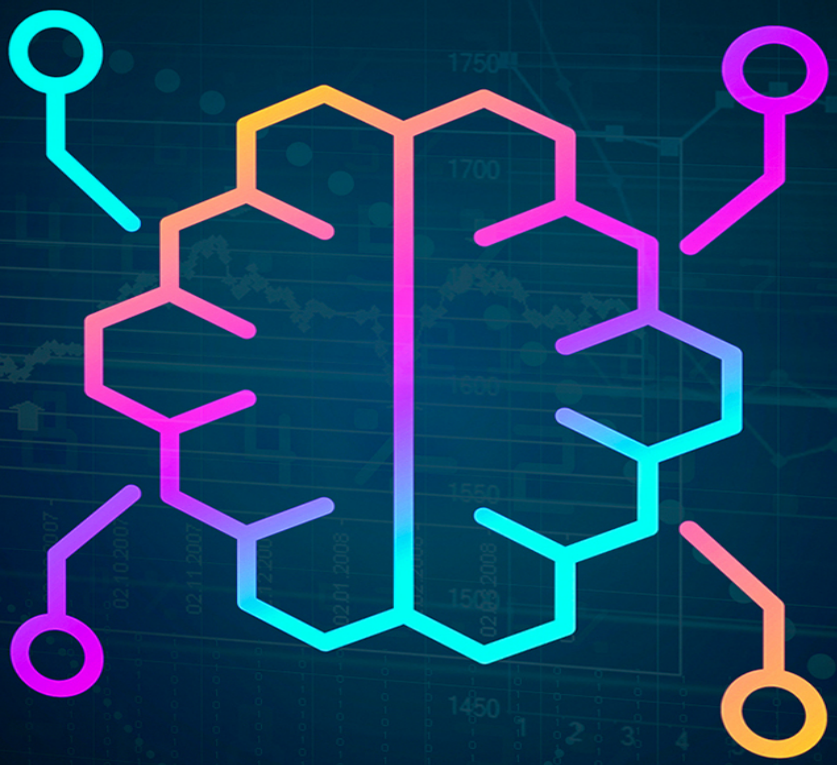
PYTHON PROGRAMMING

THE ULTIMATE BEGINNERS GUIDE TO LEARN PYTHON MACHINE LEARNING STEP-BY-STEP



ALEX STARK


PYTH
PROGRAM
THE ULTIMATE BEGINNERS G
PYTHON MACHINE LEARNIN

Python Programming

The Ultimate Beginners Guide to Learn Python Machine Learning Step-by-Step

Description

If you are brand new to machine learning in Python, then *Python Programming: The Ultimate Beginners Guide to Learn Python Machine Learning Step-by-Step* is the book for you. The book will introduce you to the basic concepts of Machine Learning, Python programming language, various program libraries, and supporting platforms.

Most programmers are amazed at how easily they can pick up the Python programming language, as it is straightforward to use, making it easy to learn. Python is one of the best platforms for those new to programming to begin with, especially if you are going to be using the integrated development environment bundled with the Python package.

Python is also a freely available open-source package that comes with loads of supporting libraries which give it its programming power. For machine learning, Python is fast becoming one of the go-to languages for newbies to cut their machine learning teeth on.

Python's concise, readable code offers programmers a simplicity that other programs do not. Thus, instead of getting lost in complex programming techniques and technical nuances, Python allows the programmer to concentrate on the application they are writing. In machine learning, the programmer needs to pay more attention to the complex algorithms and not get caught up in complex code as well.

For seasoned developers and data scientists, programming languages such as C, C ++, and so on are fine. But for someone starting out with both programming and machine learning, they can be quite cumbersome and confusing. Python programming offers a more streamlined and simpler platform to learn on.

This guide will help you with your journey into the world of Python machine learning and help you navigate your way from a newbie to an intermediate level.

Table of Contents

Introduction

Thank you for choosing *Python Programming: The Ultimate Beginners Guide to Learn Python Machine Learning Step-by-Step* as your guide to Python machine learning.

In this beginner's guide, you will learn the programming language syntax, basic programming concepts, and how to apply them to code. Other topics covered in this booking include:

- Useful Python features that beginners can use in real-world situations
- How to install and configure Python
- Python programming mechanics such as control stream, factors, records or lexicons, and classes
- Machine learning basics and how machine learning is implemented in Python, including Scikit-learn, Tensorflow, and more

Artificial Intelligence (AI)

Both a robot controlled by a computer and a digital computer that is able to perform tasks that an intelligent being can are defined as artificial intelligence (AI). There are many subsets of AI which most commonly include:

- Deep Learning (DL) — neural networks.
- Expert Systems (ES) — teaching systems and systems with decision support.
- Machine Learning (ML) — version space and decision tree learning.
- Machine Vision (MV) — understanding images and character/object recognition
- Natural Language Processing (NLP) — machine translation.
- Planning — gameplay, scheduling, and task-driven systems.
- Robotics — autonomous systems with intelligent control.
- Speech Recognition (SR)— language recognition.

For a machine to be classed as an artificial intelligence system, it needs to be able to:

- Sense
- Reason
- Act
- Adapt

For an AI to be able to carry out its goals successfully and exhibit the four qualities above, it needs to be able to think. Unlike natural intelligence, a computer's intelligence comes from cleverly designed algorithms that can learn. Therefore, one of the most important parts of AI is machine learning, as well as deep learning (which can be classified as a subset of ML).

Machine learning is the subset of AI that provides the machine with the ability to learn. ML is like the brain of an AI system, giving it the intelligence it needs to automatically learn by experience without explicit programming instructions.

Machine Learning (ML)

Machine Learning is the study of algorithms designed to be able to learn through the input of data. Using the data the programmer trained the system on, the algorithm will learn how to describe the data, improve upon the data, and predict outcomes. Most analytical and statistical data found in the modern day comes from using a machine learning algorithm.

A prime example of machine learning is the popup adverts found on search engines such as Google. You may notice that the adverts that appear will pertain to recent searches or products you have looked at on the Web. That is the machine learning algorithm assimilating your search history and predicting what it thinks you will find interesting or useful.

Some of the more common machine learning methods or algorithms include:

- Decision trees
- K-means clustering
- K-nearest neighbor
- Naive Bayes

- Random forest
- Support vector machine

Four Important Techniques Most Artificial Intelligence Systems Have in Common

- Learning — AIs have the capability to gain knowledge from their environment.
- Representation — AI's have the capability to determine how to represent the knowledge they learn from their environment.
- Rules — AIs have an explicit set of instructions or rules which are the guidelines upon which it learns and grows.
- Search — AIs have the ability to search through and find a sequence of states, the connection of weights, etc., that lead to a solution or outcome.

This guide covers an introduction to the basics of Python machine learning. Python is by far one of the easiest and most accessible programming environments to use to start learning about this subdivision of artificial intelligence.

History of Python

Guido van Rossum founded Python after working for a couple of years at the Centrum Wiskunde & Informatica (CWI), in Amsterdam, Netherlands. At CWI Guido worked on a project creating a programming language called ABC in the mid 1980s. It was the ABC programming language that inspired him to create Python in ABC's image in the late 1980s. Guido used his knowledge of what worked best with ABC to draw from and enhance Python. The things he knew only frustrated him with ABC, he found a way to fine-tune or leave out all together if they were not necessary.

Guido created Python to be free open-source software that was easy to use and allowed others to contribute to the language libraries which created a huge Python community. He chose Python for the name of the language when he started to work on the project as a big fan of Monty Python's *Flying Circus* and because he was feeling irreverent at the time.

Version 0.9.0 was the first version of Python that Guido published in February 1991. It was first published on alt.sources as a module and object-oriented system. This version included features like functions, exception handling, and core data types.

January 1994 saw Python version 1.0 released with some functional programming tools that Guido was not too happy with. These tools included a map, filter, reduce, and lambda.

The next new version to be released was done six years later in October of 2000 and was version 2.0, which included a list of new features. Version 2.0 features allowed for the support of Unicode, a full garbage collector, and list comprehensions.

Version 2.x stuck around and was constantly being updated for the next eight years until version 3.0 was released. Version 3.0 was not compatible with version 2.x but could be run alongside it as a completely separate installation. Version 3.0 was released in December of 2008 and was a newer, leaner version of Python. Guido had removed a lot of duplicate modules and constructs from the Python programming language which were found in previous versions.

Although Python 3 is the primary version used, version 2.x is still used by a lot of programmers and the version is still being updated, with the last

update being version 2.7. Development of Python 2.x ceased on version 2.7 in April 2020 with version 3.x gaining more popularity.

Chapter 1:
Python

Python is a versatile interpreted, object-oriented programming language that can be run on any system including Linux, Windows, and Mac. Over the past decade or so, Python has become a very popular programming language due to its ease of use, wide availability, and clear syntax.

Python is open-source software, which means that it can be accessed and downloaded for free. It is easy to use, understand, and follow because it is a programming language that is very similar to the English language.

Python Programming Language Top Features

Since Python's development in the late 1980s, it has become a very popular go-to programming language, especially for new programmers.

Some of the features that make Python so popular include:

- Dynamic typing, which means that Python does not need an explicit declaration of a variable being used.
- Exceptions are not syntactically heavy.
- It is easy to read and follow even if you are not an expert programmer.
- As an interpreted programming language it does not need a compiler to compile a program into machine-language instructions to run.
- Python is an object-oriented programming language similar to Ruby, Perl, and Java.
- Python is free open-source software, as are its vast amount of libraries.
- Python is continuously updated and added to; this includes its libraries.
- Python has a huge standard library, stacked with simple programming commands, efficient processing, and file handling.
- Python has the ability to search through text with commonly used expressions and commands.

- Python’s platform independence means it is available on most operating systems.
- Python is desirable for use with machine learning applications.
- With the use of interactive interfaces, Python makes testing smaller chunks of code a lot simpler.
- Python comes standard with a development and learning environment called **IDLE** . The IDLE interactive shell is a full-featured text editor for Python that makes creating Python scripts quick, easy, and with little to no syntax errors. IDLE comes with features such as a debugger, autocomplete, smart indents, and syntax highlighting.
- Python can seamlessly integrate or work with other software and programming languages like C, C++, Java, and more.
- Python can be used as a program interface for other applications.

Machine Learning

Machine learning (ML) is an algorithm-based program that is developed to encourage a computer to be able to learn without being programmed to. It is a form of artificial intelligence (AI) designed in such a way that the computer will use new data input as a means to learn, develop, and advance.

As AI and ML become more and more entwined with a person's everyday life, we become more and more reliant on them. Devices from phone apps, to cars that can drive and navigate themselves, to the refrigerator in the kitchen use a form of AI and ML. Every time you talk to Siri or Alexa, you are addressing a form of AI. Every time you ask an AI something, it is logging and tracking these choices.

Banking apps and automated assistance are in use every day. The more information fed into them, the more they learn and grow. AI and ML are the here and now as well as the future, which makes it all the more worthwhile to study ML in order to keep your place in the job market.

One of the best programming languages to learn ML on or even AI is Python. A few features that make Python a good language to get your feet

wet with ML include:

- Simplicity
- Consistency
- Extensive collection of libraries
- Extensive collection of frameworks
- Python is platform-independent
- Due to Python’s popularity, it has a large community

Common AI/ML Uses and Technologies Python is Best Suited for

Python is suitable for uses and technologies including:

- Computer Vision using OpenCV technology
- Data Analysis using NumPy, Pandas, SciPy, and Seaborn technologies
- Data Visualization using NumPy, Pandas, SciPy, and Seaborn technologies
- Machine Learning using Keras, Scikit-Learn, and TensorFlow technologies
- Natural Language Processing using NLTK and spaCy technologies

Python AI and ML Libraries

Python's large selection of libraries makes it not only easier to code with, but it also cuts down on development time. Python's rich technology stack includes a vast number of machine learning and artificial intelligence library sets. Libraries contain prewritten code makes scripting a lot easier for developers, cutting down and silly errors that can occur when writing complex algorithms.

Python’s machine learning libraries include the following:

Scikit-learn

Scikit-learn consists of a number of supervised and unsupervised learning algorithms and provides machine learning functionality that includes:

- Classification — K-Nearest Neighbors

- Clustering — K-Mean ++ and K-Means
- Model selection
- Preprocessing
- Regression — Linear and Logistic

Scikit-learn is built upon technology such as Pandas, Matplotlib, and NumPy. It is a powerful library that allows a programmer to engineer stable, robust machine learning programs.

NLTK

Natural Language Toolkit (NLTK) is not specifically designed for machine learning. However, the toolkit comes in very handy in the creation of programs that need to work with human language data. It is used to apply human language data in statistical natural language processing (NLP) and contains numerous text processing libraries.

Theano

Theona is a Python library that is used for machine learning and deep learning. Since 2007, Theano has been used to run computationally intensive scientific investigations on a massive scale. It is a powerful library that can evaluate, define, and optimize complex mathematical expressions with multi-dimensional arrays.

TensorFlow

TensorFlow is a machine learning library used to create Deep Learning models that use fast numerical computing. Although it is run as a Python library, TensorFlow was designed, developed, and released by Google. Like Python, and most of the Python libraries, TensorFlow is open-source software that is easy and free to download and install.

