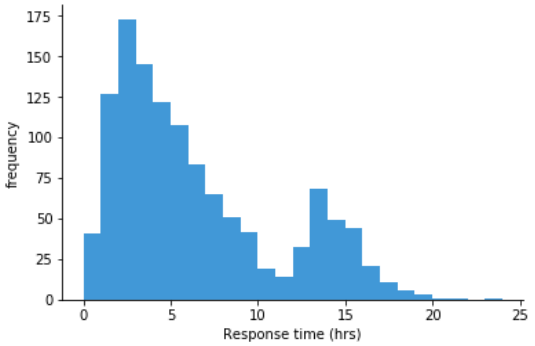
ML Interview Content

Probability & Statistics:

Histogram:

A histogram is an approximate representation of the distribution of numerical data.To construct a histogram, the first step is to "bin" (or "bucket") the range of values—that is, divide the entire range of values into a series of intervals—and then count how many values fall into each interval. The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent and are often (but not required to be) of equal size.

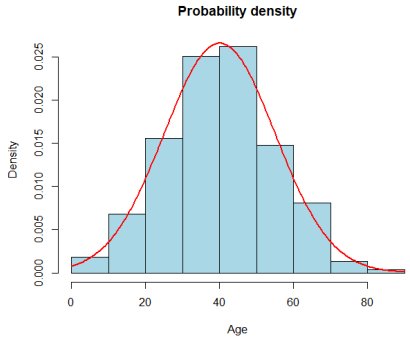
If the bins are of equal size, a rectangle is erected over the bin with height proportional to the frequency—the number of cases in each bin. A histogram may also be normalized to display "relative" frequencies. It then shows the proportion of cases that fall into each of several categories, with the sum of the heights equaling 1.



Probability Density Function(PDF):

Smoothed histogram. Can be found out using kernel density estimation.

Probability Density Function (PDF), or density of a continuous random variable, is a function whose value at any given sample (or point) in the sample space (the set of possible values taken by the random variable) can be interpreted as providing a relative likelihood that the value of the random variable would equal that sample. In other words, while the absolute likelihood for a continuous random variable to take on any particular value is 0 (since there is an infinite set of possible values to begin with), the value of the PDF at two different samples can be used to infer, in any particular draw of the random variable, how much more likely it is that the random variable would equal one sample compared to the other sample.



Cumulative Distribution Function(CDF):
CDF is the probability that a random variable, X, will take a value less than or equal to x.
CDF always lies between 0 and 1.

$$F_X(x) = \mathrm{P}(X \leq x)$$

Mean: (Or Arithmetic mean)
Mean is the average of all data points. Mean can be corrupted by a single value which can be very large or very small (outlier) that is different from the existing data points.
Mean is telling us about central tendency.

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Variance:
Measures the spread of the data from mean.

$$S^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$$

Standard deviation is the square root of Variance.
Variance/std dev can also be corrupted by one outlier.

Median:
In statistics and probability theory, the median is the value separating the higher half from the lower half of a data sample, a population, or a probability distribution. For a data set, it may be thought of as "the middle" value.

Median is not as affected by outliers as the mean.

Percentiles & Quantiles:
Percentile (or a centile) is a score below which a given percentage of scores in its frequency distribution falls (exclusive definition) or a score at or below which a given percentage falls (inclusive definition). For example, the 50th percentile (the median) is the score below which (exclusive) or at or below which (inclusive) 50% of the scores in the distribution may be found.
25th,50th,75th,100th Percentiles are called quartiles.

Median Absolute Deviation:
Median of the absolute deviations between each point and median. It’s a measure of spread like standard deviation.Robust to outliers like median.

IQR:
3rd quartile value(75th percentile) - 1st quartile value(25th percentile).
50% of the data points lies between 1st and 3rd quartiles.

Random Variable: It is the variable that can take any value from a set of possible values.
Ex:- Roll of a dice is a random variable that can take values 1,2,3,4,5,6. Flip of a coin can take either head or tail.

If the random variable can only take numbers/categories it is called **discrete random variable.**
Ex:- roll of a dice.

If the random varible can take any real value it is called **continuous random variable.**
Ex:- height of a randomly picked student.

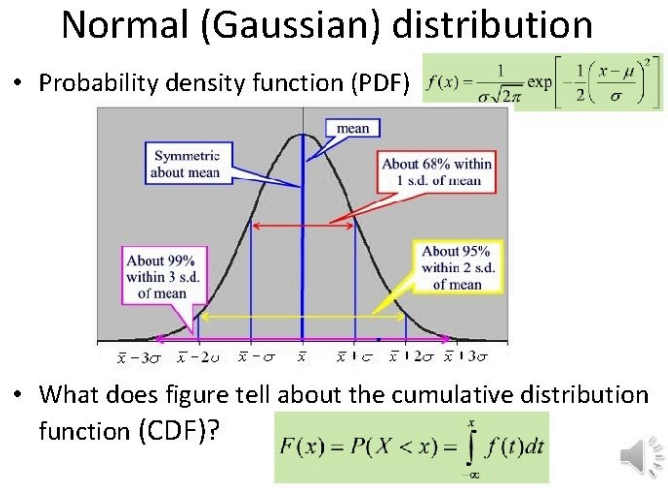
Outlier: It is a data point which is significantly different from the other points.

Population:
Ex: Set of all people in the world.
Sample:
Ex: 10,000 people randomly selected.

For example if we have to compute the mean height of all the people in the world, it is impossible to do that. So, we can take a random sample of people and estimate the mean height of the entire population using the sample heights.
As the sample size increases, the sample mean will become closer to population mean.

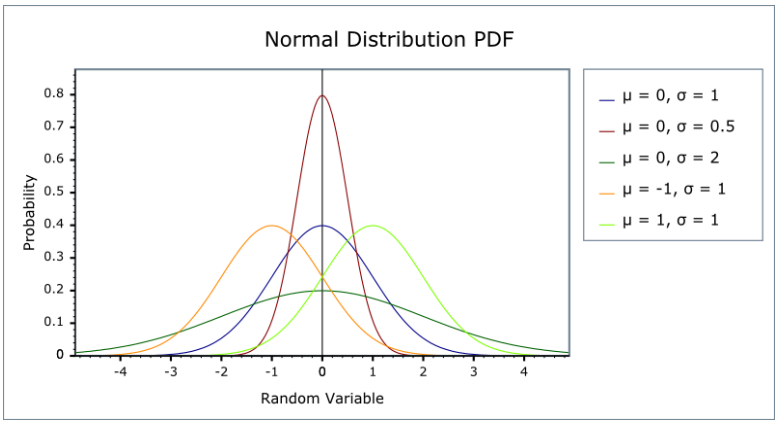
Distributions are simple models.

Gaussian(Normal) distribution:
PDF of a Gaussian distribution:

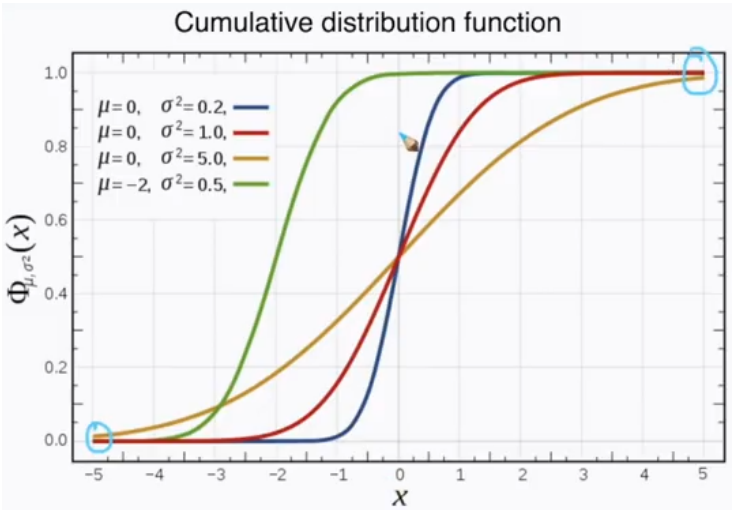


Natural phenomena usually tends to follow Gaussian distribution/normal distribution.
Ex:- Heights,weights,sepal lengths e.t.c.,

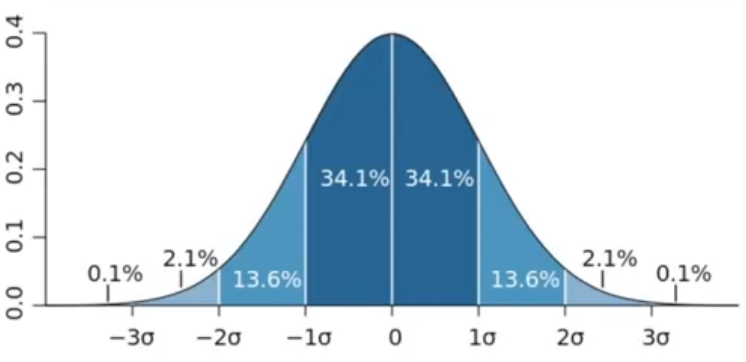
PDF for various means and std devs:



CDF of Gaussian:



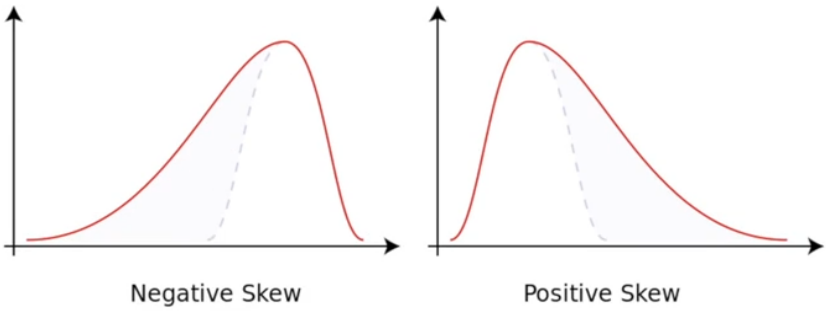
68-95-99.7Rule:



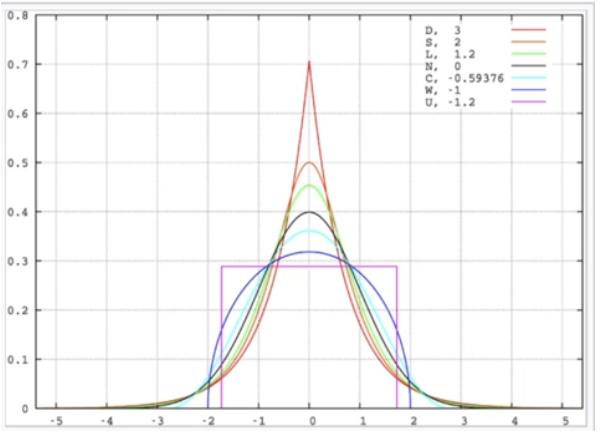
68.2% of the data lies within mean+/-1*std.
95.4% of the data lies within mean+/-2*std.
99.7% of the data lies within mean+/-3*std.

Gaussian distribution is a symmetric distribution.
Negative Skew is also called as **Left skew/left tailed** distribution as it has long tail on left side.
Positive Skew is also called as **Right skew/Right tailed** distribution as it has long tail on right side.

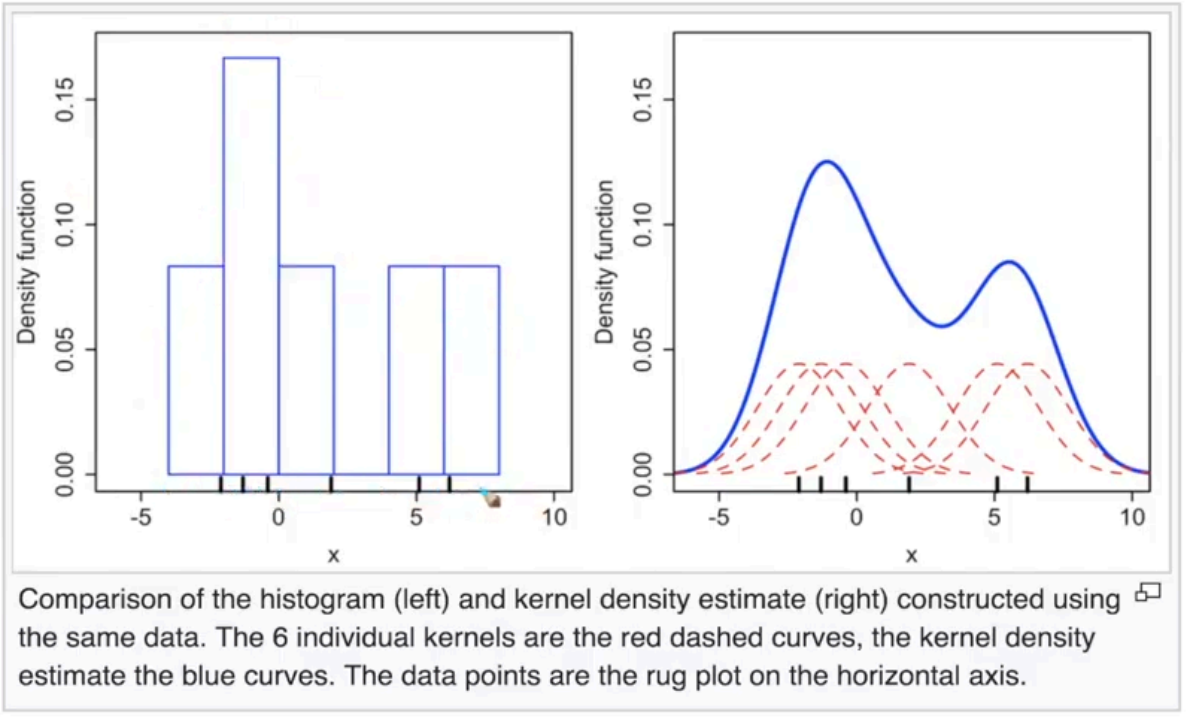
Skewness tells about how far away/how dissimilar the distribution is from the symmetric distribution.



Kurtosis measures how peaked our distribution is. Gaussian distribution has kurtosis of 0.



Standard Normal Variate:
It's a gaussian distributed random variable with a mean of zero and variance of 1.
Kernel Density Estimation:
How to convert a PDF from histogram.

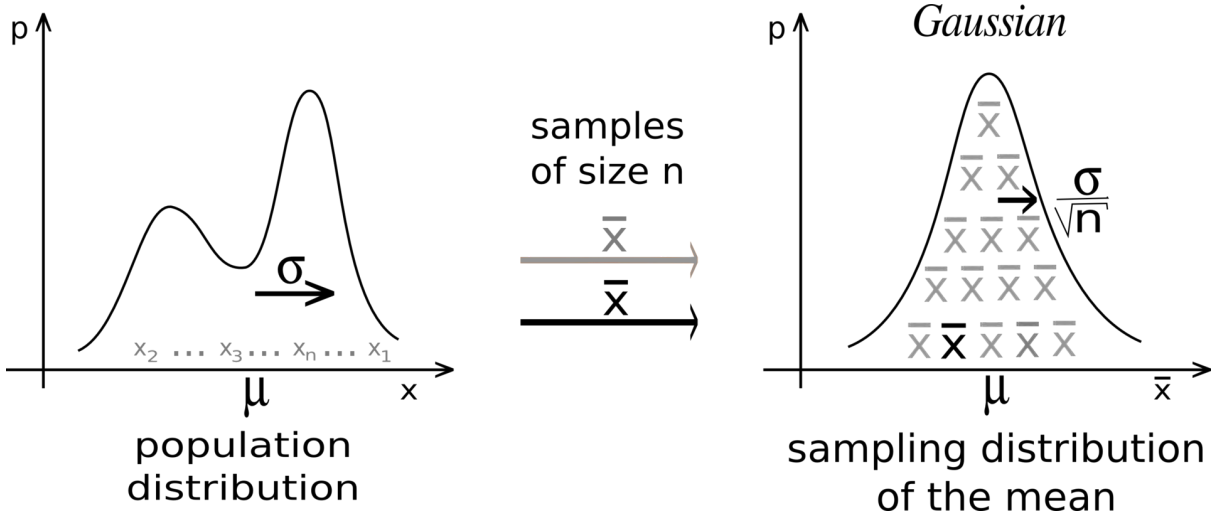


Comparison of the histogram (left) and kernel density estimate (right) constructed using the same data. The 6 individual kernels are the red dashed curves, the kernel density estimate the blue curves. The data points are the rug plot on the horizontal axis.

Construct a gaussian kernel by considering each point the Histogram as (mean) center of gaussian curve. At any point, sum up the values of the kernels to create the kde plot(Blue plot in the image above). Variance or Bandwidth of each kernel is selected to make the PDF as smooth as possible. (small bandwidth -> jagged pdf, large bandwidth-> huge deviation from the actual scenario)

Sampling Distribution and Central Limit Theorem:

Let's assume Random variable X has a distribution not necessarily gaussian(Population distribution).
If we take a random sample of size n, S1.
Similarly We'll sample S2 randomly independent of S2.
Let's say we have m samples.
If we calculate the means of each sample, the distribution of means is called the sampling distribution of sample means.
Central Limit Theorem:
If X is a distribution with finite mean and variance,
Let's assume we have m random samples of size n.
CLT says that the sampling distribution of sample means always follow gaussian distribution of mean equal to population mean and variance as variance of population divided by n.



n should be greater than equal to 30.

Quantile-Quantile Plot (Q-Q Plot):

This plot helps us answer

1. whether the given distribution is normal.
2. whether the random variables, A & B have the same distribution.

Let's consider case-1 and see how to generate a Q-Q plot.
Calculate the quantiles of the distribution.
Generate a normal distribution with mean 0 and standard deviation 1 and calculate quantiles for that as well.

Plot the corresponding quantiles of the given distribution(y-axis) and standard normal distribution (x-axis). If the resultant plot is a straight line then the given distribution is normal else it's not.
Same thing applies for case-2 as well.

Chebyshev's inequality

In probability theory, Chebyshev's inequality, also known as "Bienayme-Chebyshev" inequality guarantees that, for a wide class of probability distributions, NO MORE than a certain fraction of values can be more than a certain distance from the mean.

