**Stock Market Prediction with Historical Time Series Data and Sentimental Analysis over stock news: Insights from Data Mining using Machine Learning**

**ABSTRACT**

The art of forecasting the stock prices has been a difficult task for many of the researchers and analysts. In fact, investors are highly interested in the research area of stock price prediction. For a good and successful investment, many investors are keen in knowing the future situation of the stock market. Good and effective prediction systems for stock market help traders, investors, and analyst by providing supportive information like the future direction of the stock market. In this work, we present several Machine Learning algorithms and Long Short-Term Memory (LSTM) approach to predict stock market indices through two ways one is by analysis time series data and one is by processing news for stock prediction.

**INTRODUCTION**

Stock market prediction is a tough and challenging activity since the stock prices are directly influenced by numerous factors. The factors include political and government decisions, budgetary news, public safety events, etc. Public safety events may be man-made events like terrorist activities or natural disasters or it may be due to some epidemic. Recently, the outbreak of global pandemic novel Coronavirus (COVID 19) has greatly affected the stock market all over the world. In India, the trading activity had been stopped for about 45 minutes in an unprecedented manner. So, it is necessary to predict these fluctuations as early as possible so that the investors can make wiser decisions.

There are a lot of complicated financial indicators and also the fluctuation of the stock market is highly violent. However, as the technology is getting advanced, the opportunity to gain a steady fortune from the stock market is increased and it also helps experts to find out the most informative indicators to make a better prediction. The prediction of the market value is of great importance to help in maximizing the profit of stock option purchase while keeping the risk low. Recurrent neural networks (RNN) have proved one of the most powerful models for processing sequential data. Long Short-Term memory is one of the most successful RNNs architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

**Existing System**

Financial Market Prediction, In the past few decades, financial market prediction has become a new hot research topic in the machine learning field. With the aid of powerful models such as SVM (Support Vector Machine), feed forward neural network and recurrent neural network, researchers overcame numerous difficulties and achieved considerable progress. Their research results inspired our thesis. In this section, we will go through the representative ones among these studies.

**Literature survey:**

* **Title** : Optimized Radial Basis Functional Neural Network for Stock Index Prediction.
* **Author**: Rakhi Mahant, Trilok Nath Pandey, Alok Kumar Jagadev, and Satchidananda Dehuri
* **Abstract** : The art of forecasting the stock prices has been a difficult task for many of the researchers and analysts. In fact, investors are highly interested in the research area of stock price prediction. For a good and successful investment, many investors are keen in knowing the future situation of the stock market. Good and effective prediction systems for stock market help traders, investors, and analyst by providing supportive information like the future direction of the stock market. In this work, we present a recurrent neural network (RNN) and Long Short-Term Memory (LSTM) approach to predict stock market indices.
* **Title** : Parameters for Stock Market Prediction.
* **Author**: Prashant S. Chavan, Prof. Dr. Shrishail. T. Patil
* **Abstract** : In recent years researchers have developed a lot of interest in stock market prediction because of its dynamic & unpredictable nature. Although there were lots of methods of prediction none of them is prove to produce satisfactory results. Machine learning techniques proved to be better than other methods because of its ability of nonlinear mapping. In this paper we survey different input parameters that can be used for stock market prediction with ANN. In this paper we will try to find out most important input parameters that have major impact on accuracy. From the survey we see that most of machine learning techniques make use of Technical variables over fundamental variables for a particular stock price prediction, while Microeconomic variables are mostly used to predict stock market index. But hybridized parameters give better result that applying only single type of input variables.
* **Title**: Prediction of Stock Market Using Artificial Neural Network
* **Author**: Neelima Budhani, Dr. C. K. Jha, Sandeep K. Budhani
* **ABSTRACT:** Share Market is an untidy place for predicting since there are no significant rules to estimate or predict the price of share in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis etc are all used to attempt to predict the price in the share market but none of these methods are proved as a consistently acceptable prediction tool. Artificial Neural Network (ANN), a field of Artificial Intelligence (AI), is a popular way to identify unknown and hidden patterns in data which is suitable for share market prediction. For predicting of share price using ANN, there are two modules, one is training session and other is predicting price based on previously trained data. We used Backpropagation algorithm for training session and Multilayer Feedforward network as a network model for predicting price. In this paper, we introduce a method which can predict share market price using Backpropagation algorithm and Multilayer Feedforward network
* **Title** : A LSTM-based method for stock returns prediction: A case study of China stock market
* **Author:** Kai Chen, Yi Zhou and FangyanDai
* Abstract: Stock market prediction has always been an interesting research topic among researchers mainly due to its capital gain by trading stocks and or to understand the information hidden in stock market data. Many machine learning algorithms and statistical models have been proposed by researchers for stock price prediction and stock price movement prediction. We have studied various machine learning methods and techniques for stock market prediction. Here we present recent growth in stock market prediction methods and models, perform a comparison among these models to find out the accuracy of the prediction of the stock market values and also figuring out the advantages and disadvantages of these individual models. We are using LSTM and GRU models to predict future stock prices.
* **Title** : Detection of statistical arbitrage using machine learning techniques in Indian Stock market
* **Author:** A.U.S.S Pradeep, Soren Goyal, J. A. Bloom, I. J. Cox, and M. Miller
* **Abstract :** Pair trading strategy or statistical arbitrage strategy is a quantitative trading strategy that exploits the stock market that is out of equilibrium. Pair trading strategy is a market neutral strategy which means that we can make profit irrespective of the market trend. By choosing a pair of stocks that move together and assuming that their price is mean reverting, a trader can profit from the deviations from the mean by taking a long-short position in the chosen pair. This research analyses the performance of both Supervised and Unsupervised Machine Learning algorithms in Pair Trading and uses Python programming language to automate this trading strategy in the Indian market. Moreover, this research executes pair trading through a method called co-integration. In Supervised Learning approach, we use a Linear Regression model and in Unsupervised Learning approach, we use Principal Component Analysis for extracting risk factors of a stock and Density-Based Clustering for grouping the stock pairs together. Finally, this trading strategy was back-tested and programmed in Python for automatically triggering buy and short signals in the stock market.
* **Title**: NSE stock market prediction using deep-learning models
* **Author**: Hiransha. M, Gopalakrishnan. E. A, Menon. V. K and Soman. K. P
* **Abstract**: Hiransha.M. et. al. compare 4 deep neural networks namely Recurrent Neural Networks (RNN), Multi-Layer Perceptron (MLP), Convolution Neural Networks (CNN) and Long Short Term Memory units (LSTM) and concludes CNN performs well among other models [1].
* **Title**: CNNpred: CNN-based stock market prediction using a diverse set of variables , Stock Market Prediction Based on Generative Adversarial Network
* **Author**: Hoseinzade. E and Haratizadeh. S, Zhang K., Zhong, G., Dong, J., Wang, S. and Wang, Y.,
* **Abstract**: The prediction can be further improved by a hybrid network that combines other models. Two variants of CNN were introduced by Hoseinzade E. et. al. in which one variant can be applied to data collected from single source while the other variant can be applied to data collected from various markets [2]. But CNN is not suitable for time series prediction since its prediction does not depend on previous outcome. Zhang K. et. al., proposed Generative Adversarial Networks (GAN) based technique to predict close price of the stock where the generator is built using LSTM and discriminator is built using MLP [3]. Further, this model can be improved by using suitable optimizer and including other factors that affect financial data.
* **Title**: A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction
* **Author**: Chen. Y and Hao. Y
* **Abstract**: Four machine learning models, SVM, ANN, naïve-Bayes and random forest were used by Patel.J, et.al. with two different approaches for input data [9]. Only the closing price of a stock is predicted by them. But the prediction can be further improved by ensemble of various machine learning methods.
* **Title :** Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network
* **Author**: Minh D. L, Sadeghi-Niaraki A, Huy H. D., Min K, Moon H.,
* **Abstract**: The NBA was used with machine learning and deep learning models. A novel Two-stream Gated Recurrent Unit (TGRU) was proposed in [13] which uses both sentiment data as well as financial data. But the complexity of TGRU is doubled that of GRU and takes more training time and computational resources.

