Machine Learning

Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

Introduction

- Modelling and Evaluation - Introduction
- **Selecting a Model,** training a Model,
- Model Representation and Interpretability,
- Evaluating Performance of a Model,
- Improving Performance of a Model
- Basics of Feature Engineering: Introduction, Feature Transformation, Feature Subset Selection

Modelling and Evaluation - INTRODUCTION

- A machine is able to think and take intelligent action, and applying mathematical and statistical formulations
- machine learning struggles to build formulations or mapping based on a limited number of observations.
- the basic learning process, can be divided into three parts:
- 1. Data Input
- 2. Abstraction
- 3. Generalization

Example – Election crime

- In election, a criminal is going to launch an attack on the main candidate, and he is a criminal having a long record of serious crime.
- From the criminal database, a list of such criminals along with their photographs has been collected.
- They have to match the photos from the criminal database with the faces in the gathering to spot the potential attacker.
- based on certain salient physical features like
  - the shape of the jaw,
  - the slope of the forehead,
  - the size of the eyes,
  - the structure of the ear, etc.

Example – Election crime...

- a machine has no subjective baggage, no emotion, no bias due to past experience, and above all no mental fatigue.
- The machine can also use the same input data i.e. criminal database photos,
- apply computational techniques to abstract feature-based concept map from the input data
  - (the shape of the jaw, the slope of the forehead, the size of the eyes, the structure of the ear, etc.)
- generalize the same in the form of a classification algorithm to decide whether a face in the gathering is potentially criminal or not.

Modelling – Introduction…

- In learning process, abstraction represents raw input data in a summarized and structured format,
- such that a meaningful understanding is obtained from the data.
- This structured representation of raw input data to the meaningful pattern is called a model.

Modelling – Introduction...

- The model might have different forms.
  - It might be a mathematical equation,
  - it might be a graph or tree structure,
  - it might be a computational block, etc.
- The decision regarding which model is to be selected for a specific data set is taken by the learning task, based on
  - the problem to be solved and
  - the type of data.
- For example, when the problem is related to prediction and the target field is numeric and continuous, the regression model is assigned.

Modelling – Introduction…

- The process of assigning a model, and fitting a specific model to a data set is called model training.
- Once the model is trained, the raw input data is summarized into an abstracted form.
- with abstraction, the learner is able to summarize the knowledge.

Modelling – Introduction…

- Generalization searches through the huge set of abstracted knowledge to come up with a small and manageable set of key findings.
- If the outcome is systematically incorrect, the learning is said to have a bias

Machine Learning

Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

Selecting a Model

- Categories of Machine Learning Approaches
- Predictive models
- Descriptive models
- Popular Algorithms

Categories of Machine Learning Approaches

- Three broad categories of machine learning approaches used for resolving different types of problems
- 1. Supervised
  - 1. Classification
  - 2. Regression
- 2. Unsupervised
  - 1. Clustering
  - 2. Association analysis
- 3. Reinforcement
- For each of the cases, the model that has to be created/trained is different.
- Multiple factors play a role when we try to select the model for solving a machine learning problem

Categories of Machine Learning Approaches...

- Three types of problems

1. Predicting class values
2. Predicting numerical values
3. Predicting grouping of data

Categories of Machine Learning Approaches…

- It is very difficult to give a generic guidance related to which machine learning has to be selected, because, there is no one model that works best for every machine learning problem.
- This is what 'No Free Lunch' theorem also states.
- There is no single best optimization algorithm.
- Because of close relationship between optimization, search and machine learning.
- There is no single ML algorithm for predictive modeling problems such as classification and regression.

Categories of Machine Learning Approaches...

- It is very difficult to give a generic guidance related to which machine learning has to be selected, because, there is no one model that works best for every machine learning problem.
- This is what 'No Free Lunch' theorem also states.
- There is no single best optimization algorithm.
- Because of close relationship between optimization, search and machine learning.
- There is no single ML algorithm for predictive modeling problems such as classification and regression.

Categories of Machine Learning Approaches…

- Machine learning algorithms are broadly of two types:

1. models for supervised learning, which primarily focus on solving predictive problems and
2. models for unsupervised learning, which solve descriptive problems.

Predictive models

- Models for supervised learning or predictive models, try to predict certain value using the values in an input data set.
- The learning model attempts to establish a relation between the target feature, i.e. the feature being predicted, and the predictor features.
- The predictive models have a clear focus on what they want to learn and how they want to learn.

Predictive models …

- Predictive models, in turn, may need to predict the value of a category or class to which a data instance belongs to.
- Below are some examples:
- 1. Predicting win/loss in a cricket match
- 2. Predicting whether a transaction is fraud
- 3. Predicting whether a customer may move to another product

Predictive models - classification models

- The models which are used for prediction of target features of categorical value are known as classification models.
- The target feature is known as **a class** and the categories to which classes are divided into are called levels.
- Some of the popular classification models include
- k-Nearest Neighbor (kNN),
- Naïve Bayes, and
- Decision Tree.

Predictive models - regression models

- Predictive models may also be used to predict numerical values of the target feature based on the predictor features.
- Below are some examples:
  - 1. Prediction of revenue growth in the succeeding year
  - 2. Prediction of rainfall amount in the coming monsoon
  - 3. Prediction of potential flu patients and demand for flu shots next winter

Predictive models - regression models...

- The models which are used for prediction of the numerical value of the target feature of a data instance are known as regression models.
- popular regression models.
  - Linear Regression and
  - Logistic Regression models

Descriptive models

- Models for unsupervised learning or descriptive models are used to describe a data set or gain insight from a data set.
- There is no target feature or single feature of interest in case of unsupervised learning.
- Based on the value of all features, interesting patterns or insights are derived about the data set.
- Descriptive models which group together similar data instances, i.e. data instances having a similar value of the different features are called clustering models.

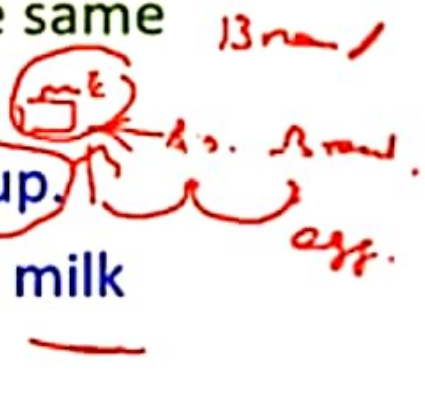
- Examples of clustering include
  - 1. Customer grouping or segmentation based on social, demographic, ethnic, etc. factors
  - 2. Grouping of music based on different aspects like genre, language, time-period, etc.
  - 3. Grouping of commodities in an inventory
- The most popular model for clustering is k-Means.

Descriptive models - Market Basket Analysis

- Descriptive models related to **pattern discovery** is used for market basket analysis of transactional data.
- In market basket analysis, based on the **purchase pattern** available in the transactional data,
- the possibility of purchasing one product based on the purchase of another product is determined.

Descriptive models - Market Basket Analysis...

- For example, transactional data may reveal a pattern that generally a customer who purchases milk also purchases biscuit at the same time.
- This can be useful for targeted promotions or in-store set up.
- Promotions related to biscuits can be sent to customers of milk products or vice versa.
- Also, in the store products related to milk can be placed close to biscuits.



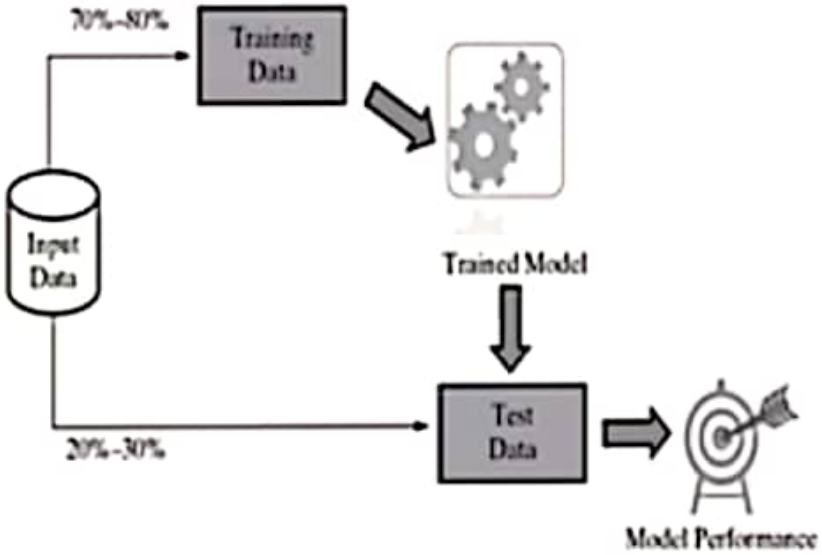
Machine Learning

Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

Training a Model (for Supervised Learning)

- Holdout method
- K-fold Cross-validation method
- Bootstrap sampling
- Lazy vs. Eager learner



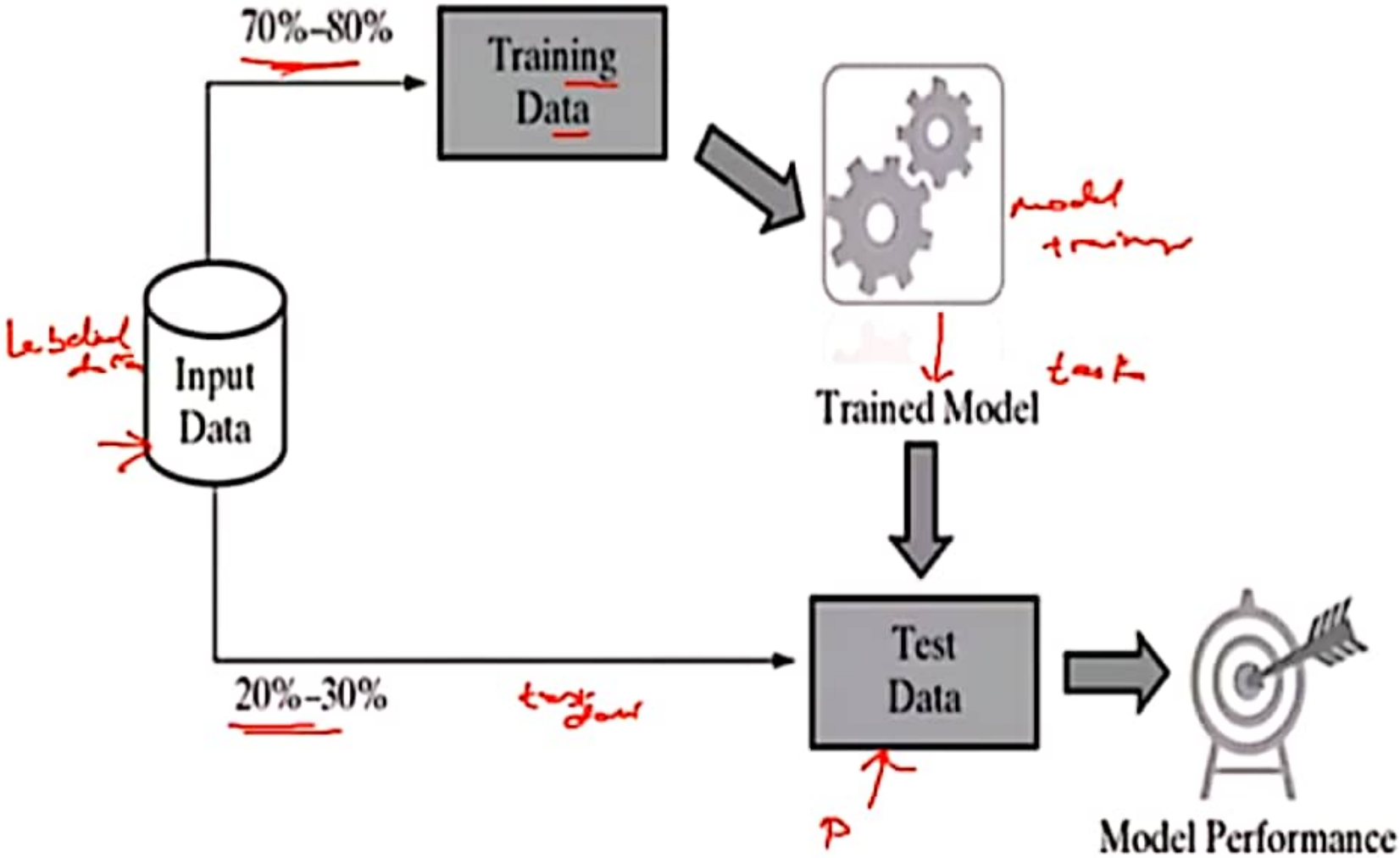
Holdout Method

- In supervised learning, a model is trained using the labelled input data.
- The test data may not be available immediately, also, the label value of the test data is not known.
- That is the reason why a part of the input data is held back (holdout) for evaluation of the model.
- This subset of the input data is used as the test data for evaluating the performance of a trained model.
- In general 70%–80% of the input data (labelled) is used for model training.
- The remaining 20%–30% is used as test data for validation of the performance of the model.

Holdout Method...

- A different proportion of dividing the input data into training and test data is also acceptable, and the division is done randomly.
- This method of partitioning the input data into two parts and both are similar in nature –
  - **training and test data,**
- which is by **holding back a part of the input data for validating the trained model is known as holdout method.**

Holdout method



Holdout Method...

- Once the model is trained using the training data, the labels of the test data are predicted using the model's target function
- Then the predicted value is compared with the actual value of the label.
- The performance of the model is in general measured by the accuracy of prediction of the label value.

Holdout Method...

- Some times, the input data is partitioned into three portions –
  - a training
  - a test data, and
  - a third validation data.
- The test data is used only for once.
- The validation data used for measuring the model performance.
- It is used in iterations and to refine the model in each iteration.

Holdout Method...

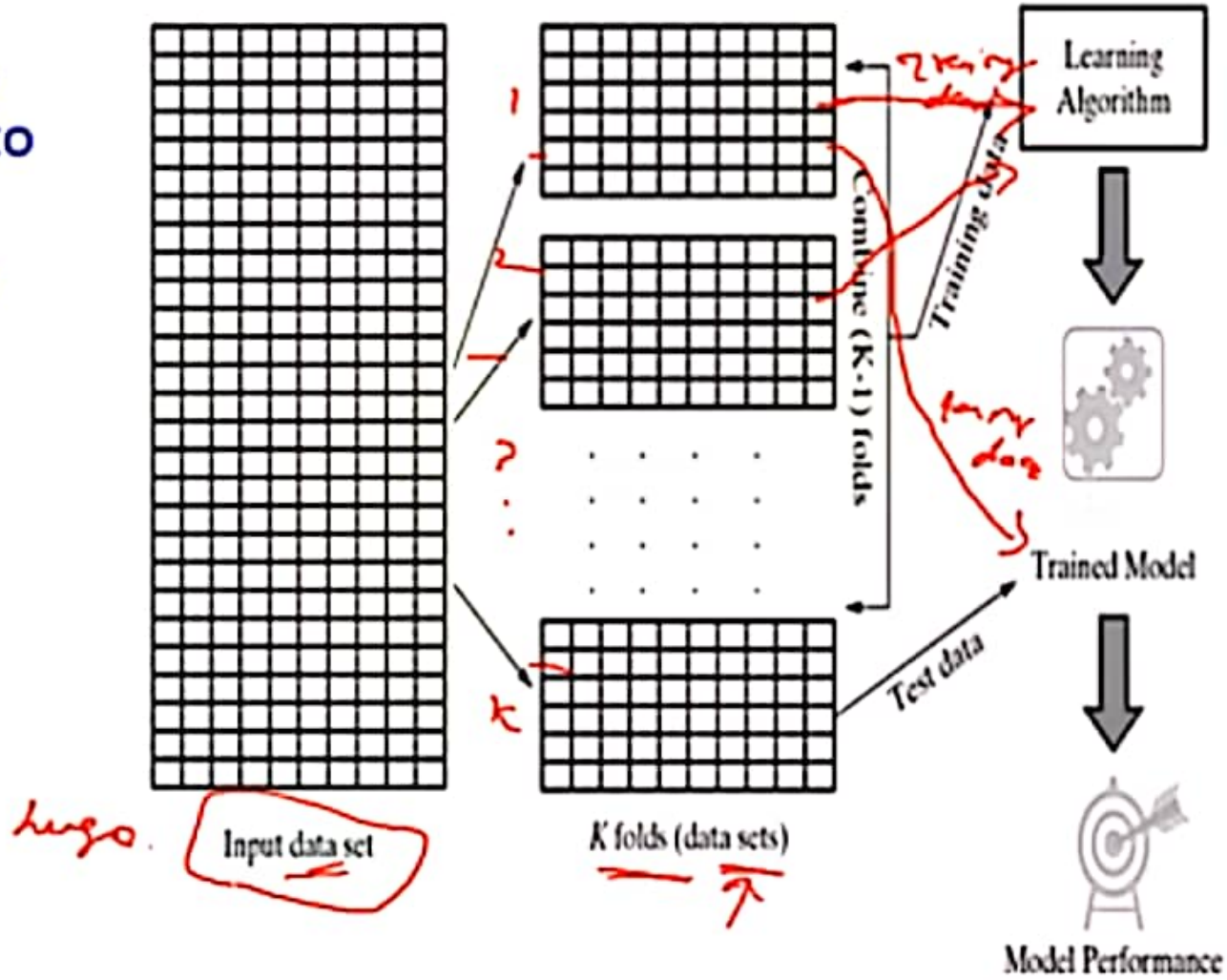
- If the volume of input data is huge, then
- Random sampling is employed for test data selection.
- the whole data is broken into several homogenous groups
- a random sample is selected from each group.
- This ensures that the generated random partitions have equal proportions of each class.

K-fold Cross-validation method

- The issues in random sampling approach, in Holdout method,
- 1. the smaller data sets - difficult to divide the data of some of the classes proportionally amongst training and test data sets.
- 2. A repeated holdout, is sometimes used to ensure the randomness of the composed data sets.
- Several random holdouts are used to measure the model performance.
- In the end, the average of all performances is taken.
- As multiple holdouts have been drawn, the training and test data (and validation data) are contain representative data from all classes and resemble the original input data closely.
- This process of repeated holdout is the basis of **k-fold cross-validation technique.**

Overall approach for K-fold cross-validation

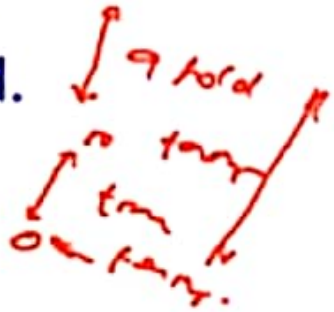
- In k-fold cross-validation, the data set is divided into k-completely distinct or non-overlapping random partitions called folds.



- The value of 'k' in k-fold cross-validation can be set to any number.
- there are two approaches which are extremely popular:
- 1. 10-fold cross-validation (10-fold CV)
- 2. Leave-one-out cross-validation (LOOCV)

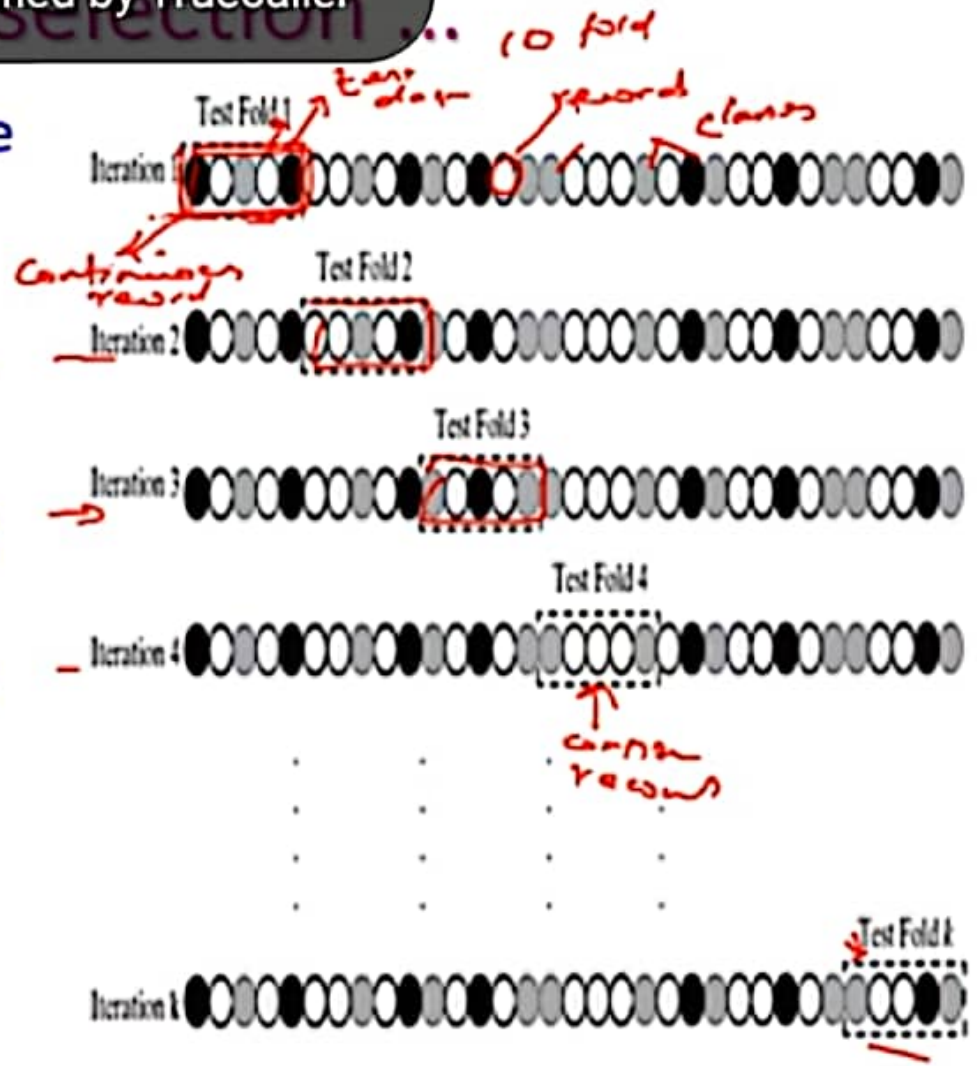
10-fold cross-validation

- 10-fold cross-validation is by far the most popular approach.
- for each of the 10-folds, each comprising of approximately 10% of the data, one of the folds is used as the **test data** for validating model performance trained based on the remaining 9 folds (or 90% of the data).
- This is repeated 10 times, once for each of the 10 folds being used as the test data and the remaining folds as the training data.
- The average performance across all folds is being reported.



Detailed approach for fold selection ...

- each of the circles resembles a record in the input data set whereas the different colors indicate the different classes that the records belong to.
- The entire data set is broken into 'k' folds – out of which one fold is selected in each iteration as the test data set.
- The fold selected as test data set in each of the 'k' iterations is different.
- the contiguous circles represented as folds, do not mean that they are subsequent records in the data set.
- the records in a fold are drawn by using random sampling technique.

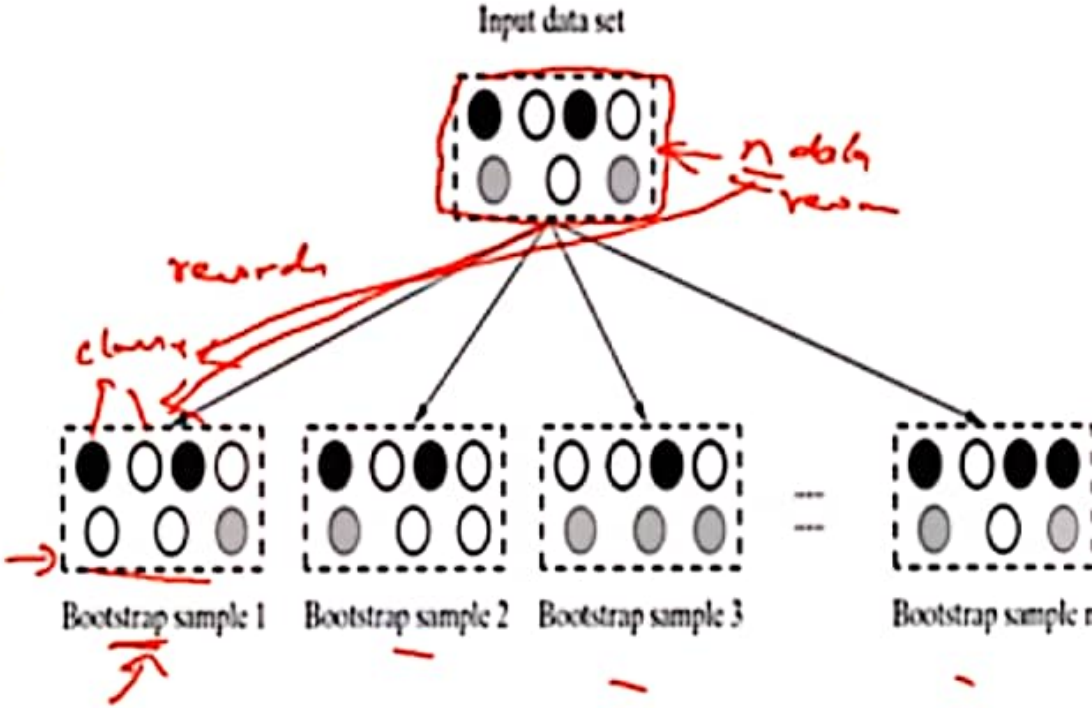


Leave-one-out cross-validation (LOOCV)

- Leave-one-out cross-validation (LOOCV) is an extreme case of k-fold cross-validation using one record or data instance at a time as a test data.
- This is done to maximize the count of data used to **train the model**.
- the number of **iterations** for which it has to be run is equal to the total number of data in the input data set.
- it is computationally very expensive and not used much in practice.

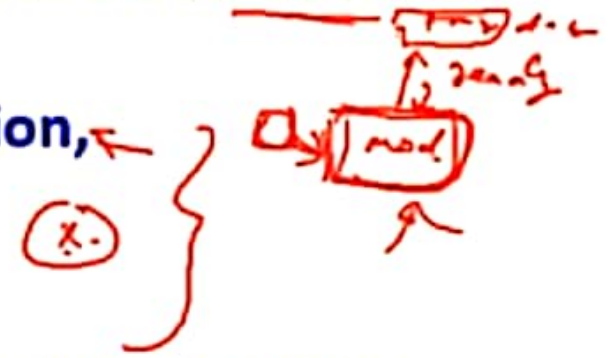
Bootstrap sampling

- It is a popular way to identify training and test data sets from the input data set.
- It uses the technique of Simple Random Sampling with Replacement (SRSWR).
- Bootstrapping randomly picks data instances from the input data set
- **the input data set having 'n' data instances,**
- **bootstrapping can create one or more training data sets having 'n' data instances,**
- **some of the data instances being repeated multiple times.**
- The Bootstrap sampling is useful in case of input data sets of small size, i.e. having very less number of data instances



Eager learner

- Eager learning follows the general principles of machine learning – it tries to construct a generalized, input independent target function during the model training phase.
- It uses Abstraction, generalization and comes up with a trained model at the end of the learning phase.
- Hence, when **the test data comes in for classification,**
- **the eager learner is ready with the model and**
- **doesn't need to refer back to the training data.**
- Eager learners take more time in the learning phase than the lazy learners.
- Some of the algorithms which adopt eager learning approach include Decision Tree, Support Vector Machine, Neural Network, etc.



Lazy learning

- Lazy learning, completely skips the abstraction and generalization processes, otherwise, lazy learner doesn't 'learn' anything.
- It uses the training data in exact, and uses the knowledge to classify the unlabelled test data.
- it is also known as rote learning (i.e. memorization technique based on repetition).
- Due to its heavy dependency on the given training data instance, it is also known as instance learning or non-parametric learning.
- Lazy learners take very little time in training because not much of training actually happens.
- it takes long time in classification as for each attribute in record of test data, a comparison-based assignment of label happens.
- The algorithm for lazy learning is k-nearest neighbor

Thank You

- **UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering**
- **Training a Model (for Supervised Learning)**
- Holdout method
- K-fold Cross-validation method
- Bootstrap sampling
- Lazy vs. Eager learner

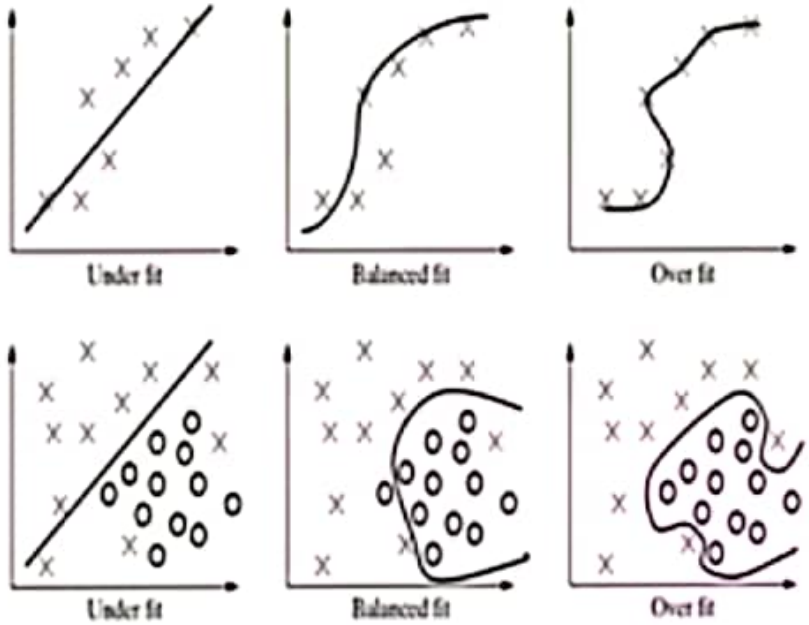
Machine Learning

Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

MODEL REPRESENTATION AND INTERPRETABILITY

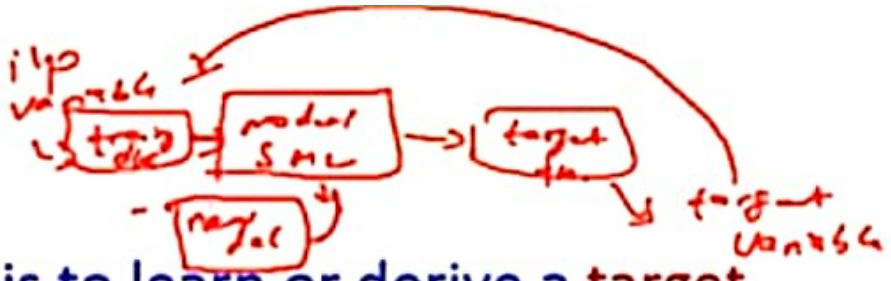
- Underfitting
- Overfitting
- Bias – variance trade-of
  - Errors due to ‘Bias’
  - Errors due to ‘Variance’



MODEL REPRESENTATION

- The goal of supervised machine learning is to learn or derive a target function which can best determine the target variable from the set of input variables.
- Learning the target function from the training data is the extent of generalization.
- The input data is a limited, and specific
- the new, unknown data in the test data set, may be differing from the training data.
- Fitness of a target function approximated by a learning algorithm determines how correctly it is able to classify a set of data, that it has never seen.

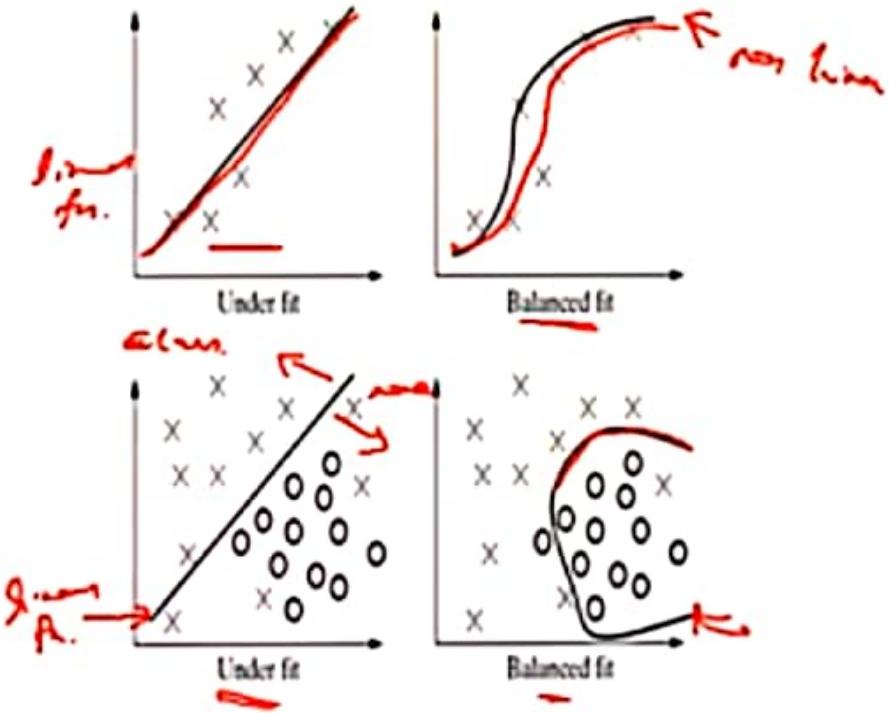
MODEL REPRESENTATION



- The goal of supervised machine learning is to learn or derive a target function which can best determine the target variable from the set of input variables.
- Learning the target function from the training data is the extent of generalization.
- The input data is a limited, and specific
- the new, unknown data in the test data set, may be differing from the training data.
- Fitness of a target function approximated by a learning algorithm determines how correctly it is able to classify a set of data, that it has never seen.

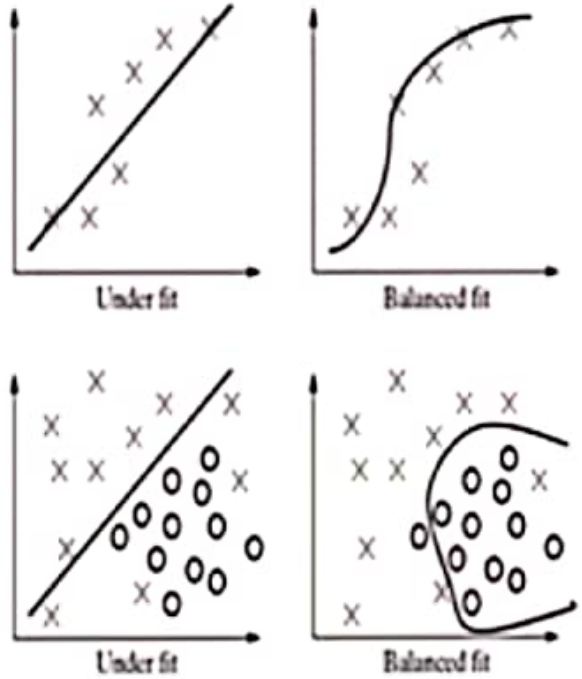
Underfitting

- If the target function is kept too simple, it may not be able to capture the essential output and represent the underlying data well.
- Underfitting may occur when trying to represent a non-linear data with a linear model as demonstrated by both cases of underfitting shown in figure



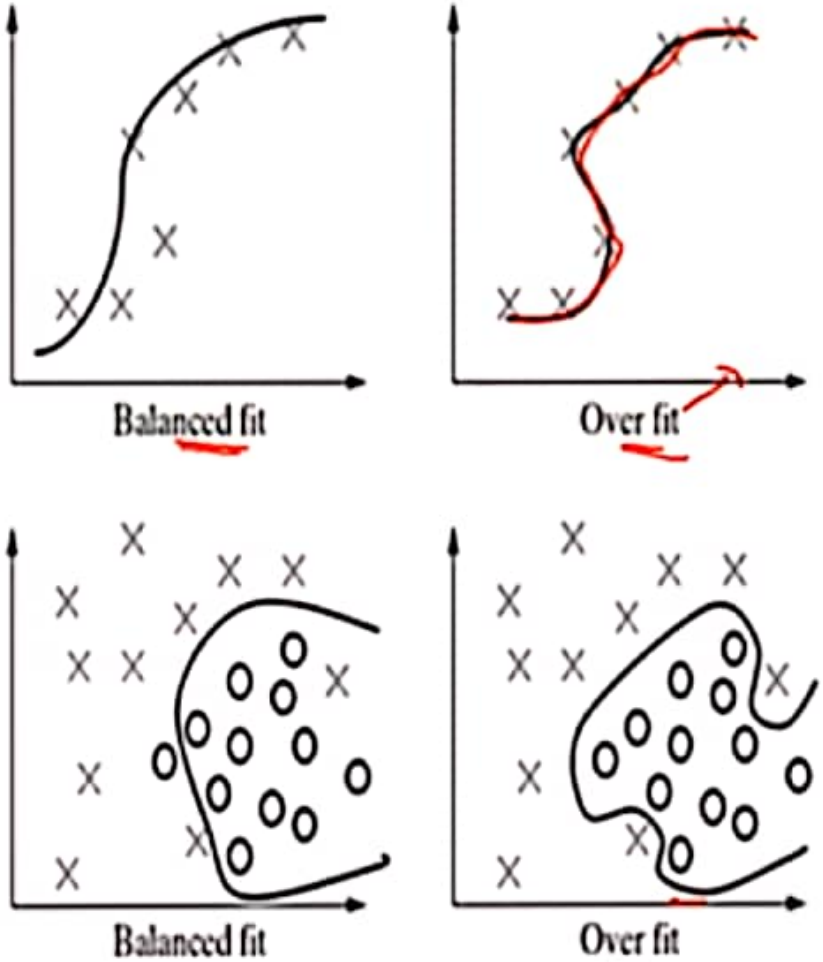
Underfitting

- Many times underfitting happens due to unavailability of sufficient training data.
- Underfitting results in both poor performance with training data as well as poor generalization to test data.
- Underfitting can be avoided by
- **1. using more training data**
- **2. reducing features by effective feature selection**



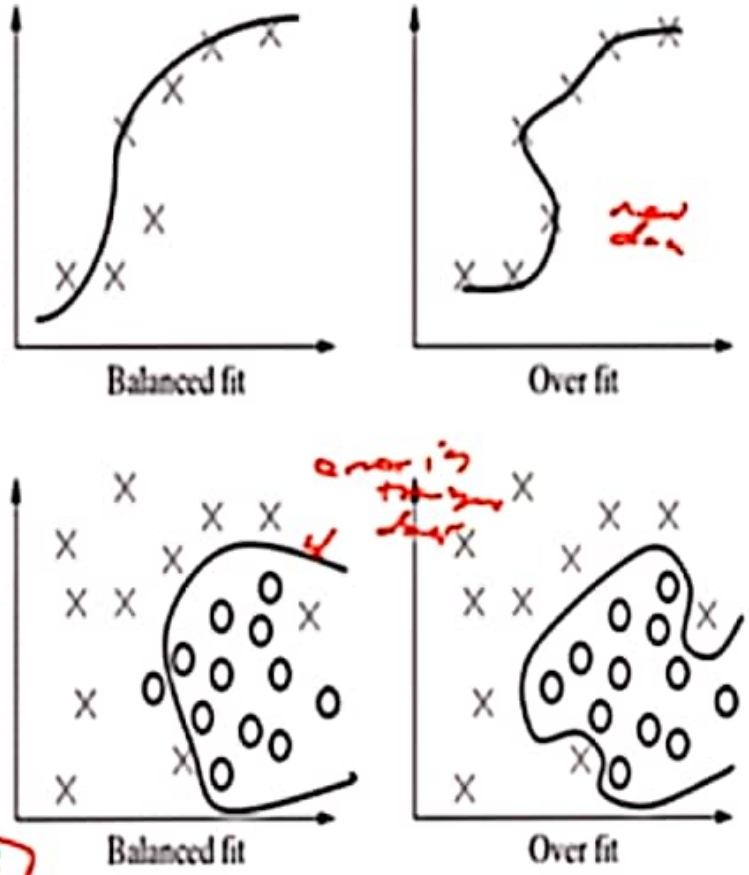
Overfitting

- Overfitting matches the training data too closely.
- If any specific deviation in the training data, like noise or outliers, gets embedded in the model, then the performance automatically reduces
- Overfitting, occur as a result of trying to fit an excessively complex model to closely match the training data.
- The target function, tries to make sure all training data points are correctly partitioned by the decision boundary.



Overfitting

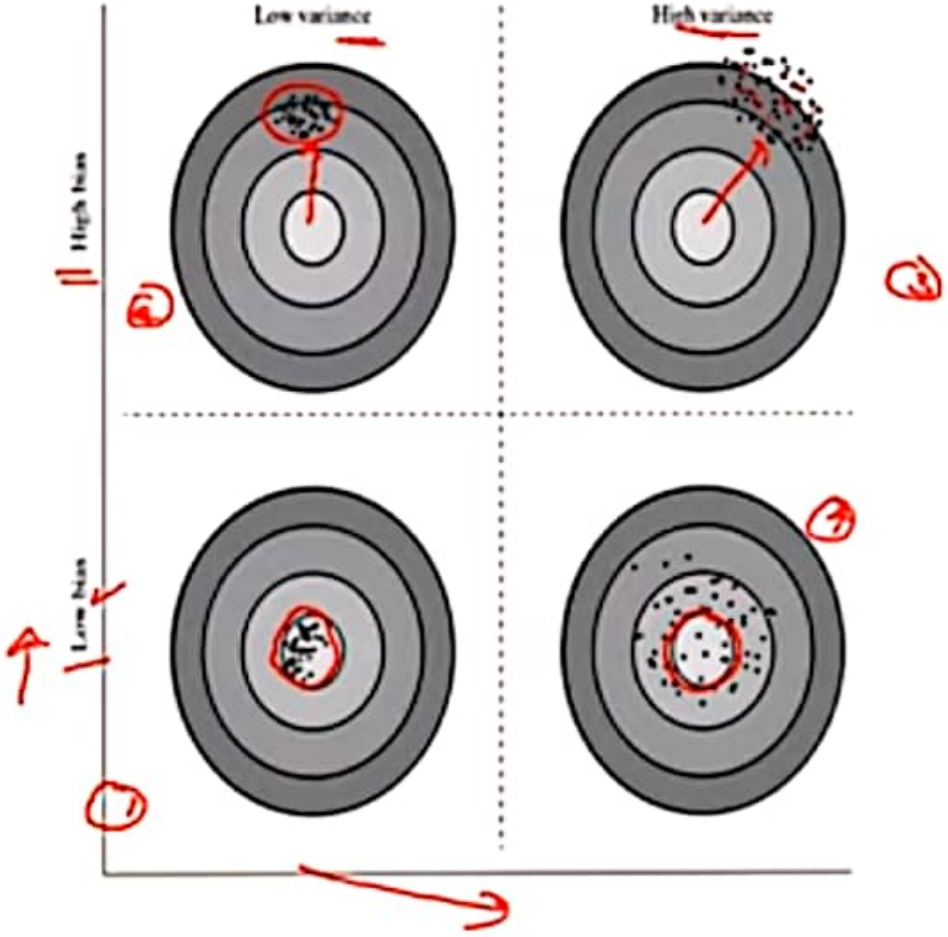
- this nature is not replicated in the unknown test data set.
- Hence, the target function results in wrong classification in the test data set..
- **Overfitting results in good performance with training data set, but poor generalization and hence poor performance with test data set.**
- Overfitting can be avoided by
  - 1. using re-sampling techniques like k-fold cross validation
  - 2. hold back of a validation data set
  - 3. remove the nodes which have little or no predictive power for the given machine learning problem.



Bias – variance trade-of

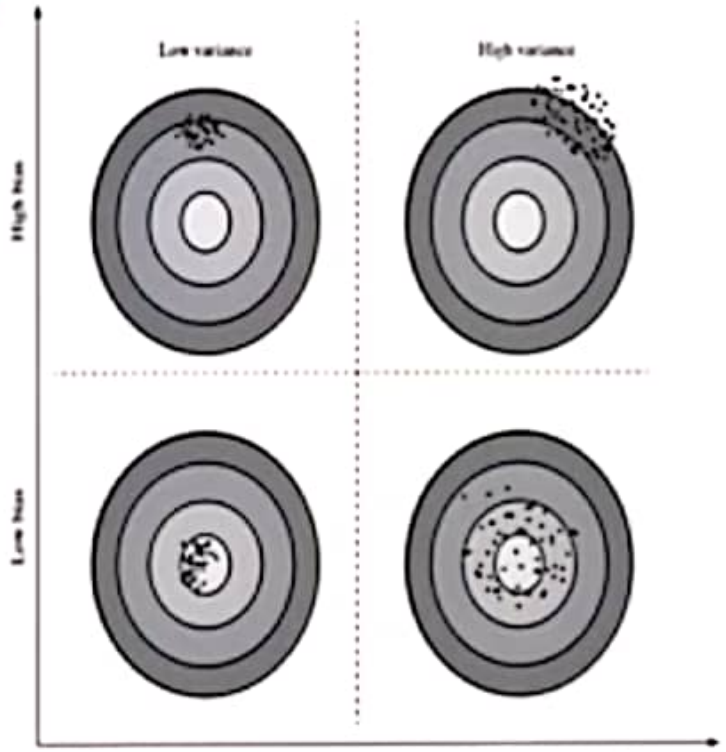
- In supervised learning, the class value assigned by the learning model built based on the training data may differ from the actual class value.
- This error in learning can be of two types –
  - errors due to 'bias' and
  - error due to 'variance'.

Bias-variance trade-off



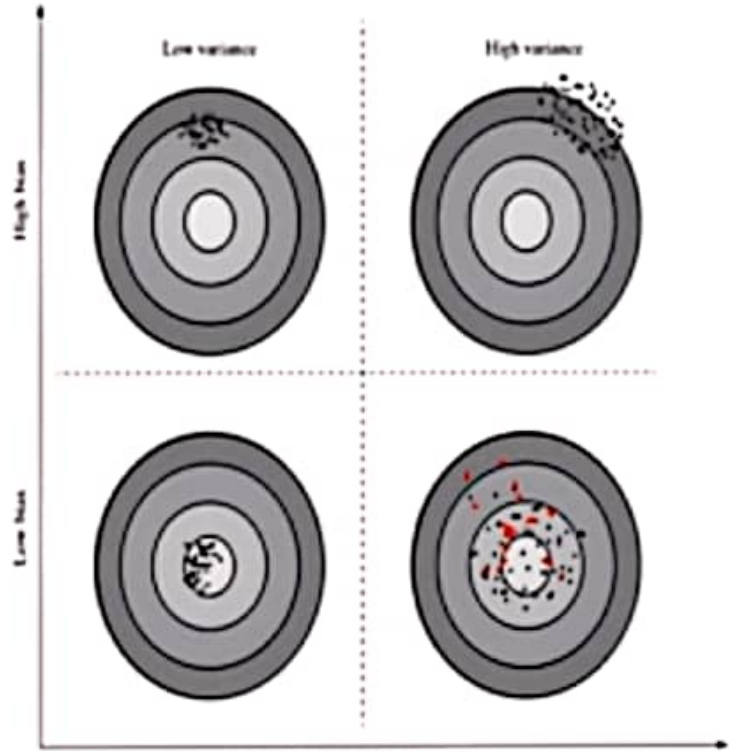
Errors due to 'Bias'

- Errors due to bias arise from simplifying assumptions made by the model to make the target function less complex or easier to learn.
- it is due to underfitting of the model.
- Underfitting results in high bias.



Errors due to 'Variance'

- Errors due to variance occur from difference in training data sets used to train the model.
- in case of overfitting, since the model closely matches the training data,
- even a small difference in training data gets magnified in the model.



Bias-variance trade-off...

- The best solution is to have a model with low bias as well as low variance.
- the goal of supervised machine learning is to achieve a balance between bias and variance.
- For example, in a supervised algorithm k-Nearest Neighbors or kNN, the user configurable parameter 'k' can be used to do a trade-off between bias and variance.
- When the value of 'k' is decreased, the model becomes simpler to fit and bias increases.
- When the value of 'k' is increased, the variance increases.

Machine Learning

Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

EVALUATING PERFORMANCE OF A MODEL – Part-1

- Supervised learning – classification
  - F-measure
  - Receiver operating characteristic (ROC) curves
  - The Area Under Curve (AUC)

Supervised learning – Classification

- In supervised learning, one major task is classification.
- The classification model is to **assign class label to the target feature based on the value of the predictor features.**
- For example, in a cricket match, the problem of predicting the win/loss, the classifier will assign a class value win/loss to target feature based on the values of other features like
  - The whether
  - the team won the toss,
  - number of spinners in the team,
  - number of wins the team had in the tournament, etc.

Supervised learning – Classification…

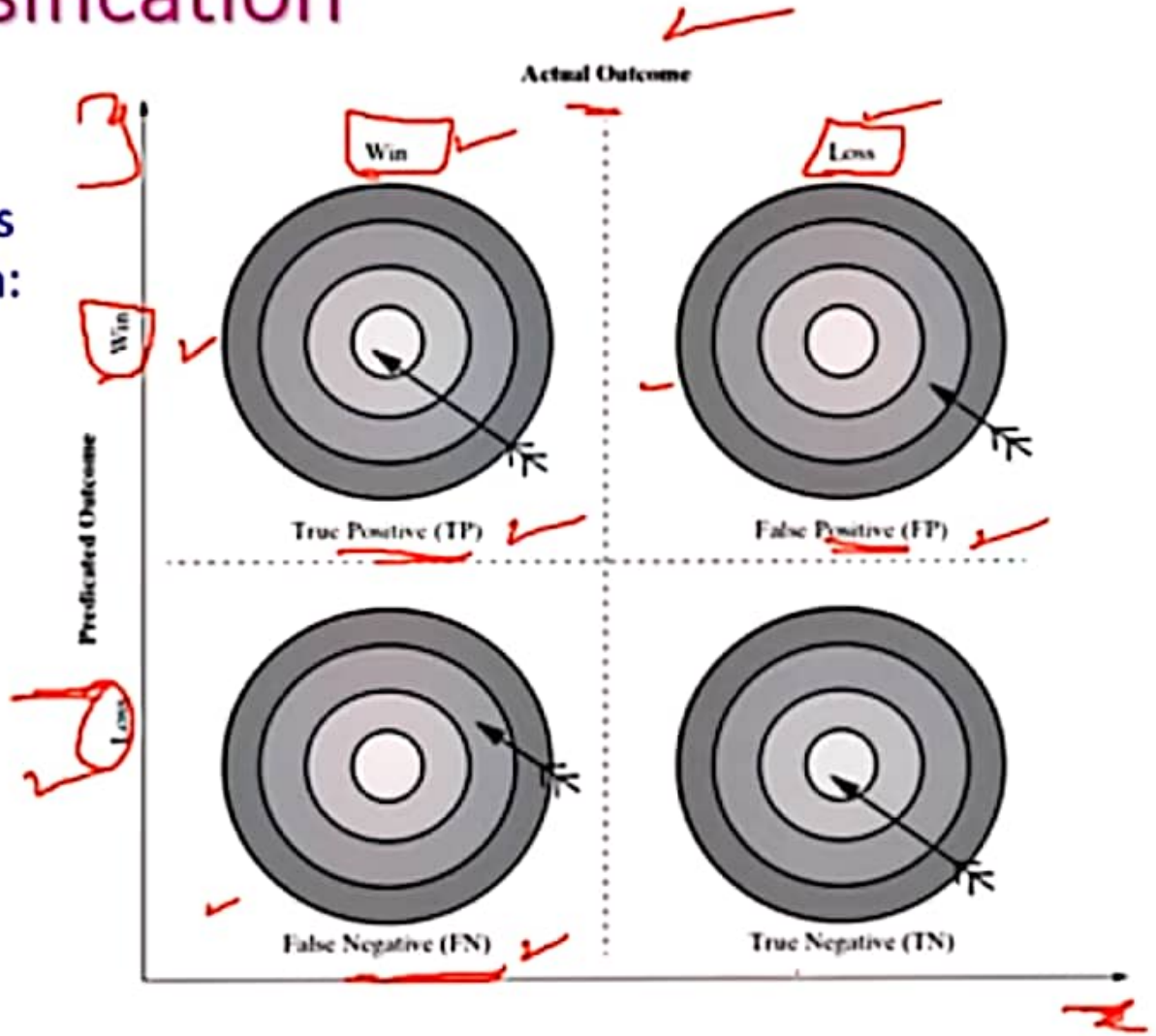
- To evaluate the performance of the model, the number of correct classifications or predictions made by the model has to be recorded.
- A classification is said to be correct if, the given problem, it has been predicted by the model that the team will win and it has actually won.

Supervised learning – Classification…

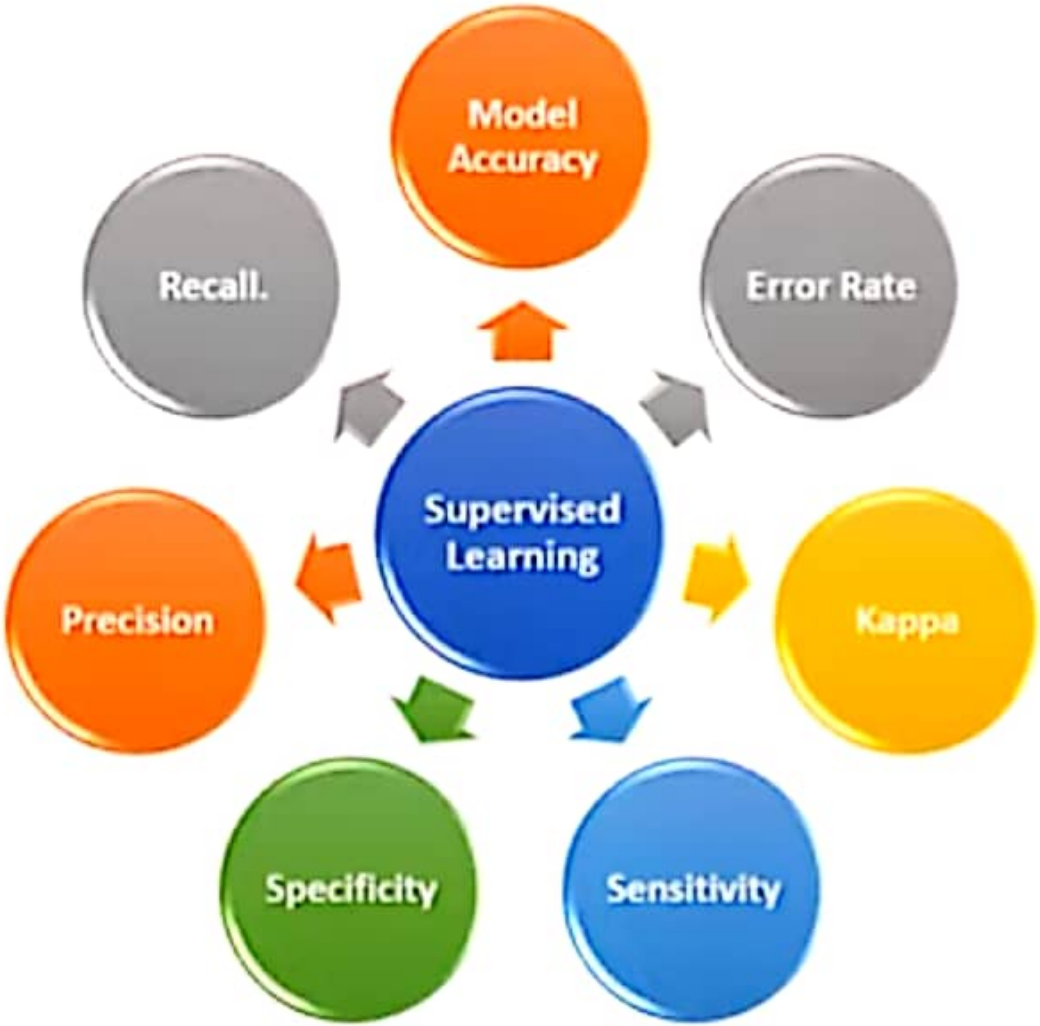
- Based on the number of correct and incorrect classifications or predictions made by a model, the accuracy of the model is calculated.
- If 99 out of 100 times the model has classified correctly,
- e.g. if in 99 out of 100 games what the model has predicted is same as what the outcome has been, then the model accuracy is said to be 99%.

Details of model classification

- There are four possibilities with regards to the cricket match win/loss prediction:
  - 1. the model predicted win and the team won – True Positive
  - 2. the model predicted win and the team lost – False Positive
  - 3. the model predicted loss and the team won – False Negative
  - 4. the model predicted loss and the team lost – True Negative



Performance Measures Of A Supervised Learning



Confusion Matrix

- A matrix containing correct and incorrect predictions in the form of TPs, FPs, FNs and TNs is known as confusion matrix.
- The win/loss prediction of cricket match has two classes of interest – win and loss.
- For that reason it will generate a 2 × 2 confusion matrix.
- For a classification problem involving three classes, the confusion matrix would be 3 × 3, etc.
- assume the confusion matrix of the win/loss prediction of cricket match problem to be as below:

	ACTUAL WIN	ACTUAL LOSS
Predicted Win	85	4
Predicted Loss	2	9

Model Accuracy

- The model accuracy is the percentage of correct classification, given by

$$\text{Model accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

- In context of the confusion matrix, total count of TPs = 85, count of FPs = 4, count of FNs = 2 and count of TNs = 9.

$$\therefore \text{Model accuracy} = \frac{TP + TN}{TP + FP + FN + TN} = \frac{85 + 9}{85 + 4 + 2 + 9} = \frac{94}{100} = 94\%$$

	ACTUAL WIN	ACTUAL LOSS
Predicted Win	85 TP	4 FP
Predicted Loss	2 FN	9 TN

2. Error Rate

- The percentage of misclassifications is indicated using error rate which is measured as

$$\text{Error rate} = \frac{FP + FN}{TP + FP + FN + TN}$$

- In context of the above confusion matrix,

$$\text{Error rate} = \frac{FP + FN}{TP + FP + FN + TN} = \frac{4 + 2}{85 + 4 + 2 + 9} = \frac{6}{100} = 6\%$$

$$= 1 - \text{Model accuracy}$$

$1 - 0.94 = 0.06$ 6 %

3. Kappa

- Kappa value of a model indicates the adjusted model accuracy.
- It is calculated using the formula below:

$$\text{Kappa value (k)} = \frac{P(a) - P(p_r)}{1 - P(p_r)}$$

P(a) = Proportion of observed agreement between actual and predicted in overall data set

$$P(a) = \frac{TP + TN}{TP + FP + FN + TN}$$

$P(p_r)$ = Proportion of expected agreement between actual and predicted data both in case of class of interest as well as the other classes

$$P(p_r) = \frac{TP + FP}{TP + FP + FN + TN} \times \frac{TP + FN}{TP + FP + FN + TN} + \frac{FN + TN}{TP + FP + FN + TN} \times \frac{FP + TN}{TP + FP + FN + TN}$$

Kappa...

- In context of the above confusion matrix, total count of TPs = 85, count of FPs = 4, count of FNs = 2 and count of TNs = 9.

$$\therefore P(a) = \frac{TP + TN}{TP + FP + FN + TN} = \frac{85 + 9}{85 + 4 + 2 + 9} = \frac{94}{100} = 0.94$$

$$P(p_r) = \frac{85 + 4}{85 + 4 + 2 + 9} \times \frac{85 + 2}{85 + 4 + 2 + 9} + \frac{2 + 9}{85 + 4 + 2 + 9} \times \frac{4 + 9}{85 + 4 + 2 + 9}$$

$$= \frac{89}{100} \times \frac{87}{100} + \frac{11}{100} \times \frac{13}{100} = 0.89 \times 0.87 + 0.11 \times 0.13 = 0.7886$$

$$\therefore k = \frac{0.94 - 0.7886}{1 - 0.7886} = 0.7162$$

	ACTUAL WIN	ACTUAL LOSS
Predicted Win	85	4
Predicted Loss	2	9

4. Sensitivity

- The sensitivity of a model measures the proportion of TP examples or positive cases which were correctly classified.
- It is measured as $$\text{Sensitivity} = \frac{TP}{TP + FN}$$
- The confusion matrix for the cricket match win prediction problem,

$$\text{Sensitivity} = \frac{TP}{TP + FN} = \frac{85}{85 + 2} = \frac{85}{87} = 97.7\%$$

Specificity

- Specificity is also another good measure to indicate a good balance of a model being excessively conservative or excessively aggressive.
- Specificity of a model measures the proportion of negative examples which have been correctly classified.

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{9}{9 + 4} = \frac{9}{13} = 69.2\%$$

Precision

- The precision gives the proportion of positive predictions which are truly positive, recall gives the proportion of TP cases over all actually positive cases.

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

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{85}{85 + 4} = \frac{85}{89} = 95.5\%$$

Recall

- Recall indicates the proportion of correct prediction of positives to the total number of positives.
- In case of win/loss prediction of cricket, recall resembles what proportion of the total wins were predicted correctly

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

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{85}{85 + 2} = \frac{85}{87} = 97.7\%$$

F-measure

- F-measure is another measure of model performance which combines the precision and recall.
- It takes the harmonic mean of precision and recall as calculated as

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

- In context of the above confusion matrix for the cricket match win prediction problem

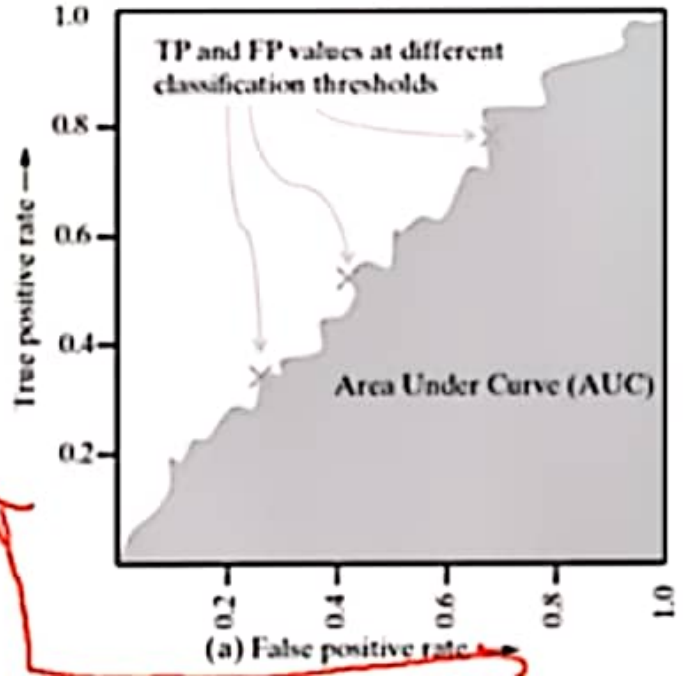
$$F\text{-measure} = \frac{2 \times 0.955 \times 0.977}{0.955 + 0.977} = \frac{1.866}{1.932} = 96.6\%$$

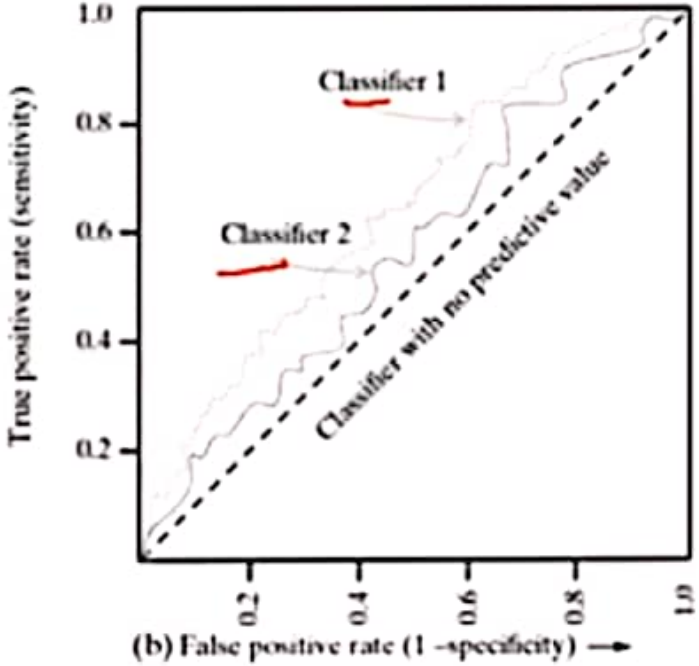
F-measure...

- F-score is a combination of multiple measures into one,
- It is used to compare the performance of different models
- the calculation is based on that precision and recall have equal weight (which may not always be true in reality).
- In certain problems, for eg. **the disease prediction problems,** precision may be given far more weightage.
- In that case, different weightages may be assigned to precision and recall.
- However, there may be a serious problem regarding what value to be adopted for each and what is the basis for the specific value adopted.

Other methods

- Visualization is an easier and more effective way to understand the model Performance
- 1. Receiver operating characteristic (ROC) curves
- 2. Area Under Curve (AUC)
- It also helps in comparing the efficiency of two models.





Receiver operating characteristic (ROC) curves

- **Receiver Operating Characteristic (ROC)** curve helps in visualizing the performance of a classification model.
- It shows the efficiency of a model in the detection of true positives while avoiding the occurrence of false positives.
- To refresh our memory, true positives the model has correctly classified data instances as the class of interest.
- For example, the model has correctly classified the tumours as malignant (serious problem), in case of a **tumour malignancy prediction problem.**.
- On the other hand, FPs are those cases where the model incorrectly classified data instances as the class of interest.

Receiver operating characteristic (ROC) curves...

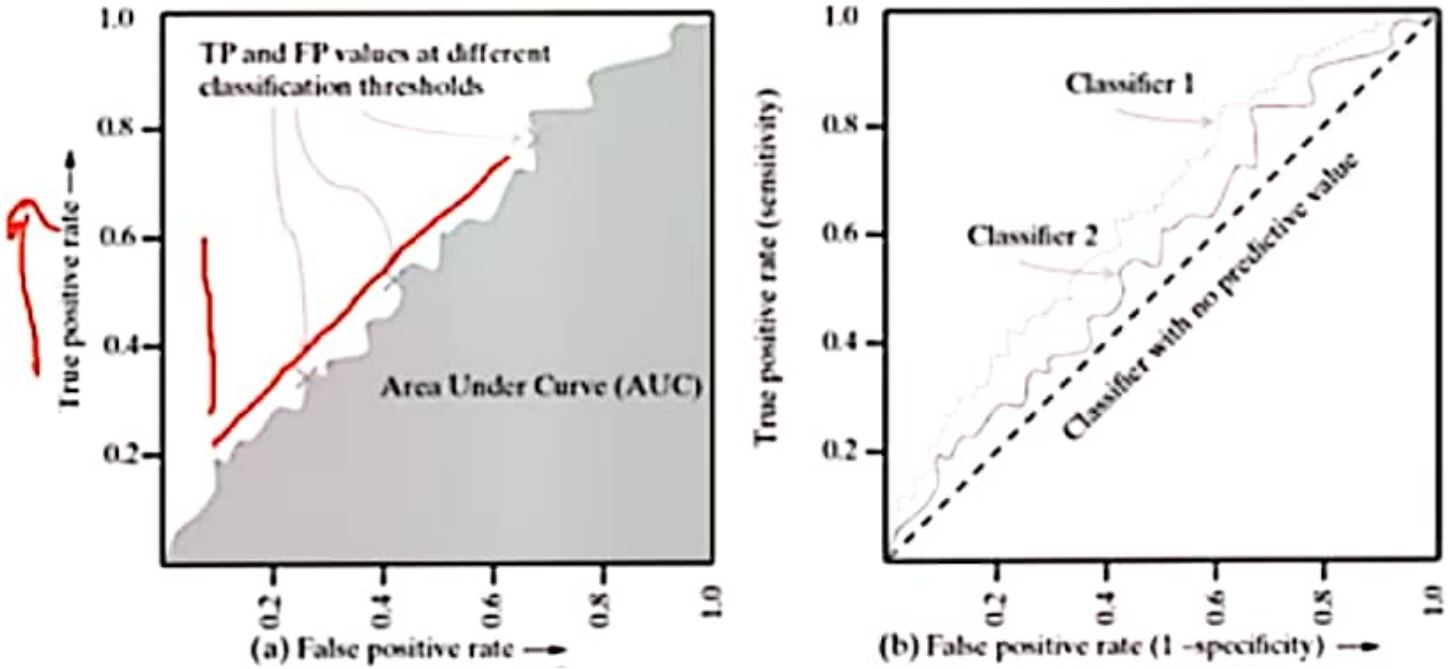
- example , the model has incorrectly classified the tumours as malignant,
- i.e. tumours which are actually benign have been classified as malignant.

$$\text{True Positive Rate TPR} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate FPR} = \frac{FP}{FP + TN}$$

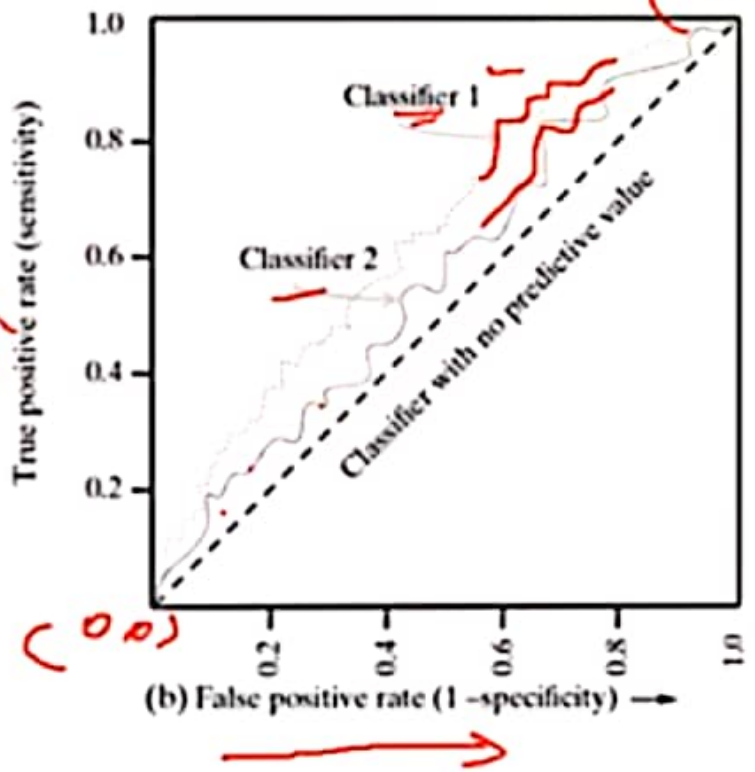
Receiver operating characteristic (ROC) curves...

- In the ROC curve, the FP rate is plotted (in the horizontal axis) against true positive rate (in the vertical axis) at different classification thresholds.
- If we assume a lower value of classification threshold, the model classifies more items as positive.
- Hence, the values of both False Positives and True Positives increase.
- The fig. ROC curve



The Area Under Curve (AUC)

- The area under curve (AUC) value, is the area of the two-dimensional space under the curve extending from (0, 0) to (1, 1),
- where each point on the curve gives a set of true and false positive values at a specific classification threshold.
- This curve gives an indication of the predictive quality of a model.
- AUC value ranges from 0 to 1, with an AUC of less than 0.5 indicating that the classifier has no predictive ability.
- Figure shows the curves of two classifiers – classifier 1 and classifier 2.
- the AUC of classifier 1 is more than the AUC of classifier 2.
- Hence, the inference that classifier 1 is better than classifier 2.



AUC

- A quick indicative interpretation of the predictive values from 0.5 to 1.0 is given below:
- 0.5 – 0.6 → Almost no predictive ability
- 0.6 – 0.7 → Weak predictive ability
- 0.7 – 0.8 → Fair predictive ability
- 0.8 – 0.9 → Good predictive ability
- 0.9 – 1.0 → Excellent predictive ability

Thank You

- **UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering**
- **EVALUATING PERFORMANCE OF A MODEL-Part-1**
  - Supervised learning – classification

- Next…
- Supervised learning – regression
- Unsupervised learning - clustering

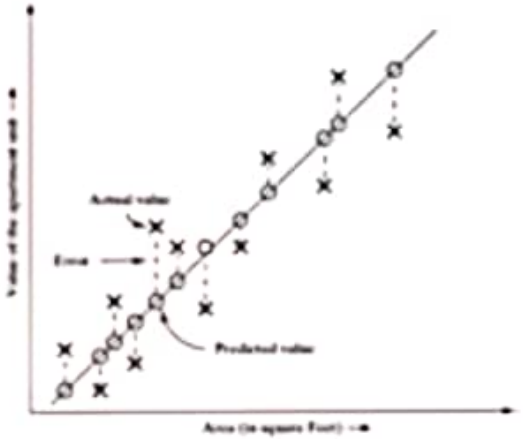
Machine Learning

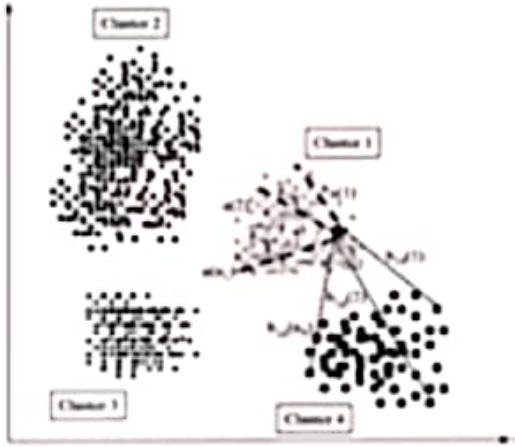
Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

EVALUATING PERFORMANCE OF A MODEL-Part-2

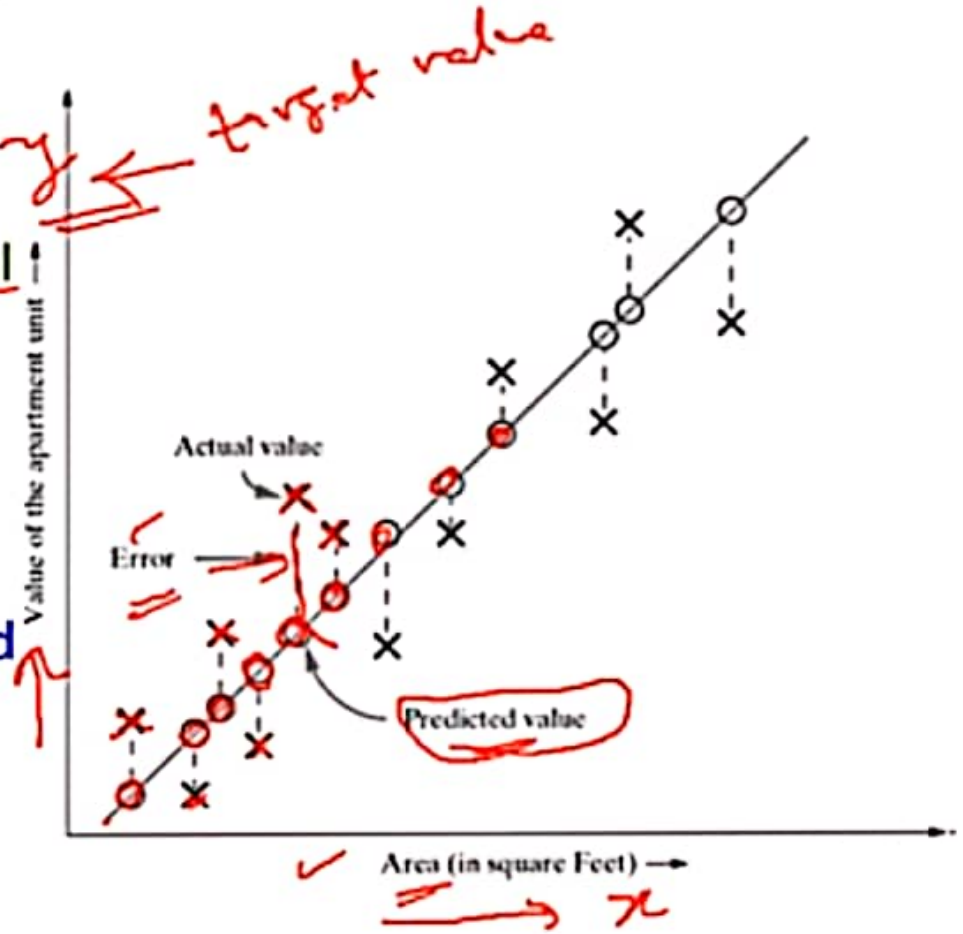
- Supervised learning – regression
  - R-squared Measure
- Unsupervised learning - clustering
  - Internal Evaluation
    - Silhouette width calculation
  - External Evaluation





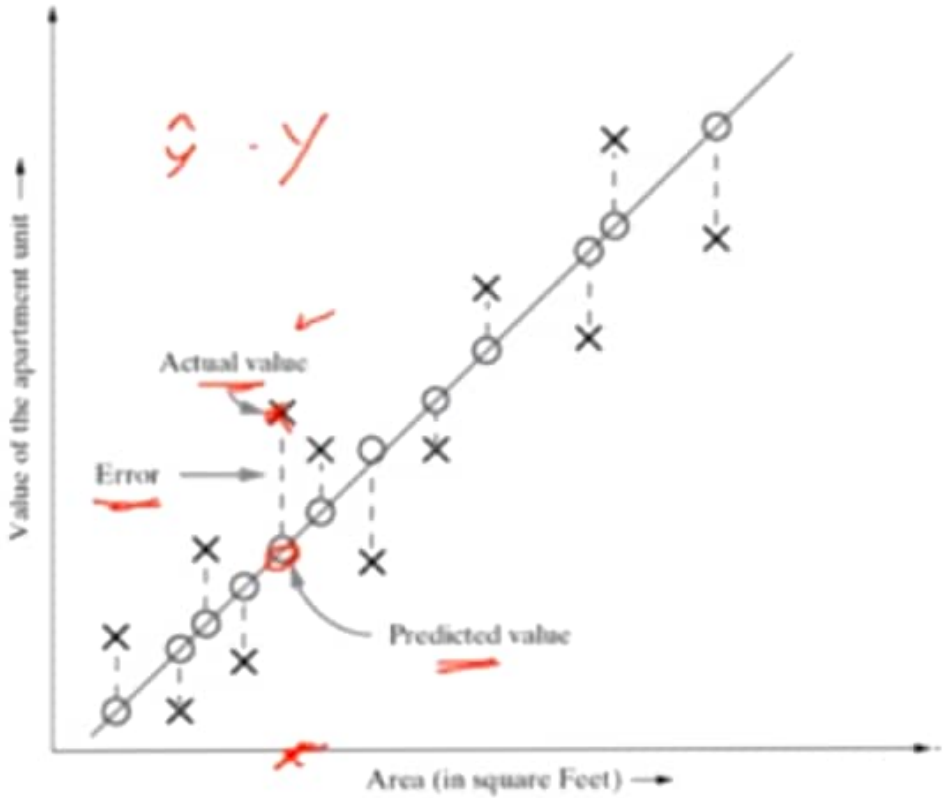
Supervised learning – Regression

- A regression model ensures the difference between predicted and actual values is low can be considered as a good model.
- Figure represents a very simple problem of real estate value prediction solved using linear regression model.
- 'area' is the predictor variable and
- 'value' is the target variable,
- the linear regression model can be represented in the form:



Supervised learning – Regression...

- For a certain value of x, say $\hat{x}$, the value of y is predicted as $\hat{y}$ whereas the actual value of y is Y (say).
- The distance between the actual value and the predicted value, i.e. $\hat{y}$ is known as residual.
- The regression model can be considered to be fitted well, if the difference between actual and predicted value is less,
- i.e. the residual value is less.



Regression - R-squared Measure

R-squared is a good measure to evaluate the model fitness.

- The R-squared value lies between 0 to 1 (0%–100%) with a larger value representing a better fit.
- It is calculated as: $$R^2 = \frac{SST - SSE}{SST}$$
- Sum of Squares Total (SST) = squared differences of each observation from
- the overall mean = $\sum_{i=1}^{n} (y_i - \bar{y})^2$
- where $\bar{y}$ is the mean.
- Sum of Squared Errors (SSE) (of prediction) = sum of
- the squared residuals = $\sum_{i=1}^{n} (Y_i - \hat{y})^2$
- where $\hat{y}$ is the predicted value of y_i and Y_i is the actual value of y_i

Unsupervised learning – Clustering

- A clustering algorithm is successful if the clusters identified using the algorithm is able to achieve the right results in the overall problem domain.
- For example, if clustering is applied for identifying customer segments for a marketing campaign of a new product launch,
- the clustering can be considered successful only if the marketing campaign ends with a success,
- i.e. it is able to create the right brand recognition resulting in steady revenue from new product sales.

Unsupervised learning – Clustering

- Two challenges of clustering:

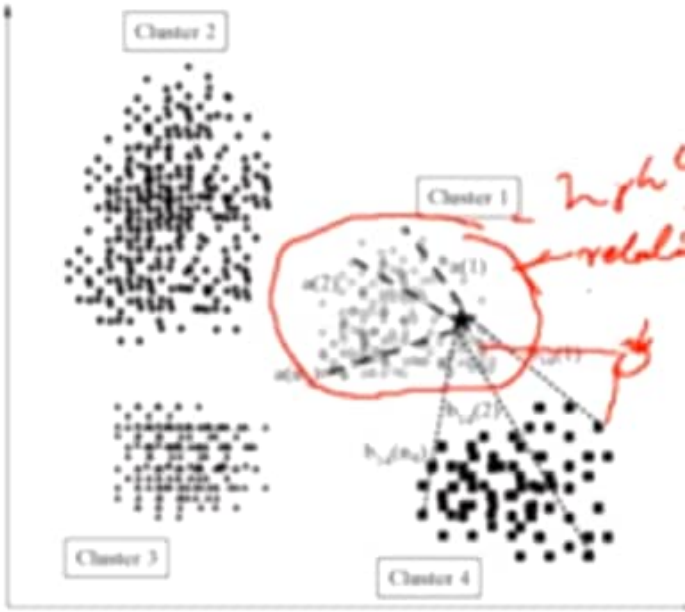
1. It is generally not known how many clusters can be formulated from a particular data set. It is completely open-ended in most cases and provided as a user input to a clustering algorithm.
2. Even if the number of clusters is given, the same number of clusters can be formed with different groups of data instances.

Unsupervised learning – Clustering...

- The popular approaches which are adopted for cluster quality evaluation.
- 1. Internal Evaluation
- 2. External Evaluation

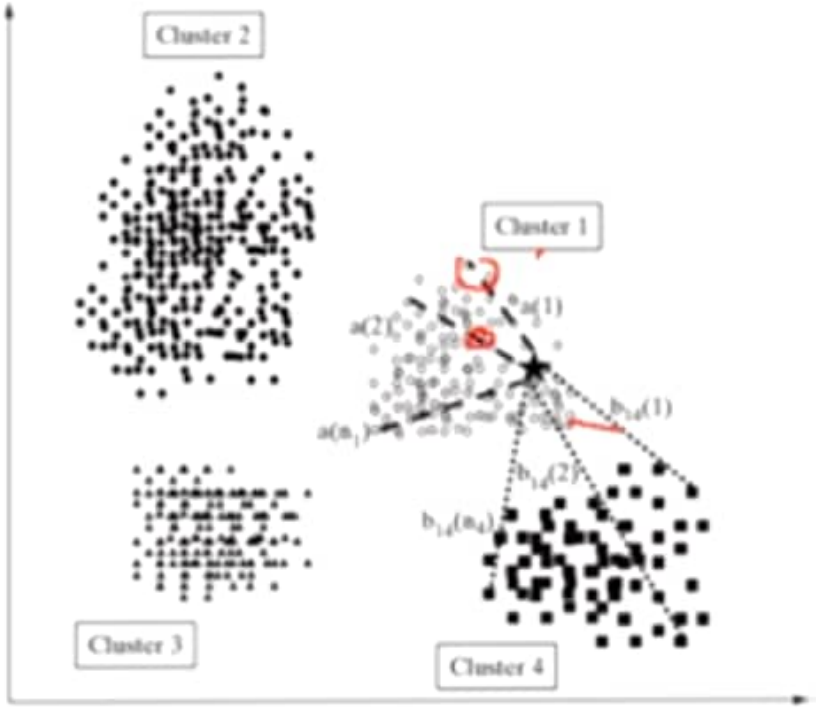
1. Internal evaluation

- The cluster is assessed based on the underlying data that was clustered.
- The internal evaluation methods generally measure cluster quality based on homogeneity of data belonging to the same cluster and heterogeneity of data belonging to different clusters.
- The homogeneity/heterogeneity is decided by some similarity measure.



Internal Evaluation - Silhouette Coefficient

- The silhouette coefficient, which is one of the most popular internal evaluation methods, uses distance (Euclidean or Manhattan distances most commonly used) between data elements as a similarity measure.
- The value of silhouette width ranges between −1 and +1, with a high value indicating high intra-cluster homogeneity and inter-cluster heterogeneity.

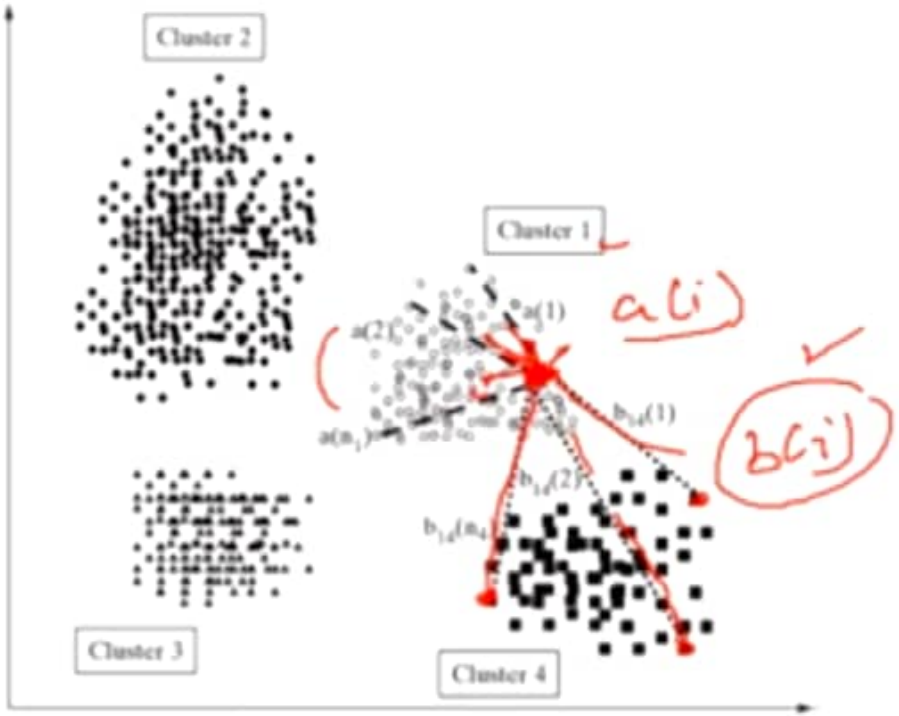


Internal evaluation...

- For a data set clustered into 'k' clusters, silhouette width is calculated as:

$$\text{Silhouette width} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

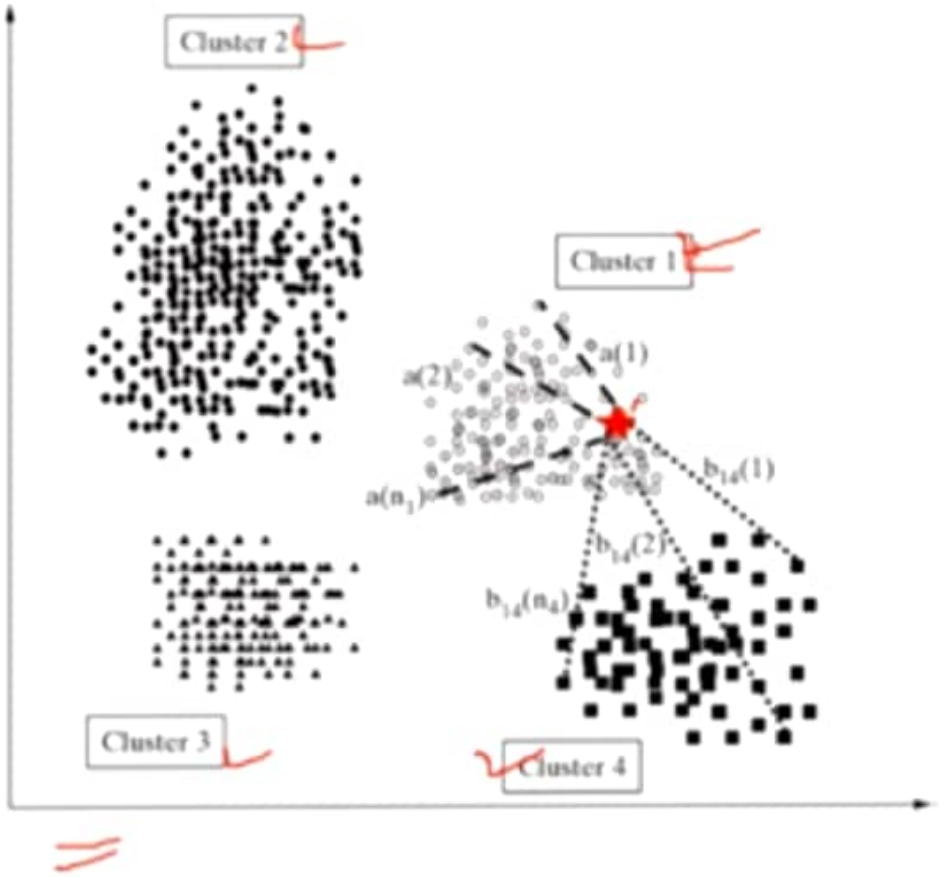
- a(i) is the average distance between the i th data instance and all other data instances belonging to the same cluster and
- b(i) is the lowest average distance between the i- the data instance and data instances of all other clusters.



Silhouette width calculation

- There are four clusters namely cluster 1, 2, 3, and 4.
- Let's consider an arbitrary data element 'i' in cluster 1, resembled by the asterisk.
- a(i) is the average of the distances a_{i1}, a_{i2}, …, a_{in} of the different data elements from the i^{th} data element in cluster 1,
- assuming there are n_1 data elements in cluster 1. Mathematically,

$$a(i) = \frac{a_{i1} + a_{i2} + \ldots + a_{in_1}}{n_1}$$



Silhouette width calculation...

- let's calculate the distance of an arbitrary data b element 'i' in cluster 1 with the different data elements from another cluster, say cluster 4 and take an average of all those distances.
- Hence,

$$b_{14}\,(\text{average}) = \frac{b_{14}(1) + b_{14}(2) + \ldots + b_{14}(n_4)}{(n_4)}$$

- where n is the total number of elements in cluster 4.
- In the same way, we can calculate the values of b_{12} (average) and b_{13} (average).
- b(i) is the minimum of all these values.
- Hence, b(i) = minimum [b_{12} (average), b_{13} (average), b_{14} (average)]

2. External evaluation

- In this approach, class label is known for the data set subjected to clustering.
- the known class labels are not a part of the data used in clustering.
- The cluster algorithm is assessed based on how close the results are compared to those known class labels.
- For example, purity is one of the most popular measures of cluster algorithms – evaluates the extent to which clusters contain a single class.

2. External evaluation

- For a data set having 'n' data instances and
- 'c' known class labels which generates
- 'k' clusters, purity is measured as:

$$\text{Purity} = \frac{1}{n}\sum_{k}\max(k \cap c)$$

Thank You

- **UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering**
- **EVALUATING PERFORMANCE OF A MODEL-Part-2**
- Supervised learning – regression
  - R-squared Measure
- Unsupervised learning - clustering
  - Internal Evaluation
    - Silhouette width calculation
  - External Evaluation

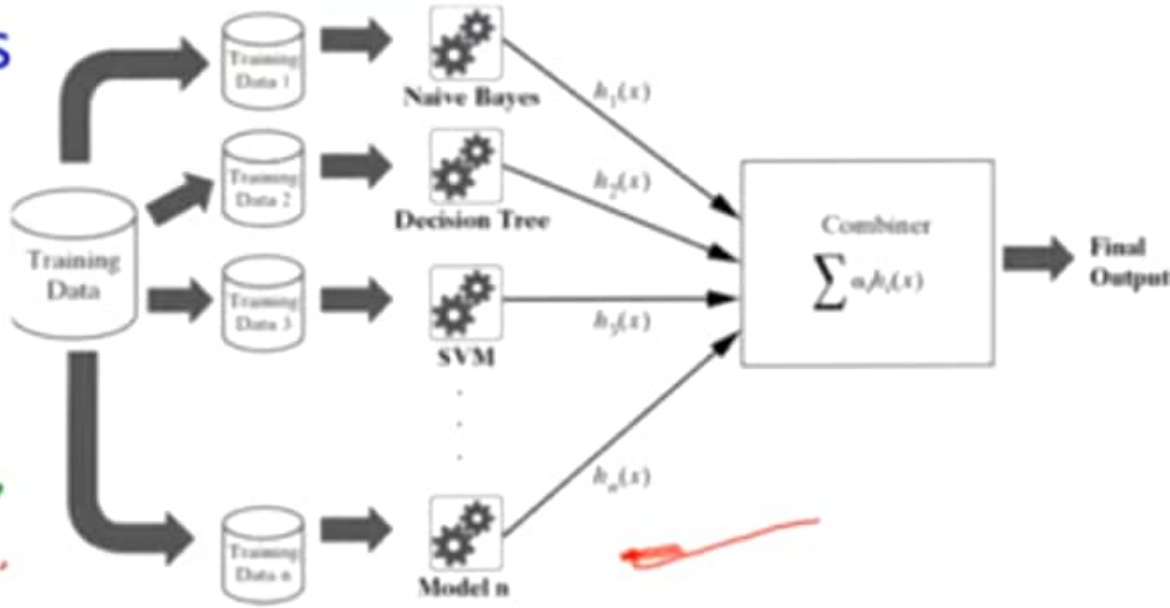
Machine Learning

Subject Code: 20A05602T

UNIT 2 – Modelling and Evaluation & Basics of Feature Engineering

IMPROVING PERFORMANCE OF A MODEL

- Model Selection Requirements
- Tuning model parameter,
  - kNN
- Combining several models,
  - Ensemble,
  - Bootstrap Aggregating/Bagging,
  - Adaptive boosting or AdaBoost



Model Selection Requirements

- The model selection is done based on several aspects:
- 1. Type of learning the task i.e. supervised or unsupervised
- 2. Type of the data, i.e. categorical or numeric
- 3. The problem domain
- 4. The experience in working with different models to solve problems

Tuning model parameter

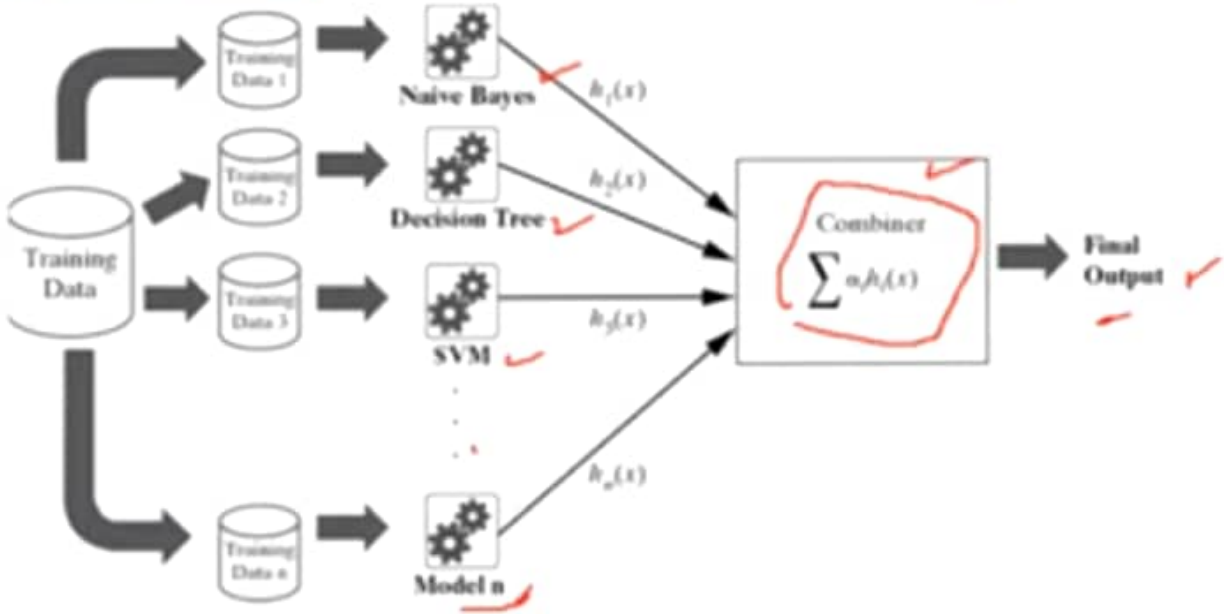
- Model parameter tuning is the process of adjusting the model fitting options, is an effective way to improve model performance
- Most machine learning models have at least one parameter which can be tuned.
- **The classification model k-Nearest Neighbour (kNN):**
- using different values of 'k' or the number of nearest neighbours to be considered, the model can be tuned.
- **The neural networks model:**
- The number of hidden layers can be adjusted to tune the performance in neural networks model.

Combining several models

- As an alternate approach of increasing the performance of one model, several models may be combined together.
- The models in such combination are complimentary to each other,
- i.e. one model may learn one type data sets well while struggle with another type of data set.
- Another model may perform well with the data set which the first one struggled with.

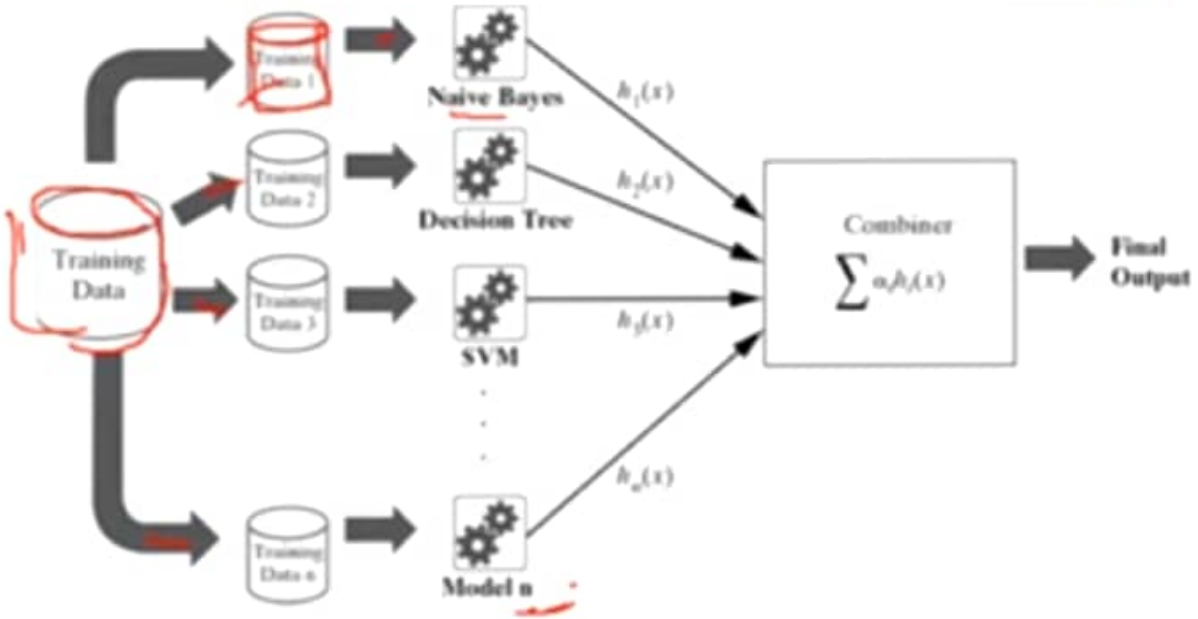
Ensemble

- This approach of combining different models with diverse strengths is known as ensemble (figure).
- Ensemble helps in averaging out biases of the different underlying models and also reducing the variance.
- Ensemble methods combine weaker learners to create stronger ones.



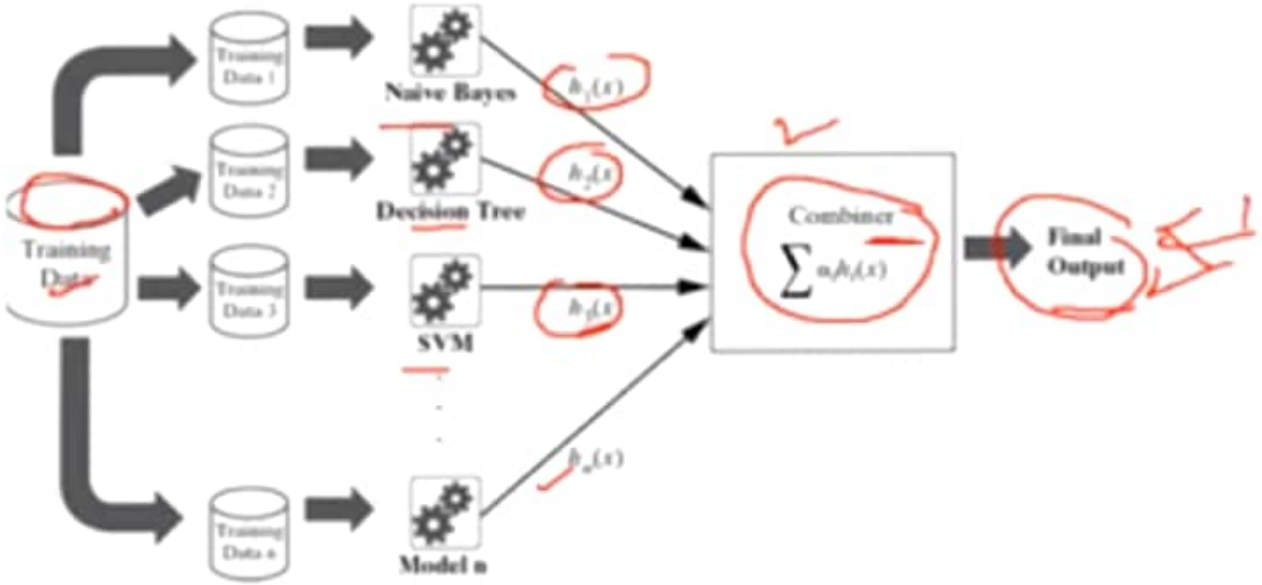
Ensemble...

- Following are the typical steps in ensemble process:
- Build a number of models based on the training data
- For diversifying the models generated, the training data subset can be varied using the allocation function.
- Sampling techniques like bootstrapping may be used to generate unique training data sets.



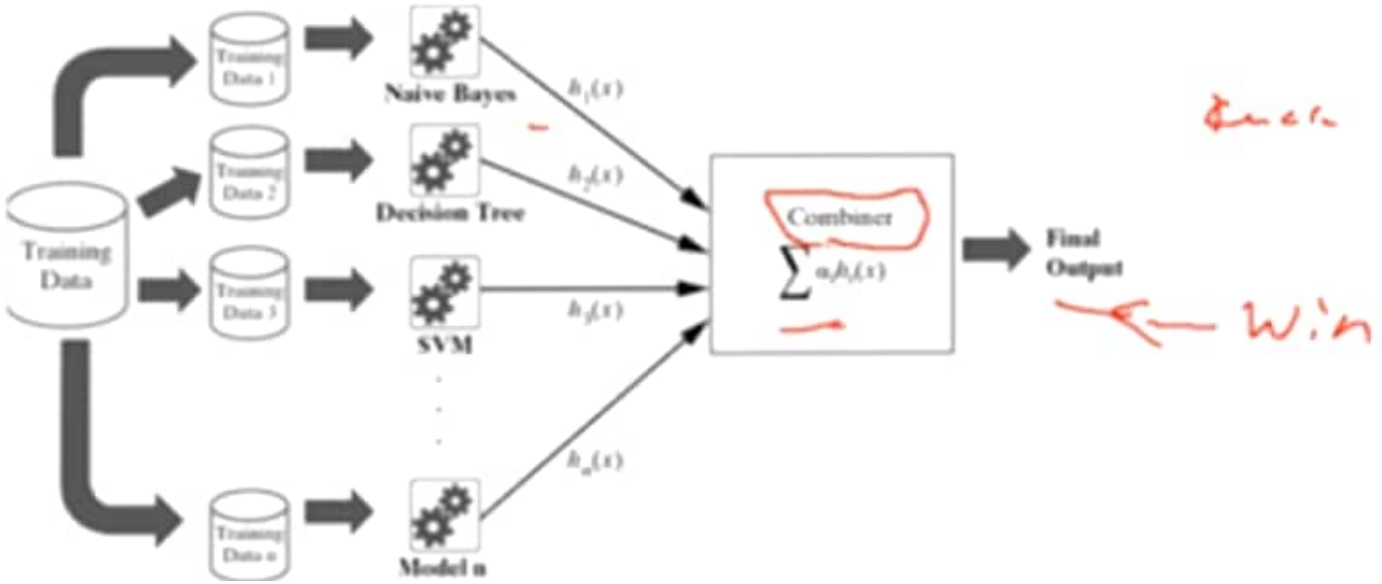
Steps in Ensemble Process...

- Alternatively, the same training data may be used but the models combined are quite varying, e.g, SVM, neural network, kNN, etc.
- The outputs from the different models are combined using a combination function.



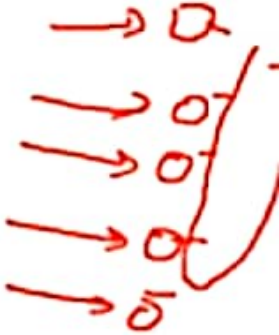
Steps in Ensemble Process...

- A very simple strategy of combining, say in case of a prediction task using ensemble, can be majority voting of the different models combined.
- For example, 3 out of 5 classes predict 'win' and 2 predict 'loss' – then the final outcome of the ensemble using majority vote would be a 'win'.



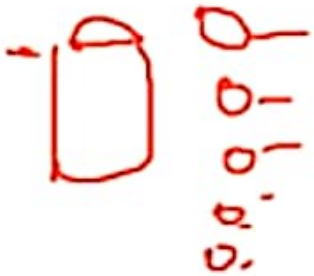
Bootstrap Aggregating or Bagging

- One of the earliest and most popular ensemble models is bootstrap aggregating or bagging.
- Bagging uses bootstrap sampling method to generate multiple training data sets.
- These training data sets are used to generate (or train) a set of models using the same learning algorithm.
- Then the outcomes of the models are combined by majority voting (classification) or by average (regression).
- Bagging suitable for unstable learners like a decision tree, in which a slight change in data can impact the outcome of a model significantly.



Adaptive boosting or AdaBoost

- Just like bagging, boosting is another key ensemble based technique.
- **The weaker learning models are trained on resampled data and the outcomes are combined using a weighted voting approach based on the performance of different models.**
- Adaptive boosting or AdaBoost is a special variant of boosting algorithm.
- It is based on the idea of generating weak learners and slowly learning.
- Random forest is another ensemble-based technique. It is an ensemble of decision trees – hence the name random forest to indicate a forest of decision trees.



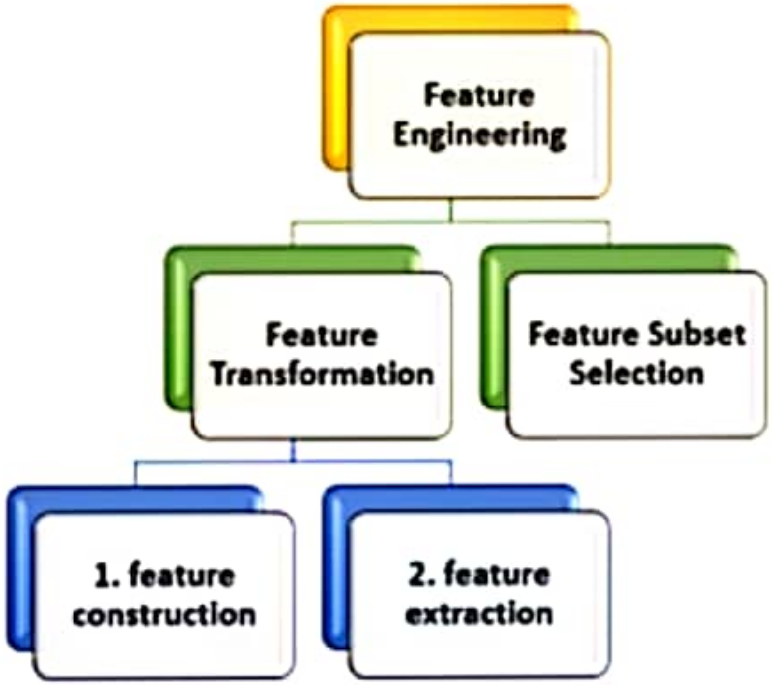
Machine Learning

Subject Code: 20A05602T

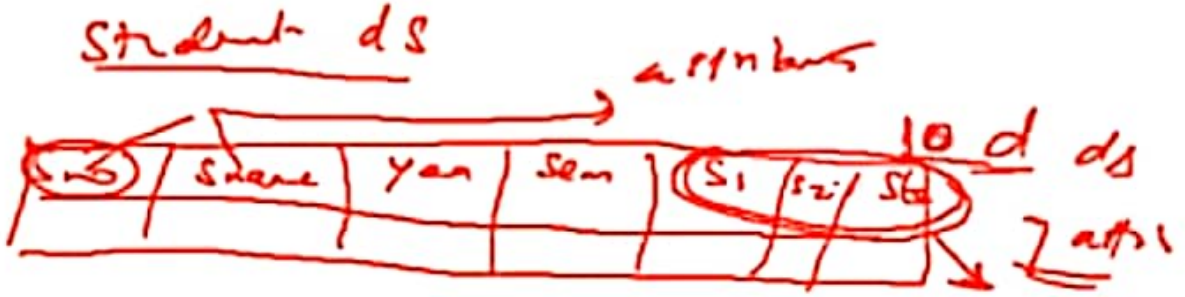
UNIT 2 –Basics of Feature Engineering

- Introduction,
- What is a feature?
- What is feature engineering?

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor
6.8	3.2	5.9	2.3	Virginica
5.5	2.5	4	1.3	Versicolor
5.1	3.5	1.4	0.2	Setosa
6.1	3	4.6	1.4	versicolor



What is a Feature?



- A feature is an attribute of a data set that is used in a machine learning process.
- The attributes which are meaningful to a machine learning problem are to be called as features.
- The features in a data set are also called its dimensions.
- So a data set having '*n*' features is called an *n*-dimensional data set.
- The selection of subset of features is an important sub part of machine learning.

Iris Data Set from UCI Repository

Iris Data Set

Download Data Folder, Data Set Description

Abstract: Famous database; from Fisher, 1936

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	5250015

Source:

Iris Data set

Source:

Creator:

R.A. Fisher

Donor:

Michael Marshall (MARSHALL%PLU '@' io.arc.nasa.gov)

Data Set Information:

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Predicted attribute: class of iris plant.

This is an exceedingly simple domain.

This data differs from the data presented in Fishers article (identified by Steve Chadwick, spchadwick '@' espeedaz.net). The 35th sample should be: 4.9,3.1,1.5,0.2,"Iris-setosa" where the error is in the fourth feature. The 38th sample: 4.9,3.6,1.4,0.1,"Iris-setosa" where the errors are in the second and third features.

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class
   -- Iris Setosa
   -- Iris Versicolour
   -- Iris Virginica

target attributes

class attributes

Iris Data set Features

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor
6.8	3.2	5.9	2.3	Virginica
5.5	2.5	4	1.3	Versicolor
5.1	3.5	1.4	0.2	Setosa
6.1	3	4.6	1.4	versicolor

- It has five attributes or features namely Sepal.Length,
- Sepal.Width, Petal.Length, Petal.Width and Species.
- Out of these, the feature 'Species' represent the class variable and the remaining features are the predictor variables.
- It is a five-dimensional data set.

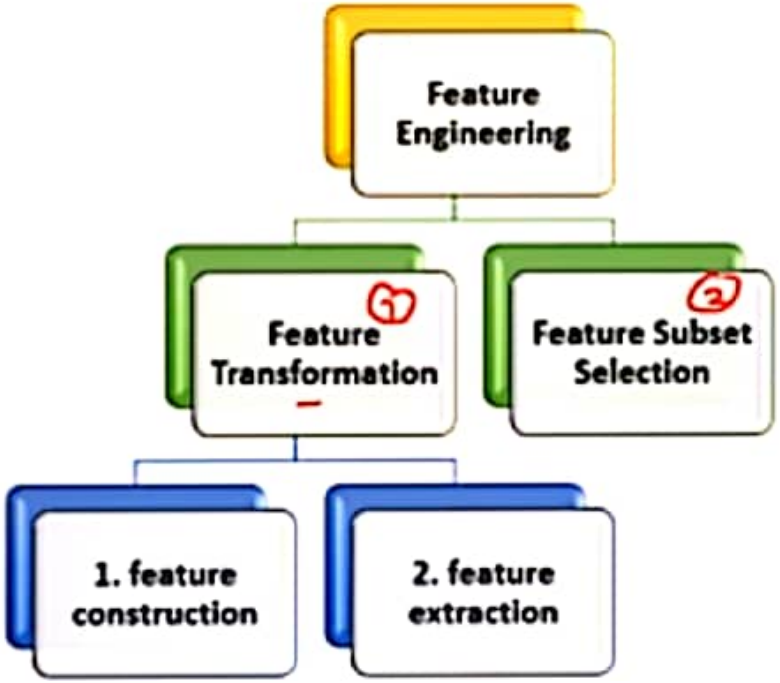
Iris Data set Features

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor
6.8	3.2	5.9	2.3	Virginica
5.5	2.5	4	1.3	Versicolor
5.1	3.5	1.4	0.2	Setosa
6.1	3	4.6	1.4	versicolor

- It has five attributes or features namely Sepal.Length,
- Sepal.Width, Petal.Length, Petal.Width and Species.
- Out of these, the feature 'Species' represent the class variable and the remaining features are the predictor variables.
- It is a five-dimensional data set.

What is Feature Engineering?

- Feature engineering is the process of translating a data set into features such that these features are able to represent the data set more effectively and result in a better learning performance.
- The feature engineering is an important pre-processing step for machine learning.
- It has two major elements:



Feature Transformation

- **Feature transformation** transforms the data (structured or unstructured), into a new set of features which can represent the underlying problem which machine learning is trying to solve.
- There are two variants of feature transformation or feature discovery:
  - 1. feature construction
  - 2. feature extraction

Feature transformation – Feature construction

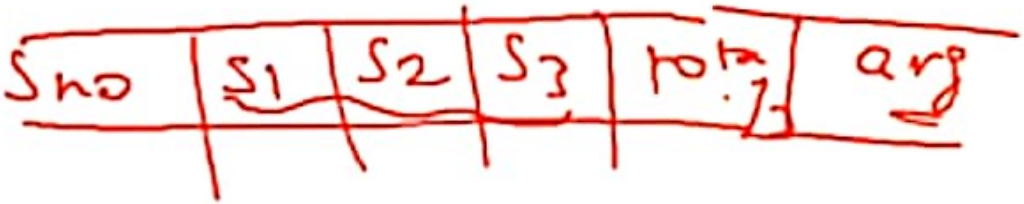
Feature construction process discovers missing information about the relationships between features and augments the feature space by creating additional features.

- Hence, if there are '*n*' features or dimensions in a data set, after feature construction '*m*' more features or dimensions may get added.
- So at the end, the data set will become '*n* + *m*' dimensional.

Feature transformation – Feature extraction

- **Feature extraction**, is the process of extracting or creating a new set of features from the original set of features using some functional mapping.

Stdt



Feature Subset selection / Feature Selection

- The objective of feature selection is to derive a subset of features from the full feature set which is most meaningful in the context of a specific machine learning problem.
- So, essentially the job of feature selection is to derive a subset F_j (F_1, F_2, ..., F_m) of F_i (F_1, F_2, ..., F_n),
- where $m < n$
- such that F_j is a set of most meaningful features and gives the best result for a machine learning problem.

Machine Learning

Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

FEATURE EXTRACTION

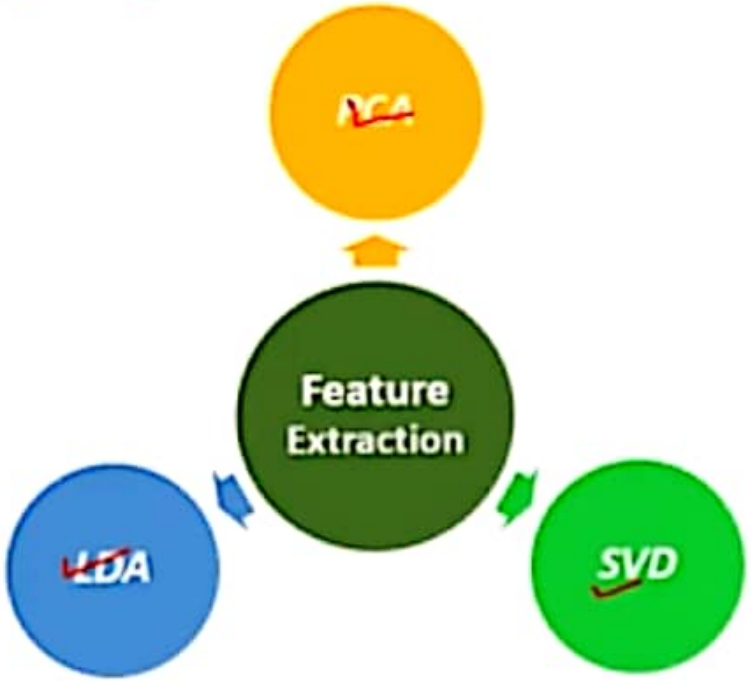
- *Principal Component Analysis (PCA)*
- *Singular value decomposition (SVD)*
- *Linear Discriminant Analysis (LDA)*

$Feat_A$	$Feat_B$	$Feat_C$	$Feat_D$
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2
11	122.32	63	23.2

→

$Feat_1$	$Feat_2$
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
3702	238.42
113.39	16274

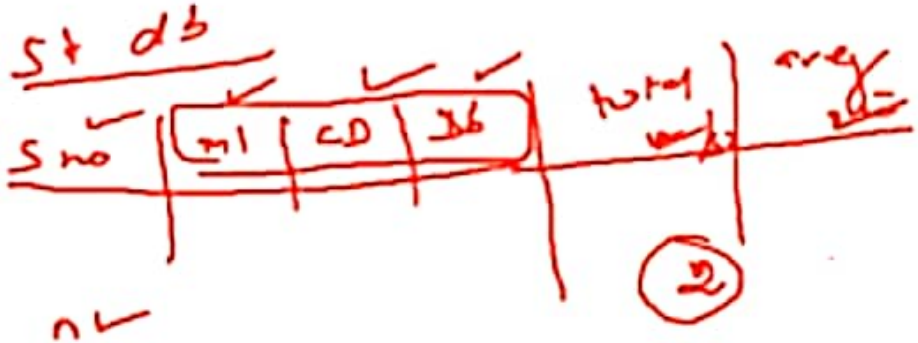
$$Feat_1 = 0.3 \times Feat_A + 0.9 \times Feat_A$$
$$Feat_2 = Feat_A + 0.5\, Feat_B + 0.6 \times Feat_C$$



Feature Extraction

- In feature extraction, new features are created from a combination of original features.
- Some of the commonly used operators for combining the original features include
- 1. For Boolean features: Conjunctions, Disjunctions, Negation, etc.
- 2. For nominal features: Cartesian product, M of N, etc.
- 3. For numerical features: Min, Max, Addition, Subtraction, Multiplication, Division, Average, Equivalence, Inequality, etc.

Feature Extraction...



- Example :
- a data set with a feature set F_i (F_1, F_2, ..., F_n).
- After feature extraction using a mapping function f
- f(F_1, F_2, ..., F_n) then we will have a set of features
- $\dot{F}_i(\dot{F}_1, \dot{F}_2, \ldots, \dot{F}_m)$ such that $\dot{F}_i = f(F_i)$ and $m < n$
- $\dot{F}_1 = k_1F_1 + k_2F_2$ etc.

Feature Extraction...

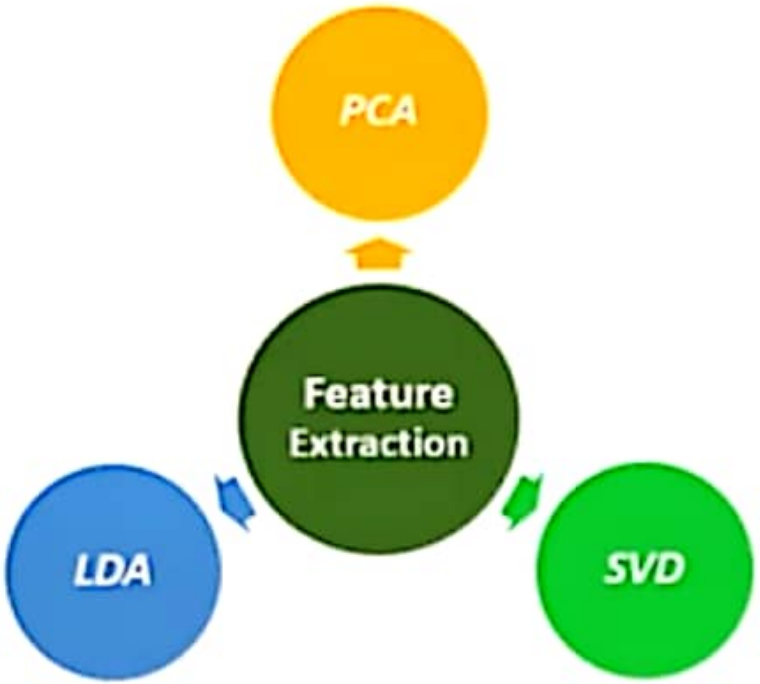
1	2	3	4
$Feat_A$	$Feat_B$	$Feat_C$	$Feat_D$
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2
11	122.32	63	23.2

1	2
$Feat_1$	$Feat_2$
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
37.02	238.42
113.39	167.74

$$Feat_1 = 0.3 \times Feat_A + 0.9 \times Feat_A$$
$$Feat_2 = Feat_A + 0.5\, Feat_B + 0.6 \times Feat_C$$

Feature Extraction…

- The most popular feature extraction algorithms used in machine learning:
- *Principal Component Analysis (PCA)*
- *Singular value decomposition (SVD)*
- *Linear Discriminant Analysis (LDA)*

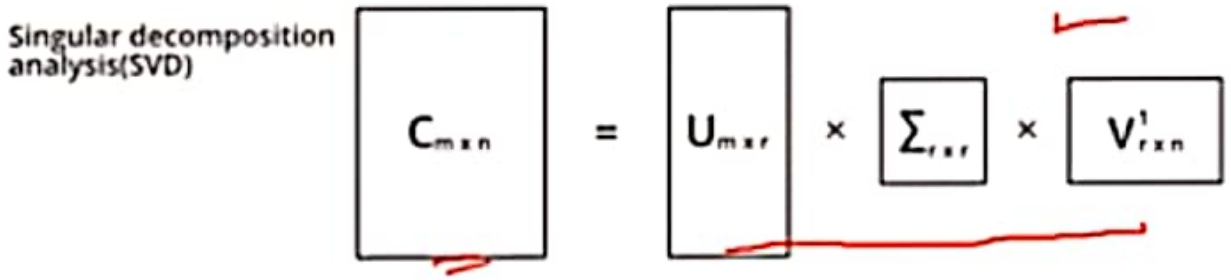


Principal Component Analysis (PCA)

- Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning.
- It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation.
- These new transformed features are called the **Principal Components**.
- It is one of the popular tools that is used for
  - exploratory data analysis and
  - predictive modeling.
- It is a technique to draw strong patterns from the given dataset by reducing the variances.

Singular value decomposition (SVD)

- The Singular Value Decomposition (SVD)) is a matrix factorization technique commonly used in linear algebra.
- i.e.a factorization of that matrix into three matrices .
- $C=UWV^T$



Linear Discriminant Analysis (LDA)

- The objective of LDA is similar to the PCA and SVD, to transform a data set into a lower dimensional feature space.
- LDA focuses on class separability, i.e. separating the features based on class separability so as to avoid over-fitting of the machine learning model.

Machine Learning

Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

FEATURE EXTRACTION – Part 2

- *Principal Component Analysis (PCA)*
- *Objectives of PCA*
- *Steps for PCA*
- *Solved Problem*
- *Applications of PCA*

Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

	Example 1	Example 2	Example 3	Example 4
First principal component PC1	P11 -4.3052	P12 3.7361	P13 5.6928	P14 -5.1238

Principal Component Analysis (PCA)

- Every data set, has multiple attributes or dimensions – many of which might have similarity with each other.
- For example, school student data set,
- the height and weight of a person, are quite related, (if the height is more, generally weight is more and vice versa.)
- So if a data set has height and weight as two of the attributes, obviously they are expected to be having quite a bit of similarity.
- If the features are less in number as well as the similarity between each other is very less then machine learning algorithm performs better.
- The principal component analysis (PCA) is a popular technique used for feature extraction.

Principal Component Analysis (PCA)...

- In PCA, a new set of features are extracted from the original features which are quite dissimilar in nature.
- So an *n*-dimensional feature space gets transformed to an *m*-dimensional feature space, where the dimensions are orthogonal to each other, i.e. completely independent of each other.

Principal Component Analysis (PCA)...

- Basis Vectors:
- A vector is a quantity having both magnitude and direction
- it can determine the position of a point relative to another point in the Euclidean space (i.e. a two or three or 'n' dimensional space).
- A vector space is a set of vectors.
- Vector spaces have a property that they can be represented as a linear combination of a smaller set of vectors, called basis vectors.

Principal Component Analysis (PCA)...

- So, any vector 'v' in a vector space can be represented as

$$v = \sum_{i=1}^{n} a_i u_i$$

- where, a represents '*n*' scalars and *u* represents the basis vectors.
- Basis vectors are orthogonal to each other.

Principal Component Analysis (PCA)...

- Orthogonality of vectors in *n*-dimensional vector space can be thought of an extension of the vectors being perpendicular in a two-dimensional vector space. i.e. Two orthogonal vectors are completely unrelated or independent of each other.
- Each vector in the original set can be expressed as a linear combination of basis vectors,
- It helps in decomposing the vectors to a number of independent components.

Principal Component Analysis (PCA)...

- The feature vector can be transformed to a vector space of the basis vectors which are termed as **principal components.**
- These principal components, like the basis vectors, are orthogonal to each other.
- So a set of feature vectors which may have similarity with each other is transformed to a set of principal components which are completely unrelated.
- the principal components capture the variability of the original feature space
- The number of principal component derived is much smaller than the original set of features.

m n

Principal Component Analysis (PCA)...

- The objective of PCA
- 1. The new features are distinct, i.e. the covariance between the new features, i.e. the principal components is 0.
- 2. The principal components are generated in order of the variability in the data that it captures.
  - the first principal component should capture the maximum variability,
  - the second principal component should capture the next highest variability etc.
- 3. The **sum of variance of the new features or the principal components should be equal to the sum of variance of the original features.**

Some Common terms used in PCA

1. **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
2. **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
3. **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
4. **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
5. **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix

Steps for PCA

- **1. Getting the dataset** – get the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

- **2. Representing data into a structure** :
  Represent our dataset into a structure. i.e. represent the two-dimensional matrix of independent variable X.
- Here each row corresponds to the data items, and the column corresponds to the Features.
- The number of columns is the dimensions of the dataset.

Steps for PCA...

- **3. Standardizing the data**
  from X. In a particular column, the features with high variance are more important, compared to the features with lower variance.
- If the importance of features is independent of the variance of the feature, then divide each data item in a column with the standard deviation of the column.
- This matrix is Z.
- **4. Calculating the Covariance of Z**
  find Z transpose, and multiply it by Z.
- i.e. Z^TZ is the Covariance matrix of Z.

Steps for PCA...

- **5. Calculating the Eigen Values and Eigen Vectors for covariance matrix Z**

  Eigenvectors are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.

- **6. Sorting the Eigen Vectors**

  all the eigenvalues will sort in decreasing order,

- And sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.

Steps for PCA...

- **7. Calculating the new features Or Principal Components** multiply the P* matrix to the Z, and the resultant matrix Z*, Each column of the Z* matrix is independent of each other.
- **8. Remove less or unimportant features from the new dataset, Z***

Example

- Given the following data, use PCA to reduce the dimension from 2 to 1

Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

- Step 1 : Given Data Set

Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

- No. of features, n : 2
- No. of samples, N : 4

- Step 2: Computation of mean of variables
- $\bar{x} = (4+8+13+7) / 4 = 8$
- $\bar{y} = (11+4+5+14) / 4 = 8.5$

- Step 3: computation of covariance matrix
- Write ordered pairs of (x,y) are
- (x,x), (x,y), (y,x), (y,y)
- i) find covariance of all ordered pairs –
- $Cov(x_i,x_j) = (1/(N-1)) \sum_{k=1}^{N} (xik - \bar{x})(xjk - \bar{x})$
- Cov(x,x) = $(1/(N-1)) \sum_{k=1}^{N} (xk - \bar{x})^2$ (both covariance are same 'x')
- $=(1/4-1)(4-8)^2+(8-8)^2+(13-8)^2-(7-8)^2=14$
- $Cov(x,y)=(1/4-1)((4-8)(11-8.5)+(8-8)(4-8.5)+(13-8)(5-8.5)+(7-8)(14-8.5))$
- $=-11$
- $Cov(y,x)=cov(x,y)=-11$
- $Cov(y,y)=(1/4-1)((11-8.5)^2+(4-8.5)^2+(5-8.5)^2+(14-8.5)^2)=23$

Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

ii) Covariance matrix nxn (2x2)

- $S=\begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix}$
- $S=\begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$

- Step 4: Eigen Value, Eigen vector and normalized eigen vector.
- i) Eigen Value
- Determinent $|S-\lambda I|=0$ $I=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ and $\lambda * I=\begin{bmatrix}\lambda & 0\\ 0 & \lambda\end{bmatrix}$
- Where S is Covariance mateix, I is Identity matrix and λ is eigen value.
- $\text{Det}(S-\lambda I)=\det\begin{bmatrix}14-\lambda & -11\\ -11 & 23-\lambda\end{bmatrix}=0$
- $(14-\lambda)(23-\lambda)-(-11*-11)=\lambda^2-37\lambda+201=0$ $\frac{\sqrt{(b2-4ac)}}{2a}$
- $\lambda=30.3849,\ 6.6151$
- $\lambda 1>\lambda 2,$
- $\lambda 1=30.3849$ (first principal component)
- and $\lambda 2=6.6151$

- ii) eigen vector of λ_1
- $(S-\lambda_1 I)U_1=0$ where S is covariance matrix, λ^1 is largest principal component, I is Identity matrix and U_1 is eigen vector.
- $= \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$
- $= \begin{bmatrix} (14-\lambda)u1 & -11u2 \\ -11u1 & (23-\lambda 1)u2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Then we get linear equations
- $(14-\lambda)u1 - 11u2 = 0$ & $-11u1 + (23-\lambda 1)u2 = 0$
- $\frac{u1}{11} = \frac{u2}{14-\lambda_1} = t$
- If t=1, then u1=11 and u2= $14-\lambda_1$
- Eigen vector u1 of $\lambda_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix} = \begin{bmatrix} 11 \\ 14-30.3849 \end{bmatrix} = \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$

- Eigen vector u1 of $\lambda_1 = \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$

- iii) Normalize the eigen vector u_1.

- e1= $\begin{bmatrix} 11/(\sqrt{11^2} - 16.38)^2 \\ -16.3849/(\sqrt{11^2} - 16.38)^2 \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$

- IIIly eigen vector of λ_2

- e2= $\begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$

- Step 5 : Derive new data set (reduced to 1 dimension which is the first principal component)

- $P11 = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$

- $= [0.5574 \ \ -0.8303] \begin{bmatrix} -4 \\ 2.5 \end{bmatrix} = -4.3052$

- $P12 = [0.5574 \ \ -0.8303] \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix} = 3.7361$

- P13=5.6928

- P14=-5.1238

$\begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$

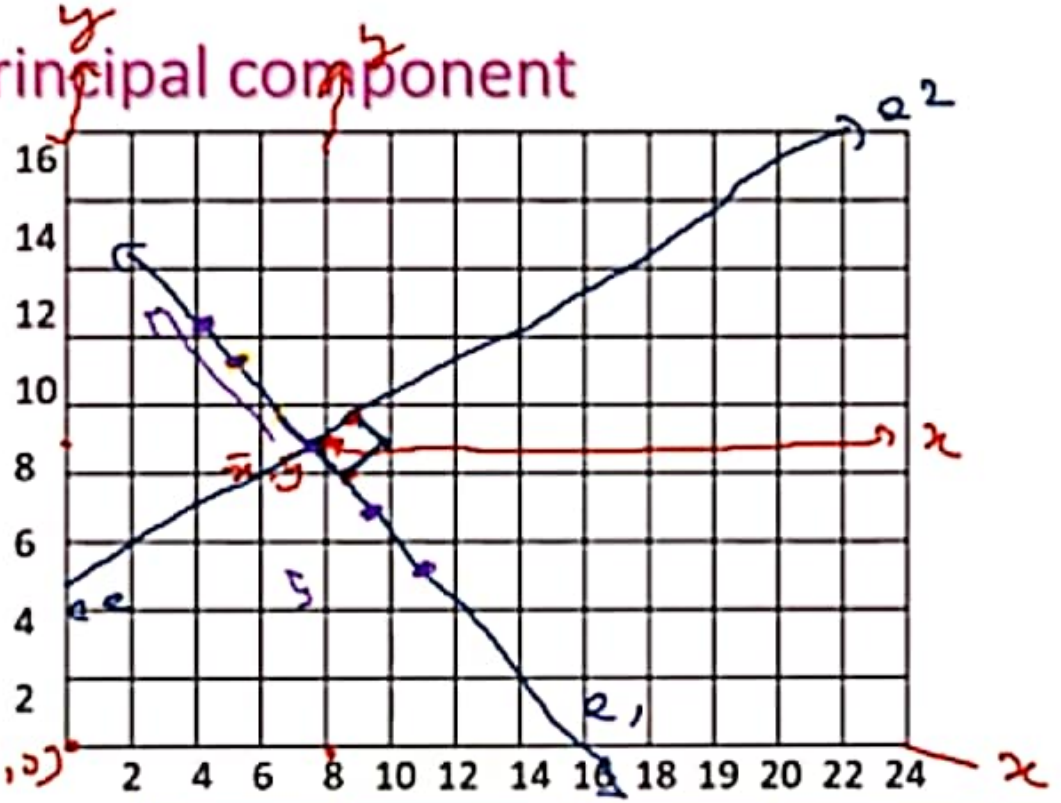
Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

PCA

	Example 1	Example 2	Example 3	Example 4
First principal component PC1	P11 -4.3052	P12 3.7361	P13 5.6928	P14 -5.1238

Step 6: Coordinate system for principal component

- Select mean of x,y (8,8.5)
- Then select e1 & e2
- e1= $\begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$
- e2= $\begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$
- Then draw the line
- e1 and e2
- Then place the table values on e1



Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

	Example 1	Example 2	Example 3	Example 4
First PC1	P11 -4.3052	P12 3.7361	P13 5.6928	P14 -5.1238

PCA...

- If every values lies on e1 i.e. on single dimension, then it is easy for computation, when compared to two dimension.

Applications of PCA

- PCA is mainly used as the dimensionality reduction technique in various AI applications such **as computer vision, image compression, etc.**
- It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.

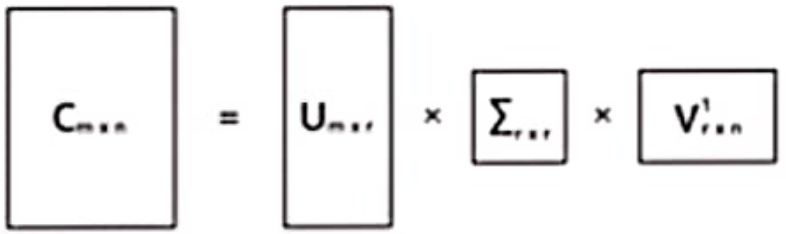
Machine Learning

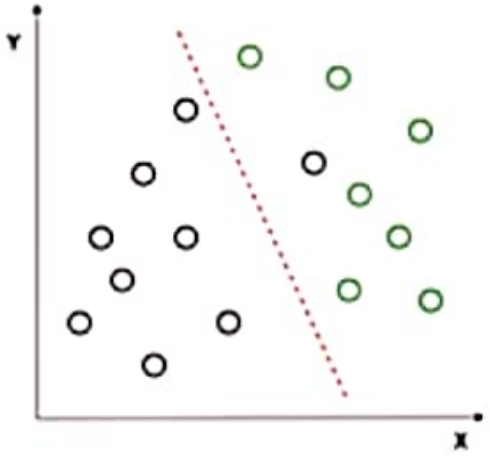
Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

FEATURE EXTRACTION – Part 3

- *Singular value decomposition (SVD)*
- *Linear Discriminant Analysis (LDA)*





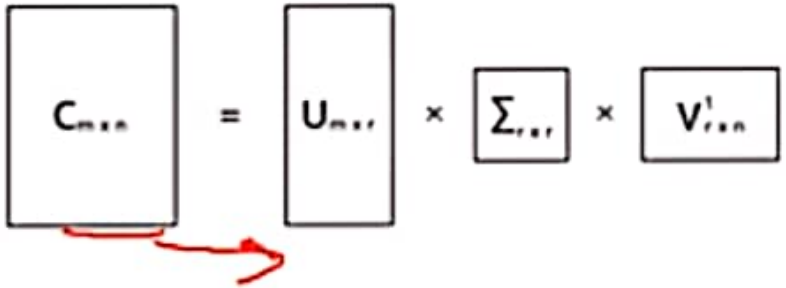
Machine Learning

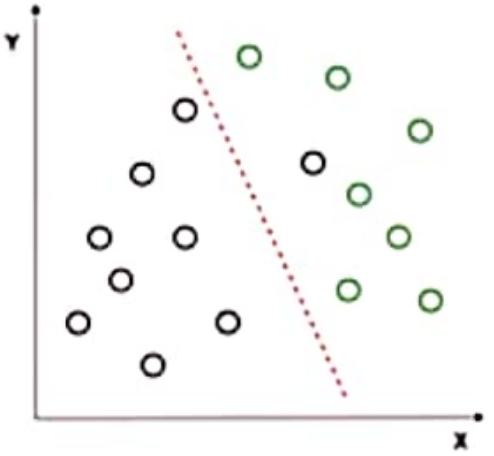
Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

FEATURE EXTRACTION – Part 3

- *Singular value decomposition (SVD)*
- *Linear Discriminant Analysis (LDA)*



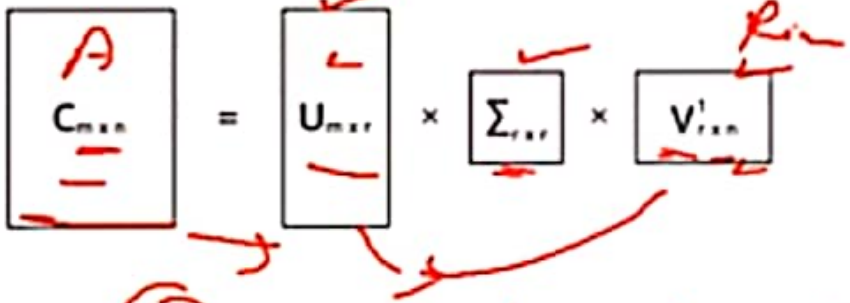


Singular value decomposition (SVD)

- Singular value decomposition (SVD) is a matrix factorization technique commonly used in linear algebra.
- SVD of a matrix A ($m \times n$) is a factorization of the form:

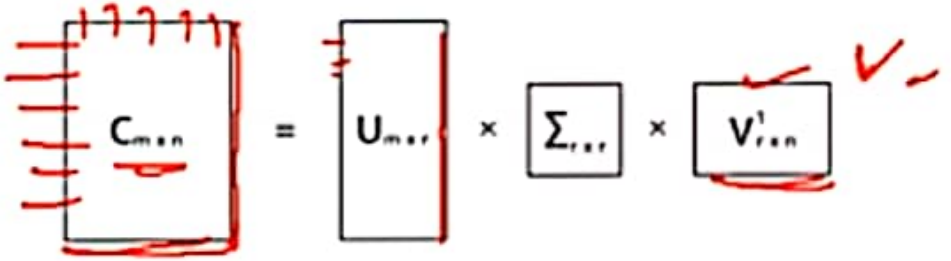
$$A = U \sum V^T$$

- where, U and V are orthonormal matrices,
- U is an $m \times m$ unitary matrix,
- V is an $n \times n$ unitary matrix and
- Σ is an $m \times n$ rectangular diagonal matrix.
- The diagonal entries of Σ are known as singular values of matrix A.
- The columns of U and V are called the left-singular and right-singular vectors of matrix A, respectively.



Singular Value Decomposition (SVD)...

- SVD of a data matrix - the properties:
- 1. Patterns in the attributes are captured by the right-singular vectors, i.e. the columns of *V*.
- 2. Patterns among the instances are captured by the left-singular, i.e. the columns of *U*.
- 3. Larger a singular value, larger is the part of the matrix A that it accounts for and its associated vectors.
- 4. New data matrix with '*k*' attributes is obtained using the equation
- $D = D \times [v_1, v_2, \ldots, v_k]$
- Thus, the dimensionality gets reduced to *k*
- SVD is often used in the context of text data.



Linear Discriminant Analysis (LDA)

- Linear discriminant analysis (LDA) is another commonly used feature extraction technique like PCA or SVD.
- The objective of LDA is, transform a data set into a lower dimensional feature space.
- LDA focuses on class separability.
- i.e. separating the features based on class separability so as to avoid over-fitting of the machine learning model.

Steps of LDA

- LDA calculates eigenvalues and eigenvectors within a class and interclass scatter matrices.
- Steps of LDA:
- 1. Calculate the mean vectors for the individual classes.
- 2. Calculate intra-class and inter-class scatter matrices.
- 3. Calculate eigenvalues and eigenvectors for S_w^{-1} and S_B, where S_w is the intra-class scatter matrix and S_B is the inter-class scatter matrix

$$S_W = \sum_{i=1}^{c} S_i$$

$$S_i = \sum_{x \in D_i}^{n} (x - m_i)(x - m_i)^T$$

- where, m_i is the mean vector of the i-th class

Linear Discriminant Analysis (LDA)...

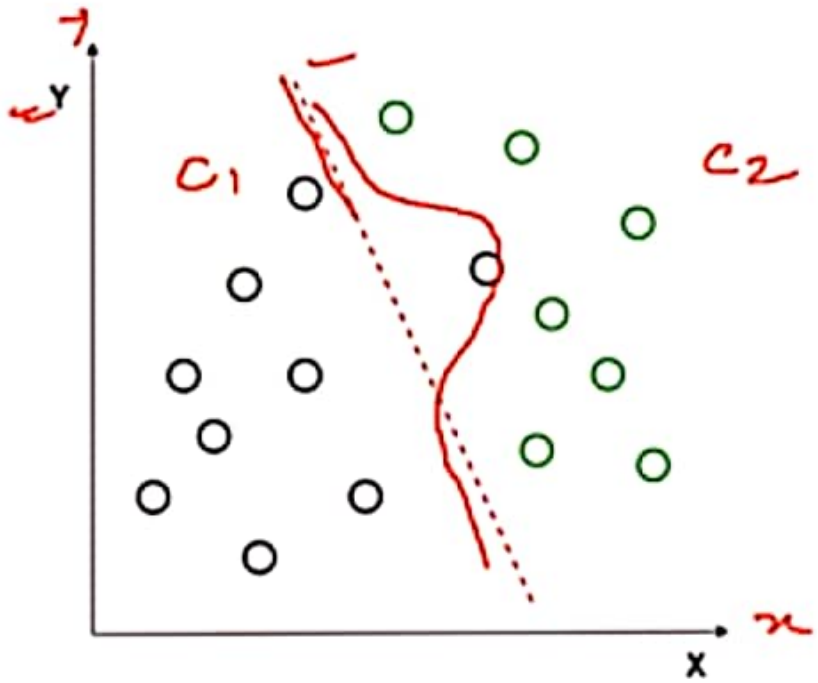
$$S_B = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$$

- where, m_i is the sample mean for each class, m is the overall mean of the data set, N_i is the sample size of each class
- 4. Identify the top '*k*' eigenvectors having top '*k*' eigenvalues

Linear Discriminant Analysis (LDA)...

- We start with the case where there are two classes, then generalize to K > 2 classes
- For example, we have two classes and we need to separate them efficiently.
- Classes can have multiple features.
- Using only a single feature to classify them, may result in some overlapping as shown in the below figure.

- two sets of data points belonging to two different classes that to classify.
- As shown in the given 2D graph, when the data points are plotted on the 2D plane, there's no straight line that can separate the two classes of the data points completely.
- LDA (Linear Discriminant Analysis) is used which reduces the 2D graph into a 1D graph in order to maximize the separability between the two classes.



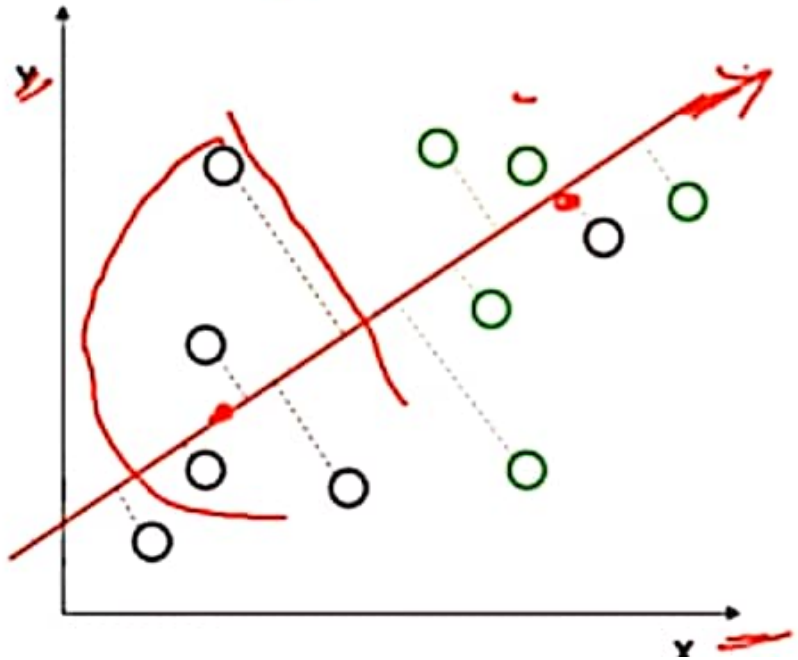
Linear Discriminant Analysis (LDA)...

- LDA uses both the axes (X and Y) to create a new axis and projects data onto a new axis
- Hence, maximize the separation of the two categories and, reducing the 2D graph into a 1D graph.

Linear Discriminant Analysis (LDA)...

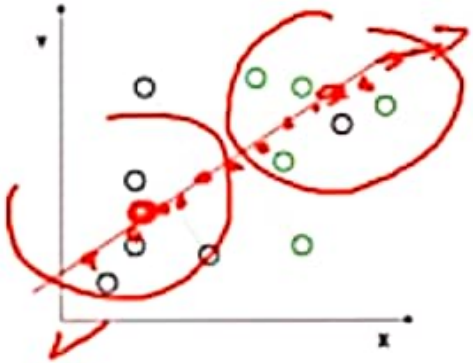
- Two criteria are used by LDA to create a new axis:

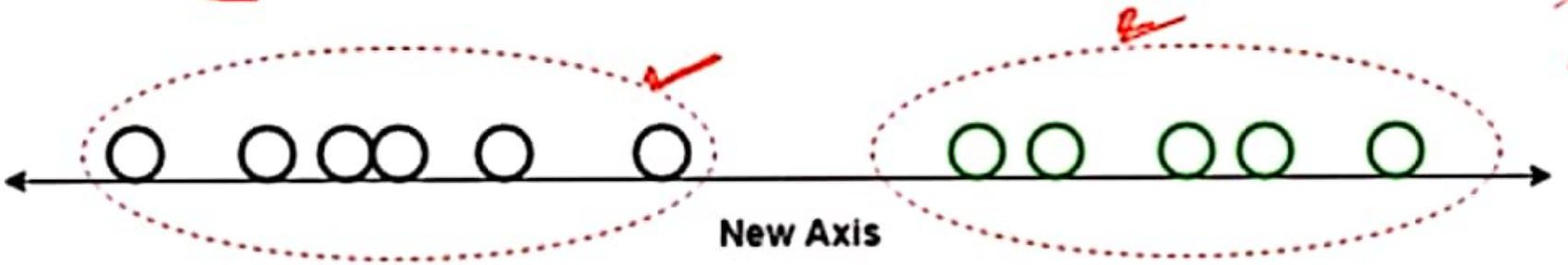
1. Maximize the distance between means of the two classes.
2. Minimize the variation within each class.



Linear Discriminant Analysis (LDA)...

- it can be seen that a new axis (in red) is generated and plotted in the 2D graph, such that it maximizes the distance between the means of the two classes and minimizes the variation within each class.
- This newly generated axis increases the separation between the data points of the two classes.
- After generating this new axis, all the data points of the classes are plotted on this new axis and are shown in the figure given below.





Machine Learning

Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

FEATURE SUBSET SELECTION – Part-1

- Issues in high-dimensional data
- Key drivers of feature selection –
  - Feature relevance
  - Feature redundancy

- Next…
- Measures of feature relevance and redundancy
- Overall feature selection process

Feature selection

- **Feature selection is the most critical preprocessing activity, it aims to select a subset of system attributes or features**
- In a student data set, predict the weight of students based on past information about similar students.
- The features are, Roll Number, Age, Height, and Weight.
- The roll number can have no bearing, in predicting student weight.
- So we can eliminate the feature roll number and build a feature subset .
- The subset of features is expected to give better results than the full set.

Roll Number	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3
38	14	1.24	25.2
45	12	1.12	23.4

→

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3
14	1.24	25.2
12	1.12	23.4

Issues in high-dimensional data

- With the rapid innovations in the digital space, the volume of data generated has increased to an unbelievable extent.
- At the same time, breakthroughs in the storage technology area have made storage of large quantity of data quite cheap
- This has further motivated the storage and mining of very large and high dimensionality data sets.

Issues in high-dimensional data...

- Two new application domains have seen drastic development.
  - The biomedical research,
  - The text categorization
- The biomedical research which includes gene selection from microarray data.
- It generates data sets having a number of features in the range of a few tens of thousands.
- The text data generated extremely high dimensions, from social networking sites, like emails, messages, articles and etc.
- In a large document corpus having few thousand documents embedded, the number of unique word tokens which represent the feature of the text data set, can also be in the range of a few tens of thousands.
- This high-dimensional data may be a big challenge for any machine learning algorithm.

Issues in high-dimensional data...

- Problems in high-dimensional data
  - very high quantity of computational resources and high amount of time will be required.
  - the performance of the model – both for supervised and unsupervised machine learning task, also degrades sharply due to unnecessary noise in the data.
  - a model built on an extremely high number of features may be very difficult to understand.
- Hence, it is necessary to take a subset of the features instead of the full set.

Issues in high-dimensional data...

- The objective of feature selection is three-fold:
- Having faster and more cost-effective (i.e. less need for computational resources) learning model
- Improving the efficiency of the learning model
- Having a better understanding of the underlying model that generated the data

Key drivers of feature selection – feature relevance

- In supervised learning, the input data set which is the training data set, has a class label attached.
- the model have to assign class labels to new, un-labelled data.
- Each of the predictor variables, is expected to contribute information to decide the value of the class label.
- A variable is not contributing any information, it is said to be irrelevant.
- The information contribution for prediction is very little, the variable is said to be weakly relevant.
- Remaining variables, which make a significant contribution to the prediction task are said to be strongly relevant variables.

Key drivers of feature selection – feature relevance...

- In unsupervised learning, there is no training data set or labelled data.
- Grouping of similar data instances are done and similarity of data instances are evaluated based on the value of different variables.
- Certain variables do not contribute any useful information for deciding the similarity of dissimilarity of data instances.
- These variables are marked as irrelevant variables in the context of the unsupervised machine learning task.

Key drivers of feature selection – feature relevance...

- Example
- Student data set: to predict Weight of a student, Roll number doesn't contribute any significant information, in supervised learning
- to group students with similar academic capabilities, Roll number can really not contribute any information, in unsupervised learning.
- The irrelevant candidate are rejected in selecting a subset of features.
- The weakly relevant features are to be rejected or not, on a case-to-case basis.

Key drivers of feature selection - Feature redundancy

- A feature information which is similar to one or more other features.
- For example, in the weight prediction problem, both the features Age and Height contribute similar information.
  - increase in Age, Weight is expected to increase.
  - increase of Height also Weight is expected to increase.
  - Age and Height increase with each other.
- So, Age and Height contribute similar information.
- when one feature is similar to another feature, the feature is said to be potentially redundant in the context of the learning problem.

Key drivers of feature selection - Feature redundancy...

- All features having potential redundancy, can be reject in the final feature subset.
- Only a small number of representative features are being a part of the final feature subset.
- **The objective of feature selection** is to remove all features which are irrelevant and redundant.

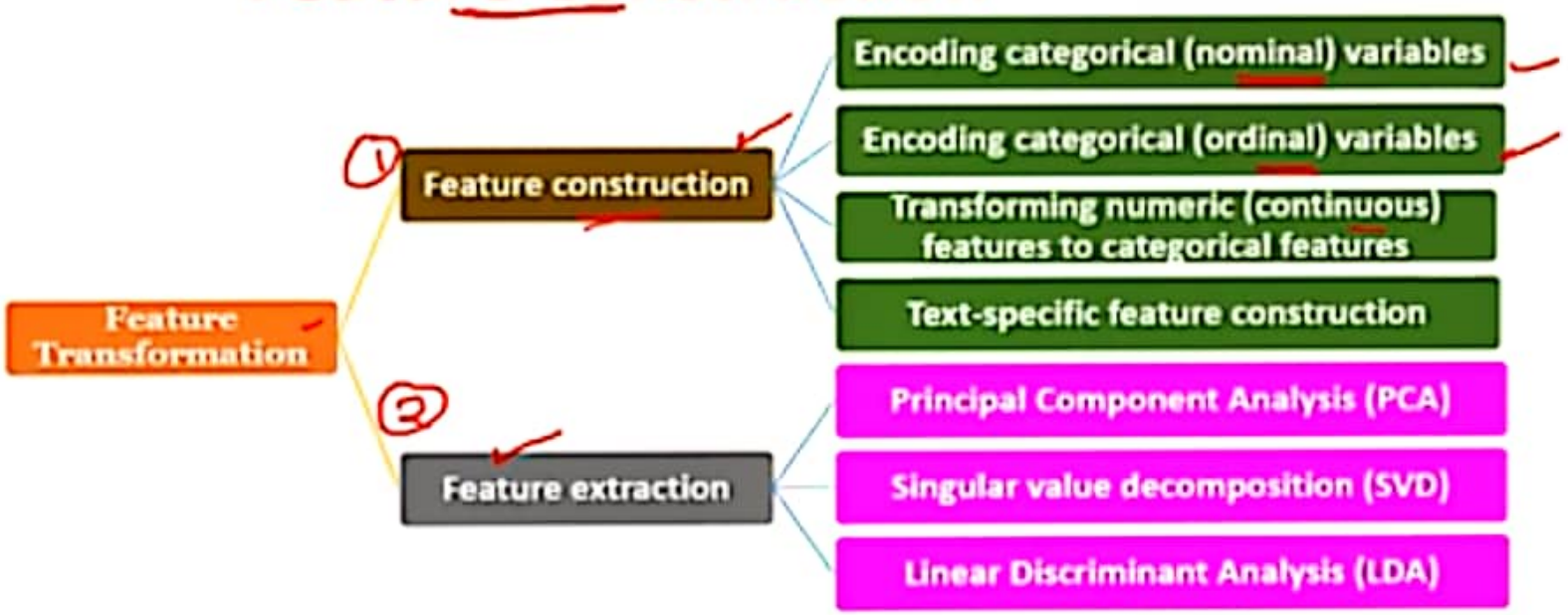
Machine Learning

Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

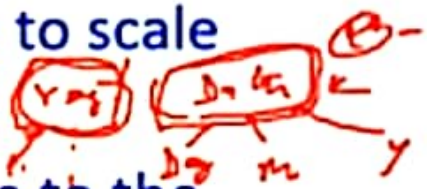
FEATURE TRANSFORMATION –

Feature Construction



FEATURE TRANSFORMATION

- The feature transformation Engineering is an important prerequisite for any machine learning model in data preprocessing.
- all available attributes of the data set are used as features.
- there will be data with different magnitudes and we have to scale down different features to the same range of magnitude.
- because most of the algorithms will give more importance to the features with high volume rather than giving the same importance to all features.
- This will lead to wrong predictions and faulty models.



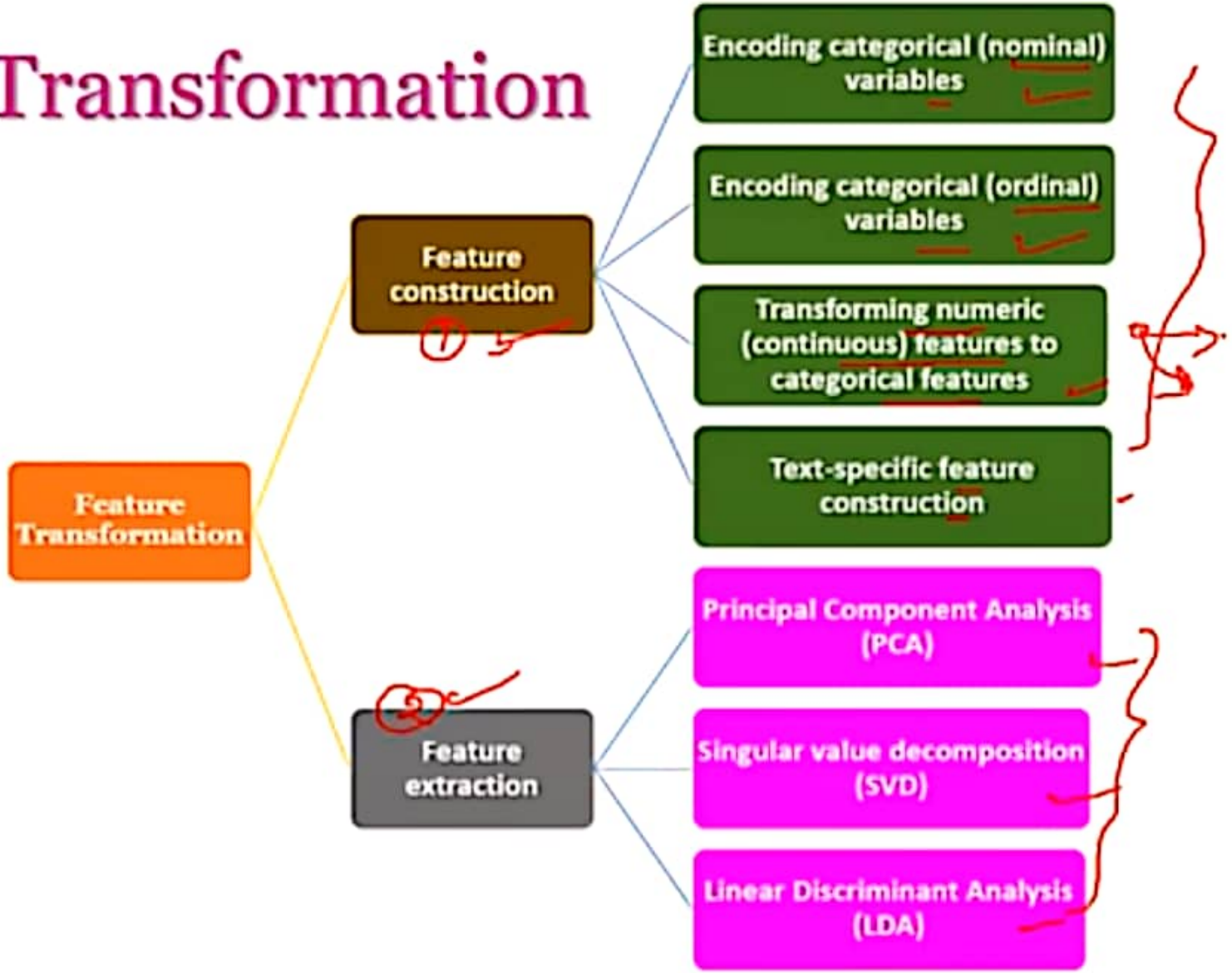
Feature Transformation...

- In case a model has to be trained to classify a document as spam or non-spam, we can represent a document as a bag of words.
- Then the feature space will contain all unique words occurring across all documents.
- This will easily be a feature space of a few hundred thousand features.
- If we start including bigrams or trigrams along with words, the count of features will run in millions.
- To deal with this problem, Feature transformation is used for dimensionality reduction and hence for boosting learning model performance.

Goals of Feature Transformation

1. Achieving best reconstruction of the original features in the data set
2. Achieving highest efficiency in the learning task

Feature Transformation



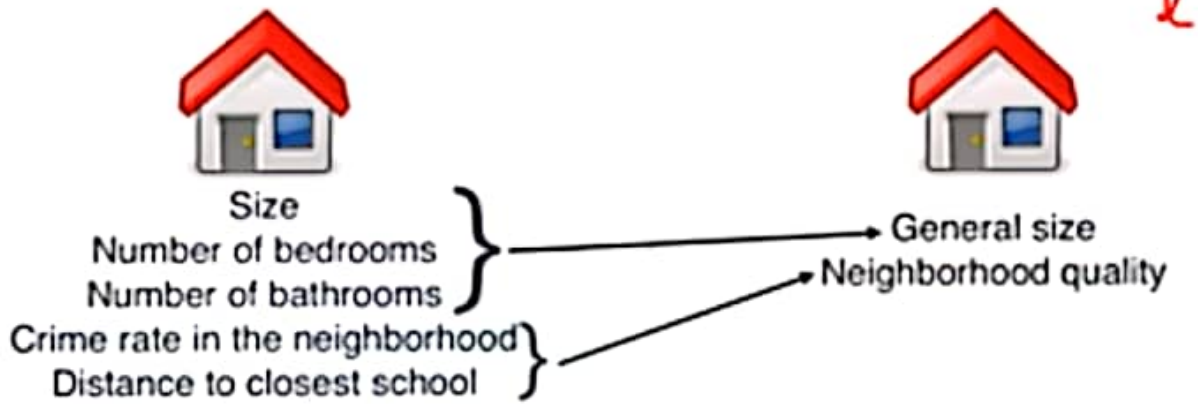
Feature Construction

- Feature construction involves transforming a given set of input features to generate a new set of more powerful features.
- Example: a real estate data set having details of all apartments sold in a specific region.
- The data set has some features – apartment length, apartment breadth, and price of the apartment.
- If it is used as an input to a regression problem, such data can be training data for the regression model.

Feature Construction...

- the model should predict the price of an apartment
- Instead of using length and breadth of the apartment as a predictor, use the area of the apartment, which is not an existing feature of the data set and can be added to the data set.
- transform the three-dimensional data set to a four-dimensional data set.

l, w, a, p



Feature construction (example 1)

apartment_ length	apartment_ breadth	apartment_ price
80	59	23,60,000
54	45	12,15,000
78	56	21,84,000
63	63	19,84,000
83	74	30,71,000
92	86	39,56,000

apartment_ length	apartment_ breadth	apartment_ area	apartment_ price
80	59	4,720	23,60,000
54	45	2,430	12,15,000
78	56	4,368	21,84,000
63	63	3,969	19,84,500
83	74	6,142	30,71,000
92	86	7,912	39,56,000

Feature Construction...

- There are certain situations where feature construction is an essential activity before we can start with the machine learning task.
- These situations are
  1. when features have categorical value and machine learning needs numeric value inputs
  2. when features having numeric (continuous) values and need to be converted to ordinal values
  3. when text-specific feature construction needs to be done

Encoding categorical (nominal) variables

- Adata set on athletes, has features
  - age,
  - city of origin,
  - parents athlete (i.e. indicate whether any one of the parents was an athlete) and
  - Chance of Win.

Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

(a)

Encoding categorical (nominal) variables...

- The feature chance of a win is a class variable while the others are predictor variables.
- So there are three features –
  - City of origin,
  - Parents athlete, and
  - Chance of win,
- which are categorical in nature and cannot be used by any machine learning task.
- feature construction can be used to create new dummy features which are usable by machine learning algorithms.

Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

(a)

Encoding categorical (nominal) variables...

- In this case, the feature 'City of origin' has three unique values namely City A, City B, and City C, three dummy features namely origin_ city_A,origin_city_B, and origin_city_C is created.
- dummy features parents_athlete_Y and parents_athlete_N are created for feature 'Parents athlete' and
- win_chance_Y and win_chance_N are created for feature 'Chance of win'.
- The dummy features have value 0 or 1 based on the categorical value

Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

Age (Years)	origin_ city_A	origin_ city_B	origin_ city_C	parents_ athlete_Y	parents_ athlete_N	win_ chance_Y	win_ chance_N
18	1	0	0	1	0	1	0
20	0	1	0	0	1	1	0
23	0	1	0	1	0	1	0
19	1	0	0	0	1	0	1
18	0	0	1	1	0	0	1
22	0	1	0	1	0	1	0

(b)

Encoding categorical (nominal) variables...

- the features 'Parents athlete' and 'Chance of win' in the original data set can have two values only.
- So creating two features from them is a kind of duplication, since the value of one feature can be decided from the value of the other.
- To avoid this duplication, we can just leave one feature and eliminate the other, as shown in Figure 4.3c.

Age (Years)	origin_ city_A	origin_ city_B	origin_ city_C	parents_ athlete_Y	parents_ athlete_N	win_ chance_Y	win_ chance_N
18	1	0	0	1	0	1	0
20	0	1	0	0	1	1	0
23	0	1	0	1	0	1	0
19	1	0	0	0	1	0	1
18	0	0	1	1	0	0	1
22	0	1	0	1	0	1	0

Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	win_chance_Y
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0
22	0	1	0	1	1

Encoding categorical (ordinal) variables

- A student data set with three variable – science marks, maths marks and grade
- The grade is an ordinal variable with values A, B, C, and D.
- To transform this variable to a numeric variable, create a feature num_grade mapping a numeric value against each ordinal value.
- grades A, B, C, and D are mapped to values 1, 2, 3, and 4 in the transformed variable

marks_science	marks_maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D
62	57	B

(a)

marks_science	marks_maths	num_grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4
62	57	2

(b)

Transforming numeric (continuous) features to categorical features

- The real estate price prediction problem, which is a regression problem,
- as a real estate price category prediction, which is a classification problem.
- the numerical data divided into multiple categories based on the data range.
- In the context of the real estate price prediction example, the original data set has a numerical feature apartment_price
- It can be transformed to a categorical variable price-grade either as shown in Figure 4.5b or as shown in Figure 4.5c.

apartment_ area	apartment_ price
4,720	23,60,000
2,430	12,15,000
4,368	21,84,000
3,969	19,84,500
6,142	30,71,000
7912	39,56,000

(a)

apartment_ area	apartment_ grade
4,720	Medium
2,430	Low
4,368	Medium
3,969	Low
6,142	High
7912	High

(b)

apartment_ area	apartment_ grade
4,720	2
2,430	1
4,368	2
3,969	1
6,142	3
7912	3

Text-specific feature construction

- In the current world, text is arguably the most predominant medium of communication.
- The social networks like Facebook or microblogging channels like Twitter or emails or short messaging services such as Whatsapp, text plays a major role in the flow of information.
- Hence, **text mining** is an important area of research in technology and commercial industries.
- The text data are unstructured nature of the data and do not have readily available features, like structured data sets, on which machine learning tasks can be executed.
- All machine learning models need numerical data as input.
- So the text data in the data sets need to be transformed into numerical features.

Text-specific feature construction

- Text data, or corpus which is the more popular keyword, is converted to a numerical representation
- following a process is known as vectorization.
- In this process, word occurrences in all documents belonging to the corpus are consolidated in the form of bag-of-words.
- There are three major steps that are followed:
- 1. tokenize
- 2. count
- 3. normalize

Text-specific feature construction...

- First tokenize a corpus, the blank spaces and punctuations are used as delimiters to separate out the words, or tokens.
- Then the number of occurrences of each token is counted, for each document.
- Lastly, tokens are weighted with reducing importance when they occur in the majority of the documents.

Feature construction (text-specific)

- A matrix is then formed with each token representing a column and a specific document of the corpus representing each row.
- Each cell contains the count of occurrence of the token in a specific document.
- This matrix is known as a **document-term matrix** (also known as a term-document matrix).

	This	House	Build	Feeling	Well	Theatre	Movie	Good	Lonely	...
	2	1	1	0	0	1	1	1	0	
	0	0	0	1	1	0	0	0	0	
	1	0	0	2	1	1	0	0	1	
	0	0	0	0	1	0	1	1	0	
	.	.	.	.	.	.	.	.	.	
	.	.	.	.	.	.	.	.	.	
	.	.	.	.	.	.	.	.	.	
	.	.	.	.	.	.	.	.	.	

Machine Learning

Subject Code: 20A05602T

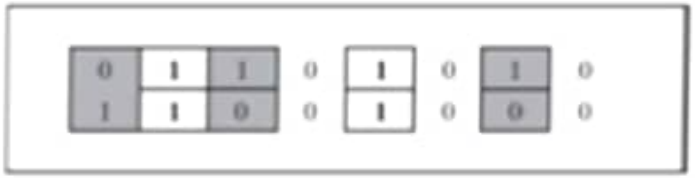
UNIT 2 –Basics of Feature Engineering

FEATURE SUBSET SELECTION – Part-2

- **Measures of feature relevance and redundancy**
- Measures of Feature Relevance
- Measures of Feature Redundancy
  1. Correlation-based measures
  2. Distance-based measures, and
  3. Other coefficient-based measure

- **Next…**
- Overall feature selection process

(a) Hamming distance measurement



(b) Jaccard coefficient measurement

(c) SMC measurement

Measures of Feature Relevance

- The feature relevance is based on the amount of information contributed by a feature.
- For supervised learning, mutual information is decide by the value of the class label.
- Mutual information can be calculated as follows:
- MI(C, f) = H(C) + H(f) - H(C, f)
- where, marginal entropy of the class, H(C) = $-\sum_{i=1}^{k} p(C_i) \log_2 p(C_i)$
- marginal entropy of the feature 'x', H(f) = $-\sum_{c} p(f = x) \log_2 p(f = x)$

- and k = number of classes, c = class variable, f = feature set that take discrete values.

Measures of Feature Relevance...

- In unsupervised learning, the entropy of features is calculated for all the features.
- Then, the features are ranked in a descending order of information gain from a feature, and top 'β' percentage of features are selected as relevant features.
- The entropy of a feature f is calculated using Shannon's formula :

$$H(f) = -\sum_x p(f = x) \log_2 p(f = x)$$

- $\sum_x$ is used only for features that take discrete values.

Measures of Feature redundancy

- Feature redundancy, is based on similar information contribution by multiple features.
- Three types of measures:
- 1. Correlation-based measures
- 2. Distance-based measures, and
- 3. Other coefficient-based measure

Correlation-based similarity measure

- Correlation is a measure of linear dependency between two random variables.
- **Pearson's product moment** correlation coefficient is used to measures the correlation between two random variables.
- For two random feature variables F1 and F2 , Pearson correlation coefficient is defined as:

$$\alpha = \frac{cov(F_1, F_2)}{\sqrt{var(F_1).var(F_2)}}$$

$$cov(F_1, F_2) = \sum (F_{1_i} - \overline{F_1}).(F_{2_i} - \overline{F_2})$$

$$var(F_1) = \sum (F_{1_i} - \overline{F_1})^2, \text{where } \overline{F_1} = \frac{1}{n}.\sum F_{1_i}$$

$$var(F_2) = \sum (F_{2_i} - \overline{F_2})^2, \text{where } \overline{F_2} = \frac{1}{n}.\sum F_{2_i}$$

Correlation-based similarity measure...

- Correlation values range between +1 and –1.
- A correlation of 1 (+ / –) indicates perfect correlation, i.e. the two features having a perfect linear relationship.
- In case the correlation is 0, then the features seem to have no linear relationship.
- Generally, for all feature selection problems,
- a threshold value is adopted to decide whether two features have adequate similarity or not.

Distance-based similarity measure

- The most common distance measure is the Euclidean distance, which, between two features F1 and F2 are calculated as:

$$d(F_1, F_2) = \sqrt{\sum_{i=1}^{n} (F_{1_i} - F_{2_i})^2}$$

- where F1 and F2 are features of an n-dimensional data set.
- The data set has two features, aptitude (F1) and communication (F2) under consideration. The Euclidean distance between the features has been calculated using the formula provided above.

Aptitude (F_1)	Communication (F_2)	(F_1 – F_2)	(F_1 – F_2)^2
2	6	–4	16
3	5.5	–2.5	6.25
6	4	2	4
7	2.5	4.5	20.25
8	3	5	25
6	5.5	0.5	0.25
6	7	–1	1
7	6	1	1
8	6	2	4
9	7	2	4
			81.75

Distance-based similarity measure...

- A more generalized form of the Euclidean distance is the **Minkowski distance**, measured as

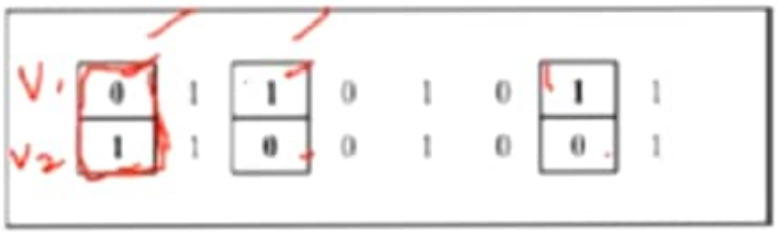
$$d(F_1, F_2) = \sqrt[r]{\sum_{i=1}^{n} (F_{1_i} - F_{2_i})^r}$$

- Minkowski distance takes the form of Euclidean distance when r = 2.
- At r = 1, it takes the form of **Manhattan distance** as shown below:

$$d(F_1, F_2) = \sum_{i=1}^{n} |F_{1_i} - F_{2_i}|$$

Distance measures between features

- To calculate the distance between binary vectors is, the **Hamming distance.**
- For example, the Hamming distance between two vectors 01101011 and 11001001 is 3



(a) Hamming distance measurement

Other similarity measures

- Jaccard index/coefficient is used as a measure of similarity between two features.
- The Jaccard distance, a measure of dissimilarity between two features, is complementary of Jaccard index.
- For two features having binary values, Jaccard index is measured as

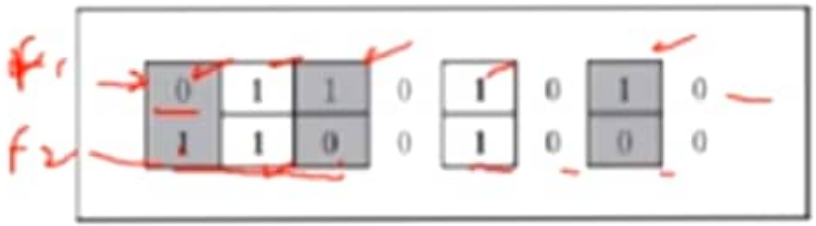
$$J = \frac{n_{11}}{n_{01} + n_{10} + n_{11}}$$

- where, n11 = number of cases where both the features have value 1
- n01 = number of cases where the feature 1 has value 0 and feature 2 has value 1
- n10 = number of cases where the feature 1 has value 1 and feature 2 has value 0
- Jaccard distance, dj = 1 - J

Other similarity measures...

- Let's consider two features F_1 and F_2 having values (0,1, 1, 0, 1, 0, 1, 0) and (1, 1, 0, 0, 1, 0, 0, 0).
- the identification of the values of n11 , n01 and n10 .
- As shown, the cases where both the values are 0 have been left out without border – as an indication of the fact that they will be excluded in the calculation of Jaccard coefficient.
- Jaccard coefficient of F and F , J =

$$\frac{n_{11}}{n_{01} + n_{10} + n_{11}} = \frac{2}{1 + 2 + 2} = \frac{2}{5} \text{ or } 0.4.$$



(b) Jaccard coefficient measurement

- ∴ Jaccard distance between F1 and F2 , dj = 1 – J = 0.6

1 - 0.4 = 0.6

Other similarity measures...

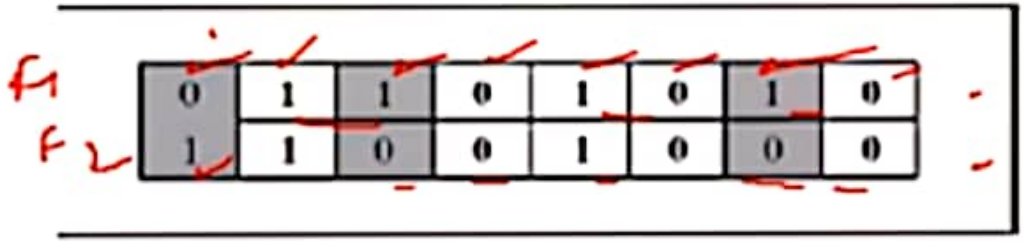
- **Simple matching coefficient (SMC)** is almost same as Jaccard coeficient except the fact that it includes a number of cases where both the features have a value of 0.

$$SMC = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

- where, n11 = number of cases where both the features have value 1
- n01 = number of cases where the feature 1 has value 0 and feature 2 has value 1
- n10 = number of cases where the feature 1 has value 1 and feature 2 has value 0
- n00 = number of cases where both the features have value 0

Other similarity measures...

- Quite understandably, the total count of rows, n = n00 + n01 + n10 + n11 .
- all values have been included in the calculation of SMC.



(c) SMC measurement

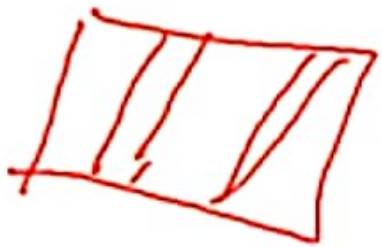
$$\therefore \text{SMC of } F_1 \text{ and } F_2 = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}} = \frac{2 + 3}{3 + 1 + 2 + 2} = \frac{1}{2} \text{ or } 0.5.$$

Cosine Similarity

- Cosine similarity which is one of the most popular measures in text classification is calculated as: $$cos\,(x,\ y) = \frac{x.y}{\|x\|.\|y\|}$$

- where, x.y = vector dot product of x and y = $\sum_{i=1}^{n} x_i y_i$

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2} \text{ and } \|y\| = \sqrt{\sum_{i=1}^{n} y_i^2}$$



Cosine Similarity...

- x = (2, 4, 0, 0, 2, 1, 3, 0, 0) and
- y = (2, 1, 0, 0, 3, 2, 1, 0, 1).
- In this case, x.y = 2*2 + 4*1 + 0*0 + 0*0 + 2*3 + 1*2 +3*1 + 0*0 + 0*1 = 19

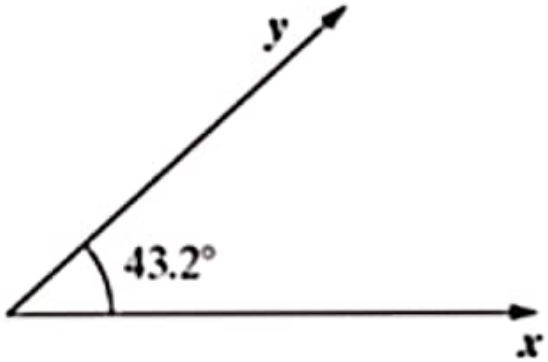
$$\|x\| = \sqrt{2^2 + 4^2 + 0^2 + 0^2 + 2^2 + 1^2 + 3^2 + 0^2 + 0^2} = \sqrt{34} = 5.83$$

$$\|y\| = \sqrt{2^2 + 1^2 + 0^2 + 0^2 + 3^2 + 2^2 + 1^2 + 0^2 + 1^2} = \sqrt{20} = 4.47$$

$$\therefore cos\,(x, y) = \frac{19}{5.83 * 4.47} = 0.729$$

Cosine Similarity...

- Cosine similarity actually measures the angle between x and y vectors.
- Hence, if cosine similarity has a value 1, the angle between x and y is 0° which means x and y are same except for the magnitude.
- If cosine similarity is 0, the angle between x and y is 90°.
- Hence, they do not share any similarity.
- In the above example, the angle comes to be 43.2°.



Machine Learning

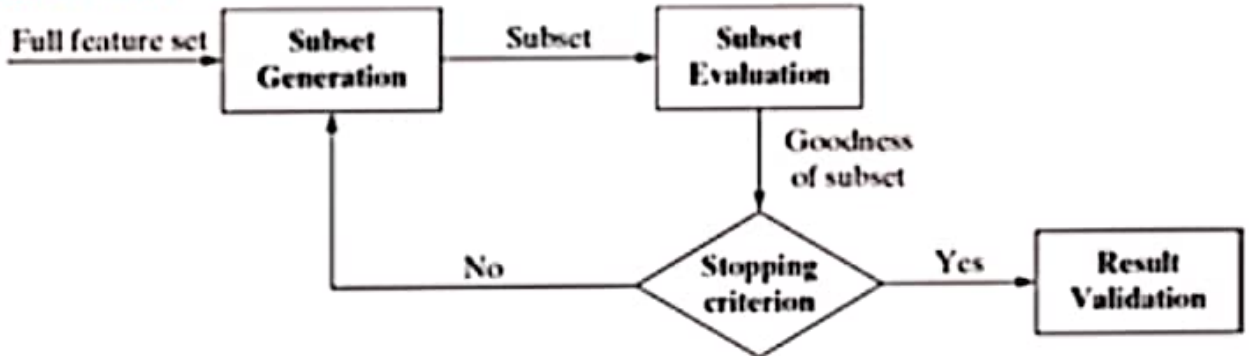
Subject Code: 20A05602T

UNIT 2 –Basics of Feature Engineering

FEATURE SUBSET SELECTION – Part-3

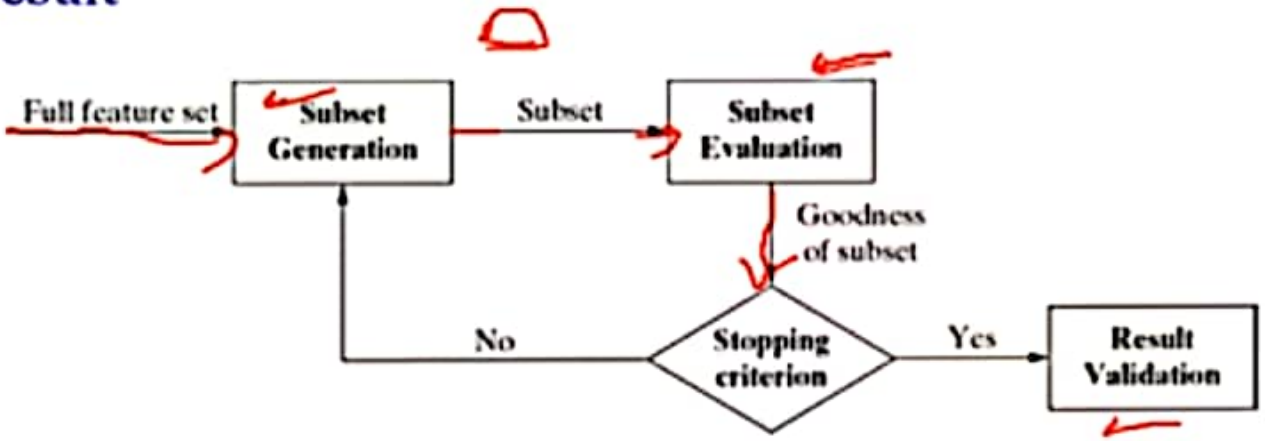
- Issues in high-dimensional data
- Key drivers of feature selection – feature relevance and redundancy
- Measures of feature relevance and redundancy
- **Overall feature selection process**
  - Feature selection approaches

1. Filter approach
2. Wrapper approach
3. Hybrid approach
4. Embedded approach



Overall feature selection process

- Feature selection is the process of selecting a subset of features in a data set. A typical feature selection process consists of four steps:
- 1. generation of possible subsets
- 2. subset evaluation
- 3. stop searching based on some stopping criterion
- 4. validation of the result



Subset Generation

- **Subset generation**, which is the first step of any feature selection algorithm, is a search procedure which ideally should produce all possible candidate subsets.
- different approximate search strategies are employed to find candidate subsets for evaluation.
  1. the search may start with an empty set and keep adding features - forward selection.
  2. a search may start with a full set and successively remove features - backward elimination.
  3. search start with both ends and add and remove features simultaneously - bi-directional selection,

Evaluation Criterion and Stopping Criterion

- Each candidate subset is then evaluated and compared with the previous best performing subset based on certain evaluation criterion.
- If the new subset performs better, it replaces the previous one.
- This cycle of subset generation and evaluation continues till a pre-defined stopping criterion is fulfilled.
- Some commonly used stopping criteria are
  - 1. the search completes
  - 2. some given bound (e.g. a specified number of iterations) is reached
  - 3. subsequent addition (or deletion) of the feature is not producing a better subset
  - 4. a sufficiently good subset (e.g. a subset having better classification accuracy than the existing benchmark) is selected

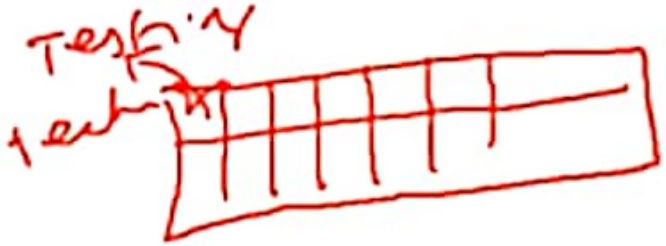
Validation

- Then the selected best subset is validated, either against prior benchmarks or by experiments.
- In case of supervised learning, the accuracy of the learning model may be the performance parameter considered for validation.
- The accuracy of the model using the subset derived is compared with accuracy of some other benchmark algorithm.
- In case of unsupervised learning, the cluster quality may be the parameter for validation.

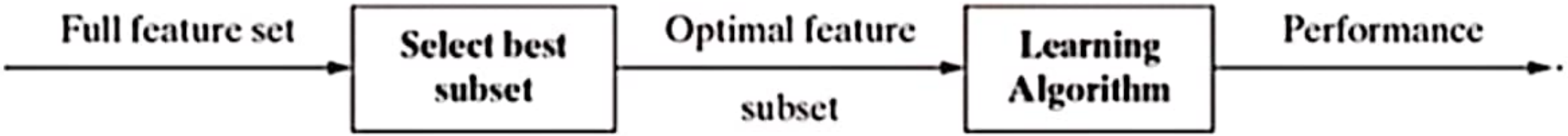
Feature selection approaches

- There are four types of approach for feature selection:
- 1. Filter approach
- 2. Wrapper approach
- 3. Hybrid approach
- 4. Embedded approach

Filter Approach

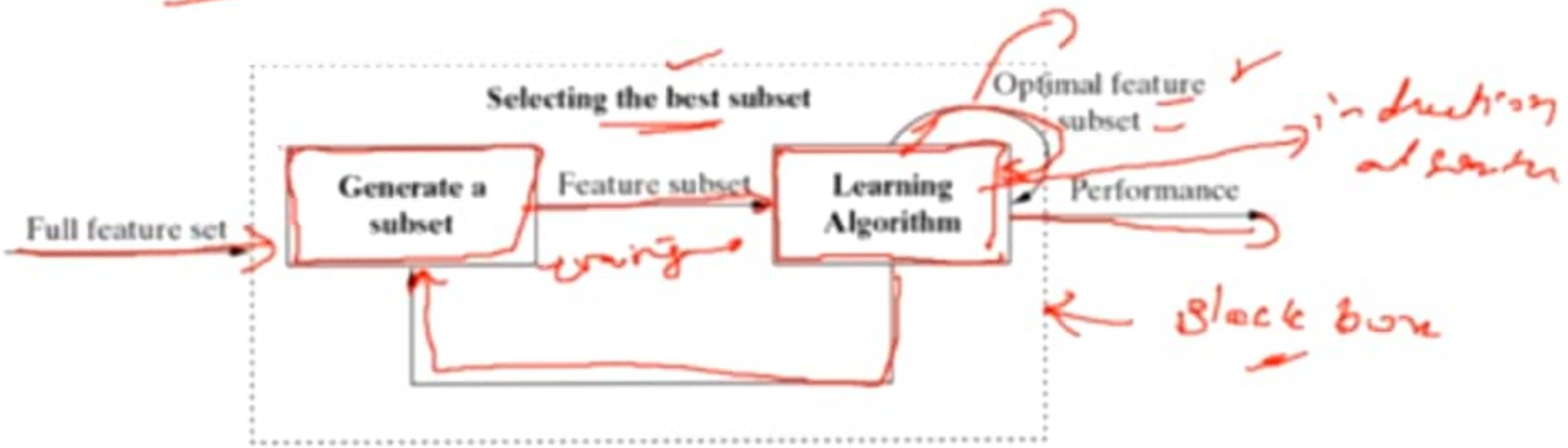


- the feature subset is selected based on statistical measures done to assess the merits of the features from the data perspective.
- No learning algorithm is to evaluate the goodness of the feature selected.
- Some of the common statistical tests conducted on features as a part of filter approach are –
- Pearson's correlation, information gain, Fisher score, analysis of variance (ANOVA), Chi-Square, etc.



Wrapper Approach

- Identification of best feature subset is done using the induction algorithm as a black box.
- The feature selection algorithm searches for a good feature subset using the induction algorithm itself as a part of the evaluation function.
- For every candidate subset, the learning model is trained and the result is evaluated by running the learning algorithm.
- wrapper approach is computationally very expensive.
- the performance is superior compared to filter approach



Hybrid approach

- Hybrid approach takes the advantage of both filter and wrapper approaches.
- A typical hybrid algorithm makes use of both the statistical tests as used in filter approach to decide the best subsets for a given cardinality and
- a learning algorithm to select the final best subset among the best subsets across different cardinalities.

Feature selection approaches...

- Embedded approach is quite similar to wrapper approach as it also uses and inductive algorithm to evaluate the generated feature subsets.
- However, the difference is it performs feature selection and classification simultaneously.