Probability and Statistics in ML :

https://towardsdatascience.com/probability-and-statistics-explained-in-the-context-of-deep-learning-ed1509b2eb3f

==========

Mean: Mean is the arithmetical average value of the data. numpy docs

$$\text{Mean} = \mu = \frac{x1 + x2 + x3 + \ldots + xn}{n}$$

```
import numpy as np
a = np.array([[1,2,3,4,5,6]])
np.mean(a,axis=0)
```

==========

Median: It is the middle value of the data. numpy docs.

```
np.median(a,axis=0)
```

==========

Mode: It is the frequently occuring value of the data. scipy docs.

```
import numpy as np
from scipy import stats
stats.mode(a)
```

==========

```
>>> from sklearn.metrics import classification_report
>>> y_true = [0, 1, 2, 2, 2]
>>> y_pred = [0, 0, 2, 2, 1]
>>> target_names = ['class 0', 'class 1', 'class 2']
>>> print(classification_report(y_true, y_pred, target_names=target_names))
              precision    recall  f1-score   support

     class 0       0.50      1.00      0.67         1
     class 1       0.00      0.00      0.00         1
     class 2       1.00      0.67      0.80         3

   micro avg       0.60      0.60      0.60         5
   macro avg       0.50      0.56      0.49         5
weighted avg       0.70      0.60      0.61         5
```

==========

Mean Absolute error: It is the average of difference between the original and predicted values.

Mean squared error: it is the average of square of difference between the original and predicted values.

$$\text{Mean absolute error} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \overline{y_i}|$$

$$\text{Mean squared error} = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \overline{y_i}\right]^2$$

==========

=

Expected value: of some variable X with respect to some distribution P(X=*x*) is the mean value of X when *x is drawn from P*.

$$\text{Expectation} = E(x) = \sum_x x.\, P(X = x)$$

==========
=

Variance: It is the measure of variability in the data from the mean value.

$$\text{Variance} = \sigma^2 = \sum_{i=1}^{n} \frac{(xi - \mu)^2}{n}$$

```
import numpy as np
a = np.array([[1,2,3,4,5,6]])
np.var(a)
```

==========
=

Standard deviation:It is the square root of variance. numpy docs.

$$\text{Standard deviation} = \sigma = \sqrt{\sum_{i=1}^{n} \frac{(xi - \mu)^2}{n}}$$

```
import numpy as np
np.std(a)
```

==========

Co variance: It shows how to two variables are linearly related to each other.

```
import numpy as np
np.cov(a)
```

==========

Uniform Distribution: It is the simplest form of continuous distribution, with every element of the sample space being **equally likely**.

Uniform distribution:

$$\text{pdf } f(x) = \frac{1}{b-a} \text{ , x belongs to [a,b]}$$

$$\text{mean E(X)} = \frac{(a+b)}{2}$$

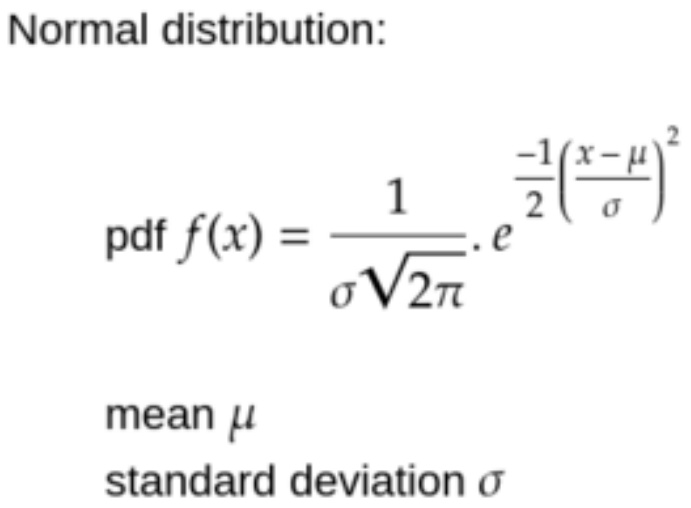
$$\text{Variance var(X)} = \frac{(b-a)^2}{12}$$

```
import numpy as np
```

np.random.uniform(low=1, high=10,size=100)

==========

Normal distribution: “Order from Chaos”
It is **the most important of all distributions**.Also known as Gaussian distribution.I




```
import numpy as np
mu = 0
sigma = 1
np.random.normal(mu,sigma,size=100)
```

==========

Standardization and Normalization :

In machine learning, you often encounter the word ‘ normalization’ and ‘standardization’. the process which we did above to obtain standard normal distribution is called standardization whereas the process of restricting the range of dataset **values between 0.0 to 1.0 is called as normalization**. However, these terms are often interchanged.

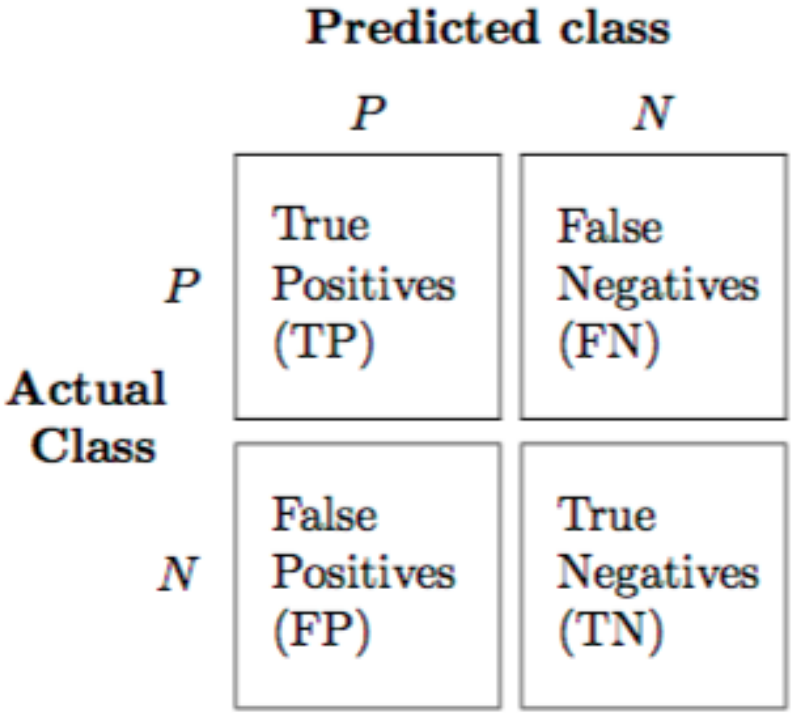
$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X_{new} = \frac{X - \mu}{\sigma}$$

```
from sklearn.preprocessing import StandardScaler
import numpy as np
data = np.array([1,2,3,4,5])
scaler = StandardScaler()
scaler.fit_transform(data)
```

==========
=

Confusion matrix: It is a matrix containing the TP, FP,TN and FN values.



confusion matrix

```
from sklearn.metrics import confusion_matrix
```

```
y_true = [2, 0, 2, 2, 0, 1]
y_pred = [0, 0, 2, 2, 0, 2]
confusion_matrix(y_true, y_pred)
```

==========

Precision and recall: So we go for two other metrics-precision and recall.

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

Precision tells you **how many of the selected objects were correct**
Recall tells you **how many correct objects were selected.**
In the above example, both precision and recall are 0.0 This indicates that the model is extremely poor.

==========

F1 score: It is the harmonic average of precision and recall.

$$\text{F1 score} = 2.\frac{precision.recall}{precision + recall}$$

F1 score of 0 means worst and 1 means best. By using this, we can resolve the chaotic behaviour of accuracy metric.
Sklearn has a classification_report function that you can invoke to get the

precision, recall, f1 score.

==========

Mean Absolute error: It is the average of difference between the original and predicted values.

Mean squared error: it is the average of square of difference between the original and predicted values.

$$\text{Mean absolute error} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \overline{y_i}|$$

$$\text{Mean squared error} = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \overline{y_i}\right]^2$$

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
mean_squared_error(y_true, y_pred)
```

==========

==========

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
```

```
y_pred = [2.5, 0.0, 2, 8]
mean_squared_error(y_true, y_pred)
```

==========

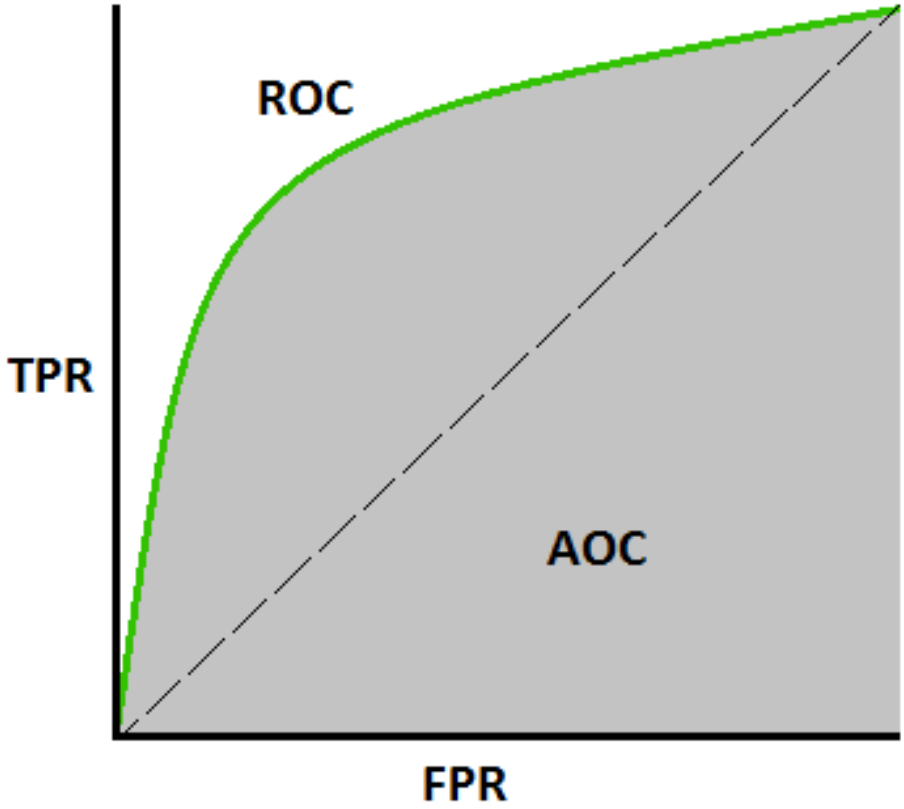
https://medium.com/swlh/recall-precision-f1-roc-auc-and-everything-542aedf322b9

Receiver operating characteristic(ROC) curve:The roc curve is a **graph showing the performance of classification models** like our digit recognizer example.It has two parameters—True Positive rate(TPR) and False Positive rate(FPR).**TPR is the same as recall and is also called as sensitivity. FPR is also 1-specificity.**

$$\text{TP rate} = \frac{TP}{TP + FN}$$

$$\text{FP rate} = \frac{FP}{FP + TN}$$

These two are plotted against each other to obtain the following graph(several values of plots are obtained by changing the classification threshold and predicting the results again repeatedly.).The area under this ROC curve is a measure of the accuracy.



Interpretation of area under the curve(AUC):When AUC=1.0 the model is best.When AUC=0.5 the model is worst.But if AUC=0.0 then the model is reciprocating the results.(Like classifying 1's as 0's and 0's as 1's).

==========

import numpy as np
from sklearn.metrics import roc_auc_score
y_true = np.array([0, 0, 1, 1])
y_scores = np.array([0.1, 0.4, 0.35, 0.8])
roc_auc_score(y_true, y_scores)

==========