STOCKPREDICTIONANDEDUCATION

## APROJECTREPORT

***Submittedby***

Kumar Gautam (UID:21BCS8445) KrishnaDubey(UID:21BCS8413)

Angel Verma(UID:21BCS8445)

***inpartialfulfillmentfortheawardofthedegreeof***

# BACHELOROFENGINEERING

**IN**

ELECTRONICS ENGINEERING



**ChandigarhUniversity**

May 2023

# BONAFIDECERTIFICATE

Certified that this project report **“Stock Prediction and Education”** is the bonafide work of“ **NAMEOFTHE**

**CANDIDATE(S) ”**who carried out the project work under my/our

supervision.

<<SignatureoftheHoD>>

**SIGNATURE**

<<NameoftheHeadoftheDepartment>>

### HEADOFTHEDEPARTMENT

<<Department>>

<<SignatureoftheSupervisor>>

### SIGNATURE

<<Name>>

### SUPERVISOR

<<AcademicDesignation>>

<<Department>>

Submittedfortheprojectviva-voceexaminationheldon

**INTERNALEXAMINER EXTERNAL EXAMINER**

# TABLEOFCONTENTS

ListofFigures 7

Listof Tables 8

Listof Standards 9

[CHAPTER1.INTRODUCTION 11](#_TOC_250012)

* 1. [IdentificationofClient/Need/RelevantContemporaryissue 11](#_TOC_250011)
  2. IdentificationofProblem 11
  3. IdentificationofTasks 11
  4. [Timeline 11](#_TOC_250010)
  5. [OrganizationoftheReport 11](#_TOC_250009)

[CHAPTER2.LITERATUREREVIEW/BACKGROUNDSTUDY 12](#_TOC_250008)

* 1. [Timelineofthereportedproblem 12](#_TOC_250007)
  2. [Existingsolutions 12](#_TOC_250006)
  3. Bibliometricanalysis 12
  4. [ReviewSummary 12](#_TOC_250005)
  5. [ProblemDefinition 12](#_TOC_250004)
  6. [Goals/Objectives 12](#_TOC_250003)

[CHAPTER3.DESIGNFLOW/PROCESS 13](#_TOC_250002)

* 1. [Evaluation&SelectionofSpecifications/Features 13](#_TOC_250001)
  2. [DesignConstraints 13](#_TOC_250000)
  3. AnalysisofFeaturesandfinalizationsubjecttoconstraints 13
  4. DesignFlow 13
  5. Designselection 13
  6. Implementationplan/methodology 13

CHAPTER4.RESULTSANALYSISANDVALIDATION 14

4.1.Implementationofsolution 14

CHAPTER5.CONCLUSIONANDFUTUREWORK 15

* 1. Conclusion 15
  2. Futurework 15

## REFERENCES .......................................................................................................

**16APPENDIX .............................................................................................................**

## 17

1.PlagiarismReport............................................................................................................... 172.

DesignChecklist 17

## USERMANUAL 18

## ListofFigures

**Figure3.1………………………………………………………………………………….**

**Figure3.2………………………………………………………………………………….**

**Figure4.1 …………………………………………………………………………….……**

## ListofTables

**Table3.1 Table**

**3.2 ………………………………………………………………………………….**

**Table4.1 …………………………………………………………………………….……**

## ListofStandards(MandatoryForEngineeringPrograms)

|  |  |  |  |
| --- | --- | --- | --- |
| **Standard** | **PublishingAgency** | **Aboutthestandard** | **Pageno** |
|  |  | IEEE802.11ispartoftheIEEE |  |
|  |  | 802setoflocalareanetwork |  |
|  |  | (LAN)technicalstandardsand |  |
|  |  | specifiesthesetofmediaaccess |  |
| IEEE802.11 | IEEE | control(MAC)andphysicallayer  (PHY) protocols forimplementing wireless local areanetwork(WLAN)computercommunication. | Mention page nowhere standardisused |

Note:TextinRedispresentedasanexample(replacewithrelevantinformation)

# ABSTRACT

**----------------------------NewPage-------------------------**

# GRAPHICALABSTRACT

**----------------------------NewPage-------------------------**

# ABBREVIATIONS

**----------------------------NewPage-------------------------**

# SYMBOLS

**----------------------------NewPage-------------------------**

**CHAPTER -1**

# INTRODUCTION

## Identification of Client/ Need/ Relevant Contemporary issue

The need for this project is due to the significant growth of the stock market and the growingpublic interest in stock trading. For example take India, India **is** one of the fastest growing stockmarkets globally and it is important to provide traders and investors with accurate and reliableinformation to make informed decisions. The need for an online application that can accuratelypredict stock prices and provide educational resources on stock trading is more critical in India,wherethestockmarketisstilldevelopingand investorsneed guidanceandsupport.

Internetandmobiletechnologieshavemadeiteasierforpeopletoaccessstockmarketinformationandparticipateinstockmarkettrading.However,manyoftheseretailinvestorsmaynothavethenecessaryknowledgeandexpertisetomakeinformeddecisions,whichincreasestherisksassociated with stock trading. Therefore, there is a need to develop web applications that canprovide retail investors with resources to teach stock trading and improve their understanding ofthestock market.