Specifically, no more than 1/k2 of the distribution's values can be more than k standard deviations away from the mean(or equivalently, at least 1-1/k2 of the distribution's values are within k standard deviations of the mean).

Now, let's formally define Chebyshev's inequality:

Let X be a random variable with mean μ with a finite variance σ2, then for any real number k>0,

P(| X-μ | < kσ) ≥ 1-1/k**2

OR

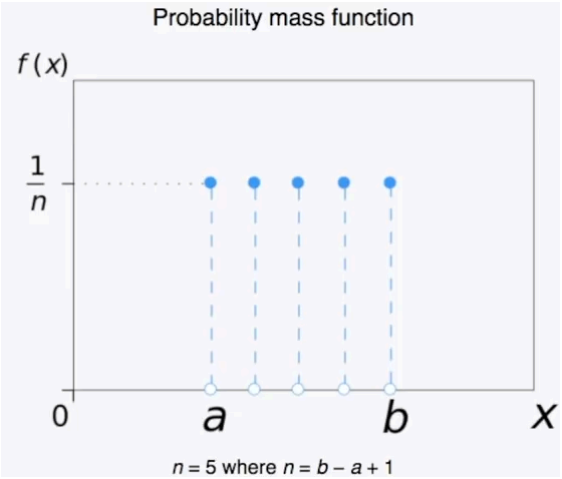
P(| X-μ | ≥ kσ) ≤ 1/k**2

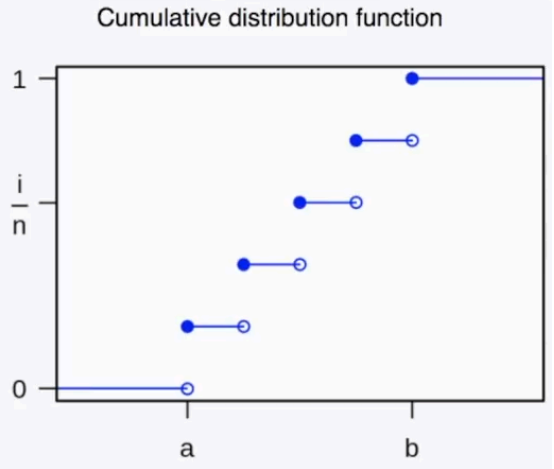
For any distribution,

$$\Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

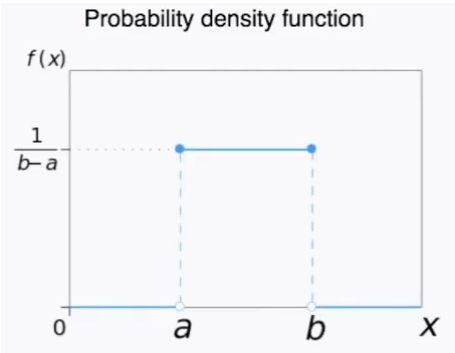
Uniform Distribution

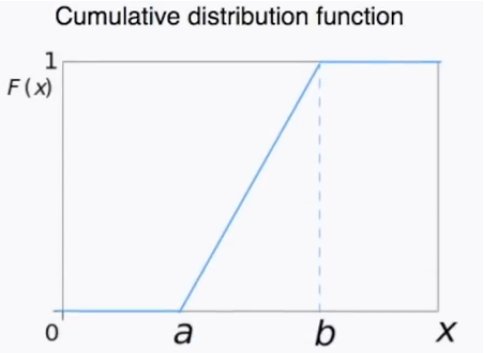
Discrete uniform distribution
All outcomes are equiprobable.





Continuous uniform distribution





Bernoulli Distribution & Binomial distribution

Both are discrete distributions.
Bernoulli distribution is a distribution with 2 outcomes. Ex: Coin toss.

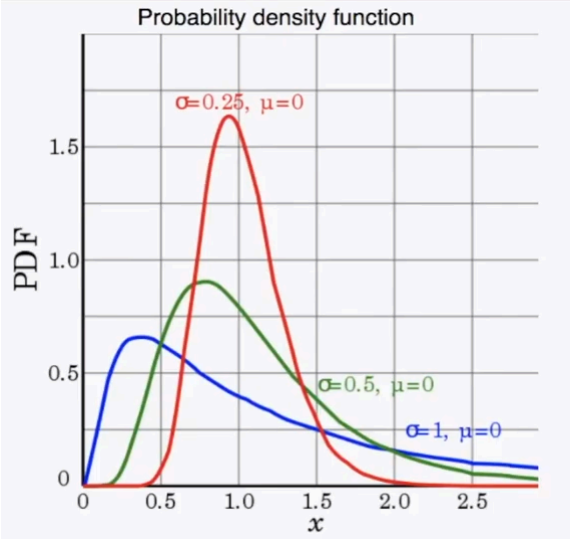
Parameters	$0 < p < 1, p \in \mathbb{R}$
Support	$k \in \{0, 1\}$
pmf	$\begin{cases} q = (1-p) & \text{for } k = 0 \\ p & \text{for } k = 1 \end{cases}$
CDF	$\begin{cases} 0 & \text{for } k < 0 \\ 1-p & \text{for } 0 \leq k < 1 \\ 1 & \text{for } k \geq 1 \end{cases}$
Mean	p
Median	$\begin{cases} 0 & \text{if } q > p \\ 0.5 & \text{if } q = p \\ 1 & \text{if } q < p \end{cases}$
Mode	$\begin{cases} 0 & \text{if } q > p \\ 0, 1 & \text{if } q = p \\ 1 & \text{if } q < p \end{cases}$
Variance	$p(1-p) (= pq)$

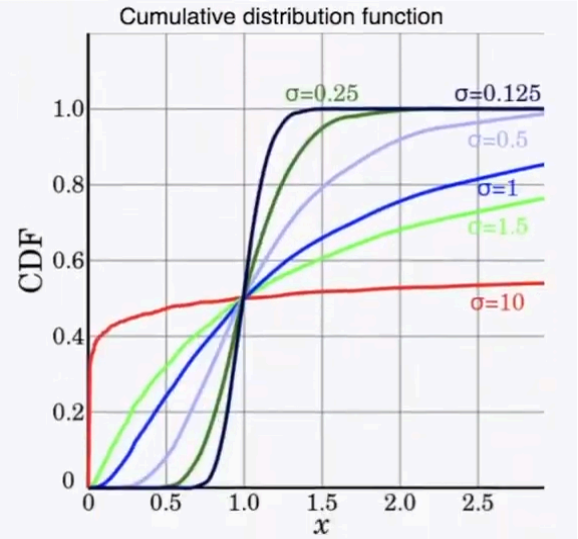
Binomial distribution is applicable when a Bernoulli experiment is conducted multiple times (n times). Ex: Coin is tossed for 15times. What is the probability that we get 13 heads?

Notation	$B(n, p)$
Parameters	$n \in \mathbb{N}_0$ — number of trials $p \in [0,1]$ — success probability in each trial
Support	$k \in \{0, \ldots, n\}$ — number of successes
pmf	$\binom{n}{k} p^k (1-p)^{n-k}$
CDF	$I_{1-p}(n-k, 1+k)$
Mean	np
Median	$\lfloor np \rfloor$ or $\lceil np \rceil$
Mode	$\lfloor (n+1)p \rfloor$ or $\lceil (n+1)p \rceil - 1$
Variance	$np(1-p)$

Log-Normal Distribution

Random variable X is considered to be log-normal if ln(X) is normally distributed.



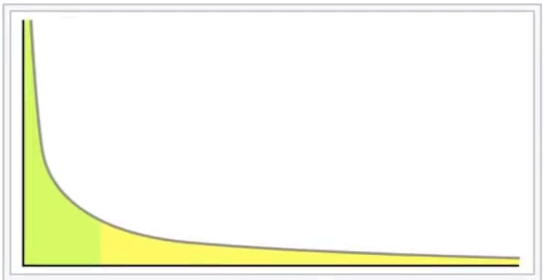


Applications:

1. The length of comments posted in Internet discussion forums follows a log-normal distribution.
2. Users' dwell time on online articles (jokes, news etc.) follows a log-normal distribution.
3. The length of chess games tends to follow a log-normal distribution.
4. Onset durations of acoustic comparison stimuli that are matched to a standard stimulus follow a log-normal distribution.
5. Rubik's Cube solves, both general or by person, appear to follow a log-normal distribution.
6. Measures of size of living tissue (length, skin area, weight).
7. Certain physiological measurements, such as blood pressure of adult humans (after separation on male/female subpopulations).
8. In reliability analysis, the log-normal distribution is often used to model times to repair a maintainable system.

Power law distribution

Also called pareto distribution.



An example power-law graph, being used to demonstrate ranking of popularity. To the right is the long tail, and to the left are the few that dominate (also known as the 80–20 rule).

Log-log plot: Used to test for power law distribution. Log y- log x plot should follow a straightline.
Applications:

1. The sizes of human settlements (few cities, many hamlets/villages).
2. File size distribution of Internet traffic which uses the TCP protocol (many smaller files, few larger ones).
3. Hard disk drive error rates.
4. The values of oil reserves in oil fields (a few large fields, many small fields).
5. The length distribution in jobs assigned to supercomputers (a few large ones, many small ones).
6. The standardized price returns on individual stocks.
7. Sizes of sand particles.
8. The size of meteorites.
9. In hydrology the Pareto distribution is applied to extreme events such as annually maximum one-day rainfalls and river discharges.

Box-Cox Transform/Power law transform
Used to transform pareto distribution to gaussian distribution.

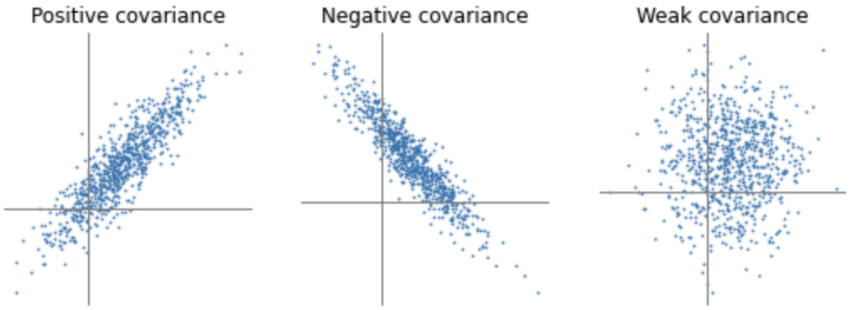
Covariance

It is the joint variability of 2 random variables.
Used to measure relationship between random variables.

$$cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

$cov_{x,y}$ = covariance between variable a and y
x_i = data value of x
y_i = data value of y
$\bar{x}$ = mean of x
$\bar{y}$ = mean of y
N = number of data values

Covariance is +ve if y increases with x and -ve if y decreases with increase in x.



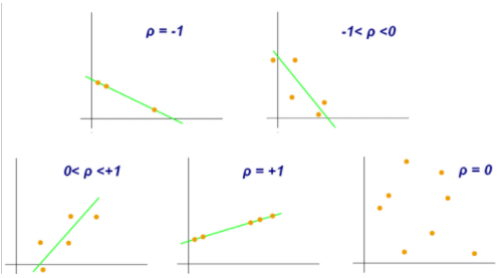
Sign of covariance is interpretable but not the magnitude. One problem with covariance is if we change the way we measure a parameter, the resultant covariance may not be same as earlier covariance.
Ex:- Covariance between height in cms and weight in kgs may not be equal to covariance between height measured in feet and weight measured in pounds.

Pearson's correlation coefficient

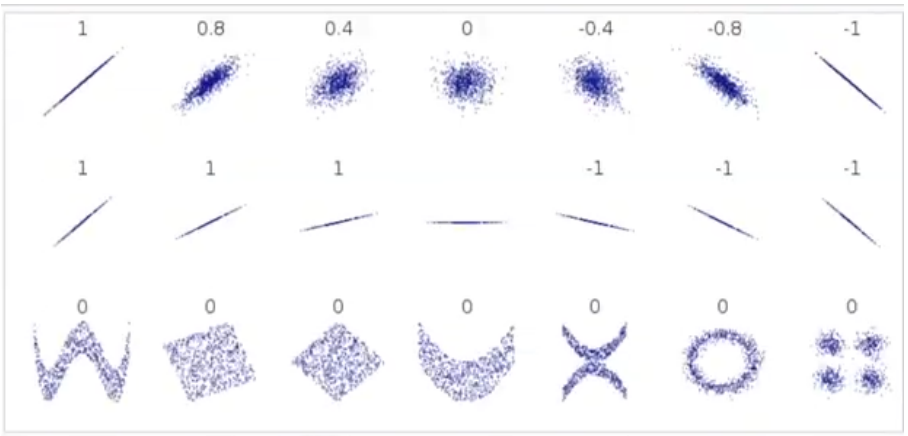
$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \quad \text{(Eq.1)}$$

where:

- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y



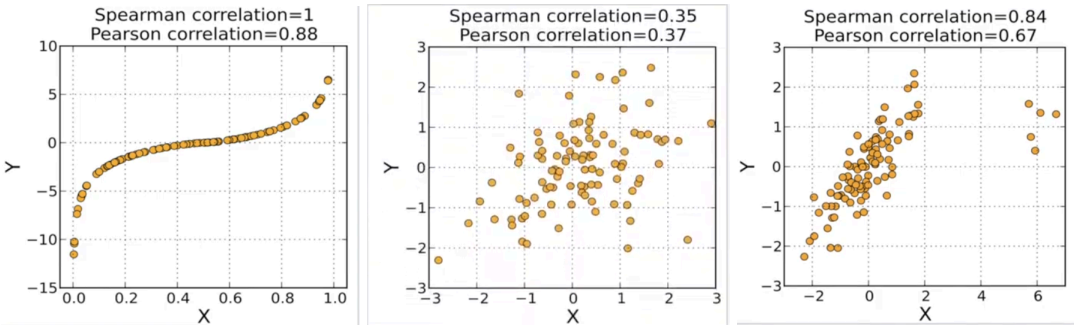
It is bound between -1 and +1. Unlike the plain covariance, the magnitude represents the strength of the relationship between X & Y. If the value is 0 then they are unrelated.



If we see the above image in the 3rd row, even though x and y are related somehow, since the relation is not linear, the pearson's correlation coefficient would be 0. i.e., pearson's correlation coefficient cannot capture non linear relationships.

Spearman rank correlation coefficient

It can capture non-linear relations as well.
Instead of looking at the actual values, it calculates the relationship between rank(X) and rank(Y).
i.e., if Y increases with X (be it any relationship) we'll have +ve coefficient with 1 and vice-versa.



Correlation doesn't imply causation

Just because X and Y are highly correlated doesn't mean that X causes Y or Y causes X.

Confidence Intervals

In statistics, a confidence interval (CI) is a type of estimate computed from the observed data. This gives a range of values for an unknown parameter (for example, a population mean). The interval has an associated confidence level chosen by the investigator. For a given estimation in a given sample, using a higher confidence level generates a wider (i.e., less precise) confidence interval. In general terms, a confidence interval for an unknown parameter is based on sampling the distribution of a corresponding estimator.
For example we have heights of 100 people (sample) and we want to estimate the population mean height. We can take the average of these 100 people but it's still an approximation of the actual value. It is point estimate.
Confidence Interval can be stated as follows:
Population mean height lies between 160.7 and 175.7 with 95% probability. Here, 160.7 to 175.7 is the interval and the 95% is confidence. We are 95% confident that the mean lies in the said interval.

Computing the confidence interval given the underlying distribution.

Let's say that we know that the underlying distribution is gaussian distribution. We can calculate the sample mean and standard deviation and then we know that the 95% of the data lies between mean +/- 2*std.

Hypothesis Testing

The best way to determine whether a statistical hypothesis is true would be to examine the entire population. Since that is often impractical, researchers typically examine a random sample from the population. If sample data are not consistent with the statistical hypothesis, the hypothesis is rejected.

There are two types of statistical hypotheses.

- **Null hypothesis**. The null hypothesis, denoted by Ho, is usually the hypothesis that sample observations result purely from chance.
- **Alternative hypothesis**. The alternative hypothesis, denoted by H1 or Ha, is the hypothesis that sample observations are influenced by some non-random cause.

For example, suppose we wanted to determine whether a coin was fair and balanced. A null hypothesis might be that half the flips would result in Heads and half, in Tails. The alternative hypothesis might be that the number of Heads and Tails would be very different. Symbolically, these hypotheses would be expressed as

$$Ho: P = 0.5$$

$$Ha: P \neq 0.5$$

Suppose we flipped the coin 50 times, resulting in 40 Heads and 10 Tails. Given this result, we would be inclined to reject the null hypothesis. We would conclude, based on the evidence, that the coin was probably not fair and balanced.

Suppose we wish to test the null hypothesis H0: The mean life length of a bulb is 500 hours against the alternative
H1: The mean life length is > 500 hours.

Suppose we take a random sample of 50 bulbs and found that the sample mean is 520 hours. Should we accept H0 or reject H0 ?
We note that even though the population mean is 500 hours, the sample mean could be more or less. Similarly even though the population mean is > 500 hours, say 550 hours, even then the sample mean could be less than 550 hours.
Thus whatever decision we may make, there is a possibility of making an error. That is falsely rejecting H0 (when it should have been accepted) and falsely accepting H0 (when it should have been rejected). We put this in a tabular form as follows:

		Decision Taken	
		Accept H_0	Reject H_0
True State	H_0 is True	Correct Decision	Type I Error
	H_0 is False	Type II Error	Correct Decision

The probability of committing type I error is denoted by **α**,alpha. It is also referred to as the size of the test or the level of significance of the test.
The probability of committing Type II error is denoted by ***β***,beta.

The Critical region C is the set of values for which we reject the null hypothesis.
The complement of the Critical region is known as the acceptance region.

The probability of rejecting the null hypothesis when it is false is called the power of the test.
Thus the power of a test is
1- (probability of accepting the null hypothesis when it is false)
= 1- Probability of Type II error
= 1 -***β***

The power of a hypothesis test is affected by three factors.

