CS6140 Machine Learning

Miniproject report

Due: 12 December 2022

Inter-Market Relationship and Alpha Portfolio construction

Team members:

Venkateshwaran Sundar - sundar.ve@northeastern.edu

Sai Rahul Vaddadi - vaddadi.s@northeastern.edu

Naren Mohan - mohan.na@northeastern.edu

Objectives and Significance:

Artificial Intelligence (AI) techniques are being increasingly deployed in finance, in areas such as asset management, algorithmic trading, credit underwriting, or blockchain-based finance, enabled by the the abundance of available data and affordable computing capacity.

There is much to gain from being able to forecasts the futures of these instruments which may *beat the market*. Millions of dollars is poured into funding research to determine reliable methods to achieve high returns while limiting the risk involved. In fact, there have been research studies that studies the relationship between the price movements of stocks and the planetary movements. But what we are trying to analyse in this project is to utilize the **inter-market relationship** between various classes of assets. This relationship is also formulated as *alpha* α that determines the price movement of the assets.

We will look at how Machine learning can be used to identify the inter-market relationship features to forecast the prices of each assets. Based on these forecasts, we will be building a optimizing model that would recommend the best set of assets to invest given the historical data. The historical data is obtained from the publicly available data sets to train and derive the inter-market relationship identify and thus predict the alpha factors of our own. This would be quantifiably compared and backtested to validate the model performance with how the market actually performs. There is existing research(Yahoo article, Staying the Course: The Impact of Investment Style Consistency on Mutual Fund Performance) that indicates that *Alpha* is particularly important because it tends to be persistent.

Once we validate claims of persistence of these factors over the last 10 years we will predict values of α for the future. We will attempt to use different machine learning model creation techniques to determine an optimal model, and then use this model to recommend the best set of assets to invest at the given point of time.

Our objective is to build portfolios for investors by generating maximal returns with a minimal risk tolerance using machine learning techniques. By leveraging publicly available data sets with pre-defined input factors, we predict average excess returns for each equity. We then make use of optimization models to construct optimal portfolio taking into account the risk tolerance. Finally, we like to evaluate our portfolio by comparing it with market values of a benchmark index such as *S&P 500*, *NYSE*, *NASDAQ*.

Important Concepts and Background:

Portfolio - A portfolio is a collection of financial investments like stocks, bonds, commodities, cash, and cash equivalents, including closed-end funds and exchange traded funds (ETFs).

Alpha - α is a measure of the active return on an investment, the performance of that investment compared with a suitable market index. An alpha of 1% means the investment's return on investment over a selected period of time was 1% better than the market during that same period; a negative alpha means the investment under-performed the market. Alpha, along with *beta*, is one of two key coefficients in the capital asset pricing model used in modern portfolio theory and is closely related to other important quantities such as standard deviation, R-squared, and the Sharpe ratio.

Beta - β often referred to as the beta coefficient, is an indication of the volatility of a stock, a fund, or a stock portfolio in comparison with the market as a whole. A benchmark index (most commonly the S&P 500) is used as the proxy measurement for the market. Knowing how volatile a stock's price is can help an investor decide whether it is worth the risk. The baseline number for beta is one, which indicates that the security's price moves exactly as the market moves. A beta of less than 1 means that the security is less volatile than the market, while a beta greater than 1 indicates that its price is more volatile than the market. If a stock's beta is 1.5, it is considered to be 50% more volatile than the overall market.

Markowitz Portfolio Selection - The Markowitz model is a single-period model, where an investor forms a portfolio at the beginning of the period. The investor's objective is to maximize the portfolio's expected return, subject to an acceptable level of risk (or minimize risk, subject to an acceptable expected return). The assumption of a single time period, coupled with assumptions about the investor's attitude toward risk, allows risk to be measured by the variance (or standard deviation) of the portfolio's return. The best that an investor can do is known as the efficient frontier.

Capital Asset Pricing Model - This model assumes that investors use the logic of Markowitz in forming portfolios. It further assumes that there is an asset (the risk-free asset) that has a certain return. With a risk-free asset, the efficient frontier in Markowitz is no longer the best that investors can do. It relies on the equation

$$E[R_j] = R_f + \beta_j(E[R_m] - R_f)$$

where, $E[R_j]$ and $E[R_m]$ are the expected returns to asset j and the market portfolio, respectively, R_f is the risk free rate, and β_j is the beta coefficient for asset j.

