CHAPTER-1

INTRODUCTION

1.1 OVERVIEW

Agriculture plays a critical role in the global economy. Pressure on the agricultural system will increase with the continuing expansion of the human population. Agri-Technology and precision farming, now also termed digital agriculture, have arisen as new scientific fields that use data intense approaches to drive agricultural productivity while minimizing its environmental impact. The data generated in modern agricultural operations is provided by a variety of different sensors that enable a better understanding of the operational environment (an interaction of dynamic crop, soil, and weather conditions) and the operation itself (machinery data), leading to more accurate and faster decision making. The agriculture sector is undergoing a transformation driven by new technologies, which seems very promising as it will enable this primary sector to move to the next level of farm productivity and profitability. Precision Agriculture, which consist of applying inputs (what is needed) when and where is needed, has become the third wave of the modern agricultural revolution (the first was mechanization and the second the green revolution with its genetic modification , and nowadays, it is being enhanced with an increase of farm knowledge systems due to the availability of larger amounts of data

Vulnerability of the agricultural sector due to climate change has substantial consequences for Macedonian economy. Considering that the majority of rural population dependents on agriculture, rural communities and especially farmers will be particularly sensitive to the forthcoming challenges of climate changes. The rural poor due to their high dependence on agriculture, relatively low ability to adapt and high share

of food in total costs will be most affected of the adverse effects of climate over agriculture. Any severe climate changes could have devastating outcome reflecting on their financial power, food supplies and the country's economy in general, including export (www, World Bank, 2, 2010).

Climate change may induce an additional price increase for major crops. Higher feed prices will cause higher meat prices that in turn will result in reduction of meat consumption and a more substantial decrease in cereal consumption (Nelson et al, 2009).

Using this model, the study examines alternative crop production patterns in India which is one of the most important agricultural regions in the Asian continent. The study is underpinned only on climate variations in the forthcoming period and based on soil composition using some simplified assumptions and features such as Analyzing the patterns to determine respective crop for the given climatic condition. Other socio-economic factors including price movements based on future supply and demand, investments in agricultural productivity, technology and infrastructure, as well as other production factors are not taken into consideration in this study.

1.2 OBJECTIVE OF THE PROJECT

Factors Affecting:

- Water Resources
- Climate Change
- Soil Composition
- Vulnerability of the agricultural sector in view of climate change, quality of soil, and undersupply of essential nutrients has substantial consequences for Economy.

- The aim of this project is to assess the potential effect of the projected climate change and other attributes upon farm profitability, in particular to crop production.
- Current challenges of water shortages, uncontrolled cost due to demand-supply, and weather uncertainty necessitate farmers to be equipped with smart Agriculture Is the primary source of livelihood which forms the backbone of our country's farming. In particular, low yield of crops due to uncertain climatic changes, poor irrigation facilities, reduction in soil fertility and traditional farming techniques need to be addressed.
- Machine learning is one such technique employed to predict crop yield in agriculture. Various machine learning techniques such as classification, regression are utilized to forecast crop yield. Neural networks, support vector machines, Random Forest are some of the algorithms used to implement prediction.

1.3 Proposed System

This scenario mainly concentrates on crop forecasting, crop yield prediction. These factors help the farmers to cultivate the best food crops and raise the right animals according to environmental components. Also, the farmers can adapt to climate changes to some degree by shifting planting dates, choosing varieties with different growth duration, or changing crop rotations. For experimental analysis, the statistical numeric data related to agriculture is undertaken. Whereas, the clustering based techniques and unsupervised algorithms are utilized for managing the collected statistical data. Additionally, the suitable classification methods like K-means Clustering, Decision Trees (DT) , Random Forest (RF) ,Support Vector Machine (SVM),Logistic Regression (LR), Neural Networks (NN) are employed for better classification outcomes.

1.4 Data Metrics

1. It is necessary to analyze correlating monitoring crop environments with statistical information
2. Based on this statistical information pattern of crop can be obtained.
3. The Database NOSQL used for data queries and plethora of work goes for Data preprocessing.
4. There are different features in the data. For visualization libraries called NumPy, pandas, matplotlib ,seaborn are used.
5. Training the data with clustering algorithms after features extraction.
6. Because climate change affects the productivity of crops with different intensity depending on the crop variety, a diversified farming system could substantially reduce risk and variability of economic returns.
7. As the amount of such information is increasing gradually, We will identify certain patterns in words and slowly start understanding.

1.5 BLOCK DIAGRAM

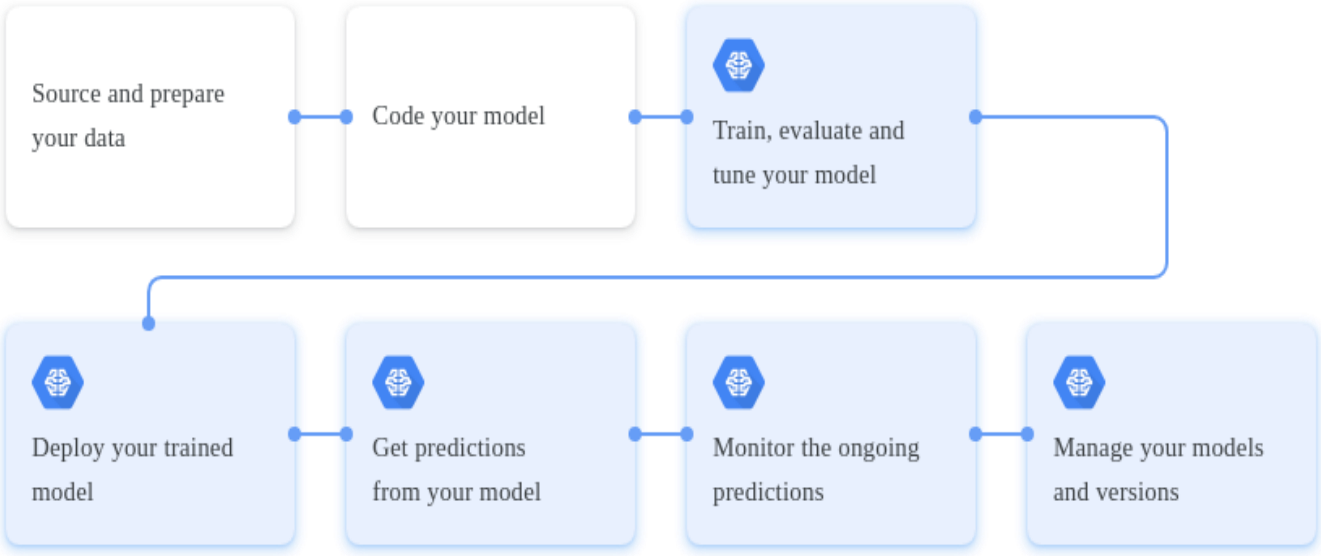

Source and prepare your data
Code your model
Train, evaluate and tune your model
Deploy your trained model
Get predictions from your model
Monitor the ongoing predictions
Manage your models and versions

Figure 1.1

1.6 ALGORITHM

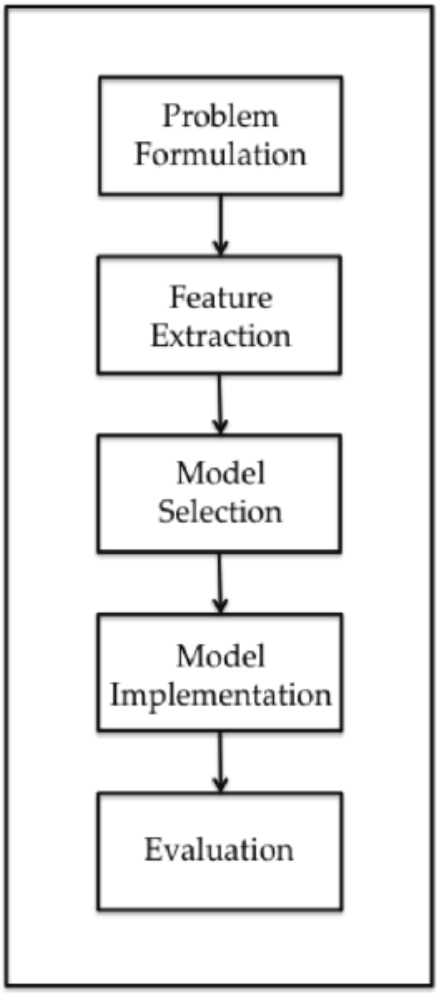


Figure 1.2

1.7 SOFTWARE TOOLS AND LIBRARIES USED

1.7.1 PROGRAMMING LANGUAGE

PYTHON - Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

1.7.2 LIBRARIES USED

NUMPY: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

PANDAS: pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python.

MATPLOT LIB: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

SEABORN: seaborn is an open-source Python library built on top of matplotlib. It is used **for data visualization and exploratory data analysis**. Seaborn works easily with data frames and the Pandas library. The graphs created can also be customized easily.

SCI-KIT LEARN: Scikit-learn (Sklearn) is the most useful and robust **library for machine learning in Python**. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

PICKLE: Python pickle module is **used for serializing and de-serializing a Python object structure**. Any object in Python can be pickled so that it can be saved on disk. Pickling is a way to convert a python object (list, dict, etc.) into a character stream.

CHAPTER 2

MACHINE LEARNING

2.1 INTRODUCTION TO MACHINE LEARNING

What is Machine Learning?

Machine learning is a branch of Artificial Intelligence that involves the design and development of systems capable of showing an improvement in performance based on their previous experiences. This means that when reacting to the same situation, a machine should show an improvement from time to time. With Machine Learning, software systems are able to predict accurately without having to be programmed explicitly.

