Ensemble Learning

-Rahmeen Habib

Intuition

a. Guess the number of candies in a jar (Law of large numbers)
b. Three Amigos

Bias - Oversimplifying Data

1. Measure of how flexible the model is
2. Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.
3. Since it oversimplifies the model, it leads to high error on training and test data.
4. Neural Nets, decision trees are low bias/unstable models whereas linear model is high bias model

Variance - Overexhausting Data

1. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before.
2. As a result, such models perform very well on training data but has high error rates on test data.

GOAL - To achieve *Low Bias* and *Low Variance*

Enter Ensembling!

Ensemble

Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem.

Base Learners

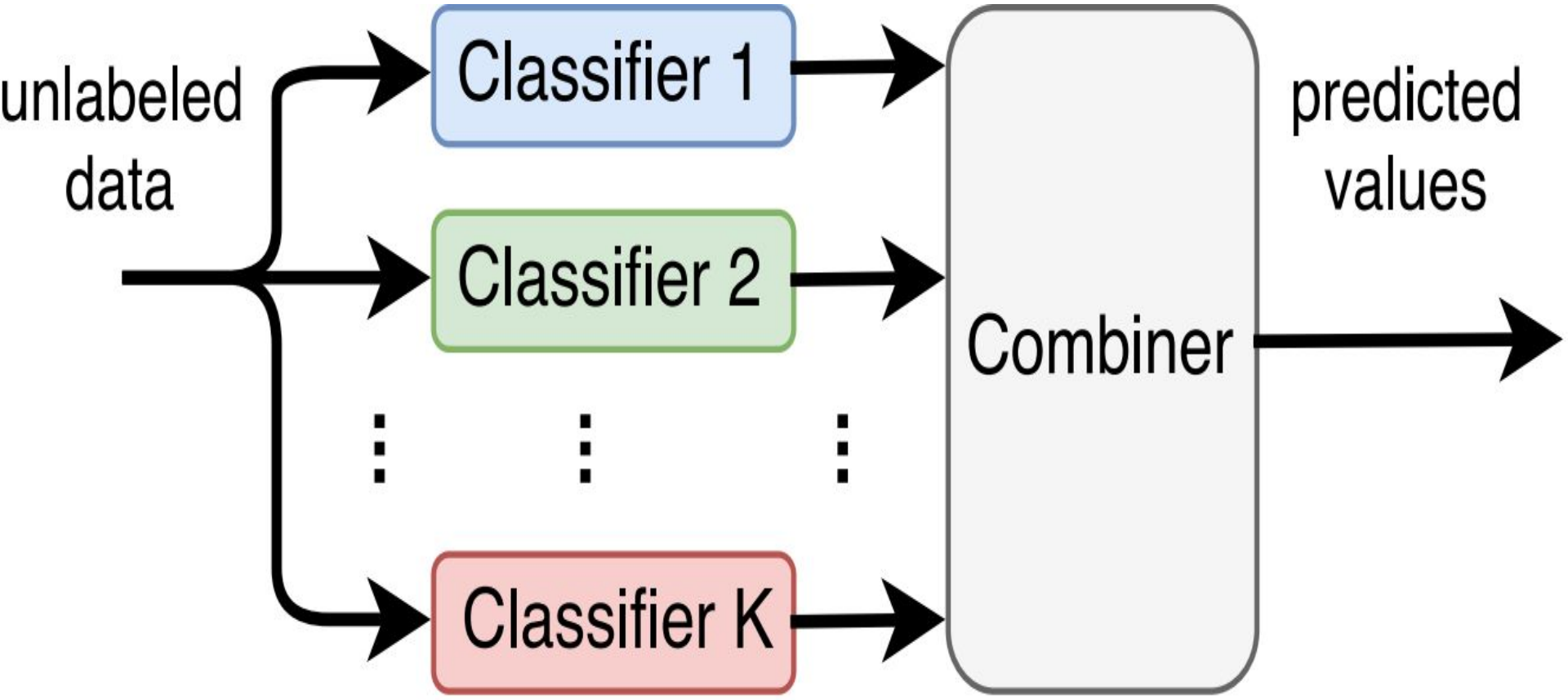
1. Group of Learners
2. Generated in multiple ways
    a. Different Algos
    b. Different Parameters/Hyperparameters (no. of nodes, hidden layers, bias values)
    c. Different Training Sets (Weak learning models with high variance can be fed subsets of data.)
    d. Different Representations (Decision trees splitting on attributes based on Split Info, Gini Index)

