010 Element ASSIGNMENT

(NO MEDIUM.COM)

Adrien Montaigne
2248806

Table of Figures

The Why, How and When of Random Forest

Data science has become more prevalent among companies, due to the increase in data storage capacities. The data storage capacities have increased exponentially in the last decades. The number of people working in data has significantly grown to satisfy this need. The job of Data scientist can be summarized in four steps: Data Extraction, Data Understanding, Data Exploitation (including modeling) and Production launch. A data scientist spends 80% of the time on Data Extraction and Data Understanding.

One of the most common machine learning methods: Random Forest Regression, will be discussed. It was introduced by Ho Tim Kam in 1995, she was the first person to explain the idea of creating several trees to improve accuracy. In 2006, Leo Breiman and Adele Cutler use the name Random Forest for the first time as a trademark.

Figure 1: Ho Tin Kam [1]

Random Forest Regression is a supervised machine learning for continuous, discrete, or qualitative variables. Supervised means that the dataset is labeled. Random Forest is inspired by the idea that unity is strength. The concept of Random Forest is to create several decision trees from the different training sets in order to create one decision tree for all the training set. The model chooses the majority or averaging answer from the different trees (the majority for discrete or qualitative outcomes, the average when the outcomes is continuous). Figure 2 represents the functioning of Random Forest.

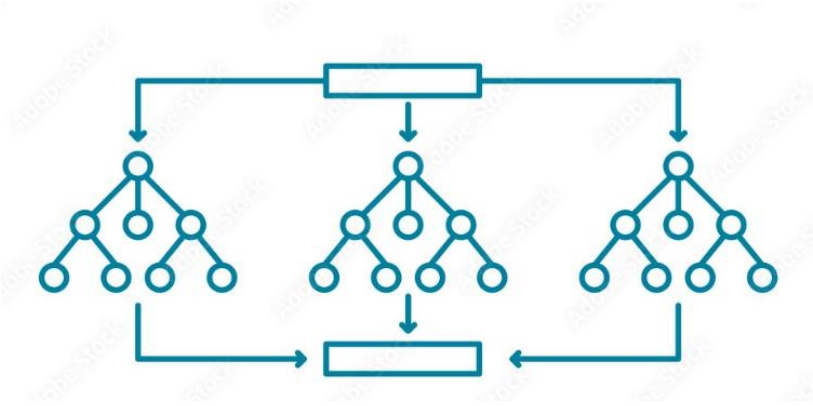

Figure 2: diagrams on Random Forest [2]

A decision tree permits to classify data based on a series of questions represents by nodes of the tree. The nodes of the tree are created thanks to a function as entropy or information gain which aims to maximize the selection. The leaves correspond to the individuals. The nodes correspond to how the decision tree classifies the data. Figure 3 represents how the decision tree work. We could determine whether the fruit is an apple, strawberry or orange based on its size and color.

