# Abstract & problem statement

This project aims to find a set of features, feature transformations and models that will best predict the direction of the Dow Jones Industrial Average (DJIA).

Established in 1885, the DJIA, is a price weighted stock index that measures the performance of the stocks of 30 large companies listed on stock exchanges in the United States with a market cap of US$6.56 trillion. It is one of the most followed indexes for equities and if often used in benchmarking portfolios, predicting recessions, US economy performance among other indicators.

We have explored the usage of multiple features with different transformation with multiple machine learning models in order to achieve the highest prediction accuracy for the DJIA index.

We modelled the problem as a classification problem with 2 signals, Buy and Sell with daily frequency. They are defined as follows:

* St = BUY if Pt+1 ≥ Pt
* St = SELL if Pt+1 < Pt

Where St is the signal at time t and Pt is the DJIA closing price at time t.

We explored the relationship between DJIA and multiple other assets in order to narrow down a few to work with and then applied various feature transformation to improve our predictions results.

Next, in this report we will describe our base set of features and justify as to why we decided to use them, then we will describe the time series analysis we conducted on our dataset, followed by some financial technical analysis feature transformations used in the project followed by the models that we decided to use and why we used them and finally we will then jump in the results. Due to large amount of results collected from various feature and feature transformation sets we will be dividing the results by the feature and feature transformations that produced them.

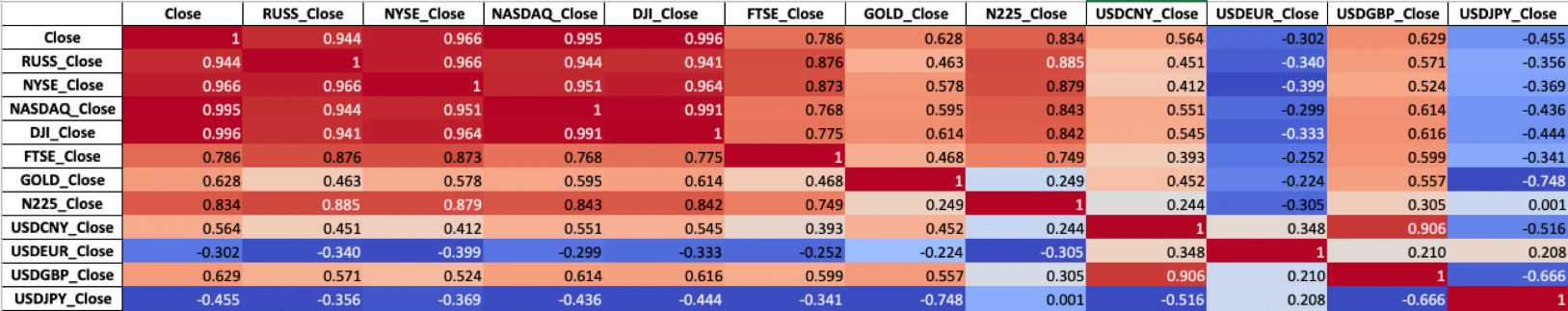
# The base features & feature transformation

In addition to the Open, High, Low and Close prices and daily Volume of the DJIA, we used daily Close prices other indexes, currencies as well as commodities. We used both 5 and 10 year historical data in an attempt to examine the effect on our prediction of simply adding more data with the same feature transformations and models.

\*\*EXPLAIN WHY THE FOLLOWING FEATURES ARE GOOD TO USE\*\*

The indexes used are NYSE, Russell, NASDAQ, S&P500, FTSE and N225. For commodities we used gold closing prices, for currency exchange rates we used the pairs USDCNY, USDEUR, USD GBP and USDJPY.

The below is a correlation matrix of our features against the Closing price of the DJIA



As the diagram above shows, there are strong correlations between the closing prices of the DJIA and the other assets. Clearly some assets are positively correlated and some are negatively which made use of as the report will later show. It is also worth noting that the above diagram is plotted based on the raw prices, which we later also transformed in order to make the data more uniform.

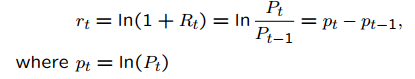
Our features set is not uniform in multiple ways. Not all of our features assets trade on the save trading venue which means that they are traded in different currencies e.g. N225 is a Japanese index and thus is traded in the Japanese Yen (JPY) while the FTSE is a British index that trades in GBP while the DJIA trades in USD. In order to take out the effect of the currencies in the prices we calculated the log returns of each of the assets and used them instead of the raw closing prices.

## Log returns \*\*REPHRASE THIS SECTION\*\* <KABIR>

Reference: <https://quantivity.wordpress.com/2011/02/21/why-log-returns/>

Given log returns are beyond the scope of this course, we will go ahead and explain it here.