## Identification of Problem

The broad problem that needs to be solved in Stock Prediction and Education web application using NLP, machine learning and deep learning are to solve the large-scale problem of accurately predicting stock prices and provide relevant educational resources to traders and investors. Stock markets are affected by many factors, such as economic conditions, geopolitical events, investor sentiment and company specific factors. It is difficult for traders and investors to keep track of all these factors and make informed decisions. Therefore, there is a need to develop web applicationsthat can use NLP, machine learning and deep learning techniques to analyze large volumes of data, identify patterns and provide accurate stock price forecasts. In addition, educational resources should be provided that can improve the understanding of the stock market and help traders and investorsmake informed decisions. The challenge is to develop an integrated solution that can provide both accurate stock price forecasts and related educational resources for traders and investors.

## IdentificationofTasks

Developing a website for stock Prediction and Education using NLP, machine learning and deeplearning involves a number of tasks that can be broadly divided into three phases: identify, build and test. Each step includes a number of task snecessary to develop a functional and reliable web application.

1. IdentificationPhase:

The identification phase involves defining the problem, understanding the customer's requirements and identifying the resources needed to develop the web application. This step includes the following tasks.

* 1. Definingtheproblem:
     1. Understand the scope of the project .
     2. Identify challenges and complexities in Stock Prediction and training.
     3. Define project goals and objectives.
  2. Understanding the client' s requirements:
     1. Identify the target group
     2. Understand the customer's needs and expectations
     3. Define the functions and features required in the web application
  3. Identify the required resources:
     1. Identification of data sources
     2. Identification of NLP,machine learning and deep learning libraries and frame works is

required.

* + 1. Defining the hardware and software requirements II. Build Phase:

The build phase involves the actual development of the web application. This step includes the following tasks.

1. Data collection and pre-processing:
   1. Collection of relevant data from various sources
   2. Data preprocessing to remove noise and inconsistencies
   3. Creation of a clean and structured data set for analysis
2. Development of a predictive model:
   1. Selection of appropriate machine learning and deep learning algorithms
   2. Model training with clean data set
   3. Model Performance Evaluation and Parameter Fine-Tuning
3. Integrating NLP Techniques:
   1. Identify relevant news articles and social media channels
   2. Using NLP techniques to analyze the mood and context of an article
   3. Acquisition of important information for inventory forecasting and trend analysis
4. Development of educational resources:
   1. Creating relevant educational content on stock trading
   2. Design a user-friendly interface for easy access to educational resources
   3. Development of interactive tools for stock market analysis and visualization III.

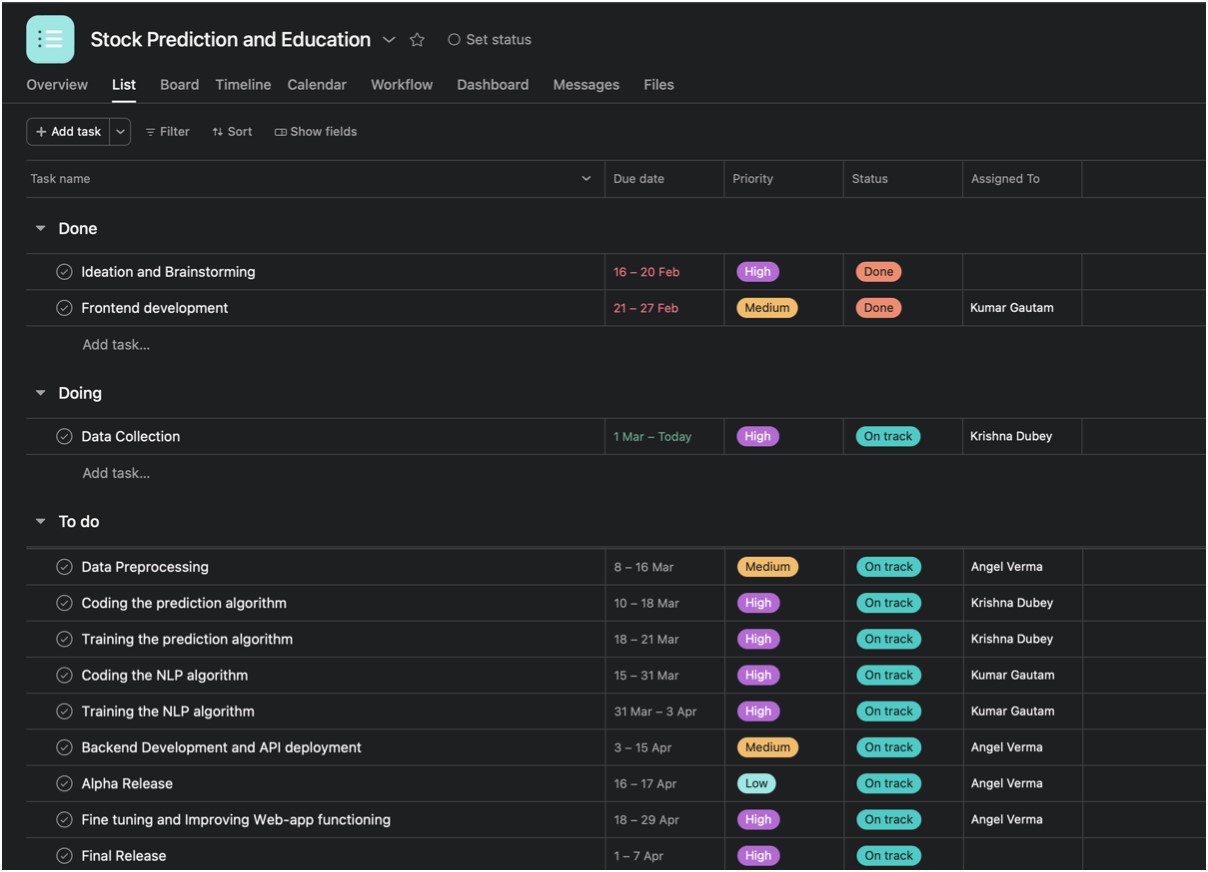
Testing Phase:

