Jurnal CoreIT, Vol.9, No.1, June 2023
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

Efficiency of the Combination of Machine Learning Models in the Evaluation of Weather Parameters

Yannick Mubakilayi[1], Simon Ntumba[2], Pierre Kafunda[3], Salem Cimanga[4], Theodore Kabangu[5]
[1,2,3,4,5]Department of Computer Science, University of Kinshasa, RD. Congo
[1]myannick@aims.ac.rw, [2]profntumba@gmail.com, [3]kafundakatalay@gmail.com,[4]cimangasalemtheophileson@gmail.com,
[5]theo.kabangu@um.ac.cd

Abstract. In this paper, we exploit the potential presented by the combination of ensemble learning models as one of the key points of the soft aspect, i.e. Tools for observing, monitoring, sampling and studying meteorological parameters in order to machine learning models composed of measurements of different meteorological parameters (temperature, rainfall, humidity rate, wind speed, etc) and then combine the results of the different models via an ensemblistic method to improve the efficiency of these automatic learning models in evaluating meteorological parameters. In this research, the authors have shown that Voting Regressor as an ensemble learning method can improve the efficiency of combining automatic learning models in evaluating meteorological parameters because we used heterogeneous models (MLP Regressor, Decision Tree Regressor and K Neighbors Regressor), and we noted that our models presented a score of at least 50 percent. as can be seen in figure 8 and table 1, with a Model score: 0.79, R-squared 0.79, MAE 2.8, RMSE 3.5. This article uses raw data in the form of a .csv file as the dataset. The authors of this study collected this dataset at Bipemba Airport (Mbujimayi/RD Congo), comprising 84 records averaged over 7 years, from 2015 to 2021.

Keywords*:* Machine learning, Voting Regressor, MLP Regressor, Decision Tree Regressor, K Neighbor sRegressor

Received February 2023 / **Revised** May 2023 / **Accepted** June 2023

INTRODUCTION

At present, it is very important for some parts of our planet to assess and closely monitor meteorological parameters to detect possible changes in order to better manage the risks associated with climate change. Climate change and climatic disturbances currently observed in certain regions of the earth have their origin in essentially the emission of greenhouse gases released by the various sources of emission which are either motor vehicles, processing plants, industries chemicals, air transport, maritime transport etc… all these different vectors emit daily disproportionate quantities of gases and particles which thus represent the main source of atmospheric pollution and which negatively influence the various weather parameters.

To this end, it is more than necessary to evaluate and predict the weather parameters to have control of the course of the climate cycle in order to prevent and anticipate possible climate changes and thus help society to adopt measures for good environmental management, to take climate resilience measures in order to also migrate towards the use of clean and renewable energy sources.
Predicting the weather is also useful for ensuring food security, because if we can better assess the climate, it can help the grower to optimize and make profitable his production in the current conditions of an environment weakened by the increase in the temperatures of the emerged lands. and various other manifestations; From an economic point of view, the agricultural, fishing and tourism sectors are directly impacted by climate change: access to water, extreme events, droughts have direct consequences on agricultural yields; the acidification of the oceans, the warming of the waters degrade marine ecosystems (such as corals) and lead to the migration of marine species, which limits the potential for fishing catches in certain areas.

Thus, regarding the various elements mentioned above, it is necessary to make use of analysis tools and powerful predictions than automatic classifiers to create a learning model that can learn from a Dataset from real data in order to provide a prediction close to reality in order to help the various environmental actors in decision-making for better management for a sustainable environment.

MACHINE LEARNING MODEL

Machine Learning solves several problems today relating to the learning approach involved. Among these approaches, we will mainly mention: supervised learning, unsupervised learning, and reinforcement learning. While

A. Neural networks

In recent years, artificial deep neural networks (including recurrent networks) have won many competitions in pattern recognition and machine learning [1]. Neural networks are a way to do machine learning, in which a computer learns to perform a task by analyzing training examples. Usually, the examples have been hand labeled in advance. An object recognition system, for example, might receive thousands of labeled images of cars, houses, coffee cups, etc., and it would find visual patterns in the images that consistently match the labels. particular.

