PYTHON

FOR DATA SCIENCE

IVOR OSBORNE

PYTHON DATA SCIENCE

The Complete Step-by-Step Python Programming Guide. Learn How to Master Big Data Analysis and Machine Learning (2022 Edition For Beginners)

Ivor Osborne

Table of Contents

TABLE OF CONTENTS

CHAPTER ONE

DATA VISUALIZATION AND MATPLOTLIB

CHAPTER TWO

NEURAL NETWORKS

CHAPTER THREE

DECISION TREES

CHAPTER FOUR

APPLICATIONS OF BIG DATA ANALYSIS

CONCLUSION

Chapter One
Data Visualization and Matplotlib

At some point throughout your journey of dealing with all of that data and attempting to understand it, you will determine that it is time to truly view and visualize the data. Sure, we could put all of that information into a table or write it out in long, monotonous paragraphs, but it will make it difficult to focus and understand the comparisons and other information that comes with it. One of the greatest ways to do this is to take that information and, after it has been thoroughly examined, apply data visualization to make it work for our purposes.

The good news is that the Matplotlib library, which is a NumPy and SciPy extension, can assist us in creating all of them. The Matplotlib library can

help you visualize your data as a line graph, a pie chart, a histogram, or in another way.

When it comes to this topic, the first thing we're going to try and investigate is data visualization. We need a better understanding of what data visualization is, how it works, why we would want to utilize it, and so on. When we can put all of these pieces together, it becomes lot easier to take all of the data we've been collecting and visualize it in a way that allows us to make smart business decisions.

Data visualization is the presenting of quantitative information in a more pictorial way. In other words, data visualization may be used to transform a set of data, whether tiny or vast, into visuals that are much easier for the brain to interpret and process. It is a fairly typical event in your daily life, but you may recognize them best from graphs and charts that assist you better understand the facts. Infographics will be defined as a combination of more than one of these visualizations, as well as some additional information thrown in for good measure.

You will discover that there are numerous advantages to employing this type of data visualization. It can be used to assist with trends and facts that are unknown and may be concealed deep inside the information that you have, particularly in bigger quantities of data. These visualizations include inline charts, which show how something changes over time, column and bar charts, which allow you to make comparisons and observe the relationship between two or more objects, and pie charts, which indicate how much of a whole something takes up.

These are just a few instances of data visualization that you may come across while working.

The second thing we need to consider is what will make for good data

visualization. These will be built when we can combine design, data science, and communication into one. When done correctly, data visualization may provide us with vital insights into complex collections of data, and this is done in a way that makes them more relevant for those who use them, and more intuitive overall.

Many businesses will profit from this type of data visualization, and it is a tool that should not be overlooked. When you are attempting to evaluate your data and want to ensure that it matches up properly and that you fully appreciate all of the information that is being presented, it is a good idea to look at some of the plots later on and pick which ones can best talk about your data and which one you should utilize.

You may be able to extract all of the information you require from your data and complete your Data Analysis without the use of these graphics. However, if you have a large amount of data and don't want to miss anything and don't have time to look through it all, these visualizations will come in handy. The finest data visualizations will incorporate a variety of complicated ideas that can be communicated in a way that is efficient, precise, and clear, to mention a few.

To assist you in creating a good type of data visualization, ensure that your data is clean, well-sourced, and thorough. This is a step that we covered previously in the guidebook, so your data should be ready by now. When you're ready and the data is ready, it's time to look at some of the charts and plots that are available for use. This might be difficult and challenging at times depending on the type of information you are working with at the moment. You must select the chart type that appears to work best for the data you have available.

After you've done your research and determined which style of chart is best

for you, it's time to go through and create, as well as customize, that chosen graphic to work best for you. Keep your graphs and charts as simple as possible because these are frequently the greatest forms of graphs. You don't want to waste time throwing in components that aren't necessary and will merely distract us from the data.

The visualization should be finished at this point. You've chosen the one you want to use, after sorting and cleaning your chosen data, and then you've chosen the proper chart to use, bringing in the code that goes with it to obtain the greatest results. Now that this section is complete, it is time to take that visualization, publish it, and share it with others.

Why is it important to use data visualization

With that information in mind, let's take a look at some of the advantages of using data visualization and why so many people enjoy working with it. While it is possible to do the analysis and more on your own without the graphs and other visuals, this is often a poor way to make decisions and does not ensure that you understand what is going on with the data in front of you or that you see the full amount of information and trends that are presented. In addition, some of the additional reasons that data analysts prefer to work with these types of visualizations are as follows:

It assists them in making better decisions. Today, more than ever before, businesses have decided to examine a number of data tools, including data visualizations, in order to ask the appropriate questions and make the best decisions for them. Emerging computer technology and user-friendly software programs have made it a little easier to learn more about your firm and guarantee that you are making the greatest decisions for your organization, supported by solid facts.

The significant emphasis on KPIs, data dashboards, and performance

measurements already demonstrates the necessity of taking all of the data collected by the organization and then measuring and monitoring it. Some of the best quantitative information that a business may already be measuring right now and that they may put to good use after the analysis is the firm's market share, the expenses of each department, the revenue collected by quarter, and even the units or products that the company sells.

The next advantage of using this type of data visualization is that it can assist us in telling a tale with a lot of meaning behind it. When we look at some of the work done by the mainstream media, data visualizations and other informational visuals have become crucial tools. Data journalism is a growing industry, and many journalists rely on high-quality visualization tools to tell stories about what's going on in the world around them.

This is something that has gained traction in recent years. Many of the world's most prestigious institutions, like The Washington Post, The New York Times, CNN, and The Economist, have completely embraced the concept of data-driven news.

Marketers can also enter the scene and benefit from the combination of emotional storytelling and quality data that they have at their disposal at all times. A skilled marketer will be able to make data-driven decisions on a daily basis but communicating this information with their clients will necessitate a somewhat different strategy. This strategy must be able to appeal to both the emotional and the rational sides of the brain at the same time. You will discover that using heart and statistics, the graphics from the data may ensure that marketers are able to get their message out there.

Another factor that may influence our decision to work with this type of data visualization is data literacy. The ability to interpret and then understand data visualization has become a need in our modern environment. Because many

of the resources and tools associated with these graphics are widely available in our current society, it is believed that professionals, even those who are not technically savvy, will be able to look at these visuals and obtain the necessary information from them.

As a result, boosting the amount of data literacy discovered around the world will be one of the most important pillars that we will observe when it comes to a lot of data visualization companies and more. To assist individuals in business in making better decisions, it is critical to have the proper information and the correct tools, and data visualization graphs will be the key to accomplishing this. The precedence that comes with data visualization.

We might also seek assistance from Florence Nightingale in this regard. She was best renowned for her work as a nurse during the Crimean War, but she was also a data journalist known for her rose or coxcomb diagrams. These were a form of revolutionary chart that helped here to acquire better hospital conditions, which in turn helped to save many lives of the soldiers that were there.

In addition, Charles Joseph Minard is responsible for one of the most well-known data visualizations. Minard was a civil engineer from France who was well-known for his use of maps to represent numerical data. He is well known for his work on the map depicting Napoleon's Russian campaign of 1812, which depicted the tragic loss of his army while attempting to advance in Moscow, as well as parts of the retreat that ensued.

Why should we use of data visualization?

Before delving deeper into what the matplotlib library can do (so that we have additional ways to depict our data), let's take a look at data visualization and why we should use it in the first place. Some of the several reasons why you might wish to work with data visualization are as follows:

1. It can take the data you already have and make it easier for you to remember and understand it, rather than simply skimming through it and hoping it makes sense.
2. It will provide you with the capacity to identify previously unknown facts, trends, and even outliers in the data that may be valuable.
3. It makes your life easier by allowing you to rapidly and effectively visualize relationships and patterns.
4. It ensures that you can ask better questions and make the greatest judgments for your organization.

What exactly is matplotlib?