**Proposed System**

In our proposed approach we have used several Machine learning algorithms like support vector machine (SVM) , Random Forest (RF), Logistic Regression (LR) and applied k-fold cross validation over SVM algorithm to get optimized results and we have make 10 folds over training data. Another approach we used is neural network where LSTM stands out to be good for text processing.

In this stage, the data is fed to the neural network and trained for prediction assigning random biases and weights. Our LSTM model is composed of a sequential input layer followed by 2 LSTM layers and dense layer with ReLU activation and then finally a dense output layer with linear activation function. , the output value generated by the output layer of the RNN is compared with the target value. The error or the difference between the target and the obtained output value is minimized by using back propagation algorithm which adjusts the weights and the biases of the network.

**System Architecture**

Dataset

Pre-Processing

Data Analysis

Pre-processed Dataset

Graph Analysis

Pre-processed Dataset

RF/SVM/LR Classifier and LSTM NN

Predication Result

**SYSTEM REQUIREMENTS**

**Hardware:**

1. Windows 7,8,10 64 bit
2. RAM 4GB

**Software:**

1. Python 2.7 or 3.x / Anaconda Navigator

**Modules**

* **DATA COLLECTION**
* **DATA PRE-PROCESSING**
* **FEATURE EXTRATION**
* **Natural Language Processing**
* **EVALUATION MODEL**

**DATA COLLECTION**

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

**DATA PRE-PROCESSING**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

1. Formatting
2. Cleaning
3. Sampling

**Formatting:** The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

**Cleaning:** Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonym zed or removed from the data entirely.

**Sampling:** There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

**FEATURE EXTRATION**

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python

We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

Natural Language Processing

1. **Feature Selection and Preprocessing**

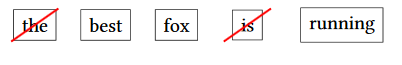
Feature selection and preprocessing are significant tasks in Artificial Intelligence and mainly represent the data preparation step in the CRISP-DM. Especially in NLP, this task does have tremendous impact on the success of text analysis. This is mostly caused by the unstructured and arbitrary nature of text data. Furthermore, machines need structure and numerical data. A couple of approaches for this transformation task e.g. word embedding or the vector space model, exist. This section’s scope lies on the theoretical foundation of different preprocessing and feature selection techniques. This section will be accompanied by the English phrase „the best fox is running “as an example to illustrate the application of preprocessing. Nevertheless, every routine should be used with care. It is not always the case that a reasonably good preprocessing method leads to better results in every application. The so-called no free lunch theorem makes it necessary to evaluate every suggested method separately in the practical part of this thesis.

1. **Tokenization**

For processing written natural language it is inevitable to split texts into smaller units, which are called tokens. Computers need to distinguish single entities of a text and tokenization is used to create them. Usually tokens represent simple words, which are the smallest independent units of natural language. Furthermore, token scan consists of idioms or a hyphens, e.g. „user-generated “. Tokenization breaks running texts into short text entities and is the very first task in any text preprocessing cycle. Besides the partition of small units, whole sentences can also be the output of a tokenizer. A simple word tokenizer can be realized in many languages by splitting the text at the occurrences of space symbols. This simple baseline approach does have a couple downsides, due to the lack of identifying words that semantically belong together. However, a simple tokenizer divides the phrase, which was introduced above, is into the following five tokens:



using tokens, so-called n-grams can be created, which indicate a token set with the length of n. „Gramma “is the Greek word for letter or token. When talking about a set of n letters in words, it is about characterngrams.2.2.2.2. Stop Word Removal A very important approach to reduce the huge raw input space in NLP is stop word removal(swr). Most languages have specific words, which do appear more often than others or do not include much information about the content of the text, e.g. auxiliary verbs or articles. Due to this, it often makes sense to exclude this so-called stop words in further analysis. In English such words could be "the", "a" or "an" and for German typical stop words are the articles "der", "die" and "das". The elimination could be done by checking the words against a standardized stop word list. These lists are available in literature and are often implemented in different software packages [DS17]. In our example, „the“ and„ is“ are eliminated. Stop word removal should be used with care, especially in sentiment analysis, which attempts to predict a positive or negative intention of a text. The removal would exclude words that are able to change a whole statement, such as „not “or „none“.