The goal of Machine Learning is to build algorithms which can receive input data then use statistical analysis so as to predict the output value in an acceptable range. Machine learning originated from pattern recognition and the theory that computers are able to learn without the need for programming them to perform any tasks. Researchers in the field of Artificial Intelligence wanted to determine whether computers are able to learn from data.

Machine learning has been a subject of interest because of its ability to use informa- tion to solve complex problems like facial recognition or handwriting detection. Many times, machine learning algorithms do this by having tests baked in. Examples of these tests are formulating statistical hypotheses, establishing thresholds, and minimizing mean squared errors over time. Theoretically, machine learning algorithms have built a solid foundation. These algorithms have the ability to learn from past mistakes and minimize errors over time.

Machine learning is an iterative approach, and this is why models are able to adapt as they are being exposed to new data. Models learn from their previous computations so as to give repeatable, reliable results and decisions Machine learning is the intersection between theoretically sound computer science and practically noisy data. Essentially,

it's about machines making sense out of data in much the same way that humans do. Machine learning is a type of artificial intelligence whereby an algorithm or method will extract patterns out of data.

2.2 MACHINE LEARNING ALGORITHMS

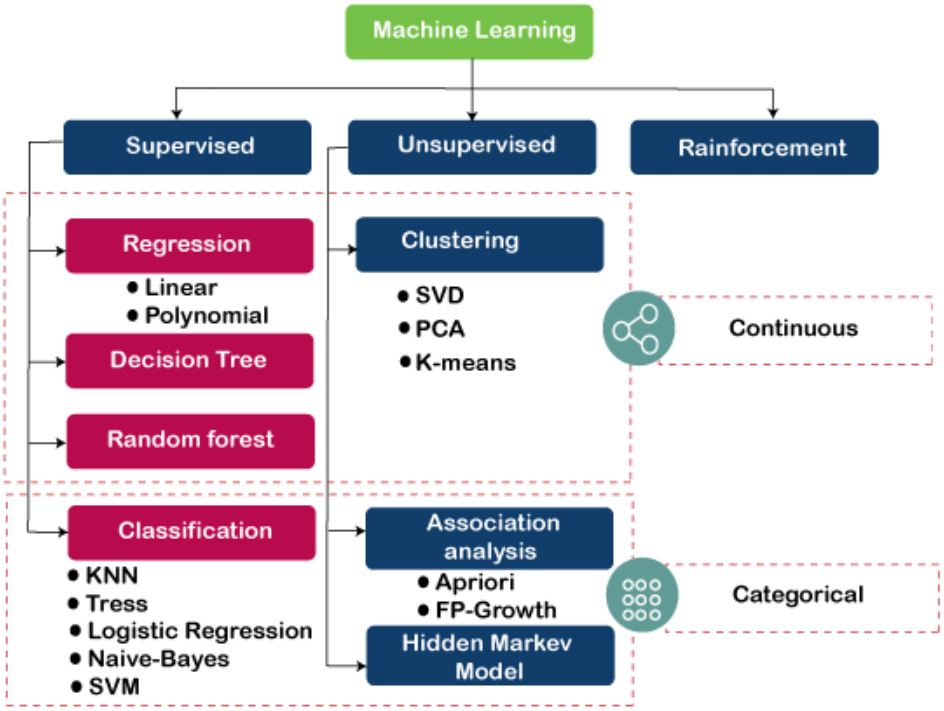


Figure 2.1

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 divides 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 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 votes; 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

2.3 FEATURES OF MACHINE LEARNING

Features are nothing but the variables in machine learning models. What is required to be learnt in any specific machine learning problem is set of these features (variables), coefficients of these features and parameters for coming up with appropriate function (also termed as hyper parameters).

Features can be raw data which are very straightforward and can be derived from real-life as it is. However, not all problems can be solved using raw data or data in its original form. Many times, they need to be represented or encoded in different form. For example, a color can be represented in RGB format or HSV format. Thus, a color can have two different representations or encodings. And, both of these representations or encodings can be used to solve different kind of problems. Some tasks that may be difficult with one representation can become easy with another. For example, the task "select all red pixels in the image" is simpler in the RGB format, whereas "make the image less saturated" is simpler in the HSV format.

In case of machine learning, it is responsibility of data scientists to hand-craft some useful representations / features of data. In case of deep learning, the feature representations are learnt automatically based on the underlying algorithm. One of the most important reasons why deep learning took off instantly is that it completely automates what used to be the most crucial step in a machine-learning workflow: feature engineering

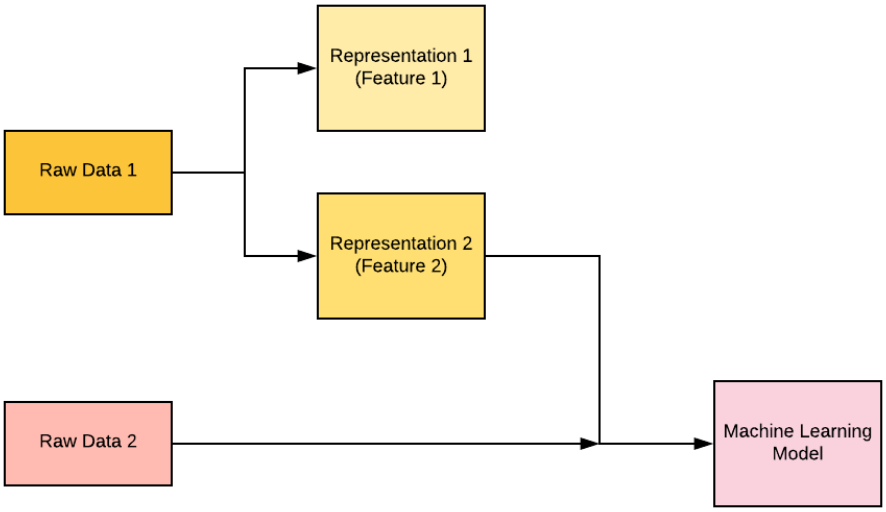


Figure 2.2

2.3 Key characteristics of machine learning

In order to understand the actual power of **machine learning**, you have to consider the characteristics of this technology. There are lots of examples that echo the characteristics of **machine learning** in today's data-rich world. Here are seven key characteristics of **machine learning** for which companies should prefer it over other technologies.

2.3.1- The ability to perform automated data visualization

A massive amount of data is being generated by businesses and common people on a regular basis. By visualizing notable relationships in data, businesses can not only make better decisions but build confidence as well. **Machine learning** offers a number of tools that provide rich snippets of data which can be applied to both unstructured and structured data. With the help of user-friendly automated data visualization platforms in

machine learning, businesses can obtain a wealth of new insights in an effort to increase productivity in their processes.

2.3.2- Automation at its best

One of the biggest characteristics of **machine learning** is its ability to automate repetitive tasks and thus, increasing productivity. A huge number of organizations are already using **machine learning**-powered paperwork and email automation.

2.3.3- Customer engagement like never before

For any business, one of the most crucial ways to drive engagement, promote brand loyalty and establish long-lasting customer relationships is by triggering meaningful conversations with its target customer base. **Machine learning** plays a critical role in enabling businesses and brands to spark more valuable conversations in terms of customer engagement. The technology analyzes particular phrases, words, sentences, idioms, and content formats which resonate with certain audience members. You can think of Pinterest which is successfully using **machine learning** to personalize suggestions to its users. It uses the technology to source content in which users will be interested, based on objects which they have pinned already.

2.3.4- The ability to take efficiency to the next level when merged with IoT

Thanks to the huge hype surrounding the IoT, **machine learning** has experienced a great rise in popularity. IoT is being designated as a strategically significant area by many companies. And many others have launched pilot projects to gauge the potential of IoT in the context of business operations. But attaining financial benefits through IoT isn't easy. In order to achieve success, companies, which are offering IoT consulting services and platforms, need to clearly determine the areas that will change with the implementation of IoT strategies. Many of these businesses have failed to address it. In

this scenario, **machine learning** is probably the best technology that can be used to attain higher levels of efficiency. By merging **machine learning** with IoT, businesses can boost the efficiency of their entire production processes.

2.3.5- The ability to change the mortgage market

It's a fact that fostering a positive credit score usually takes discipline, time, and lots of financial planning for a lot of consumers. When it comes to the lenders, the consumer credit score is one of the biggest measures of creditworthiness that involve a number of factors including payment history, total debt, length of credit history etc. But wouldn't it be great if there is a simplified and better measure? With the help of **machine learning**, lenders can now obtain a more comprehensive consumer picture. They can now predict whether the customer is a low spender or a high spender and understand his/her tipping point of spending. Apart from mortgage lending, financial institutions are using the same techniques for other types of consumer loans.

2.3.6- Accurate data analysis

Traditionally, data analysis has always been encompassing trial and error method, an approach which becomes impossible when we are working with large and heterogeneous datasets. **Machine learning** comes as the best solution to all these issues by offering effective alternatives to analyzing massive volumes of data. By developing efficient and fast algorithms, as well as, data-driven models for processing of data in real-time, **machine learning** is able to generate accurate analysis and results.

2.3.7- Business intelligence at its best

Machine learning characteristics, when merged with big data analytical work, can generate extreme levels of business intelligence with the help of which several different industries are making strategic initiatives. From retail to financial services to healthcare,

and many more — **machine learning** has already become one of the most effective technologies to boost business operations.