With the rest of the information from above in mind, we can see how critical it is to work with data visualization and to have a technique in place to assist us comprehend what is going on in all of the data that we acquired earlier. This assures that we will be ready to go and that the entire analysis will operate as expected.

However, this raises another question that we must address. We need to know what approaches we may employ to assist us in creating some of these charts and graphs, as well as the other visuals that we choose to employ. Your company most likely has a lot of data, and you want to make sure that you are not just selecting the proper type of visualization, but that you are also able to put it all together and create the right visual, which is where matplotlib comes in.

The concept behind matplotlib is that a picture is worth a thousand words. Fortunately, this type of library will not require a thousand words of code to create the graphics that you desire, but it is there to create the visuals and graphics that are required to accompany any information that you have.

Even though you can rest assured that this library will not take a thousand

words to use, you will discover that it is a massive library to look through, and getting the plot to behave the way that you want, as well as choosing the right kind of plot, will be something that you will need to achieve at some points through trial and error. Using one-liners to generate some of the basic graphs that come with this type of coding does not have to be difficult; but, much of the rest of the library can be intimidating at times. And it is for this reason that we will look at what matplotlib is all about and why you should consider learning it so that you can incorporate some of these graphs and visualizations into your data.

First, we must investigate why matplotlib is sometimes regarded as perplexing. Learning how to deal with this type of library can be hard at first, but this isn't to suggest that the matplotlib documentation is inadequate; there is a wealth of information available. However, there are a few obstacles that programmers may face, and some of these are as follows:

1. *The library that you will be able to use will be quite large. In fact, it will comprise approximately 70,000 lines of code, and it is constantly evolving, so this figure is likely to grow over time.*
2. *Matplotlib will be the home of more than one sort of interface, or method of constructing a figure, and it will be able to connect with a wide range of backends. The backend will handle the entire process that occurs when the chart is rendered, not simply the structure. This can cause some issues along the way.*
3. *While this library will be quite extensive, it will be a part of NumPy and SciPy, so you should make sure you understand how to use these languages ahead of time to make things easier.*

To see how this one can operate, we need to know a little bit about its past. John D. Hunter, a neurobiologist, started working on this collection in 2003. He was initially inspired to emulate the commands found in the Mathworks-

supplied MATLAB software.

Hunter died in 2012, and the matplotlib library is now a collaborative effort built and maintained by a slew of others.

One of the important advantages of MATLAB is that it has a global style. The Python notion of importing is useful at times, but it will not be utilized much in MATLAB, and most of the functions that we use with this will be available to the user when they are on the top level.

Knowing that matplotlib has its origins in the MATLAB process can assist to explain why pylab exists in the first place. Pylab is a module that exists within our matplotlib library and was designed to mimic some of the global style that we can find in MATLAB. It exists solely to assist in bringing together several classes and even functions from matplotlib and NumPy to form a namespace. This will be useful for those who want to switch from the previous MATLAB without having to import the statements.

The most significant issue that did emerge here is something that some Python users may have seen in the past. Using the form pylab import in a session or a script was possible, but it was widely regarded as bad practice. In some of the lessons it has published, Matplotlib does not advise against doing this at all. Internally, there are many potentially conflicting imports that are used and masked in the short-come source, and as a result, the matplotlib library has abandoned some of the convenience that comes with this model and recommends to all users that they do not work with pylab, bringing them more in line with some of the key parts of the Python language, so that these can work better together.

The matplotlib object hierarchy is another aspect that we must consider. If you've gone through any kind of beginner's lesson on this library, you've probably used plt.plot([1, 2, 3]) or something similar. This will be an

excellent one to utilize on occasion, but keep in mind that this one-liner will hide the fact that the plot will be a hierarchy with Python objects nested and hidden inside. In this situation, the hierarchy will imply that there will be a structure in the objects that is similar to a trial that is hidden with each of the plots that we have.

A figure object will be the container for this type of image on the outside. Instead, we'll observe that there are numerous Axes objects that we can then use. The name of Axes can be a source of misunderstanding for us. This can result in what we perceive of as an individual graph or plot, rather than the axis that we are accustomed to seeing on a chart or graph.

Consider the Figure function that we are utilizing as a box-like container that will hold at least one, but frequently more than one, Axes or real plots. There will be a hierarchy of smaller objects behind the Axes that we see, such as text boxes, legends, individual lines, and tick marks, to mention a few, and practically all of the parts that appear in this type of chart may be changed by a Python object on its own. This will cover even the smallest details, such as labels and ticks.

The stateless and stateful approaches to matplotlib are the next things we need to look at. For the time being, we will look at the stateful interfaces, which include the state machine and the state-based, as well as the stateless, which are the object-oriented interfaces.

Almost all of the functions available in pyplot, including plt.plot(), will either refer to the existing current Figure as well as the current Axes that you worked with, or will assist you in creating a new one if none exist.

Those who have spent a significant amount of time using MATLAB before switching over may prefer to phrase the differences above as something more along the lines of plt.plot() is a state machine interface that will implicitly

follow the tracks of the current figures. This may not make much sense unless you have spent a significant amount of time working on programming, but it signifies the following in more common terms:

1. The standard interface that you can use will be called up with plt.plot) and other top-level functions in pyplot. Remember that there will only be one Figure or Axes that you are attempting to manipulate at any given time, and you do not need to go through and explicitly refer to it to get things done.
2. Directly modifying the underlying objects will be regarded an object-oriented approach. It is common to do this by using the calling methods with an Axes object, which can represent the plot on its own.

```
def plot(*args, **kwargs):
  """An abridged version of plt.plot()."""
  ax = plt.gca()
  return ax.plot(*args, **kwargs)
def gca(**kwargs):
  """Get the current Axes of the current Figure."""
  return plt.gcf().gca(**kwargs)
```

We can boil it all down and see how it works in just a few lines of code to help us obtain a bit more information about how this plt.pot() function is going to work because we have already spent some time talking about it in this chapter. These codes will be as follows:

Chapter Two
Neural Networks

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep neural network that has proven to be particularly effective in a variety of computer science applications such as object recognition, object categorization, and computer vision. ConvNets have been used for many years to differentiate between faces, recognize objects, and power vision in self-driving cars and robotics.

A ConvNet can recognize and provide captions for a large number of visual scenes. ConvNets may also recognize everyday items, animals, and humans. Convolutional Neural Networks have recently been employed successfully in natural language processing tasks such as sentence classification.

Convolutional Neural Networks are thus one of the most essential tools for machine learning and deep learning problems. LeNet was the first Convolutional Neural Network to be introduced, and it played a vital role in propelling the broader area of deep learning. Yann LeCun proposed the very first Convolutional Neural Network in 1988. It was mostly used to solve character recognition challenges like reading digits and codes.

Convolutional Neural Networks, which are widely utilized in computer science nowadays, are extremely similar to the first Convolutional Neural Network developed in 1988.

LeNet, like today's Convolutional Neural Networks, was employed for a wide range of character recognition applications. The conventional Convolutional Neural Networks that we use today, like LeNet, have four major operations: convolution, ReLU non-linearity activation functions, sub-sampling or pooling, and classification of their fully-connected layers.

These operations are the foundational steps in the construction of any Convolutional Neural Network. To go on to dealing with Convolutional Neural Networks in Python, we must first delve further into these four fundamental functions to gain a better understanding of the idea underlying Convolutional Neural Networks.

Every image, as you are aware, may be easily represented as a matrix comprising several values. We will use the usual term channel to refer to a specific component of images. An image captured by a normal camera typically contains three channels: blue, red, and green. Consider these images to be three two-dimensional matrices layered on top of each other. Each of these matrices also has unique pixel values ranging from 0 to 255.

A grayscale image, on the other hand, only has one channel because there are no colors visible, simply black and white. In this case, we'll be looking at

grayscale images, thus the example we're looking at is just a single-2D matrix that represents a grayscale image. Each pixel in the matrix must have a value between 0 and 255. In this situation, 0 denotes a black color while 255 denotes a white color.

