**CHAPTER-1 Technology Stack**

**1.1) Hardware Used:**

For the development of the project, I have used the following hardware configuration:

♣ Processor: AMD Ryzen 5 25000U with Radeon Vega Mobile 2.00GHZ

♣ Ram: 8.00 GB

♣ Hard Disk Drive: 1TB

♣ Operating System: Windows 10

**1.2. Language Used:**

**1.2.1. Python (version - 3.7.3):**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help programmers write clear, logical code for small and large-scale projects. Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)).

**1.2.2. Why use Python?**

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

**1.3. Tools Used:**

**1.3.1. Anaconda:**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 15 million users and includes more than 1500 popular data-science packages suitable for Windows, Linux, and MacOS. Anaconda distribution comes with more than 1,500 packages as well as the [Conda](https://en.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

**1.3.2. Jupyter Notebook:**

Jupyter [Notebook](https://en.wikipedia.org/wiki/Notebook_interface) (formerly IPython Notebooks) is a [web-based interactive](https://en.wikipedia.org/wiki/Rich_Internet_application) computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a [JSON](https://en.wikipedia.org/wiki/JSON) document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.wikipedia.org/wiki/Markdown)), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

Jupyter Notebook can connect to many kernels to allow programming in many languages. By default Jupyter Notebook ships with the IPython kernel. As of the 2.3 release (October 2014), there are currently 49 Jupyter-compatible kernels for as many programming languages, including [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [R](https://en.wikipedia.org/wiki/R_(programming_language)), [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)) and [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)).

**1.4. Packages Used:**

**1.4.1. Numpy:**

NumPy is the fundamental package for scientific computing in Python. **Numpy**is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

NumPy’s array class is called ndarray. It is also known by the alias array. Note that numpy.array is not the same as the Standard Python Library class array.array, which only handles one-dimensional arrays and offers less functionality.

**The important attributes of an ndarray object are:**

* **ndarray.ndim:** The number of axes (dimensions) of the array.
* **ndarray.shape:** The dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with n rows and m columns, shape will be (n, m).
* **ndarray.size:** The total number of elements of the array. This is equal to the product of the elements of shape.
* **ndarray.dtype:** An object describing the type of the elements in the array.
* **ndarray.itemsize:** The size in bytes of each element of the array. For example, an array of elements of type float64 has itemsize 8 (=64/8), while one of type complex32 has itemsize 4 (=32/8). It is equivalent to ndarray.dtype.itemsize.
* **ndarray.data:** The buffer containing the actual elements of the array.

**1.4.2. Pandas:**

Pandas is a [Python](http://www.python.org/) package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

**1.4.2.1. Pandas Dataframe:**

**Pandas DataFrame** is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the **data**, **rows**, and **columns**

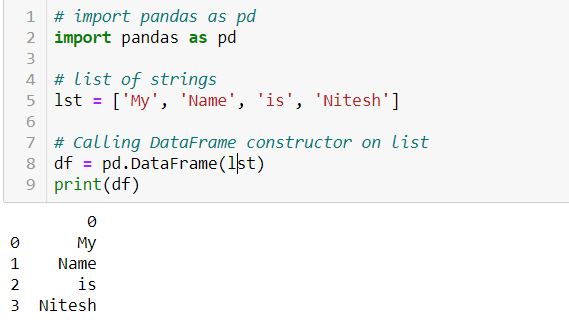


Fig. 1.1 Pandas Dataframe

**1.4.2.2. Pandas Series:**

Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index. Pandas Series is nothing but a column in an excel sheet.  
Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.

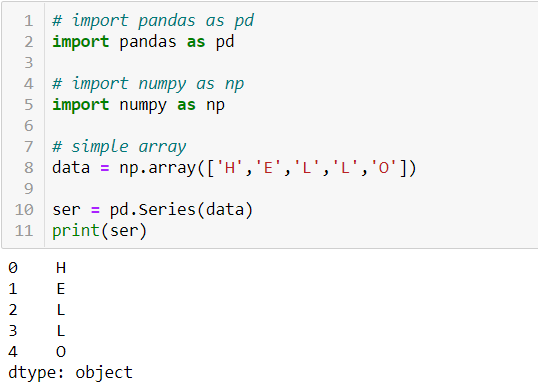


Fig. 1.2 Pandas Series

**1.4.3. Matplotlib:**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python Scripts, the Python and IPython Shells, the Jupyter Notebook, Web Application Servers and for graphical user interface toolkits. It is low level, provides lots of freedom. Matplotlib tries to make easy things and hard things possible. We can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc. with just few lines of code.

**1.4.4. Seaborn:**

Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It has great default styles. Seaborn aims to make visualization a central part of exploring and understanding the data. Its dataset-oriented plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

**1.5. Technology Used:**

**1.5.1. Data Analysis Using Python:**

Python is an increasingly popular tool for data analysis. In recent years, a number of libraries have reached maturity, allowing R and Stata users to take advantage of the beauty, flexibility, and performance of Python without sacrificing the functionality these older programs have accumulated over the years. Data analysis is the process of evaluating data using analytical and statistical tools to discover useful information and aid in business decision making. There are a several data analysis methods including data mining, text analytics, business intelligence and data visualization.

