Opinion

Challenges in predicting disease state with apache spark

Abstract

Advances in the Big Data platform, Open Source software and affordability of commodity hardware have made it possible to implement scalable application. Assuming that Predictive Analytic technology has reached level of maturity, we attempted to implement Disease State Analytics within Apache Spark Hadoop platform. Disease State Analytics designed to predict current state of a patient disease diagnosis based up on the laboratory test results. Major purpose of this study is to evaluate current state of the art of Machine Learning technology and discuss various challenges faced.

Reference materials researched from Apache website, various books explains hypothetical scenario under academic background, while real life situations could be totally different. Most of the authors begin Machine Learning with clean delimited numeric data set and mostly assume fixed number of features. Majority of them ignores the relational nature of data as well as underlying data quality and complexity. This makes proposed solution so rigid that it is impractical to deploy it in real life. Thus we attempt to develop and implement flexible solution with all bells and whistles such as data quality, transformation, consistent feature translation, smarter predictive models with dedicated functionality. While domesticating machine with human interaction to improve decision we faces various technological challenges and tried to find workaround to make path. Ultimately we expect that Apache Spark technology should evolve by overcoming these challenges.

Volume 3 Issue 1 - 2016

Saratkar Nilesh
Department of Informatics Analytics & Business Intelligence, Quest Diagnostics Inc, USA

Correspondence: Saratkar Nilesh, Big Data Lead Department of Informatics Analytics & Business Intelligence, Quest Diagnostics Inc, 1290 Wall Street, Lyndhurst-NJ, USA, Tel +19739302015, Email Nilesh.S.Saratkar@questdiagnostics.com

Received: January 29, 2016 | **Published:** February 01, 2016

Introduction

Typically most authors publish their success stories, at this time we will share challenges face while predicting disease state of a patient from laboratory test results. Performing prediction with the manually curated data under controlled environment is very trivial work, but automating process and scaling out to thousands of tests and diseases is a real challenge. Hence we choose to work with Apache Spark platform to develop and scale up within Big Data platform. One of the major challenges we face is the volume and variety of data across each laboratory test and variation exists in laboratory test ordering pattern while diagnosing a specific disease. We hope that these challenges would be good lesson learned for the future state of the art of Apache Spark.

Discussion

Disease State Analytics utilized Apache Spark 1.3 stable version to utilize scalable predictive analytics capability. Current state of the art of the Spark technology is very hypothetical in nature and does not withstand in real-life scenarios. This limitation makes it more challenging to utilize technology to achieve results above and beyond its capabilities. However its ability to scale up to Big Data platform holds potential benefits.

1. Machine Learning is also known as domestication of machine where human to machine interaction is critical to complete feedback loop. Current state of technology focus so much on the algorithms, it almost missed to address how automatic and manual curation process benefits enhanced learning experience and make correction overtime.
2. Various machine learning algorithms expects feature data set in the tabular matrix format. However relational databases are predominant across industry. Hierarchical and relational nature of data makes it challenging to transform every bit of information into single row within tabular matrix format as well as maintain consistent relationship.
3. Data Scientists have to deal with everlasting data wrangling operations instead of focusing on data analysis. Typically data scientists manually filter, curate data and move on. But cutting corner simply adds challenges to the automation. Technology need to mature enough to record all manual curation events and translate into the reproducible process.
4. Machine learning algorithms primarily designed to work with numeric matrix format, which force every bit of information into numeric form. As we know we lose lot of information in translation and also loose capability to trace back to the original data. In reality not everything can be seamlessly translated into numbers, normalize and reverse translate back.
5. Except Decision Tree algorithm, majority of the machine learning algorithms does not publish decision tree plan. This pose huge challenge to scientific approach, which typically based up on making conclusion on the proofs and heavily rely on reproducibility of previously published result.
6. Translation of every categorical feature into numeric value usually accomplished by preparing index of unique value. This method is resource intensive and also leads to inconsistent feature values. Alternatively Hashing algorithms are used for consistent feature value. However hashing also produce very large value, this is unsuitable for array index and often requires scaling or normalization.

7. Performance of every machine learning algorithm is different and requires data in specific format. Choosing right machine learning algorithm is one of the big challenge and critical for the success. It is often done on trial and error basis. Ensemble technique holds good hope in choosing right algorithm for a given situation.
8. Machine learning algorithms evolved since many years and usually developed for centralized data processing. It's challenging to rebuilding such algorithms for Big Data platform, where data located and processed locally in distributed environment. Apache Spark community is making good progress in this direction, but lot of predictive algorithms still remains inaccessible within this community.
9. Mechanism for Cleansing and Standardization of data is completely missing from this stack. This leads to break critical feature hash keys and resulted in loss of confidence in predicted target. Data scientists usually discard substantially high and low cardinality data at to improve prediction accuracy.
10. Usually one to one relation between field name and field value assumed during feature transformation into numeric value. This preconceived notion breaks when field with mix data type comes across. In reality freeform field containing numeric, text, ratio, ordinal and binary values requires more dynamic transformation than metadata based logic.
11. Accuracy of predictive model in practice is accomplished by cross validation and hyper parameterization techniques. Since every algorithm is different and requires variety of evaluation techniques such as ROC, RMSE, AUC. Standard model evaluation framework will simplify it further.
12. Real-life situation could be so diverse that building single predictive model may not be viable solution. Current state of technology does not provide method for effective data stratification and automatically route relevant data to multiple predictive models. Ability to build smaller models for specific situation and integration with other models would help lot in pipeline maner
13. Typically predictive models assumes fixed list of features and designed for dense matrix. In order to address it, Apache Speak introduced sparse matrix to support large amount of diverse features and often requires labeling features into tuple form. Building flexible algorithm to handle varying amount of features would be ideal to scale up models to similar entity such as region or product.
14. Complete makeover of data is expected between original data and feature matrix. Most of the algorithms also loose various key fields necessary for unique identification of the record. Thus lack of traceability during wrangling and post prediction typically adds more challenges to this complex scenario. Apache Spark also evolving with machine learning pipeline framework, which hold promise to pass through identifier for traceability.
15. Insufficient data quality is the worst enemy of data scientists. Thus they often tend to filter out and avoid bad data and risk losing major chunk of data. At this time Apache Spark lacks framework for data cleansing and standardization capability. Reduction in data variation will definitely avoid false features and also improve prediction capability.
16. Evolving open source community such as Spark MLlib and ML end up in frequent releases and often lacks integration capabilities within release. Adopting fewer stable releases will definitely avoid various challenges in production deployment and large amount of system testing. Data Scientist often hit the wall when they realize that certain capability they have investigated is not supported by their application version.
17. Scalability of Data Product relies up on capability to persist predictive model into disk in PMML format. Most of the Apache Spark algorithms still does not support PMML. This capability will enable Apache Spark seamlessly deploy predictive models designed within other modeling technologies.
18. Usually prediction algorithms expect data with associated feature index and value with tuple format. It also expects that all index should be non-zero, non-repeating and sorted order. With larger data volume chances of hash key conflicts increase chances of duplication of feature index. In such situation data scientists end up compromising out come by aggregating duplicate index and values.
19. Apache Spark at this time does not focus on process deployment strategy as well as maintenance of predictive capability overtime. It will be unrealistic to change production system parameters every week. At the same time concept of integrating automated curation to manual curation for correction by data steward is not completely clearly defined.
20. High latency observed in the prediction process makes it unsuitable for the web service call within interactive application. High amount of initiation time further adds challenges to implementation. However advances in implementing algorithms within Spark Streaming as well as Spark Pipeline holds good hope. At this time very few algorithms supports this streaming capability.

Conclusion

Apache Spark is one of the fastest growing open source communities. While we encountered various challenges working with its capabilities in predicting disease state of a patient diagnosis, but this technology still holds enormous potential benefits for the Big Data Technology. Success is a journey and challenges we face today should leads to various future enhancements. We hope that all discussed challenges and workaround will enable Apache Spark community to make improvement and take it to the next level.

Acknowledgements

None.

Conflict of interest

The author declares no conflict of interest.