ASSIGNMENT SUBMISSION FORM

Course Name: **Foundational Project - 1**
Assignment Title: **Stock Value Prediction using Sentiment Analysis**
Submitted by: **Group 8**

Student Name	PG ID
Anjali Muralidharan	12110006
George Abraham	12110121
Mrinal Nidhi	12110047
Sonal Rai	12110054
Sreevatsan Sama Srijaganathan	12110060
Vikrant Dhawan	12110001

ISB Honour Code

- I will represent myself in a truthful manner.
- I will not fabricate or plagiarise any information with regard to the curriculum.
- I will not seek, receive or obtain an unfair advantage over other students.
- I will not be a party to any violation of the ISB Honour Code.
- I will personally uphold and abide, in theory and practice, the values, purpose and rules of the ISB Honour Code.
- I will report all violations of the ISB Honour Code by members of the ISB community.
- I will respect the rights and property of all in the ISB community.
- I will abide by all the rules and regulations that are prescribed by ISB.

Note: Lack of awareness of the ISB Honour Code is never an excuse for a violation. Please go through the Honour Code in the student handbook, understand it completely. Please also pay attention to the following points:

- Please do not share your assignment with your fellow students under any circumstances if the Honour Code scheme prohibits it. The HCC considers both parties to be guilty of an Honour Code violation in such circumstances.
- If the assignment allows you to refer to external sources, please make sure that you cite all your sources. Any material that is taken verbatim from an external source (website, news article etc.) must be in quotations. A much better practice is to paraphrase the source material (it still must be cited).

Chosen Project: Project 1-

Background

Stock prices fluctuate rapidly multiple times within a second. Investing in stocks using a hunch may throw us into losses but taking an informed decision will always yield good results. There are a lot of stock value prediction models out in the market, but alas! all those models have poor accuracy.

Objective: Content Creator

You must work on developing a stock value prediction model which will provide better accuracies. To build the model, the factors which influence the market value have to be extracted. The objective of this project is to build a Stock Market sentiment analysis model using news articles. You have to scrape the relevant information about a company from various news channels, perform sentimental analysis and emotion mining along with other NLP-related tasks to define a quantified index that can further help in predicting the stock price.

Data Sources:

Social media
News Channels
Wikipedia
Public forums

Phase 1: Planning (CRISP-ML(Q)) –

1. Business Understanding:

- **Business Problem :** Build a Stock Prediction Model using Sentiment Analysis
    a. **Objective:** Maximize the returns on investment.
    b. **Constraints:** Minimize the loss incurred.

- **Feasibility:**
    a. **Applicability of ML Technology:** The outcome of the study showed that news did not only correlate with the stock market but also was a high predictor of stock movement. Pagolu et al. [3] argue that a firm equity value depends not only on historical stock price data, but also on the current events, news, and product announcements.
    b. **Legal Constraints:** Company will not be involved in the Insider Trading. Company will not be held accountable in case of loss incurred by investor by using the stock prediction. We will not be involved in Insider's Trading.
    c. Requirements of the application:
        i. System Requirements: Cloud Computing Platform: Microsoft Azure
        ii. Human Resource: Data Scientist, Data Engineers, Quality Engineers, Project Manager
        iii. Robustness (less than 5%), Explain-ability: (Satisfactory output from LOCO, PDP)

- **Assumptions:** Uninterrupted access to the historical stock data from the financial websites and news articles and social networks.

- **Risk**
    a. The sparsity of stock related news will affect the prediction of the model
    b. User may tweet from multiple accounts to sway opinion, which can affect the emerging market with less Twitter users and tweets
    c. Data imbalance i.e., if the data is collected only from a particular country/region could result in biased predictions.

