ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 3, March 2018

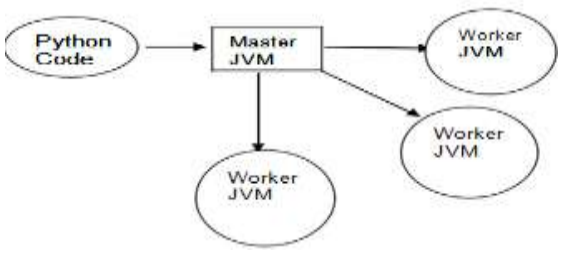
Data Analysis and Machine Learning using PySpark

[1] Kavitha C, [2] Athithya J, [3] Dhinesh Kumar K, [4] Mohamad Imran A
[1] Assistant Professor, [2][3][4] UG Student
[1][2][3][4] Department of Computer Science and Engineering RMK College of Engineering and Technology Puduvoyal, Chennai.

***Abstract* - Data analysis and machine learning have the potential to become an integral part of every existing industry. Using collective data from different sources on a specific topic or issue, an extensive scientific analysis could be done to create models and patterns that enable us to predict future outcomes with a comfortable measure of accuracy. This paper focuses on using the predefined methods available in the PySpark API to conduct data analysis and create an efficient model to predict future outcomes.**

I. INTRODUCTION

Data analysis, also known as analysis of data or data analytics, is an extensive process of inspecting, cleansing, transforming, and modeling the given data with the goal of discovering useful information, suggesting results, and to support decision-making. Data analysis has multiple avatars and approaches, encompassing diversified techniques under different names, in different business, science, and social science domain. Many Organizations in the business world use the concept of Data Analysis and the concept of Machine learning to predict pitfalls and fallacies in their revenue and business paths. Machine learning [1] is a specific field of computer science that gives computers the ability to "learn" (i.e. progressively improve performance and accuracy on a specific task) with a predefined backload of data, without being explicitly programmed do the same. Machine learning implements a lot of decisive and specific algorithms to implement the learning factor in the computer systems. Spark is an open-source cluster-computing framework developed by the Apache Software Foundation. Spark was originally developed at the UC Berkeley's AMPLab, and sometime later, its codebase was donoate to the Apache Software foundation. The entire architecture of Spark is written in the programming language Scala. But, it also provides API's with other programming languages like Python, Java. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance



The generic architecture of the PySpark API in a given machine is as given in the diagram. The PySpark API is very useful for data scientists and machine learning enthusiasts as it helps to implement an entire cluster-computing framework in a familiar python environment. Spark improves the concept of usability by offering a rich set of APIs and making it ultimately easy for developers to write efficient code. Programs in spark are very small when compared to the programs of MapReduce. The Spark Python API (PySpark) implements the Spark programming model to Python programming language. PySpark provides an easy-to-use programming abstraction and parallel runtime.

The typical life cycle of a Spark program is –

Create RDDs from some external data source or parallelize a collection in your driver program.

Lazily transform the base RDDs into new RDDs using transformations.

Cache some of those RDDs for future reuse. Perform actions to execute parallel computation and to produce results.

II. MACHINE LEARNING

Machine learning is a developing field of computer science that gives the computers the ability to learn patterns and differences based upon the given data, without being explicitly programmed. Some widely implemented methods and algorithms of Machine learning are explained below: Decision tree learning [2] uses a decision tree (as a predictive model) to go from observations about an item to conclusions about the item's target value. The leaf nodes in the decision tree represent the conclusions derived from the dataset. It is one of the widely used predictive modelling approaches used in statistics, data mining and machine learning. Association

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, March 2018

rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness

Deep learning [4] (also known as deep structured learning or hierarchical learning) is an integral part of a broader family of machine learning. Deep Learning models are primarily dependent upon the communication patterns based on the biological nervous systems. Deep Learning can be implemented by specific architectures like Neural Networks, Deep Belief Networks

Reinforcement learning (RL)[4] is another prominent area of machine learning that is primarily inspired by art of behavior and psychology. Reinforcement learning is primarily concerned with how agents of software ought to take actions in a given environment. Reinforcement learning has become the basis of many other sought-after disciplines like game theory, operations research, swarm intelligence, etc.

Representation learning indicates a set of techniques that allows a system to automatically discover the representations needed for concepts of feature detection or classification from raw data. This implementation of machine learning replaces the concept of feature engineering for the extraction of featured from the dataset given. However, real-world data such as images, video, and sensor data has failed to satisfy the algorithms in Representation learning. An alternative is to discover such features or representations through examination. [5] PySpark provides a lot of resources that are highly useful for the concept of Machine learning. Every Algorithm that I relevant to machine learning is available as a library in PySpark. By using PySpark, we can attain the goal of implementing machine learning in large sets of collective data.

III. ALGORITHM

1. Import the dataset in CSV format into the PySpark API.

2. Define a RDD in the Spark shell and import the dataset into the defined RDD

3. Define a schema for the imported Dataset to prevent incoming unicode error.

4. Using the schema, convert the RDD into a Spark Dataframe.

5. Perform Modifications in the Dataset to give out efficient results on the model that is to be created (Removal of dependent variables).

6. Perform feature extraction according to the prediction model that we are implementing. Some models do not require the concept of feature extraction as they can implicitly handle categorical variables and their conversion.

7. Based on the extracted features, create a machine learning model using any of the ML algorithms (Decision Tree, Regression, Clustering, etc.).

8. Create a Grid of standard parameters, in relevance with the imported dataset, to cross-validate the model and evaluate the model.

9. Derive predictions and label them according to Dataset column values.

10. Using the parameters grid and evaluator, evaluate the accuracy and recall values of the labels of the dataset. Using the above algorithm, we can positively predict the future outcome of a given label in the dataset.
The above algorithm is most effective with the Decision Tree model[2] since it does not create any problems with categorical variables in the Dataset. The Decision Tree algorithm operates by creating leaf nodes and root nodes and assigns specific categorical variables to these node positions and evaluates outcomes accordingly. The Cross-Validator function compares all of the created Decision trees on the Dataset and compares them with the grid of hyperparameters listed to decide upon the best available decision tree for further processing.

IV. CONCLUSION

Hence, by the concepts of Machine Learning we can efficiently predict future outcomes of the imported Dataset. This methodology and functionality of Data Analysis and Machine Learning can be applied to real-time Datasets to predict pitfalls and fallacies.

REFERENCES

[1]" Supervised Machine Learning: A Survey of Classification techniques" by S. B. Kostiantis.

[2] "Data Mining tasks Classification: Decision Tree Reconvery" and methods: by Ronny Kohavi and J. Ross Quinlan.

[3] " Applications of Machine Learning and rule Induction" by Pat Langely and Herbert A. Simon.

[4] "Data Mining: Practical Machine Learning tools and techniques" by Ian H. Witten, Ebbie frank, Mark A. Hall and Christopher J. Pal.

[5] "Mllib: Machine Learning in Apache Spark" by Xiangrui Meng, Joseph Bradley.