ITI104 Assignment (19 Aug 2023)

Magdalene Cheong

23A467J

Task: Study and explore the Voting vs Stacking ensemble methods

Introduction To Ensemble Methods:

As the meaning of ensemble implies - a group of items viewed as a group rather than individually – the ensemble methods engage more than 1 base predictor. The number of base predictors is specified in the model. The base predictors used in the model can be of the same machine learning algorithm, as in parallel homogeneous ensembles and sequential boosting ensembles. Or they can be different, as in parallel heterogeneous ensembles. The results of predictors are then aggregated to return the prediction. Therefore, in essence, ensembles are made up of individual base predictors (base estimator/base model) with the objective of achieving better performance (prediction) than each of the base predictor. There are three types of ensembles:

- Parallel homogeneous ensembles
    - Also known as bagging.
    - Example: random forest, pasting
    - This ensemble uses the same base model, each trained on a different random subset of the dataset of the same size. Each base predictor is totally independent of the other base models and are not optimised predictors. It suffices that each of them performs slightly better than random probability. The outcomes of the base models are then aggregated to produce the final performance indicators by hard voting or soft voting.
- Sequential boosting ensembles
    - Also known simply as boosting.
    - Example: AdaBoost, gradient boosting, XGBoost
    - This ensemble also uses the same base predictor. Each base predictor is, however, run sequentially on the same dataset. Like bagging, each base predictor is a weak learner. The difference between them is that succeeding base predictor tries to correct the errors made by the preceding one by increasing the weights of those wrongly classified training instances and reducing those correctly classified. More emphasis is thus placed on the wrongly classified instances when the succeeding base predictor is run. Therefore, the weights keep changing over each boosting run until it has completed all the specified number of iterations.
- Parallel heterogeneous ensembles
    - Also known as stacking.
    - This ensemble method uses multiple machine learning models so it could be a combination of decision, SVM, logistic regression etc. It makes use of the strengths of individual model to produce better performance. Stacking method combines the predictions from these models. All the base models run on the same dataset.
    - The architecture of the stacking model includes 2 or more models:

    - L-0 Models (Base models). At this level, the base models are complex and diverse. A diverse range of models is often encouraged as each model make different assumptions and have different strengths. Thus, by putting them together, the ensemble seeks to leverage from the strengths of each model to arrive at a better prediction. These models perform well on its own, unlike in bagging and boosting. Their predictions are then compiled.
    - L-1 Model (Meta model) is the model that learns how to combine the predictions from models in L-0 in the best possible way.
- There are 2 approaches for L-1 Model
    - Voting ensemble
        - This is further broken down into Regression Voting Ensemble and Classification Voting Ensemble.
        - In Classification Voting Ensemble, the model combines the predictions from multiple models and returns the class with the highest number of votes (Hard Voting) or the class with the highest summed probability (Soft Voting).
        - In Regression Voting Ensemble, prediction is made by averaging the predictions from the models in L-0.
    - Stacking ensemble (Stacked Generalization)
        - Unlike the complex base models, the meta model (L-1) is typically simple. It is designed to interpret the predictions from the base model. Therefore, linear models are often used (linear regression and logistic regression.
        - Another method is to assign weights to different base model base on its importance and the prediction will therefore be a weighted average. This process is also known as Weighted Average Ensemble or blending.

How Voting and Stacking Ensemble Methods Work:

Classification Voting Ensemble:

A voting ensemble works by combining the predictions from L-0 models. The prediction for each label is added up and the label with the highest majority vote is the prediction. As mentioned above, hard voting sums up the total number of votes and return the class with the highest number of votes as the prediction. Soft voting, on the other hand, averages the probability across the classifiers and the class with the highest average value is the prediction. The ensemble assumes equal weightage for all base models.

To illustrate hard voting with a classifier voting ensemble with 3 classifier and 2 classes, because Class 2 'wins' the majority voting, Class 2 is the predicted outcome of the voting ensemble:

L-0 Model	P(Class 1)	P(Class 2)
Classifier 1	0.3	**0.7**
Classifier 2	0.4	**0.6**
Classifier 3	**0.9**	0.1
Number of votes (mode)	1	**2**

To illustrate soft voting with the same setup, Class 1 is the prediction based on highest average of all the probabilities:

L-0 Model	P(Class 1)	P(Class 2)
Classifier 1	0.3	0.7
Classifier 2	0.4	0.6
Classifier 3	0.9	0.1
Average probability	**0.53**	0.47

Therefore, generally hard voting is used when the objective is to predict class label. If class probabilities are the desired output, then soft voting should be used instead.