- **Data Quality Verification**
    a. **Data Description:**
        i. 1 year of historical stock prices for companies is obtained from Yahoo Finance (Date, Open, Close, High, Low, Volume).
        ii. 1 year of financial news articles related to companies was collected using the Beautiful Soup API calls to train the model and another 1 year of data to test the data. We also included the data from the comments section to not just include the news title sentiment.
        iii. 10% of test data will be assigned for Blind testing used in Evaluation phase.
    b. **Data Verification:** Follow below steps before building a machine learning model include:
        i. Variable identification: define each variable and its role in the dataset.
        ii. Perform Univariate analysis and bivariate analysis to determine the interaction between the variables.
        iii. Detect and treat missing values and outliers.

- **Data Collection:** Establish a CI/CD pipeline using GIT for Model Version Control. Data Version control is an extension of GIT that will be used for maintaining the data versioning.

- **Success Criteria:**
    1. **Business Success Criteria:** Quarterly 10% increase in the number of user acquisition rate.
    2. **ML Success:** ML algorithm to have an accuracy of 75%.
    3. **Economic Success Criteria**: Build automated test suite for future validation, deployment, and operations frameworks to save cost by 20%

2. Data Preparation:

- **Selecting Data :**
    a. The stock price related features we are going to extract are stock opening and closing price, volume, high and low for a period of 1 year. For textual data, we will extract N gram features. We are going to use embedded method for feature selection as it encompasses the benefits of both the wrapper and filter methods by including interactions of features but also maintaining reasonable computational cost. The algorithm we will use is Lasso Regularization, which adds a penalty to the different parameters of our model to reduce its freedom i.e., to avoid over-fitting.
    b. In stocks data we will select data for each stock for a period of one year (daily). In textual data, we will extract about five sentences surrounding the name of the company from articles and tweets in which the name of the company is mentioned.

- **Cleaning Data :**
    a. Collecting more data and principal component analysis are 2 ways through which we aim to reduce the noise/irrelevant features in our dataset. The new dataset will be evaluated for the SNR again.
    b. With the stock prices data, we need to check for the presence of duplicate records and null values. With textual data, we need to remove HTML tags, lowercasing, remove stop words and punctuation, tokenize sentences, perform noun phrasing, document weighting, and obtain TFIDF scores.

- **Constructing Data :**
    a. The new features that we will derive using our existing features are – exponential moving average (to give greater weight and significance on the most recent data points), average true range (measures market volatility), rate of change (measures rate of change of price b/w the present and a certain period ago)
    b. The data augmentation methods we will incorporate for our time series stock data are magnifying (interpolating a random slice of the data to the original size), jittering (adding gaussian noise) and reversing (reversing the time series data).

- **Standardizing Data :**
    a. The file format we are going to use to store the data set is CSV (comma separated values)
    b. We will apply a log transformation on the stock prices to reduce the difference between high and low stock prices. Another method that can be used for normalizing a time series data is using the formula : y = (x - min) / (max - min). For textual data we will perform stemming and lemmatization.

3. Data Modelling:

- **Literature Research :**
    a. Detailed research on patents, internal reports, etc. for existing solutions on similar business problems.
    b. Deep dive into the reasons behind poor accuracy of existing stock prediction models – Selection Bias, Portfolio Construction, and Incorrect data preprocessing.

- **Defining quality measures of the model:**
    a. Building a 75% accurate model with a close to 0.0 error margin.
    b. Using KNN algorithm to enhance robustness, performance, and complexity of the model.

- **Model Selection**
    a. Selection of the most appropriate model that aligns with the data available and best tailored to solve the business problem among a set of candidate models using resampling models or K-Fold Cross-Validation.

- **Model Training**
    a. Building the best mathematical representation of the relationship between data features and a target label.
    b. Defining high quality training data and algorithms to make the model as accurate as possible using Empirical Risk Minimization.

- **Ensuring Reproducibility**
    a. To repeatedly run the model on different datasets and try to obtain the same (or similar) results.
    b. Maintaining dynamism and high coding standards, recording dataset versioning, usage of latest libraries and maintaining a common framework.

4. Evaluation:

- **Validate Performance:**
    a. Perform Blind Testing i.e., validating the model performance on a disjoint dataset stored initially separate from training and test datasets.

