Unit 6: Big Data Analytics using R

6.0 Overview

Units 1-5 have covered the fundamentals of R, Big Data and the Big Data ecosystem. This unit gives an overview of Big Data analytics techniques and explains the phases of the data analytics life cycle. Moreover, some real world examples where Big Data analytics could be applied, are described. The unit also emphasizes on Machine Learning as a technique to analyse big datasets. Supervised Machine Learning techniques namely Linear Regression, Logistic Regression and Random Forest and unsupervised Machine Learning techniques namely K-Means algorithm and Principal Components Analysis (PCA) have been discussed. Some algorithms have been applied to a dataset using Spark.

6.1 Learning Outcomes

Upon completion of this unit, you will be able to:

- Understand the techniques for Big Data Analytics
- Discuss the phases of the data analytics project life cycle
- Obtain an insight on the Big Data analytics problems
- Identify tools for Big Data analytics
- Assess the importance of Machine Learning
- Differentiate between the supervised and unsupervised machine learning algorithms
- Apply supervised and unsupervised algorithms using SparkR on a dataset

6.2 Introduction to Big Data Analytics

Analyzing Big Data follows a different path from traditional systems. Big Data analytics refers to “a set of procedures and statistical models to extract the information from a large variety of data sets” (Kune et al., 2016). Big data analytics can provide valuable insights that may provide substantial advantages. This section highlights the main Big Data techniques:

- Text Analytics

A vast proportion of unstructured data comes from social media, email and newspapers and is therefore in textual format (Chen et al., 2012). Text analytics derives information from these textual sources (Kune et al., 2016). Modern text analytics make use of statistical models and text mining to extract valuable information from vast amount of data.

- In Memory Analytics

In-memory analytics is an approach used “for querying data when it resides in a computer’s random access memory (RAM), as opposed of querying data stored on physical disks” (Kune et al., 2016). The adoption of in-memory analytics has led to a paradigm shift where the technique has resulted in faster query and calculation and improved performance (Hota, 2013). This has resulted in quicker decision making for businesses (Kune et al., 2016).

- Graph Analytics

Graph analytics is another technique that is widely adopted to analyse large volume of data. It studies the behavior of connected components such as social networks (Kune et al., 2016). Additionally, the technique extracts intelligence between data sets by inferring paths through complex relationships. A number of graph analytics frameworks such as GraphLab, CombBLAS, Giraph, SociaLite and Galois exist (Satish et al., 2014).

- Statistical methods

Statistical methods are used "to exploit relationships and causal relationships between different objectives" (Chen and Zhang, 2014). However, traditional statistical methods are not appropriate to manage Big Data, according to Chen and Zhang (2014).

- Data Mining

"Data mining has form a branch of applied artificial intelligence" and its main purpose is to retrieve required data from large amount of data (Liao et al, 2012). There are various techniques such as classification, clustering, pattern matching that are used in data mining. Different algorithms such as k-means, clustering, decision forest algorithms and regression trees are available for processing of data, calculation and reasoning. The algorithms for data mining comprise of three parts namely the model, preference criterion and the searching algorithm (Fayyad, 1996). The model can be either classification or clustering and the search algorithm is used to find particular model or attributes. Data mining also involves dynamic prediction which is suitable for the use of healthcare purposes such as diagnosis.

- Machine Leaning

Machine learning (ML) is defined as a "field of study that gives computers the ability to learn without being explicitly programmed" (Samuel, 2000). It is a branch of artificial intelligence (AI) that uses various statistical, probabilistic and optimization techniques to allow computers to "learn" from previous examples. It is used to detect complex pattern from huge and complex data sets (Cruz and Wishart, 2006). ML concepts are used to enable applications to take a decision from the available datasets. ML is a field that has covered nearly every scientific domain, which has eventually had a great impact on the science and society.

- Social Media Analytics

Social Media analytics refer to "the analysis of structured and unstructured data from social media channels" (Gandomi and Haider, 2015). Social media analytics are classified into content-based analytics and structure based analytics. Content-based analytics analyse data posted by users whereas structure based analytics analyse data with respect to structural attributes of a social network and determine links and intelligence between relationships among the participating entities. A number of techniques have emerged to extract data from the structure of a social network. Some of these include community detection, social influence analysis and link prediction (Gandomi and Haider, 2015).

- Predictive Analytics

Predictive analytics aim to uncover patterns and relationships in data by using above discussed techniques such as optimization methods, statistical methods, data mining, machine learning, visualization approaches and social media analytics (Gandomi and Haider, 2015). Predictive analytics seek to predict the future by analyzing current and historical data (Kune et al., 2016).

6.3 Big Data Analytics Lifecycle

Data science projects are different from most traditional Business Intelligence. They are more exploratory in nature. Thus, it is of utmost importance to consider a process that can govern the development that allows exploration (Dietrich et al., 2015). It is useful to consider a framework that would help in the organization of the work and to obtain a clear insight of the big data. Thus, a framework consisting of some stages have been identified so that the expected output can be obtained (Prajapati, 2013). The framework can be termed as the big data analytics lifecycle. The stages of this lifecycle are not linear, that is, they are related to each other. The data analytics processes that are defined in the lifecycle should be followed sequentially so that proper mining and analytics are achieved. The main processes of the data analytics lifecycle are shown in Figure 6.1 and further described.