Existing work : Ma, Y. et al. (2021) applied a two-stage process for portfolio construction by using different machine learning and deep learning based models. Initially, they performed stock selection by predicting the expected returns based on historical data of 9 years from 2007 to 2015 of component stocks from China securities 100 index. Finally, they utilized these results in advancing mean–variance (MV) and omega portfolio optimization models for optimizing portfolio. Yu et al. (2020) advanced six portfolio optimization methods by combining the projections from the ARIMA model (i.e., MV, MAD, DSR, LVaR, CVaR and omega models). Prior to extending these portfolio optimization methods, they first employed the ARIMA model to forecast future stock return. Experimental results shown that extended MV and omega models with ARIMA prediction outperformed single portfolio models and outperformed advanced portfolio optimization models with ARIMA prediction. LSTM neural networks, along with the equal weighted method, Monte Carlo simulation, and MV model, were used by Ta et al. (2020) to construct portfolios. Additionally, they used SVM and linear regression as comparisons for choosing stocks. According to experimental findings, LSTM neural networks were more accurate predictors than SVM and linear regression, and their built-in portfolios outperformed others.

Implementation: We have day-to-day data on equities, bonds, currencies, and commodities from 1997 to 2021. Since some commodities and currencies are not traded on all days, we've done some preprocessing on data to include data only All prices are in USD. *Yahoo Finance* - Yfinance offers an excellent range of market data on stocks, bonds, currencies, and cryptocurrencies. It also offers market news, reports, analysis, and additional options - setting it apart from some of its competitors.[6]

Yfinance is available as a Python package and can be easy to use to fetch data on stocks and commodities. Here's a snippet of the list of commodities and stocks we've fetched data for and what it looks like.

Equities to collect

Tech stocks

1. Microsoft MSFT
2. Amazon AMZN
3. Meta META
4. Alphabet GOOG

Healthcare stocks

1. Pfizer PFE

Consumer stocks

1. Coca cola KO
2. Walmart WLMT
3. Costco COST
4. AT&T T
5. Dollar General Corp

Financials

1. PayPal PYPL
2. Berkshire Hathaway BRK.A
3. Bank of America BAC

The following example code contains tickers of each of the stocks and commodities mentioned above among more tickers.

```
stocks_tickers = ["MSFT", "AMD", "AMZN", "GE", "AAPL", "PFE", "KO", "WMT", "COST", "T", "CMCSA", "PEP", \
    "KO", "UL", "BAC", "JPM", "GS", "MS", "NOC", "LMT", "BA", "F", "LUV", "SLB", "NEE", "MCD",
        "SBAC", "AMT", "JNJ"]
index_tickers = ["^GSPC", "^DJI", "^IXIC", "^NYA", "^RUT", "^N100"]
currency_tickers = ["JPY=X"]
bonds_tickers = ["^IRX", "^FVX", "^TNX"]
mf_tickers = ["FDCAX", "GMCFX", "FRESX", "FRRSX", "FSLBX", "VDIGX", "SGGDX", "OPGSX"]
comm_tickers = ["CL", "HE", "B", "NBP", "GL"]

tickers = stocks_tickers + index_tickers + currency_tickers  + bonds_tickers + mf_tickers + comm_tickers
```

```
period = "2000-01-01"

df = yf.download(tickers, start="2000-01-01")
df["Adj Close"]
```

The following dataframe depicts sample data collected from Yfinance. This is a time series data, where every column is the ticker of a stock/commodity being traded. The values in dataframe are the adjusted closing price of every stock price on the given date.