**1.5.1.1. Steps for Data Analysis:**

* **Importing Data with Pandas**

The first step is to read the data. The data is stored as a comma-separated values, or csv, file, where each row is separated by a new line, and each column by a comma (,). In order to be able to work with the data in Python, it is needed to read the csv file into a Pandas DataFrame. A DataFrame is a way to represent and work with tabular data.

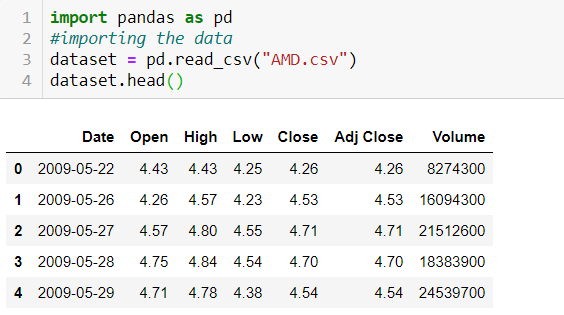


Fig. 1.3 Read CSV file in Dataframe

* **Handling the Missing Data**

The Data Analysis Phase also comprises of the ability to handle the missing data from our dataset, and not so surprisingly Pandas live up to that expectation as well. This is where dropna and/or fillna methods comes into the play. While dealing with the missing data, you as a Data Analyst are either supposed to drop the column containing the NaN values (dropna method) or fill in the missing data with mean or mode of the whole column entry (fillna method), this decision is of great significance and depends upon the data and the affect would create in our results.

* **Deletion:** It is of two types: List Wise Deletion and Pair Wise Deletion.

In **List Wise Deletion**, we delete observations where any of the variable is missing. Simplicity is one of the major advantage of this method, but this method reduces the power of model because it reduces the sample size.

In **Pair Wise Deletion**, we perform analysis with all cases in which the variables of interest are present. Advantage of this method is, it keeps as many cases available for analysis. One of the disadvantage of this method, it uses different sample size for different variables.

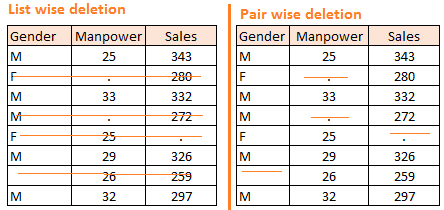
[](https://www.analyticsvidhya.com/wp-content/uploads/2015/02/Data_Exploration_2_2.png)

Fig. 1.4 Deletion Techniques

* **Mean/ Mode/ Median Imputation**

Imputation is a method to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. Mean / Mode / Median imputation is one of the most frequently used methods. It consists of replacing the missing data for a given attribute by the mean or median or mode of all known values of that variable.

* **Correlation and Correlation Computation**

Correlation is a simple relationship between two variables in a context such that one variable affects the other. Correlation is different from act of causing. One way to calculate correlation among variables is to find Pearson correlation. Here we find two parameters namely, Pearson coefficient and p-value. We can say there is a strong correlation between two variables when Pearson correlation coefficient is close to either 1 or -1 and the p-value is less than 0.0001.

Correlation can be derived using following formula-

**Correlation = Covariance(X, Y) / SQRT (Var(X)\* Var(Y))**

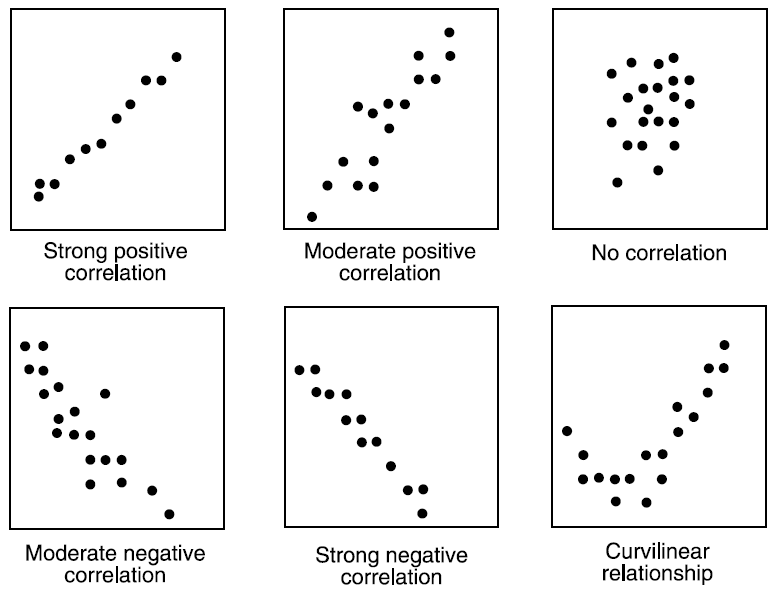


Fig. 1.5 Types of Correlation

**1.5.2. Machine Learning**

Machine learning (ML) is the [scientific study](https://en.wikipedia.org/wiki/Branches_of_science) of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [statistical models](https://en.wikipedia.org/wiki/Statistical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to perform a specific task without using explicit instructions, relying on patterns and [inference](https://en.wikipedia.org/wiki/Inference) instead. It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Or in simpler way we can explain machine learning as:

Two definitions of Machine Learning are offered. Arthur Samuel described it as: "the field of study that gives computers the ability to learn without being explicitly programmed." This is an older, informal definition.

