Portfolio3 –Week 7-Random Forest

Qiaoyu Wang

0 Introduction

This week we looked at Ensemble Models, such as Random Forest, which luckily aligned with the questions I posed in Portfolio2 last week. I've been contemplating **how to naturally handle feature importance and interactions between features**, as well as **how can we use simple decision trees to handle larger datasets more effectively**. I guess Ensemble Models might give me some insights.

So I divided the portfolio into 2 parts.

In the first part of the portfolio, I persisted in utilizing the same dataset (https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality/data) to delve deeper into the first question, seeking a better understanding of feature importance.

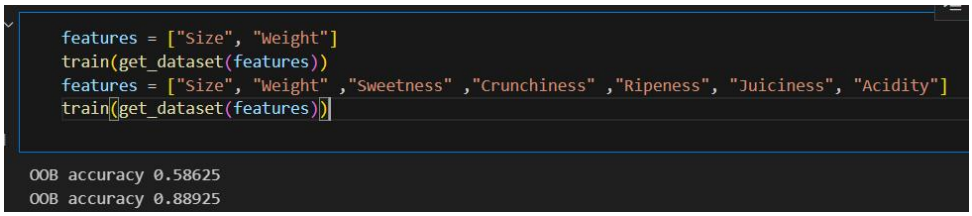
For the second part, I diverged by downloading a new, larger dataset (https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction) and fed it into the model to assess prediction accuracy, trying to find some insights into how the model behaves with more extensive data.

1 Explore Feature Importance

1.1 Train Random Forest Model

Firstly, I applied the dataset to the random forest model discussed in this week's lecture notebook and obtained the accuracy results successfully. Compared to the simple decision

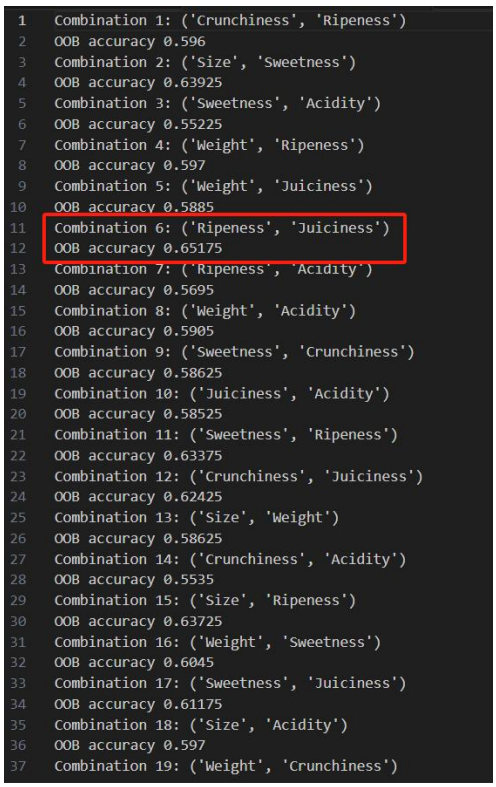
tree model, the accuracy based on 2 features appeared to be nearly identical, but the accuracy based on the entire features list increased significantly from 80 to 88. **This suggests that even if we enhance the complexity of the model, if the features aren't sufficiently aligned with the target prediction, we may still fall short of achieving an ideal prediction.**