2.4 Architecting the Machine Learning Process

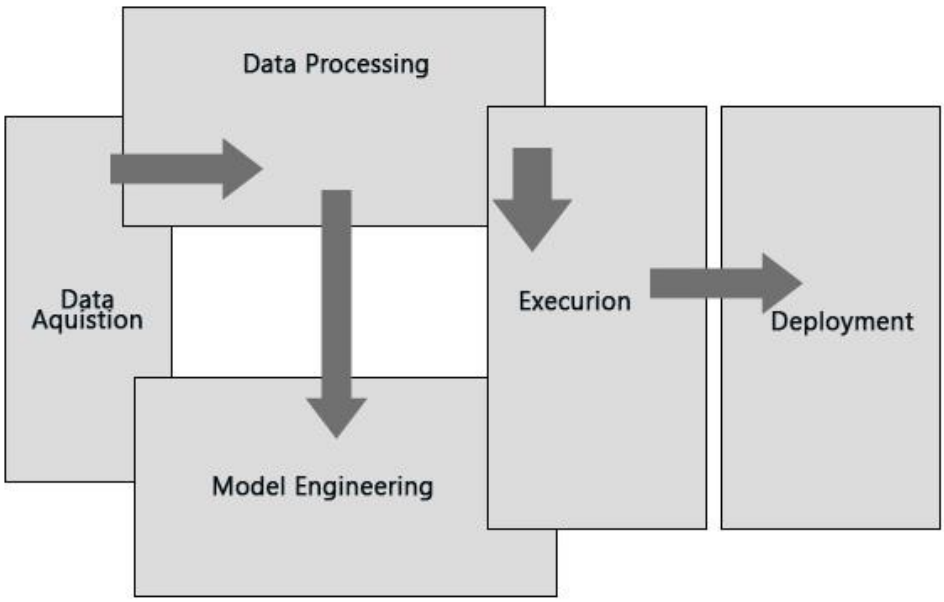


Figure 2.3

1. Data Acquisition

As machine learning is based on available data for the system to make a decision hence the first step defined in the architecture is data acquisition. This involves data collection, preparing and segregating the case scenarios based on certain features involved with the decision making cycle and forwarding the data to the processing unit for carrying out further categorization. This stage is sometimes called the data preprocessing stage. The data model expects reliable, fast and elastic data which may be discrete or continuous in nature. The data is then passed into stream processing systems (for continuous data) and stored in batch data warehouses (for discrete data) before being passed on to data modeling or processing stages.

2. Data Processing

The received data in the data acquisition layer is then sent forward to the data processing layer where it is subjected to advanced integration and processing and involves normalization of the data, data cleaning, transformation, and encoding. The data processing is also dependent on the type of learning being used. For e.g., if supervised learning is being used the data shall be needed to be segregated into multiple steps of sample data required for training of the system and the data thus created is called training sample data or simply training data. Also, the data processing is dependent upon the kind of processing required and may involve choices ranging from action upon continuous data which will involve the use of specific function-based architecture, for example, lambda architecture, Also it might involve action upon discrete data which may require memory-bound processing. The data processing layer defines if the memory processing shall be done to data in transit or in rest.

3. Data Modeling

This layer of the architecture involves the selection of different algorithms that might adapt the system to address the problem for which the learning is being devised, These algorithms are being evolved or being inherited from a set of libraries. The algorithms are used to model the data accordingly, this makes the system ready for the execution step.

4. Execution

This stage in machine learning is where the experimentation is done, testing is involved and tunings are performed. The general goal behind being to optimize the algorithm in order to extract the required machine outcome and maximize the system performance, The output of the step is a refined solution capable of providing the required data for the machine to make decisions.

5. Deployment

Like any other software output, ML outputs need to be operationalized or be forwarded for further exploratory processing. The output can be considered as a non-deterministic query which needs to be further deployed into the decision-making system.

It is advised to seamlessly move the ML output directly to production where it will enable the machine to directly make decisions based on the output and reduce the dependency on the further exploratory steps.

TYPES OF MACHINE LEARNING

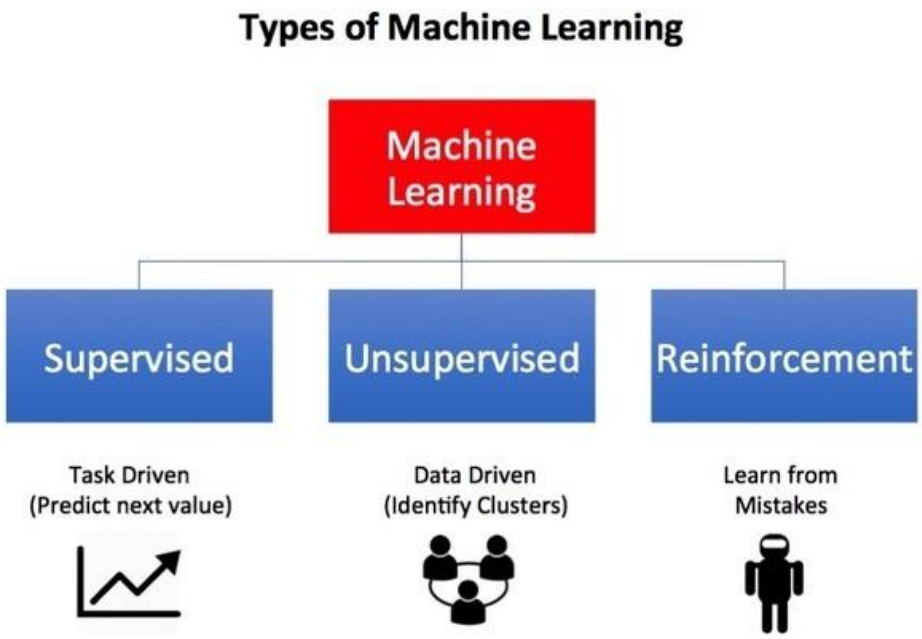


Figure 2.4

Supervised Learning

Supervised learning is the most popular paradigm for machine learning. It is the easiest to understand and the simplest to implement. It is very similar to teaching a child with the use of flash cards.

Unsupervised Learning

Unsupervised learning is very much the opposite of supervised learning. It features no labels. Instead, our algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and/or organize the

data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data.

Reinforcement Learning

Reinforcement learning is fairly different when compared to supervised and unsupervised learning. Where we can easily see the relationship between supervised and unsupervised (the presence or absence of labels), the relationship to reinforcement learning is a bit murkier. Some people try to tie reinforcement learning closer to the two by describing it as a type of learning that relies on a time-dependent sequence of labels, however, my opinion is that that simply makes things more confusing.

CHAPTER 3

INTRODUCTION TO PYTHON PROGRAMMING LANGUAGE

3.1 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 whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

3.1.1 Why Python?

Simple and Easy to Learn he syntax and the keywords used in it are more usually in English. The quality of this language is such that it allows the programmer to concentrate on problem-solving instead of worrying about learning the syntax and how to use it. All this baggage is usually associated with several other high-level languages.

High-level Language

A high-level language means that from a developer's perspective, various internal details like memory management are abstracted from you and are then automatically taken care of by the language. This is the best high-level language for a non-programmer to learn.

Fast and Efficient to Use

Python is incredibly quick to act when it comes to the execution of actions and this is another feature that makes Python a powerful language. Any program that is written in Python can be embedded and executed just like a script within programs coded in other languages like C or C++. You can also write a simple Python script and then use it to execute any other C/C++ programs.

Open Source

Since this language is an open source, it is available free of cost. It essentially means that you can write and distribute a code written in Python. The code of Python is well-maintained so it can be constantly reused and improved upon by developed all over the world.

Object Oriented

As with any other modern language, even Python has an object-oriented approach towards programming. The code is organized in classes that are referred to as templates and "objects" is an example of such a class. Objects are the building blocks of an object-oriented programming language-it effectively combines data along with the methods that are used to perform any functions on this data.

GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

3.1.2 NumPy, Pandas and Scikit-learn

Python offers a large inbuilt image and video library that come handy while dealing with the feature extraction phase. This feature makes Python desirable and easy to use language for Machine Learning. The Scikit-learn package also helps in different stages of building a Machine Learning model; training the model and evaluating the system, thereby making the whole pipeline come together seamlessly. Pytorch is a good alternative for beginners.

3.2 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.

3.3 Python Features

Python's features include:

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined 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 and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable: You can add low-level modules to the Python interpreter. These Modules enable programmers to add to or customize their tools to be more efficient.

Databases: Python provides interfaces to all major commercial databases.

Scalable: Python provides a better structure and support for large programs than shell scripting.

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

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

CHAPTER 4

DATA ENGINEERING

4.1 Collecting Data

Data is one of the most valuable resources today's businesses have. The more information you have about your customers, the better you can understand their interests, wants and needs. This enhanced understanding helps you meet and exceed your customers' expectations and allows you to create messaging and products that appeal to them.

Determine What Information You Want to Collect

The first thing you need to do is choose what details you want to collect. You'll need to decide what topics the information will cover, who you want to collect it from and how much data you need. Your goals — what you hope to accomplish using your data — will determine your answers to these questions. As an example, you may decide to collect data about which type of articles are most popular on your website among visitors who are between the ages of 18 and 34. You might also choose to gather information about the average age of all of the customers who bought a product from your company within the last month.

Set a Timeframe for Data Collection

Next, you can start formulating your plan for how you'll collect your data. In the early stages of your planning process, you should establish a timeframe for your data collection. You may want to gather some types of data continuously. When it comes to transactional data and website visitor data, for example, you may want to set up a method for tracking that data over the long term. If you're tracking data for a specific campaign, however, you'll track it over a defined period. In these instances, you'll have a schedule for when you'll start and end your data collection.