Convolutional Neural Networks: How Do They Work?

Convolutional Neural Network structures are commonly employed to solve deep learning challenges. Because of their structure, Convolutional Neural Networks are utilized for object recognition, object segmentation, detection, and computer vision, as previously stated. CNNs learn directly from picture input, eliminating the requirement for manual feature extraction, which is prevalent in traditional deep neural networks.

The use of CNNs has grown in popularity due to three major considerations. The first is the structure of CNNs, which eliminates the requirement for manual data extraction because Convolutional Neural Networks learn all data features directly. The second reason for CNNs' growing popularity is that they generate fantastic, cutting-edge object identification results. The third argument is that CNNs can be easily retained for many new object recognition tasks to assist in the development of further deep neural networks.

A CNN can include hundreds of layers, each of which learns to detect many different aspects of picture input automatically. Furthermore, filters are usually applied to each training image at varying resolutions, thus the output of each convolved image is utilized as the input to the next convolutional layer.

The filters can also begin with extremely simple picture attributes such as edges and brightness, and as the convolutional layers proceed, they can typically enhance the complexity of those image features that describe the

object. As a result, filters are often applied to every training image at various resolutions, as the output of each convolved image serves as the input to the following convolutional layer.

Convolutional Neural Networks can be trained on images ranging from hundreds to thousands to millions. When working with big volumes of image data and particularly complicated network topologies, GPUs should be used because they can greatly reduce the processing time required for training a neural network model.

Once your Convolutional Neural Network model has been trained, you may utilize it in real-time applications such as object recognition, pedestrian detection in ADAS (Advanced Driver Assistance Systems), and many more.

The output layer is the final fully-connected layer in typical deep neural networks, and it represents the overall class score in every classification setting.

Because of these characteristics, typical deep neural nets are incapable of scaling to entire images. For example, in CIFAR-10, all images are 32x32x3. This means that all CIFAR-10 images have three color channels and are 32 inches wide and 32 inches tall. This suggests that the weights in a single fully-connected neural network in a first regular neural net would be 32x32x3 or 3071. This is a more difficult number to manage because those fully-connected structures are incapable of scaling to larger images.

Furthermore, you would like to have more identical neurons so that you can quickly put up more parameters. In the case of computer vision and other related problems, however, using fully-connected neurons is inefficient because your parameters will quickly lead to over-fitting of your model. As a result, Convolutional Neural Networks use the fact that their inputs are images to solve these types of deep learning challenges.

Convolutional Neural Networks limit visual design in a much more rational way due to their structure. Unlike a traditional deep neural network, the layers of the Convolutional Neural Network are made up of neurons that are arranged in three dimensions: depth, height, and breadth. For example, the CIFAR-10 input images form part of the input volume of all layers in a deep neural network, which has the size 32x32x3.

Instead of all layers being fully connected like in typical deep neural networks, the neurons in these levels can be connected to only a tiny part of the layer preceding it. Furthermore, the output of the final layers for CIFAR-10 would have dimensions of 1x1x10 because the Convolutional Neural Networks architecture would have compressed the complete image into a vector of class score arranged only along the depth dimension at the conclusion of the design.

To recapitulate, a ConvNet, unlike traditional three-layer deep neural networks, composes all of its neurons in only three dimensions. Furthermore, each layer in the Convolutional Neural Network turns the 3D input volume into a 3D output volume with varied neuron activations.

A Convolutional Neural Network is made up of layers that all have a basic API and produce a 3D output volume with a differentiable function that may or may not include neural network parameters.

A Convolutional Neural Network is made up of subsamples and convolutional layers, which are sometimes followed by fully-connected or dense layers. As you may be aware, the input of a Convolutional Neural Network is a n x n x r picture, where n denotes the height and width of the input image and r represents the total number of channels present. Convolutional Neural Networks may also have k filters referred to as kernels. When kernels are present, their q is determined, which can be the same as the

number of channels.

Each Convolutional Neural Network map is subsampled using max or mean pooling over p x p in a contiguous area, where p typically varies from 2 for small images to more than 5 for bigger images. Every feature map is subjected to sigmoidal non-linearity and additive bias, either after or before the subsampling layer. Following these convolutional neural layers, there may be numerous fully-connected layers, the structure of which is the same as that of ordinary multilayer neural networks.

Stride and Padding

Second, after setting the depth, you must additionally indicate the stride with which you slide the filter. When you have a stride of one, you can only move one pixel at a time. When you have a stride of two, you can move two pixels at a time, but this results in lower spatial volumes of output. The stride value is one by default. However, if you desire less overlap between your receptive fields, you can make larger strides, but as previously said, this will result in smaller feature maps because you are skipping over picture spots.

If you utilize larger strides but wish to keep the same dimensionality, you must use padding, which surrounds your input with zeros. You can pad with either the values on the edge or with zeros. Once you've determined the dimensionality of your feature map that corresponds to your input, you may proceed to adding pooling layers, which are typically employed in Convolutional Neural Networks to retain the size of your feature maps.

If no padding is used, your feature maps will shrink with each layer. When you wish to pad your input volume with zeros all around the border, zero-padding comes in handy. This is known as zero-padding, and it is a hyperparameter. You can regulate the size of your output volumes by

utilizing zero-padding. The spatial size of your output volume may be simply calculated as a straightforward function of your input volume size, the convolution layers receptive field size, the stride you employed, and the amount of zero- padding you used in your Convolutional Neural Network border.

For example, if you have a 7x7 input and apply the formula to a 3x3 filter with stride 1 and pad 0, you will receive a 5x5 output. If you choose stride two, you will obtain a 3x3 output volume, and so on, using the formula where W represents the size of your input volume, F represents the receptive field size of your convolutional neural layers, S represents the stride utilized, and P indicates the amount of zero-padding you employed.

(W-F +2P)/S+1

You can quickly calculate how many neurons can fit in your Convolutional Neural Network using this formula. When possible, try to use zero- padding. For example, if your input and output dimensions are both five, you can use a zero-padding of one to generate three receptive fields. If you do not employ zero-padding in situations like these, your output volume will have a spatial dimension of 3, because 3 is the number of neurons that can fit inside your original input.

Mutual constraints are prominent in spatial arrangement hypermeters. For example, applying stride to an input size of 10 with no zero-padding and a filter size of three is impossible. As a result, your hyperparameter set will be invalid, and your Convolutional Neural Networks library will either throw an exception or zero pad the rest fully to make it fit.

Fortunately, appropriately sizing the convolutional layers so that all dimensions incorporate zero-padding can make any job easier.

Parameter Sharing

In your convolutional layers, you can use parameter sharing strategies to completely regulate the amount of parameters used. If you designate a single two-dimensional depth slice as your depth slice, you can force the neurons in each depth slice to utilize the same bias and weights. Using parameters sharing approaches, you will obtain a one-of-a-kind set of weights, one for each depth slice. As a result, you can considerably reduce the number of parameters in your ConvNet's first layer. By completing this step, all neurons in your ConvNet's depth slices will use the same settings.

In other words, every neuron in the volume will automatically compute the gradient for all of its weights during backpropagation.

However, because these computed gradients stack up across each depth slice, you only need to update a single collection of weights each depth slice. As a result, all neurons within a single depth slice will use the same weight vector. As a result, when you forward the convolutional layer pass in each depth slice, it is computed as a convolution of all neurons' weights alongside the input volume. This is why the collection of weights we acquire is referred to as a kernel or a filter, which is convolved with your input.

However, there are a few circumstances where this parameter sharing assumption is invalid. This is frequently the case with a large number of input photos to a convolutional layer with a specific centered structure, where you must learn different features based on the location of your image.

For example, if you have an input of numerous faces that have been centered in your image, you may anticipate to acquire various hair-specific or eye-specific traits that could be easily learned at many spatial places. When this occurs, it is fairly usual to simply loosen the parameter sharing strategy and employ a locally connected layer.