The testing phase includes testing the web application to ensure its functionality, reliability and usability. This step includes the following tasks.

1. Unit testing:
   1. Testing the accuracy and consistency of the predictive model
   2. Accuracy and reliability testing of NLP techniques
   3. Testing the usability and relevance of training resources
2. Integration testing:
   1. Testing the Integration of Prediction Model and NLP Techniques
   2. Integration testing of training resources and user interface
   3. Testing the general functionality of the web application
3. Testing the system:
   1. Testing the web application on different platforms and devices
   2. Web Application Testing and Scalability
   3. Identify and resolve potential problems or deficiencies

## Timeline

**Gantt Chart**



## Organization of the Report

### Chapter1:

In this, identification of problems and tasks has been defined. Moreover, it includes a Gantt Chart report of the timeline of different phases of our project.

### Chapter2:

In this, we have added different resources that we used for our project knowledge. These resourcesinclude various websites, research papers and books based on Stock predication (Prediction +Education)with existing solutions.

### Chapter3:

This chapter includes the designing part of the project as well as design constraints we faced and how we implemented the design with code at last. Back-end work has been done in this phase.

### Chapter4:

This chapter has the solutions to the problems identified in the earlier phase of this project. The implementation of the solution has been carried out, which means dealing with the back end of the website.

### Chapter5:

In this, the final project as well as conclusion and future work has been shown

**CHAPTER -2**

# LITERATURE REVIEW/BACKGROUND STUDY

Stock prediction and education websites have been a popular topic in the field of finance and artificial intelligence (AI). A literature survey of this topic could cover a wide range of research areas, including:

* + 1. Machine learning techniques for stock prediction: Many studies have explored the use of various machine learning algorithms for predicting stock prices, including support vectormachines, artificial neural networks, decision trees, and random forests. Researchers have also developed ensemble models that combine multiple algorithms to improve prediction accuracy.
    2. Sentiment analysis for stock prediction: Sentiment analysis techniques, which analyze social media and news sentiment to predict stockprices, have become increasingly popular in recent years. Researchers have used natural language processing (NLP) techniques to analyze social media data and news articles to identify sentiment patterns that can be used to predict stock prices.
    3. Hybrid models for stock prediction: Researchers have also developed hybrid models that combine both machine learning and sentiment analysis techniques to improve prediction accuracy. These models incorporate both technical and fundamental analysis to provide a more comprehensive approach to stock prediction.
    4. Educational resources for investors: In addition to stock prediction, many studies have focused on the development of educational resources for investors. These resources include online courses, tutorials, and webinars that help investors to better understand the stock market, investment strategies, and financial concepts.
    5. User behavior and website design: Researchers have also explored the impact of website design and user behavior on stock prediction and education websites. Studies have examined the impact of website design features such as user interface and navigability on user engagement and satisfaction. Other studies have analyzed user behavior data to identify patterns and preferences that can inform website design and content development.

Overall, there is a wealth of literature on stock prediction and education websites, covering a wide range of topics related to machine learning, sentiment analysis, hybrid models, educational resources, and user behavior.

* 1. **Timeline of the reported problem:-**

It is difficult to provide a comprehensive timeline of stock market prediction problems, as there have been numerous incidents through out history. However, here are some notable events:

1. Black Monday, October 19, 1987: The Dow Jones Industrial Average fell 22.6% in a single day, the largest percentage drop in stock market history. Many analysts were unable to predict the market crash, leading to widespread panic and losses.
2. Dot-com bubble, 1997-2001: During this period, the stock market experienced a rapid rise in the value of internet-based companies, which ultimately led to a bubble that burst in 2000-2001. Many analysts failed to predict the eventual collapse of the market, resulting in significant losses for investors.
3. 2008 Financial Crisis: The collapse of the US housing market and the subsequent financial crisis had a significant impact on global stock markets. Many analysts failed to predict the extent of the crisis, leading to widespread losses for investors.
4. Flash Crash, May 6, 2010: The Dow Jones Industrial Average fell 9% in a matter of minutes, before quickly recovering. The cause of the crash was never fully determined, but it highlighted the fragility of the stock market and the potentiall fo run expected events to cause significant disruptions.
5. COVID-19 pandemic, 2020: The pandemic had a significant impact on global stock markets,leading to widespread uncertainty and volatility. Many analysts were unable to predict the extent of the market's decline, leading to significant losses for investors.