- Sample size (*n*). Other things being equal, the greater the sample size, the greater the power of the test.
- Significance level (α). The lower the significance level, the lower the power of the test. If you reduce the significance level (e.g., from 0.05 to 0.01), the region of acceptance gets bigger. As a result, you are less likely to reject the null hypothesis. This means you are less likely to reject the null hypothesis when it is false, so you are more likely to make a Type II error. In short, the power of the test is reduced when you reduce the significance level; and vice versa.
- The "true" value of the parameter being tested. The greater the difference between the "true" value of a parameter and the value specified in the null hypothesis, the greater the power of the test. That is, the greater the effect size, the greater the power of the test.

Steps to conduct hypothesis testing:

1. State the hypothesis.
2. Formulate an analysis plan.
3. Analyse sample data.
4. Interpret results.

Applications:

1. Proportions
2. Difference between proportions

3. Regression slope
4. Means
5. Difference between means.
6. Difference between matched pairs.
7. Goodness of fit
8. Homogeneity
9. Independence.

Hypothesis test for a proportion:

1. Each sample point can result in just two possible outcomes. We call one of these outcomes a success and the other, a failure.
2. The sample includes at least 10 successes and 10 failures.
3. The population size is at least 20 times as big as the sample size.

Null: Proportion of success is equal to P
Alternate: Not equal to P.

Test method: Use the one-sample z-test to determine whether the hypothesized population proportion differs significantly from the observed sample proportion.

Standard deviation. Compute the standard deviation (σ) of the sampling distribution.
$\sigma = \sqrt{P * (1 - P) / n}$

where P is the hypothesized value of population proportion in the null hypothesis, and n is the sample size.
Test statistic. The test statistic is a z-score (z) defined by the following equation.
$z = (p - P) / \sigma$
where P is the hypothesized value of population proportion in the null hypothesis, p is the sample proportion, and σ is the standard deviation of the sampling distribution.
P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a z-score, use the Normal Distribution Calculator to assess the probability associated with the z-score.

If the sample findings are unlikely, given the null hypothesis, the researcher rejects the null hypothesis. Typically, this involves comparing the P-value to the significance level, and rejecting the null hypothesis when the P-value is less than the significance level.

Difference between two Proportions:

1. The sampling method for each population is simple random sampling.
2. The samples are independent.
3. Each sample includes at least 10 successes and 10 failures.
4. Each population is at least 20 times as big as its sample.

Set	Null hypothesis	Alternative hypothesis	Number of tails
1	$P_1 - P_2 = 0$	$P_1 - P_2 \neq 0$	2
2	$P_1 - P_2 \geq 0$	$P_1 - P_2 < 0$	1
3	$P_1 - P_2 \leq 0$	$P_1 - P_2 > 0$	1

The first set of hypotheses (Set 1) is an example of a two-tailed test, since an extreme value on either side of the sampling distribution would cause a researcher to reject the null hypothesis. The other two sets of hypotheses (Sets 2 and 3) are one-tailed tests, since an extreme value on only one side of the sampling distribution would cause a researcher to reject the null hypothesis.
When the null hypothesis states that there is no difference between the two population proportions (i.e., d = P1 - P2 = 0), the null and alternative hypothesis for a two-tailed test are often stated in the following form.

$$H_o: P_1 = P_2$$
$$H_a: P_1 \neq P_2$$

Test method: Use the two-proportion z-test (described in the next section) to determine whether the hypothesized difference between population proportions differs significantly from the observed sample difference.

Pooled sample proportion: Since the null hypothesis states that P1=P2, we use a pooled sample proportion (p) to compute the standard error of the sampling distribution.
p = (p1 * n1 + p2 * n2) / (n1 + n2)
where p1 is the sample proportion from population 1, p2 is the sample proportion from population 2, n1 is the size of sample 1, and n2 is the size of sample 2.

Standard error. Compute the standard error (SE) of the sampling distribution difference between two proportions.
SE = sqrt{ p * (1 - p) * [(1/n1) + (1/n2)] }
where p is the pooled sample proportion, n1 is the size of sample 1, and n2 is the size of sample 2.

Test statistic. The test statistic is a z-score (z) defined by the following equation.
z = (p1 - p2) / SE
where p1 is the proportion from sample 1, p2 is the proportion from sample 2, and SE is the standard error of the sampling distribution.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a z-score, use the Normal Distribution Calculator to assess the probability associated with the z-score.

If the sample findings are unlikely, given the null hypothesis, the researcher rejects the null hypothesis. Typically, this involves comparing the P-value to the significance level, and rejecting the null hypothesis when the P-value is less than the significance level.

Hypothesis test for mean:

1. The sampling method is simple random sampling.
2. The sampling distribution is normal or nearly normal.

Generally, the sampling distribution will be approximately normally distributed if any of the following conditions apply.

1. The population distribution is normal.

2. The population distribution is symmetric, unimodal, without outliers, and the sample size is 15 or less.
3. The population distribution is moderately skewed, unimodal, without outliers, and the sample size is between 16 and 40.
4. The sample size is greater than 40, without outliers.

Set	Null hypothesis	Alternative hypothesis	Number of tails
1	$\mu = M$	$\mu \neq M$	2
2	$\mu \geq M$	$\mu < M$	1
3	$\mu \leq M$	$\mu > M$	1

Test method: Use the one-sample t-test to determine whether the hypothesized mean differs significantly from the observed sample mean.

Standard error. Compute the standard error (SE) of the sampling distribution.
SE = s * sqrt{ (1/n) * [(N - n) / (N - 1)] }

where s is the standard deviation of the sample, N is the population size, and n is the sample size. When the population size is much larger (at least 20 times larger) than the sample size, the standard error can be approximated by:
SE = s / sqrt(n)

Degrees of freedom. The degrees of freedom (DF) is equal to the sample size (n) minus one. Thus, DF = n - 1.
Test statistic. The test statistic is a t statistic (t) defined by the following equation.
t = (x - μ) / SE

where x is the sample mean, μ is the hypothesized population mean in the null hypothesis, and SE is the standard error.
P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a t statistic, use the t Distribution Calculator to assess the probability associated with the t statistic, given the degrees of freedom computed above.

Hypothesis test for Difference between means:

1. The sampling method for each sample is simple random sampling.
2. The samples are independent.
3. Each population is at least 20 times larger than its respective sample.

The sampling distribution is approximately normal, which is generally the case if any of the following conditions apply.

1. The population distribution is normal.
2. The population data are symmetric, unimodal, without outliers, and the sample size is 15 or less.
3. The population data are slightly skewed, unimodal, without outliers, and the sample size is 16 to 40.
4. The sample size is greater than 40, without outliers.

Set	Null hypothesis	Alternative hypothesis	Number of tails
1	$\mu_1 - \mu_2 = d$	$\mu_1 - \mu_2 \neq d$	2
2	$\mu_1 - \mu_2 \geq d$	$\mu_1 - \mu_2 < d$	1
3	$\mu_1 - \mu_2 \leq d$	$\mu_1 - \mu_2 > d$	1

When the null hypothesis states that there is no difference between the two population means (i.e., d = 0), the null and alternative hypothesis are often stated in the following form.

Ho: μ1 = μ2

Ha: μ1 ≠ μ2

Test method: Use the two-sample t-test to determine whether the difference between means found in the sample is significantly different from the hypothesized difference between means.

Standard error. Compute the standard error (SE) of the sampling distribution.
SE = sqrt[(s1**2/n1) + (s2**2/n2)]
where s1 is the standard deviation of sample 1, s2 is the standard deviation of sample 2, n1 is the size of sample 1, and n2 is the size of sample 2.

Degrees of freedom. The degrees of freedom (DF) is:
DF = (s1**2/n1 + s2**2/n2)**2 / { [(s1**2 / n1)**2 / (n1 - 1)] + [(s2**2 / n2)**2 / (n2 - 1)] }
If DF does not compute to an integer, round it off to the nearest whole number. Some texts suggest that the degrees of freedom can be approximated by the smaller of n1 - 1 and n2 - 1; but the above formula gives better results.

Test statistic. The test statistic is a t statistic (t) defined by the following equation.
t = [(x1 - x2) - d] / SE
where x1 is the mean of sample 1, x2 is the mean of sample 2, d is the hypothesized difference between population means, and SE is the standard error.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a t statistic, use the t Distribution Calculator to assess the probability associated with the t statistic, having the degrees of freedom computed above. (See sample problems at the end of this lesson for examples of how this is done.)

The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a t statistic, use the t Distribution Calculator to assess the probability associated with the t statistic, having the degrees of freedom computed above.

Difference between paired means:

1. The sampling method for each sample is simple random sampling.
2. The test is conducted on paired data. (As a result, the data sets are not independent.)

The sampling distribution is approximately normal, which is generally true if any of the following conditions apply.

1. The population distribution is normal.
2. The population data are symmetric, unimodal, without outliers, and the sample size is 15 or less.
3. The population data are slightly skewed, unimodal, without outliers, and the sample size is 16 to 40.
4. The sample size is greater than 40, without outliers.

The hypotheses concern a new variable d, which is based on the difference between paired values from two data sets.

$$d = x1 - x2$$

where x1 is the value of variable x in the first data set, and x2 is the value of the variable from the second data set that is paired with x1.

The table below shows three sets of null and alternative hypotheses. Each makes a statement about how the true difference in population values μd is related to some hypothesized value D.

Set	Null hypothesis	Alternative hypothesis	Number of tails
1	$\mu_d = D$	$\mu_d \neq D$	2
2	$\mu_d \geq D$	$\mu_d < D$	1
3	$\mu_d \leq D$	$\mu_d > D$	1

Test method: Use the matched-pairs t-test to determine whether the difference between sample means for paired data is significantly different from the hypothesized difference between population means.

Standard deviation. Compute the standard deviation (sd) of the differences computed from n matched pairs.
sd = sqrt [(Σ(di - d)2 / (n - 1)]
where di is the difference for pair *i*, d is the sample mean of the differences, and n is the number of paired values.

Standard error. Compute the standard error (SE) of the sampling distribution of d.
SE = sd * sqrt{ (1/n) * [(N - n) / (N - 1)] }
where *s*d is the standard deviation of the sample difference, *N* is the number of matched pairs in the population, and *n* is the number of matched pairs in the sample. When the population size is much larger (at least 20 times larger) than the sample size, the standard error can be approximated by:
SE = sd / sqrt(n)

Degrees of freedom. The degrees of freedom (DF) is: DF = n - 1 .

Test statistic. The test statistic is a t statistic (t) defined by the following equation.
t = [(x1 - x2) - D] / SE = (d - D) / SE
where x1 is the mean of sample 1, x2 is the mean of sample 2, d is the mean difference

between paired values in the sample, D is the hypothesized difference between population means, and SE is the standard error.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a t statistic, use the t Distribution Calculator to assess the probability associated with the t statistic, having the degrees of freedom computed above.

Chi-Square Goodness of fit:

1. The sampling method is simple random sampling.
2. The variable under study is categorical.
3. The expected value of the number of sample observations in each level of the variable is at least 5.

The test is applied when you have one categorical variable from a single population. It is used to determine whether sample data are consistent with a hypothesized distribution.

For a chi-square goodness of fit test, the hypotheses take the following form.

- Ho: The data are consistent with a specified distribution.
- Ha: The data are *not* consistent with a specified distribution.

Typically, the null hypothesis (Ho) specifies the proportion of observations at each level of the categorical variable. The alternative hypothesis (Ha) is that *at least* one of the specified proportions is not true.

Test method: Use the chi-square goodness of fit test to determine whether observed sample frequencies differ significantly from expected frequencies specified in the null hypothesis.

Degrees of freedom. The degrees of freedom (DF) is equal to the number of levels (k) of the categorical variable minus 1.
DF = k - 1

Expected frequency counts. The expected frequency counts at each level of the categorical variable are equal to the sample size times the hypothesized proportion from the null hypothesis
Ei = npi
where Ei is the expected frequency count for the *i*th level of the categorical variable, n is the total sample size, and pi is the hypothesized proportion of observations in level *i*.

Test statistic. The test statistic is a chi-square random variable (X2) defined by the following equation.
X**2 = Σ [(Oi - Ei)**2 / Ei]
where Oi is the observed frequency count for the *i*th level of the categorical variable, and Ei is the expected frequency count for the *i*th level of the categorical variable.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a chi-square, use the Chi-Square Distribution Calculator to assess the probability associated with the test statistic. Use the degrees of freedom computed above.

Chi-square test of Homogeneity:

1. For each population, the sampling method is simple random sampling.
2. The variable under study is categorical.

3. If sample data are displayed in a contingency table (Populations x Category levels), the expected frequency count for each cell of the table is at least 5.

The test is applied to a single categorical variable from two or more different populations. It is used to determine whether frequency counts are distributed identically across different populations.

$$H_o: P_{\text{level 1 of pop 1}} = P_{\text{level 1 of pop 2}} = \ldots = P_{\text{level 1 of pop r}}$$
$$H_o: P_{\text{level 2 of pop 1}} = P_{\text{level 2 of pop 2}} = \ldots = P_{\text{level 2 of pop r}}$$
$$\ldots$$
$$H_o: P_{\text{level c of pop 1}} = P_{\text{level c of pop 2}} = \ldots = P_{\text{level c of pop r}}$$

The alternative hypothesis (Ha) is that *at least* one of the null hypothesis statements is false.

Test method: Use the chi-square test for homogeneity to determine whether observed sample frequencies differ significantly from expected frequencies specified in the null hypothesis.

Degrees of freedom. The degrees of freedom (DF) is equal to:
DF = (r - 1) * (c - 1)
where r is the number of populations, and c is the number of levels for the categorical variable.

Expected frequency counts. The expected frequency counts are computed separately for each population at each level of the categorical variable, according to the following formula.
Er,c = (nr * nc) / n
where Er,c is the expected frequency count for population r at level *c* of the categorical variable, nr is the total number of observations from population r, nc is the total number of observations at treatment level *c*, and n is the total sample size.

Test statistic. The test statistic is a chi-square random variable (X2) defined by the following equation.
X**2 = Σ [(Or,c - Er,c)**2 / Er,c]
where Or,c is the observed frequency count in population *r* for level *c* of the categorical variable, and Er,c is the expected frequency count in population *r* for level *c* of the categorical variable.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a chi-square, use the Chi-Square Distribution Calculator to assess the probability associated with the test statistic. Use the degrees of freedom computed above.

Chi-square test for independence:

1. The sampling method is simple random sampling.
2. The variables under study are each categorical.
3. If sample data are displayed in a contingency table, the expected frequency count for each cell of the table is at least 5.

The test is applied when you have two categorical variables from a single population. It is used to determine whether there is a significant association between the two variables.

H_o: Variable A and Variable B are independent.

H_a: Variable A and Variable B are not independent.

Test method. Use the chi-square test for independence to determine whether there is a significant relationship between two categorical variables.

Degrees of freedom. The degrees of freedom (DF) is equal to:
DF = (r - 1) * (c - 1)
where r is the number of levels for one categorical variable, and c is the number of levels for the other categorical variable.

Expected frequencies. The expected frequency counts are computed separately for each level of one categorical variable at each level of the other categorical variable. Compute r * c expected frequencies, according to the following formula.
Er,c = (nr * nc) / n
where Er,c is the expected frequency count for level *r* of Variable A and level *c* of Variable B, nr is the total number of sample observations at level r of Variable A, nc is the total number of sample observations at level *c* of Variable B, and n is the total sample size.

Test statistic. The test statistic is a chi-square random variable (X2) defined by the following equation.
X2 = Σ [(Or,c - Er,c)**2 / Er,c]
where Or,c is the observed frequency count at level *r* of Variable A and level *c* of Variable B, and Er,c is the expected frequency count at level *r* of Variable A and level *c* of Variable B.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a chi-square, use the Chi-Square Distribution Calculator to assess the probability associated with the test statistic. Use the degrees of freedom computed above.

Hypothesis test for Regression slope:

The test focuses on the slope of the regression line

$$Y = B0 + B1X$$

where B0 is a constant, B1 is the slope (also called the regression coefficient), X is the value of the independent variable, and Y is the value of the dependent variable.

If we find that the slope of the regression line is significantly different from zero, we will conclude that there is a significant relationship between the independent and dependent variables.

1. The dependent variable Y has a linear relationship to the independent variable X.
2. For each value of X, the probability distribution of Y has the same standard deviation σ.
3. For any given value of X,the Y values are independent.
4. The Y values are roughly normally distributed (i.e., symmetric and unimodal). A little skewness is ok if the sample size is large.

If there is a significant linear relationship between the independent variable *X* and the dependent variable *Y*, the slope will *not* equal zero.

$$Ho: B1 = 0$$

$$Ha: B1 \neq 0$$

The null hypothesis states that the slope is equal to zero, and the alternative hypothesis states that the slope is not equal to zero.

Test method. Use a linear regression t-test (described in the next section) to determine whether the slope of the regression line differs significantly from zero.

Standard error. Many statistical software packages and some graphing calculators provide the standard error of the slope as a regression analysis output. The table below shows hypothetical output for the following regression equation: y = 76 + 35x .

Predictor	Coef	SE Coef	T	P
Constant	76	30	2.53	0.01
X	35	20	1.75	0.04

In the output above, the standard error of the slope (shaded in gray) is equal to 20. In this example, the standard error is referred to as "SE Coeff". However, other software packages might use a different label for the standard error. It might be "StDev", "SE", "Std Dev", or something else. If you need to calculate the standard error of the slope (SE) by hand, use the following formula:

$$SE = sb1 = \text{sqrt}\ [\ \Sigma(yi - \hat{y}i)^{**}2 / (n - 2)\ ] / \text{sqrt}\ [\ \Sigma(xi - x)^{**}2\ ]$$

where yi is the value of the dependent variable for observation *i*, ŷi is estimated value of the dependent variable for observation *i*, xi is the observed value of the independent variable for observation *i*, x is the mean of the independent variable, and n is the number of observations.

Slope. Like the standard error, the slope of the regression line will be provided by most statistics software packages. In the hypothetical output above, the slope is equal to 35.

Degrees of freedom. For simple linear regression (one independent and one dependent variable), the degrees of freedom (DF) is equal to:
DF = n - 2
where n is the number of observations in the sample.

Test statistic. The test statistic is a t statistic (t) defined by the following equation.
t = b1 / SE
where b1 is the slope of the sample regression line, and SE is the standard error of the slope.

P-value. The P-value is the probability of observing a sample statistic as extreme as the test statistic. Since the test statistic is a t statistic, use the t Distribution Calculator to assess the probability associated with the test statistic. Use the degrees of freedom computed above.

Resampling and Permutation test:

A permutation test (also called a randomization test, re-randomization test, or an exact test) is a type of statistical significance test in which the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under all possible rearrangements of the observed data points.

To illustrate the basic idea of a permutation test, suppose we collect random variables X_A and X_B for each individual from two groups A and B whose sample means are $\bar{x}_A$ and $\bar{x}_B$, and that we want to know whether X_A and X_B come from the same distribution. Let n_A and n_B be the sample size collected from each group. The permutation test is designed to

determine whether the observed difference between the sample means is large enough to reject, at some significance level, the null hypothesis H0 that the data drawn from A is from the same distribution as the data drawn from B.

The test proceeds as follows. First, the difference in means between the two samples is calculated: this is the observed value of the test statistic, T_{obs} .

Next, the observations of groupsA and B are pooled, and the difference in sample means is calculated and recorded for every possible way of dividing the pooled values into two groups of size n_A and n_B (i.e., for every permutation of the group labels A and B). The set of these calculated differences is the exact distribution of possible differences (for this sample) under the null hypothesis that group labels are exchangeable (i.e., are randomly assigned).

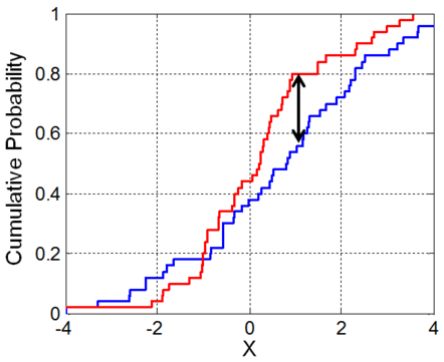
The one-sided p-value of the test is calculated as the proportion of sampled permutations where the difference in means was greater than or equal to T_{obs} . The two-sided p-value of the test is calculated as the proportion of sampled permutations where the absolute difference was greater than or equal to $|T_{\text{obs}}|$.

Alternatively, if the only purpose of the test is to reject or not reject the null hypothesis, one could sort the recorded differences, and then observe if T_{obs} is contained within the middle $(1-\alpha) \times 100$ of them, for some significance level α . If it is not, we reject the hypothesis of identical probability curves at the $\alpha \times 100\%$ significance level.

KS Test:

This test is used to determine whether the two random variables X1 and X2 have the same distribution(null hypothesis) or not(alternate).

Test Statistic is the maximum difference between the two cdf's at any point.



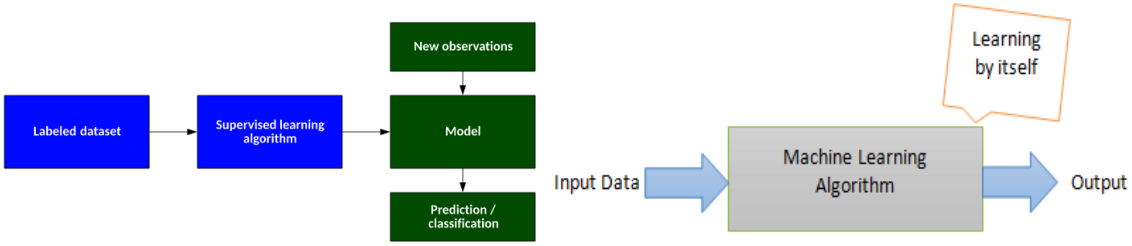
Proportional Sampling:

Proportional sampling is the method of picking an element proportional to its weight, i.e., the higher the weight of the object, the better are its chances of being selected.

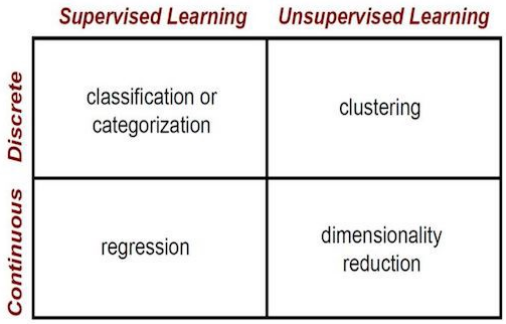
Machine Learning

Supervised Learning vs Unsupervised Learning

Supervised learning is a machine learning approach that's defined by its use of labeled datasets. These datasets are designed to train or "supervise" algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time.Supervised learning can be separated into two types of problems: classification and regression.



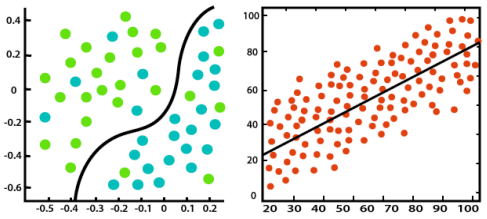
Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention (hence, they are "unsupervised"). Unsupervised learning models are used for three main tasks: clustering, association and dimensionality reduction



Regression vs Classification

Regression and Classification algorithms are Supervised Learning algorithms. Both the algorithms are used for prediction in Machine learning and work with the labeled datasets. But the difference between both is how they are used for different machine learning problems.

The main difference between Regression and Classification algorithms that Regression algorithms are used to predict the continuous values such as price, salary, age, etc. and Classification algorithms are used to predict/Classify the discrete values such as Male or Female, True or False, Spam or Not Spam, etc.



Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters. In Classification, a computer program is trained on the training dataset and based on that training, it categorizes the data into different classes. The task of the classification algorithm is to find the mapping function to map the input(x) to the discrete output(y).
Example: The best example to understand the Classification problem is Email Spam Detection. The model is trained on the basis of millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam or not. If the email is spam, then it is moved to the Spam folder.

Different Classification Algorithms:
Logistic Regression
K-Nearest Neighbours
Support Vector Machines
Kernel SVM
Naïve Bayes
Decision Tree Classification
Random Forest Classification

Classification can again be divided into types:
Binary classification (2 classes) and multi-class classification.

Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of Market Trends, prediction of House prices, etc.
The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).
Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

Different Regression Algorithms:
Simple Linear Regression
Multiple Linear Regression
Polynomial Regression
Support Vector Regression
Decision Tree Regression
Random Forest Regression

Metrics in Machine Learning

Classification Metrics

Accuracy: No. of correctly classified points/ Total No.of points.

It lies between 0 & 1, the higher the better.

Problems with accuracy:

1. It doesn't give the correct picture of how the model is performing on imbalance data sets.

   Ex: Let's say we build model for cancer or not dataset. The cases with cancer are very less when compared to the cases with no cancer. Let say 1% cancer and 99% no cancer. If our model is predicting all the points as no cancer then the accuracy will 0.99 but still the model is performing bad since it's not able to predict the cancer cases at all.

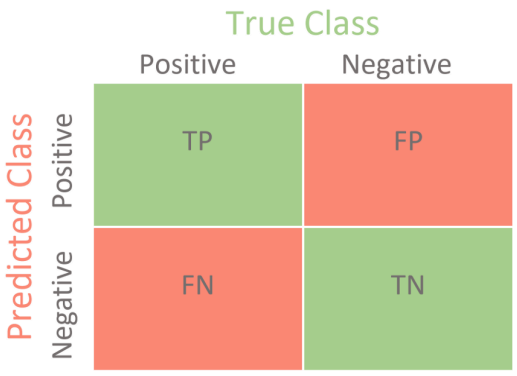
   That's why we should never use accuracy for imbalance datasets.

2. Let's consider a case where we need to compare 2 models that give probability as the output instead of class labels. Accuracy is not a good metric to choose.

   Consider the below case, if we take the threshold of 0.5 (>0.5 is 1 and <=0.5 is 0), the predicted class labels will be same for the 2 models and the classification accuracy is same but clearly we can see that M1 is performing far better than M2.

Confusion Matrix:

Let's consider a binary classification task, confusion matrix looks like:



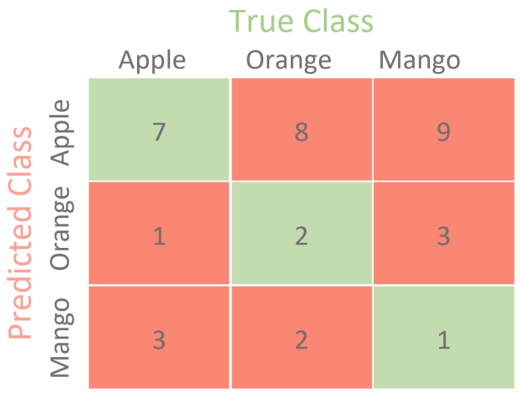
TP: True Positive (Actual +, Predicted +)

FP: False Positive(Actual -,Predicted +)

FN: False Negative(Actual +, Predicted -)

TN: True Negative(Actual -, Predicted -)

Confusion Matrix for Multi-class classification problem:



Ideally in a confusion matrix, the diagonal elements should be as high as possible and the rest should be as small as possible.

Many metrics can be derived from confusion matrix:

1. True positive rate TPR: True Positives/Total Positives

   TP/(TP+FN)

2. True negative rate TNR: True Negatives/Total Negatives

   TN/(TN+FP)

3. False Positive Rate FPR: False Positives/Total Negatives

   FP/(TN+FP)

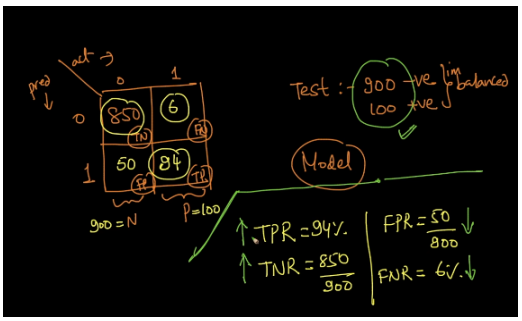
4. False Negative Rate FNR: False Negatives/Total Positives

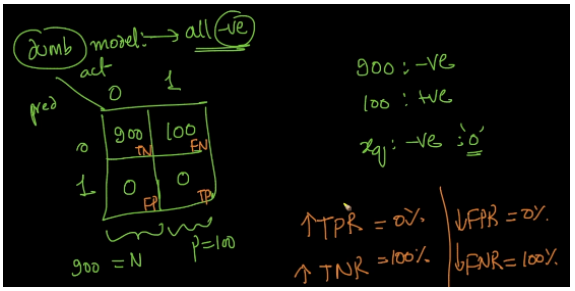
   FN/(TP+FN)

TPR & TNR should be high and FPR,FNR should be low.

If we derive metrics by considering all or most of them, the data imbalance issue that accuracy had can be solved.

Let’s consider 2 examples in imbalance data.





Case 1 is a decent one.in which we have fairly good TPR and TNR and low FPR and FNR. But case-2 is pretty bad. Even though we have high TNR and 0 FPR, our TPR is 0 & FNR is 100% that means we are not predict Positives at all which is really bad.

Depending on the problem we can choose which one is of more importance to us. Let's say there is a problem of cancer vs no cancer, in that even if we falsely predict a person has cancer(FP) it's fine he can undergo more tests to confirm. But we shouldn't falsely predict that a person has no cancer. I.e., FN should be as low as possible. In this case, FN and TP are important to us.

Few important metrics that can be derived from confusion matrix are Precision, Recall & F1-score. (used predominantly in information retrieval)

Precision: Of all the points that the model predicted, how many are actually +ve.

TP/(TP+FP)

Recall: Of all the +ve points, how many are actually predicted as +ve.

TP/(TP+FN)

Precision & Recall mainly care about +ve class. They are mainly useful if we care about the +ve class more.

We always want precision & recall to be high. **F1-score** does the job to bring the 2metrics together. F1-score is harmonic mean of Precision & Recall. It's high only when Pr and Re are high.

F1-score = 2*Precision*Recall/(Precision+Recall)

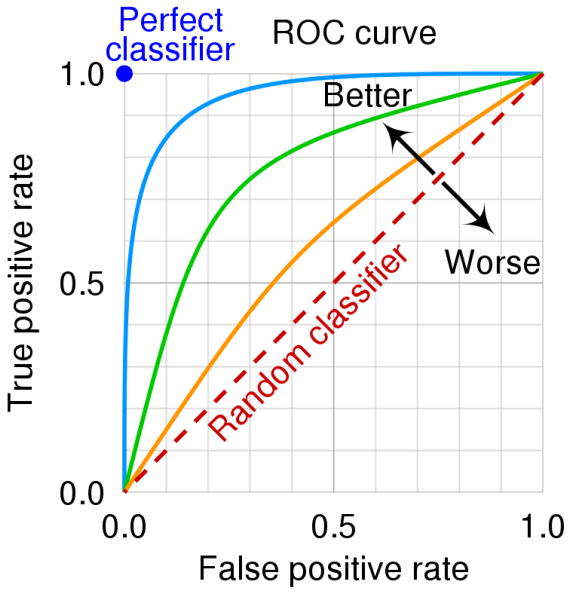
Precision & Recall are interpretable but F1-score isn't.

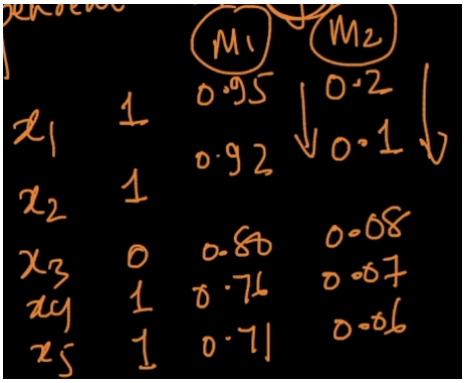
Receiver Operating Characteristic(ROC) curve & Area under the curve(auc):

Used in binary classification. Model should output the probability score instead of the class label.ROC curve is a plot between TPR and FPR.

Steps to plot ROC curve:

1. Sort the predicted probabilities in descending order.
2. Consider each probability as threshold (>=probability is class 1 and < is class 0).
3. Calculate corresponding TPR and FPR for this setup and plot a point.
4. Repeat this process.





Area under this curve is called auc. Auc lies between 0 & 1, the higher the better. The red line passing through the center is the ROC curve for a random model. Area under the red line is 0.5.

Points to note:

1. AUC can be impacted by imbalanced data in some cases.
2. AUC is not dependent on the probability scores themselves. It only cares about the ordering. In the above figure AUC of M1 & M2 is same even though M1 is doing much much better.
3. If AUC of a model is <0.5 (worse than a random model), we'll just have to invert the class label to make it better than the random. (AUCnew = 1-AUC).

Log loss

Log-loss is indicative of how close the prediction probability is to the corresponding actual/true value (0 or 1 in case of binary classification). The more the predicted probability diverges from the actual value, the higher is the log-loss value. log-loss lies between 0 to ∞ Smaller the log-loss the better the model. It's hard to interpret.

Binary classification:

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N} -\left(y_i * \log(p_i) + (1-y_i) * \log(1-p_i) \right)$$

Multiclass classification:

$$logloss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij} \log(p_{ij})$$

- N is the number of rows
- M is the number of classes

Regression Metrics

R-square or Coefficient of determination:
R-squared, also known as the coefficient determination, defines the degree to which the variance in the dependent variable (or target) can be explained by the independent variable (features).
Let us understand this with an example — say the R-squared value for a particular model comes out to be 0.7. This means that 70% of the variation in the dependent variable is explained by the independent variables.
Ideally, we would want the independent variables to be able to explain all the variation in the target variable. In that scenario, the r-squared value would be 1. Thus we can say that higher the r-squared value, the better the model.
So, in simple terms, higher the R squared, the more variation is explained by your input variables and hence better is your model. Also, the r-squared would range from 0 to 1. Here is the formula for calculating R-squared-
The R-squared is calculated by dividing the sum of squares of residuals from the regression model (given by SSres) by the total sum of squares of errors from the average model (average model is predicting average value of the dependent variable no matter what the input is) (given by SStot) and then subtracting it from 1.

$$r^2 = 1 - \frac{\text{SS Error}}{\text{SS Total}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

One drawback of r-squared is that it assumes every variable helps in explaining the variation in the target, which might not always be true. For instance, if we add new features to the data (which may or may not be useful), the r-squared value for the model would either increase or remain the same but it would never decrease.
This is taken care of by a slightly modified version of r-squared, called the **adjusted r-squared.**
Similar to R-squared, the Adjusted R-squared measures the variation in the dependent variable (or target), explained by only the features which are helpful in making predictions. Unlike R-squared, the Adjusted R-squared would penalize you for adding features which are not useful for predicting the target.
Let us mathematically understand how this feature is accommodated in Adjusted R-Squared. Here is the formula for adjusted r-squared

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$R^2_{adj} = 1 - ((1 - R^2)\frac{N-1}{N-M-1}$$

Here R^2 is the r-squared calculated, N is the number of rows and M is the number of columns. As the number of features increases, the value in the denominator decreases.
If the R2 increases by a significant value, then the adjusted r-squared would increase.
If there is no significant change in R2, then the adjusted r2 would decrease.
R-square is not robust to outliers.

Root mean squared error(RMSE):
Impacted by outliers

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation
i = variable i
N = number of non-missing data points
x_i = actual observations time series
$\hat{x}_i$ = estimated time series

Mean absolute deviation:

$$\frac{1}{n}\sum_{i=1}^{n}|x_i - m(X)|$$

$m(X)$ = average value of the data set
n = number of data values
x_i = data values in the set

Median Absolute deviation (MAD): Robust to outliers.

$\mathrm{MAD} = \mathrm{median}(|X_i - \tilde{X}|) \quad \tilde{X} = \mathrm{median}(X)$:

Distribution of errors:
Apart from deriving metrics like mean,median e.t.c., of errors, we can also directly look at their distribution. We can compare 2 models with its c.d.f. Whichever model's c.d.f is on top, that is the better model.

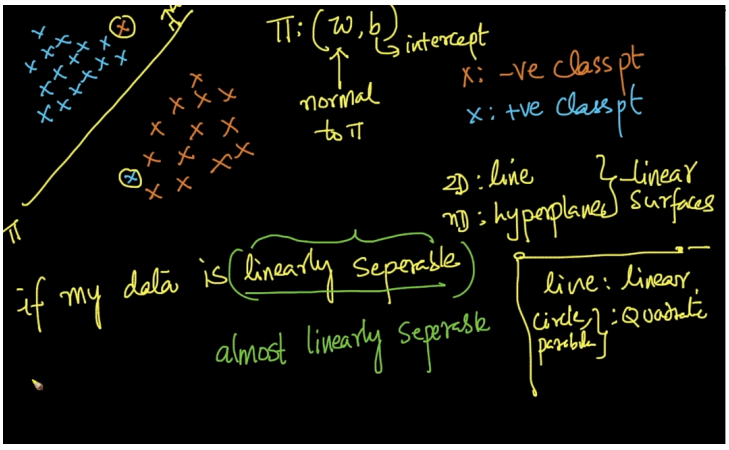
Logistic Regression

Classification algorithm.
Intuition:
Multiple interpretations: Geometric, Probabilistic, loss-function

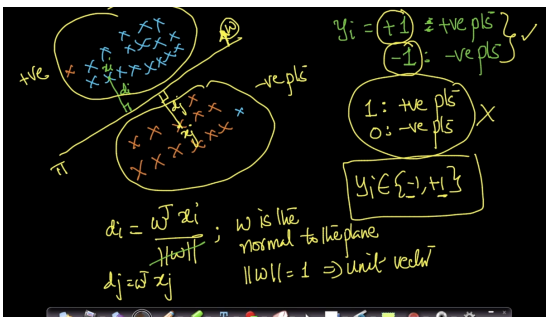
Geometric intuition:

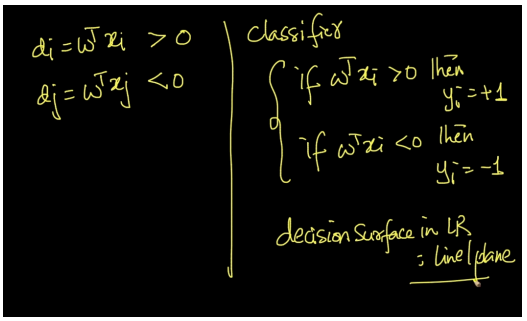
Assumption of Logistic Regression: Data is linearly seperable or almost linearly seperable.

Task is to find a plane Π that best separates, +ve and -ve class points.



For every point, we calculate d, the perpendicular distance between the point to the plane. If it is +ve we will consider it as a +ve point else a negative point. The Plane here is the decision surface.

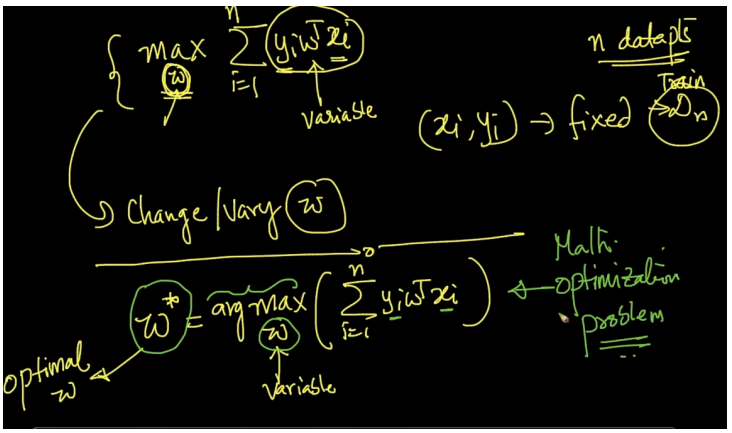




If yi is a +ve point and w.Txi>0 then it is correctly classified and yi*wT*xi>0.If yi is a -ve point and w.Txi<0 then it is correctly classified and yi*wT*xi>0.

If yi is a +ve point and w.Txi<0 then it is incorrectly classified and yi*wT*xi<0.If yi is a -ve point and w.Txi>0 then it is incorrectly classified and yi*wT*xi<0.

If we want our classifier to be very good i.e., minimum no.of misclassifications, we want as many points as possible for which yi*wT*xi>0.So, we can try to maximize the sum of yi*wT*xi across all the points. Here yi & xi are train data points so they are fixed. So, w should be taken in such a way that it maximises sum of yi*wT*xi across all the points. This is an optimisation problem to solve.

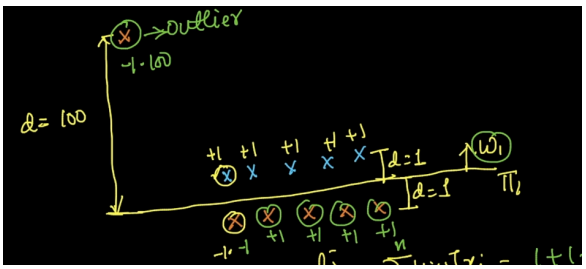


Let w* be the w that maximises the optimization problem.

yi*wT*xi can be called as signed distance since, wT*xi is the distance from xi to the plane that separates both the classes and the sign is +ve for correctly classified points, -ve for incorrectly classified points.

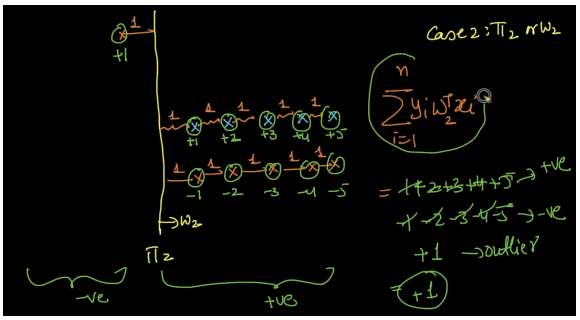
There are few problems with this formulation.

In the below figure, Π1 is the plane that seperates +ve and -ve classes. This looks good except for 1 outlier that is misclassified. Our sum of signed distances will be +10-100=-90.Just because of a single outlier. Since we are trying maximise the sum of signed distances, Π1 will not be considered as decision boundary eventhough it does a very good job.



In the below figure there is another decision boundary Π2. It misclassifies all the -ve points just to tune to the outlier. There by making our sum of squared distances as +1.

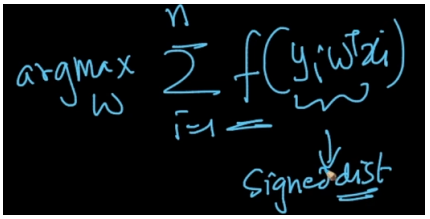
This might be preferred over Π1.



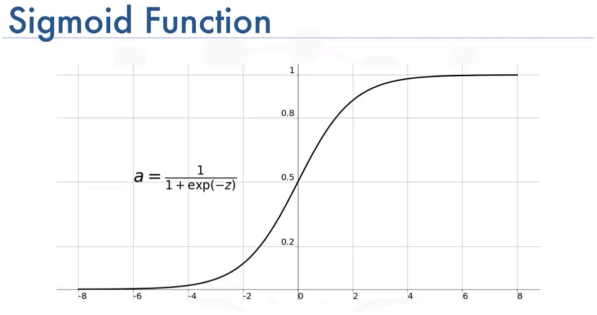
What we are trying say is, this current setup of maximising sum of signed distances can be affected a lot by the presence of outliers.

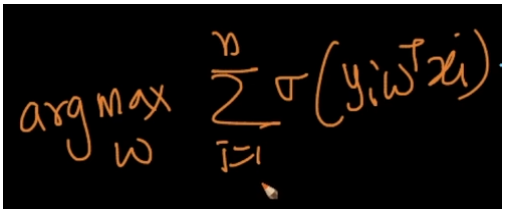
We need to modify the formulation.

We can use squashing to reduce the effect of outlier. If the signed distance is small, use it as it is. If it's large then make it small.Sp, instead of maximising signed distance, we can try to maximise f(signed distance).



One such function is sigmoid function. It is bounded between 0 & 1. It looks like a linear plot between -2 & 2 and then it starts to taper off.

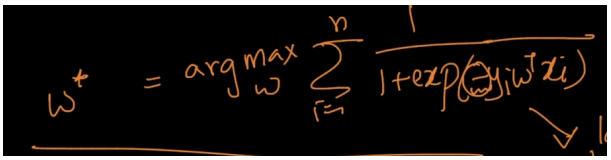




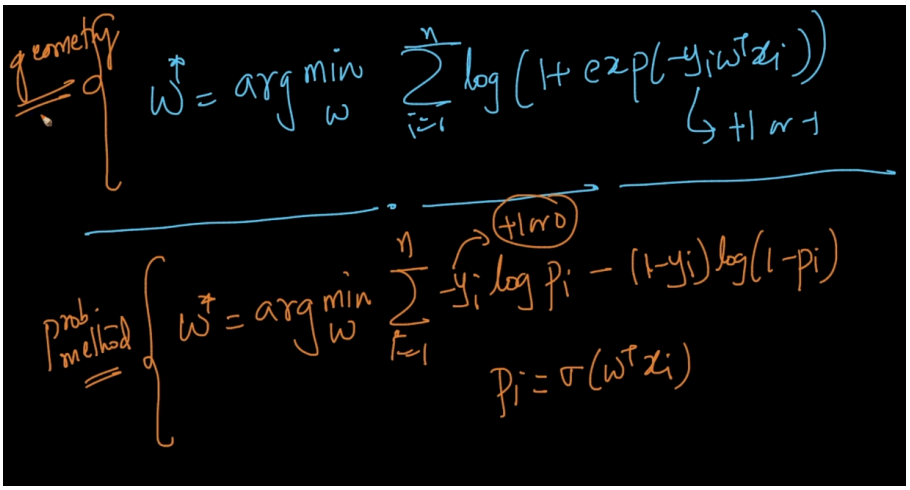
When the signed distance is 0, sigmoid value is 0.5.

There are many functions that can fit the criteria of tapering off, but the reason we choose sigmoid is because it has a nice probabilistic interpretation as well.

If the point is very far from the hyperplane, the value of the sigmoid can be close to either 0 or 1 based on which side of the hyperplane it is in. If the point is on the hyperplane that means we can't be sure whether it's a positive point or not, in this case sigmoid gives 0.5 (which is random as per probability).



This formulation is less impacted by the presence of outliers.



Both are similar.

The optimal w* is called weight vector.

Decision on query points: wT*xq>0 +ve class & wT*xq<0 -ve class.

Interpretation:

If wi (weight corresponds to feature i) is +ve, if xqi increases, wi*xqi increases, wTxq increases, Sigmoid(wTxq) also increases, the p(yq=+1) also increase.

If wi (weight corresponds to feature i) is -ve, if xqi decreases, wi*xqi decreases, wTxq decreases, Sigmoid(wTxq) also decreases, the p(yq=+1) also decrease.

Overfitting:

As per the optimisation problem given in the above figure, for min value of the expression for each point, yi*wT*xi should be as high as possible may be tending to infinity.

If we pick w such that,

1. All training points are correctly classified &
2. yi*wT*xi tends to infinity

Such a w is the best w. But there is a problem with this. What if some of the training points are outliers and we are doing a perfect job on them as well.

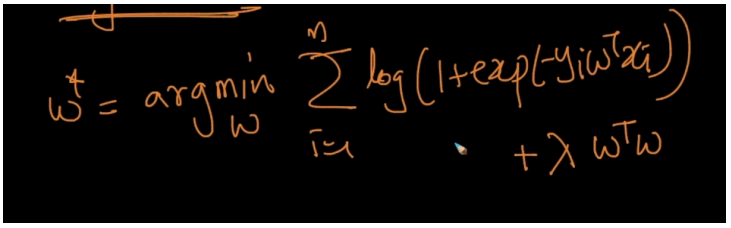
The issues with this are

1. Weights tend to overfit (do well on train but badly on test).
2. Weights may become very high and tends to infinity(+/-).

Regularisation:

Keep a check on weights and tend to reduce overfitting.

Minimize(Loss Function + Regularisation)



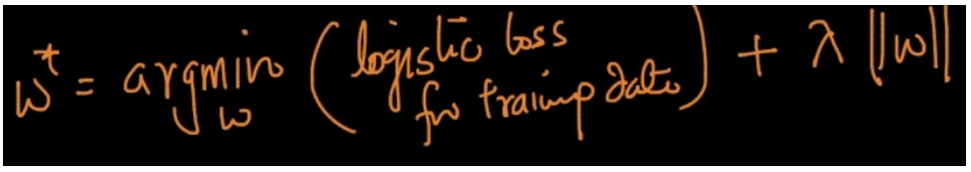
The 2nd term is called square of l2 norm of w (w**2). 2nd term is the regularization term. Now we are trying to minimise the square of the weights as well. (This will reduce the chance of weights becoming very large or weights being tuned exactly to train data). This type of regularization is called L2-Regularization.

ƛ is the regularisation coefficient and is a hyper parameter. If ƛ is very small (almost 0) then there is no regularization and the overfitting will still be there. If ƛ is very large then the minimization problem will try to reduce the weights a lot inorder reduce the overall term (weights will almost become 0).Then the model will be dumb model.(Underfitting).

Underfit-> High bias

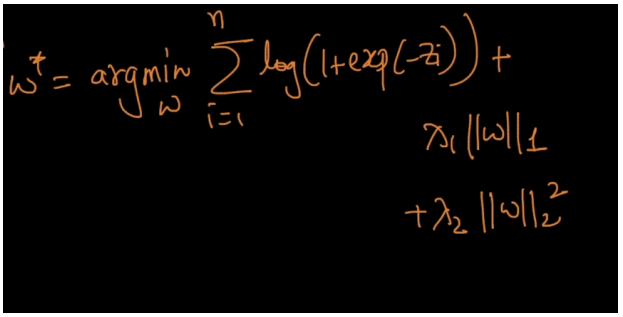
Overfit->High variance.

L1 Regularisation



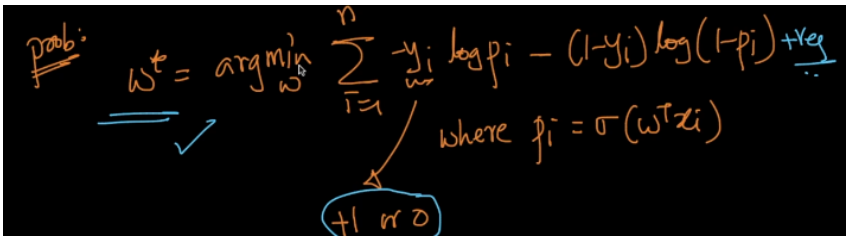
Only the regualrisation term will change and it serves the same purpose as the L2 regularizer. It has one major advantage. This can create sparsity. Some of the unimportant feature weights can become 0 or close to 0.

Elastic-net: Combination of L1 and L2



Assumptions of probabilistic derivation of logistic regression:

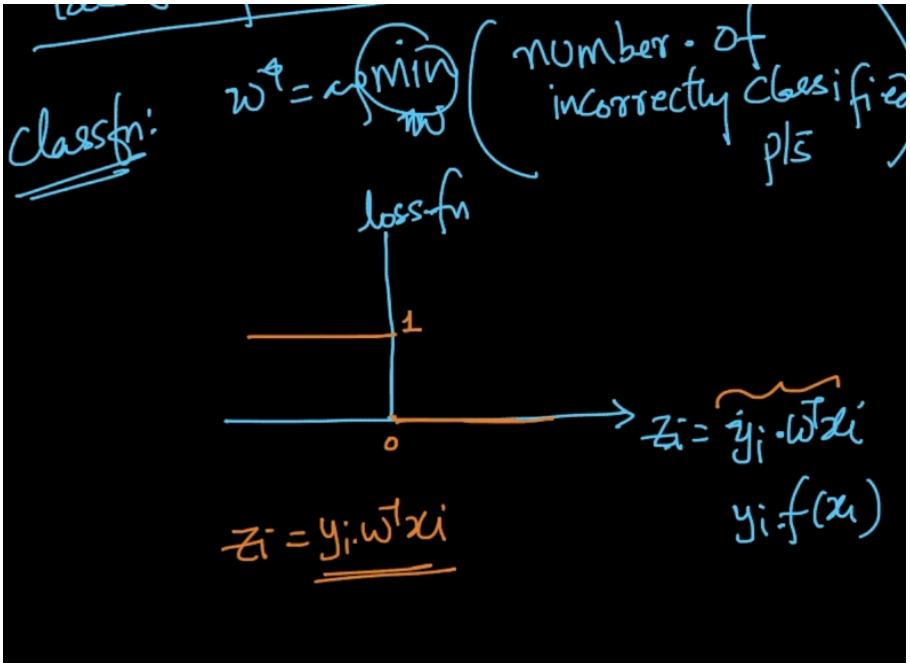
1. Class label should follow a bernoulli distribution (0 or 1).
2. For real valued features, Condition probabilities of Y given each Xi should follow gaussian distribution.
3. Xi and Xj are conditionally independent.



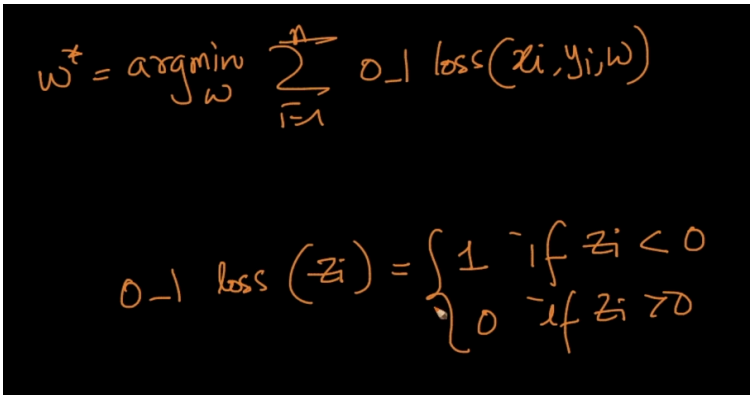
Loss minimization interpretation of Logistic Regression:

Ideal optimization problem is to minimize the no.of incorrectly classified points.

If we give +1 for each incorrectly classified points and 0 for correctly classified points… we want to reduce the sum.(0-1 loss function)

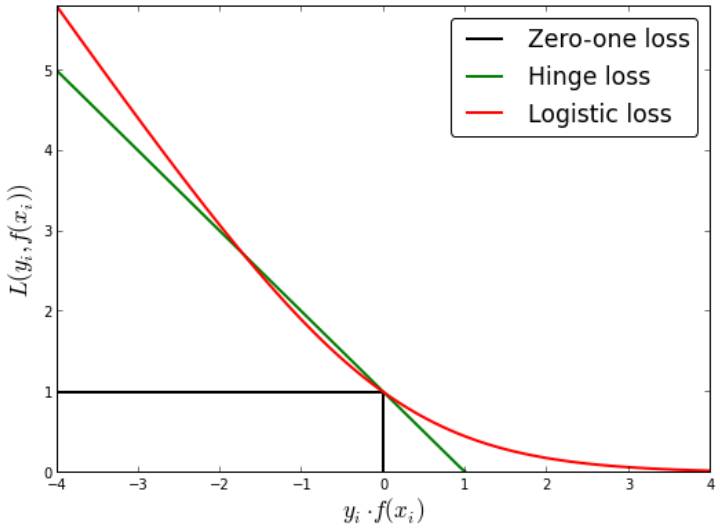


If Zi>0, yi*wT*xi>0 that means, the point is correctly classified. Else it's incorrectly classified.



But this loss function is not differentiable since there is a discontinuity at z=0. This can't be used in our optimization problem even though it is ideal.

We can use some functions that can approximate this behaviour. Logistic los is one such approximation, hinge loss is another.



Logistic loss is used in Logistic Regression, Hinge loss is used in SVMs. Exponential loss is used in Ada Boost.

Hyperparameters:

ƛ, regularisation coefficient is the hyper parameter.

ƛ1 and ƛ2 for elastic net.

Column/Feature standardisation:

Given each feature, we'll transform this such that

X=(x-mean(x))/std(x)

The scale of the feature can easily impact our result.

Feature importance and Model interpretability:

If we have all the features independent of one another, we can use the absolute value of weights value as feature importance.

If the weight value is negative, then it contributes to the probability that the point belongs to the negative class and vice-versa.

Interpreting the model is very easy in logistic regression. We can take the top-n features with highest +ve weights and say that these contribute more for the +ve class and top-n features with highest -ve weights and say that these contribute more for the -ve class.

Multicollinearity:

When there is multicollinearity in the data i.e., multiple features are collinear, we can't exactly represent the weights as feature importance. If features are collinear, then weight vectors can change arbitrarily.

Every data will have some kind of multicollinearity.

Given a dataset, if we want to use wi as feature importance or not:

Perturbation technique:

Compute weight vector as it is.

For every value of xij, add small noise term.

Compute new weights and compare with old weights, if they differ significantly then we can't use weights as feature importances.If they don't differ significantly, then we can use the weights as feature importances.

Space and Time complexity:

Train time: O(n*d) n: no .of points, d: dimensionality

Test Space: O(d)

Test time: O(d)

If d is small then Logistic Regression is very very good for low latency applications.

If d is large, we can use L1 regularisation and increase ƛ to introduce sparsity in weights to adjust the model for latency requirements. But again this is a tradeoff between bias,variance and latency.

Real world cases:

Decision surface: Linear/Hyper plane.

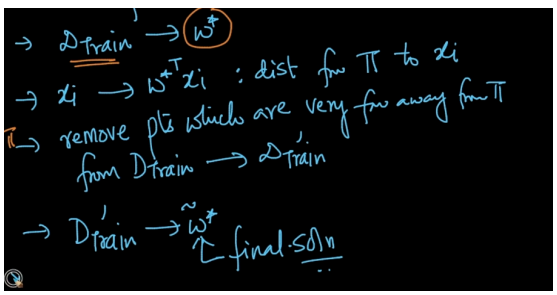
Assumption: Data is linearly seperable/almost linearly seperable.

Feature importance/Interpretability.

Imbalance data: Upsampling/Down sampling/ Class weights.

Outliers: Less impact due to usage of sigmoid function.

Hack to avoid outliers:

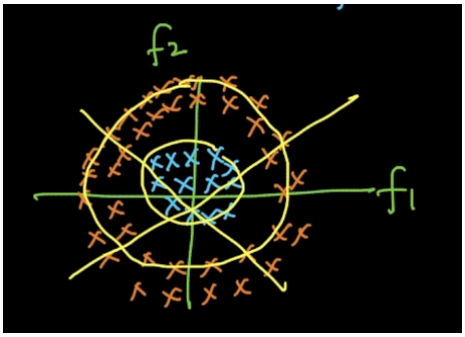


Multiclass classification: One vs Rest,Max entropy models, Softmax classifiers,Multinomial Logistic Regression.

What if the similarity matrix is given instead of data: Kernel Logistic Regression can be used.

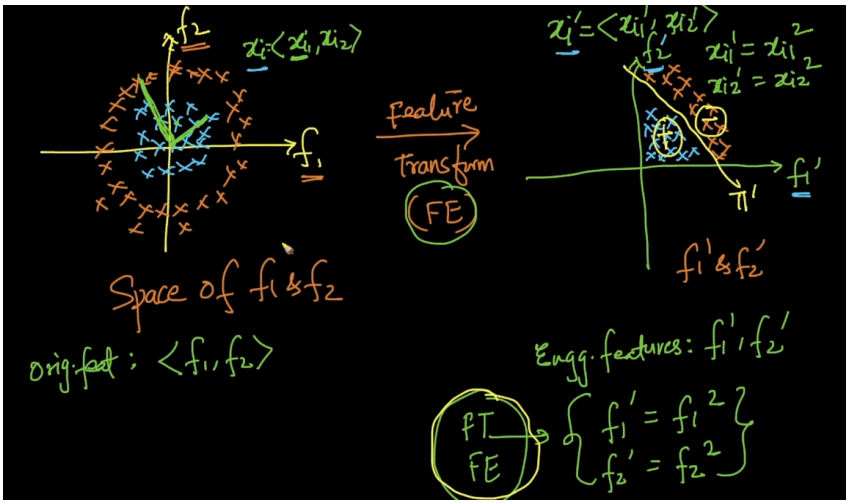
If d is large then Logistic regression work fairly well because in a high dimensional system, find the hyperplane that separates the classes is fairly easy. But again inorder to adjust the model for latency, L1 Regularisation should be used.

What if the data is not linearly seperable:



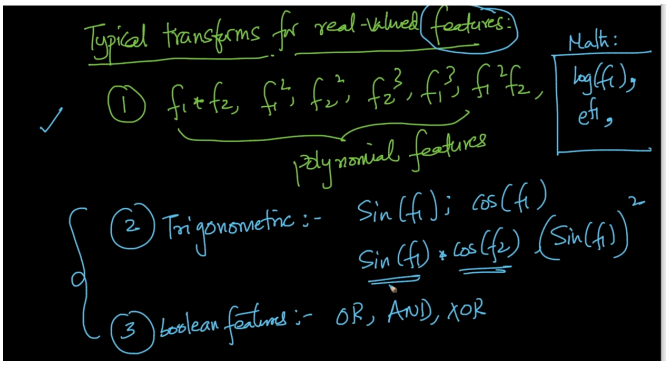
Drawing a hyperplane that best separates the above data is impossible in the given feature space.

We need to transform the features to some other space.(Feature engineering).

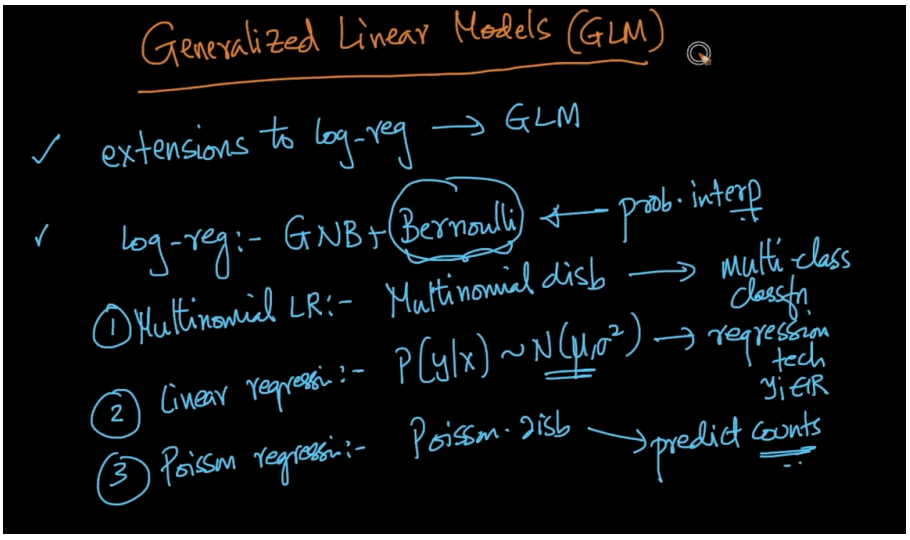


In the above example if we plot, f1' and f2' which are f1**2 and f2**2 respectively, we can create a separating hyperplane between the 2 classes. (Transformation/Feature engineering).

How do we know which transform to apply? Comes with practice.



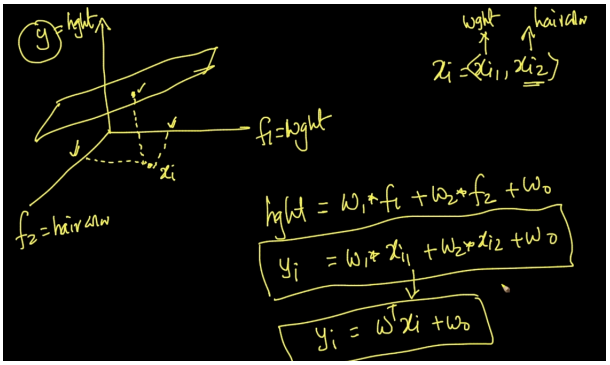
Generalized Linear Models:



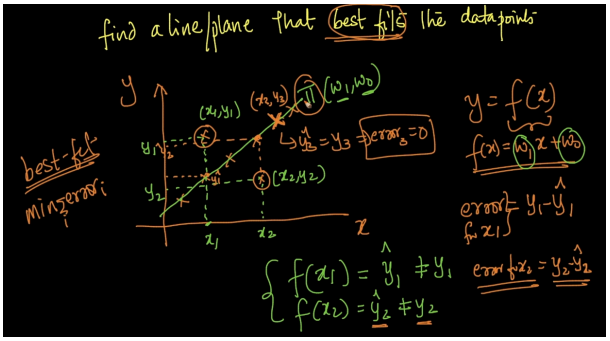
Linear Regression

Regression technique unlike logistic regression.

Geometry behind Linear Regression:
Linear regression finds a line or plane that best fits the given data.



Best fit line means that most of the points should lie as close to the line as possible.



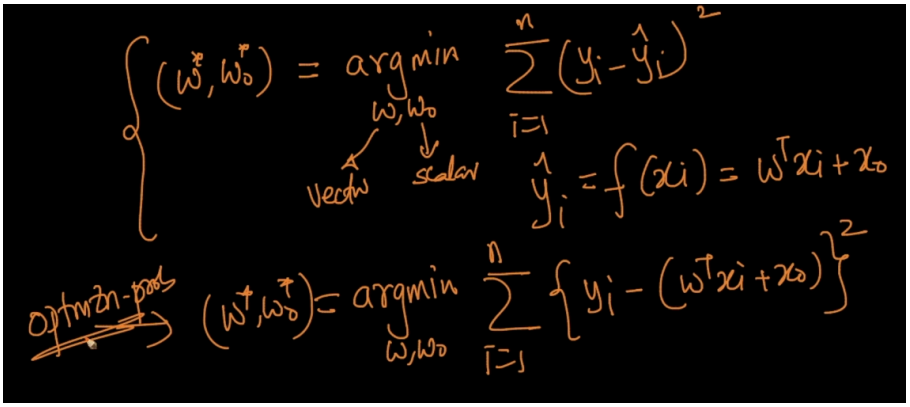
In the above picture, (x1,y1) and (x2,y2) don't lie on the line but (x3,y3) does.
Error e1=y1-y1online,e2=y2-y2online and e3 is 0 since the point (x3,y3) lies on the line.
The intuition is to minimize the error across all the points in training data.

Mathematical Formulation:

In the above image, e1 is +ve since the actual value is above the predicted value or value corresponding to x1 on the line & e2 is -ve. The sum of the error could get cancelled out and the resulting value may not represent the actual scenario.

In order to avoid this, we can try to reduce the square of the errors across all the points.

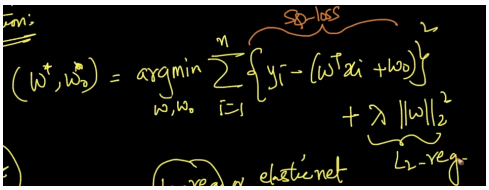
The optimization problem can be given as:



It is often referred to as ordinary least squares or linear least squares. The loss term is called squared loss.

Just like logistic regression, we can apply regularization to avoid overfitting.

We can either use L1,L2 or ElasticNet.

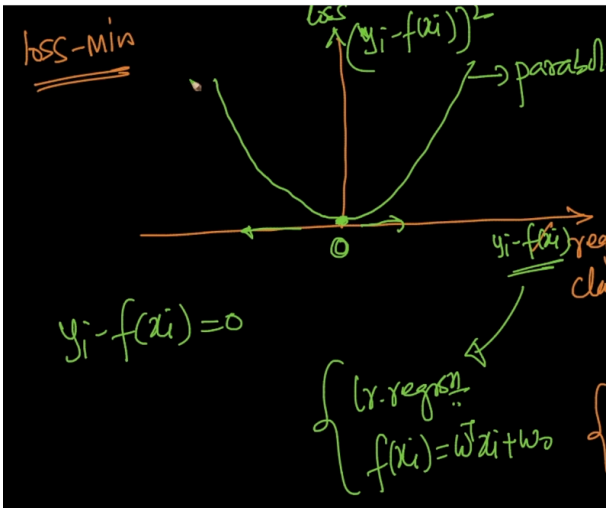


Loss function perspective:

In classification problems, we usually try to maximize sum of yi*f(xi).

In regression we will try to reduce, yi-f(xi)... That means we are trying to reduce the distance between actual and predicted values.

Loss function can be shown as :



We can see from the plot that even if the error is +ve or -ve our loss will be positive and we trying to reduce that.

Real world cases:
Most cases are like Logistic regression.

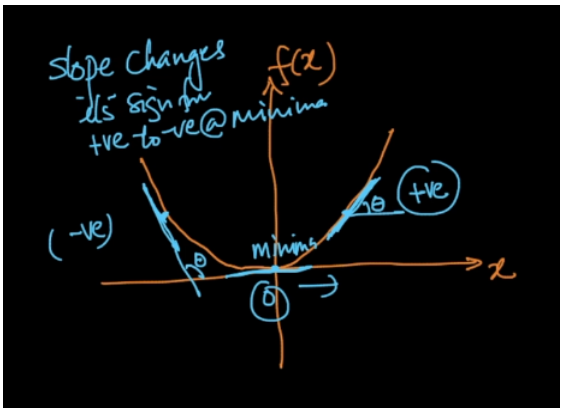
Feature importance & interpretability:
If the data is not multicollinear then we can take abs(weight) as the feature importance.

Feature engineering/Transformations:
Same as logistic regression.

If we use L1 regularization we can get sparsity.

One major difference is the impact of outliers. It'll be high in linear regression. We have sigmoid function Logistic regression that is limiting the error but in linear regression we are squaring the error term and that makes things worse. We can follow the same hack that we explained in Logistic Regression.
That technique is called RANSAC (Random Sampling Consensus).

Gradient Descent:
Not all maximas and minimas will be found out by making their differentiation=0. Some problems are very very difficult to solve.
Gradient descent is an iterative approach which can help us reach the minimum (loss) and it is initialised randomly.

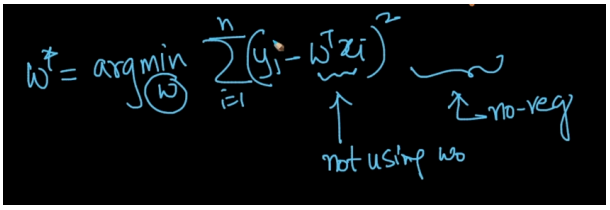


From the above picture, we can see that at minima the slope changes sign. I.e., slope right to minima is +ve and slope left to minima is -ve and at minima is zero. Also, we can see that the magnitude of slopes becomes less as we approach the minima. This is important to understand about Gradient descent.
Steps:

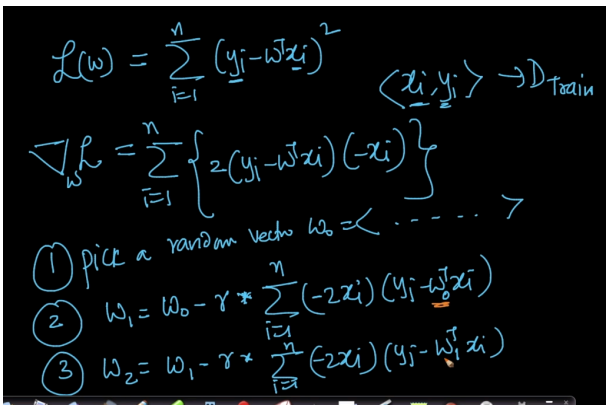
1. Initialize randomly (x0).
2. We need to move towards a point so that we move close to x*.
   Update the value,
   x1 = x0-r*(df/dx) at x0.
   r is called as learning rate and is constant.
   df/dx tells us in which direction to move and by how much we can move so that we can come close to x*.
3. Repeat the steps till we reach x* or our update term(r*df/dx) becomes very very less. i.e., xk+1-xk is very very small.
   And xk will be our x*.

Learning rate should be kept constant. As we go near the minima, the df/dx becomes small and Learning rate will start dominating and we may not reach minima and oscillate around minima. Remedy is to change r with each iteration (reduce).
If r is large, we may diverge instead of converging. If r is very small, our weight updates will be small and our convergence will take lot of steps.

Gradient Descent for Linear Regression:
Let's keep it simple and consider no regularization, w0 as 0.

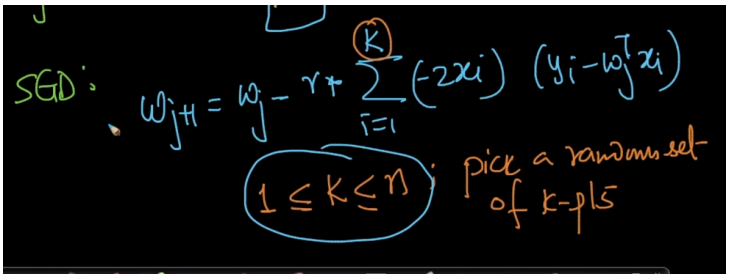


The following are the Gradient descent steps for Linear regression:



and the steps goes on till we converge.

One problem with this is if n is very large, computing the summation is very expensive and time consuming. We have to go through the whole data in order to update the weights once. To solve this, there is a technique called stochastic gradient descent(SGD), where in we need not go through the whole data and still end up solving the problem and get same weights as Gradient descent.

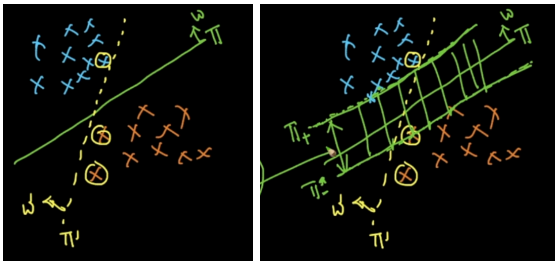


Take a random set of k-points and update the weights at each iteration.
Suppose if we converge in 1000 iterations in Gradient descent, SGD may take 10K or 100K iterations, but still we converge.
When k=1, it is called SGD and k>1 it is called batch-SGD.

Why does L1-regularization creates sparsity?

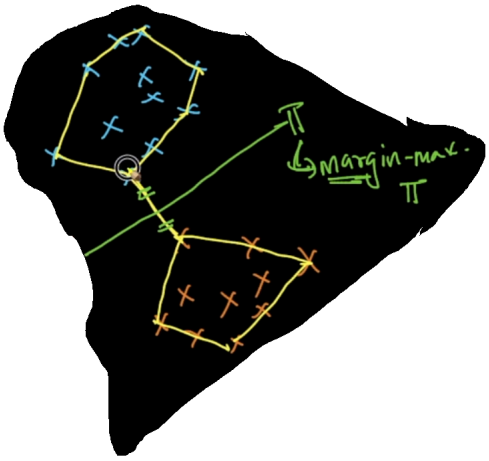
Support Vector Machines(SVM)

Intuition:
Suppose there are many hyperplanes that separate +ve and -ve classes, we try to find a hyperplane that separates +ve points to -ve points as widely as possible.

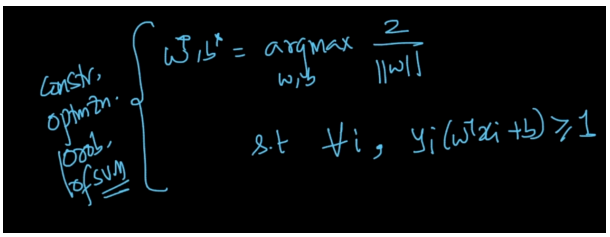


In the above figure Π is a better separating hyperplane than Π'. Since Π' has -ve and positive points closer to it. But Π has the boundary points far away from it. Π here is called margin maximising hyper plane.
Π+ is the hyperplane which is parallel to Π and touches a +ve point(1st) and Π- is the hyperplane which is parallel to Π and touches a -ve point(1st). Distance between Π+ and Π- is called the margin.
SVMs try to find a hyperplane that maximises the margin. As margin increases, the generalisation accuracy increases.
Support Vectors: The points through which Π+ and Π- passes through are called support vectors.
Alternative Geometric intuition to SVMs:

1. Construct 2 convex hulls using all the +ve points and all the -ve points.
2. Find the shortest line connecting the hulls.
3. Bisect the line and that is the margin maximising Π.

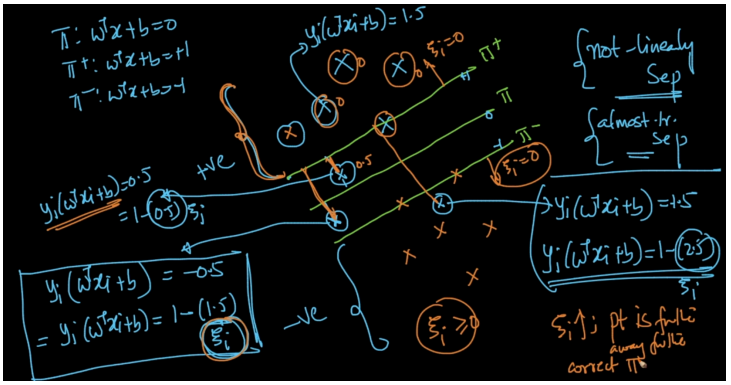


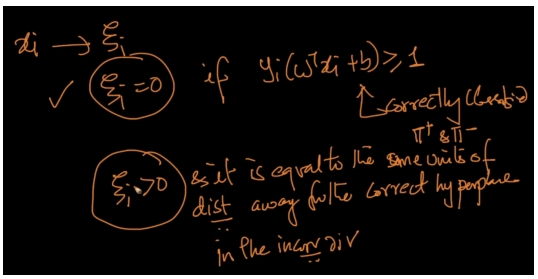
Mathematical Formulation:
Let Π be the margin maximising hyperplane.
Then Π is $w^T x+b =0$
Π+ is $w^T x+b =+1$ and Π- is $w^T x+b =-1$
Margin is given by $2/||w||$.
We need to maximise it.
The mathematical formulation will be to maximise $2/||w||$.

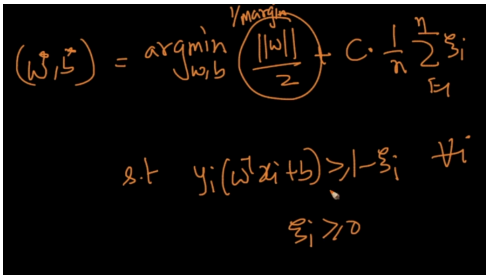


This works when our data is linearly separable. What if it's not exactly linearly separable but almost?
We may never be able to solve the problem stated above. This formulation is called hard margin SVM. We have to modify the formulation to make this work for the cases where its not exactly linearly separable but almost.

For all the points we create a new variable called **ζ**i such that if a +ve point lies above Π+ its **ζ**i =0 and if a -ve points lies below Π- its **ζ**i =0. If a +ve point lies anywhere else it's **ζ**i >0 and if a -ve points lies anywhere else it's **ζ**i >0. If the distance of the point away from the correct hyperplane increases, its **ζ**i increases.



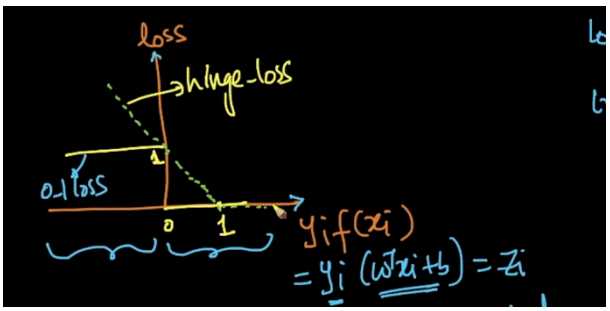




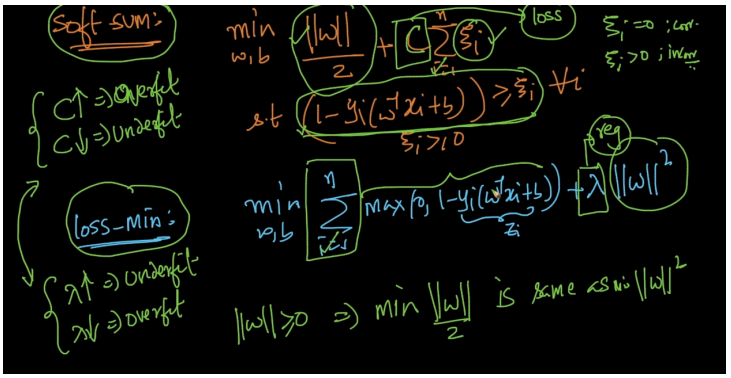
ζi can be interpreted as the average distance of misclassified points from the correct hyper planes.
2nd term including **ζ**i is like loss and the 1st time is like regularization.
C is the hyperparameter which is like 1/lambda.
As C increases, overfitting tendency increases, as C decreases underfitting tendency increases.
This formulation is called soft margin SVM.

Why did we take +1 and -1 for Π+ and Π- equations?
It can be +k and -k also. It doesn't matter if we are maximising for w,b on 2/||w|| or 2*k/||w||.

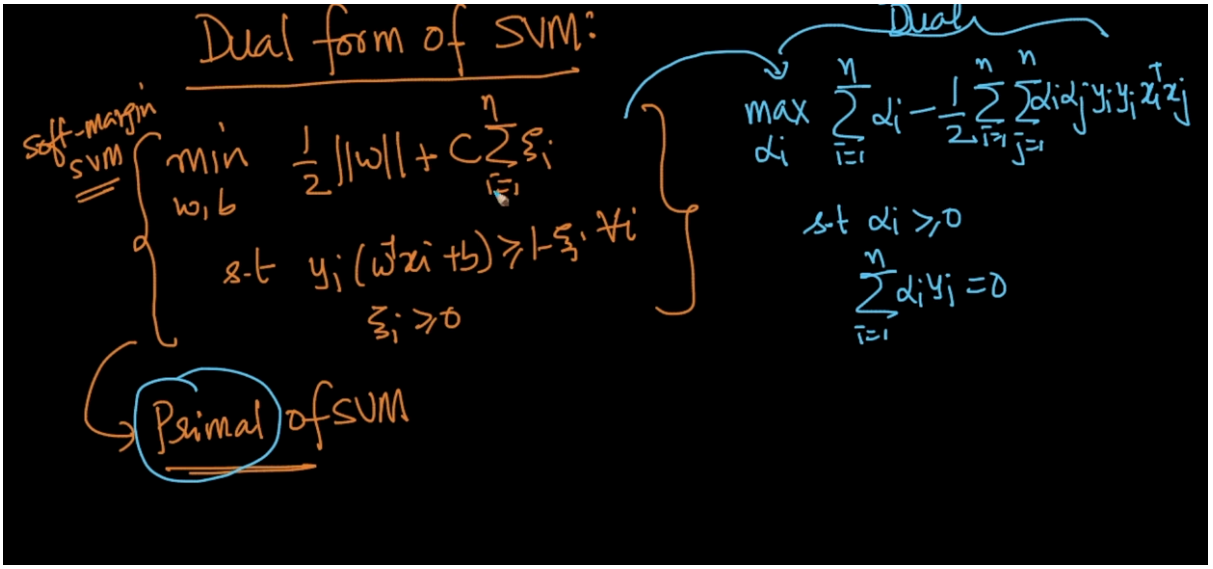
Loss minimization-Hinge Loss based interpretation:
If zi>=1, Hingeloss is 0.
Else Hingeloss is 1-Zi Hinge loss is not differentiable at 1. But there are hacks.



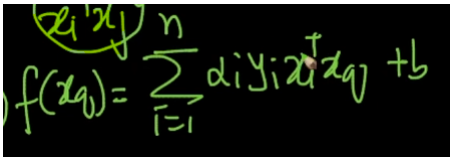
Hingeloss can be written as max(0,1-zi)



Dual form of SVM:

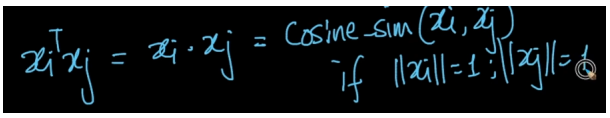


This is a very useful form.
For every xi, there is a corresponding $\underline{\alpha}$i.
xi's occur only as dot product with xj's



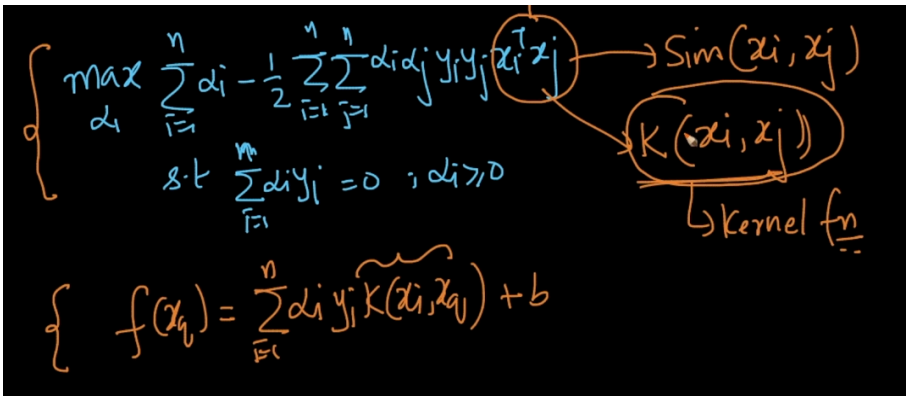
$\underline{\alpha}$i's only existing for support vectors and are 0 for non support vectors.

To compute f(xq) only, Support vectors are required. Everyother points doesn't matter.



Instead of using Cosine similarity of xi and xj we can replace that with any similarity matrix (K(xi,xj)). K is kernel function.

Kernel Trick:



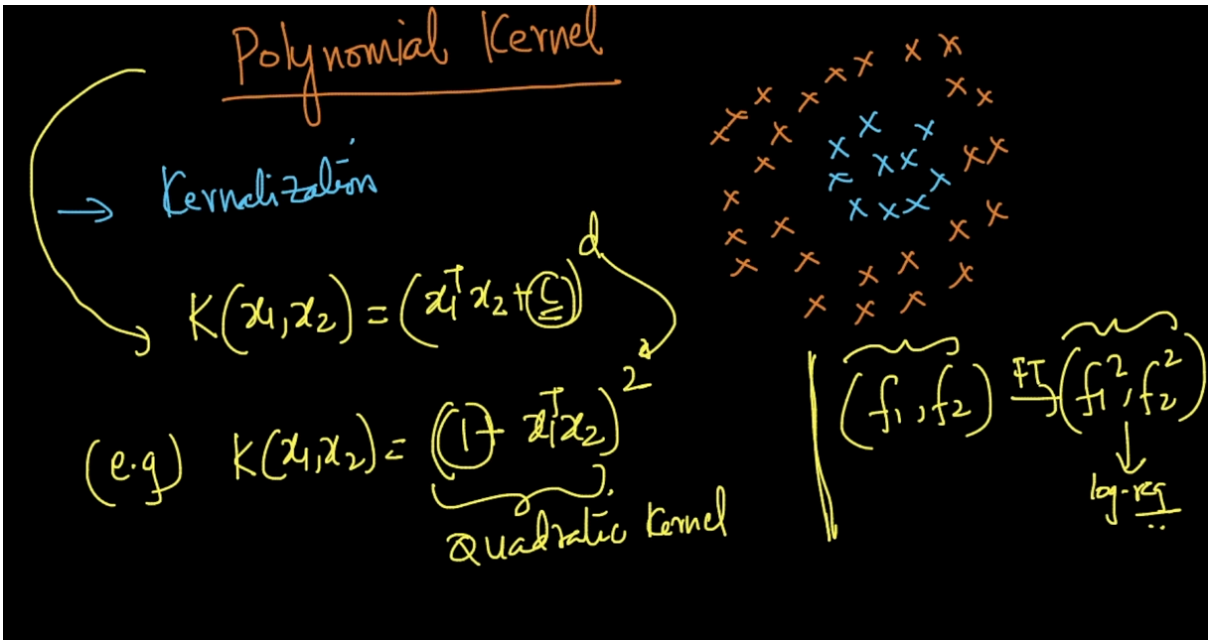
The most important Idea in SVM is Kernel trick.
If we don't replace xi.T*xj with kernel it's Linear SVM. If it's replacing then it's kernel SVM.

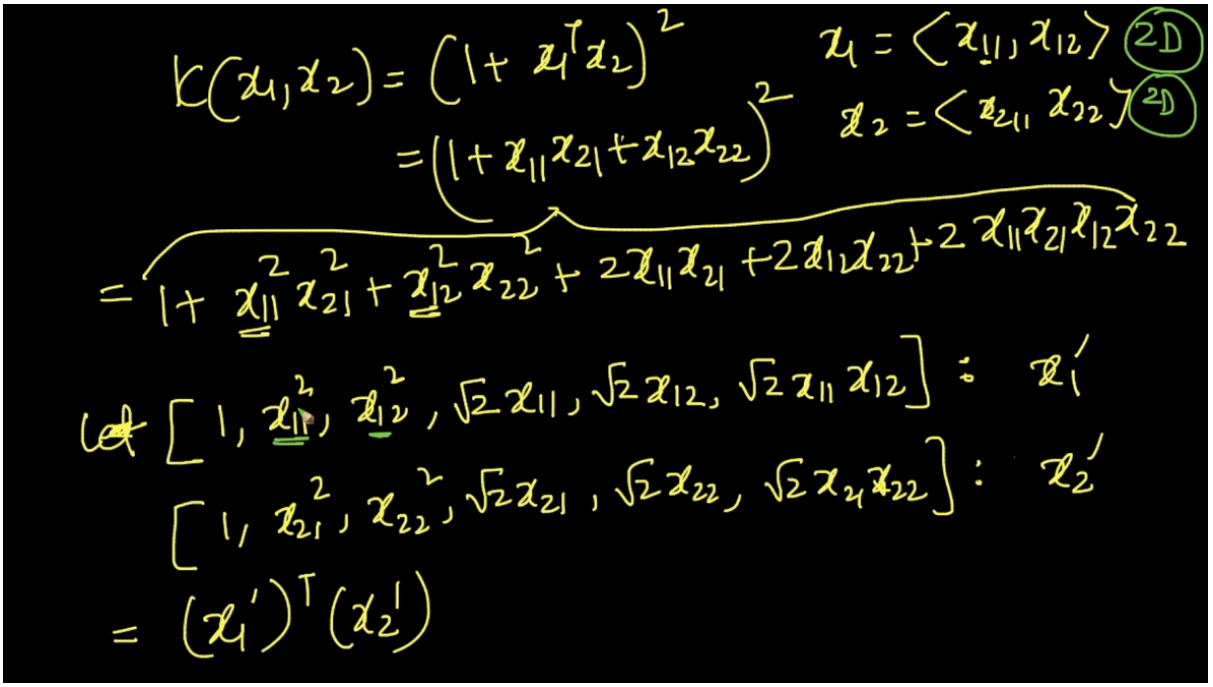
Linear SVM tries to find maximum margin hyperplane in the space of xi's and Logistic regression tries to find a hyperplane that minimises logistic loss in the space of xi's. Hence sometimes their results look similar.

There are different kinds of kernels that can be used.

Kernel-SVM using kernel trick can handle non linearly separable datasets also.

Polynomial Kernel:





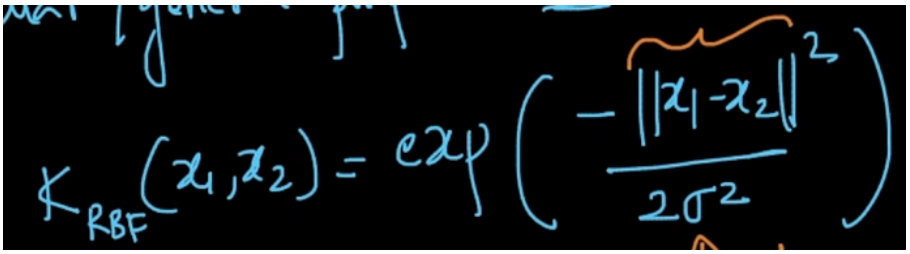
Kernelization takes data in d dimension and internally mapping this data to a dimension d' where d'>d.

Kernel SVM implicitly transforms features.But in Logistic regression, we explicitly transform features.

Challenge is what is the right kernel for a dataset.

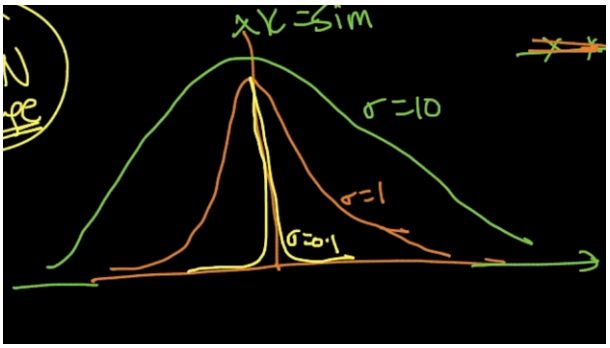
RBF Kernel:

General purpose kernel and most used.



Sigma is a hyperparameter along with C.

As distance between 1 and 2 increases, K decreases. (like similarity).



If Sigma is less, the points which are very close to the given point have non-zero similarity values. But as Sigma increases the points which are farther also can have non-zero similarity value.

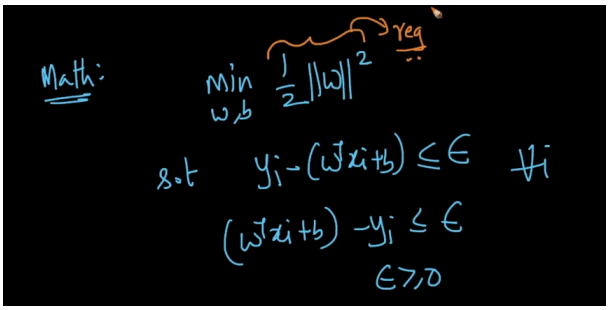
A large value of Sigma in RBF kernel is intuitively equivalent to higher k in k-NN and a small value of Sigma is equivalent to lower k in k-NN.

If we don't know which Kernel to use, we can directly use RBF. There are 2 hyper parameters, Sigma and C.

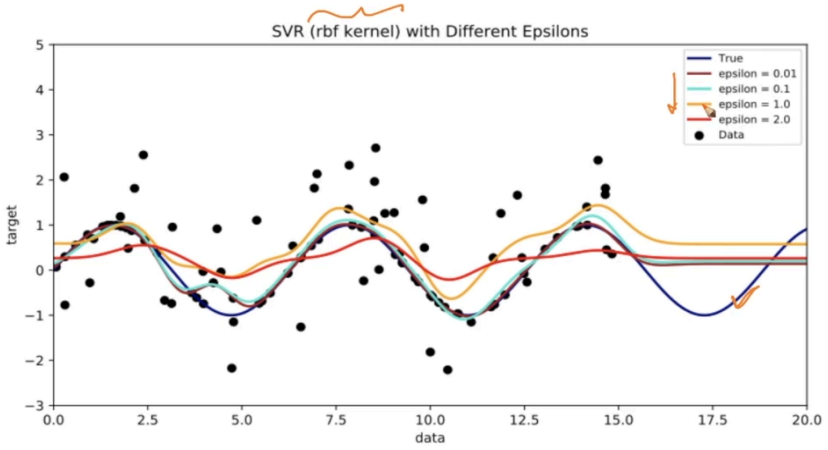
Train and Run time complexity:
Optimization problem solved for dual formulation: Sequential Minimal Optimization.
Training time: O(n*n)
Typically don't use SVM when n is large.
Run time: Depends on the number of support vectors k.
O(k*d)
There is no clean way of how to control the no.of support vectors.

nu-SVM:
Original formulation is called C-SVM.
Hyperparameter in nu-SVM is nu.
0<=nu<=1.
nu is an upper bound on fraction of errors and lower bound on fraction of support vectors.

Support Vector Regression (SVR):
This can be kernalized as well to fit any curve. ϵ is the hyper parameter.



ϵ is very less, overfitting tendency is more.
ϵ is very high, underfitting tendency is more.



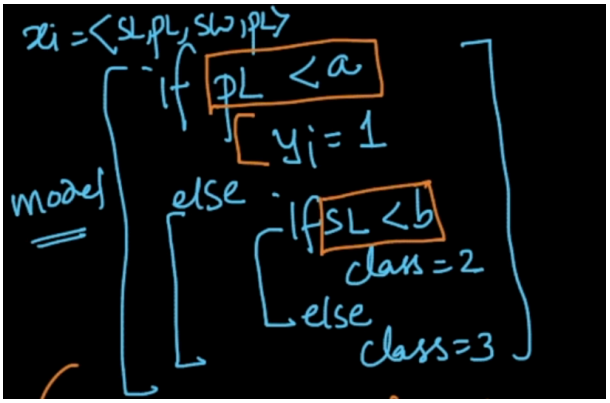
Real world cases:
Feature engineering/Transforms: It's about finding the right kernel.

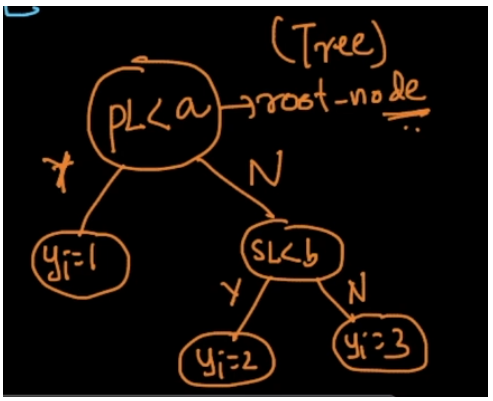
Decision surface: for linear SVM, the decision surface is a hyperplane/linear surface. For kernel SVM it is a non linear surface in original space but linear in transformed space.
Similarity/Distance function: Can be used in training SVM.
Interpretability/Feature Importance: For linear SVMs,We can't easily get the feature importance.
For linear SVMs it's like Linear/Logistic Regression.
Outliers: Outliers have very little impact since only Support vectors matter and not all the data points. RBF SVM with small sigma can be impacted by outliers.

Large dimensionality is very good for SVM.

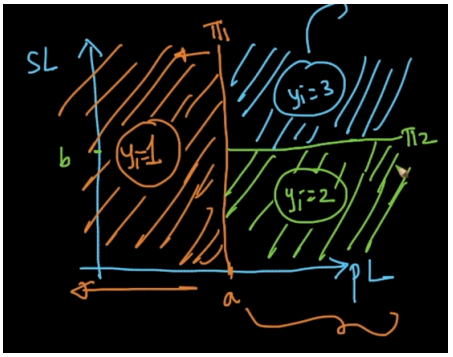
Decision Trees

Geometric Intuition:
Similar to nested if else conditions.



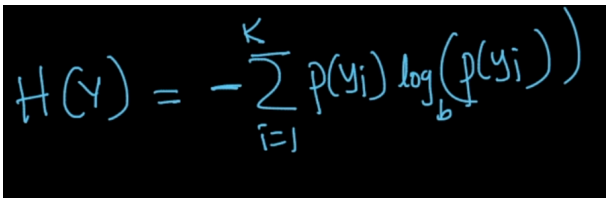


Root node: 1st node in a tree.
Leaf node: all the end nodes are leaf nodes.
Internal node: A node which is neither root node nor leaf node.

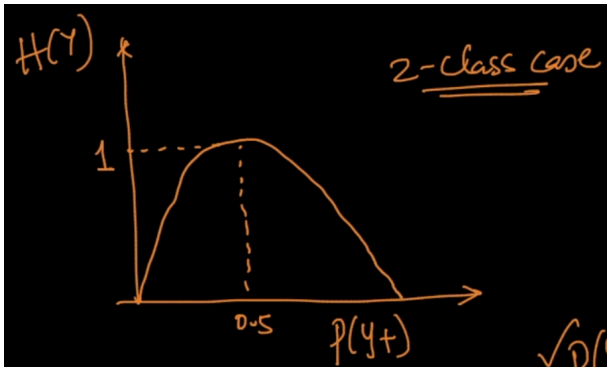


All the hyperplanes are axis parallel. Decision tree is a set of axis parallel hyperplanes.

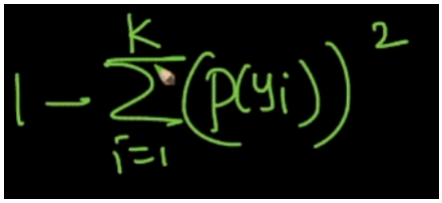
Entropy: Measure of randomness in the data



If the classes are equally probable then the entropy is maximum and if the decision making easy 100% single class then the entropy is 0.



Information Gain: Information gain at every step is the difference between the weighted entropy after split - entropy before split.
Gini Impurity: Similar to entropy but more simple.



Max value of Gini Impurity occurs when all classes are equiprobable(like Entropy) and the max value is 0.5.
Min value of Gini impurity is 0 when the decision making is easy 100% single class.
Calculation is simple here since there is no log.

How do we construct a decision tree?

1. Choosing the root node: Calculate Information gain for each feature and then select the one in which information gain is high.
2. If any of the node is pure node (0 impurity/ 100% single class, we don't have to go further).
3. For impure nodes follow step-1.
4. If we don't have enough points to continue we'll stop there. (we won't go a node further if we have very few points---Hyper parameter).
5. If we are too deep we'll stop (Depth: Hyperparameter).

We recursively break each node based on the Information Gain.
Depth of the tree is the main hyperparameter: High depth Overfititng and Low depth Underfitting. As depth increases interpretability of the model reduces.
How does a split happen in continuous features?

1. Sort the features in ascending order.
2. For each point calculate information gain, f1<that point value.
3. Select the point which gives us the maximum information gain.

Feature Standardization:
It's not required for Decision trees since it is not a distance based method.
Categorical features with many categories:
Example, PIN codes. (1000s of levels). And data is small.

One feature engineering hack, Instead of using the PIN code as a categorical feature, convert it to continuous feature. Calculate P(yi=1|Pincodej) and then replace it with Pincodej. So, instead of having many categorical values, we can have continuous feature which makes things easier for splitting to happen.

Overfitting and Underfitting:

High depth: Over fitting
Low depth: Under fitting

Train and Run time complexity:

Train time: O(n*log(n)*d)
When we have numerical features, calculation is high.
If we have large d Decision trees may not be best option.
Runtime Space: O(no.of internal nodes + no.of leaf nodes) reasonable.
Runtime time complexity: O(depth).

Decision trees are very good, if we have large data, less dimensionality, low latency applications.

Regression using Decision Trees:

Instead of Information gain, splits happen based on change in the regression metrics like mean squared error, Mean Absolute Deviation, Median Absolute deviation. It should reduce the weighted error after split.
Yi predicted = average of all the yi's in that bucket.

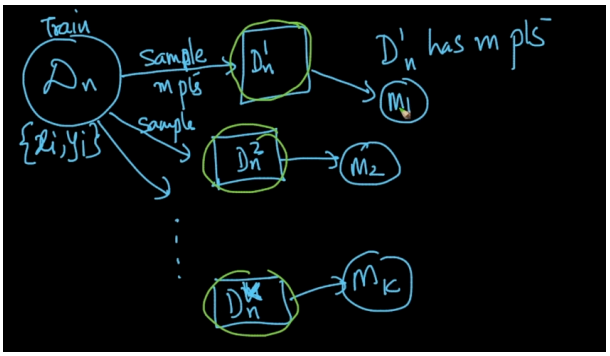
Real world cases:

Imbalance data: balance it. Imbalance effects Entropy calculations.
Large d: train time complexity is very high.
Categorical Feature: We should avoid one hot encoding as it increases the dimensionality.
If given a similarity matrix, decision trees cannot work.
Multiclass classification works.
Decision surface is non-linear.
Feature interactions: Feature interactions happen to decide the class label.
Outliers: Affected by outliers especially if depth is large.
Interpretability is there.
Feature importance: Can be calculated by knowing the reduction in entropy/gini impurity due to the features.

Ensembles:

Ensemble is a group of things. With respect to ML, it is multiple simple/base models combined together to become a more powerful model. 4 Types of ensembles: Bagging (bootstrapped aggregation),Boosting,Cascading,Stacking. The more different the models are, the better we can combine them.

Bagging:

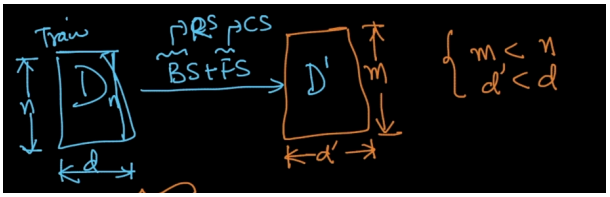


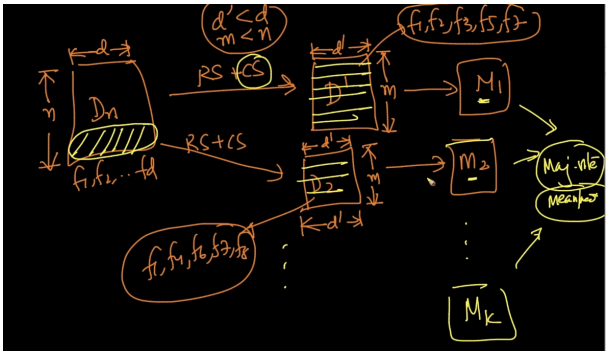
Sampling with replacement(bootstrap), we will create multiple datasets and build models M1,M2,...Mk. Combine the models to make a strong learner (aggregation).
Aggregation for Classification: Majority vote, for Regression: Mean/Median.
None of the models, see the whole data. If the data changes, models can change a lot implying the presence of high variance.
M1,M2,...,Mk can each have high variance but the aggregated model won't have high variance.
Bagging can reduce variance in a model.
Bunch of low bias, high variance models combined to form low bias and reduced variance model.

Tree when grown to the max. Depth is an example of low bias and high variance model.

Random Forests:
Most popular bagging algorithm.
Base learners are decision trees.

We'll do row sampling(selecting rows randomly for training each model) as well as column sampling(selecting few features randomly for training each model).





Depth of base models is reasonable...It can be max depth as well.
All the individual models trained can be very different from each other since we are doing row sampling as well as column sampling.
The points that are not part of training a Model are called out of bag points for that model. It can be used as a CV data for that model.

Bias-Variance tradeoff:
Bias is always the bias of individual models.
As the no.of models (k) increases, variance reduces.

No.of estimators (k) is a hyper parameter.
Row sampling ratio, Col sampling ratio can be tuned as well. If they are less then the data received by each individual models will be very different. And they shouldn't be too less to train.

Train and Runtime complexity:
Train time complexity: O(nlogn*d*k).
But since the models are independent we can parallelize the training process.
Run time complexity: O(d*k).
Space complexity: O(nodes*k).

Extremely Randomised Trees:
Difference from RF/DT is how they handle continuous data.
Randomly select few points and then select the threshold out of them.
Extremely Randomised Trees: DT+Row sampling+Col Sampling+randomised Threshold selection.
Can reduce variance better than RFs.

Real world cases:
Same DT cases hold true.
Bias-Variance: No.of base learners/Estimators (k)
Feature Importance: Overall reduction in entropy/gini impurity @ various levels for all models.

Boosting:

Base models here will be of low variance and high bias and additively combine them.
A decision tree with very less depth can be considered as high bias low variance model.
Popular algorithms are Gradient Boosting, Adaptive Boosting.
The initial model is trained on the training data. (M0)
The next model (M1) will be trained on the errors in training data.
The predicted output can be given as linear combination of Predictions of M0 and M1.
The next Model (M2) will be trained on the errors in data up to M1 level.
And the process continues.

Gradient boosting:

Input: training set $\{(x_i, y_i)\}_{i=1}^{n}$, a differentiable loss function $L(y, F(x))$, number of iterations M.

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma).$$

2. For m = 1 to M:
    1. Compute so-called *pseudo-residuals*:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

    2. Fit a base learner (e.g. tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^{n}$.
    3. Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$

    4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

Linear Regression