Conditions for ensembling models

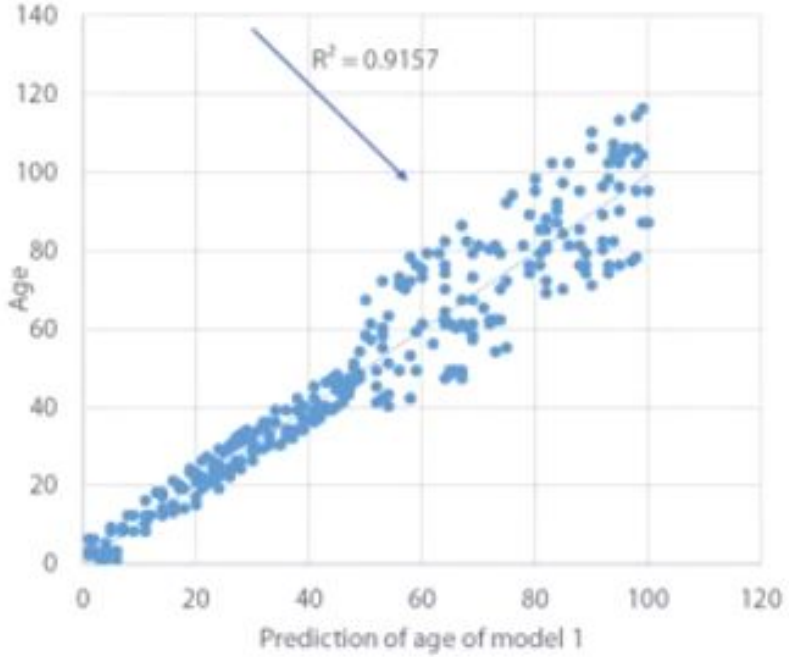
1. Diversity
2. Individual Accuracy (Base Learners cannot be absolutely random, error>0.5)
3. Independent Base Learners that generate different kinds of errors and work well on different kinds of data

