Weather Forecasting With Random Forest Algorithm

Bahadir Erkam Bakoglu - 001089837 - bb0948y

Abstract

Random forests or Random Decision Forests are ensemble learning methods for classification, regression, and other classification tasks. They involve the construction of a large number of decision trees at the time of training. When solving classification problems, random forests are used to select the class that is selected by the majority of trees. In the case of regression tasks, the mean prediction of individual trees is returned. Random forest algorithms may be utilized for predicting various parameters in a set of data. Additionally, it also includes weather forecasts, as well as temperature and weather conditions. This paper will describe and argue the differences between different algorithms including linear regression, logistic regression, decision trees and random forest algorithms that are used for weather forecasting and predictions.

1. Introduction

Forecasting the weather is the process of predicting future weather conditions. The process relies on the input of previous records of temperature, weather condition and other details to predict the weather condition on a particular day according to a trained algorithm (Hill, Schumacher, and Jirak 2022). It is now imperative that we develop learning algorithms that can keep up with the volume and quality of data in modern data sets. This is while maintaining sufficient statistical efficiency at the same time. Random forest algorithms are trained using bootstrap aggregation, which is a general technique used in machine learning (Biau and Scornet 2016). Among the aggregation schemes employed by random forests, the modified tree learning algorithm is also employed, which selects randomly at each candidate split during the learning process. There is also a term called "feature aggregation" that is used to describe this process.

This paper will describe the concept behind the random forest algorithm and provide examples to explain the algorithm. Furthermore, an overview of the algorithm and the associated libraries will be presented. Then, the explanation, evaluation and results of the study will be described. This will establish a foundation for comparison between other decision algorithms including linear regression, logistic regression, and decision trees. Finally, an evaluation of the development process of the project and the possibilities for further work will be provided along with an assessment of the efficiency of the project.

2. Methods

It is the goal of random forests to reduce variance by averaging multiple deep decision trees that have been trained on different segments of the same training set. Although the performance of the final model is generally greatly enhanced, there is a small increase in bias and some loss of interoperability. It can be viewed as a symbiotic relationship between decision tree algorithms and random forests. Enhancing the performance of a single random tree by combining the skills of many random trees. Despite the fact that random forests are not identical, they provide the same results as k-fold cross validation.

It is imperative that the algorithm be trained in order to achieve maximum accuracy. To achieve this, random forests are trained using the general technique of bootstrapping (Livingston 2005). This technique is known as bootstrap aggregation. Given a training set

$$X = x_1, ..., x_n \tag{1}$$

with responses

$$Y = y_1, ..., y_n \tag{2}$$

,aggregating repeatedly, B times, by using the training set as a replacement, selects a random sample and fits trees to if:

$$b = 1, ..., B : \tag{3}$$

Sample *n* training examples from *X, Y*. As a result, these

become X_b, Y_b. When all the regression trees have been trained, predictions for unseen samples can be made by averaging their predictions. Averaging all the predictions of the regression trees on $x^{'}$ can be used to determine predictions for unseen samples $x^{'}$ after training:

$$\frac{1}{B}\sum_{b=1}^{B} f_b(x^{'}). \tag{4}$$

Due to the bootstrapping method, the model performs better because it reduces variance without increasing bias. Consequently, the average of many trees is not greatly affected by noise in its training set. However, the predictions of a single tree are highly sensitive to noise in its training set (Biau 2012). The reason for this is that the trees are not associated with each other. In the event that many trees are trained on a single training set, the result would be strongly correlated trees. The bootstrap sampling method involves showing the trees different sets of training data in order to de-correlate their structures.

Furthermore, the standard deviation of all the individual regression trees on $x^{'}$ may be used to estimate the uncertainty of the prediction:

$$uncertanity = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - f)^2}{B-1}} \tag{5}$$

It is a parameter of free choice whether there will be a certain number of samples or trees, *B*. According to the size and nature of the training set, a few hundred to several thousand trees are typically required. Cross-validation can be used to determine the optimal number of trees. This can be determined by using only those trees in the bootstrap sample that did not have x_i in their training sample. Once a certain number of trees have been fitted, training and testing errors tend to level off. As described above, the original bagging algorithm for trees can be found here. As well as random forests, there is another type of bagging scheme that is implemented as part of the learning process - namely, the modified tree learning algorithm which selects a random subset of features from among each candidate split in the learning process. There is a term sometimes used to describe this process, which is "feature bagging". Using a bootstrap sample, this is done because of the correlation between *B* trees in an ordinary bootstrap sample: if one or a few features have a high degree of predictive power for the response variable, they will be selected for inclusion in many of the *B* trees, leading to these trees becoming correlated with each other.

Typically, for a classification problem having a number of *p* features, *sqrt(p)* features are considered in each split. In order to achieve the most accurate results in regression problems, it is optimal to consider $p/3$ with a minimum of 5 nodes. This is a reasonable default value. There are a number of parameters that can be tuned here but the most appropriate values should be selected and fine-tuned on a case-by-case basis for each issue. Added one more step of randomization results in extremely randomized trees, which are known as Extra trees. Even though it is similar to ordinary random forests in that they are ensembles of individual trees, there are two main differences between them: the first is that each tree is trained using the entire learning sample, and the second is that the split between the trees is randomized in the tree learner. As an alternative to computing the most optimal cut-point locally, a random cut-point is chosen for each feature under consideration. A uniform distribution is used to select a value within the range of empirical values for this feature. Following that, the node is split based on the split that produces the highest score out of all the randomly generated splits. Random forests can be configured to consider a certain number of randomly selected features at each node, similar to ordinary random forests (Probst, Wright, and Boulesteix 2019). This parameter has a default value of *sqrt(p)* for classification, *p* for regression, where p is the number of features in the model.