1. **Stemming**

Besides stop word elimination, stemming is a useful technique to map words to their word stems and further reduce the input dimension. This helps to extract the real meaning of a text and makes the unstructured data better accessible for a machine. The first stemming algorithm based on deleting longest suffixes and spelling exceptions was developed in 1968. By now, the porter stemming algorithm is a state-of-the-art approach and strips suffixes from words to retain the word stem. While this method performs well in English, there are some drawbacks for the German language, due to the fact that German words are not usually build by adding suffixes. However, there is a German equivalent based on Porter’s idea and the string processing language Snowball. By using the English Porter Stemmer, the words „best “, „fox “and „running “are assigned to the following words:

Best → best fox →fox running →run

1. **Lemmatization**

Lemmatization is the process of mapping every word in a text to their dictionary type or intended originating structure. Verbs are transformed to their infinite form, a noun is reconstructed to its singular representation and adverbs or adjectives anticipate their positive format [Liu+12]. The method is based on morphological analysis and often uses a dictionary, for example WordNet [Fel98], where the lemma of every modified word form could be retrieved. This preprocessing step is similar to stemming and reduces the input space, by mapping different word forms to their common representation. Natural Language Processing lemmatization is supported by dictionary entries; it is able to map „best “to its lemma„ good “:

Best → good fox→ fox running→ run

1. **Vector Space Model**

Besides, preprocessing the words themselves, their representations have to be changed into a machine readable format. Meanwhile, a couple of different approaches have been developed to transform texts into different kinds of numerical representations. Some of them only represent statistics of a word, such as the one-hot-encoding, and other formats also include the word’s context, e.g. word2vec The Vector Space Model is an approach that transforms a text into one vector. It is based on one-hot-encoding of words. Given a set of textual documents (corpus), it is possible to create a vocabulary with the length of N. The one-hot-encoded word vector represents a word by1at the corresponding vocabulary entry.

**EVALUATION MODEL**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, we have used Cross-Validation , to avoid under fitting, this method use a training set and then evaluate model performance on testing set. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

**Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

**ALGORITHM**

**RANDOM FOREST**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

**HOW RANDOM FOREST WORKS**

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

**ADVANTAGES OF USING RANDOM FOREST**

pros of using random forest for classification and regression.

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features.
4. The random forest algorithm also works well when data has missing values or it has not been scaled we.

**SVM(Support Vector Machine):**

More formally, a support-vector machine constructs a hyperplane or set of hyperplanes in a [high-](https://en.wikipedia.org/wiki/High-dimensional_space) or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier . Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly seperable in that space. For this reason, it was propose that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel functionk ( x , y ) {\displaystyle k(x,y)} selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane.

#### SVM Advantages

* SVM’s are very good when we have no idea on the data.
* Works well with even unstructured and semi structured data like text, Images and trees.
* The kernel trick is real strength of SVM. With an appropriate kernel function, we can solve any complex problem.
* Unlike in neural networks, SVM is not solved for local optima.

**Logistic Regression:**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

**Proposed Approach**

**Sentiment Analysis on IMDB Reviews using LSTM and Keras**

Steps

1. Load the dataset (50K IMDB Movie Review)
2. Clean Dataset
3. Encode Sentiments
4. Split Dataset
5. Tokenize and Pad/Truncate Reviews
6. Build Architecture/Model
7. Train and Test

Import all the libraries needed

**Stop Word** is a commonly used words in a sentence, usually a search engine is programmed to ignore this words (i.e. "the", "a", "an", "of", etc.)

Declaring the english stop words

Load and Clean Dataset

In the original dataset, the reviews are still dirty. There are still html tags, numbers, uppercase, and punctuations. This will not be good for training, so in **load\_dataset()** function, beside loading the dataset using **pandas**, I also pre-process the reviews by removing html tags, non alphabet (punctuations and numbers), stop words, and lower case all of the reviews.

Encode Sentiments

In the same function, I also encode the sentiments into integers (0 and 1). Where 0 is for negative sentiments and 1 is for positive sentiments.

Split Dataset

In this work, I decided to split the data into 80% of Training and 20% of Testing set using **train\_test\_split** method from Scikit-Learn. By using this method, it automatically shuffles the dataset. We need to shuffle the data because in the original dataset, the reviews and sentiments are in order, where they list positive reviews first and then negative reviews. By shuffling the data, it will be distributed equally in the model, so it will be more accurate for predictions.

Tokenize and Pad/Truncate Reviews

A Neural Network only accepts numeric data, so we need to encode the reviews. I use **tensorflow.keras.preprocessing.text.Tokenizer** to encode the reviews into integers, where each unique word is automatically indexed (using **fit\_on\_texts** method) based on **x\_train**.  
**x\_train** and **x\_test** is converted into integers using **texts\_to\_sequences** method.

Each reviews has a different length, so we need to add padding (by adding 0) or truncating the words to the same length (in this case, it is the mean of all reviews length) using **tensorflow.keras.preprocessing.sequence.pad\_sequences**.

**post**, pad or truncate the words in the back of a sentence  
**pre**, pad or truncate the words in front of a sentence

Build Architecture/Model

**Embedding Layer**: in simple terms, it creates word vectors of each word in the *word\_index* and group words that are related or have similar meaning by analyzing other words around them.

**LSTM Layer**: to make a decision to keep or throw away data by considering the current input, previous output, and previous memory. There are some important components in LSTM.

