INDUSTRIAL TRAINING DAILY DIARY
DAY 15

11 July, 2025

Topic : Basics of Machine Learning, with examples

Objectives:

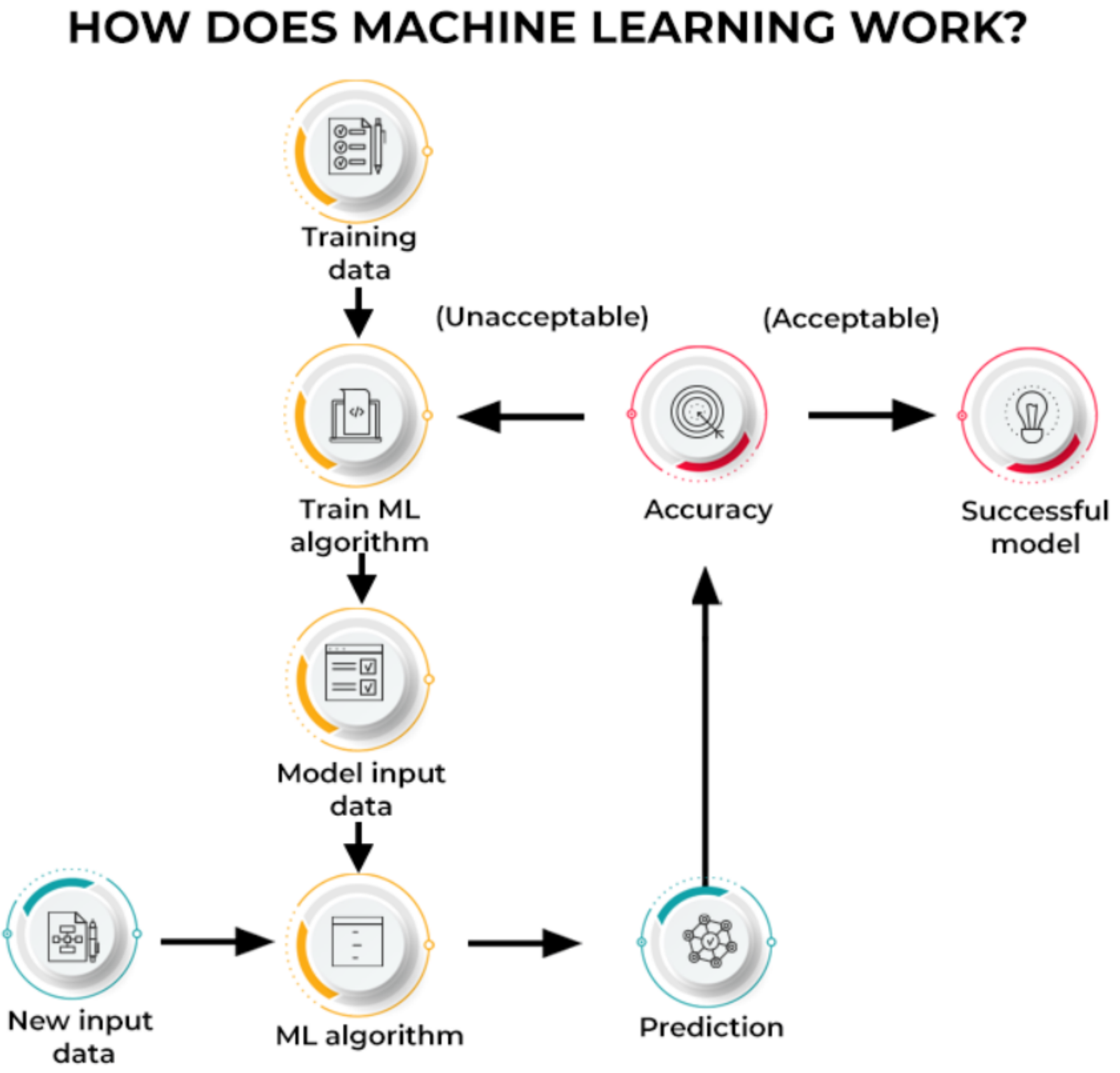
- To understand what Machine Learning is and how it differs from traditional programming.
- To explore the key components of a machine learning system: data, model, algorithm, and evaluation.
- To learn about different types of Machine Learning:
    - Supervised Learning
    - Unsupervised Learning
    - Reinforcement Learning
- To study real-life applications of Machine Learning in various fields like healthcare, finance, and recommendation systems.
- To implement basic examples of supervised learning (e.g., linear regression or classification).
- To understand the process of training a model using datasets and evaluating its performance.
- To gain hands-on experience by writing and running simple machine learning code using Python.

What Is Machine Learning?

Machine learning (ML) is defined as a discipline of artificial intelligence (AI) that provides machines the ability to automatically learn from data and past experiences to identify patterns and make predictions with minimal human intervention.