Matrix Multiplication

Those dot products between the local regions of the input and between the filters are usually performed by the convolution operation. In these instances, a typical convolutional layer implementation strategy is to take full use of this feature and design the specific forward pass of the primary convolutional layer as one huge matrix multiply.

Matrix multiplication is implemented when the local portions of an input image are entirely stretched out into separate columns during the im2col procedure. For example, if you have an input of size 227x227x3 and convolve it with a filter of size 11x11x3 at a stride of 4, you must stretch every block of pixels in the input into a column vector of size 363.

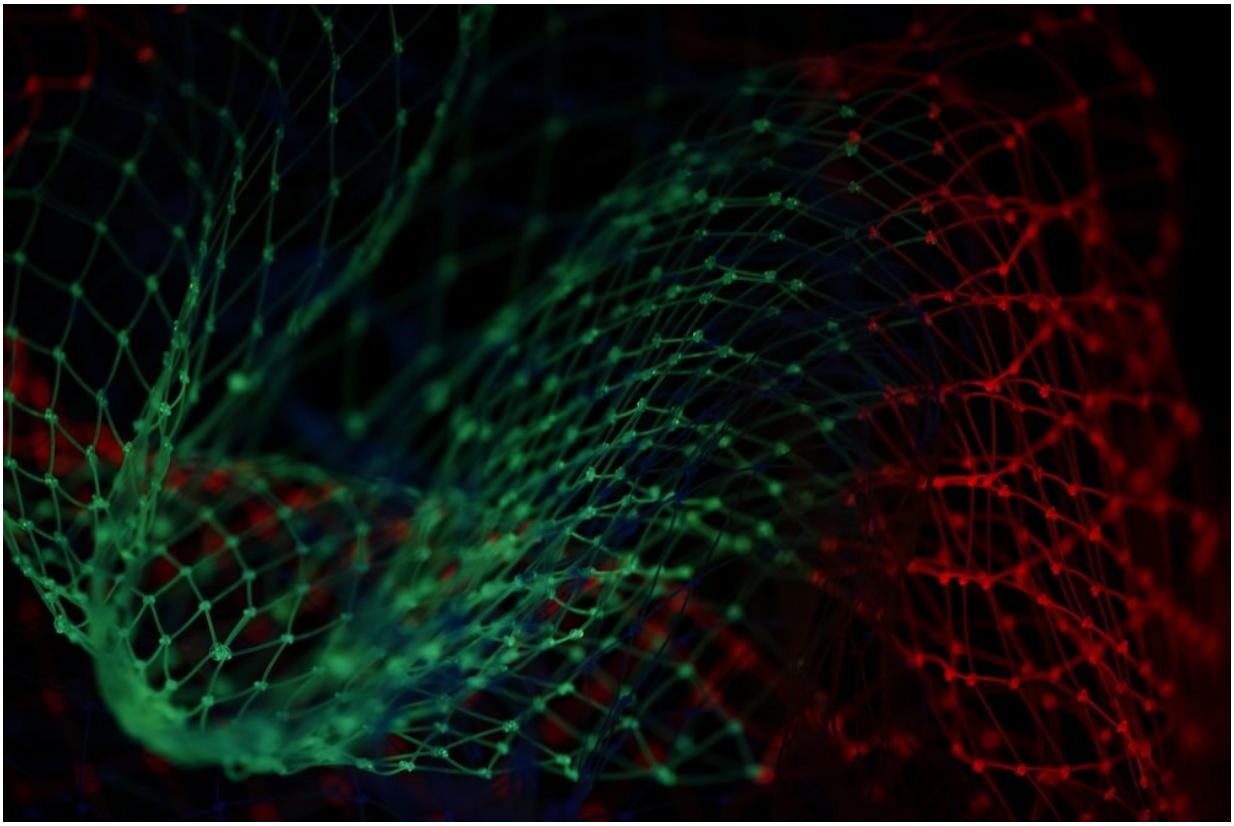
When you run this process in your input stride of 4, you obtain 55 locations as well as weight and height, which leads to an output matrix of x columns, where each column is a maximally stretched out receptive field and you get 3025 fields in total.

Each number in your input volume can be repeated in several columns. Also, keep in mind that the weights of the convolutional layers are similarly spread out into certain rows. For example, if you have 95 filters with dimensions of 11x11x3, you will receive a matrix with w rows with dimensions of 96x363.

In terms of matrix multiplications, the output of your convolution will be equivalent to conducting one massive matrix multiply that evaluates the dot products between every receptive field and between every filter, resulting in the output of your dot production of every filter at every position. Once you get your result, you must reshape it to the correct output dimension, which in this example is 55x55x96. This is a fantastic strategy, but it has a drawback. The biggest disadvantage is that it consumes a lot of memory because the values in your input volume are reproduced numerous times. The fundamental advantage of matrix multiplications, however, is that various

implementations can improve your model. Furthermore, while conducting pooling operations, this im2col can be reused several times.

Chapter Three
Decision Trees



Decision trees are formed in the same way as support vector machines, and they are a type of supervised machine learning technique that can solve both regression and classification problems. They are effective for dealing with large amounts of data.

You must go beyond the fundamentals in order to process huge and complex datasets. Furthermore, decision trees are utilized in the construction of random forests, which are often regarded as the most powerful learning method. Because of their popularity and efficiency, we will primarily focus

on decision trees in this chapter.

An Overview on Decision Trees

Decision trees are fundamentally a tool that supports a decision that will influence all subsequent decisions. This means that everything from expected outcomes to consequences and resource utilization will be altered in some way. Take note that decision trees are typically represented in a graph, which can be thought of as a type of chart in which the training tests appear as nodes. For example, the node may be a coin flip with two possible outcomes. Furthermore, branches sprout to represent the results individually, and they have leaves that serve as class labels. You can see why this method is referred to as a decision tree now. The structure is reminiscent of a tree. Random forests are, as you might expect, exactly what they sound like. They are collections of decision trees, but that's all there is to it.

Decision trees are one of the most powerful supervised learning methods, particularly for beginners. Unlike more complex algorithms, they are relatively simple to implement and have a lot to offer. Any common data science task can be performed by a decision tree, and the results obtained at the end of the training process are highly accurate. With that in mind, let's look at a few more benefits and drawbacks to gain a better understanding of their use and implementation.

Let's start with the good news:

1. Decision trees are basic in concept and thus straightforward to execute, even if you have no formal education in data science or machine learning. This algorithm's notion can be summarized with a formula that follows a popular style of programming statement: If this, then that, otherwise that. Furthermore, the results will be very simple to interpret, thanks to the graphic depiction.

2. A decision tree is one of the most efficient approaches for examining and deciding the most significant factors, as well as discovering the relationship between them. You may also easily create new features to improve measurements and forecasts. Don't forget that data exploration is one of the most crucial stages of working with data, especially when there are a lot of variables to consider. To prevent a time-consuming procedure, you must be able to discover the most useful ones, and decision trees excel at this.
3. Another advantage of using decision trees is that they are fantastic at removing outliers from your data. Remember that outliers are noise that lowers the accuracy of your forecasts. Furthermore, noise has little effect on decision trees. Outliers have such a minor impact on this method in many circumstances that you can choose to ignore them if you don't need to maximize the accuracy ratings.

Finally, decision trees are capable of working with both numerical and categorical information. Keep in mind that several of the algorithms we've already discussed can only be employed with one sort of data or the other. Decision trees, on the other hand, have been shown to be adaptable and capable of handling a far broader range of tasks.

As you can see, decision trees are extremely strong, diverse, and simple to create, so why would we use anything else? As is customary, nothing is flawless, so let's look at the drawbacks of using this type of algorithm:

1. Overfitting is a major issue that arises during the implementation of a decision tree. Take notice that this technique has a tendency to generate very sophisticated decision trees that, due to their complexity, will have difficulty generalizing data. This is referred to as overfitting, and it occurs when implementing other learning algorithms as well, though not to the same extent. Fortunately, this

does not exclude you from employing decision trees. To decrease the impact of overfitting, all you need to do is invest some time in implementing certain parameter limits.