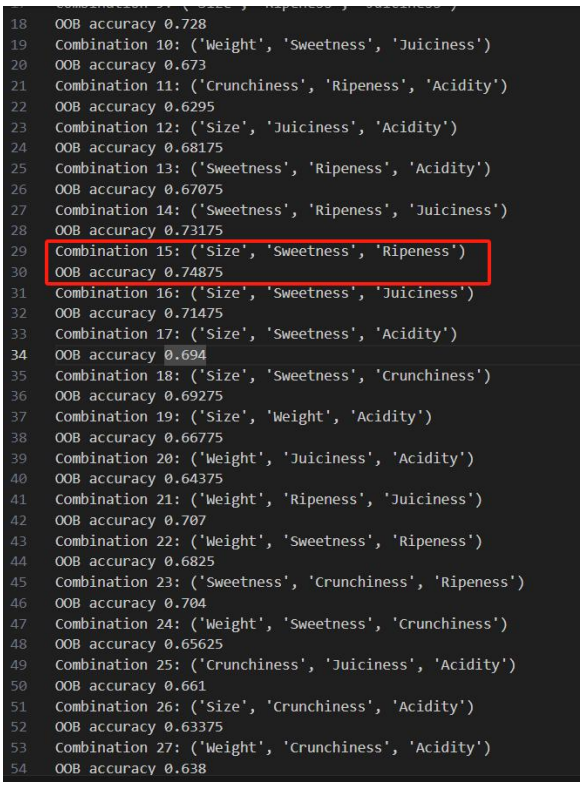


1.2 Explore Feature Importance and Interactions

1.2.1 For-Loop of all possible feature sets

Before delving into feature importance, I initially conducted a preliminary exploration using a for-loop to find some insights into the connection between features. It showed 21 combinations of 2-feature sets, with the top accuracy observed in the 'Ripeness' and 'Juiciness' feature set, reaching 65. I also experimented with 3-feature set, observing an increase to nearly 75 in accuracy, particularly with the 'Size', 'Sweetness', and 'Ripeness' feature set.





(Left: Accuracy result based on random 2 feature set;

Right: Accuracy result based on random 3 feature set)

Now I got a foundational understanding of feature connections and started to use feature importance to identify the most relevant features. I searched online and found that there were two common ways for calculating feature importance in random forest models: Gini Importance and Mean Decrease Accuracy.

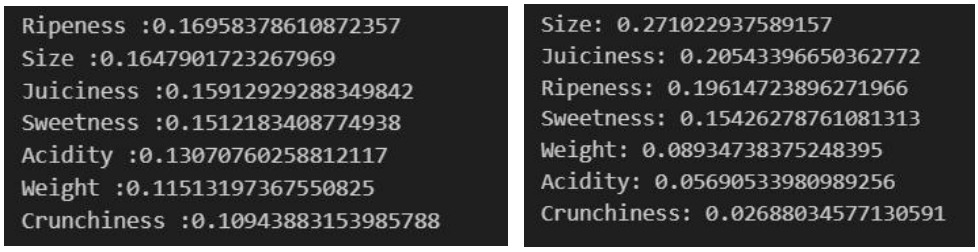
1.2.2 Gini Importance (Gini impurity or Mean Decrease impurity)

In decision trees, nodes are split based on features to minimize impurity. Gini impurity is a method of measuring how each feature decrease the impurity of the split in the node.[1] I used “model.feature_importances_” to calculate it, and the result is shown in the following

picture, along with a comparison to that of the simple decision tree model from last week.

It's clear that there were some differences between Random Forest and Decision Tree models because of their different mechanisms for calculating feature importance, leading to variations in the importance scores assigned to features. **We can observe that the distribution of each feature differs significantly in the Decision Tree model. Instead, it is evenly distributed in the Random Forest model.** This phenomenon aligns well with the nature of both models: the Decision Tree works as a simple model, which inevitably tends to inflate the importance of some features. In contrast, the Random Forest ensemble comprises a set of decision trees forming a forest, result in a more complicated but accurate prediction.

Stepping away from the data and considering the real world, each of these features has a significant connection with apple quality, making it challenging to discard any one of them for defining quality.



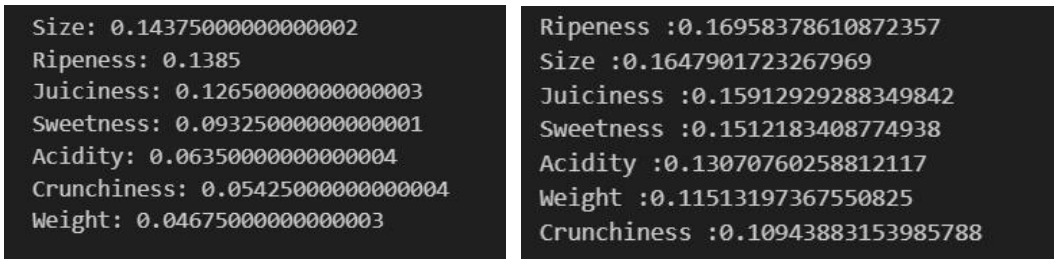
(Left: Gini Importance in random forest model;

Right: Feature Importance in decision tree model)

1.2.3 Permutation Feature Importance

Permutation Importance measures feature importance by re-shuffling values from one feature in the selected dataset, passing the dataset to the model again to obtain predictions and calculate the metric for this modified dataset. The feature importance is the difference

between the benchmark score and the one from the modified (permuted) dataset. [2]



(Left: Permutation Feature Importance in random forest model,

Right: Gini Importance in random forest model)

