**## 2. \*\*Ensemble: Random Forest\*\*:**

For our next methods, we move from regression analysis to another form of mathematical methodology called, Statistical Ensemble. Statistical Ensemble, or simply just ‘Ensemble’, is an idealization consisting of a larger number of virtual copies of a system, considered all at once, each of which represents a possible state that the real system might be in. In other words, we create many virtual copies of our system that sample each run with a sample of the data and create their own predictions. We then take all the predictions and combined them to create a prediction that should theoretically be better than each of the single predictions on their own.

In machine learning, we utilize the Ensemble method to create multiple base models or base learners, also known as weak learners. Each of these weak models have poor prediction abilities by design. Some of the models will generate high bias, some will have high variance, and vice versa, etc. Once they have combined their training, all of these base models will be combined into a single, higher-performing model that will be more accurate and have low variance.

The way that these Ensemble methods run is where we get the three categories of ensemble methods, Bagging, Boosting, and Stacking/Blending. In bagging, we create many different weak learners, all who each have a random sample of the training data. These are all ran at the same time in the same way and then are ultimately combined into a single model at the end. With Boosting, the process is more iterative, with each weak learner running sequentially, learning from its predecessor and becoming its successors foundation. This method works to improve upon itself and builds up reducing the errors with each iteration. Stacking/Blending is similar to bagging but instead of utilizing the same model to train each random sample, we use several different models that are trained independently, before being combined into a single model at the end.

One of these ensemble learning methods is known as Random Decision Forest or Random Forest, which is what we used for our second model. The Random Forest method is a bagging model that creates a multitude of weak decision trees learners. Each of these decision trees leaners are then ran with a random sample of data and their ending output results are held until all learns complete their works. Once all of the learners have finished the combined result is determines either our class or prediction. When used for regression the outputs are added and average to get the final prediction, while when used for classification, the class selected by most tree results becomes our classification.

The Random Forest Ensemble method is a very strong and versatile model to utilize in machine learning. Since one of the model’s core functions is the use of random samples of data each run, this model corrects common habit of standard decision tree models to overfit their data while learning. Not only that but the model is capable of handling large datasets that have a mix of data types, missing data, or that are even unbalanced.

It’s from this flexibility and speed that you will see this model being used in almost every field. Do you need spam detection in email? How about identifying objects and people for self-driving cars? Human body position recognition for video games? Targeted advertising based on a multitude of factors? Random forest is typically the go to starter model for every use case.

Unfortunately, it does have it’s drawbacks and limitations. Although the Random Forest model helps eliminate the overfitting issue of a standard decision tree, this only works if the data set does not contain a high degree of noise. Furthermore, as the data relationship becomes more complex or the dataset becomes larger, the speed and computational expense needed to run can rise quickly.

**\*\*References:\*\***:

<https://en.wikipedia.org/wiki/Ensemble_learning>

<https://en.wikipedia.org/wiki/Ensemble_(mathematical_physics)>

<https://en.wikipedia.org/wiki/Random_forest>

<https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/>

<https://blog.devops.dev/a-deep-dive-into-the-random-forest-algorithm-d4a8950507a3>

<https://medium.com/data-science/random-forests-algorithm-explained-with-a-real-life-example-and-some-python-code-affbfa5a942c>

<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>

<https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/>

<https://www.geeksforgeeks.org/random-forest-regression-in-python/>