Tom Mitchell provides a more modern definition: **“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”**

Machine leaning problem can be assigned to one of two broad classifications:

**1.5.2.1. Supervised Learning:**

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

Supervised learning is the [machine learning](https://en.wikipedia.org/wiki/Machine_learning) task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled [training data](https://en.wikipedia.org/wiki/Training_set) consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyses the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

Supervised learning problems are categorized into:

* **Regression:**

In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.

* **Classification:**

In here, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories. Or in other words we can say that in classification we are just predicting the value of the class to which the input data belongs.

**1.5.2.2. Unsupervised Learning:**

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

**1.5.3. Machine Learning Models**

There are various types of machine learning models based on the types of the time of machine learning (i.e. – supervised and unsupervised) but in the training was constrained by the limited time so the trainer was only able to cover the supervised learning part of the course so only one regression machine learning model and two classification model were possible so the following models mentioned with their working principle and the related statistics were taught in training.

**1.5.4. What are Machine Learning Models?**

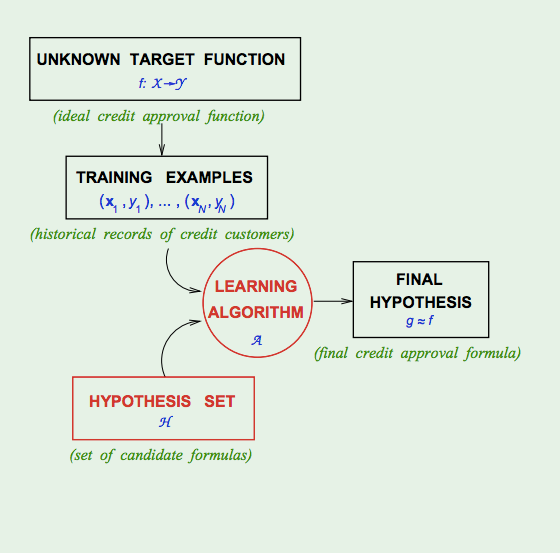
* A model is the relationship between features and the label.
* A Machine Learning model is a mathematical model that generates predictions by finding patterns in your data.
* Machine Learning Models generate predictions using the patterns extracted from the input data.
* The hypothesis set and the learning problem, together, can be referred to as “learning model”.

Fig. 1.6 Mechanism of ML model

* Learning in the supervised model entails creating a function that can be trained by using a training data set, then applied to unseen data to meet some predictive performance.
* An algorithm is the general approach you will take. The model is what you get when you run the algorithm over your training data and what you use to make predictions on new data. You can generate a new model with the same algorithm with different data, or a different model from the same data with a different algorithm.

**1.5.4.1. Regression Models**

* **Linear Regression**

In linear regression, the relationships are modelled using [linear predictor functions](https://en.wikipedia.org/wiki/Linear_predictor_function) whose unknown model [parameters](https://en.wikipedia.org/wiki/Parameters) are [estimated](https://en.wikipedia.org/wiki/Estimation_theory) from the [data](https://en.wikipedia.org/wiki/Data). Such models are called [linear models](https://en.wikipedia.org/wiki/Linear_model). Most commonly, the [conditional mean](https://en.wikipedia.org/wiki/Conditional_expectation) of the response given the values of the explanatory variables (or predictors) is assumed to be an [affine function](https://en.wikipedia.org/wiki/Affine_transformation) of those values; less commonly, the conditional [median](https://en.wikipedia.org/wiki/Median) or some other [quantile](https://en.wikipedia.org/wiki/Quantile) is used. Like all forms of [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), linear regression focuses on the [conditional probability distribution](https://en.wikipedia.org/wiki/Conditional_probability_distribution) of the response given the values of the predictors, rather than on the [joint probability distribution](https://en.wikipedia.org/wiki/Joint_probability_distribution) of all of these variables, which is the domain of multivariate analysis.

**1.5.4.2. Classification Models**

* **LOGISTIC REGRESSION:-**

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) that in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), although many more complex [extensions](https://en.wikipedia.org/wiki/Logistic_regression#Extensions) exist. In [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), logistic regression (or logit regression) is [estimating](https://en.wikipedia.org/wiki/Estimation_theory) the parameters of a logistic model (a form of [binary regression](https://en.wikipedia.org/wiki/Binary_regression)). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an [indicator variable](https://en.wikipedia.org/wiki/Indicator_variable), where the two values are labeled "0" and "1". In the logistic model, the [log-odds](https://en.wikipedia.org/wiki/Log-odds) (the [logarithm](https://en.wikipedia.org/wiki/Logarithm) of the [odds](https://en.wikipedia.org/wiki/Odds)) for the value labeled "1" is a [linear combination](https://en.wikipedia.org/wiki/Linear_function_(calculus)) of one or more [independent variables](https://en.wikipedia.org/wiki/Independent_variable) ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a [continuous variable](https://en.wikipedia.org/wiki/Continuous_variable) (any real value). The corresponding [probability](https://en.wikipedia.org/wiki/Probability) of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labelling; the function that converts log-odds to probability is the logistic function.