In summary, there have been numerous incidents throughout history where analysts and prediction models have failed to accurately predict stock market movements. These events highlight the challenges of accurately predicting the stock market and the need for caution when investing.

* 1. **Existing solutions**

There are many existing solutions for stock prediction and education websites that use NLP and machine learning. Some examples include:

* + 1. Yahoo Finance - Yahoo Finance provides users with real-time stock quotes, financial news, and analysis using natural language processing (NLP) and machine learning algorithms.
    2. Seeking Alpha - Seeking Alpha is a financial news and analysis platform that uses machine learning algorithms to provide users with investment insights, stock analysis, and market predictions.
    3. Stock Rover - Stock Rover is a stock analysis platform that uses NLP and machine learning to provide users with investment research, portfolio analysis, and stock screening tools.
    4. Alpha Vantage - Alpha Vantage is a financial data provider that uses machine learning algorithms to provide users with real-time stock quotes, historical data, and technical analysis tools.
    5. Kavout - Kavout is an AI-driven investment platform that uses NLP and machine learning algorithms to provide users with stock analysis, trading signals, and market predictions.
  1. **Bibliometric analysis**

Bibliometric analysis of Stock Prediction and education Web application would involve analyzing the scientific and academic literature that cites or refers to the platform. However, Stock Prediction and education Web application is not typically the subject of academic research or publication.

Therefore, it may not be possible to perform a comprehensive bibliometric analysis of Stock Prediction.

However, some possible approaches to conducting a bibliometric analysis of related topics could include:

* + 1. Key features: Key features of estimate usually include the ability to input information about aspecific company stock, such as dividend yield, market valuation, vision and condition, and get an estimated value for that stock. Some other common features of used Stock Prediction and education Web application may include:

1. Market Analysis: Ability to analyse market trends and sales data to provide an accurate estimate of stock value
2. Historical Data: The ability to incorporate historical data about a specific Stock, such asprevious sales or ownership history, to provide a more accurate estimate.
3. Customization: The ability to adjust the estimated value based on specific Stock features, such as upgrades or modifications. Transparency: A clear explanation of how the estimated value is calculated
4. Comparison tools: Ability to compare the estimated value of a particular stock with other similar stock on the market
5. History Reports: Option to purchase a stock history report that provides information on details that can affect the value of the stock prediction.Overall, the key features of a used stock price estimate can vary depending on the specific tool and its intended audience. However, the ability to provide an accurate estimate basedon factors such as market trends and stock history is a common feature of most Stock pricing tools
   1. **Review Summary**

A stock prediction and education web application typically aims to provide investors with accurate stock market predictions and educational resources to help them make informed investment decisions.

In terms of stock prediction, such applications often use machine learning and artificial intelligence algorithms to analyze past and current stock market data, identify patterns, and make predictions about future market movements.

The accuracy of such predictions can vary, and it's important for investors to exercise caution when making investment decisions based on these predictions.

In addition to stock prediction, such web applications may also offer educational resources such as online courses, tutorials, and webinars to help investors understand the stock market, investmentstrategies, and financial concepts. These resources can be valuable for both novice and experienced investors.

The success of a stock prediction and education web application depends on various factors, such as the accuracy of the prediction models, the quality and relevance of educational resources, the user interface and experience, and the abilityto provide timely and relevant information to users.

Overall, a well-designed stock prediction and education web application can provide investors with valuable insights and resources to help them make informed investment decisions. However, it'simportant to approach such predictions with caution and to consider other factors such as market conditions, industry trends, and individual financial goals when making investment decisions.

* 1. **Problem Definition**

The problem that a stock prediction and education web application aims to address is the challenge faced by investors in making informed investment decisions in the stock market. This challenge arises due to the complexity of the stock market, the unpredictability of market movements, and the abundance of information available to investors.

Stock prediction and education web applications seek to address this problem by providing investors with accurate stock market predictions and educational resources to help them understand the market and make informed investment decisions.

The application uses machine learning andartificial intelligence algorithms to analyze past and current stock market data and identify patterns that can be used to predict future market movements.

In addition, the web application provides educational resources such as online courses, tutorials, and webinars to help investors understand the stock market, investment strategies, and financial concepts.

