International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Oct.-2021)

http://dx.doi.org/10.12785/ijcds/100189

Machine Learning model to predict the number of cases contaminated by COVID-19

Allae Erraissi[1] and Mouad Banane[2]

[1] *Chouaib Doukkali University, El Jadida, Morocco*
[2] *Laboratory of Artificial Intelligence & Complex Systems Engineering (AICSE), ENSAM Casablanca*

Received 11 May. 2020, Revised 22 Jul. 2021, Accepted 05 Aug. 2021, Published 28 Oct. 2021

Abstract: This paper presents a dedicated machine learning model to predict the number of cases infected by the Corona Virus; the case of Morocco was chosen to validate this study. Completely realized in Spark ML with the 'Scala' language and tested for a certain number of algorithms generated on datasets coming from dedicated sources to gather Covid19 data in the world. The results show the possibility of achieving better scores prediction after using the proposed method. We tested our model on the case of China and the results were relevant. The proposed Machine Learning model can be applied to data from any country in the world. We have applied it in this paper to the case of Morocco and China.

Keywords: Machine Learning, Spark ML Model, Artificial Intelligence Model, Coronavirus, Predicting COVID-19.

1. Introduction

The field of artificial intelligence is undergoing a tremendous evolution these days and it risks changing the world through us categorically [1]. Fraud detection, recommendation systems, facial recognition, and decision-making automation are just a few of a variety where artificial intelligence has already taken place.

As its name suggests, artificial intelligence is intelligence (a set of skills) acquired by machines in order to function and react like humans. Machine learning [2] is a branch of artificial intelligence that concerns the design and development of algorithms that allow a computer (a machine in the broad sense) to learn to perform very complex tasks without having been explicitly programmed.

So this study aims to propose a model based on artificial intelligence, especially on machine learning [2]. This model uses a series of predefined instructions which leads a machine to generate a prediction of the number of cases infected with the Coronavirus COVID-19 in Morocco per day. Then we will test this model in the case of China since it has recorded enough data that can be used for verification. As part of this study, we used the library dedicated to distributed learning methods Spark ML [3].

This paper is detailed as follows: section 2 presents the related work on which we are concerned to carry out this study. Then, in section 3 we talk about the Apache Spark ML solution dedicated to the use of Machine Learning algorithms and methods on Big Data [4,5]. Section 4 presents our proposed Machine Learning model to predict the number of cases infected with the coronavirus (COVID-19). Finally, section 5 presents the results obtained after the application of our model on the case of Morocco and China as well as a summary discussion of the work carried out.

2. Related work

The use of machine learning techniques better known by its Anglicism "Machine Learning" is becoming more and more popular. It applies to a wide variety of fields, from image processing to processing human language, including computer security [35] and the video game industry [2]. One of the main purposes of applying machine learning techniques is prediction. Any machine learning process can be subdivided into the following steps: data preprocessing, model design and training, and model evaluation. The preprocessing phase is still known as “Pre-processing” very often consists of choosing the most informative variables from the set of incoming vectors and normalizing them if necessary. This technique is also called "feature selection / feature engineering". The purpose of the selection of informative variables is to

E-mail: allae.erraissi-etu@etu.univh2c.ma, mouad.banane-etu@etu.univh2c.ma

eliminate nonessential variables which can be considered as noise by the model and thus reduce its accuracy of prediction. Several research studies have been carried out in recent decades to propose either algorithms or criteria for the selection of informative variables.

In [6], authors are offered a technology for extracting informative variables based on the ontology of the field studied. The first phase of this technology consists of a preliminary extraction by the domain expert, of the variables which he deems useful without any other restriction than the variables chosen must be objects or attributes of the ontology. The second phase, still described in [6], consists of aggregating and filtering the preselected variables during the previous phase. The mathematical translation of this procedure is as follows: either $\Omega = \{ {}_1 ,\ldots\ldots ,\omega_n\}$ - the class labels, $X = \{ X_1 , \ldots , X_n\}$- the set of variable identifiers. x^i_s - a value taken by X_i. $\aleph_i$ - the definition domain of X_i(x^i_s □ $\aleph_i$). Let $\sum$ be the set of learning vectors.

For a x^i_s value of X_i: x^i_s □ $\aleph_i$ and in the class ω_k an aggregate $\aleph_i(\omega_k)$ □ $\aleph_i$ is defined as follows:

$$x^i_s \in \aleph_i(\omega_k) \text{ if and only if for } \forall \omega_v \in \Omega,\ v \neq k : p(\omega_k / x^i_s) > p(\omega_v / x^i_s) + \Delta, \quad (1)$$

Where Δ a positive real number defining the dominance threshold.

At the end of phase 2, the author introduces the unitary predicates Bi (ωk) which take the value 'true' (1) if and only if x^i_s □ $\aleph_i(\omega_k)$. The final results of the second phase will, therefore, be the aggregates $\aleph_i(\omega_k)$, the unitary predicates $B_i(\omega_k)$ and i □ $I(\omega_k)$: $I(\omega_k)$ - a subset of the indexes of the variables X_i that passed the test of inequality (1).

During Phase 3 a probabilistic approach was introduced to determine cause-and-effect dependencies in the conjunction of the predicates $B_i(\omega_k)$ and i □ $I(\omega_k)$, $\Omega = \{ \omega_1 ,\ldots\ldots ,\omega_n \}$ and ω_j □ Ω.

Let { $\aleph_i(\omega_j)$ }| i □ $I(\omega_k)$, j=1, ..., m - the set of aggregates where $p(\aleph_i(\omega_j))=|\aleph_i(\omega_j)|/|\aleph_i|$, where | | represents the standard of the corresponding assembly. It is obvious that: $p(B_i(\omega_j))= p(\aleph_i(\omega_j))$.