* **Forget Gate**, decides information is to be kept or thrown away
* **Input Gate**, updates cell state by passing previous output and current input into sigmoid activation function
* **Cell State**, calculate new cell state, it is multiplied by forget vector (drop value if multiplied by a near 0), add it with the output from input gate to update the cell state value.
* **Ouput Gate**, decides the next hidden state and used for predictions

**Dense Layer**: compute the input with the weight matrix and bias (optional), and using an activation function. I use **Sigmoid** activation function for this work because the output is only 0 or 1.

The optimizer is **Adam** and the loss function is **Binary Crossentropy** because again the output is only 0 and 1, which is a binary number.

Training

For training, it is simple. We only need to fit our **x\_train** (input) and **y\_train** (output/label) data. For this training, I use a mini-batch learning method with a **batch\_size** of *128* and *5* **epochs**.

Also, I added a callback called **checkpoint** to save the model locally for every epoch if its accuracy improved from the previous epoch.

Testing

To evaluate the model, we need to predict the sentiment using our **x\_test** data and comparing the predictions with **y\_test** (expected output) data. Then, we calculate the accuracy of the model by dividing numbers of correct prediction with the total data. Resulted an accuracy of **86.63%**

**Domain Specification**

**MACHINE LEARNING**

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

**Machine Learning vs. Traditional Programming**

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

**COMPUTER**

**DATA RULES**

**Machine Learning**

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and**inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi3.png)

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

#### Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi4.png)

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

## Machine learning Algorithms and where they are used?

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi5.png)

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

#### Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

* Classification task
* Regression task

#### Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

#### Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

|  |  |  |
| --- | --- | --- |
| **Algorithm Name** | **Description** | **Type** |
| **Linear regression** | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| **Logistic regression** | Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) | Classification |
| **Decision tree** | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made | Regression Classification |
| **Naive Bayes** | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| **Support vector machine** | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| **Random forest** | The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Regression Classification |
| **AdaBoost** | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| **Gradient-boosting trees** | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

#### Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Description** | **Type** |
| **K-means clustering** | Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) | Clustering |
| **Gaussian mixture model** | A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters | Clustering |
| **Hierarchical clustering** | Splits clusters along a hierarchical tree to form a classification system.  Can be used for Cluster loyalty-card customer | Clustering |
| **Recommender system** | Help to define the relevant data for making a recommendation. | Clustering |
| **PCA/T-SNE** | Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances. | Dimension Reduction |

**Application of Machine learning**

**Augmentation**:

* Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

**Automation**:

* Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**Finance Industry**

* Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

**Government organization**

* The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

**Healthcare industry**

* Healthcare was one of the first industry to use machine learning with image detection.

**Marketing**

* Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

**Example of application of Machine Learning in Supply Chain**

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

**Example of Machine Learning Google Car**

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**Reinforcement Learning**

Reinforcement learningis a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

* Q-learning
* Deep Q network
* State-Action-Reward-State-Action (SARSA)
* Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

**AI in Finance:**The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

**AI in HR:**Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose HireVue, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

**AI in Marketing:**AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

**Artificial Intelligence**

**Deep Learning**

**Machine Learning**

ML

**REQUIREMENTS ANAYLSIS**

**SOFTWARE REQUIREMENTS**

* Python
* Anaconda Navigator
* Python built-in modules
* Numpy
* Pandas
* Matplotlib
* Sklearn
* Seaborm

**ANACONDA NAVIGATOR**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

## Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

## **WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR**?

The following applications are available by default in Navigator:

* JupyterLab
* Jupyter Notebook
* QTConsole
* Spyder
* VSCode
* Glueviz
* Orange 3 App
* Rodeo
* RStudio

Advanced conda users can also build your own Navigator applications

## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

## What’s new in 1.9?

* Add support for **Offline Mode** for all environment related actions.
* Add support for custom configuration of main windows links.
* Numerous bug fixes and performance enhancements.

**PYTHON OVERVIEW**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted:** Python is processed at runtime by the interpreter.You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
* **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Python Features**

Python's features include:

* **Easy-to-learn:** Python has few keywords, simple structure, and a clearlydefined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
* **A broad standard library:** Python's bulk of the library is very portable andcross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.
* **Portable:** Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.
* **Extendable:** You can add low-level modules to the Python interpreter. Thesemodules enable programmers to add to or customize their tools to be more efficient.
* **Databases:** Python provides interfaces to all major commercial databases.
* **GUI Programming:** Python supports GUI applications that can be created andported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable:** Python provides a better structure and support for large programsthan shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* IT supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

**Python’s standard library**

* Pandas
* Numpy
* Sklearn
* seaborn
* matplotlib

**PANDAS**

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of the most preferred and widely used tools in [data munging/wrangling](https://en.wikipedia.org/wiki/Data_wrangling) if not THE most used one. Pandas is an open source

What’s cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

**Installation and Getting Started**

In order to “get” Pandas you would need to install it. You would also need to have Python 2.7 and above as a pre-requirement for installation. It is also dependent on other libraries (like [NumPy](http://www.numpy.org/)) and has optional dependancies (like Matplotlib for plotting). Therefore, I think that the easiest way to get Pandas set up is to install it through a package like the [Anaconda distribution](https://www.continuum.io/downloads), “a cross platform distribution for data analysis and scientific computing.”

In order to use Pandas in your Python IDE ([Integrated Development Environment](https://en.wikipedia.org/wiki/Integrated_development_environment)) like [Jupyter Notebook](http://jupyter.org/" \t "_blank) or [Spyder](https://pythonhosted.org/spyder/) (both of them come with Anaconda by default), you need to import the Pandas library first. Importing a library means loading it into the memory and then it’s there for you to work with. In order to import Pandas all you have to do is run the following code:

* **import pandas as pd**
* **import numpy as np**

Usually you would add the second part (‘as pd’) so you can access Pandas with ‘pd.command’ instead of needing to write ‘pandas.command’ every time you need to use it. Also, you would import numpy as well, because it is very useful library for scientific computing with Python. Now Pandas is ready for use! Remember, you would need to do it every time you start a new Jupyter Notebook, Spyder file etc.

