**Introduction to Data Science in Python**

**Chapter 1**

**What is Data Science?**

Data science is an interdisciplinary method to analyze and find complicated patterns in data and extract meaningful insights. A data science project often entails the following steps:

* **Problem Identification** **-** Identifying the problem that needs to be solved.
* **Data Collection** **-** Obtaining or extracting relevant data.
* **Data Processing -** Data analysis, visualization, and preparation.
* **Model Development –** Building a model to detect patterns.
* **Model Assessment -** Evaluating and improving the performance of a model.
* **Insights Communication –** Presenting the gained insights and findings.
* **Model Deployment -** Putting a model in place and keeping it up to date

As its name implies, you need data to solve problems through data science. But it is far more vital to have a clear business challenge to solve first. A project may produce inaccurate findings if the problem is not framed correctly, as you may have used incorrect data, improperly prepared the data, or caused a model to discover the wrong patterns. As a result, properly defining the scope and aim of a data science project with your stakeholders is crucial.

**Applications of Data Science:**

Data science has a wide range of applications in both real-world and corporate settings. A few examples are as follows:

* **Recommender Systems:**

One of the essential applications of Data Science is targeted recommender systems. Whatever a user searches for on the internet, they will find many posts. Let us understand this through an example. A user wants to buy a laptop, they do a Google search for it before buying it offline. Companies that pay for laptop advertisements can benefit from data science. So they will see the recommendation of that particular laptop that they were searching online, in social media, on websites, and in apps. As a result, they will be forced to shop online.

* **Health Care:**

Data science is a blessing in the healthcare industry. The following are some of the applications of data science:

* + Tumor detection.
  + Drug development.
  + Image analysis in medicine.
  + Medical ChatBots.
  + Genetics and Genomics.
  + Predictive Modeling for Diagnosis.
* **Finance**

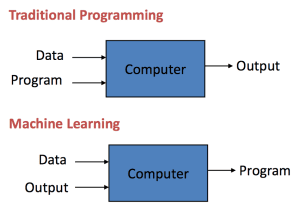
In the financial industry, data science is crucial. The financial sector has long had a problem with fraud and the possibility of losing money. As a result, Financial Industries must automate risk of loss analysis to make strategic decisions for the organization. Financial Industries also employ Data Science Analytics techniques to forecast the future. It enables businesses to predict client lifetime value and stock market movements.

**What Is Machine Learning?**

Machine learning is a subset of Artificial Intelligence that allows computers to learn from past experiences without being explicitly programmed. Contrary to classical programming, in which the computer uses a predefined set of rules/algorithms/programs to generate results, the Machine Learning approach uses Data to learn rules against which results are generated.

***Classical Programming: (Input + Set of Rules = Result)***

***Machine Learning: (Data+ Outputs = Set of Rules + Input = Results)***



Fundamentally, Machine Learning consists of the following three types of learnings:

* Supervised Learning
* Unsupervised Learning
* Reinforcement Learning.

**Supervised Learning:**

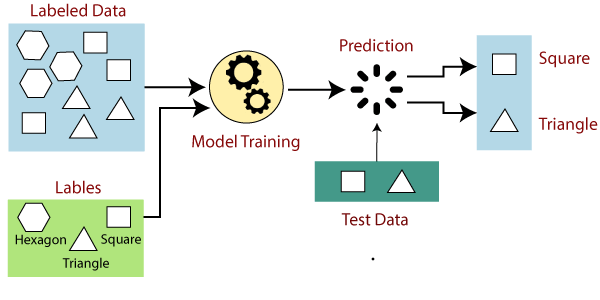
Supervised machine learning technique uses labeled data, i.e., data with known answers, to train algorithms for predicting future events.

Supervised learning is mainly used for:

* Regression problems (predicting a value outcome).
* Classification problems

**How does it work?**

The computer learns a function to predict output values from the data analysis. After sufficient training, the algorithm can predict targets for any unseen input. The learning algorithm can also compare its output with the correct, intended output and find errors to modify the model accordingly, commonly known as validation.



Supervised learning can be used to classify data like:

1. What is spam in an e-mail, based on known spam examples
2. What kind of videos do you like, based on videos you have played.

**Unsupervised Learning:**

Unsupervised learning is an ML technique that uses neither classified nor labeled data. This technique allows the algorithm to act on the information without any guidance. Here the goal is to group unsorted data according to similarities, patterns, and differences without any prior training of the model or determine how the Data is distributed in the space, known as density estimation.

In a nutshell, for an n-sampled space X1 to Xn (n training examples), true class labels are not provided for each sample, hence known as learning without a teacher(unsupervised learning).

Unsupervised learning is mainly used for:

* **Clustering** is a machine learning technique used for grouping unlabeled data based on their similarities or differences.
* **Association** is another machine learning technique that uses different rules to find relationships between variables in a given dataset.
* **Dimensionality** reduction is an unsupervised learning technique used when the number of features (or dimensions) in a given dataset is too high. It reduces the number of data inputs to a manageable size while also preserving the data integrity so that the training can be computationally less expensive.

To conclude, unsupervised learning is about creating computer algorithms that can improve themselves. It is expected that machine learning will eventually shift to unsupervised learning to allow programmers to solve problems without even creating models.

Diagram

Description automatically generated

**Reinforcement Learning:**

Reinforcement learning is a machine learning technique that rewards desirable behaviors while penalizing undesirable ones. A reinforcement learning agent can perceive and comprehend its surroundings, act, and learn through trial and error in general.

For example, in the game of chess, the model will learn to protect its most essential pieces while killing the opponents after multiple rounds of training in which it receives high penalties every time it loses.

***Note: Reinforcement Learning is out of scope and will not be covered in this course.***

**Overview of Python:**

Python is one of the most popular programming languages for data science. But, before we get into Python's data science applications, let's go over some fundamental Python principles.

**Types of Variable:**

Variables are containers that we use in Python to store data. In Python, you can handle and manipulate different types of variables. Each has its specificities and benefits. Let's look at some of the variables that we will be using during this course:

**Numeric Variables:**

The most basic type of variable is the numeric variable. These variables are used to store integers and decimal numbers. You can perform all basic mathematical operations such as addition, subtraction, multiplication, and division on these variables.

To declare a numeric variable in Python, you can use the following code:

numeric\_variable\_1 **=** 50

numeric\_variable\_2 **=** 60.5

You can perform mathematical operations on these variables by adding the operator between the two variables:

numeric\_variable\_1 **+** numeric\_variable\_2

You will get the following output on your Google Colab notebook:

**110.5**

**Text Variables:**

To store text values in your Python code, you can use the text variables. The declaration of these variables is precisely similar to the declaration of numeric variables.

var1 **=** 'I love'

var2 **=** "Data Science"

You can use single quotes to declare your string(another name for text variable) or double quotes. Both work equally well.

To display the contents of your variables, you can use the print() function.

**print(**var1**)**

**print(**var2**)**

Running these functions in your colab notebook should give you the following output:

**I love**

**Data Science**

You can use string interpolation to print variables with text. Python provides a method known as the f strings to do that. To print variables with text, you can use the following code:

**print(**f"My variables have the following values: {var1} {var2}!"**)**

This code will produce the following output:

**"My variables have the following values: I love Data Science !"**

You can also use string concatenation to concatenate two string-type variables. To add two strings, you can use the following code:

**print(**var1 **+** var2**)**

This code will produce the following output:

**I love Data Science**

**Python List:**

If you want multiple values in the same container/ variable, you can use the Python list.

To declare a list, you need to use the square brackets [] like this:

my\_list **=** **[**'Hello' **,** 'World'**]**

**print(**my\_list**)**

You will get the following output:

**['Hello', 'World']**

A Python list can have multiple data types in it. Say if you want both the text and numeric values inside a list, you can do that as follows:

multi\_list **=** **[**'I love'**,** 2022**,** 1689**,** 'Data Science'**]**

**print(**multi\_list**)**

You will get the following output:

**['I love', 2022, 1689, 'Data Science']**

You can also access a particular value in the list. To do this, you need to access the list through its index. The list index starts from 0 and can be used as follows:

**print(**multi\_list**[**0**])**

**print(**multi\_list**[**2**])**

You will get the following output:

**I love**

**1689**

Also, to access a particular range of a list, you can use the **:** operator. Just specify the starting index on the left side of the operator and the last index on the right side of the operator. For example:

multi\_list**[**0**:**2**]**

Would give you:

**['I love', 2022, 1689]**

You can also iterate through the items in a list using the ***for*** loop. For printing every element in the multi\_list one by one, you can use the following code:

**for** item **in** multi\_list**:**

**print** **(**item**)**

This will give you the following output:

**'I love'**

**2022**

**1689**

**'Data Science'**

To add an item to the list, you can use the append function.

multi\_list**.**append**(**6**)**

**print(**multi\_list**)**

Would give you the following output:

**['I love', 2022, 1689, 'Data Science' , 6]**

You can use the remove function to remove an item from the list.

multi\_list**.**remove**(**1689**)**

**print(**multi\_list**)**

Would give you the following output:

**['I love,' 2022, 'Data Science,' 6]**

**Python Dictionary:**

One of the most popular data types in Python are the dictionaries. Similar to the list, a dictionary is used to store multiple values. A key-value pair organizes all of these values stored in the dictionary. To simply put, a dictionary is not indexed by numbers. Instead, it is indexed by keys. So to access a specific value, you need to call it by its key.

You can declare a dictionary using the following code:

dict **=** **{**'Learning'**:** 'Data Science'**,** 'Language'**:** 'Python'**}**

**print(**dict**)**

Would give you the following output:

**{'Learning': 'Data Science', 'Language': 'Python'}**

To access a specific value, you need a key:

For example;

dict**[**'Language'**]**

would yield in :

**Python**

To access all the keys in the dictionary, you can use the keys function:

dict**.**keys**()**

Would result in:

**dict\_keys['Learning', 'Language']**

Similarly, to access all the values, you can use the values function:

dict**.**values**()**

Would result in:

**dict\_values['Data Science’, 'Python']**

You can also iterate through the a for loop:

**for** key**,** value **in** dict**.**items**():**

**print(**key**)**

**print(**value**)**

Would give you:

**'Learning'**

**'Data Science'**

**'Language'**

**'Python'**

**Exercise 1.01:**

We have gone through the basic data structures required for Data Science Algorithms. Let us do a hands-on exercise to practice these concepts:

In this exercise, we will be creating a dictionary that will contain machine learning algorithms.

Open the following link to get started:

***https://bit.ly/35uVTrk***

**Python for Data Science:**

In this section we will learn about the packages that we have to use with Python for our data science applications. The two packages we will be learning about are:

* Pandas
* Scikit-Learn

**The PANDAS Package:**

The pandas package provides an enormous variety of APIs for manipulating data structures. The two major data structures defined in the pandas package are DataFrame and Series

**DataFrame:**

A DataFrame is a two-dimensional table that represents a tabular data structure. Rows, columns, indexes, and cells make up the structure. It's fairly comparable to an Excel sheet or a database table.

You can create a dataframe using the following code:

**import** pandas **as** pd

data **=** **{**

"algorithm"**:** **[**'Linear Regression'**,** 'K-Means'**],**

"type"**:** **[**'Supervised Learning' **,** 'Unsupervised Learning'**]**

**}**

#load data into a DataFrame object:

df **=** pd**.**DataFrame**(**data**)**

**print(**df**)**

You will get the following output:

Graphical user interface, text

Description automatically generated

Each Key is a coulnm and the elements inside the value are the rows.

The DataFrame class in pandas represents a DataFrame. A pandas DataFrame is made up of 1-dimensional arrays called pandas Series. In a DataFrame, a pandas Series is essentially a single column.

Now, the data can come from many different sources. Some of the most common sources are:

1. CSV file
2. Excel Spreadsheets
3. JSON and etc.

We will look into CSV files first.

**CSV:**

CSV stands for Comma Separated Values. These files use a comma (,) to separate colunms and newlines. The previous example of the dataframe woud look like this in a CSV file.

algorithm, type

Linear Regression, K-Means

Supervised Learning, Unsupervised Learning

To use a CSV file for your data science project, you need to import the file using the read\_csv() method from the pandas package.

This can implemented as:

pd**.**read\_csv**(**'https://raw.githubusercontent.com/fenago/pythonml/main/Chapter%201/Social\_Network\_Ads.csv'**)**

This will give you the following results:

Table

Description automatically generated

**Excel Spreadsheets:**

Excel is a Microsoft program that is widely used in the industry. It has its internal structure for storing information like the data type of each cell and even Excel formulae. Excel files can easily be imported using the read\_excel() method.

pd**.**read\_excel**(**'file\_name\_here.xlsx'**)**

**JSON:**

JSON is a widely used file format primarily to transfer data from online APIs. It has a similar structure to a Python dictionary with key-value pairs. In JSON format, the above example DataFrame would look like this:

**{**

"algorithm"**:** **[**

"Linear Regression"**,**

"K-Means"

**],**

"type"**:** **[**

"Supervised Learning"**,**

"Unsupervised Learning"

**]**

**}**

To read a JSON file, you can use the read\_json() method from the pandas package.

pd**.**read\_json**(**'file\_name\_here.json'**)**

**Exercise 1.02:**

We have gone through the essential files that we have to deal with while working on a Data Science project. Let us do a hands-on exercise to practice these concepts:

In this exercise, we will be loading CSV, TSV, and XLSX files into a Pandas Dataframe.

Open the following link to get started:

[**https://bit.ly/3IupU94**](https://bit.ly/3IupU94)

**The Scikit-Learn package:**

Another widely used tool among data scientists is Scikit-learn (commonly known as sklearn). The primary goal of sklearn is to provide APIs for data processing and machine learning algorithm training.

**The Sklearn API.**

Sklearn package has builtin a substantial amount of machine learning algorithms, such as linear regression, logistic regression, K-nearest neighbor, K-means clustering, decision trees, and random forests.

The algorithms in sklearn are grouped based on a family. For example:

Random forest and GradientBoosting are part of the ensemble module.

You can access these algoritms by importing the Sklearn package as follows:

**from** sklearn**.**ensemble **import** RandomForestClassifier

Another advantage of the Sklearn API is that all the algorithms follow the same API structure. As a result, once you've figured out how to train one algorithm, it's straightforward to train another with only minor code modifications. To train a machine learning model with sklearn, follow these four steps:

1. **Model Creation -** Create a model with the hyperparameters you wish to use: this will set up the machine learning model you want to use.
2. **Model Training -** Train the model with training data: the model will learn the optimal parameters to generate predictions near the target's actual values as feasible during this stage.
3. **Prediction -** Predict the result from input data: the model will use the learned parameter to predict the outcome for new data.
4. **Accuracy Calculation -** Evaluate the model's prediction accuracy: to see if the model learned the correct patterns for reliable predictions.

To instatiate a model, you need to set its hyper parameters. It is recommended to set the random\_state hyperparameter in order to get reproducible results every time you run the code.

rf\_model **=** RandomForestClassifier**(**random\_state**=**1**)**

After the instantiation, you need to train the model with the data.

rf\_model**.**fit**(**features**,** target**)**

Once the model is trained, you now need to predict the values using an input. To do that use the following code:

preds **=** rf\_model**.**predict**(**features**)**

After getting the predictions, we now need to evaluate the accuracy of the model. To do that you can use the following code:

**from** sklearn**.**metrics **import** accuracy\_score

accuracy\_score**(**target**,** preds**)**

**Exercise 1.03:**

Now we have seen how we can develop our models using the Sklearn package.

Now we will load a dataset from GitHub and apply a logistic regression model to it. We will be predicting whether a patient will have a stroke depending on the input features.

Open the following link to get started:

**https://bit.ly/35a2RlM**

**Summary:**

This chapter gave you a broad idea of what data science entails. We also learnt about supervised and unsupervised machine learning methods, as well as regression and classification algorithms. We got a basic overview of Python and how to work with the key data structures (lists and dictionaries) that will be utilised throughout the course.

Then we went through what a DataFrame is and how to make one using the well-known pandas tool to import data from various file types. Finally, we learnt how to train a machine learning model and generate predictions using the sklearn package.

This has just been a brief introduction to the intriguing field of data science. This book will teach you a lot more and introduce you to new approaches for managing data science projects from start to finish.

The next chapter demonstrates how to run regression on a real-world dataset