Journal of Big Data

RESEARCH Open Access

Using Big Data-machine learning models for diabetes prediction and flight delays analytics

Thérence Nibareke* and Jalal Laassiri

*Correspondence:
therence.nibareke@uit.ac.ma
Informatics Systems and Optimization Laboratory, Ibn Tofail University, Kenitra, Morocco

Abstract

Introduction: Nowadays large data volumes are daily generated at a high rate. Data from health system, social network, financial, government, marketing, bank transactions as well as the censors and smart devices are increasing. The tools and models have to be optimized. In this paper we applied and compared Machine Learning algorithms (Linear Regression, Naïve bayes, Decision Tree) to predict diabetes. Further more, we performed analytics on flight delays. The main contribution of this paper is to give an overview of Big Data tools and machine learning models. We highlight some metrics that allow us to choose a more accurate model. We predict diabetes disease using three machine learning models and then compared their performance. Further more we analyzed flight delay and produced a dashboard which can help managers of flight companies to have a 360° view of their flights and take strategic decisions.

Case description: We applied three Machine Learning algorithms for predicting diabetes and we compared the performance to see what model give the best results. We performed analytics on flights datasets to help decision making and predict flight delays.

Discussion and evaluation: The experiment shows that the Linear Regression, Naive Bayesian and Decision Tree give the same accuracy (0.766) but Decision Tree outperforms the two other models with the greatest score (1) and the smallest error (0). For the flight delays analytics, the model could show for example the airport that recorded the most flight delays.

Conclusions: Several tools and machine learning models to deal with big data analytics have been discussed in this paper. We concluded that for the same datasets, we have to carefully choose the model to use in prediction. In our future works, we will test different models in other fields (climate, banking, insurance.).

Keywords: Big Data, Hadoop, Spark, HBase, Machine learning, Data analytics, Accuracy, K-Nearest Neighbor, K means

Introduction

In recent decades, increasingly large amounts of data are generated from a variety of sources. The size of generated data per day on the Internet has already exceeded two Exabyte. Within 1 min, 72 h of videos are uploaded to YouTube, around 30.000 new

posts are created on the Tumble blog platform, more than 100.000 Tweets are shared on Twitter and more than 200.000 pictures are posted on Facebook [1]. Those amount of data are stored and analyzed by different tools which allow distributed storage, real time processing and data analysis. Data is considered as a key source for promoting growth and wellbeing of the society.

Every day, more than 2 quintillion bytes of data are being created in this info-centric digitized world from various sources like scientific instruments, sensors, mobile phones, social network [2–4], web authoring, telecommunication industry, social media, etc.

Big Data can be defined as high-volume, high-velocity, and high variety data that demands cost-effective, innovative forms of information. Big Data processing and analysis can be applied in many disciplines, such as science, engineering, finance, business, social as well as healthcare [5]. The aim of analysis is to get valuable insight which can help in decision making [6]. For data processing and analytics, many tools have been developed. The most used is Hadoop and Spark. Hadoop basically has two main components: Hadoop Distributed File System (HDFS) for distributed storage and second part is MapReduce for distributed processing [7–9].

Spark is a fault-tolerant, in-memory data analytics engine [10, 11]. It has many components [1, 12, 13] which allow to run Scala, Java, Python, as well as R jobs.

Machine learning is a subfield of computer science. It is a type of artificial intelligence (AI) that provides machines with the ability to learn without explicit programming. Machine learning evolved from pattern recognition and computational learning theory [14]. Machine Learning algorithms can be grouped three main categories of learning: supervised, unsupervised, and reinforcement [14]. We can use Nearest Neighbors, Naïve Bayesian, Linear Regression, K-Means, Support vector machines (SVM) and many others which will be presented in this paper.

The increase in the quantities of data as well as their varieties has motivated several researchers to investigate about the powerful tools to store, process and analyze the data and gets meaning insights which help decision-making. Further more, Machine Learning algorithms could help in the health field, in particular by predicting a disease, which would allow the informed person to take the necessary measures in time.

In this paper, we present an overview of Big Data, Machine learning tools and models. We perform diabetes prediction using three Machine Learning algorithms and compare their performance according to the accuracy, error, and the score.

This paper is organized as follow: The first section is a background of tools, models and Machine Learning algorithms that can be used for storing, processing and analyzing datasets. The "Related works" section gives an overview of some research works carried out on the same theme as this paper. In "Our methods" section, we describe the case study. The "Experiments" section presents the experimental setup, the dataset description. We discuss the results of our experiments in "Results and discussion" section. We end by a conclusion and give some future works.

Literature review

Hadoop ecosystem

In order to extract knowledge from Big Data, various models, programs, software, hardware and technologies have been proposed. To choose a technology, many parameters

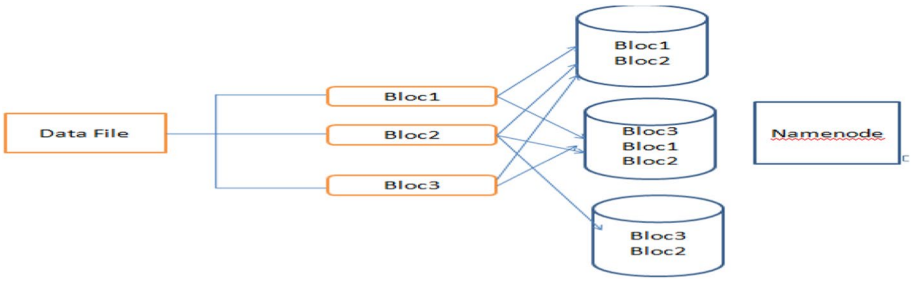

Fig. 1 Block replication in Hadoop cluster

must be considered: technological compatibility, deployment complexity, cost, efficiency, performance, reliability, security risk [1, 12]. Data scientists are facing several challenges when dealing with high volumes of data. This includes data capture, data storage, searching, sharing, analysis and visualization.

Hadoop consists of two main components for data storage and Hadoop MapReduce [1, 15]. The main advantage of Hadoop is its capacity to rapidly process data sets. This is due to its parallel clusters and its distributed file system. The second advantage of Hadoop is the fault-tolerance: Data is replicated to several nodes.

HDFS

HDFS is designed to store large amounts of data across multiple nodes of commodity hardware. HDFS has a master–slave architecture made up of data nodes which each store blocks of the data, retrieve data on demand, and report back to the name node with inventory. The name node keeps records of this inventory (references to file locations and metadata) and directs traffic to the data nodes upon client requests. If the name node fails, a secondary name node will write backups of metadata to multiple file systems [15, 16]. Figure 1 shows the replication of a file into differents data nodes.

MapReduce

A MapReduce job consists of two parts, a map phase, which takes raw data and organizes it into key/value pairs, and a reduce phase which processes data in parallel [16, 17]. A list of data elements are provided, one at a time, to a function called the Mapper, which transforms each element individually to an output data element. The Map function divides the input into ranges by the Input Format and creates a map task for each range in the input. The Job Tracker distributes those tasks to the worker nodes. The output of each map task is partitioned into a group of key–value pairs for each reducer [13, 18].

The Fig. 2 shows an example of a MapReduce job.

Limitations of Hadoop

In Hadoop version 1, MapReduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node. The Job Tracker coordinates all jobs running on the cluster and assigns map and reduce tasks to run on the Task Trackers while Task Trackers run assigned tasks and periodically report the progress to the Job Tracker [1].

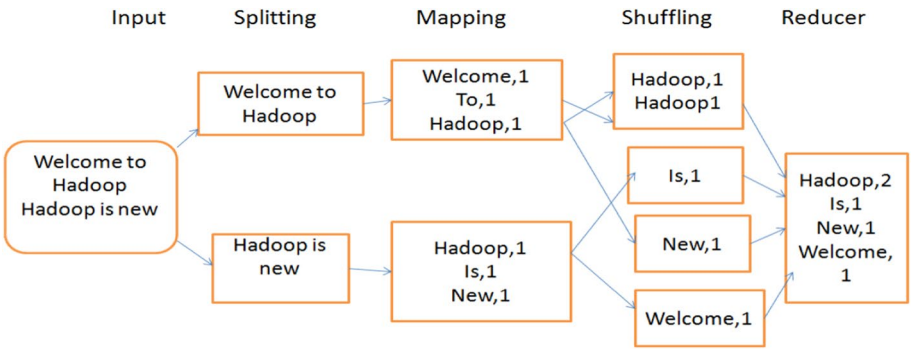

Fig. 2 Execution of MapReduce job

The Job Tracker is responsible of Data Processing and Resource management (Maintaining the list of live nodes, maintaining the list of available and occupied map and reduce slots, allocating the available slots to appropriate jobs and tasks according to selected scheduling policy).

The large Hadoop clusters revealed a limitation involving a scalability bottleneck [18] (https://data-flair.training/blogs/13-limitations-of-hadoop/) caused by having a single Job Tracker: -If job tracker fails, all jobs are lost -According to Yahoo, the practical limits of such a design are reached with a cluster of 5000 nodes and 40,000 tasks running concurrently -A node cannot run more map tasks than map slots at any given moment, even if no reduce tasks are running-This harms the cluster utilization because when all map slots are taken (and we still want more), we cannot use any reduce slots, even if they are available, or vice versa-Hadoop was designed to run MapReduce jobs only-Hadoop cannot run other applications like: Graph (graph processing).

In the second version of Hadoop called YARN [8] the two major features of the Job Tracker have been split into separate daemons: a global Resource Manager and per-application Application Master [1].

Apache Spark

Spark was introduced by Apache Software Foundation for speeding up the Hadoop computational computing software process [11]. Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming. Spark has the following features:

- Speed: Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory [19].
- Supports multiple languages: Spark provides built-in Application Programming Interface (API) in Java, Scala, or Python. Therefore, you can write applications in different languages.

- Advanced analytics: Spark not only supports Map and reduce. It also supports Structured Query Language (SQL) queries, Streaming data, machine learning, and Graph algorithms.

Spark applications run as independent sets of processes on a cluster, coordinated by the Spark Context object in the program on the master node. The driver program connects to one or more worker nodes through the cluster manager [9]. Spark can be divided into the following components [20]: Spark SQL: Spark SQL is a component on top of Spark Core that introduces a new data abstraction called Schema RDD, which provides support for structured and semi-structured data -Park Streaming: perform streaming analytics. -MLlib (Machine Learning Library) [21]: is a distributed machine learning framework above Spark -Graphx: a distributed graph-processing framework on top of Spark.

Spark is often used with distributed data stores such as MapR-XD, Hadoop's HDFS, and Amazons S3, with popular Not only SQL (NoSQL) databases such as MapR-DB, Apache HBase, Apache Cassandra, and MongoDB, and with distributed messaging stores such as MapR-ES and Apache Kafka [22]. This is a scalable and distributed NoSQL database that sits atop the HFDS. It was designed to store structured data in tables that can have billions of rows and millions of columns. HBase is not a relational database and was not designed to support transactional and other real-time applications [18].

Apache HBase enables random, real-time read/write access to big data. This has the capability and capacity of accommodating billions of rows and millions of columns atop clusters of commodity servers. Just as Bigtable leverages the distributed data storage provided by the Google file system, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS [18].

Data analytics and machine learning

Use cases of machine learning

Machine learning is used in many domains like marketing, health, e-commerce, finance, opinion analysis [22, 23].

Classification: Gmail uses a machine learning called classification to designate if an email is spam or not, based on the data of an email: the sender, recipients, subject, and message body. Classification takes a set of data with known labels and learns how to label new records based on that information.

Clustering: Google News uses clustering to group news articles into different categories, based on title and content. Clustering algorithms discover groupings that occur in collections of data.

Collaborative filtering: Amazon uses a machine learning technique called collaborative filtering (commonly referred to as recommendation) to determine which products users will like, based on their history and similarity to other users.

Related works

In recent years, many studies have been focusing on Big Data analytics and machine learning. Machine learning models can help in predicting disease [24] experimented the design of a prediction algorithm using machine learning and find the optimal classifier

to give the closest result comparing to clinical outcomes. Their results show the Decision Tree algorithm and the Random forest has the highest specificity of 98.20% and 98.00%, respectively holds best for the analysis of diabetic data [25] give a detailed version of predictive models from base to state-of-art, describing various types of predictive models, steps to develop a predictive model, their applications in health care in a broader way and particularly in diabetes. Farooq and Hussain [26] developed a hybrid clinical decision support mechanism by combining evidence, extrapolated through legacy patient data to facilitate cardiovascular preventative care.

Sternberg et al. [27] propose a taxonomy and summarize the initiatives used to address the flight delay prediction problem, according to scope, data, and computational methods. Chen and Li [28] propose a delay propagation model as a link to connect features to build a chained delay prediction model. Zettam et al. [29] described a MapReduce-based Adjoint method for preventing brain disease [30], mobile phone data were collected and the customer's gender and age were predicted. The authors analyzed Call Data Records (CDR), billing data and other customer's information and applied different types of Machine Learning algorithms to provide marketing campaigns with more accurate information about customer demographic attributes (age, gender). The model applied to 18,000 users information achieved 85.6% accuracy in terms of user gender prediction and 65.5% of user age prediction. Dahdouh [31] developed a distributed courses recommender system for the e-learning platform that aims to discover relationships between student's activities (historical logs) using association rules method in order to help students to choose the most appropriate learning materials. Their experimental results show the effectiveness and scalability of the proposed system.

Al-Saqqa, Al-Naymat and Awajan [4] used apache Spark with MLIB to perform sentiment analysis on customer's reviews on a product. His experimental results showed that Support vector machine classifier outperforms Naïve Bayes and logistic regression classifiers [21], as electronic commerce, customers can make electronic payments. However, the system needs confidence by making fraud detection a critical factor. Authors presented a Scalable Real-time Fraud Finder (SCARFF).

Our methods

Diabetes prediction

Diseases prevention has been one of the uses of new technology in particular Machine Learning algorithms [32, 33] medical devices with sensor, health cloud and continuously generating a huge amount of data which is often called as streaming big data. Over the last few decades, heart diseases and diabetes are the most common cause of global death. So early detection of these diseases and continuous monitoring can reduce the mortality rate [32].

Diabetes is a chronic disease or group of metabolic disease where a person suffers from an extended level of blood glucose in the body, which is either the insulin production is inadequate, or because the body's cells do not respond properly to insulin [24]. There are four types of diabetes which are type 1, type 2, gestational and pre diabetes [25].

We used machine learning based algorithms to predict diabetes. The model were tested using python as the programming language. The Python language has diversified application in the software development companies such as in gaming, web frameworks

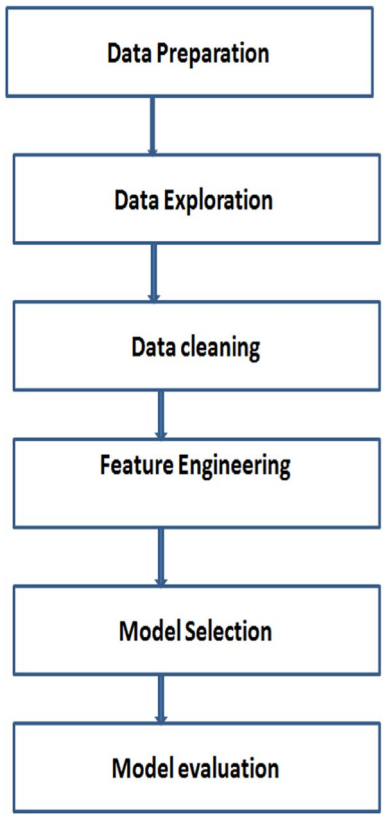

Fig. 3 Workflow of the proposed model

and applications, language development, prototyping, graphic design applications. This provides the language a higher plethora over other programming languages used in the industry. Some of its advantages are-Three Machine Learning algorithms were carried on diabetes datasets: Linear regression, Naive Bayes and Decision Tree [26]. The dataset used contains 7 features and we want to predict the class of a given person (1: positive, 0: negative). We calculated the accuracy, the score and the RMSE of each model. The Fig. 3 shows the Chart flow of the model.

Proposed algorithm

Data cleaning refers to the process of removing invalid data points from a dataset. Many statistical analyses try to find a pattern in a data series, based on a hypothesis or assumption about the nature of the data. Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. Data cleaning is one of those things that everyone does but no one really talks about in the process we ignore these particular data points, and conduct our analysis on the remaining data. The algorithm used is as follows:

1. Data loading and cleaning

    1.1 Import the library pandas.

1.2 Save the CSV file which contains the data in the same folder as the python file.
1.3 Use the function "read_csv" to load the data.
1.4 Handling missing Data and empty lines
1.5 Features transformation (tested_Negative: 0; tested_positive: 1)

2. Apply logistic Regression model and evaluate the model.

2.1 Set the cross validation (cv)parameter to 10.
2.2 Calculate the error RMSE=np.sqrt(((y-model.predict(X))**2).sum()/len(y) where x represents the features of a person and y the class (0 or 1), and np is a variable of the numpy library.

3. Apply the Naïve bayes Model on the same data and calculate the RMSE with the cv parameter = 10.
4. Apply the Decision Tree Model and calculate the error with the same formula as in.
5. Compare the values of RMSE for the three models.

Before performing any machine learning algorithm, first we analyzed the dataset. This step is necessary to familiarize with the data, to gain some understanding about the potential features and to see if data cleaning is needed. Data cleaning is considered to be one of the crucial steps of the work flow, because it can make or break the model. Dataset cleaning includes correct duplicate or irrelevant observations, missing or null data points etc.

Feature engineering is the process of transforming the gathered data into features that better represent the problem that we are trying to solve to the model, to improve its performance and accuracy. After the data was cleaned, Model selection or algorithm selection phase is the heart of machine learning. It is the phase where we select the model which performs best for the data set. It is a general practice to avoid training and testing on the same data. The reasons are that, the goal of the model is to predict out-of-sample data, and the model could be overly complex leading to over fitting. The model evaluation is done by the following metrics [34]:

- Accuracy: The percentage of correct classifications (values from 0 to 100). It indicates the classifier's ability to correctly guess the proper class for each element.

The Eq. 1 gives the accuracy a model:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}. \quad (1)$$

The accuracy is obtained by calculating the following parameters:

- True Positives (TP): The cases in which we predicted YES and the actual output was also YES.
- True Negatives (TN): The cases in which we predicted NO and the actual output was NO.
- False Positives (FP): The cases in which we predicted YES and the actual output was NO.

- False Negatives (FN): The cases in which we predicted NO and the actual output was YES.
  From (1)

$$\begin{aligned} Accuracy &= \big(True\,positive + True\,Negative\big)/Total\,number\,of\,samples \\ &=> Accuracy(TP + TN)/(TP + TN + FP + FN). \end{aligned} \tag{2}$$

- F-score: The weighted average of precision and recall of classifications (values from 0 to 1). F1 score is the Harmonic Mean between precision and recall. It allows knowing how many instances the model classifies correctly, as well as how robust it is. High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 score, the better is the performance of the model. The Eq. 3 gives the expression of F-score

$$Score = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{3}$$

where

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5}$$

.

- Mean Absolute Error
  Mean Absolute Error is the average of the difference between the original values and the predicted values. It gives us the measure of how far the predictions were from the actual output. However, they don't give us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data. Mathematically, it is represented as:

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N} |y_j - \hat{y}_j|. \tag{6}$$

- RMSE: Standard deviation between the real and predicted values via regression. It is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the square of the difference between the original values and the predicted values. The advantage of Mean Absolute Error (MSE) being that it is easier to compute the gradient, whereas Mean Absolute Error requires complicated linear programming tools to compute the gradient. As, we take square of the error, the effect of larger errors become more pronounced then smaller error, hence the model can now focus more on the larger errors.

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N} |y_j - \hat{y}_j|^2 \tag{7}$$

Flight delays analytics

Flight delays create problems in scheduling, passenger inconvenience, and economic losses, there is growing interest in predicting flight delays before hand in order to optimize operations and improve customer satisfaction.[1]

With rapid growth of air traffic, increasing flight delays in the United States (US) have become a serious and prominent problem. According to the Bureau of Transportation Statistics (BTS), nearly one in four airline flights arrived at its destination over 15 min late (https://www.sita.aero/air-transport-it-review/articles/using-artificial-intelligence-to-predict-flight-delays, https://data-flair.training/blogs/13-limitations-of-hadoop/). It is reported that the annual total cost of air transportation delays was over $30 billion, which poses a significant challenge to the development of Next Generation Air Transportation System [28]. Delay is one of the most remembered performance indicators of any transportation system [27].

A flight delay is said to occur when an airline lands or takes off later than its scheduled arrival or departure time respectively. Conventionally if a flight's departure time or arrival time is greater than 15 min than its scheduled departure and arrival times respectively, then it is considered that there is a departure or arrival delay with respect to corresponding airports. Notable reasons for commercially scheduled flights to delay are adverse weather conditions, air traffic congestion, late reaching aircraft to be used for the flight from previous flight, maintenance and security issues [35].

