RANDOM FOREST EXPERIMENT REPORT

GANESHREDDY ANNAPAREDDY

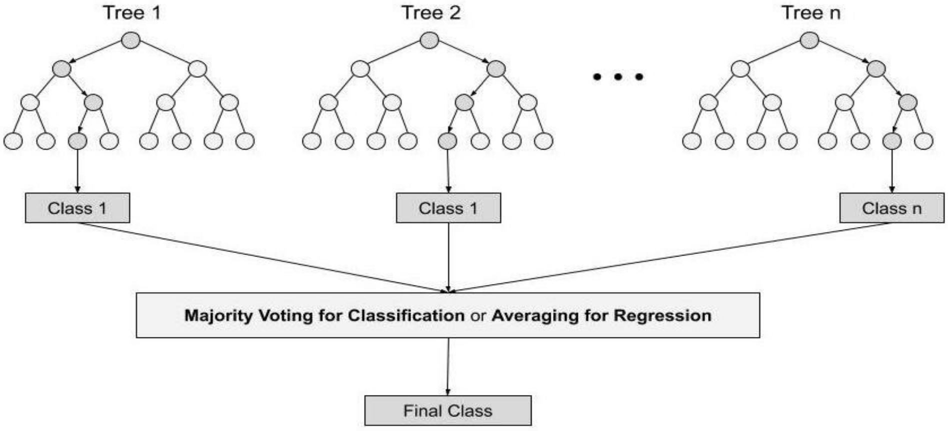
50442295

ganeshre@buffalo.edu

1 RANDOM FOREST

For classification and regression, supervised ensemble learning models called "Random Forests" are utilized. Classification is a key component of machine learning; we want to identify the class to which an observation belongs. For many commercial applications, like as predicting whether a certain user will purchase a product or predicting whether a specific loan would fail or not, the ability to correctly categorize observations is quite significant. In fact there are a lot of other algorithm's like support vector machine, naïve bayes classifier, logistic regression and decision trees to name some of them. But the random forest classifier is on top of the classifier hierarchy among all the classifiers in them.

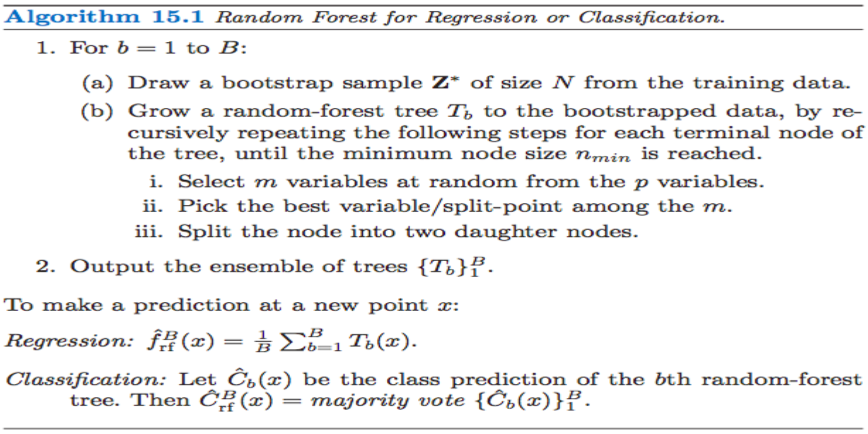
“”” To provide more precise and dependable predictions, Random Forests combine many decision trees “””


Tree 1
Tree 2
Tree n
Class 1
Class 1
Class n
Majority Voting for Classification or Averaging for Regression
Final Class

“The above given figure shows us an example on how the random forest works”

Multiple machine learning models are aggregated using ensemble learning models for improving performance overall. This is justified by the fact that each model utilized is poor when used independently but powerful when used as part of an ensemble. The "strong" ensemble is produced by aggregating the outputs of a large number of decision trees, which serve as the "weak" elements in that example of random forest. The random forest algorithm consists of two steps: creating the random forest and using the classifier produced in the first stage to generate a prediction. The random forest method differs from the decision tree algorithm in that the process of locating the root node and dividing the feature nodes occurs at random.