- **Determine Robustness:**
    a. Test Model performance on noisy data, by imputing wrong / extreme values which may occur on rare instance in production environment.
    b. Re-train the model to handle noisy data basis testing outputs.
    c. Repeat this process until model matches the Robustness criteria specified in quality measure of the model

- **Increase explain ability for ML practitioner & end user**
    a. Identify the features which impact the model's prediction the most and check whether they are in-line with domain understanding.
    b. It is important to understand what features contribute to the model's prediction and why they do it.
    c. Test it using techniques like:
        i. Partial Dependence Plots (PDP): visual representation of how features influence the predicted outcome, with other features held constant.
        ii. Leave One Column Out (LOCO): leaves one column out, retrains the model, and then computes the differences of each LOCO model to the original model prediction score. Is score changes lot the variable which is left out must be important.

- **Compare results with defined success criteria**
    a. Check output consistency
    b. Record and validate overall cost. Check whether it is in-line with our business success criteria
    c. Record and provide feedback & suggestions to relevant team members as well make a note of constraints violation.

5. Deployment:

- **Inference Hardware :**
    a. Deploy the ML model exposing its predictive functionality as precomputed predictions (yes/no) through a web service endpoint on a cloud platform
    b. Model compression might be required if we want to extend the services on mobile applications

- **Model evaluation under production condition**
    a. Accuracy checks on live predictions will be made under incremental production conditions by running evaluation at each step. Same accuracy benchmarks will be

used which were used at evaluation stage. A confidence range can be established with a stop button.
    b. Live Data drifts check. The input data distribution should remain similar.

- **Assure user acceptance and usability**
    a. Design a user acceptance and usability test for the final user interface. Deploy only if it is passed by 90% of the sample users.
    b. Create a user guide to assure seamless usability for new users on platform

- **Minimize the risks of unforeseen errors**
    a. Prepare a fallback option in case a rare event hits (e.g., Pandemic induced stock market crashes). In this case the accuracy of the model will be compromised. Fallback option would be to reduce the confidence % by certain number.
    b. Prepare fall back for each data source. For e.g.., if twitter is down, we will have a secondary data source.

- **Deployment strategy**
    a. Prepare a document listing out the steps of the deployment strategy and the instructions for carrying out the steps in detail. Follow the steps while deploying the model into production.
    b. Disaster management: We should prepare a multi-server deployment strategy in case one of the servers is down due to unforeseen circumstances

6. Monitoring and maintenance:

- **Non-stationary data distribution:**
    a. Stock market is highly volatile and therefore the words/hashtags/expressions in social media/blogs are likely to vary with new events.
    b. Model needs to be re-trained with new vocabulary at regular intervals (say every three months) or when the output varies by 5% from the ML success criteria

