MACHINE LEARNING

UNIT -I:The ingredients of machine learning, Tasks: the problems that can be solved with Machine learning, Models: the output of machine learning, Features, the workhorses of machine learning.

Binary classification and related tasks: Classification, Scoring and ranking, Class probability estimation

A)Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that which makes it more similar to humans: ***The ability to learn***. Machine learning is actively being used today, perhaps in many more places than one would expect

I-THE PROBLEMS THAT CAN BE SOLVED WITH MACHINE LEARNING:

The most common machine learning tasks are predictive, in the sense that they concern predicting a target variable from features.

Binary and multi-class classification: categorical target

Regression: numerical target

Clustering: hidden target

Descriptive tasks are concerned with exploiting underlying structure in the data

8 PROBLEMS SOLVED BY MACHINE LEARNING

- Manual data entry.
- Detecting Spam.
- Product recommendation.
- Medical Diagnosis.
- Customer segmentation and Lifetime value prediction.
- Financial analysis.
- Predictive maintenance.
- Image recognition (Computer Vision).

1. MANUAL DATA ENTRY

Inaccuracy and duplication of data are major business problems for an organization wanting to automate its processes. Machines learning (ML) algorithms and predictive modelling algorithms can significantly improve the situation. ML programs use the discovered data to improve the process as more calculations are made. Thus machines can learn to perform time-intensive documentation and data entry tasks. Also, knowledge workers can now spend more time on higher-value problem-solving tasks. Arria, an AI based firm has developed a natural language processing technology which scans texts and determines the relationship between concepts to write reports.

2. DETECTING SPAM

Spam detection is the earliest problem solved by ML. Four years ago, email service providers used pre-existing rule-based techniques to remove spam. But now the spam filters create new rules themselves using ML. Thanks to 'neural networks' in its spam filters, Google now boasts of 0.1 percent of spam rate. Brain-like "neural networks" in its spam filters can learn to recognize junk mail and phishing messages by analyzing rules across an enormous collection of computers. In addition to spam detection, social media websites are using ML as a way to identify and filter abuse.

3. PRODUCT RECOMMENDATION

Unsupervised learning enables a product based recommendation system. Given a purchase history for a customer and a large inventory of products, ML models can identify those products in which that customer will be interested and likely to purchase. The algorithm identifies hidden pattern among items and focuses on grouping similar products into clusters. A model of this decision process would allow a program to make recommendations to a customer and motivate product purchases. E-Commerce businesses such as Amazon has this capability. Unsupervised learning along with location detail is used by Facebook to recommend users to connect with others users.

4.MEDICAL DIAGNOSIS

Machine Learning in the medical field will improve patient's health with minimum costs. Use cases of ML are making near perfect diagnoses, recommend best medicines, predict readmissions and identify high-risk patients. These predictions are based on the dataset of anonymized patient records and symptoms exhibited by a patient. Adoption of ML is happening at a rapid pace despite many hurdles, which can be overcome by practitioners and consultants who know the legal, technical, and medical obstacles.