In the following three filters will be introduced to select the predicates:

Filter 1 :

$$I(B_i(\omega_j),\omega_k) = |p(B_i(\omega_j)\omega_k) - p(B_i(\omega_j))p(\omega_k)|/[p(B_i(\omega_j))p(\omega_k)] \geq \delta_{min} > 0\text{-a} \quad (2)$$

Filter 2:

$$R(\widehat{B}_\iota(\omega_j),\omega_k) = |p(\omega_k/ \widehat{B}_\iota(\omega_j)) - p(\omega_k/ \widehat{B}_\iota(\omega_j))|\{p(\widehat{B}_\iota(\omega_j)))[1 - p(\widehat{B}_\iota(\omega_j))]\} = |p(\widehat{B}_\iota(\omega_j)\ \omega_k) - p((\widehat{B}_\iota(\omega_j)))\ p(\omega_k)\ |/\{p(\widehat{B}_\iota(\omega_j))\ [1- p(\widehat{B}_\iota(\omega_j))] \geq \delta_{min},\ \delta_{min} > 0\text{-a} \quad (3)$$

Filter 3:

$$p(\widehat{\omega_k}/\widehat{B}_\iota(\omega_j)) = p(\widehat{B}_\iota(\omega_j)\ \widehat{\omega_k})/\ p(\widehat{B}_\iota(\omega_j)) \geq \gamma_{min} \quad (4)$$

The authors of [7] in Chapter 6 entitled "Dimensionality reduction" discuss the PCA (Principal Component Analysis [8]) algorithm based on the analysis of linear dependence. The idea is to replace the redundant variables with a new one that sums up the information contained in the original vector space. One of the first concepts to appear is SVD (Singular Value Decomposition) [9]. Let X be a matrix of dimension n * d; n- being the number of lines and d - the dimension of the initial vector space to be reduced. Let x be a column vector (transposed from one of the rows of X) and let v - be one of the new vector entities (principal components) that we are looking for. According to the SVD theorem, matrix X can be decomposed sum follows $X = U\Sigma V^T$, U and V being orthogonal matrices such that $U^TU = I$ *et* $V^TV = I$. $\sum$ a diagonal matrix containing the singular values of X.

In the following, the author develops the main stages of PCA [8]. The first step is to refocus the data in the initial vector space. The second step is the linear projection of the initial data vector x into new vectors v. The third step is to maximize the variance of the coordinates. The equation of the coordinate vector after projection is z=Xv.

The objective function for the main components is maxw W^TW, where $W^TW = 1$. It appears that the optimal form of W is the eigenvector of X^TX.

In [10] the authors based on Bayes' theorem:

$$P(A_j|B) = \frac{P(A_j)P(B|A_j)}{\sum_{j=1}^{n} P(A_j)P(B|A_j)} \quad (5)$$

and on Information theory [11], in particular, the notion of self-information $I(x_i)=-log_2P(x_i)$ proposes an algorithm supposed to improve the forecasting accuracy of the NBC (Naive Bayesian Classifier) model [12].

The algorithm is divided into two stages consists of its first phase of calculating the weight of each variable $W_{Fi}=-log_2P(C|F_i)$. The second phase consists of selecting the variables whose weight W_{Fi} would have exceeded a certain threshold δ determined by the user.

A new method called LFE (Learning Feature Engineering) is proposed in [13]. At the heart of the LFE [14], we find a set of multilayer classifying perceptrons. Each perceptron corresponding to a transformation. LFE takes as input a dataset and recommends a set of paradigms allowing reconstructing a subset of the informative variables of the initial dataset. Each paradigm is made up of a transformation and the ordered list of variables for which the transformation was most efficient.

In the following, we will propose a model based on the machine learning methods already mentioned to predict the number of COVID-19 cases in Morocco per day. Through the use of the library dedicated to distributed learning methods Spark ML [3].

3. SPARK ML

Spark ML is an essential software brick of the Apache Spark platform [3]. The Spark ML API is dedicated to the implementation of learning methods. Spark ML offers services covering data preparation, enrichment, development of learning methods, and deployment.

The Spark ML API offers several transformers and algorithms for developing and fine-tuning methods of:

- Classification
- Regression
- Clustering
- Recommendation

A. Concepts for implementing Spark ML algorithms

Here we cite the key concepts for implementing a Spark ML algorithm [15]:

- DataFrame: is a distributed, column-oriented data structure suitable for statistical data or learning methods.
- Estimators: these are algorithms aimed at adjusting a function between the explanatory variables and the target variable. The algorithms are based on the data of a DataFrame.
- Evaluators: measures to assess the performance of a learning method.
- Transformers: these are algorithms used to transform one DataFrame into another.
- Parameters: Transformers and Estimators share the same API to specify parameters.
- Pipeline: a pipeline consists of several Transformers and Estimators for the implementation of a ML workflow.
- Cross-Validation: a technique to develop a learning method.

B. Pipeline & PipelineStage

A learning method involves processing and learning phases. A Pipeline, in Spark MLIB [16], is a sequence of elements of type pipelineStage. Each PipelineStage consists of a Transformer or an Estimator. Therefore, the order of steps (PiplineStage) is important.

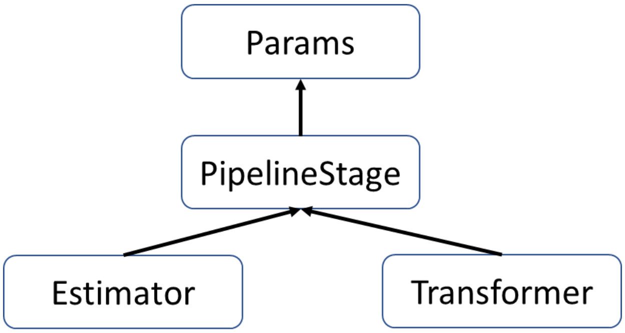


Figure 1. Hierarchy: Estimator and Transformer [16].

C. Transformation Stage

A transformer transforms data. In Spark ML, for a transformation step, the transform() method is called on a DataFrame and returns a DataFrame. The transformer is an abstract class that extends the PipelineStage abstract class.

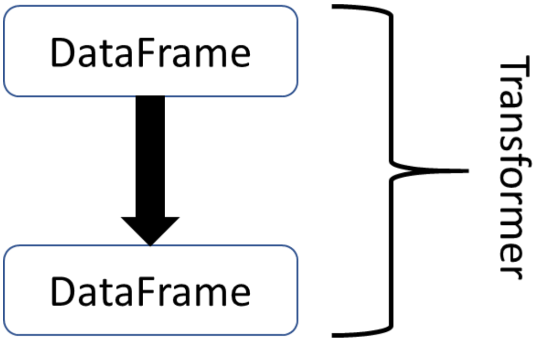


Figure 2. The role of Transformer [16].

Some examples of Transformers:

- Tokenization: segment text into a list of words.
- Binarization: transform numerical values into binary values.
- Normalizer: normalize vectors using p-norm.
- N-gram: calculate the next character from an article sequence θ MinMaxScaler: transform a quantitative variable to a quantitative variable with values in a given interval.
- Etc.

D. Estimation Stage

An estimator allows us to evaluate or approach an unknown parameter. A statistical estimator aims to evaluate an unknown parameter via a sample of data. In Spark ML, for an estimation step, the fit() method of an Estimator is called on a DataFrame to produce a Transformer.

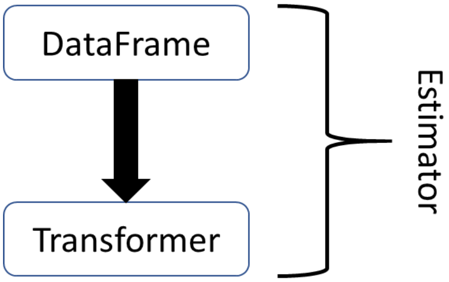


Figure 3. The role of Estimator [16].

