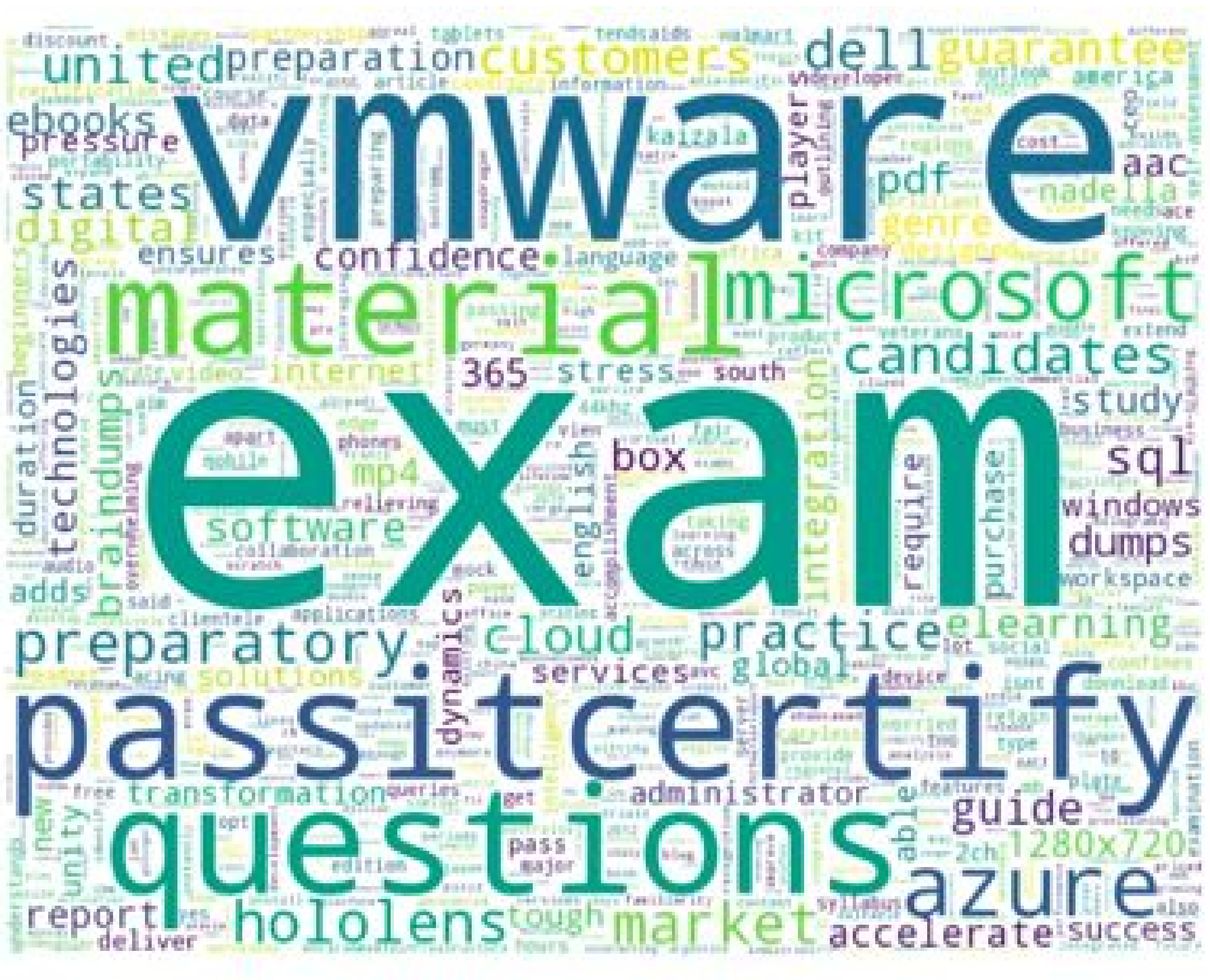
Natural Language Processing on News/Blogs related to Microsoft:

To perform a sentiment analysis to identify the sentiment the company is currently portraying in the market (positive or negative) we used Natural Language processing on news articles and blogs related to Microsoft in the past 2 months. We used NEWS API to fetch news articles and blogs from over 30,000 different sources to retrieve 57000+ news articles and blogs and then performed sentiment analysis on those articles using LDA.

We compared the opening and closing prices of Microsoft stock, mapped it with the date and the corresponding articles for that date and came up with the concept of positive or negative sentiment (if the difference was positive if corresponded to a positive sentiment and if the difference was negative it corresponded to a negative sentiment for the company).



 Following is the WordCloud generated using LDA for words corresponding to a negative sentiment (words that appear in articles when the stock closes on a low in comparison to the opening price):


vmware
exam
passitcertify
questions
material
microsoft
customers
dell
guarantee
united
preparation
ebooks
pressure
states
digital
pdf
aac
nadella
genre
confidence
ensures
candidates
technologies
braindumps
365
stress
study
internet
sql
box
mp4
software
english
integration
require
purchase
windows
dumps
workspace
adds
preparatory
cloud
practice
elearning
dynamics
solutions
services
global
transformation
administrator
guide
unity
azure
1280x720
report
hololens
market
tough
accelerate
success
deliver
duration
player
kaizala
language
video
south
america
discount
tablets
walmart
information
article
pass
major
free
features
2ch
syllabus
edition
collaboration
applications
clientele
phones
mobile

Conclusion

During the course of years, millions of dollars have been spent and thousands of people have been trying to predict the most accurate stock price trends. As per the current scenario, people have been able to predict the trends pretty accurately.

But the sheer amount of effort needed in the data sourcing, data transformation and ability to come up with the most significant dependent variables and then correctly use them to build an accurate and high performing model is only imaginable.

We were able to understand and deep dive into the real world scenario of data sourcing, data transformation and dealing with different kinds of data. We also were able to expand our horizon into the financial domain. We also gained a deep understanding of neural networks and how to implement them. We also experienced dealing with a large amount of unstructured data, implementing cloud solutions using AWS and also built effective presentation skills.