5. CUSTOMER SEGMENTATION AND LIFETIME VALUE PREDICTION

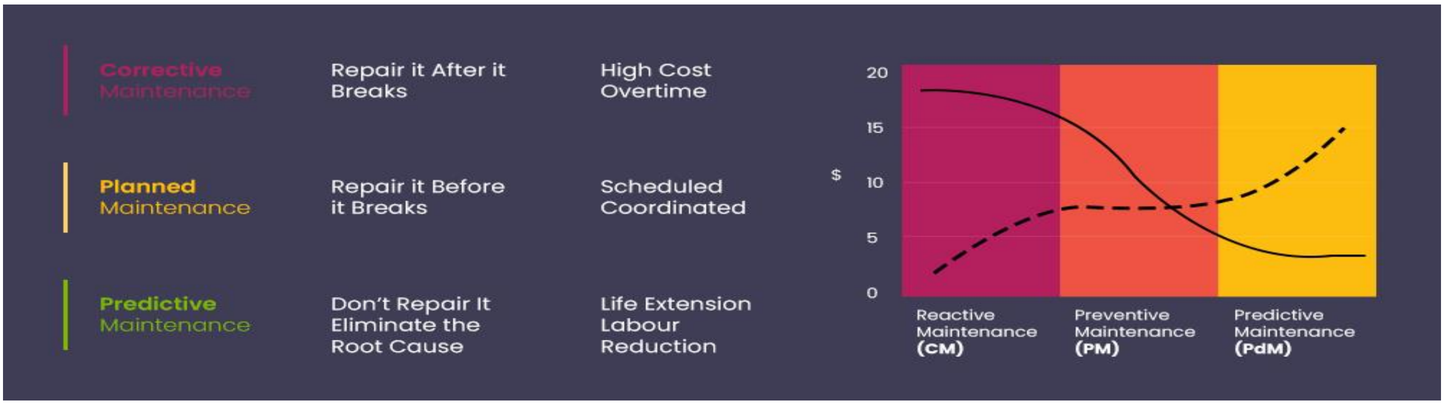
Customer segmentation, churn prediction and customer lifetime value (LTV) prediction are the main challenges faced by any marketer. Businesses have a huge amount of marketing relevant data from various sources such as email campaign, website visitors and lead data. Using data mining and machine learning, an accurate prediction for individual marketing offers and incentives can be achieved. Using ML, savvy marketers can eliminate guesswork involved in data-driven marketing. For example, given the pattern of behavior by a user during a trial period and the past behaviors of all users, identifying chances of conversion to paid version can be predicted. A model of this decision problem would allow a program to trigger customer interventions to persuade the customer to convert early or better engage in the trial.

6. FINANCIAL ANALYSIS

Due to large volume of data, quantitative nature and accurate historical data, machine learning can be used in financial analysis. Present use cases of ML in finance includes algorithmic trading, portfolio management, fraud detection and loan underwriting. According to Ernst and Young report on 'The future of underwriting' – Machine learning will enable continual assessments of data for detection and analysis of anomalies and nuances to improve the precision of models and rules. And machines will replace a large no. of underwriting positions. Future applications of ML in finance include_chatbots and conversational interfaces for customer service, security and sentiment analysis.

7. PREDICTIVE MAINTENANCE

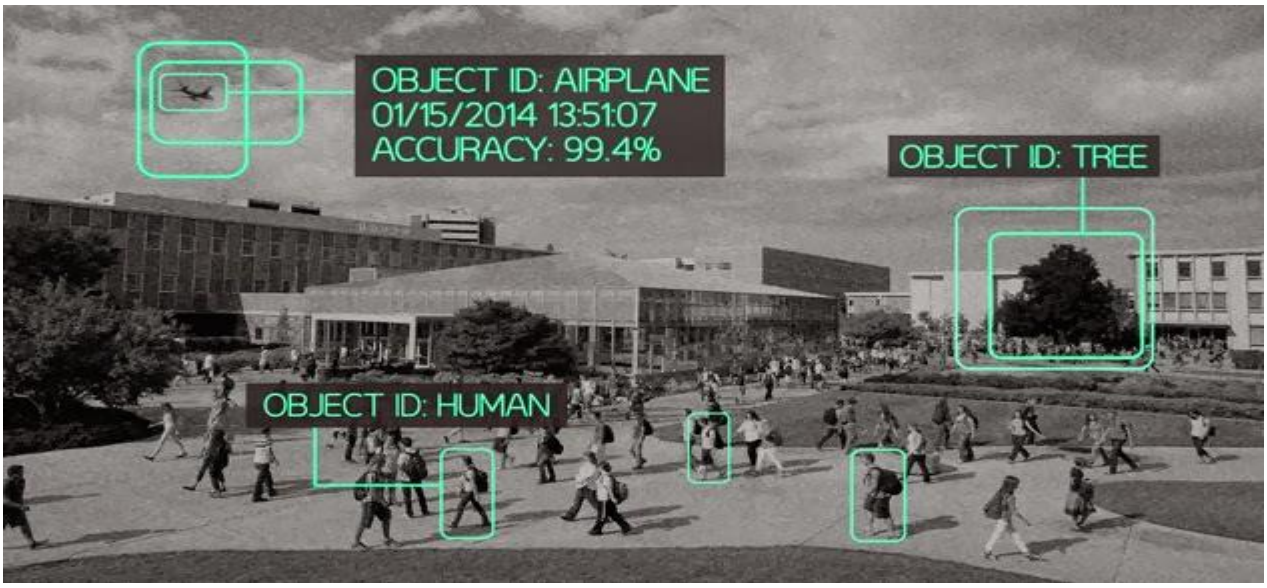
Manufacturing industry can use artificial intelligence (AI) and ML to discover meaningful patterns in factory data. Corrective and preventive maintenance practices are costly and inefficient. Whereas predictive maintenance minimizes the risk of unexpected failures and reduces the amount of unnecessary preventive maintenance activities.