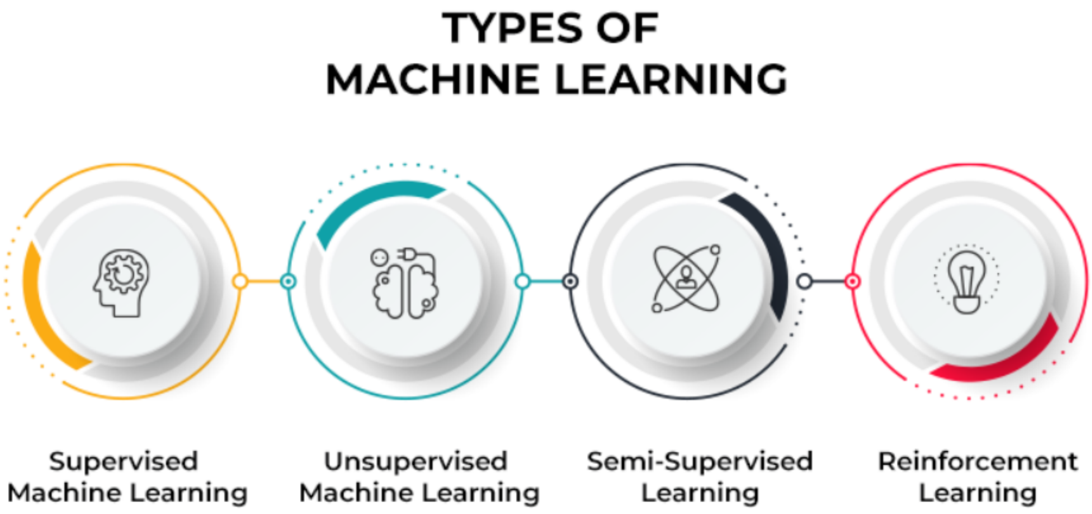
How does machine learning work?

Machine learning algorithms are molded on a training dataset to create a model. As new input data is introduced to the trained ML algorithm, it uses the developed model to make a prediction.



Types of Machine Learning

Machine learning algorithms can be trained in many ways, with each method having its pros and cons. Based on these methods and ways of learning, machine learning is broadly categorized into four main types:



1. Supervised machine learning

This type of ML involves supervision, where machines are trained on labeled datasets and enabled to predict outputs based on the provided training. The labeled dataset specifies that some input and output parameters are already mapped. Hence, the machine is trained with the input and corresponding output. A device is made to predict the outcome using the test dataset in subsequent phases.

Supervised machine learning is further classified into two broad categories:

•**Classification**: These refer to algorithms that address classification problems where the output variable is categorical; for example, yes or no, true or false, male or female, etc. Real-world applications of this category are evident in spam detection and email filtering.
Some known classification algorithms include the Random Forest Algorithm, Decision Tree Algorithm, Logistic Regression Algorithm, and Support Vector Machine Algorithm.
•**Regression**: Regression algorithms handle regression problems where input and output variables have a linear relationship. These are known to predict continuous output variables. Examples include weather prediction, market trend analysis, etc.

Popular regression algorithms include the Simple Linear Regression Algorithm, Multivariate Regression Algorithm, Decision Tree Algorithm, and Lasso Regression.

2. Unsupervised machine learning

Unsupervised learning refers to a learning technique that's devoid of supervision. Here, the machine is trained using an unlabeled dataset and is enabled to predict the output without any supervision. An unsupervised learning algorithm aims to group the unsorted dataset based on the input's similarities, differences, and patterns.

Unsupervised machine learning is further classified into two types:

•**Clustering**: The clustering technique refers to grouping objects into clusters based on parameters such as similarities or differences between objects. For example, grouping customers by the products they purchase.
Some known clustering algorithms include the K-Means Clustering Algorithm, Mean-Shift Algorithm, DBSCAN Algorithm, Principal Component Analysis, and Independent Component Analysis.

•**Association:** Association learning refers to identifying typical relations between the variables of a large dataset. It determines the dependency of various data items and maps associated variables. Typical applications include web usage mining and market data analysis.
Popular algorithms obeying association rules include the Apriori Algorithm, Eclat Algorithm, and FP-Growth Algorithm.

3. Semi-supervised learning

Semi-supervised learning comprises characteristics of both supervised and unsupervised machine learning. It uses the combination of labeled and unlabeled datasets to train its algorithms. Using both types of datasets, semi-supervised learning overcomes the drawbacks of the options mentioned above. Consider an example of a college student. A student learning a concept under a teacher's supervision in college is termed supervised learning. In unsupervised learning, a student self-learns the same concept at home without a teacher’s guidance. Meanwhile, a student revising the concept after learning under the direction of a teacher in college is a semi-supervised form of learning.