2. Continuous variables might cause problems for decision trees. When dealing with continuous numerical variables, decision trees lose some information. This issue arises when the variables are classified. A continuous variable is a value that is set to be within a range of numbers if you are unfamiliar with them. If anyone between the ages of 18 and 26 are regarded to be of student age, this numerical range becomes a continuous variable because it can hold any value between the defined minimum and maximum.

1. While some downsides may necessitate greater work in the implementation of decision trees, the benefits clearly exceed them.

Classification and Regression Trees

We previously described how decision trees are utilized for both regression and classification applications. However, this does not imply that you use the same decision trees in both circumstances. Classification and regression trees must be separated from decision trees. They deal with separate problems, although they are related in some ways because they are both forms of decision trees.

Keep in mind that classification decision trees are used when the dependent variable is categorical. A regression tree, on the other hand, is only used when the dependent variable is continuous. Furthermore, in the case of a classification tree, the training data result is the mode of the total relevant observations. This means that all observations that we cannot specify will be predicted based on this value, which reflects the most often identified observation.

Regression trees, on the other hand, operate in a slightly different manner. The value obtained during the training step is not the mode value, but rather the mean of all observations. In this manner, unidentified observations are announced with the mean value derived from known data.

Both forms of decision trees go through a binary split, but from top to bottom. This means that observations in one location will generate two branches, which will then be partitioned inside the predictor space. This is also referred to as a greedy method since the learning algorithm seeks the most relevant variable in the split while neglecting future splits that could contribute to the creation of a more powerful and accurate decision tree.

As you can see, the two have some differences as well as commonalities. What you should take away from all of this is that the splitting has the most impact on the decision tree implementation's accuracy scores. Regardless of the form of tree, decision tree nodes are separated into subnodes. This tree split is carried out to provide a more consistent set of nodes.

Now that you've grasped the principles of decision trees, let's delve a little more into the issue of overfitting.

Overfitting Problem

Overfitting is one of the most common issues when working with decision trees, and it can have a significant impact on the results. If no constraints are imposed, decision trees can get a 100 percent accuracy score for the training set. The key disadvantage here is that overfitting occurs when the algorithm attempts to decrease training errors while instead increasing testing errors. Regardless of the score, this imbalance leads to poor prediction accuracy in the end outcome. What causes this to happen? In this situation, the decision trees generate a lot of branches, which causes overfitting. To tackle this problem, you must limit how much the decision tree can grow and how many

branches it can produce. Furthermore, you may prune the tree to keep it under control, just as you would with a real tree to ensure it produces a plentiful supply of fruit.

1. The first thing you may do is alter the option min samples split to define how many observations a node needs to execute the splitting. You can declare anything with a sample count ranging from one to a maximum of ten. Just keep in mind that increasing the amount will prevent the training model from determining connections that are particularly common to a specific decision tree. In other words, greater values can be used to constrain the decision tree.
2. Min samples leaf: This is the parameter you need to adjust to determine how many observations a node, or leaf, requires. The overfitting control mechanism functions in the same way that the samples split parameter does.
3. Max features: Modify this option to determine the features that are chosen at random. These are the characteristics that are used to perform the best split. Calculate the square root of the total characteristics to find the most efficient value. Just bear in mind that in this case, the greater number tends to exacerbate the overfitting problem we're attempting to address. As a result, you should experiment with the value you choose. Furthermore, not every case is the same. A greater value will sometimes work without causing overfitting.
4. Max depth: Finally, we have the depth parameter, which comprises of the decision tree's depth value. However, in order to restrict the overfitting problem, we are only interested in the greatest depth value. Keep in mind that a high value corresponds to a high number of splits, and thus a significant amount of information. You may

influence how the training model learns the connections in a sample by adjusting this parameter.

1. Changing these parameters is merely one part of getting control over our decision trees in order to reduce overfitting and improve performance and accuracy. Following the application of these constraints, the trees must be pruned.

Pruning

This strategy may appear to be too absurd to be true; nonetheless, it is a realistic machine learning idea that is utilized to optimize your decision tree by practically eliminating overfitting. Pruning, like real tree pruning, reduces the size of the trees in order to focus resources on giving extremely accurate outcomes. However, you should keep in mind that the pruned segments are not chosen at random, which is a good thing. Sections that are removed are those that do not aid in classification and do not result in any performance improvements. Less complex decision trees result in a more optimal model.

Visualize the following situation to better comprehend the difference between an unmodified decision tree and one that has been trimmed and optimized. Assume there is a roadway with one lane for automobiles traveling at 80 mph and another lane for slower vehicles traveling at 50 mph. Assume you're driving down the highway in a red car and you're faced with a decision. You have the option of going to the fast lane to pass a slow-moving automobile; however, this means that you will have a truck in front of you who is unable to accomplish the high speed he should in the left lane, and therefore you will be stranded in that lane. In this instance, the cars in the other lane are gradually overtaking you since the truck is unable to keep up. The other option is to stay in your lane and not attempt a pass. The best option here is one that permits you to travel a greater distance in a shorter amount of time. As a result, if you continue in the slow lane until you gradually pass the truck

that is obstructing the fast lane, you will ultimately be able to switch to that lane and pass all of the other vehicles. As you can see, the second choice appears to be the slowest at the time of evaluation; but, in the long term, it is the most efficient. The decision trees are all the same. When you set limitations for your trees, they will not become greedy and switch you to the left lane, where you would be caught behind a truck. However, pruning the decision tree will help you to explore your surroundings in greater detail and foresee a greater number of possibilities you have to make a better choice.

As you can see, performing the trimming procedure has several significant advantages that should not be overlooked. However, putting this strategy into action necessitates a number of actions and conditions. A decision tree, for example, must have a high depth value in order to be acceptable for pruning. Furthermore, in order to avoid bad outcomes, the process must begin at the bottom. This problem must be avoided because if a negative node split occurs at the bottom and another at the top, we will end up with a decision tree that stops when the first division occurs. If the tree is pruned, it will continue to grow, resulting in greater profits.

When all you have is theory, it might be tough to visualize decision trees, so let's start with a step-by-step implementation to see them in action.

Decision Tree Implementation

Making a decision tree begins with the root node. The first step is to choose one of the data attributes and create a logical test around it. Once you have a set of results, you may branch out and write another series of tests to build the subnode. Once we have at least one subnode, we may use a recursive splitting method to assess whether or not we have clean decision tree leaves. Keep in mind that the purity level is defined by the number of cases that arise from a single class. At this point, you can begin pruning the tree to remove

everything that does not improve the classification stage's accuracy. You will also need to examine each and every split that is conducted depending on each attribute. This stage must be completed in order to decide which characteristic, as well as split, is the most ideal.

But that's enough theory for now. At this stage, all you should be concerned with is the core concept of decision trees and how to make them efficient. Once you believe you have a firm handle on the fundamentals, you must begin the implementation process.

In the following example, we will once again rely on the Iris dataset and the Scikit-learn module to provide the data.

K-means Clustering

Unsupervised learning methods, as previously stated, are appropriate for working with unlabeled data. To be more specific, one of the best, if not the best, strategy is to utilize a form of clustering algorithm. The cluster analysis is the basic principle underlying this approach, which entails reducing data observations to clusters, or subdivisions of data, where each cluster has information that is similar to that of a preset feature. Clustering entails a number of strategies that all work toward the same aim because they are all concerned with developing a range of beliefs about the data structure.

K-means clustering is one of the most widely used unsupervised learning algorithms and clustering approaches. The idea behind this notion is to create data clusters based on the similarity of the values. The first step is to calculate k, which is represented by the total number of clusters we create. These clusters are made up of k-many points, each of which has the average value for the entire cluster. Furthermore, the values are assigned based on the closest average value. Remember that clusters have a core, which is defined as an average value that pushes the other averages aside, causing them to

change. After a sufficient number of rounds, the core value will move to a lower performance metric. We have the solutions when we reach this step because there are no observations available to be designated.