**Working with Pandas**

Loading and Saving Data with Pandas

When you want to use Pandas for data analysis, you’ll usually use it in one of three different ways:

* Convert a Python’s list, dictionary or Numpy array to a Pandas data frame
* Open a local file using Pandas, usually a CSV file, but could also be a delimited text file (like TSV), Excel, etc
* Open a remote file or database like a CSV or a JSONon a website through a URL or read from a SQL table/database

There are different commands to each of these options, but when you open a file, they would look like this:

* **pd.read\_filetype()**

As I mentioned before, there are different filetypes Pandas can work with, so you would replace “filetype” with the actual, well, filetype (like CSV). You would give the path, filename etc inside the parenthesis. Inside the parenthesis you can also pass different arguments that relate to how to open the file. There are numerous arguments and in order to know all you them, you would have to read the documentation (for example, the [documentation for pd.read\_csv()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html)would contain all the arguments you can pass in this Pandas command).

In order to convert a certain Python object (dictionary, lists etc) the basic command is:

* **pd.DataFrame()**

Inside the parenthesis you would specify the object(s) you’re creating the data frame from. This command also has [different arguments](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html).

You can also save a data frame you’re working with/on to different kinds of files (like CSV, Excel, JSON and SQL tables). The general code for that is:

* **df.to\_filetype(filename)**

**Viewing and Inspecting Data**

Now that you’ve loaded your data, it’s time to take a look. How does the data frame look? Running the name of the data frame would give you the entire table, but you can also get the first n rows with df.head(n) or the last n rows with df.tail(n). df.shape would give you the number of rows and columns. df.info() would give you the index, datatype and memory information. The command s.value\_counts(dropna=False) would allow you to view unique values and counts for a series (like a column or a few columns). A very useful command is df.describe() which inputs summary statistics for numerical columns. It is also possible to get statistics on the entire data frame or a series (a column etc):

* df.mean() Returns the mean of all columns
* df.corr() Returns the correlation between columns in a data frame
* df.count() Returns the number of non-null values in each data frame column
* df.max()Returns the highest value in each column
* df.min()Returns the lowest value in each column
* df.median()Returns the median of each column
* df.std()Returns the standard deviation of each column

**Selection of Data**

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary. You can select a column (df[col]) and return column with label col as Series or a few columns (df[[col1, col2]]) and returns columns as a new DataFrame. You can select by position (s.iloc[0]), or by index (s.loc['index\_one']) . In order to select the first row you can use df.iloc[0,:] and in order to select the first element of the first column you would run df.iloc[0,0] . These can also be used in different combinations, so I hope it gives you an idea of the different selection and indexing you can perform in Pandas.

**Filter, Sort and Groupby**

You can use different conditions to filter columns. For example, df[df[year] > 1984] would give you only the column year is greater than 1984. You can use & (and) or | (or) to add different conditions to your filtering. This is also called boolean filtering.

It is possible to sort values in a certain column in an ascending order using df.sort\_values(col1) ; and also in a descending order using df.sort\_values(col2,ascending=False). Furthermore, it’s possible to sort values by col1 in ascending order then col2 in descending order by using df.sort\_values([col1,col2],ascending=[True,False]).

The last command in this section is groupby. It involves splitting the data into groups based on some criteria, applying a function to each group independently and combining the results into a data structure. df.groupby(col) returns a groupby object for values from one column while df.groupby([col1,col2]) returns a groupby object for values from multiple columns.

**Data Cleaning**

Data cleaning is a very important step in data analysis. For example, we always check for missing values in the data by running pd.is null() which checks for null Values, and returns a boolean array (an array of true for missing values and false for non-missing values). In order to get a sum of null/missing values, run pd. Is null().sum(). Pd .not null() is the opposite of pd. Is null(). After you get a list of missing values you can get rid of them, or drop them by using df. Drop na() to drop the rows or df. drop na(axis=1) to drop the columns. A different approach would be to fill the missing values with other values by using df. Fill na(x) which fills the missing values with x (you can put there whatever you want) or s .fill na(s.mean()) to replace all null values with the mean (mean can be replaced with almost any function from the statistics section).

It is sometimes necessary to replace values with different values. For example, s. replace(1,'one') would replace all values equal to 1 with 'one'. It’s possible to do it for multiple values: s. replace([1,3],['one', 'three'])would replace all 1 with 'one' and 3 with 'three'. You can also rename specific columns by running:  df. rename(columns={'old\_name': 'new\_ name'})or use df. set\_ index('column\_one') to change the index of the data frame.

**Join/Combine**

The last set of basic Pandas commands are for joining or combining data frames or rows/columns. The three commands are:

* df1.append(df2)— add the rows in df1 to the end of df2 (columns should be identical)
* df. concat([df1, df2],axis=1) — add the columns in df1 to the end of df2 (rows should be identical)
* df1.join(df2,on=col1,how='inner') — SQL-style join the columns in df1with the columns on df2 where the rows for col have identical values. how can be equal to one of: 'left', 'right', 'outer', 'inner'

**NUMPY**

Numpy is one such powerful library for array processing along with a large collection of high-level mathematical functions to operate on these arrays. These functions fall into categories like Linear Algebra, Trigonometry, Statistics, Matrix manipulation, etc.

Getting NumPy

NumPy’s main object is a homogeneous multidimensional array. Unlike python’s array class which only handles one-dimensional array, NumPy’s nd array class can handle multidimensional array and provides more functionality. NumPy’s dimensions are known as axes. For example, the array below has 2 dimensions or 2 axes namely rows and columns. Sometimes dimension is also known as a rank of that particular array or matrix.