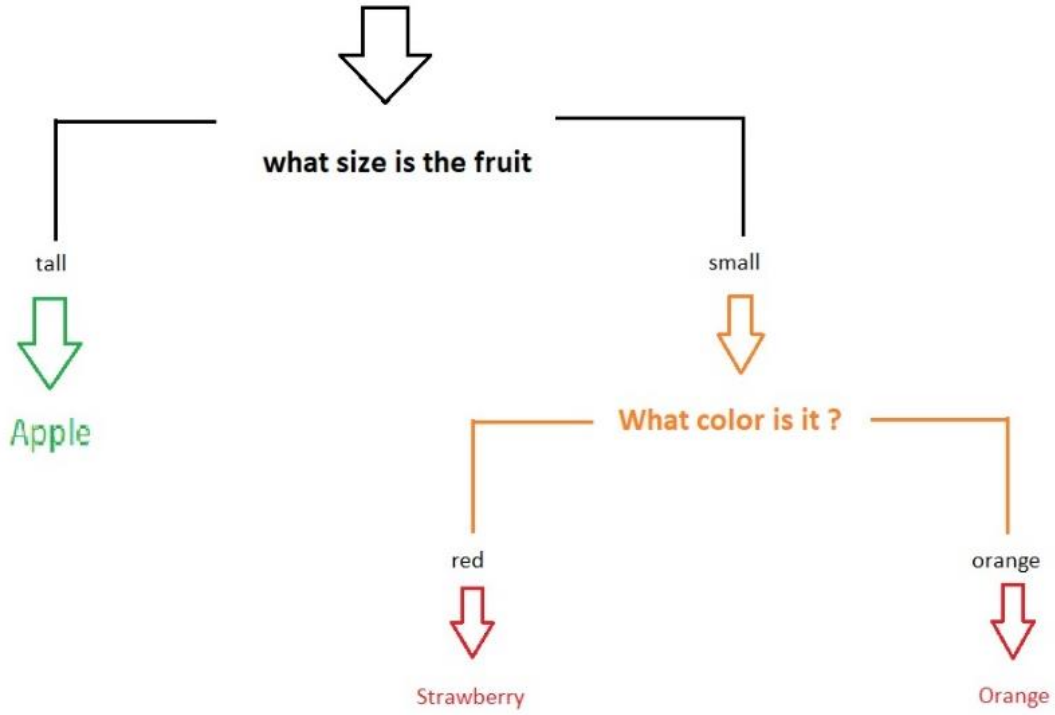

Figure 3: Diagrams on the decision tree

How to use random forest thanks to Python?

Random Forest is very simple to implement. We will see in this instance the data preprocessing, the implementation of Random Forest, the evaluation of the model, how to optimize the model, and the importance feature.

How to preprocess the data for Random Forest?

First of all, we need to use different libraries like pandas, numpy, sklearn and matplotlib and a dataset named titanic.csv (the link of the dataset is in description) in order to get an example of a random forest. The shape of the dataset is (890, 12). The dataset has 12 features: Age, Sex, PassengerID, Pclass, Name, Ticket, SibSp, Parch, Fare, Cabin, and Embarked. We will try to predict if a person survives in function diverse features. We split randomly the dataset to get a test and training set for X and Y. Afterwards, we encode the columns for the gender and P-class. That is transforming male or female in 1 or 0, or transforming S, C, or Q in 0, 1, or 2. It exists different ways to make it.

```
Pclass        0
Name          0
Sex           0
Age         144
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       547
Embarked      2
dtype: int64
```

Figure 4: Missing value

We observe that it misses some values in the columns named “Age” and “Embarked”. We will treat missing values in replacing missing values by mean or by the majority (majority for the values “Embarked” and the mean for the values “Age”.

We will remove some columns that are not useful for the model. The feature “Name”,” Cabin” and” Ticket” are columns containing string characters. We will standardize and normalize the variables to obtain the same dimension for each one.

How to implement Random Forest?

We will initialize a random forest thanks to the library: sklearn from the training set. In this instance, I've selected 20 trees (n estimators = 20).

```
[ ]  from sklearn.ensemble import RandomForestClassifier
     from sklearn.datasets import make_classification
     clf = RandomForestClassifier(n_estimators=20 , random_state=0)
     clf.fit(X_train, Y_train)
     result = clf.predict(X_test)
```

Figure 5: Implementation of the model

we will predict y_pred from X_test and calculate the confusion matrix, f1 score, and accuracy score.

How to evaluate Random Forest?

```
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score
cm = confusion_matrix(Y_test, result)
print(cm)
print(accuracy_score(Y_test, result))
print(f1_score(Y_test, result))
```

```
[[95 11]
 [21 52]]
0.8212290502793296
0.7647058823529411
```

Figure 6: Evaluation of the model

The more the accuracy score and F1 score are close to 1, the more the model is good. In contrast, the more the accuracy score and F1 score are close to 0, the more the model is bad. In this instance, the model is good because the accuracy score and F1 score are close to 1. It's preferable to use the f1 score when the dataset is unbalanced and preferable to use the accuracy score when the dataset is balanced. 0.82 and 0.76 for the accuracy score and f1 score, respectively is a decent score. Other measures, such as sensitivity, threshold, or specificity, exist.

How to optimize the model?

We could adjust some hyperparameters, such as the number of trees or the maximum number of features, to raise the accuracy score. The ideal n estimator in this case is 20. n_estimator is the number of trees in the model. The default values for the other hyperparameters are good, so there is no need to alter them. We could choose different node-splitting functions as entropy or information gain. By default, the node-splitting function is Gini. Other functions, including entropy or log loss, produced accuracy that was equal to or inferior compared to the Gini function.

Accuracy may be increased by using ensemble learning techniques like Boosting, voting classifiers, or stacking. The idea of a Voting classifier is to train different models and to vote on the various models. Boosting involves building a number of models. For boosting, each model is complementary to the others. The final model predicts new variables. Stacking is instead of gathering predictions in the form of a vote, a model is asked to learn which model is right or wrong.

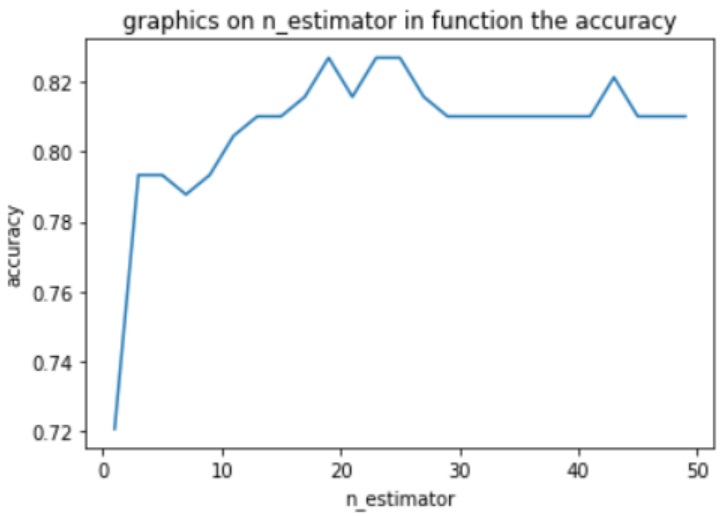

Figure 7: Graphics on the optimum number of trees

Understand Random Forest: Feature Importance

It can be observed that Age, Sex, and Fare contribute a lot to the construction of this model compared to SibSp, Parch, and Embarked. It seems coherent for sex, age, and Fare are strongly correlated to the survival column. It is sometimes necessary to select the features to reduce the complexity of the model and get a better interpretation of the model.

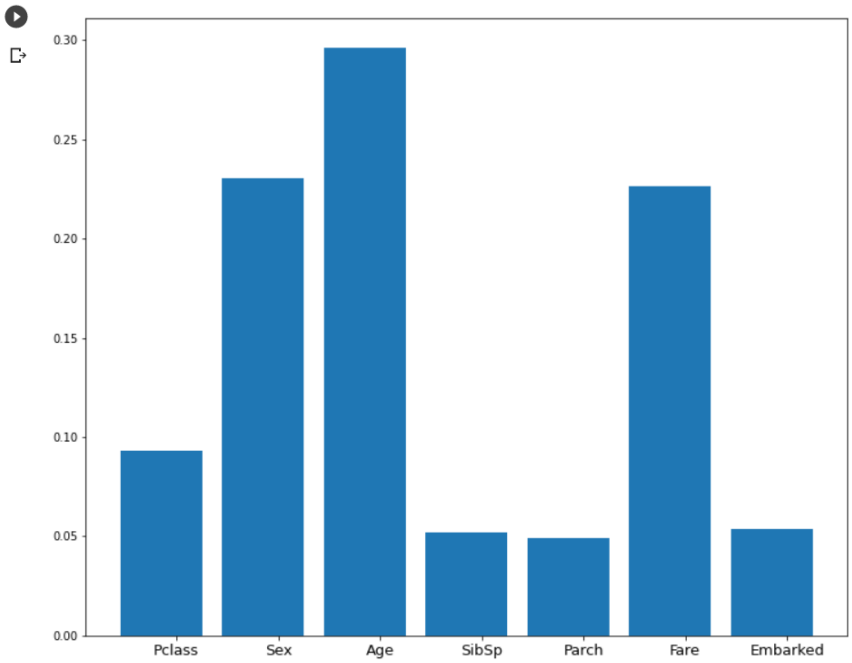

Figure 8: Graphic on the feature importance

What are the benefits of Random Forest?