It's okay if all of this theory has left you perplexed. You'll see that this strategy is much simpler than it appears. Let's have a look at how it's done in practice. We'll utilize the UCI handwritten digits dataset in this example. It's free, and you don't need to download it if you're using Scikit-learn in conjunction with the book. With that said, here is the code:

```
from time import timeimport numpy as np
import matplotlib.pyplot as plt
from sklearn import datasetsnp.random.seed()
digits = datasets.load_digits()
data = scale(digits.data)
n_samples, n_features = data.shape
n_digits =len(np.unique(digits.target))
labels = digits.target
sample_size = 300
print("n_digits: %d, \t n_samples %d, \t n_features
%d"
% (n_digits, n_samples, n_features))
print(79 * '_')
print('% 9s' % 'init''time      inertia  homo   compl
v-meas
ARI AMI silhouette')
def bench_k_means(estimator, name, data):
estimator.fit(data)
```

```
% (name, (time() - t0), estimator.inertia_,
metrics.homogeneity_score(labels,
estimator.labels_),
metrics.completeness_score(labels,
estimator.labels_),
metrics.v_measure_score(labels,
estimator.labels_),
metrics.adjusted_rand_score(labels,
estimator.labels_),
metrics.silhouette_score(data,
estimator.labels_,
metric='euclidean'
```

If you examine the code line by line, you will realize that the implementation is rather basic, logical, and simple to grasp. In fact, it is similar in several ways to other strategies we have used previously. However, there is one significant difference to note, and that is the performance measurements we use to appropriately analyze the data.

First, we have a score for homogeneity. This measure can have a value ranging from 0 to 1. It is mostly interested in clusters that have only one class system. The concept is that if we have a score that is near to one, then the cluster is primarily made up of samples from a single class. If, on the other hand, the score is near to zero, we have attained a low level of homogeneity.

The completeness score comes next. This metric supplements the measure of homogeneity. Its objective is to enlighten us about how the measures become a part of a given class. The two ratings allow us to conclude that we either performed excellent clustering or just failed.

The third metric is known as the V-metric or, more colloquially, the V-measure. The harmonic mean of the preceding two scores is used to calculate this score. The V-metric validates the validity by assigning a zero to one value to the homogeneity and completeness score.

The adjusted Rand index statistic comes next. This is a score that is used to verify the labeling's similarity. The Rand index simply determines the relationship between the distribution sets by using a number between zero and one.

Finally, the silhouette metric is used to determine whether the performance of the clustering is adequate in the absence of labeled data. The measurement ranges from a negative to a positive number and evaluates whether or not the clusters are well-structured. If the value is negative, we have a problem with

faulty clusters. To ensure dense clusters, we must acquire a score near to a positive one. Remember that in this scenario, we could also have a score close to zero. In this situation, the silhouette measurement indicates that we have clusters that overlap.

Now that you understand the measuring system, we need to take one more step to ensure that the results are accurate. We may use the bench k means function to validate the clustering scores as follows:

```
bench_k_means(KMeans(init='k-means++', n_clusters=n_digits, n_
init=10),

name="k-means++", data=data) print(79 * '_')
```

Now let's see what conclusion we can draw from the scores. Here's how your results should look:

```
n_digits: 10,        n_samples 1797,        n_features 64

init                 time                 inertia      homo
compl

k-means++ 0.25s          69517                  0.596        0.643

init                 v-meas      ARI              AMI
silhouette

k-means++          0.619        0.465        0.592        0.123
```

As you can see, we have fairly good results with a basic k-means implementation; nonetheless, there is much room for improvement. Although clustering is sufficient, we could improve the scores by using other supervised or unsupervised learning approaches. In this circumstance, for example, you might think about employing the principal component analysis approach as well. Another possibility is to use various dimensionality reduction algorithms. These results, however, will enough to understand how

to build the K-means clustering algorithm. However, keep in mind that in the real world of data science, you will frequently combine various methods and techniques. You will almost never be able to achieve useful results with just one algorithm, especially if you are working with raw datasets rather than practice ones.

In this chapter, we learned about unsupervised learning algorithms, specifically K-means clustering. The goal of this part was to demonstrate a technique that can be used to more complex datasets. Clustering algorithms are a data science staple and are frequently employed, particularly in conjunction with other algorithms and learning techniques. Furthermore, as you will discover later, clustering techniques, particularly K-means clustering, are extremely effective when working with Big Data.

Chapter Four
Applications of Big Data Analysis

Big Data and Big Data Analytics applications serve both small and large businesses in a variety of industrial fields. In this chapter, we will delve deeper into such applications.

eCommerce

Customers and potential customers for any organization are among the over 2.6 billion active social media users. The race is on to develop more successful marketing and social media tactics powered by machine learning, with the goal of providing an improved customer experience and converting prospective consumers into raving fans. Sifting through and analyzing vast

amounts of data has not only become viable, but also simple. Artificial intelligence marketing solutions have helped to bridge the gap between execution and large data analysis. Artificial Intelligence (AI) marketing is a technique in which you use artificial intelligence technology such as machine learning on available customer data to predict consumer requirements and expectations while dramatically improving the customer's trip. Marketers may improve campaign performance and ROI with little to no extra effort thanks to big data insights supplied by artificial intelligence marketing solutions. The following are the important components that contribute to the effectiveness of AI marketing:

- Big data - The ability of a marketing firm to aggregate and segment a massive amount of data with minimal manual labour is referred to as Big Data. The marketer can then use the proper media to guarantee that the appropriate message is delivered to the target audience at the appropriate moment.
- Machine learning platforms enable marketers to spot trends or recurrent occurrences and gain useful insights and answers, allowing them to grasp the core cause and probability of recurring events.
- Platform that is intuitive – AI marketing relies on applications that are lightning fast and simple to use. Artificial intelligence technology can sense emotions and communicate like a human, allowing AI-powered systems to understand open form content such as email responses and social media.

Predictive Analysis

All solutions based on artificial intelligence technology are capable of extracting information from data assets in order to forecast future trends. AI technology has enabled the modeling of trends that could previously only be determined retroactively. These predictive analysis algorithms can be utilized

to make sound decisions and examine customer purchasing behavior. The algorithm can accurately predict when a consumer is more likely to make a new purchase or reorder an existing one. Marketing firms can now reverse engineer customers' experiences and activities to develop more profitable marketing tactics. FedEx and Sprint, for example, use predictive analytics to identify consumers who are at risk of switching to a competitor.

Smart searches

Only a decade ago, if you type in "women's flip flops" on Nike.com, the probability of you finding what you were looking for would be next to zero. However, today's search engines are not only accurate but also much faster. This upgrade has largely been brought on by innovations like "semantic search" and "natural language processing" that enable search engines to identify links between products and provide relevant search results, recommend similar items, and auto-correct typing errors. The artificial intelligence technology and big data solutions can rapidly analyze user search patterns and identify key areas that the marketing companies should focus on.Only a decade ago, if you typed "women's flip flops" into Nike.com, the chances of finding what you were looking for were almost none. Today's search engines, on the other hand, are not only more accurate, but also more faster. This advancement has been largely driven by advancements such as "semantic search" and "natural language processing," which allow search engines to recognize links between products and deliver relevant search results, propose related things, and auto-correct typing errors. Artificial intelligence technologies and big data solutions can quickly assess user search trends and highlight crucial areas where marketing firms should focus their efforts.

Google launched the first Artificial Intelligence-based search algorithm, "RankBrain," in 2015. Following in Google's footsteps, other major e-

commerce websites, such as Amazon, have incorporated big data analysis and artificial intelligence into their search engines to provide smart search experiences for their customers, allowing them to find desired products even when they don't know exactly what they're looking for. Smart search solutions such as "Elasticsearch" are available to even modest e-commerce enterprises. Companies that provide data-as-a-service, such as "Indix," enable businesses to train their product search models by learning from larger data sources.