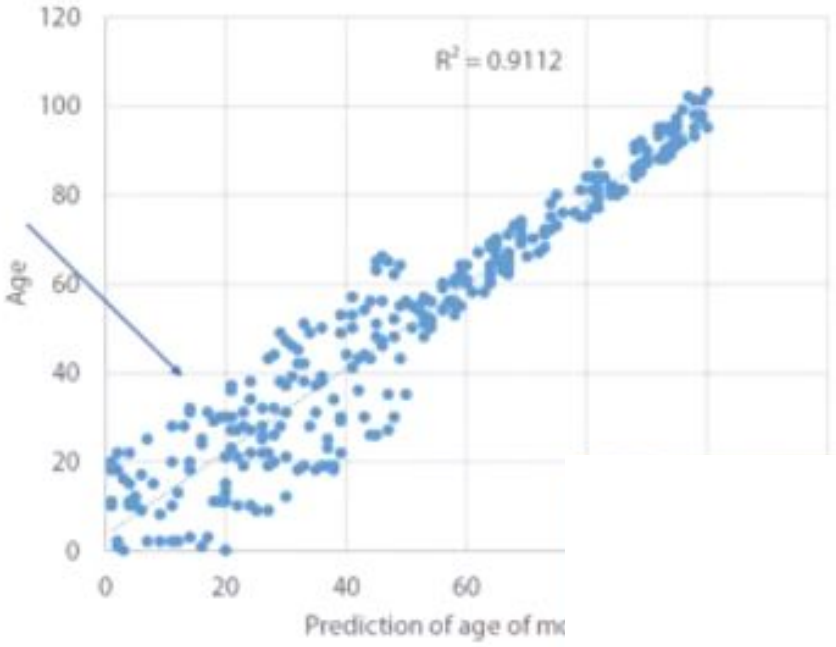
Goal of ensembling models

1. Honoring Diversity
2. Maximising Accuracy
3. Reducing Bias and Variance Errors


unlabeled
data
Classifier 1
Classifier 2
Classifier K
Combiner
predicted
values

An simple example of Ensembling two models

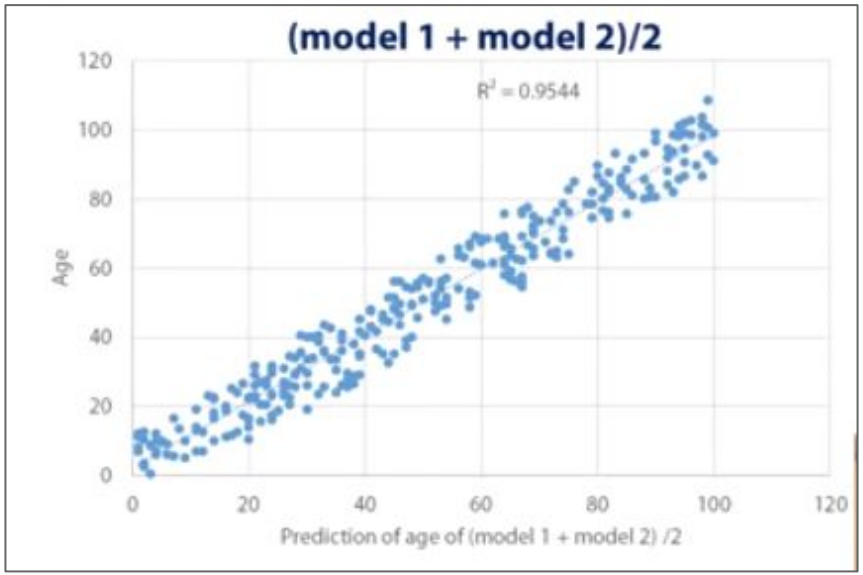




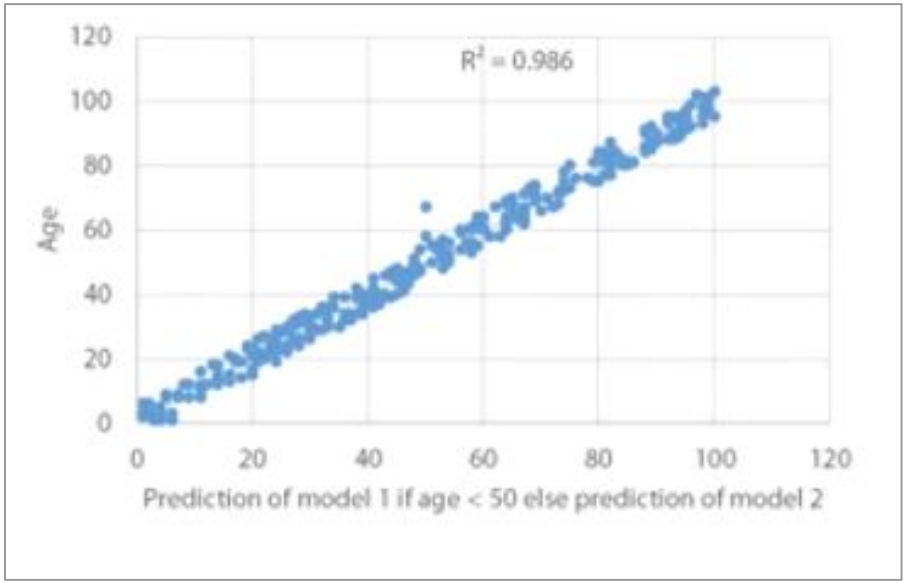
Questions Alert!

1. How are these Models Different? Spot any differences that you can see.
2. What is the performance evaluation metric used here?
3. Can you use ensembling to improve on performance?
4. How?

Blending



Conditional Averaging



Ways of Combining/Ensembling

1. Averaging
2. Weighted Averaging (based on accuracy, variance)
3. Conditional Averaging
4. Bagging
5. Boosting
6. Stacking

What is Bagging?