Determine Your Data Collection Method

At this step, you will choose the data collection method that will make up the core of your data-gathering strategy. To select the right collection method, you'll need to consider the type of information you want to collect, the timeframe over which you'll obtain it and the other aspects you determined.

Collect the Data

Once you have finalized your plan, you can implement your data collection strategy and start collecting data. You can store and organize your data in your DMP. Be sure to stick to your plan and check on its progress regularly. It may be useful to create a schedule for when you will check in with how your data collection is proceeding, especially if you are collecting data continuously. You may want to make updates to your plan as conditions change and you get new information.

Analyze the Data and Implement Your Findings

Once you've collected all of your data, it's time to analyze it and organize your findings. The analysis phase is crucial because it turns raw data into valuable insights that you can use to enhance your marketing strategies, products and business decisions. You can use the analytics tools built into our DMP to help with this step. Once you've uncovered the patterns and insights in your data, you can implement the findings to improve your business.

4.2 Data Preprocessing

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

Dataset and Pre-processing:

We needed a dataset of fake and genuine profiles. Various attributes included in the dataset are number of friends, followers, status count. Dataset is divided into training and testing data. Classification algorithms are trained using training dataset and testing dataset is used to determine efficiency of algorithm. From the dataset used, 80% of both

profiles (genuine and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset.

Three common data pre-processing steps are :

- **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 anonymized 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.

4.3 Feature Selection:

Features are selected to apply classification algorithms. The classification algorithm is discussed further. Attributes are selected as features if they are not dependent on other attributes and they increase efficiency of the classification. The features that we have chosen are discussed further.

After selection of attributes, the dataset of profiles that are already classified as fake or genuine are needed for the training purpose of the classification algorithm. We have used a publicly available dataset of 1337 fake users and 1481 genuine users consisting of various attributes including name, status count, number of friends, followers count, favorites, languages known etc.

4.4 Classification:

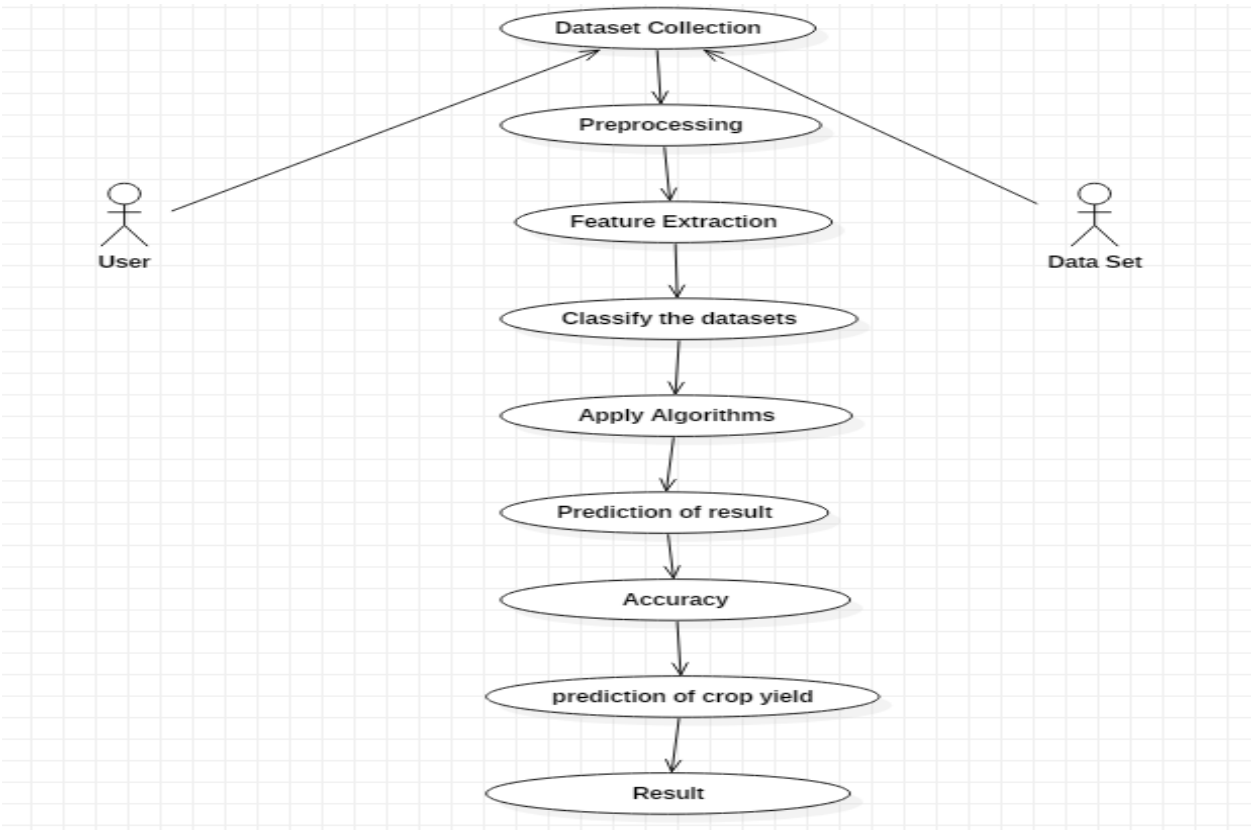
Classification is the process of categorizing a data object into categories called classes based upon features/attributes associated with that data object. Classification uses a classifier, an algorithm that processes the attributes of each data object and outputs a class based upon this information. In this project, we use Support Vector Machine, Decision tree, Random Forest ,Logistic Regression and Neural Network as a classifier. Support Vector Machine is an elegant and robust technique for classification on a large data set not unlike the data sets of Social Network with several millions of profiles.

CHAPTER 5

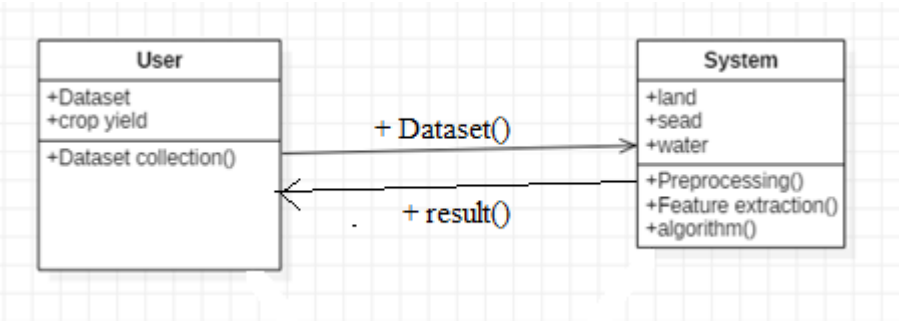
SCHEMATIC DIAGRAM AND ITS WORKING

5.1 UML DIAGRAMS

USE CASE DIAGRAM:



5.2 CLASS DIAGRAM:



5.3 SEQUENCE DIAGRAM:

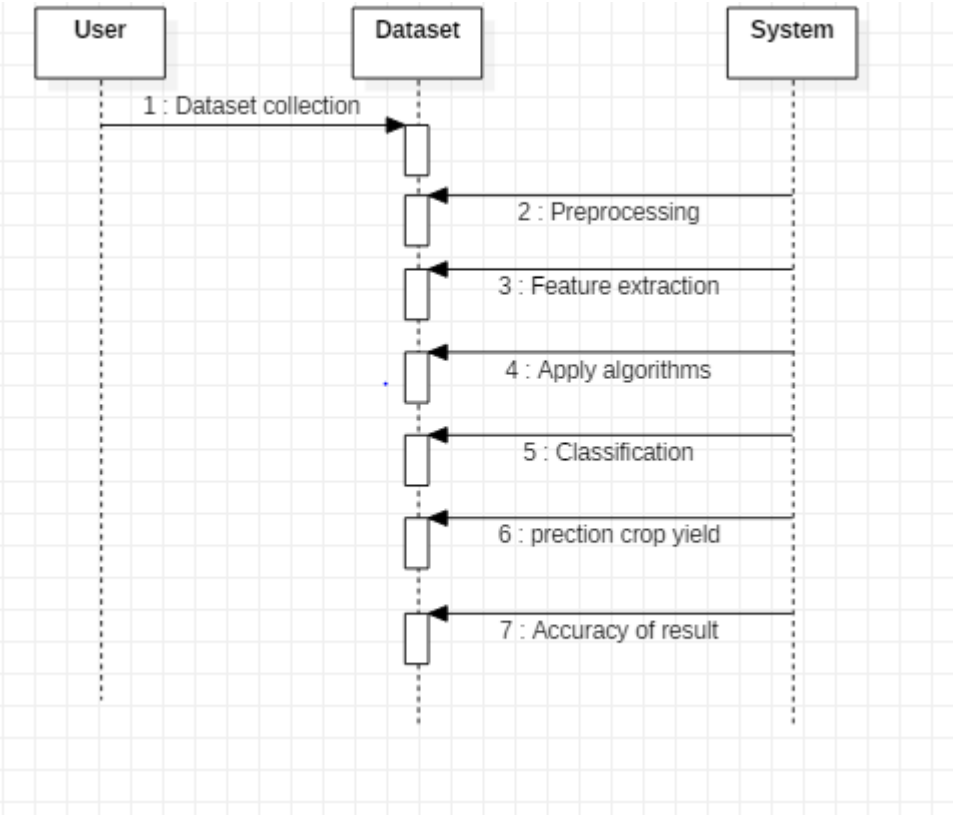
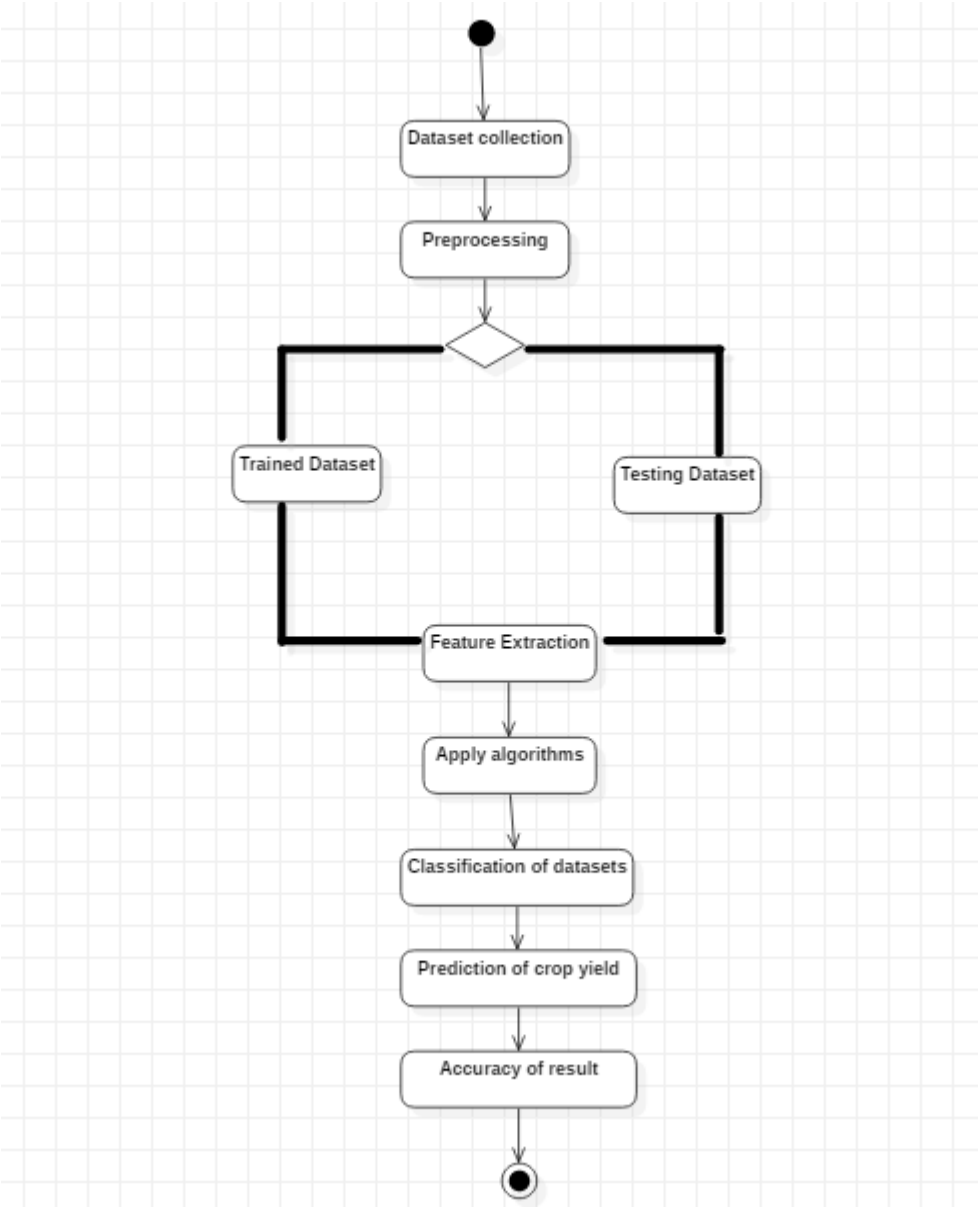


Figure 4.3

5.4 ACTIVITY DIAGRAM:



5.5 END TO END PROCESS

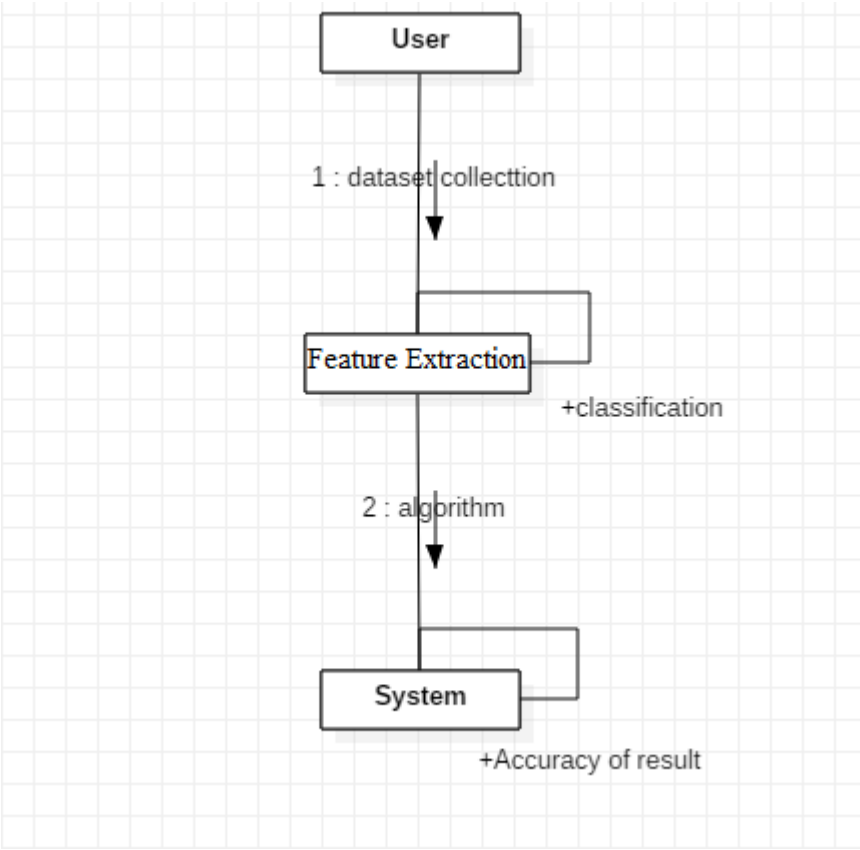


Figure 4.5

5.6 WORKING

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

5.6.1 K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs to only one group that has similar properties.

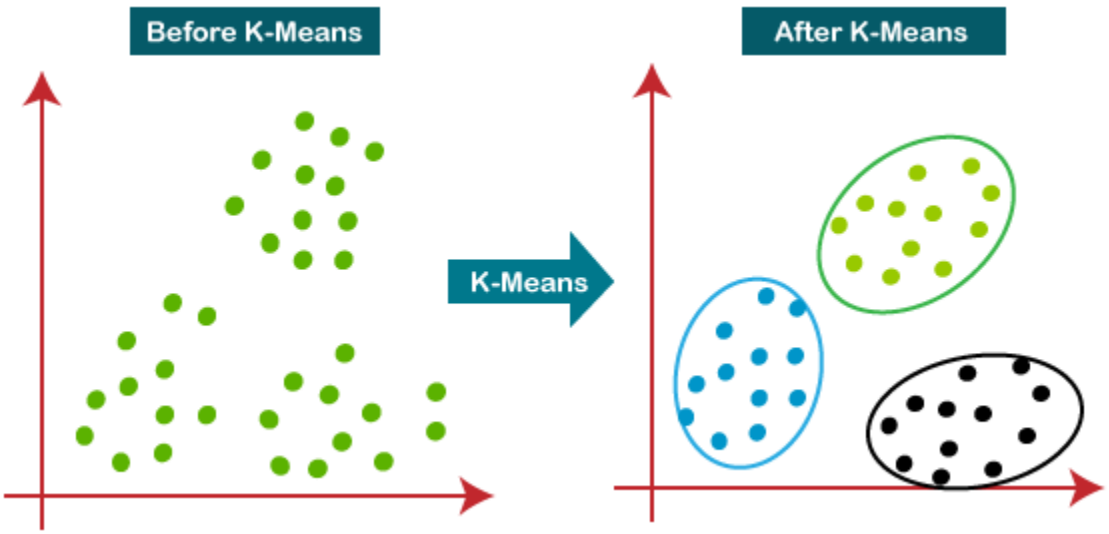


Figure 5.1

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

How does it work?

How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

5.6.2 SciKit Learn

In general, a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. If each sample is more than a single number and, for instance, a multi-dimensional entry (aka multivariate data), it is said to have several attributes or features.

Learning problems fall into a few categories:

supervised learning, in which the data comes with additional attributes that we want to predict. This problem can be either:

classification: samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class.

regression: if the desired output consists of one or more continuous variables, then the task is called *regression*. An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight.

unsupervised learning, in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.

Training set and testing set

Machine learning is about learning some properties of a data set and then testing those properties against another data set. A common practice in machine learning is to evaluate an algorithm by splitting a data set into two. We call one of those sets the **training set**, on which we learn some properties; we call the other set the **testing set**, on which we test the learned properties.

5.6.3 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.

- 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.

Advantages of k-means:

Relatively simple to implement.

Scales to large data sets.

Guarantees convergence.

Can warm-start the positions of centroids.

Easily adapts to new examples.

Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

5.7 PYTHON ENVIRONMENT

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
- Importing Datasets

5.7.1 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 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 a table in 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) and has optional dependencies (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 , "a cross platform distribution for data analysis and scientific computing."

In order to use Pandas in your Python IDE (Integrated Development Environment) like Jupyter Notebook or 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 file types 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() 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 .

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 a column with label col as Series or a few columns (df[[col1, col2]]) and return 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.

5.7.2 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.isnull() 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.isnull().sum(). pd.notnull() is the opposite of pd.isnull(). After you get a list of missing values you can get rid of them, or drop them by using df.dropna() to drop the rows or df.dropna(axis=1) to drop the columns. A different approach would be to fill the missing values with other values by using df.fillna(x) which fills the missing values with x (you can put there whatever you want) or s.fillna(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 colhave identical values. how can be equal to one of: 'left', 'right', 'outer', 'inner'

5.7.3 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 ndarray 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.

5.7.4 Sklearn

In python, scikit-learn library has a pre-built functionality under sklearn. 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 the 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.

5.8 SEABORN

5.8.1 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 an excellent library for you.

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

- **Matplotlib:** low level, provides lots of freedom
- **Pandas Visualization:** easy to use interface, built on Matplotlib
- **Seaborn:** high-level interface, great default styles
- **ggplot:** based on R's ggplot2, uses Grammar of Graphics
- **Plotly:** 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.

5.9 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 and conda can be used.

2. pip install matplotlib

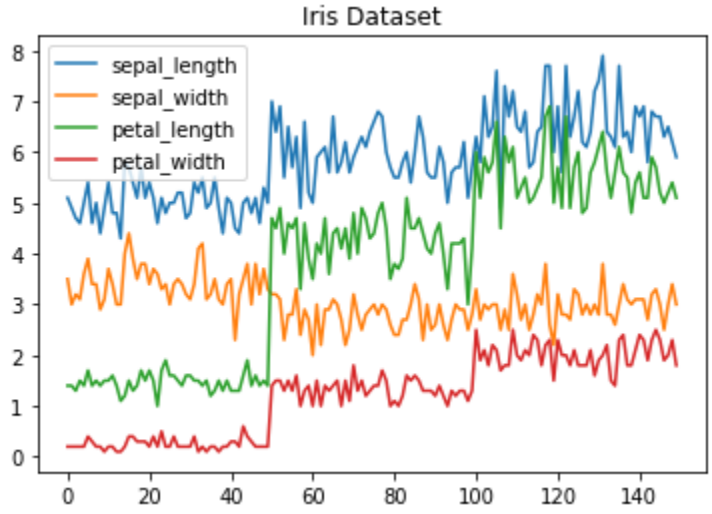
3. conda install matplotlib

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.

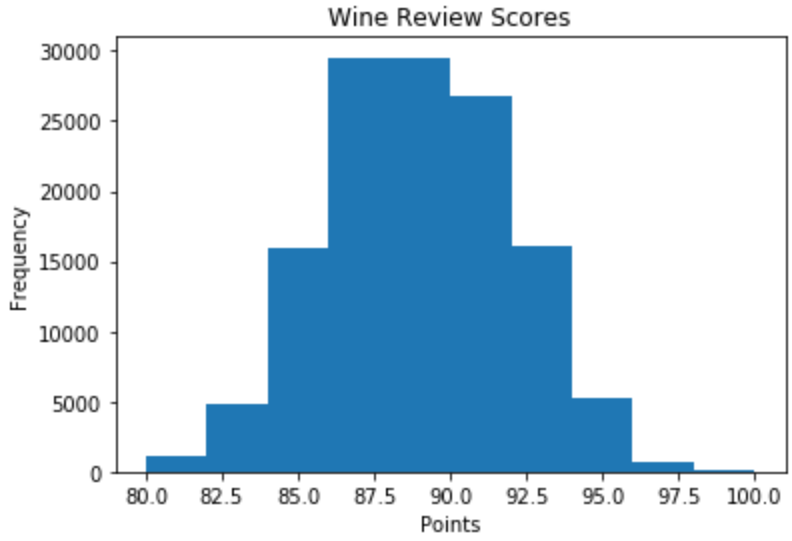


Line Chart

Figure 5.2

Histogram

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

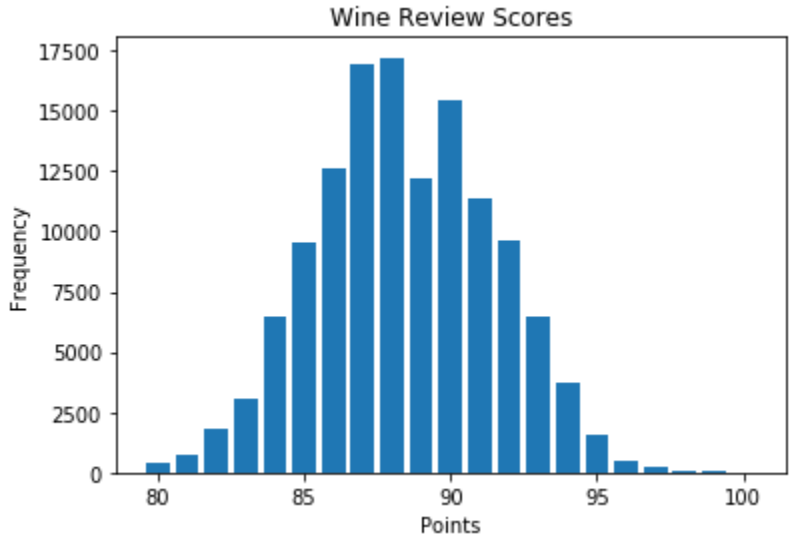


Histogram

Figure 5.3

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 otherwise it can get quite messy.



Bar-Chart

Figure 5.4

Pandas Visualization

Pandas is an 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.

1. **Pandas can be installed using either pip or conda.**

2. **pip install pandas**

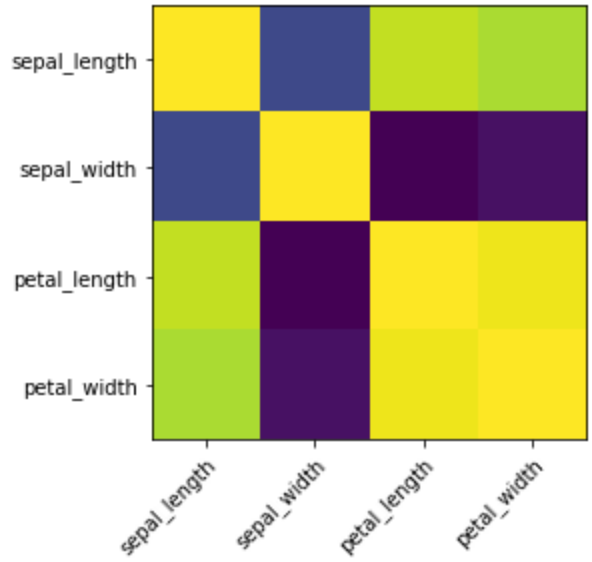
3. **conda install pandas**

Heatmap

A Heatmap is a graphical representation of data where the individual values contained in a matrix 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.

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

Matplotlib:



Heatmap without annotations

Figure 5.5

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.

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:

- Functions: Identified functions must be exercised.
- Output: Identified classes of software outputs must be exercised.
- Systems/Procedures: system should work properly

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.

Data Visualization

Data visualization is the process of translating large data sets and metrics into charts, graphs and other visuals. The resulting visual representation of data makes it easier to identify and share real-time trends, outliers, and new insights about the information represented in the data.

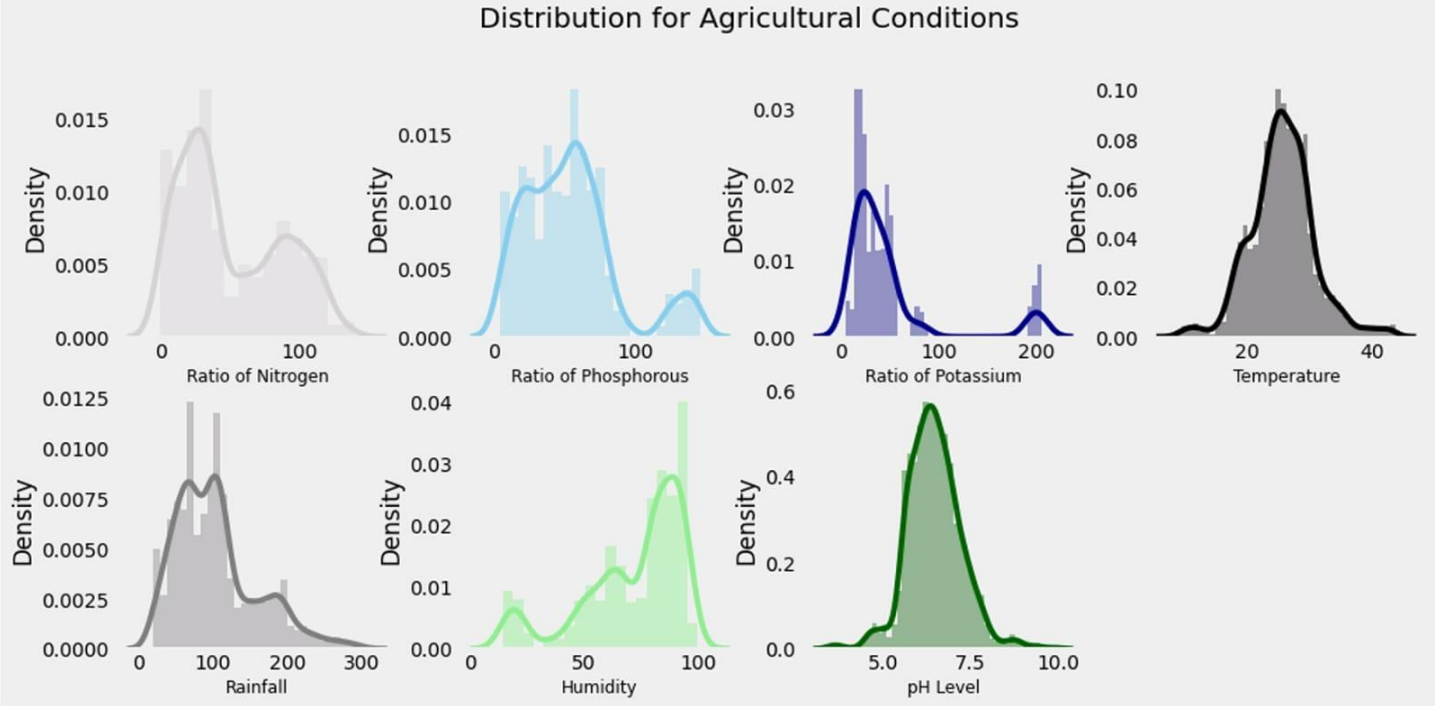


Figure 5.6

In cluster analysis, the elbow method is **a heuristic used in determining the number of clusters in a data set**. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use.

