BITI2223 Machine Learning

Ensemble Models

- Ensemble Models
- Bagging
- Randomization
- Boosting
- Stacking

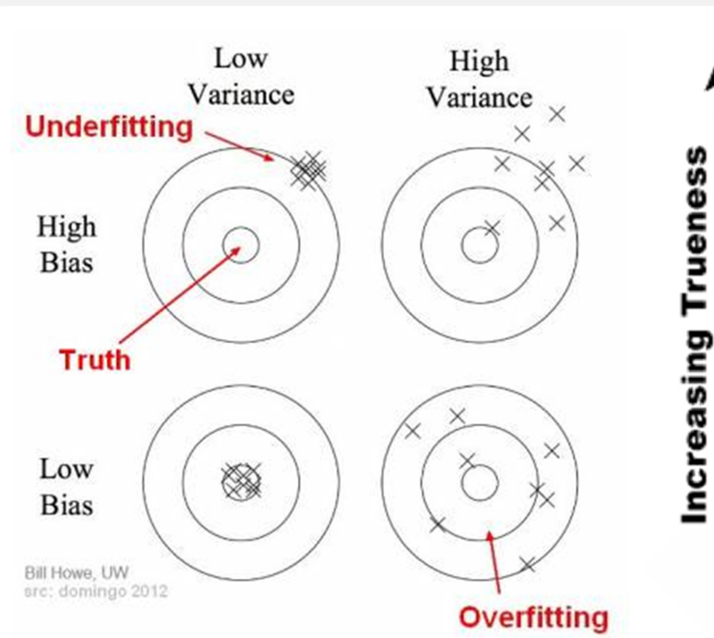
Outlines:

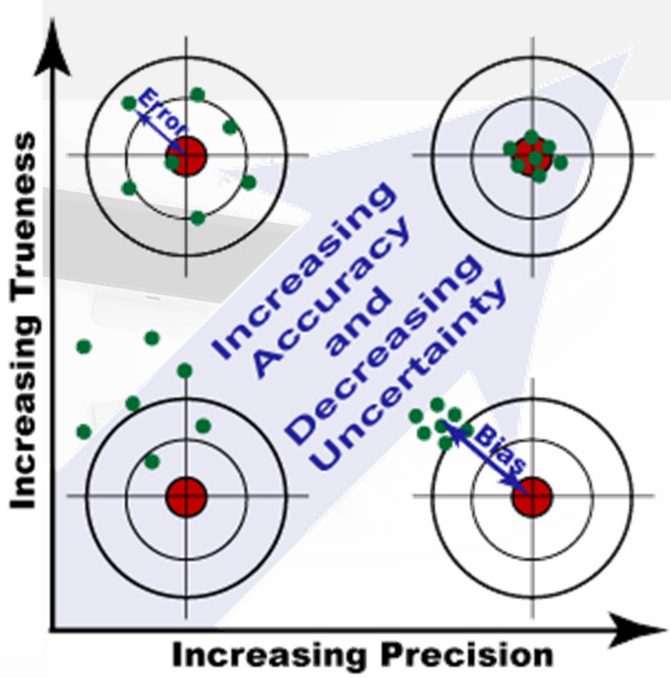
Ensemble Learning

- **Ensemble learning** is a **technique** that is used to create **multiple Machine Learning models**, which are then **combined** to produce more accurate results.

- A general Machine Learning model is built by using the entire training data set.

- However, in **Ensemble Learning** the **training data set is split into multiple subsets**, where in **each subset** is used to build a **separate model.**
  - After the models are trained, they are then **combined to predict an outcome** in such a way that the variance in the output is reduced.





Ensemble Models

- Basic idea:
  - build **different "experts"**, let them **vote**
- **Advantage:**
  - often **improves** predictive performance
- **Disadvantage:**
  - usually produces output that is **very hard to analyze**
  - but: there are approaches that aim to produce a single comprehensible structure

Bagging

- **Randomly resampling** the training data.
- Examples: Bagging, MetaCost
  - Drawing *m* training sets of the same size from the original data. Hopefully, each tricky part of the data will be chosen for training.
  - Combine the *m* resulting models using voting / averaging.
- Specifically **effective on small dataset** or on unstable learning model.
  - *Decreases error by decreasing the variance in the results due to unstable learners, algorithms (like decision trees) whose output can change dramatically when the training data is slightly changed.*

Bagging

- **Combine** multiple models.
- **Take the benefit** of differe nt models / learners on diversified data

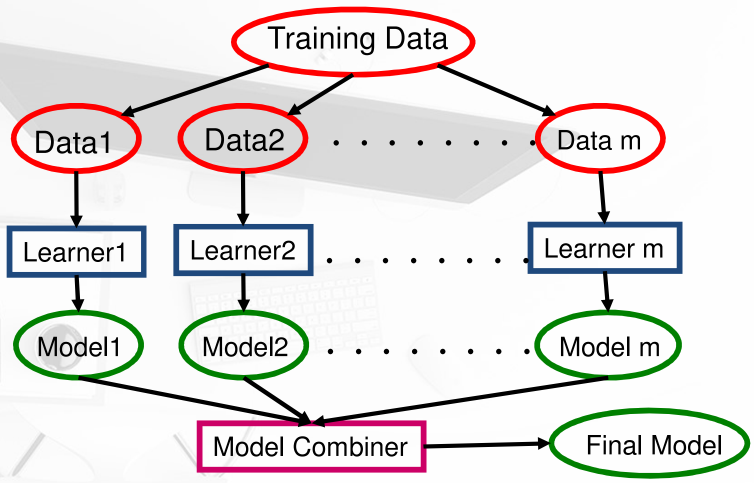


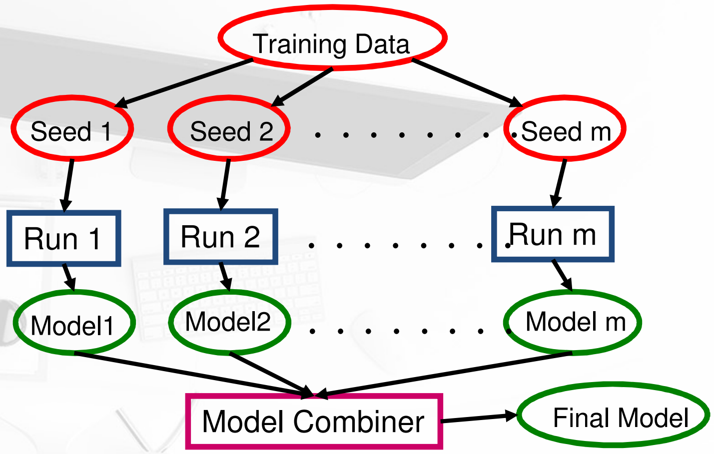
Figure is adopted form Raymond J. Mooney, Machine Learning Ensembles

Bagging

- Bagging works because **it reduces variance by voting/averaging**
  - Note: in some pathological hypothetical situations the overall error might increase
  - Usually, the more classifiers the better
  - Aside: bias-variance decomposition originally only known for numeric prediction
- Problem: we only have one dataset!
  - Solution: generate new ones of size *n* by sampling from it with replacement
- Can help a lot if data is noisy
- Can also be applied to numeric prediction
- Bagging unpruned decision trees known to produce good probability estimates
- Where, instead of voting, the individual classifiers' probability estimates are averaged

Randomization

- Use **random number seeds** to combine classifiers' results on several runs by voting or averaging.
- **<u>Examples:</u>** RandomForest, RansomCommittee, Ransom SubSpace



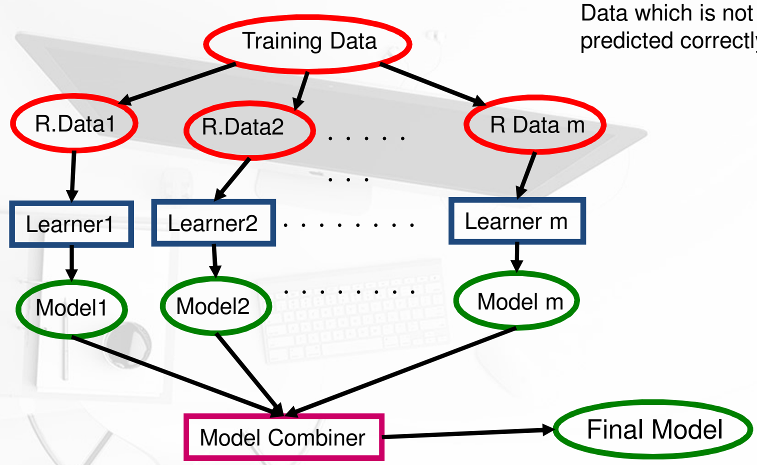
Randomization

- **Can randomize learning algorithm** instead of input
- Some algorithms already have a random component: eg. initial weights in neural net
- **Most algorithms can be randomized**, eg. greedy algorithms:
- Pick from the *N* best options at random instead of always picking the best options. E.g.. attribute selection in decision trees
- <u>More generally applicable than bagging</u>, e.g. random subsets in nearest-neighbor scheme
- Can be combined with bagging

Boosting

- Each new model is **influenced** by the performance of those built previously.
- Encourages new models to become experts for instances handled incorrectly by earlier ones by **assigning greater weight** to those instances.
- Weights a model's contribution by its **confidence** rather than giving equal weight to all models

R. Data = Remaining Data which is not predicted correctly



Stacking

- **To combine predictions of base learners** (different schemes) use meta learner
- **Not widely used** because it is difficult to analyse.
- Stacking can be applied to numeric prediction too.
- Many different implementations: choose best learners, use unweighted voting, use weighted voting, etc.