Managers would like to know which day of week has the most delayed flight, the number of delayed flights for each airport of departure and airport destination as well as other statistics which can help decision making. The objective of this study is to perform analysis of the historical flight data to gain valuable insights. Using machine learning models, we can build a predictive model to predict whether a flight will be delayed or not given a set of flight characteristics. The analysis could be able to answer to the following questions: Which Airports have the Most Delays? Which routes are typically the most delayed? Airport Origin delays per month, Airport Origin delay per day/hour, what are the primary causes for flight delays? In our model we used spark as the storage and data processing, Scala and Spark SQL for our analytics. The Fig. 4 shows the workflow of the model.

As we saw in the previous section, for getting reliable information from data, we have to perform many steps from the data acquisition to the results analytics.

Spark is an in-memory tool for a fast data processing. We used flight information from the United States Department of Transportation. After downloading flight data, the first step was to load our data into a DataFrame. We used a Scala case class and StructType to define the schema, corresponding to a line in the JSON (JavaScript Object Notation) data file. We loaded the data from 2 months (January and February 2018). We performed a column transformation; we added a new column "orig_dest" for the origination-destination airport, in order to use this as a feature. Then we could query the DataFrame to get useful information like the count of departure delays by origin_destination, the day of the week with delayed flight.

[1] https://www.sita.aero/air-transport-it-review/articles/using-artificial-intelligence-to-predict-flight-delays).

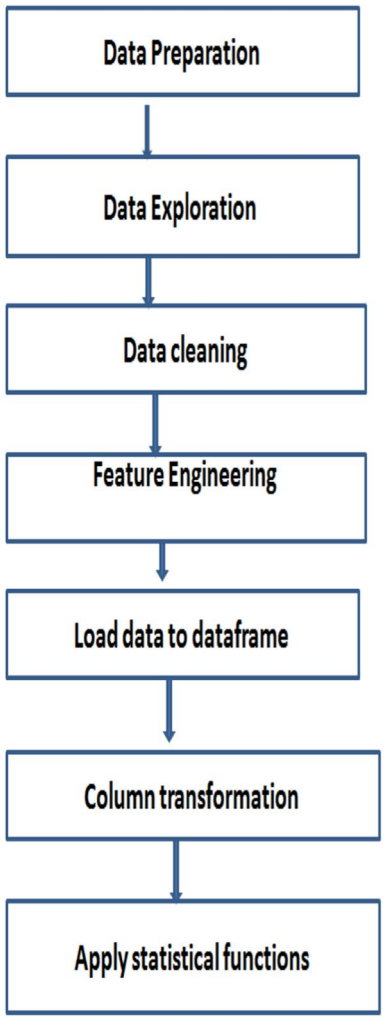

Fig. 4 Flight delay analytics workflow

Proposed algorithm

In our experiment, we used Scala as the programming language and the databriks cloud platform which contains a notebook and a spark session for executing the Scala code. Using cloud plateform provided at the link: https://community.cloud.databricks.com/login.html that can be accessed by creating user account which is the mail address. We created a cluster which run the Scala jobs. The first step is to load the json file in the data directory of cloud plateform and then we create a cluster and start it.

1. Access to the cloud plateform at https://community.cloud.databricks.com/login.
   1.1 Create an account with a username and password.
2. Create a cluster which able to run scala jobs.
3. Load flight datasets into the data directory (file store) of the cluster.
4. Create a new notebook and import the apache.spark packages.
5. Transform each json file line to an object with attributes.
6. Use Spark.sql package to run queries and display the results.

Table 1 Diabetes dataset features

Index feature	Feature	Feature name
0	Preg	Pregnancies (number of times pregnant)
1	Plas	Glucose
2	Pres	Blood pressure (mm Hg)
3	Skin	Skin thickness (mm)
4	Insu	Insulin (mu U/ml)
5	Mass	BMI (Body Mass Index:Weight (kg)/Height^2 (m^2)
6	Predi	Diabetes pedigree function
7	Age	Age (years)
8	Class	Class (1: positive, 0: negative)

Experiments

For our experiment we used a single node cluster. The setup includes an operating system Windows 10, 64 bytes with an I5 Control Processing Unit (CPU), each containing 8 cores clocked at 3 GHz, and 4 GB Random Access Memory (RAM) and a total storage space of 350 GB. For speeding up the computation, an hybrid plateform with many nodes or cloud can be used for storing and processing data. We used anaconda as plateform which can be installed on a single node cluster.

For the performance evaluation of algorithms we compared three algorithms for predicting diabetes disease for a person. We installed anaconda version 3 available at the link (https://repo.anaconda.com/archive) and used Python 3+ as the programming language. We used Spyder which is a Python Development Environment with a rich interactive testing and debugging. We imported some libraries to perform our algorithm like ScikitLearn, Numpy, Pandas, Matplotlib, and Seaborn.

Diabetes is a disease which occurs when the blood glucose level becomes high, which ultimately leads to other health problems such as heart diseases, kidney disease etc. The datasets was provided from (https://datahub.io/machine-learning/diabetes#resource-diabetes). The diabetes dataset have been produced by the National Institute of Diabetes and Digestive and Kidney Diseases and provided by Vincent Sigillito (vgs@aplcen.apl.jhu.edu) Research Center, RMI Group Leader Applied Physics Laboratory The Johns Hopkins University Johns Hopkins Road Laurel, MD 20707. The diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria: if the 2 h post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). All patients here are females. For the male patients other parameters should been considered. The data used for our experiment is a Comma Separated Value (csv) file of 36 KB (diabetes.csv). The file contains 768 datasets with 7 features and each dataset is labeled with the class (positive: 1, negative: 0) as shown in Table 1.

For the flight delays analytics, the datasets was provided on https://github.com/mapr-demos/mapr-spark2-ebook. The input file is a JSON format of 8.41 MB with 413,348 rows and 12 features (Table 2).

In our experiment, we used Scala as the programming language and the databriks cloud platform which contains a notebook and a spark session for executing the Scala code. Using cloud plateform provided at https://community.cloud.databricks.com/login

Table 2 Flight features

Index feature	Feature	Feature name
0	_id	ID composed of carrier, date, origin, destination, flight number
1	dofW	Day of week (1 = Monday, 7 = Sunday)
2	Carrier	Carrier code
3	Origin	Origin airport code
4	Dest	Destination airport code
5	Crsdephour	Scheduled departure hour
6	Crsdeptime	Scheduled departure time
7	Depdelay	Departure delay in minutes
8	Crsarrtime	Scheduled arrival time
9	Arrdelay	Arrival delay minutes
10	Crselapsedtime	Elapsed time
11	Dist	Distance

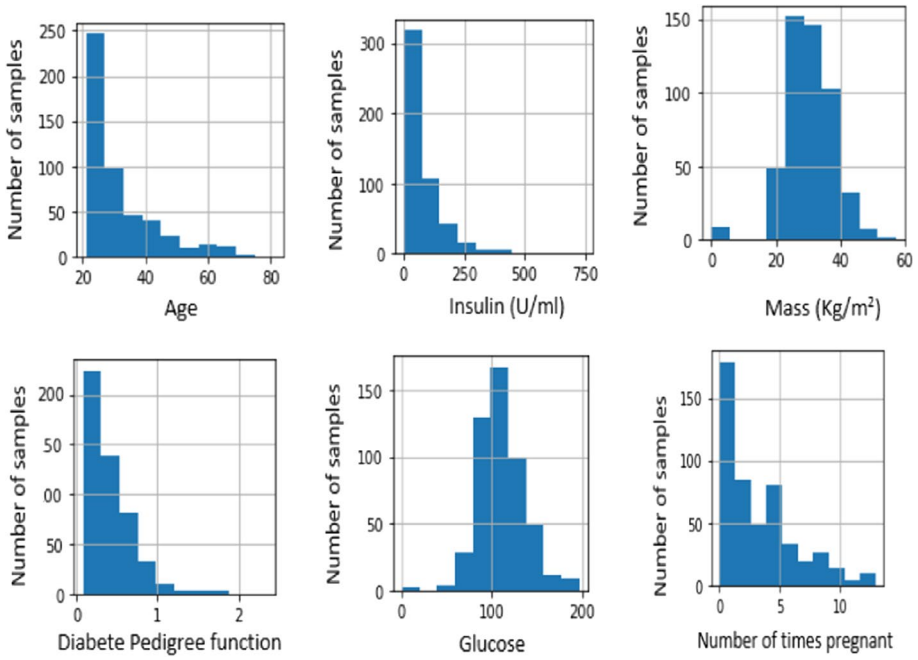

Fig. 5 Diabetes data distribution

.html that can be accessed by creating user account which is the mail address. We created a cluster which run the Scala jobs. The first step is to load the json file in the data directory of cloud plateform and then we create a cluster and start it.

Results and discussion

The figure shows the graphics of diabetes datasets distribution for each feature:

The Fig. 5 shows that almost 250 persons in our sample were between 20 and 30 years old, more than 250 persons had between 0 and 250 U/ml for insulin. We can see on the graphs that diabetes pedigree was between 0 and 2 with the majority of persons having between 0 and 0.3 as the diabetes pedigree function. We tested three

Table 3 Model evaluation metrics

Model	Accuracy	Score	RMSE
Linear regression	0.766	0.774	0.476
Naive Bayes	0.766	0.763	0.486
Decision Tree	0.766	0.956	0.125

Fig. 6 Performance of the models

models (Logistic Regression, Naïve Bayes, and Decision Tree) and evaluated them by calculating the accuracy, the score and the error (RMSE). The table shows the value of three metrics for each model.

The performance evaluation of the three models—Decision tree, Naïve Bayesian and Linear Regression was done by calculating the three parameters (accuracy, score and RMSE). A good model is the model having the score close to 1 and RMSE close to 0. These parameters determine the accuracy of the model and a good prediction with minimum errors.

Table 3 shows the accuracy, score and RMSE of the three models. For Decision Tree model, the RMSE was closed to 0 while the trend of the score is 1. This can occur when the expected classes outputs are exactly matched by actual outputs. That means the model is ideally trained and goal is 100%, this means the features were enough strong to result in 100% classification rate. These values mean that the model was accurate in prediction.

The three models show the same accuracy (0.766), Linear regression and Naïve bayesian give errors (RMSE) which are close equal (Fig. 6).

The experiment shows that the three models give the same accuracy but Decision tree outperforms the two other models with the greatest score (1) and the smallest error (0). Given a dataset of features, the model is able to predict the positivity or negativity of a person. For example .Testing our model with [6, 109, 60, 27, 0, 25, 0.206, 27] gives the 0 as class which means the person was tested negative.

While analyzing the flight delays, the results showed that Sunday and Monday are the days of the week with the highest average of departure delay flights while on Wednesday we had the lowest average (Fig. 7). The ORD airport has the highest

dofW	avgdelay
7	18.75420240137221
1	17.404697121580984
6	16.829883897021706
4	14.609846568875401
2	13.50552611657835
5	13.10193236714976
3	11.3646816543676

Fig. 7 Average delay per day

origin	count(depdelay)
ORD	679
ATL	637
SFO	542
EWR	518
DEN	484
IAH	447
LGA	432
MIA	429
BOS	296

Fig. 8 Delayed flight by airport

orig_dest	count
DEN_SFO	172
ORD_SFO	168
ATL_LGA	155
ATL_EWR	141
SFO_DEN	134

Fig. 9 Delay statistics

delayed	count
0.0	36790
1.0	4558

Fig. 10 Delayed flight by route

number of delayed departures (Fig. 8). The routes ORD->SFO and DEN->SFO have the highest delays, possibly because of weather in January and February (Fig. 9).

Figure 10 shows that the number of delayed flight was equal to 4558 and the flight who was not delayed equal to 36790. We can calculate the rate = delayed flight/not delayed

flight = 0.12 this ration shows the quality of service. If the ration is greater than 1, which means the number of delayed flight is greater than the number of not delayed flight, managers will have to take strategic decisions to improve the service. These analytics gave a good dashboard for the managers.

Conclusion

Machine Learning algorithms can be used to get insights value from data. This can help in decision making and make predictions in many domains (Healthcare, Financial, Marketing, Recommendation engines etc.). Machine Learning algorithms are commonly classified into supervised, unsupervised, semi-supervised, reinforcement learning and transfer learning). These algorithms can be used to make classification, clustering, collaborative filtering from a dataset. The accuracy of each algorithm depends on the values of some parameters like accuracy, score and error. Diabetes is defined as a chronic disease where a person presents an extended level of blood glucose in the body (The production of insulin is inadequate or the body's cells are not responding to insulin). Diabetes can cause serious complication to people's health. The prediction of diabetes can help in taking strategic decision of preventing one's health. In this paper, we give an overview of Big Data and tools, Machine Learning algorithms and performance comparison. We experimented a use case of Big Data analytics using Spark by analyzing flight delays.

To make a good prediction or classification, we have to use a adequate machine learning algorithm. In our experiment we try to predict diabetes by using Naïve Bayesian, Linear Regression and Decision Tree. We calculated and compared three parameters: accuracy, error and score for the three models. With decision tree, a score equals to 1 and an error equal to 0 were achieved. Decision tree is the most appropriated model for the case of study. We concluded that Decision tree is the best model for this use case while Naïve bayesian gave worst values of score, accuracy and error (RMSE).

We analyzed flight data to help decision making. The model used can show on which day of week, which airport of departure and airport destination we have the most delayed flight. From this information managers can adopt strategy in order to minimize and anticipate the impact of the flight's delay on customer satisfaction.

In our future works we will continue to explore other domain like business, marketing, social networks, climate in order to find out how big data can be analyzed for getting value. We will experiment how different tools and models can be associated to achieve the highest performance in time, correctness, and resources optimization.

Abbreviations

AI: Artificial intelligence; API: Application Programming Interface; CDR: Call Data Records; CPU: Control Processing Unit; ETL: Extract transform load; HDFS: Hadoop Distributed File System; JDBC: Java Database Connectivity; JSON: JavaScript object notation; KNN: K-Nearest Neighbors; MLib: Machine learning library; MSE: Mean Absolute Error; NoSQL: Not only SQL; ODBC: Open Database Connectivity; RAM: Random access memory; RDD: Resilient distributed dataset; RMSE: Root mean square error; SQL: Structured Query Language; SVM: Support vector machines; YARN: Yet another resource negotiator.

Acknowledgements

Not applicable.

Authors' contributions
TN has conducted the experimental work and wrote the manuscript. JL was the supervisor of the work. He suggested the structure of the manuscript and corrected the final version of the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The diabetes datasets was provided from the link https://datahub.io/machine-learning/diabetes#resource-diabetes. For the flight delays analytics, the datasets was provided on the link https://github.com/mapr-demos/mapr-spark2-ebook. The input file is a JSON format of 8.41 MB with 413,348 rows and 12 features.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 23 October 2019 Accepted: 3 September 2020
Published online: 17 September 2020

References

1. Inoubli W, Aridhi S, Mezni H, Maddouri M, Mephu Nguifo E. An experimental survey on big data frameworks. Future Gener Comput Syst. 2018;86:546–64.
2. Petrov M, Butakov N, Nasonov D, Melnik M. Adaptive performance model for dynamic scaling Apache Spark Streaming. Procedia Comput Sci. 2018;136:109–17.
3. Brahmwar M, Kumar M, Sikka G. Tolhit—a scheduling algorithm for Hadoop Cluster. Procedia Comput Sci. 2016;89:203–8.
4. Al-Saqqa S, Al-Naymat G, Awajan A. A large-scale sentiment data classification for online reviews under apache spark. Procedia Comput Sci. 2018;141:183–9.
5. Zheng W, Qin Y, Bugingo E, Zhang D, Chen J. Cost optimization for deadline-aware scheduling of big-data processing jobs on clouds. Future Gener Comput Syst. 2018;82:244–55.
6. Akhavan-Hejazi H, Mohsenian-Rad H. Power systems big data analytics: an assessment of paradigm shift barriers and prospects. Energy Rep. 2018;4:91–100.
7. Uzunkaya C, Ensari T, Kavurucu Y. Hadoop ecosystem and its analysis on tweets. Procedia Soc Behav Sci. 2015;195:1890–7.
8. Naik NS, Negi A, Anitha R. A data locality based scheduler to enhance MapReduce performance in heterogeneous environments. Future Gener Comput Syst. 2019;90:423–34.
9. Sarumi OA, Leung CK, Adetunmbi AO. Spark-based data analytics of sequence motifs in large omics data. Procedia Comput Sci. 2018;126:596–605.
10. Hernández ÁB, Perez MS, Gupta S, Muntés-Mulero V. Using machine learning to optimize parallelism in big data applications. Future Gener Comput Syst. 2018;86:1076–92.
11. Hidalgo N, Rosas E, Vasquez C, Wladdimiro D. Measuring stream processing systems adaptability under dynamic workloads. Future Gener Comput Syst. 2018;88:413–23.
12. Lu S, Wei X, Rao B, Tak B, Wang L, Wang L. LADRA: log-based abnormal task detection and root-cause analysis in big data processing with Spark. Future Gener Comput Syst. 2019;95:392–403.
13. JayaLakshmi ANM, Krishna Kishore KV. Performance evaluation of DNN with other machine learning techniques in a cluster using Apache Spark and MLlib. J King Saud Univ Comput Inf Sci. 2018. https://doi.org/10.1016/j.jksuci.2018.09.022.
14. Mahdavinejad MS, Rezvan M, Barekatain M, Adibi P, Barnaghi P, Sheth AP. Machine learning for internet of things data analysis: a survey. Digit Commun Netw. 2018;4(3):161–75.
15. Rao Chandakanna V. REHDFS: a random read/write enhanced HDFS. J Netw Comput Appl. 2018;103:85–100.
16. Landset S, Khoshgoftaar TM, Richter AN, Hasanin T. A survey of open source tools for machine learning with big data in the Hadoop ecosystem. J Big Data. 2(1). 2015. http://www.journalofbigdata.com/content/2/1/24.
17. Subramaniyaswamy V, Vijayakumar V, Logesh R, Indragandhi V. Unstructured data analysis on big data using map reduce. Procedia Comput Sci. 2015;50:456–65.
18. Raj P. The Hadoop ecosystem technologies and tools. In: Advances in computers, vol. 109. Elsevier; 2018. pp. 279–320.
19. Mustafa S, Elghandour I, Ismail MA. A machine learning approach for predicting execution time of spark jobs. Alex Eng J. 2018;57(4):3767–78.
20. Chambers B, Zaharia M. Spark: The definitive guide; 2018. p. 600.
21. Carcillo F, Dal Pozzolo A, Le Borgne Y-A, Caelen O, Mazzer Y, Bontempi G. SCARFF: a scalable framework for streaming credit card fraud detection with spark. Inf Fusion. 2018;41:182–94.
22. McDonald C. Getting started with Apache Spark from inception to production; 2018. p. 174.
23. Garcia-Ceja E, Riegler M, Nordgreen T, Jakobsen P, Oedegaard KJ, Tørresen J. Mental health monitoring with multimodal sensing and machine learning: a survey. Pervasive Mob Comput. 2018;51:1–26.

24. Sneha N, Gangil T. Analysis of diabetes mellitus for early prediction using optimal features selection. J Big Data. 2019;6(1):13. https://doi.org/10.1186/s40537-019-0175-6.
25. Jayanthi N, Babu BV, Rao NS. Survey on clinical prediction models for diabetes prediction. J Big Data. 2017;4(1):26. https://doi.org/10.1186/s40537-017-0082-7.
26. Farooq K, Hussain A. A novel ontology and machine learning driven hybrid cardiovascular clinical prognosis as a complex adaptive clinical system. Complex Adapt Syst Model. 2016;4(1):12. https://doi.org/10.1186/s40294-016-0023-x.
27. Sternberg A, Soares J, Carvalho D, et al. A review on flight delay prediction. 2017. arXiv preprint arXiv:1703.06118. https://arxiv.org/abs/1703.06118.
28. Chen J, Li M. Chained predictions of flight delay using machine learning. In: AIAA Scitech 2019 Forum. 2019. p. 1661. https://www.researchgate.net/publication/330185077.
29. Zettam M, Laassiri J, Enneya N. A MapReduce-based Adjoint method for preventing brain disease. J Big Data. 2018. https://doi.org/10.1186/s40537-018-0136-5.
30. Al-Zuabi IM, Jafar A, Aljoumaa K. Predicting customer's gender and age depending on mobile phone data. J Big Data. 2019. https://doi.org/10.1186/s40537-019-0180-9.
31. Dahdouh K, Dakkak A, Oughdir L, Ibriz A. Large-scale e-learning recommender system based on Spark and Hadoop. J Big Data. 2019. https://doi.org/10.1186/s40537-019-0169-4.
32. Ed-daoudy A, Maalmi K. A new Internet of Things architecture for real-time prediction of various diseases using machine learning on big data environment. J Big Data. 2019;6(1):104. https://doi.org/10.1186/s40537-019-0271-7.
33. Hosseinzadeh F, Kayvanjoo AH, Ebrahimi M, et al. Prediction of lung tumor types based on protein attributes by machine learning algorithms. SpringerPlus. 2013;2(1):238.
34. Behera M, Fowler EE, Owonikoko TK, et al. Statistical learning methods as a preprocessing step for survival analysis: evaluation of concept using lung cancer data. Biomed Eng Online. 2011;10(1):97.
35. Chakrabarty N. A data mining approach to flight arrival delay prediction for american airlines. 2019. arXiv preprint arXiv:1903.06740.

Publisher's Note