* **K-nearest neighbour:-**

The k-nearest neighbour’s algorithm (k-NN) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

* In k- NN Classification, the output is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k=1, then the object is simply assigned to the class of that single nearest neighbour.
* In k- NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbours.

k- NN is a type of [instance-based learning](https://en.wikipedia.org/wiki/Instance-based_learning), or [lazy learning](https://en.wikipedia.org/wiki/Lazy_learning), where the function is only approximated locally and all computation is deferred until classification.

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry.

**CHAPTER-2 Demonstration of Tech. in Project**

**3.1. Problem Statement:**

Stock market prediction is basically defined as trying to determine the stock value and offer a strong idea for the people to know and predict the market and the stock prices. It is generally presented using the quarterly financial ratio using the dataset. Thus, relying on a single dataset may not be sufficient for the prediction and can give a result which is inaccurate. Hence, we are contemplating towards the study of machine learning with various datasets integration to predict the market and the stock trends.

**2.2. A Potential Solution to the Problem:**

**2.2.1. Stock Analysis:**

Stock market analysis is the analysis which enables investors to identify the intrinsic worth of a security even before investing in it. All stock market tips are formulated after thorough research by experts. Stock analysts try to find out activity of an instrument/sector/market in future.

By using stock analysis, investors and traders arrive at equity buying and selling decisions. Studying and evaluating past and current data helps investors and traders to gain an edge in the markets to make informed decisions. Fundamental Research and Technical Research are two types of research used to first analyse and then value a security.

Performing a research before making an investment is a must. Taking control and planning one's financial future has become very important for many people. It is only after a thorough research that you can make some assumptions into the value and future performance of an investment.

Stock research is important because taking the time to look over the financial history of the companies that one is thinking of investing in, will give the prospective buyer a better sense of the future.

When someone is putting their hard earned money into a stock, they need to research that stock in order to make sure that the company is not laden with too much debt, is generating sufficient, have satisfied customers, are growing cash flows, investing in their future and are trading at a reasonable market valuation. By reviewing the stock's financial reports, one can make an educated decision whether the company is stable, growing and has an improving future. There are two methods of research for the prediction of the Stock Market—

* **Fundamental Research**

In fundamental research, you try to find out value of an equity share using the information provided in the[financial statements](https://cleartax.in/s/financial-reports) of the company. The investor tries to analyse various aspects of the business like competitive advantage, financial soundness, quality of management and competition. The main aim is to ascertain the relative attractiveness of the underlying business.

Here, it is assumed that the market price doesn’t reflect the true value of the company due to some uncontrollable external factors like investor sentiments. As the market will attain equilibrium, the real value will be equal to its market price in the long run. It believes that paying a higher price for a stock will affect return on investment adversely. Thus, by means of [financial ratios](https://cleartax.in/s/profitability-ratio), investors try to arrive at the true value at which a stock should ideally trade in the market.

* **Technical Research**

Technical research relates to the study of past stock prices to predict the trend of prices in future. It shows you the direction of movement of the share prices. With the help of technical research, you can identify whether there will be sharp rise or fall in the price of share. It is not dependent on recent news or events which has already been incorporated in the price of the share.

As the stock prices are dependent on investor psychology which keeps changing according to news and events, technical research emphasises the use of Stop-losses. It will save investors from suffering a big loss in future. Technical research gives meaningful results only for stocks which are high in demand and traded in huge volumes. Technical research uses different types of charts like bar chart, candlestick chart; to understand the pattern of stock prices. Daily charts are used by short term traders to examine the immediate movement in the stock prices. Weekly / monthly charts are used by medium/long term traders to ascertain the probability to earn higher more in the long run.

**2.3. Literature Survey:**

During a literature survey, we collected some of the information about Stock market prediction mechanisms currently being used.

**2.3.1. Survey of Stock Market Prediction Using Machine Learning Approach**-

The stock market prediction has become an increasingly important issue in the present time. One of the methods employed is technical analysis, but such methods do not always yield accurate results. So it is important to develop methods for a more accurate prediction. The technique that was employed in this instance was a regression. Since financial stock marks generate enormous amounts of data at any given time a great volume of data needs to undergo analysis before a prediction can be made. Each of the techniques listed under regression has its own advantages and limitations over its other counterparts. One of the noteworthy techniques that were mentioned was linear regression. The way linear regression models work is that they are often fitted using the least squares approach, but they may alternatively be also be fitted in other ways, such as by diminishing the "lack of fit" in some other norm, or by diminishing a handicapped version of the least squares loss function. Conversely, the least squares approach can be utilized to fit nonlinear models.

**2.3.2. Impact of Financial Ratios and Technical Analysis on Stock Price Prediction Using Random Forests**-

The use of machine learning and artificial intelligence techniques to predict the prices of the stock is an increasing trend. Due to the vast number of options available, there can be n number of ways on how to predict the price of the stock, but all methods don’t work the same way. The output varies for each technique even if the same data set is being applied. In the cited paper the stock price prediction has been carried out by using the random forest algorithm is being used to predict the price of the stock using financial ratios form the previous quarter. This is just one way of looking at the problem by approaching it using a predictive model, using the random forest to predict the future price of the stock from historical data. However, there are always other factors that influence the price of the stock, such as sentiments of the investor, public opinion about the company, news from various outlets, and even events that cause the entire stock market to fluctuate. By using the financial ratio along with a model that can effectively analyse sentiments the accuracy of the stock price prediction model can be increased.

