Polynomial Regression

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial.

y= b0+b1x1+ b2x12+ b2x13+...... bnx1n

It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.

It is a linear model with some modification in orde

earest Neighbour is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.

* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



## **Why do we need a K-NN Algorithm?**

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



## **How does K-NN work?**

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



* Firstly, we will choose the number of neighbors, so we will choose the k=5.
* Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



* By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



ADVERTISEMENT

* As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

## **How to select the value of K in the K-NN Algorithm?**

Below are some points to remember while selecting the value of K in the K-NN algorithm:

ADVERTISEMENT

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

## **Advantages of KNN Algorithm:**

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

ADVERTISEMENT

## **Disadvantages of KNN Algorithm:**

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples

**Apache Hadoop** is a platform that got its start as a Yahoo project in 2006, which became a top-level Apache open-source project afterward. This framework handles large datasets in a distributed fashion. The Hadoop ecosystem is highly fault-tolerant and does not depend upon hardware to achieve high availability. This framework is designed with a vision to look for the failures at the application layer. It’s a general-purpose form of distributed processing that has several components:

* **Hadoop Distributed File System (HDFS):**This stores files in a Hadoop-native format and parallelizes them across a cluster. It manages the storage of large sets of data across a Hadoop Cluster. Hadoop can handle both structured and unstructured data.
* **YARN:**YARN is Yet Another Resource Negotiator. It is a schedule that coordinates application runtimes.
* **MapReduce:** It is the algorithm that actually processes the data in parallel to combine the pieces into the desired result.
* **Hadoop Common:**It is also known as Hadoop Core and it provides support to all  other components it has a set of common libraries and utilities that all other modules depend on.

Hadoop is built in Java, and accessible through many programming languages, for writing MapReduce code, including Python, through a Thrift client.  It’s available either open-source through the Apache distribution, or through vendors such as Cloudera (the largest Hadoop vendor by size and scope), MapR, or HortonWorks.

### Advantages and Disadvantages of Hadoop –

#### Advantage of Hadoop:

1. Cost effective.

2. Processing operation is done at a faster speed.

3. Best to be applied when a company is having a data diversity to be processed.

4. Creates multiple copies.

5. Saves time and can derive data from any form of data.

#### Disadvantage of Hadoop:

1. Can’t perform in small data environments

2. Built entirely on java

3. Lack of preventive measures

4. Potential stability issues

 5. Not fit for small data

### What is Spark?

Apache Spark is an open-source tool. It is a newer project, initially developed in 2012, at the AMPLab at UC Berkeley. It is focused on processing data in parallel across a cluster, but the biggest difference is that it works in memory. It is designed to use RAM for caching and processing the data. Spark performs different types of big data workloads like:

* Batch processing.
* Real-time stream processing.
* Machine learning.
* Graph computation.
* Interactive queries.

**There are five main components of Apache Spark:**

* **Apache Spark Core:**It is responsible for functions like scheduling, input and output operations, task dispatching, etc.
* **Spark SQL:**This is used to gather information about structured data and how the data is processed.
* **Spark Streaming:**This component enables the processing of live data streams.
* **Machine Learning Library:**The goal of this component is scalability and to make machine learning more accessible.
* **GraphX:**This has a set of APIs that are used for facilitating graph analytics tasks.

### Advantages and Disadvantages of Spark-

#### Advantage of Spark:

1. Perfect for interactive processing, iterative processing and event steam processing
2. Flexible and powerful
3. Supports for sophisticated analytics
4. Executes batch processing jobs faster than MapReduce
5. Run on Hadoop alongside other tools in the Hadoop ecosystem

#### Disadvantage of Spark:

1. Consumes a lot of memory
2. Issues with small file
3. Less number of algorithms
4. Higher latency compared to Apache fling

### Hadoop vs Spark

This section list the differences between Hadoop and Spark. The differences will be listed on the basis of some of the parameters like performance, cost, machine learning algorithm, etc.

* Hadoop reads and writes files to HDFS, Spark processes data in RAM using a concept known as an RDD, Resilient Distributed Dataset.
* Spark can run either in stand-alone mode, with a Hadoop cluster serving as the data source, or in conjunction with Mesos. In the latter scenario, the Mesos master replaces the Spark master or YARN for scheduling purposes.
* Spark is structured around Spark Core, the engine that drives the scheduling, optimizations, and RDD abstraction, as well as connects Spark to the correct filesystem (HDFS, S3, RDBMS, or Elasticsearch). There are several libraries that operate on top of Spark Core, including Spark SQL, which allows you to run SQL-like commands on distributed data sets, MLLib for machine learning, GraphX for graph problems, and streaming which allows for the input of continually streaming log data.

Below is a table of differences between Hadoop and Spark:

| **Basis** | **Hadoop** | **Spark** |
| --- | --- | --- |
| **Processing Speed & Performance** | Hadoop’s MapReduce model reads and writes from a disk, thus slowing down the processing speed. | Spark reduces the number of read/write cycles to disk and stores intermediate data in memory, hence faster-processing speed. |
| **Usage** | Hadoop is designed to handle batch processing efficiently. | Spark is designed to handle real-time data efficiently. |
| **Latency** | Hadoop is a high latency computing framework, which does not have an interactive mode. | Spark is a low latency computing and can process data interactively. |
| **Data** | With Hadoop MapReduce, a developer can only process data in batch mode only. | Spark can process real-time data, from real-time events like Twitter, and Facebook. |
| **Cost** | Hadoop is a cheaper option available while comparing it in terms of cost | Spark requires a lot of RAM to run in-memory, thus increasing the cluster and hence cost. |
| **Algorithm Used** | The PageRank algorithm is used in Hadoop. | Graph computation library called GraphX is used by Spark. |
| **Fault Tolerance** | Hadoop is a highly fault-tolerant system where Fault-tolerance achieved by replicating blocks of data.  If a node goes down, the data can be found on another node | Fault-tolerance achieved by storing chain of transformations If data is lost, the chain of transformations can be recomputed on the original data |
| **Security** | Hadoop supports LDAP, ACLs, SLAs, etc and hence it is extremely secure. | Spark is not secure, it relies on the integration with Hadoop to achieve the necessary security level. |
| **Machine Learning** | Data fragments in Hadoop can be too large and can create bottlenecks. Thus, it is slower than Spark. | Spark is much faster as it uses MLib for computations and has in-memory processing. |
| **Scalability** | Hadoop is easily scalable by adding nodes and disk for storage. It supports tens of thousands of nodes. | It is quite difficult to scale as it relies on RAM for computations. It supports thousands of nodes in a cluster. |
| **Language support** | It uses Java or Python for MapReduce apps. | It uses Java, R, Scala, Python, or Spark SQL for the APIs. |
| **User-friendliness** | It is more difficult to use. | It is more user-friendly. |
| **Resource Management** | YARN is the most common option for resource management. | It has built-in tools for resource management. |

What is the difference between transformations and actions in PySpark?

Transformations create a new DataFrame without immediately computing the result, while PySpark actions trigger the computation of the DataFrame and return a value or perform a side effect on the data.