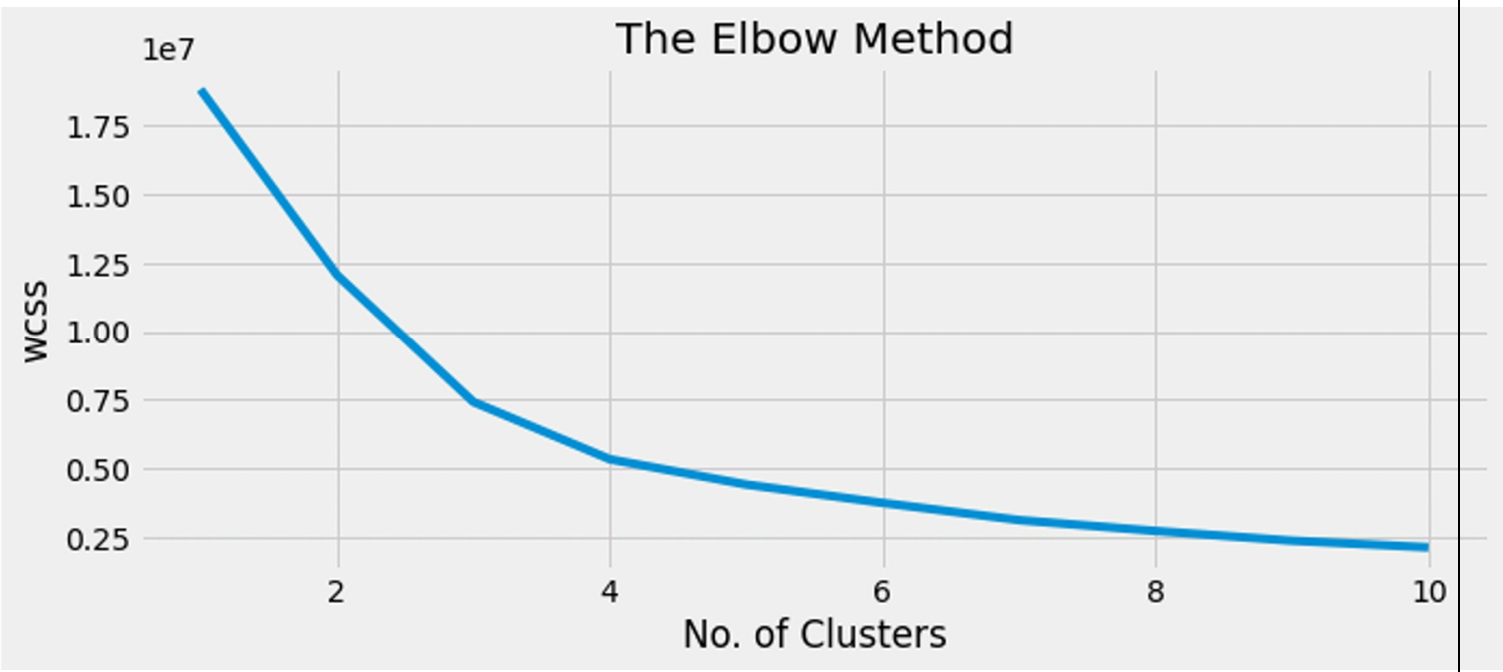


Figure 5.7

Statistical Analysis of data

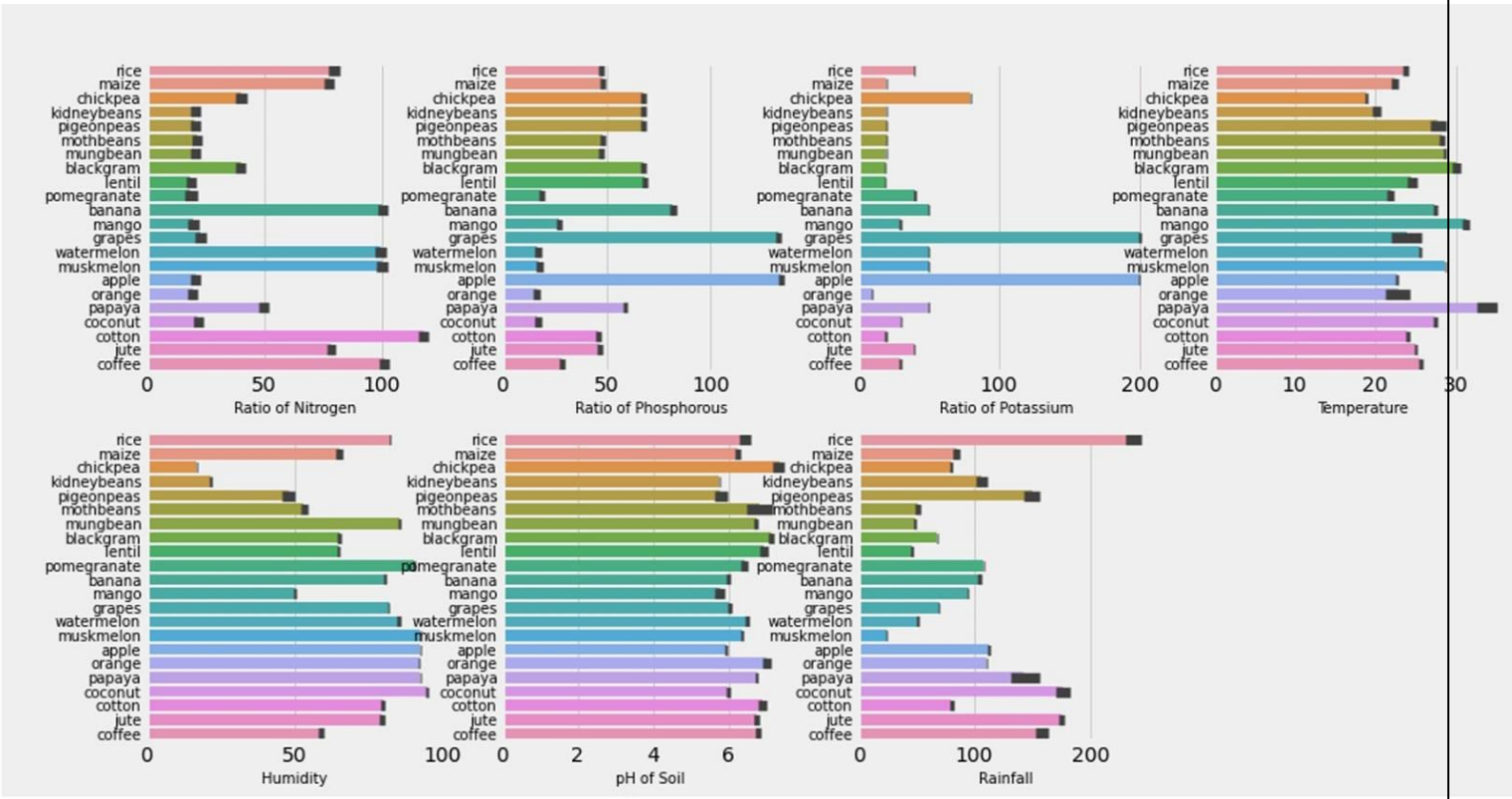


Figure 5.8

Prediction Using K-Means Clustering Algorithm

In [3]:
```
# lets check the head of the dataset
data.head()
```

Out[3]:

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Figure 5.9

12/11/2021, 15:31 Optimizing Agricultural Production (1)

In [20]:
```
# lets create Training and Testing Sets for Validation of Results
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, ra

print("The Shape of x train:", x_train.shape)
print("The Shape of x test:", x_test.shape)
print("The Shape of y train:", y_train.shape)
print("The Shape of y test:", y_test.shape)
```

```
The Shape of x train: (1760, 7)
The Shape of x test: (440, 7)
The Shape of y train: (1760,)
The Shape of y test: (440,)
```