**2.3.3. A Survey on Stock Market Prediction Using SVM**-

The recent studies provide a well-grounded proof that most of the predictive regression models are inefficient in out of sample predictability test. The reason for this inefficiency was parameter instability and model uncertainty. The studies also concluded the traditional strategies that promise to solve this problem. Support vector machine commonly known as SVM provides with the kernel, decision function, and sparsity of the solution. It is used to learn polynomial radial basis function and the multi-layer perceptron classifier. It is a training algorithm for classification and regression, which works on a larger dataset. There are many algorithms in the market but SVM provides with better efficiency and accuracy. The correlation analysis between SVM and stock market indicates strong interconnection between the stock prices and the market index.

**2.3.4. Stock Market Prediction: Using Historical Data Analysis**-

The stock market prediction process is filled with uncertainty and can be influenced by multiple factors. Therefore, the stock market plays an important role in business and finance. The technical and fundamental analysis is done by sentimental analysis process. Social media data has a high impact due to its increased usage, and it can be helpful in predicting the trend of the stock market. Technical analysis is done using by applying machine learning algorithms on historical data of stock prices. The method usually involves gathering various social media data, news to extract sentiments expressed by individuals. Other data like previous year stock prices are also considered. The relationship between various data points is considered, and a prediction is made on these data points. The model was able to make predictions about future stock values.

**2.4. About Dataset:**

Each row represent the daily stock Market data, each columns contains the attributes that described on the columns metadata. The raw data contains 2335 rows and 7 columns.

* **Date:** date of the stock.
* **Open:** price of the stock at market open
* **High:** Highest price reached in the day
* **Low:** Lowest price reached in the day
* **Close:** price of the day at market close
* **Adj. Close:** Adjusted closing price amends a stock's [closing price](https://www.investopedia.com/terms/c/closingprice.asp) to accurately reflect that stock's value after accounting for any corporate actions.
* **Volume:** Number of shares traded

It is not always necessary that the dataset we get is useful for the final result so, We used the raw data to create new columns/features that are more related or more useful to our project using feature engineering, the feature that we introduce are –

* **High-low:** we can check the daily volatility of the stock by high-low change.
* **Daily% change:** we can also check the daily volatility of the stock by daily% change.
* **Predic.Price:** label is the final price of the stock which we want to predict.

**2.5. Working of the Project:**

My whole project is based on the regression problem. Now for the Stock Price Prediction there are plenty of algorithms that can predict the Price of the stock but here for the scope of this project report only one machine learning model will be sufficient. Here, we are going to train Linear Regression Model on Stock Dataset and get their result and check their predictive ability for future Prices.

* **Linear Regression**

The basic concept of this algorithm mentioned are and given in chapter 2 of this project report and from this point, this project report will be oriented towards understanding of the working of the Linear Regression model. How this model work? We are going to train Linear Regression Model on the Dataset and check the predictive ability of the model. How good this model work, the steps related to this project are:

* Data extraction
* Missing value detection
* Missing value handling
* Feature engineering
* Visualization
* Appling and comparing the results of the model
* Conclusion of the model analysis

**2.5.1. Working of Linear Regression:**

In linear regression, the relationships are modelled using [linear predictor functions](https://en.wikipedia.org/wiki/Linear_predictor_function) whose unknown model [parameters](https://en.wikipedia.org/wiki/Parameters) are [estimated](https://en.wikipedia.org/wiki/Estimation_theory) from the [data](https://en.wikipedia.org/wiki/Data). Such models are called [linear models](https://en.wikipedia.org/wiki/Linear_model). Most commonly, the [conditional mean](https://en.wikipedia.org/wiki/Conditional_expectation) of the response given the values of the explanatory variables (or predictors) is assumed to be an [affine function](https://en.wikipedia.org/wiki/Affine_transformation) of those values; less commonly, the conditional [median](https://en.wikipedia.org/wiki/Median) or some other [quantile](https://en.wikipedia.org/wiki/Quantile) is used. Like all forms of [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), linear regression focuses on the [conditional probability distribution](https://en.wikipedia.org/wiki/Conditional_probability_distribution) of the response given the values of the predictors, rather than on the [joint probability distribution](https://en.wikipedia.org/wiki/Joint_probability_distribution) of all of these variables, which is the domain of [multivariate analysis](https://en.wikipedia.org/wiki/Multivariate_analysis).

Linear regression with multiple variables is also known as "multivariate linear regression".

We now introduce notation for equations where we can have any number of input variables.

|  |
| --- |
|  |

The multivariable form of the hypothesis function accommodating these multiple features is as follows:



In order to develop intuition about this function, we can think about θ0​ as the basic price of a house, θ1​ as the price per square meter, θ2​ as the price per floor, etc. x1​ will be the number of square meters in the house, x2​ the number of floors, etc.

Using the definition of matrix multiplication, our multivariable hypothesis function can be concisely represented as:

|  |
| --- |
|  |

This is a vectorization of our hypothesis function for one training example.

**USES:-**

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

* If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.
* If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

**CHAPTER-3 Interface Screenshots**

**CODE:**

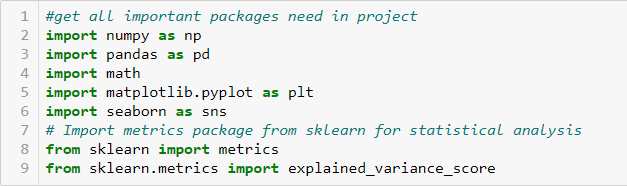


Fig. 4.1. Importing all Packages

In the above piece of code we are importing all the packages required in the successful completion of this project.

