**To overcome the issue of overfitting and underfitting we use holdout method but sometimes we don’t have much data to perform even holdout method so there are many method like**

**1)k-fold cross validation (here k is user-define) (we model is over or underfitting)**

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation

Cross-validation is usually used in machine learning for improving model prediction when we don't have enough data to apply other more efficient methods like the 3-way split (train, validation and test) or using a holdout dataset **its especially use for when we have small data**

**Steps involved in k-fold cross validation**

**First**  step is suppose you have chosen k as 10 and you have 10 parts of dataset now what you will do you pick any one part of the dataset and that dataset is your testing dataset and now **secondly** remaining k-1 part dataset is now your training dataset now you will take every dataset as testing dataset and remaining is to training dataset and **lastly**  take out the average test results from it

from sklearn.model\_selection import **cross\_val\_score**

**parametric and non parametric model(for more info. Refer basic python docs )**

**decision tree is really use for**

classification

outlier and anomaly detection

feature engineering

decision tree are really high variance(overfitting) to overcome this issue we will use

depending on k it will split the dataset for us now it will select the split data depending on randomly for test and training the data

technique that evaluates statistics of a given population by testing a dataset by replacing the sample.

The purpose of those evaluation should be low variance and high biased optimal level for the model so what we are doing is we are taking repeated data itself to train the model but combination is different we can use in any learning algorithm but mostly use for decision tree

And the every split have mean standard deviation should match with the population

For feature selection in decision tree for sampling in bootstrap method should be root of n/2 where n is number of feature

So what random forest will do it will aggregate the result which come most of the time

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Random Forest is a supervised machine learning algorithm made up of decision trees. Random Forest is used for both classification and regression—for example, classifying whether an email is “spam” or “not spam”

3)**bootstrap method(we used mainly for decision tree)(when you have high variance)**

**But why bootstrap method came**

Lets understand why bootstrapping has come in case suppose you have done a experiment and found out result is bad but there are external factor also that they are responsible so you have done the experiment several times and then you have found out the bad results has come only rare time and good results has come several times now to find out which one is rare and which one several so this will be time consuming so that’s why here bootstrap method come

**Why only bootstrap method ?**

We can calculate the standard deviation or confidence interval of mean without regardless of bootstrap method then why bootstrap ? because we can apply it to any statistic to create a histogram of what might happen if repeated the experiment a bunch of time

**Steps involved in bootstrap**

1)create bootstrap dataset (but how to create)

First take the dataset and repeat so datapoint thrice or twice or once simple you have created your bootstrap dataset (the datapoint that we have created by doing repeating is called **sampling with replacement**)

2)calculate mean(depends on domain expert what he/she wants to calculate)

3)and keep track using histogram or anything (note:-1,23method process called **bootstrapping** )

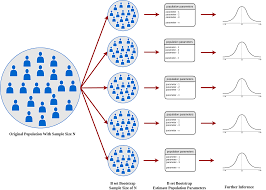
**Usually bootstrapping is done 1000 and more time in computer**

so what we will do is we will create forest out of decision tree means we will create multiple decision tree

so how going to work is

depending on  **k**  it will split the dataset for us now it will select the split data depending on randomly for test and training the data

technique that evaluates statistics of a given population by testing a dataset by replacing the sample.



The **purpose**  of those evaluation should be **low variance** and **high** **biased** optimal levelfor the modelso what we are doing is we are taking repeated data itself to train the model but combination is different we can use in **any learning algorithm** but mostly use for decision tree

And the every split have mean standard deviation should **match** with the population

For feature selection in decision tree for sampling in bootstrap method should be root of n/2 where n is number of feature

So what random forest will do it will aggregate the result which come most of the time

**Random forest**

**Small note:-how do we infer the data ?**

By descriptive statistics ,mean, median, standard deviation should be similar to population

**Decision tree is made on basis of entropy score(entropy is most relevant feature)**

**But why random forest ?**

Decision tree work great with the data used to create them but they are not flexible when it comes to **classifying new sample** so to curb this problem we will come up with **random forest**

**So how random forest works ?**

1)create bootstrap dataset(randomly select item and repeat more than once(pick anyone datapoint to repeat))**note:- the datapoint will not include the entries called out of bag dataset**

2)create a decision tree using the bootstrapped dataset but use a random subset of variables (or column) at each step.

-make decision tree(pick optimal variable) then built out of it

3)now go back to step 1 and repeat make a new bootstrapped dataset and build a tree considering a subset of variable(column) at each step

(ideally its done 100s of time)

**Now we created the random forest but now how to used it**

Now take the y variable test the each random forest for the y variable now whoever got the highest majority ans we will win

**Terminology**

**Bootstrapping** the data plus using the **aggregate** to make a decision is called **bagging**

so what we will do is we will create forest out of decision tree means we will create multiple decision tree

**out of Bag dataset**

the entries that didn’t make it into the bootstrap dataset

**so how going to work is**

the entries that didn’t make that dataset(out of the bag dataset) that will be tested by our random forest

ultimately we can measure how accurate our random forest is by the proportion of the out of bag sample that were correctly or incorrectly classified by the random forest

the proportion of out of bag samples that were incorrectly classified is the **out of bag error**

**what should we do if accuracy is not good**

then we should change the variable that we took in initial may be we should take variable 3 instead of 2 or try bunch of other variables also

**now we had got how to**

1)build a random forest

2)use a random forest

3)estimate the accuracy of the random forest

4)change the number of variable used per step1 1(if accuracy is not good)

**Random forest consider 2 types of missing data**

1)missing data in original dataset used to create the random forest

2)missing data in new sample that you want to categorize(new sample)

For the 1st missing values we should guess the values manually by analyzing the data (those who have numeric missing values should take median rest of all dataset and should apply to it) then run that dataset in random forest

But to confirm that which **sample is similar** to which missing values we have to…..

**For categorical missing values**

1)built random forest and run all of the data down all of the tree

After doing first step now plot proximity matrix which sample is similar and which sample is not

Then calculate the frequency of missing values column **x** proximity matrix values = the majority of highest value will win (we will impute that value)

**For numerical**

We will take that proximity values of categorical variable **X** the numerical values = the new value for the missing column

We should do this process again until we get the closet accurate value

**How to calculate the distance plot out 0f proximity matrix**

**Now how to calculate the proximity value to distance of closet similar value**

1-proximity values = the closet distance of similar values

Now by the distance plot we can plot heat map scatter plot for most similar values

Now will come to next 2nd method of imputing missing values

**For categorical**

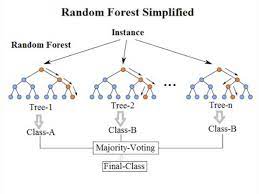
We will first impute all possible values that can be

**For numerical**

Do what we have did in 1st method (take all the median of the value and impute it)

**Then finally**

Run the random forest for the missing data sample and whoever come in the majority we will win



**There is advance version of random forest is stacking (stack)**

We build the model with different algorithm like knn,svm,dt,lr and take out the most similar result (aggregation)