A low correlation between models or trees is essential for the random forest algorithm. The ensemble forecasts made by the uncorrelated models can be more precise than any individual prediction. In this instance, each tree is shielded from the specific sources of mistake by the others. Simply said, many trees will be right while other trees may be incorrect. The performance of a bigger group of interconnected uncorrelated trees will be superior to that of any single model. The bagging approach is used to train the random forest model, which is composed of a group of decision trees. The bagging technique consists only of combining many models to improve the end outcome. A random sample is chosen from the dataset through bagging. Row sampling is the process of creating each model from the sample. Bootstrap is the term for the row sample stage in row sampling. Aggregation is the process of combining all the results and producing a result based on a majority vote. The following illustration shows how the Random Forest algorithm works:



```
Algorithm 15.1 Random Forest for Regression or Classification.
  1. For $b = 1$ to $B$:
       (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size $N$ from the training data.
       (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by re-
           cursively repeating the following steps for each terminal node of
           the tree, until the minimum node size $n_{min}$ is reached.
             i. Select $m$ variables at random from the $p$ variables.
            ii. Pick the best variable/split-point among the $m$.
           iii. Split the node into two daughter nodes.
  2. Output the ensemble of trees $\{T_b\}_1^B$.
To make a prediction at a new point $x$:
Regression: $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.
Classification: Let $\hat{C}_b(x)$ be the class prediction of the $b$th random-forest
      tree. Then $\hat{C}_{\mathrm{rf}}^B(x) = majority\ vote\ \{\hat{C}_b(x)\}_1^B$.
```

We employ a decision tree as part of the random forest technique, therefore choosing a root node is crucial in order to construct the tree. Therefore, we choose the feature that will serve as the root node using the Gini Index. How clean or impure the split will be is determined by the Gini Index. In order to compute it, use the equation:

$$\text{GINI INDEX} = 1 - {}_{i=1}\sum^{n} (P_i)^2 = 1-[(P_+)^2 + (P_-)^2]$$

P+ stands for the likelihood of the positive class, and P- for the likelihood of the negative class. The program experiments with all potential split combinations and selects the feature as the root node that will result in the lowest Gini Index (Low Impurity). Entropy can be used to assess impurities, however ensemble models prefer the Gini index since it is computationally efficient and executes more quickly because there isn't a logarithmic term present like there is in entropy.

$$E(S) = {}_{i=1}\sum^{C} - p_i \log_2 p_i$$

The decision tree's disadvantage is that it frequently overfits the training set of data. Each choice tree varies somewhat from the others, just like the random forest. While some people may overfit, others may do well. By averaging the outcomes, the algorithm will thereby decrease overfitting. To address this high variance problem, we combine many decision trees in a random forest, which allows us to minimize the variance and overcome overfitting.

2 CREATION

- Sample N examples at random, but with replacement from the original data, if the training set has N cases. This sample will serve as the tree's training set.
- If there are M input variables, a number is supplied so that m variables are randomly chosen from the M at each node, and the node with the best split on this m is split.

3 PREDICTION

- Takes the test characteristics, creates a decision tree at random using its rules, and then stores the projected outcome (target).
- Determine how many votes each predicted target received.
- Think of the random forest algorithm's final prediction as being the most voted predicted target.

4 BENEFITS

- Provides flexibility, easy to determine feature importance and Reduce risk of over lifting.

5 CHALLENGES

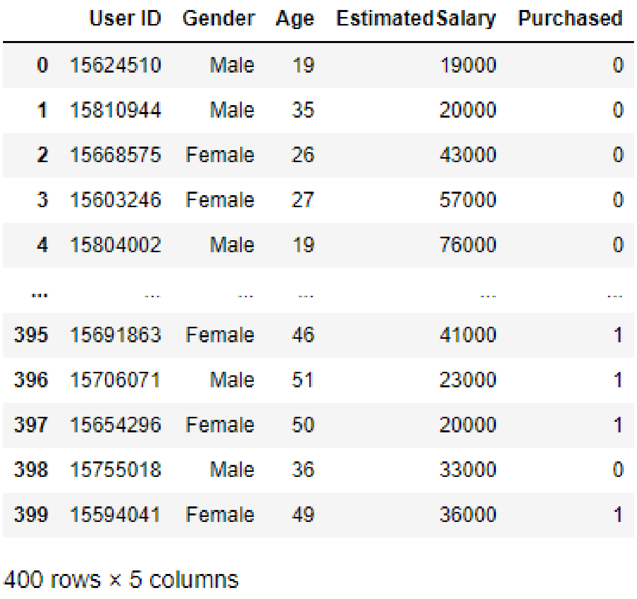
- More complex, requires more resources and is time consuming process.