Figure 5.10

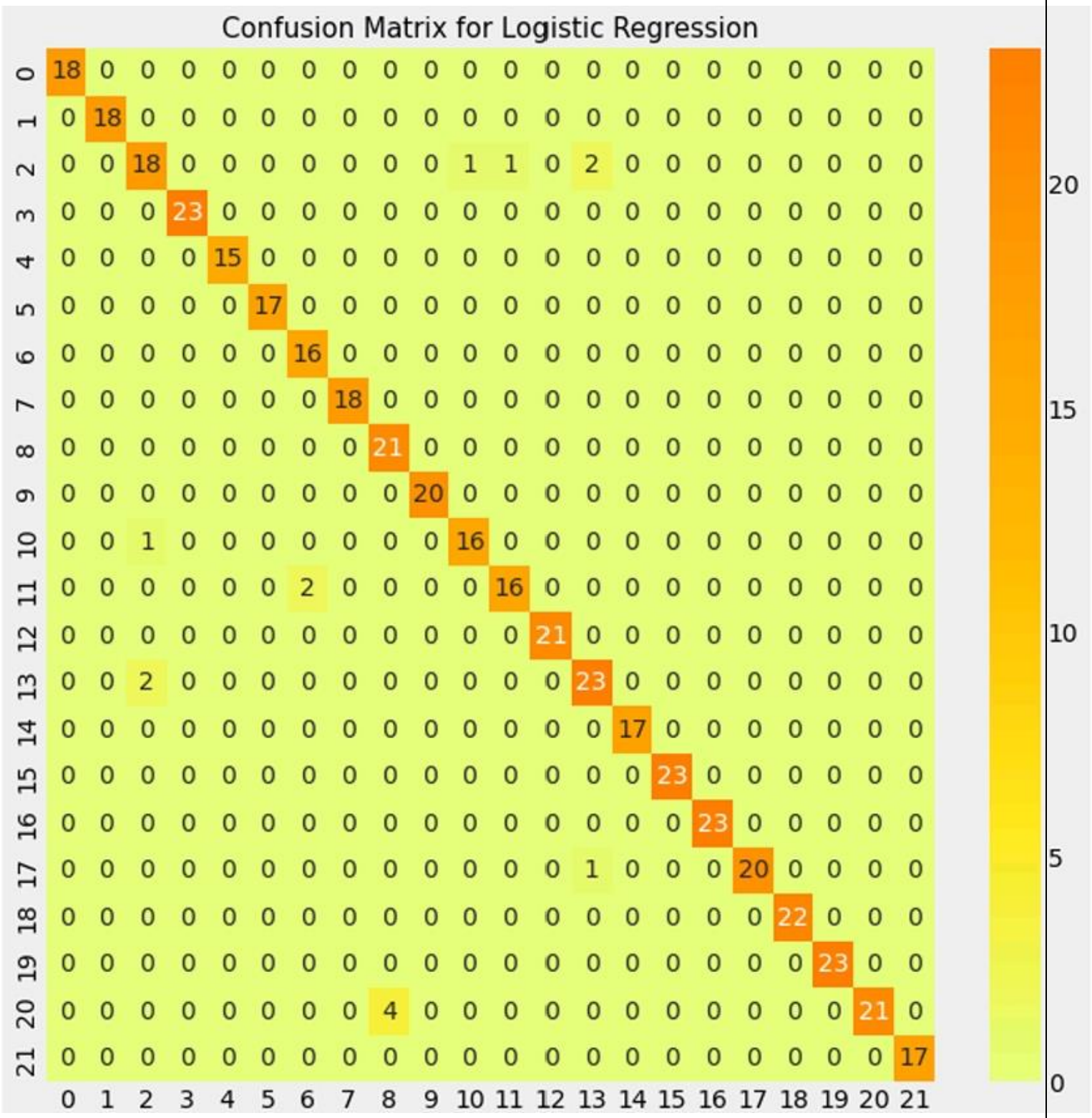

Confusion Matrix for Logistic Regression
0 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 18 0 0 0 0 0 0 0 1 1 0 2 0 0 0 0 0 0 0 0
3 0 0 0 23 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 21 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 1 0 0 0 0 0 0 0 16 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 2 0 0 0 0 16 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0 0 0 21 0 0 0 0 0 0 0 0 0
13 0 0 2 0 0 0 0 0 0 0 0 0 0 23 0 0 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17 0 0 0 0 0 0 0
15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 23 0 0 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 23 0 0 0 0 0
17 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 20 0 0 0 0
18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 22 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 23 0 0
20 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 21 0
21 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
20
15
10
5
0

Figure 5.11

CHAPTER 6

ADVANTAGES AND APPLICATIONS

6.1 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 its massive face recognition. The government uses Artificial intelligence to prevent jaywalkers.

Healthcare industry

- Healthcare was one of the first industries 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 developed advanced mathematical tools like

Bayesian analysis to estimate the value of a customer. With the boom of data, the 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 the past year the stock manager relied 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 terms 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 the 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 terms of AI. In deep learning, the learning phase is done through a neural network.

Reinforcement Learning

Reinforcement learning is 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. 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 beginning 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, they hired higher quality staff.

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.

6.2 Advantages of Machine Learning

1. Automates Repetitive Tasks

Automation is getting commonplace almost everywhere in the world. One benefit of machine learning's nature is that your business is guaranteed to save time and money. Developers and data analysts can have more time for other higher-level tasks which a computer can't handle. Meanwhile, let your machine learning platform take over the functions that have already been redundantly performed.

Let the algorithm do the complicated jobs in your company, allowing you to cut on the workload. Large volumes of data will be reviewed by identifying patterns and cycles that many people can't easily pinpoint. This results in the most efficient data mining system, which then results in the expansion of technological discoveries.

2. Keeps Improving Over Time

Machine learning algorithms improve on their own as they continue to be generated. Its technology is continually evolving and enhancing its efficiency and accuracy. Because of the increasing data being processed and evaluated, the system becomes even more accurate than it originally is. Such an algorithm and closer to a 100% accuracy rate become useful for making better decisions and forecasts in your business.

A computer's hardware and software benefit from machine learning, as it processes various data and networks, which makes the processing power of the system faster. As the current algorithms become error-free, they can reliably design more efficient algorithms further. The more data you input in your data set, the more accurate your forecasts will be. This will lead to having machine learning as a leading technology in the coming years.

3. Has Wide Application

Different businesses and organizations can capitalize on the merits of machine learning in helping their market growth and increasing human work performance. Healthcare providers, e-commerce site owners, and manufacturers all use machine learning to stay ahead of the game in their respective niches.

Even the education industry also embraces machine learning to help out students in their study routines. Currently, Chinese students are utilizing machine learning platforms to improve their focus.

As for online shopping, many online sites rely on machine learning processes to study their target audience's searches. Many people have now grown accustomed to remote working systems, and many businesses have shifted to online platforms. Machine learning will help them manage their businesses better.

CHAPTER 7

7.1 RESULTS:

In [35]:
```
# lets do some Real time Predictions
prediction = model.predict((np.array([[90,
                                       60,
                                       70,
                                       50,
                                       90,
                                       5.0,
                                       90]])))
print("The Suggested Crop for Given Climatic Condition is :", prediction)
```

```
The Suggested Crop for Given Climatic Condition is : ['papaya']
```

CHAPTER 8

8.1 Conclusion

The Results show that we can attain an accurate crop prediction using the K- Means Clustering algorithm. K-Means Clustering algorithm achieves a large number of crop yield models with the lowest models. It is suitable for massive crop yield prediction in

agricultural planning. This makes the farmers take the right decision for the right crop such that the agricultural sector will be developed by innovative ideas.

8.2 Future Scope

This paper describes the crop yield prediction ability of the algorithm. In the future we can determine the efficient web enhancement or application based on their accuracy metrics that will help to choose an efficient algorithm for crop yield prediction.

REFERENCES

[1] D. R. Legates, R. Mahmood, D. F. Levia, T. L. DeLiberty, S. M. Quiring, C. Houser, and F. E. Nelson, "Soil moisture: A central and unifying theme in physical geography," Progress in Physical Geography, vol. 35, no. 1, pp. 65–86, 2011.
[2] Y. H. Kerr, P. Waldteufel, J.-P. Wigneron, J. Martinuzzi, J. Font, and M. Berger, "Soil moisture retrieval from space: The soil moisture and ocean salinity (smos) mission," IEEE transactions on Geoscience and remote sensing, vol. 39, no. 8, pp. 1729–1735, 2001.
[3] E. G. Njoku, T. J. Jackson, V. Lakshmi, T. K. Chan, and S. V. Nghiem, "Soil moisture retrieval from amsr-e," IEEE transactions on Geoscience and remote sensing, vol. 41, no. 2, pp. 215–229, 2003.
[4] R. H. Reichle, R. D. Koster, J. Dong, and A. A. Berg, "Global soil moisture from satellite observations, land surface models, and ground data: Implications for data assimilation," Journal of Hydrometeorology, vol. 5, no. 3, pp. 430–442, 2004.
[5] S. Lambot, E. C. Slob, I. van den Bosch, B. Stockbroeckx, and M. Vanclooster, "Modeling of ground-penetrating radar for accurate characterization of subsurface electric properties," IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 11, pp. 2555–2568, 2004.
[6] M. S. Dawson, A. K. Fung, and M. T. Manry, "A robust statisticalbased estimator for soil moisture retrieval from radar measurements," IEEE transactions on geoscience and remote sensing, vol. 35, no. 1, pp. 57–67, 1997. [7] G. Satalino, F. Mattia, M. W. Davidson, T. Le Toan, G. Pasquariello, and M. Borgeaud, "On current limits of soil moisture retrieval from ers-sar data," IEEE Transactions on Geoscience and Remote Sensing, vol. 40, no. 11, pp. 2438–2447, 2002.