

Learning Curve in Machine Learning in Python

- Learning curve is very famous among the data scientist.
- Learning curve shows the efficiency and way your machine learning model learns.
- So,the learning is widely used in diagnostics tool in machine learning for the algorithm that learning from training datasets incrementally.
- That means we increase our datasets by some steps and then we see the performance of our model.
- Model can be evaluate on the training datasets and on hold out validation datasets after each update during training.
- Plots the measure accuracy/Performance metric that can shown as like -><https://www.youtube.com/watch?v=2Bkp468u12V&list=PL8t186t>
- In that we will get to know that the performances of accuracy on the trainng and testing datasets (cross validation) as we are increasing the number of observations That's means the size of our data.
- If we increase the size of the datasets how fit in overall performance of your model.
- That means whether it is desirable to increase the data to improve your performance or you need to work on your model.
- The metrics which we used to evaluate learning curve that is known as score actually.That could be accuracy score or we can say loss of your model.
- We can plot our model either accuracy or loss.
- It is more common to use score that minimizing such as loss or error,where by better score indicates more learning and value of 0.0 indicates that the training datasets was learned perfectly and no mistake was made.
- No mistakes was made that means the low bias when your model having really very low bias in that case your model having the high variance which is not desirable.

Variance And Bias Trade Off.

- In this case of bias and variance.
- The variance is inversly proportional to the Bias.When we increases the bias then the varinace will be decreases but the error will be high that time.
- Similarly for the When we decreases the bias then the variance will be increases but the error will be high that time also.
- During this period we have to find the some sweet spot there we can minimized the total error.Not only the bias not only the variance but we minimize there total error.
- This is known as Bias and Variance Trade Off.
- So,we need to find such place where we can get the low variance and low base and low error with optimum complexity.

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
matplotlib inline

In [3]: from sklearn import datasets
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import learning_curve

In [4]: cancer = datasets.load_breast_cancer()

In [5]: cancer.keys()

Out[5]: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename'])

In [6]: print(cancer.DESCR)

.. _breast_cancer_dataset

Breast cancer wisconsin (diagnostic) dataset
-----

**Data Set Characteristics:**

 :Number of Instances: 569

 :Number of Attributes: 30 numeric, predictive attributes and the class

 :Attribute Information:
    = radius (mean of distances from center to points on the perimeter)
    = texture (standard deviation of gray-scale values)
    = perimeter
    = area
    = compactness (local variation in radius lengths)
    = compactness (perimeter2 / area - 1.0)
    = concavity (severity of concave portions of the contour)
    = concave points (number of concave portions of the contour)
    = symmetry
    = fractal dimension ("coastline approximation" - 1)

    The mean, standard error, and "worst" or largest (mean of the three
    worst/largest values) of these features were computed for each image,
    resulting in 30 features.  For instance, field 0 is Mean Radius, field
    10 is Radius SE, field 20 is Worst Radius.

    = class:
        = WDBC-Malignant
        = WDBC-Benign

 :Summary Statistics:

-----
              Min      Max
-----
radius (mean):      6.981  28.11
texture (mean):      9.71   39.28
perimeter (mean):   43.79  188.5
area (mean):        143.5  2501.0
smoothness (mean):  0.053  0.163
compactness (mean): 0.019  0.345
concavity (mean):    0.0    0.421
concave points (mean): 0.0    0.201
symmetry (mean):     0.106  0.304
fractal dimension (mean): 0.05  0.097
radius (standard error): 0.112  2.873
texture (standard error): 0.36   4.885
perimeter (standard error): 0.757  21.98
area (standard error):  6.802  542.2
smoothness (standard error): 0.002  0.031
compactness (standard error): 0.002  0.135
concavity (standard error): 0.0    0.396
concave points (standard error): 0.007  0.053
symmetry (standard error): 0.008  0.079
fractal dimension (standard error): 0.001  0.03
radius (worst):        7.93   36.04
texture (worst):       12.02  49.54
perimeter (worst):     58.41  251.2
area (worst):          183.2  4254.0
smoothness (worst):    0.071  0.223
compactness (worst):   0.027  1.058
concavity (worst):     0.0    1.252
concave points (worst): 0.0    0.293
symmetry (worst):      0.156  0.664
fractal dimension (worst): 0.055  0.208
-----

:Missing Attribute Values: None

:Class Distribution: 212 - Malignant, 357 - Benign

:Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

:Donor: Nick Street

:Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
https://goo.gl/U2Uwz2

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass.  They describe
characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/qpso-dataset/machine-learn/WDBC/
..
topic:: References

- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction
  for breast tumor diagnosis, ISBT/SPIE 1993 International Symposium on
  Electronic Imaging: Science and Technology, volume 1905, pages 861-870,
  San Jose, CA, 1993.
- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and
  prognosis via linear programming. Operations Research, 43(4), pages 570-577,
  July-August 1995.
- W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques
  to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994)
  163-171.
```

In this kind of datasets we having the the 569 rows and 30 features columns.Here is the two classes.In the attributes information we will get the feature parameter.

```
In [7]: x = cancer.data
        y = cancer.target

In [8]: #569 rows and 30 feature columns.
        x.shape

Out[8]: (569, 30)
```

Learning curve.

- Determines cross-validated training and test scores for different training set sizes.
- A cross-validation generator splits the whole dataset k times in training and test data. Subsets of the training set with varying sizes will be used to train the estimator and a score for each training subset size and the test set will be computed. Afterwards, the scores will be averaged over all k runs for each training subset size.

Returns

- train_sizes_abs : array, shape (n_unique_ticks,), dtype int
Numbers of training examples that has been used to generate the learning curve. Note that the number of ticks might be less than n_ticks because duplicate entries will be removed.
- train_scores : array, shape (n_ticks, n_cv_folds)
Scores on training sets.
- test_scores : array, shape (n_ticks, n_cv_folds)
Scores on test set.

```
In [13]: train_sizes,train_scores,test_scores = learning_curve(RandomForestClassifier(),x,y,cv=10,n_jobs=1,scoring="acc")
```