#### Importing NumPy

NumPy is imported using the following command. Note here np is the convention followed for the alias so that we don't need to write numpy every time.

* import numpy as np

NumPy is the basic library for scientific computations in Python and this article illustrates some of its most frequently used functions. Understanding NumPy is the first major step in the journey of machine learning and deep learning.

#### 

**Sk learn**

In python, scikit-learn library has a pre-built functionality under sk learn. Pre processing.

Next thing is to do feature extraction Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally our models are trained using Classifier algorithm.. We use nltk . classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered . The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre processed data. The chosen classifiers were Decision tree , Support Vector Machines and Random forest. These algorithms are very popular in text classification tasks.

**SEABORN**

# **Data Visualization in Python**

Data visualization is the discipline of trying to understand data by placing it in a visual context, so that patterns, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has a excellent library for you.

**To get a little overview here are a few popular plotting libraries:**

* [**Matplotlib:**](https://matplotlib.org/)low level, provides lots of freedom
* [**Pandas Visualization:**](https://pandas.pydata.org/pandas-docs/stable/visualization.html)easy to use interface, built on Matplotlib
* [**Seaborn:**](https://seaborn.pydata.org/)high-level interface, great default styles
* [**ggplot:**](http://ggplot.yhathq.com/)based on R’s ggplot2, uses [Grammar of Graphics](https://www.amazon.com/Grammar-Graphics-Statistics-Computing/dp/0387245448)
* [**Plotly:**](https://plot.ly/python/)can create interactive plots

In this article, we will learn how to create basic plots using Matplotlib, Pandas visualization and Seaborn as well as how to use some specific features of each library. This article will focus on the syntax and not on interpreting the graphs.

Matplotlib

Matplotlib is the most popular python plotting library. It is a low level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code.

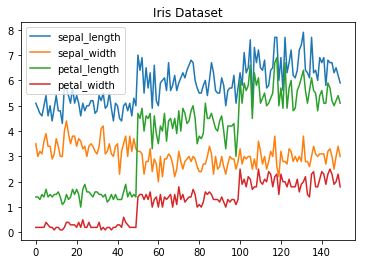
1. To install Matplotlib pip anaconda can be used.
2. pip install matplotlib
3. conda install matplotlib

Matplotlib is specifically good for creating basic graphs like line charts, bar charts, histograms and many more. It can be imported by typing:

* **import matplotlib.pyplot as plt**

#### Line Chart

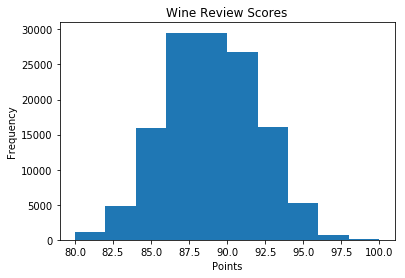
In Matplotlib we can create a line chart by calling the plot method. We can also plot multiple columns in one graph, by looping through the columns we want, and plotting each column on the same axis.



**Line Chart**

#### Histogram

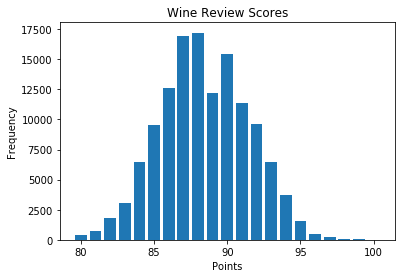
In Matplotlib we can create a Histogram using the hist method. If we pass it categorical data like the points column from the wine-review dataset it will automatically calculate how often each class occurs.



**Histogram**

#### Bar Chart

A bar-chart can be created using the bar method. The bar-chart isn’t automatically calculating the frequency of a category so we are going to use pandas value\_counts function to do this. The bar-chart is useful for categorical data that doesn’t have a lot of different categories (less than 30) because else it can get quite messy.



**Bar-Chart**

Pandas Visualization

Pandas is a open source high-performance, easy-to-use library providing data structures, such as dataframes, and data analysis tools like the visualization tools we will use in this article.

Pandas Visualization makes it really easy to create plots out of a pandas dataframe and series. It also has a higher level API than Matplotlib and therefore we need less code for the same results.

## Pandas can be installed using either pip or conda.

## pip install pandas

## conda install pandas

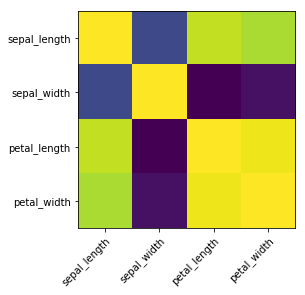
#### Heatmap

A Heatmap is a graphical representation of data where the individual values contained in a [matrix](https://en.wikipedia.org/wiki/Matrix_%28mathematics%29) are represented as colors. Heatmaps are perfect for exploring the correlation of features in a dataset.

To get the correlation of the features inside a dataset we can call <dataset>.corr() , which is a Pandas dataframe method. This will give use the [correlation matrix](https://www.displayr.com/what-is-a-correlation-matrix/).

We can now use either Matplotlib or Seaborn to create the heatmap.

**Matplotlib:**



**Heatmap without annotations**

Data visualization is the discipline of trying to understand data by placing it in a visual context, so that patterns, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. In this article we looked at Matplotlib, Pandas visualization and Seaborn.