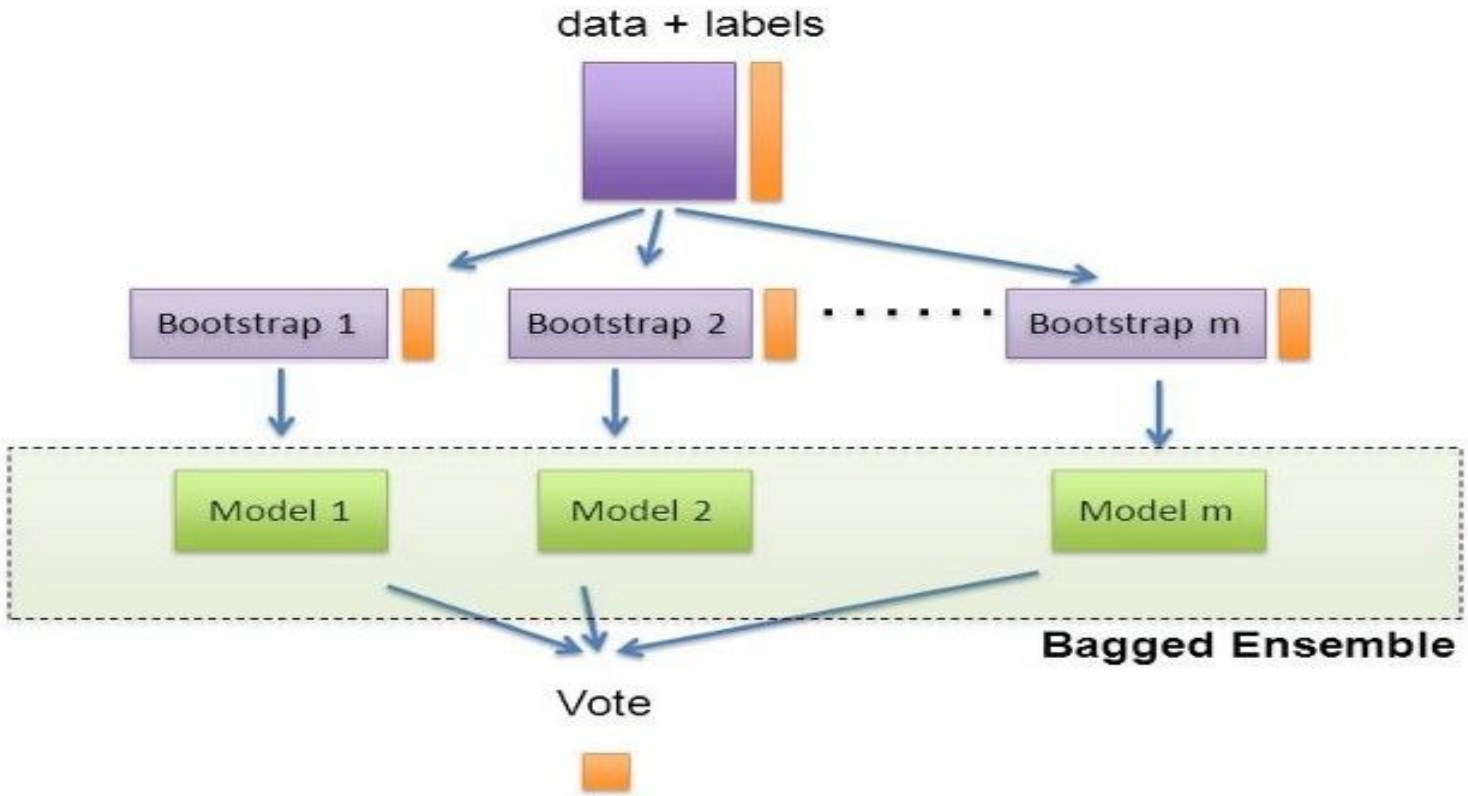
Averaging slightly different versions of the same model to improve accuracy

Bootstrap + Aggregation

Random sampling as opposed to Disjoint partitioning of Data, Sampling with replacement

Averaging all the results

“Bagging” : **B**ootstrap **AGG**regat**ING**



Parameters that Control Bagging

- Row (Sub) sampling or Bootstrapping
- Shuffling
- Column (Sub) sampling
- Model-specific parameters
- Number of models (or bags)
- (Optionally) parallelism

Let's Code!

BaggingClassifier and **BaggingRegressor** from Sklearn

```
# train is the training data
# test is the test data
# y is the target variable
model=RandomForestRegressor()
bags=10
seed=1
# create array object to hold bagged predictions
bagged_prediction=np.zeros(test.shape[0])
#loop for as many times as we want bags
for n in range (0, bags):
    model.set_params(random_state=seed + n)# update seed
    model.fit(train,y) # fit model
    preds=model.predict(test) # predict on test data
    bagged_prediction+=preds # add predictions to bagged predi
#take average of predictions
bagged_prediction/= bags
```

What is Boosting?

Boosting is an ensemble method for improving the model predictions of any given learning algorithm. The idea of boosting is to train weak learners sequentially, each trying to correct its predecessor.

- Weight based boosting
- Residual boosting

Weight Boosting

Rownum	x0	x1	x2	x3	y	pred	abs.error	weight
0	0.94	0.27	0.80	0.34	1	0.80	0.20	1.20
1	0.84	0.79	0.89	0.05	1	0.75	0.25	1.25
2	0.83	0.11	0.23	0.42	1	0.65	0.35	1.35
3	0.74	0.26	0.03	0.41	0	0.40	0.40	1.40
4	0.08	0.29	0.76	0.37	0	0.55	0.55	1.55
5	0.71	0.76	0.43	0.95	1	0.34	0.66	1.66
6	0.08	0.72	0.97	0.04	0	0.02	0.02	1.02

Residual Boosting

Rownum	x0	x1	x2	x3	y	pred	error
0	0.94	0.27	0.80	0.34	1	0.80	0.20
1	0.84	0.79	0.89	0.05	1	0.75	0.25
2	0.83	0.11	0.23	0.42	1	0.65	0.35
3	0.74	0.26	0.03	0.41	0	0.40	-0.40
4	0.08	0.29	0.76	0.37	0	0.55	-0.55
5	0.71	0.76	0.43	0.95	1	0.34	0.66
6	0.08	0.72	0.97	0.04	0	0.02	-0.02

Rownum	x0	x1	x2	x3	y	new pred	old pred
0	0.94	0.27	0.80	0.34	0.2	0.15	0.80
1	0.84	0.79	0.89	0.05	0.25	0.20	0.75
2	0.83	0.11	0.23	0.42	0.35	0.40	0.65
3	0.74	0.26	0.03	0.41	-0.4	-0.30	0.40
4	0.08	0.29	0.76	0.37	-0.55	-0.20	0.55
5	0.71	0.76	0.43	0.95	0.66	0.24	0.34
6	0.08	0.72	0.97	0.04	-0.02	-0.01	0.02

Final Prediction : old_pred + eta*new_pred

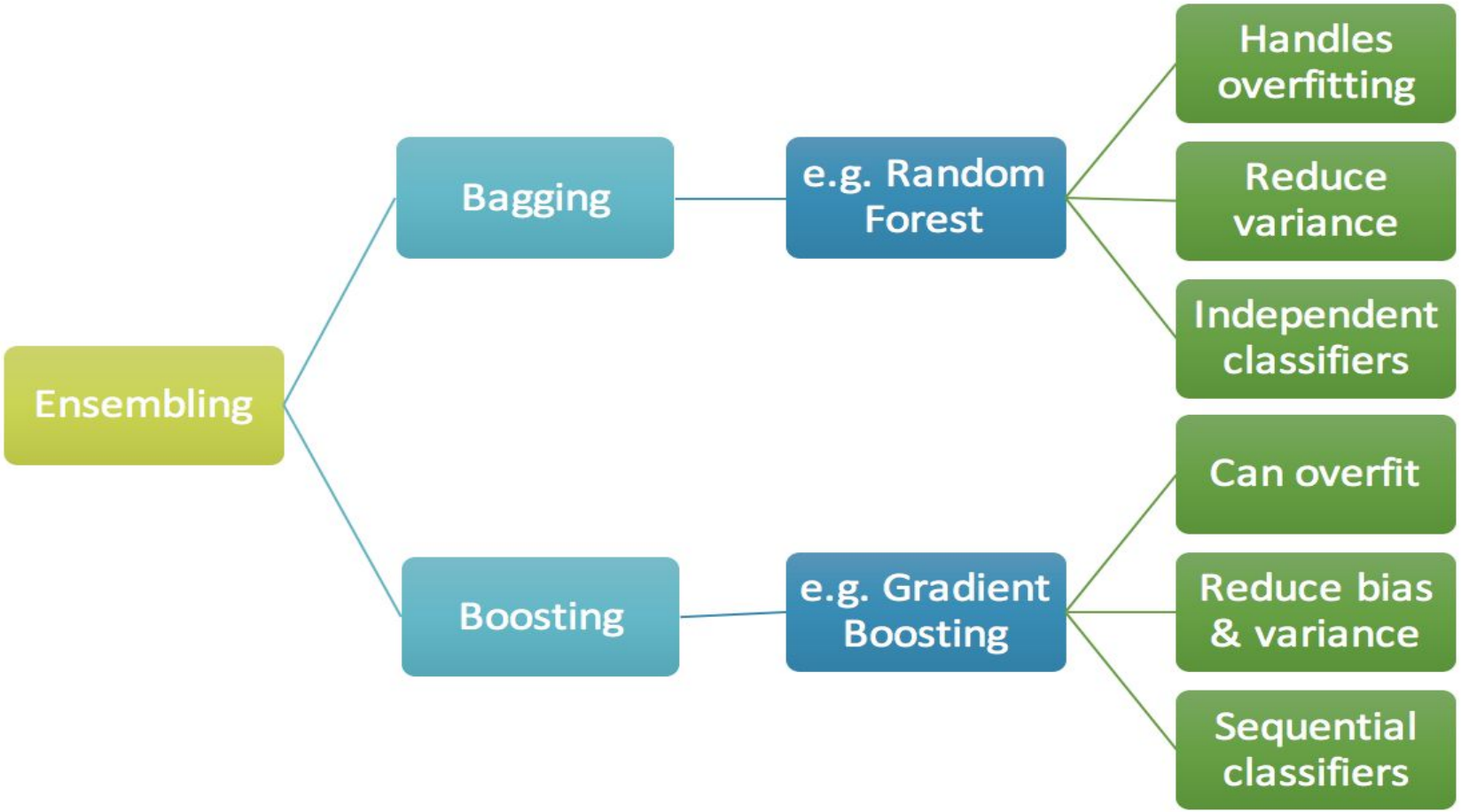
↠ 0.75 + 1*0.20 = 0.95 (pretty close to the actual value = 1)

Boosting Parameters

1. Learning rate
2. Number of estimators
3. Subsampling

Experiment with Boosting!

- AdaBoost
- XgBoost
- Sklearn’s GBM


Ensembling
Bagging
e.g. Random Forest
Handles overfitting
Reduce variance
Independent classifiers
Boosting
e.g. Gradient Boosting
Can overfit
Reduce bias & variance
Sequential classifiers

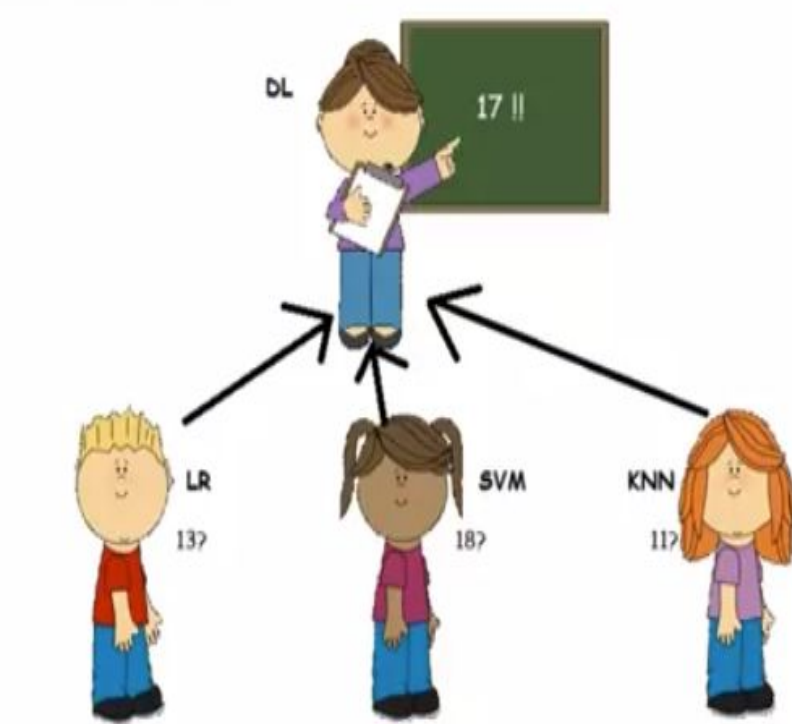
Stacking - Another popular Ensembling model

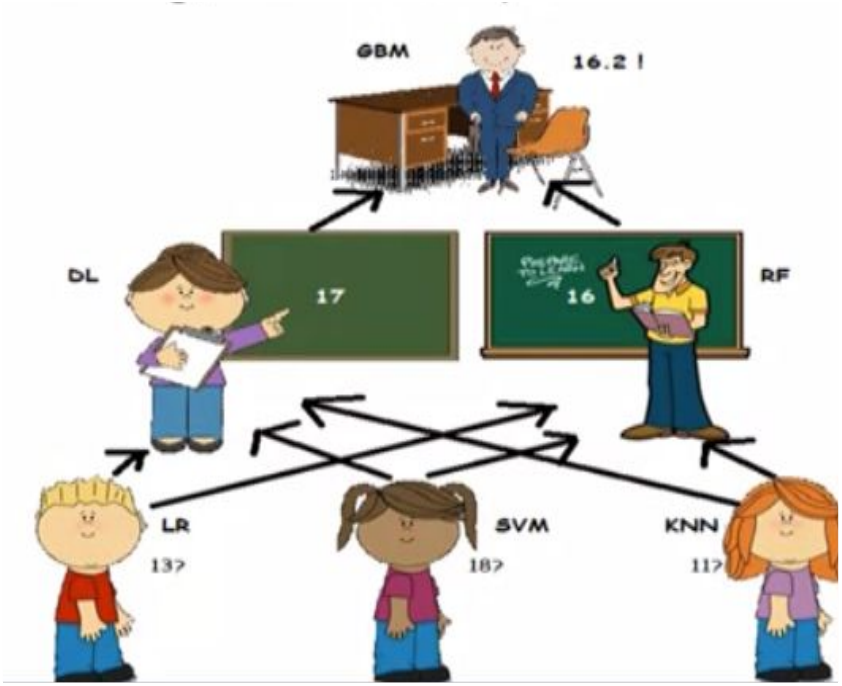
Stacking is a way to ensemble multiple classification or regression models.

Stacking is a different paradigm however.

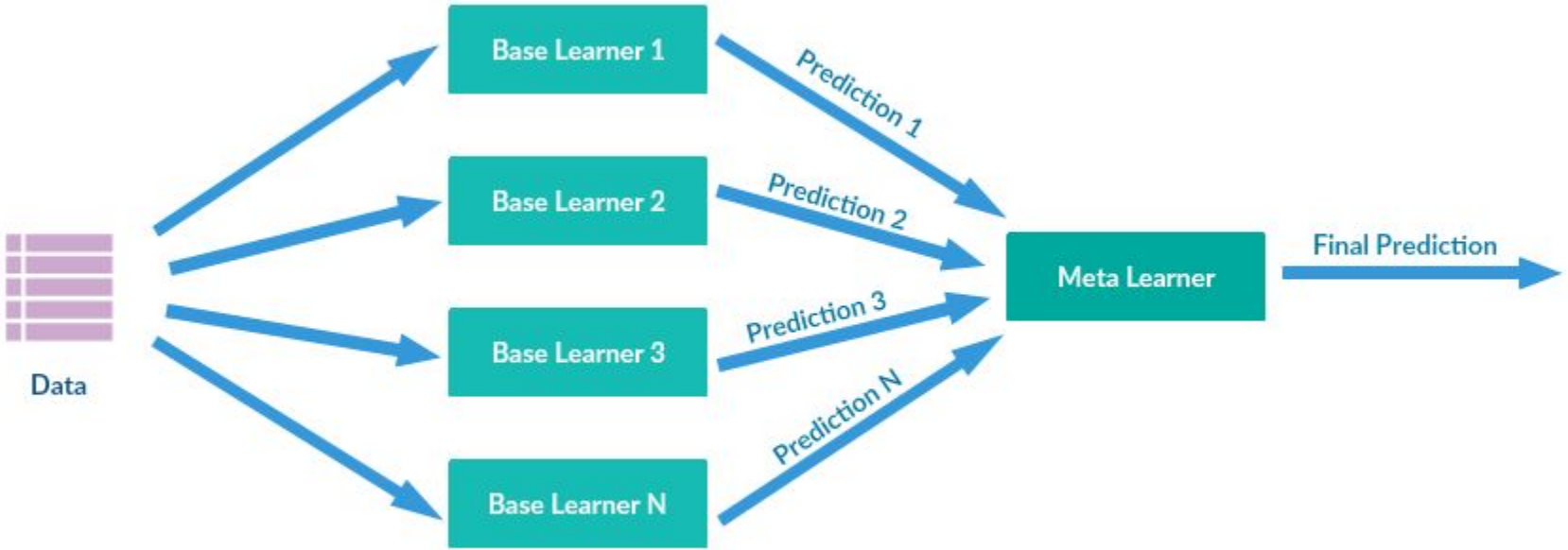
1. The point of stacking is to explore a space of different models for the same problem.
2. The idea is that you can attack a learning problem with different types of models which are capable to learn some part of the problem, but not the whole space of the problem.
3. So you can build multiple different learners and you use them to build an intermediate prediction, one prediction for each learned model.
4. Then you add a new model which learns from the intermediate predictions the same target.
5. This final model is said to be stacked on the top of the others, hence the name.
6. Thus you might improve your overall performance, and often you end up with a model which is better than any individual intermediate model.

An Illustrative Example





A one layer Stacking Model



Formalising the method of Stacking

- Wolpert in 1992 introduced stacking. It involves:

1. **Splitting** the train set into two disjoint sets.
2. **Train** several base learners on the first part.
3. **Make predictions** with the base learners on the second (validation) part.

A numerical example of Stacking

A

X0	x1	x2	xn	y
0.17	0.25	0.93	0.79	1
0.35	0.61	0.93	0.57	0
0.44	0.59	0.56	0.46	0
0.37	0.43	0.74	0.28	1
0.96	0.07	0.57	0.01	1

B

X0	x1	x2	xn	y
0.89	0.72	0.50	0.66	0
0.58	0.71	0.92	0.27	1
0.10	0.35	0.27	0.37	0
0.47	0.68	0.30	0.98	0
0.39	0.53	0.59	0.18	1

C

X0	x1	x2	xn	y
0.29	0.77	0.05	0.09	?
0.38	0.66	0.42	0.91	?
0.72	0.66	0.92	0.11	?
0.70	0.37	0.91	0.17	?
0.59	0.98	0.93	0.65	?

Train algorithm **0** on A and make predictions for B and C and save to **B1**, **C1**
Train algorithm **1** on A and make predictions for B and C and save to **B1**, **C1**
Train algorithm **2** on A and make predictions for B and C and save to **B1**, **C1**

B1

pred0	pred1	pred2	y
0.24	0.72	0.70	0
0.95	0.25	0.22	1
0.64	0.80	0.96	0
0.89	0.58	0.52	0
0.11	0.20	0.93	1

C1

pred0	pred1	pred2	y	Preds3
0.50	0.50	0.39	?	0.45
0.62	0.59	0.46	?	0.23
0.22	0.31	0.54	?	0.99
0.90	0.47	0.09	?	0.34
0.20	0.09	0.61	?	0.05

Train algorithm 3 on B1 to obtain pred3

Why is stacking so useful?

1. Because it brings a lot of diversity
2. Weak learners also bring in a lot of information

Let’s Code!

Recommended Reading

1. https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions/
2. https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de
3. https://www.coursera.org/lecture/competitive-data-science/ensembling-tips-and-tricks-XqLc1
4. https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

Thank You!

feel free to reach out at rahmeenwill99@gmail.com