Loosely inspired by the human brain, a neural network consists of thousands, if not millions, of unique, tightly interconnected processing nodes. Most neural networks today are organized in layers of nodes, and they are "feed-forward", which means that data passes through them in one direction only. An individual node can be connected to multiple nodes in the underlying layer, from which it receives data, and to multiple nodes in the upper layer, to which it sends data.

To each of its incoming connections, a node will assign a number called "weight". When the network is active, the node receives a different piece of data — a different number — on each of its connections and multiplies it by the associated weight. It then adds the resulting products, resulting in a single number. If this number is less than a threshold value, the node does not transmit any data to the next layer. If the count exceeds the threshold value, the node "fires", which in today's neural networks usually means sending the count - the sum of the weighted inputs - along all its outgoing connections.

When a neural network is trained, all its weights and thresholds are initially set to random values. The training data is transmitted to the lower layer - the input layer - and it passes through the following layers, multiplying and adding in complex ways, until it finally arrives, radically transformed, at the output layer. During training, weights and thresholds are continually adjusted until training data with the same labels consistently produce similar outputs [2].

B. Decision Tree (DT)

One of the widely used techniques in data mining is systems that create classifiers [3]. In data mining, classification algorithms can process a large volume of information. It can be used to make assumptions about categorical class names, to classify knowledge based on training sets and class labels, and to classify newly obtained data [4]. Fig. 1 illustrate a structure of DT.

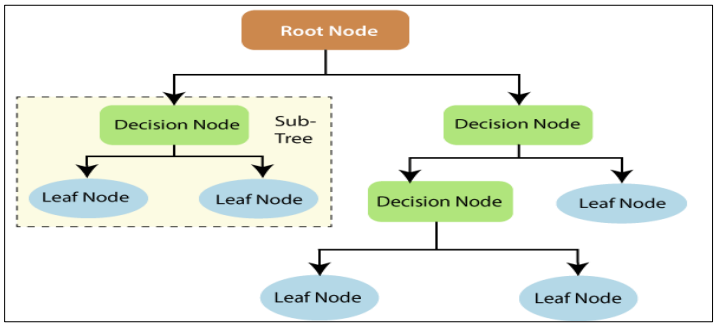

Figure 1. Decision Tree [4]

Decision trees are one of the powerful methods commonly used in various fields, such as machine learning, image processing, and pattern identification [5]. DT is a successive model that unites a series of basic tests in an efficient and consistent way where a numerical characteristic is compared to a threshold value in each test [6]. Conceptual rules are much easier to construct than numerical weights in the neural network of connections between nodes [7, 8]. Mainly for bundling purposes, DT is used. Moreover, DT is a commonly

used classification model in Data Mining [9]. Nodes and branches are composed of each tree. Each node represents features in a category to be classified and each subset defines a value that can be taken by the node [10, 11]. Due to their simplicity of analysis and accuracy over multiple forms of data, decision trees have found many fields of implementation [12].

C. K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a method that uses a supervised algorithm for the results of the newly ranked query instance based on the majority of KNN categories. This algorithm aims to classify new objects based on attributes and training samples. The KNN algorithm is very simple, based on the shortest distance between the query instance and the training sample to determine its KNN. The training sample is projected into a multidimensional space, where each dimension represents a characteristic of the data. The space is divided into sections-part according to the classification of the training sample. A point in this space is denoted class c if class c is the most common classification in the k nearest neighbor of this point.

