Ensemble Methods

Jayanth Rasamsetti
Founder & Chief Scientist
www.sgmoid.com
ex-American Express, ex-KPMG
Columbia University (MS)
IIT Madras (B.Tech & M.Tech)

Introduction to machine learning

What is an ensemble?

Introduction to machine learning

What is an ensemble?

Introduction to machine learning

What is <u>NOT</u> an ensemble?

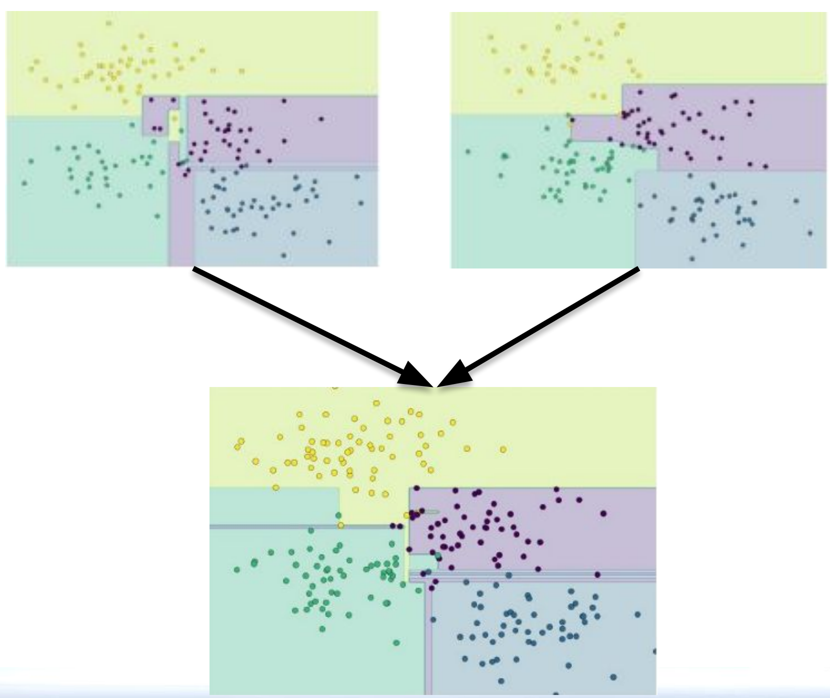
Image Source: www.bharatstudent.com

Introduction to machine learning

Consider the same dataset trained by two different models.

Is the learning better in the *combined* space? Guesses?

Meta models
Trained on slightly
differ data



Combined result with
better accuracy than
individual models

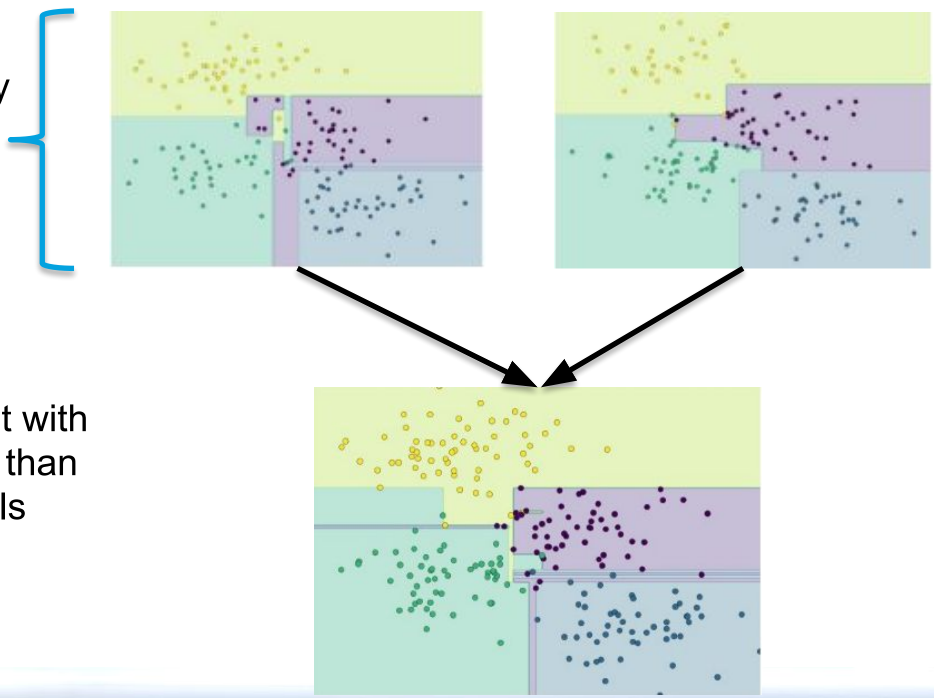
Source: Python Data
Science Handbook

Introduction to machine learning

Ensembles:

1) Train multiple *weak* predictors on a dataset such that they get slightly different results some learn some patterns better and others learn other patterns
2) Combine their predictions to get an overall better performance
3) The combined group of learners is called meta model or an ensemble

Meta models
Trained on slightly
differ data



Combined result with
better accuracy than
individual models

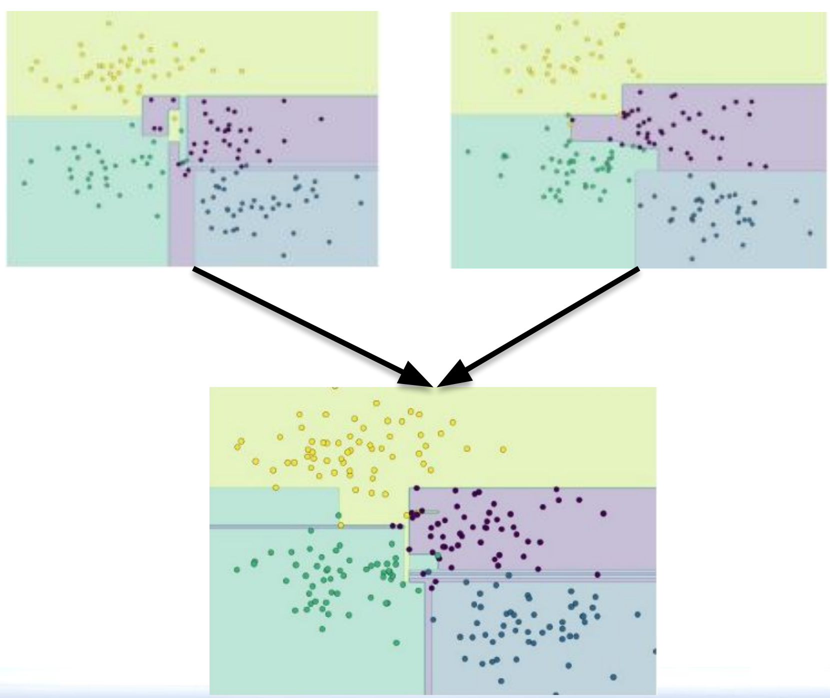
Source: Python Data
Science Handbook

Introduction to machine learning

In some parts of the feature space, the different instances produce similar results for e.g. extreme regions

In regions where the data points from different classes overlap, the instances give different results. By using information from all the instances, may give overall better result than individual instances

Meta models
Trained on slightly
differ data



Combined result with
better accuracy than
individual models

Source: Python Data
Science Handbook

Introduction to machine learning

Each learner gets to see slightly different data can be done in many ways.

Average: The driving principle is to build several estimators independently and then to average / vote their predictions. On average, the combined estimator is usually better than any of the single base estimator because its variance is reduced.

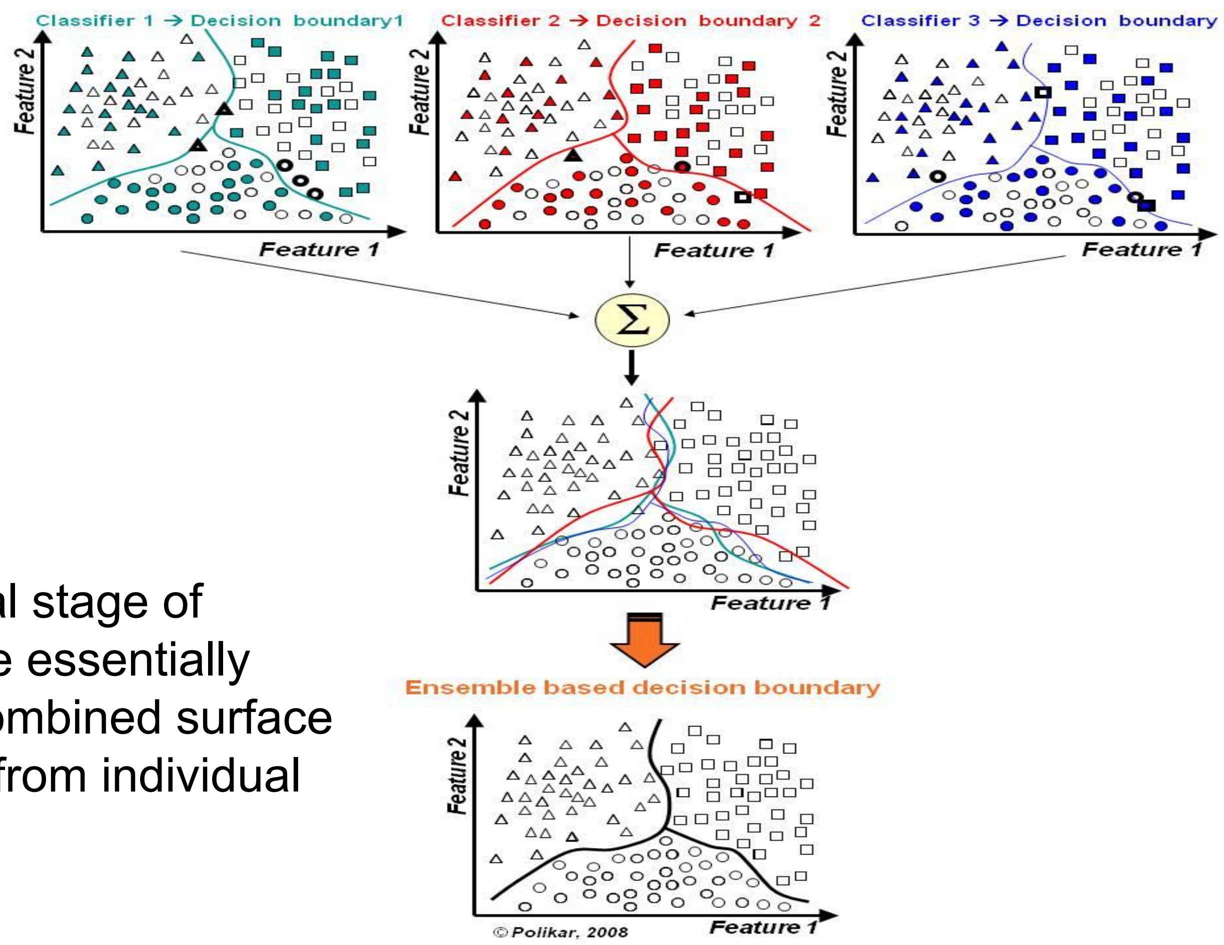
e.g. Bagging, Random Forest

Boosting: base estimators are built sequentially and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful ensemble.

e.g. AdaBoost, Gradient Tree Boosting

Introduction to machine learning

Ensembles



In the final stage of voting, we essentially have a combined surface resulting from individual surfaces

Source: https://github.com/MenuPolis/MLT/wiki/Bagging

Introduction to machine learning

What could bagging mean?

Introduction to machine learning

What could bagging mean?

Bagging (Bootstrap **Agg**regation)

1. Uses sampling with replacement to generate multiple samples of a given size. Sample may contain repeat data points
2. Multiple sample sets are created from the same data set using random function
3. Each sample data set is used to create a predictor

Data	Kohli	Dhoni	Sharma
BS1	?	?	?
BS2	?	?	?
BS3	?	?	?

Can you find out the bags?

Bagging (Bootstrap **Agg**regation)

1. Uses sampling with replacement to generate multiple samples of a given size. Sample may contain repeat data points
2. Multiple sample sets are created from the same data set using random function
3. Each sample data set is used to create a predictor

Data	Kohli	Dhoni	Sharma
BS1	Kohli	Dhoni	Dhoni
BS2	Sharma	Dhoni	Dhoni
BS3	Kohli	Kohli	Sharma

……. (Several possibilities!)

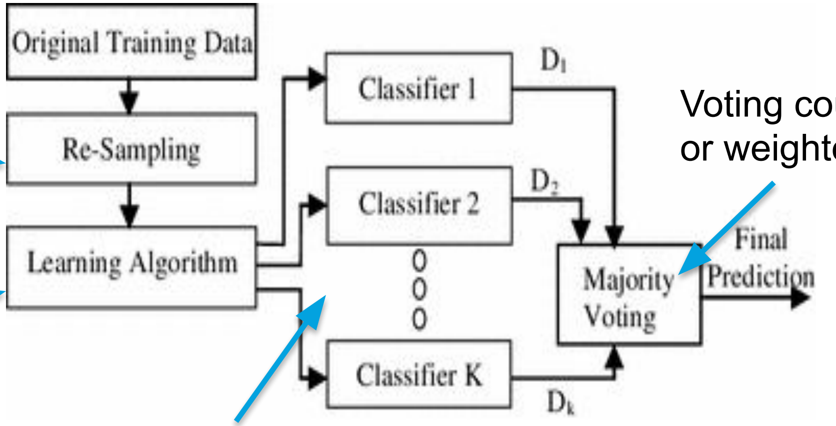
Bagging (Bootstrap **Agg**regation)

```
# configure bootstrap
n_iterations = 10   # Number of bootstrap samples to create
n_size = int(len(data) * 0.50) # picking only 50 % of the given data in every bootstrap sample

# run bootstrap
stats = list()
for i in range(n_iterations):
    # prepare train and test sets

    train = resample(values, n_samples=n_size) # Sampling with replacement
    test = np.array([x for x in values if x.tolist() not in train.tolist()])
    # picking rest of the data not considered in sample
```

Bagging (Bootstrap **Agg**regation)

Re-sampling done for every classifier using a random function. For large n, 63.2% unique samples likely to be selected

Algorithm to generate classifiers. Could be Decision Tree, Naïve Bayes etc



Voting could be simple or weighted

K classifiers created in parallel and independently on respective training data

Source: https://link.springer.com/article/10.1007/s13721-013-0034-x

Bagging (Bootstrap **Agg**regation)

1. Reduces variance errors and helps to avoid overfitting

2. Can be used with any type of machine learning model, *mostly used with Decision Tree*

3. For classification bagging is used with voting to decide the class of an input while for regression average or median values are calculated

4. For large sample size, sample data is expected to have roughly 63.2% (1 – 1/e) unique data points and the rest being duplicates

```
from sklearn.ensemble import BaggingClassifier
```

Introduction to machine learning

Case Study

Defaulting on debt by customers is over a USD 50 Billion industry. Large Retail banks are frequently susceptible to this. You are hired as a Machine Learning Engineer by Deutsche Bank to predict the defaulter prediction amongst customers. Let us try to improve defaulter prediction of the decision tree using bagging ensemble technique

Source: *https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)*

Introduction to machine learning

Dataset has 1000 rows and 17 columns

	checking_balance	months_loan_duration	credit_history	purpose	amount	savings_balance	employment_duration
0	< 0 DM	6	critical	furniture/appliances	1169	unknown	> 7 years
1	1 - 200 DM	48	good	furniture/appliances	5951	< 100 DM	1 - 4 years
2	unknown	12	critical	education	2096	< 100 DM	4 - 7 years
3	< 0 DM	42	good	furniture/appliances	7882	< 100 DM	4 - 7 years
4	< 0 DM	24	poor	car	4870	< 100 DM	1 - 4 years
5	unknown	36	good	education	9055	unknown	1 - 4 years
6	unknown	24	good	furniture/appliances	2835	500 - 1000 DM	> 7 years
7	1 - 200 DM	36	good	car	6948	< 100 DM	1 - 4 years
8	unknown	12	good	furniture/appliances	3059	> 1000 DM	4 - 7 years
9	1 - 200 DM	30	critical	car	5234	< 100 DM	unemployed

Decision Tree: 54% (Test)
Bagging: 67% (Test)

Source: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Introduction to machine learning

Bagging (Bootstrap **Agg**regation)

Lab: Improve defaulter prediction of the decision tree using bagging ensemble technique

Description: Sample data is available at local file system as credit.csv

Introduction to machine learning

What could boost mean?

Introduction to machine learning

What could boost mean?

Introduction to machine learning

Boosting

1. Similar to bagging, but the learners are grown sequentially; except for the first, each subsequent learner is grown from previously grown learners

2. If the learner is a Decision Tree, each of the trees can be small, with just a few terminal nodes (determined by the parameter d supplied)

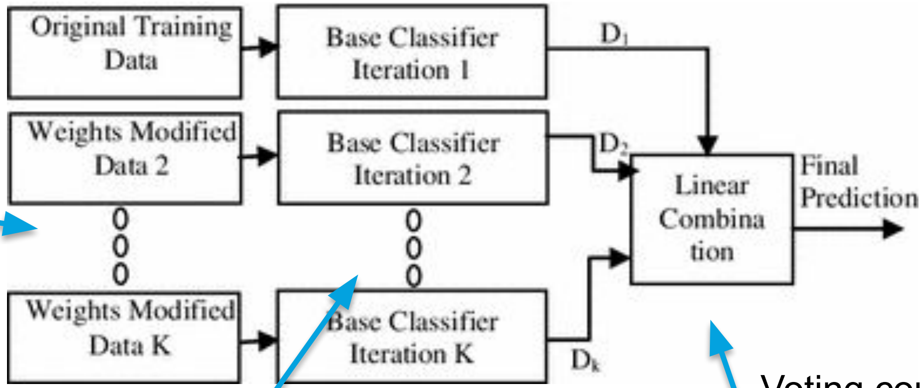
3. During voting higher weight is given to the votes of learners which perform better in respective training data unlike Bagging where all get equal weight

4. Boosting slows down learning (because it is sequential) but the model generally performs well

Introduction to machine learning

Boosting (**AdaBoost**)

Training data from base data with focus on instances which were incorrectly classified by earlier model (if any)



K similar classifiers created in sequence with respective training data with focus on addressing the misclassified data, not the usual cost functions

Voting could be simple or weighted

It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instance

Source: https://link.springer.com/article/10.1007/s13721-013-0034-x

Introduction to machine learning

Boosting

7. Two prominent boosting algorithms are AdaBoost, short for Adaptive Boosting and Gradient Descent Boosting

8. In AdaBoost, the successive learners are created with a focus on the ill fitted data of the previous learner

9. Each successive learner focuses more and more on the harder to fit data i.e. their residuals in the previous tree

```
]: from sklearn.ensemble import AdaBoostClassifier
```

Introduction to machine learning

Recently asked Interview Question:

1. Boosting is faster compared to Bagging

A. True

B. False

2. Which of the following algorithm is not an example of an ensemble method?

A. Extra Tree Regressor

B. Random Forest

C. Gradient Boosting

D. Decision Tree

Introduction to machine learning

Recently asked Interview Question:

1. Boosting is faster compared to Bagging

A. True
B. False

Ans) False

2. Which of the following algorithm is not an example of an ensemble method?

A. Extra Tree Regressor
B. Random Forest
C. Gradient Boosting
D. Decision Tree

Ans) D

Boosting (**AdaBoost**)

Adapting weights with focus on erroneously classified instances

Given: $(x_1,y_1), \dots, (x_m,y_m)$, *Where* $x_i \in X$, $y_i \in Y = \{1, 2, \dots, K\}$

1. *Initialize the weights* $w_i^1 = 1/m$, $i=1, 2, \dots, m$
2. *For* $t=1$ *to* T
   - (a) *Fit a classifier* $h^t(x)$ *to the training data using weights* w_i^t
   - (b) *Compute*

$$err^t = Pr_{i \sim w_i^t}[h^t(x_i) \neq y_i] = \sum_{i=1,\, h^t(x_i) \neq y_i}^{m} w_i^t \Big/ \sum_{i=1}^{m} w_i^t$$

If $err^t > 1/2$*, then* $t=T-1$ *and abort loop.*

   - (c) *Compute*

$$\alpha^t = \log \frac{1 - err^t}{err^t}$$

   - (d) *Set*

$$w_i^t \leftarrow \begin{cases} w_i^t \cdot \exp(\alpha^t) & if \quad h^t(x_i) \neq y_i \\ w_i^t & otherwise \end{cases} \qquad i = 1, 2, \dots, m$$

   - (e) *Renormalize* w_i^t
3. *Output*

$$H(x) = \arg\max_{y \in Y} \sum_{t=1,\, h^t(x)=y}^{T} \alpha^t$$

Initialize weights, equal weights to all instances

Generate first classifier with equal focus on all instances

Total up weights of all error instances, express it as a ratio to total weights

If error ratio is > 50%

Calculate predictor weights (i.e. weight of the classifier)

Assign new weights to instances misclassified, else keep the weights same

Renormalize the weights across all the instances and fit next classifier

For a test instance use weighted voting to identify the class

AdaBoost:

Lab - 7 Improve defaulter prediction of the decision tree using Adaboosting

Description – Sample data is available at local file system as credit.csv

The dataset has 16 attributes described at
https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)
or in the notes page of this slide

`sklearn.ensemble`.**GradientBoostingClassifier**

Sol: Adaboost+Credit+Decision+Tree.ipynb

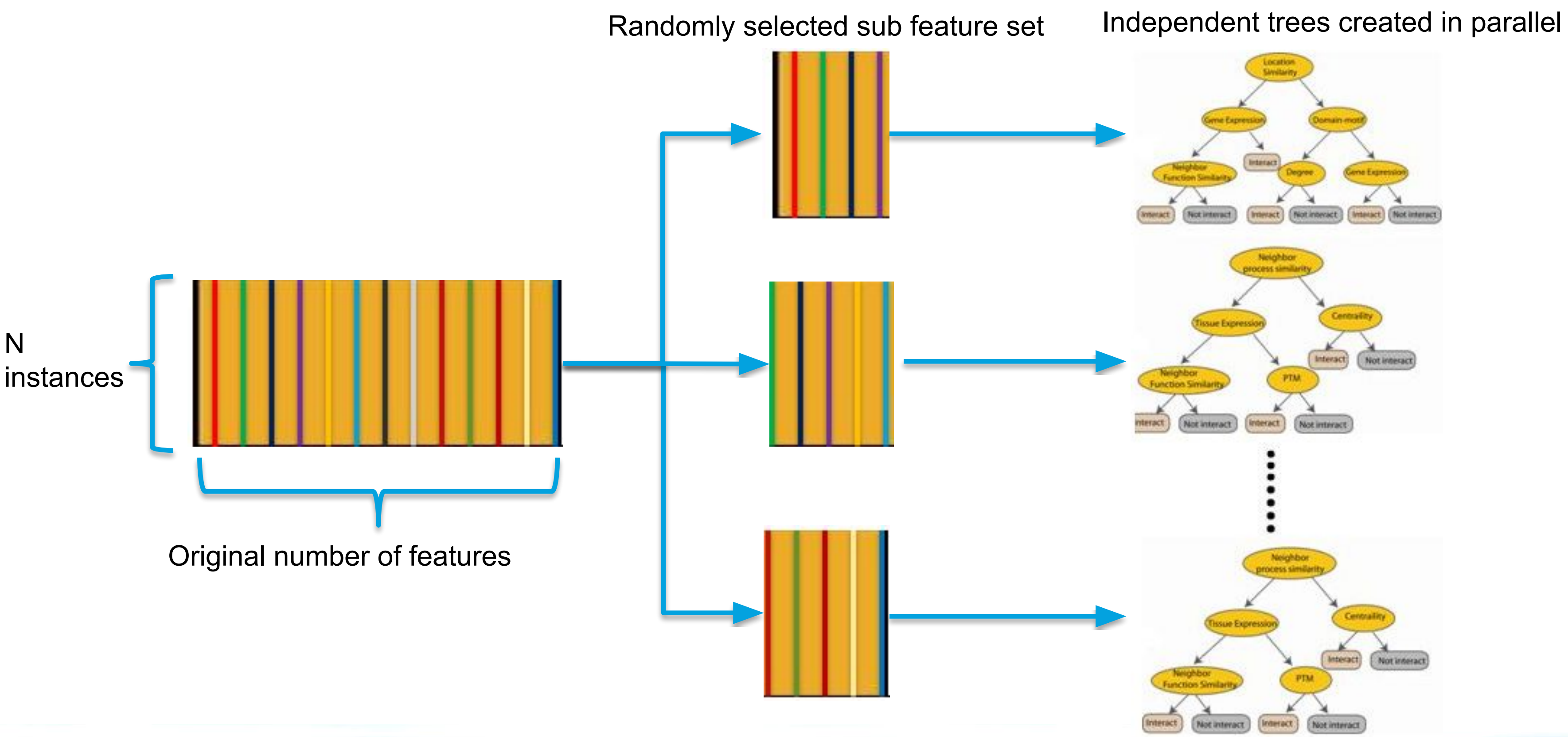
Random Forest

1. Each tree in the ensemble is built from a sample drawn with replacement (bootstrap) from the training set
2. In addition, when splitting a node during the construction of a tree, the split that is chosen is no longer the best split among all the features
3. Instead, the split picked is the best split among a random subset of the features
4. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree)
5. Due to averaging, its variance decreases, usually more than compensating the increase in bias, hence yielding overall a better result

```
from sklearn.ensemble import RandomForestClassifier
rfc1 = RandomForestClassifier(n_estimators = 50)
```

Random Forest

Used with Decision Trees. Create different trees by providing different sub-features from the feature set to the tree creating algorithm. The optimization function is Entropy or Gini index



Random Forest (Tuning):

All the parameters of decision trees and more

max_features: The number of features to consider when looking for the best split

class_weight: Weights associated with classes in the form {class_label: weight}.

If not given, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of y.

```
from sklearn.ensemble import RandomForestClassifier
rfcl = RandomForestClassifier(n_estimators = 50)
```

Random Forest (Tuning):

bootstrap: Whether bootstrap samples are used when building trees.

oob_score: bool (default=False) Whether to use out-of-bag samples to estimate the generalization accuracy.

n_jobs: The number of jobs to run in parallel for both fit and predict. None means 1 unless in a joblib.parallel_backend context. -1 means using all processors. See Glossary for more details.

Attributes

feature_importances_: Return the feature importances (the higher, the more important the feature).

```
from sklearn.ensemble import RandomForestClassifier
```

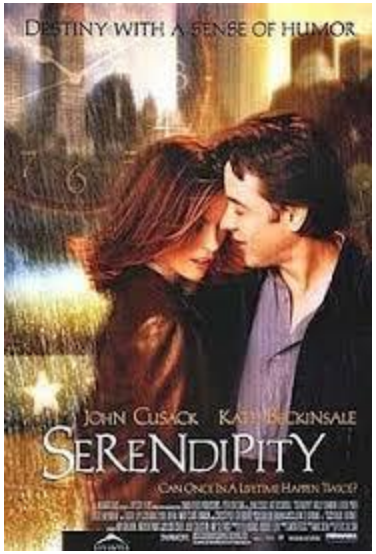
Random Forest is incredibly useful for another trick

Can you spot that?

checking_balance	months_loan_duration	credit_history	purpose	amount	savings_balance	employment_duration
< 0 DM	6	critical	furniture/appliances	1169	unknown	> 7 years
1 - 200 DM	48	good	furniture/appliances	5951	< 100 DM	1 - 4 years
unknown	12	critical	education	2096	< 100 DM	4 - 7 years
< 0 DM	42	good	furniture/appliances	7882	< 100 DM	4 - 7 years
< 0 DM	24	poor	car	4870	< 100 DM	1 - 4 years
unknown	36	good	education	9055	unknown	1 - 4 years
unknown	24	good	furniture/appliances	2835	500 - 1000 DM	> 7 years
1 - 200 DM	36	good	car	6948	< 100 DM	1 - 4 years
unknown	12	good	furniture/appliances	3059	> 1000 DM	4 - 7 years
1 - 200 DM	30	critical	car	5234	< 100 DM	unemployed

Introduction to machine learning

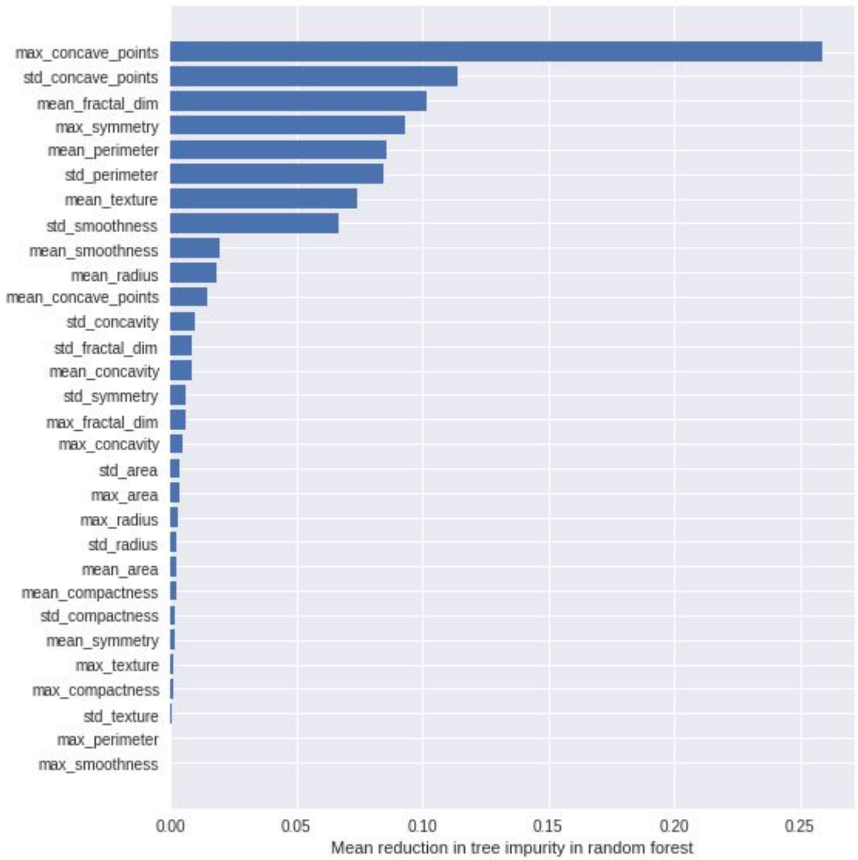
Random Forest is incredibly useful for another trick



Introduction to machine learning

Random Forest is incredibly useful for another trick

feature_importances_: Return the feature importances (the higher, the more important the feature).



Random Forest

Lab - 9 Improve defaulter prediction of the decision tree using Random Forest

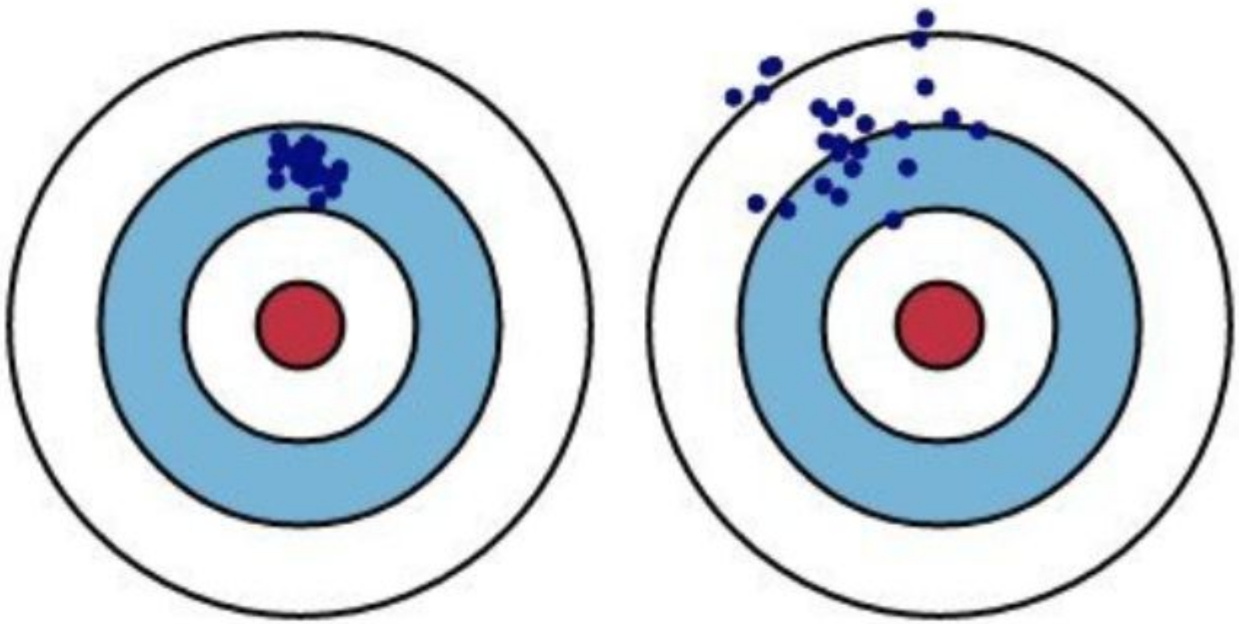
Description – Sample data is available at local file system as credit.csv

The dataset has 16 attributes described at
https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)
or in the notes page of this slide

Sol: RF+Credit+Decision+Tree.ipynb

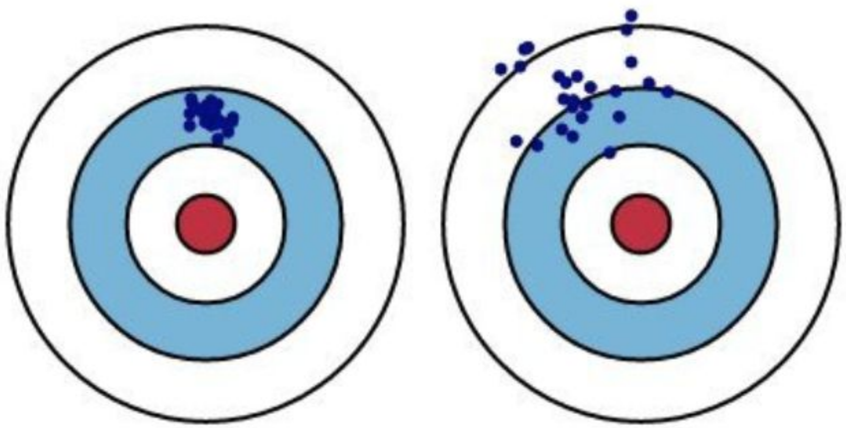
Introduction to machine learning

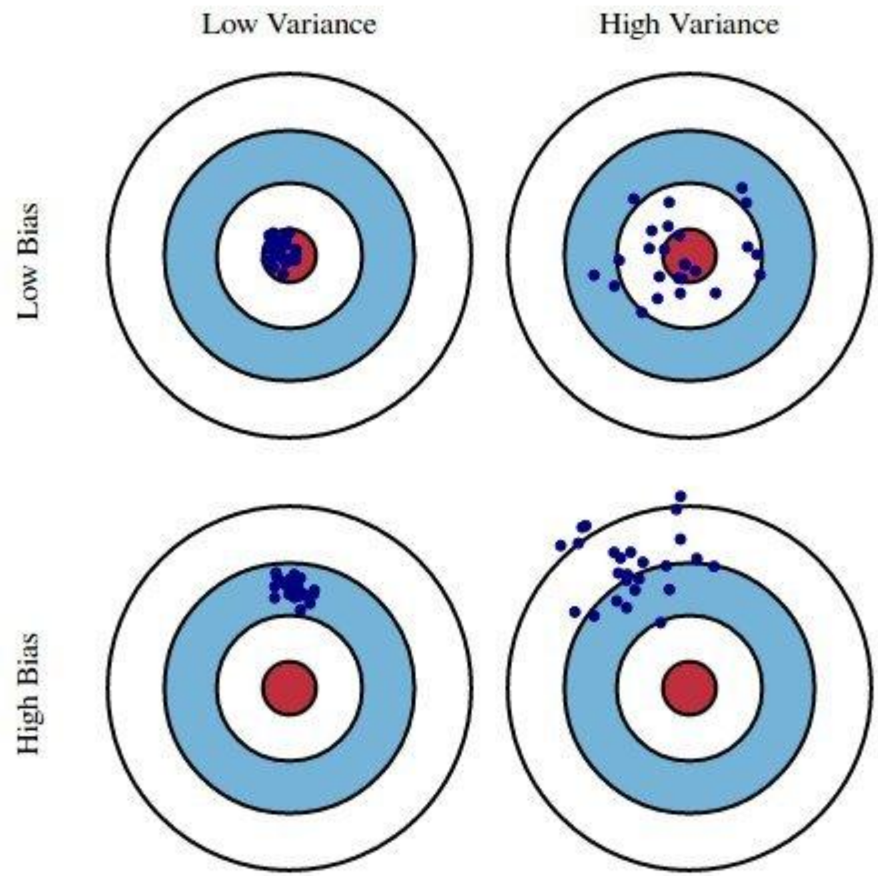
Which image is high bias and which is high variance??



Introduction to machine learning

1. Bagging decreases the model's variance
2. Boosting decreases the model's bias





Stacking

1. Similar to bagging, but apply several different models to original data
2. The weights for each model is determined based on how well they perform on the given input data
3. Similar classifiers usually make similar errors (bagging), so forming an ensemble with similar classifiers may not improve the classification rate
4. Presence of a poorly performing classifier may cause performance deterioration in the overall performance

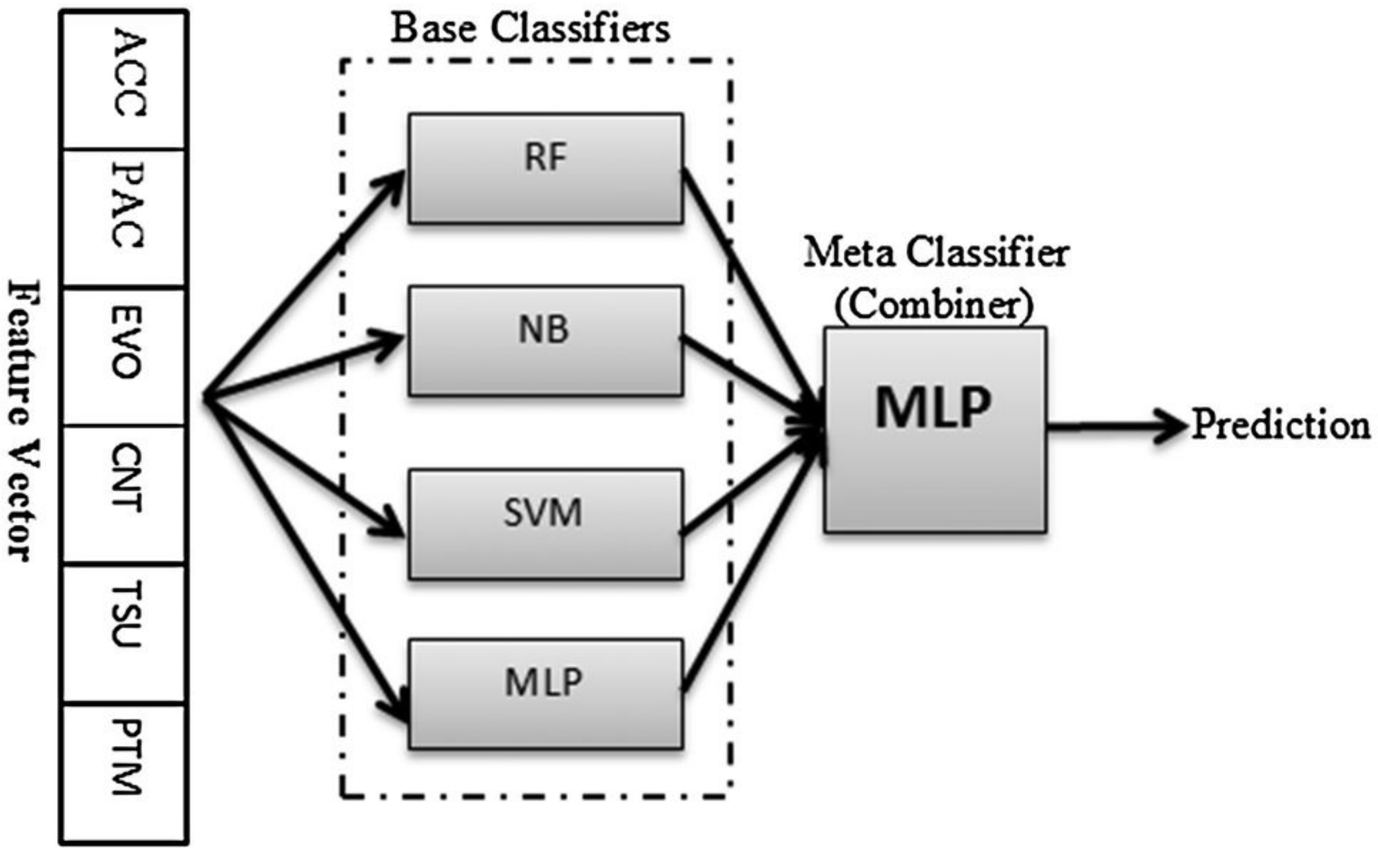
Introduction to machine learning

Stacking

1. Similarly, even on presence of a classifier that performs much better than all of the other available base classifiers, may cause degradation in the overall performance
2. Another important factor is the amount of correlation among the incorrect classifications made by each classifier
3. If the consistent classifiers tend to misclassify the same instances, then combining their results will have no benefit
4. In contrast, a greater amount of independence among the classifiers can result in errors by individual classifiers being overlooked when the results of the ensemble are combined.

Introduction to machine learning

Stacking



Source:
http://pubs.rsc.org/-/content/articlelanding/2014/mb/c4mb00410h/unauth#!divAbstract

Introduction to machine learning

Stacking

Lab- 10 Improve defaulter prediction of the decision tree using Stacking

Description – Sample data is available at local file system as credit.csv

The dataset has 16 attributes described at
https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)
or in the notes page of this slide

Sol: Stacking+Credit+Decision+Tree.ipynb

Introduction to machine learning

K-Folds Cross Validation

1. Divide data into k parts
2. Use k-1 of the parts for training, and 1 for testing
3. Repeat the procedure k times, rotating the test set
4. Determine an expected performance metric (MSE, Accuracy, etc) based on the results across the iterations


```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(model, iris.data, iris.target, cv=5)
scores
```

```
array([0.96, 1.  , 0.96, 0.96, 1.  ])
```

Introduction to machine learning

Regularization

1. Simple models preferred over complex ones
2. Complex models lead to overfit
3. L1 (Lasso) and L2 (Ridge) are elegant ways of achieving this

$$L(x,y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \boxed{\lambda \sum_{i=1}^{n} |\theta_i|}$$

$$L(x,y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \boxed{\lambda \sum_{i=1}^{n} \theta_i^2}$$

Introduction to machine learning

L1 Regularization

$$L(x,y) \equiv \sum_{i=1}^{n}(y_i - h_\theta(x_i))^2 + \boxed{\lambda \sum_{i=1}^{n}|\theta_i|}$$

1. L1 penalizes sum of absolute value of weights.
2. L1 has multiple solutions L1 has built in feature selection
3. L1 is robust to outliers
4. L1 generates model that are simple and interpretable but cannot learn complex patterns

L2 Regularization

$$L(x,y) \equiv \sum_{i=1}^{n}(y_i - h_\theta(x_i))^2 + \boxed{\lambda \sum_{i=1}^{n}\theta_i^2}$$

1. L2 regularization penalizes sum of square weights.
2. L2 has one solution
3. L2 has no feature selection
4. L2 is not robust to outliers
5. L2 gives better prediction when output variable is a function of all input features
6. L2 regularization is able to learn complex data patterns

Introduction to machine learning

Bonus: Xgboost (Extreme Gradient Boost) (Best ML Algorithm right now!)

n_estimators - Number of trees to be formed

max_depth [default=6] Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. 0 indicates no limit.

learning_rate - eta [default=0.3, alias: learning_rate] Step size shrinkage used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features, and eta shrinks the feature weights to make the boosting process more conservative. range: [0,1]

```
:  from xgboost import XGBClassifier
   xgb_model = XGBClassifier()
   xgb_model.fit(X_train, train_labels)
```

```
:  XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
          colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0,
          max_depth=3, min_child_weight=1, missing=None, n_estimators=100,
          n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
          reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
          silent=True, subsample=1)
```

Bonus: Xgboost (Extreme Gradient Boosting)

Discussion: Look into the parameter document and make notes on any 3 of the Xgb parameter email to jr3281@columbia.edu

```
: from xgboost import XGBClassifier
  xgb_model = XGBClassifier()
  xgb_model.fit(X_train, train_labels)
```

```
: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
         colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0,
         max_depth=3, min_child_weight=1, missing=None, n_estimators=100,
         n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
         reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
         silent=True, subsample=1)
```

Sol: Stacking+Credit+Decision+Tree.ipynb

Introduction to machine learning

Recently asked Interview Question:

Which of the following algorithm is not an example of an ensemble method?

A. Extra Tree Regressor
B. Random Forest
C. Gradient Boosting
D. Decision Tree

Introduction to machine learning

Recently asked Interview Question:

Which of the following algorithm is not an example of an ensemble method?
A. Extra Tree Regressor
B. Random Forest
C. Gradient Boosting
D. Decision Tree

Ans D

Introduction to machine learning

Recently asked Interview Question:

Which of the following algorithm uses the clever trick of oob_error?
A. Decision Tree
B. Random Forest
C. Linear Regression
D. Naive Bayes

Introduction to machine learning

Recently asked Interview Question:

Which of the following algorithm uses the clever trick of oob_error?
A. Decision Tree
B. Random Forest
C. Linear Regression
D. Naive Bayes

Ans D

Microsoft

Supervised Machine Learning

Further Reading:

1) https://news.ycombinator.com/news
2) http://course.fast.ai/ml.html
3) Best way to increase programming speed is “pair programming”

Don’t:
1) Do Andrew NG’s Machine Learning Course (Uses’ Octave an obsolete language)
2) Look at too many resources/books

Thank you!

Jayanth Rasamsetti
Founder www.sgmoid.com
ex-American Express, ex-KPMG
Columbia University (MS)
IIT Madras (B.Tech & M.Tech)