Here, we are importing Numpy, Pandas for data, importing matplotlib and seaborn for Visualization importing sklearn metrics for the statistical analysis.

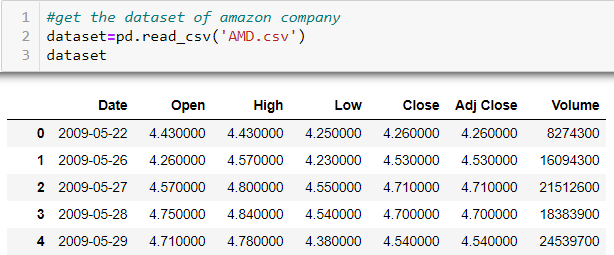


Fig. 3.2 Read the data

In the above snippet if code, Data is being read from the CSV file named ‘AMD.csv’ and a Dataframe is created using pandas named dataset.

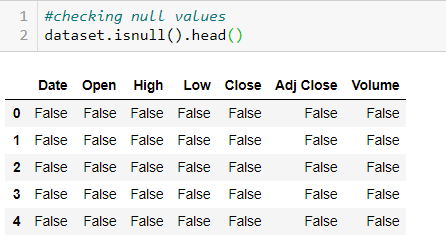


Fig. 3.5 checking null values

In this, We checking the any null values in the dataset by using isnull() function.

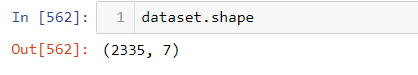


Fig. 3.6 Shape of Data

Here we check the shape of data (i.e. how many rows and columns are present in the dataset). And here you can see that we have 2335 rows and 7 columns.

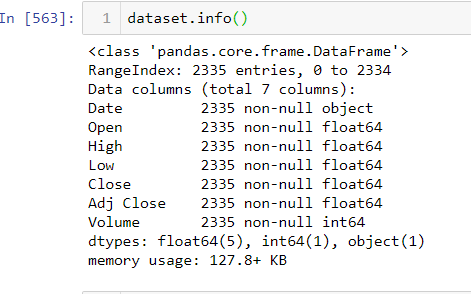


Fig. 3.7 about the Dataset

In the above code, we used the info () function to get the concise summary of the dataset like types of element used, how much memory used.

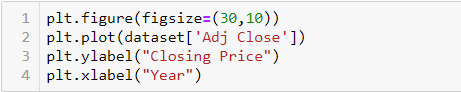


Fig. 3.8 Code for Visualization

In the above code we using matplotlib to visualize the ‘Adj Close’ feature in the dataset where take x axis as year and y axis as Closing price are.

Output of the given code is given below –

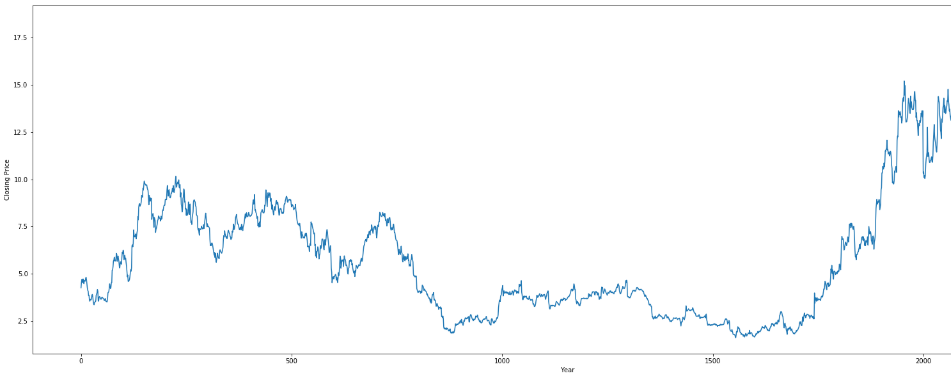


Fig. 3.9 Visualization of Adj Close.

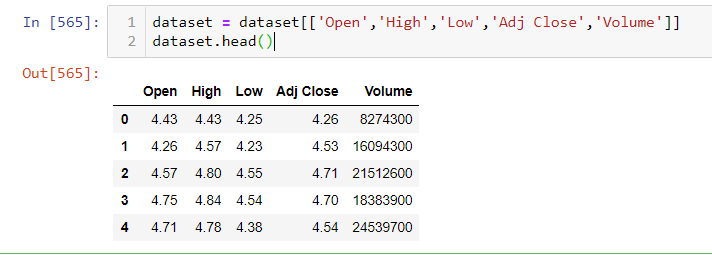


Fig. 3.10 a look on dataset

In above code you can see the there is no relation between the price change and volume over time.

It is not necessary that the dataset we are using is useful for the final result so, we used the raw data to create new columns/features that are more useful to our project using feature engineering. So we consider daily volatility, such as with the high minus low % change and daily percent change with the help of other features like open, low, high, close, adj. close and volume.