In terms of the ranking of features, there isn't much disparity between these two methods, albeit they yield different distribution values, likely owing to the nuances of their calculation. **Notably, "Size" and "Ripeness" emerge as crucial features, nearly tied in importance, while "Juiciness" and "Sweetness" consistently occupy the third and fourth positions, respectively. Let's get back to the for-loop analysis of all possible feature sets (Chapter 2.1), it's expected to observe that both top 2-feature set ('Ripeness', 'Juiciness') and 3-feature set ('Size', 'Sweetness', 'Ripeness') correspond closely with our rank results. Now we can figure out features that are relatively correlated with quality and those are less so.**

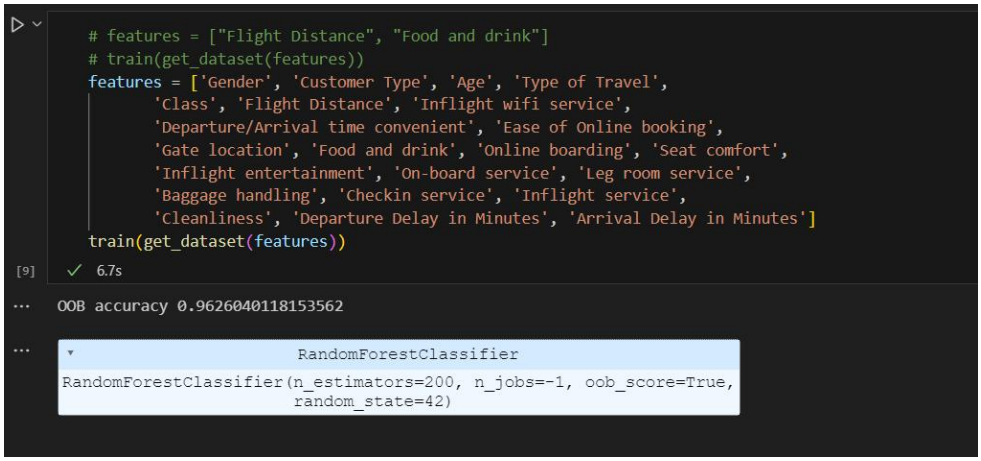
And also in terms of comparing both two methods of feature importance, **it's regrettable that due to the small scale and simplicity of the dataset, we couldn't glean substantial differentiation between them.** I searched online and found lots of profound research papers on this topic, indicating that delving deeper into it in the future would be worthwhile.

2 Explore Larger Dataset

After figuring out my first question, I got into next one: **how can we use simple decision trees to handle larger datasets more effectively**? Now we have a forest instead of one simple tree, let's try how it worked on a larger dataset.

I found a dataset on Kaggle, which had more than 100k rows of data. I fed it into the model and **it just took 6.7s to be trained, with a comprehensive 22-feature set.** The result was also pleasing to the eye, **which reached to 96, a very accurate prediction.**

This performance suggested that the RF model could adeptly handle larger and more complex dataset. **But how large and how complex? Were there any limitations?** I searched online and found a case where an individual attempted to train a RF model on a dataset comprising 30 million rows, and got crashed because of limited memory resources, which highlighted a key aspect of the problem: **as the dataset size increases, the trees' depth grows, demanding substantial memory resources that might exceed the capacity of a laptop.** It may be worth trying if we can have hardware resources such as high-memory servers or cloud-based solutions that offer scalable computing capabilities, allowing for the processing of larger datasets without memory limitations.



3 Conclusion

It's gratifying to gain a preliminary understanding of random forests. I've acquired knowledge about two new methods for calculating feature importance and successfully applied them to my dataset. Additionally, I've ventured into analyzing a larger dataset to observe how the random forest model performs under such conditions.

However, as I delved deeper, the more questions come out in my mind. One such query emerged when considering scenarios where the dataset for model training is insufficient: how to generate synthetic data based on the original dataset? While researching this topic, I found a relevant project online, but it took time to understand it. (https://medium.com/analytics-vidhya/a-step-by-step-guide-to-generate-tabular-synthetic-data set-with-gans-d55fc373c8db)

Furthermore, as I mentioned in this article, how to compare different methods for calculating feature importance, especially when the relationships between features grow increasingly complex. Additionally, how to determine the boundaries of the scale and complexity for a random forest model? It is reassuring to discover that there is so much more to explore. I'll keep on moving.

4 Bibliography

[1] Piotr Płoński, 2020. Random Forest Feature Importance Computed in 3 Ways with Python. Available at: https://mljar.com/blog/feature-importance-in-random-forest/

[2] Eryk Lewinson, 2019. Explaining Feature Importance by example of a Random Forest.

Available at: https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e