Random forest is one of the best-known prediction models. Most data scientists know or use random forest in their projects. It usually gets a better accuracy than a decision tree or another machine learning algorithm. It is also good at dealing with missing values because it automates missing values present in the data or outliers. It's a model with little hyperparameter (number of trees, number of nodes, and the depth of the tree). Random Forest is easily comprehensible from a human point of view compared to another machine learning like an artificial neural network. We can interpret what is going on inside the algorithm. As a decision tree, Random Forest is fast and can handle a large dataset. It's a model that can give a measure of feature importance. It can be useful if we want to select and remove some variables. Random Forest is a model very easy to implement compared to model like SVM and perform well with high dimension data. To finish, Random Forest can solve different problems like regression or classification tasks. That is the biggest advantage of Random Forest: Versatility.

What are the disadvantages of Random Forest?

Random forest is a model that requires more data stockage than a decision tree. Random Forest can easily overfit. The performance of the model will increase with the number and length of the roots. Random Forest is a model sensitive to the environment since it has a good performance in the condition that the environment does not change a lot. Random Forest does not provide a coefficient as linear or multiple regression. Random Forest is not as good when the dataset has a low dimension or little feature. The number of trees and the depth of each tree can make the algorithm too slow. Random forest is usually fast to train. But, the prediction can take much more time. Random Forest should not use when the training set and the test set iare completely different.

This is a table summarizing the advantages and the disadvantages of Random Forest.

Advantages of random forest	Disadvantages of random forest
- Robustness against missing values or outliers - Fast, can handle a large dataset - Easy to implement - Versatility (regression/classification tasks) - Comprehensive for human point of view - Little hyperparameter	- Over – fitting - Memory - Sensitive to the environment - Bad to Low dimension - Complexity

Figure 9: Table on the advantage and the disadvantages of Random Forest

Conclusion

In this article, an introduction to Random Forest was presented. Random Forest is a super tool for Data Scientist. I hope you will use it more often Random Forest as machine learning. Thank you for taking the time to read my article. I hope you learned more about Random Forest. Do not hesitate to comment on your opinion on the blog. It was a joy writing this article.

Reference

If you want to know more about the subject or check the source, I recommend you to check these websites.

E R, S. (2021a). *Random Forest | Introduction to Random Forest Algorithm*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

[1] Ibm.com. (2016). *Tin Kam Ho - IBM*. [online] Available at: https://researcher.watson.ibm.com/researcher/view.php?person=us-tho [Accessed 23 Mar. 2023]

kaggle.com. (2012). *Titanic - Machine Learning from Disaster*. [online] Available at: https://www.kaggle.com/competitions/titanic/data.

Ravindran, S.K. (2021). *Random Forest in Simple English: Why is it so popular?* [online] Medium. Available at: https://towardsdatascience.com/random-forest-in-simple-english-why-is-it-so-popular-3ba04d0374d.

Tat, M.J. (2017). *Seeing the random forest from the decision trees: An explanation of Random Forest*. [online] Medium. Available at: https://towardsdatascience.com/seeing-the-random-forest-from-the-decision-trees-an-intuitive-explanation-of-random-forest-beaa2d6a0d80#:~:text=Random%20Forests%20allow%20us%20to [Accessed 23 Mar. 2023].

[2] Vector, T. (n.d.). *Random forest line icon. Decision trees symbol. Machine learning technique that's used to solve regression and classification problems. Complex problems solution. Vector illustration, flat, clip art. Stock Vector*. [online] Adobe Stock. Available at: https://stock.adobe.com/hu/images/random-forest-line-icon-decision-trees-symbol-machine-learning-technique-that-s-used-to-solve-regression-and-classification-problems-complex-problems-solution-vector-illustration-flat-clip-art/474661732 [Accessed 23 Mar. 2023].