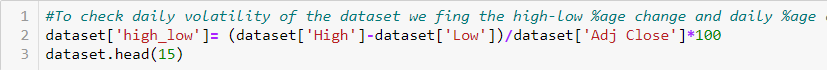


Fig. 3.11 high—low % change



Fig. 3.12. Daily % change

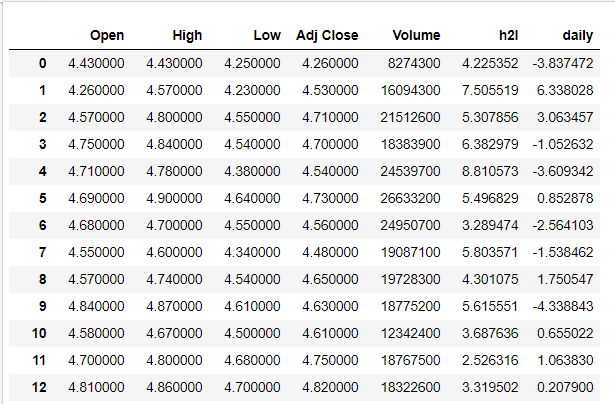


Fig. 3.13 a look at our new dataset

In the above code you can see that we add the new features in our dataset with the help of feature Engineering. That are daily % change and the high—low % change which is more helpful in our model.

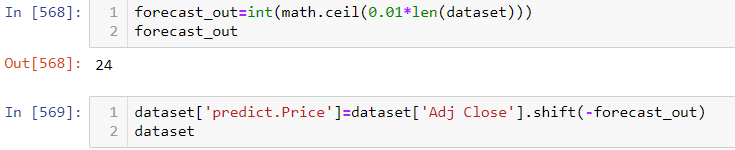
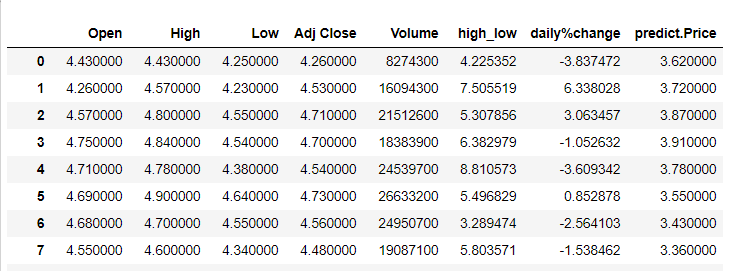


Fig. 3.14 work on new feature

Now, in the above code we are adding the new feature called predict. Price which we want to predict in our model so, here we define the new forcast\_out column and we take 1% of the entire length of dataset in the forcase\_out column and we add our new column predict. Price in the dataset with the help of simple pandas operation.



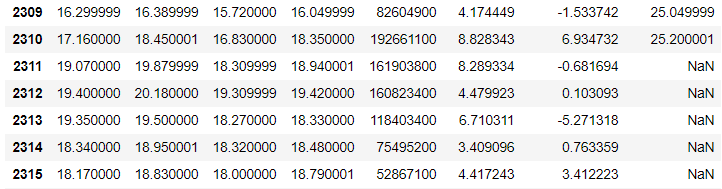


Fig. 3.16 a look at the NaN values

In the above code we have some NaN values because of the shifting operation we do before. So,

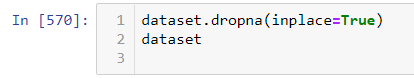


Fig. 3.17 using dropna () function

We use the dropna () function to drop all the NaN values from the dataset.

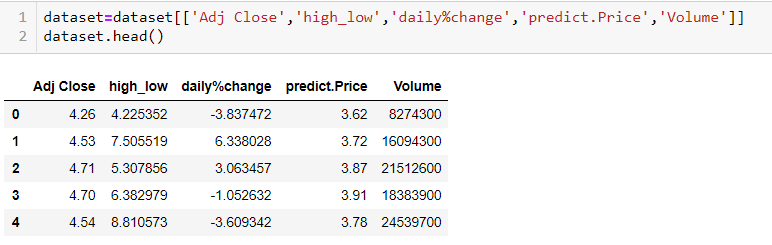


Fig. 3.15 a look at our new features in the dataset

In the above code as you can see that we only take those columns in our dataframe which we are going to use in our Linear Model for the Stock Price Prediction

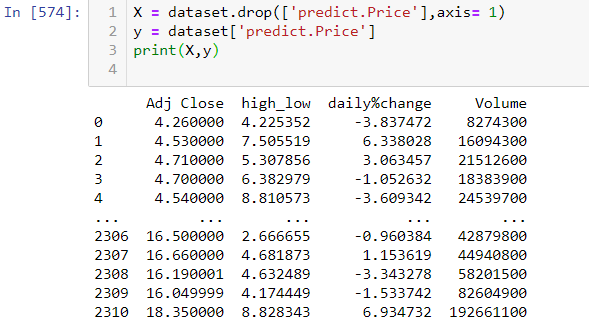


Fig. 3.16 a look at dependent and independent variable

In the above code we take dependent variable i.e. predict. Price which we need to predict is selected as **Target variable** are stored it in ‘y’. And we take independent variable i.e. Adj Close, high-low, daily% change and Volume and stored it in ‘y’.

