Corona Outbreak Visualisation and Prediction Using Machine Learning

Summer Training Report submitted in partial fulfilment of the requirement for the degree of

B.Tech

In

Computer Science &Engineering

Mrs. Himani Sharma

(Faculty coordinator of class)

By

Ashish Upadhyay
(01820802718)

Bhagwan Parshuram Institute of Technology

PSP-4, Sector-17, Rohini, Delhi - 89

June-July 2020

DECLARATION

This is to certify that Report entitled “**Corona Outbreak Visualisation and Prediction using Machine Learning**” which is submitted by me in partial fulfilment of the requirement for the award of degree B.Tech in Computer Engineering to BPIT, GGSIP University, Dwarka, Delhi comprises only my original work and due acknowledgement has been made in the text to all other material used.

Date: **Name of Student**

October 30,2020 Ashish Upadhyay
(01820801718)

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

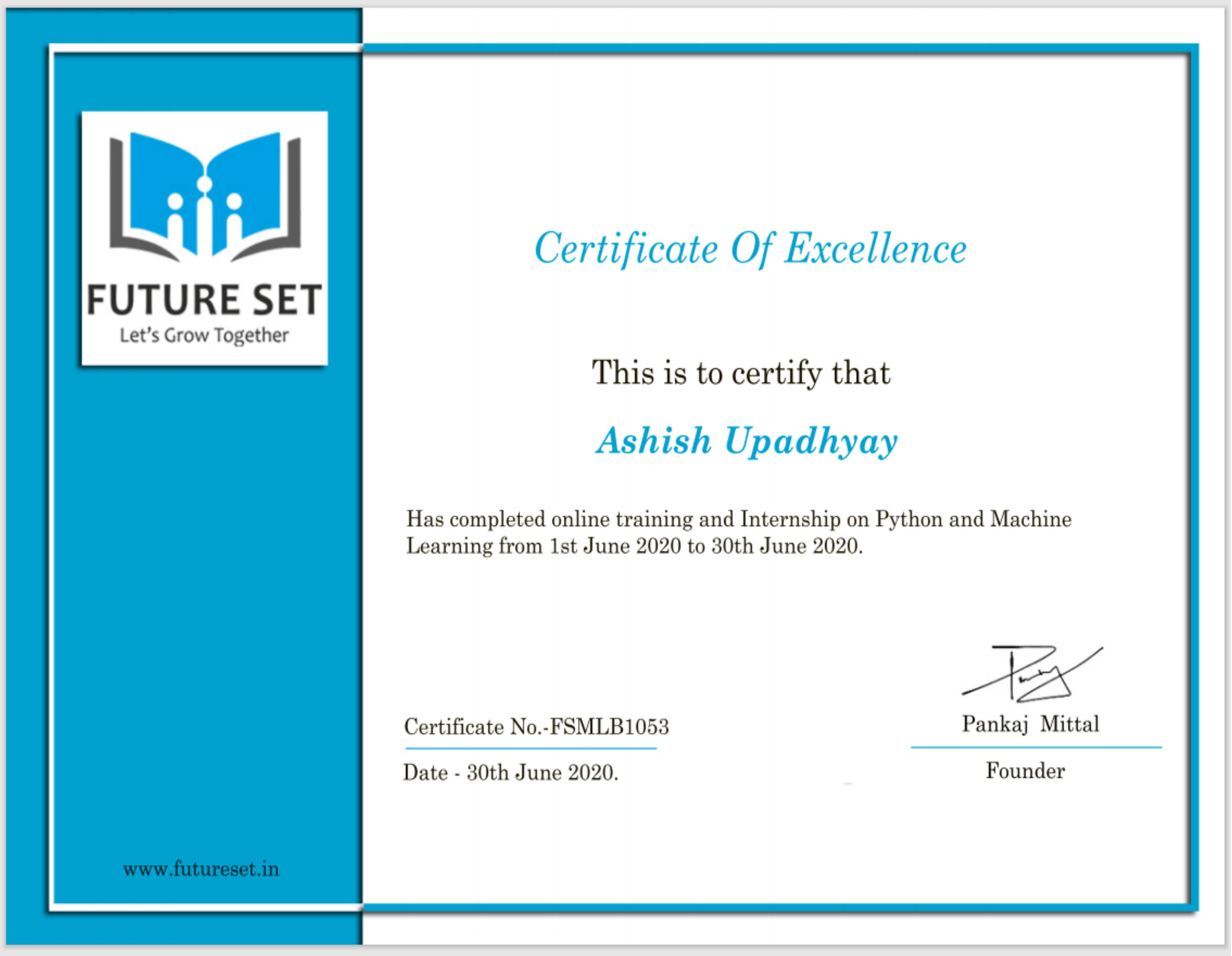
I am highly indebted to organization *Future Set* for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards *my parents & member of Future Set* for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons especially *Mr. Vikas Dubey* who guided me throughout this project and for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Company Certificate



FUTURE SET

Let's Grow Together

www.futureset.in

Certificate Of Excellence

This is to certify that

Ashish Upadhyay

Has completed online training and Internship on Python and Machine Learning from 1st June 2020 to 30th June 2020.

Certificate No.-FSMLB1053

Date - 30th June 2020.

Pankaj Mittal

Founder

Training Coordinator Certificate

This is to certify that Report entitled "***Corona Outbreak Visualisation and Prediction using Machine Learning***" which is submitted by ***Ashish Upadhyay*** in partial fulfilment of the requirement for the award of degree B.Tech in Computer Engineering to BPIT, GGSIP University, Dwarka, Delhi is a record of the candidate own work and the matter embodied in this report is adhered to the given format.

October 30,2020 **Mrs. Himani Sharma**

Date: **Coordinator**

List of Content

List of figures

List of tables

ABSTRACT

Background and objective

COVID-19 outbreak was first reported in Wuhan, China and has spread to nearly whole world. WHO declared COVID-19 as a Public Health Emergency of International Concern (PHEIC) on 30 January . Naturally, a rising infectious disease involves fast spreading, endangering the health of large numbers of people, and thus requires immediate actions to prevent the disease at the community level. Therefore, this model go through the previous data and gives out upcoming cases as well as statistics and analysis on COVID-19. This model aims to predict and forecast COVID19 cases, deaths, and recoveries through predictive modelling. This aims to comprehensively review the role of ML as one significant method in the arena of screening, predicting, forecasting, contact tracing, for SARS-CoV-2 and its related epidemic.

Method

A selective assessment of information on the model was executed on the databases related to the application of ML and technology on Covid-19. Rapid and critical analysis of the three crucial parameters, i.e., abstract, methodology, and the conclusion was done to relate to the model's possibilities for tackling the SARS-CoV-2 epidemic.

Result

This paper addresses on recent studies that apply ML technology towards augmenting the researchers on multiple angles. It also addresses a few errors and challenges while using such algorithms in real-world problems. The model also discusses suggestions conveying researchers on model design, medical experts, and policymakers in the current situation while tackling the Covid-19 pandemic and ahead.

Conclusion

COVID-19 is still an unclear infectious disease, which means we can only obtain an accurate SEIR prediction after the outbreak ends. The outbreak spreads are largely influenced by each country's policy and social responsibility. As data transparency is crucial inside the government, it is also our responsibility not to spread unverified news and to remain calm in this situation. This project has shown the importance of information dissemination that can help in improving response time, and help planning in advance to help reduce risk. Further studies need to be done to help contain the outbreak as soon as possible.

Keywords: COVID-19, data analysis, sentiment analysis, predictive modelling, SEIR

CHAPTER 1

Intro with COVID-19

1.1 INTRODUCTION

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus.

Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

The best way to prevent and slow down transmission is to be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face.

The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it’s important that you also practice respiratory etiquette (for example, by coughing into a flexed elbow). This project aims to predict future cases by analyzing data of patients using machine-learning algorithms.

1.2 PROBLEM

There are several disease outbreaks that invaded humanity in World history. World Health Organization (WHO), its co-operating clinicians and various national authorities around the globe fight against these pandemics to date. Since the first Covid-19 (Coronavirus) disease case confirmed in China December 2019 Wuhan District, the outbreak continues to spread all across the world, and on 30th January 2020 WHO declared the pandemic as an international concern of public health emergency . The novel Coronavirus (SARS-CoV-2) disease spread on more than 185 countries infecting more than 7,145,800 individuals and causing 407,067 deaths by June 09, 2020. To address this global novel pandemic, WHO, scientists and clinicians in medical industries are searching for new technology to screen infected patients in various stages, find best clinical trials, control the spread of this virus, develop a vaccine for curing infected patients, trace contact of the infected patient.

1.3 MOTIVATION

Recent studies identified that Machine Learning and Artificial Intelligence are promising technology employed by various healthcare providers as they result in better scale-up, speed-up processing power, reliable and even outperform human in specific healthcare tasks. Therefore, healthcare industries and clinicians worldwide employed various ML and AI technology to tackle the Covid-19 pandemic to address the challenges during the outbreak. In medical industries, AI is not applied to replace the human interactions, but to provide decision support for clinicians on what they are modeled for.

This project focuses on the novel Covid-19 epidemic and how the modern AI and ML technology were recently employed to solve the challenges during the outburst. We present comprehensive reviews of studies on the model and technology applied to tackle the novel Covid-19 pandemic. The studies further discuss types of AI and ML methods recently employed integration and types of the dataset, the final performance of each proposed model, and present on the pros and cons of modern techniques.

CHAPTER – 2

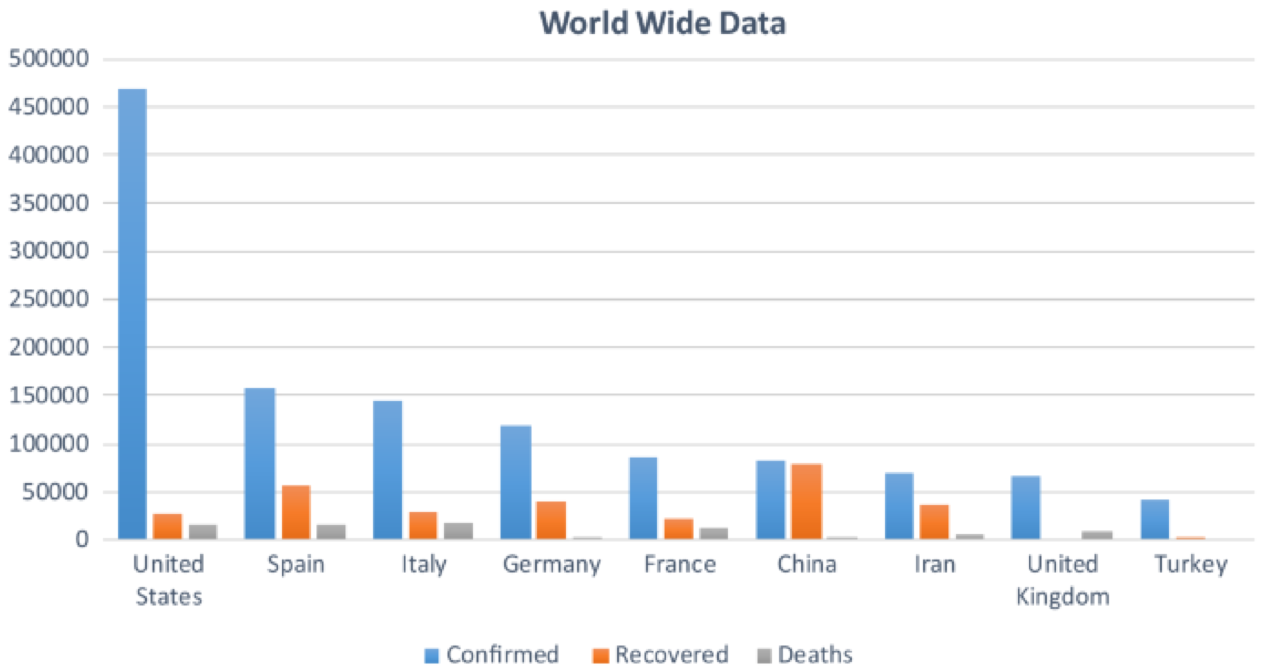
Possibilities of ML & AI in analysing COVID-19

2.1 ML and AI recently employed to tackle health care SARS-CoV-2 outbreak

AI and ML technology are used to improve the accuracy of prediction for screening both infectious and non-infectious diseases. The relation with health care begins with the evolution of the first expert system called MYCIN developed in 1976. MYCIN was designed to use 450 rules collected from a medical expert to treat bacterial infection by suggesting antibiotics to the patients. Such an expert system serves as clinical decision support for clinicians and medical experts. Recent studies evident on the prospect of ML and AI technology for the various pandemic outbreak, it supports healthcare experts in various communicable diseases (SARS, EBOLA, HIV, COVID_19) and non-communicable diseases (Cancer, Diabetic, Heart disease, and Stroke) outbreak.

2.2 ML and AI technology in COVID -19 screening and treatment

Early detection of any disease, be it infectious and non-infectious, is critically an important task for early treatment to save more lives . Fast diagnosis and screening process helps prevent the spread of pandemic diseases like SARS-CoV-2, cost-effective, and speed up the related diagnosis. The development of an expert system for health care assists in the new order of identification screening and management of SARS-CoV-2 carrier by more cost-effective compared to the traditional method. ML and AI are used to augment the diagnosis and screening process of the identified patient with radio imaging technology akin to Computed Tomography (CT), X-Ray, and Clinical blood sample data. The healthcare expert can use radiology images like X-ray and CT scans as routine tools to augment traditional diagnosis and screening. Unfortunately, the performance of such devices is moderate during the high outburst of the SARS-CoV-2 pandemic. In this regard, studies show the potential of AI and ML tools by suggesting a new model that comes with rapid and valid method SARS-CoV-2 diagnosis using Deep Convolutional Network. The study shows that diagnosis utilizing an expert system employing AI and ML on 1020 CT images of 108 Covid-19 infected patients along with viral pneumonia of 86 patients, the remarkable performance suggests the use of the convolutional neural network (Resnet-101) as an adjuvant tool for radiologist resulting 86.27%, 83.33% of accuracy and specificity respectively.

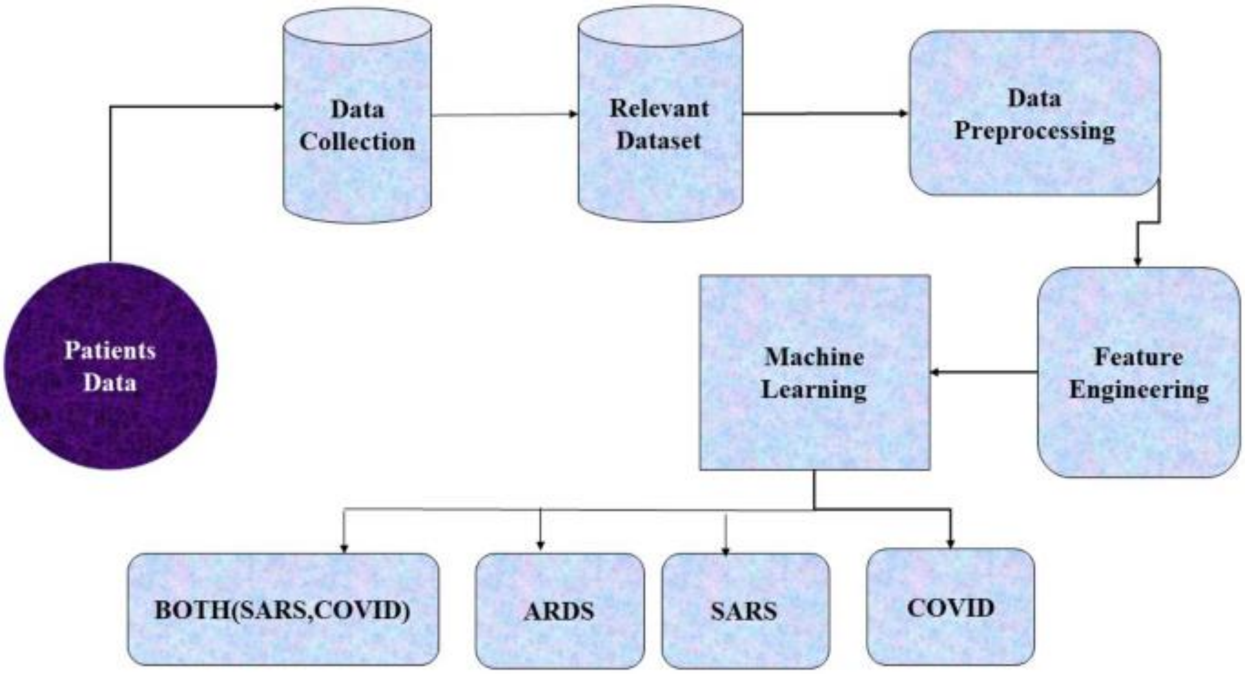


2.3 ML and AI technology in SARS-Cov-2 contact tracing

If a person diagnoses and is confirmed with Covid-19, the next important step is contact tracing prevention of the wider spread of the disease. According to WHO, the infection spreads from person-to-person primarily through saliva, droplets, or discharges from the nose through contact transmission. To take control on the spread of SARS-Cov-2, contact tracing is an essential public health tool used to break the chain of virus transmission. The process of contact tracing is to identify and manage people who are recently exposed to an infected Covid-19 patient to avoid further spread. Generally, the process identifies the infected person with a follow-up for 14 days since the exposure. If employed thoroughly, this process can break the transmission chain of the current novel coronavirus and suppress the outbreak by giving a higher chance of adequate controls and helping reduce the magnitude of the recent pandemic. In this regard, various infected countries come up with a digital contact tracing process with the mobile application, utilizing different technologies like Bluetooth, Global Positioning System (GPS), Social graph, contact details, network-based API, mobile tracking data, card transaction data, and system physical address. The digital contact tracing process can perform virtually real-time and much faster compared to the non-digital system. All these digital apps are designed to collect individual personal data, which will be analyzed by ML and AI tools to trace a person who is vulnerable to the novel virus due to their recent contacted chain.