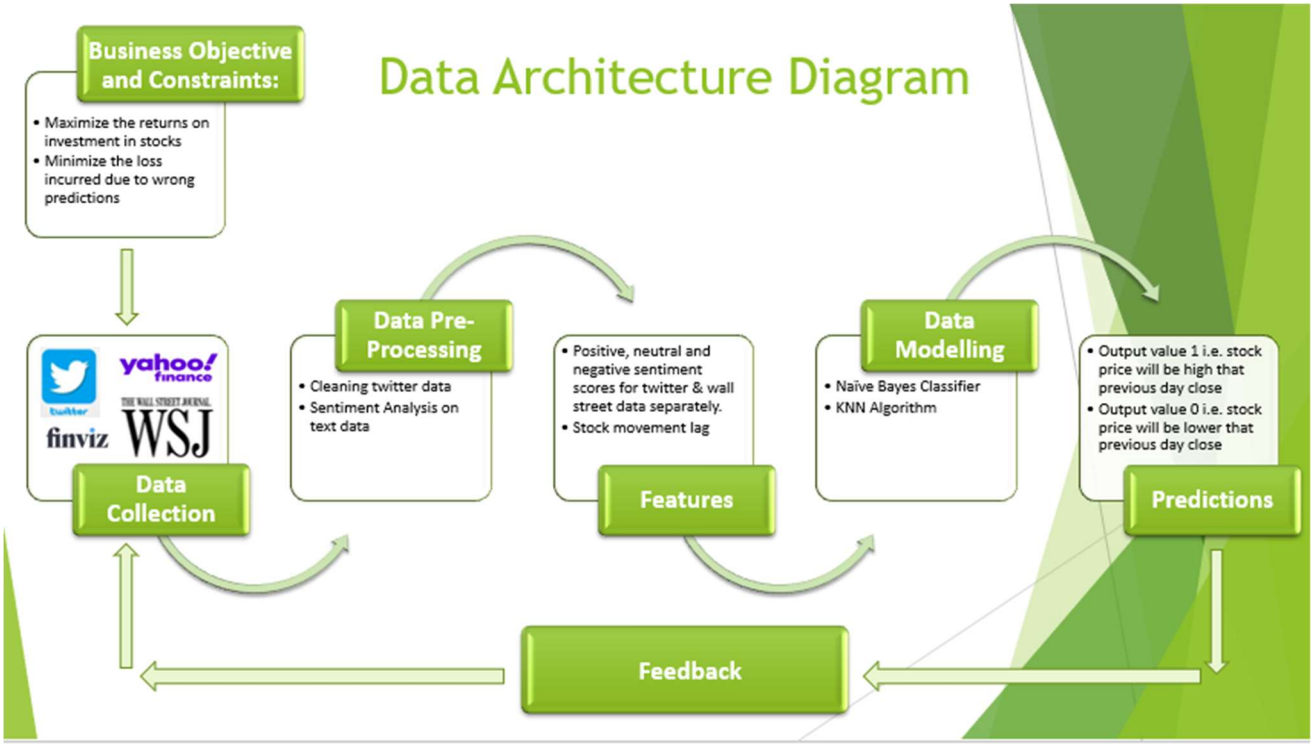
- **Degradation of hardware**
    a. Deployment of model in cloud will obviate issues related to aging of hardware.
    b. No sensor hardware is used in the model under consideration

- **System updates**
    a. Updates to system software and DLLs / libraries will be implemented only after testing for impact on the model.

- **Monitor**
    a. Compare the incoming stream of words with training data to assess the degree of variation and model staleness on a continuous basis.
    b. Need to monitor model performance by comparing statistics of training data with outputs from live data
    c. Re-train model if the variation is more than 5% from the ML success criteria

- **Update**
    a. Continuously collect data and retrain the model at regular intervals (say every three months).
    b. Updates will be tested for performance and accuracy. Updates having same or better accuracy as that of the previous version only will be released.

Phase 2 : Implementation-



1. Data Collection:

- **The stock we've chosen to train and test our model is TESLA stocks. This model can be scaled to work on any other stock too. The predictions are based on the dataset provided to the model hence, this is a scalable model.**

- **Data was collected from the following sources,**
    a. **Twitter:** 100 tweets were taken for each day for a period of one year that contained the query term TSLA

		POSITIVE_TW	NEGATIVE_TW	NEUTRAL_TW
Date				
01/01/21	**mean**	0.090564	0.036653	0.872772
01/02/21	**mean**	0.088891	0.042178	0.868901
01/03/21	**mean**	0.104495	0.046248	0.849248
01/04/21	**mean**	0.081693	0.038960	0.879356
01/05/21	**mean**	0.070178	0.029584	0.900228
...	**...**	...	...	...
31/05/21	**mean**	0.078802	0.042010	0.879079
31/07/21	**mean**	0.092733	0.037485	0.869673
31/08/21	**mean**	0.110436	0.051446	0.838079
31/10/21	**mean**	0.071832	0.040723	0.887396
31/12/21	**mean**	0.128871	0.024891	0.846178

b. **Finviz and Finnhub: I**n their free subscription model we were only able to collect news articles relating to Tesla stocks for a week. Hence, we decided to not use this data in our model.

c. **Yahoo finance:** Daily TSLA stock data was collected for a period of 1 year (stock opening and closing price, volume, high and low)

	POSITIVE_TW	NEGATIVE_TW	NEUTRAL_TW	POSITIVE_WS	NEGATIVE_WS	NEUTRAL_WS
Date						
01/01/21	0.090564	0.036653	0.872772	0.157500	0.051000	0.791000
01/02/21	0.088891	0.042178	0.868901	0.062000	0.031000	0.907000
01/03/21	0.104495	0.046248	0.849248	0.165750	0.069000	0.765500
01/04/21	0.081693	0.038960	0.879356	0.106333	0.051333	0.842667
01/05/21	0.070178	0.029584	0.900228	0.099000	0.040000	0.861000
...	...	...	...	...	...	...
31/05/21	0.078802	0.042010	0.879079	0.131550	0.050875	0.814750
31/07/21	0.092733	0.037485	0.869673	0.131550	0.050875	0.814750
31/08/21	0.110436	0.051446	0.838079	0.131550	0.050875	0.814750
31/10/21	0.071832	0.040723	0.887396	0.167000	0.096000	0.738000
31/12/21	0.128871	0.024891	0.846178	0.144833	0.065333	0.789667

d. **Wallstreet journal:** News articles from 2018 were scraped which contained news relating to the Tesla stocks. We then filtered this and used the latest 1 year's data.

	Date	TIME	SUMMARY	COMPOUND_TW	NEGATIVE_TW	NEUTRAL_TW	POSITIVE_TW
0	01/01/21	23:53:44	Do you guys foresee any dip on Monday at all? ...	0.0000	0.000	1.000	0.000
1	01/01/21	23:53:09	P/E ratios are very useful for determining the...	0.8066	0.026	0.832	0.142
2	01/01/21	23:46:22	To tap a vein just pricked... I would rather ...	0.0000	0.000	1.000	0.000
3	01/01/21	23:38:34	I became debt free in 2020. This year my accou...	0.4939	0.048	0.843	0.109
4	01/01/21	23:34:17	Although I'm not sure I see a lot of mix senti...	0.2354	0.064	0.842	0.094
5	01/01/21	23:25:33	$TSLA is incredibly overvalued. It's priced fo...	0.5719	0.000	0.684	0.316
6	01/01/21	23:25:20	Happy New Years Everyone! What company do you ...	0.8619	0.000	0.811	0.189
7	01/01/21	23:14:30	ThingsIPlanOnCancelling\n- margin calls\n- due...	0.0000	0.000	1.000	0.000
8	01/01/21	23:07:14	Everyone worried about $TSLA Model Y pricing a...	-0.7269	0.191	0.809	0.000
9	01/01/21	23:04:33	Monday Market Open!\n\nHappy New Year to all. ...	0.6114	0.000	0.889	0.111

2. Data Pre-Processing:

- **Data cleaning:** Tweets are cleaned by removing hashtags, twitter handle mentions, retweets, and hyperlinks.
- **Sentiment analysis:** Data from tweets and wall street journal is clubbed into a single dataset and then sentiment analysis is done on this dataset using the python package SentimentIntensityAnalyzer - VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of a sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only talks about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.
- **Feature engineering:** From the daily stock price data collected, the closing price is used to obtain a new feature which is termed as stock movement lag which is the difference between the closing stock price of the present day and closing stock price of the previous day.

Code snippet of the final dataset with 7 features,

	Date	stock_movment	POSITIVE_TW	NEGATIVE_TW	NEUTRAL_TW	POSITIVE_WS	NEGATIVE_WS	NEUTRAL_WS	stock_movment_lag
0	04/01/21	0	0.106337	0.032634	0.861010	0.102667	0.035333	0.862000	0.0
1	05/01/21	0	0.115218	0.033119	0.851663	0.131550	0.050875	0.814750	1.0
2	06/01/21	1	0.092733	0.046446	0.860802	0.078333	0.045000	0.877000	1.0
3	07/01/21	1	0.130970	0.050139	0.818901	0.263000	0.088000	0.649000	0.0
4	08/01/21	0	0.127604	0.043950	0.828446	0.131550	0.050875	0.814750	0.0
...	...	...	...	...	...	...	...	...	...
246	23/12/21	1	0.114129	0.035099	0.850822	0.146250	0.034750	0.819000	0.0
247	27/12/21	0	0.097990	0.042653	0.859347	0.125667	0.046333	0.828000	1.0
248	28/12/21	1	0.096376	0.032238	0.871386	0.151000	0.022333	0.826667	0.0
249	29/12/21	0	0.079713	0.051515	0.868762	0.153333	0.036000	0.810667	1.0
250	30/12/21	1	0.092515	0.057079	0.850376	0.145000	0.073000	0.782333	0.0

3. Data Modelling:

In the modelling phase, before building the model, we have done extensive research on the existing patents that model solutions to similar business problems.

When we deep dived into the reasons behind poor accuracy of these existing stock prediction models, we found 3 major issues –

1. Selection Bias – Selection Bias is the selection of outdated data. We have overcome selection bias by using randomization of data.
2. Portfolio Construction – Portfolio Construction are issues that occur due to the interest of people wanting to take minimum risk to achieve maximum returns. We have handled this by balancing the risk and rewards.
3. Incorrect Data Pre-processing – This usually happens because the distribution of stock prices is seldom normal. We have handled this by removing and adding few data attributes like Dummy Variables and by performing transformation by removal of skewed data.

Based on the above research done, we will define few quality measures of the model.
We aimed at building a 60% accurate model with a close to 0.0 error margin if not equal to 0.0 and we are able to achieve that by using the Naïve Bayes model.

In Model Selection, our objective was to select from a set of candidate models, the most appropriate model that aligns with the data available and is best tailored to solve our business problem. We identified the most appropriate model by using Resampling Method and K-Fold Cross Validation technique.
Resampling method helped us understand If the model will generalize well by measuring performance on data samples that it has not been trained on.
The K - cross-validation technique helped in randomly shuffling the dataset and then splitting it into k groups. One of the K groups was considered the test data and the other k-1 groups were clubbed into a training set and this process repeats for k iterations and the best model from the K runs was selected.

After selecting the best model, we moved into the next phase of training the model.
We built the best possible mathematical representation of the relationship between data features and the target label. Also ensured that we avoid overfitting while doing the same.

Once a model was trained, we ensured that there is reproducibility.
Hence, the model was repeatedly run-on different datasets and obtained similar results.
Across all the phases, with respect to code, we have maintained dynamism and high coding standards.
With respect to data, we recorded the dataset versioning for measuring the execution trends.
And with respect to the environment, we have maintained a common framework and stuck to the usage of latest libraries.

Algorithm and process followed:

- **Grouping the Sentiment Data at day level:**
    a. Around 100 tweets were considered for the sentiment analysis every day. So, the twitter and wall street data were grouped at day level
    b. We used 6 independent variables {positive, negative and neutral sentiments from both twitter and wall street data} which are summed at day level

- **Treating missing data in the Independent Variables**
    a. We took median of the column to replace any missing values
    b. The twitter & wall street data corresponding to weekend is removed from the dataset as we won't have stock price data on weekend

- **Designing the Dependent Variable**
    a. The dependent variable is calculated as the change is the stock price today vs the stock price of yesterday. If the stock price of today is higher than of yesterday's the variable would be "1" else, it would be "0".
    b. The hypothesis is that the predicted variable is dependent on the 6 independent variables with sentiment score.
    c. We use the lag of the stock movement variable as today's tweet would affect the stock movement tomorrow. The final dependent variable is named "stock_movment_lag"

- **Merging the Data**
    a. Both the datasets(one with "stock_movment_lag" and other with sentiment scores) are merged on the date variable. The total rows we have are 52*5 = 210 days (excluding weekends)
    b. We remove the nan, infinity, and NA values from the dataset as these values are not catered by the model

- **Building the classifier**
    a. The data(both dependent and independent) is divided into test and train at 20-80 split using "train_test_split" from "sklearn.model_selection"
    b. We build "Naïve Bayes" and "KNN" classifier to classify the data into "1" and "0" based on the sentiment score.