It is, however, possible that the voting ensemble is not able to make a prediction. This is illustrated with the following example:

L-0 Model	P(Class 1)	P(Class 2)
Classifier 1	0.5	0.5
Classifier 2	0.4	**0.6**
Classifier 3	**0.9**	0.1
Number of votes (mode)	**1**	**1**

Therefore, a voting ensemble is best suited in situations where the base models perform comparatively well. If half of the base models do not perform well, then the voting ensemble would not be able to perform well. If a base model has better prediction than the voting ensemble and that model should be used instead.

The rule of thumb would thus be to use voting ensembles when:

- All the base models have comparatively good performance.
- All the base models mostly have similar predictions.

Regression Voting Ensemble:

In regression voting ensemble, it takes the average of the predictions from the base model. Each model is given equal weightage. The method is illustrated below:

L-0 Model	Prediction
Regressor 1	79.8
Regressor 2	75.4
Regressor 3	83.4
Average of the predictions	79.53

The average of the predictions (79.53) will thus be the prediction of the regression voting ensemble.

The weights for the models can be adjusted higher for the stronger base model(s) to for a better prediction. This is also known as the Weighted Average Ensemble.

Weighted Average Ensemble:

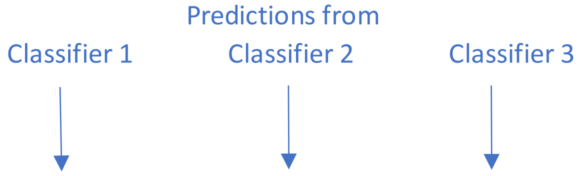
This ensemble is an extension of the voting ensemble. It gives more weights to base models whose strengths we want to leverage from. In this ensemble, each base model is assigned a weight coefficient. The weight coefficient could be a value ranging from 0 to 1 which represents the percentage of the weight. The higher the value, the higher the weight. It could also be the number of votes given to each base model, starting with the value of 1. This is illustrated in the table below:

L-0 Model	Weight	Prediction	Weight x Prediction	Weighted Prediction
Regressor 1	0.90	79.8	0.90 * 79.8	71.82
Regressor 2	0.85	75.4	0.85 * 75.4	64.09
Regressor 3	0.95	83.4	0.95 * 83.4	79.23
Prediction (sum(Weighted Prediction)/Total Weights)				79.68

The same method of calculation applies to Classification Voting Ensemble using soft voting. The values under Prediction will then be the probabilities predicted by each base model.

Stacking Ensemble:

The stacking ensemble takes in predictions of the base models and create a new dataset. These are known as meta-features. Meta-features can be predictions or probabilities of the predictions. Take for example, if there are 3 classifiers as L-0 base models, Classifier 1, Classifier 2 and Classifier 3, the dataset created would look like this:


Predictions from

Classifier 1 Classifier 2 Classifier 3

	Meta-feature 1	Meta-feature 2	Meta-feature 3
x_1	0.90	0.83	0.92
x_2	0.88	0.85	0.91
x_3	0.90	0.81	0.94
…	…	…	…

This new dataset is used as the training set. A simple machine learning model is often used, like linear regression or logistic regression. This is known as linear stacking. More complex and powerful non-linear models can also be used (e.g. SVM) but at the expense of interpretability and computational efficiency that linear models have. In summary, the Stacking Ensemble (Meta Model) makes its own prediction with the outputs from the base models as its inputs.

Advantages and Benefits of Voting vs Stacking Methods:

Voting Methods	**Stacking Methods**
Simpler: Voting Methods are straight forward, simpler, easier to understand, interpret and implement.	More complex but flexible: Stacking Methods are more complex because it allows selection from a range of machine learning algorithm to combine the predictions. The way the predictions from the base models is combined is thus more complex and less obvious as compared to straightforward voting and averaging. Training a machine learning algorithm on these predictions can capture complex patterns and interactions between the models better and make better prediction. Stacking is therefore flexible in that one can select a machine learning model that is best suited for the data. As mentioned under Hard Voting Method, it is possible that all classed have the same number of votes. This situation is very rare but possible. In such cases, Stacking Method would be helpful as a more complex understanding of the data via the meta model would help to give a more accurate prediction.
Efficient: As a simpler method, it is computational efficient because it does not require training as Stacking Method does	Less efficient but potential for better performance: The additional training in Stacking Method allows room for optimisation, therefore it has a higher chance to achieve better performance.
Better variance: By averaging out the predictions, variance is reduced – a consequence of averaging. Therefore as ensemble size increases, variance reduces.	Better bias: By training a dataset which is a collection of the predictions from different base models, it is possible to get a better bias than that of each of the base models.
Diverse: If Voting Method is used on a range of models of diverse strengths and weaknesses where their predictions vary accordingly, the outcome of the Voting Methods could be more generalised and robust.	Extensible: Stacking ensemble can be part of a larger ensemble, combining multiple tiers for deeper gains on information.

Advantages and Benefits of Ensemble Methods Over Single Classifier/Regressor:

1) Ensembles achieve higher accuracy as it combines the strengths of individuals models and overcome the assumptions made by each model. It 'smooths' out mistakes made by individual base model. In other words, it is more robust to mistakes and therefore generalises better than a single base model.
2) Variance is reduced by virtue of averaging. As also mentioned in the above point, ensembles even out the mistakes of other base models hence variance is reduced. Therefore, the predictions can be more consistent and stable. In same vein, ensembles could also overcome effects overfitting on the base models.
3) Ensembles are more robust to noise and outliers. In stacking ensemble for example, the dataset the meta model uses is based on the base model predictions where there would be no outliers. Any effect of outliers or noise that the base models experience are not present in the metadata therefore output from the meta model will be consistent and has less variance. The effects of noise and outlier on the base models are also evened out by the relative strengths of different base models.
4) Generalisation is better as ensembles captures a broader range of patterns and makes more balanced predictions. In voting ensembles, the final prediction takes into account predictions from different models (which make different assumptions and have different strengths) and therefore averaging their predictions consolidates different perspectives of the data. While in stacking ensemble, the meta model selected could be non-linear and therefore might be able to uncover underlying complex patterns in the data.
5) Because different base models are used, ensembles better capture complex underlying pattern of the data as each of the base models has different strengths hence the bias could be lower.
6) Ensembles make use of weak learners to correct wrong predictions made by its predecessor to return a better prediction. This is more superior than a single model as it allows the error to be 'rediscovered' and corrected for a better prediction.
7) By virtue that it engages different base models, ensembles allow wider exploration of the model space, combining models with different hyperparameters than on a single model which is confined to a set of hyperparameters. The fit on the data is then likely more comprehensive.

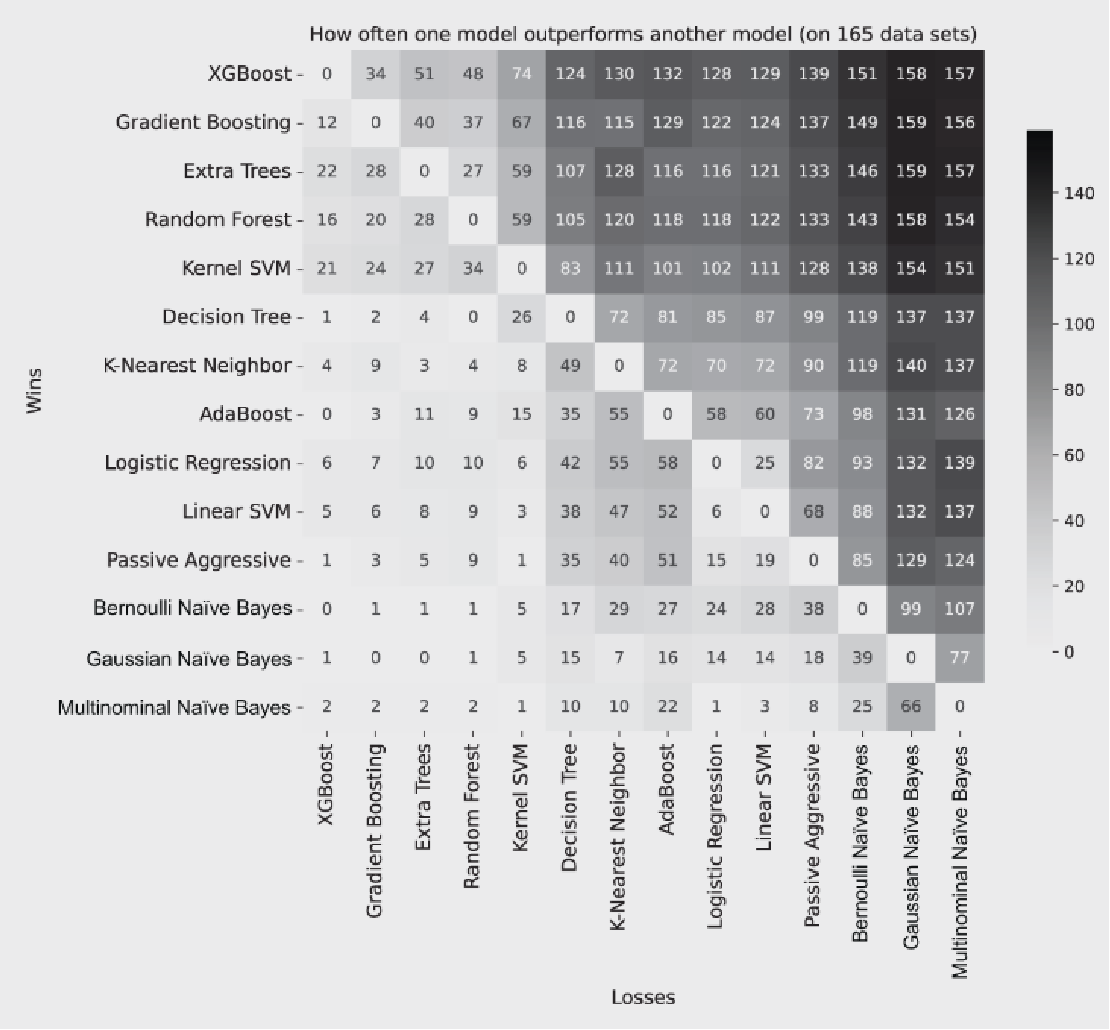
The main considerations if ensemble or single model should be used are:

- Performance - if an ensemble gives better performance/predictions over a single model, it should be used otherwise it does not justify the increased cost involved for additional development and computation time.
- Robustness – if an ensemble is able to reduce the spread of the performance/predictions and is less affected by changes in datasets, it should be used after balancing other cost factors.

Interesting Facts About Voting, Stacking or Other Ensemble Methods:

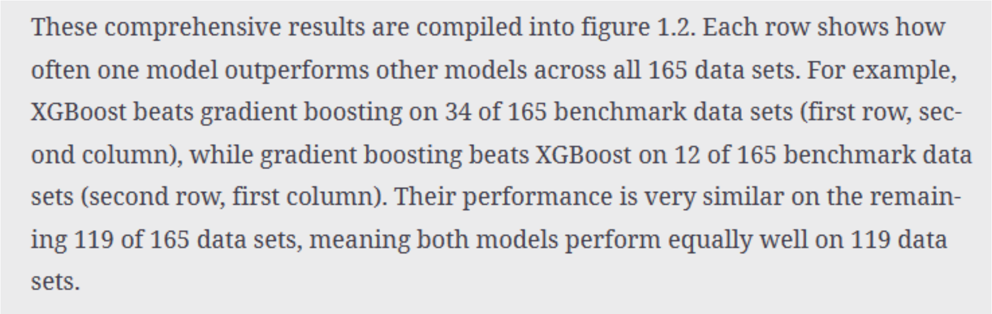
1) Comparison between models

A study was conducted by Olson et al. in 2018. He conducted a comprehensive analysis of 14 popular machine-learning algorithms and their variants on 165 classification datasets. The algorithms in the study include ensemble as well. Two models are considered to have the same performance if their prediction accuracy is within 1% difference. The number of wins and losses are then compiled and plotted in the heatmap below:



	XGBoost	Gradient Boosting	Extra Trees	Random Forest	Kernel SVM	Decision Tree	K-Nearest Neighbor	AdaBoost	Logistic Regression	Linear SVM	Passive Aggressive	Bernoulli Naïve Bayes	Gaussian Naïve Bayes	Multinominal Naïve Bayes
XGBoost	0	34	51	48	74	124	130	132	128	129	139	151	158	157
Gradient Boosting	12	0	40	37	67	116	115	129	122	124	137	149	159	156
Extra Trees	22	28	0	27	59	107	128	116	116	121	133	146	159	157
Random Forest	16	20	28	0	59	105	120	118	118	122	133	143	158	154
Kernel SVM	21	24	27	34	0	83	111	101	102	111	128	138	154	151
Decision Tree	1	2	4	0	26	0	72	81	85	87	99	119	137	137
K-Nearest Neighbor	4	9	3	4	8	49	0	72	70	72	90	119	140	137
AdaBoost	0	3	11	9	15	35	55	0	58	60	73	98	131	126
Logistic Regression	6	7	10	10	6	42	55	58	0	25	82	93	132	139
Linear SVM	5	6	8	9	3	38	47	52	6	0	68	88	132	137
Passive Aggressive	1	3	5	9	1	35	40	51	15	19	0	85	129	124
Bernoulli Naïve Bayes	0	1	1	1	5	17	29	27	24	28	38	0	99	107
Gaussian Naïve Bayes	1	0	0	1	5	15	7	16	14	14	18	39	0	77
Multinominal Naïve Bayes	2	2	2	2	1	10	10	22	1	3	8	25	66	0

Explanation from the source:



These comprehensive results are compiled into figure 1.2. Each row shows how often one model outperforms other models across all 165 data sets. For example, XGBoost beats gradient boosting on 34 of 165 benchmark data sets (first row, second column), while gradient boosting beats XGBoost on 12 of 165 benchmark data sets (second row, first column). Their performance is very similar on the remaining 119 of 165 data sets, meaning both models perform equally well on 119 data sets.

This data is very interesting. By comparing the wins and losses, we can gauge if the 2 models in comparison are in fact more similar than different. Take the example quoted from the book, if we were to compare just the wins, we might probably conclude that XGBoost is way more superior than gradient boosting (34 vs 12). However, if we consider that they are on par for a whopping 119 datasets, we might not be so quick to throw gradient boosting out of the window yet.

This chart certainly helps in shortlisting models for machine learning projects. Most importantly, it demonstrates that ensemble methods are indeed much more superior than individual models.

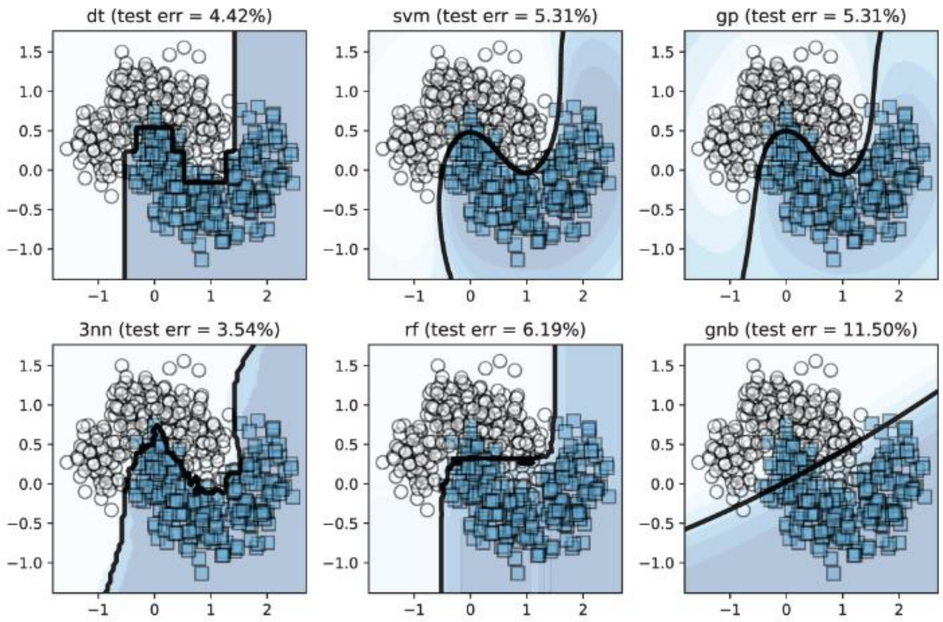
Information from:

Ensemble Methods For Machine Learning by Gautam Kunapuli (Manning Publications)

Link: 1 Ensemble methods: Hype or hallelujah? | Ensemble Methods for Machine Learning (oreilly.com)

2) Comparison between base models

The following picture shows how each of the base models perform individually.



This picture convincingly proves that each base model performs very differently and the decision boundaries differ greatly. What is interesting is how ensemble is actually able to combine these base models together to produce a more superior prediction. It is also interesting to see that random forest has a higher error than decision tree. Perhaps there were not enough trees to make up for the weak performance of each tree stump. Possibilities are aplenty. This chart simply demonstrates that machine learning is an art as much as it is science.

Information from:

Ensemble Methods For Machine Learning by Gautam Kunapuli (Manning Publications)

Link: 3 Heterogeneous parallel ensembles: Combining strong learners | Ensemble Methods for Machine Learning (oreilly.com)

References:

1) Ensemble Methods For Machine Learning by Gautam Kunapuli (Manning Publications) (https://learning.oreilly.com/library/view/ensemble-methods-for/9781617297137/)
2) Hard vs. Soft Voting Classifiers | Baeldung on Computer Science
3) Stacking Ensemble Machine Learning With Python - MachineLearningMastery.com
4) How to Develop Voting Ensembles With Python - MachineLearningMastery.com
5) Stacking Ensemble Machine Learning With Python - MachineLearningMastery.com
6) https://github.com/FernandoLpz/Stacking-Blending-Voting-Ensembles