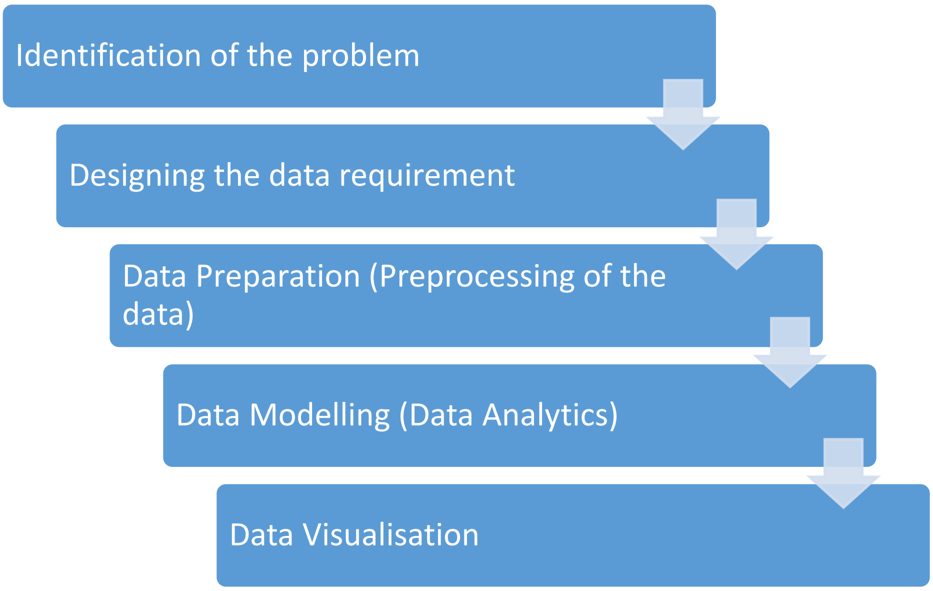

Identification of the problem
Designing the data requirement
Data Preparation (Preprocessing of the data)
Data Modelling (Data Analytics)
Data Visualisation

Figure 6.1: Data Analytics Project Lifecycle

Identification of the Problem

This phase is also termed as discovery. In this process, the main focus is understanding the requirements and objectives of the project from a business perspective. It is essential for the team to understand the business domain and to convert this knowledge into a data mining problem definition. An initial plan is devised and a decision model is often used. In this part, the team also assesses the resources available to support the project in terms of people, technology, time, and data (Tutorial point, 2015).

Designing the data requirement

In this phase, the datasets to be used for the data analytics are identified. Data to be used are collected and the attributes of the datasets are being defined based on the domain and the problem specification. This phase is important since it allows the user to discover first insights into the data and to determine relevant and interesting subsets to form hypotheses for hidden information.

Data Preparation (Preprocessing of the data)

The final outcome of the data preparation phase is to obtain the final dataset that will be used for modelling. In data analytics, various formats of data are required at different time, that is, different applications would require different data sources, algorithms and attributes. Thus, it is important to provide the data in a format so that all the algorithms and data tools can use. In this phase, the data is cleansed, aggregated, augmented, sorted and formatted. Thus, after preprocessing, a fixed data set format is generated.

Data Modelling (Data Analytics)

Data modelling, also known as data analytics, is performed to discover meaningful information from the data. There are several techniques such as regression, classification and clustering, that can be used to analyse and discover patterns in data. Likewise, the same algorithms can be used for big data where the data analytics can be sent to the MapReduce job. Data analytics enable organisations to deal with large volume of data. In this phase, the user understands the relationship between the features and consequently devise data mining methods that can be used for prediction. Machine learning techniques are capable of discovering patterns in very large datasets and this is useful for decision making (Khan *et al*, 2014).

Data Visualisation

Data visualisation is the phase where the output of data analytics is being displayed. Visualisation is an interactive way of representing the results. Plots and charts can be used to visualize the data by using the required packages available in the data visualisation software.

6.4 Big Data Analytics Problems

In this section, some examples of data analytics problems are being described. This will allow you to understand the relevant techniques and functions to perform data analytics. Various packages are available in R along with the computational power of Hadoop, analytics and predictions.

Big Data Analytics and Deep Learning are gaining much attention with regards to data science. Companies are in possession of huge amount of information regarding problems such as national intelligence, cyber security, fraud detection, marketing, and medical informatics (Najafabadi et al., 2015). Deep Learning algorithms are analyzing massive amount of unsupervised data and thus becoming an important tool for data analytics. Complex patterns are being extracted, which are eventually helpful for decision making. Analysis is done on large data sets to uncover correlations to find business trends, combat crime, prevent diseases amongst others.

6.4.1. Semantic Indexing

There exist different types of data namely text, audio, video and images that are available across social networks, marketing applications, shopping systems, security systems, fraud detection applications amongst others. The efficient storage and retrieval of these information are becoming a challenging task. Thus, semantic indexing can be used instead of storing the data using big strings. Semantic indexing stores the data in a more efficient way, thus making it useful for discovery. Complex associations between the data and factors can be depicted. For semantic indexing, deep learning can be used to generate high level data abstraction of data instead of using raw input (Najafabadi et al., 2015). The packages available in R and other data analytics software are capable of uncovering the underlying trends and patterns of the data.

6.4.2. Exploring web pages categorization

In web analytics, it is important to determine the importance of web pages. Based on the information such as content, colours, design, no of visits, ease of navigations and other details, the webpages can be customized accordingly. Data collected with respect to the data, source, title and page path are captured. After data collection, it is subject to the MapReduce algorithm. Depending on the popularity of the website, it can be categorized as high, medium, or low (Prajapati, 2013).

6.4.3. Prediction of Cancer

Despite the rapid advancement in technology, the early detection and prognosis of cancer is still a challenge. Detection of cancer is concerned with the analysis of petabytes of data. This involves high dimensional data, which are collected from various sources such as scientific experiments, literature, computational analysis and research. Prognosis is being used to determine the survival pattern using various attributes such as specific drug administered to a patient, treatment given and response of the patient. Lots of data are involved and thus data analytics can be used to determine trends and patterns which will eventually help doctors in taking the proper decisions. Data mining techniques can be used to determine trends and acquire knowledge using the information available.

6.5 Big Data Analytics using Machine Learning Techniques

Machine learning (ML) focuses on the data analysis using different statistical tools and learning processes to obtain more knowledge from the data (Faruk and Cahyono, 2018). ML has been applied on many problems such as recognition systems, informatics and data mining and autonomous control systems (Qui et al., 2016). The ways of defining the machine learning algorithms differs but they are commonly subdivided mainly into supervised learning, unsupervised learning and semi-supervised learning which combines both supervised learning and unsupervised learning.