The flow/steps that will be performed in this study can be seen in Figure 2 below:

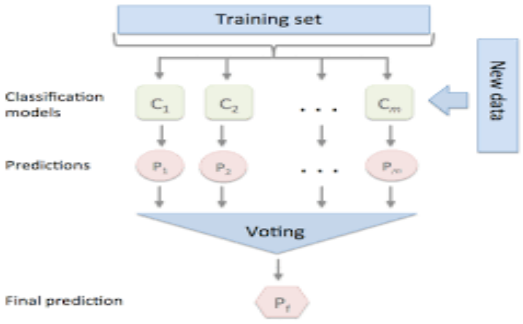

Figure 2. Voting Regressor Algorithm

The Voting Regressor (VR) is created on an intuitive and simple concept. The concept is to combine multiple machine learning models and a final predicted value is calculated by using either their average predicted value or a value predicted by the majority of the machine learning algorithms in the ensemble. The working of VR is presented in Figure 2. VR is considered very useful in the machine learning models, which are equally well-performing. It will help to predict more accurately by balancing out their weaknesses.

RESULTS AND DISCUSSION

Data collection is obtained from Mbujimayi Bipemba airport in the Democratic Republic of Congo. Below are some examples of real meteorological data which has been collected by the Bipemba weather station which contains the climatological information from 2015 to 2021 and consists of the following columns: temperature, humidity, rainfall, year, month, rainfall, rain frequency, wind speed, and storm.

A. Dataset

```
dataset.tail()
```

	Year	Month	Rainfall	weather	Storm	Humidiy	Temperature	Wind Speed	Rain Frequency
79	2021	Aout	79	3	2	45.1	34	26	5
80	2021	Septembre	110	4	9	60.0	33	30	13
81	2021	Octobre	121	4	9	71.0	33	25	13
82	2021	Novembre	101	2	14	72.0	31	22	16
83	2021	Decembre	106	4	15	73.0	31	26	19

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 84 entries, 0 to 83
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Year            84 non-null     int64
 1   Month           84 non-null     object
 2   Rainfall        84 non-null     int64
 3   weather         84 non-null     int64
 4   Storm           84 non-null     int64
 5   Humidiy         84 non-null     float64
 6   Temperature     84 non-null     int64
 7   Wind Speed      84 non-null     int64
 8   Rain Frequency  84 non-null     int64
dtypes: float64(1), int64(7), object(1)
memory usage: 6.0+ KB
```

Figure 3. Data information

The data shows us that we have 10 parameters for a total of 84 entries in the space from 2015 to 2021 and concerning the data types, we have integers.

B. Model creation

We created our model and to compare their performance, performance measures such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R-squared) were calculated.

```
from sklearn.neural_network import MLPRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean_absolute_error,r2_score,mean_squared_error

def run_experiment(model):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    score = cross_val_score(model, X_train, y_train, scoring = 'r2', cv = 2)
    print("Model score : ", model.score(X_test, y_test))
    print("R^2 score : ", r2_score(y_test, y_pred))
    print("Mean Absolute Error :", mean_absolute_error(y_test,y_pred))
    print("Root Mean Squared Error:",np.sqrt(mean_squared_error(y_test, y_pred)))
```

Figure 4. Model creation

In this article, MLPRegressor used five metrics, Model score, Validation score, MAE, RMSE, and R-squared, as shown were used in order to compare the performances of the single ML models. As shown in the figure below: Model score 0.73, Validation score 0.5, R-squared 0.73, MAE 3.2, RMSE 4,04.

```
modelMLP = MLPRegressor(hidden_layer_sizes=(84, 84, 84), max_iter= 500)

run_experiment(modelMLP)
```

```
Model score :  0.7351154886318894
Validation score :  0.5074934104333981
R^2 score :  0.7351154886318894
Mean Absolute Error : 3.2936318262974105
Root Mean Squared Error: 4.0405055169051725
```

Figure 5. MLPRegressor model

In this article, DecisionTreeRegressor used five metrics, Model score, Validation score, MAE, RMSE, and R-squared, as shown were used in order to compare the performances of the single ML models. As shown in the figure below: Model score 0.5, Validation score 0,1, R-squared 0.5, MAE 3.2, RMSE 5,5.

```
modelDT = DecisionTreeRegressor(max_depth= 3)

run_experiment(modelDT)
```

```
Model score :  0.503882017877703
Validation score :  0.10097660638580314
R^2 score :  0.503882017877703
Mean Absolute Error : 4.135448179271707
Root Mean Squared Error: 5.529678810060336
```

Figure 6. DecisionTreeRegressor model

In this article, KNeighborsRegressor used five metrics, Model score, Validation score, MAE, RMSE, and R-squared, as shown were used in order to compare the performances of the single ML models. As shown in the figure below: Model score 0.6, Validation score 0,4, R-squared 0.6, MAE 3.9, RMSE 4,9.

```
modelKNN = KNeighborsRegressor(n_neighbors= 10)

run_experiment(modelKNN)
```

```
Model score :  0.603390152706041
Validation score :  0.4339524579066285
R^2 score :  0.603390152706041
Mean Absolute Error : 3.9247058823529395
Root Mean Squared Error: 4.944121881231346
```

Figure 7. KNeighborsRegressor model

C. Combination of models: Ensemble Learning

Ensemble learning is a technique that involves training several ML models in order to take into account all their results. To do this, in this article we combined three models, setting a criterion that the score must be at least 50 percent. We tried other models that gave less than 50 percent. And the score was improved by using VotingRegressor, one of sklean's ensemblistic methods.

The choice of this method is due to the fact that we are dealing with heterogeneous models, and when we are dealing with heterogeneous models, we need to combine them in a simple way. votingregressor is a simple way of combining, but also our models give at least a 50 percent score, which is why the three models have been maintained. they are: MLPRegressor, DecisionTreeRegressor and KNeighborsRegressor

```
from sklearn.ensemble import VotingRegressor
modelVoing = VotingRegressor([('MLP', modelMLP), ('DT', modelDT), ('KNN', modelKNN)])
run_experiment(modelVoing)
```

```
Model score :  0.7924670207752081
R^2 score :  0.7924670207752081
Mean Absolute Error : 2.8551431744311686
Root Mean Squared Error: 3.576440318153632
```

Figure 8: Combination of models

By combining the different models using Votingregressor, we obtained an improvement as shown in the figure above and obtained the following result: Model score: 0,79, R-squared 0.79, MAE 2.8, RMSE 3,5.

DISCUSSION

In Table 2 can be seen comparison of methods.

Table 2. Models comparison and improvement

Models/Methods	Model Sscore	Validation score	R-squared	MAE	RMSE
MLPRegressor	0,73	0,50	0,73	3,29	4,04
DecisionTreeRegressor	0,50	0,10	0,50	4,13	5,52
KNeighborsRegressor	0,60	0,43	0,60	3,92	4,94
Votingregressor	0,79		0,79	2,85	3,57

The Figure 9 below shows how the votingregressor improves the model score and R-squared and minimizing the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

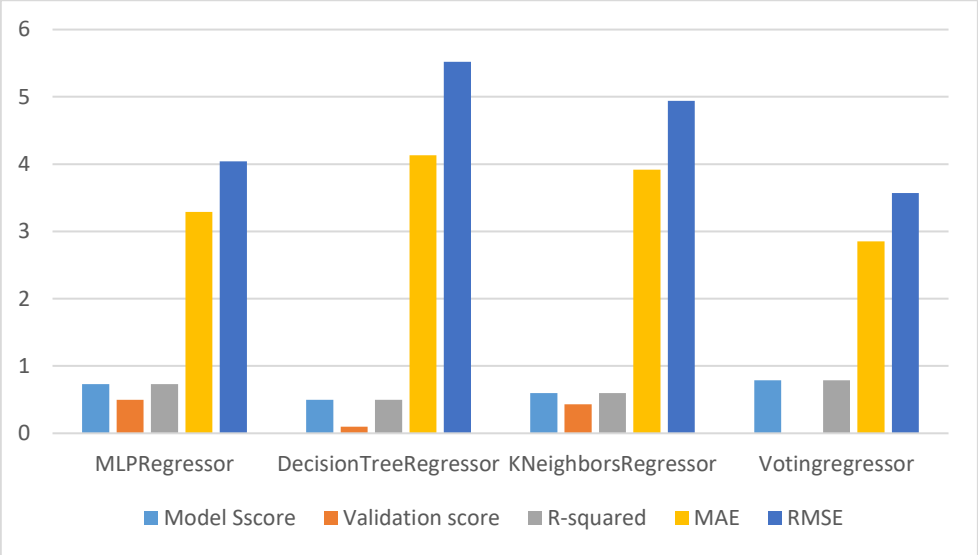

Figure 9. Comparison and improvement

CONCLUSION

This article presents the Efficiency of the combination of machine learning models in the evaluation of the meteorological parameters of bipemba airport in Mbujimayi and the combination of models with the VotingRegressor method would be the most suitable because the best performance on the prediction would therefore be to considerably reduce the differences between the estimated values of the predicted parameters and their true values when they are taken and thus we can better anticipate certain situations that can permanently affect the lives of the inhabitants and therefore propose to decision-makers to take effective measures in the management of the environment and the fight against climate change. For the purposes of this article, we made a parallel combination of the models (MLPRegressor, DecisionTreeRegressor and KNeighborsRegressor) with the Voting Regressor method, and the efficiency was improved, as can be seen in Table 1.

REFERENCES

[1] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. **61**: 85-117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
[2] Hardesty, Larry (14 April 2017). "Explained: Neural networks". MIT News Office. Retrieved 2 June 2022.
[3] S. S. Nikam, “A comparative study of classification techniques in data mining algorithms,” Oriental journal of computer science & technology, vol. 8, no. 1, pp. 13–19, 2015.

[4] C. Z. Janikow, “Fuzzy decision trees: issues and methods,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 28, no. 1, pp. 1–14, 1998.
[5] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, “Decision tree classifier for network intrusion detection with GA-based feature selection,” in Proceedings of the 43rd annual Southeast regional conferenceVolume 2, 2005, pp. 136–141.
[6] I. S. Damanik, A. P. Windarto, A. Wanto, S. R. Andani, and W. Saputra, “Decision Tree Optimization in C4. 5 Algorithm Using Genetic Algorithm,” in Journal of Physics: Conference Series, 2019, vol. 1255, no. 1, p. 012012.
[7] R. Barros, M. Basgalupp, A. de Carvalho, and A. Freitas, “A Survey of Evolutionary Algorithms for Decision-Tree Induction,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 42, pp. 291–312, Jan. 2012, doi: 10.1109/TSMCC.2011.2157494.
[8] G. Gupta, “A self-explanatory review of decision tree classifiers,” in International conference on recent advances and innovations in engineering (ICRAIE-2014), 2014, pp. 1–7.
[9] S. S. Gavankar and S. D. Sawarkar, “Eager decision tree,” in 2017 2nd International Conference for Convergence in Technology (I2CT), Mumbai, Apr. 2017, pp. 837–840, doi: 10.1109/I2CT.2017.8226246.
[10] P. H. Swain and H. Hauska, “The decision tree classifier: Design and potential,” IEEE Transactions on Geoscience Electronics, vol. 15, no. 3, pp. 142–147, 1977.
[11] A. Dey, “Machine learning algorithms: a review,” International Journal of Computer Science and Information Technologies, vol. 7, no. 3, pp. 1174–1179, 2016.
[12] J. Mrva, Š. Neupauer, L. Hudec, J. Ševcech, and P. Kapec, “Decision Support in Medical Data Using 3D Decision Tree Visualisation,” in 2019 E-Health and Bioengineering Conference (EHB), Nov. 2019, pp. 1–4, doi: 10.1109/EHB47216.2019.8969926.
[13] Y. Bengio, O. Delalleau, and C. Simard, “Decision Trees Do Not Generalize To New Variations,” Computational Intelligence, p. 19.
[14] S.-Y. Liang, D.-Q. Han, and C.-Z. Han, “A novel diversity measure based on geometric relationship and its application to design of multiple classifier systems,” Acta Automatica Sinica, vol. 40, no. 3, pp. 449–458, 2014.