2.4 ML and AI technology in SARS-CoV-2 prediction and forecasting

A new novel model, that forecast and predicting 1-3 to 6 days ahead of total Covid-19 patient of 10 Brazilian states, using stacking-ensemble with support vector regression algorithm on the cumulative positive Covid-19 cases of Brazilian data was proposed, thus augmenting the short-term forecasting process to alert the healthcare expert and the government to tackle the pandemic. Recent studies suggested a novel model using a supervised multi-layered recursive classifier called XGBoost on clinical and mammographic factor datasets. After applying the model, researchers found out those three significant key features (high-sensitivity C-reactive protein, lymphocyte and lactic dehydrogenase (LDH)) of the 75 features clinical and blood test samples result to be the highest rank of 90% accuracy in predicting and assessing Covid-19 patient into general, severe and mortality rate. Furthermore, comparatively higher value in single lactic dehydrogenase appears to be a significant factor in classifying most patients in need of intensive medical care, as LDH degree related to various respiratory disorder diseases, namely asthma and bronchitis, and pneumonia. The forecast model employed decision rule to forecast rapidly and predict infected individuals at the highest risk, authorized patients to be manageable for intensive care, and possibly lessen the transience rate. A Canadian based forecasting model using time-series was developed employing Deep learning algorithm for the long-short-term-memory network, the studies found out a key factor intended for predicting the course with an ending point estimation of the current SARS-CoV-2 epidemic in Canada and all over the globe. The suggested model forecast ending point of this SARS-CoV-2 outbreak in Canada will be around June 2020. Based on the data collected from John Hopkins University, the prediction was likely to be accurate as newly infected cases have dropped rapidly and proven the applicability of the expert system in predicting and forecasting for the current pandemic outbreak by revealing key significant features.



2.5 Conclusion and discussion

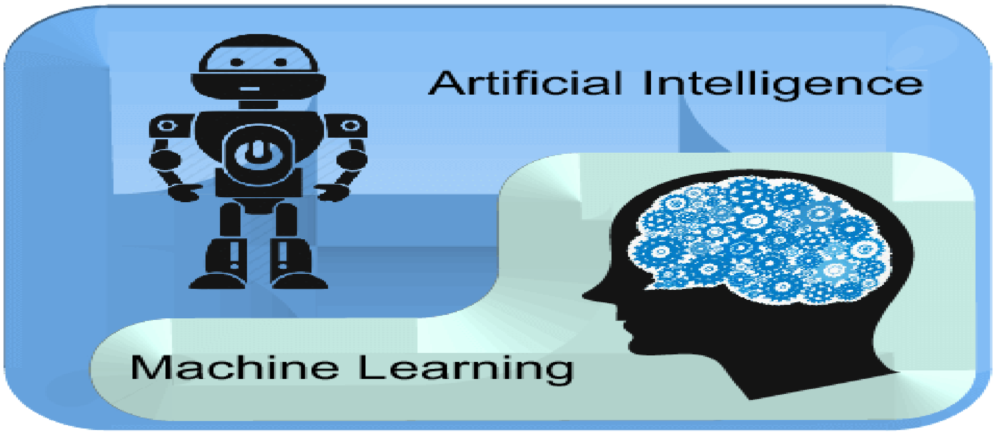
Since the outbreak of the novel SARS-CoV-2, scientists and medical industries around the globe ubiquitously urged to fight against the pandemic, searching alternative method of rapid screening and prediction process, contact tracing, forecasting, and development of vaccine or drugs with the more accurate and reliable operation. Machine Learning and Artificial Intelligence are such promising methods employed by various healthcare providers. This paper addresses on recent studies that apply such advance technology in augmenting the researchers in multiple angles, addressing the troubles and challenges while using such algorithm in assisting medical expert in real-world problems. This paper also discusses suggestions conveying researchers on AI/ML-based model design, medical experts, and policymakers on few errors encountered in the current situation while tackling the current pandemic. This review shows that the use of modern technology with AI and ML dramatically improves the screening, prediction, contact tracing, forecasting, and drug/vaccine development with extreme reliability. Majority of the paper employed deep learning algorithms and is found to have more potential, robust, and advance among the other learning algorithms. However, the current urgency requires an improved model with high-end performance accuracy in screening and predicting the SARS-CoV-2 with a different kind of related disease by analyzing the clinical, mammographic, and demographic information of the suspects and infected patients. Finally, it is evident that AI and ML can significantly improve treatment, medication, screening & prediction, forecasting, contact tracing, and drug/vaccine development for the Covid-19 pandemic and reduce the human intervention in medical practice. However, most of the models are not deployed enough to show their real-world operation, but they are still up to the mark to tackle the pandemic.

CHAPTER – 3

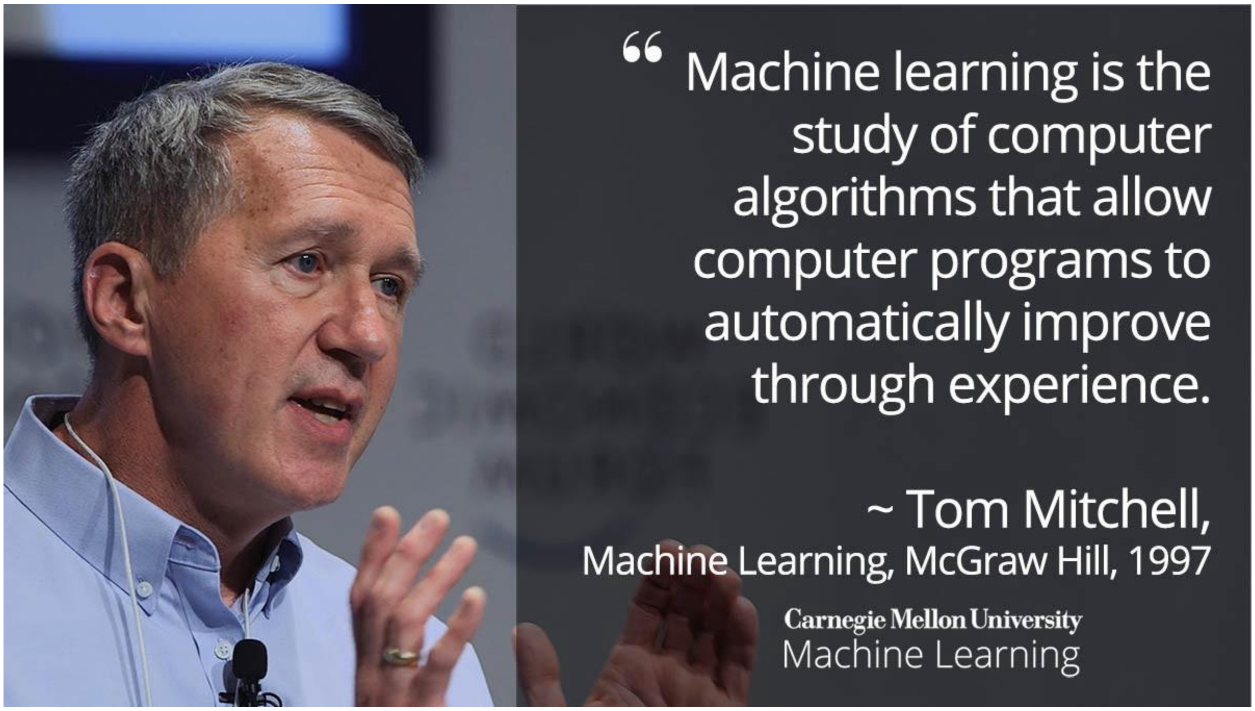
Intro with technology learned & used

3.1 Definition of Artificial Intelligence

Artificial Intelligence (**AI**) is the branch of computer sciences that emphasizes the development of **intelligence** machines, thinking and working like humans. For **example**, speech recognition, problem-solving, learning and planning.



3.2 Definition of Machine Learning

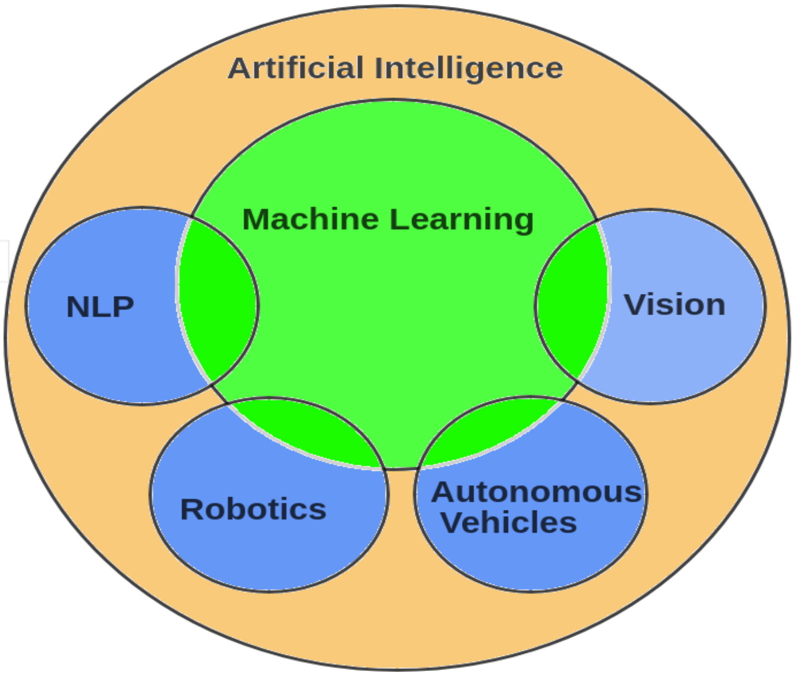


“Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience.

~ Tom Mitchell,
Machine Learning, McGraw Hill, 1997

3.3 How ML & AI are interconnected

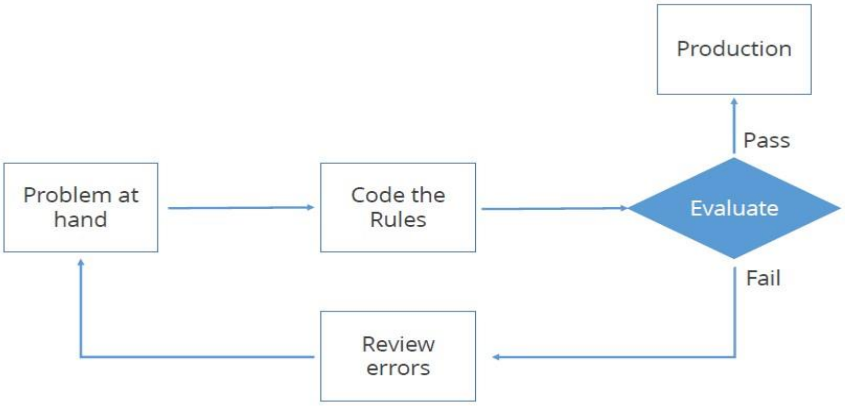
While **machine learning** is based on the idea that **machines** should be able to learn and adapt through experience, **AI** refers to a broader idea where **machines** can execute tasks "smartly." **Artificial Intelligence** applies **machine learning**, **deep learning** and other techniques to solve actual problems.



3.4 Traditional Approach vs. ML Approach

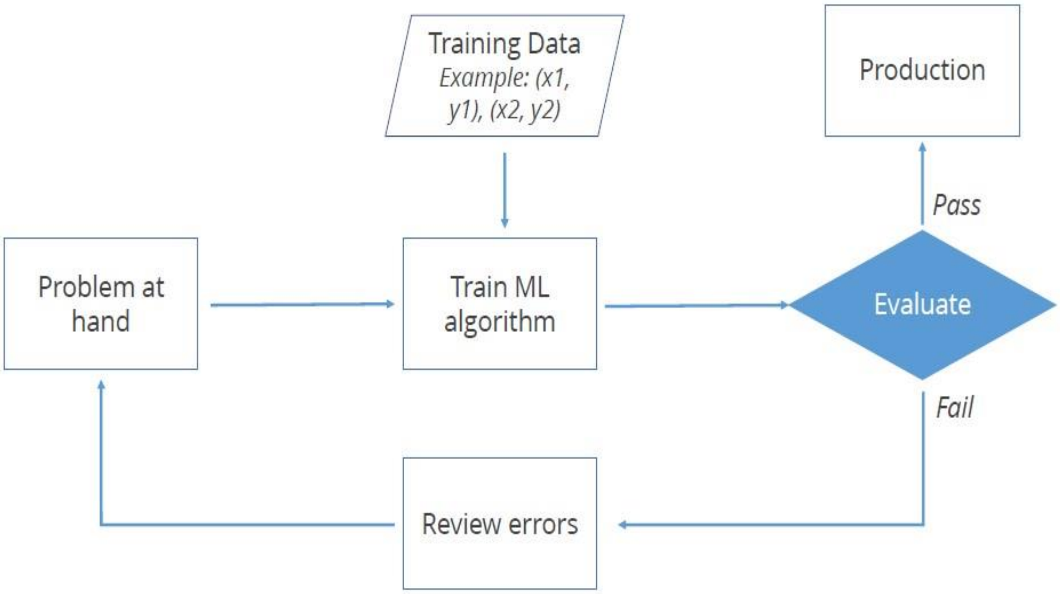
1. **Traditional Approach**

   Tradition Programming relies on **hard-coded rules.**



2. **Machine Learning Approach**

   Machine Learning relies on learning patterns based on sample data.



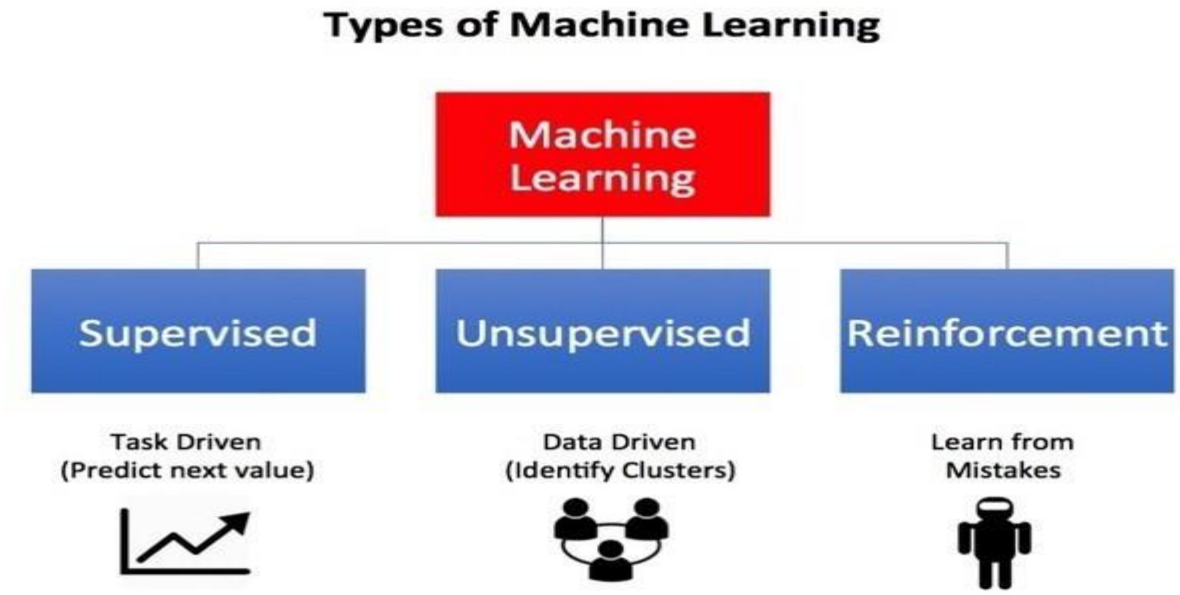
3.5 Why Machine Learning Matters

With the rise in big data, machine learning has become a key technique for solving problems in areas, such as:

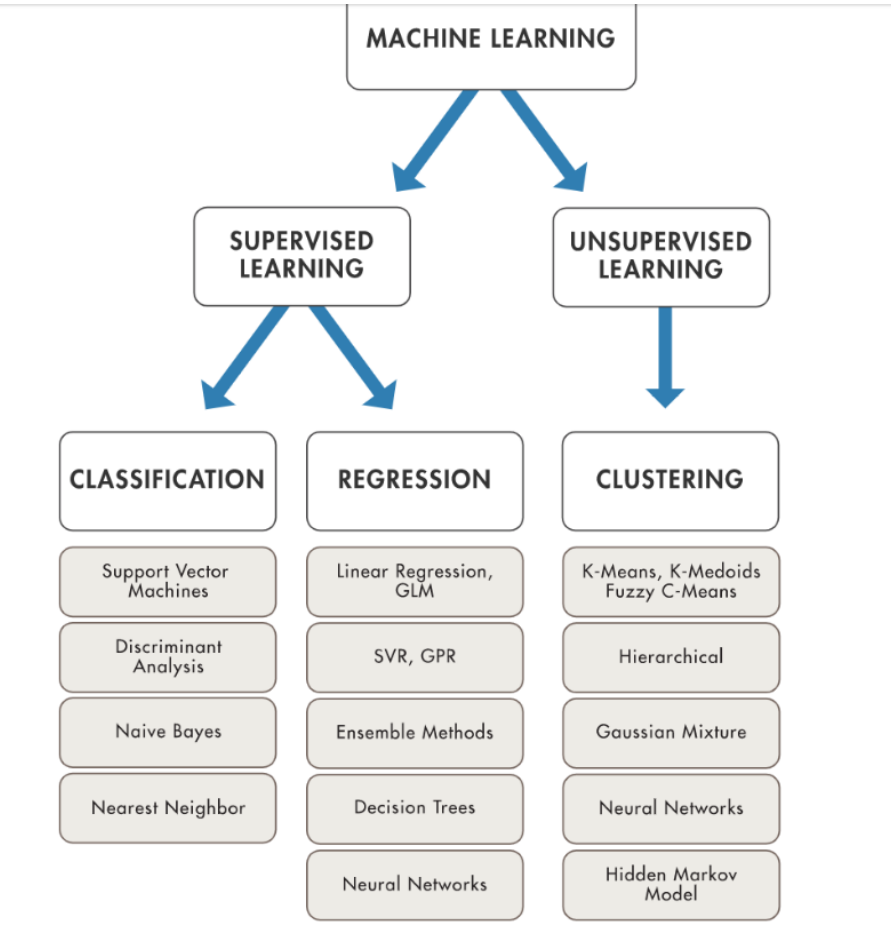
- **Computational finance**, for credit scoring and algorithmic trading
- **Image processing and computer vision**, for face recognition, motion detection, and object detection
- **Computational biology**, for tumor detection, drug discovery, and DNA sequencing
- **Energy production**, for price and load forecasting
- **Automotive, aerospace, and manufacturing**, for predictive maintenance
- **Natural language processing**, for voice recognition applications