3. Experiments

The data set acquired for the project included an enormous amount of weather condition and temperature data from year 2003 to 2020. The data has been stored into a CSV file to be integrated for use of the project. The setting of the project and the evaluation criteria is specified as followed.

3.1. Experimental settings

The data set for the project consists of eight columns. However, for the output of the weather prediction requires only one of the columns which is weather condition. In addition to the data set, the hyper parameters used in the random forest algorithm to get the most accurate data possible include twenty decision trees where every tree has a depth of seven, bootstrap number of three-thousand, five number of features and classification task. Furthermore, the splitting of the data is by using a 70 percent, 20 percent, 10 percent split for the training, validation and test sets.

3.2. Evaluation criteria

A random forest algorithm is evaluated based on accuracy of 80 percent in predicting weather conditions and a mean squared error score of 5 percent. Due to the nature of the random forest algorithm, the accuracy criteria to be met are quite appropriate. In comparison with other specified algorithms, this algorithm requires a much higher number of parameters and the evaluation period of the algorithm is more challenging to implement. An average value of 5 percent is considered an optimal goal when it comes to the mean squared error of the algorithm since the random forest is a larger scaled algorithm then the other specified algorithms.

3.3. Results

The random forest algorithm's accuracy for the weather condition predicted by using the specified data set is recorded as 84-percent. It is estimated that the root mean squared error (RMSE) for the algorithm is 4 percent. The results obtained meet all the evaluation criteria for the algorithm to be deemed successful. Taking into account the depth used in each individual decision tree built in the algorithm, it can be stated that the results obtained are sufficient considering the depth used. If we were to use much larger decision trees and train the algorithm accordingly, the accuracy output may have been higher (Yao et al. 2020).

3.4. Discussion

The Random forest algorithm to predict weather conditions proved to be more efficient compared with the other decision algorithms such as linear regression, logistic regression, decision trees.

The linear regression algorithm is an algorithm that is based on supervised learning in machine learning. The program performs a regression analysis. Based on independent variables, a regression model predicts a target value. In most cases, it is used to determine the relationship between variables and forecasting.

In artificial intelligence, decision trees are used to reach conclusions based on the information available from previous decisions. Additionally, each of these conclusions is assigned a value that is used in the prediction of what course of action should be taken in the future as a result of these conclusions. Decision trees are the smaller scale implementations of random forests. The primary difference between decision trees and random forests are the number of operations carried out in the algorithm to acquire an output

Logistic regression, as a statistical algorithm, is used to calculate or predict the probability of occurrence of binary events such as yes/no. Similar to decision trees the usability of logistic regression is limited compared to random forest algorithms.

As for the comparison part of the algorithms the decision tree algorithm used a maximum depth of twenty for the decision tree and the number of maximum number of leaf node of two-hundred as hyper parameters. Ultimately the decision tree algorithm got an accuracy of 93 percent.

Logistic regression algorithm used a saga solver class with a balanced weight and number of maximum iterations in the algorithm of thousand as the hyper parameters. As a result, the logistic regression algorithm achieved a total of 71 percent of accuracy in case of weather condition prediction.

Linear regression algorithm used days taken as hyper parameters and achieved a 74 percent accuracy of weather prediction and related information about weather forecast.

This proves sufficient evidence that random forest algorithm is more throughout and detailed in case of predicting a future event while looking thought previous data. The capability and scale of random forests are greater than the specified algorithms, even though decision tree algorithm achieved a higher accuracy score in this specific instance the accuracy difference can be explained by the different maximum decision tree depth in both algorithms. As the decision trees and the number of nodes increases, the accuracy of the prediction enhances.

4. Conclusion

Using the specified data set and hyper-parameters, the random forest algorithm has been shown to be more effective than linear regression, logistic regression, but less effective than decision tree algorithms when considering the specified data set and hyper-parameters. In spite of this, it is clear that the random forest algorithm is more detailed and large scaled than other algorithms. It is also deemed that the algorithm will work better in the case of larger amounts of data if it receives a larger input to the hyper parameters.

Acknowledgements

I appreciate my teammates during the development of the project for lending me their knowledge and advertise to be achieve a successful outcome. The code that has been developed during the project and the data set used are referenced from here[1] and deserves our thanks for providing an excellent example.

References

Biau, Gérard (2012). "Analysis of a random forests model". In: *The Journal of Machine Learning Research* 13.1, pp. 1063–1095.

Biau, Gérard and Erwan Scornet (2016). "A random forest guided tour". In: *Test* 25.2, pp. 197–227.

Hill, Aaron J, Russ S Schumacher, and Israel Jirak (2022). "A new paradigm for medium-range severe weather forecasts: probabilistic random forest-based predictions". In: *arXiv preprint arXiv:2208.02383*.

Livingston, Frederick (2005). "Implementation of Breiman's random forest machine learning algorithm". In: *ECE591Q Machine Learning Journal Paper*, pp. 1–13.

Probst, Philipp, Marvin N Wright, and Anne-Laure Boulesteix (2019). "Hyperparameters and tuning strategies for random forest". In: *Wiley Interdisciplinary Reviews: data mining and knowledge discovery* 9.3, e1301.

Yao, Han et al. (2020). "Application of random forest algorithm in hail forecasting over Shandong Peninsula". In: *Atmospheric research* 244, p. 105093.

[1]The code supporting the document is referenced from (https://github.com/AbdullahMakhdoom/Weather_Prediction)