Corrective Maintenance	Repair it After it Breaks	High Cost Overtime
Planned Maintenance	Repair it Before it Breaks	Scheduled Coordinated
Predictive Maintenance	Don't Repair It Eliminate the Root Cause	Life Extension Labour Reduction

8. IMAGE RECOGNITION (COMPUTER VISION)

Computer vision produces numerical or symbolic information from images and high-dimensional data. It involves machine learning, data mining, database knowledge discovery and pattern recognition. Potential business uses of image recognition technology are found in healthcare, automobiles – driverless cars, marketing campaigns, etc. Baidu has developed a prototype of DuLight for visually impaired which incorporates computer vision technology to capture surrounding and narrate the interpretation through an earpiece. Image recognition based marketing campaigns such as Makeup Genius by L'Oreal drive social sharing and user engagement.

- 

II-MODELS: THE OUTPUT OF MACHINE LEARNING, FEATURES, THE WORKHORSES OF MACHINE LEARNING.

MODELS: THE OUTPUT OF MACHINE LEARNING

1.Geometric models

2.Probabilistic models

3.Logical models Grouping and grading

Machine learning models can be distinguished according to their main intuition:

Geometric models :use intuitions from geometry such as separating (hyper-)planes, linear transformations and distance metrics.

Probabilistic models: view learning as a process of reducing uncertainty, modelled by means of probability distributions.

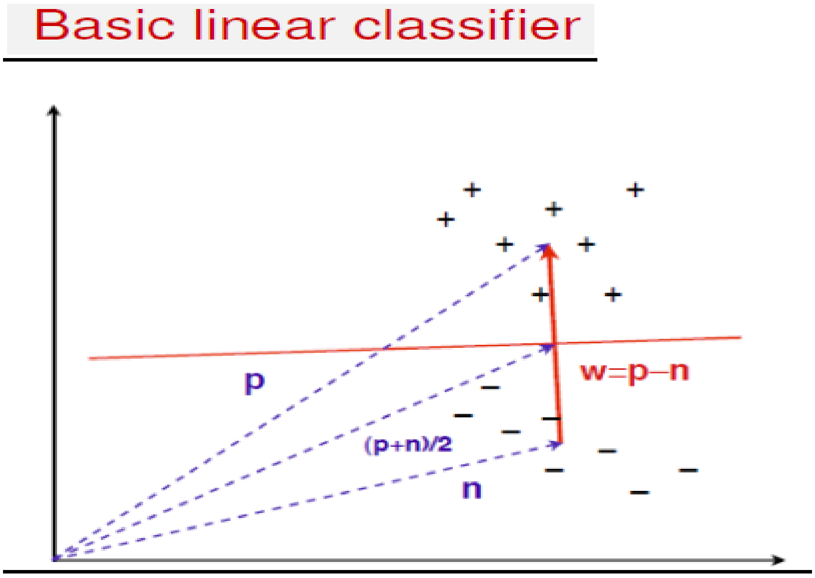
Logical models are defined in terms of easily interpretable logical expressions.

Alternatively, they can be characterised by their modus operandi:

Grouping models divide the instance space into segments; in each segment a very simple (e.g., constant) model is learned.

Grading models learning a single, global model over the instance space

1.Geometric models



Basic linear classifier

The basic linear classifier constructs a decision boundary by half-way intersecting the line between the positive and negative centres of mass. It is described by the equation

$\mathbf{w}\cdot\mathbf{x} = t$, with $\mathbf{w} = \mathbf{p} - \mathbf{n}$; the decision threshold can be found by noting that ($\mathbf{p}$+$\mathbf{n}$)/2 is on the decision boundary, and $t = (\mathbf{p}-\mathbf{n})\cdot(\mathbf{p}+\mathbf{n})/2 = (||\mathbf{p}||^2 - ||\mathbf{n}||^2)/2$, where ||$\mathbf{x}$|| denotes the length of vector $\mathbf{x}$.

2.Probabilistic models

- A model describes data that one could observe from a system
- If we use the mathematics of probability theory to express all forms of uncertainty and Noise associated with our model...
- Then inverse probability (i.e. Bayes rule) allows us to infer unknown quantities, adapt our models, Make predictions and learn from data

Bayes Rule:

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})}$$

- Bayes rule tells us how to do inference about hypotheses from data.
- Learning and prediction can be seen as forms of inference.