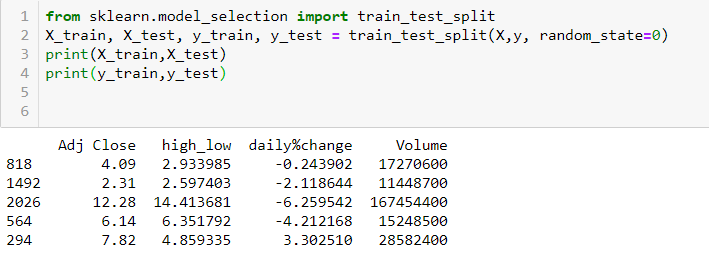


Fig. 3.17 train test split

And then we split the data in the X\_train, X\_test and y\_train, y\_test. For the training and testing of the data.

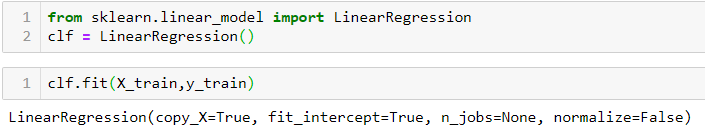


Fig. 3.18 accuracy of model

Now here we applied the linear model on the dataset to predict the value.

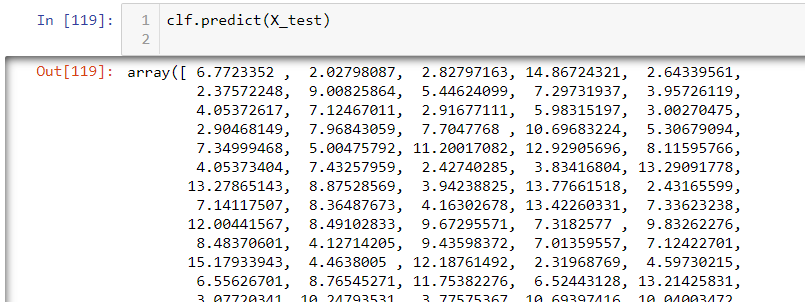


Fig. 3.19 predictions by the model

Here, In above code shows the predictions made by the model for the next ‘n’ days for the Stock.

**CHAPTER-4 Conclusion**

Linear regression model is applied for technical analysis and prediction of stock market and it is showing 89% accuracy which is far more accurate than fundamental predictions.

**4.1. Accuracy score**

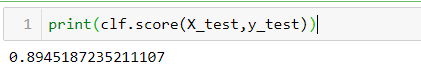


Fig. 4.1 accuracy of our model

**4.2. Error Evaluation Metrics**

To evaluate the effectiveness of our methods, we will use the root mean square error (RMSE), mean absolute error (MAE) metrics and mean squared error (MAE). For this, the lower the value, the better the prediction.

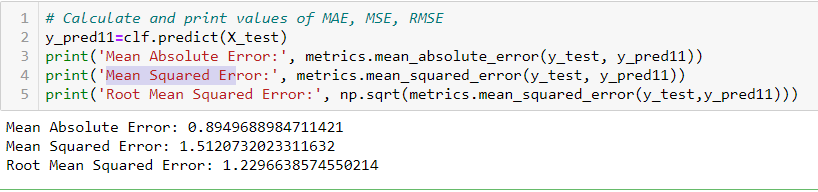


Fig. 4.2 evaluate the effectiveness

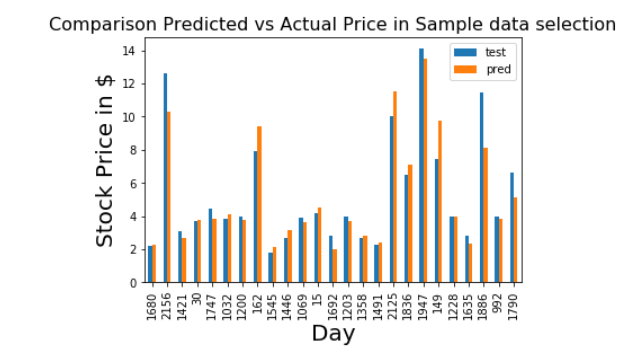


Fig. 4.3 comparison predicted vs actual price

Here you can see the comparison between the actual price and the predicted price by the model, this shows the accuracy of the model

**CHAPTER-5 BIBLOGRAPHY**

**5.1. Help from online sources**

* <http://scikit-learn.org/stable/>.
* Few tutorials on cross validation from YouTube channel Data School <https://www.youtube.com/user/dataschool>.
* <http://seaborn.pydata.org/>.
* For incites of notebooks and help in my project organisation <https://www.kaggle.com/>
* For dataset <https://www.kaggle.com/>
* For working out my basic code for experimentation <https://colab.research.google.com/>
* Very useful articles on various topics related to data science and data analytics <https://towardsdatascience.com/>
* Statistical knowledge for almost every machine learning model <https://www.coursera.org/learn/machine-learning> teacher - dr. Andrew ng
* <https://markdunne.github.io/public/mark-dunne-stock-market-prediction.pdf>
* <https://zerodha.com/varsity/chapter/volatility-calculation-historical/>
* <https://www.slideshare.net/jshivank/stock-market-prediction-49019834>
* <https://www.geeksforgeeks.org/python-programming-language/>

**5.2. Citations for books**

* For beginner level motivation and knowledge for machine learning models Hands-On Machine Learning with Scikit-Learn and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems.