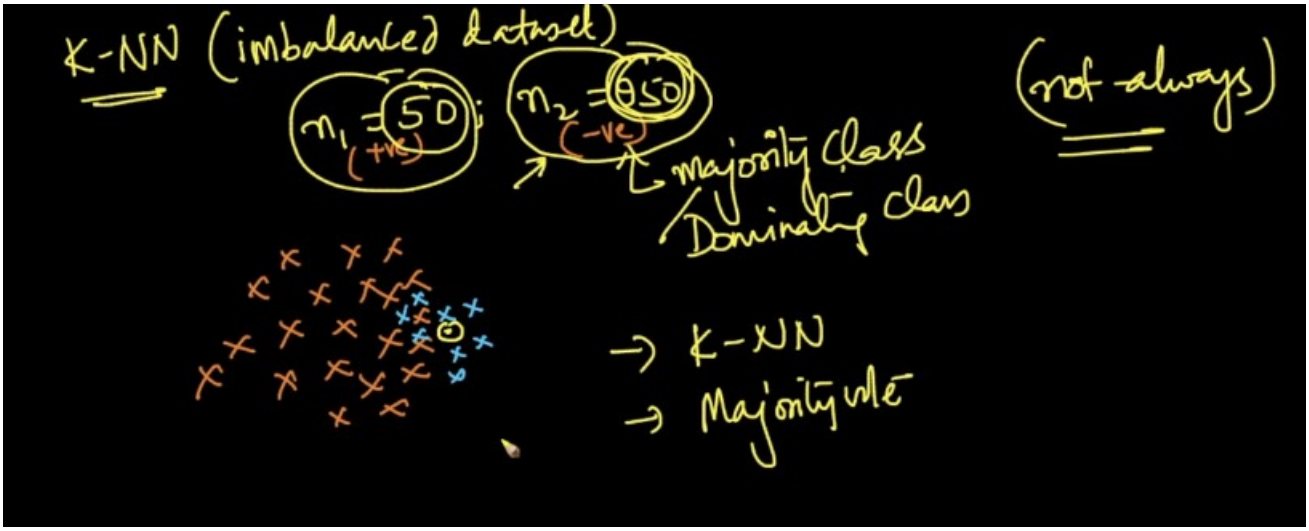
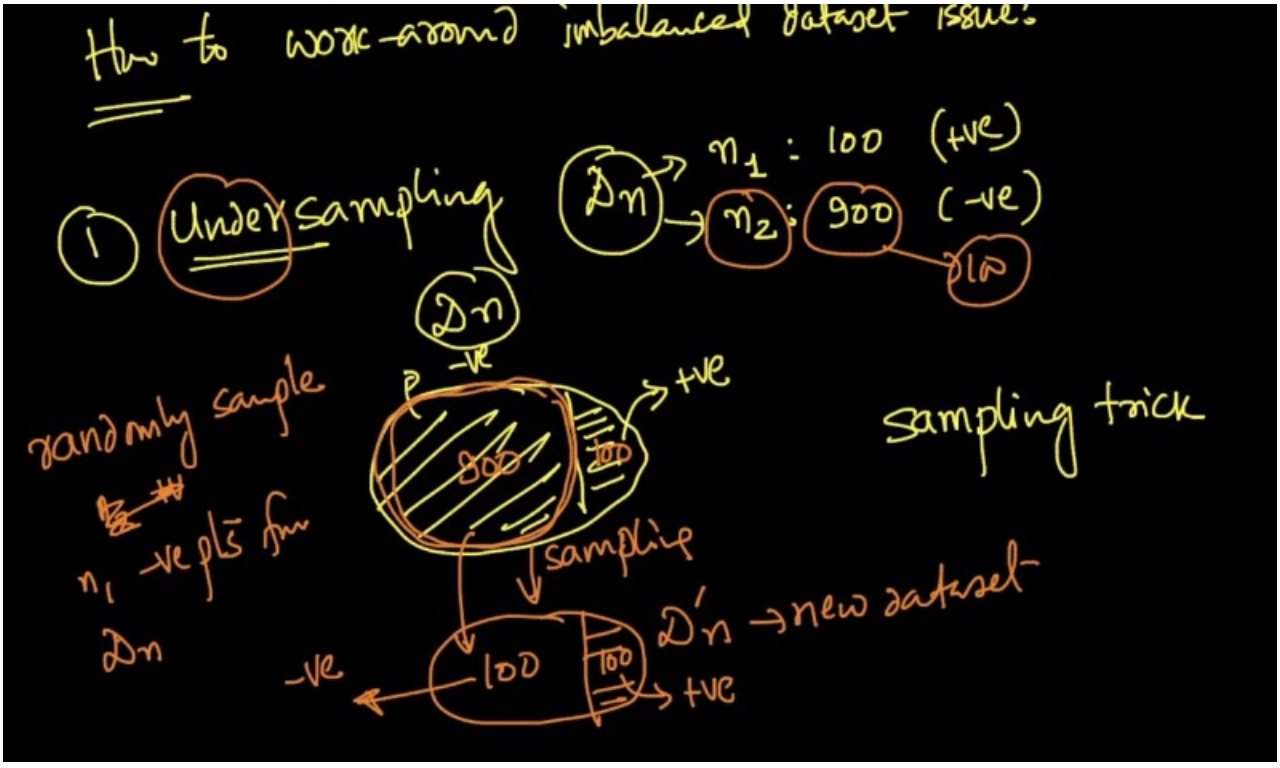
NOTES

Balanced vs. Imbalanced data set:
Imbalanced data-sets:
In case of imbalanced data sets majority class will be dominate over the minority class. In case of KNN the majority will get high priority. This may occur.



Resolving imbalanced data sets:

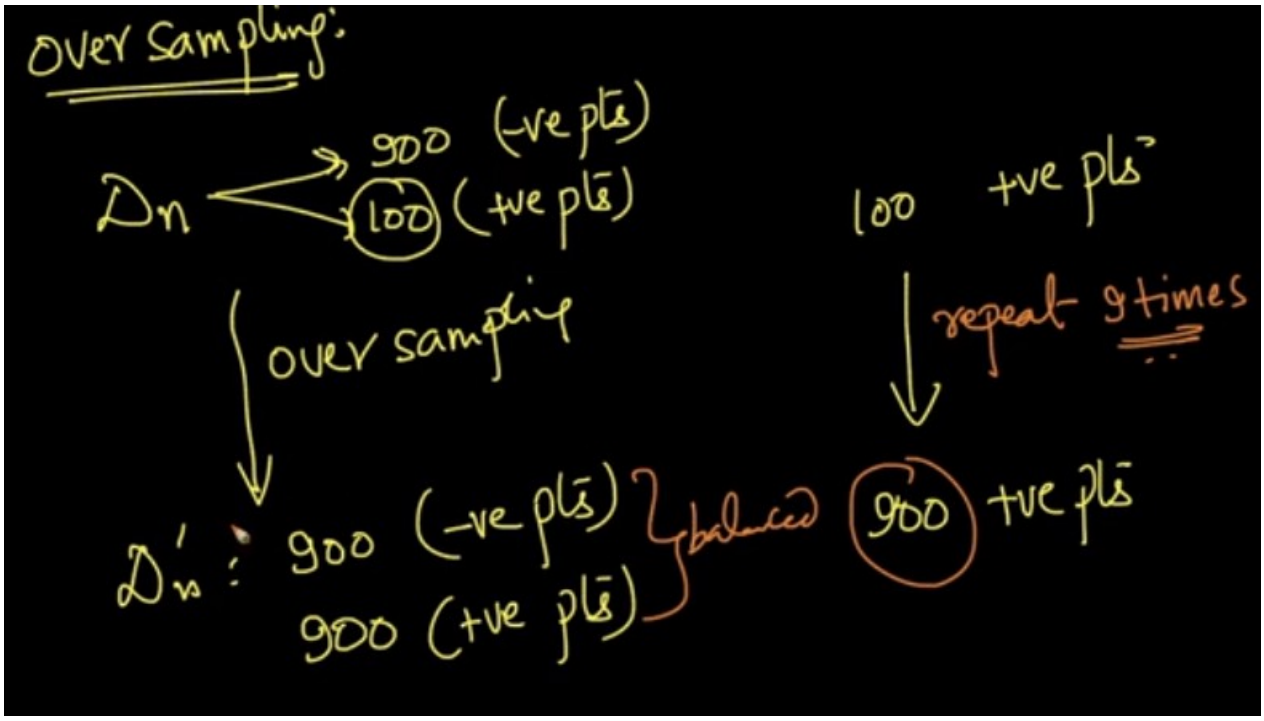
1. Under sampling.
2. Over sampling.



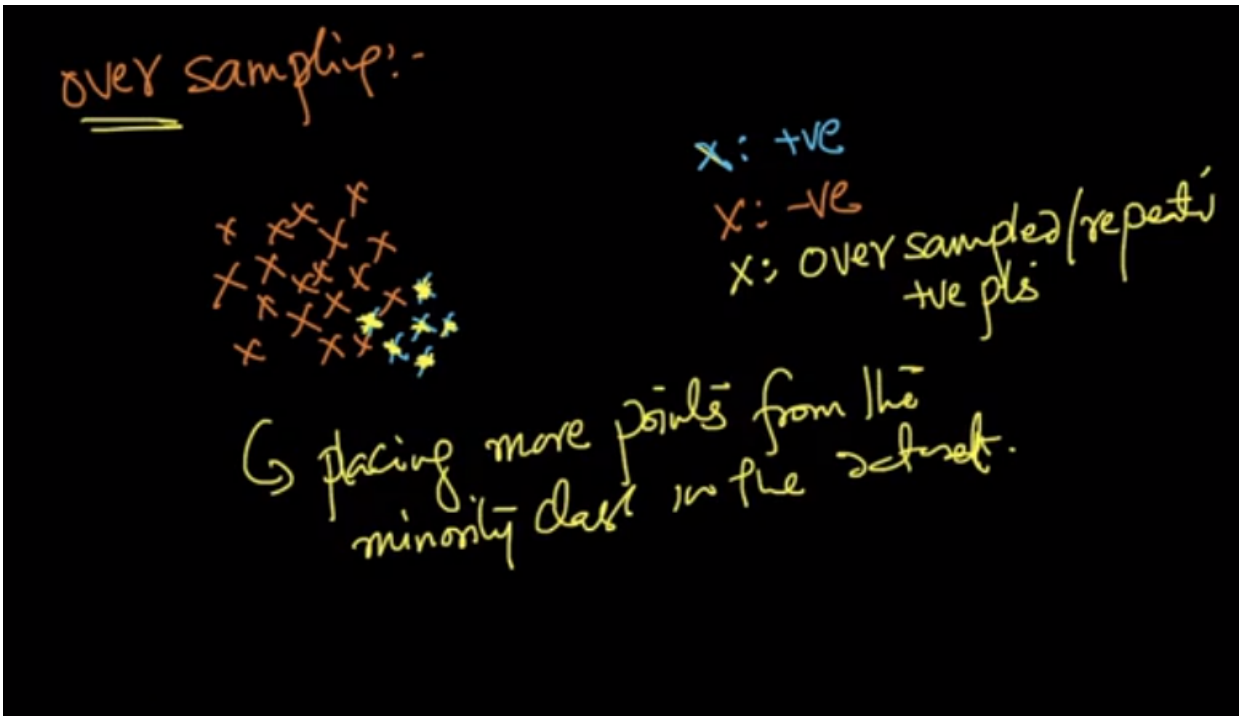
The sampling must be done **randomly.**

There is data loss in case of under sampling. We can overcome this situation by Oversampling the data.

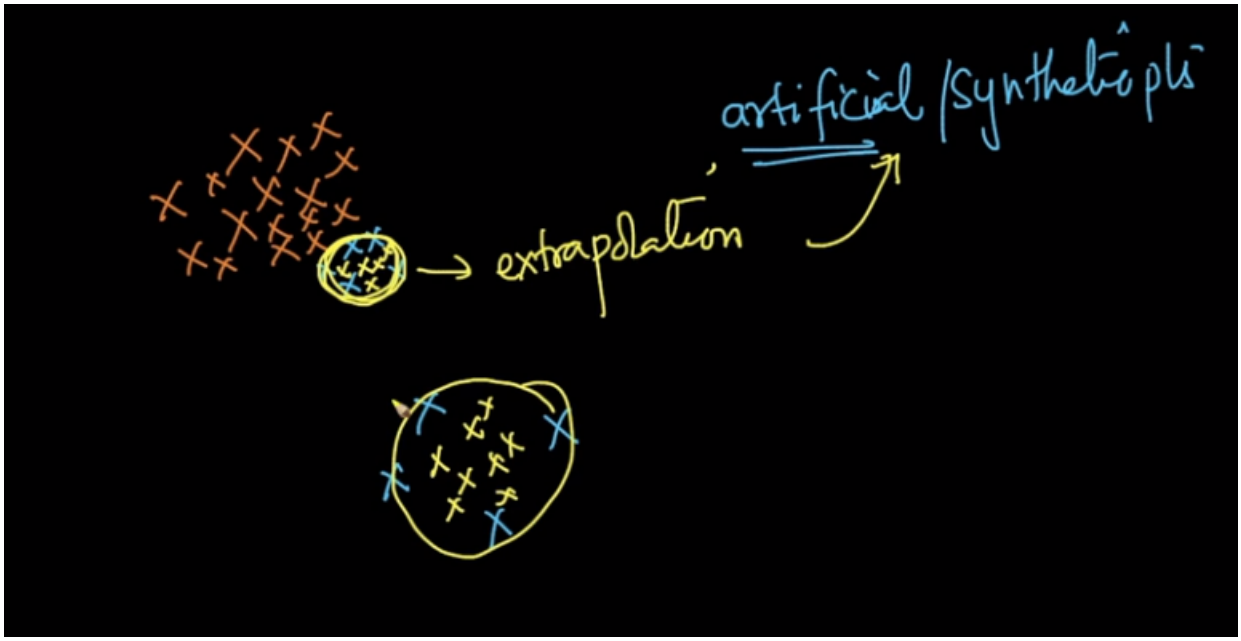
Oversampling is taking the same data point repeatedly.



The data points lie on top of each other.

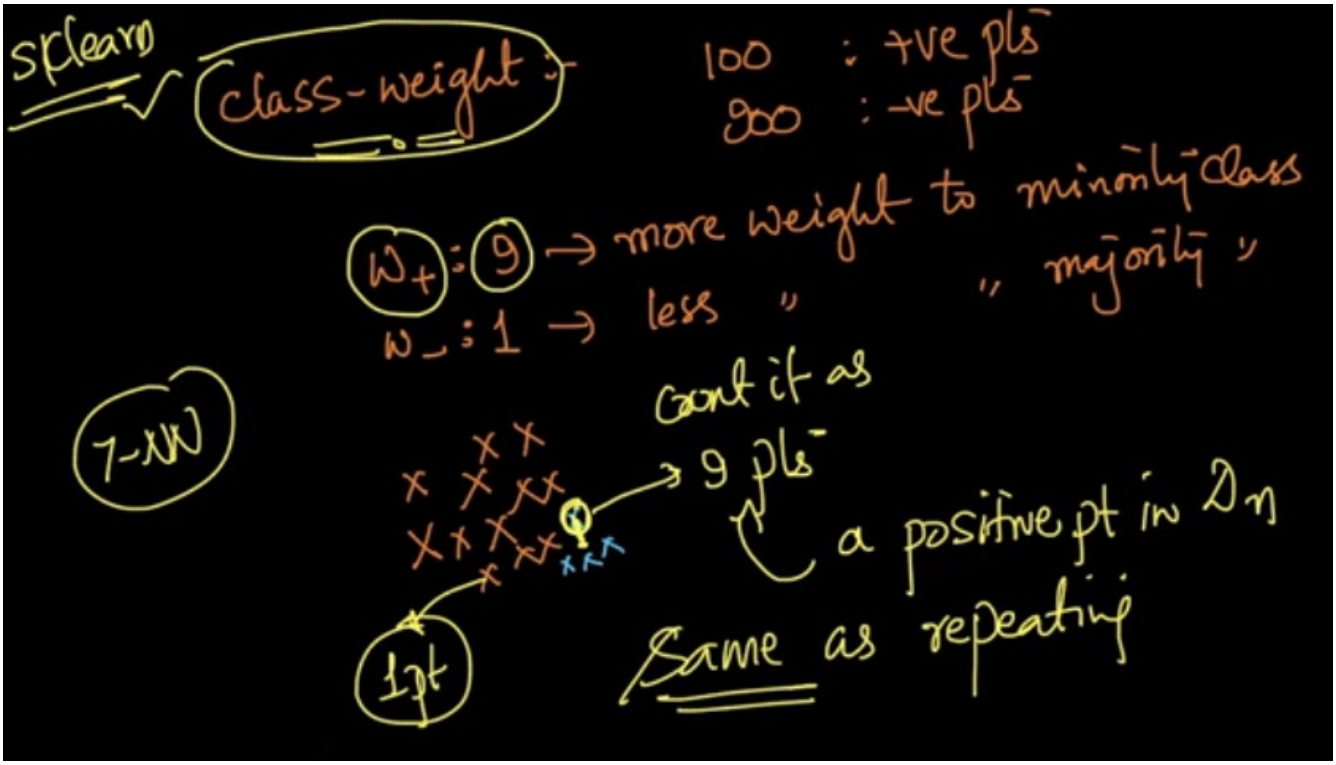


Making synthetic points is the other alternative to oversampling. There artificial points are created with in the same class points.

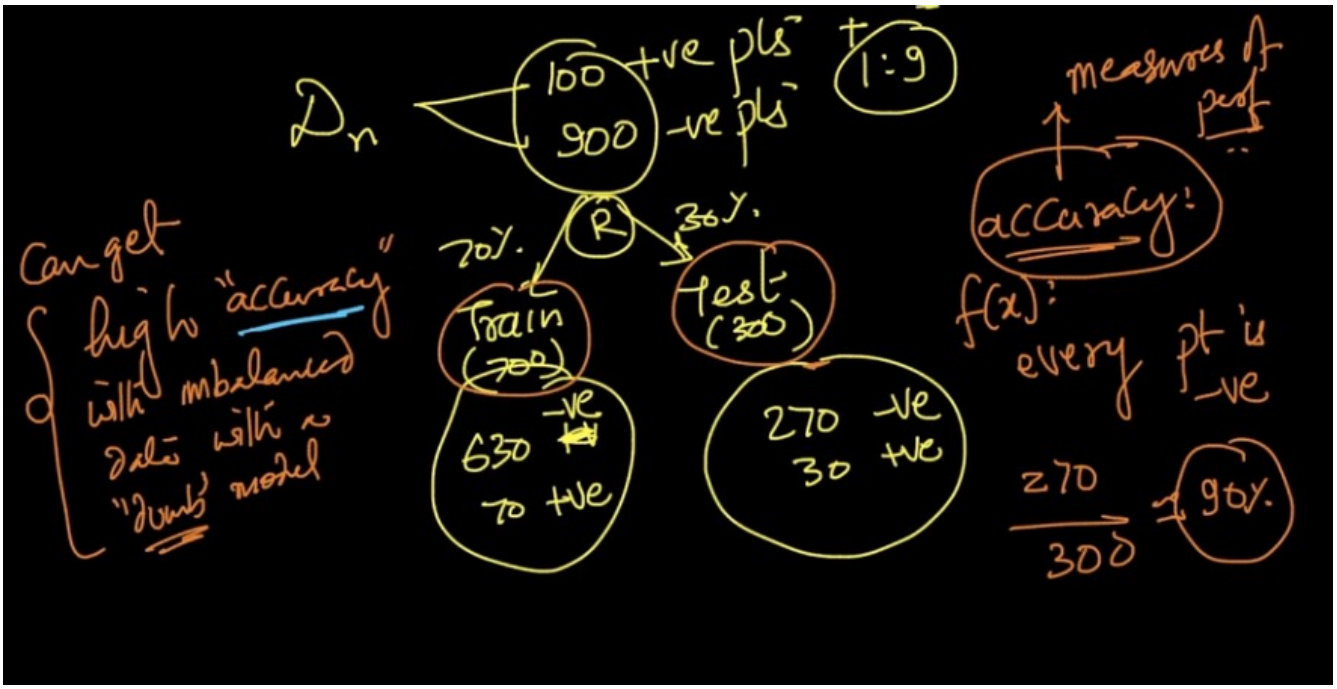


Implementation of class – weight for oversampling:

The more weight is given to the minority class the less weight to the majority class. This is same as repeating the points.

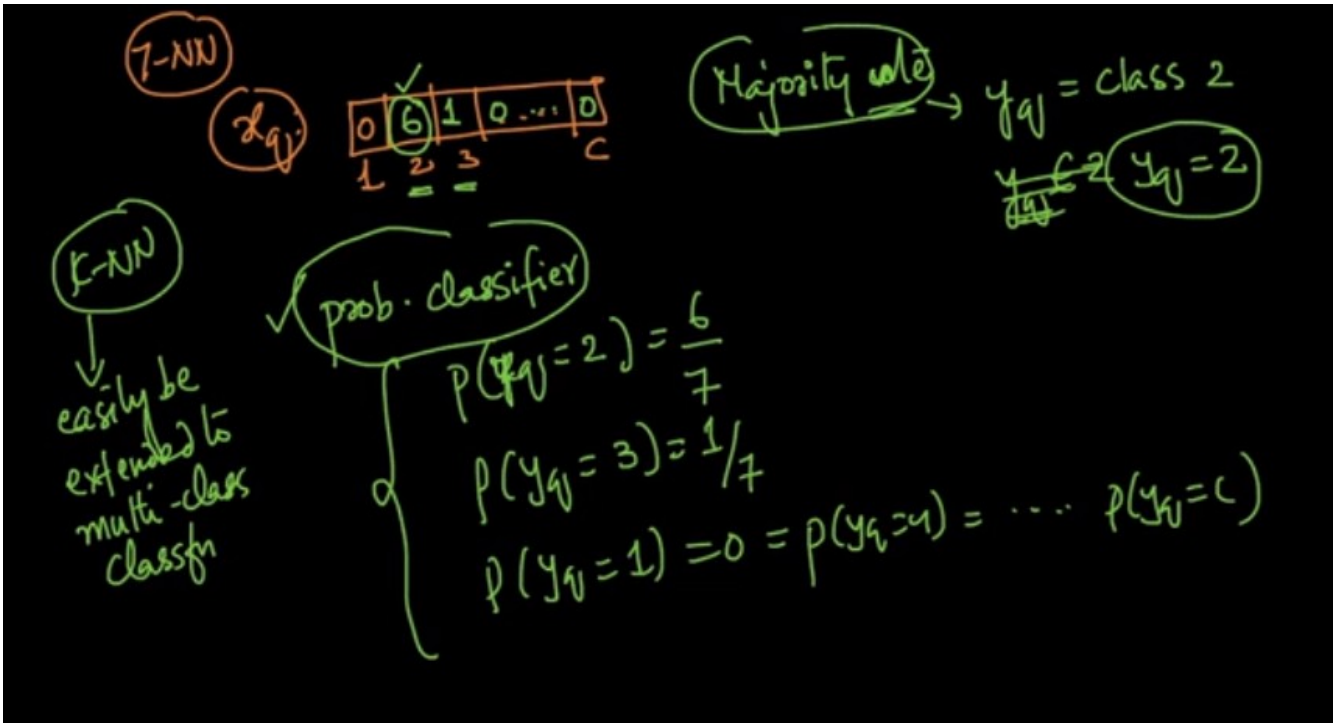


In case of Imbalanced data sets we can get high accuracy with a dumb model.

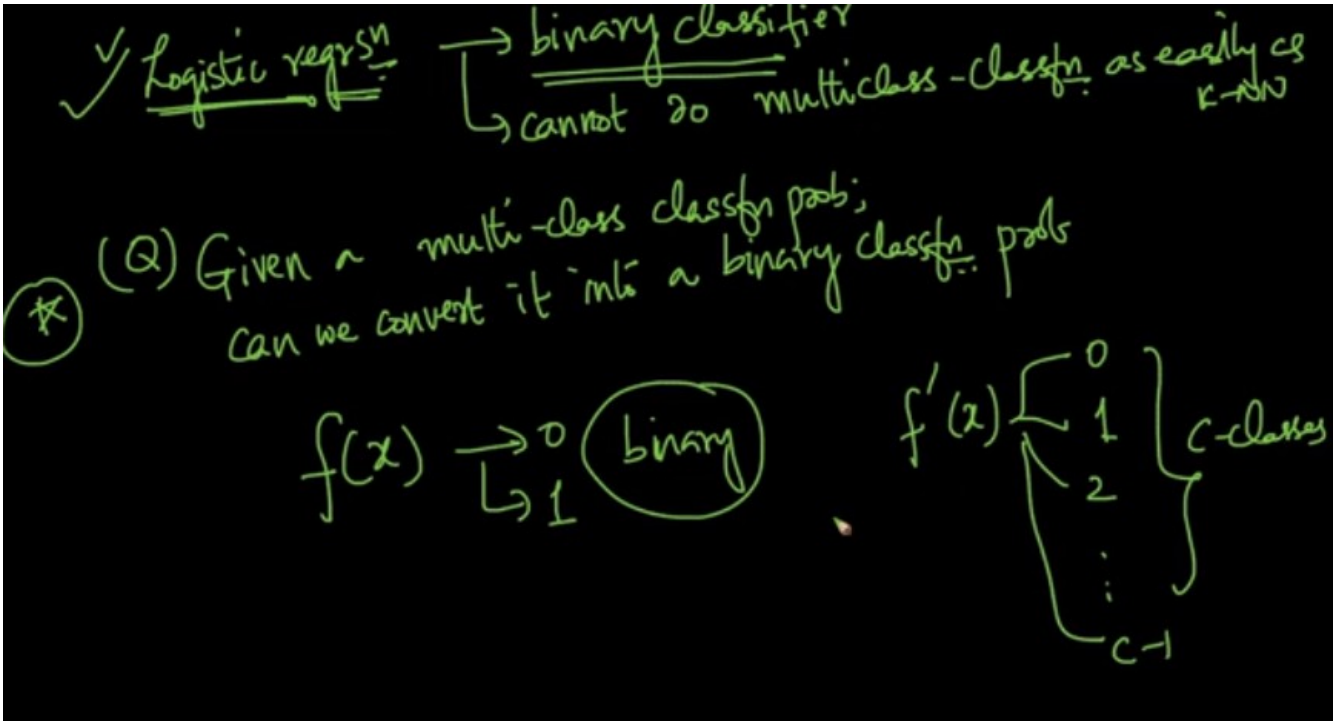


Trying oversampling is the good practical strategy.

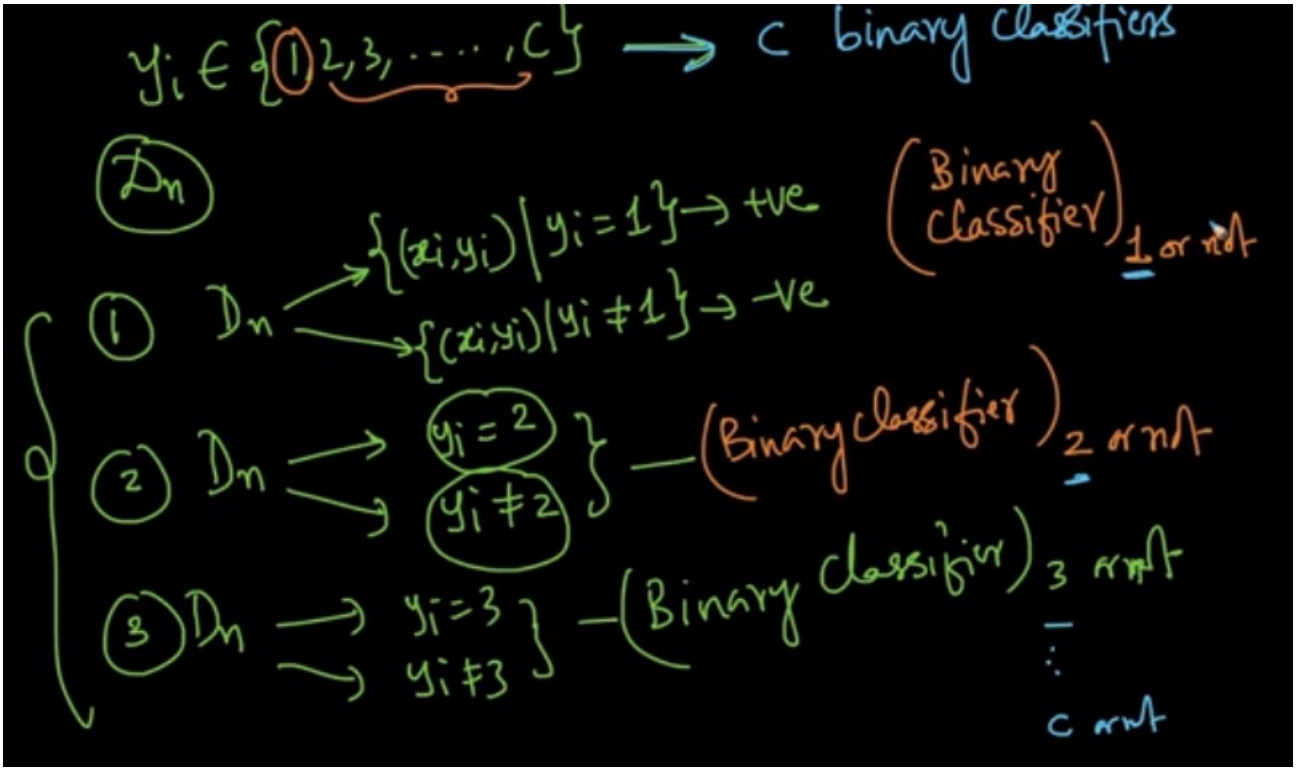
Multi – class classification:
Example:
K – nearest neighbors: KNN can be extended to multi – class classification by using probability scores.



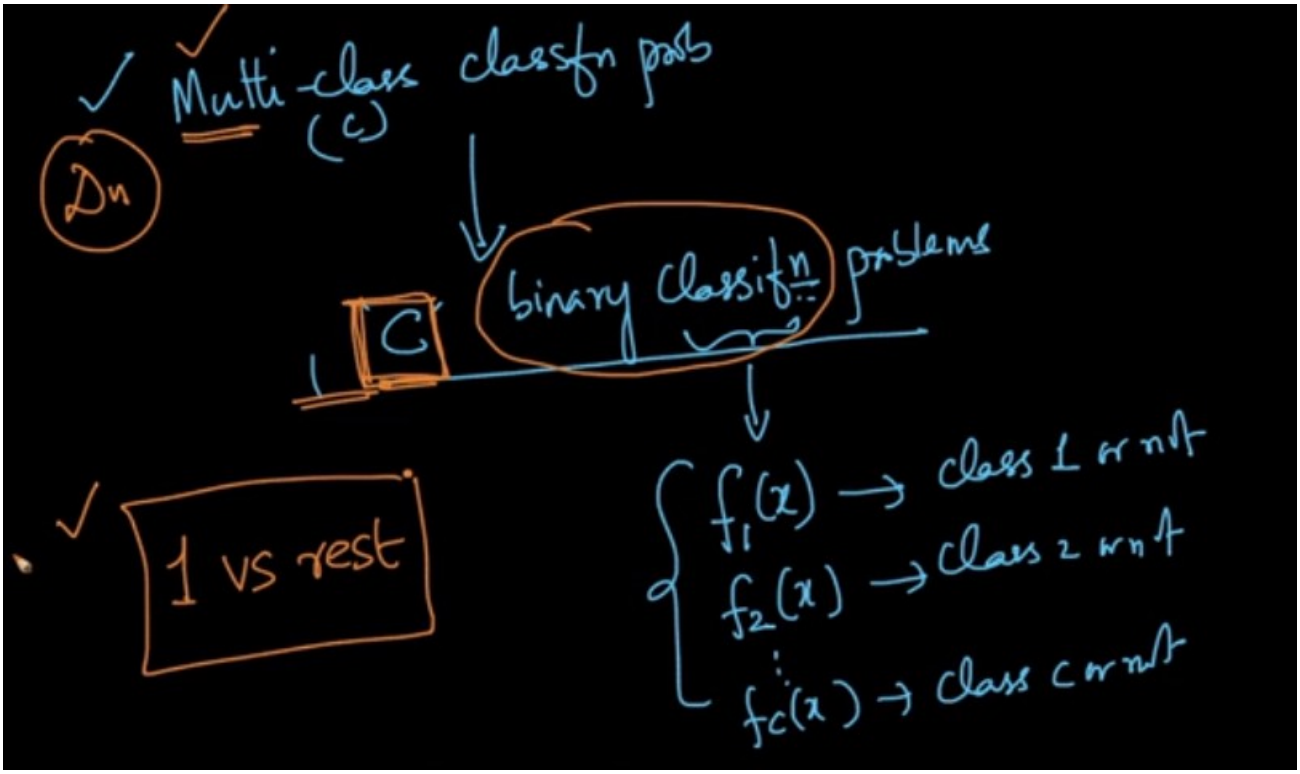
Logistic regression can only do binary classification, that can be extended to multi – class by using soft - max classification.



Breaking the data – set to small parts with each class. If we have ‘c’ classes we can break into ‘c’ binary classification problems.

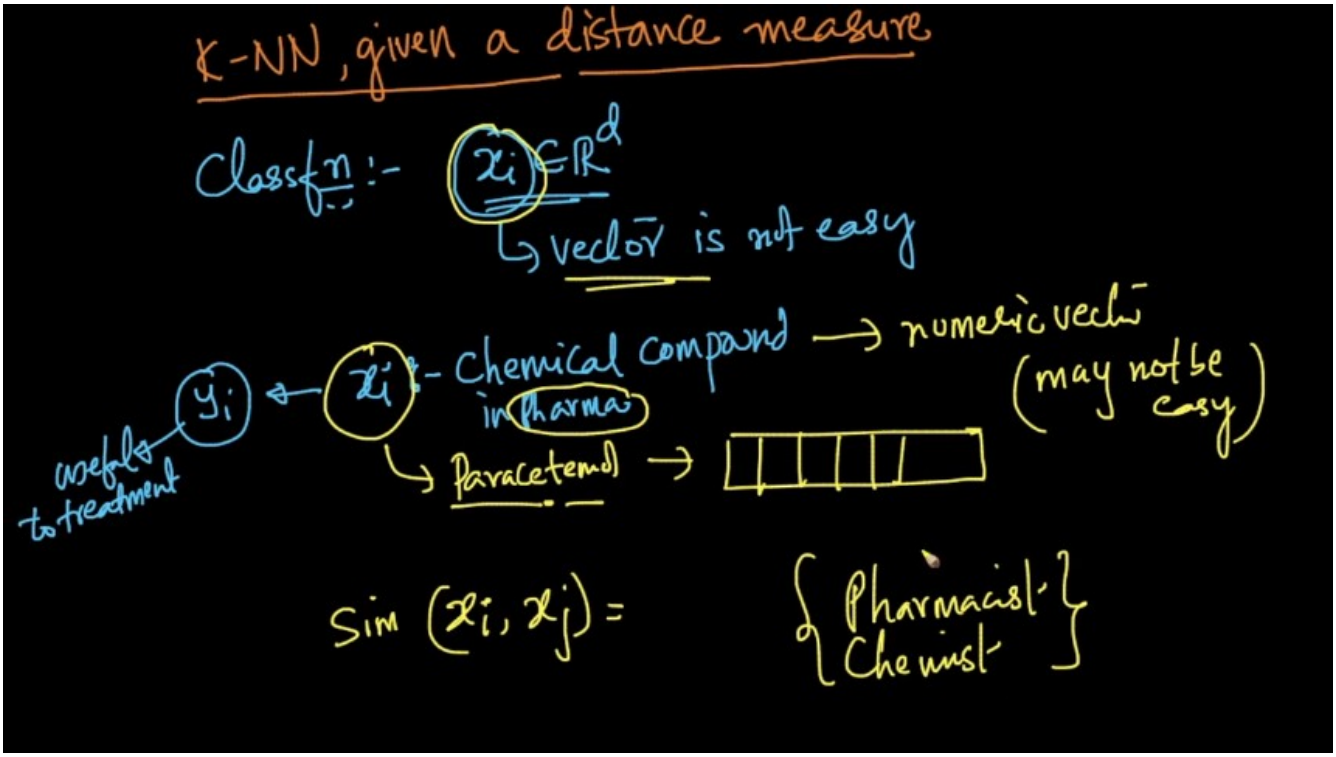


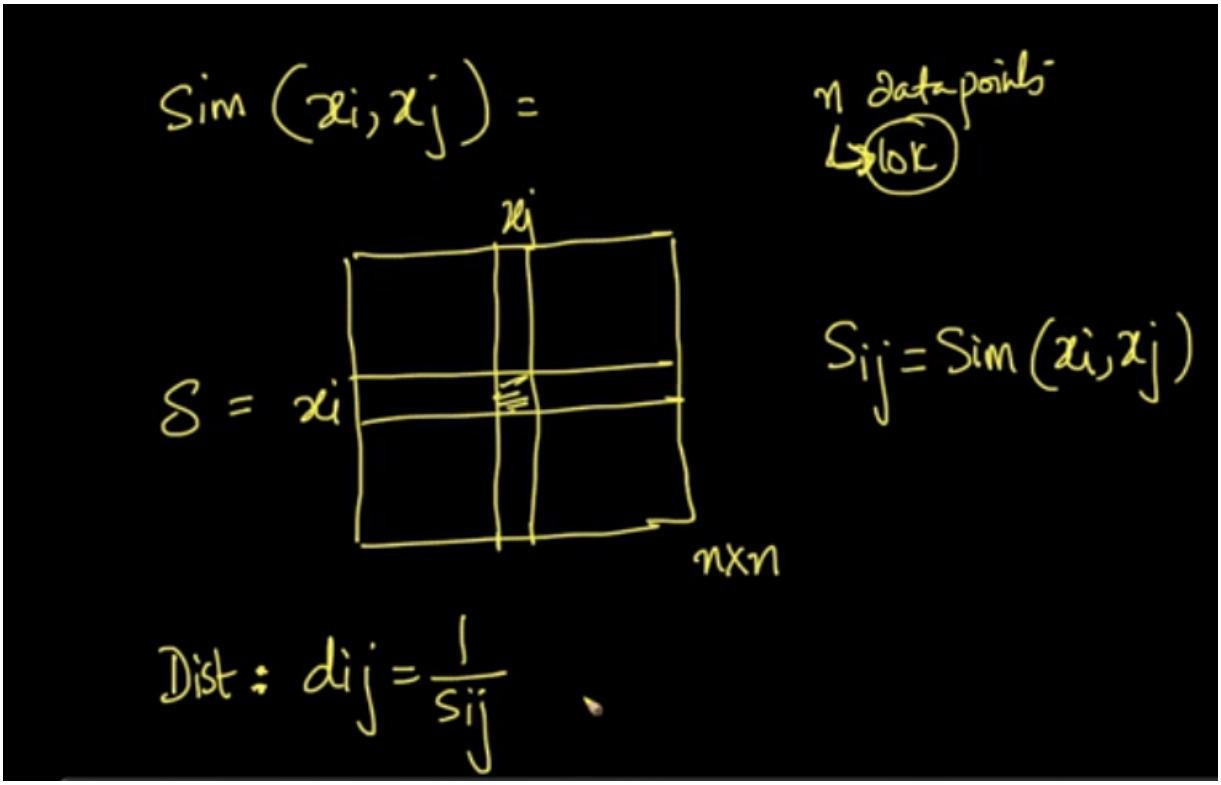
We must train ‘c’ classifiers in this case.
This solution is called one vs. rest solution.



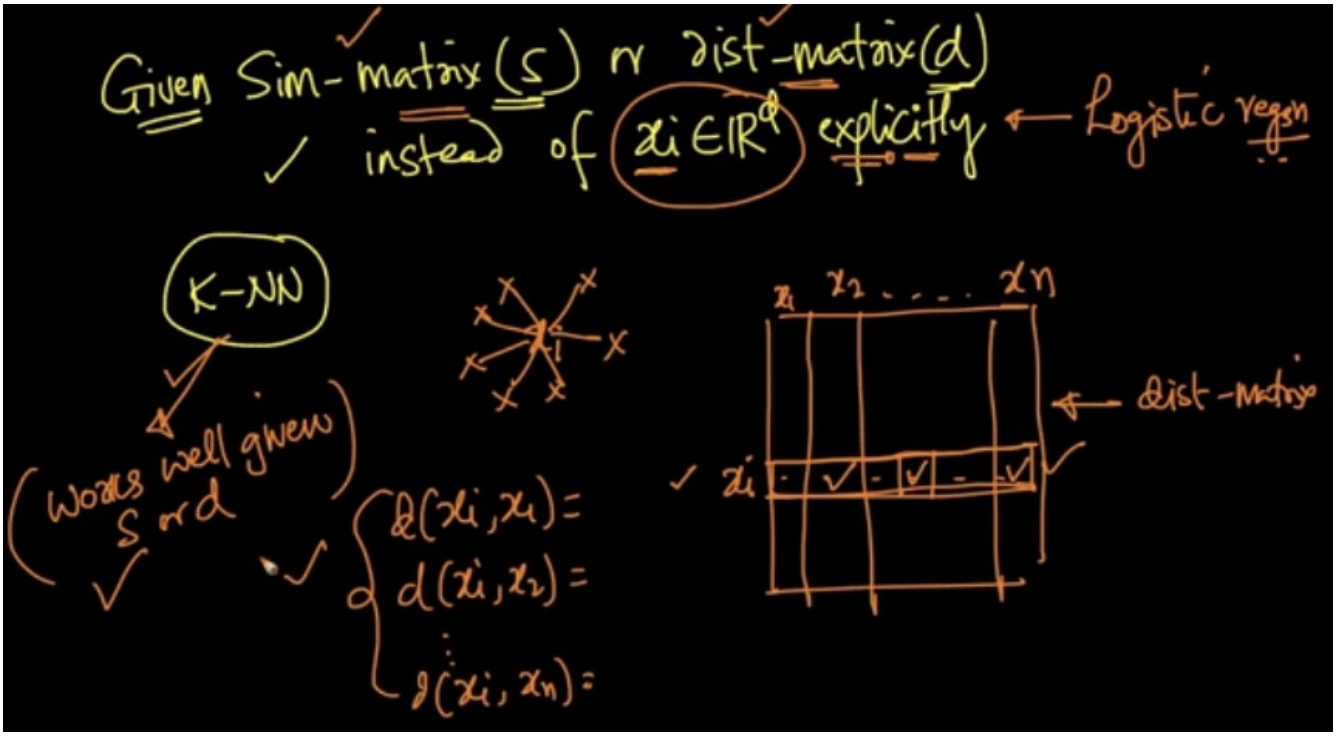
Distance similarity matrix:

Example: Chemical compounds cannot be easy to express in the form of numeric value.



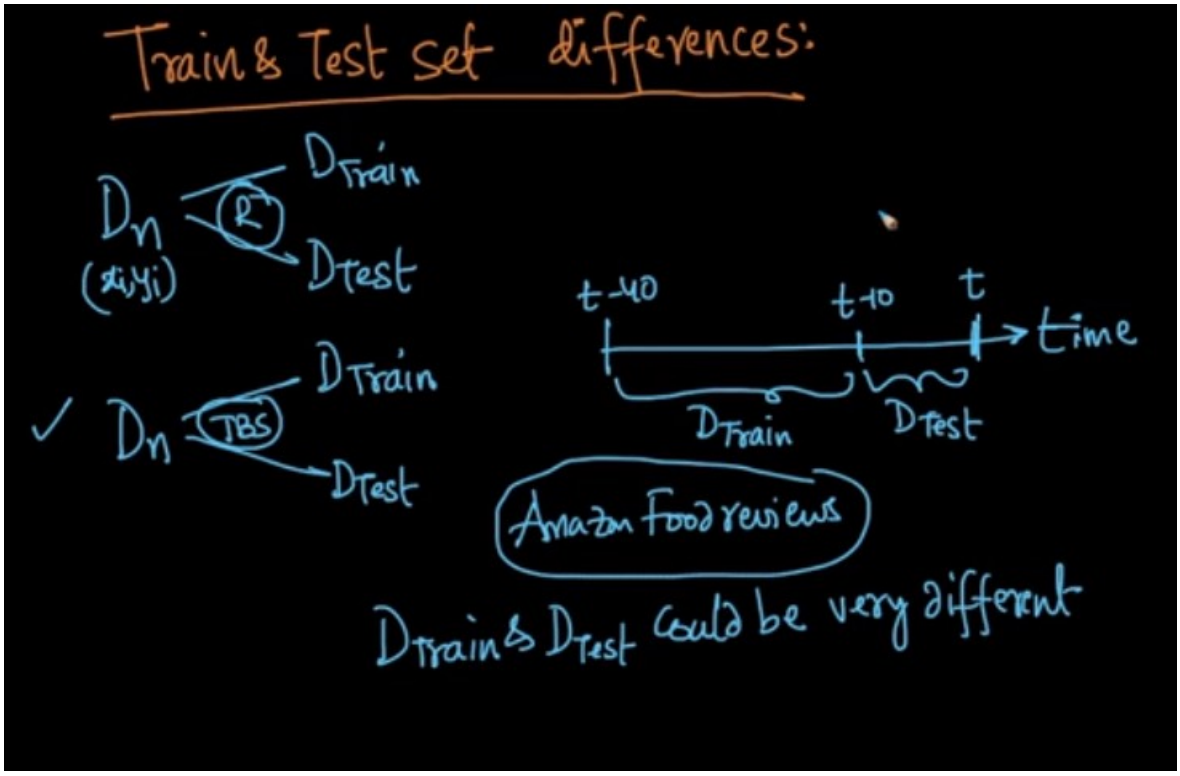


Distance matrix is calculated by taking inverse of the similarity value of each feature.

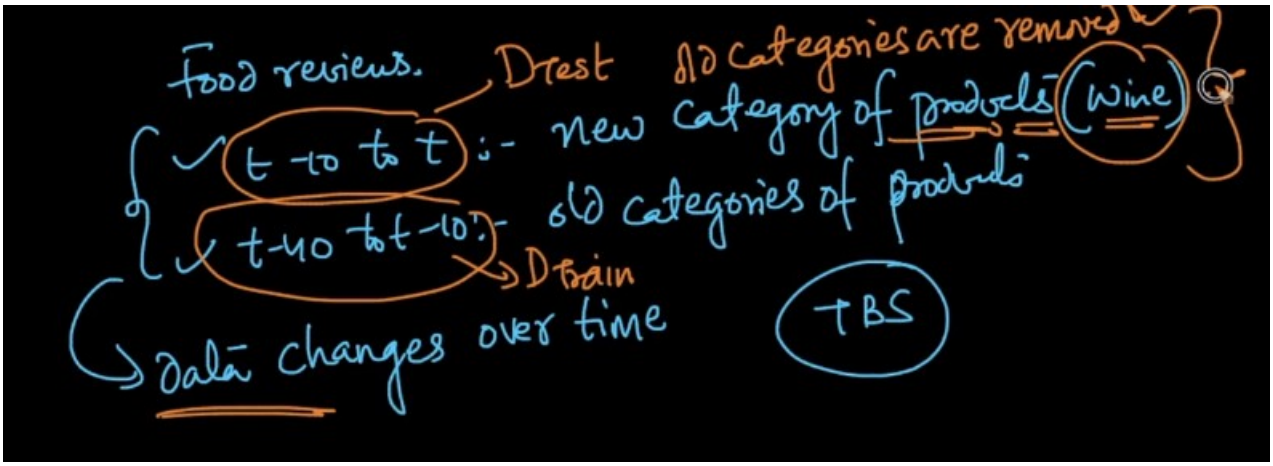


Train – test set differences:

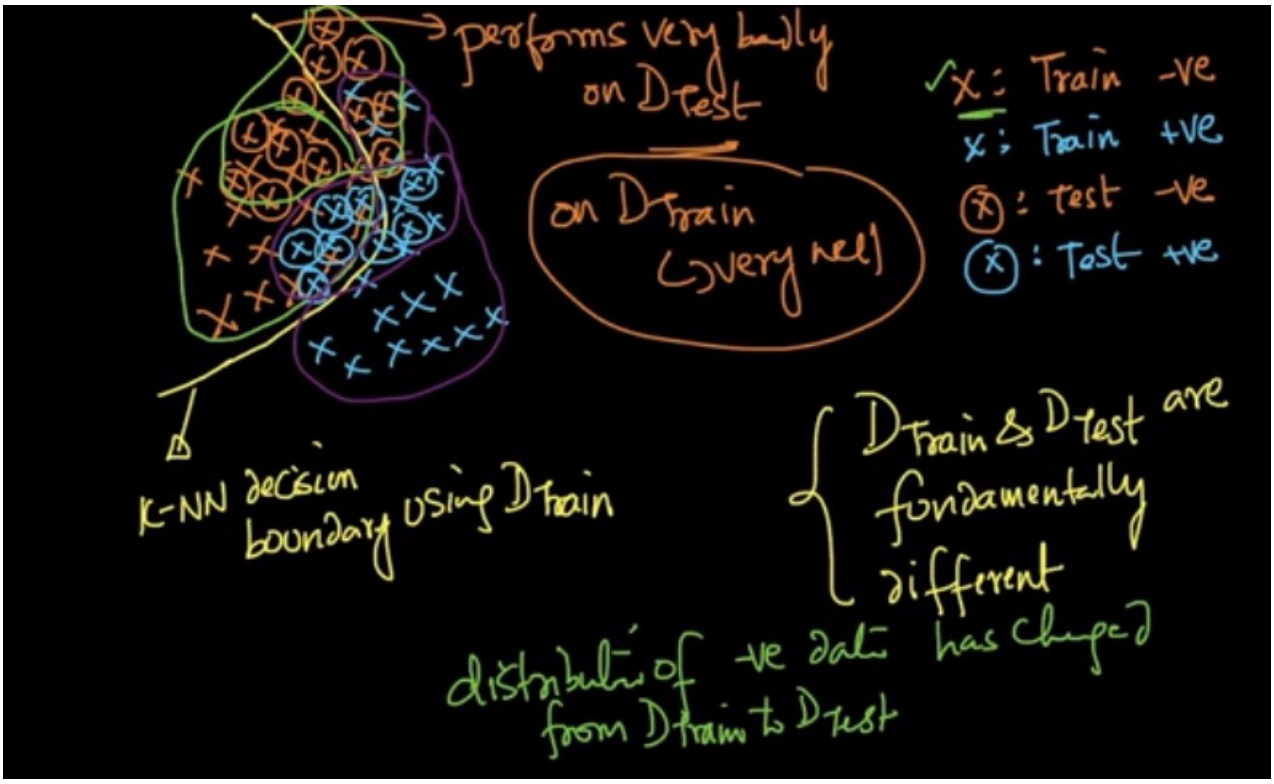
1. Random splitting.
2. Time – based splitting.



The underlying data changes over time then choosing time – based slicing is the good choice of splitting the data.

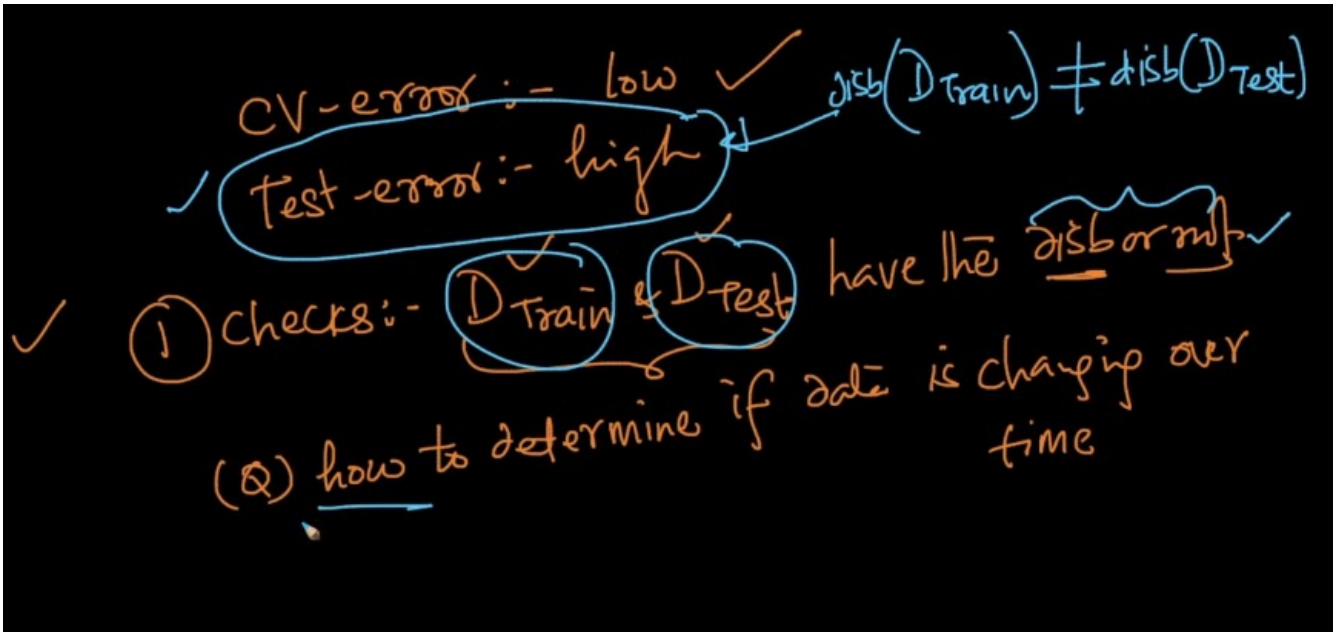


There will be miss match of the train and test data sets. The distribution of train and test data-sets changes.



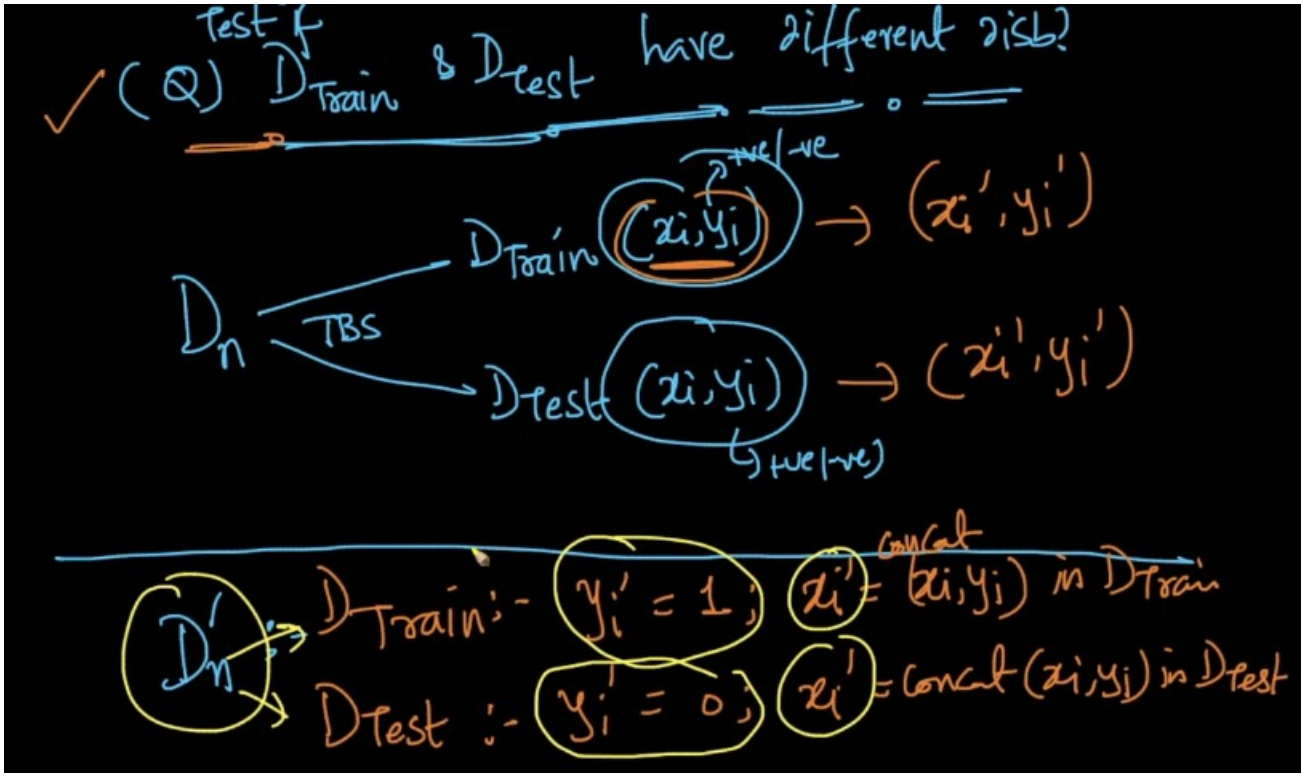
The cross – validation error will be low but the test accuracy will be low.

The distribution of the train and test must be checked.

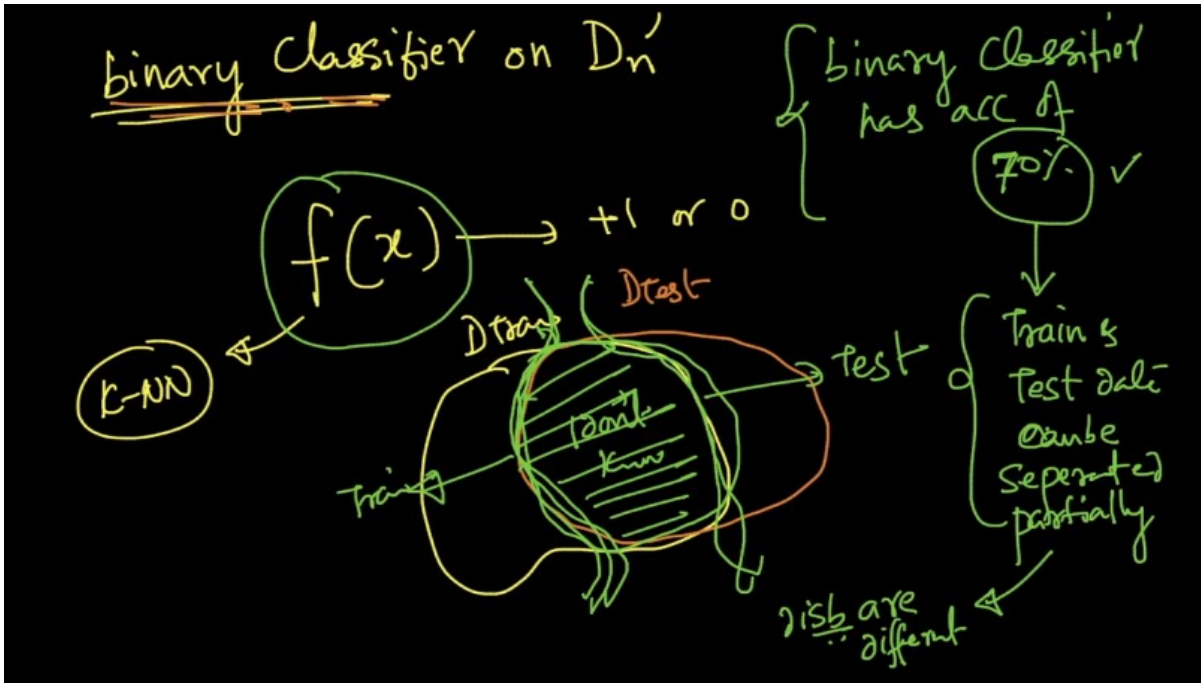


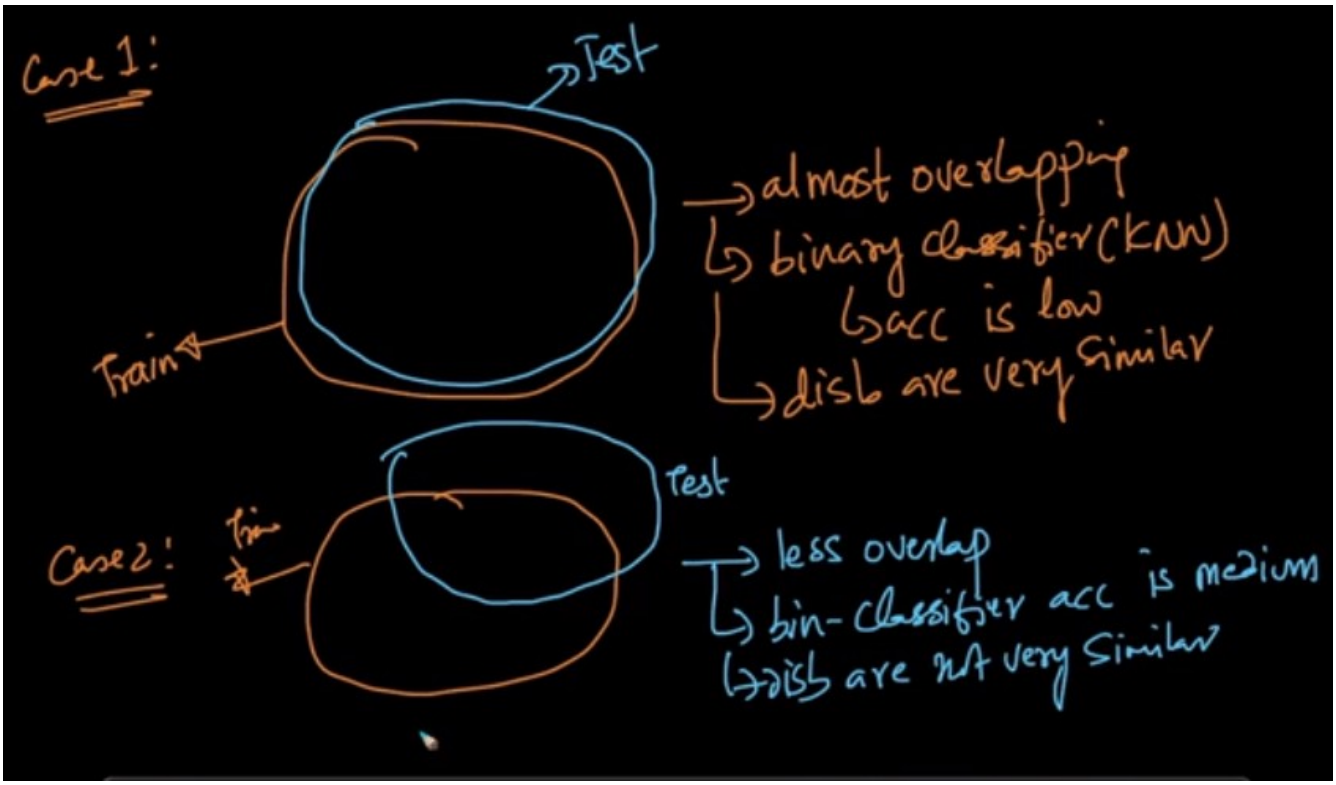
Q) Determining the train and test distributions.

We can use a classifier to determine the distribution of train and test data of the new data-set and the old data-set.



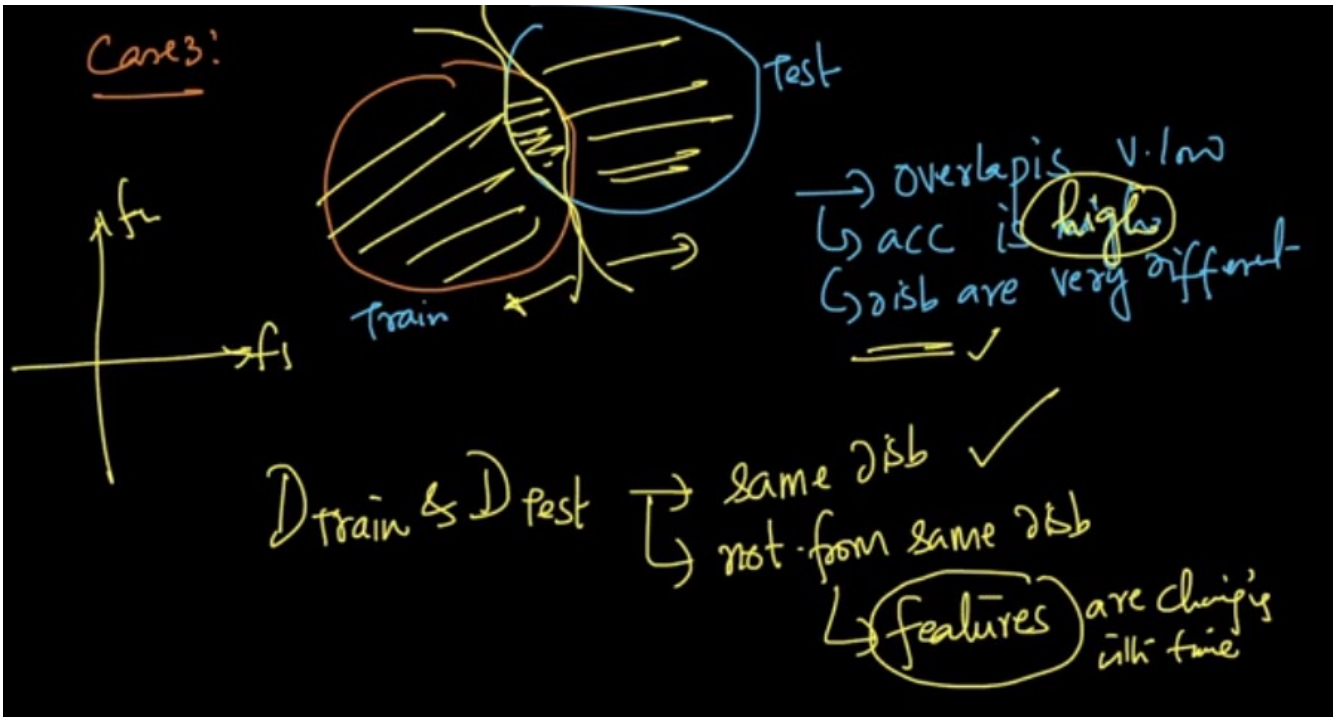
Build the binary classifier on the new data set.
The accuracy of the model gives the differences of the train and test data-sets.



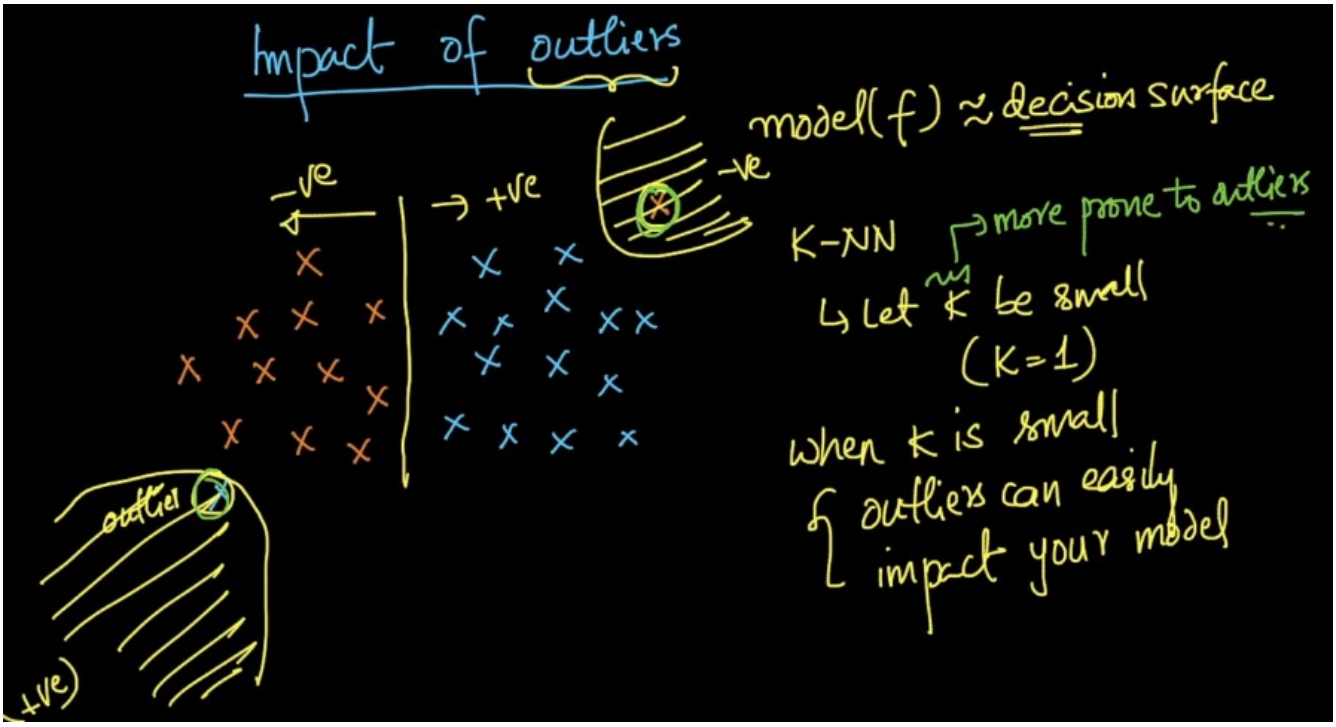


We can use any binary classifier.
This accuracy will be low as the distributions are very different.

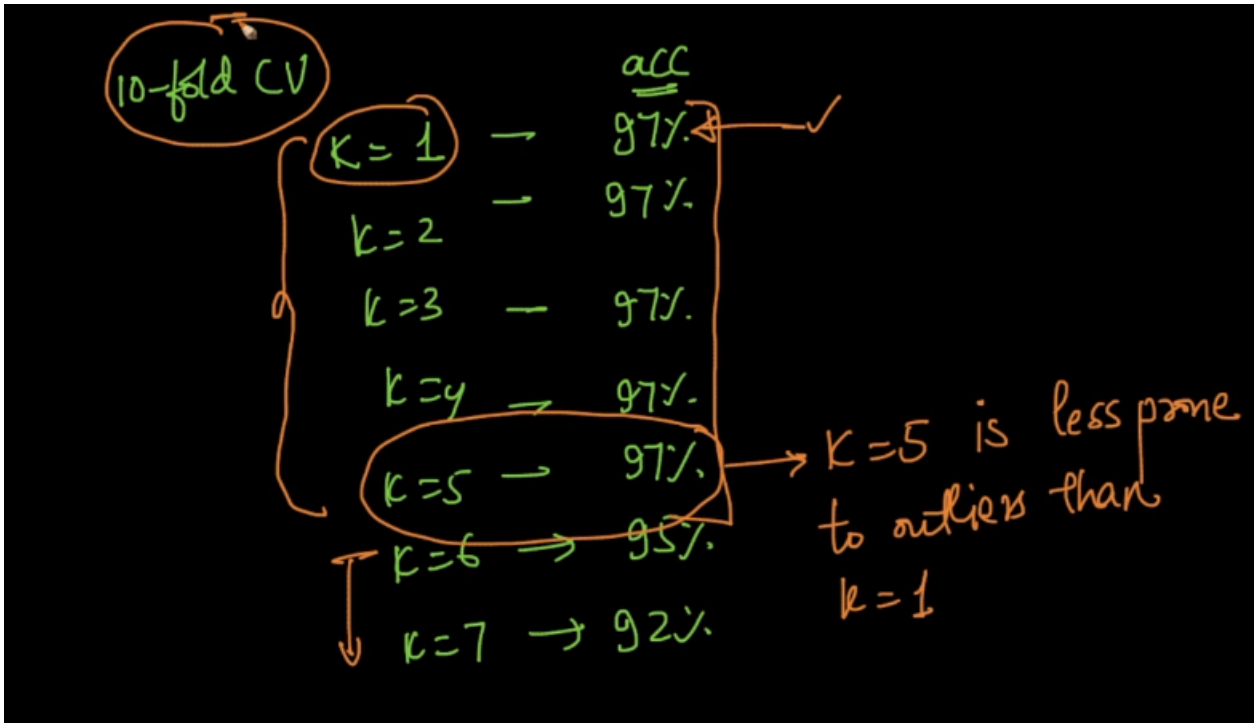
If the data is not coming from the same distribution then feature space is changing, building the new feature will be the way.



Impact of outliers:
In case of KNN, when K is small outliers can easily impact the model.



It is always to pick the greater value of L, Since the model is less prone to outliers.

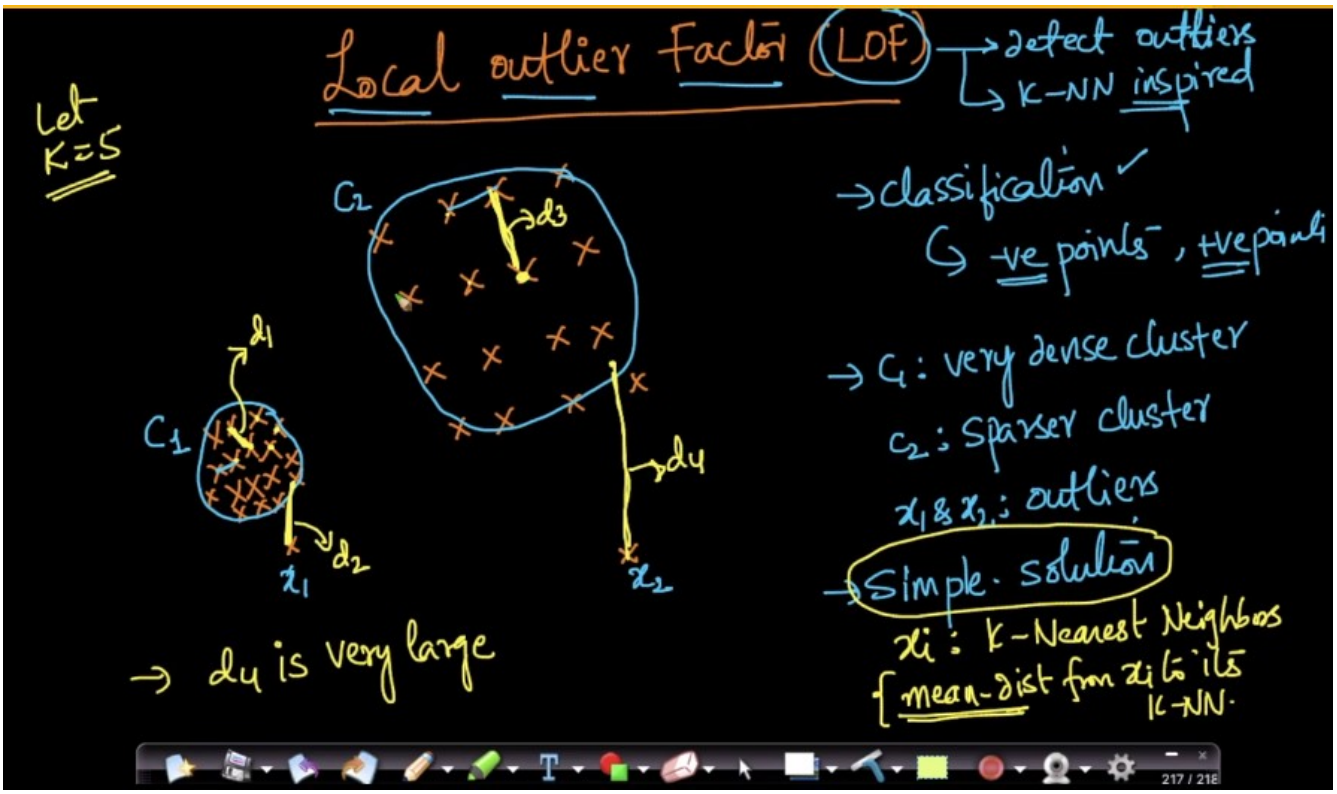


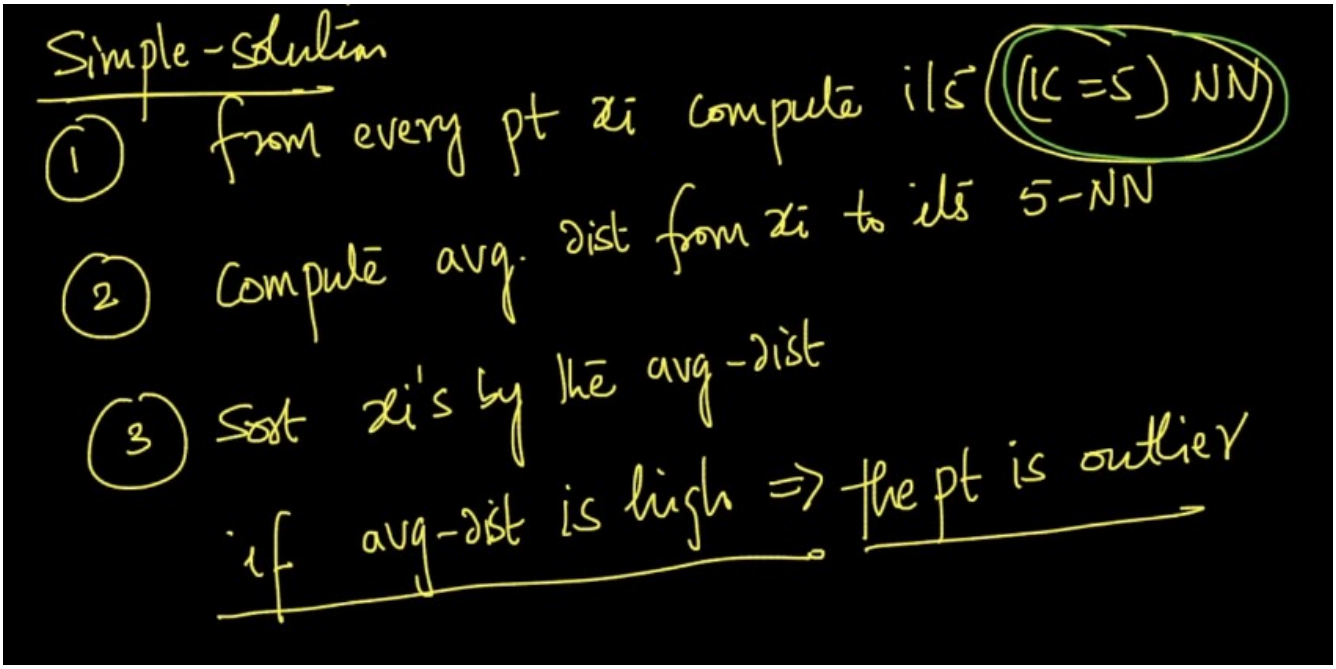
Local outlier – factor is used to detect outliers.

Local – outlier factor:

Mean distance to KNN:
It is being inspired by KNN.

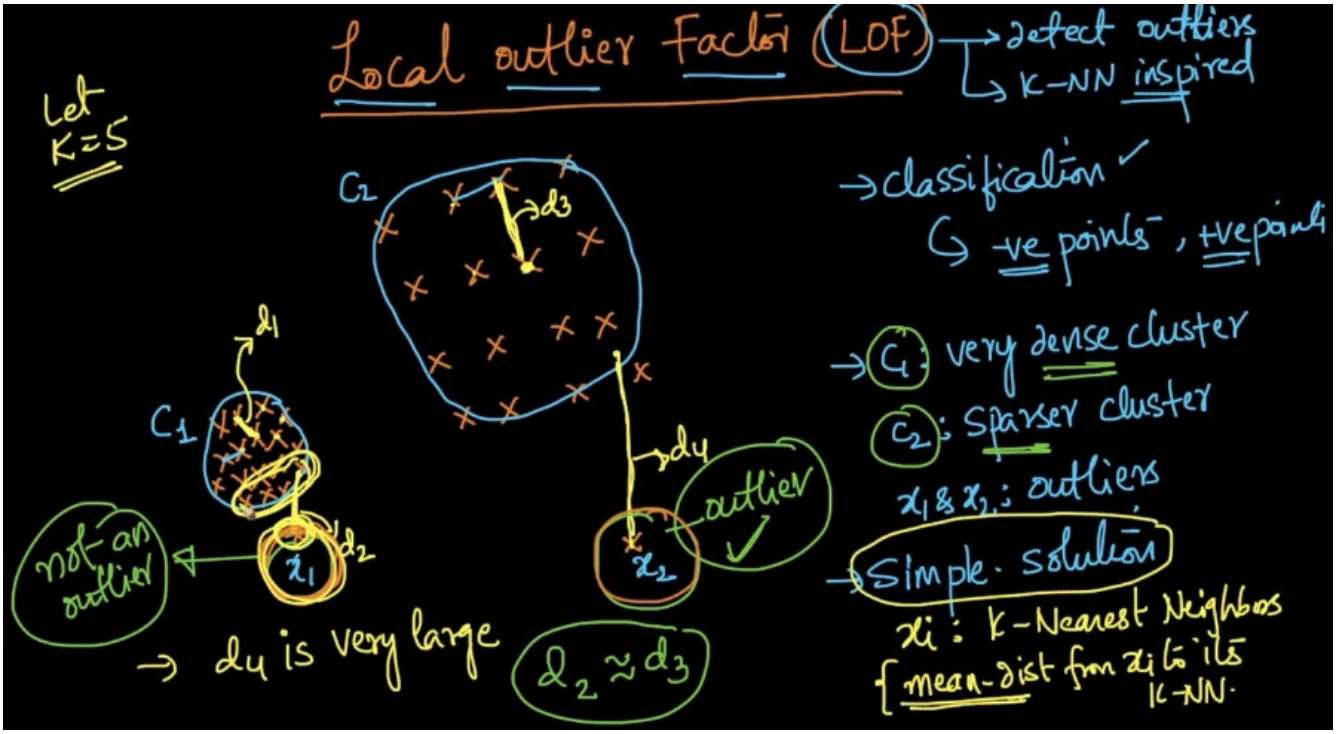
Detect outliers:
Consider the two clusters of points C1 and C2, dense and sparse respectively. Both X1 and X2 are outliers.