**UML DIAGRAMS:**

**UML DIAGRAMS**

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

* actors
* business processes
* (logical) components
* activities
* programming language statements
* database schemas, and
* Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

**Sequence Diagram:**

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

* **Activity Diagrams-:**
* Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency.In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**Usecase diagram:**

* UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
* UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
* OMG is continuously putting effort to make a truly industry standard.
* UML stands for **U**nified **M**odeling **L**anguage.
* UML is a pictorial language used to make software blue prints

**Class diagram**

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

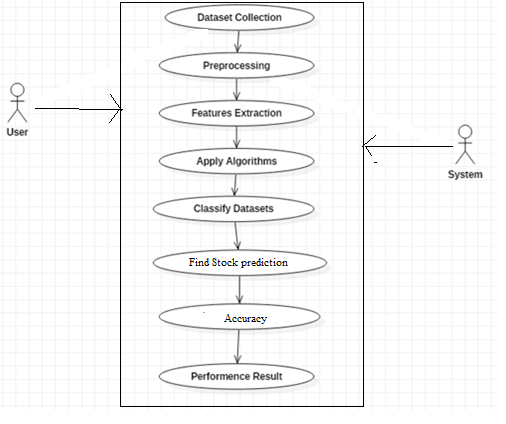
In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

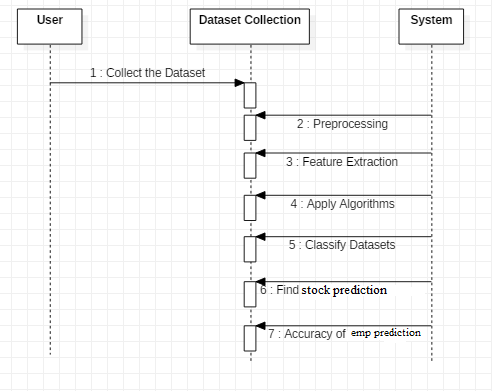
**Use Case :**



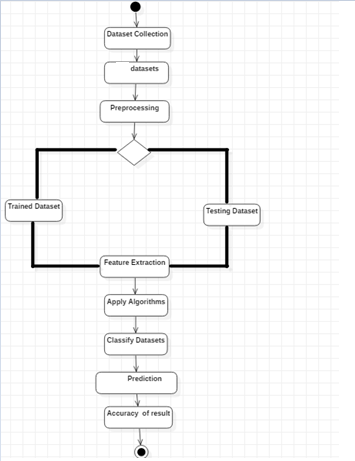
**Class Diagram :**



**Sequence Diagram:**



**Activity Diagram:**



**TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

* Meets the business and technical requirements that guided its design and Development.
* Works as expected and can be implemented with the same characteristics.

**TESTING METHODS**

**Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Valid messages: Identified classes of valid messages must be accepted.
* Invalid messages: Identified classes of invalid messages must be rejected.
* Functions: Identified functions must be exercised.
* Output: Identified classes of application outputs must be exercised.
* Systems/Procedures: Interfacing systems or procedures must be invoked.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

**TEST CASES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SL #** | **TEST CASE NAME** | **DESCRIPTION** | **STEP NO** | **ACTION TO BE TAKEN (DESIGN STEPS)** | **EXPECTED (DESIGN STEP)** | **Test Execution Result ( PASS/FAIL)** |
| 1 | Excel Sheet verification | **Objective:** There should be an excel sheet. Any number of rows can be added to the sheet. | Step 1 | Excel sheet should be available | Excel sheet is available | Pass |
|  |  |  | Step 2 | Excel sheet is created based on the template | The excel sheet should always  be based on the template | Pass |
|  |  |  | Step 3 | Changed the name of excel sheet | Should not make any modification on the name of excel sheet | Fail |
|  |  |  | Step 4 | Added 10000 or above records | Can add any number of records | Pass |

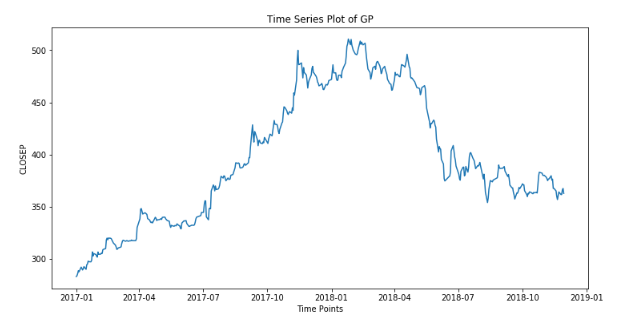
**Results: **

Fig 1. Time series analysis between closing price and dates

****

Fig 2. Candlestick analysis between closing price vs opening price vs high vs low and dates