4. Reinforcement learning

Reinforcement learning is a feedback-based process. Here, the AI component automatically takes stock of its surroundings by the hit & trial method, takes action, learns from experiences, and improves performance. The component is rewarded for each good action and penalized for every wrong move. Thus, the reinforcement learning component aims to maximize the rewards by performing good actions.

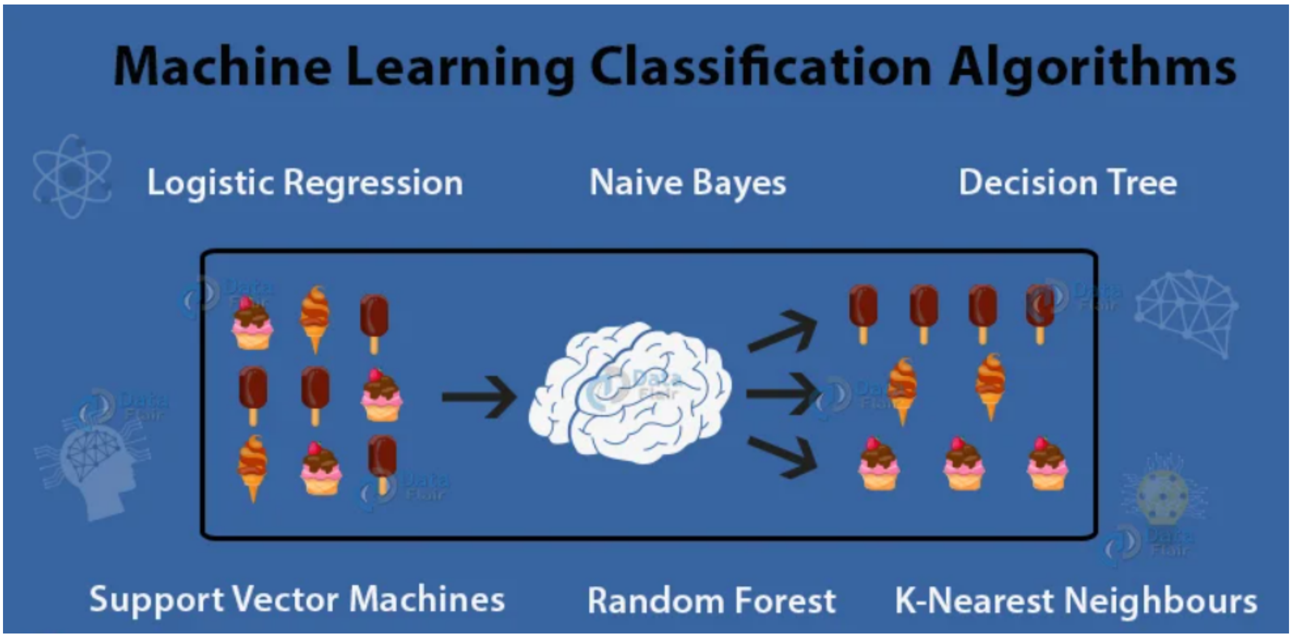
Unlike supervised learning, reinforcement learning lacks labeled data, and the agents learn via experiences only. Consider video games. Here, the game specifies the environment, and each move of the reinforcement agent defines its state. The agent is entitled to receive feedback via punishment and rewards, thereby affecting the overall game score. The ultimate goal of the agent is to achieve a high score.

Reinforcement learning is applied across different fields such as game theory, information theory, and multi-agent systems. Reinforcement learning is further divided into two types of methods or algorithms:

- **Positive reinforcement learning**: This refers to adding a reinforcing stimulus after a specific behavior of the agent, which makes it more likely that the behavior may occur again in the future, e.g., adding a reward after a behavior.
- **Negative reinforcement learning**: Negative reinforcement learning refers to strengthening a specific behavior that avoids a negative outcome.

Machine Learning Classification Algorithms

Classification is one of the most important aspects of **supervised learning**.

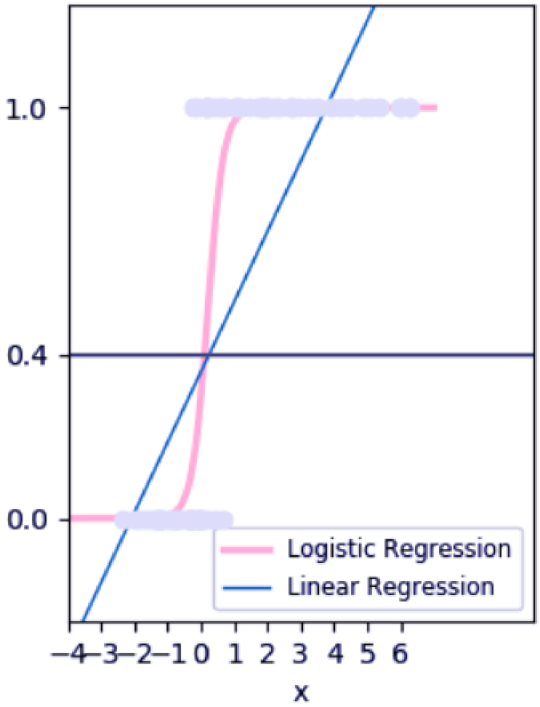


1. Logistic Regression Algorithm

Logistic regression may be a **supervised learning** classification algorithm wont to **predict the probability** of a target variable. It's one among the only ML algorithms which will be used for various classification problems like **spam detection**, **Diabetes prediction**, **cancer detection** etc.

We use logistic regression for the binary classification of **data-points**. We perform categorical classification such that an output belongs to either of the **two classes (1 or 0)**.
For example – we can predict whether it will rain today or not, based on the **current weather conditions**.
Two of the important parts of logistic regression are **Hypothesis and Sigmoid Curve**. With the help of this hypothesis, we can derive the **likelihood** of the event.
The data generated from this hypothesis can fit into the **log function** that creates an **S-shaped curv**e known as "**sigmoid"**. Using this **log function**, we can further predict the **category of class**.

We can represent the sigmoid as follows:



The produced graph is through this logistic function:

1 / (1 + e^-x)

The '**e'** in the above equation represents the **S-shaped curve** that has values between **0 and 1**.

We write the equation for **logistic regression** as follows:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

In the above equation, **b0** and **b1** are the **two coefficients** of the **input x**. We estimate these two coefficients using "**maximum likelihood estimation"**.

2. Naïve Bayes Algorithm

Naïve Bayes algorithm may be a supervised learning algorithm, which is predicated on **Bayes theorem** and used for **solving classification problems**. It's not one algorithm but a **family of algorithms** where all of them share a standard principle, i.e. every pair of features being classified is **independent** of every other.

Naïve Bayes Classifier is one among the **straightforward** and **best Classification** algorithms which helps in building the **fast machine learning** models which will make **quick predictions**.

Naive Bayes is one of the **powerful machine learning** algorithms that is used for classification. It is an **extension** of the Bayes theorem wherein each feature assumes **independence**. It is used for a variety of tasks such as **spam filtering** and other areas of **text classification**.

Naive Bayes algorithm is useful for:

- It is an **easy** and **quick way** to **predict** the **class** of the **dataset.** Using this, one can perform a **multi-class prediction**.
- When the assumption of **independence** is **valid,** Naive Bayes is much more **capable** than the other algorithms like **logistic regression**.
- Furthermore, you will require **less training data**.

Naive Bayes however, suffers from the following drawbacks:

- If the **categorical variable** belongs to a category that wasn't followed up in the **training set**, then the model will give it a probability of **0** which will **inhibit** it from **making any prediction**.
- Naive Bayes assumes independence between its **features**. In real life, it is difficult to gather data that involves completely **independent features**.

It still has some shortcomings though. If the categorical variable falls in any of the categories that the model was not trained on then it will assign this feature a probability of zero which will limit the ability of the model to make predictions . Furthermore, Naive Bayes work under the premise of features independence, which is rarely true in the real world datasets.

3. Decision Tree Algorithm

Decision Tree algorithms are used for **both predictions** as well as **classification** in machine learning.

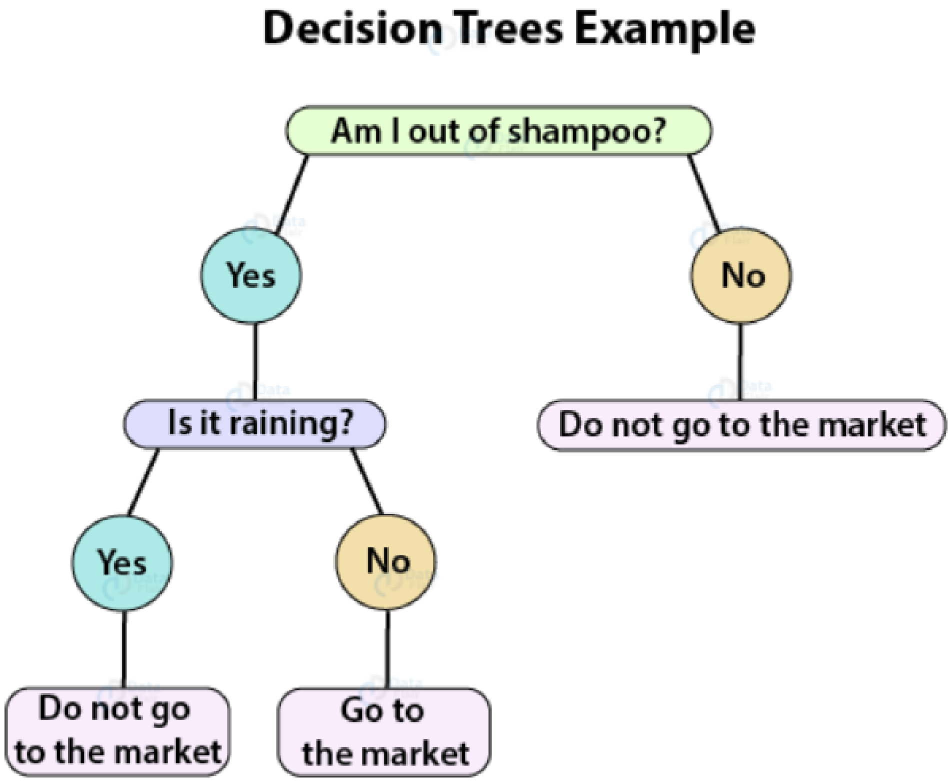
Using the decision tree with a given **set of inputs**, one can **map** the **various outcomes** that are a result of the **consequences** or **decisions**.

We can understand decision trees with the following **example**:

Let us assume that you have to go to the market to buy some products. At first, you will assess if you really need the product.

Suppose, you will only buy shampoo if you run out of it. If you do not have the shampoo, you will evaluate the weather outside and see if it is raining or not. If it is not raining, you will go and otherwise, you will not.

We can **visualize** this in the form of a decision tree as follows:



This decision tree is a result of various **hierarchical steps** that will help you to reach certain decisions. In order to build this tree, there are two steps – **Induction** and **Pruning**. In induction, we **build a tree** whereas, in pruning, we **remove** the **several complexities** of the tree.

Decision Trees are very flexible because they work with continuous and nominal variables. They are also easy to depict graphically, which makes them valuable for analyzing the decision-making processes. However, they can be computationally very intensive and could easily over fit when used with large data sets. These concerns are trimmed by techniques such as pruning and other ensemble methods inclusive of Random Forest.

4. K-Nearest Neighbors Algorithm

K-nearest neighbors is one of the most **basic** yet **important** classification algorithms in machine learning.

KNNs belong to the **supervised learning domain** and have several applications in **pattern recognition**, **data mining**, and **intrusion detection**. These KNNs are used in real-life scenarios where **non-parametric** algorithms are required. These algorithms do not make any assumptions about how the **data** is **distributed**.

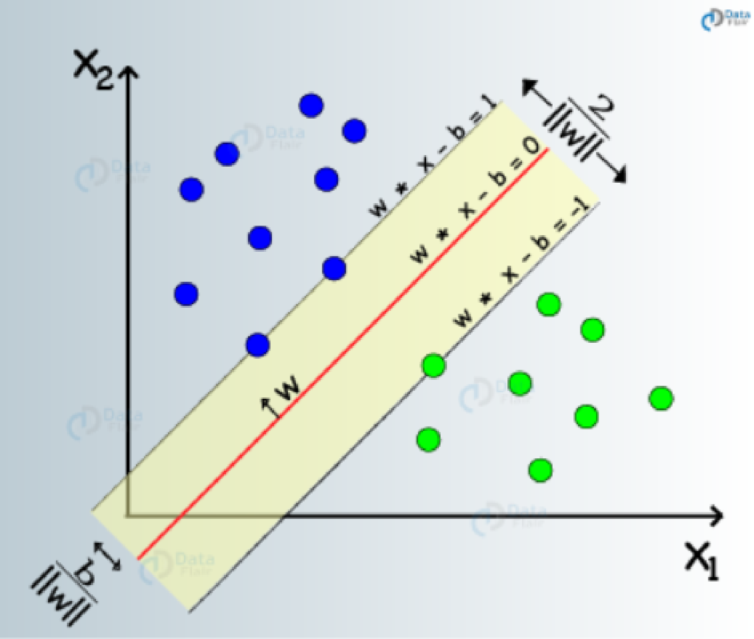
When we are given prior data, the KNN classifies the coordinates into groups that are identified by a **specific attribute**.

5. Support Vector Machine Algorithm

Support Vector Machines are a type of supervised machine learning algorithm that provides **analysis of data** for **classification** and **regression analysis**.

While they can be used for **regression**, **SVM** is mostly used for **classification**. We carry out plotting in the **n-dimensional space**. The value of each feature is also the value of the **specified coordinate**. Then, we find the ideal **hyperplane** that differentiates between the two classes.

These support vectors are the **coordinate representations** of **individual observation**. It is a **frontier method** for **segregating** the **two classes**.



6. Random Forest Algorithm

Random Forest classifiers are a **type of ensemble** learning method that is used for **classification**, **regression** and **other tasks** that can be performed with the help of the **decision trees**. These decision trees can be **constructed** at the **training time** and the **output** of the class can be either **classification** or **regression.**

With the help of these random forests, one can correct the habit of **overfitting** to the training set.

Some of the advantages and disadvantages of random forest classifiers are as follows:

Advantages – Random Forest Classifiers facilitate the **reduction** in the **over-fitting** of the **model** and these classifiers are more **accurate** than the **decision trees** in several cases.