The ultimate goal of a stock prediction and education web application is to empower investors to make informed investment decisions and reduce the risk associated with stock market investments. By providing accurate predictions and educational resources, the application can help investors to better understand the stock market and make investment decisions based on data-driven insight

* 1. **Goals/Objectives**

The goals and objectives of a stock prediction and education web application are to:

1. Provide accurate stock market predictions: The primary goal of such an application is to provide investors with accurate stock market predictions based on past and current market data. This can help investors make informed investment decisions and reduce theriskassociated with stock market investments.
2. Offer educational resources: The application should provide investors with educational resources such as online courses, tutorials, and webinars to help them understand the stock market, investment strategies, and financial concepts. This can help investors to better understand the market and make more informed investment decisions.
3. Increase user engagement: The web application should be designed to engage users and keep them coming back for more. This can be achieved through a user-friendly interface, personalized content, and timely updates on market movements.
   * + 1. Provide timely and relevant information: The application should provide users with timely and relevant information about market movements, news, and trends. This can help investors stay up-to-date on market developments and adjust their investment strategies accordingly.
       2. Improve prediction accuracy over time: The application should continually improve its prediction models over time to provide more accurate predictions to users. This can be achieved through the use of machine learning algorithms and regular updates to the application's data sources.
       3. Build user trust and loyalty: The web application should be designed to build trust and loyalty among users by providing accurate predictions and educational resources, as well

as a positive user experience. This can help to ensure that users continue to use theapplicationand recommend it to others.

Overall, the goals and objectives of a stock prediction and education web application are to provideinvestors with accurate predictions, educational resources, and a positive user experience to help them make informed investment decisions and reduce the risk associated with stock market investments.

CHAPTER 3.

DESIGN FLOW/PROCESS

## 3.1. Evaluation & Selection of Specifications/Features

## Evaluation

To evaluate a stock price prediction using Python, you'll need to follow these general steps:

**Prepare the data**: Clean and pre-process the data to remove any missing values or outliers. You may also need to convert categorical variables to numerical data using techniques such as one-shot coding

**Data Partitioning**: Partition the data into training and test sets. The training set will be used to train the model and the test set will be used to evaluate its performance

.

**Train a model:** Use a machine learning algorithm to train a model that can predict Stock prices based on input data. Popular algorithms for regression tasks like this include linear regression, decision trees, and random forests.

**Improving the model:** If the performance of the model is not satisfactory, you can try to improve it by adjusting the hyper parameters or using a different algorithm.

**Deploy the model**: Once you are satisfied with the performance of the model, you can deploy it and make predictions for new data

## Selection of Specifications/Features:

1. Input data: The input data for the LSTM model include historical stock price data, financial statements, and market news. The data are b pre-processed to remove missing values and outliers, and feature engineering techniques is applied to extract relevant features. A time series split or k-fold cross-validation is used to ensure the model is trained on representative data.
2. LSTM architecture: A deep LSTM architecture with several layers is used to capture complex relationships in the data. The number of neurons in each layer can be determined through experimentation, with larger layers used for larger datasets. The activation function for the LSTM cells is selected based on the problem, with commonly used functions including sigmoid, tanh, and ReLU.
3. Hyperparameters: The learning rate, batch size, and optimizer is set before training. A low learning rate helps the model converge more slowly but more accurately, while a high learning rate leads to faster convergence but may result in overshooting the optimal solution. A larger batch size result in more stable gradient estimates but requires more memory.
4. Evaluation metrics: Evaluation metrics such as accuracy, precision, recall, and F1 score is used to evaluate the model's performance. A high accuracy and F1 score are desirable. The confusion matrix is used to evaluate the model's performance for different classes.
5. Deployment: The LSTM model is to be deployed as a web application using a cloud service provider such as AWS or Google Cloud.Security measures such as encryption and authentication should be implemented to protect user data. Regular monitoring and testing should be performed to ensure the system is functioning correctly.

## 3.2. Design Constraints

Designing a Stock Price Preducation using Python may have some limitations or restrictions that need to be considered. Here are some of the common design limitations:

1. **Availability of data:** The quality and quantity of available data can significantly affect the accuracy of a Stock price prediction. Estimate performance will depend on the availability of saccurate and comprehensive stock pricing data.
2. **Model performance**: The performance of stock price prediction depends on the accuracy and efficiency of the machine learning algorithm used. The chosen algorithm must be able to handle the size and complexity of the data while providing accurate predictions.
3. **Feature Selection:** The selection of features that go into an estimator can affect its performance. The selected characteristics should therefore be relevant, important and meaningful for predicting the price of a Stock
4. **Scalability**: Stock price prediction should be scalable to handle large data sets and large number of users. It should be able to provide fast and accurate forecasts while being able to handle peak traffic.
5. **Privacy and Security:** It is important to ensure that the privacy and security of user data is maintained throughout the development and deployment of Stock pricing estimator. Collected data should be anonymized and encrypted to prevent any unauthorized access to sensitive information.
6. **Budget/Cost:** Developing and deploying a stock price prediction can require a budget for hardware, software, and personnel. The budget should be considered to ensure that the development process is feasible and sustainable.
7. **Regulatory compliance**: Stock price prediction should comply with all relevant regulatory standards or guidelines, such as data protection laws and anti discrimination laws, to ensure that the model does not have any unintended biases or ethical issues. These limitations should be taken into account when designing and implementing stock price prediction using Python to ensure that the model is accurate, efficient, scalable, and compliant

Here are some common standards to consider when developing a Stock preducation using Python:

1. **Code Quality**: The code used in developing the estimator should be of high quality, well documented, and easy to understand and maintain. This can be achieved by using coding standards such as PEP 8 and performing code reviews.
2. **Data quality:** The quality of the data used to train and test the estimator is critical to its accuracy. Data should be cleaned and pre-processed to ensure that it is consistent, accurate and representative of the population. It is also important to ensure that the data is not skewed.
3. **Documentation:** The estimator development process should be well documented, including the problem statement, data sources and pre-processing, algorithm used, performance metrics, and any assumptions made during development.
4. **Testing and Validation:** The estimator should be thoroughly tested and validated to ensure its accuracy and reliability. This includes using appropriate testing methods, such as cross-validation, and performing sensitivity analysis to identify potential weaknesses in the model
5. **Security:** The estimator should be designed to be secure and protect user data from unauthorized access or tampering. This can be achieved by using secure data storage and transfer methods as well as implementing access control

By adhering to these standards, stock price prediction can be developed to be accurate, reliable, safe, and compliant with applicable regulations and ethical standards

## 3.3. Analysis of Features and finalization subject to constraints

1. Features: The features used in a stock prediction web application using LSTM can include historical stock prices, financial ratios, technical indicators, and news sentiment. Historical stock prices are the most important feature and should be used to train the LSTM model. Financial ratios such as price-to-earnings ratio (P/E), price-to-book ratio (P/B), and dividend yield can provide insight into a company's financial health. Technical indicators such as moving averages and relative strength index (RSI) can provide insight into market trends. News sentiment analysis can help predict how news headlines can affect stock prices.
2. Constraints: The constraints for a stock prediction web application using LSTM can include data availability, model complexity, overfitting, interpretability, real-time prediction, and security and privacy. Data availability is a significant constraint as the quality and quantity of historical data can affect the accuracy of the model's predictions. Model complexity is another constraint as the model's architecture must be designed to work within the available computational resources and time constraints. Overfitting is a constraint as the model must be able to generalize to new data. Interpretability is a constraint as users and stakeholders may require an explanation of how the model arrived at its predictions. Real-time prediction is a constraint as users may require up-to-date information. Security and privacy are constraints as users' data must be protected.
3. Finalization: The finalization of a stock prediction web application using LSTM will depend on the analysis of the features and the constraints. The LSTM model's architecture should be designed to work within the available computational resources and time constraints while considering the feature analysis. Techniques such as regularization, cross-validation, and ensemble methods can help mitigate the risk of overfitting. Feature importance analysis and visualization can help improve the interpretability of the model

Above as explained lists common features that may be included in our project. Once these elements are selected, it is important to complete them with respect to the design constraints and standards. For example, to ensure accuracy, it may be necessary to collect data from multiple sources and use a diverse and representative training data set.

**3.4. Design Flow**

Here are two alternative design flows for estimating the price of Stock using Python:

***Design Procedure 1:***

**Data Collection**: Collect Stock data from various sources such as google ,yahoo and other browsers.

**Data cleaning and preparation:** Clean and prepare data for use in pricing, including tasks such as removing duplicates, filling in missing values, and converting categorical variables to numerical representations.

**Feature Engineering:** Engineer features that are likely to be useful in predicting Stock price, such as combining existing features, creating interaction terms, or transforming variables.

***Design process 2:( Random Forest)***

1. **Define the problem**: Clearly define the stock price prediction problem, including input properties, constraints, and performance metrics.
2. **Collect data**: Collect Stock data from a variety of sources, such as google , websites or publicly available datasets.
3. **Clean and pre-process the data:** Clean and pre-process the collected data, including tasks such as removing duplicates, filling in missing values, and converting categorical variables to numerical representations.
4. **Explore and visualize the data:** Explore and visualize the cleaned data to gain insight into the relationships between the input features and the target variable (price).
5. **Feature Engineering:** Design new features that can be useful in predicting Stock price, such as combining existing features, creating interaction terms, or transforming variables.
6. **Feature Selection:** Use feature selection techniques to identify the most important features for predicting Stock Prices.
7. **Create a user interface:** Create a user interface, such as a web application or a command-line interface, that allows users to enter the relevant features of a stock and get an estimated price.
8. **Test and validate solutions**: Test and validate solutions by comparing estimated prices with actual prices of different stocks.
9. **Continually improve the solution:** Continually improve the solution by collecting new data, retraining the model and updating the user interface based on user feedback. These two alternative design flows for used stock pricing using Python share many similarities, such as data collection, cleaning, preparation, feature engineering, model selection, training, evaluation, deployment, and maintenance. However, it differs in the additional feature selection step in Design Flow 2, which can improve the accuracy and efficiency of the model by identifying the most important features for predicting Stock prices.

## 3.5. Design selection

Here are some design options for Stock preducation using Python:

1. **Linear Regression**: Predicting stock prices in Python using linear regression is easy is a simple and interpretable model that can provide quick estimates of Stock prices based on a set of input features. It assumes a linear relationship between the input properties and the target variable (price) and can be easily implemented in Python using libraries such as scikit-learn or stats models.  [Stock market forecasting is a attractive application of linear regression. machine learning packages like scikit-learn](https://www.google.com/search?rlz=1C1VDKB_enIN1014IN1014&q=Stock+market+forecasting+is+an+attractive+application+of+linear+regression.+Model+machine+learning+packages+like+scikit-learn&spell=1&sa=X&ved=2ahUKEwjAsemSrsX-AhWTxTgGHYjbB80QBSgAegQIBxAB) .As easy as these analyses are to implement, selecting features with ample enough predictive power to turn a profit is more of an art than science. In training our model, we’ll take a look at how to easily add common technical indicators to our data to use as features in training our model. Let’s take this in a step-by-step approach starting with getting our historic pricing data.
2. **Decision trees**: Decision trees can capture complex relationships between input elements and a target variable and can handle non-linear relationships. They are also easy to interpret and can be used for feature selection. However, they can be prone to overfitting and require careful tuning of the hyper-parameters.
3. **Neural Networks:** Neural networks can capture complex non-linear relationships between input functions and a target variable and can be used for function engineering. They are also highly scalable and can handle large datasets. However, they can be computationally expensive, require large amounts of data, and can be difficult to interpret. The choice of design for Stock Price estimation using Python depends on the specific requirements of the problem and the available resources. It is important to consider factors such as model accuracy, interpretability, and scalability, as well as available data and computational resources. It is an ensemble method that combines multiple decision trees to produce more accurate and stable predictions
4. **Random Forests:** Random forests are a set of decision trees that can improve estimation accuracy and reduce overfitting. They can handle large data sets and are robust to outliers and missing values. However, they can be computationally expensive and may not be as interpretable as linear regression or decision trees.

Here are a few reasons why Random Forest can be a good choice for a stock price prediction:

1. **Robust to noise and outliers:** Random Forest can handle noisy and outlier data because it builds trees based on random subsets of features and data. This helps avoid overfitting and improve the robustness of the model.
2. **Handles non-linear relationships:**Random Forest can handle complex, nonlinear relationships between the input elements and the target variable. This is useful for estimating the price of a Stock because the relationship between stock features and price may not be linear.
3. **Scalability:** Random Forest is scalable and can handle large datasets with a large number of features. This is useful for estimating the price of a stock because there may be many features that are relevant to predicting the price.
4. **Ease of use**: Random Forest is fairly easy to implement in Python using libraries like scikit-learn. It also has a number of hyperparameters that can be tuned to improve performance.
5. **Provides Feature Importance:** Random Forest provides a feature importance measure that can help identify the most important features to predict the price of a Stock. When using Random Forest for Stock price prediction it is important to carefully select the features to include in the model and tune the hyperparameters to achieve the best performance. In addition, it is important to properly train and test the model to ensure that it is accurate and reliable

## 3.6. Implementation plan/methodology

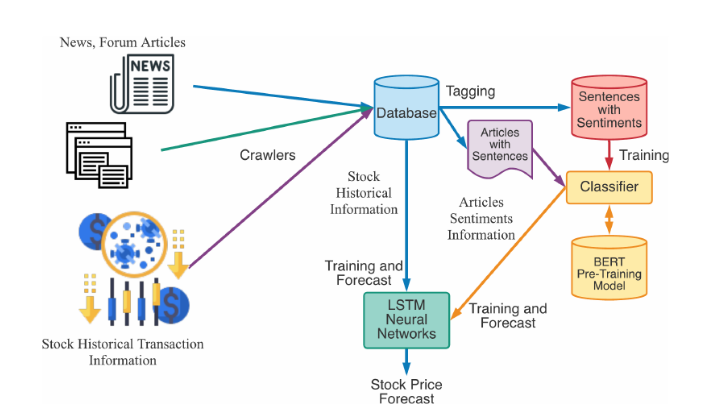
Machine learning models implemented in trading are often trained on historical stock prices and other quantitative data to predict future stock prices. However, natural language processing (NLP) enables us to analyze financial documents such as 10-k forms to forecast stock movements. LSTM (Long Short-term Memory) is one of the extremely powerful algorithms for time series. It can catch historical trend patterns & predict future values with high accuracy.Considered among the most potent tree-based techniques, Random Forest can predict the stock process as it can also solve regression-based problems.

However, with the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Price Prediction using machine learning advancements have begun to include such approaches in analyzing stock market data. The Opening Value of the stock, the Highest and Lowest values of that stock on the same day, as well as the Closing Value at the end of the day are all indicated for each date.

Methodology consisted of the following steps: Data Collection, Data Pre-processing, splitting

the data set into training and test data, building a LSTM model, then make predictions. They incorporated various text pre-processing methods such as stop-words removal, normalization, lemmatization, stemming, tokenization, BOW, TF/IDF and LSTM(Long Short Term Memory) which is a special kind of Recurrent Neural Network, capable of learning long-term dependencies in time series. After training, their model [13] learns from previous stock price close and improves in terms of accuracy. They used API to get data of AAPL stocks which can be used well with Bombay Stock Exchange, National Stock Exchange, or NASDAQ.