3.6 How Machine Learning Works

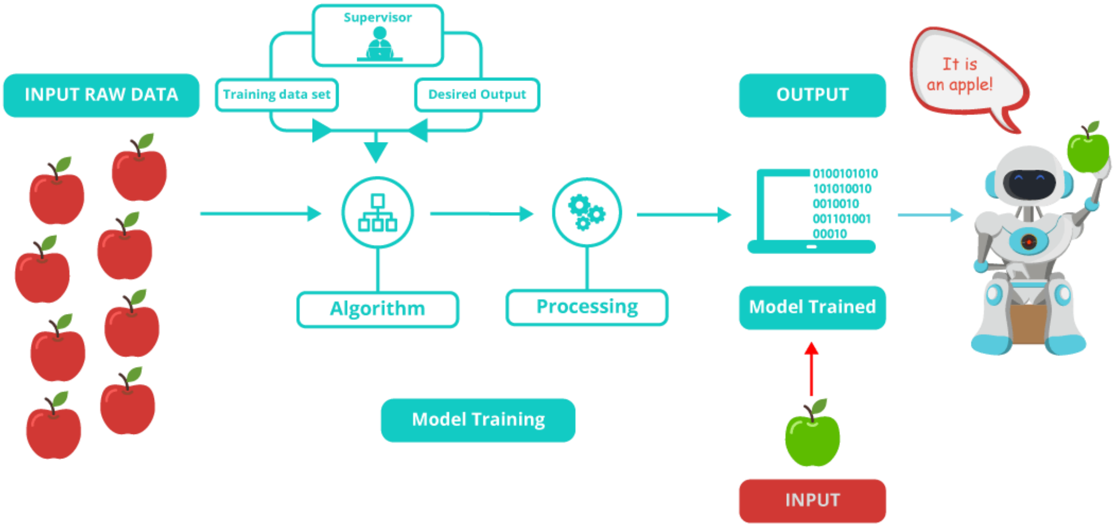
Machine learning uses two types of techniques: **supervised learning**, which trains a model on known input and output data so that it can predict future outputs, and **unsupervised learning**, which finds hidden patterns or intrinsic structures in input data.



- Choose **supervised learning** if you need to train a model to make a prediction--for example, the future value of a continuous variable, such as temperature or a stock price, or a classification—for example, identify makes of cars from webcam video footage.
- Choose **unsupervised learning** if you need to explore your data and want to train a model to find a good internal representation, such as splitting data up into clusters.
- **Reinforcement learning** (RL) is an area of **machine learning** concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward. **Reinforcement learning** is one of three basic **machine learning** paradigms, alongside supervised **learning** and unsupervised **learning**.



Supervised Learning



Supervised machine learning builds a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Use supervised learning if you have known data for the output you are trying to predict.

Supervised learning uses classification and regression techniques to develop predictive models.

Classification techniques predict discrete responses—for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring.

Use classification if your data can be tagged, categorized, or separated into specific groups or classes. For example, applications for hand-writing recognition use classification to recognize letters and numbers. In image processing and computer vision, unsupervised pattern recognition techniques are used for object detection and image segmentation.

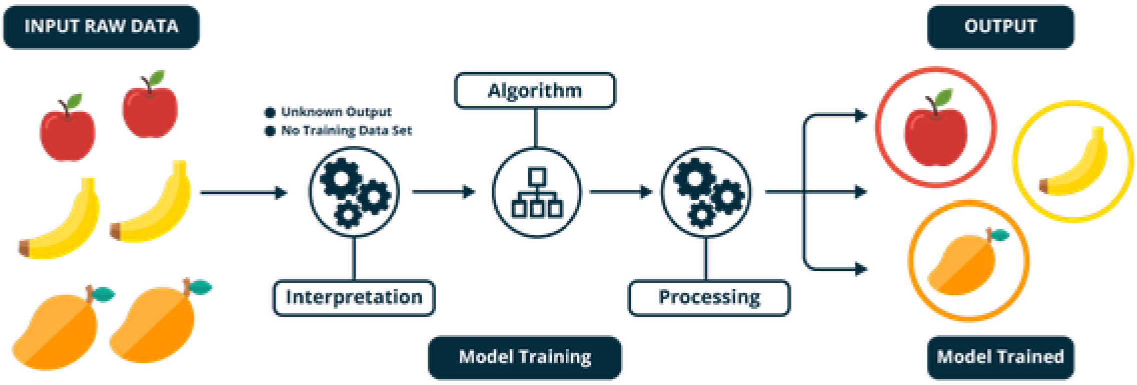
Common algorithms for performing classification include support vector machine (SVM), boosted and bagged decision trees, *k*-nearest neighbor, Naïve Bayes, discriminant analysis, logistic regression, and neural networks.

Regression techniques predict continuous responses—for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading.

Use regression techniques if you are working with a data range or if the nature of your response is a real number, such as temperature or the time until failure for a piece of equipment.

Common regression algorithms include linear model, nonlinear model, regularization, stepwise regression, boosted and bagged decision trees, neural networks, and adaptive neuro-fuzzy learning.

Unsupervised Learning



Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used t o draw inferences from datasets consisting of input data without labeled responses.

Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for cluster analysis include gene sequence analysis, market research, and object recognition.

For example, if a cell phone company wants optimize the locations where they build cell phone towers, they can use machine learning to estimate the number of clusters of people relying on their towers. A phone can only talk to one tower at a time, so the team uses clustering algorithms to design the best placement of cell towers to optimize signal reception for groups, or clusters, of their customers.

Common algorithms for performing clustering include k-means and k-medoids, hierarchical clustering, Gaussian mixture models, hidden Markov models, self-organizing maps, fuzzy c-means clustering, and subtractive clustering.

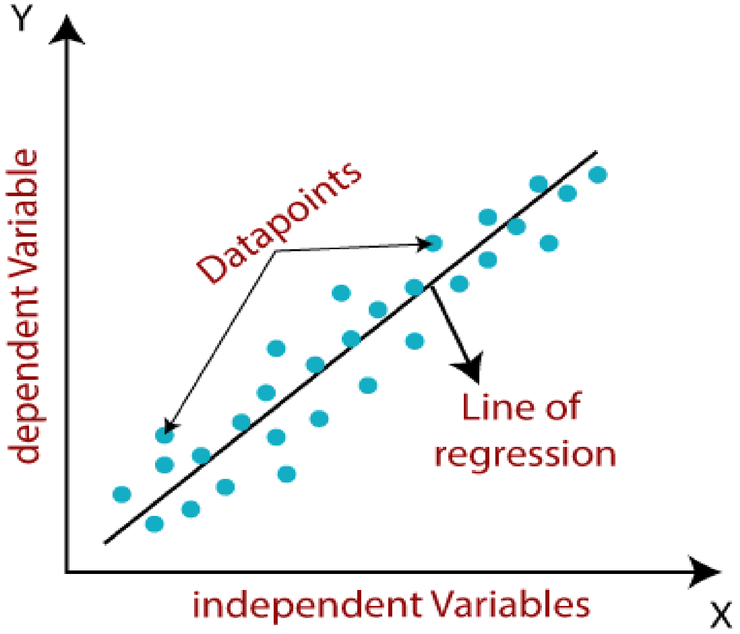
CHAPTER – 4

Various common ML Algorithms

What are the most common and popular machine learning algorithms?

1. *Linear Regression (Supervised Learning/Regression)*

Linear regression is the most basic type of regression. Simple linear regression allows us to understand the relationships between two continuous variables.



Mathematically, we can represent a linear regression as:

$y= a_0+a_1x+ \varepsilon$

Here,

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)

a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

Cost function-

- The different values for weights or coefficient of lines (a_0, a_1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
- Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
- We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**.

For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

For the above linear equation, MSE can be calculated as:

$$\text{MSE}= 1\frac{1}{N}\sum_{i=1}^{n}(y_i \; - (a_1x_i + a_0))^2$$

Where,

N=Total number of observation
Yi = Actual value
($a1x_i$+a_0)= Predicted value.

Residuals: The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.

Gradient Descent:

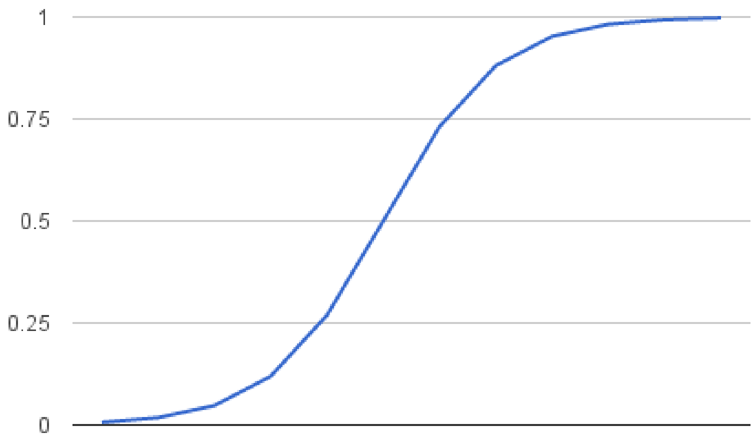
- Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
- A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
- It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

Model Performance:

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called **optimization**.

2. *Logistic Regression (Supervised learning – Classification)*

Logistic regression focuses on estimating the probability of an event occurring based on the previous data provided. It is used to cover a binary dependent variable, that is where only two values, 0 and 1, represent outcomes.



- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or

1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

3. *Naïve Bayes Classifier Algorithm (Supervised Learning - Classification)*

The Naïve Bayes classifier is based on Bayes' theorem and classifies every value as independent of any other value. It allows us to predict a class/category, based on a given set of features, using probability.

Despite its simplicity, the classifier does surprisingly well and is often used due to the fact it outperforms more sophisticated classification methods.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

4. *K-Means Clustering Algorithm (Unsupervised Learning - Clustering)*

The K Means Clustering algorithm is a type of unsupervised learning, which is used to categorise unlabelled data, i.e. data without defined categories or groups. The algorithm works by finding groups within the data, with the number of groups represented by the variable K. It then works iteratively to assign each data point to one of K groups based on the features provided.

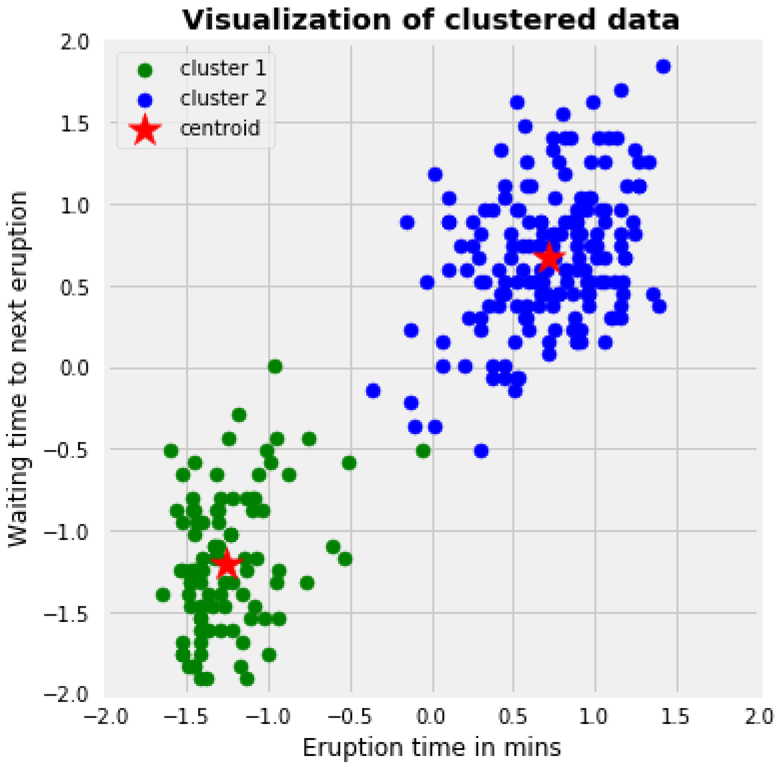
K-means algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way kmeans algorithm works is as follows:

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).

Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.



The above graph shows the scatter plot of the data colored by the cluster they belong to. In this example, we chose K=2. The symbol '*' is the centroid of each cluster. We can think of those 2 clusters as geyser had different kinds of behaviors under different scenarios.

Next, we'll show that different initializations of centroids may yield to different results. I'll use 9 different random_state to change the initialization of the centroids and plot the results. The title of each plot will be the sum of squared distance of each initialization.

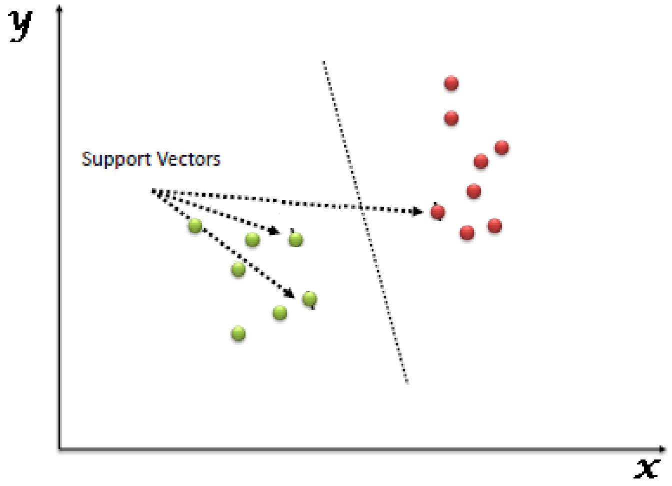
As a side note, this dataset is considered very easy and converges in less than 10 iterations. Therefore, to see the effect of random initialization on convergence, I am going to go with 3 iterations to illustrate the concept. However, in real world applications, datasets are not at all that clean and nice!

The approach k-means follows to solve the problem is called **Expectation-Maximization**. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.

5. ***Support Vector Machine Algorithm (Supervised Learning - Classification)***

Support Vector Machine algorithms are supervised learning models that analyse data used for classification and regression analysis. They essentially filter data into categories, which is achieved by providing a set of training examples, each set marked as belonging to one or the other of the two categories. The algorithm then works to build a model that assigns new values to one category or the other.

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However , it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



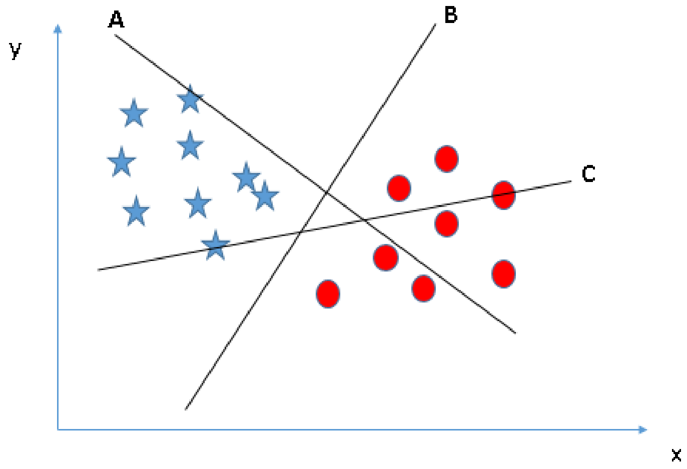
Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

How does it work?

Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is “How can we identify the right hyper-plane?”. Don’t worry, it’s not as hard as you think!

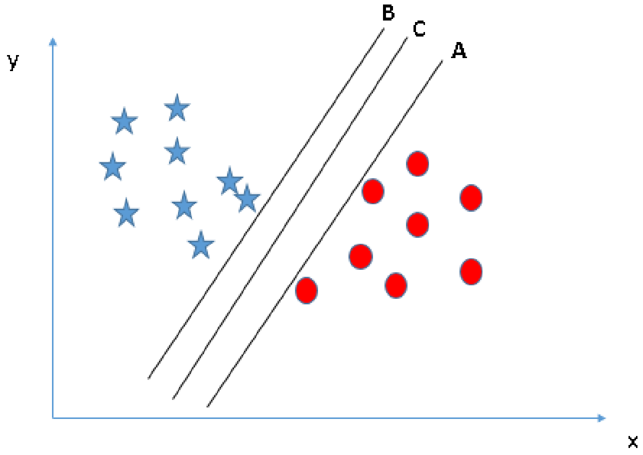
Let’s understand:

- Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

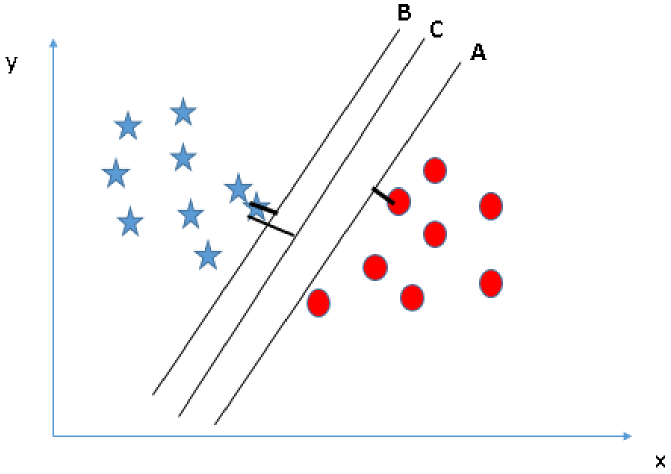


- You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

- Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

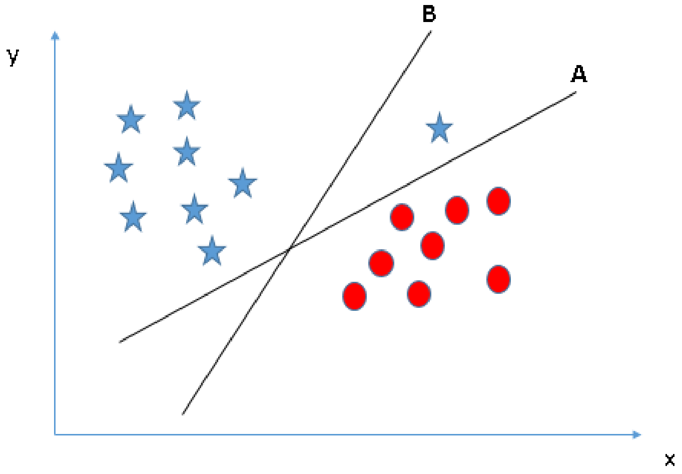


Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin. Let's look at the below snapshot:



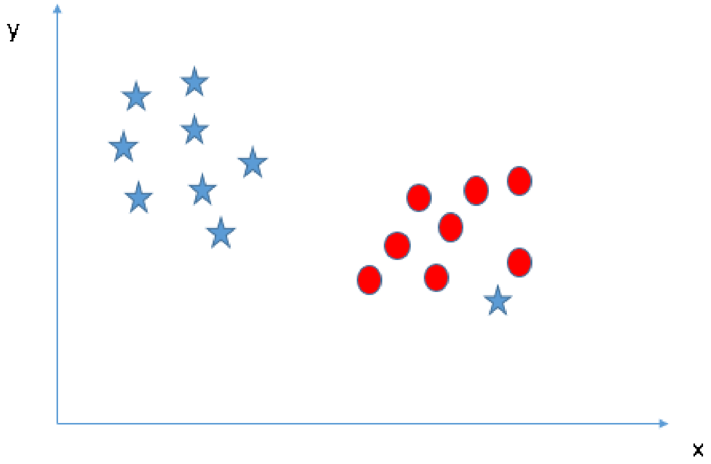
Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- Identify the right hyper-plane (Scenario-3):Hint: Use the rules as discussed in previous section to identify the right hyper-plane

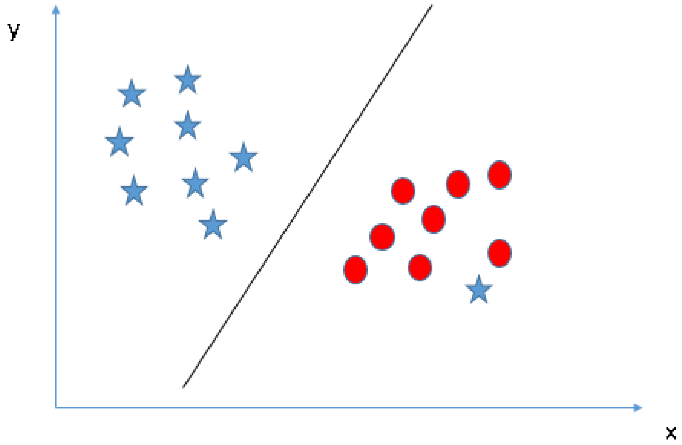


Some of you may have selected the hyper-plane B as it has higher margin compared to A. But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is A.

- Can we classify two classes (Scenario-4)?: Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



- As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



- In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

6. ***Artificial Neural Networks (Reinforcement Learning)***

An artificial neural network (ANN) comprises 'units' arranged in a series of layers, each of which connects to layers on either side. ANNs are Inspired by biological systems, such as the brain, and how they process information. ANNs are essentially a large number of interconnected processing elements, working in unison to solve specific problems.

ANNs also learn by example and through experience, and they are extremely useful for modelling non-linear relationships in high-dimensional data or where the relationship amongst the input variables is difficult to understand.

A neural network is a machine learning algorithm based on the model of a human neuron. The human brain consists of millions of neurons. It sends and process signals in the form of electrical and chemical signals. These neurons are connected with a special structure known as synapses. Synapses allow neurons to pass signals. From large numbers of simulated neurons neural networks forms.

An Artificial Neural Network is an information processing technique. It works like the way human brain processes information. ANN includes a large number of connected processing units that work together to process information. They also generate meaningful results from it.

We can apply Neural network not only for classification. It can also apply for regression of continuous target attributes.

Neural networks find great application in data mining used in sectors. For example economics, forensics, etc. and for pattern recognition. It can be also used for data classification in a large amount of data after careful training.

A neural network may contain the following 3 layers:

- Input layer – The activity of the input units represents the raw information that can feed into the network.
- Hidden layer – To determine the activity of each hidden unit. The activities of the input units and the weights on the connections between the input and the hidden units. There may be one or more hidden layers.
- Output layer – The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

Artificial Neural Network Layers

Artificial Neural network is typically organized in layers. Layers are being made up of many interconnected 'nodes' which contain an 'activation function'. A neural network may contain the following 3 layers:

a. Input layer

The purpose of the input layer is to receive as input the values of the explanatory attributes for each observation. Usually, the number of input nodes in an input layer is equal to the number of explanatory variables. 'input layer' presents the patterns to the network, which communicates to one or more 'hidden layers'.

The nodes of the input layer are passive, meaning they do not change the data. They receive a single value on their input and duplicate the value to their many outputs. From the input layer, it duplicates each value and sent to all the hidden nodes.

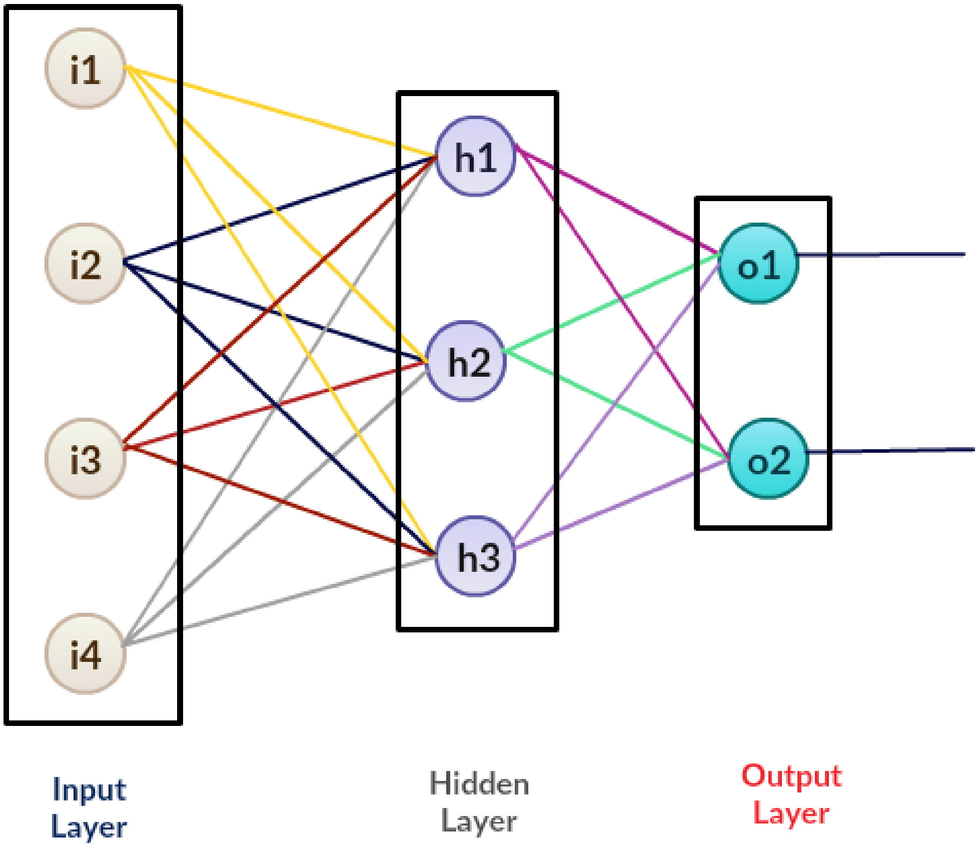
b. Hidden layer

The Hidden layers apply given transformations to the input values inside the network. In this, incoming arcs that go from other hidden nodes or from input nodes connected to each node. It connects with outgoing arcs to output nodes or to other hidden nodes. In hidden layer, the actual processing is done via a system of weighted 'connections'. There may be one or more hidden layers. The values entering a hidden node multiplied by weights, a set of predetermined numbers stored in the program. The weighted inputs are then added to produce a single number.

c. Output layer

The hidden layers then link to an 'output layer'. Output layer receives connections from hidden layers or from input layer. It returns an output value that corresponds to the prediction of the response variable. In classification problems, there is usually only one output node. The active nodes of the output layer combine and change the data to produce the output values.

The ability of the neural network to provide useful data manipulation lies in the proper selection of the weights. This is different from conventional information processing.



Applications of Artificial Neural Networks:

a. Classification of data:

Based on a set of data, our trained neural network predicts whether it is a dog or a cat?

b. Anomaly detection:

Given the details about transactions of a person, it can say that whether the transaction is fraud or not.

c. Speech recognition:

We can train our neural network to recognize speech patterns. Example: Siri, Alexa, Google assistant.

d. Audio generation:

Given the inputs as audio files, it can generate new music based on various factors like genre, singer, and others.

e. Time series analysis:

A well trained neural network can predict the stock price.

f. Spell checking:

We can train a neural network that detects misspelled spellings and can also suggest a similar meaning for words. Example: Grammarly

g. Character recognition:

A well trained neural network can detect handwritten characters.

h. Machine translation:

We can develop a neural network that translates one language into another language.

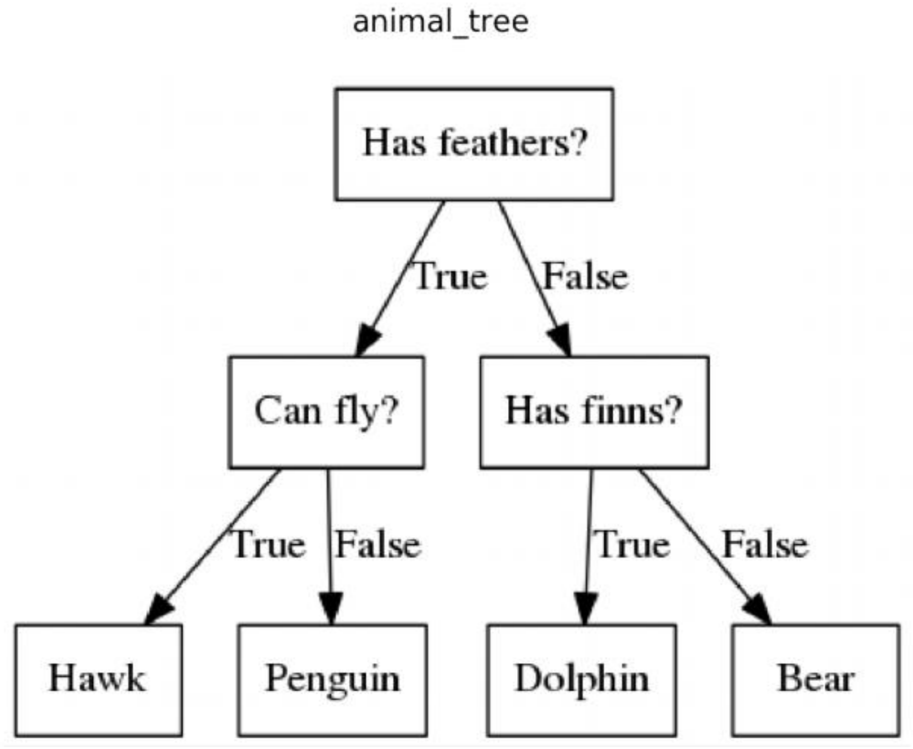
i. Image processing:

We can train a neural network to process an image and extract pieces of information from it.

7. *Decision Trees (Supervised Learning – Classification/Regression)*

A decision tree is a flow-chart-like tree structure that uses a branching method to illustrate every possible outcome of a decision. Each node within the tree represents a test on a specific variable – and each branch is the outcome of that test.

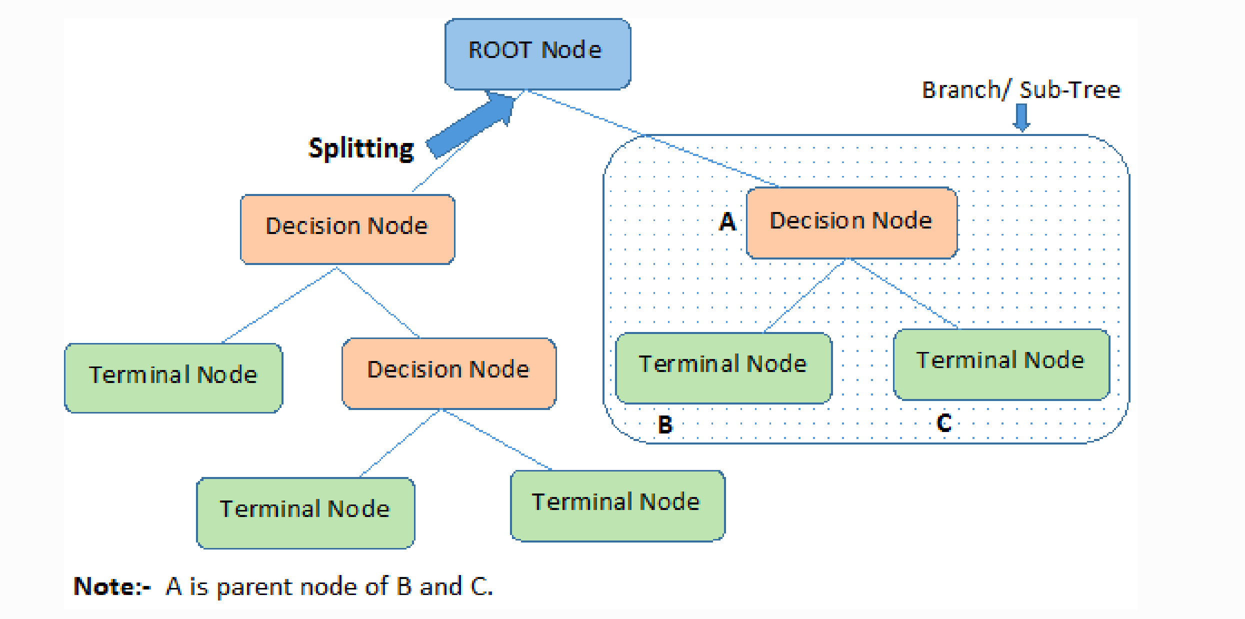
A decision tree is one of the supervised machine learning algorithms. This algorithm can be used for regression and classification problems—yet, is mostly used for classification problems. A decision tree follows a set of if-else conditions to visualize the data and classify it according to the conditions.
For example,



Before we dive deep into the working principle of the decision tree's algorithm you need to know a few keywords related to it.

Important terminology

1. **Root Node:** This attribute is used for dividing the data into two or more sets. The feature attribute in this node is selected based on Attribute Selection Techniques.
2. **Branch or Sub-Tree:** A part of the entire decision tree is called branch or sub-tree.
3. **Splitting:** Dividing a node into two or more sub-nodes based on if-else conditions.
4. **Decision Node:** After splitting the sub-nodes into further sub-nodes, then it is called as the decision node.
5. **Leaf or Terminal Node:** This is the end of the decision tree where it cannot be split into further sub-nodes.
6. **Pruning:** Removing a sub-node from the tree is called pruning.



Note:- A is parent node of B and C.

8. _**Random Forests (Supervised Learning – Classification/Regression)**_

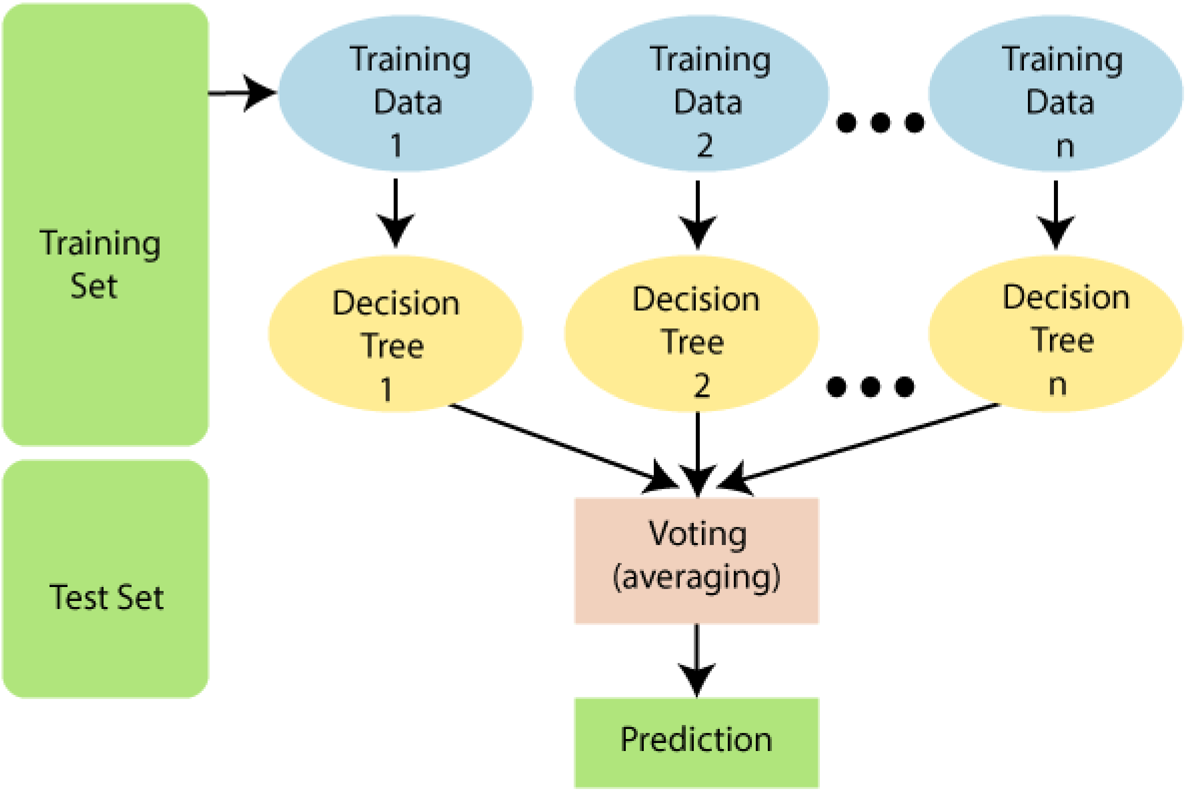
Random forests or 'random decision forests' is an ensemble learning method, combining multiple algorithms to generate better results for classification, regression and other tasks. Each individual classifier is weak, but when combined with others, can produce excellent results. The algorithm starts with a 'decision tree' (a tree-like graph or model of decisions) and an input is entered at the top. It then travels down the tree, with data being segmented into smaller and smaller sets, based on specific variables.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

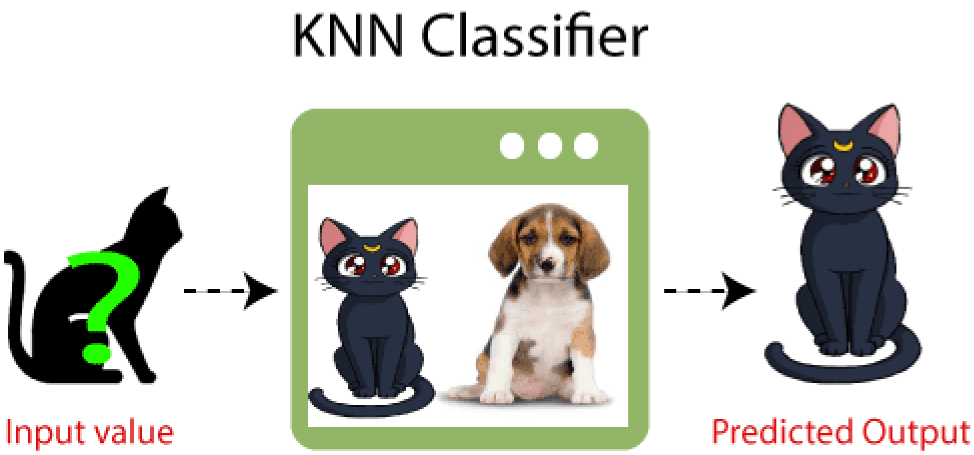
The below diagram explains the working of the Random Forest algorithm:



9. *K-Nearest Neighbours (Supervised Learning)*

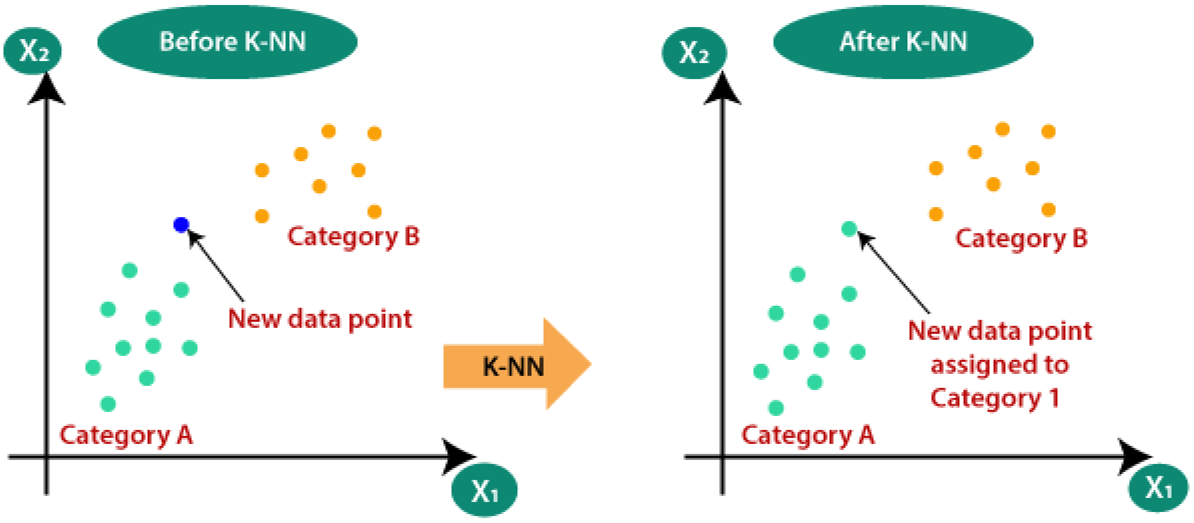
The K-Nearest-Neighbour algorithm estimates how likely a data point is to be a member of one group or another. It essentially looks at the data points around a single data point to determine what group it is actually in. For example, if one point is on a grid and the algorithm is trying to determine what group that data point is in (Group A or Group B, for example) it would look at the data points near it to see what group the majority of the points are in.

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



CHAPTER – 5

Our Model

5.1 Introduction:

Our predictive model machine learning project can be broken down into 6 top-level tasks:

1. Define Problem: Investigate and characterize the problem in order to better understand the goals of the project.

2. Analyze Data: Use descriptive statistics and visualization to better understand the data you have available.

3. Prepare Data: Use data transforms in order to better expose the structure of the prediction problem to modeling algorithms.

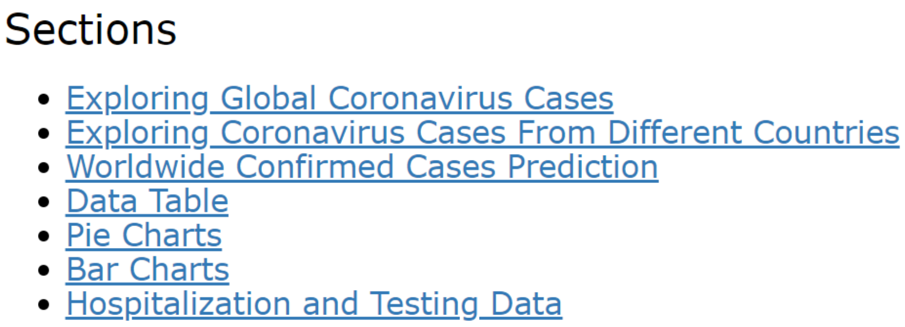
4. Evaluate Algorithms: Design a test harness to evaluate a number of standard algorithms on the data and select the top few to investigate further.

5. Improve Results: Use algorithm tuning and ensemble methods to get the most out of well-performing algorithms on your data.

6. Present Results: Finalize the model, make predictions and present results.

5.2 Requirements:

As per the requirements   , we divided our project in the following sections:



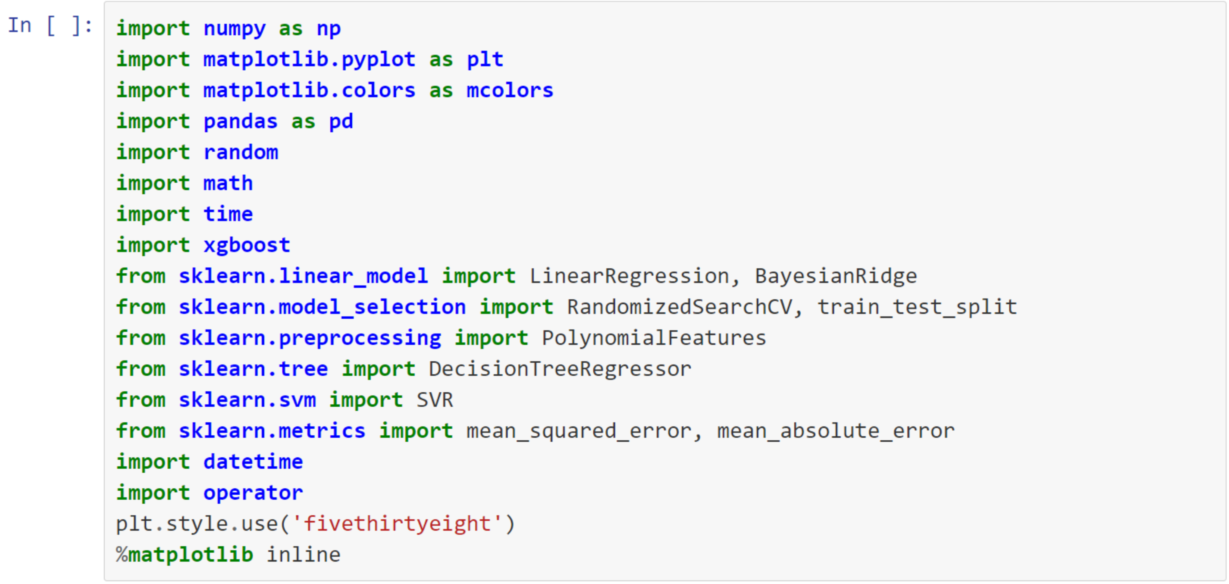
Sections

- Exploring Global Coronavirus Cases
- Exploring Coronavirus Cases From Different Countries
- Worldwide Confirmed Cases Prediction
- Data Table
- Pie Charts
- Bar Charts
- Hospitalization and Testing Data

5.3 Process with Code :

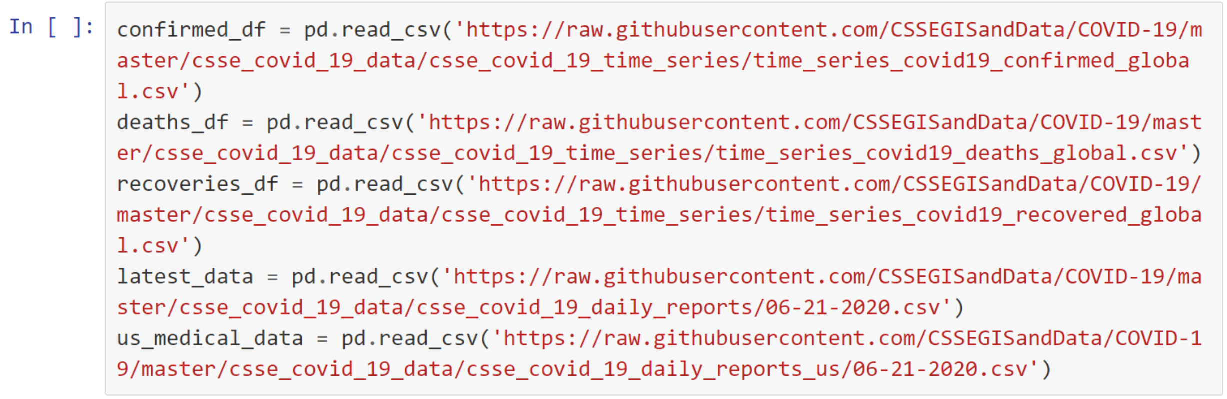
5.3.1 Loading the dataset

Lets start by loading the libraries required for this project.



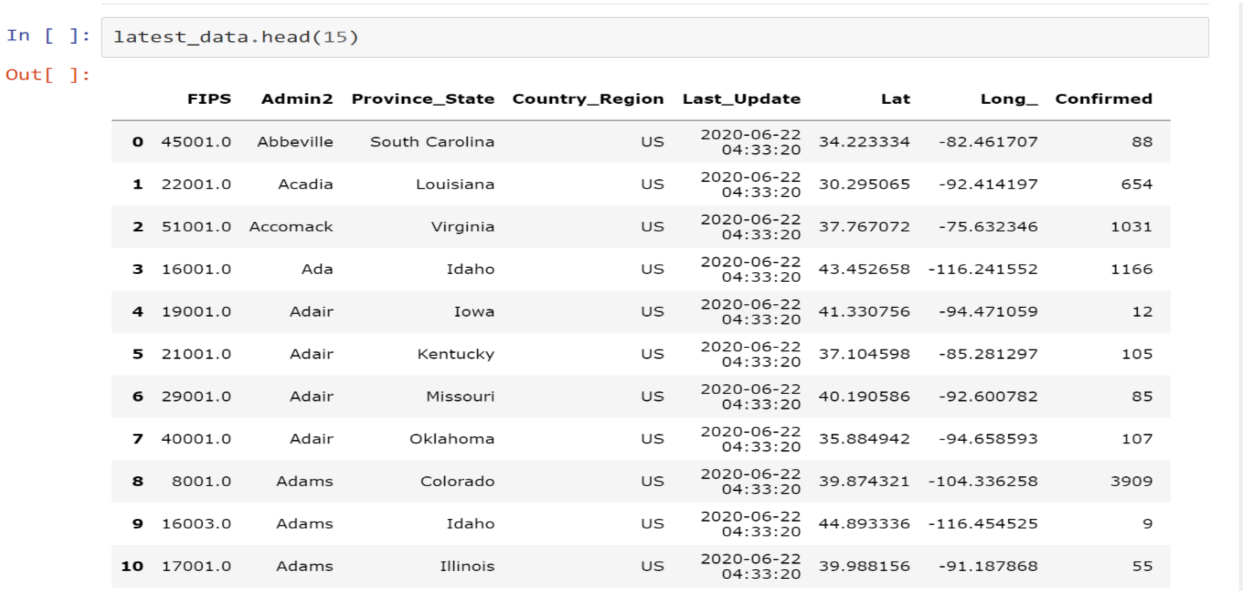
```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import pandas as pd
import random
import math
import time
import xgboost
from sklearn.linear_model import LinearRegression, BayesianRidge
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
plt.style.use('fivethirtyeight')
%matplotlib inline
```

And then taking datasets



```
In [ ]: confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
deaths_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv')
recoveries_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_global.csv')
latest_data = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/06-21-2020.csv')
us_medical_data = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports_us/06-21-2020.csv')
```

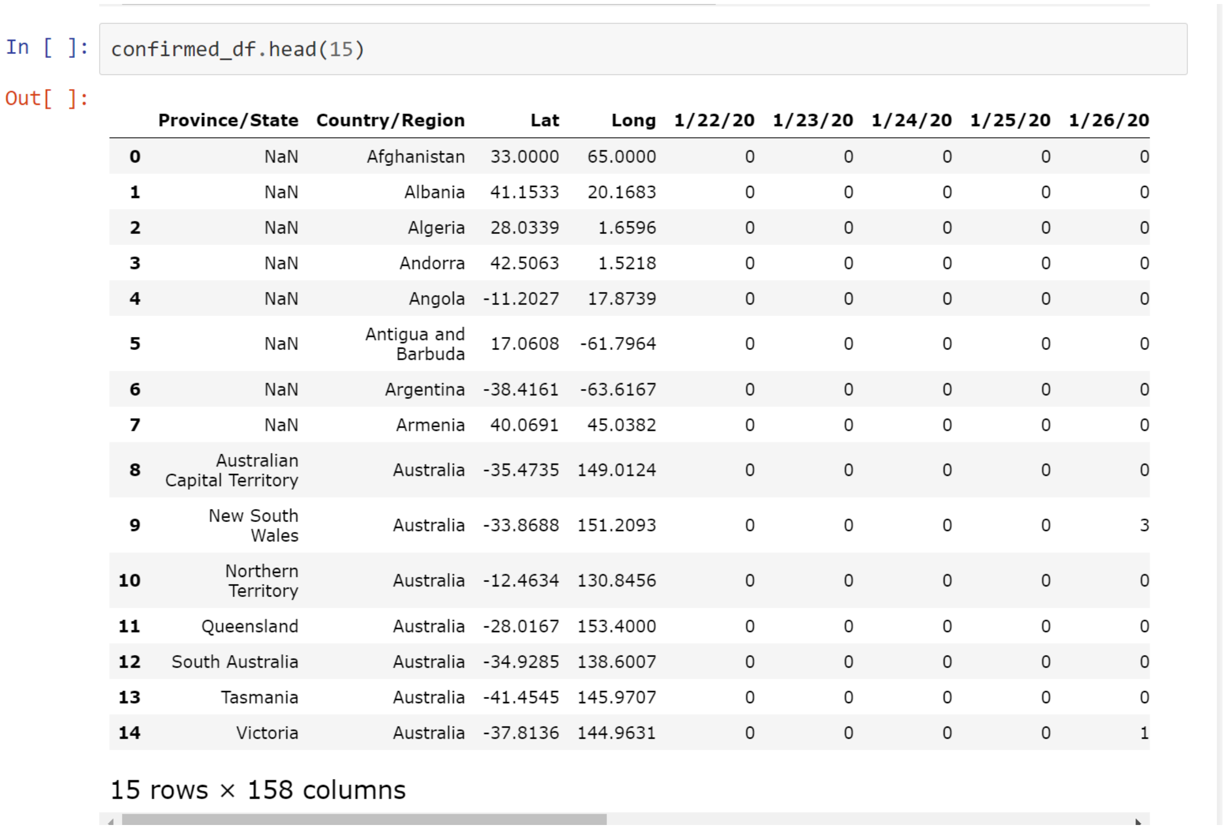
Now let us have a look at our loaded data:



In [ ]:
```
latest_data.head(15)
```

Out[ ]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed
0	45001.0	Abbeville	South Carolina	US	2020-06-22 04:33:20	34.223334	-82.461707	88
1	22001.0	Acadia	Louisiana	US	2020-06-22 04:33:20	30.295065	-92.414197	654
2	51001.0	Accomack	Virginia	US	2020-06-22 04:33:20	37.767072	-75.632346	1031
3	16001.0	Ada	Idaho	US	2020-06-22 04:33:20	43.452658	-116.241552	1166
4	19001.0	Adair	Iowa	US	2020-06-22 04:33:20	41.330756	-94.471059	12
5	21001.0	Adair	Kentucky	US	2020-06-22 04:33:20	37.104598	-85.281297	105
6	29001.0	Adair	Missouri	US	2020-06-22 04:33:20	40.190586	-92.600782	85
7	40001.0	Adair	Oklahoma	US	2020-06-22 04:33:20	35.884942	-94.658593	107
8	8001.0	Adams	Colorado	US	2020-06-22 04:33:20	39.874321	-104.336258	3909
9	16003.0	Adams	Idaho	US	2020-06-22 04:33:20	44.893336	-116.454525	9
10	17001.0	Adams	Illinois	US	2020-06-22 04:33:20	39.988156	-91.187868	55



In [ ]:
```
confirmed_df.head(15)
```

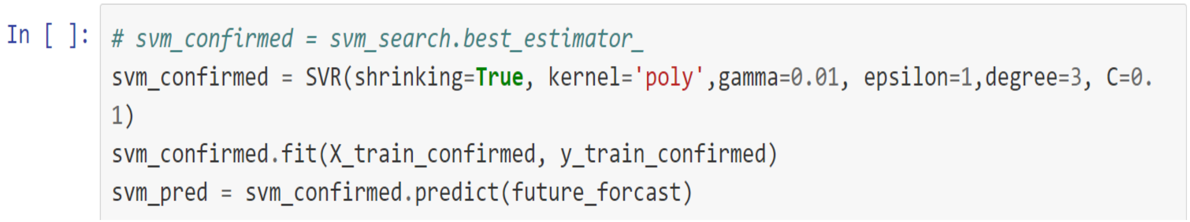
Out[ ]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0
5	NaN	Antigua and Barbuda	17.0608	-61.7964	0	0	0	0	0
6	NaN	Argentina	-38.4161	-63.6167	0	0	0	0	0
7	NaN	Armenia	40.0691	45.0382	0	0	0	0	0
8	Australian Capital Territory	Australia	-35.4735	149.0124	0	0	0	0	0
9	New South Wales	Australia	-33.8688	151.2093	0	0	0	0	3
10	Northern Territory	Australia	-12.4634	130.8456	0	0	0	0	0
11	Queensland	Australia	-28.0167	153.4000	0	0	0	0	0
12	South Australia	Australia	-34.9285	138.6007	0	0	0	0	0
13	Tasmania	Australia	-41.4545	145.9707	0	0	0	0	0
14	Victoria	Australia	-37.8136	144.9631	0	0	0	0	1

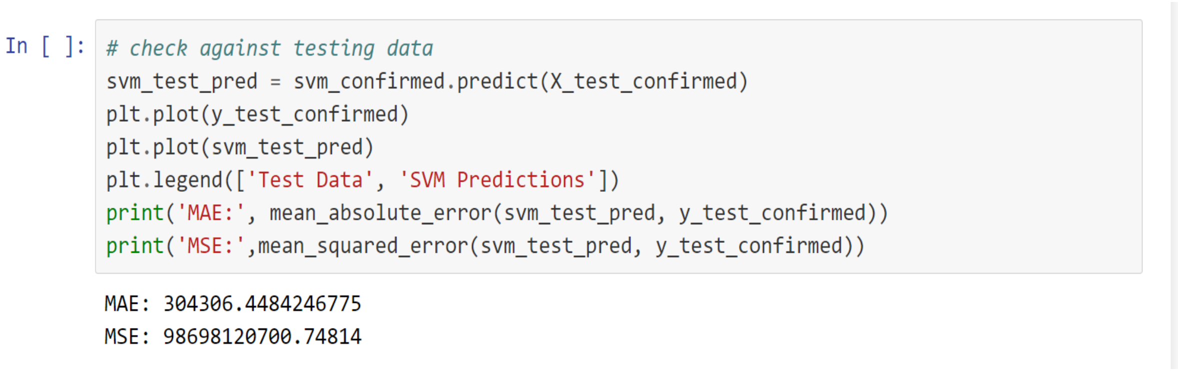
15 rows × 158 columns

We've explored the data, now we'll try to use machine learning to predict our target. After doing this , we need to decide our algorithms which we are going to perform on our loaded data .In our case we looked for various algorithms and finally decided to use Support Vector Machine(SVM),Linear Regression and Bayesian Ridge Regression.

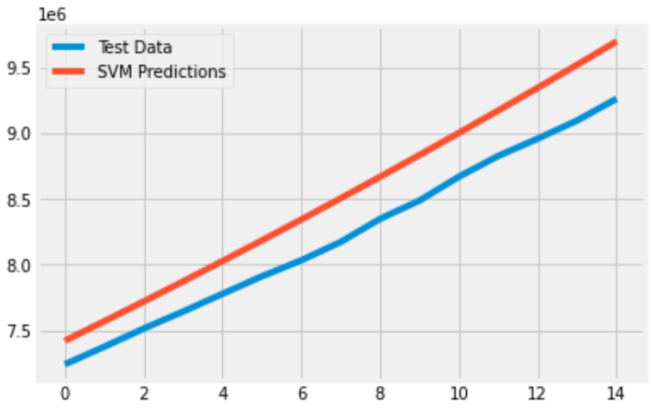
1.Prediction using SVM



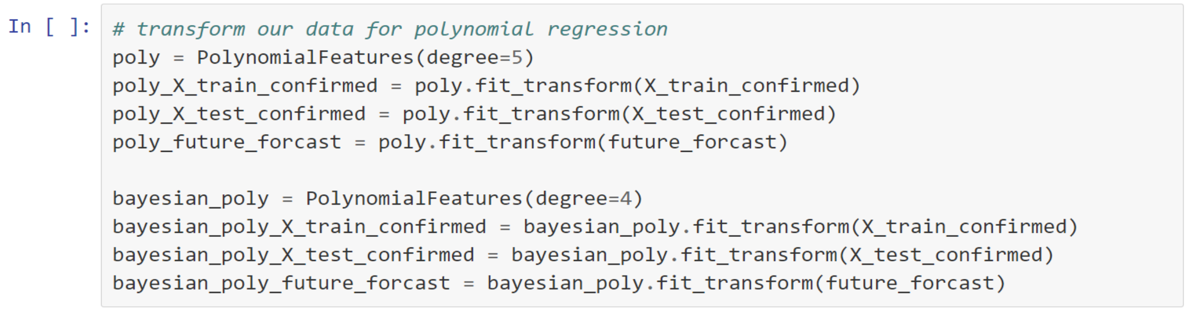
```
In [ ]: # svm_confirmed = svm_search.best_estimator_
        svm_confirmed = SVR(shrinking=True, kernel='poly',gamma=0.01, epsilon=1,degree=3, C=0.
        1)
        svm_confirmed.fit(X_train_confirmed, y_train_confirmed)
        svm_pred = svm_confirmed.predict(future_forcast)
```



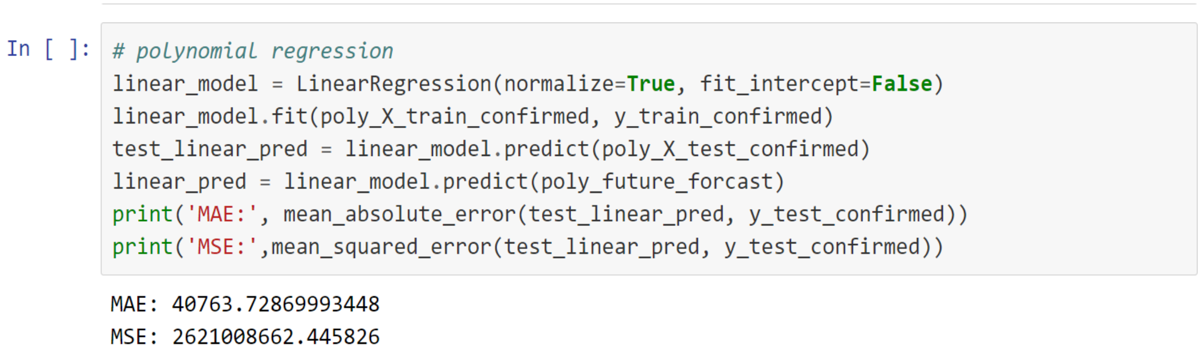
```
In [ ]: # check against testing data
        svm_test_pred = svm_confirmed.predict(X_test_confirmed)
        plt.plot(y_test_confirmed)
        plt.plot(svm_test_pred)
        plt.legend(['Test Data', 'SVM Predictions'])
        print('MAE:', mean_absolute_error(svm_test_pred, y_test_confirmed))
        print('MSE:',mean_squared_error(svm_test_pred, y_test_confirmed))

        MAE: 304306.4484246775
        MSE: 98698120700.74814
```



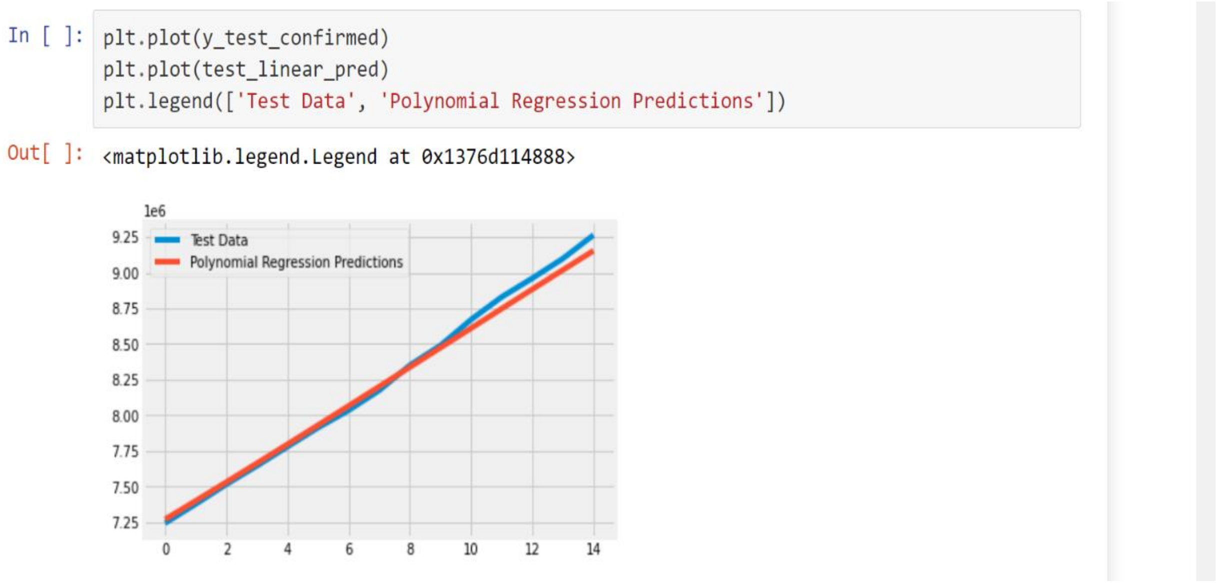
2.Prediction using Polynomial Regression



```
In [ ]: # transform our data for polynomial regression
        poly = PolynomialFeatures(degree=5)
        poly_X_train_confirmed = poly.fit_transform(X_train_confirmed)
        poly_X_test_confirmed = poly.fit_transform(X_test_confirmed)
        poly_future_forcast = poly.fit_transform(future_forcast)

        bayesian_poly = PolynomialFeatures(degree=4)
        bayesian_poly_X_train_confirmed = bayesian_poly.fit_transform(X_train_confirmed)
        bayesian_poly_X_test_confirmed = bayesian_poly.fit_transform(X_test_confirmed)
        bayesian_poly_future_forcast = bayesian_poly.fit_transform(future_forcast)
```

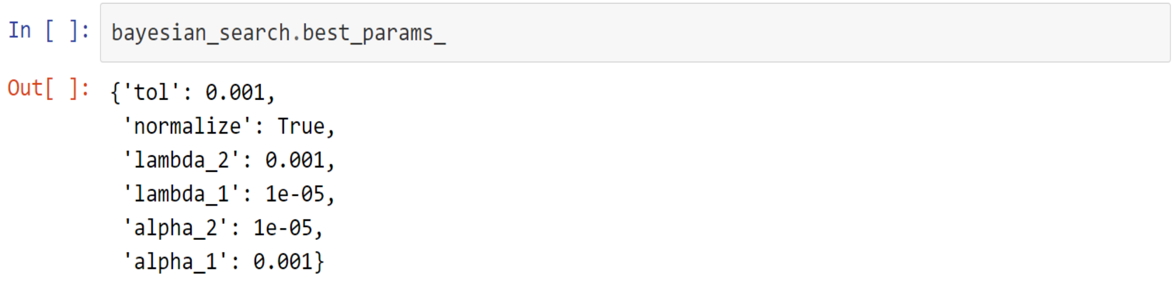


```
In [ ]: # polynomial regression
        linear_model = LinearRegression(normalize=True, fit_intercept=False)
        linear_model.fit(poly_X_train_confirmed, y_train_confirmed)
        test_linear_pred = linear_model.predict(poly_X_test_confirmed)
        linear_pred = linear_model.predict(poly_future_forcast)
        print('MAE:', mean_absolute_error(test_linear_pred, y_test_confirmed))
        print('MSE:',mean_squared_error(test_linear_pred, y_test_confirmed))

        MAE: 40763.72869993448
        MSE: 2621008662.445826
```

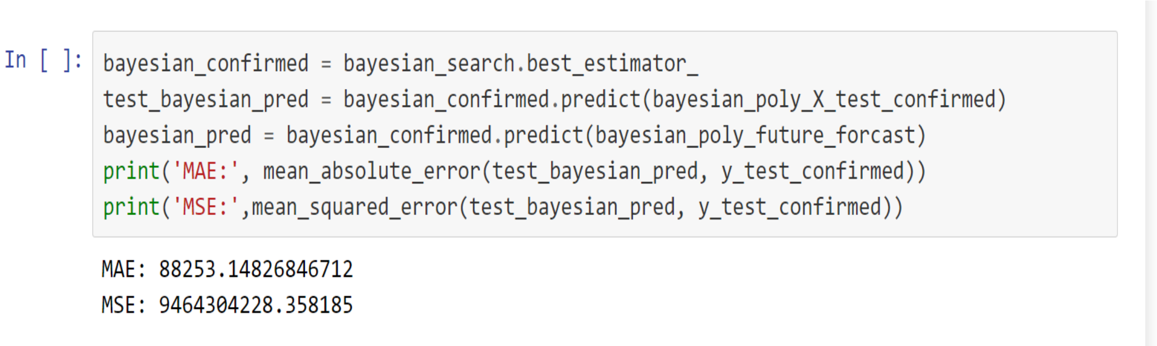


```
In [ ]: plt.plot(y_test_confirmed)
        plt.plot(test_linear_pred)
        plt.legend(['Test Data', 'Polynomial Regression Predictions'])

Out[ ]: <matplotlib.legend.Legend at 0x1376d114888>
```

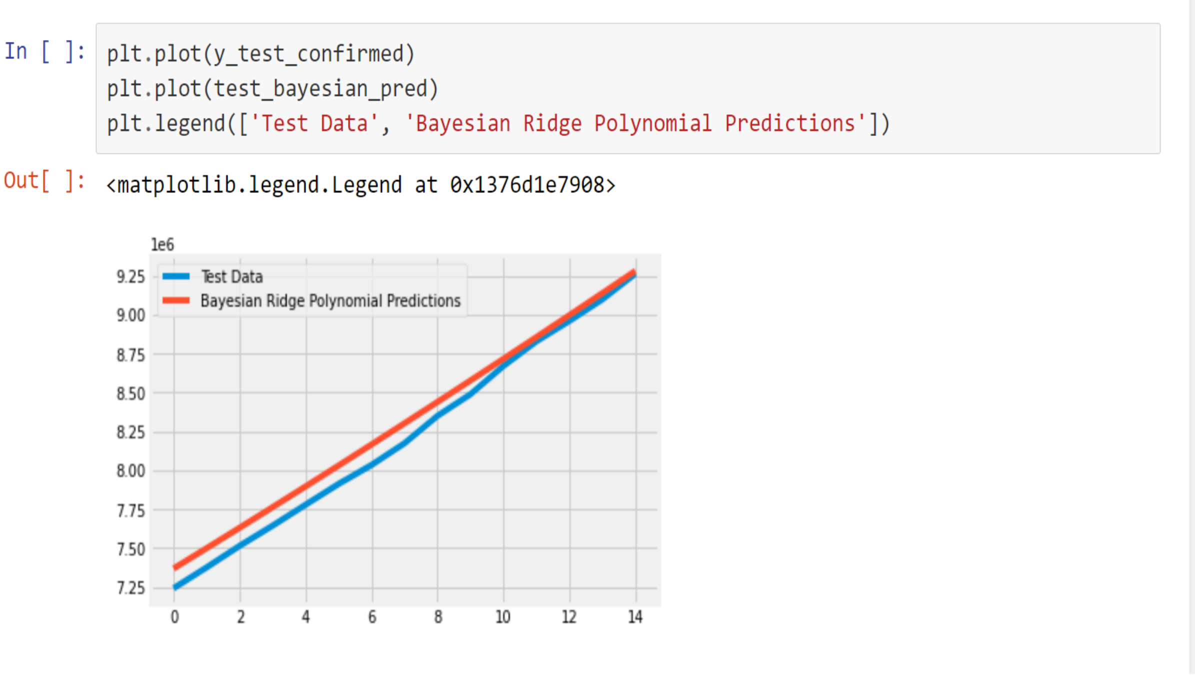
3.Prediction using Baysian Ridge Regression



```
In [ ]: bayesian_search.best_params_

Out[ ]: {'tol': 0.001,
          'normalize': True,
          'lambda_2': 0.001,
          'lambda_1': 1e-05,
          'alpha_2': 1e-05,
          'alpha_1': 0.001}
```



```
In [ ]: bayesian_confirmed = bayesian_search.best_estimator_
        test_bayesian_pred = bayesian_confirmed.predict(bayesian_poly_X_test_confirmed)
        bayesian_pred = bayesian_confirmed.predict(bayesian_poly_future_forcast)
        print('MAE:', mean_absolute_error(test_bayesian_pred, y_test_confirmed))
        print('MSE:',mean_squared_error(test_bayesian_pred, y_test_confirmed))

        MAE: 88253.14826846712
        MSE: 9464304228.358185
```



```
In [ ]: plt.plot(y_test_confirmed)
        plt.plot(test_bayesian_pred)
        plt.legend(['Test Data', 'Bayesian Ridge Polynomial Predictions'])

Out[ ]: <matplotlib.legend.Legend at 0x1376d1e7908>
```

- **Mean Absolute Error** (**MAE**): This measures the absolute average distance between the real data and the predicted data, but it fails to punish large errors in prediction.

- **Mean Square Error** (**MSE**): This measures the squared average distance between the real data and the predicted data.

5.4 VISUALISATION

Graphing the Confirmed cases, deaths and recovery cases :

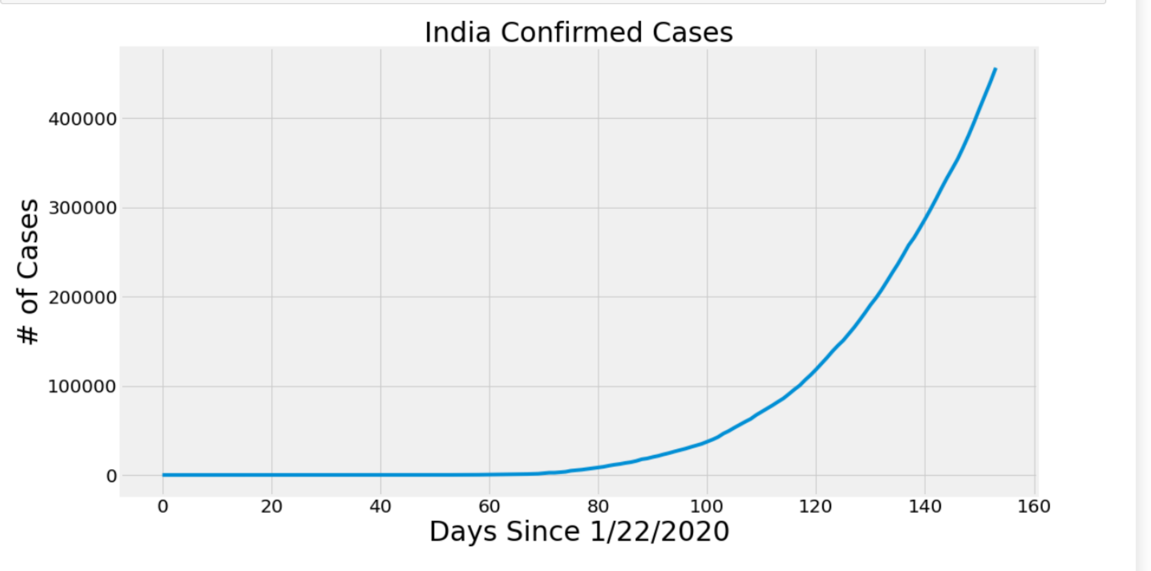
Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

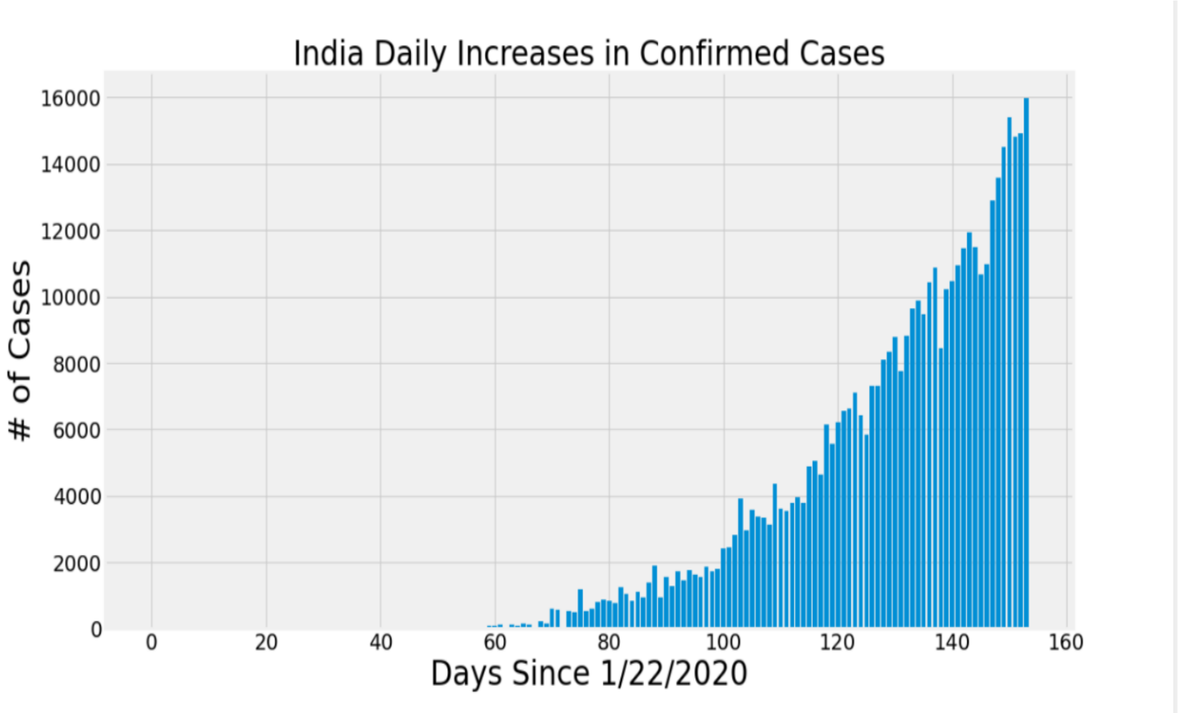
In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Data is necessary for our understanding of the world, and particularly for the emergence of phenomena such as the COVID-19 outbreak. A viral pandemic is not a scenario in which intuition can provide a sense of how the spread is advancing, nor are feelings a sufficient approach to dealing with and ultimately defeating such an unseen enemy. Collecting, analyzing, sharing, and ultimately making use of data is what is needed.

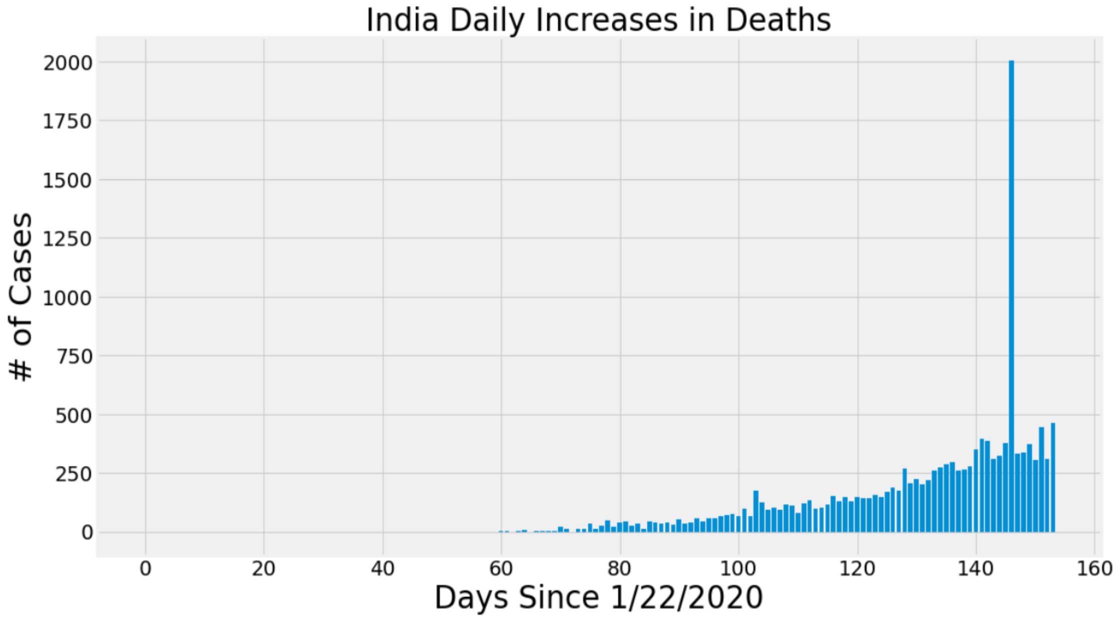
This collected data needs to be efficiently conveyed to all sorts of individuals and groups, from lay people to experts and everything in between. Visualizing collected data can make its dissemination easy, and can help others understand quickly what has taken others so long to collect and analyze. After all, a picture is worth a thousand words.

# Having a look upon confirmed cases in India

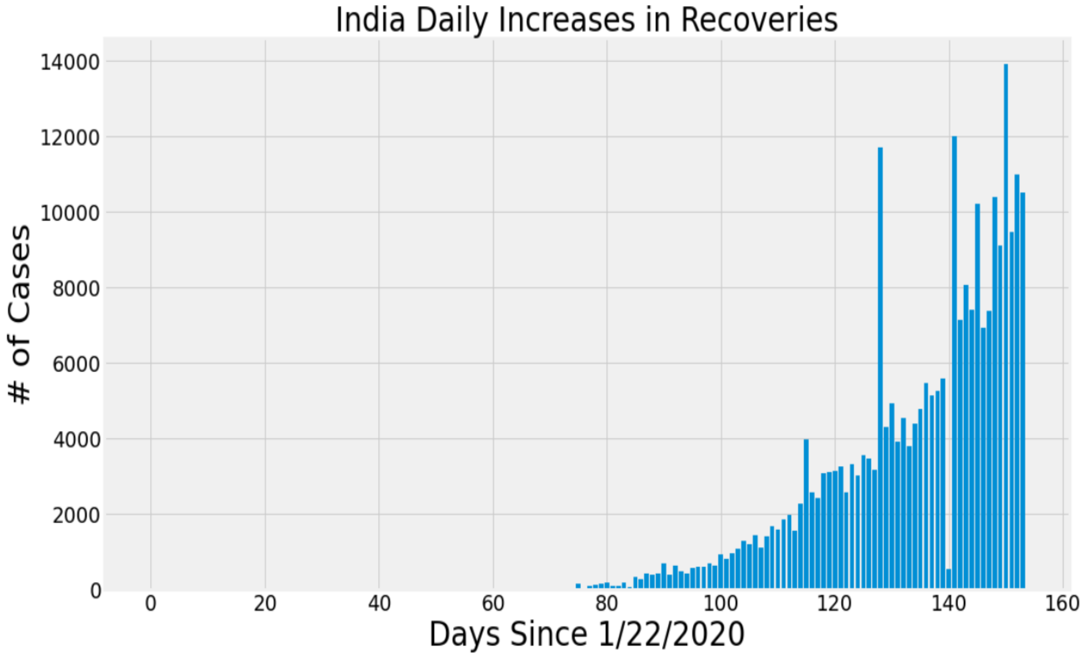




# Having a look upon Increase in Death cases in India



# Having a look upon Increase in Recovery cases in India

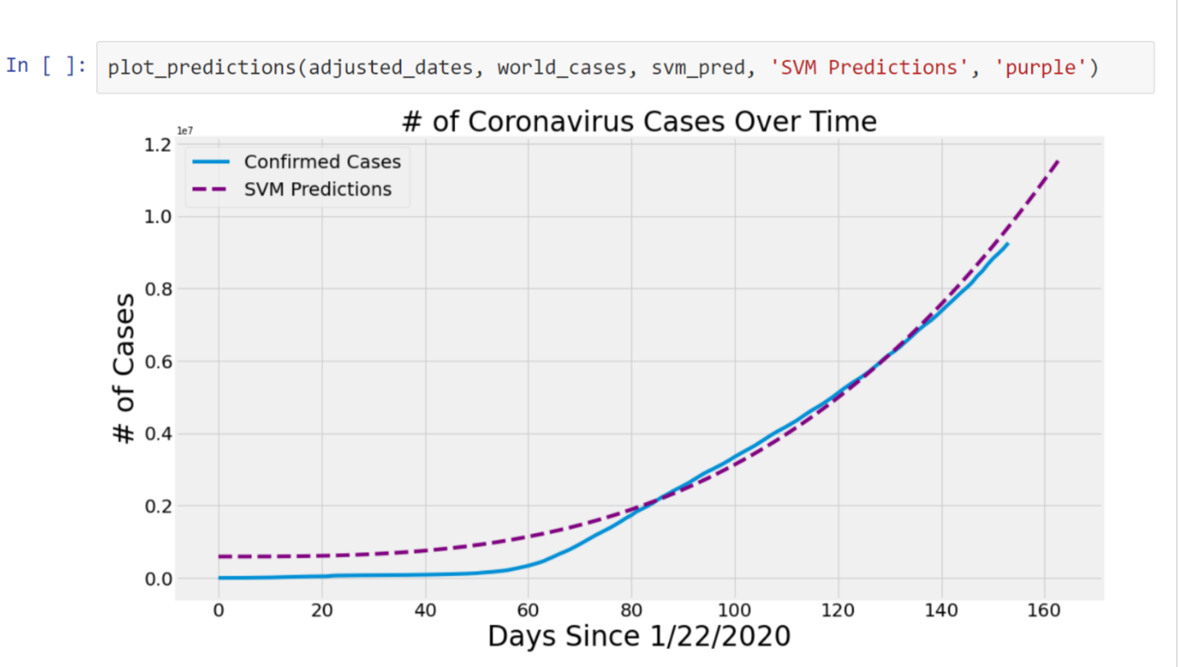


5.5 PREDICTION

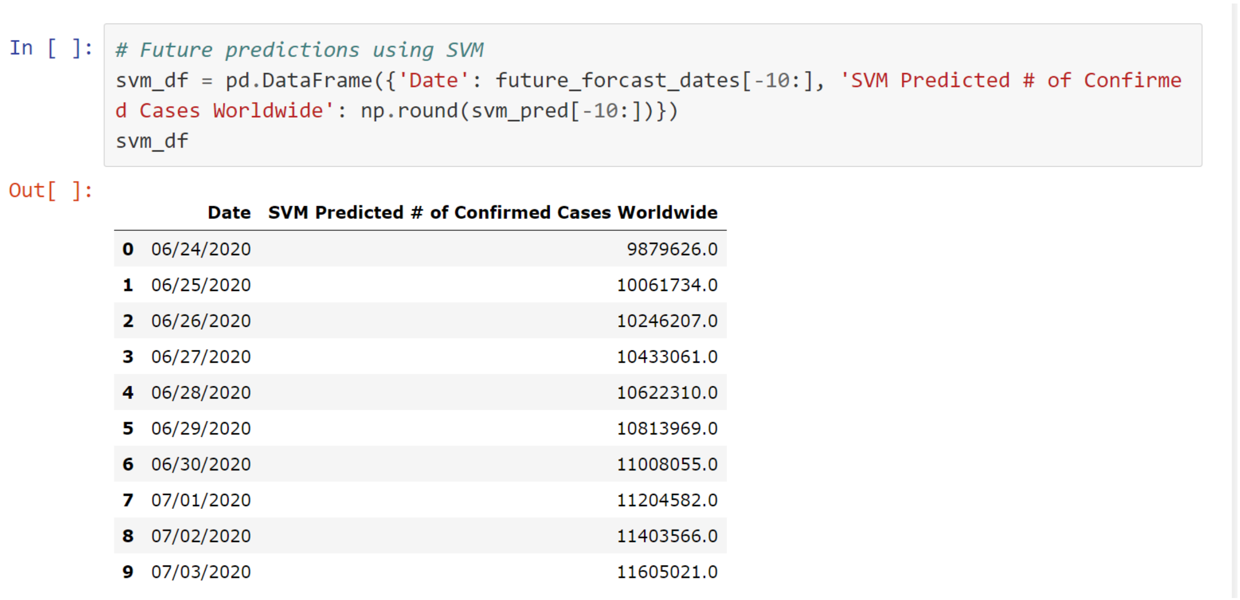
Prediction for confirmed corona virus cases world-wide :

# Prediction using SVM Algorithm

The support vector machine (SVM) is a predictive analysis data-classification algorithm that assigns new data elements to one of labeled categories. SVM is, in most cases, a binary classifier; it assumes that the data in question contains two possible target values. But here we will use it for forcasting.



```
In [ ]: plot_predictions(adjusted_dates, world_cases, svm_pred, 'SVM Predictions', 'purple')
```



In [ ]:

```
# Future predictions using SVM
svm_df = pd.DataFrame({'Date': future_forcast_dates[-10:], 'SVM Predicted # of Confirme
d Cases Worldwide': np.round(svm_pred[-10:])})
svm_df
```

Out[ ]:

	Date	SVM Predicted # of Confirmed Cases Worldwide
0	06/24/2020	9879626.0
1	06/25/2020	10061734.0
2	06/26/2020	10246207.0
3	06/27/2020	10433061.0
4	06/28/2020	10622310.0
5	06/29/2020	10813969.0
6	06/30/2020	11008055.0
7	07/01/2020	11204582.0
8	07/02/2020	11403566.0
9	07/03/2020	11605021.0

# Prediction using Polynomial Regression Algorithm

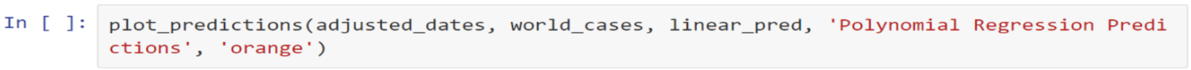
The basic goal of regression analysis is to model the expected value of a dependent variable y in terms of the value of an independent variable x. In simple regression, we used following equation –

```
y = a + bx + e
```

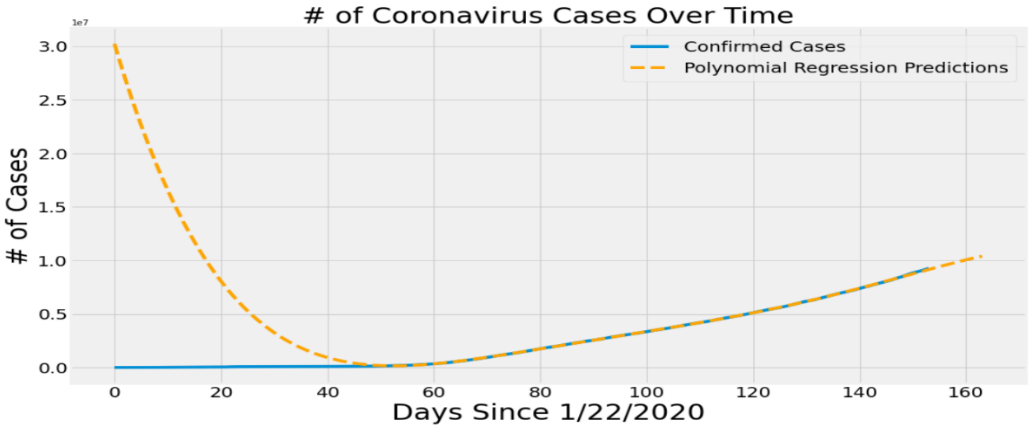
Here y is dependent variable, a is y intercept, b is the slope and e is the error rate.