Some examples of estimators:

- StringIndexer: code categories.
- IndexToString: decode indexes.
- VectorIndexer: identify categorical variables according to a frequency threshold.
- QuantileDiscretizer: transform a quantitative variable into a qualitative variable using the quantile.
- StandardScaler: normalize a vector using the variance.
- Etc.

E. Evaluate a learning method

To evaluate the performance of a learning method, we use dedicated measures. For a given individual, an evaluation measure uses the adjusted value and the reference value. We have measures for regression methods and classification methods.

- **Evaluate a Learning method: Classification**

It aims to measure the quality of a classification method. It indicates whether the method succeeds in correctly classifying individuals.

TABLE I. CONFUSION MATRIX [17].

		Predicted	
		Positive	Negative
Real	Positive	TP	FN
	Negative	FP	TN

The evaluation measures are:

- Sensitivity: the rate of correctly identified positive cases.
- Specificity: the rate of correctly identified negative cases.
- Accuracy: the rate of correctly identified positive or negative cases.
- F-measure: the rate that combines sensitivity and specificity.

- **Evaluate a learning method: regression**

It aims to measure the quality of a regression method by calculating a global error. An error close to zero indicates the relevance of the model. The main measures are [15]:

- MSE: Mean Squared Error.
- RMSE: this is the square root of MSE (root Mean Squared Error).
- R: Coefficient of determination.
- MAE: mean Absolute Error.
- MAPE: Mean Absolute Percentage Error.

F. API for the evaluation of a learning algorithm

The Spark ML API provides three implementations for the Evaluator abstract class. The classes are: RegressionEvaluator for regression methods, BinaryClassificationEvaluator for binary classification methods, and MulticlassClassificationEvaluator for multi-class classification methods.

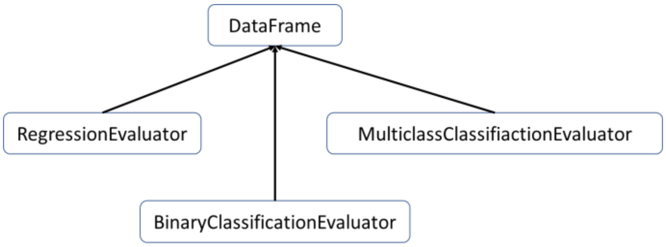

DataFrame
RegressionEvaluator
MulticlassClassifiactionEvaluator
BinaryClassificationEvaluator

Figure 4. Implementations for the Evaluator abstract class [18].

4. PROPOSED MODEL

In order to better propose a Machine Learning model [33,34] to predict the number of cases of the Corona COVID-19 [19] virus in Morocco over the coming weeks.

As it is established that, during epidemics, the daily number of infected follows a bell-shaped diffusion curve which gives, in cumulative, an S curve, the choice of a logistic function was noticeable.

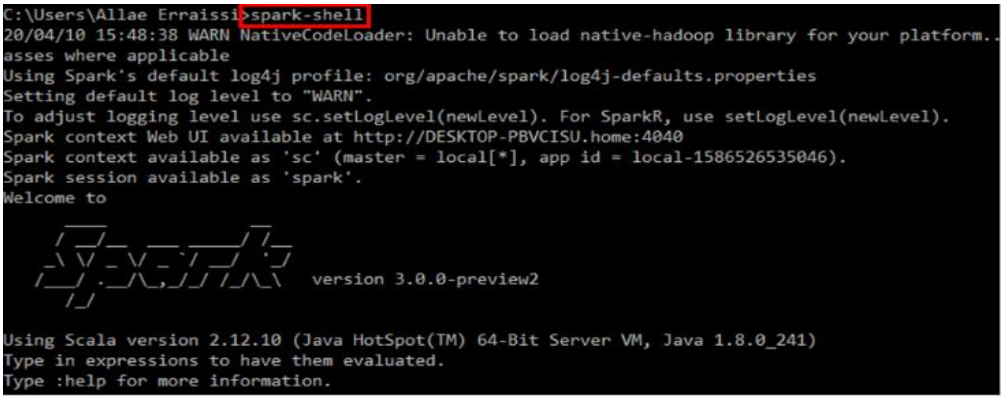
The data we had captured the results of tests by health authorities in different countries around the world. Nevertheless, we knew without question that these data did not reflect reality. Some countries have chosen not to be over-tested either by choice of public health policy or because they simply did not have the means. Moreover, even for the countries that tested more significantly, it can be reasonably assumed that, in the first periods, the number of tests concerning the number of real cases was lower than in the following periods. The awareness of the dangers increases with time. In order to verify the relevance of our proposed ML model, we conducted a series of tests on data from the site of the Moroccan

Ministry of Health (www.covidmaroc.ma) [20]. So the dataset first gathered was of the type (CSV file [21]).

```
Date;02/03/2020;03/03/2020;04/03/2020;05/03/2020;06/03/202
0;07/03/2020;08/03/2020;09/03/2020;10/03/2020;11/03/2020;1
2/03/2020;13/03/2020;14/03/2020;15/03/2020;16/03/2020;17/0
3/2020;18/03/2020;19/03/2020;20/03/2020;21/03/2020;22/03/2
020;23/03/2020;24/03/2020;25/03/2020;26/03/2020;27/03/2020
;28/03/2020;29/03/2020;30/03/2020;31/03/2020;01/04/2020;02/
04/2020;03/04/2020;04/04/2020;05/04/2020;06/04/2020;07/04/
2020;08/04/2020;09/04/2020;
Number of cases;1;0;0;1;0;0;0;1;0;3;0;2;10;10;9;
7;10;9;16;17;19;28;27;55;50;70;57;77;77;61;37;54;83;128;102;9
9; 64;91; 99;
```

This CSV type file must be transformed into a DataFrame, because we are going to use Spark ML, which is based on the DataFrame API. To do this, we used version 3.0.0 of Spark. And as for programming language, we used version 2.12.10 of the Scala programming language [22]. You need to make sure that Java is installed on your machine. All of these tools were installed on a machine with a 2.70 GHz Intel (R) Core (TM) i7 processor. With 1 TB storage space and 16 GB RAM. The Java version "jre1.8.0_241" is used for the development of our Machine Learning model.

