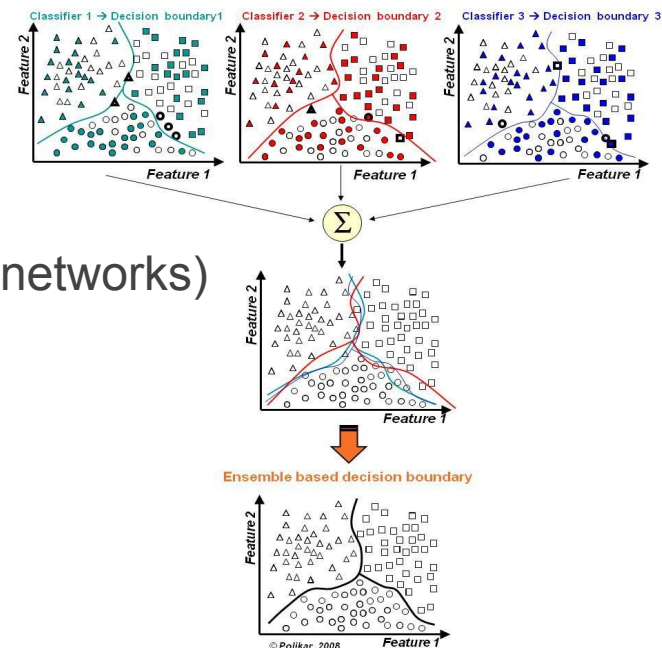


Ensemble Learning

What is Machine Learning Ensembles?

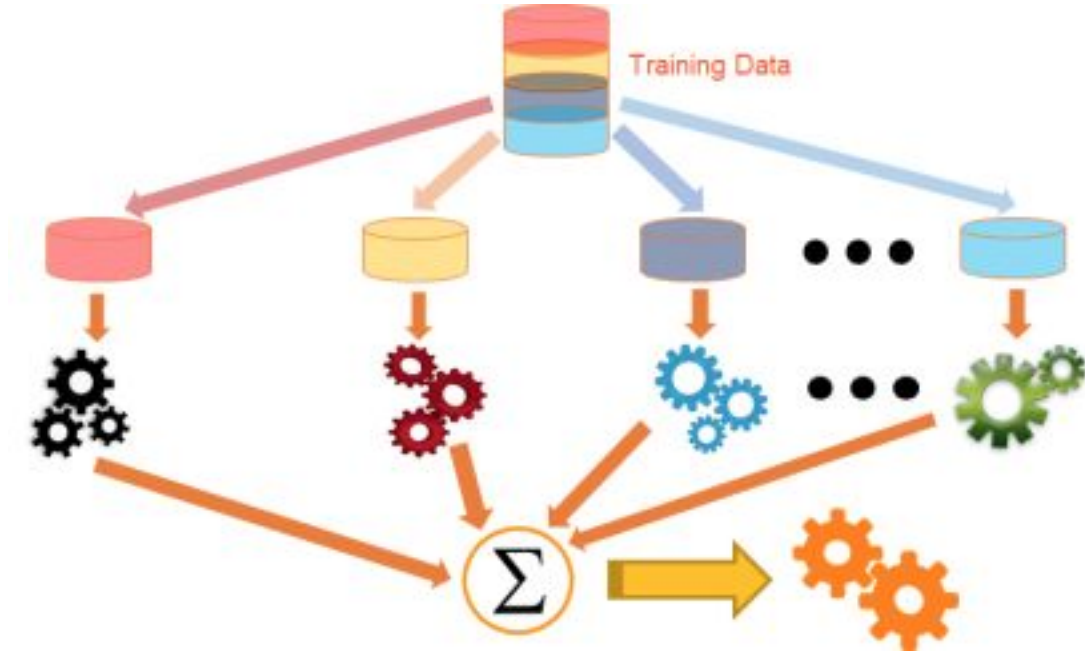
Machine Learning Ensembles

- Techniques that generate a group of base learner with when combined have higher accuracy
- Strong v.s. Weak learner
- Stable (kNN) v.s. Unstable (decision trees, neural networks) machine learning algorithms.