Keras

Keras is a Python library that gives a top-level Application Programming Interface (API) to neural-networking systems. It works best running alongside or on top of Theano, R, PlaidML, MS Cognitive Toolkit, or TensorFlow.

PyTorch

Pytorch is a popular deep learning library written in Lua. It is an open-source model of Torch for Python. It gives programmers access to the same Torch libraries. PyTorch is good for beginners and has some useful tutorials with lots of examples.

Chapter 2:
Start Learning Python

Scikit-learn is one of the most popular machine learning Python libraries, to begin with. It gives new developers a good base from which to learn how the various aspects of ML work, such as labeling, modeling, and testing.

Another good machine learning base to start with is Keras, which is known for its ease of use and straightforward modeling structure. As you become more adept with Python and machine learning, you will be able to make more informed Python machine learning library choices.

Brief Overview of Machine Learning Processes

Before you get into Python machine learning, there are a few machine learning processes that you need to be aware of. These processes include:

- **Supervised Learning** — Supervised Learning is where the system will develop a predictive outcome based on both input and output data. Models of supervised learning include:
    - Classification
    - Regression
- **Unsupervised Learning** — Unsupervised Learning is based on input data. The system will interpret and group data based on the input data. Models of unsupervised learning include:
    - Clustering

The Basic Principles of Python Machine Learning

The basis of machine learning is for the system to learn based on experience. For example, think about teaching a child the difference between shark fins and dolphins fins. You would show them that a shark's tail fin is vertical and that it sways from side to side, while a dolphin's tail is horizontal and sways up and down. Once the child has gathered that information, they will be able to use that to distinguish between the two creatures.

In a similar way, a machine learning algorithm is fed base data upon which to learn and grow. For instance, an estate agent may have statistical information about neighborhoods, the houses in those neighborhoods, and

the average cost per house. This information would be the training information for the ML. From this information, the system would be able to output the average costs of two or three bedroomed houses based on certain criteria. This criteria could be the cost of a two bedroom house in a neighborhood with two bathrooms and a double garage.

Basic Principles of Python Machine Learning

To start designing a Python Machine Learning algorithm, you will need to:

- Determine a problem
- Gather and prepare the required data
- Analyze and evaluate the algorithms
- Increase the outcome of positive results
- Define results & present them in a certain way

Stages of Python Machine Learning

Once you have determined the problem your machine learning algorithm is going to be designed for, you will need to design the ML algorithm.

The stages of Python machine learning are:

- Collection of data
- Sorting of the collected data
- Analyzing the sorted collection of data
- Designing the ML algorithm
- Developing the ML algorithm
- Training the ML algorithm
- Testing the ML algorithm
- Using the ML algorithm in a live environment

Python Machine Learning Example Training Datasets

When you are learning to use ML, you are not going to want to learn on a live dataset. The good news is that there are quite a few free-to-use example datasets available for Python ML coding.

Some of the more popular example datasets are:

- Fisher Iris Flower Dataset — This dataset has been around for many years and is used in most ML training scenarios. It is a CSV file that contains different species of the Iris plant (*Iris Species* , n.d.).
- Prima Indians Diabetes Dataset — This dataset can be used freely as a training dataset for Python machine learning. Like the Iris dataset, you must download the CSV file. This dataset is used to predict a person's chances of getting diabetes based on certain criteria (*Predict Diabetes From Medical Records* . n.d.).

These datasets can both be obtained at a website called **Kaggle.com** .

Place your dataset in a directory that you have administration rights for and note the full path to the dataset.

Both these datasets are small and have enough columns as well as data to be useful as training datasets but will not get out of control. When you are learning to create an ML algorithm, you do not want to start with huge amounts of data that could have loads of variable criteria.

Getting Started with Python

One of the first machine learning Python programs to attempt is the simple “welcome world” machine learning program.

Some Python features that will be discussed in this section are:

- Introducing the Python platform and libraries
- Stacking the data set
- Condensing the data set
- Introducing the data set
- Working with algorithms
- Predicting some outcomes

While you are starting out and learning, it is important to remember to take as much time as needed. Progress at your own pace and join a Python support group. There are many out there, and some of the more popular ones can be found on the Python website (The Python Community, n.d.).

Step 1: Download and Install

The latest version of Python is available on the Python website (Python.org). It is best to download the latest version of Python on all the supporting libraries. At the date this guide was published, Python version 3.8.5 was the latest version. Version 2.7.18 was still supported and will work with the following exercises.

- Download Python from the Python.org website.
- Follow the instructional guide provided by **Python.org** to install and set up the Python environment in readiness for ML.

Step 2: Install the Python Libraries

Python Machine Learning Supporting Libraries you will need are:

- Matplotlib
- Numpy
- Pandas
- Scipy
- Scikit-learn

Once again, follow the comprehensive instructional guide on the **Python.org** website on how to install Python libraries.

Step 3 - Build Your Environment

The easiest way to code with Python ML and build an environment is with Anaconda using Jupyter Notebook (which is part of the Anaconda suite).

To download Anaconda, go to the **Anaconda** website (*Anaconda Individual Edition* , n.d.). Download the latest version to ensure it works with the installed version of Python on your machine.

Both the Python website and the Anaconda website have comprehensive guides on how to download, install, and set up the Python ML environment with Anaconda.

Alternatively, you can use Python SciPy. The following examples run with Python SciPy. To run and set up the Python SciPy environment, go to the SciPy webpage, and follow the comprehensive installation and setup guidelines *(SciPy Installation* , n.d.).

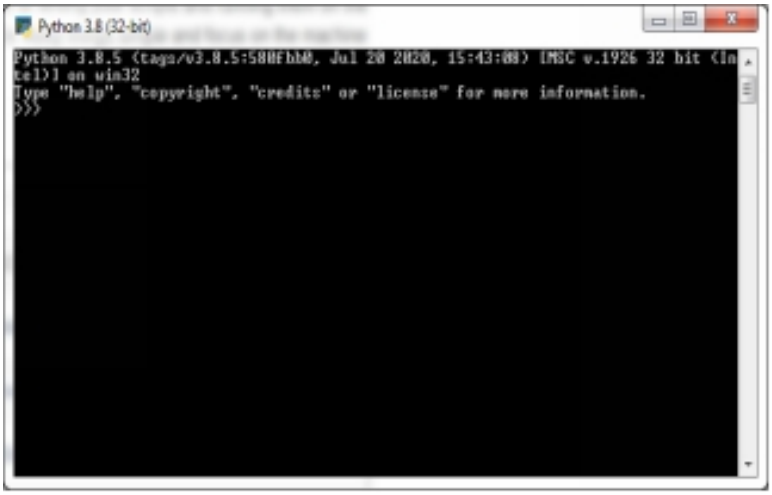
Step 4: Check the Python Installation and Import the Libraries

For the first few exercises, it is best to use the Python 3.8 command line.

At a new command line in the Python interpreter, type the following:

python

The following screen should appear:



Type the following commands to import the needed libraries:

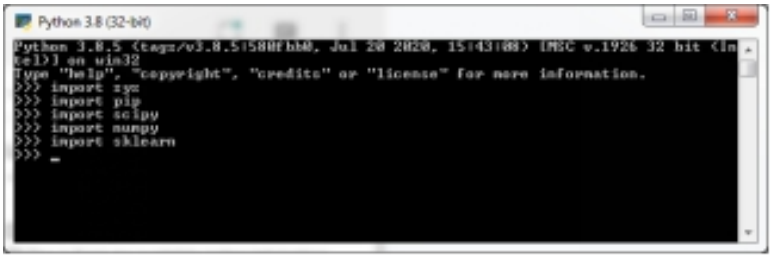
import sys

import pip

import scipy

import numpy

import sklearn



You will need to install Matplotlib and Pandas. To do this, type the following commands at the command line prompt.

To install Matplotlib type the following:

python -m pip install matplotlib

The installation will gather the matplotlib files and then install them. If the installation ends with a warning that your version may need to be upgraded, run an upgrade by using the following command:

python -m pip install matplotlib --upgrade

To install Pandas type the following:

python -m pip install pandas

The installation will gather the Pandas files and then install them. If the installation ends with a warning that your version may need to be upgraded, run an upgrade by using the following command:

python -m pip install pandas --upgrade

Once Matplotlib and Pandas are installed and up to date, type the following at the command line prompt:

import matplotlib

import pandas

This will import the modules and you should now have all the required libraries/modules imported and ready to use.

To check the modules that are installed in Python, type the following at the command line prompt:

pip list

or

help(“modules”)

The system can take a minute or two to gather the necessary information and will display a list of all installed and available libraries in Python.

Chapter 3:
Working with Data

To work with data, you are going to download the Fisher Iris Flower Dataset from the following website (download the ZIP file):

https://gist.github.com/curran/a08a1080b88344b0c8a7#file-iris-csv

If you want to read up on this dataset you can do so in Wikipedia, which has a host of information on it. The dataset was designed by Ronald Fisher in 1936 as a linear discriminant analysis example (*Iris Flower Data Set* , n.d.). There are three species of iris listed in the dataset, with over fifty samples per species. The length and width of the petals and sepals were measured for all the samples to differentiate between the iris species.

Download the Iris data into Python using CSV format and place it in an easily accessible directory.

For example:

c:\datasets\

Step 1: Load Sklearn, Pandas, and Matplotlib Supporting Libraries

Once you have the Iris dataset CSV file loaded, you need to load the supporting libraries for Sklearn, Pandas, and Matplotlib.

Import Sklearn Supporting Libraries by typing the following into the command line:

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
```

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

from sklearn.svm import SVC

from sklearn.model_selection import StratifiedKFold

from sklearn.model_selection import train_test_split

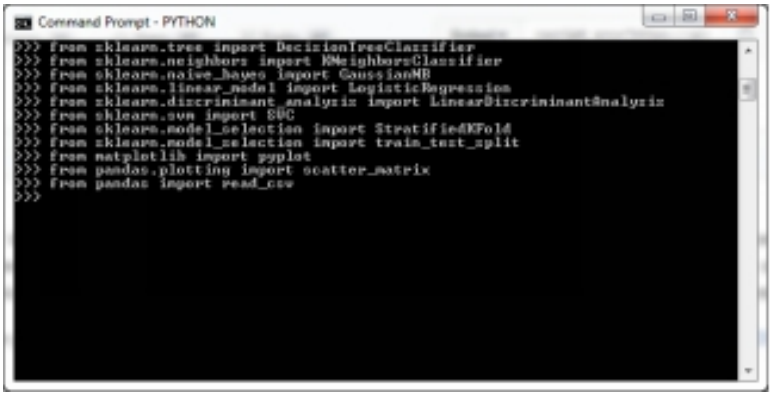
Import Matplotlib Supporting Libraries by typing the following into the command line:

from matplotlib import pyplot

Import Pandas Supporting Libraries by typing the following into the command line:

from pandas.plotting import scatter_matrix

from pandas import read_csv



If you encounter an error, check that you have Sklearn, Matplotlib, and Pandas correctly installed and imported into Python.

Step 2: Import the Dataset into Python

Pandas is used to import the dataset and set up column names. This is how we set up the visualization of the output data.

You will need to know the exact path to the directory where you downloaded the Iris CSV file to (ex. c:\datasets\iris.csv).

For the following exercises, you will be setting up a **DataFrame** . A DataFrame is like an Excel spreadsheet that contains a number of columns listing the data types. It will also contain a number of rows containing the

data pertaining to each data type. In Python, a DataFrame is classified as a 2-dimensional data structure (*Intro to data structures* , n.d.).

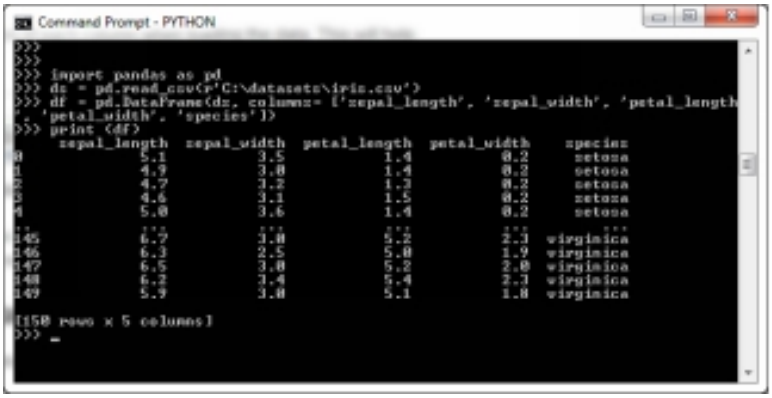
At the command line prompt type in the following:

import pandas as pd

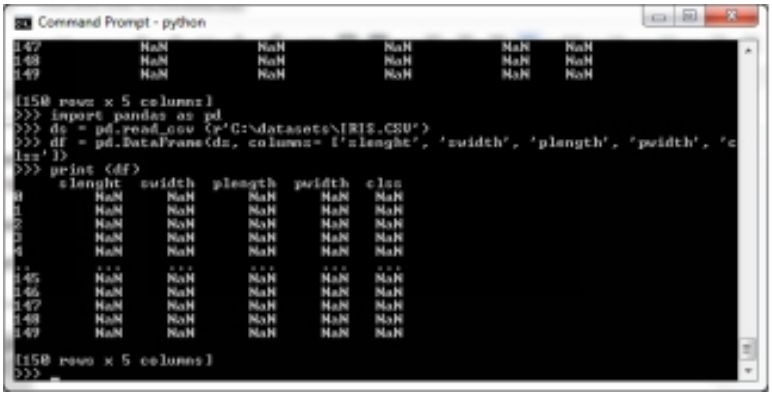
ds = pd.read_csv (r'c:\datasets\iris.csv')

df= pd.DataFrame(ds, columns= ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

print (df)



Make sure the column headings you use in the "df" definition match the column names in the .CSV file. If they do not, you will get NaN values in the table when it prints as per the image below.



Step 3: Working With the Shape Property

There are different ways to view the data and different metrics that can be pulled from. One such metric is the dimensions of the dataset.

When there are massive amounts of data set out in rows and columns, there is a quick way to figure out how many instances and attributes there are.

You can do this by using the **.shape** property of the defined dataset.

For this exercise, you are going to use the already defined DataFrame, called **df** , that you created in the previous exercise.

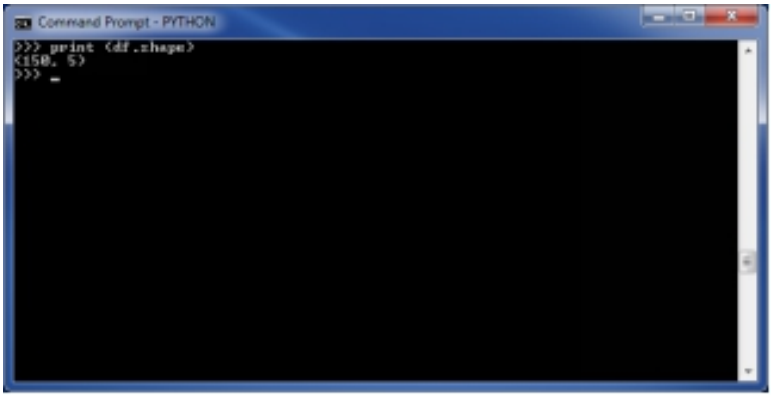
To find how many attributes and instances there are in the iris.csv dataset, type the following into the command line prompt:

print (df.shape)

The command will return:

(50, 5)

This translates to 50 instances of 5 attributes. Instances are the rows of data (data) and the attributes (describes the data) are the columns of data.



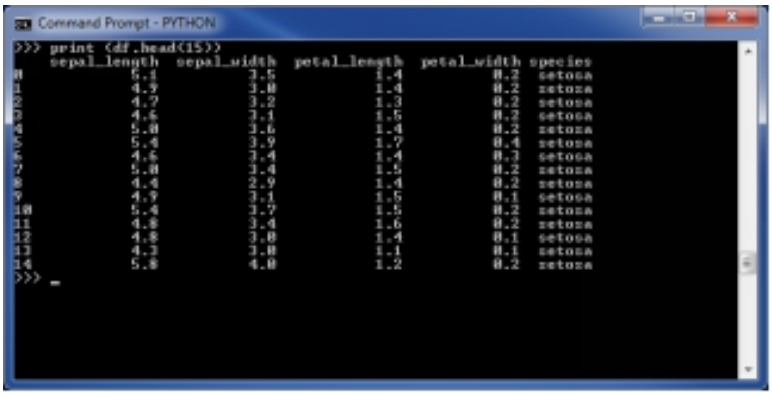
Step 4: Working With the Head() Property

Once you know how much data there is, you can find out if you have the correct data type or view the quality of data by looking at the first few rows of it. To view the first few rows of data in the dataset you will use the **.head()** property.

Use the DataFrame you defined in the first exercise as **df** . In the command line prompt type the following:

print (df.head(15))

This tells the system to print the first 15 rows in the DataFrame **df** ; the system should return information similar to the image below.



Step 5: Working With the Describe() Propert y

Python Pandas includes a number of Descriptive Statistic functions, listed in the table.

Descriptive Statistics Function	Use of the Function
abs()	Absolute Value
count()	Amount of non-null observations
cumsum()	Cumulative Sum
cumprod()	Cumulative Product
max()	Maximum Value
mean()	Mean of Values
median	Median of Values
min()	Minimum Value
mode()	Mode of Values
prod()	Product of Values
std()	Standard Deviation of Values

It is important to note that some generic operations will not work with all of the descriptive functions because a DataFrame is a heterogeneous data structure. Some functions will work with both character/string and numeric data, while some only work with numeric data.

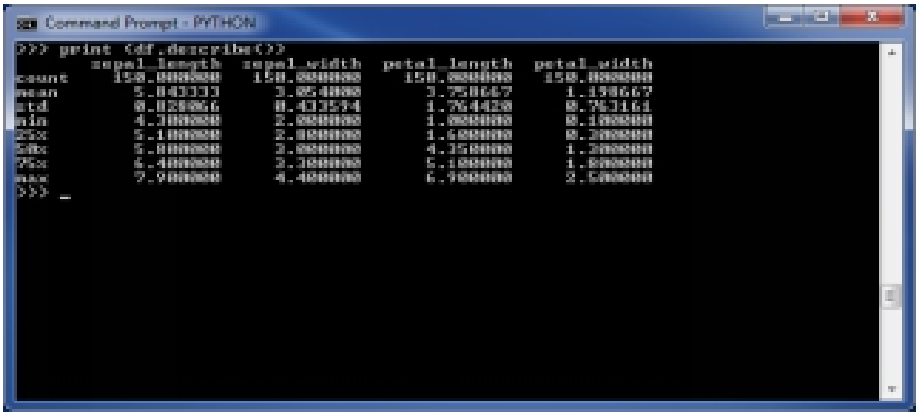
- Functions such as abs() and cumprod() will return an exception if a string or character data is used.
- Functions such as sum() and cumsum() will work with numeric and character/string data.

For a statistical summary of the data columns in the DataFrame the .**describe()** function is used.

Type the following into the command line to summarize the data in the **ds** Dataframe columns.

print (ds.describe())

The system should return a screen that looks similar to the one in the image below.



The .describe() function returns:

- count — Total count of the objects listed per data item excluding any null values in the DataFrame.
- mean — The mean value (average) of all the data in the DataFrame.
- std —The standard deviation for all the listed data in the DataFrame.
- min — The minimum value for the data in the DataFrame.
- IQR — The rows listed as 25%, 50%, and 75% are the **interquartile range** of the data values in the DataFrame. This measures the difference between the 25th, 50th, and 75th percentile of the data.
- max — The maximum value for the data in the DataFrame.

The function excludes any columns with characters, which is why it does not list or include the “species” column in the output table.

The Iris dataset is measured in centimeters. From this summary a person can see that the values in the table fall within a scale of between 0 to 8 centimeters.

Step 6: Working With the Groupby() Property

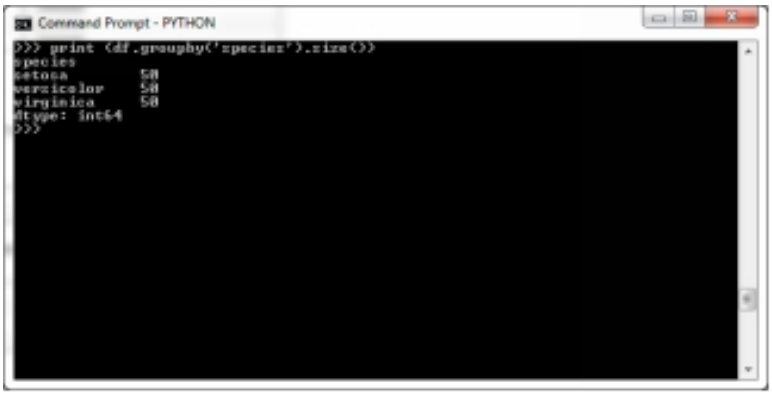
In a massive table of data, it is easier to view and work with data that is split up into sets. To split the data into sets the .groupby() property is used. This property is used to perform functions such as:

- Aggregation — This function will compute a statistical summary of the data.
- Filtration — This function will sort through and discard conditional data.
- Transformation — This function is used for group-specific information.

To use the .groupby() property, type in the following at the command line prompt to group the df DataFrame by species and size.

print (df.groupby('species').size())

The system should return a table that looks similar to the one in the image below.



Step 7: Creating Your Own Datasets

Why not try the following exercise and create your own dataset?

Think up a table of statistics such as the sales statistics per product per year.

Create a worksheet and fill it with data, for example:

Item	Sales	Year	Rank
Butter	20000	2010	2

Cream	15000	2011	3
Eggs	35000	2012	1
Milk	45000	2013	1
Chocolate	40000	2014	2

The above table is a list of inventory for a shop, total sales for each year, and where their sales ranked within their territory.

You can create the table in Excel and save it as a CSV file and then import it into Python. Or, you can create it directly in the command line to practice with the functions you have learned in the chapter.

Type the following into a new command line:

import pandas as pd

sales_data = {'Item' : ['Butter', 'Cream', 'Eggs', 'Milk', 'Chocolate'],

'Sales' : [20000, 15000, 35000, 45000, 40000],

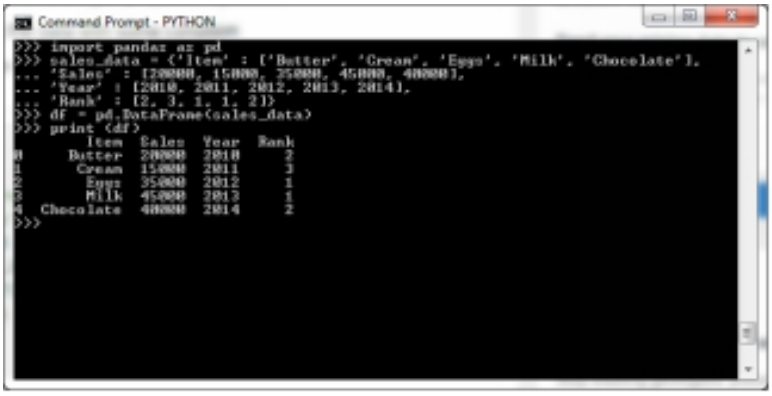
'Year' : [2010, 2011, 2012, 2013, 2014],

'Rank' : [2, 3, 1, 1, 2]}

df = pd.DataFrame(sales_data)

print (df)

A table similar to the one below will be the systems output.



Create your own datasets and use the functions in this chapter to gain various statistical information from it.

Chapter 4:
Working with Data Visualization

The concept of data visualization is to view or understand data in a visual form. Laying data out in a visual form makes it easier to distinguish trends, patterns, and correlations that may otherwise be missed.

Python has quite a few libraries for data visualization. Each of these graphing libraries has useful features. The library you choose to use should be one that best suits your data visual needs. Some of the Python graphing libraries that can be used are:

- Matplotlib — This library is one of the more popular graphing libraries in Python. It is used to create bar charts, histograms, line charts, and other basic graphs. Matplolib takes quite a bit of coding to create its data visualization output.
- Pandas Visualization — This library has an application programming interface (API). It is for high-level visualization with less coding requirements than Matplotlib.
- Plotly — This library is good for creating interactive plots for exploratory visualizations.
- Seaborn — This library is based on the Matplotlib library. It is a high-level interface designed to give more informational statistics attractively displayed.

There are different kinds of plots that can be used to visualize data. Some of the more commonly used plots include:

- Bar Graph
- Line Chart or Graph
- Histogram Plot
- Scatter Plot
- Box and Whisker Plot

Matplotlib is Python's core plotting function. Understanding how this library works helps you better understand the plotting functions in Pandas and Seaborn.

There are three types of exploratory data analysis methods used for plots. These are:

- Bivariate analysis — This method is used to find the relationship between the variables in a dataset.
    - Box plot
    - Violin plot
- Multivariate analysis — This method is used to find the interaction between two or more variables in the dataset.
    - Scatter plot
    - Pair plot
- Univariate analysis — This method is used to summarize each field in the dataset statistics based on one variable.
    - Box plot
    - Cumulative distributive function (CDF)
    - Probability density function (PDF)
    - Distribution plots
    - Violin plot

For this chapter, you will need to download the Wine Review Analysis CSV file from the following website:

https://github.com/gorbulus/WineReviewAnalysis

Place the CSV in the same directory you placed the iris.csv file.

C:\datasets\

You will find that the CSV file has a long name, so it is best to rename this to wine.csv.

The dataset contains the following column heading:

- *_blank heading_*
- country
- description
- designation

- points
- price
- region_1
- region_2
- variety
- winery

Remember when setting up dataset descriptions that the names of the columns have to be an exact match to the ones in the CSV file. This includes the lower and uppercase words (it is case sensitive). The first column heading in the CSV file is blank. It is best to give it a name, for instance “num”.

Getting Started With Data Visualization

Install Seaborn by typing the following in the command line prompt:

python -m pip install seaborn

import seaborn

The last line will test whether or not Seaborn was successfully installed. If you are returned to the command line prompt with no errors, the library has been installed.

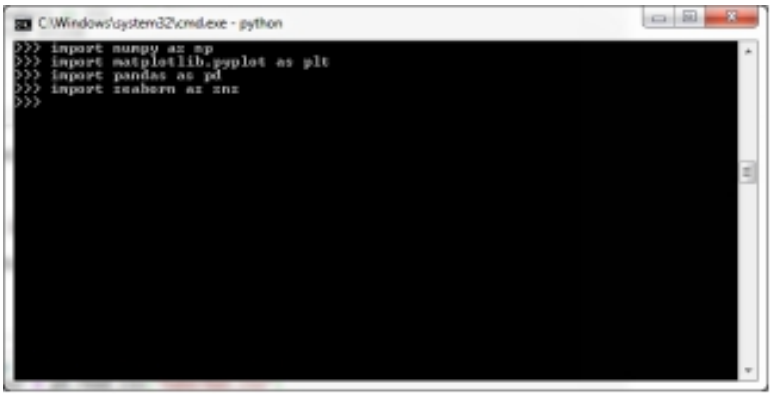
The first thing you have to do is import and rename the libraries in a new command line.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns



Next, you will need to load the wine.csv dataset into the system by typing in the following at the command line prompt:

wine = pd.read_csv(r'C:\datasets\wine.csv')

df = pd.DataFrame(wine, columns= ['num', 'country', 'description', 'designation', 'points', 'price', 'province', 'region_1', 'region_2', 'variety', 'winery'])

Basic Wine Review Analysis

The basic analysis would be what the **.shape** , **.head()** , and **.tail()** functions would give of the wine review analysis data.

Type the following into the command line prompt to find out the number of rows and columns in the file:

print (wine.shape)

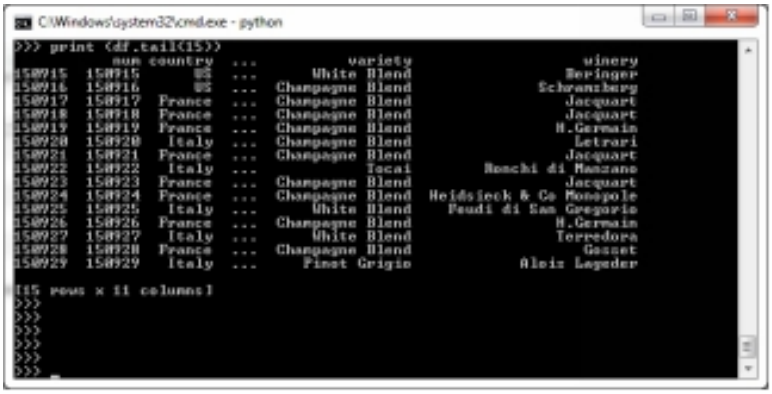
There are 150930 rows x 11 columns in the dataset

View the first 15 rows of data in the dataset by using the **.head(15)** property.

print (df.head(15))

View the last 15 rows of data in the dataset by using the **.tail(15)** property.

print (df.tail(15))

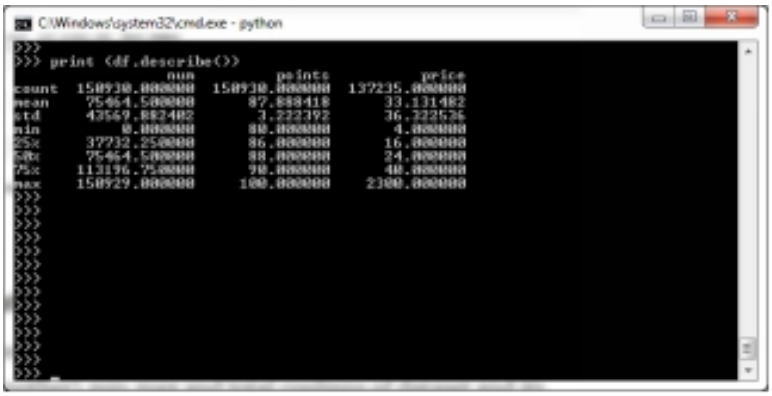


High-Level Wine Review Analysis

High-level analysis can supply you with more in-depth statistical information on the dataset. This is information you would get from using the .describe() and .info() properties. These types of statistics can be used to return the IQR stats, the means values, standard deviation values, maximum values, and so on.

Type the following into the command line prompt to output some high-level wine review statistics:

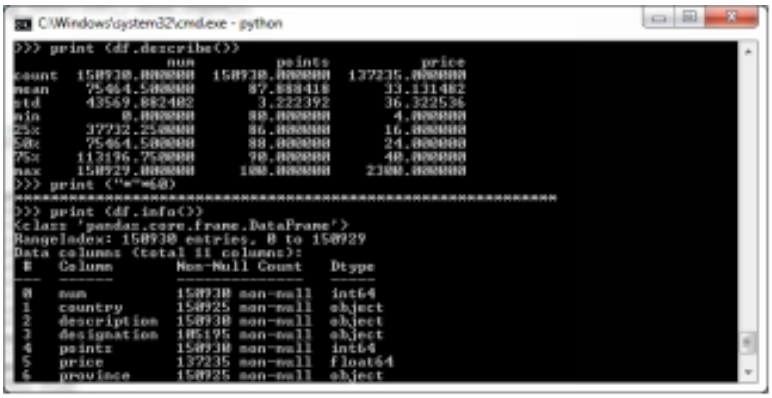
print (df.describe())



Type the following into the command line prompt:

print (“”*60)*

print (df.info())



Univariate Analysis

There are a few plots that you can do in order to do graph analysis based on one variable. For these examples, you will be using the iris.csv DataFrame.

Distribution Plots

As the data in the dataset is numerical, it is easy enough to create **box and whisker** graphs. These types of plots offer you a visual of the data distribution.

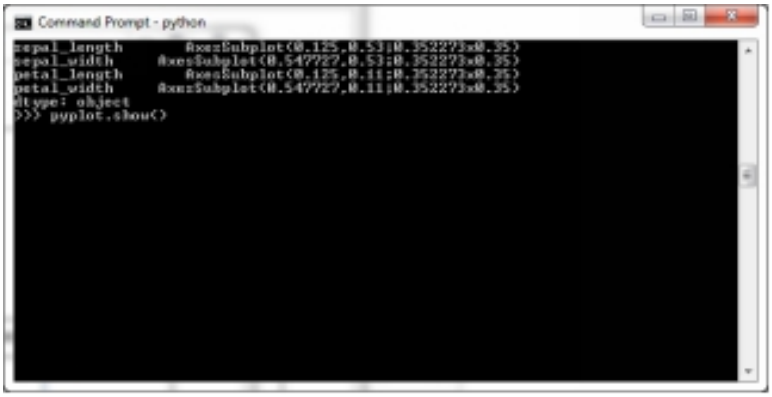
Type the following into the command line prompt:

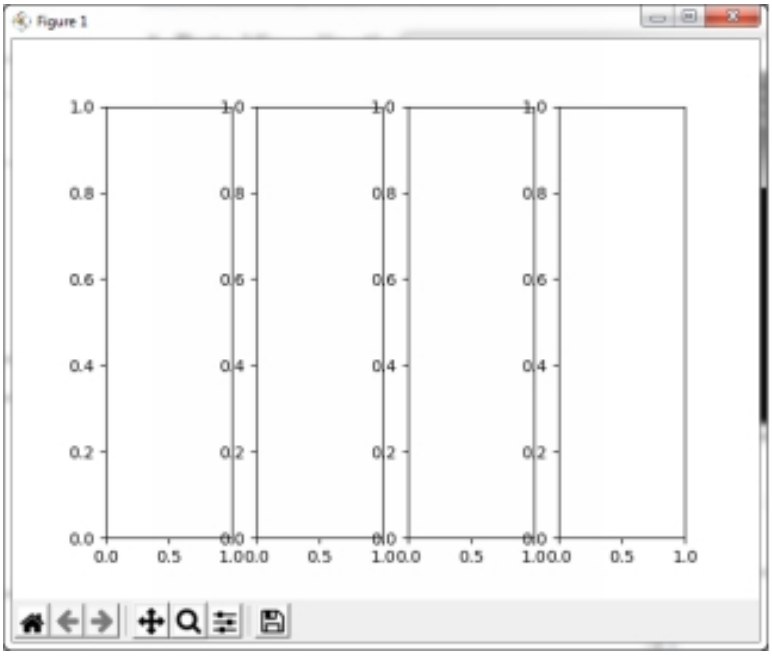
iris = pd.read_csv(r’C:\datasets\iris.csv’)

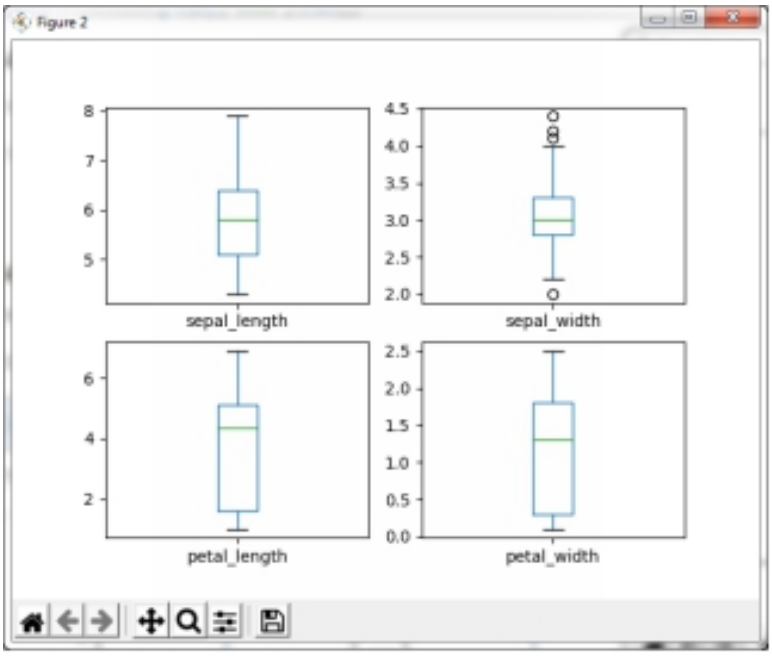
df= pd.DataFrame(iris, columns= [‘sepal_length’, ‘sepal_width’, ‘petal_length’, ‘petal_width’, ‘species’])

df.plot(kind= 'box', subplots=True, layout=(2,2), sharex=False, sharey=False)

pyplot.show()



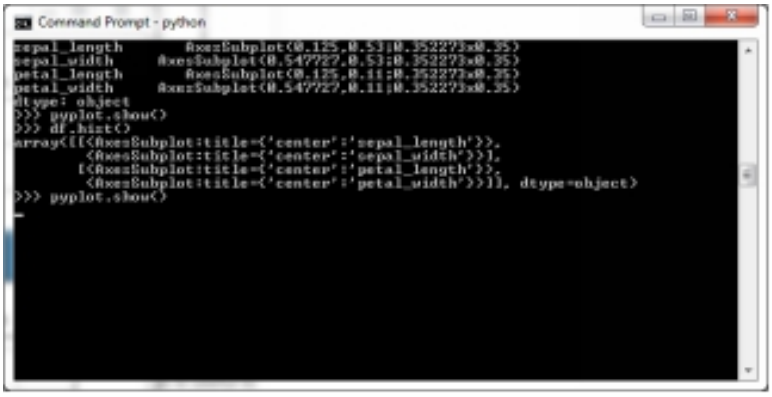


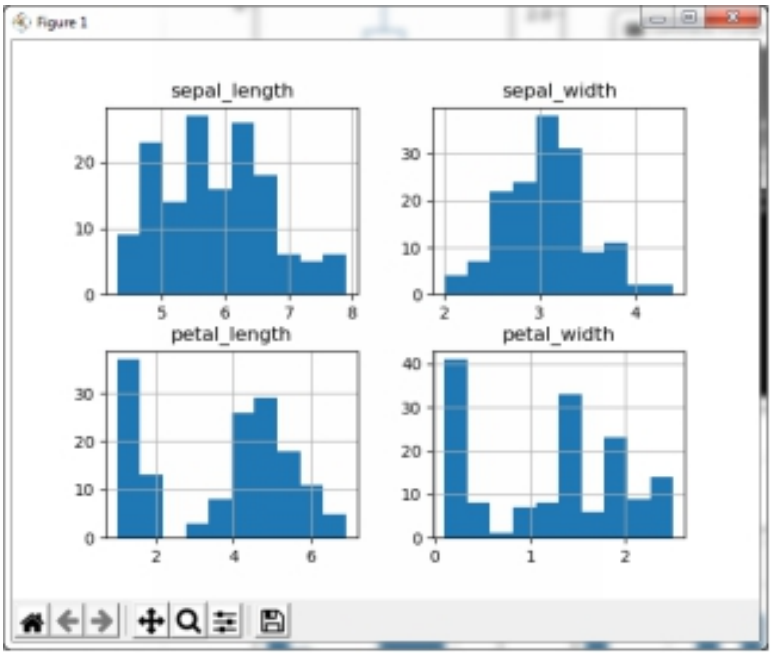


The **histogram** is another way to get a distribution plot. Type the following at the command line prompt:

df.hist()

pyplot.show()





Multivariate Analysis

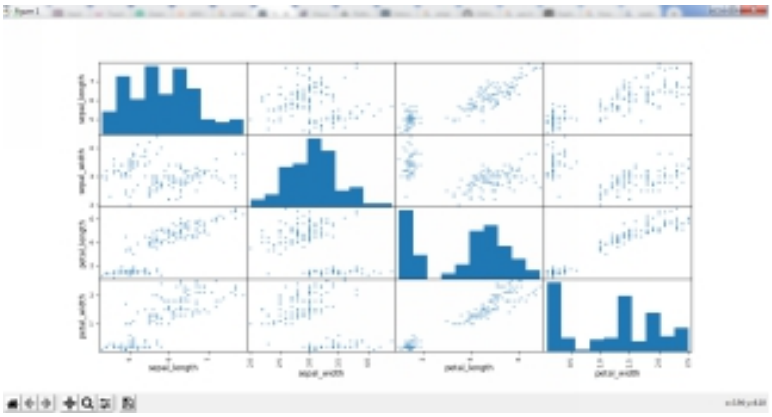
Using the wine.csv DataFrame that you have already set up, you are going to continue using the iris DataFrame you created above.

The Multivariate analysis allows you to determine the correlations between the data in the dataset. This example uses the scatter plot.

Type the following at the command line prompt:

scatter_matrix(df)

pyplot.show()


sepal_length
sepal_width
petal_length
petal_width

Chapter 5:
Predictive Analytics

There are very few organizations, especially the larger ones, that do not use predictive analytics to grasp new opportunities or solve current problems.

What is Predictive Analytics?

Historical data is used by various machine learning techniques and statistical algorithms to predict future outcomes. By learning about what happened in the past, predictive analytics can be used to predict possible future outcomes.

In the past, predictive analytics was left to the statisticians and mathematicians. With the advances in technology business analysts have seen the advantages of using predictive analytics to increase business revenue, especially in today's competitive business environment with social media and interactive software being more readily available.

The world of technology as it stands today allows for faster and cheaper computer systems with easy to use analytical software. This gives computer users access to larger volumes of different types of data. This data can be used to provide organizations with valuable insights into various business, market, and customer trends.

Business environments in this century face tough economic conditions and to keep ahead or maintain their edge in the marketplace they need a competitive edge. Having access to large volumes of data from which they can pull various statistics gives them this edge.

The Use of Predictive Analytics

Predictive analytics is used in many systems these days and plays an important role in:

- Fraud detection
- Operations improvements
- Marketing campaign optimization
- Reducing risk
- Predicting market trends

As systems become more advanced so does cybercrime. With predictive analytics, systems can more easily identify certain patterns that can quickly detect anomalies. Fraud detection systems run high-performance behavioral analysis algorithms to find any indications of advanced threats.

Businesses now use predictive analytics to enhance their marketing campaigns to grow their customer base and ensure their current ones keep coming back. This is done with the analysis of their customer's responses to various items, services, and so on.

Risk reduction comes in being able to assess a customer's likelihood of defaulting on loans, higher purchases, etc. Insurance companies use it for insurance claims, and much more.

The hotel industry is a prime example of using predictive analytics to increase their revenue and ensure high room occupancy per night.

Who Uses Predictive Analytics?

There are not many industries that do not use some form of predictive analytics.

Some real-world examples include:

Aviation

As airlines expand the globe they fall into the category of large complex businesses that come with large complex processes. Predictive analytics help airlines improve upon network route planning, being able to quickly adapt to emergencies, and avoiding overbooking.

Airlines are also striving to offer their customers more personalized services. With predictive analytics, the system can offer these to the customer based on their previous choices.

Banking and Financial

The banking and financial industries use predictive analytics to detect fraud and can do so within a fraction of the time a human could. Customers' credit risks can be assessed within minutes instead of days or weeks. With data being collected on their customers thanks to Internet banking, these institutes are able to offer more personalized packages to their customers.

Car Industry

Connected car technology means that there is bidirectional communication between the car and other systems on the Internet it is connected to. These communications include various statistics such as car mileage, malfunctions, and more to the car company or registered dealer. Car tracking companies will keep a log of where the car is going, where it has been, the speeds it was traveling at, etc. All this data is being collected and used to improve upon car performance, response times to breakdowns, security issues, traffic problems, and more.

Car manufacturers are able to assess the quality of various products used in a particular vehicle. They also use predictive analysis to assess how their latest in-car technology is working or being received.

Government and Public Sectors

Governments have been taking advantage of predictive analytics for a lot longer than most sectors. They use statistics from the Bureau of the Census to help with population trends, get a better understanding of their customers, and to help improve upon services.

It also helps them find where a district may be in need of schools, bus routes, and various town services upgrades or implementations such as sewages. If a neighborhood is becoming more popular, the city needs to be able to ensure the area can sustain the growth. Predictive analysis can easily ascertain this and help the city to plan for this growth.

Health and Medical

Health and medical industries use predictive analysis to help predict who is more likely at risk of developing certain diseases. By gathering a patient's medical history, lifestyle choices, and habits, predictive analysis can determine if they are at risk for developing heart disease, cancer, diabetes, and more.

The medical industry is fast advancing by using machine learning techniques that utilize complex predictive algorithms.

Insurance

Insurance companies use predictive analytics to prevent fraud, speed up the claim process, and can detect when a person is not adhering to the terms of

their contract. For instance, in the medical insurance sector, insurance companies have access to doctors and pharmacy databases. Medical insurance systems can detect a client's purchase history and medical visits. This results in the system being able to spot patterns where the customer is not keeping to their script or health care instructions.

Manufacturing

Manufacturers use predictive analytics to ensure the quality of parts, service resources, and reduce waste or risk of systems failure. By being able to access things like warranty claims, customer complaints, product recalls, and sales data, manufacturers are able to improve their products. With the explosion of social media gaining ground, it has opened up a new advertising stream. Cheaper advertising that has a larger outreach means that large manufacturers are not only trying to keep a foothold with large corporations, but smaller companies, too.

Predictive analytics helps manufacturers keep a foot in the market, quickly assess inferior products, and enables them to keep up with or stay ahead of market trends.

Oil, Gas, and Utilities

Being able to predict when a large power-generating turbine needs maintenance or supply may shortly not meet demand is a huge plus for the oil, gas, and utility industry. With the deregulation of power companies at the turn of the century, consumers have access to private power supplies. Predictive analytics gives power companies the edge to keep ahead of their consumer's needs in order to stop them from going elsewhere.

Chapter 6:
Working With Algorithms

By now you should have an idea about the workings of some of the basic machine learning functions in Python and know what predictive analytics is. In this chapter, you are going to work with algorithms.

What is an Algorithm?

An algorithm in programming is a set of instructions that need to be executed in an orderly fashion from start to finish in order to achieve the desired outcome. Designing an algorithm is universal and not code specific. One algorithm may be used across many different programming languages. Although it may be written differently, the functionality and outcome will be the same no matter the language it is written in.

Important Categories of an Algorithm

There are different categories that algorithms fall into that a programmer needs to take into account when designing an algorithm.

The following are import algorithm categories to be aware of:

- Delete — This would be an algorithm to delete duplicate data from an existing data structure.
- Insert — This would be an algorithm to insert data into an existing data structure.
- Search — This would be an algorithm to search for specific data in an existing data structure.
- Sort — This would be an algorithm to sort data in a certain order in an existing data structure.
- Update — This would be an algorithm that enables data in an existing data structure to be updated/modified.

Algorithm Characteristics

Algorithms have well-defined characteristics and if a procedure does not display these characteristics, they cannot be called algorithms.

The following are characteristics of an algorithm:

- Effective — An algorithm needs to be useful with no unnecessary steps that could make it less effective. It has to be feasible with the resources available.
- Explicit — An algorithm must be unambiguous; it can have only one meaning, so it has to be clear and concise with each step laid out.
- Finite — An algorithm needs to display the characteristic of finiteness and terminate after a specific number of steps.
- Input — An algorithm must have well-defined inputs. This means that the programmer needs to know the amount of data, the form of the data, and what kind of data they are dealing with. All algorithms should have zero or more inputs.
- Output — As with the input data, to get the required output a programmer needs to know the kind of data they are working with, how much data there is, and the form of the data. Algorithms need 1 or more outputs that must be well-defined.

Getting Started With Algorithms

As algorithms are not code specific, programmers generally write them using basic common code constructs that are shared with the different programming languages. There are no set hard and fast rules or standards for designing or writing algorithms either, rather they are written depending on the available resources for a problem that exists or may arise.

Writing algorithms is a process that starts once the problem it is being written for has been well-defined. As an algorithm is a step-by-step set of procedures, writing the algorithm is generally a step-by-step procedure, too.

How Do You Start Writing an Algorithm?

An algorithm is like a recipe. You have certain ingredients that are used (data), and these ingredients are mixed in a step by step procedure (performing a task) in order to get the desired dish (outcome).

Using a recipe as an example, you are going to learn how to write an algorithm.

Step 1: The Problem

The first step is to define the problem. In this case, it is a recipe, but for what?

- You want to make a cake.

What type of cake:

- Chocolate cake.

How many cakes?

- 1

Now the problem has been defined:

- You want to make 1 chocolate cake.

It must be noted that problems in real-life situations are not as simple as above. There is a lot that goes into defining a problem. Usually, developers do not get involved at this level unless they are developing a system they want to develop for themselves.

Although most developers shy away from getting involved in defining the problem, they are always roped in. If you go back to the start of this chapter part of a characteristic of an algorithm is a “well-defined problem.”

Some of the common features of a well-defined problem should include:

- The descriptions should not be ambiguous.
- The problem should be clear and concise with no unstated presumptions.
- There should be no internal contradictions.
- The description of the problem should be complete and detailed.

Step 2: Analyze the Problem

Once the problem has been identified as a well-defined one, the programmer needs to analyze it. During this step, you need to answer questions such as:

- What kind of data is it?
- Are there rules pertaining to working with the data?
- Is the data readily available?
- Where is the data?

- What is the relationship between the data?
- What items need to be removed?
- What items need to be changed, updated, or modified?
- What are the changes, modifications, or updates?
- What is the desired outcome or end results?

Step 3: The Algorithm

- Input the ingredients (input data)
- Mix the ingredients in a certain way and put it in the oven (perform the task)
- Output the baked cake (output data)

Now that you have defined the problem and analyzed it, it is time to figure out the solution algorithm to obtain the desired outcome. But if you look at baking cakes, for example, you will see that there is more than one way to bake a cake.

As with any problem, there is usually more than one solution that can output the desired results. You can't just take the first solution you have presented, because the second, third, or even fourth solution may be a better fit for the situation.

When designing algorithms, you will more than likely end up with a number of solution algorithms that could be suitable for the defined problem.

Step 4: Analyzing and Choosing the Best Solution Algorithm for the Problem

The best solution algorithm will depend on a number of factors, such as:

- The resources at your disposal — you can eliminate solutions that may incur extra time, effort, or resources.
- Costs — The most cost-effective solution that does not cut back on quality and puts no extra stress on other resources.
- Speed — An algorithm needs to be streamlined and finite. If one solution has fewer procedures to deliver the desired outcome, you are going to choose it.

Step 5: Develop the Algorithm

Once the solution algorithm has been chosen, you will need to develop a high-level algorithm. This is the basic structure of the algorithm which will give a satisfactory day-to-day outcome if run as is.

This structure is best to start with, like the framework for a house. Once we know it will effectively perform the basic task, for instance, if the foundation and framework are solid, we can add detail to it.

Step 6: Refine the Chosen Solution Algorithm

Once the solution algorithm has been chosen, it needs to be refined by adding more detail. What else does the algorithm need, what are potential problems that could arise, etc.

Like baking a cake, when you are designing an algorithm it is not only the data or problem that has to be taken into account. You have to think about the more technical aspects like computing power. What is it going to take to run the algorithm?

A good rule of thumb is to over-stack the algorithm with detail rather than to under-stack it. It is easier to take away detail than it is to add it.

Step 7: Review

Step 7 is like a never-ending loop. Once the algorithm is done it must be constantly reviewed for changes, updates, or ways to make it better. If there are updates, changes, or additions it will go back a step or two.

Choosing a Model

Not only could there are many different solutions to a problem, but there could also be a few models to base the algorithm on.

Some of the machine learning algorithm models in Python are:

- Classification and Regression Trees (CART) — Used for classification and prediction modeling.
- Decision Tree — This model is used for decision-based outcomes like a bank would use to approve a loan.
- K-Means Clustering — This model is used for unsupervised learning models. It is used to find patterns in specific datasets.

- K-Nearest Neighbor (KNN) — Determines patterns by observing its closest neighbor.
- Linear Regression — A basic yet powerful model, used to predict an outcome based on one or more independent variables.
- Logistic Regression — This model works on the same principles as a classification model. It is used to predict new probable outcomes based on a particular category.
- Principal Component Analysis (PCA) — This model takes a huge chunk of data and condenses it into a more manageable one.

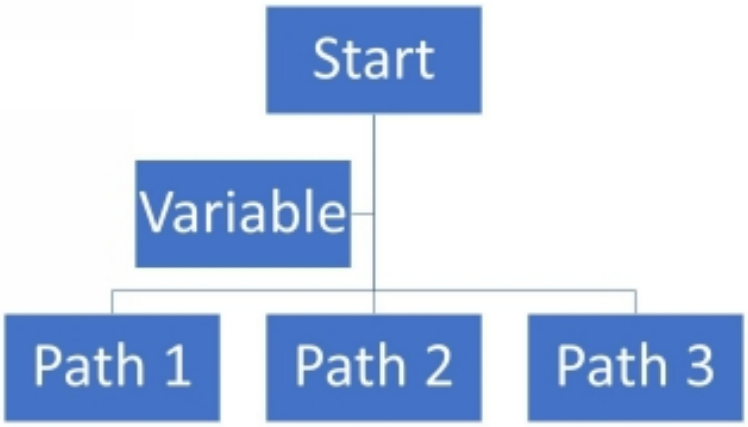
In the next chapter, you will have a brief overview of some of the models listed above.

Chapter 7:
Machine Learning Algorithms

Decisions Tree (Example of a Classification Tree)

Anyone who has done math, programming, or business courses knows what a flow chart is. They can be used as graphical representations of a transactional flow for most business, mechanical, technical, and programming scenarios.

The image below represents what could pass as a basic flowchart, easily recognizable to anyone.



What is a Decision Tree?

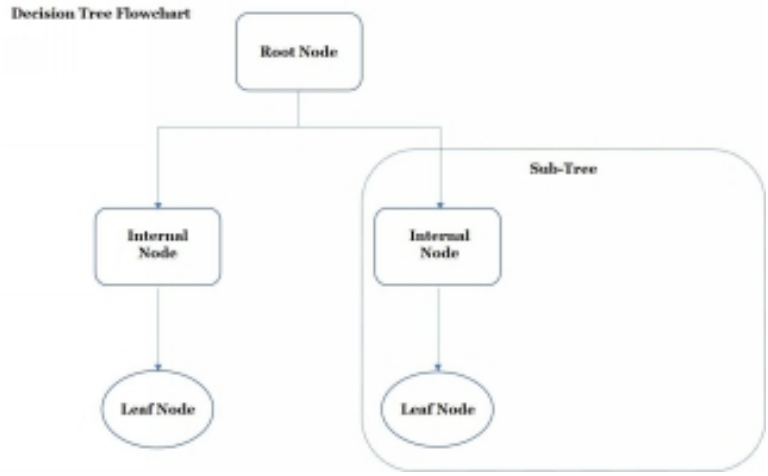
A decision tree is a flowchart with a tree-like structure and works the same way any other flowchart would.

The Components of a Decision Tree Flowchart

A decision tree also has the same sort of flowchart bits and pieces that represent the flow, a decision, and an outcome. These nodes are as follows:

- Decision node
- Branch
- Leaf node
- Sub-tree

The decision tree starts with a decision node called the root node. Each line that flows from a node is known as a branch. The decision nodes/internal nodes can branch out to be quite a few. Some may even span to a sub-tree. All the outcomes are known as the leaf node(s).



The decision tree flowchart is designed to help visualize decision making and can be used in most business practices, program designs, and or anywhere decisions are to be made.

When to Use a Decision Tree

Decision trees are good models to use when there is more than one outcome to a problem (if … then … else … situations). They are also good to use for categorization problems where there are items that need to be classified or categorized, such as animal species, plant species, and so on.

Decision tree is a popular model used in machine learning for application in industries such as energy, financial, pharmaceutical, business, manufacturing, healthcare, engineering, and more.

Decision Tree Algorithm

The Decision tree is an algorithm that mainly gets used for classification models as a supervised learning algorithm. It is therefore referred to as a Classification Tree model.

There are a few types of decision tree models including:

- Categorical Variable Decision Tree — An algorithm that has a categorical target variable.

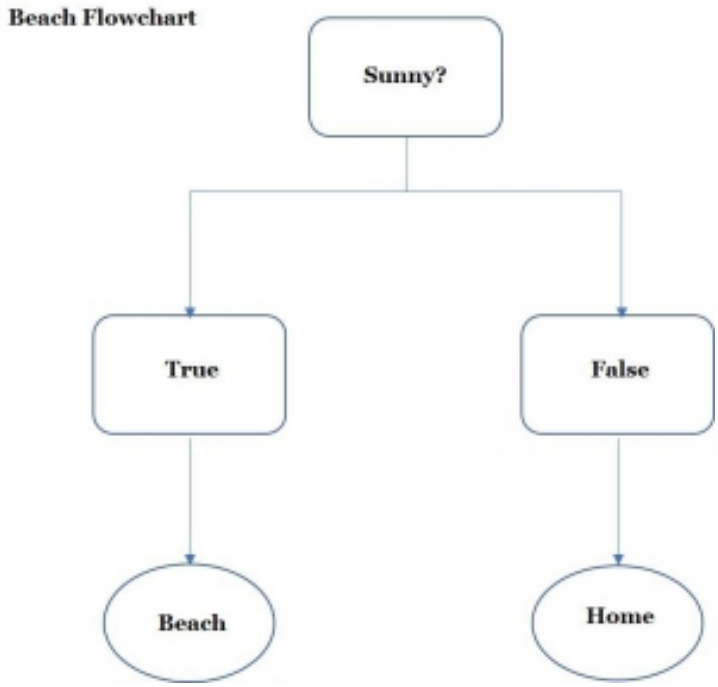
- Continuous Variable Decision Tree — An algorithm that has a continuous target variable.

Structure of a Decision Tree in Python?

There are five steps to the structure of the decision tree.

- Lay out the dataset
- Train a model
- Create decision nodes to accomplish target criteria
- Create leaf nodes as the predictions
- Achieve the targeted outcome

As an example, you will make the decision to go to the beach if it is sunny or stay at home if it is not.



In the command line prompt type the following:

```
from sklearn import tree
from sklearn.model_selection import train_test_split
import numpy as np
features = [‘sunny’]
X = [[0], [1]]
Y = [0, 1]
des = tree.DecisionTreeClassifier()
des.fit(X, Y)
 desfile = open(“dtree.dot”, ‘w’)
tree.export.graphviz(des, out_file = desfile, feature_names= filled=True, rounded=True, impurity=False, class_names= [ ‘Beach’, ‘Home’]
desfile.close()
print (des.predict([[0]]) )
```

Try the *print* statement with the [[1]], [0], and [1] predictions to view the outcomes.

Advantages of Decision Tree

The advantages of using the Decision Tree model are:

- It is easy to use and understand
- It is great for feature selection
- It is great for performing variable screening
- There is little to no data cleaning required
- There is not a lot of preparation required as compared to other models like regression tree models
- It is a lot easier when it comes to generating rules

Disadvantages/Limitations of Decision Trees

The disadvantages or limitation of the decision tree model are:

- Trees can get complicated. This is called overfitting and is a tree that does not generalize data.
- Rectangular partitioning is used for classification.
- The data can become quite large and cumbersome if not pruned.
- It is not partial to non-numerical data.

Entropy and Gini Index in Python

A node, in a Python Decision Tree algorithm, is split by calculating information gain. This split is done by Entropy and Gini Index which are measures of impurity of the data or node. An impure node is a node that has many classes associated with it, as there is a disorder in the node. A node with only a single class in it is more orderly and therefore the node is pure. Both Entropy and Gini Index are used for selection criteria in decision trees.

In most datasets, there will not be a big difference between using either Entropy or Gini Index in a decision tree model. It is, however, important to be aware of these two measures.

Entropy

Entropy measures how to reduce uncertainty in a split. It is not as easy as the Gini index as it is more computational. Entropy tends to favor splits that have a lot of unique values but smaller counts. The formula for entropy is as follows:

$$\text{Entropy} = \sum_{i=1}^{c} P_i * \log_2(P_i)$$

Gini Index

The Gini Index is used in the CART algorithm. It uses a squared proportion of classes and is calculated by minus the sum of the squared probabilities per class from 1.

The Gini formula looks as follows:

$$\text{Gini} = -\sum_{i=1}^{c} (P_i)^2$$

The Gini index is usually used for continuous attributes; it will seek out the largest class and is good at minimizing misclassification.

Chapter 8:
Regression Trees and CART

Data sets are split based on data that is of the same or similar nature in classification tree models. For instance, take a dataset that will determine if a consumer will buy a garment based on their age and gender. If 95% of people who bought the garment were women, this is where the split would be made and the top node in the tree would become women (the split would now be 95% pure). This is where the Gini Index and entropy measures come in, as they are used to quantify data's homogeneity. Classification models in terms of machine learning are great for algorithms that have scaled target features. In the Decision Tree example in Chapter 7, you used an example of going to the beach. Here you could add features such as stay at home or go to the beach; the variables were known.

What happens when there is a situation where one classification could have a number of variables?

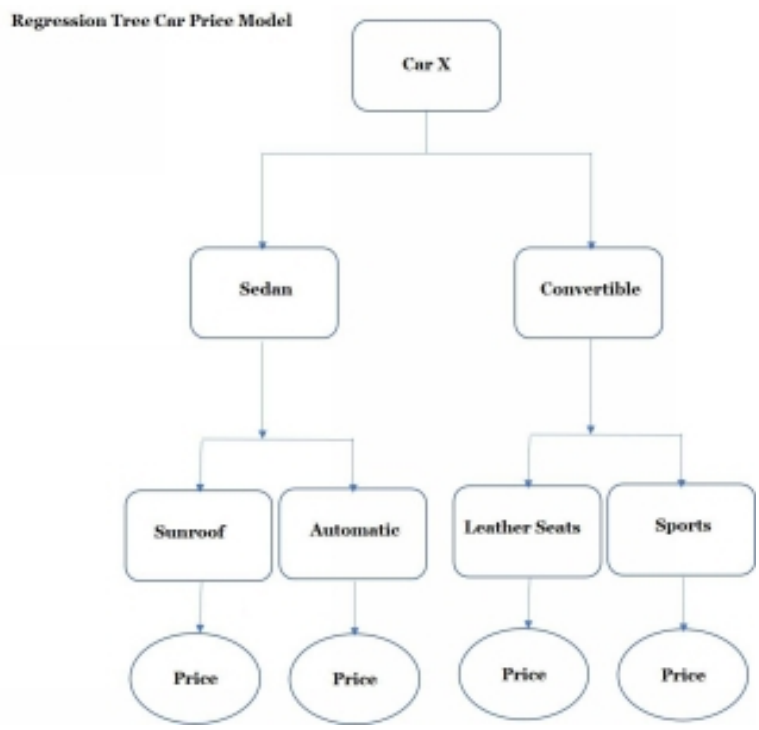
Regressions Trees

If, for instance, there is a dataset where you need to predict the price of a brand new car from the showroom floor, the decision tree model is not going to work. Most models of cars have different classes and even the different classes of model prices can differ depending upon the additional extras you can get with the vehicle. A classification tree model is not going to work in this situation, but a regression tree model will. Regression trees allow for an infinite number of continuously scaled values of the target feature.

The regression tree formula is calculated on the variance and is as follows:

$$\text{Variance} = \quad \text{Var}(x) = \frac{\Sigma_i^n = 1(yi^{-\bar{y}})}{n-1}$$

A regression tree is set up much like the decision tree, where it starts off with a root node and searches for the descriptive feature, splitting the target feature into pure values. Once it has those values the dataset is divided into the descriptive feature. This process is then repeated within the subsets of the split until there are no more splits and a leaf node is formed.



When there is a prediction type of problem, the best models to use are regression tree models.

CART Models

CART stands for Classification and Regression Tree, which is a predictive machine learning algorithm. CART is one of the oldest methodologies in machine learning and forms the foundation for algorithms such as Boosted Decision Trees, Bagged Decision Trees, and Random Forests.

Based on certain predictor variables, the CART algorithm can predict outcomes. In other words, it will show that a target variable can be predicted by a number of values. CART can handle the tasks of both regression and Classification Tree models.

Trees that are used for classification of data are known as classification models (Decision Trees), such as grouping animals by species. Trees that are used for estimation are the regression trees, such as estimating the cost of a two-door convertible with leather seats, special rims, and certain color paint.

The CART model allows you to use both classification and regression tree models; it also supports numerical target variables but it will not calculate rule sets.

For a more complex tree where you would have to first classify a species and then subsets of that species, you would use the CART tree model.

Chapter 9:
Random Forests

Most predictions have variances. For instance, you may create a model that will predict the weather for the next day. This prediction will be based on various factors which you will feed into training models such as the season, the month, the weather pattern for the past couple of years to get a historical average, etc. The model will take all the training data and look for a relationship between the features in order to predict the targeted outcome. The model learns by example, like humans do, and formulates the best questions to get to the most accurate prediction.

The tree has a target for which it learns how to map related data to, such as the weather. It may know that a sunny day is hot and the average temperatures that fall into the hot category. Unlike humans, the model has no previous knowledge of temperature and to it, it is generating outputs based on its training model that agree with data being inputted. In a classification model such as a decision tree model, it is basing its predictions on a single decision tree model. At this level, the model would not be able to answer different questions about the weather.

In order to enable the model to learn a lot more about a problem, you can use random forests to create a more diverse knowledge base. If we look at problem-solving in human terms, we know that a single human could solve multiple small problems. As the problem grows one person alone may not be able to cope. There are many variables, such as not having the scope of knowledge, they only have their point of view, and so on. But, two or more people working on the same problem brings about a whole new dimension to solving it. A think tank consists of a group of like-minded people working together, each bringing their own unique set of skills and knowledge to the table.

Look at a decision tree or regression tree model as one person working on a part of the problem they have the information, training, and knowledge about, while a random forest is a think tank of decision trees and regression tree models working together to create a more robust and diverse model. The more trees there are in the random forest, the higher the accuracy rate will be of the model. For instance, if your model was categorizing large

cats, a Random Forest of datasets with different cat species ensures more accurate, diverse data, data that is not biased by one particular dataset.

Building the Random Forest

The first step will be to create multiple Decision or Regression Trees that will make up the random forest. These trees can be made up of various decision tree algorithms such as Gigi Index, Information Gain, etc.

In the CART model, there are only single trees. In a random forest, you will grow multiple trees. Any new classification that needs to be classified is done so on various attributes. The trees already in the model will give a classification of those attributes based on the data in their dataset. The classification of the tree with the most votes will be what the new classification will fall under. In the regression model, the average of the votes is used.

Advantages of Random Forest

- The model will not be overfitted
- Higher dimensionality for large data set handling
- Random forests maintain accuracy for any missing data and are able to handle it
- Random forests can perform both regression and classification tasks

Disadvantages of Random Forest

- Random forests are better at handling classification models than they are with regression.
- Random forests are more of a black box approach as there is little to no control over what the model does.

Uses of Random Forest

- Medical applications — Medicine validation.
- Disease analytics — Identify a person's risk of getting certain diseases from their medical history.
- Banking applications — Fraud detection

- Stock markets — Identifying patterns of sock behavior as well as expected losses or gains.
- Search engines — Product identification and recommendations based on a customer's search history.
- Image classification — Computers use cameras to identify different body parts for applications such as games on the Xbox.
- Voice classification — Speech recognition and lip-reading system.

How the Random Forest Works

1. In a training set, assume that a number of cases are A.
2. A sample, with replacement, of the A cases is randomly taken.
3. For B input features or variables, b<B will be specified.
4. This allows for each nodes b variable to be randomly selected out of the B input values.
5. The node is split on the b values selected.
6. While the forest grows the value of b will be held constant.
7. There is no pruning of data, allowing for each tree in the Random Forest to grow to its fullest potential.
8. By aggregating the predictions of the b values new data can be predicted. This is where the majority of votes for classification models and average votes for regression models come in.

The test features of the random forest algorithm have to run through all the rules as laid out by each tree in the random forest. A subset of the forest will vote on what the predicted outcome should be. For instance, if the subject is a face, a few trees in the forest may return data pertaining to the face, such as a nose, an eye, or an ear. If in a forest of, say, 800 trees only 200 return information pertaining to the face, the vote will be cast out of 800. If the majority of the nodes list an eye, the outcome will be an eye. If some of the other trees have put forward an ear, nose, etc., then the higher levels of the tree can vote that the result is a face.

Ensemble Algorithm

Random forests are known as ensemble algorithms that are based on a concept that a single learner may not have all the strength needed to solve a problem. They are then known as a weak learner. But a group of weak learners banded together can become a collectively strong learner.

The same applies to classification trees. On their own they are weaker learners, but collectively they become a force of knowledge. As they share knowledge and grow the applications they are capable of becoming endless.

Ensemble Learners

Single learners consist of training and testing data.

- Training data = a learning method such as K-nearest neighbor (KNN)
- A model is formed from the training data which, if you punch in a query, you will get an outcome from.

Ensemble learners also consist of training and testing data but incorporate an ensemble of different learning methods. Each learning method is trained and tested with the same data. Each model would be individually queried to get a predicted outcome and will return their prediction; the answer with the most votes would be the answer. In regression, the mean value will be the answer.

Ensemble learners are good to use because each model you use has its own bias and variables. When you combine these methods, their biases and variables are radically reduced due to a voting (classification) or averaging system (regression).

Bagging or Bootstrap Aggregation

Bagging or bootstrap aggregation is used in regression and statistical classification. It is primarily used to improve upon an algorithm's accuracy. It does this by reducing variance and improving the stability of the model.

In the ensemble learners in the above section, we looked at training different models on the same data. Bagging or bootstrap aggregation is another way of creating ensemble learners by using the same method, with the dataset, but training each learner on a different section of the data.

To bag the data, you would need to create subsets of the data (bags filled with different sections of the data).

Data is chosen at random from the same set of data per bag with replacement. With replacement means that the random grab for data may select the same data it has already collected.

- n = The number of data instances in the dataset.
- n^1 = The number of data instances per bag. The rule of thumb for the training instances per bag should be no more than 60% of the dataset training instances.
- m = The number of bags in the ensemble of learners.

Each model is trained on the data it has randomly gathered and tested on the outcome as either voted by the models or the mean value.

Boosting

The boosting machine learning ensemble meta-algorithm is used in supervised learning to cut down on variance, reduce bias, and boost weak learners. Boosting is similar to bagging but it's used to boost ensemble learners that are not performing as they should be performing.

When using an algorithm such as Ada Boost, you set up the learning as you would to select random data (like in bagging) from the dataset. The model will get trained in the same way, the only difference will be that not just 40% or so of the dataset will be used to test the model. All of the training data is used to test the learner’s model. The outcome of this test usually pinpoints errors or inconsistencies in the testing dataset.

The next step will be to build the next bag of randomly selected data from the dataset but weighted towards the erroneous data. This means that instances in the dataset that displayed the error will be the instances that will be picked first over other instances of the data. This will continue for the amount of bagging instances needed to complete the model.

Boosting helps to fix poorly modeled data and cut down on errors in the model.

Chapter 10:
Overview of Neural Networks, Big Data, the Internet of Things (IoT), and Cloud Computing

When you are starting out with machine learning, there are a lot of terms and types of machine learning that you will come across. Although you may not go in-depth with learning them, it is still a best practice to know what they are all about. In this chapter, we will touch on a few of these topics with a brief overview of them.

Neural Networks

Neural networks have been around since the late 1950s and were modeled around the human brain by mimicking characteristics of the brain's neurons. Neural networks work by passing data through various interconnected nodes. Each node will analyze, classify, and then pass the data to the next node. There are usually more than two or three layers hidden in a neural network. Some neural networks have hundreds of hidden layers, which is why most Neural Network models fall under the category of deep learning. As you know, deep learning is a subset of both machine learning and AI.

Types of Neural Networks

There are quite a few types of neural networks, and each of them has its own unique application value and uses different methods to determine rules.

The different types of neural networks include the following list.

Convolutional Neural Network

Convolutional Neural Networks (CNN) layers are either pooled or interconnected. They can have from one to many layers. The use of a convolution operation runs on the input to make the network less parameter hungry, and it is a very deep network.

These neural networks are used in applications such as Natural Language Processing (NLP), Video Recognition, and Recommender Systems.

Feedforward Neural Network

The feedforward neural network is one of the simplest forms of neural network, wherein data only moves in one direction from input to output. It uses a classifying activation function to create a front propagated wave.

This type of neural network can have one layer or multiple hidden layers, but it has no backpropagation.

A feedforward neural network is quite easy to maintain and is usually used in datasets that contain a lot of noise. Data that contains a lot of noise refers to data that has a lot of meaningless information mixed with it. They are used in applications where the classes of targets are hard to classify, such as computer vision and face recognition.

A Single-layer Perceptron (SLP) is a type of feedforward neural network. It needs a binary target of which it can only classify cases that can be separated linearly. It is a threshold-based neural network that activates when the target reaches a predetermined threshold.

Modular Neural Network

As the name states, a modular neural network has independently performing modular networks. These networks all perform separate sub-tasks without signaling or interacting in any way with each other during computation. They work this way even though they are working towards a common goal. They're like a company with different departments that all work separately, but at the end of the day all are needed to keep the company operational. Each of the tasks performed is vital to the operation of the system.

Large complex data structures get broken down into smaller bits that speed up the computational process. Once again, you can think of it similar to that of an office situation. Different departments take the load off of one or two people trying to keep the office running efficiently. By increasing the workforce, each with their own separate tasks to do, the work is done faster and more efficiently. This is the same for large and complex data systems that need the data to be broken down so as not to create bottlenecks in the system.

Multi-layer Perceptron

The multi-layer perceptron neural network is used extensively in applications such as speech recognition or machine translation. It is a neural network that consists of three or more layers. Each node in one layer is connected to the nodes in the next layer. It is a fully connected neural network.

Nonlinear activation function is used in a multi-layer perceptron to classify non-linear data.

Radial Basis Function Neural Network

The radial basis function is used in applications for systems such as large power grids. They consist of two layers, the inner layer and the outer layer, of which the network measures the distance of any point in the network relative to the center point.

It calculates the output from the center point to the next point in order to find the shortest distance between the two points. As power stations have grown and the systems have become more complex, it is important to be able to determine the quickest way to restore power in the event of an outage.

Recurrent Neural Networks

To help predict the outcome of a recurrent neural network layer, the output of a layer is saved and fed back to the input. It uses a long short-term memory function and starts the network off with the first layer being the same as a feedforward neural network. The layers after the first layer incorporate the recurrent neural network function.

Recurrent neural networks start with front propagation, keeping the information it needs stored as memory in a cell. While it is holding the information in its memory it will continue to perform its next set of instructions. This system is a self-learning system and if it gets a wrong prediction, it will go back during back propagation and right the error.

Big Data

When computers first started out, a kilobyte of information was considered quite large. In modern computing terms, a kilobyte is a drop in the ocean compared to the huge amounts of data that are used.

The table below shows how data is measured from the smallest unit to the largest unit in existence to date.

Data Storage Unit	Capacity of the Unit	Storage Unit Abbreviation
Bit	1 or 0	b
Byte	8 bits	B

Kilobyte	1024 bytes	KB
Megabyte	1024 kilobytes	MB
Gigabyte	1024 megabytes	GB
Terabyte	1024 gigabytes	TB
Petabyte	1024 terabytes	PB
Exabyte	1024 petabytes	EB
Zettabyte	1024 exabytes	ZB
Yottabyte	1024 zettabytes	YB

Very large amounts of data are known as Big Data that is collected, processed, learned, and grows exponentially regardless of its current size.

5 Key Elements (5V's) of Big Data

Big data has five key elements that make it the huge business it is today. These five key elements are:

Volume

In the early 2000's, data available for analysis started to amass into huge volumes. Ordinary systems could not handle the copious amounts of data that were overwhelming them. That's when the term "Big Data" was coined in reference to data being too big to fit into conventional systems. Volume in the 5V's refers to the sheer volume of information available to analysts these days in a data system that continues to grow. Most Big Data systems potentially double in size every two years.

Velocity

The velocity refers to the speed at which the volume of data is produced. Thanks to real-time data applications there is a huge amount of data produced nearly every second of every day. Real-time data applications include systems such as mobile phones, RFID chips, and facial recognition systems. As real-time data has to be dealt with as soon as it is captured, it has a knock-on effect with its supporting systems such as networking bandwidth. It can also take up a lot of systems storage and processing power.

Veracity

There is also a lot of data that is captured every single day that is considered noise or of no value. The systems designed to handle Big Data have to be adept at being able to quickly, effectively, and efficiently deal with this junk data to ensure valid and useful output data.

Value

The bigger a dataset gets, the less structure it tends to have. The less structured data is, the harder a system has to work to process it. That is why normal conventional systems are not equipped to handle Big Data. As data is a valuable commodity in this day and age, Big Data has only started to get a foothold in the machine learning field.

Big Data Uses

Why is data so valuable? Because in the modern-day business world, competition is extremely fierce. Even smaller businesses have the potential to reach millions of potential customers all around the globe. With payment and delivery methods being as easy as the World Wide Web has made them, customers will support the company that pays the most attention to their needs.

Having access to large amounts of statistical data helps organizations get an edge by being able to predict the wants and needs of their target audiences. This is where Big Data and machine learning go hand in hand. Some of the applications Big Data is used for include the following.

Predicting Machinery Maintenance

Correctly analyzed data of a system can predict when a machine's parts are likely to fail or the lifespan of the part prolonged by providing various services. It can also quite accurately predict a failure in advance as an applied warning system for production lines, for example.

Predicting Market Trends

Big data has the ability to accurately predict market trends based on historical data collected from customer purchases over a certain number of months or years. Predicting what the customer wants before they know they will want is a marvelous edge to have. It also shows the customer that you were taking notice of what they wanted, need, and may like in the future.

Fraud and Compliance

One of the biggest uses for Big Data is in fraud detection. With the large amount of data, the system has a lot to base certain patterns on. As a result, it may be able to accurately predict a fraudulent transaction or one that has been tampered with. Because a computer can process this much faster than the human brain can, fraud can be dealt with in a timely manner, if not prevented.

Product Development

Based on the latest market trends, manufacturers can base new and improved products upon this. It can greatly cut down on cost and quickly show which products are worth upgrading or ending the production of.

Customer Relations

For a customer, filling out those customer satisfaction surveys can be a bother. Most will fill them in because they feel it is giving them a say, which they do have a right to do as the consumer. What this means for a business is that the data collected from this information not only helps with production but also customer interaction. In this day and age businesses strive to keep a happy customer, especially as word of mouth is now word of fingertip thanks to the ever- expanding web of social media.

Internet of Things (IoT)

Carnegie Mellon University was one of the very first institutions to implement an Internet of Things (IoT) system in the early 1980s. This system was a Coke machine that was connected to the Internet and allowed for a remote system to monitor the temperature of the machine. Being connected over the Internet also allowed the supplier of the machine to know when it ran out of stock.

The Internet of Things is found in many applications in the 21st century, from watches to appliances, cars, buildings, and even towns. Any device that is connected over the Internet is known by the term Internet of Things (IoT). If you take a look around your home right now, you are bound to find at least one device that belongs to the category of IoT. A smart device, the thermostat, and even jet engines these days have sensors on them reporting status.

Uses of the Internet of Things (IoT)

Basically, any device that has an on/off button that either works with or through the Internet is considered to be an IoT device. As we live in a high-tech world, soon there will not be many devices that are not classified as such. In a smart city, even the roads are connected through the Internet.

There are many modern-day uses for the Internet of Things applications, including the following.

Manufacturing

Thanks to the applications that support the Internet of Things, manufacturers are able to increase the productivity of new goods turning out higher volumes, faster, and with less waste. Machinery is now fitted with various devices such as sensors, identification devices, actuation monitoring, and communication. This cuts down various high-risk factors that used to plague the manufacturing industry. These devices cut down on the room for human error, safety issues, and redundant processes.

By integrating machine learning into various manufacturing processes, companies can employ statistical evaluation and predictive maintenance to maximize reliability. IoT together with ML has completely changed the way the manufacturing industry operates.

Disabled or Elderly Care

With monitoring systems installed into smart homes, it allows for a sort of assisted living giving the frail and elderly a little more independence. IoT allows for 24/7 monitoring and assistance when or where needed without having a full-time caretaker.

Transportation

When it comes to transportation, IoT connects roads, traffic control, and even cars. IoT uses smart sensors that can help with fleet maintenance. It is also useful for GPS tracking, driver anomalies while driving, vehicle problems, and can help cut down on road accidents.

For traffic control, sensors alongside and in the road can sense congestion. With this data cities can figure out if road systems need to be upgraded. Traffic lights can sense when there is more traffic backed up on the one side than the other and switch accordingly.

Smart cars are making their debut and are fully equipped with ML, AI, and IoT systems that enable them to not only drive, but adapt to the needs of its

passengers. The car learns as it drives, collecting and processing the information based on the habits of its owner or passenger.

Home or Building Automation

Home or building automation are systems in a home or building that can be monitored and controlled. These include lighting, temperature, electric, air control, fire systems, and more.

These I0T systems can help to prevent fires, carbon monoxide poisoning, and help reduce energy footprint while cutting down on household running costs.

Agriculture

IoT has helped the agricultural industry by using various sensors and weather devices. These systems allow for the collection of valuable data such as soil composition, pest infestations, temperature, humidity, wind speeds, etc. This information helps farmers to reduce crop maintenance, various risks, cut down on waste, and improve not only the quality of their products, but quantity as well.

Monitoring of the Environment

With the great emphasis on environmental issues these days, IoT helps with the capturing of information such as air quality, soil conditions, seismic activity, wildlife through habitat, and more. Some of the more sophisticated systems can save lives by being able to predict a natural disaster for early warning detection systems. These systems are vital for early evacuations and emergency response systems. As the World Wide Web spans the globe, IoT paired with ML allows for geographical scalability.

City Management

Slowly but surely, smart cities are starting to take root in certain countries around the world. With the IoT and ML working hand in hand, cities now have the ability to have greater control of things like sewage control, electrical grids, and bridges, monitor various wind conditions, and more. It can also be effective in the monitoring of rural and urban infrastructure. This is needed to ensure the infrastructure of those areas can support the growing or current population within the area.

Energy Management

There are very few electronic devices on the market today that cannot connect to the Internet. Energy management functions are built into a lot of smart houses already. These management functions control the temperature, lights, and make sure the house is energy efficient. This helps to limit the energy footprint and save on household costs.

Some smart houses can be controlled remotely or allow for the scheduling of switching on lights, heating, air conditioning, or even turn on the oven and coffee pot.

Health Care

Health care has its own Internet of Things, a system that has taken off in recent years. The Internet of Health Things (IoHT) has driven the health care system to become digitized in order to remotely monitor a person's health. These devices can be found in systems such as pacemakers and artificial limbs.

Health care systems that can be monitored remotely can cut down on dosage errors and notify healthcare providers of potential health issues. This is also valuable in helping with emergency response times, especially if the system can predict a possible fatal health issue.

IoHT, AI, and ML have combined to provide a person with a better quality of life and quicker responses to potential health-related issues.

Consumer Products

Smart devices, machine ordering devices, information robots, connected vehicles, and smartwatches are all IoT devices. As you know, consumers have become reliant on most of their smart devices for keeping track of appointments, contacts, banking information, and operations.

IoT vs Security

A big concern with the Internet of Things in any application is security. Modern-day criminals don't really have to leave the comfort of their home to rob a bank account or take control of a connected car. With the amount of personal information data that is collected from around the world on a daily basis, there is always the possibility that that information is vulnerable.

We are taught never to leave ATM slips around without tearing them up or shredding account information before binning it. But with data, once it is

out there on the World Wide Web, it is pretty hard to retract without changing your lifestyle or identity. As you have now learned, Big Data is worth a lot of money. So, it is a targeted commodity for some hackers.

In order to fill the gaps in these IoT vulnerabilities, machine learning applications are put in place. As a result, IoT devices no longer just connect to the Internet to transfer or collect information. They are getting smarter and learning to predict things. For instance, Airbus has many different sensors monitoring various aspects of the aircraft, things like cabin temperature and air pressure, engine functions, and sensors that can predict accurate outside temperatures. This information is fed to a cloud-based network and can be monitored both in the plane as well as remotely.

With the application of deep learning integration, IoT vulnerabilities can be overcome, with fraud detection and unusual activities picked up easily by deep learning algorithms. NASA uses IoT and ML integration for systems like ZigBee and Xbee to monitor the performance function of its Exo-Brake devices. NASA uses extremely modified sensors along with ML to help increase the lifespan of the rovers and make them more effective. NASA has been working on implementing ML and IoT in their Lunar Lander project which uses the limitless amounts of solar power available in space.

The military uses IoT and ML integration for use with drones that collect data on various military operations. Thanks to each drone's unique ML algorithm, the data the drone collects when it flies is automatically sifted through to determine if it is relevant or not.

As IoT devices collect huge volumes of data on a daily basis, ML analytics are needed to sort through the data to sift out data that is not needed.

Cloud-Based Machine Learning

With Big Data on the rise and companies needing more processing power or moving to integrated systems to cut down on costs, more and more are moving to cloud-based environments. Cloud-based systems allow organizations the freedom to test out various infrastructure, software applications, and give them a cheaper alternative for creating in-house systems.

Amazon AWS offers packages that can offer on-demand resources that are scalable to the customer’s needs as and when they require them. This means

that should the customer need a lot more processing power or storage for a limited time, they only pay for the space and length of use. There are many different companies offering cloud-based packages such as Google Cloud Platform and Microsoft Azure.

So why slot cloud-based systems into this guide? Because cloud-based systems are not only beneficial to large organizations, but to smaller ones and individuals too. Machine learning programming languages like Python may be open-source systems along with most of their libraries, but they can still take a lot of processing power and storage. Because of scalable on-the-fly resources offered by cloud-based systems providers, resources that may otherwise not be an option due to costs or accessibility are made available.

As an individual starting out with machine learning, your current system may be able to handle the smaller projects, but as you grow you will find your system's resources may become limited. Finding a cloud-based system that is both affordable and suitable for your needs will be highly beneficial to your machine learning training and growth.

Conclusion

Hopefully, you found this book useful in helping you get started with Python machine learning. You should now have the foundational tools you need to help you move onto the more intermediary Python machine learning.

Some key points to remember when starting out with Python machine learning are:

- Learn all the buzzwords and make sure you understand them
- Have a good grasp of basic Python programming
- If you are a beginner, work through the basics of Python and learn the functions, libraries, and commands
- Try to write a small Python program using the supporting platforms like Anaconda or IDLE
- Understand the basic concepts of machine learning as laid out in this guide
- Practice the exercises and start to add to them the more confident you feel
- Look for various YouTube videos to expand your Python machine learning knowledge
- The Internet is a valuable source of information and has plenty of Python machine learning support groups to help you or bounce ideas around with.

As with any technology environment, you should strive to join groups that keep you updated and informed on breakthrough technologies within the field. Read data science magazines and articles. Sign up to popular data science newsletters from reliable sources such as TechRepublic, Data Science Magazine, and sites like Medium.com.

Some websites with newsletters to follow or podcasts of interest to keep yourself updated on the latest AI or ML developments and news include:

- AI Weekly — http://aiweekly.co/ — This site is a good site for newsletters and has some very interesting blogs that are updated regularly and added to.

- AI News — https://artificialintelligence-news.com/ — AINEWS is another website with regularly updated articles, blogs, and newsletters.
- Talking Machines — http://www.thetalkingmachines.com/ — This is a website for podcasts on the latest in AI advancements. There are always a lot of interesting talks, shows, debates, and more.
- Concerning AI — https://concerning.ai/ — This is a website for podcasts on the innovations, advancements, and potential for AI.
- O'Reilly Data Show Podcast — https://www.oreilly.com/radar/topics/oreilly-data-show-podcast/ — A podcast that explores the drive behind Big Data, exploration into AI, and data science.
- TWIML — https://twimlai.com/shows/ — This podcast site is a top site for those wanting to keep abreast of all the latest AI and ML trends. It has a variety of hosts to choose from that discuss all sorts of topics, not only those relating to AI or ML.

Thank you for buying this book and if you are reading the conclusion, congratulations on working your way through it! Keep practicing, as the more you do, the easier machine learning will become, and you will be programming your own ML algorithm in no time.

Good luck on your Python machine learning journey, and if you found this book helpful, please let us know. It is always great to hear from our readers!

References

A Beginner's Guide to Classification and Regression Trees. (n.d.). Digital Vidya. https://www.digitalvidya.com/blog/classification-and-regression-trees/#:~:text=In%20a%20regression%20tree%2C%20a,Squared%20Errors%E2%80%9D(SSE) .

A Comprehensive Guide to Types of Neural Networks. (n.d.). Digital Vidya. https://www.digitalvidya.com/blog/types-of-neural-networks/

Adkoli, V. (2019, October 5). *Machine Learning (ML) and IoT can Work Together to Improve Lives* . OpenSource. https://www.opensourceforu.com/2019/10/machine-learning-ml-and-iot-can-work-together-to-improve-lives/#:~:text=Machine%20Learning%20(ML)%20and%20IoT%20can%20Work%20Together%20to%20Improve%20Lives,-By&text=They%20collect%20huge%20amounts%20of,and%20make%20them%20more%20efficient .

All images in this book were of the author's own making, including screenshots of any coding.

Anaconda Individual Edition. (n.d.). Anaconda. https://www.anaconda.com/products/individual

Advantages and Disadvantages of Python Programming Language. (2017, April 24). Medium. https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121

Beklemysheva, A. (n.d.). *Why Use Python for AI and Machine Learning?* Steel Kiwi. https://steelkiwi.com/blog/python-for-ai-and-machine-learning/#:~:text=Benefits%20that%20make%20Python%20the,overall%20popularity%20of%20the%20language .

Brownlee, J. (2019, February 10). *Your First Machine Learning Project in Python Step-By-Step* . Machine Learning Mastery. https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Copeland, B.J. (2020, August 11). *Artificial Intelligence* . Britannica. https://www.britannica.com/technology/artificial-intelligence

History of Python. (n.d.). Python Course. https://www.python-course.eu/python3_history_and_philosophy.php

How to Import a CSV File into Python using Pandas. (2019, December 21). Data to Fish. https://datatofish.com/import-csv-file-python-using-pandas/

Intro to Data Structures. (n.d.). Pandas. https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html

Iris Flower Data Set. (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Iris_flower_data_set

Iris Species. (n.d.). kaggle. https://www.kaggle.com/uciml/iris

Khareem Sudlow. (2019, August 3). *Your First Machine Learning Project in Python Step-By-Step - Start Machine Learning in Python?* [Video]. YouTube. https://www.youtube.com/watch?v=_J-U7jcycLI

Koehrsen, W. (2017, December 27). *Random Forest Simple Explanation* . Medium. https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

Machine Learning Random Forests. (n.d.). Python Machine Learning Tutorial. https://www.python-course.eu/Random_Forests.php

Modern Machine Learning Algorithms: Strengths and Weaknesses. (n.d.). https://elitedatascience.com/machine-learning-algorithms#regression

Predict Diabetes From Medical Records. (n.d.). kaggle. https://www.kaggle.com/paultimothymooney/predict-diabetes-with-python-starter-kernel

Python Language advantages and applications . (n.d.). GeeksforGeeks. https://www.geeksforgeeks.org/python- language-advantages-applications/

Python Pandas - Descriptive Statistics. (n.d.). Tutorialspoint. https://www.tutorialspoint.com/python_pandas/python_pandas_descriptive_statistics.htm

Python Pandas - Function Application. (n.d.). Tutorialspoint. https://www.tutorialspoint.com/python_pandas/python_pandas_function_application.htm

Regression Trees. (n.d.). Python Machine Learning Tutorial. https://www.python-course.eu/Regression_Trees.php

SciPy Installation. (n.d.). SciPy. https://www.scipy.org/install.html

The Iris Dataset . (n.d.). GitHub. https://gist.github.com/curran/a08a1080b88344b0c8a7#file-iris-csv

The Python Community . (n.d.). Python. https://www.python.org/community-landing/

What are Decision Trees? (n.d.). Python Machine Learning Tutorial. https://www.python-course.eu/Decision_Trees.php