***The experimental ﬂow chart of a proposed stock price forecast:***



**Building a random forest model:**

**A.**  Train the Random Forest model on the training set using the Random Forest Regressor class from the scikit-learn library.

**B.** Tune the hyperparameters of the model, such as the number of trees, the maximum depth of each tree, and the minimum number of samples needed to split a node. This can be done using techniques such as grid search or random search.

**Model Evaluation**: Evaluate the performance of the Random Forest model on the test set using metrics such as mean squared error, mean absolute error, and Rsquared. Adjust the hyperparameters as needed to improve performance.

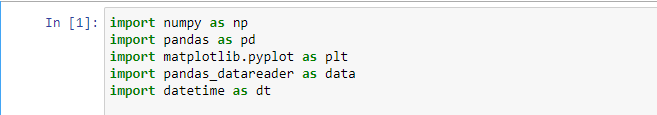
**Prediction:** Once a model is trained and evaluated, it can be used to make predictions on new data. This can be done using the forecast () method from the scikit-learn library.

**Deploy**: Deploy the model in a suitable format for production use. This may include wrapping the model as a REST API, web application, or command-line tool, depending on the project's requirements. The methodology for implementing Stock Price estimation using Python with Random Forest algorithm involves several steps, including data collection, data cleaning and pre-processing, feature selection, data partitioning, Random Forest model building, model evaluation, prediction, and deployment. The process can be iterative, with modifications to the model and features based on evaluation results until an accurate and reliable price estimate is achieved.

Here is the pseudocode for training a Random Forest model on the used STOCK price prediction dataset:

*Importing the libraries-*

ln[1]:



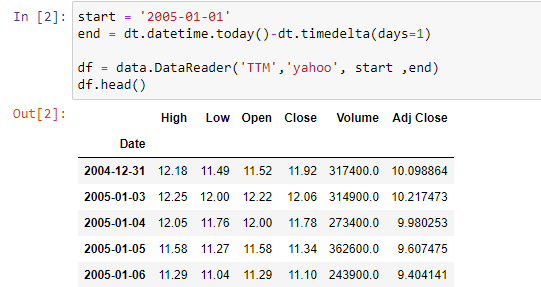
The libraries imported above will be used as follows:

1. pandas\_datareader- this will be used to fetch the historical data of the every stock from yahoo finance.
2. numpy - to perform the data manipulation on BAC stock price to compute the input features and output. If you want to read more about numpy then it can be found here.
3. Sklearn - Sklearn has a lot of tools and implementation of [machine learning](https://blog.quantinsti.com/trading-using-machine-learning-python/) models. Random Forest Classifier will be used to create Random Forest classifier model.
4. matplotlib.pyplot – to perform the and show the calculations and variations through the graph.

#### Fetching the data

The next step is to import the price data of TTM stock from yahoo finance. The data.DataReader function from pandas is used to get the TTM data for 18 years from 1 Jan 2005 to yesterday as shown below. The data is stored in the dataframe data.

In [2]:



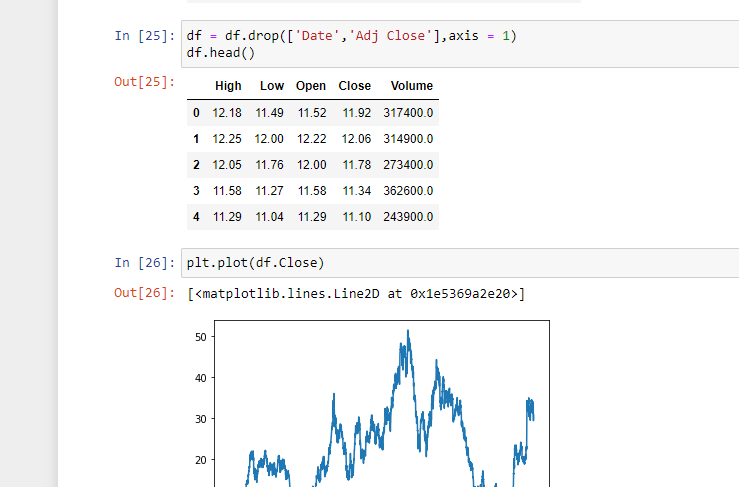
#### Creating input and output dataset

In this step, I will create the input and output variable.

1. Input variable: I have used '(Open - Close)/Open', '(High - Low)/Low', standard deviation of last 5 days returns (std\_5), and average of last 5 days returns (ret\_5)
2. Output variable: If tomorrow’s close price is greater than today's close price then the output variable is set to 1 and otherwise set to -1. 1 indicates to buy the stock and -1 indicates to sell the stock.

The choice of these features as input and output is completely random

In [3]:



**CHAPTER -4**

# RESULT ANALYSIS AND VALIDATION

**Implementation of solution**

The implementation of the solution has been carried out, which means dealing with the front-end design of our home page and back-end design of our stock preduction of the web application.

**Develop the web application :** This involves building the web application with the front end and backend components including the interface and database.