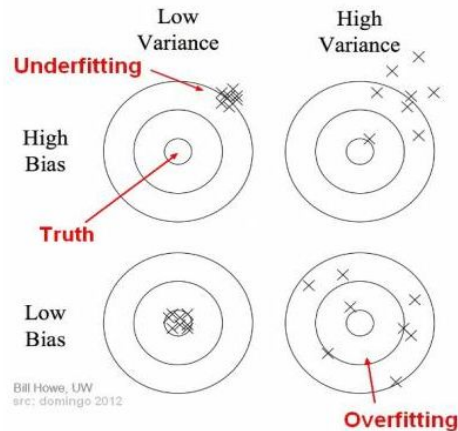
Why Ensemble?

- Reduce Bias
- Reduce Variance
- Prediction Error:
= Bias 2
+ Variance
+ Irreducible Error



Bias-Variance

- **Bias:** the difference between the average prediction of our model and the correct value which we are trying to predict
- **Variance:** the variability of model prediction for a given data point or a value which tells us spread of our data



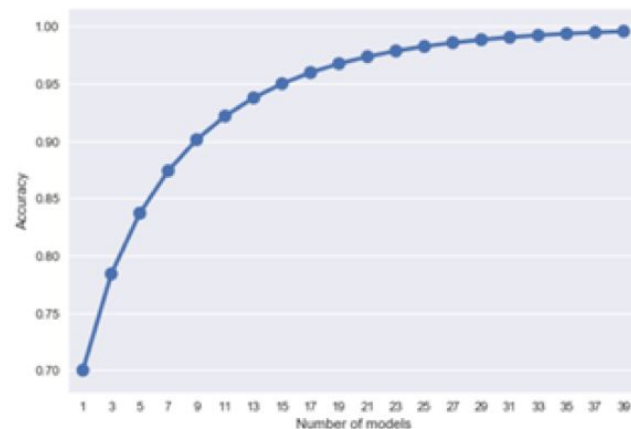
Reduce Bias

- Assume a test set of 10 samples and k (assume k is odd) independent binary classifiers, where each classifier has p accuracy. By combining these k classifiers using majority voting, the improved accuracy will be

$$\sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{i} p^{k-i} (1-p)^i$$

If $p = 0.7$, then we have

k	Ensemble Accuracy
1	0.7
3	0.784
5	0.83692
11	0.92177520904
101	0.999987057446

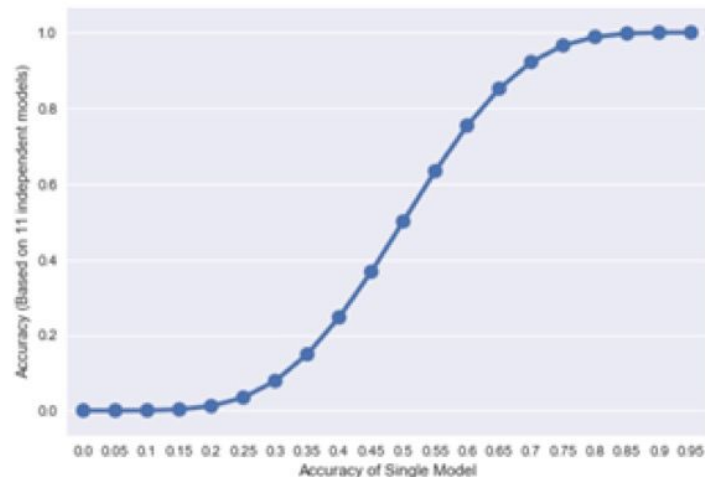


Reduce Bias

- Assume a test set of 10 samples and k (assume k is odd) independent binary classifiers, where each classifier has p accuracy. By combining these k classifiers using majority voting, the improved accuracy will be

$$\sum_{i=\lfloor \frac{k}{2} \rfloor}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{i} p^{k-i} (1-p)^i$$

Fix # of classifiers to be
11

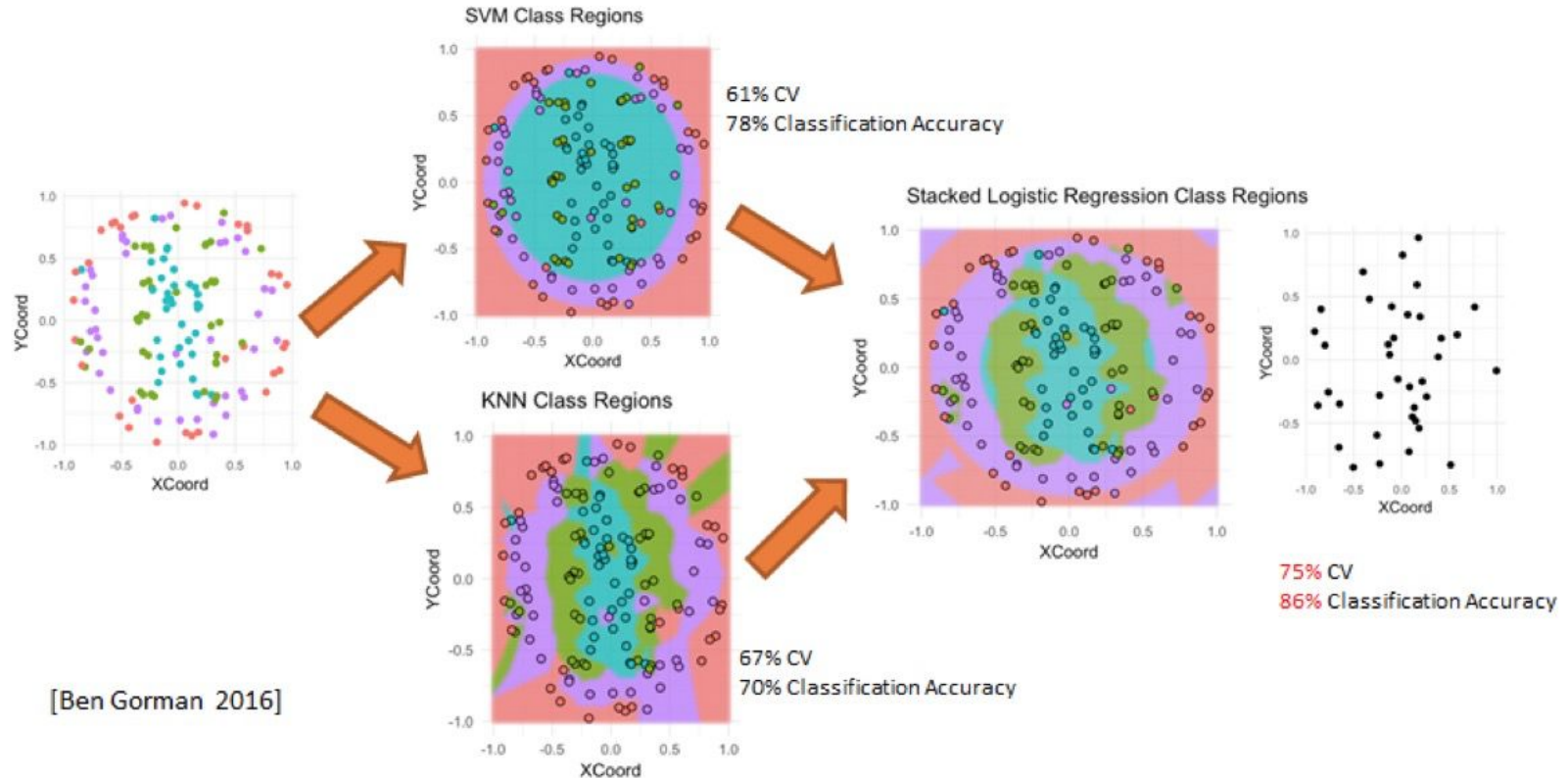


Reduce Variance

- Suppose we have n independent models: M_1, M_2, \dots, M_n with the same variance σ^2 . The ensemble constructed from these models using averaging will have the variance as follows:

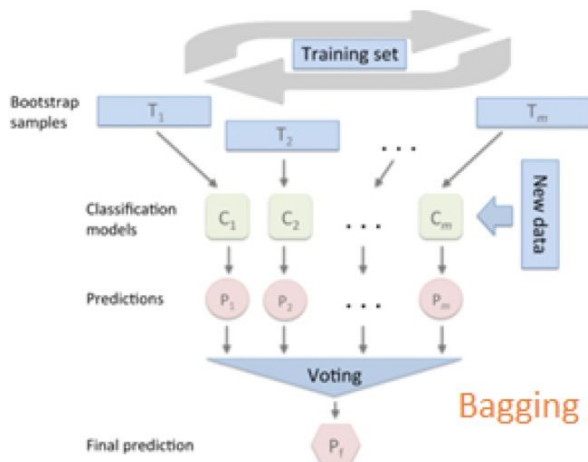
$$\begin{aligned}\text{Var}(M^*) &= \text{Var}\left(\frac{1}{n} \sum_i M_i\right) \\ &= \frac{1}{n^2} \text{Var}\left(\sum_i M_i\right) \\ &= \frac{1}{n^2} \cdot n \cdot \text{Var}(M_i) \\ &= \frac{\text{Var}(M_i)}{n}\end{aligned}$$

Machine Learning Ensembles

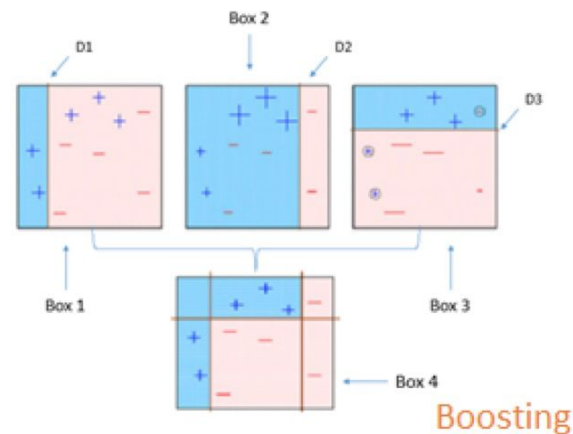


Common Ensemble Techniques

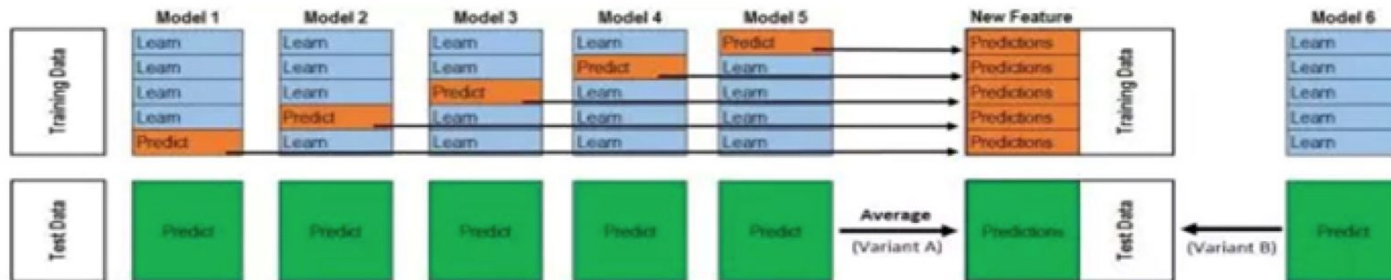
Overview



Bagging



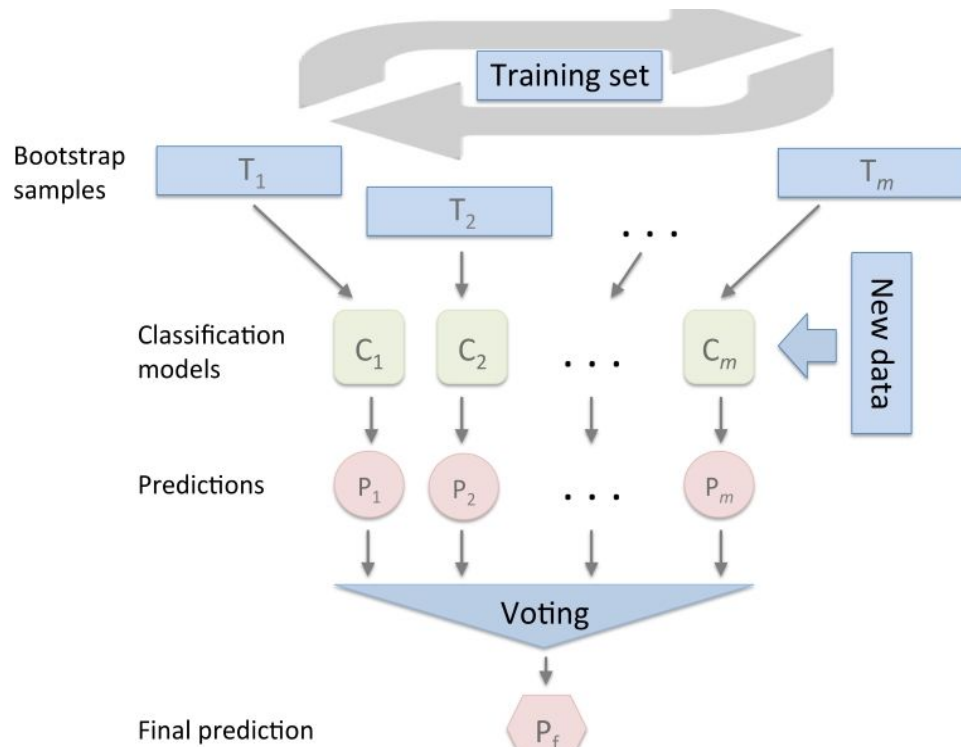
Boosting



Blending/Stacking

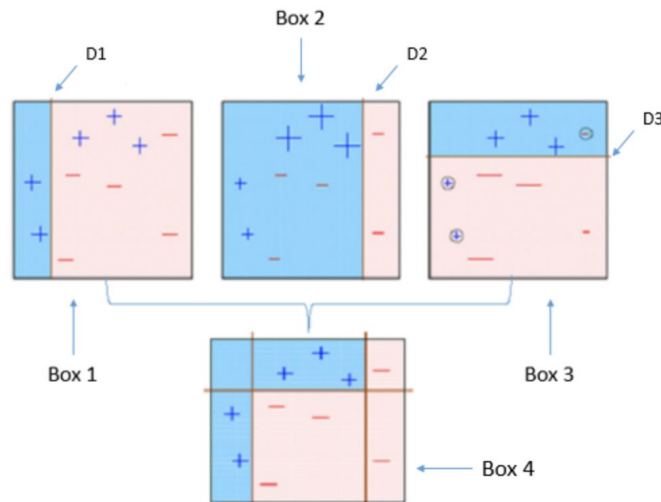
Bagging

- A.k.a Bootstrap aggregation
- Train m classifier from m bootstrap replica
- Combine outputs by voting
- Decreases error by decreasing the variance
- **Random Forest** (Randomly select features)
- **ExtraTrees** (Randomized top-down split)



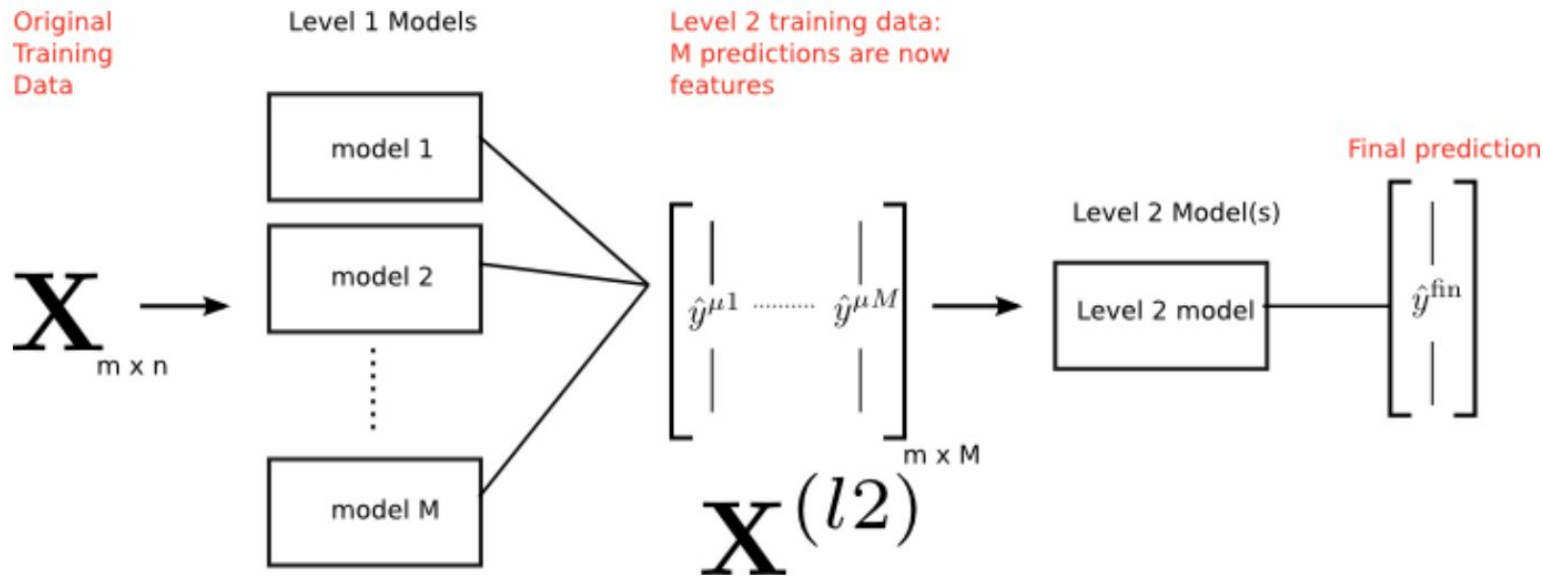
Boosting

- Training samples are given weights (initially same weight)
- At each iteration, a new hypothesis is learned.
- Training samples are reweighted to focus the model on samples that the most recently learned classifier got wrong
- Combine output by voting
- Gradient Boosting, Adaboost, XGBoost, LightGBM



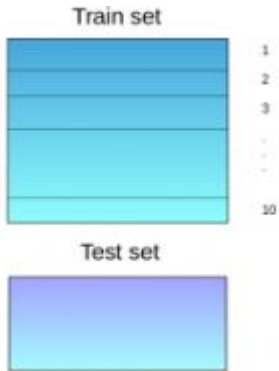
Stacking/Blending

- They are both ensemble learning technique that uses predictions from multiple models (such as KNN, SVM or Decision Tree) to build a new model.

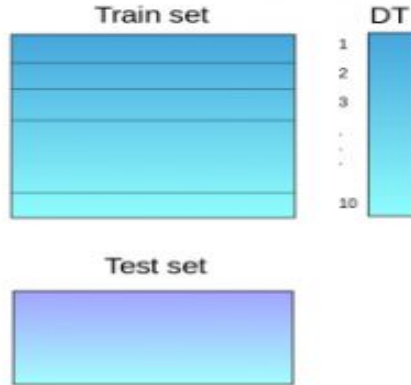


Stacking

1 Train set is split into 10 parts

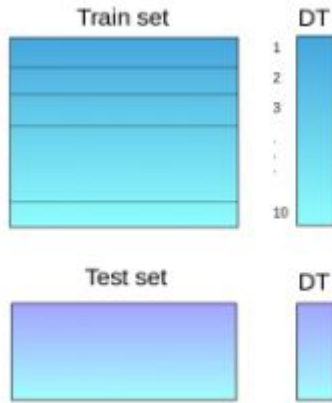


2 A base model is trained on 9 parts and predictions are made for the 10th part. It is looped for each part of data. And the prediction is regarded as new features.

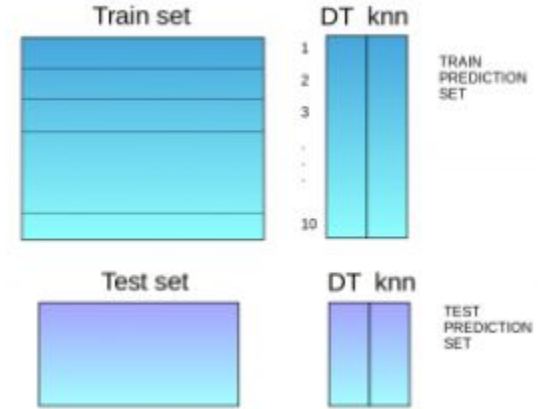


Stacking

3 The based model is then fitted on the whole train dataset. Then, predictions are made on the test dataset as the new features.



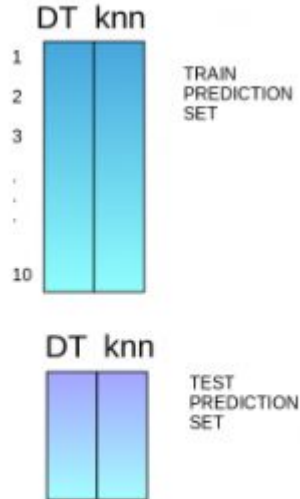
4. Steps 2-3 are repeated for another base model. Then, we are going to have another set of predictions for the train set and test set.



Stacking

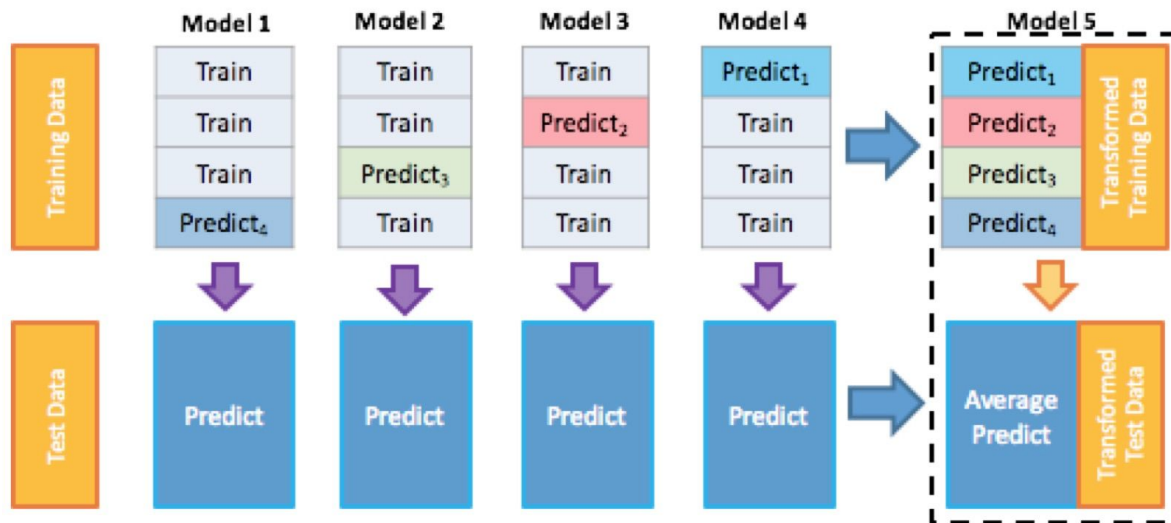
5 The predictions from the train set are used as features to build a new model.

6. This model is used to make final predictions on the test prediction set.



Stacking

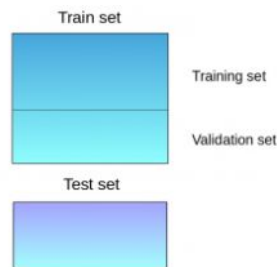
- Slightly different: in step 3, we just take the average predictions from one base model applied over different folders instead of re-training of the base model over the whole training dataset.



Blending

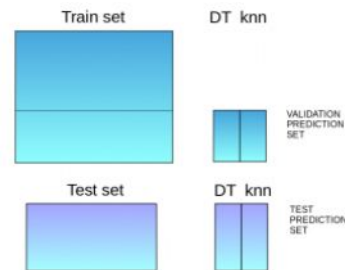
- Different from stacking, blending does not perform cv. The holdout set and its predictions are used to build a model which is run on the test set.

1 The train set is split into training and validation sets.



3 The validation set and its predictions are used as features to build a new model (level-2).

2 Models are fitted on the training set. And the predictions are made on validation set and test set



4 This model is used to make final predictions on test dataset and meta-features.

Some possible pitfalls

- Exponentially increasing training times and computational requirements
- Increase demand on infra. to maintain and update these models.
- Greater chance of data leakage between models or stages in the whole training.

In a nutshell

- **No Free Lunch Theorem:** There is no one algorithm that is always the most accurate.
- Our efforts should focus on obtaining base models which make different kinds of errors, rather than obtaining highly accurate base models
- What we need to do is to build weak learners that are at least more accurate than random guessing
- Feature Engineering !!!
- Keep trying (experimenting, tuning, etc.) !