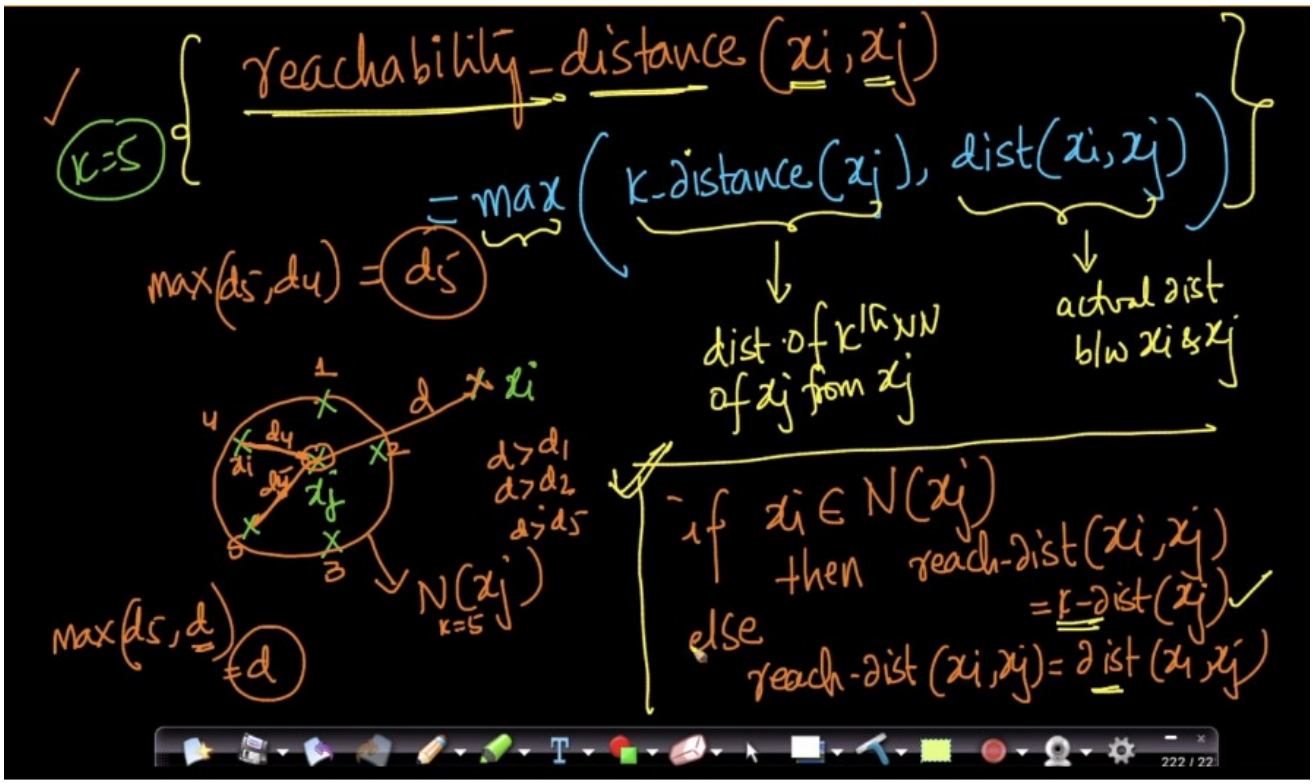
By this process we cannot conclude X2 as an outlier. Though it is an outlier to the dense cluster.

There by this situation leads to computing local density.

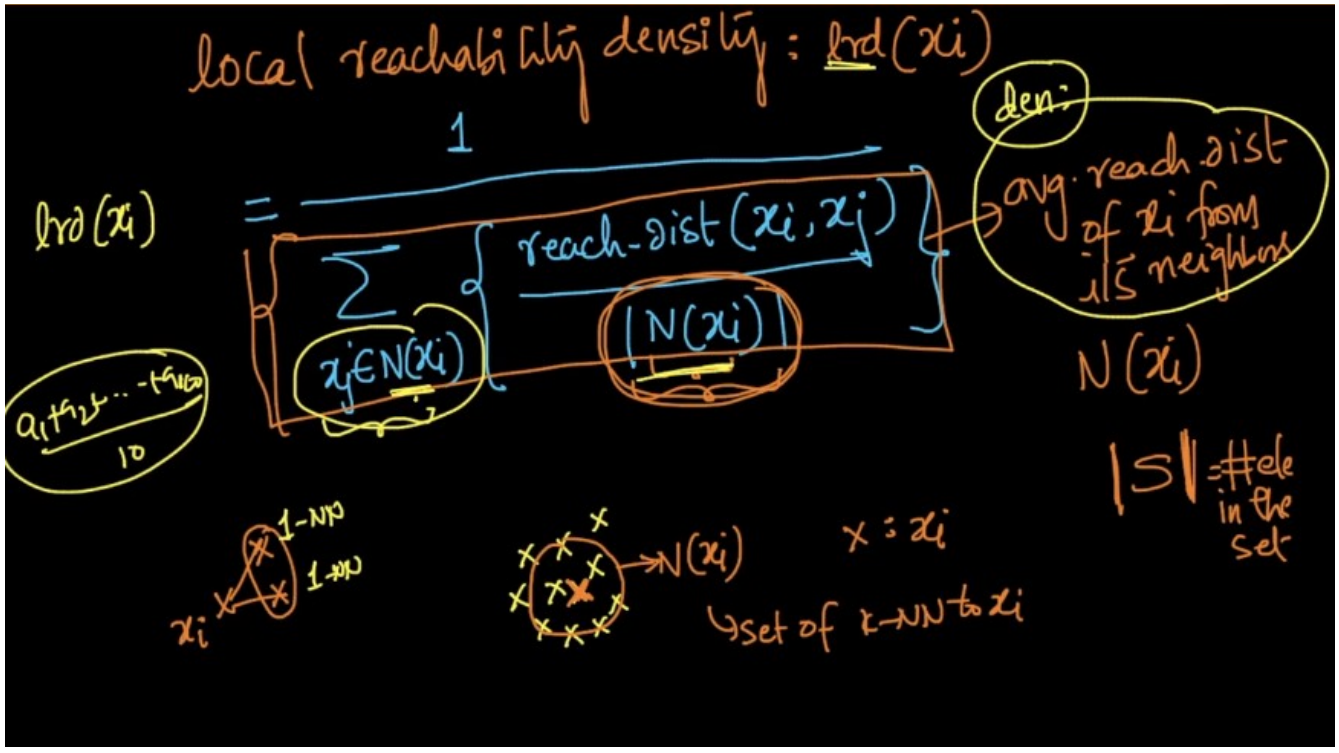


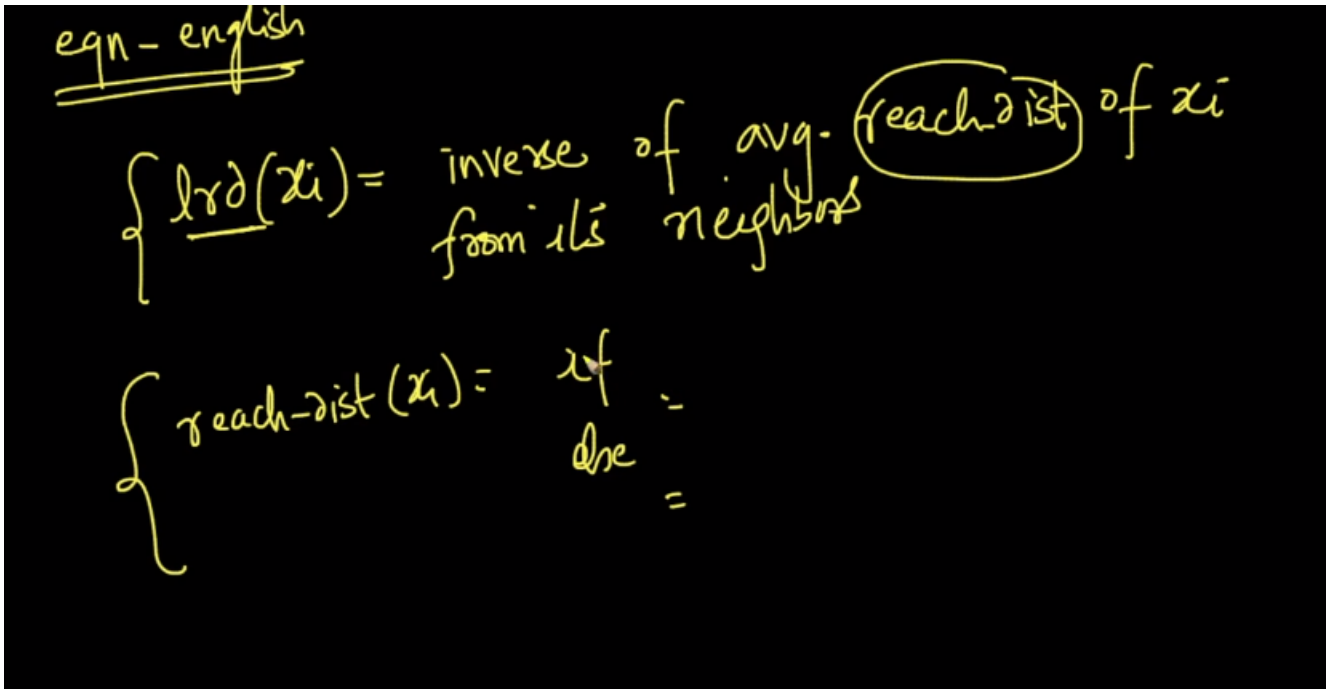
Reachability – Distance(A, B):

The reachability distance of two points x1 and xj is given as follows:

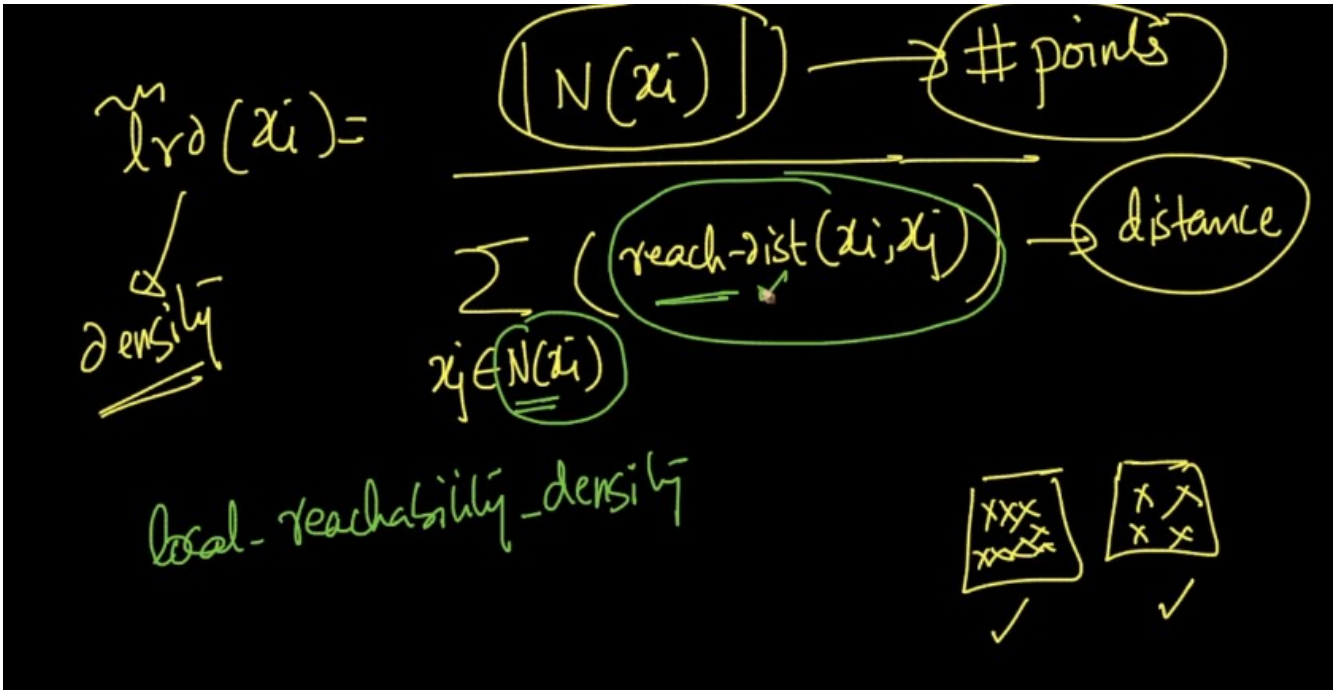


Local reach-ability – density(A):

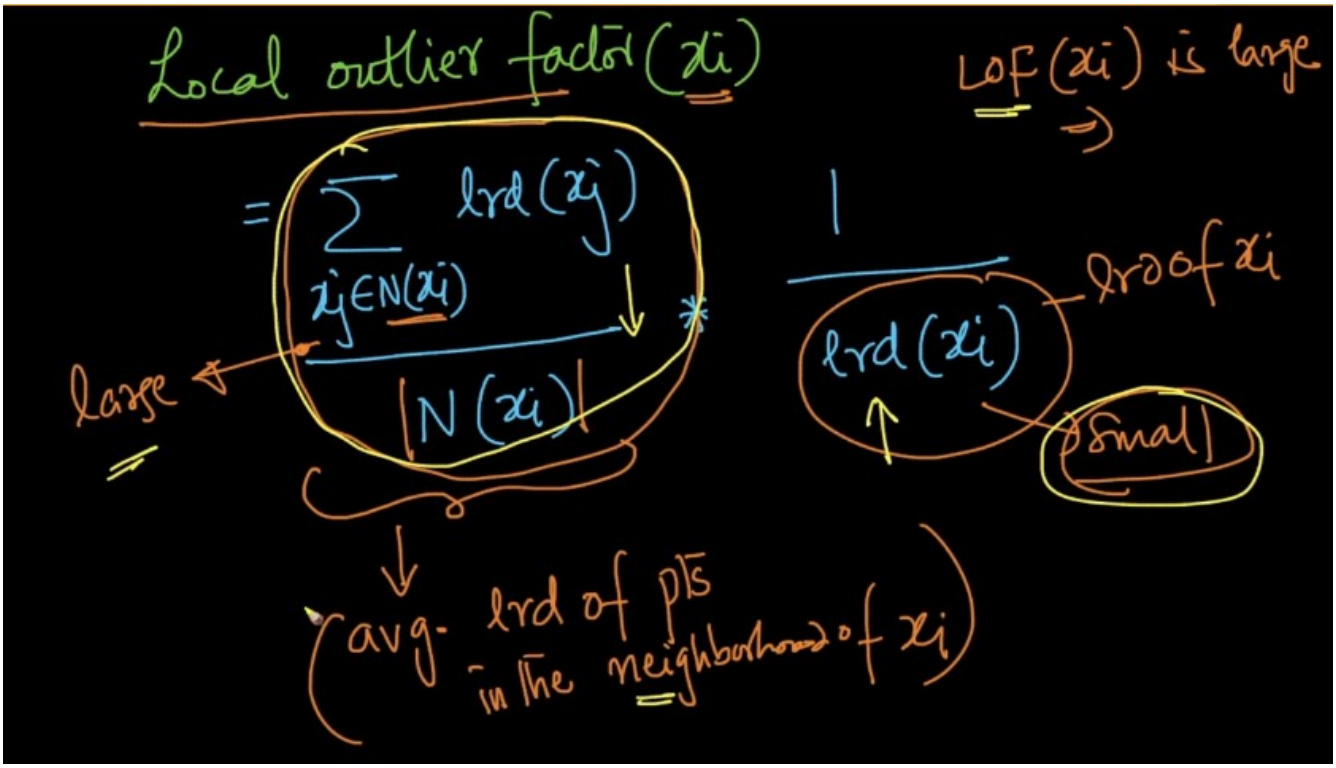


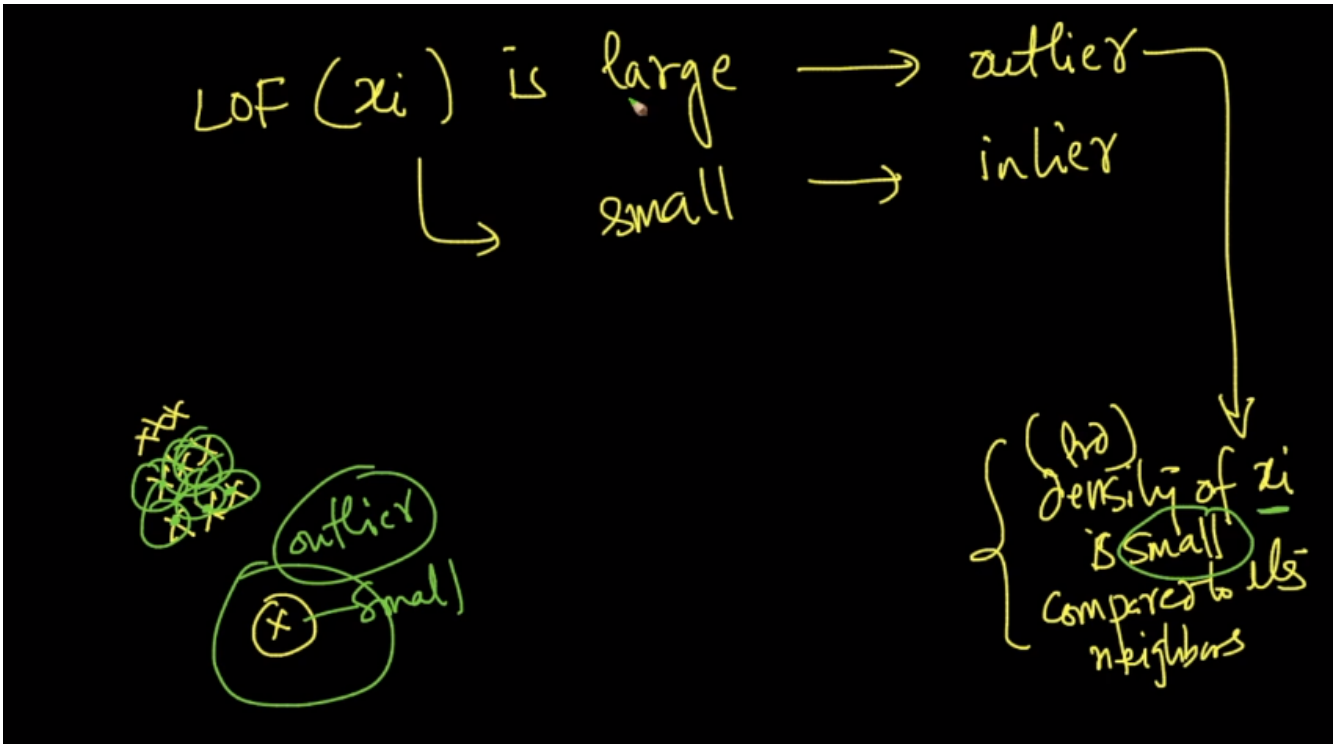


Reach-ability distance lead to local reach-ability – density.



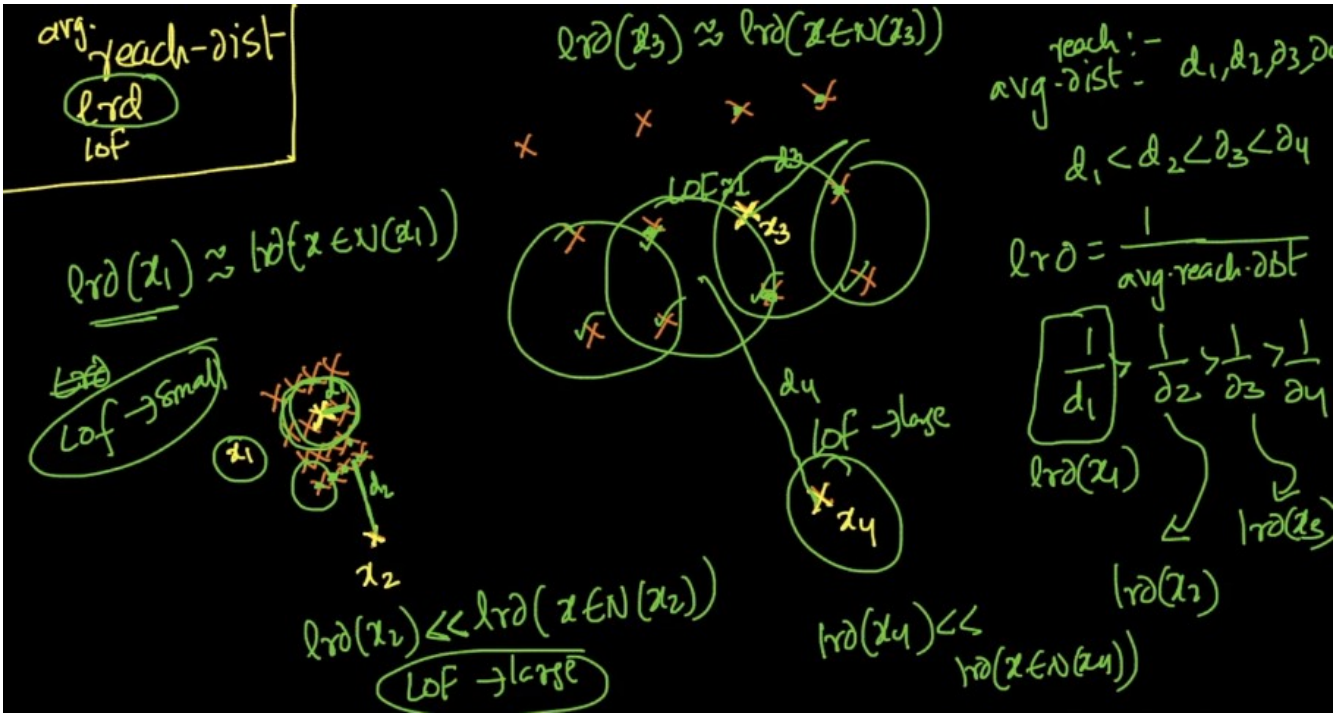
Local outlier Factor:

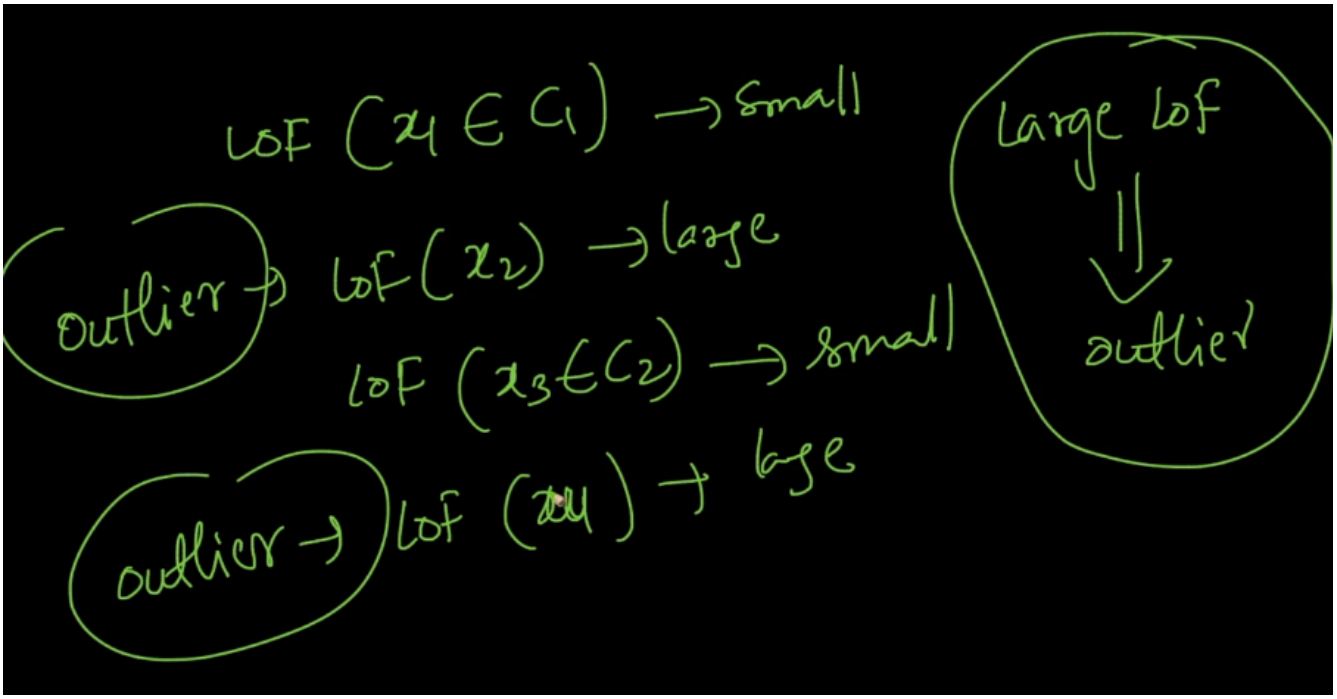




Example:

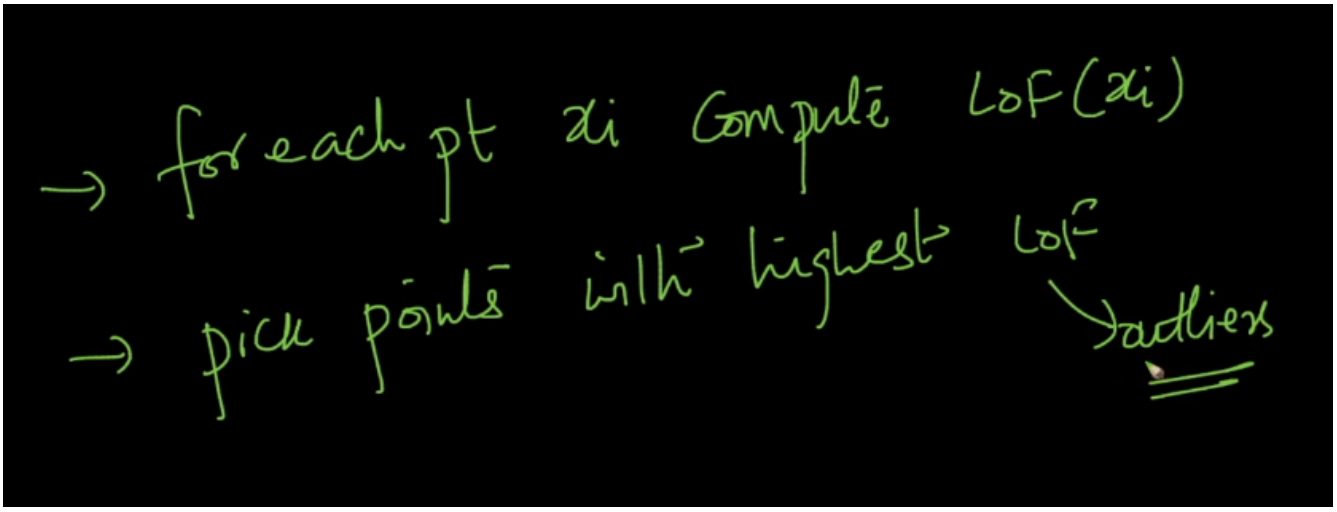
Explanation of local outlier factor with example.



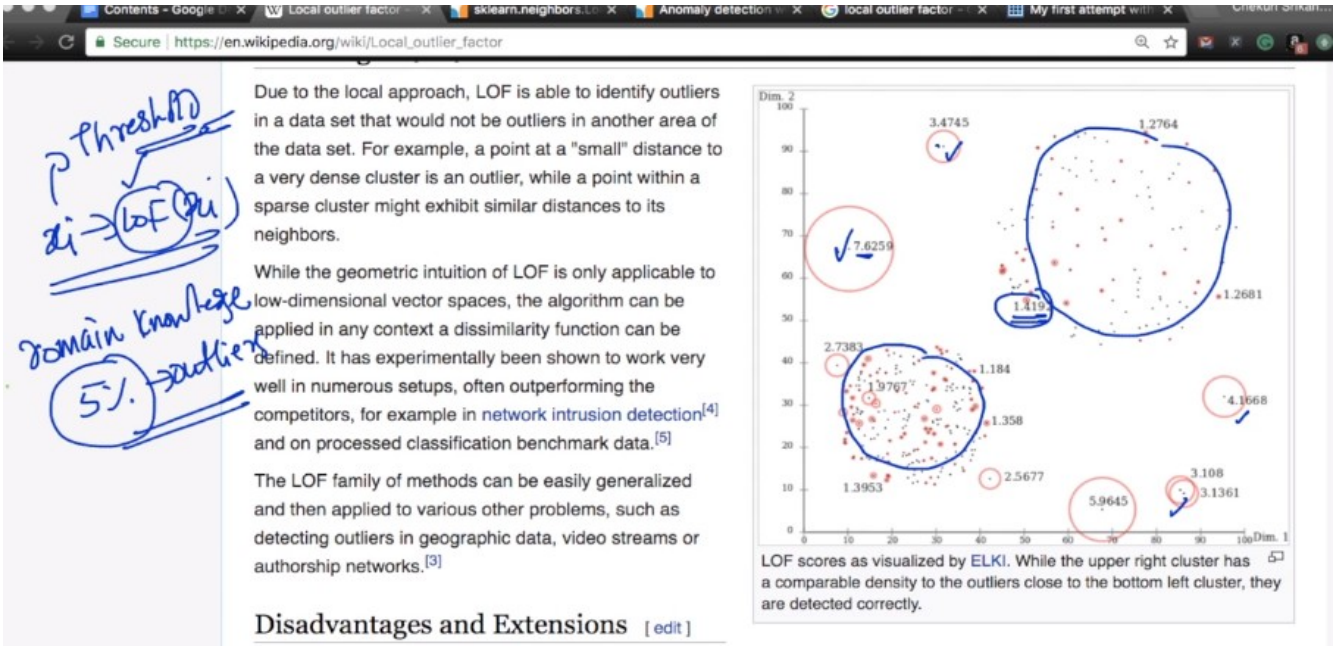


The larger the LOF is the outlier.

The threshold of LOF values depends on the application.

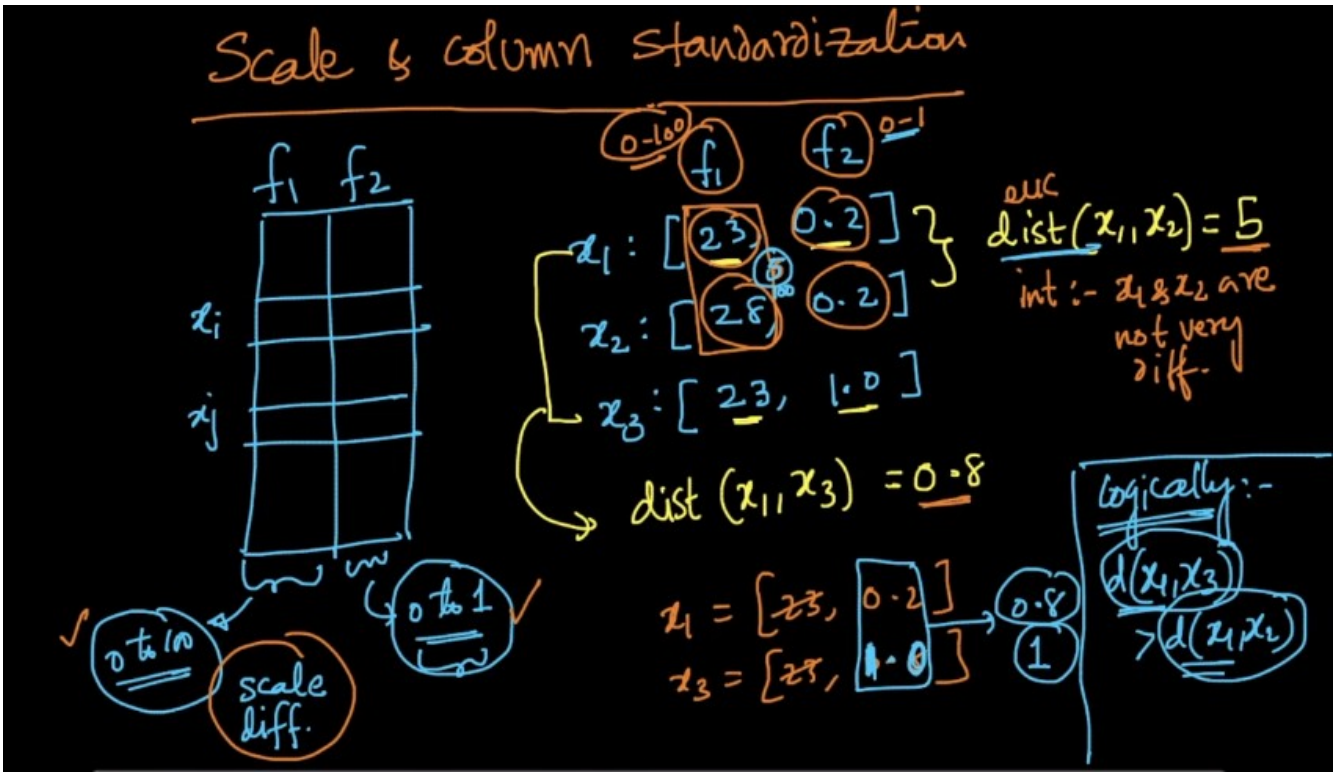


From the domain knowledge if we know the percentage of the points that will be outliers, then we can pick right LOF distance value and remove the points that are greater than the distance chosen.

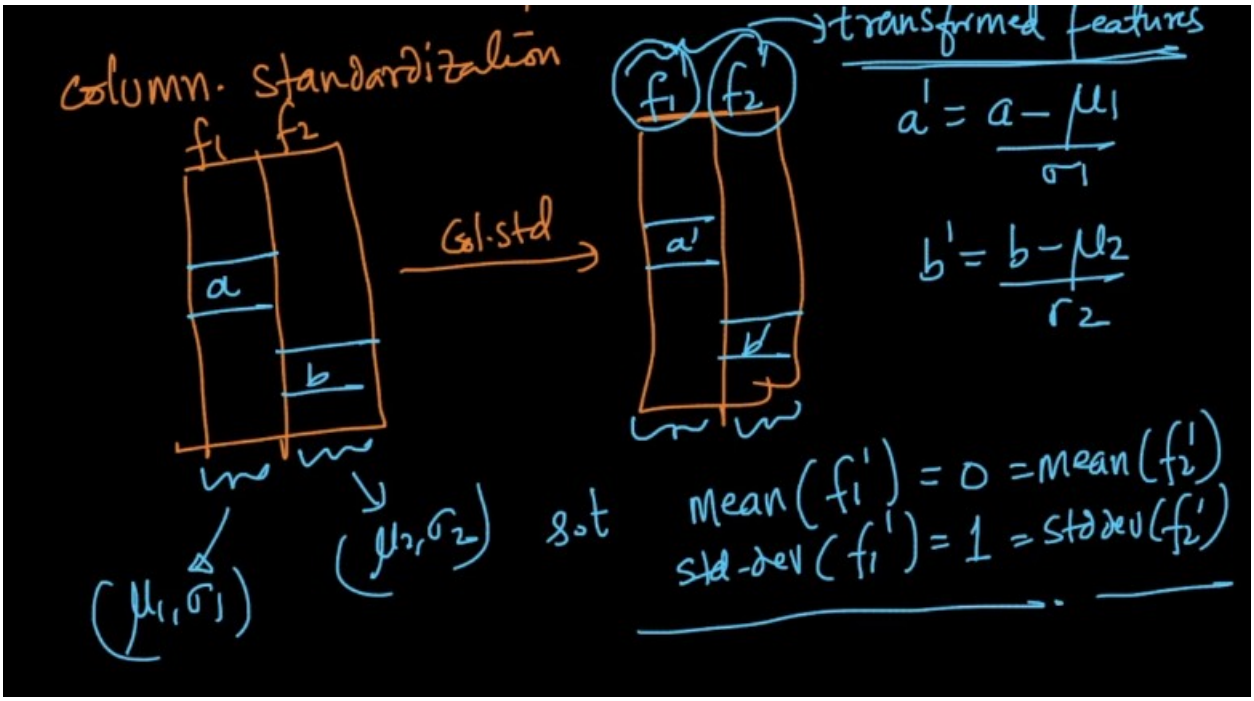


Impact of scale and column standardization:

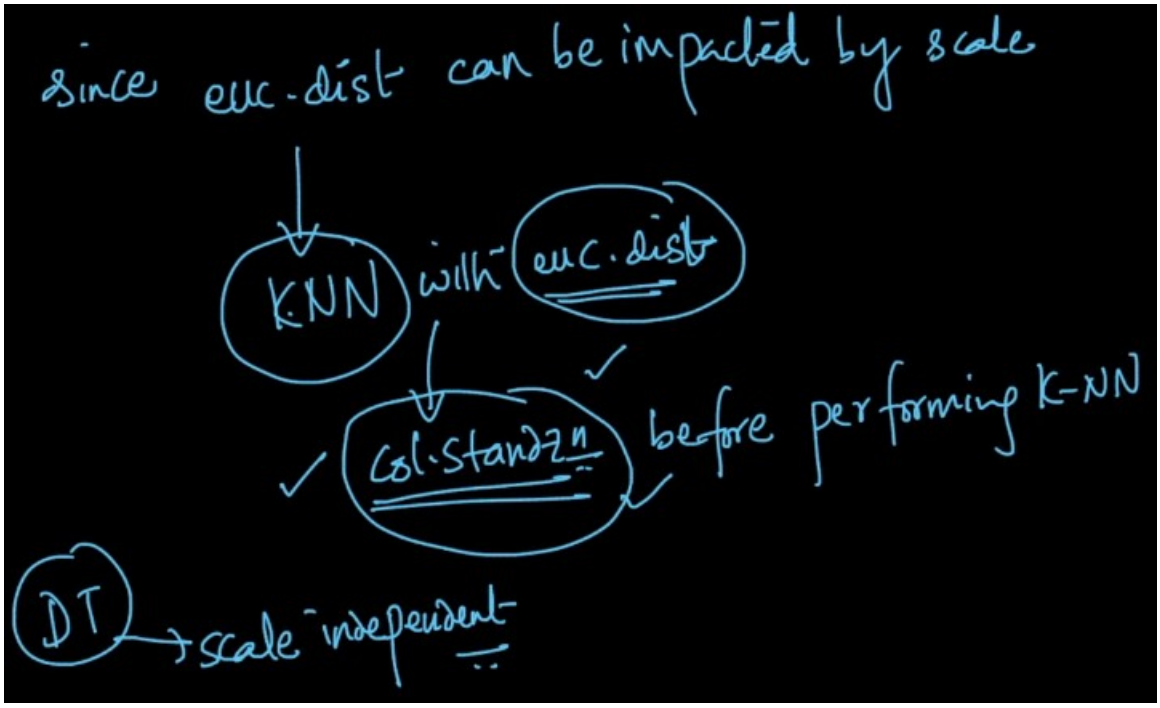
The values of the attributes will be different in scale compared one another. The distance between the points will be different when the scale of the attributes are different.



Column standardization solves the scale difference of the attributes.

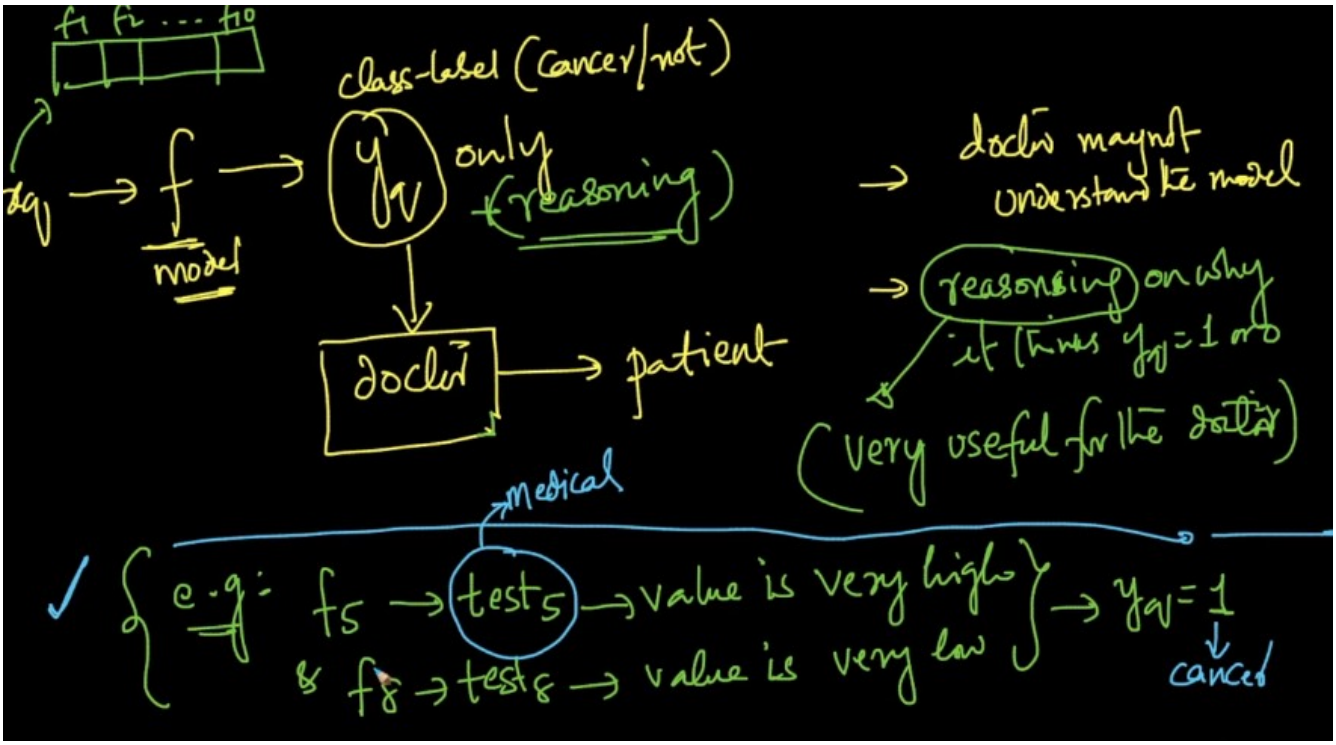


The distance measures will be impacted scale. So, we need column standardization.



Interpret-ability:

If we provide reasoning of the prediction of the model, we can better understand the results.



Models that cannot give the reasoning is called black – box model.

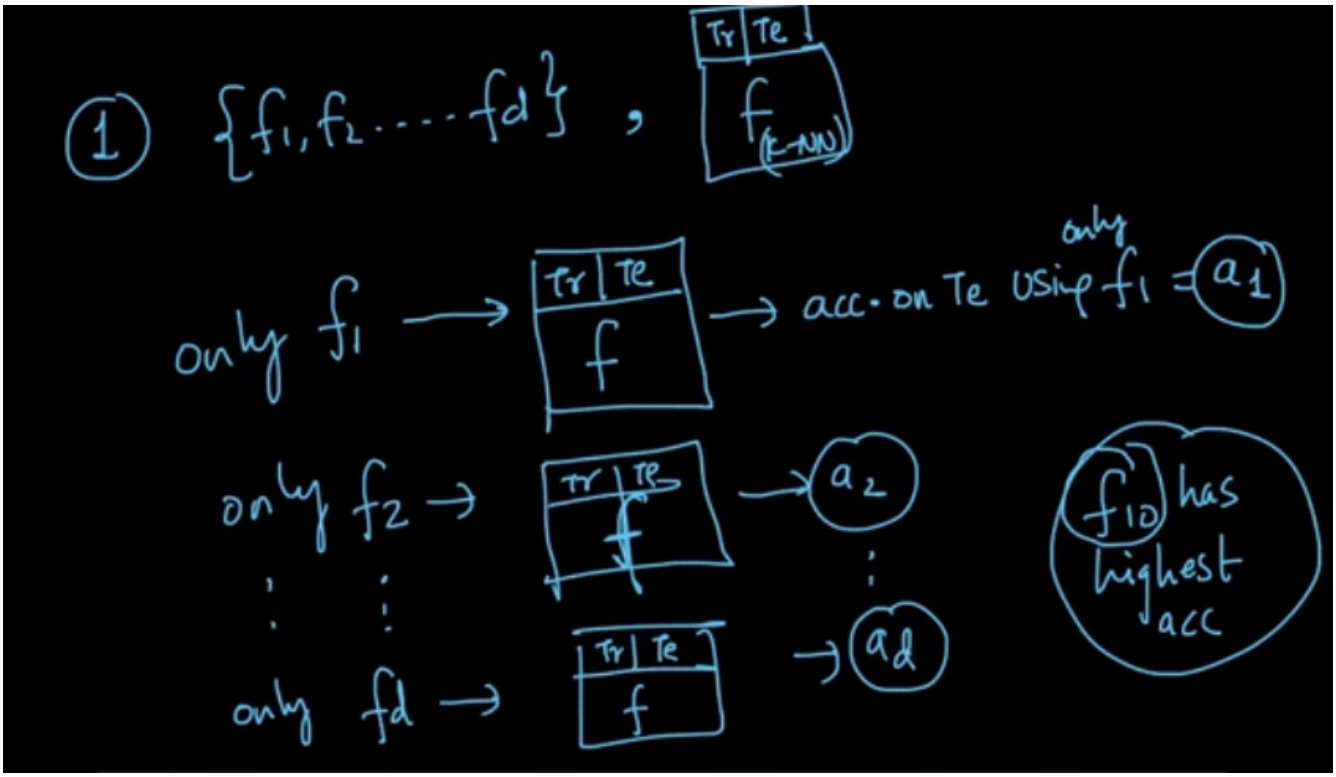
The interpret-ability plays important role in reasoning the results of the model.
KNN is interpret able when d is small and k is small.

Feature importance and forward feature importance:

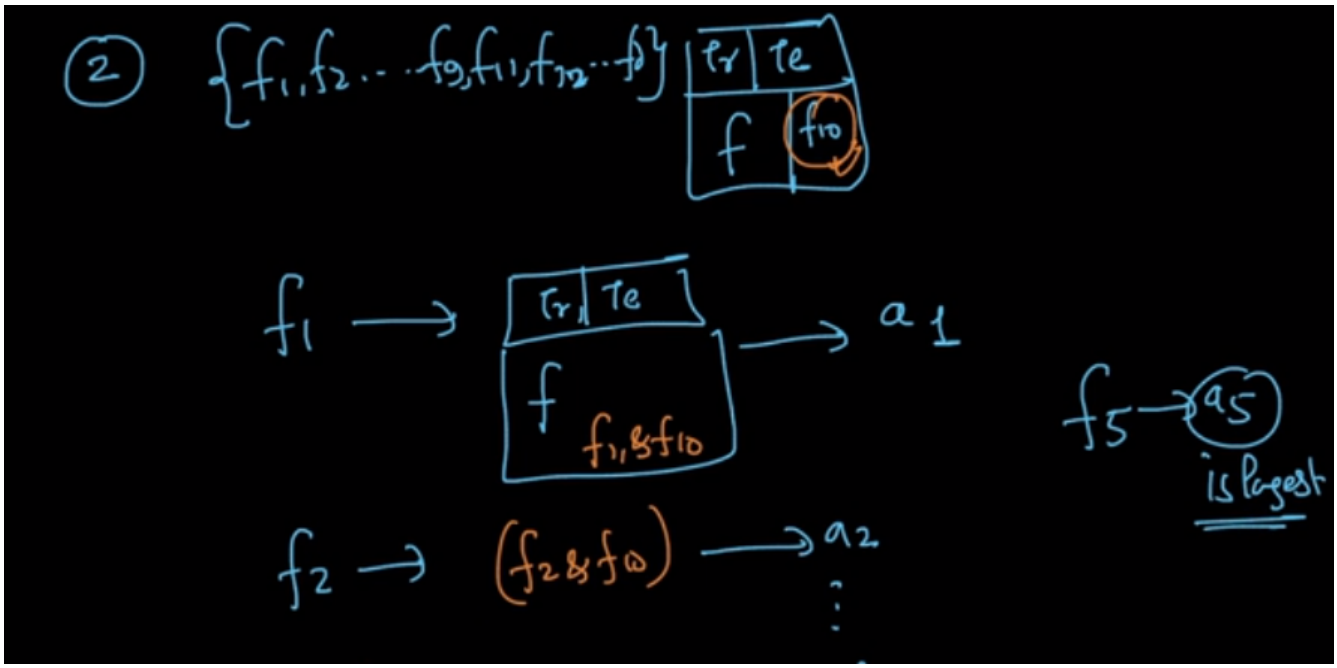
Forward feature selection:

Picking the top features in the data set is called forward feature selection.

For every feature we will make train, test and develop model to get the accuracy scores and the model with highest accuracy is chosen as the most favoring feature towards the accuracy.



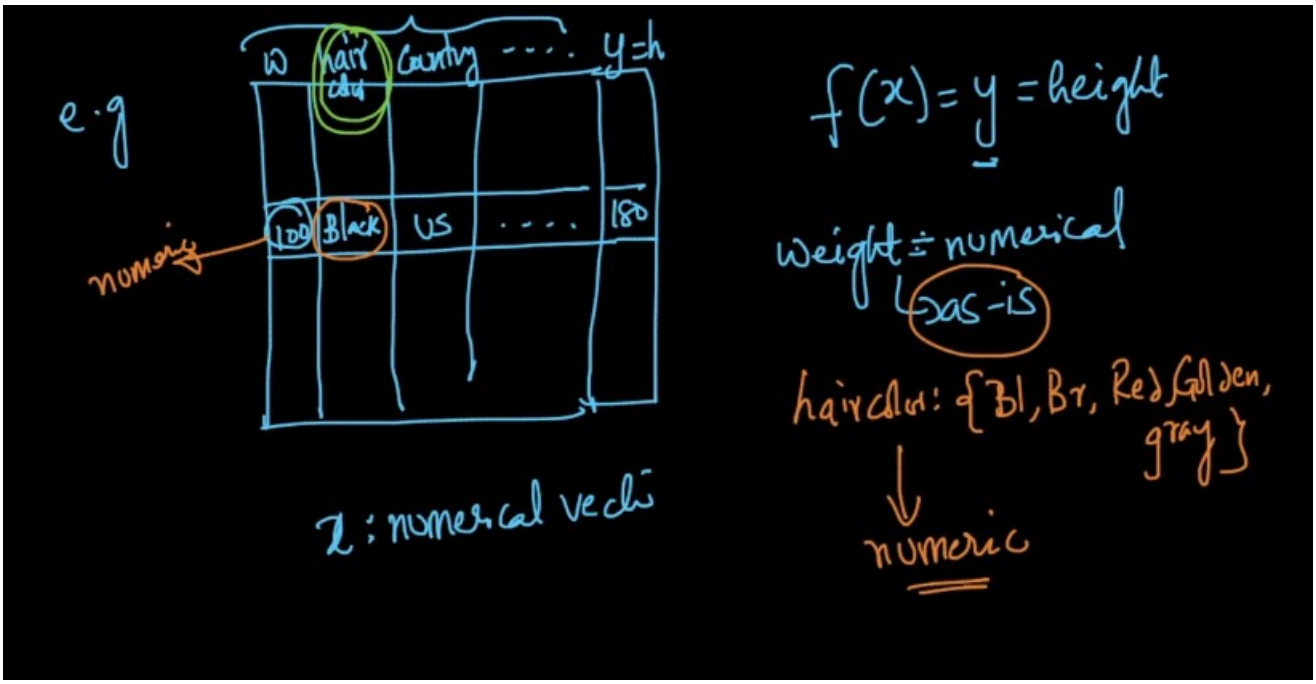
The the chosen features are again compared with other features and accuracy is calculated. This is called forward feature selection.



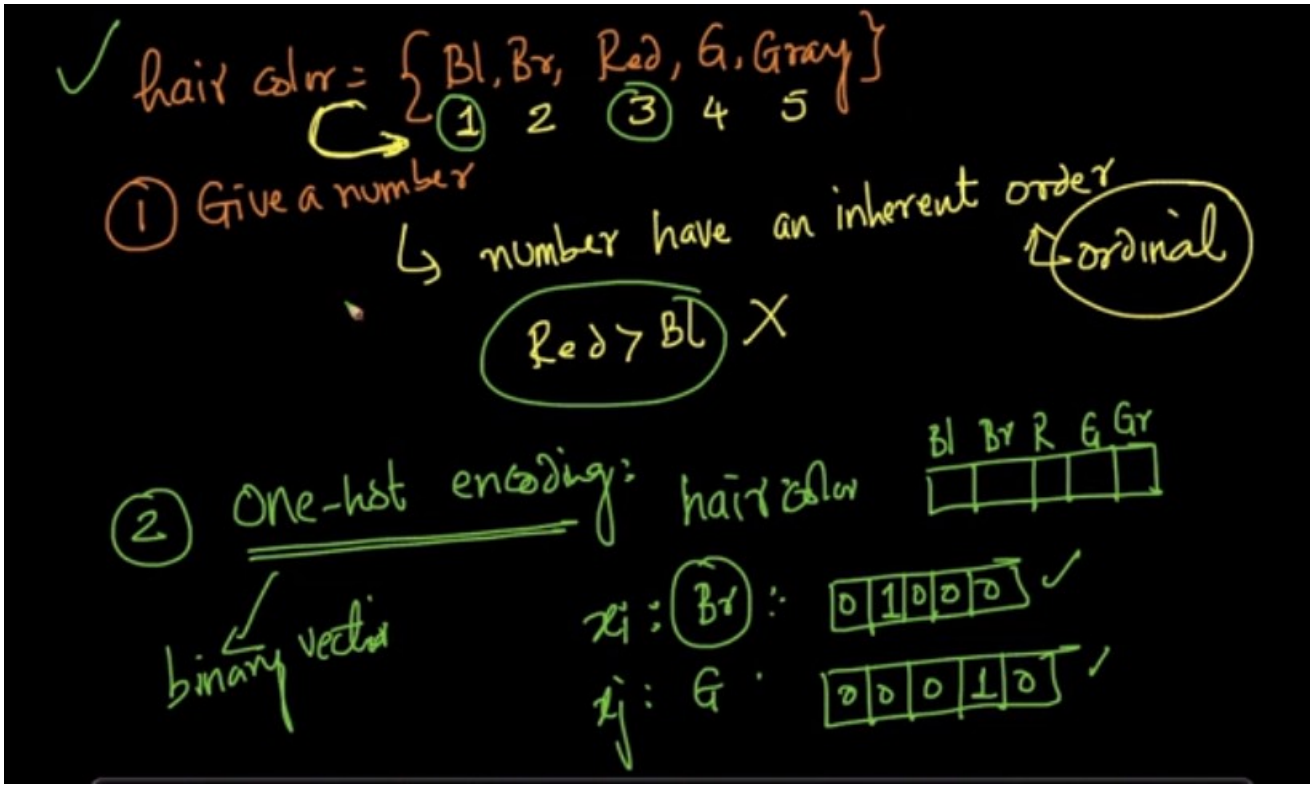
At every stage we will choose the feature that add most value to the model.

Backward feature selection:
In each iteration we will remove the feature which results in lowest drop in accuracy.
Handling categorical and numerical features:

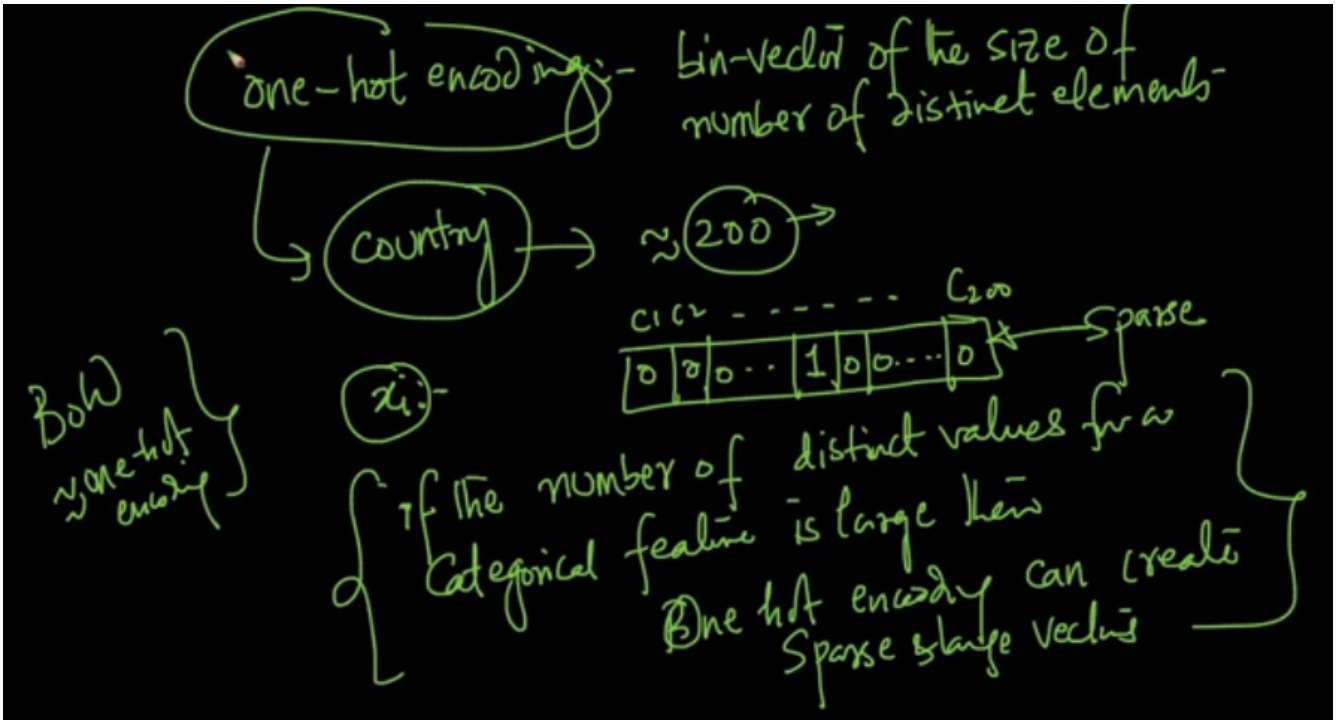
Converting categorical features to numerical – Converting the categories to numerical will give an artificial constraint that changes the meaning of the categories.
To overcome this situation we will use one – hot encoding.



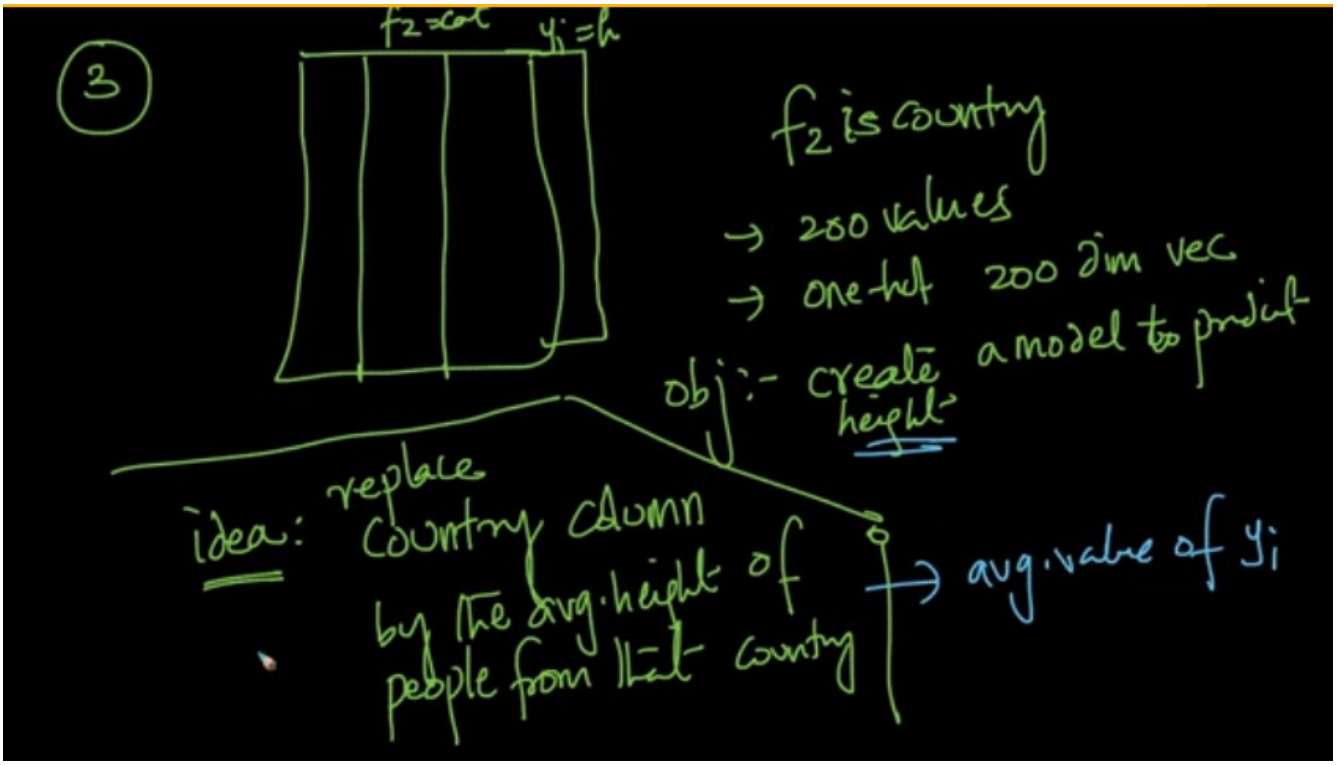
One – hot encoding makes the binary vector of each category of the size number of categories in the feature.



The problem with one – hot encoding, it increases the dimensions of the data.

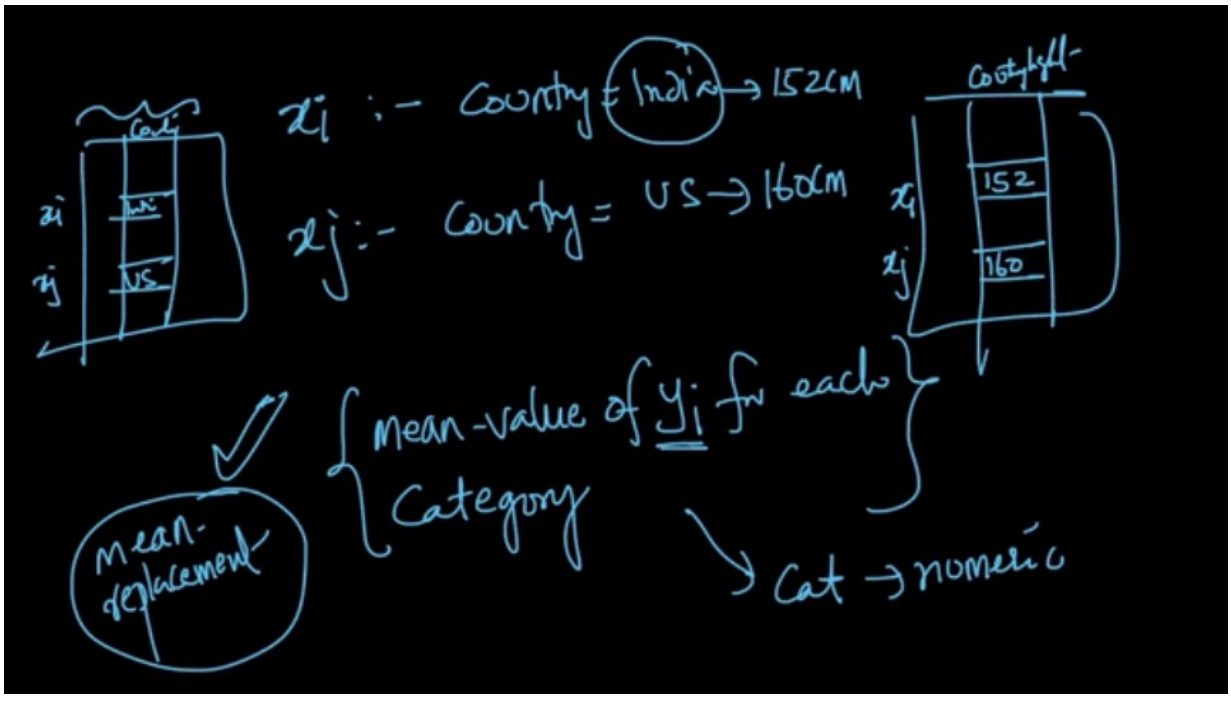


Other types of encoding:



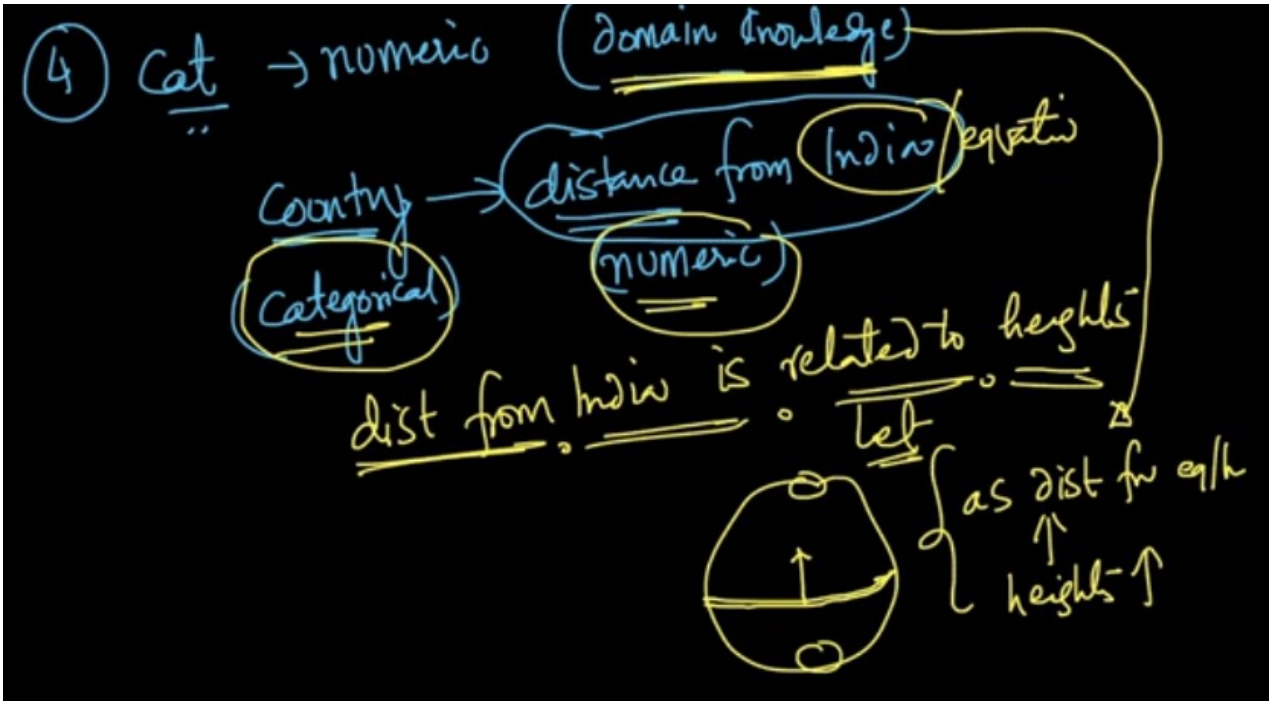
Example:

We can replace the each category with the mean value of the predicting variable. This is called mean replacement of the categorical data.

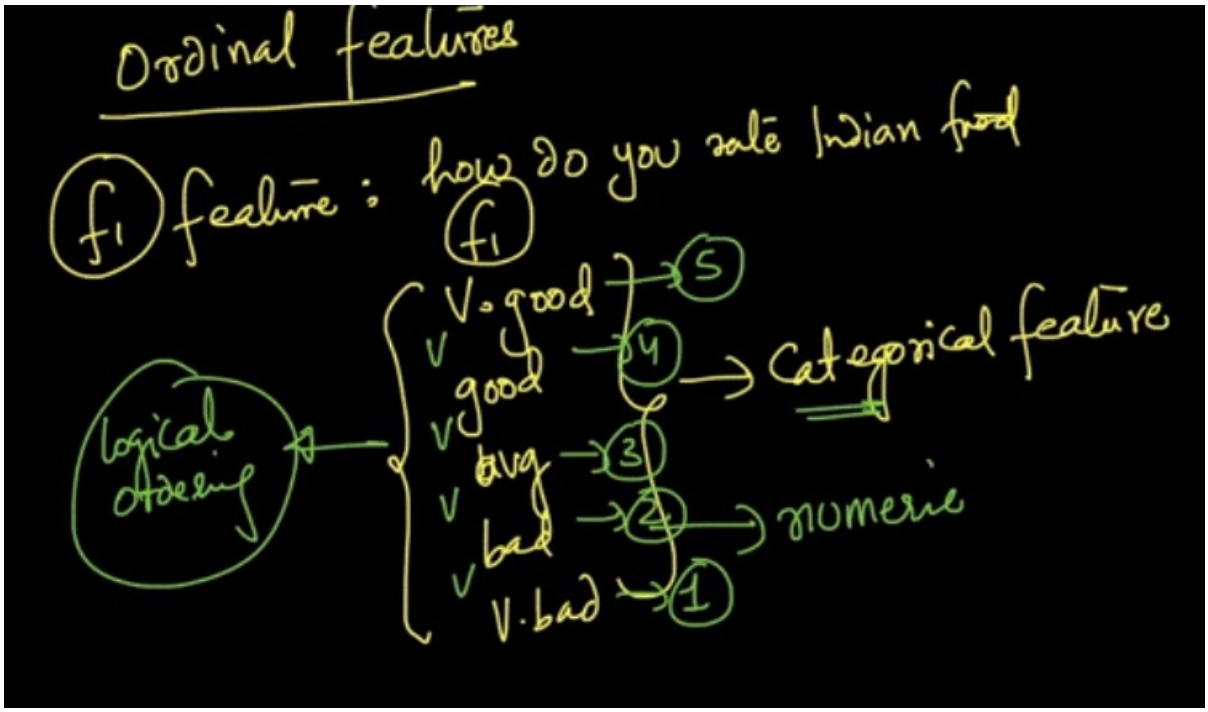


This strategy is the problem specific approach.

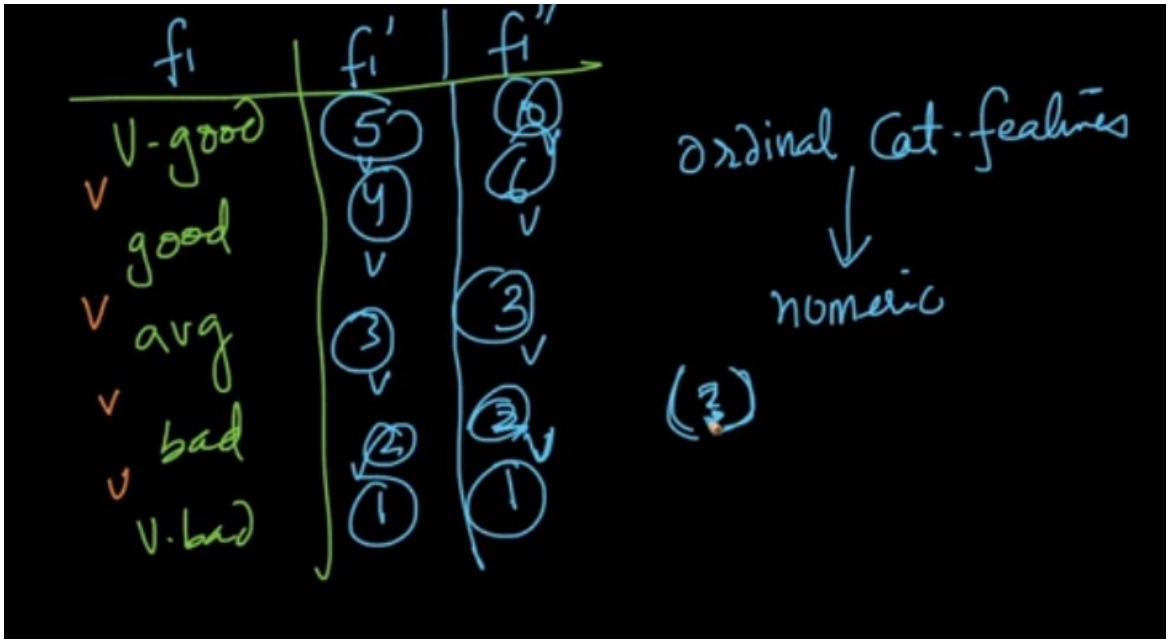
Another approach:
From domain knowledge we know that heights are related from one specific country.



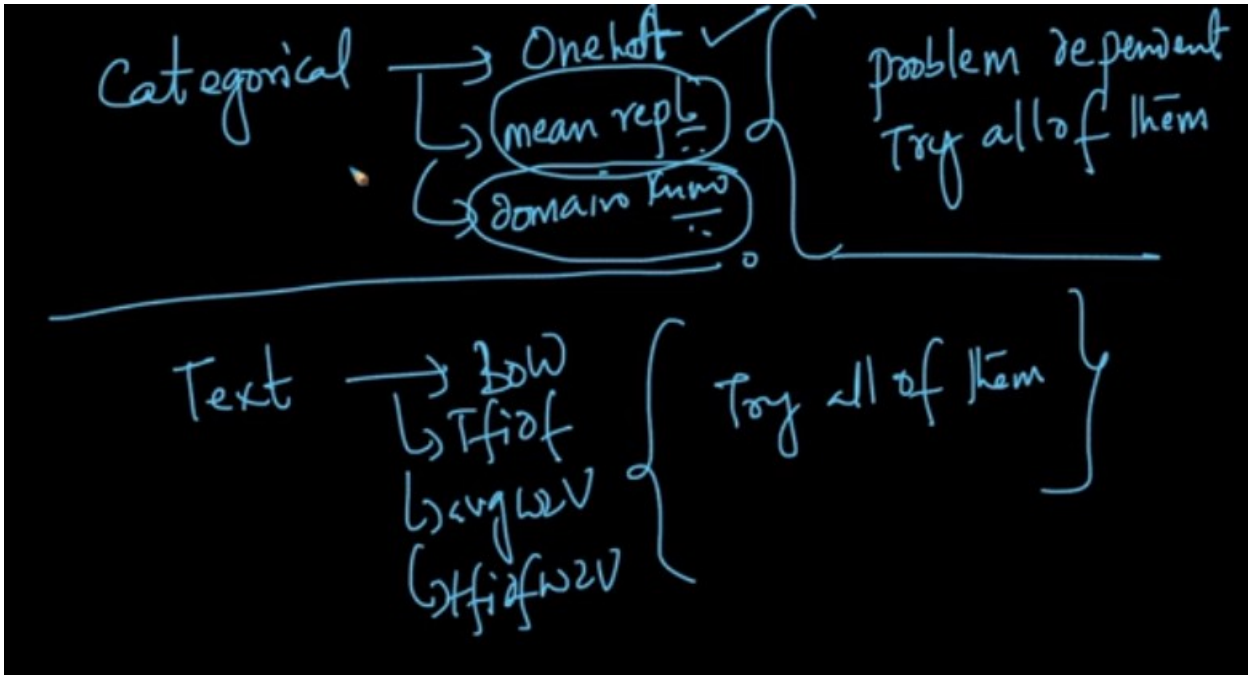
Ordinal features:
Features that can be ordered in one specific order. If there is an order in the categories of the feature then we can convert them into numeric features by assigning the numbers in the same order.



Ways of numeric encoding:
The order of giving numeric values is very problem specific.



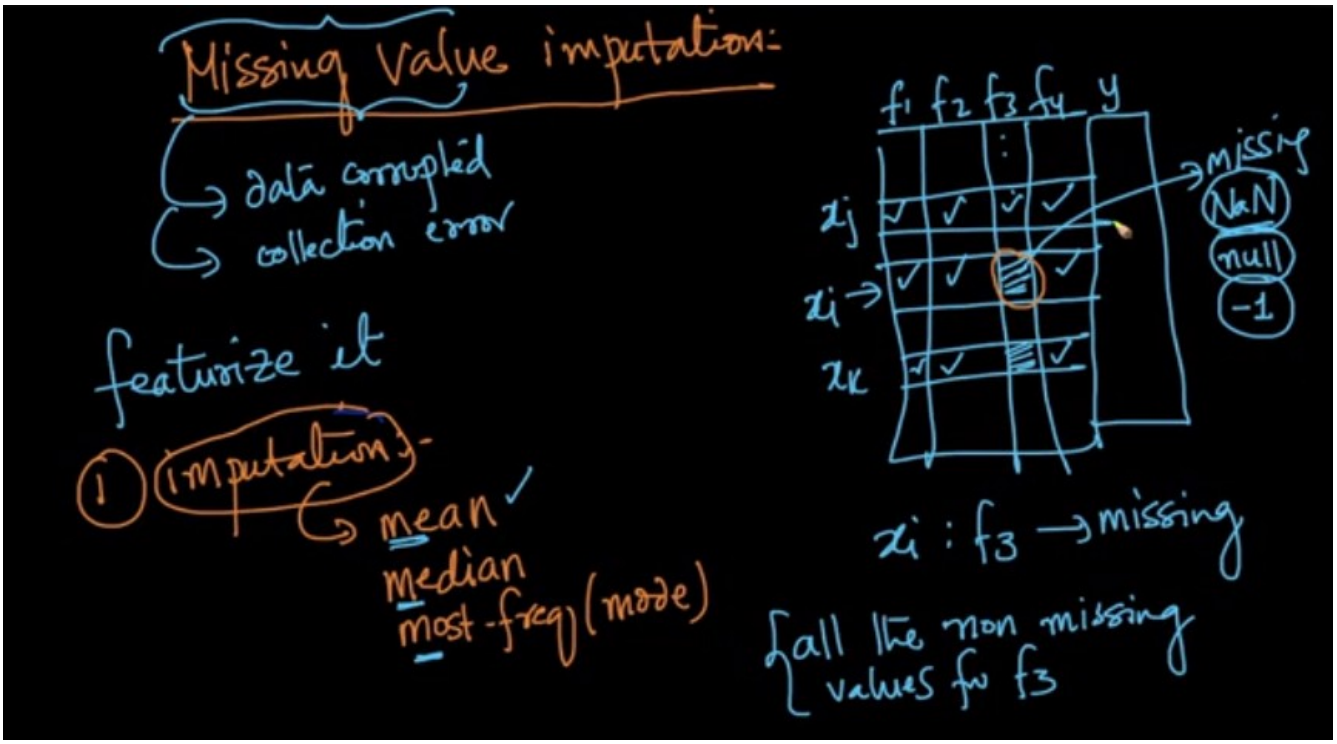
It is very problem specific to choose which will be the best for encoding the data.



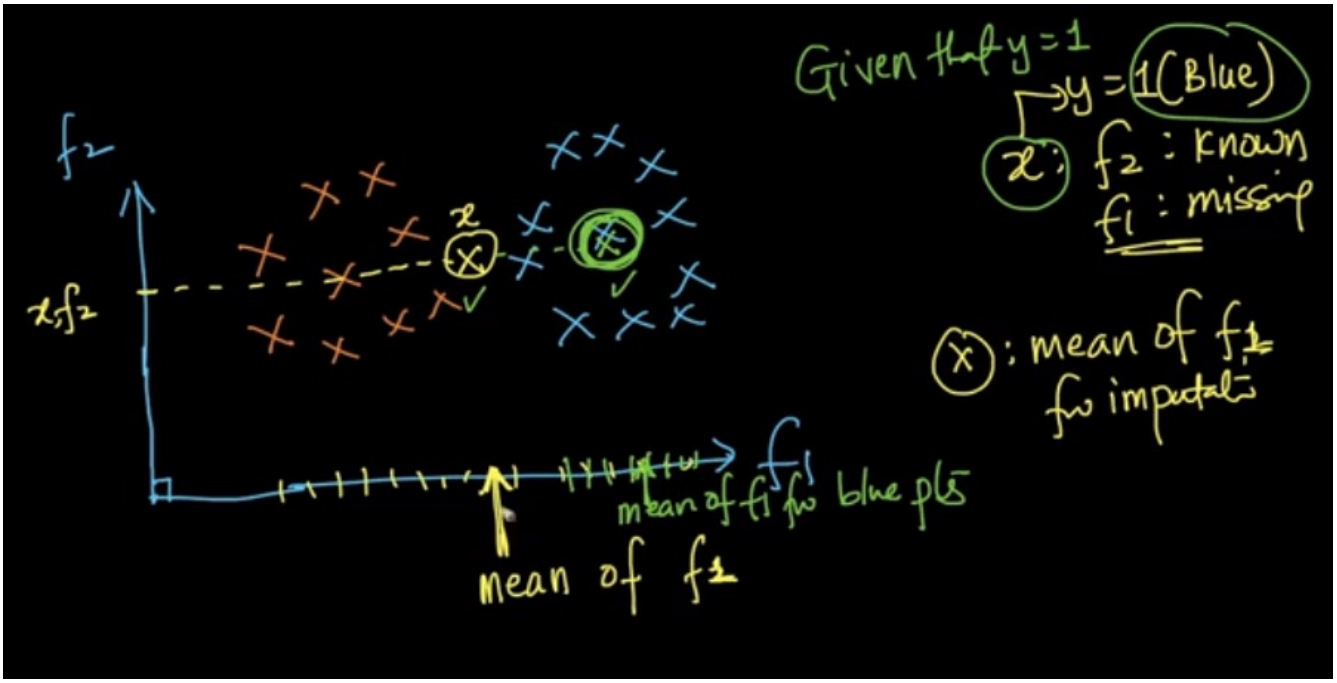
Handling missing values by imputation:

The missing values can be replaced by

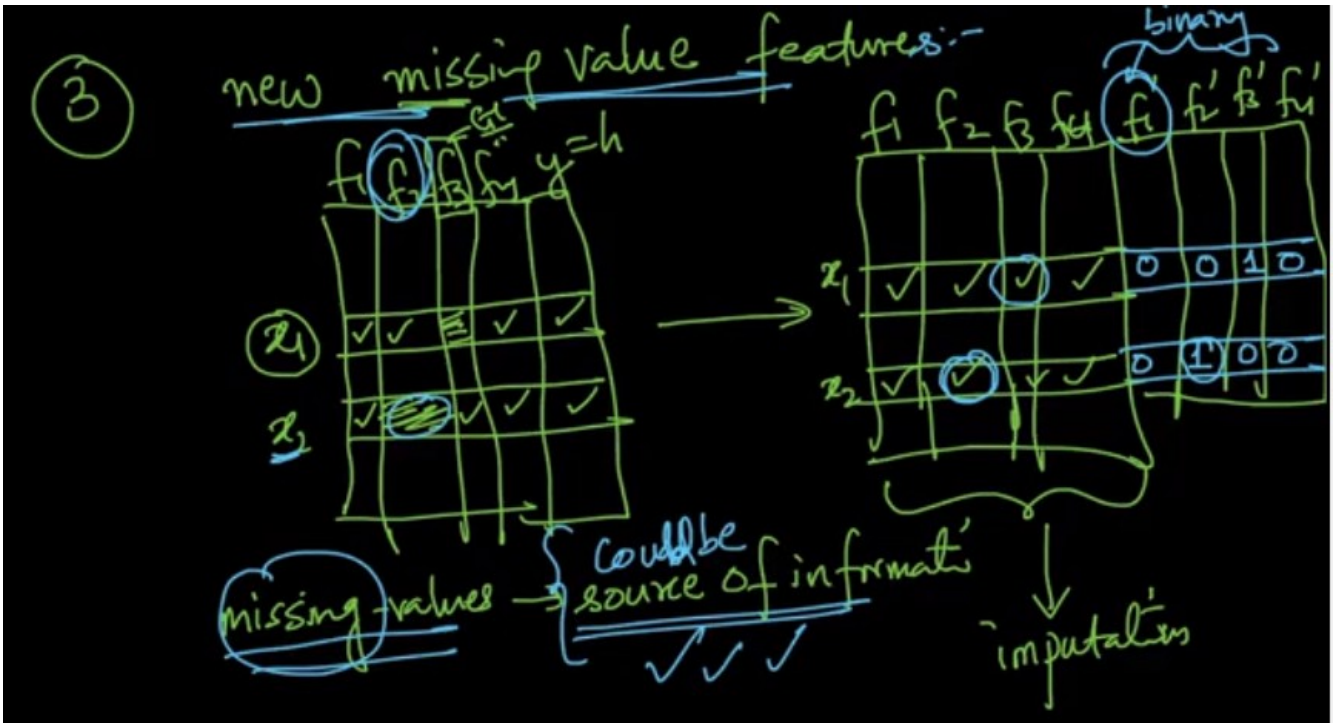
1. Mean
2. Median or
3. Mode.



Edge case of applying mean value.

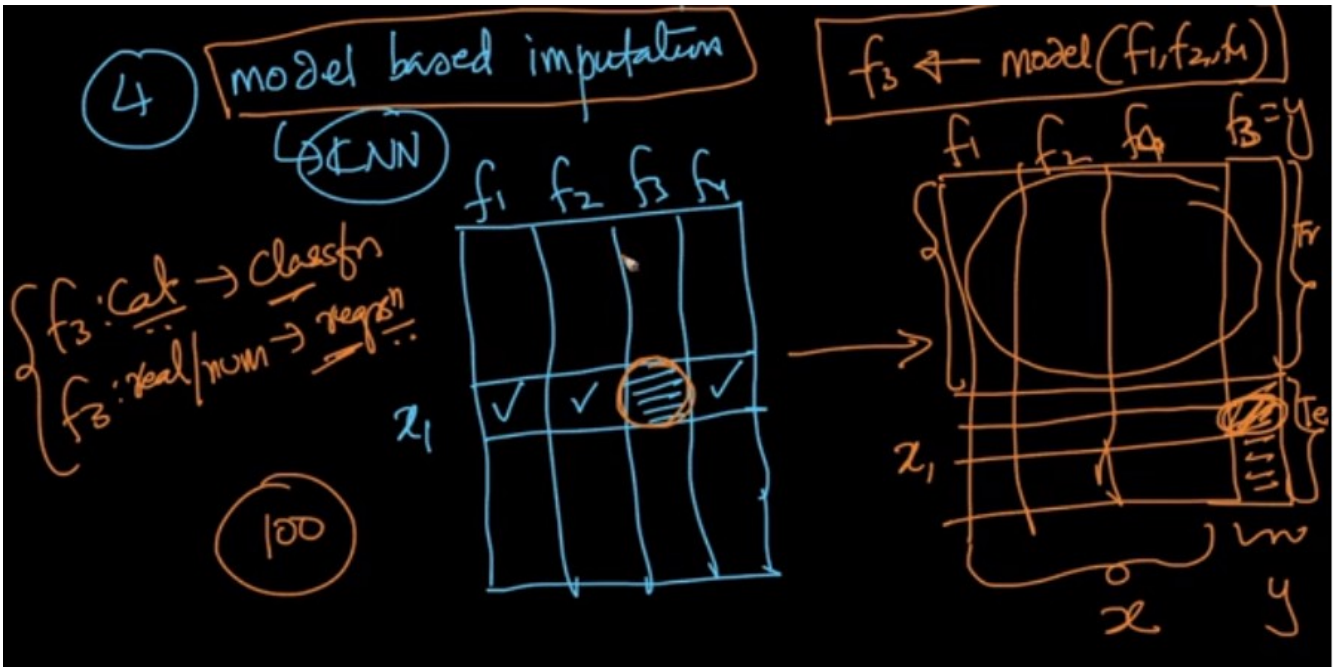


One more strategy of imputation is make a new features marking the features that are missing.



Model based imputation:

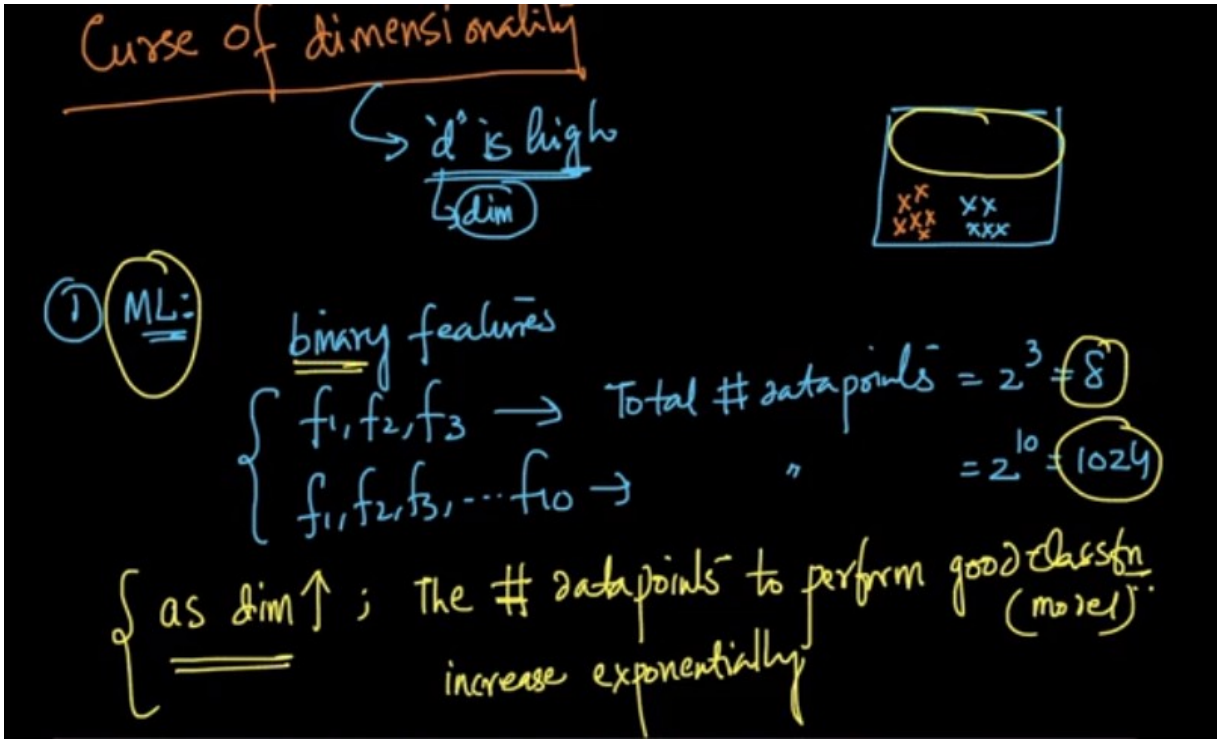
In this strategy we will make all the missing values as the test data and non – missing as training data and train KNN and predict the missing values in the test data.



KNN is most often used for model based imputation.

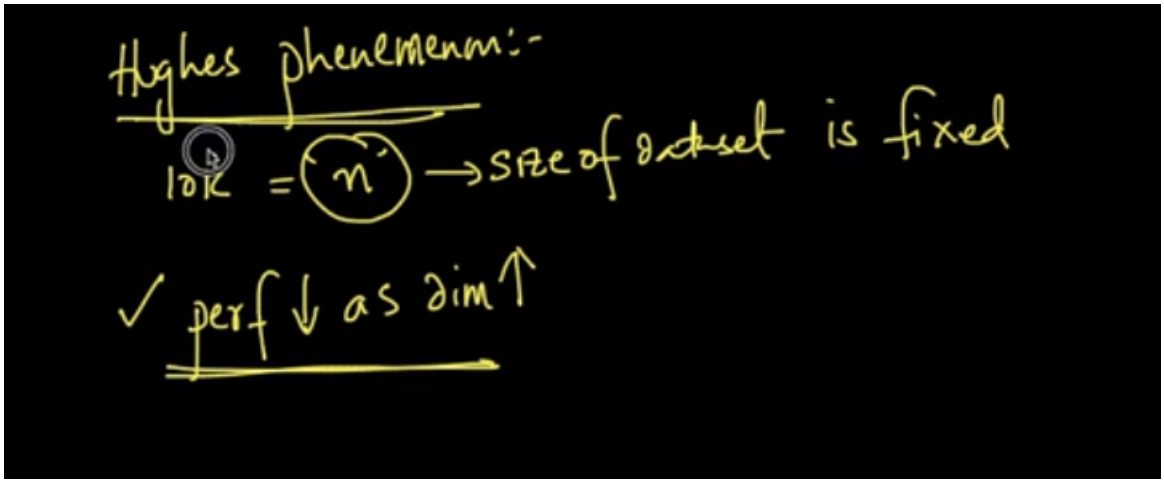
Curse of dimensionality:

As dimensionality increases then the number of data points to perform good classification increase exponentially.

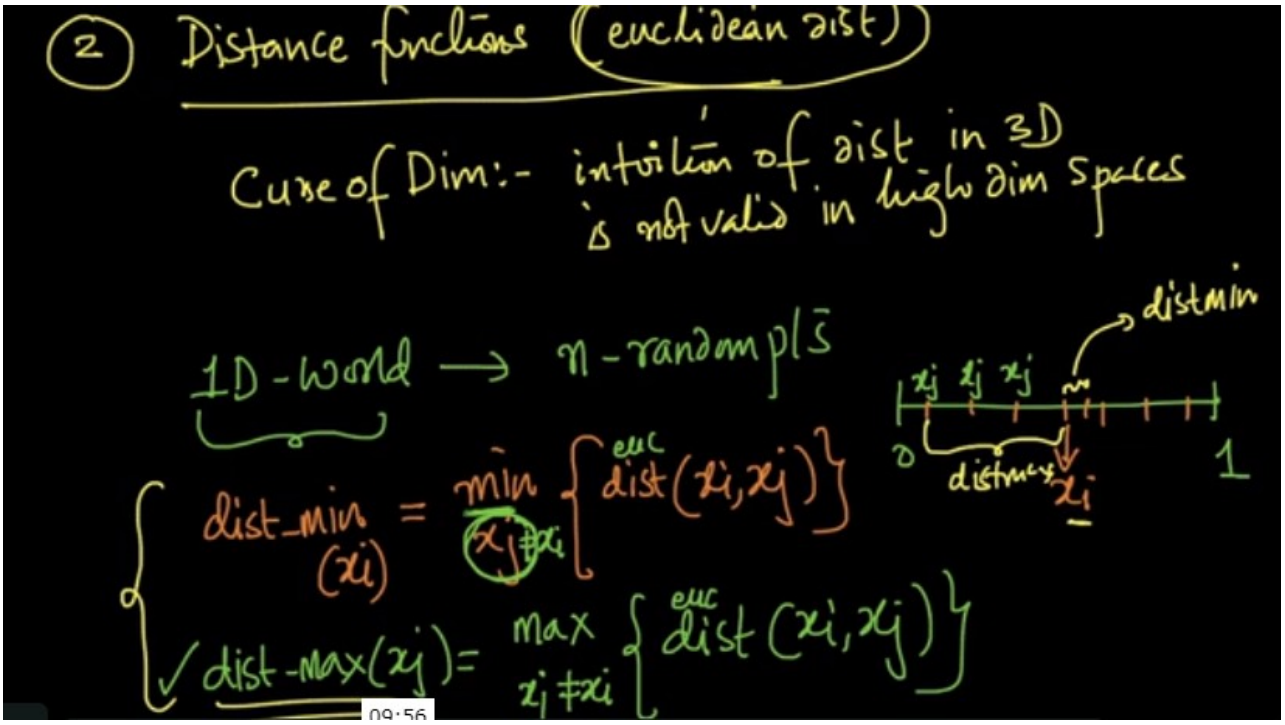


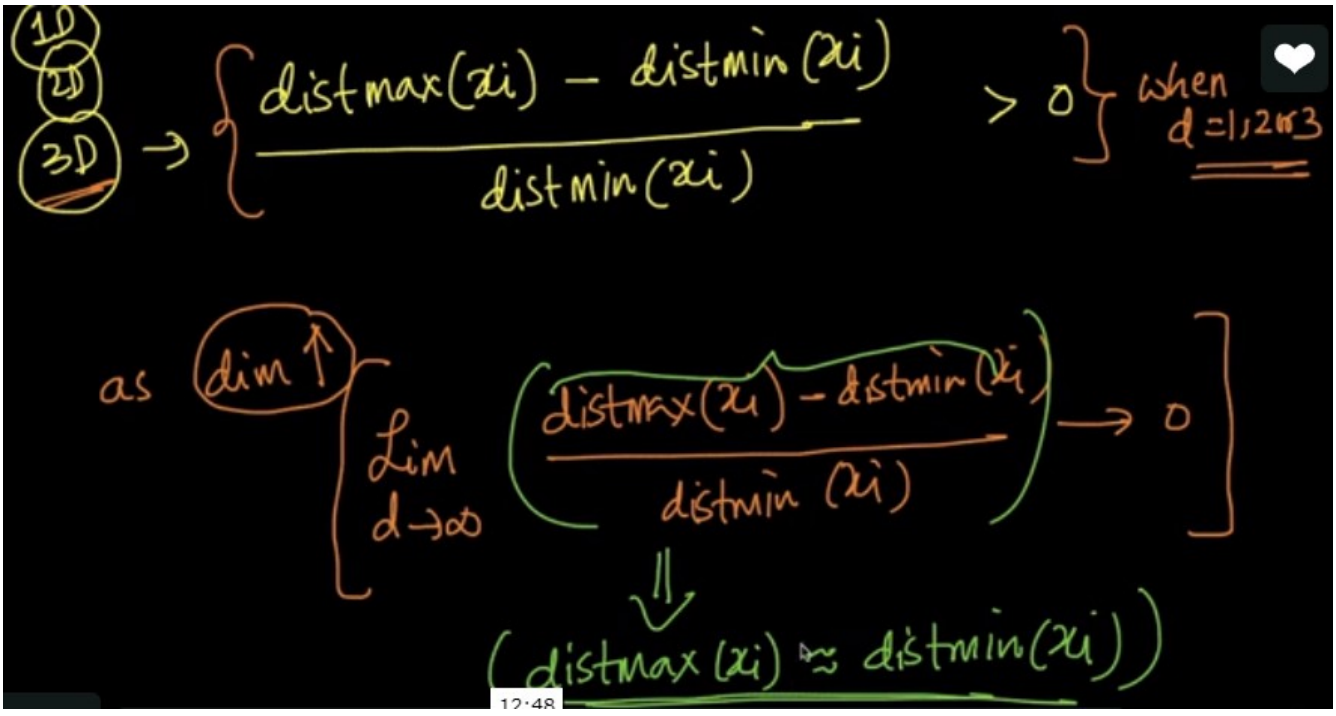
Hughes phenomenon:

If the size of the data set is fixed, the performance of the model decreases as the dimensions increase.

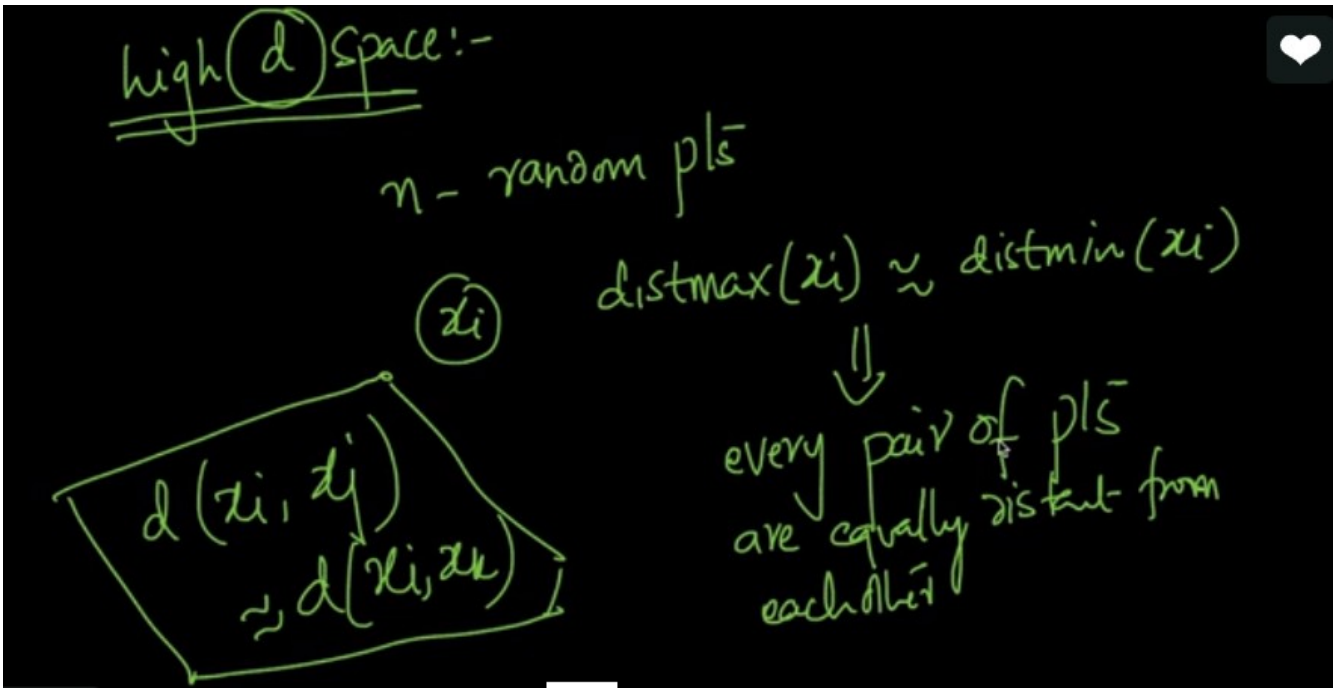


Effect on distance functions if dimensions increase:



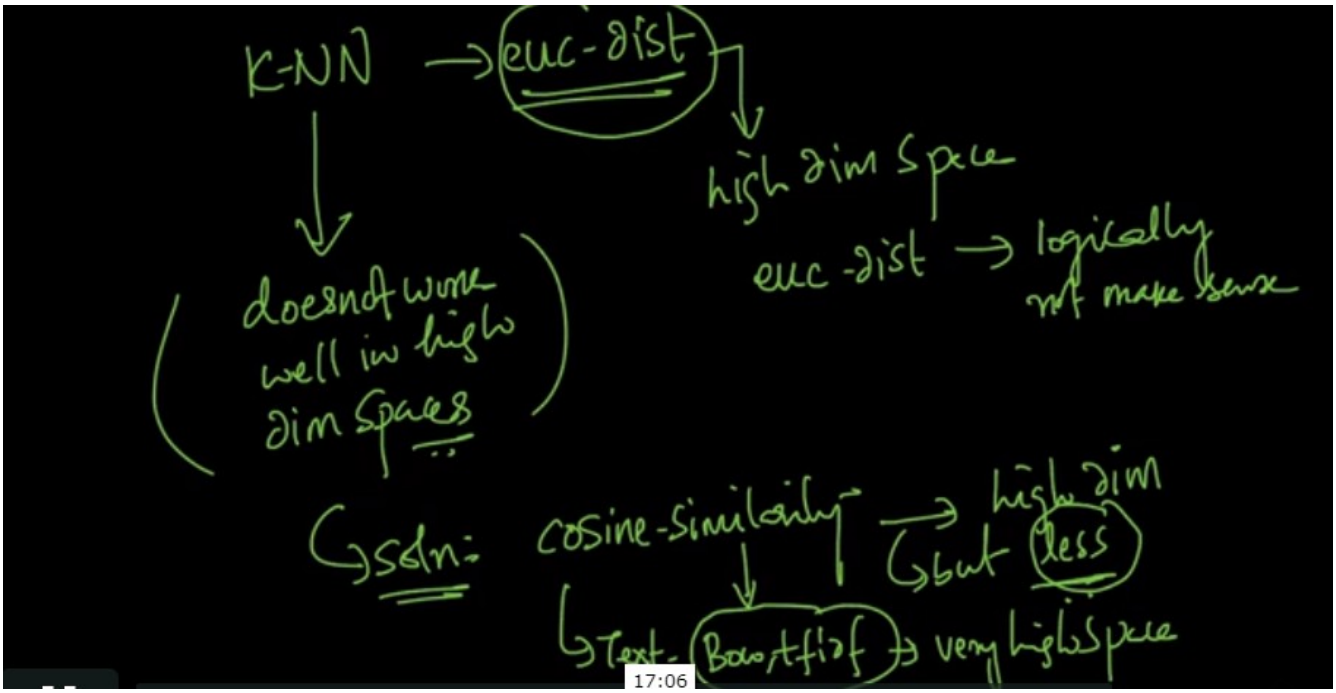


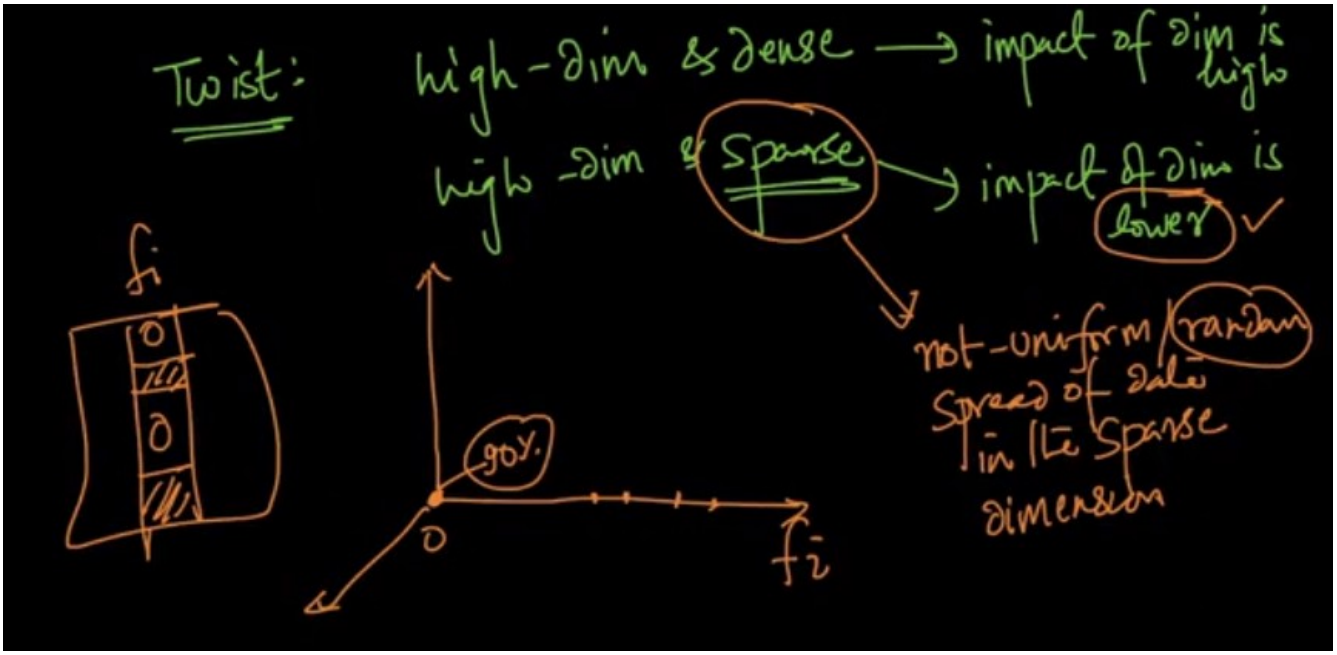
In a high dimensional space, if we take ‘n’ random points then its distance max is equal to distance min form a point. Every pair of points are equally distant from each other.



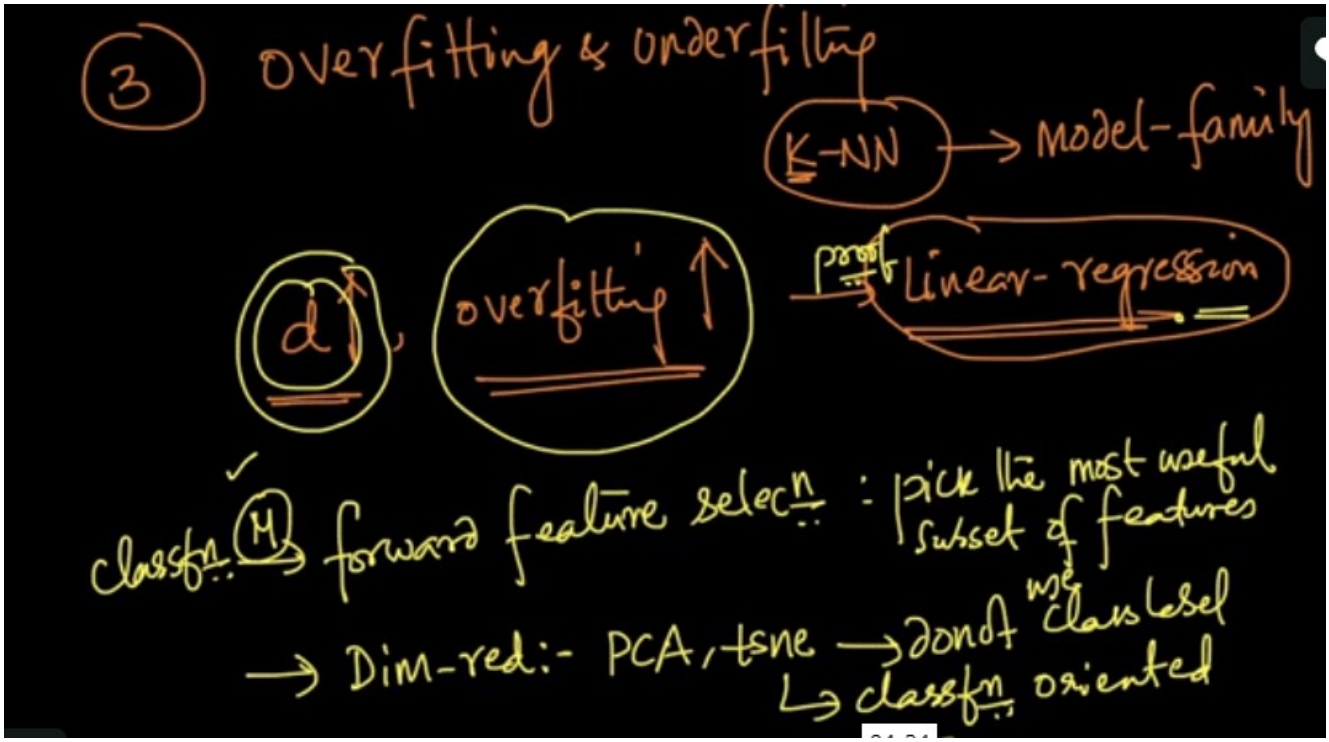
In high dimensional space euclidean distance does not logically make sense.

KNN does not work very well in high dimensional data.



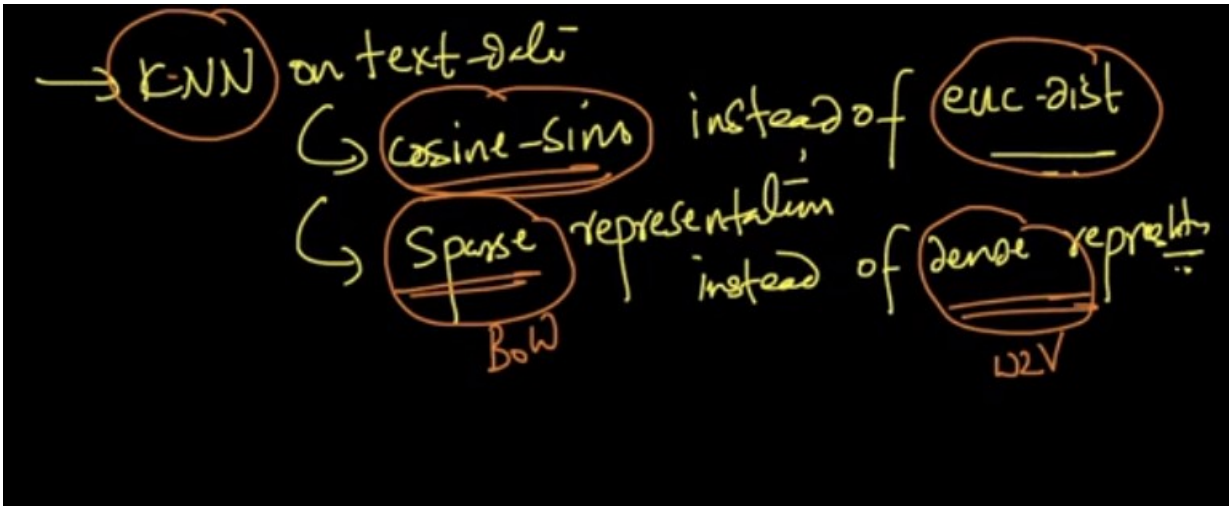


Overfitting and Underfitting:



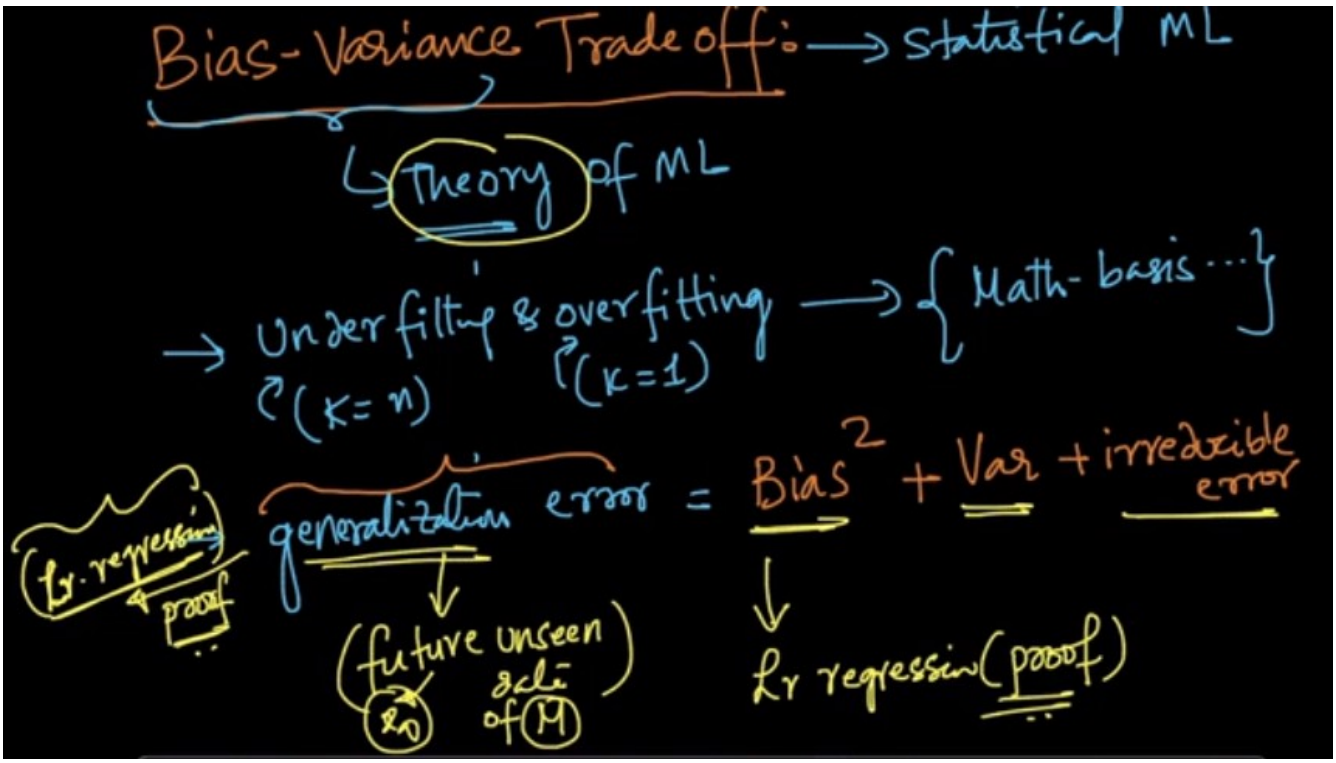
If there are large dimensions we can use forward feature selection for classification.

Ways implementing KNN when dimensions are more.



Bias – Variance trade off:

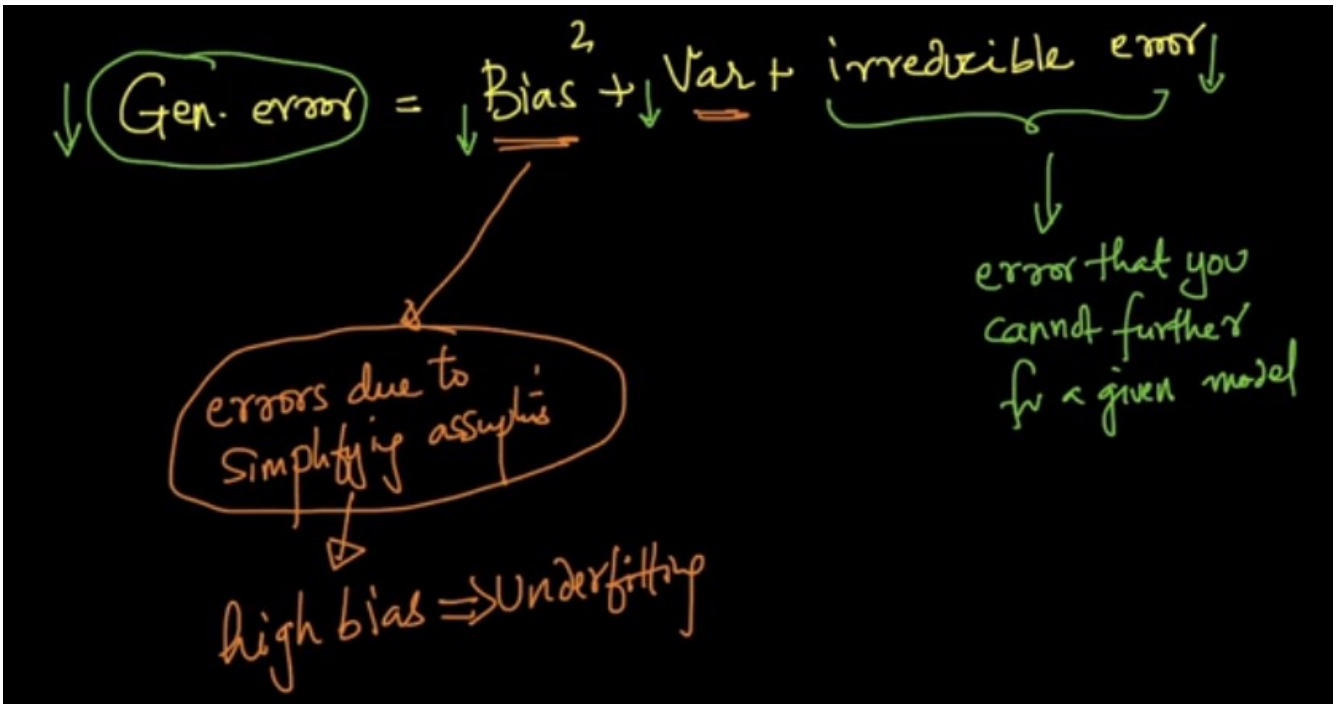
Equation of generalization error:

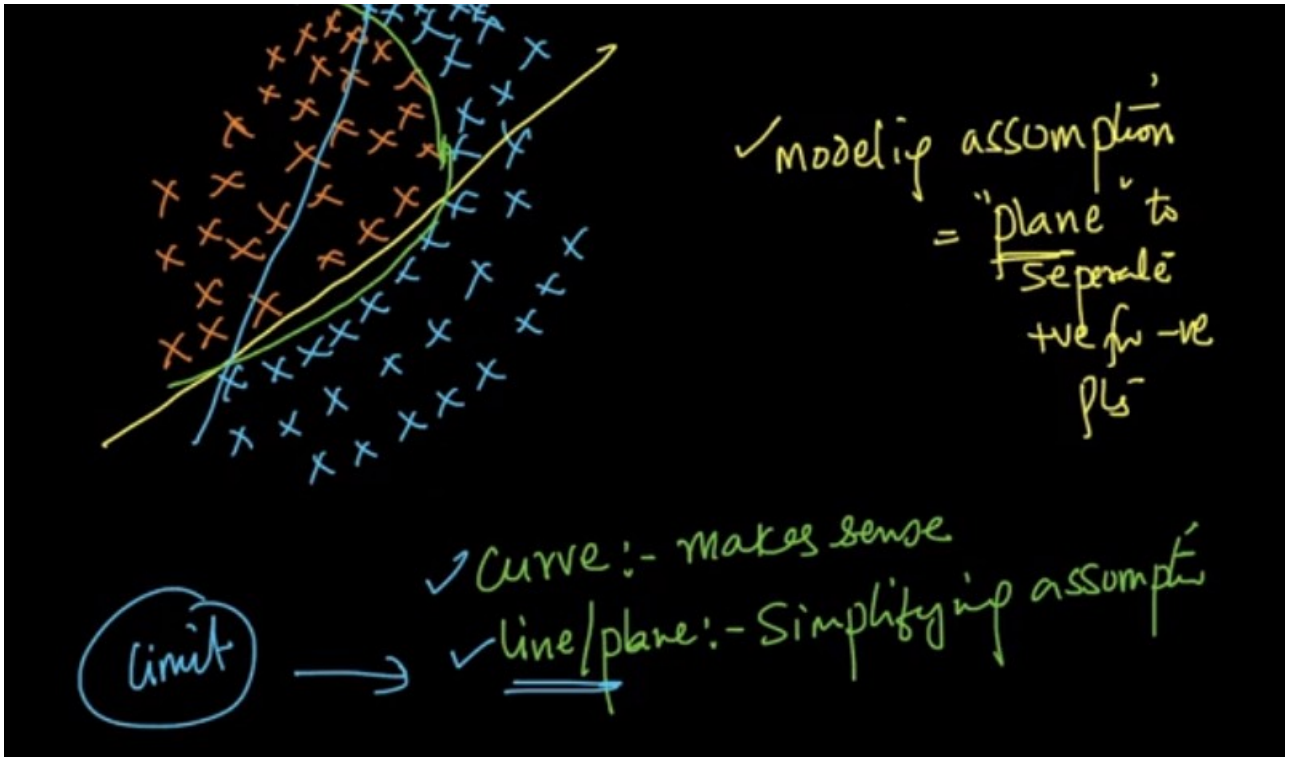


Irreducible error: This is the error that cannot be reduced further for a given model.

Bias error: This is due to simplified assumptions.

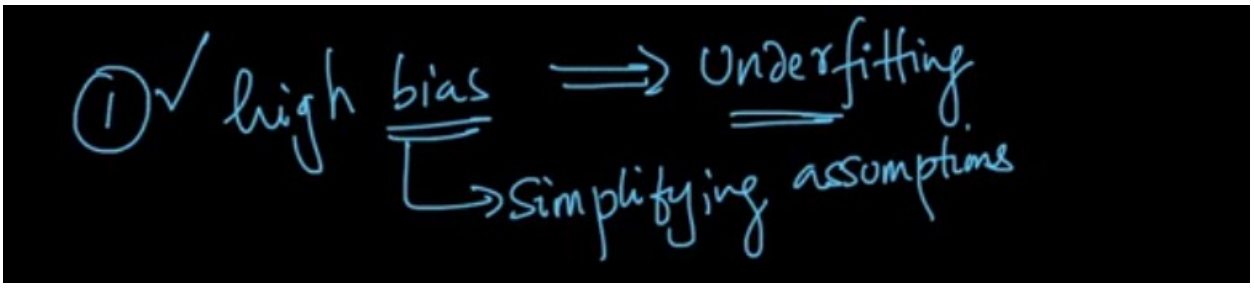
Variance:



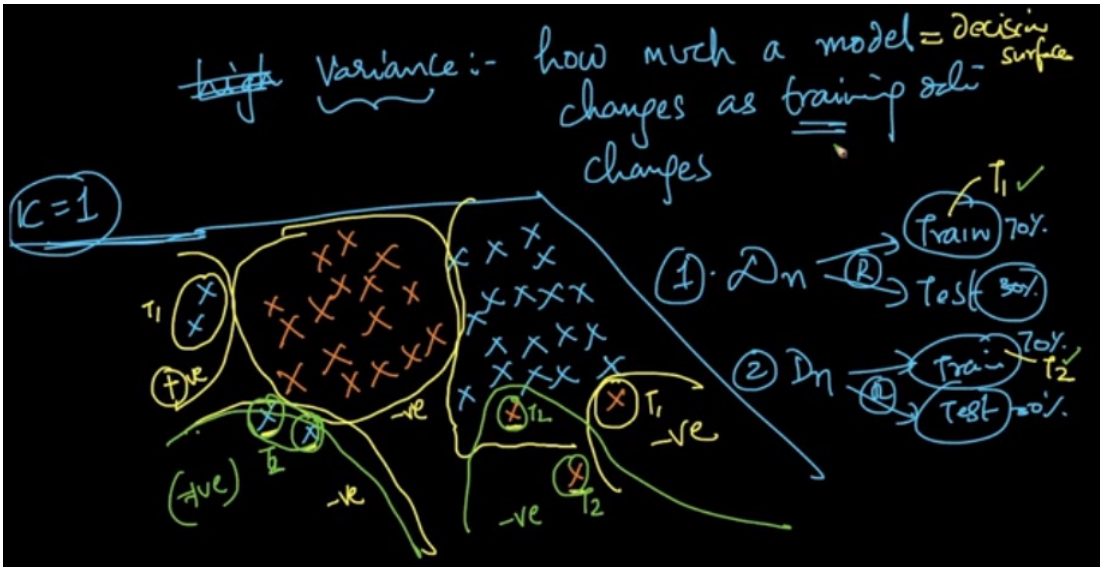


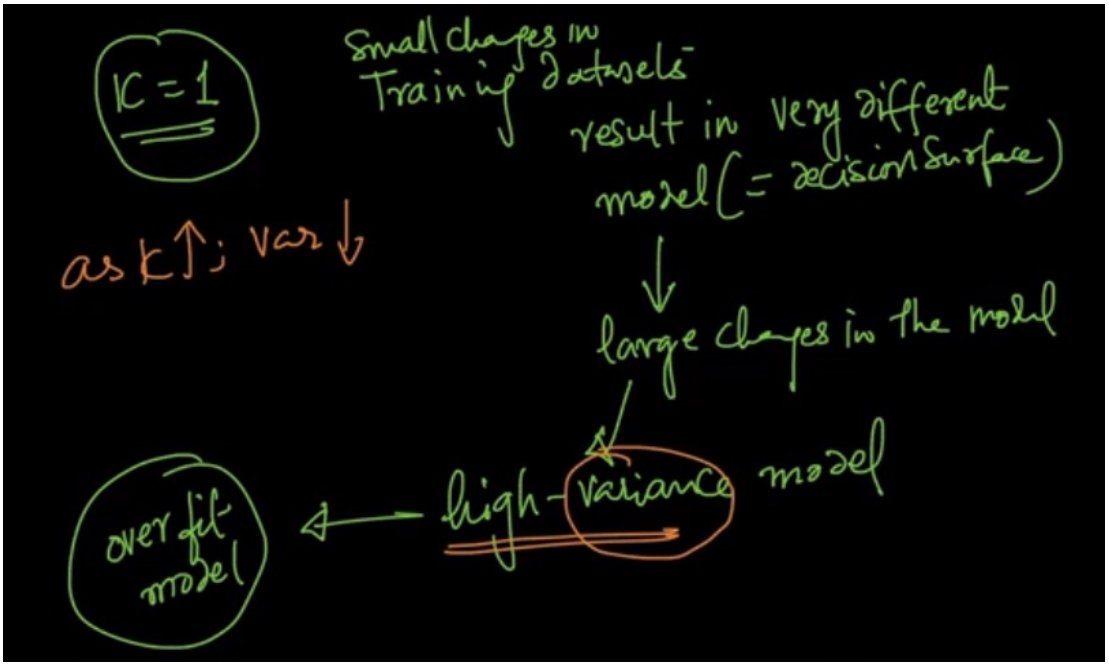
In KNN, when k = n the dominant class will be the class label of the query point.

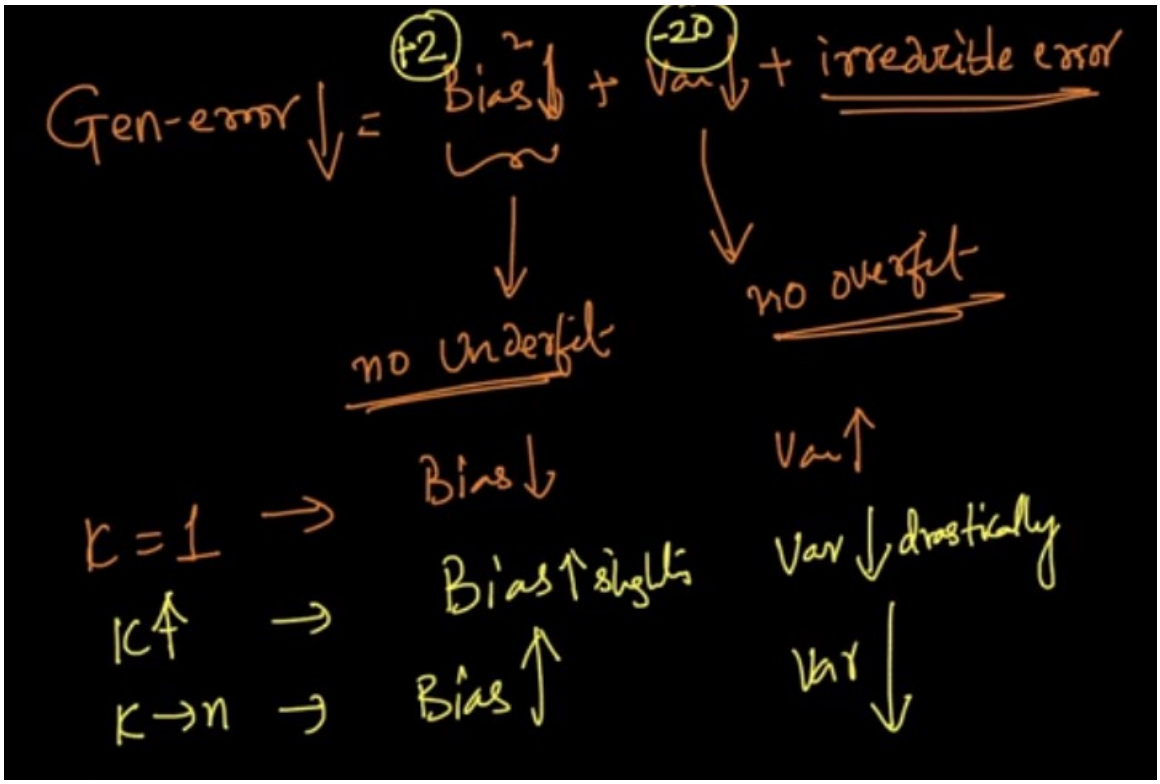
This means we are underfitting the data-set.



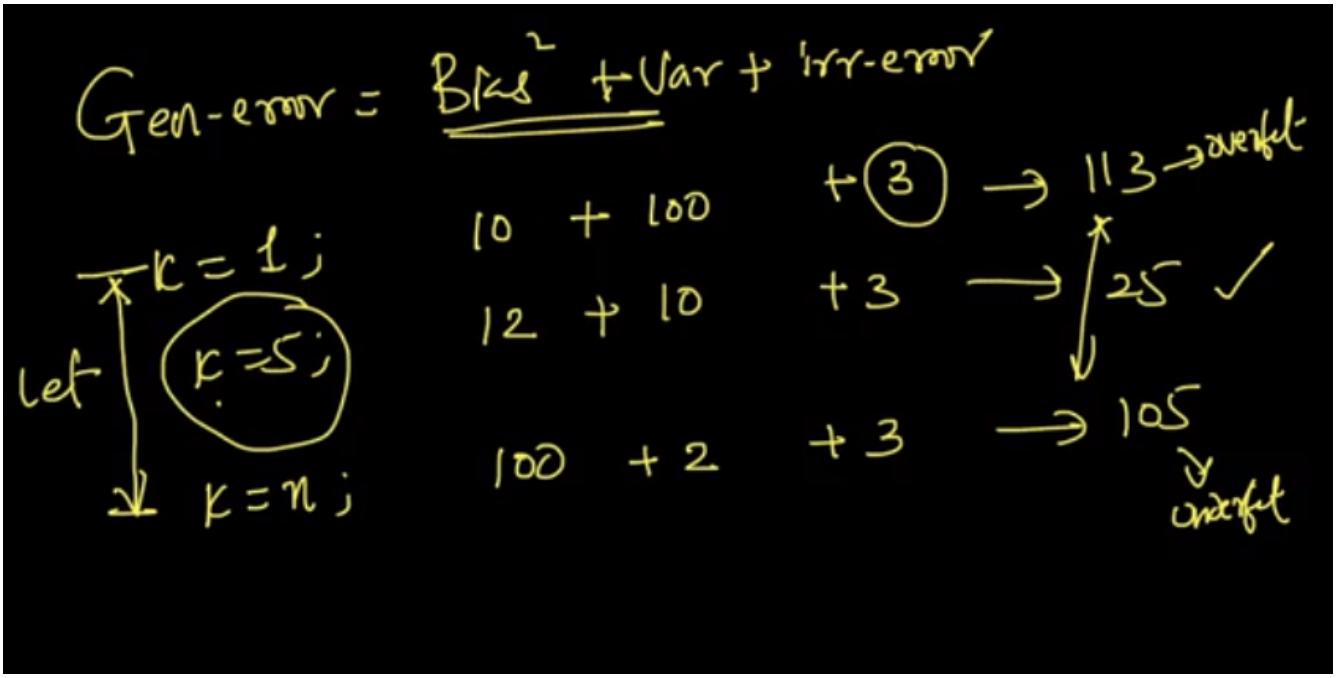
Variance: how much a model changes/decision surface as training data changes.







Balancing bias and variance will give the good model as output.



Best and worst cases for an algorithm: