

Intro to ensemble methods

By Marios Michailidis



Background

- Research data scientist at **H₂O.ai**
- PhD in ensemble methods
- Former kaggle #1



Μαριος Μιχαηλιδης **KazAnova**

Data Scientist at H2O ai
Volos, Greece

Joined 4 years ago · last seen in the past day

<https://www.facebook.com/StackNet/>

Followers 465
Following 35



Competitions
Grandmaster

[Home](#) [Competitions \(101\)](#) [Kernels \(11\)](#) [Discussion \(539\)](#) [Datasets \(1\)](#) ...

[Edit Profile](#)

Competitions Grandmaster



Current Rank
3
of 65,862

Highest Rank
1

26

23

21

- [Homesite Quote Conversion](#) **1st**
2 years ago · Top 1% of 1764
- [Truly Native?](#) **1st**
2 years ago · Top 1% of 274
- [Acquire Valued Shoppers C...](#) **1st**
3 years ago · Top 1% of 952

Kernels Contributor



Unranked

0

0

0

- [Xgboost python scores aro...](#) **5** votes
6 months ago
- [Your Second Round vs the ...](#) **4** votes
2 years ago
- [enhanced](#) **3** votes
2 years ago

Discussion Master



Current Rank
2
of 36,751

Highest Rank
1

36

43

271

- [The 'Magic' \(Leak\) feature i...](#) **224** votes
5 months ago
- [Score 0.53776 \(or 0.52879\)...](#) **149** votes
6 months ago
- [My Approach](#) **94** votes
5 months ago



What is ensemble modelling?

It means combining different machine learning models to get a better prediction.

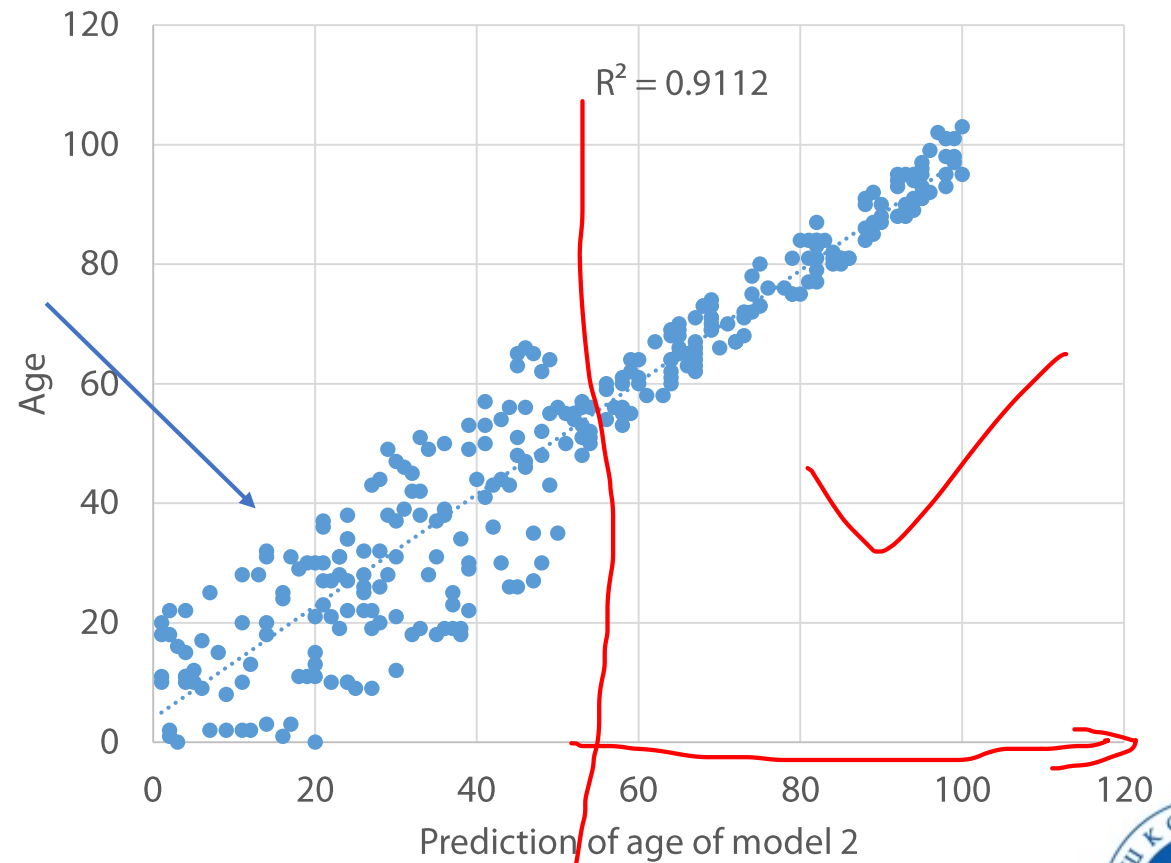
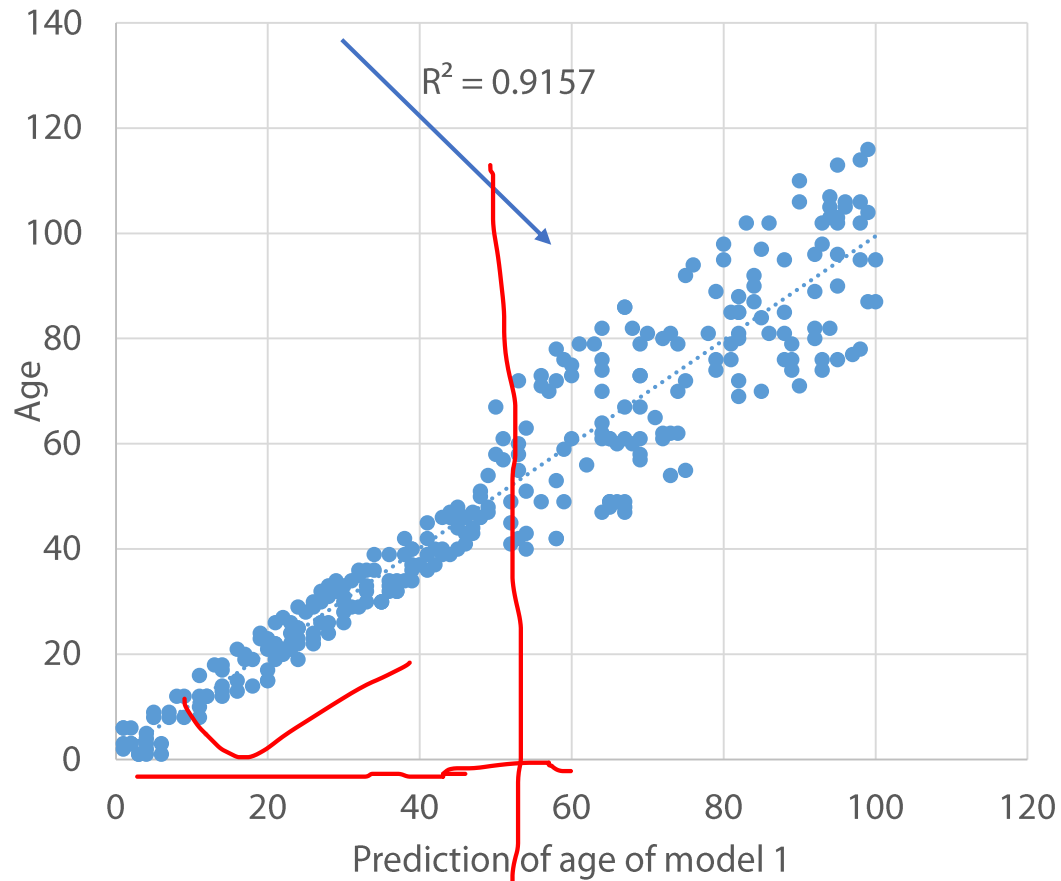


Examined ensemble methods

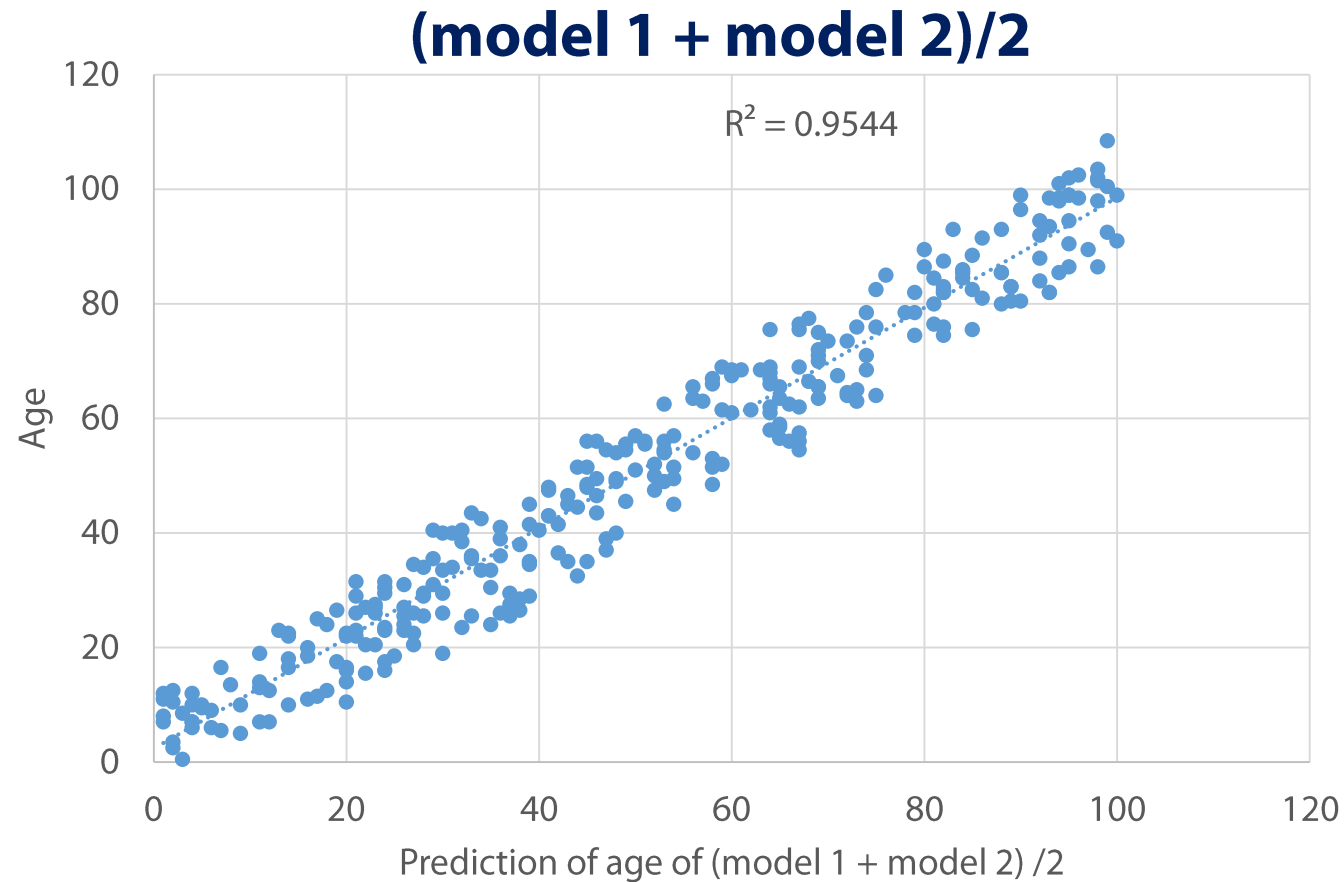
- Averaging (or blending)
- Weighted averaging
- Conditional averaging
- Bagging
- Boosting
- Stacking
- StackNet



Averaging ensemble methods

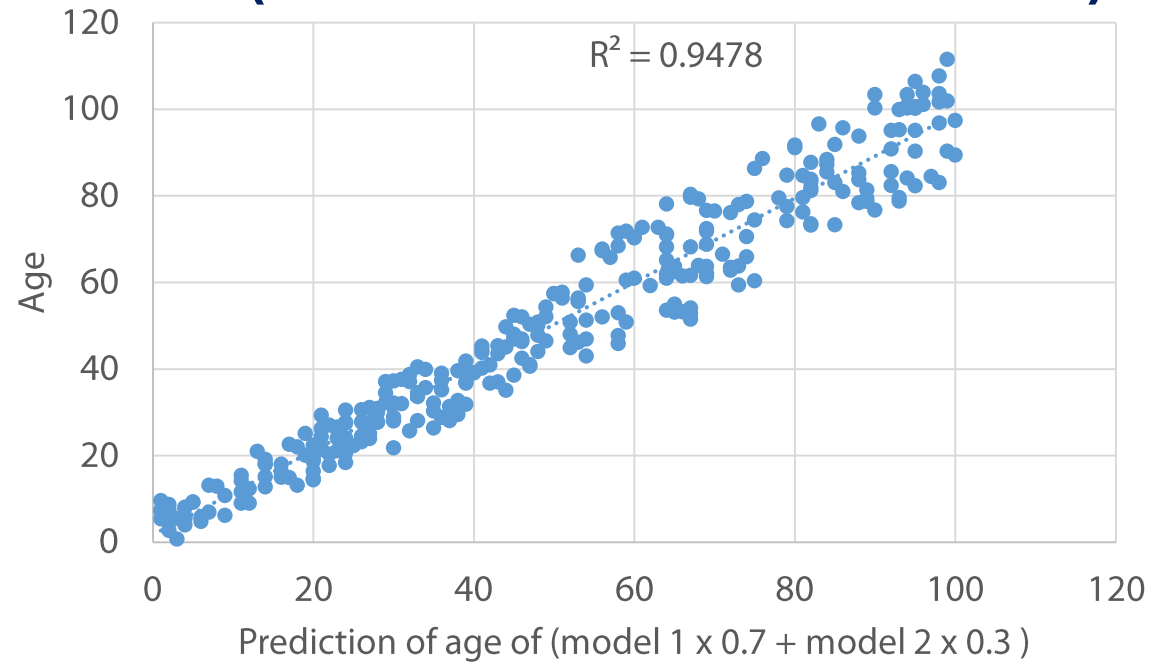


Averaging ensemble methods

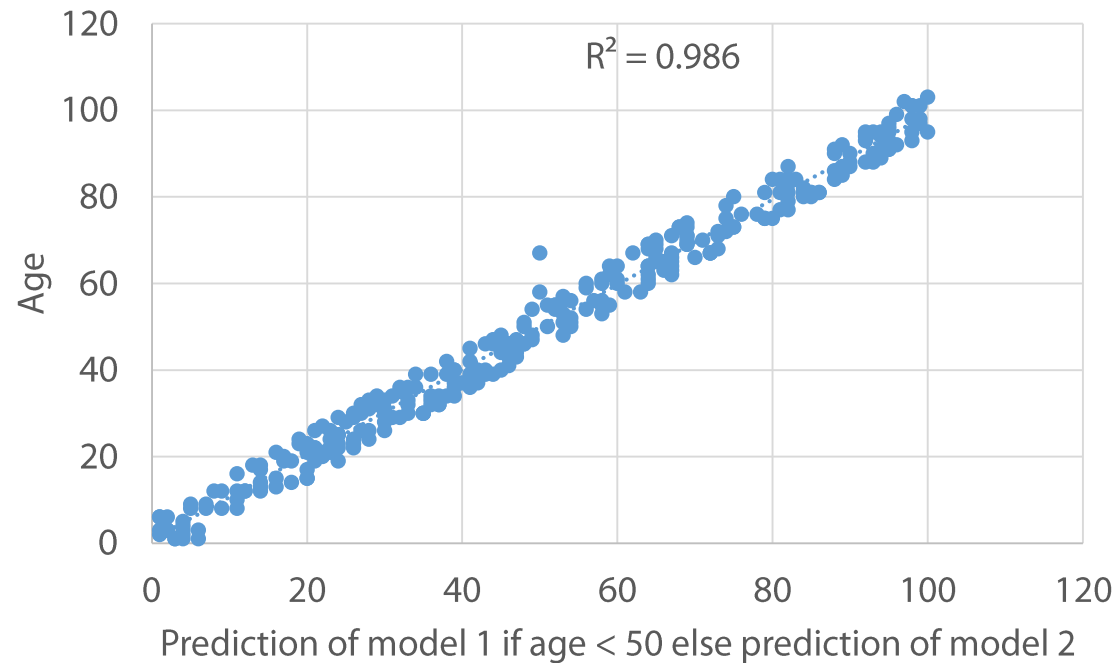


Averaging ensemble methods

(model 1 x 0.7 + model 2 x 0.3)



Averaging ensemble methods



Ensemble methods: bagging

By Marios Michailidis



Examined ensemble methods

- Averaging (or blending)
- Weighted averaging
- Conditional averaging
- Bagging
- Boosting
- Stacking
- StackNet



What is Bagging

Means **averaging** slightly different versions of the same model to improve accuracy



Why Bagging

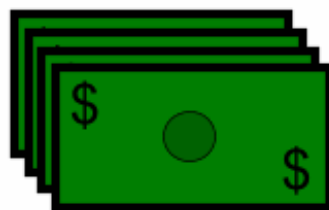
- There are 2 main sources of errors in modelling:
 1. Errors due to **Bias** (underfitting)
 2. Errors due **Variance** (overfitting)



Why Bagging



Why Bagging



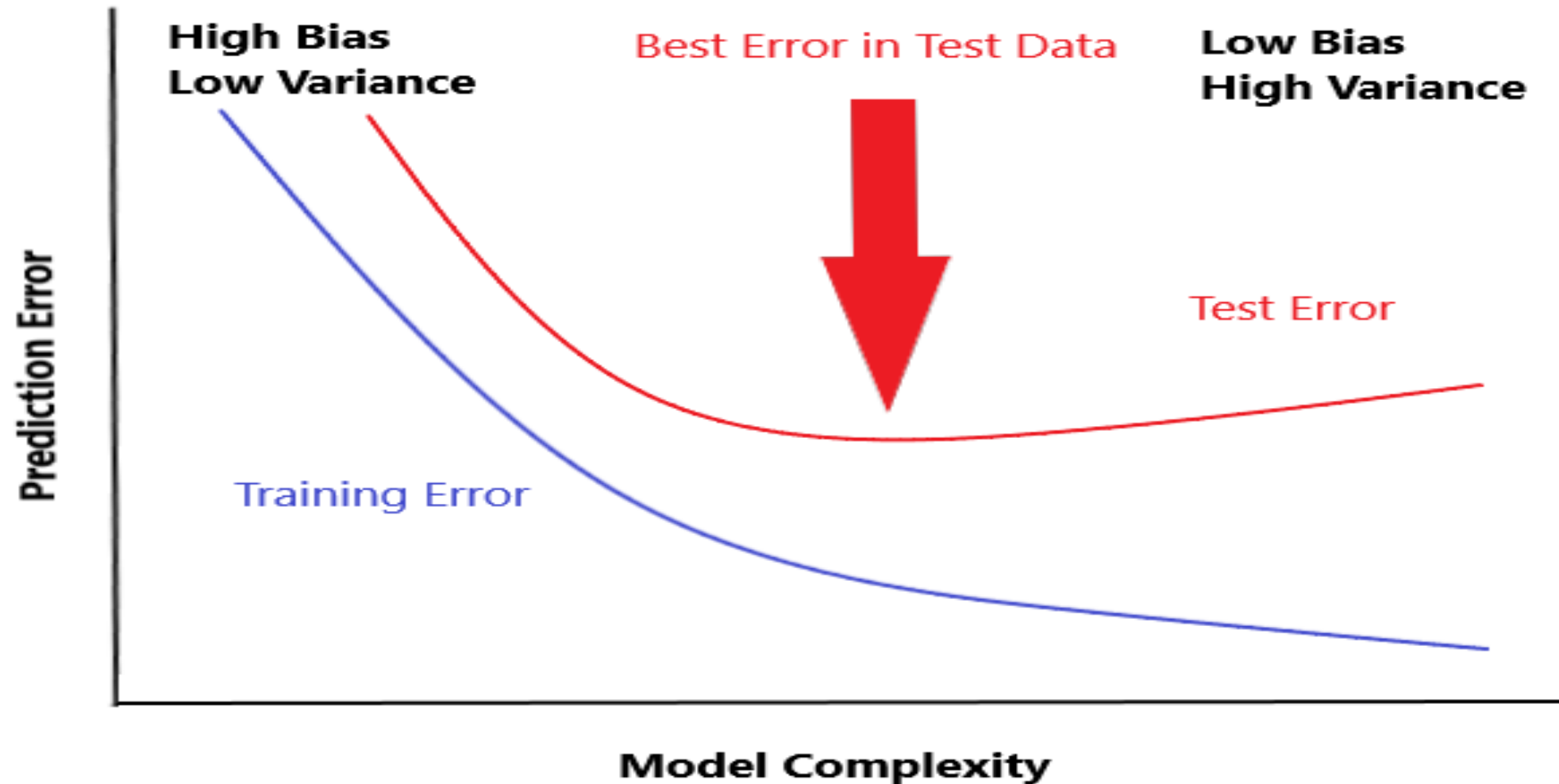
Why Bagging



Jon



Why Bagging



Parameters that control bagging?

- Changing the seed
- Row (Sub) sampling or Bootstrapping At the same time, you can run a model with less rows or you could use bootstrapping
- Shuffling
- Column (Sub) sampling
- Model-specific parameters
- Number of models (or bags)
- (Optionally) parallelism



Examples of bagging

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 predictions  
#take average of predictions  
bagged_prediction/= bags
```



Examples of bagging

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 predictions  
#take average of predictions  
bagged_prediction/= bags
```



Examples of bagging

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 predictions  
#take average of predictions  
bagged_prediction/= bags
```



Examples of bagging

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 predictions  
#take average of predictions  
bagged_prediction/= bags
```



Examples of bagging

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 predictions
#take average of predictions
bagged_prediction/= bags
```



Examples of bagging

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 predictions
#take average of predictions
bagged_prediction/= bags
```