Disadvantages – Random forests exhibit **real-time prediction** but that is **slow** in **nature.** They are also difficult to **implement** and have a **complex algorithm**.

7. Stochastic Gradient Descent Algorithm

Stochastic Gradient Descent (SGD) is a class of **machine learning algorithms** that is apt for **large-scale learning**. It is an efficient approach towards **discriminative learning** of **linear classifiers** under the **convex loss function** which is **linear (SVM)** and **logistic regression**.

We apply SGD to the large scale machine learning problems that are present in **text classification** and other areas of **Natural Language Processing**. It can **efficiently scale** to the problems that have more than **10^5 training** examples provided with more than **10^5 features**.

Following are the advantages of Stochastic Gradient Descent:

- These algorithms are **efficient**.
- We can **implement** these algorithms quite easily.

However, **Stochastic Gradient Descent (SGD)** suffers from the following disadvantages:

- The SGD algorithm requires a number of **hyperparameters** such has regularization and a **number of iterations**.
- It is also quite **sensitive** to **feature scaling**, which is one of the most important steps under data**-preprocessing**.
-

8. Kernel Approximation Algorithm

In this **submodule**, there are various functions that perform an **approximation** of the feature maps that correspond to certain kernels which are used as examples in the **support vector machines**. These feature functions perform a **wide array** of **non-linear transformations** of the input which serves as the **basis of linear** classifications or the other algorithms.
An advantage of using the approximate features that are also **explicit** in nature compared with the kernel trick is that the **explicit mappings** are better at **online learning** that can significantly **reduce the cost** of learning on **very large datasets**.
The **standard kernelized SVMs** cannot scale properly to the large datasets but with an approximate kernel map, one can utilize many efficient **linear SVMs**.

Support Vector Machine (SVM) Classifier With Python Implementation

When you enter the Machine Learning dimension, it is highly likely that one of the first classifier algorithms you might come across is SVM, you will find that SVM is all over the place. SVM which stands for Support Vector Machine is one of the most popular classification algorithms used in Machine Learning.

What is SVM?

Support Vector Machine or SVM is a supervised and linear Machine Learning algorithm most commonly used for solving classification problems and is also referred to as Support Vector Classification. There is also a subset of SVM called SVR which stands for Support Vector Regression which uses the same principles to solve regression problems. SVM also supports the kernel method also called the kernel SVM which allows us to tackle non-linearity.

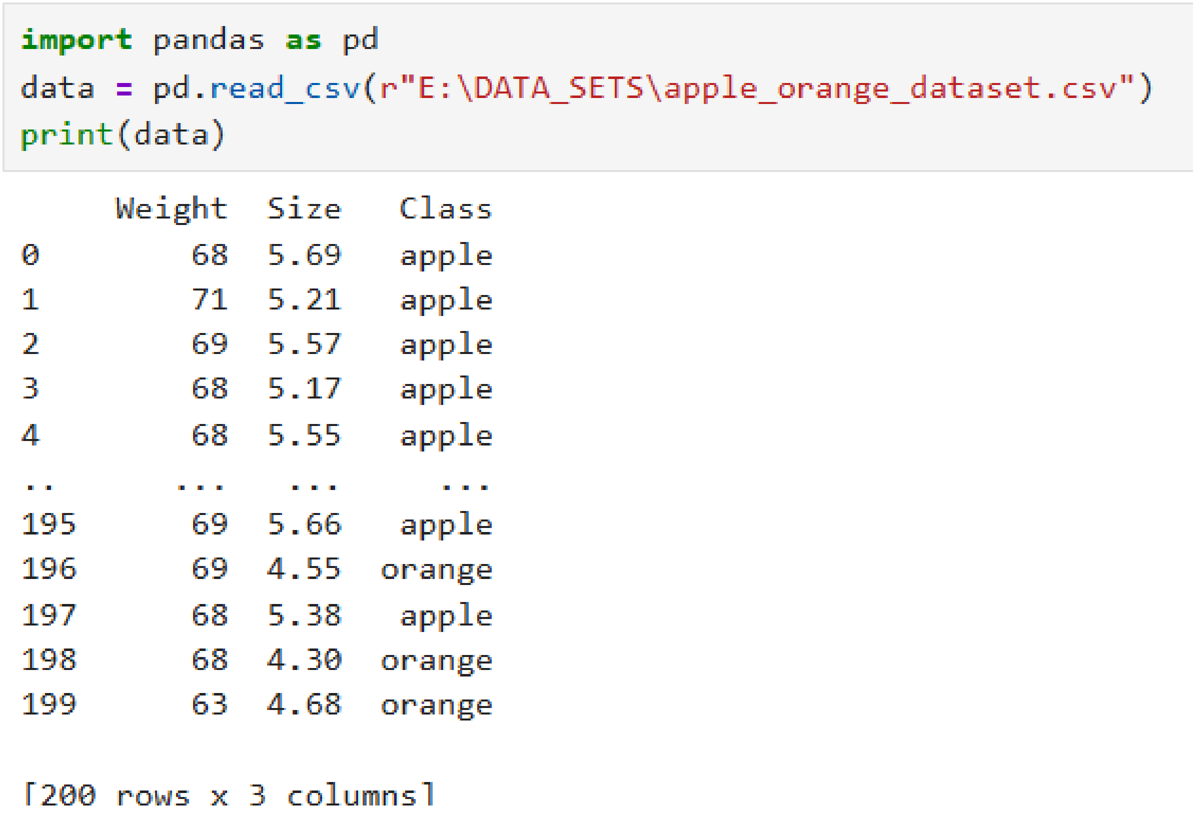
How SVM works?

Just for the sake of understanding, we will leave the machines out of the picture for a minute. Now how would a human being like you and me classify a set of objects scattered on the surface of a table? Ofcourse we will consider all their physical and visual characteristics and then identify based on our prior knowledge. We can easily identify and distinguish apples and oranges based on their colour, texture, shape etc.

Implementing SVM in Python

Now that we have understood the basics of SVM, let's try to implement it in Python. Just like the intuition that we saw above the implementation is very simple and straightforward with Scikit Learn's svm package.

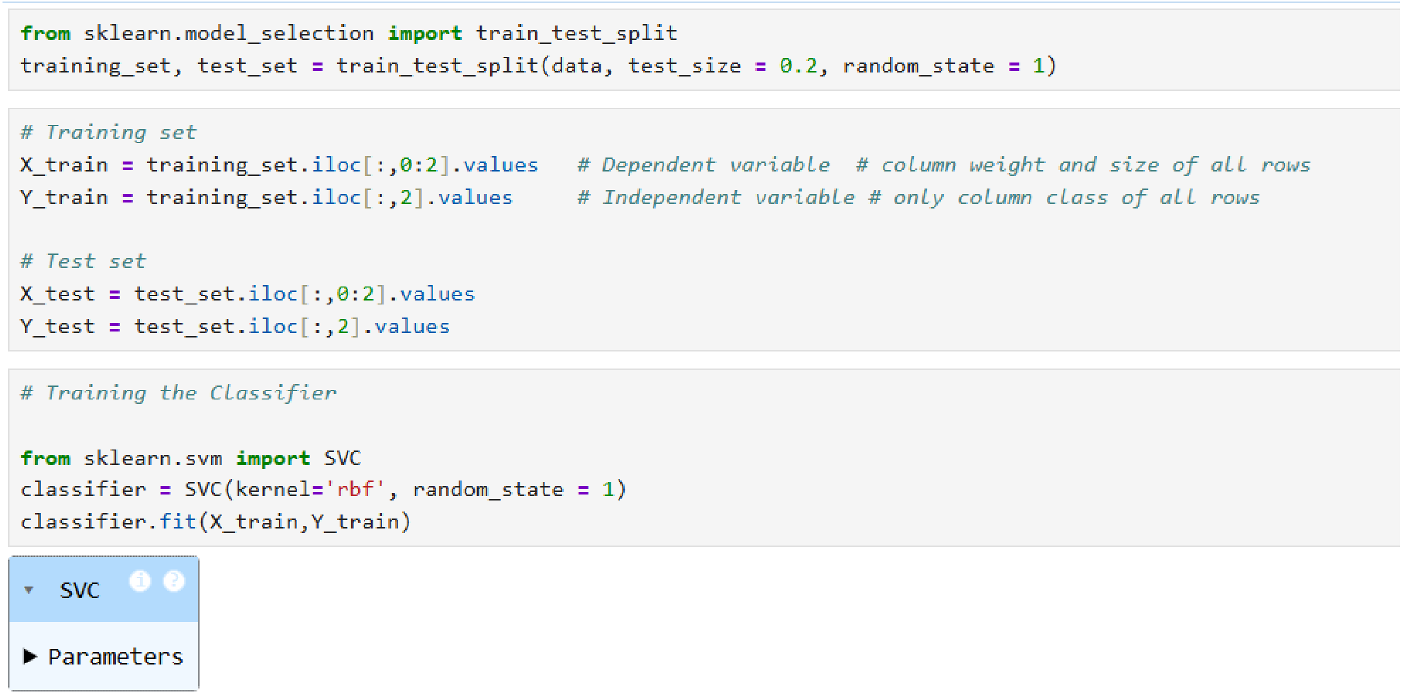
Let's use the same dataset of apples and oranges. We will consider the Weights and Size for 20 each. Click here to download the dataset or you can simply create a dataset of random values which are linearly separable.



```python
import pandas as pd
data = pd.read_csv(r"E:\DATA_SETS\apple_orange_dataset.csv")
print(data)
```

```
     Weight  Size   Class
0        68  5.69   apple
1        71  5.21   apple
2        69  5.57   apple
3        68  5.17   apple
4        68  5.55   apple
..      ...   ...     ...
195      69  5.66   apple
196      69  4.55  orange
197      68  5.38   apple
198      68  4.30  orange
199      63  4.68  orange

[200 rows x 3 columns]
```



```python
from sklearn.model_selection import train_test_split
training_set, test_set = train_test_split(data, test_size = 0.2, random_state = 1)
```

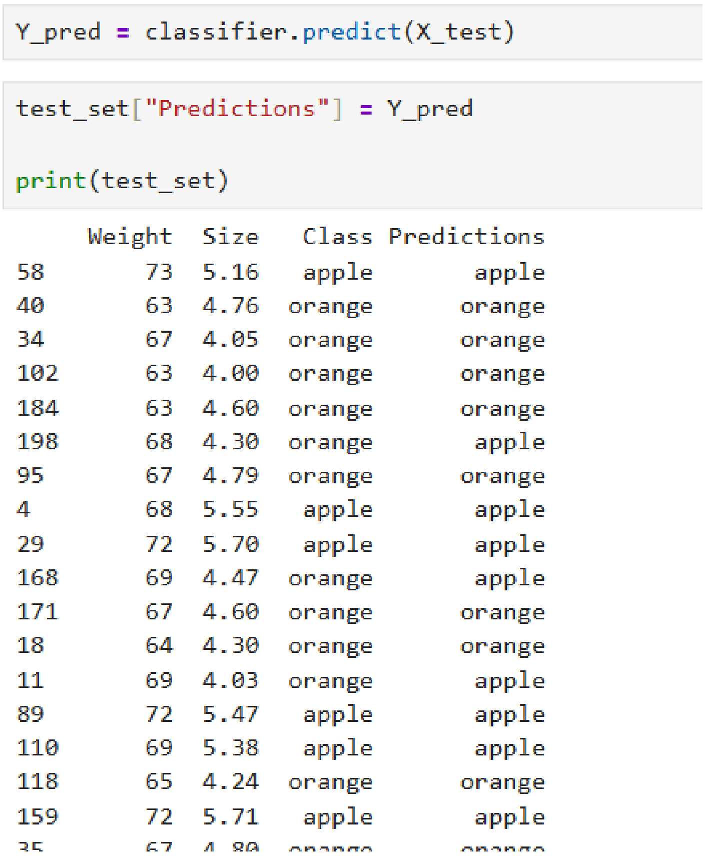
```python
# Training set
X_train = training_set.iloc[:,0:2].values   # Dependent variable  # column weight and size of all rows
Y_train = training_set.iloc[:,2].values     # Independent variable # only column class of all rows

# Test set
X_test = test_set.iloc[:,0:2].values
Y_test = test_set.iloc[:,2].values
```

```python
# Training the Classifier

from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state = 1)
classifier.fit(X_train,Y_train)
```

```
▾ SVC
▶ Parameters
```

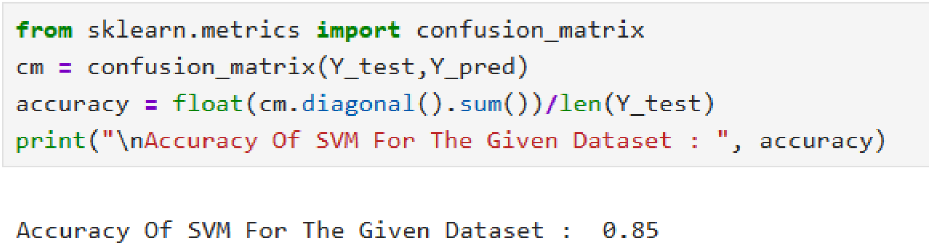


```
Y_pred = classifier.predict(X_test)
```

```
test_set["Predictions"] = Y_pred

print(test_set)
```

```
     Weight  Size   Class Predictions
58       73  5.16   apple       apple
40       63  4.76  orange      orange
34       67  4.05  orange      orange
102      63  4.00  orange      orange
184      63  4.60  orange      orange
198      68  4.30  orange       apple
95       67  4.79  orange      orange
4        68  5.55   apple       apple
29       72  5.70   apple       apple
168      69  4.47  orange       apple
171      67  4.60  orange      orange
18       64  4.30  orange      orange
11       69  4.03  orange       apple
89       72  5.47   apple       apple
110      69  5.38   apple       apple
118      65  4.24  orange      orange
159      72  5.71   apple       apple
35       67  4.80  orange      orange
```



```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test,Y_pred)
accuracy = float(cm.diagonal().sum())/len(Y_test)
print("\nAccuracy Of SVM For The Given Dataset : ", accuracy)
```

```
Accuracy Of SVM For The Given Dataset :  0.85
```