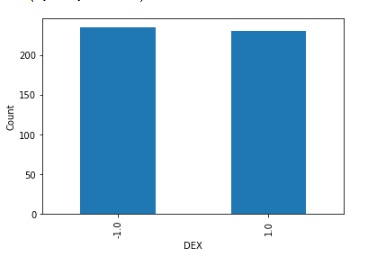
****

Fig 3. Graph for Class Count

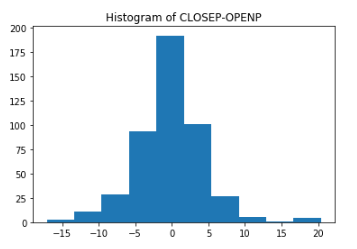
****

Fig 4. Time series analysis between closing price and opening price

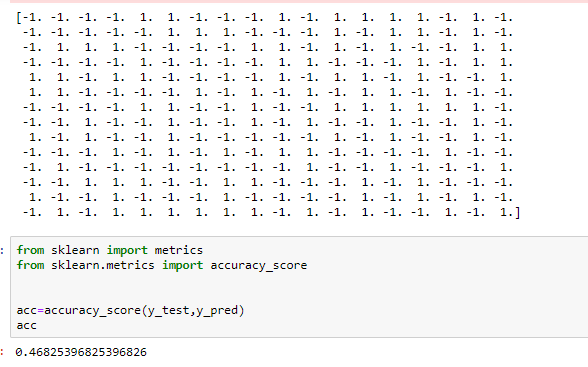


Fig. 5 Accuracy result - SVM with k-fold cross validation

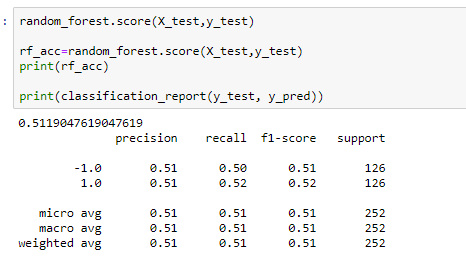


Fig. 6 Accuracy result – Random Forest

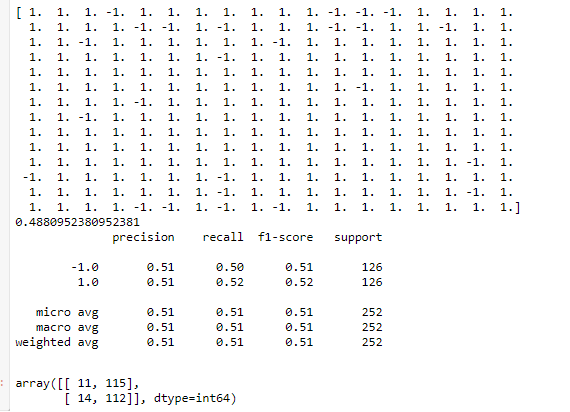


Fig. 7 Result with Logistic Regression

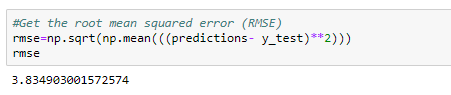


Fig. 8 Result with LSTM

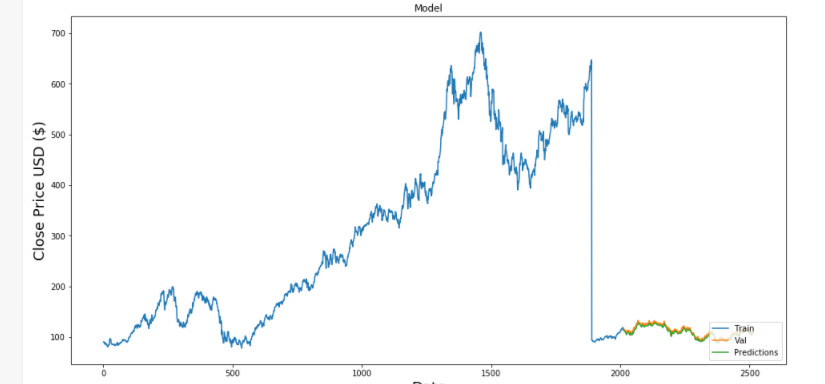


Fig. 9 Closing Price Prediction for upcoming days



Fig. 10 Wordcloud over Stock News For sentiment purpose

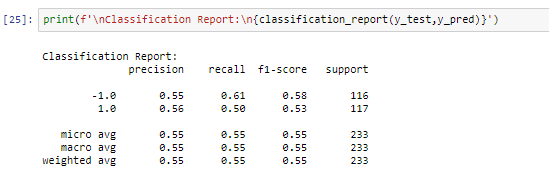


Fig. 11 Sentiment analysis accuracy

**Conclusions :**

The popularity of stock market trading is growing rapidly, which is encouraging researchers to find out new methods for the prediction using new techniques. The forecasting technique is not only helping the researchers but it also helps investors and any person dealing with the stock market. In order to help predict the stock indices, a forecasting model with good accuracy is required.

In this work, we have used forecasting technology using machine learning algorithms like SVM, RF, LR and LSTM which helps analysts through two ways one is with time series data or another with stock news based sentiment or any person interested in investing in the stock market by providing them a good knowledge of the future situation of the stock market.

**References**

[1] Hiransha. M, Gopalakrishnan. E. A, Menon. V. K and Soman. K. P, “NSE stock market prediction using deep-learning models”, Procedia computer science, Vol. 132, pp. 1351-1362, 2018.

[2] Hoseinzade. E and Haratizadeh. S, “CNNpred: CNN-based stock market prediction using a diverse set of variables”, Expert Systems with Applications, Vol. 129, pp. 273-285, 2019.

[3] Zhang K., Zhong, G., Dong, J., Wang, S. and Wang, Y., “Stock Market Prediction Based on Generative Adversarial Network”, Procedia computer science, Vol. 147, pp. 400-406, 2019.

[4] Patel. J, Shah. S, Thakkar. P and Kotecha. K, “Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques”, Expert Systems with Applications, Vol. 42(1), pp.259-268, 2015.

[5] Minh D. L, Sadeghi-Niaraki A, Huy H. D., Min K, Moon H., “Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network”, IEEE Access, Vol. 6, pp. 55392-404, Sept 2018.

[6] RakhiMahant, TrilokNathPandey, Alok Kumar Jagadev, and SatchidanandaDehuri ―Optimized Radial Basis Functional Neural Network for Stock Index Prediction,‖ International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016.

[7] Kai Chen, Yi Zhou and FangyanDai ―A LSTM-based method for stock returns prediction: A case study of China stock market,‖ IEEE International Conference on Big Data,2015.

[8] A.U.S.S Pradeep, SorenGoyal, J. A. Bloom, I. J. Cox, and M. Miller, ―Detection of statistical arbitrage using machine learning techniques in Indian Stock market,‖ IIT Kanpur, April 15, 2013.

[9] Prashant S. Chavan, Prof. Dr. Shrishail. T. Patil ―Parameters for Stock Market Prediction,‖ Prashant S Chavan et al, Int.J.Computer Technology & Applications, Vol 4 (2),337-340.

[10] NeelimaBudhani, Dr. C. K. Jha, Sandeep K. Budhani―Prediction of Stock Market Using Artificial Neural Network,‖ International Conference on Soft Computing Techniques for Engineering and Technology (ICSCTET)- 2014. [6] SharvilKatariya, Saurabh Jain―Stock Price Trend Forecasting using Supervised Learning Methods.

[11] DhirajMundada, GauravChhaparwal, SachinChaudhari, and TruptiBhamare― Stock Value Prediction System,‖ International Journal on Recent and Innovation Trends in Computing and Communication, April 2015

[12] Hochreiter, Sepp, and Jürgen Schmidhuber― Long short-term memory,‖ Neural computation 9.8 (1997): 1735-1780