4. Evaluation (Model Results):

- **Naïve Bayes Results**
    - Accuracy: 60.78%
    - Precision: 64.86%
    - Recall: 77.41%

- **KNN Algorithm Results**
    - Accuracy: 49.01%
    - Precision: 64.70%
    - Recall: 35.48%

Final decision :
We choose **Naïve Bayes** to predict the stock movement based on the sentiment score as it gives us better results

5. Deployment Strategy:

a. Deploy the Naïve Bayes ML model exposing its predictive functionality as precomputed predictions (higher/ lower) through a web service endpoint like Flask API on Python.
b. The application will be hosted on Google cloud servers using Google App Engine
c. Design a user acceptance and usability test for the final user interface. Deploy only if it is passed by 90% of the sample users.
d. Create a user guide to assure seamless usability for new users on platform
e. Prepare fall back for each data source. For e.g.., if twitter is down, we will have a secondary data source like Facebook and Mastodon

6. Monitoring and Maintenance Strategy:

a. Model will be re-trained with new vocabulary at regular intervals (say every three months) or when the output varies by 5% from the ML success criteria
b. Compare the incoming stream of words with training data to assess the degree of variation and model staleness on a continuous basis.
c. Need to monitor model performance by comparing statistics of training data with outputs from live data
d. Continuously collect data and retrain the model at regular intervals (say every three months).
e. Updates will be tested for performance and accuracy. Updates having same or better accuracy as that of the previous version only will be released.
f. Prepare a multi-server deployment strategy in case one of the servers is down due to unforeseen circumstances (disaster management)

Steps to implement the code:

Run the .py files in the following order -> readHistoryStockPrice.py -> Twitter.py -> finnhub.py -> finviz.py -> WSJ_Scraping.py -> SentimentAnalysis.py -> Modelling.py

Data Sources and References:

- Stock Price Prediction Using News Sentiment Analysis: Saloni Mohan1, Sahitya Mullapudi1, Sudheer Sammeta1, Parag Vijayvergia1 and David C. Anastasiu1
- https://www.dummies.com/article/technology/information-technology/data-science/general-data-science/phase-6-of-the-crisp-dm-process-model-deployment-148174
- https://www.datascience-pm.com/crisp-dm-2/
- https://www.pluralsight.com/blog/machine-learning/3-steps-train-machine-learning
- https://arxiv.org/abs/2003.05155
- https://machinelearningmastery.com/improve-model-accuracy-with-data-pre-processing/
- https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/
- https://towardsdatascience.com/learning-theory-empirical-risk-minimization-d3573f90ff77
- https://neptune.ai/blog/how-to-solve-reproducibility-in-ml
- https://thirdeyedata.io/robustness-measurement-of-machine-learning-models-with-examples-in-python/
- https://www.kaggle.com/residentmario/denoising-algorithms
- https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/
- https://medium.com/@Nikhilkohli1/extracting-features-for-stock-prediction-streamlit-based-application-a97afc55d926
- https://arxiv.org/pdf/2010.15111.pdf
- https://tcoil.info/normalize-stock-prices-and-time-series-data-with-python-2/
- https://www.twitter.com
- https://finnhub.io/
- https://finviz.com/
- https://finance.yahoo.com/
- https://www.wsj.com/
- https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/
- https://thenewstack.io/deployment-strategies/
- https://towardsdatascience.com/learning-theory-empirical-risk-minimization-d3573f90ff77
- https://gist.github.com/YohanObadia/b310793cd22a4427faaadd9c381a5850
- https://patents.google.com/patent/US8285619B2/en