6.5.1 Supervised Machine Learning algorithms

Supervised learning is defined as where a function that maps an input variable to an output variable based on example input-output pairs. Linear Regressions and Random Forest are the two examples of supervised learning algorithms that will be considered for this course.

6.5.1.1 Linear Regressions

Linear regression is a statistical tool that is mainly used for predicting and forecasting values based on historical information subject to some important assumptions:

- There requires a dependent variable and a set of independent variables
- There exist a linear relationship between the dependent and the independent variables, that is:

$$y = a_1x_1 + a_2x_2 \, , \dots + b + e$$

Where

- y : is the response variable.
- xj : is the predictor variable j where j=1,2,3,……….p.
- e : is the error term that is normally distributed with mean 0 and constant variance
- aj and b: are the regression coefficients to be estimated the coefficients

Regression is a technique used to identify the linear relationship between target variables and explanatory variables (Prajapati, 2013). Other terms are also used to describe the variable. One of these variable is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

6.5.1.2 Logistic Regression

In statistics, logistic regression is known to be a probabilistic classification model. Logistic regression is widely used in many disciplines, including the medical and social science fields. Logistic regression can be either binomial or multinomial. It is very popular to predict a categorical response. Binary logistic regression is used in cases where the outcome for a dependent variable have two possibilities. As for multinomial, the logistic regression is concerned with possibilities where there are three or more possible types.

Using logistic regression, the input values (x) are combined linearly using weights or coefficient values to predict an output value (y) based on the log odds ratio. One major difference between linear regression and logistic regression is that in linear regression, the output value being modeled is a numerical value while in logistic, it is a binary value (0 or 1) (Prajapati, 2013).

The logistic regression equation can be given as follows:

$$Py = e^{\wedge}(b0 + b1*x_i) / (1 + e^{\wedge}(b0 + b1*x_i))$$

Where Py is the expected probability for the y(f) subject, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x_i). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

It is quite simple to make predictions using logistic regression since there is a need to plug in numbers into the logistic regression equation to obtain the output.

6.5.1.3 Random Forest

Random forests (RF) are known to be very popular for the classification and regression methods. RF is a very powerful machine learning algorithm which is concerned with combining the decisions output from several trees (Faruk and Cahyono, 2018). In fact, they combine tree predictors in such a way that each tree depends on the values of a random vector sampled independently. The generalization error of a forest of tree classifiers relies on the strength of the individual trees in the forest and the correlation between them (Breiman, 2001). RF follows the same principles of the decision trees but it does not select all the data points and variables in each of the trees. It randomly samples data points and variables in each of the tree that it creates and then combines the output at the end. It removes the bias that a decision tree model might introduce in the system. It also improves the predictive power significantly.

6.5.2 Unsupervised Machine Learning algorithms

Compared to supervised learning, unsupervised learning makes use of input data but have no corresponding output variables. In other words, unsupervised learning algorithms have no outcome to be predicted. They aim at discovering the structure and distribution of data in order to learn more about the data. Two examples of unsupervised learning algorithms are K-Means algorithm and Principal Components Analysis (PCA) algorithm.

6.5.2.1 K-Means Algorithm

K-Means algorithm is an unsupervised machine learning algorithm which aims at clustering data together, that is, finding clusters in data based on similarity in the descriptions of the data and their relationships (Hans-Hermann, 2008). Each cluster is associated with a center point known as a centroid. Based on the center, the length of space of each cluster with respect to the center is calculated and the clusters are formed by assigning points to the closest centroid. Various algorithms, such as Euclidean distance, Euclidean squared distance and the Manhattan or City distance, are used to determine which observation is appended to which centroid. The number of clusters is represented by the variable *K*.

6.5.2.2 Principal Components Analysis (PCA)

PCA algorithm aims at analyzing data to identify patterns and expressing the data in such a way to highlight similarities and differences (Smith, 2002). Once the patterns are found, the data can be compressed by reducing the dimensions of the dataset with minimal loss of information. This technique is commonly used for image compression and can be applied in the fields such as finance and bioinformatics to find patterns in data of high dimension.

6.6 Setting up the environment for Big Data Analytics using Spark

To be able to run R on Hadoop you now have to install packages. SparkR has been chosen as it has an in-built Machine Learning library consisting of a number of algorithms that run in memory. Proceed with the following steps to install SparkR.

Steps:

1. From the VM console type the following to install the packages for curl, libcurl and libxml (if not already installed). If they are already installed, the messages displayed will indicate same to you.

```
sudo yum -y install curl libcurl-devel
sudo yum -y install libxml2 libxml2-devel
```

2. For installing SparkR, the tutorial from http://spark.rstudio.com/ has been used.

3. Open Rstudio and install the package sparklyr as follows:

```
install.packages("sparklyr")
```

Warning: In Case the following error is being encountered:

ERROR: configuration failed for package 'stringi'

- Install the *stringi* dependency as follows:

```
install.packages(c(“stringi”),configure.args=c(“—
disable-cxxll”),repos=”https://cran.rstudio.com”)
```

- Install the sparklyr package again as follows:

```
install.packages("sparklyr")
```

4. Install local version of R

```
library(sparklyr)
spark_install(version = "2.1.0")
```

5. Install latest version of sparklyr

```
Install.packages(c('devtools','curl'))

devtools::install_github("rstudio/sparklyr")
```

Your SparkR environment should now be ready to use.

6. You need to connect to the local instance of Spark and remote Spark clusters.

Use the following codes to connect to a local instance of Spark via the *spark_connect* function:

```
library(sparklyr)
library(dplyr)
sc <- spark_connect(master = "local")
```

7. Read the dataset

Note that before reading any dataset in Spark, you need to upload the database from the source (if it is not already in the Spark) using the following codes:

```
databaseName <- read.csv("name of the databse. csv")
```

8. Load data from R dataset to Spark

```
library(dplyr)
tableName_tbl <- copy_to(sc, TableName)
```

6.7 Applying supervised Machine Learning techniques using Spark

This section provides examples of how supervised Machine Learning techniques linear regression, random forest and the logistic regression can be applied to a dataset based on the following case study.

Case Study Description

The case study chosen is on the birth weights of babies. The dataset has been downloaded from the following link: https://www.sheffield.ac.uk/mash/data. It contains details on the weight of newborn babies and their parents. The dataset contains mostly continuous variables that is most useful for correlation and regression. Supervised techniques are applied on the dataset to determine which variables have an influence on the babies' birthweight. The table below describes the different variables that have been used for the birthweight datasets.

Name	Variable	Data type
ID	Baby number	
length	Length of baby (inches)	Scale
Birthweight	Weight of baby (lbs)	Scale
headcirumference	Head Circumference	Scale
Gestation	Gestation (weeks)	Scale
smoker	Mother smokes 1 = smoker 0 = non-smoker	Binary
motherage	Maternal age	Scale
mnocig	Number of cigarettes smoked per day by mother	Scale
mheight	Mothers height (inches)	Scale
mppwt	Mothers pre-pregnancy weight (lbs)	Scale
fage	Father's age	Scale
fedyrs	Father's years in education	Scale
fnocig	Number of cigarettes smoked per day by father	Scale
fheight	Father's height (inches)	Scale
lowbwt	Low birth weight, 0 = No and 1 = yes	Binary
mage35	Mother over 35, 0 = No and 1 = yes	Binary

6.7.1 Linear Regression

The first supervised technique that is being applied on the birthweight dataset is the linear regression.

1. The libraries have to be imported and the database is loaded as explained in section 6.6 (Steps 1 -3)

2. Load the data in R dataset

```
birthweight <- read.csv("birthweight_reduced.csv")
birthweight_tbl <- copy_to(sc,birthweight)
head(birthweight_tbl)
```

The details of the dataset can be obtained by using head( ) function. After running the codes, the following is being displayed:

```
# Database: spark_connection
    id headcirumference length Birthweight Gestation smoker motherage mnocig mheight
mppwt

  <int>          <int>  <int>     <dbl>     <int>  <int>     <int>  <int>   <int> <int>

1  1313            12     17       5.8       33     0       24      0      58    99

2   431           12     19       4.2       33     1       20      7      63   109

3   808           13     19       6.4       34     0       26      0      65   140

4   300           12     18       4.5       35     1       41      7      65   125

5   516           13     18       5.8       35     1       20     35      67   125

6   321           13     19       6.8       37     0       28      0      62   118

# ... with 7 more variables: fage <int>, fedyrs <int>, fnocig <int>, fheight <int>,

#   lowbwt <int>, mage35 <int>, LowBirthWeight <chr>
```

The summary can be obtained by using the function summary ( )

```
summary(birthweight_tbl)
```

3. Apply the linear regression function on the dataset. To apply the linear regression, it is important to know which factor to take as the response variable and which one(s) to take as predictor variable.

4. Different linear models are formulated based on questions 1, 2 and 3.

Question 1: Do all the factors listed in the birthweight table influence the birthweight?

The lm_model1 ( ) is formulated as follows based on question 1:

```
lm_model1<- birthweight_tbl %>% select(LowBirthWeight,length,motherage,smoker,
Gestation,Birthweight, headcirumference, smoker, motherage, mnocig, mppwt, fage, fedyrs,
fnocig, fheight, mage35) %>%
ml_linear_regression(LowBirthWeight~length+motherage+smoker+
Gestation+Birthweight+ headcirumference+ smoker+ motherage + mnocig + mppwt + fage
+ fedyrs + fnocig + fheight + mage35)

summary (lm_model1)
```

Note that it if formulated based on the equation of the linear model.

After running the above codes, the following results are obtained:

```
Deviance Residuals:

    Min      1Q   Median      3Q      Max

-0.42533 -0.10785 -0.01961  0.11140  0.42949
```

```
Coefficients:

    (Intercept)         length       motherage         smoker       Gestation

    3.522021303    -0.119732120    -0.010909114     0.101237784    -0.031251285

    Birthweight headcirumference          mnocig           mppwt            fage

   -0.013699240    -0.031169860    -0.005629405    -0.002252918    -0.003029553

         fedyrs          fnocig         fheight          mage35

   -0.008982498     0.004628336     0.019740191     0.428123876
```

```
R-Squared: 0.6456

Root Mean Squared Error: 0.2083
```

Interpretation of Results

It can be seen that there are coefficients that are being displayed for the various fields. These values represent the percentage influence that these predictor variables have on the response variable, that is, their influence on the outcome. For example: length has a value of -0.119732120, which shows that as length increases, it decreases (negative value in front of the coefficient) the chance of having a low birth weight by 11.9%. The value of smoker is 0.101237784, which shows that, if the mother smokes, the chance of having a low birth weight increases by 10.1%.

Activity 1:

Interpret the results of the other variables based on the value of the coefficients.

Hint: When the coefficient is negative, the influencing variables decrease the chance of a low birth weight.

Question 2: Do mother's height and father's height have any influence on the baby's length?

A second linear model is formulated as follows based on question 2:

```
lm_model2<- birthweight_tbl %>% select(length,mheight, fheight) %>%
ml_linear_regression(length~ mheight+ fheight)

summary(lm_model2)
```

```
R-Squared: 0.1779

Root Mean Squared Error: 0.997
```

The following results is obtained:

```
Deviance Residuals:

    Min       1Q   Median       3Q      Max

-2.44405 -0.68707 -0.03086  0.70399  2.33639
```

```
Coefficients:

(Intercept)     mheight     fheight

 6.65996234  0.17164827  0.03128298
```

Interpretation of Results

It can be seen that there are coefficients that are being displayed for the various fields. These values represent the percentage contribution that these attributes have on the result, that is, their influence on the outcome. For example: mheight has a value of 0.17164827, which shows that mheight has a 17% contribution in determining the length of the child. As for fheight, it has only 3% contribution in determining the baby's length.

Question 3: Does gestational age and smoking have an influence on the birthweight?

A third linear model is formulated based on question 3 and is as follows:

```
lm_model3<- birthweight_tbl %>% select(LowBirthWeight, Gestation, smoker) %>%
ml_linear_regression(LowBirthWeight~ smoker+ Gestation)

summary (lm_model3)
```

```
R-Squared: 0.4022

Root Mean Squared Error: 0.2705
```

The following results is obtained:

```
Deviance Residuals:

    Min       1Q   Median       3Q      Max

-0.47660 -0.16350 -0.06716  0.13003  0.77629
```

```
Coefficients:

(Intercept)      smoker   Gestation

 3.13796250  0.13849085 -0.07827535
```

Interpretation of Results

From the result, it can be deduced that smoking has 13% of influence on the low birthweight. This means that the variable smoke increases the chance of having a low birthweight. As for Gestation, the chance of having a low birth weight is decreased by 7% upon increasing the gestation age.

5. Prediction

Prediction is used to estimate a particular value based on the model formulated. In this case, we want to predict the birthweight using the influencing factors as derived in the linear model.

To do the prediction, we have partitioned our dataset into two categories namely: training and test. In this example we have taken 50% of the dataset for the training and 50% for the test.

The following code is applied for the partitioning:

```
partitions <- birthweight_tbl %>% sdf_partition(training=0.5, test=0.5)
```

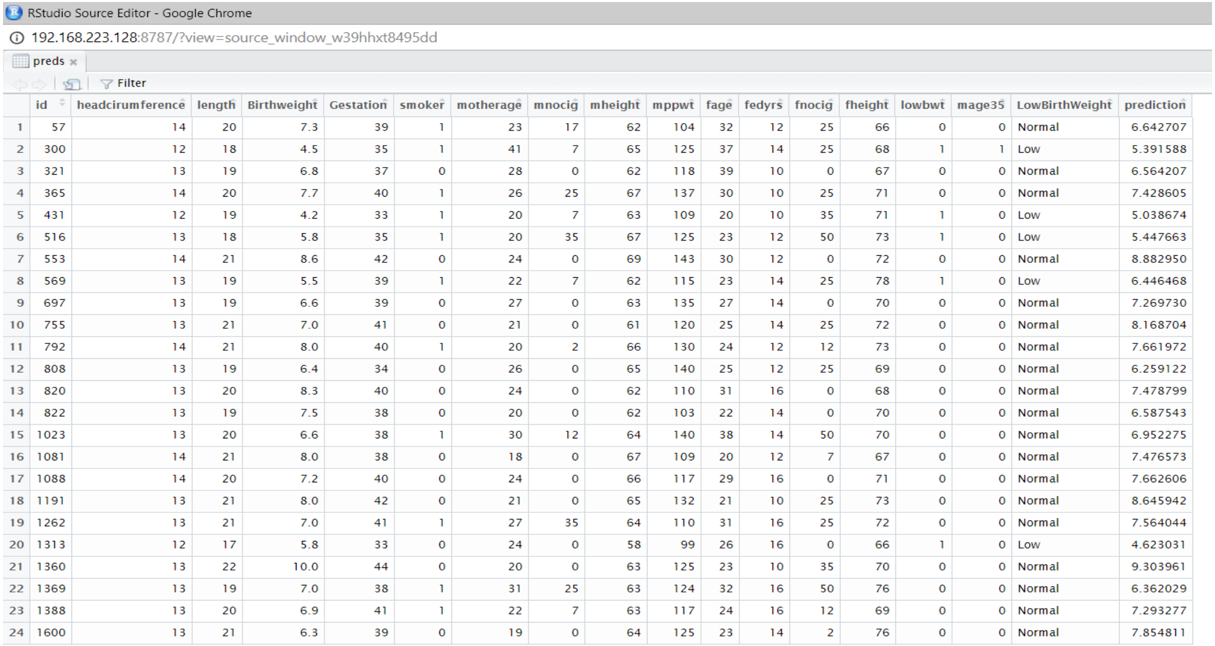
To formulate the prediction based on linear model 3, the following codes are used:

```
preds <- sdf_predict(partitions$test, lm_model3)
```

To obtain the values of the predicted field (in this case birthweight), the following code is used:

```
View(preds)
```

A column prediction is appended to the original table. This column shows the predicted value of the birthweight generated from the model applied. These values are compared with the birthweight already in the table. It can be seen that for nearly all the values, the predicted birthweight is close to the real birthweight, which indicates that linear regression is an appropriate model that can be used for this type of data.


RStudio Source Editor - Google Chrome
192.168.223.128:8787/?view=source_window_w39hhxt8495dd
preds
Filter

	id	headcirumference	length	Birthweight	Gestation	smoker	motherage	mnocig	mheight	mppwt	fage	fedyrs	fnocig	fheight	lowbwt	mage35	LowBirthWeight	prediction
1	57	14	20	7.3	39	1	23	17	62	104	32	12	25	66	0	0	Normal	6.642707
2	300	12	18	4.5	35	1	41	7	65	125	37	14	25	68	1	1	Low	5.391588
3	321	13	19	6.8	37	0	28	0	62	118	39	10	0	67	0	0	Normal	6.564207
4	365	14	20	7.7	40	1	26	25	67	137	30	10	25	71	0	0	Normal	7.428605
5	431	12	19	4.2	33	1	20	7	63	109	20	10	35	71	1	0	Low	5.038674
6	516	13	18	5.8	35	1	20	35	67	125	23	12	50	73	1	0	Low	5.447663
7	553	14	21	8.6	42	0	24	0	69	143	30	12	0	72	0	0	Normal	8.882950
8	569	13	19	5.5	39	1	22	7	62	115	23	14	25	78	1	0	Low	6.446468
9	697	13	19	6.6	39	0	27	0	63	135	27	14	0	70	0	0	Normal	7.269730
10	755	13	21	7.0	41	0	21	0	61	120	25	14	25	72	0	0	Normal	8.168704
11	792	14	21	8.0	40	1	20	2	66	130	24	12	12	73	0	0	Normal	7.661972
12	808	13	19	6.4	34	0	26	0	65	140	25	12	25	69	0	0	Normal	6.259122
13	820	13	20	8.3	40	0	24	0	62	110	31	16	0	68	0	0	Normal	7.478799
14	822	13	19	7.5	38	0	20	0	62	103	22	14	0	70	0	0	Normal	6.587543
15	1023	13	20	6.6	38	1	30	12	64	140	38	14	50	70	0	0	Normal	6.952275
16	1081	14	21	8.0	38	0	18	0	67	109	20	12	7	67	0	0	Normal	7.476573
17	1088	14	20	7.2	40	0	24	0	66	117	29	16	0	71	0	0	Normal	7.662606
18	1191	13	21	8.0	42	0	21	0	65	132	21	10	25	73	0	0	Normal	8.645942
19	1262	13	21	7.0	41	1	27	35	64	110	31	16	25	72	0	0	Normal	7.564044
20	1313	12	17	5.8	33	0	24	0	58	99	26	16	0	66	1	0	Low	4.623031
21	1360	13	22	10.0	44	0	20	0	63	125	23	10	35	70	0	0	Normal	9.303961
22	1369	13	19	7.0	38	1	31	25	63	124	32	16	50	76	0	0	Normal	6.362029
23	1388	13	20	6.9	41	1	22	7	63	117	24	16	12	69	0	0	Normal	7.293277
24	1600	13	21	6.3	39	0	19	0	64	125	23	14	2	76	0	0	Normal	7.854811

6.7.2 Logistic Regression

1. The libraries have to be imported and the database is loaded as explained in section 6.6 (Steps 1 -3).

2. Load the data in R dataset

```
birthweight <- read.csv("birthweight_reduced.csv")
birthweight_tbl <- copy_to(sc, birthweight)
```

3. Select the appropriate table, fields and labels that will be used to formulate the glm ( ) model and that will be used for prediction.

```
birthweight$LowBirthWeight <- factor(birthweight$LowBirthWeight, labels = c("Low",
"Normal"))

 birthweight_tbl <- tbl(sc, "birthweight")
```

4. Apply the logistic regression model. Note that there are many variables as discussed in the case study. These variables have been recorded to determine whether they have any influence on normal birthweight or low birthweight. Different models can be formulated.

In the first case, the variable **smoker** will be used to build the model. The following question can be formulated:

Quesion 1: Does the variable smoke has any effect on the baby birthweight?

The glm_model( ) function is used as follows:

glm_model1 (smoker as variable)

```
glm_model1 <- birthweight_tbl %>%

  mutate(binary_response = as.numeric(LowBirthWeight == "Normal")) %>%

  ml_logistic_regression(binary_response ~ smoker)
```

```
glm_model1
```

The following result is obtained:

```
Formula: binary_response ~ smoker

Coefficients:

(Intercept)     smoker

  -2.944439    1.720663
```

Interpretation:

Replacing the value 1.720663 in the logistic regression formula, log odds increase by (exp(1.720663)-1) which is a value of 45.8% for a mother who smoke as compared to a mother who does not smoke. This means that the risk of low birth weight increase by 45.8% for a mother who smokes compared to a mother who does not smoke.

Question 2: Does the Gestation variable has any effect on the baby birthweight?

glm_model2 (Gestation as variable)

```
glm_model2 <- birthweight_tbl %>%

  mutate(binary_response = as.numeric(LowBirthWeight == "Normal")) %>%

  ml_logistic_regression(binary_response ~ Gestation)

glm_model2
```

The following result is obtained:

```
Formula: binary_response ~ Gestation

Coefficients:

(Intercept)   Gestation

 31.3126582  -0.8773096
```

Interpretation:

Replacing the value -0.8773096 in the logistic regression formula, log odds decrease by (exp(-0.8773096)-1) which is a value of 58.4% . This means that the risk of low birth weight decreases by 58.4% for the gestation variable. Thus, birth weight value being low is not influenced by a high gestation. That is why, the log odds decreases by 58.4%.

Question 3: Do both the variables smoke and gestation have any effect on the baby birthweight?

A third logistic model is formulated based on question 3 and is as follows:

```
glm_model3 <- birthweight_tbl %>%

  mutate(binary_response = as.numeric(LowBirthWeight == "Normal")) %>%

  ml_logistic_regression(binary_response ~ Gestation + smoker)

glm_model3
```

The following result is obtained:

```
Formula: binary_response ~ Gestation + smoker

Coefficients:

(Intercept)   Gestation     smoker

  49.162036   -1.468332    5.484319
```

Activity 2

Interpret the results obtained with respect to the Gestation and smoker variables.

Hint: the exp ( ) has to be computed for both variables separately using the coefficients generated.

5. Prediction

To formulate the prediction based on glm model 3, the following codes are used:

```
predicted <- sdf_predict(glm_model3, birthweight_tbl)
View(predicted)
```

A new table *predicted* is created with a number of additional variables. The actual values of the birthweight and the predicted value can easily be compared.

```
Database: spark_connection
      id headcirumference length Birthweight Gestation smoker motherage
mnocig mheight mppwt  fage fedyrs fnocig fheight lowbwt
   <int>            <int>  <int>       <dbl>     <int>  <int>     <int>
<int>   <int> <int> <int>  <int>  <int>   <int>  <int>
 1  1313               12     17         5.8        33      0        24
0      58    99    26     16      0      66      1
 2   431               12     19         4.2        33      1        20
7      63   109    20     10     35      71      1
 3   808               13     19         6.4        34      0        26
0      65   140    25     12     25      69      0
 4   300               12     18         4.5        35      1        41
7      65   125    37     14     25      68      1
 5   516               13     18         5.8        35      1        20
35      67   125    23     12      50      73      1
 6   321               13     19         6.8        37      0        28
0      62   118    39     10      0      67      0
 7  1363               12     19         5.2        37      1        20
7      64   104    20     10     35      73      1
 8   575               12     19         6.1        37      1        19
7      65   132    20     14      0      72      0
 9   822               13     19         7.5        38      0        20
0      62   103    22     14      0      70      0
10  1081               14     21         8          38      0        18
0      67   109    20     12      7      67      0
# ... with more rows, and 9 more variables: mage35 <int>, LowBirthWeight
<chr>, features <list>, rawPrediction <list>,
#   probability <list>, prediction <dbl>, predicted_label <chr>,
probability_1_0 <dbl>, probability_0_0 <dbl>
```

Lab Activity 1:

1. Apply the Random Forest model (described in section 6.5) on the birthweight dataset.

2. Display and interpret the result.

Hint:

```
rf_model <- tableName_tbl %>%

  ml_random_forest(Variable To be determined, type = "classification")
```

3. Develop the prediction model and interpret the values obtained.

6.8 Applying unsupervised Machine Learning Techniques

This section provides examples of how unsupervised Machine Learning techniques namely K-Means algorithm can be applied to a dataset based on the following case study.

Case Study Description

The Breast Cancer dataset has been chosen for demonstrating the K-Means algorithm. According to Dubey et al. (2016), clustering is an important activity that enables grouping of data based on the nature or a symptom of the disease. The Breast Cancer Wisconsin has been downloaded from UCI Repository https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+ (original). The details of the fields are given in table below. The dataset has 11 variables with 699 observations, first variable is the identifier and has been excluded in the analysis. Thus, there are **9 predictors** and **a response** variable (class). The response variable denotes "Malignant" or "Benign" cases.

Name	Variable
Clump thickness (CT)	It indicates grouping of cancer cells in multilayer.
Uniformity of cell size (UCS)	It indicates metastasis to lymph nodes.
Uniformity of cell shapes (UCSh)	It identifies cancerous cells, which are of varying size.
Marginal adhesion (MA)	It suggests loss of adhesion, i.e., a sign of malignancy but the cancerous cells lose this property. ~~So~~ This retention of adhesion is an indication of malignancy.
Single epithelial cell size (SECS)	If the SECS becomes larger, it may be a malignant cell.
Bare nuclei (BN)	Bare nuclei without cytoplasm coating which are found in benign tumors.
Bland chromatin (BC)	It usually found in benign cell.
Normal nucleoli (NN)	It is generally very small in benign cells.
Mitoses	It is the process in cell division by which nucleus divides.

Steps:

1. The libraries have to be imported and the database is loaded as explained in section 6.7 (Steps 1 -3)

2. Load the data in R dataset.

```
library(sparklyr)

library(ggplot2)

library(dplyr)

sc <- spark_connect(master="local")

library(readxl)

BreastCancerData <- read_excel("/home/cloudera/Downloads/BreastCancerData.xlsx")
```

3. Select the appropriate table, fields and labels that will be used to formulate the K-Means model. The predictors Uniformity*_of_Cell_Shape* and *Uniformity_of_Cell_Size* are chosen to formulate the KMeans model.

4. Write the codes for the KMeans model.

```
breastcancer_tbl <- copy_to(sc,BreastCancerData, "BreastCancerData", overwrite = TRUE)


kmeans_model <- breastcancer_tbl %>%

  ml_kmeans(formula= ~ Uniformity_of_Cell_Shape + Uniformity_of_Cell_Size, centers = 2)

kmeans_model
```

The output is displayed:

```
Cluster centers:
  Uniformity_of_Cell_Shape Uniformity_of_Cell_Size
1                 7.325000                7.415000
2                 1.557114                1.418838

Within Set Sum of Squared Errors =  2781.015
```

5. Prediction is made based on the model. Write the following codes.

```
predicted <- sdf_predict(kmeans_model, breastcancer_tbl) %>%
  collect
table(predicted$Class_2benign_4malignant, predicted$prediction)
```

The output below is displayed:

```
      0   1
  2   9 449
  4 191  50
```

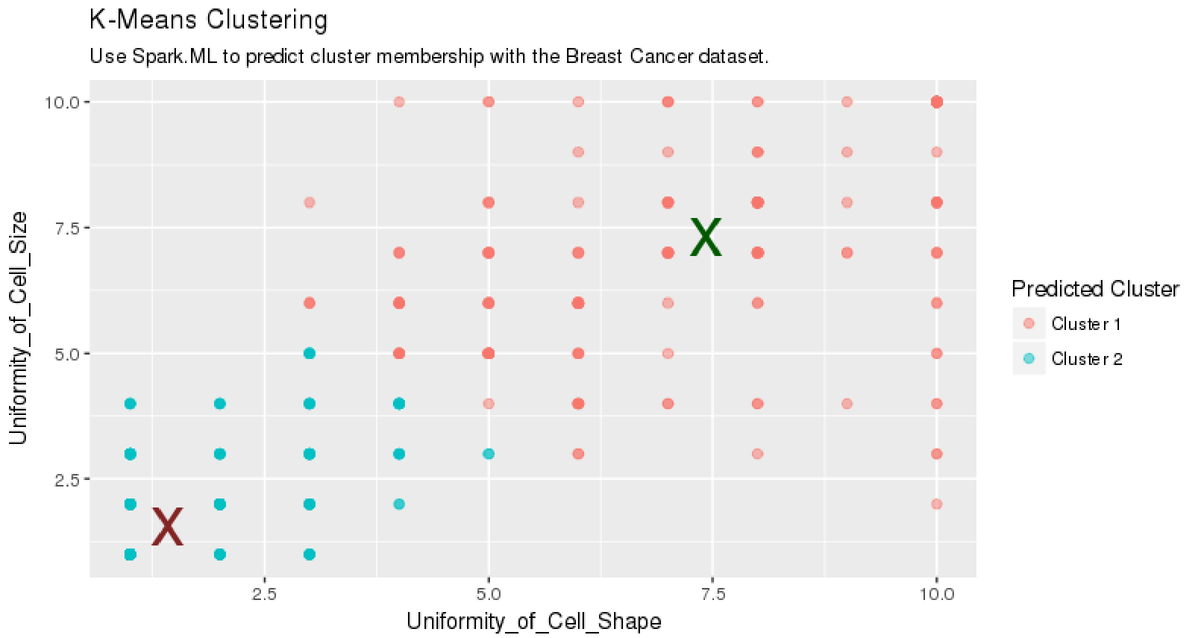
Interpretation of results

449 of the benign cases have been classified under 1 and 9 cases have been misclassified. 191 of the malignant cases have been grouped together and 50 have been misclassified.

6. Display the results in the form of a graph using the following codes.

```
# plot cluster membership
sdf_predict(kmeans_model) %>%
  collect() %>%
  ggplot(aes(Uniformity_of_Cell_Shape, Uniformity_of_Cell_Size)) +
  geom_point(aes(Uniformity_of_Cell_Size, Uniformity_of_Cell_Shape, col = factor(prediction + 1)),
             size = 2, alpha = 0.5) +
  geom_point(data = kmeans_model$centers, aes(Uniformity_of_Cell_Size,
Uniformity_of_Cell_Shape),
             col = scales::muted(c("green", "red")),
             pch = 'x', size = 12) +
  scale_color_discrete(name = "Predicted Cluster",
                       labels = paste("Cluster", 1:2)) +
  labs(
    x = "Uniformity_of_Cell_Shape",
    y = "Uniformity_of_Cell_Size",
    title = "K-Means Clustering",
    subtitle = "Use Spark.ML to predict cluster membership with the Breast Cancer dataset."
  )
```

The following graph is displayed showing two clusters indicated by **X**.



Activity 3

Change the predictor variables and see the effect on the clusters formed.

Lab Activity 2:

1. Apply the PCA model (described in section 6.5) on the Breast Cancer dataset.

2. Display and interpret the result.

Hint:

Use ml_pca()

3. Develop the prediction model and interpret the values obtained.

6.10 Unit Summary

In this Unit, you learned the different techniques that can be used for Big Data analytics. You became familiar with the Big Data analytics lifecycle and problems where Big Data analytics can be applied. You became acquainted with supervised and unsupervised Machine Learning techniques. Some of the algorithms have been applied to large data sets using Spark.

6.11 Additional Readings

1. Zikopoulos, P. and Eaton, C., 2011. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media.

2. Pentreath, N., 2015. *Machine Learning with Spark*. Packt Publishing Ltd.

3. Kotsiantis, S.B., Zaharakis, I. and Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, pp.3-24.

4. Random Forests, https://www.r-bloggers.com/how-to-implement-random-forests-in-r/

5. Logistic Regression, https://machinelearningmastery.com/logistic-regression-for-machine-learning/

6.12 References

Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.

Chen, C.P. and Zhang, C.Y., 2014. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, pp.314-347.

Chen, H., Chiang, R.H. and Storey, V.C., 2012. Business intelligence and analytics: from big data to big impact. *MIS quarterly*, pp.1165-1188.

Chen, M., Mao, S. and Liu, Y., 2014. Big data: A survey. *Mobile Networks and Applications*, *19*(2), pp.171-209.

Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. Cancer informatics, 2, 117693510600200030.

Dietrich, D., B. Heller, and B. Yang. "Data Science and Big Data Analytics: Discovering." *Analyzing, Visualizing and Presenting Data* (2015).

Gandomi, A. and Haider, M., 2015. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), pp.137-144.

Faruk, A., & Cahyono, E. S. (2018). Prediction and Classification of Low Birth Weight Data Using Machine Learning Techniques. *Indonesian Journal of Science and Technology*, *3*(1), 18-28.

Fayyad, U.M., 1996. Data mining and knowledge discovery: Making sense out of data. *IEEE Expert: Intelligent Systems and Their Applications*, *11*(5), pp.20-25.

Hans-Hermann, B.O.C.K., 2008. Origins and extensions of the k-means algorithm in cluster analysis. *Journal Electronique d'Histoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistics*, *4*(2).

Hota, J., 2013. Adoption of in-memory analytics. *CSI Communications,* pp.20-22.

Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Ali, M., Kamaleldin, W., ... & Gani, A. (2014). Big data: survey, technologies, opportunities, and challenges. The Scientific World Journal, 2014.

Liao, S.H., Chu, P.H. and Hsiao, P.Y., 2012. Data mining techniques and applications–A decade review from 2000 to 2011. *Expert systems with applications*, *39*(12), pp.11303-11311.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1.

Prajapati, V., 2013. *Big data analytics with R and Hadoop*. Packt Publishing Ltd.

Samuel, A., 2000. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 44(1.2):206–226.

Satish, N., Sundaram, N., Patwary, M.M.A., Seo, J., Park, J., Hassaan, M.A., Sengupta, S., Yin, Z. and Dubey, P., 2014, June. Navigating the maze of graph analytics frameworks using massive graph datasets. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 979-990). ACM.

Smith, L.I., 2002. *A tutorial on principal components analysis*.

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing, 2016(1), 67.

Tutorial Point 2015, Available from: https://www.tutorialspoint.com/big_data_analytics/big_data_analytics_lifecycle.htm [Last accessed: March 2018]