So, we start with the command: spark-shell. The following figure shows the execution result and the configuration used:

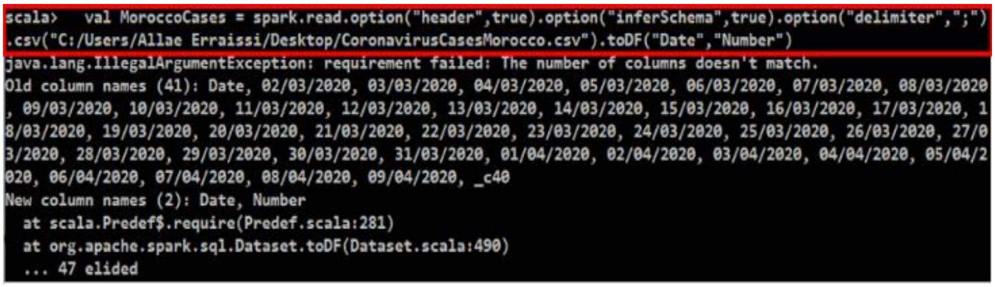


```
C:\Users\Allae Erraissi>spark-shell
20/04/10 15:48:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform..
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://DESKTOP-PBVCISU.home:4040
Spark context available as 'sc' (master = local[*], app id = local-1586526535046).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.0-preview2
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_241)
Type in expressions to have them evaluated.
Type :help for more information.
```

Figure 5. Spark execution and system configuration.

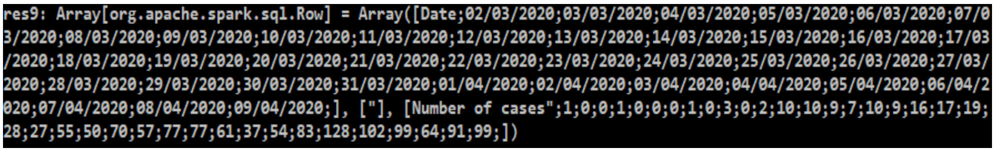
Then we transform the input file of type CSV to a DataFrame with the following command:



```
scala>   val MoroccoCases = spark.read.option("header",true).option("inferSchema",true).option("delimiter",";")
.csv("C:/Users/Allae Erraissi/Desktop/CoronavirusCasesMorocco.csv").toDF("Date","Number")
java.lang.IllegalArgumentException: requirement failed: The number of columns doesn't match.
Old column names (41): Date, 02/03/2020, 03/03/2020, 04/03/2020, 05/03/2020, 06/03/2020, 07/03/2020, 08/03/2020
, 09/03/2020, 10/03/2020, 11/03/2020, 12/03/2020, 13/03/2020, 14/03/2020, 15/03/2020, 16/03/2020, 17/03/2020, 1
8/03/2020, 19/03/2020, 20/03/2020, 21/03/2020, 22/03/2020, 23/03/2020, 24/03/2020, 25/03/2020, 26/03/2020, 27/0
3/2020, 28/03/2020, 29/03/2020, 30/03/2020, 31/03/2020, 01/04/2020, 02/04/2020, 03/04/2020, 04/04/2020, 05/04/2
020, 06/04/2020, 07/04/2020, 08/04/2020, 09/04/2020, _c40
New column names (2): Date, Number
  at scala.Predef$.require(Predef.scala:281)
  at org.apache.spark.sql.Dataset.toDF(Dataset.scala:490)
  ... 47 elided
```

Figure 6. Transformation of the CSV file to a DataFrame.

Here is the result of the creation of the DataFrame from a data source which is of type CSV in this work:



```
res9: Array[org.apache.spark.sql.Row] = Array([Date;02/03/2020;03/03/2020;04/03/2020;05/03/2020;06/03/2020;07/0
3/2020;08/03/2020;09/03/2020;10/03/2020;11/03/2020;12/03/2020;13/03/2020;14/03/2020;15/03/2020;16/03/2020;17/03
/2020;18/03/2020;19/03/2020;20/03/2020;21/03/2020;22/03/2020;23/03/2020;24/03/2020;25/03/2020;26/03/2020;27/03/
2020;28/03/2020;29/03/2020;30/03/2020;31/03/2020;01/04/2020;02/04/2020;03/04/2020;04/04/2020;05/04/2020;06/04/2
020;07/04/2020;08/04/2020;09/04/2020;], ["], [Number of cases";1;0;0;1;0;0;0;1;0;3;0;2;10;10;9;7;10;9;16;17;19;
28;27;55;50;70;57;77;77;61;37;54;83;128;102;99;64;91;99;])
```

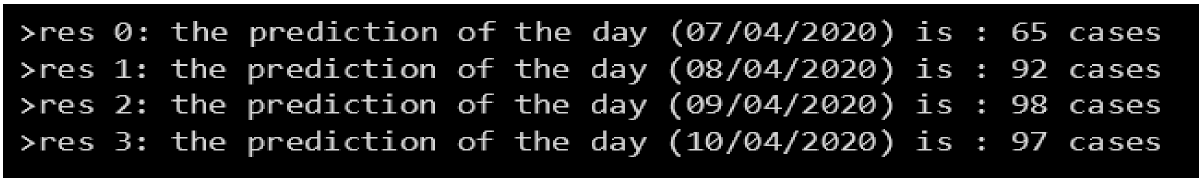
Figure 7. Content of the DataFrame.

Now that the DataFrame is well created, it will be used by the Spark Machine Learning library to predict the number of cases of the Corona Covid-19 virus in Morocco per day. This part shows an extract from the code of the model that we have developed for this study:

```
import org.apache.spark.ml.{Pipeline, PipelineModel}
import org.apache.spark.ml.classification.
{DecisionTreeClassifier,DecisionTreeClassificationMo
del}
import org.apache.spark.ml.feature.{StringIndexer,
VectorAssembler}
// load CovidMorocco data and rename columns
val MoroccoCases = spark.read.option("header",
true).option("inferSchema","true")
.option("delimiter";" ").csv("C:/Users/Allae
Erraissi/Desktop/CoronaVirusCasesMorocco.txt").toD
F("Date", "Number)
// Create assemble to group the columns
val assembler = new
VectorAssembler().setInputCols(Array("MorningDate"
,"EveningDate")) .setOutputCol("ResultOfDay")
```

```
// transform the Species variable into an indexer
val indexer = new
StringIndexer().setInputCol("Number").setOutputCol("N
umberIndex")
// Create the learning model DT
val dt = new
DecisionTreeClassifier().setLabelCol("NumberIndex").
.setPredictionCol("NumberPredictCol").setFeaturesCol("
features")
// create a pipeline: assembler, indexer, dt
val pipeline = new Pipeline().setStages(Array(assembler,
indexer, dt))
// adjust the model
val model = pipeline.fit(MoroccoCases)
// evaluate the model on learning data
val predictions = model.transform(MoroccoCases)
// adjust results
predictions.show
```

The execution of the COVID 19 model on the sample of data from Morocco gives us the results of this form:



```
>res 0: the prediction of the day (07/04/2020) is : 65 cases
>res 1: the prediction of the day (08/04/2020) is : 92 cases
>res 2: the prediction of the day (09/04/2020) is : 98 cases
>res 3: the prediction of the day (10/04/2020) is : 97 cases
```

Figure 8. The result of the execution of our model.