Recommendation Engines

Customers and marketing firms both adore recommendation engines, which have quickly turned into fan favorites. "Apple Music" already knows your music preferences better than your partner, and Amazon always shows you a list of things you might be interested in purchasing. This type of discovery tool, which can filter through millions of potential possibilities and zero in on an individual's wants, is becoming vital for major corporations with massive physical and digital inventory.

Jussi Karlgren, a Swedish computational linguist, investigated the practice of clustering customer activities to anticipate future behaviors in his study titled "Digital bookshelves" in 1998. Similarly, Amazon employed collaborative filtering to generate consumer suggestions. The predictive analysis-based systems' collection and analysis of consumer data, together with individual profile information and demographics, enables the system to continuously learn and adapt based on consumer activities such as likes and dislikes on products in real-time. For example, "Sky" has created a predictive analysis-based model capable of recommending content based on the viewer's mode. The savvy consumer expects such an improved experience not only from their music and on-demand entertainment providers, but also from all other e-commerce companies.

Product Categorization and Pricing

E-commerce enterprises and marketing firms are increasingly incorporating artificial intelligence into their inventory categorization and labeling processes. Marketing firms must deal with bad data just as much, if not more, than they do with well-organized, clean data. This collection of positive and negative samples is used to train predictive analysis-based classification algorithms. Different detailers, for example, may have different descriptions for the same product, such as sneakers, basketball shoes, trainers, or Jordans, but the AI system can recognize that they are all the same things and categorize them properly. If the data set lacks the principal keyword, such as skirts or shirts, the artificial intelligence algorithm can recognize and classify the item or product as skirts or shirts based simply on the surrounding context.

We're all aware with seasonal hotel rate fluctuations, but with the arrival of artificial intelligence, product prices may be tailored to suit demand with a whole new degree of precision. Machine learning algorithms are utilized for dynamic pricing by evaluating consumer data patterns and creating near-accurate forecasts of what they are prepared to pay for that specific product as well as their receptivity to special offers. This enables firms to precisely target their consumers and determine whether or not a discount is required to complete the deal. Dynamic pricing also enables firms to compare their product pricing to market leaders and competitors and alter their prices accordingly in order to close the sale. For example, "AirbnB" has created a dynamic pricing system that gives property owners with "Price Tips" to help them select the best possible listing price for their home. The method considers a number of contributing elements, including geographical location, local events, property photos, property reviews, listing characteristics, and, most crucially, booking timings and market demand. The

system will also monitor the property owner's ultimate decision to follow or disregard the offered 'pricing advice,' as well as the success of the listing, and will then process the results and alter its algorithm accordingly.

Customer Targeting and Segmentation

Marketing firms must target increasingly granular categories in order to reach their customers with a high level of personalization. Machine learning algorithms can be trained against "gold standard" training sets using current customer data to find common characteristics and relevant factors. Data segments could be as simple as location, gender, and age, or as complicated as the buyer's identity and previous behavior. Segmentation is possible using AI Dynamics, which accounts for the fact that customers' actions change all the time, and that people might take on various personalities in different settings.

Marketing firms must target increasingly granular categories in order to reach their customers with a high level of personalisation. Machine learning algorithms can be trained against "gold standard" training sets using current customer data to find common characteristics and relevant factors. Data segments could be as simple as location, gender, and age, or as complicated as the buyer's identity and previous behavior. Segmentation is possible using AI Dynamics, which accounts for the fact that customers' actions change all the time, and that people might take on various personalities in different settings.

Sales and Marketing Forecast

The construction of sales and marketing forecasting models is one of the most easy artificial intelligence applications in marketing. The machine learning algorithms use a large amount of quantifiable data, such as clicks, purchases, email answers, and time spent on webpages, as training materials.

Sisense, Rapidminer, and Birst are three of the market's leading business intelligence and production organizations. Marketing firms are always improving their marketing efforts, and they can forecast the success of their marketing activities or email campaigns using AI and machine learning. Artificial intelligence technology can predict short and long-term sales performance and forecast sales outcomes by analyzing prior sales data, economic trends, and industrywide comparisons. The sales forecasting model assists in estimating product demand and assisting businesses in managing production to maximize sales.

Programmatic Advertisement Targeting

Bidding on and targeting program-based advertisements has grown substantially more efficient since the development of artificial intelligence technologies. The automated process of purchasing and selling ad inventory to an exchange that connects advertisers and publishers is known as programmatic advertising. Artificial intelligence technology is utilized to enable real-time bidding for inventory through social media channels, mobile devices, and television. This also relates to predictive analytics and the ability to model data that was previously only determined retrospectively. Artificial intelligence can help determine the ideal time of day to deliver a specific ad, the likelihood of an ad converting into sales, the user's receptiveness, and the likelihood of engagement with the ad.

Programmatic organizations can collect and analyze data and behaviors from visiting customers in order to optimize real-time campaigns and target the audience more precisely. The usage of "demand-side platforms" (to assist the process of buying ad inventory on the open market) and "data management platforms" is included in programmatic media buying (to provide the marketing company an ability to reach their target audience). The data management solutions are meant to collect and analyze a large volume of

website "cookie data" in order to empower marketing representatives to make informed decisions about their prospective consumers. Search engine marketing (SEM) advertising, for example, is used by channels such as Facebook, Twitter, and Google. Programmatic advertisements provide a considerable advantage over competitors in terms of efficiently managing a large inventory of website and application viewers. Google and Facebook are the gold standard for efficient and effective advertising, with a user-friendly platform that allows non-technical marketing organizations to launch, operate, and measure their initiatives and campaigns online.

Visual Search and Image Recognition

Artificial intelligence-based image identification and analysis technology has advanced by leaps and bounds, resulting in eerie visual search features. With the emergence of technologies such as Google Lens and platforms such as Pinterest, consumers may now use the visual search functionality to find results that are aesthetically similar to one another. The visual search function is comparable to typical text-based searches in that it returns results on a related topic. Major shops and marketing firms are increasingly utilizing visual search to provide a more enriched and engaging customer experience. Instead of the consumer's previous behavior or purchases, visual search can be used to improve merchandising and make product recommendations based on the style of the product.

Target and Asos have made significant investments in the development of visual search technologies for their e-commerce websites. Target established a partnership with Pinterest in 2017 that permits the integration of Pinterest's visual search technology, known as "Pinterest lens," into Target's mobile application. As a result, buyers may take photos of stuff they want to buy while out and about and find similar items on Target's e-commerce site. Similarly, Asos' "Asos' Style Match" visual search application allows buyers

to take a photo or upload an image on the Asos website or apps and search their product catalog for similar things. These technologies entice buyers to visit businesses for products they may see in a magazine or while out and about by assisting them in shopping for the appropriate purchase even if they do not know what the product is.

Image recognition has greatly aided marketing firms in gaining an advantage on social media by helping them to find a range of uses for their brand logos and products in order to keep up with visual trends. This phenomenon, also known as "visual social listening," enables businesses to detect and understand where and how customers interact with their brand, logo, and product even when the firm is not referred to directly by name.

Healthcare Industry

Big Data Analysis has resulted in a paradigm shift in healthcare due to the rising availability of healthcare data. The investigation of correlations between patient outcomes and the treatment or prevention approach utilized is the core focus of big data analytics in the healthcare business. Big Data Analysis-driven Artificial Intelligence algorithms for patient diagnoses, treatment protocol generation, drug research, and patient monitoring and care have all been established effectively. The advanced AI approaches can filter through large amounts of clinical data and help unlock clinically important information to aid decision making.

Some medical fields that are seeing an increase in big data analysis-based AI research and applications include:

- Radiology - AI's capacity to analyze imaging results supports the clinician's ability to spot changes in a picture that the human eye can easily miss. An artificial intelligence algorithm developed recently at Stanford University may pinpoint particular spots in the lungs of

pneumonia sufferers.

- Electronic Health Records — The requirement for digital health records in order to improve information dissemination and access necessitates the rapid and correct logging of all health-related data in the systems. Humans are prone to errors and might suffer from cognitive overload and exhaustion. AI has successfully automated this process. At baseline, the use of predictive models on electronic health record data allowed for the prediction of customized treatment response with 70-72 percent accuracy.
- Imaging - Ongoing AI research is assisting clinicians in assessing the outcome of corrective jaw surgery as well as cleft palate therapy to predict facial beauty.

Entertainment Industry

Big Data Analysis, in collaboration with Artificial Intelligence, is increasingly running in the background of entertainment sources ranging from video games to movies, providing us with a richer, more engaging, and realistic experience. Big Data Analysis is being used by entertainment providers such as Netflix and Hulu to provide consumers with customised suggestions based on their prior activity and behavior. Big Data Analysis-based solutions have been used by computer graphics and digital media content creators to improve the speed and efficiency of their production processes. Machine learning algorithms are rapidly being used by film studios in the creation of film trailers and commercials, as well as in pre- and post-production procedures. For example, big data analysis and an artificial intelligence-powered technology called "RivetAI" enable producers to automate and read the processes of movie screenplay breakdown, storyboard, budgeting, scheduling, and shot-list generation. Certain time-consuming operations performed during film post-production, such as synchronization

and clip assembly, can be easily automated using artificial intelligence.

Advertising and Marketing

A machine learning algorithm created as a result of big data analysis can be simply trained using text, still images, and video segments as data sources. It may then extract items and concepts from various sources and make recommendations for effective marketing and advertising solutions. Alibaba, for example, developed a program called "Luban" that can generate banners at the speed of a human designer. Luban generated a hundred and seventeen million banner designs in 2016 for the Chinese online shopping fiesta known as "Singles Day" at a rate of 8000 banner designs per second.

"20th Century Fox" partnered with IBM to create the trailer for their horror film "Morgan" using their AI system "Watson." To learn the proper "moments" or snippets that should be in a video.

Watson was trained to classify and analyze input "moments" from audio-visual and other composition aspects in over a hundred horror movies. Watson used this training to create a six-minute movie trailer in under 24 hours, which would have taken a human professional and weeks to create.

An AI marketing platform may accelerate the marketing process dramatically by utilizing machine learning, computer vision technologies, natural language processing, and predictive analytics. Albert Intelligence Marketing's artificial intelligence-based marketing platform, for example, may produce autonomous campaign management plans, construct unique solutions, and execute audience targeting. The adoption of their AI-based platform resulted in an 183 percent increase in client transaction rate and a 600 percent increase in conversation efficiency, according to the company.

McCann Erickson Japan introduced the "AI-CD ß" artificial intelligence-based creative director in March 2016 as the world's first robotic creative

director. "AI-CD ß" received instruction on selected aspects of various TV shows as well as the winners of the previous ten years of the All Japan Radio and Television CM competition. "AI-CD ß" can extract concepts and themes that meet each client's unique campaign needs by utilizing data mining skills.

User Experience Personalization

On-demand entertainment users' expectations for a rich and engaging individualized user experience are always rising. Netflix, one of the leading on-demand entertainment platforms, has released "Meson," an artificial intelligence-based workflow management and scheduling application comprised of various "machine learning pipelines" capable of creating, training, and validating personalization algorithms to provide personalized recommendations to users. Netflix teamed with the University of Southern California to create "Dynamic Optimizer," a revolutionary Machine Learning method that can compress video for high-quality streaming without sacrificing image quality. By enhancing video fluency and definition, this artificial intelligence solution will address streaming issues in developing countries and among mobile device users.

IBM Watson recently partnered with IRIS.TV to provide a business-to-business solution to media businesses such as CBS, The Hollywood Reporter, and Hearst Digital Media by measuring and optimizing their customers' introduction to their web content. IBM Watson is enhancing IRIS.TV's machine learning algorithms, which can 'learn' from users' search histories and recommend related material. According to reports, the Hollywood reporter experienced a 50 percent boost in view or retention of a short PDF three months after using the IRIS.TV program.

Search Optimization and Classification

The capacity to digitize text, audio, and video content has resulted in an

explosion of media availability on the Internet, making it harder for users to find exactly what they're looking for. Machine learning technology is being improved to improve the accuracy of search results. Google, for example, is incorporating artificial intelligence into its platform to improve image search accuracy. Instead of putting in keywords for their search, people can now simply upload a sample image to Google Image. Google Image's image recognition engine will automatically recognize and handle elements of the submitted user image and return search results with comparable images. Google also employs artificial intelligence technology in ad placement across the network. A pet food advertisement, for example, will only appear on a pet-related website, whereas a chicken wings advertisement will not appear on a vegetarian-targeted page.

Vintage Cloud has collaborated with "ClarifAI," an artificial intelligence-based business, to create a film digitization platform. Vintage Cloud was able to increase the speed of video content classification and categorization by utilizing ClarifAI's computer vision API.

A startup called "Zorroa" has created a visual asset management platform that is combined with machine learning algorithms. This technology, known as a "Analysis Pipeline," allows users to search for specific content within enormous databases. The database has processors that can uniquely tag each visual asset as well as machine learning algorithms that have been 'trained' to recognize certain components of the visual data. This visual content is then sorted and cataloged so that high-quality search results can be delivered.

Conclusion

Almost everyone will agree that big data has arrived in a big manner and has taken over the commercial sector. But, what is the future of data analysis, and will it expand? What innovations will emerge in its wake? What will the future of big data look like? Will it continue to grow? Is big data on its way to becoming a museum exhibit? What exactly is cognitive technology? What does the future of rapid data look like? Let's have a look at the responses to these questions. To obtain a better understanding, we'll look at some predictions from specialists in Data Analysis and Big Data.

The volume of data will continue to expand. There is virtually no doubt in people's minds that we will continue to create a larger and larger quantity of data, especially given that the number of internet-connected gadgets and mobile devices will expand dramatically. In the future years, we will see a significant advance in the methods we use for data analysis. Although SQL will remain the standard tool, new tools such as Spark will emerge as a complementary way for data analysis, and their number will continue to expand, according to reports.

More and more tools for data analysis will become available, and some of them will not require the analyst's assistance. Microsoft and Salesforce have unveiled several joint capabilities that would enable non-programmers to construct apps for accessing corporate data. Prescriptive analytics will be embedded into business analytics software, and IDC expects that by 2020, half of all software linked to business analysis will be accessible with all of the business intelligence it requires.

In addition to these features, real-time streaming insight into big data will become a distinguishing attribute for data victors in the future. Users will

seek to use data to make informed decisions in real-time by utilizing programs such as Spark and Kafka. Machine learning will be the most important strategic trend to emerge. In the future, machine learning will be required for massive data preparation and predictive analysis in enterprises. Big data will confront significant hurdles as well, particularly in terms of user privacy. The European Union's new private standards are intended to protect consumers' personal information. Various businesses will be required to address privacy controls and processes. In the next years, it is expected that the majority of business ethical infractions would be tied to data.

You can pretty well anticipate every company to have a chief data officer by the end of the year. According to Forrester, this officer's importance will expand quickly, but some types of organizations and age gaps may reduce their importance in the near future. According to Gartner, autonomous agents will continue to play an important role and will be a major trend. Autonomous vehicles, smart advisers, virtual personal assistants, and robots are examples of these agents.

The manpower necessary for Data Analysis will continue to grow, and people ranging from scientists to analysts to architects to experts in the field of data management will be required. However, a scarcity of big data talent may force huge corporations to devise alternative strategies. According to certain significant institutes, numerous organizations will employ internal training to resolve their challenges. On the horizon is a business model based on big data in the form of a service.