Log returns are calculated as follows:



Where rt = return at time t, Pt is the price at time t

Benefit of using returns, versus prices, is normalization: measuring all variables in a comparable metric, thus enabling evaluation of analytic relationships amongst two or more variables despite originating from price series of unequal values. This is a requirement for many multidimensional statistical analysis and machine learning techniques.

Several benefits of using log returns, both theoretic and algorithmic.

First, log-normality: if we assume that prices are distributed [**log normally**](http://en.wikipedia.org/wiki/Log-normal_distribution) (which, in practice, may or may not be true for any given price series), then log(1 + r_i) is conveniently [**normally distributed**](http://en.wikipedia.org/wiki/Normal_distribution), because:

1 + r_i = \frac{p_i}{p_j} = \exp^{\log(\frac{p_i}{p_j})} 

This is handy given much of classic statistics presumes normality.

Second, approximate raw-log equality: when returns are very small (common for trades with short holding durations), the following approximation ensures they are close in value to raw returns:

\log(1 + r) \approx r , r \ll 1 

Third, time-additivity: consider an ordered sequence of n trades. A statistic frequently calculated from this sequence is the compounding return, which is the running return of this sequence of trades over time:

\displaystyle (1 + r_1)(1 + r_2)  \cdots (1 + r_n) = \prod_i (1+r_i)

This formula is fairly unpleasant, as probability theory reminds us the product of normally-distributed variables is not normal. Instead, the sum of normally-distributed variables is normal (important technicality: only when all variables are uncorrelated), which is useful when we recall the following logarithmic identity:

\log(1 + r_i) = log(\frac{p_i}{p_j}) = \log(p_i) - log(p_j) 

Thus, compounding returns are normally distributed. Finally, this identity leads us to a pleasant algorithmic benefit; a simple formula for calculating compound returns:

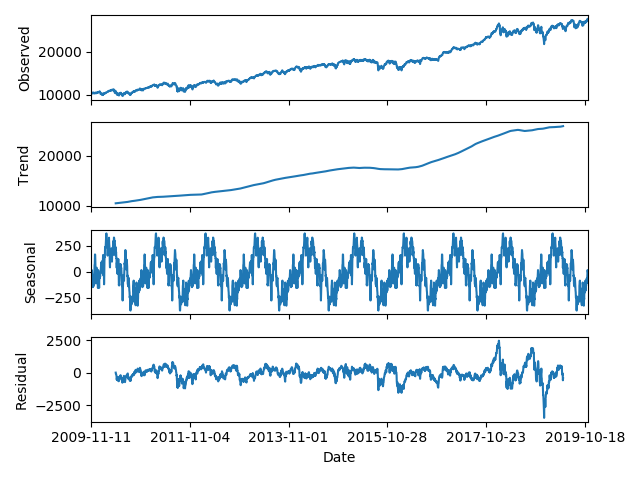
\displaystyle \sum_i \log(1+r_i) = \log(1 + r_1) + \log(1 + r_2)  + \cdots + \log(1 + r_n) = \log(p_n) - \log(p_0)

Thus, the compound return over n periods is merely the difference in log between initial and final periods. In terms of [**algorithmic complexity**](http://en.wikipedia.org/wiki/Big_O_notation), this simplification reduces O(n) multiplications to O(1) additions. This is a huge win for moderate to large n. Further, this sum is useful for cases in which returns diverge from normal, as the [**central limit theorem**](http://en.wikipedia.org/wiki/Central_limit_theorem) reminds us that the sample average of this sum will converge to normality (presuming finite first and second moments).

Most of the results we will show later used log returns instead of raw prices unless otherwise indicated.

# Time series analysis <DANISH>

Given that we are working with time series, we went ahead and visualized each of our features and the DJIA for a better understanding. The below is a diagram of the decomposed time series of the DJIA close price:



Next we will introduce some concepts that are beyond the scope of the course but were used for feature transformations in our project.

## Stationarity

## Autocorrelations and partial autocorrelation

## Seasonality

As we can observe, there is clearly a seasonal factor in the prices of the past 10 years of the DJIA.

# 

# Technical analysis & feature transformation <KABIR>

# The models\*\*ADD BRIEF DESCRIPTIONS OF EACH MODELS WITH DIAGRAMS\*\* <YUKI>

## Random Forest Classifier

## Adaboost Classifier

## Support Vector Machine

## K Nearest Neighbors

## Gradient Boosting

## Logistic Regression

## Forward Feed Neural Networks

## Long short-term memory

## Autoregressive integrated moving average

# Results \*\*SHOW RESULTS CATEGORICALLY\*\* <CLIFF>

# Conclusion \*\*FROM EACH RESULT CATEGORY PICK A WINNER MODEL AND DESCRIBE WHY IT WON\*\*<CLIFF>

# Teamwork

|  |  |
| --- | --- |
| **Member** | **Contribution** |
| Danish Alsayed |  |
| Yuki Ng |  |
| Cliff Ng |  |
| Kabir Rajput |  |