Then we present a part of code which presents the techniques relating to the evaluation of our learning method:

```
// Import of class for evaluation
import
org.apache.spark.ml.evaluation.MulticlassClassificationE
valuator
// Create the assessment instance
// Fix the label column
// Fix the prediction column
// Set the evaluation measure
val evalution = new
MulticlassClassificationEvaluator().setLabelCol("Numbe
rIndex")
.setPredictionCol("NumberPredictCol").setMetricName("
accuracy")
// calculate the value of Accuracy
val ACCURACY= evalution.evaluate(predictions)
// display the value of ACCURACY
print(s" Accuracy= ${ACCURACY}")
```

To evaluate our proposed Machine Learning model, we followed all of these steps:

- Load data.
- Prepare the data.
- Adjust the ML model.
- Initiate the evaluator and position the params.
- Evaluate the method via the Accuracy measure.

5. Discussion and Evaluation

In order to verify the relevance of the proposed methods, we had conducted a series of tests on data from the site of the Moroccan Ministry of Health (www.covidmaroc.ma) [20]. The classification algorithms used are: 'Decision Tree' (DTC) [23], 'Gaussian Naive Bayes' (NBC) [24], 'Support Vector Machine' (SVM) [25], 'Logistic Regression' (LR) [26], 'Random Forest'(RFC) [27] and finally 'Voting Classifier' (VC) [28]. The characteristics of the input data are shown in figure 9. These data show the number of COVID-19 cases in Morocco per day.

To make predictions, whether we have the partial differential equation (PDE) model [29] which determines how the different parameters involved in the phenomenon interact with each other or ordinary differential equations (ODE) [30] when we only have two quantities, one in function on the other which act on the phenomenon. In this case, the data collected is injected into the equations and it is the resolution of the equations which gives the forecasts. The examples illustrating this are numerous and diverse, we can cite the example of the weather. What you also need to know is that the models are approximate, and approximation methods make their resolution. If the forecast in the case of the weather is more and more precise. This is because the models keep getting better and the resolution methods are more and more efficient and fast, so they give fewer errors.

Furthermore, to clarify is that the phenomena for which we have PDE, or ODE equation models are phenomena that have a certain regularity, even if the phenomenon is complex, the regularity can be concealed. If we don't have the equation model, can we make predictions? The answer is yes if the phenomenon is random and if we can find the law of probability that governs it, we can use it to make forecasts. Now, if we don't have an equation model or a probability law can we forecast anyway? In this case, we use extrapolation methods to extrapolate the history. The least-squares method consists in looking for a polynomial that best approaches the data represented in the form of a point cloud. When we say who "approaches the best" we would need a means of measurement, a criterion to select the best approximation, this is what is called standard. Following the course of evolution, this polynomial is considered as the mathematical model that governs the evolution of the phenomenon, we speak of the polynomial model. The polynomial model does not work for all phenomena.

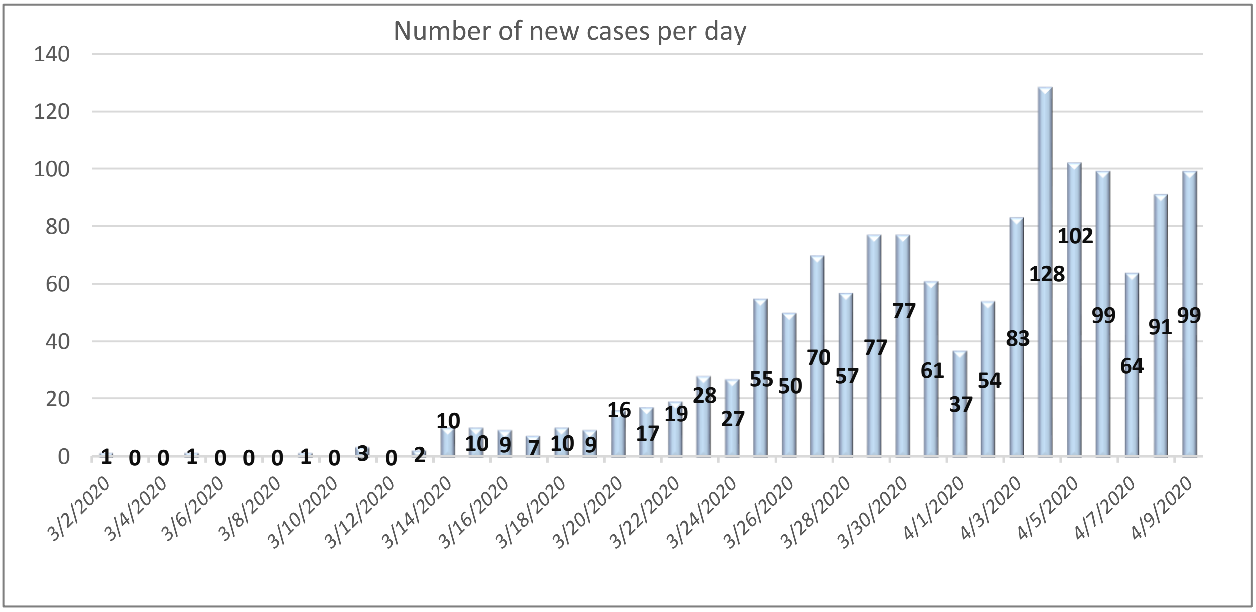


Figure 9. Number of COVID-19 cases per day in Morocco.

Case of Covid-19:

We are faced with a completely random phenomenon. The number of people infected depends on the trajectory of each infected individual and the people they met on their way and also depended on the people who have touched anywhere the virus is found. Assuming that the model follows a Gaussian law is not meaningless, without doubt, the evolution continues its growth, relatively fast, it will not grow indefinitely, it will go through a peak then it will decrease to 0 Maybe not with the same speed. The whole problem is to approach this Gaussian. We are going to show on a theoretical example that polynomials cannot "efficiently" approach a Gaussian using a reduced number of data. Therefore, the polynomials cannot be used as predictors in this case.

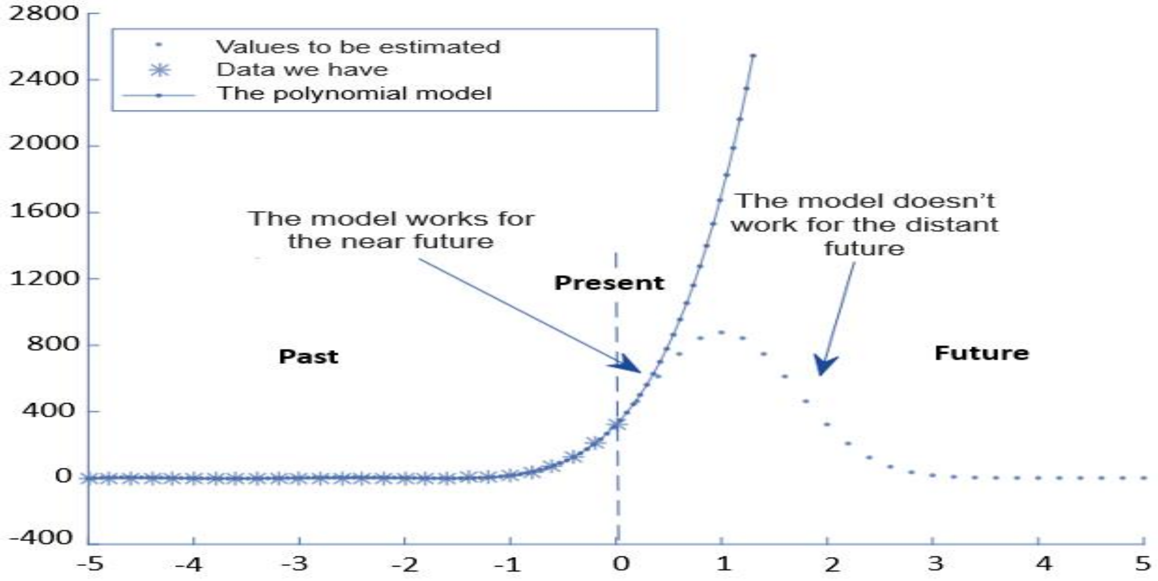


Figure 10. Result of the polynomial approach.

We can see in the graph that polynomials cannot predict the long term. To remedy this, the use of our Machine Learning model with Spark ML to give satisfactory results. We have given as input to our model several parameters related to the situation of our paid in this confinement period. We have taken into consideration the percentage of illiteracy in Morocco and its relationship with the application of the instructions of the various ministries of the country which oblige people to isolate themselves and stay at home. Also, consideration has been given to the approaching of the holy month of Ramadan for Muslims. It is, for this reason, our Machine Learning model predicted two possible scenarios for the COVID-19 [31] in Morocco.

As already mentioned, we have applied this model to data extracted from the site: www.marocovid.com. The result is shown in the following figure:

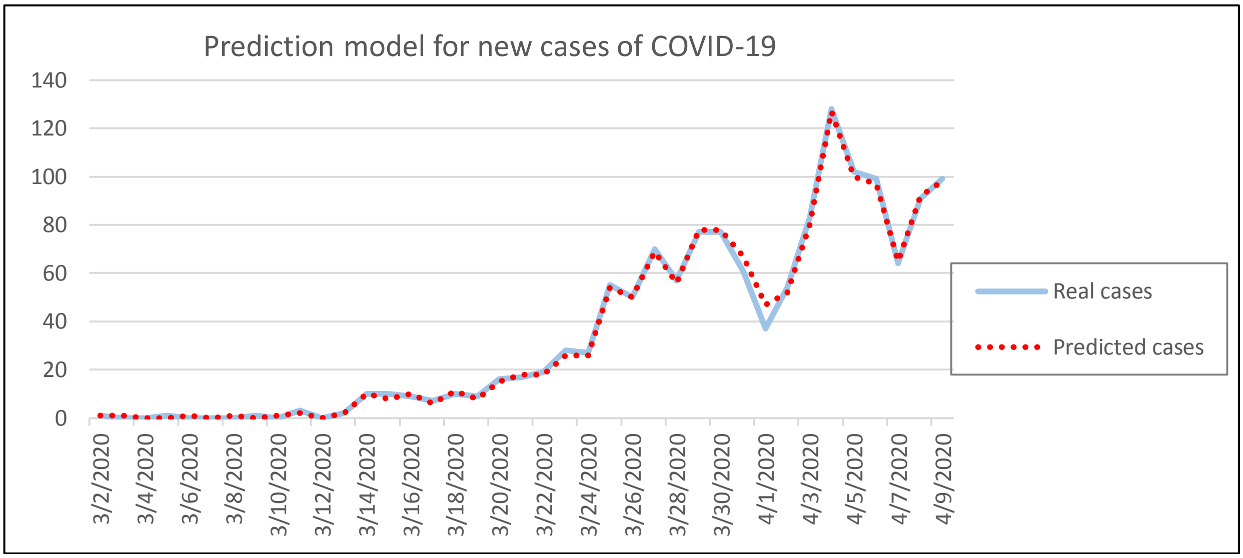


Figure 11. Prediction model for new cases of COVID-19 in Morocco.

We applied our model to detect the number of cases of the Coronavirus in Morocco on the DataFrame. We have defined two possible scenarios for our study. The first scenario consists in that the Moroccan population has respected the instructions of the ministries responsible for fighting against the COVID-19 [31].

With the application of our Cvid-19 model, our estimate is as follows for the first scenario:

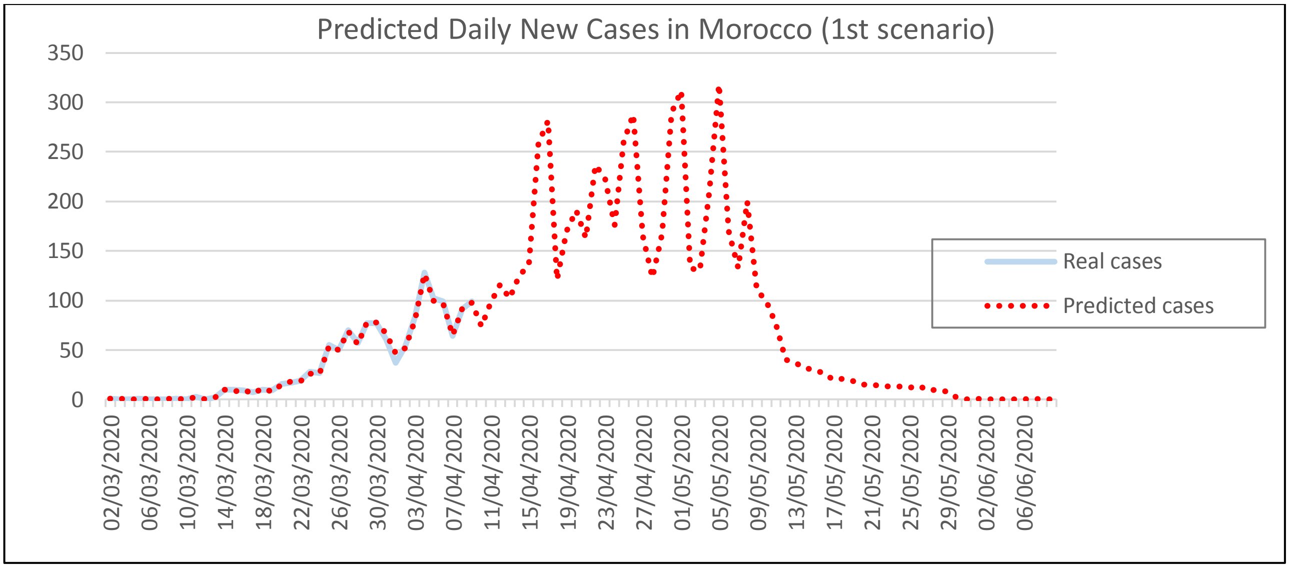


Figure 1. Predicted daily new cases in Morocco (1st scenario).

We can see that the growth rate has decreased significantly which has considerably influenced the peak and also the date of attenuation. The second scenario was defined based on the place of the holy month of Ramadan among Muslims in general, and especially for Moroccans. So, during this month, people will not respect the instructions of the Moroccan ministries.

To take this information into account, we have changed the weighting percentages for the country's illiteracy, as well as the percentage of non-compliance with the instructions of the Moroccan authorities.

With the application of our COVID-19 model, our estimate is as follows for the second scenario:

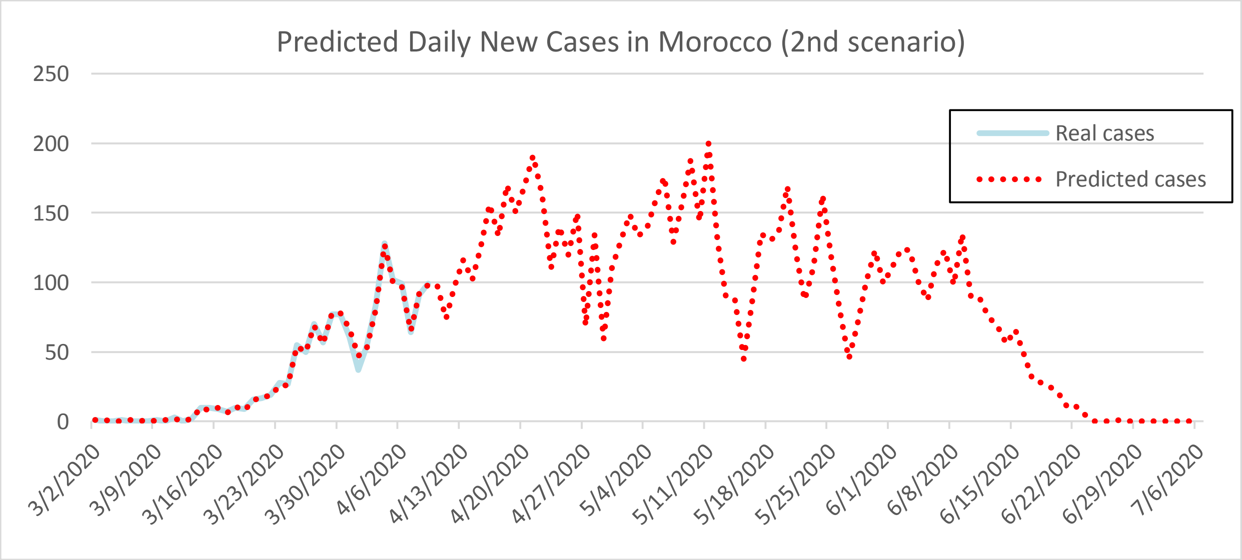


Figure 2. Predicted daily new cases in Morocco (2nd scenario).

- **Verification of our model on the case of China:**

China has recorded enough data that can be used for verification. The data on which we do this verification is extracted from the site: https://www.worldometers.info/coronavirus/country/china/ [32].

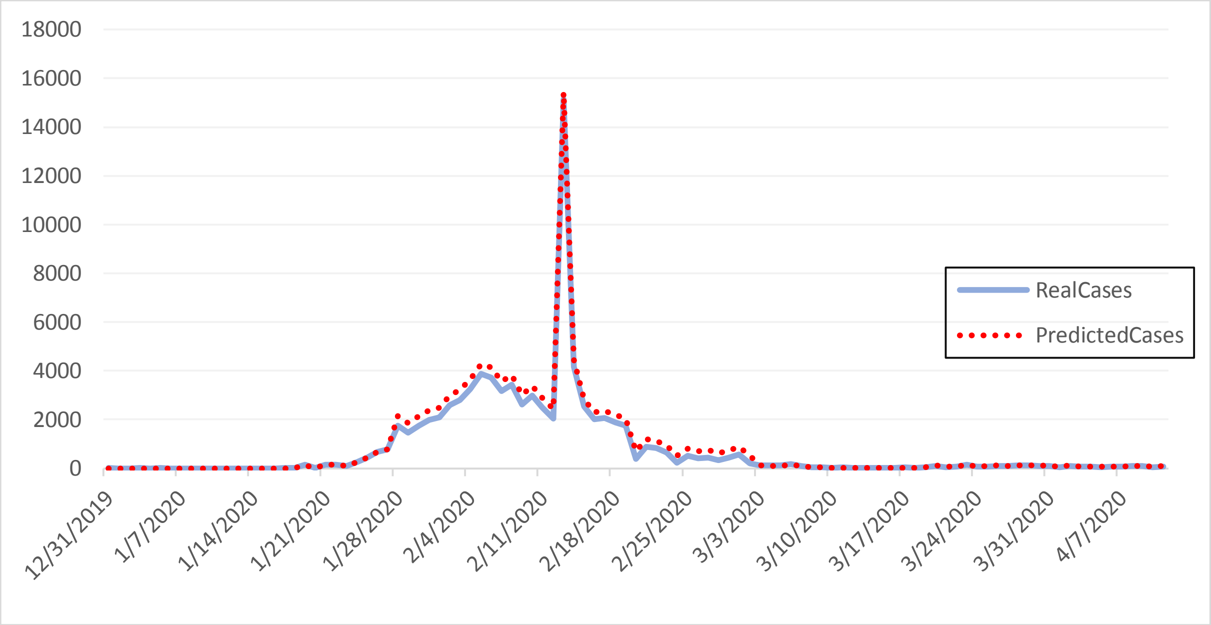


Figure 3. Application of our model on the case of China.

The accuracy of forecasts depends on the "method" approach first, on the number of data and the distance from the future to be predicted.

6. CONCLUSION

This work presents a dedicated machine learning model to predict the number of cases infected with the Corona Virus; the case of Morocco was chosen to validate this study. Completely realized in Spark ML with the 'Scala' language and tested for a certain number of algorithms generated on datasets from dedicated sources to group COVID-19 data to the world, the results show the possibility of achieving better scores prediction after using the proposed methods. We are sending this work to the world to help them in our way to fight this 2019 Coronavirus pandemic.

REFERENCES

[1] Kulkarni S, Seneviratne N, Baig MS, Khan AH. Artificial Intelligence in Medicine: Where Are We Now?. Academic radiology. 2019 Oct 19.

[2] Waring, J., Lindvall, C., & Umeton, R. (2020, April 1). Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. Artificial Intelligence in Medicine. Elsevier B.V. https://doi.org/10.1016/j.artmed.2020.101822

[3] Karau, Holden, and Rachel Warren. High performance Spark: best practices for scaling and optimizing Apache Spark. " O'Reilly Media, Inc.", 2017.

[4] Erraissi, Allae, and Abdessamad Belangour. « Meta-Modeling of Big Data Management Layer ». International Journal of Emerging Trends in Engineering Research 7, no 7, 36-43, 2019. https://doi.org/10.30534/ijeter/2019/01772019.

[5] Erraissi, Allae, and Abdessamad Belangour. « Hadoop Storage Big Data Layer: Meta-Modeling of Key Concepts and Features ». International Journal of Advanced Trends in Computer Science and Engineering 8, n□ 3, 646-53, 2019. https://doi.org/10.30534/ijatcse/2019/49832019

[6] Gorodetsky V., Samoylov V., 'Feature Extraction for Machine Learning: Logic-Probabilistic Approach', JMLR: Workshop and Conference Proceedings 10: 55-65 The Fourth Workshop on Feature Selection in Data Mining.

[7] Zheng A., Casari A., 'Feature Engineering for Machine Learning', O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[8] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

[9] Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. In Linear Algebra (pp. 134-151). Springer, Berlin, Heidelberg.

[10] Mani K., Kalpana P., 'An Efficient Feature Selection based on Bayes Theorem, Self-Information and Sequential Forward Selection', I.J. Information Engineering and Electronic Business, 2016, 6, 46-54.

[11] Blahut, R. E., & Blahut, R. E. (1987). Principles and practice of information theory (Vol. 1). Reading, MA: Addison-Wesley.

[12] Kononenko, I. (1991, March). Semi-naive Bayesian classifier. In European Working Session on Learning (pp. 206-219). Springer, Berlin, Heidelberg.

[13] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B. Khalil, Deepak Turaga, 'Learning Feature Engineering for

Classification', Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).

[14] Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. S. (2017, August). Learning Feature Engineering for Classification. In IJCAI (pp. 2529-2535).

[15] Spark, Apache. "Apache spark." Retrieved January 17 (2018): 2018.

[16] Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark." The Journal of Machine Learning Research 17.1 (2016): 1235-1241.

[17] Townsend, J. T. (1971). Theoretical analysis of an alphabetic confusion matrix. Perception & Psychophysics, 9(1), 40-50.

[18] Bifet, Albert, et al. "Streamdm: Advanced data mining in spark streaming." 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, 2015.

[19] Lai, C. C., Shih, T. P., Ko, W. C., Tang, H. J., & Hsueh, P. R. (2020). Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and corona virus disease-2019 (COVID-19): the epidemic and the challenges. International journal of antimicrobial agents, 105924.

[20] Morocco Covid-19 cases. http://www.covidmaroc.ma/pages/Accueil.aspx

[21] Mahmud, S. H., Hossin, M. A., Jahan, H., Noori, S. R. H., & Bhuiyan, T. (2018, May). CSV-ANNOTATE: Generate annotated tables from CSV file. In 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD) (pp. 71-75). IEEE.

[22] Odersky, Martin, Lex Spoon, and Bill Venners. Programming in scala. Artima Inc, 2008.

[23] Safavian, S. Rasoul, and David Landgrebe. "A survey of decision tree classifier methodology." IEEE transactions on systems, man, and cybernetics 21.3 (1991): 660-674.

[24] Ontivero-Ortega, Marlis, et al. "Fast Gaussian Naïve Bayes for searchlight classification analysis." Neuroimage 163 (2017): 471-479.

[25] Ben-Hur, Asa, and Jason Weston. "A user's guide to support vector machines." Data mining techniques for the life sciences. Humana Press, 2010. 223-239.

[26] Kleinbaum, David G., et al. Logistic regression. New York: Springer-Verlag, 2002.

[27] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

[28] Ruta, Dymitr, and Bogdan Gabrys. "Classifier selection for majority voting." Information fusion 6.1 (2005): 63-81.

[29] Strikwerda, J. C. (2004). Finite difference schemes and partial differential equations (Vol. 88). Siam.

[30] Hale, J. K. (1971). Functional differential equations. In Analytic theory of differential equations (pp. 9-22). Springer, Berlin, Heidelberg.

[31] World Health Organization. "Coronavirus disease 2019 (COVID-19): situation report, 72." (2020).

[32] Coronavirus Cases in China. https://www.worldometers.info/coronavirus/country/china/.

[33] Erraissi, A., Mouad, B., & Belangour, A. (2019). A big data visualization layer meta-model proposition. Paper presented at the 2019 8th International Conference on Modeling Simulation and Applied Optimization, ICMSAO 2019, doi:10.1109/ICMSAO.2019.8880276

[34] Kalna, F., Erraissi, A., Banane, M., & Belangour, A. (2019). A scalable business intelligence decision-making system in the era of big data. International Journal of Innovative Technology and Exploring Engineering, 8(12), 2034-2042. doi:10.35940/ijitee.L3251.1081219

[35] Erraissi, A., & Belangour, A. (2019). A big data security layer meta-model proposition. Advances in Science, Technology and Engineering Systems, 4(5), 409-418. doi:10.25046/aj040553

Allae Erraissi is an Associate Professor at Chouaib Doukkali University, El Jadida, Morocco. He obtained his PhD on Computer Science from Hassan II University, Faculty of Sciences Ben M'Sik, Casablanca, Morocco. His main interests engaged in research on various aspects of Information technologies namely Model-Driven Engineering approaches and their applications on new emerging technologies such as Big Data, Cloud Computing, Business Intelligence, Internet of Things, etc.

Mouad Banane is an Associate Professor at Hassan II University. Laboratory of Artificial Intelligence & Complex Systems Engineering (AICSE). He obtained his Ph.D at Faculty of Sciences Ben M'Sik, Hassan II University of Casablanca, Morocco. His research fields: Model-Driven Engineering, Semantic Web, Artificial Intelligence, and Big Data.