Out[83]:

Date	AAPL	AMD	AMT	AMZN	B	BA	BAC	CL	CMCSA	COST	...	WMT	^DJI	^FVX
2000-01-03	0.851942	15.500000	23.893417	4.468750	4.890864	25.940285	13.443506	19.092175	11.772097	30.975521	...	44.220287	11357.509766	6.457
2000-01-04	0.780115	14.625000	23.638689	4.096875	4.777997	25.899942	12.645577	18.593782	10.890766	29.278828	...	42.565632	10997.929688	6.396
2000-01-05	0.791530	15.000000	24.351927	3.487500	4.740375	27.513643	12.784347	17.961220	10.607478	29.779150	...	41.696960	11122.650391	6.489
2000-01-06	0.723033	16.000000	24.097200	3.278125	4.702754	27.796045	13.877175	17.922878	11.016667	30.377323	...	42.151985	11253.259766	6.450
2000-01-07	0.757282	16.250000	25.421778	3.478125	4.665131	28.602890	13.512897	18.996332	10.670428	32.389442	...	45.337154	11522.559570	6.397

Exploratory analysis:
The data as fetched from Yfinance had many dates for which values were invalid. While we attempted steps of extrapolation to fill in missing data we realized that this was leading to a bias, we've instead dropped all data points with null values. Since the goal was to determine the relationship between different financial instruments whose prices we have already loaded on a dataframe, we began my determining relationship between multiple variables in the time series.[7]

Granger causality is an econometric test used to verify the usefulness of one variable to forecast another. It can also be used to test the null hypothesis that the coefficients of past values in the regression equation are zero. So, if the p-value obtained from the test is lesser than the significance level of 0.05, then, you can safely reject the null hypothesis. This has been performed on the original data set.
Following is an excerpt of all the p-values which were determined from the test.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1		^IRX	^FVX	^TNX	FDCAX	GMCFX	FRESX	FRRSX	FSLBX	VDIGX	SGGDX	OPGSX	CL	HE	B	NBP	GL
2	AAPL	0.953623	0.731127	0.76569	0.103387	0.434168	0.007069	0.055264	0.33863	0.029869	0.059414	0.124079	0.000116	0.005498	0.015635	1.57E-08	2.96E-05
3	PFE	0.984367	0.604839	0.717764	0.002916	0.146905	0.219634	0.467003	0.004705	0.001721	0.45571	0.368371	0.052925	0.068377	0.229286	0.517292	0.001315
4	WMT	0.039282	0.18998	0.226382	0.005353	0.057234	0.026124	0.027874	0.050003	0.000691	0.102762	0.115676	0.023701	0.166885	0.002846	0.105015	0.056769
5	COST	0.094629	0.056945	0.024965	0.026977	0.091663	0.008933	0.0143	0.065392	0.002895	0.533887	0.380557	0.009625	0.042213	0.084359	3.53E-06	0.017112
6	T	0.010761	0.25936	0.206689	0.026828	0.179105	0.576762	0.838954	0.011733	0.003641	0.021035	0.016397	0.368762	0.028927	0.640229	0.526853	0.002083
7	CMCSA	0.275274	0.437257	0.696494	0.007821	0.037917	0.238071	0.23392	0.366479	3.91E-05	0.416702	0.531969	0.076022	0.002395	0.146534	0.016233	0.017598
8	PEP	0.091052	0.028937	0.018728	0.000485	0.088798	0.167374	0.365903	2.65E-05	0.029399	0.489214	0.580252	0.092213	0.109214	0.058751	0.000114	0.000722
9	KO	0.352318	0.102889	0.072257	2.63E-06	0.002247	0.409562	0.561049	7.56E-05	0.001053	0.666203	0.376084	0.115389	0.088527	0.069329	0.208924	0.005001
10	UL	0.102527	0.138413	0.082625	0.06239	0.977099	0.123023	0.291368	0.012762	0.22628	0.358843	0.335438	0.000339	0.102305	0.003319	0.000792	0.001697
11	BAC	2.54E-29	0.104574	0.571466	0.02833	0.168739	0.175393	0.195554	0.046575	0.116258	3.35E-07	9.92E-08	0.348671	0.709573	0.423371	0.998826	0.004396
12	JPM	8.97E-11	0.658516	0.757372	0.561894	0.031907	2.61E-05	0.000195	0.291654	0.030614	2.23E-05	3.72E-07	0.139334	0.297994	0.650536	0.209341	0.000469
13	GS	3.30E-06	0.484271	0.49561	0.013051	0.661172	0.078483	0.081553	0.033129	0.036968	0.015862	0.001161	0.073434	0.249454	0.092054	0.325577	0.013017
14	MS	0.1721	0.44885	0.532103	0.581585	0.850949	0.709024	0.753587	0.783638	0.918603	0.010168	0.002032	0.229793	0.978277	0.973954	0.970123	0.083391

From the list of p-values generated we can see that while there are some p-values that are below the threshold, most of the features are not very dependent on past values and other features.

More about p-values:

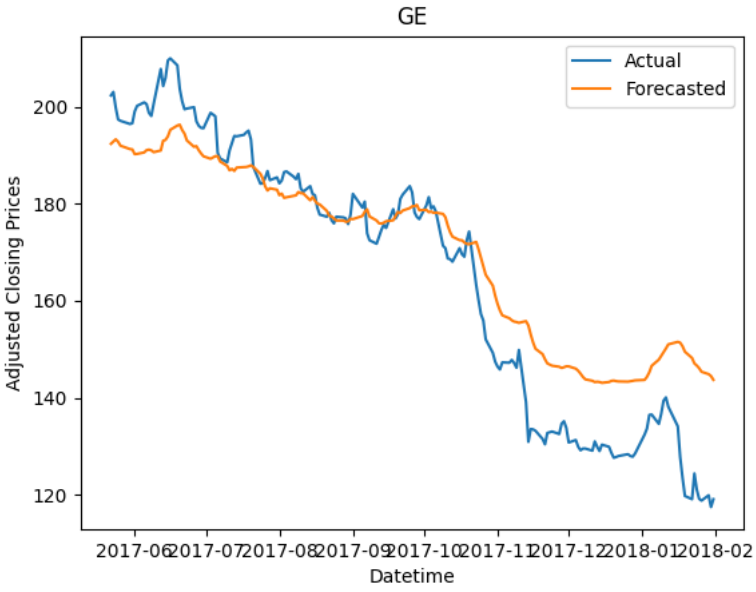
Model selection: We are dealing with multivariate time series data, with some research that one model apt to train such data is a vector autoregressive (VAR) model. [8] For the purposes of this project, we decided to make predictions by taking into account a lag period of 14 days. The VAR model is a workhouse multivariate time series model that relates current observations of a variable with past observations of itself and past observations of other variables in the system. VAR models differ from univariate autoregressive models because they allow feedback to occur between the variables in the model. When we trained this model on our data, we realized that the accuracy wasn't as high as we'd liked. It appeared like the data we had may have required a model which could draw a complex decision boundary. Furthermore, AutoRegressive models require 3 hyperparameters to be tuned namely p(dependency between current and past values), d(stationarity), and q(linear function of past errors) followed by assumption on the stationarity of our data, to work[8].

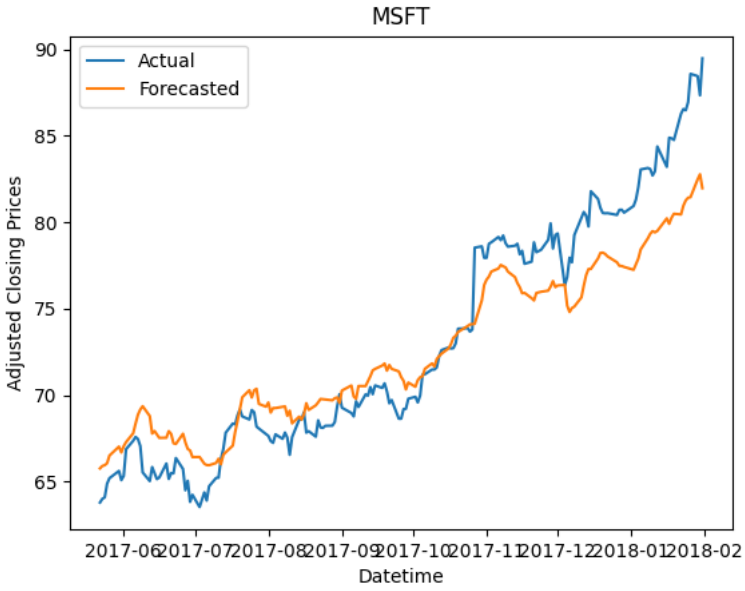
We later switched to a *Long Short-Term Memory Neural network*, a type of recurrent neural network capable of learning order dependence in sequence prediction problems[9]. With the LSTM model, we

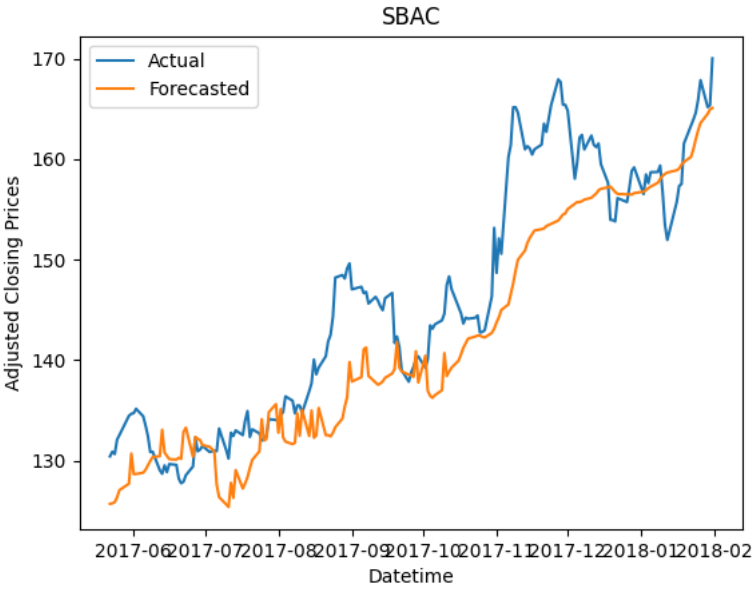
were able to better forecast stock prices.

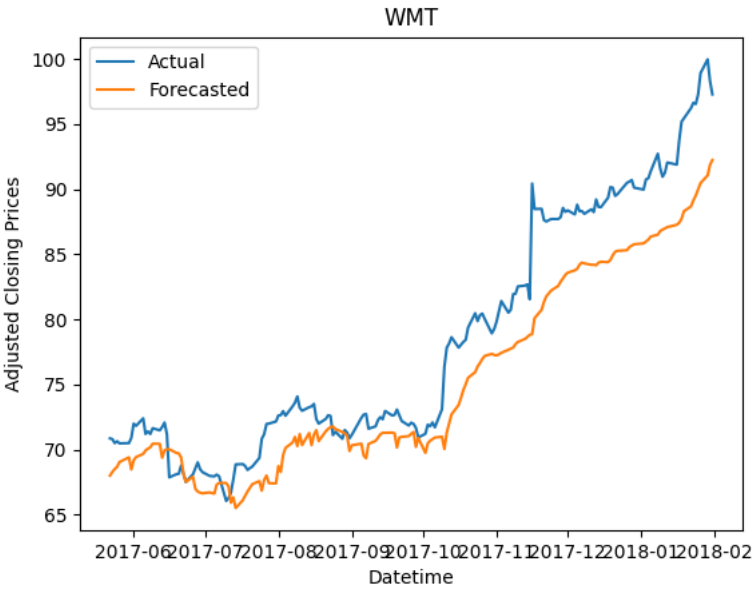
Once forecasts on stock prices were made successfully by the mode, we were able to calculate alpha factors for the stocks which then could be used to estimate a list of stocks that will likely perform better than the market.

Results: Present results and findings Describe experiments done for each figure and table Provide headers for the table and make it easy to understand.
The forecasts from LSTM model were accurate for many stocks, however, there were a handful of stocks where the model failed to make a satisfactory prediction. Here are few examples of stocks prices where the forecasts where accurate:

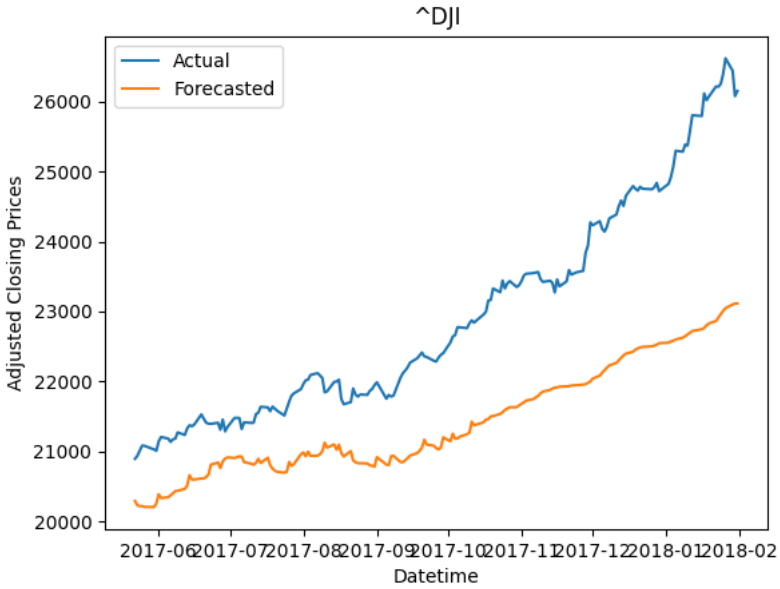


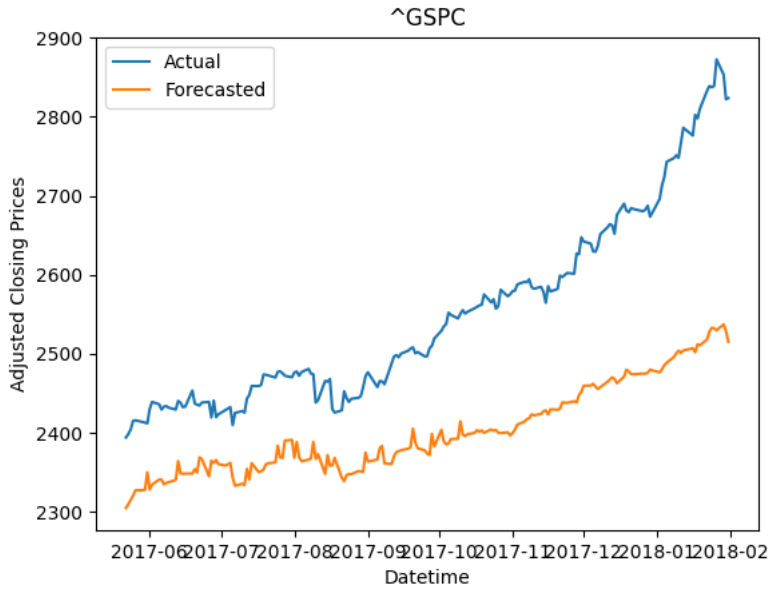


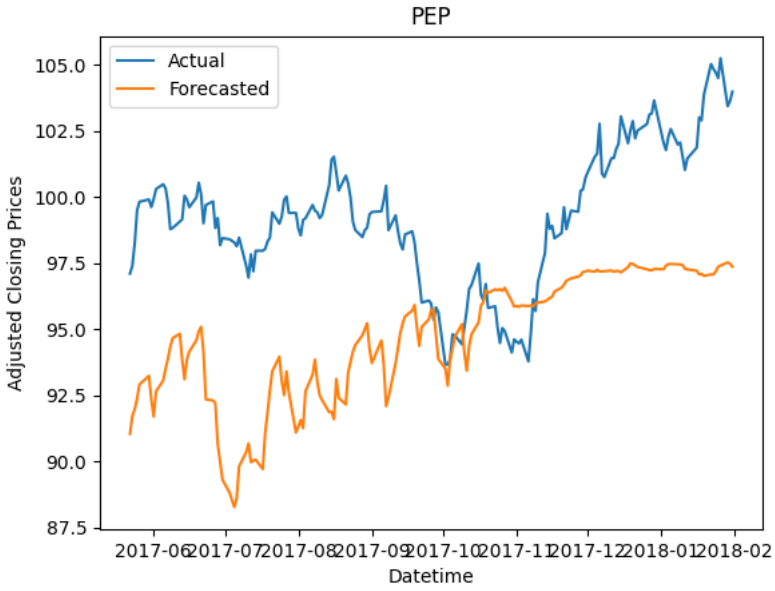

SBAC
Actual
Forecasted
Adjusted Closing Prices
170
160
150
140
130
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
Datetime


WMT
Actual
Forecasted
Adjusted Closing Prices
100
95
90
85
80
75
70
65
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
Datetime

There are examples for which the stock prices predictions weren't satisfactory:







The entire list of scatter plots for each forecast made are submitted along with the project.

Computing the Alpha Factor:

Once the forecasts are made using the deep learning model, we then go ahead to compute the alpha factor. Investments. This alpha factor is a number that signifies the potential returns of a stock from the current time point accounting for the risks in investment as well as for the fluctuations in the forecasts. Alpha factor would be computed for all the instruments individually that has been considered for forecasting. This allows it to be compared and classified based on the risk threshold that the user would want to consider. The procedure for computing the alpha factor is as follows –

Compute 7-day period SMA of the forecasts Get the maximum return that one could get as per the smoothened forecast

$$maxreturn = argmax(SMAforecastedprices)$$

Get the significance of the forecast by fitting a straight line using linear regression and computing its R squared Compute alpha -

$$alpha = ((maxreturn/currentPrice) - 1) * R^2$$

The first step focuses on computing the SMA with a 7-day period to smoothen the forecasts. This prevents abrupt fluctuations in the forecasts made by the machine learning models. And then the maximum of the SMA smoothed forecast value is taken to be the maximum return price that one could get out of the considered instrument. Then a straight line is fit using linear regression on the smoothened forecasts. Once the line has been fit, the R squared value is considered which acts as the confidence level of the forecasts. Then using the formula mentioned above, the alpha factor is computed. This formula multiplies the returns in percent with the R-squared value of the fit to get the alpha factor.

Stock recommendation:
Once the alpha factors for all the instruments are computed, recommendations can be performed using it. The user can input the risk threshold parameter based on which the stock recommendations would be made. The risk threshold parameter serves as how aggressive the returns are to be expected. It is similar to the terminology used in Mutual Funds where higher risk equals higher returns, but

the returns are not guaranteed and lower risk equals lesser returns but is more secure. Similarly, the risk threshold parameter would allow the recommender to suggest instruments that it has the most confidence in if the lesser risk is given as input and more instruments if the higher risk is given as input. The recommender function is the ultimate output of the entire project that gives out the list of instruments and its corresponding confidence levels.

```
recommender(df_pred, 0.5)
```

```
{'AMZN': 0.6139771795813136,
 'JPM': 0.5831290700007775,
 'BA': 0.5710359822236253}
```

Conclusion We started with the idea of determining if there exists a relationship between different financial instruments, if so how can we leverage this information to make predictions on stock prices? We find that there indeed exists an Inter-market relationship, the prices of commodities affect stock prices of the day. We then train a model to be able to determine the relationship between commodities and stock prices so that we can effectively forecast stock prices from this model. The results from this model were useful to then calculate alpha factors. While in reality, the calculation of alpha is more involved, we have used a simplified version of the formula for this project. Future improvements on this project will be to more accurately calculate values of alpha. Furthermore, we've only considered adjusted closing rates, and dropped all invalid values, with more domain knowledge an educated guess could be made on how to handle missing data. There is also room to include more features in the dataset since there are a plethora of external factors which affect stock prices. In conclusion, this project makes a jab of determining inter-market relationships and using insights from these relationships to predict high-performing stock prices. While our model isn't near ready to be deployed in a production environment, the results show that there is scope for an improved version of this model.

Individual tasks:

Sai Rahul Vaddadi Performed analysis on data from yfinance, determined causality, and trained the LSTM model to predict stock prices

Naren Mohan Used forecasts made from LSTM model to calculate Alpha factors. These alpha factors were used to estimate stocks that will likely perform well in the market.

Venkateshwaran Sundar Performed analysis on sources stocks and commodity prices. Collaborated with team members on methods to solve obstacles and collated all results to write up the project report.

References:

1. Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. Expert Syst. Appl., 165, 113973.
2. Yu, J., Chiou, W.P., Lee, W., & Lin, S. (2020). Portfolio models with return forecasting and transaction costs. International Review of Economics & Finance, 66, 118-130.
3. Ta, V., Liu, C., & Tadesse, D.A. (2020). Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading. Applied Sciences.
4. https://www.investopedia.com/articles/fundamental-analysis/09/intermarket-relations.asp
5. Jensen, L.N., Astro Cycles, and Speculative Markets
6. https://algotrading101.com/learn/yfinance-guide/
7. https://towardsdatascience.com/granger- causality-and-vector-auto-regressive-model-for-time-series-forecasting-3226a64889a6
8. https://www.aptech.com/blog/introduction-to-the-fundamentals-of-vector-autoregressive-models/
9. https://towardsdatascience.com/exploring- the-lstm-neural-network-model-for-time-series-8b7685aa8c