EXPERIMENT :

Dataset :

For the random forest experiment I have used the social network ads from Kaggle. It has the data which consists of users ID, their gender, age and an estimated salary along with them purchasing or not. The dataset has 400The columns and 5 rows and the description goes as

Column	Description
User ID	The user id of the customer
Gender	It tells us if they are male or a female
Age	The age of the user
Estimated Salary	The estimated salary of the user
Purchased	1 if purchased or 0 if not purchased



	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

Training :

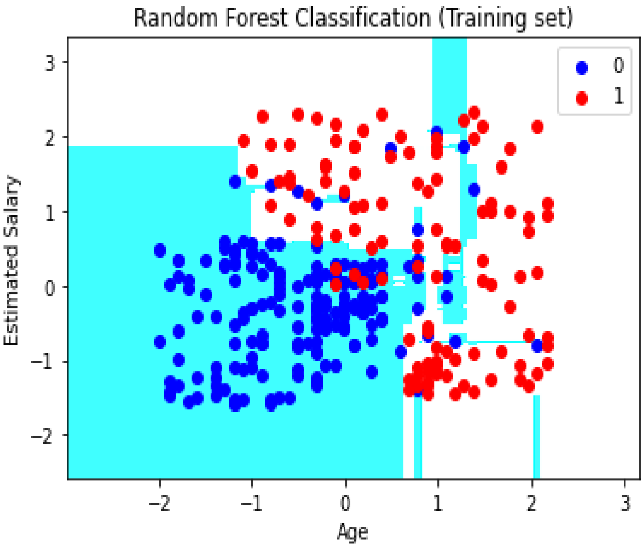
I used the dataset social network ads and see the evaluation result. Python was used for this task on the Jupiter notebook. I have used NumPy, matplotlib, pandas and sk-learns libraries for the experiment.

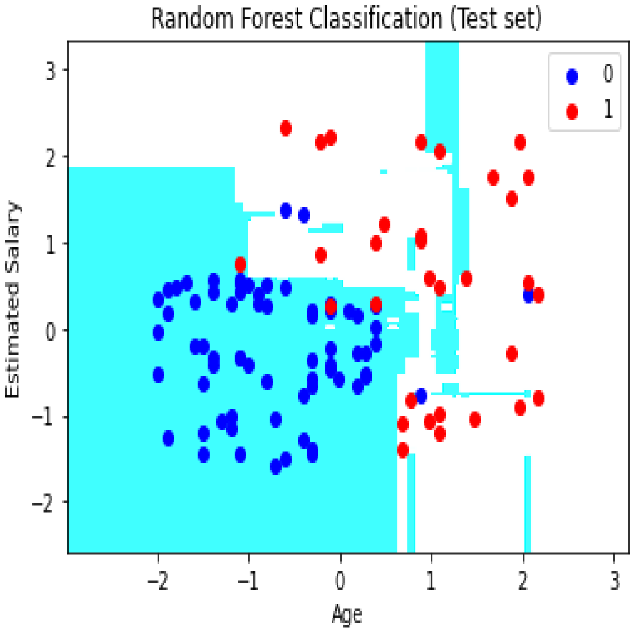
For importing the training model I have used sklearn.model_selection and then used scikit learns for the standard scalar function where I am training and testing. I am importing the random forest classifier model from sklearn.svm. I have imported confusion matrix for predicting the classifier for the y prediction. And then I have plotted the graphs for both the training and testing dataset .

RESULTS :

Training set:

In the training set I have compared the estimated salary against their age to see if the user had made any purchase according to the dataset. Here there are all the values from the dataset. We can see that the 0 are in blue dots spread across the graph which means the user haven't made any purchase and the 1 are in the red dots which means they have made a purchase. Here the most number of purchases are made if the user has more salary and age compared to the users who has less salary and age compared to rest of the users.





Testing set :

In the testing set we can see the number of points have decreased. We can see that the 0 are in blue dots spread across the graph which means the user haven't made any purchase. While the 1 are in the red dots spread across the graph which means they have made a purchase. The number of purchased have reduced than that of those who didn't make a purchase. From this we can see that the more age and the more salary, the more the chances of making a purchase.