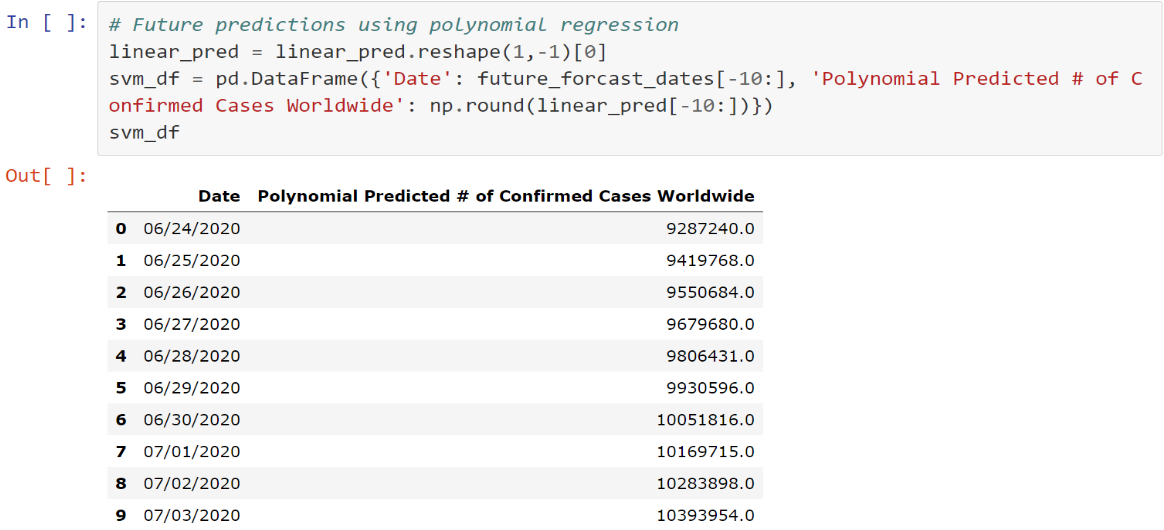
In general for Polynomial regression, we can model it for nth value.

```
y = a + b1x + b2x^2 +....+ bnx^n
```



```
In [ ]: plot_predictions(adjusted_dates, world_cases, linear_pred, 'Polynomial Regression Predi
        ctions', 'orange')
```





```
In [ ]: # Future predictions using polynomial regression
        linear_pred = linear_pred.reshape(1,-1)[0]
        svm_df = pd.DataFrame({'Date': future_forcast_dates[-10:], 'Polynomial Predicted # of C
        onfirmed Cases Worldwide': np.round(linear_pred[-10:])})
        svm_df

Out[ ]:
```

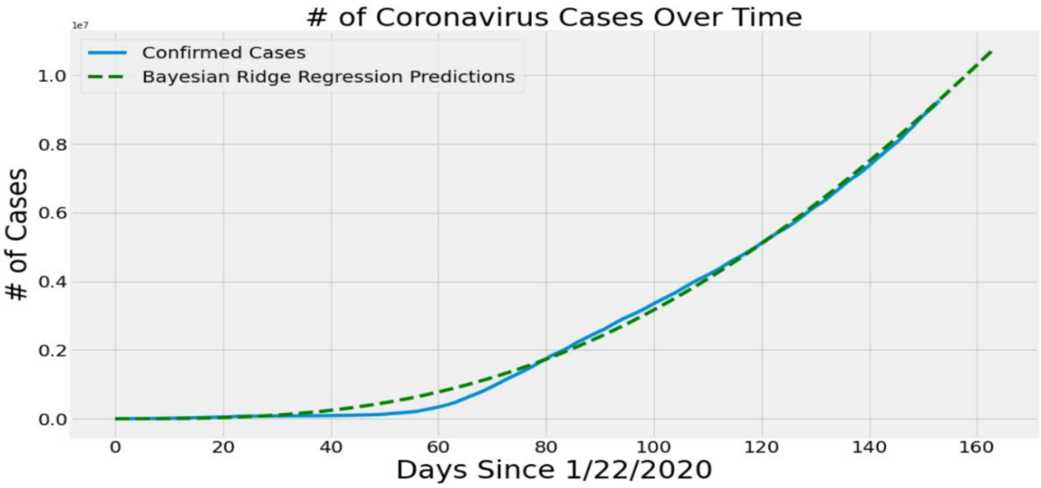
	Date	Polynomial Predicted # of Confirmed Cases Worldwide
0	06/24/2020	9287240.0
1	06/25/2020	9419768.0
2	06/26/2020	9550684.0
3	06/27/2020	9679680.0
4	06/28/2020	9806431.0
5	06/29/2020	9930596.0
6	06/30/2020	10051816.0
7	07/01/2020	10169715.0
8	07/02/2020	10283898.0
9	07/03/2020	10393954.0

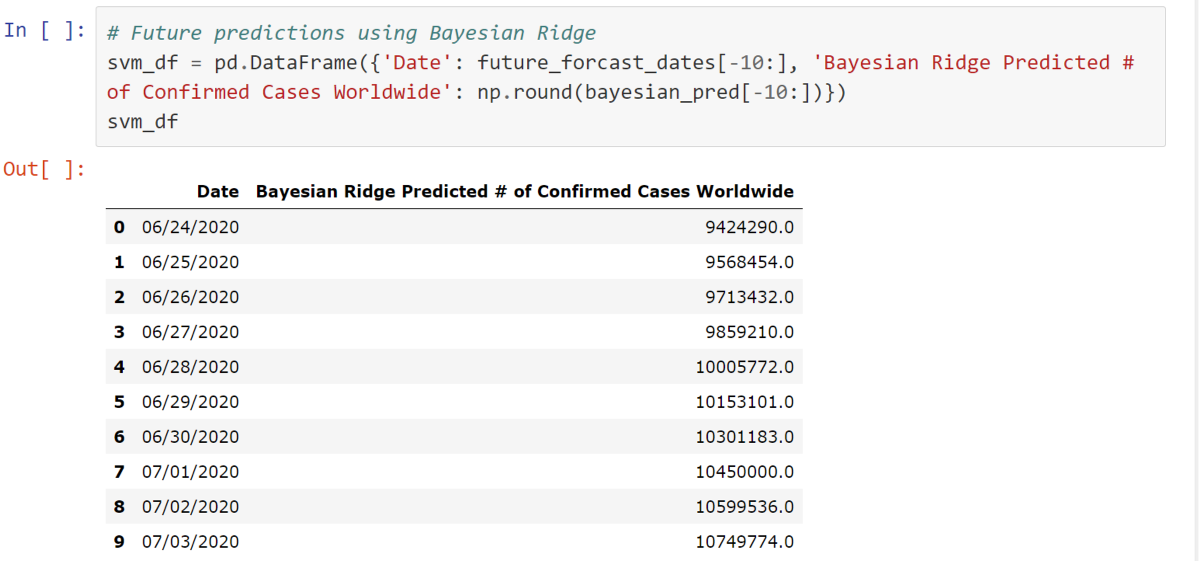
# Prediction using Bayesian Ridge Regression Algorithm

Ridge Regression is a popular type of regularized linear regression that includes an L2 penalty. This has the effect of shrinking the coefficients for those input variables that do not contribute much to the prediction task.

- Ridge Regression is an extension of linear regression that adds a regularization penalty to the loss function during training.

```
In [ ]: plot_predictions(adjusted_dates, world_cases, bayesian_pred, 'Bayesian Ridge Regression
        Predictions', 'green')
```





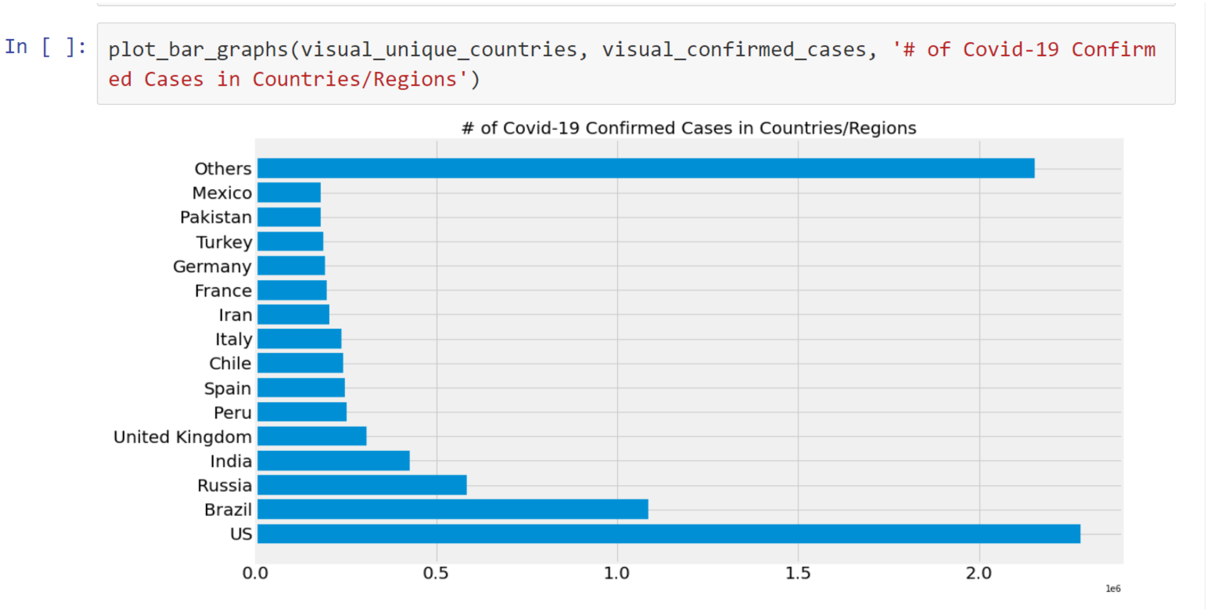
```
In [ ]: # Future predictions using Bayesian Ridge
        svm_df = pd.DataFrame({'Date': future_forcast_dates[-10:], 'Bayesian Ridge Predicted #
        of Confirmed Cases Worldwide': np.round(bayesian_pred[-10:])})
        svm_df
```

Out[ ]:

	Date	Bayesian Ridge Predicted # of Confirmed Cases Worldwide
0	06/24/2020	9424290.0
1	06/25/2020	9568454.0
2	06/26/2020	9713432.0
3	06/27/2020	9859210.0
4	06/28/2020	10005772.0
5	06/29/2020	10153101.0
6	06/30/2020	10301183.0
7	07/01/2020	10450000.0
8	07/02/2020	10599536.0
9	07/03/2020	10749774.0

Bar Chart Visualisation for Covid-19

The **Bar Chart** is a **chart visualization** that you can customize in the Reports section. Each choice in a question is represented as a **bar**, and the length of each **bar** is proportional to the value being measured. This is one of the most easier and efficient way to compare the conditions of various countries here.



```
In [ ]: plot_bar_graphs(visual_unique_countries, visual_confirmed_cases, '# of Covid-19 Confirmed Cases in Countries/Regions')
```

Result

Our Outbreak Visualisation and Prediction Model project successfully visualises and predicts . Visualisation is an efficient way to analyse and understand huge data . Prediction may not be accurate but it is close to accuracy.

Comparison & Analysis

SVM algorithm: support-vector machine constructs a hyperplane or set of hyperplanes in a high 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.

Polynomial Regression algorithm: **polynomial regression** is a form of regression analysis in which the relationship between the independent variable *x* and the *y* is modelled as an dependent variable of *n*th degree polynomial in *x*.

Bayesian Ridge Regression algorithm: **Bayesian linear regression** is an approach to linear regression in which the statistical analysis is undertaken within the context of Bayesian inference. When the regression model has errors that have a normal distribution, and if a particular form of prior distribution is assumed, explicit results are available for the posterior probability distribution of the model's parameters.

Conclusion

The global pandemic of the severe acute respiratory syndrome Coronavirus 2 (SARS-CoV-2) has become the primary national security issue of many nations. Advancement of accurate prediction models for the outbreak is essential to provide insights into the spread and consequences of this infectious disease. Due to the high level of uncertainty and lack of crucial data, standard epidemiological models have shown low accuracy for long-term prediction. This project presents a comparative analysis of ML algorithms to predict the COVID-19 outbreak. Due to the highly complex nature of the COVID-19 outbreak and differences from nation-to-nation, this study suggests ML as an effective tool to model the time series of outbreak. We should note that this paper provides an initial benchmarking to demonstrate the potential of machine learning for future research.

For the advancement of higher performance models for long-term prediction, future research should be devoted to comparative studies on various ML models for individual countries. Due to the fundamental differences between the outbreak in various countries, advancement of global models with generalization ability would not be feasible. As observed and reported in many studies, it is unlikely that an individual outbreak will be replicated elsewhere.

Although the most difficult prediction is to estimate the maximum number of infected patients. The mortality rate is particularly important to estimate accurately the number of patients and the required beds in intensive care units. For future research , modeling the mortality rate would be of the utmost importance for nations to plan for new facilities. For future research integration of machine learning models is suggested to enhance the existing standard epidemiological models in terms of accuracy and longer lead time.

REFERENCES

Novel coronavirus 2019-nCoV: Early estimation of epidemiological parameters and epidemic predictions
medRxiv (2020)
https://scholar.google.com/scholar_lookup?title=Transmission%20of%202019-nCoV%20infection%20from%20an%20asymptomatic%20contact%20in%20Germany&publication_year=2020&author=C.%20Rothe&author=M.%20Schunk&author=P.%20Sothmann

Novel Coronavirus (2019-nCoV) situation report - 1. World Health Organization. 2020 Jan 21.
https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200121-sitrep-1-2019-ncov.pdf

Coronavirus disease: What you need to know. World Health Organization.
https://www.afro.who.int/news/coronavirus-disease-what-you-need-know

Novel coronavirus disease 2019 (COVID-19) pandemic: increased transmission in the EU/EEA and the UK – sixth update. European Centre for Disease Control and Prevention
https://www.ecdc.europa.eu/sites/default/files/documents/RRA-sixth-update-Outbreak-of-novel-coronavirus-disease-2019-COVID-19.pdf

Countries where coronavirus has spread. Worldometer.
https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/

Is India entering stage 3 of covid-19 outbreak? India Today.
https://www.indiatoday.in/programme/to-the-point/video/is-india-entering-stage-3-of-covid-19-outbreak-1661544-2020-03-30

Coronavirus update: has Covid-19 entered Stage 3? Experts, government disagree. Hindustan Times. 2020.
https://www.hindustantimes.com/india-news/has-covid-19-entered-stage-3-experts-government-disagree/story-u22337reY9uO1ZSeHPIUiK.html

#COVID19 Government Measures Dataset. acaps.
https://www.acaps.org/covid19-government-measures-dataset

COVID-19 India.
https://www.covid19india.org/

Das S. Prediction of COVID-19 disease progression in India: under the effect of national lockdown.
http://arxiv.org/abs/2004.03147

Indian Council of Medical Research.
https://icmr.nic.in/sites/default/files/whats_new/ICMR_testing_update_19April_9PM_IST.pdf

Kaggle platform
https://www.kaggle.com

BOOK

Machine Learning Mastery With Python
Understand Your Data, Create Accurate Models and Work Projects End-To-End
© Copyright 2016 Jason Brownlee. Edition: v1.4