3.Logical models

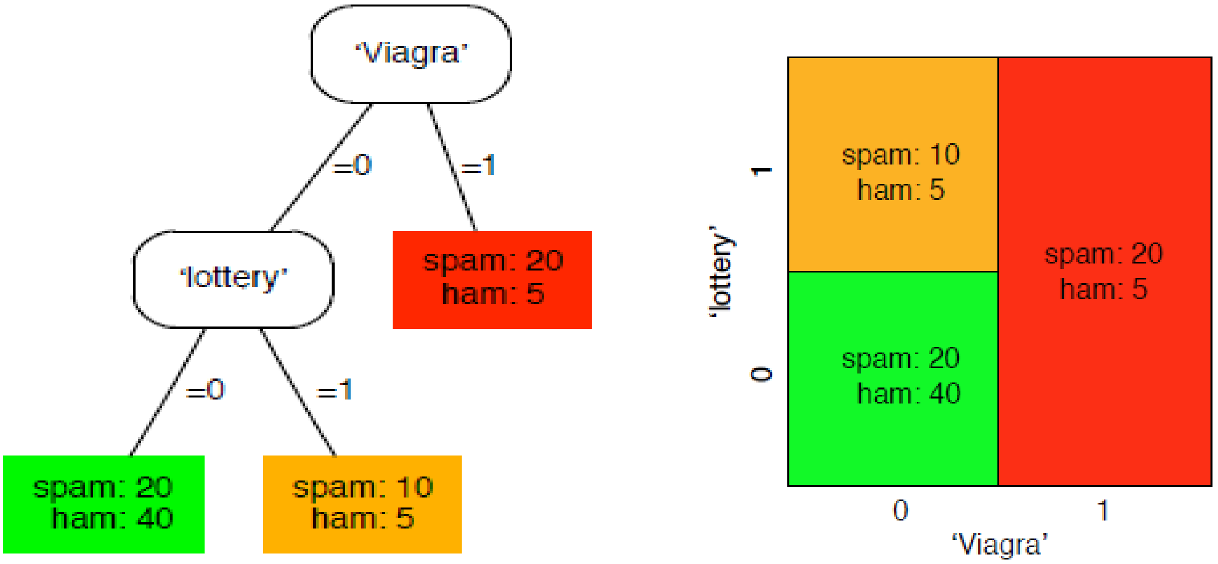


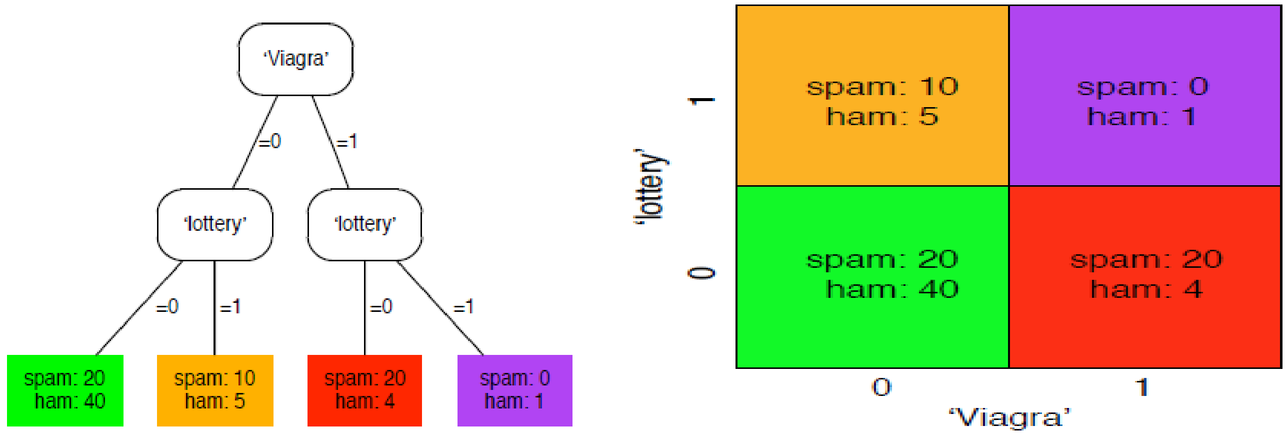
Fig:1:4-Feature Tree for Logical models

Labelling a feature tree

(left) A feature tree combining two Boolean features. Each internal node or split is labelled with a feature, and each edge emanating from a split is labelled with a feature value. Each leaf therefore corresponds to a unique combination of feature values. Also indicated in each leaf is the class distribution derived from the training set. (right) A feature tree partitions the instance space into rectangular regions, one for each leaf. We can clearly see that the majority of ham lives in the lower left-hand corner.

- The leaves of the tree in Figure 1.4 could be labelled, from left to right, as ham – spam – spam, employing a simple decision rule called majority class.
- Alternatively, we could label them with the proportion of spam e-mail occurring in each leaf: from left to right, 1/3, 2/3, and 4/5.
- Or, if our task was a regression task, we could label the leaves with predicted real values or even linear functions of some other, real-valued features.

A complete feature tree



Consider the following rules:

·if lottery = 1 **then** Class = Y = spam·

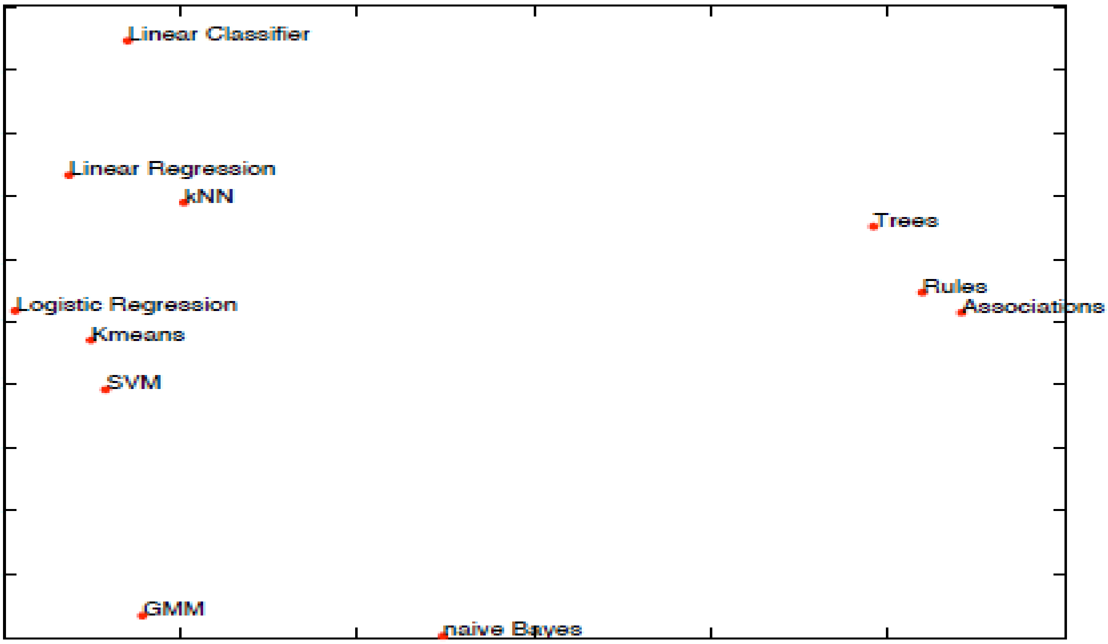
·if Peter = 1 **then** Class = Y = ham·

4.Grouping and Grading models

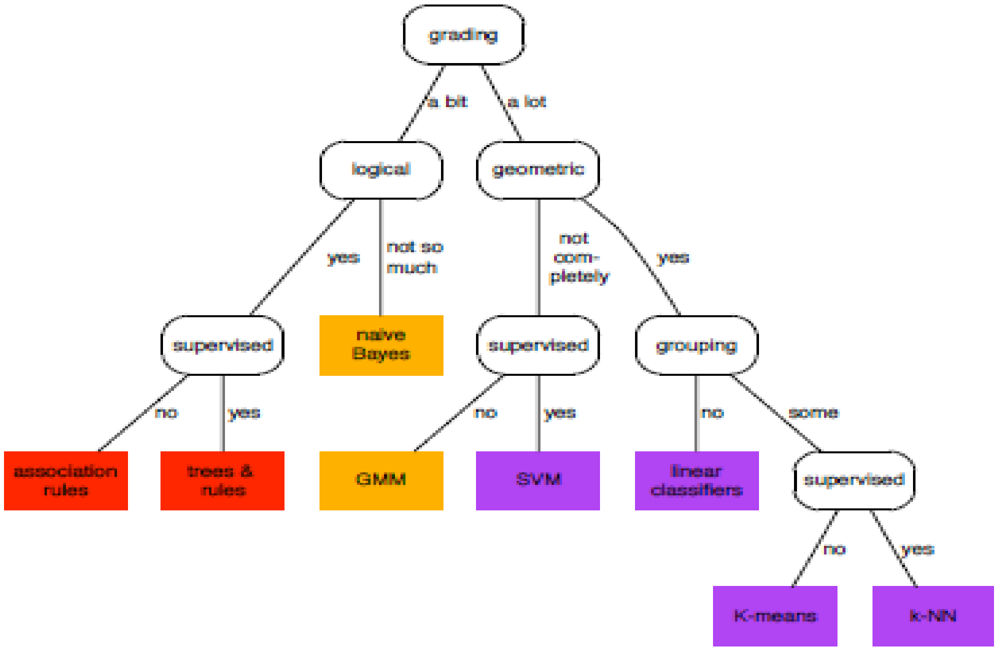
Mapping machine learning models:

A 'map' of some of the models that will be considered in this book. Models that share characteristics are plotted closer together: logical models to the right, geometric models on the top left and probabilistic models on the bottom left. The horizontal dimensiroughly ranges from grading models on the left to grouping models on the right.

Mapping machine learning model diagram



ML taxonomy diagram



A taxonomy describing machine learning methods in terms of the extent to which they are grading or grouping models, logical, geometric or a combination, and supervised or unsupervised. The colours indicate the type of model, from left to right: logical (red), probabilistic (orange) and geometric (purple).

III-FEATURES :-THE WORKHORSES OF MACHINE LEARNING

Suppose we have a number of learning models that we want to describe in terms of a number of properties:

→The extent to which the models are geometric, probabilistic or logical;

→Whether they are grouping or grading models;

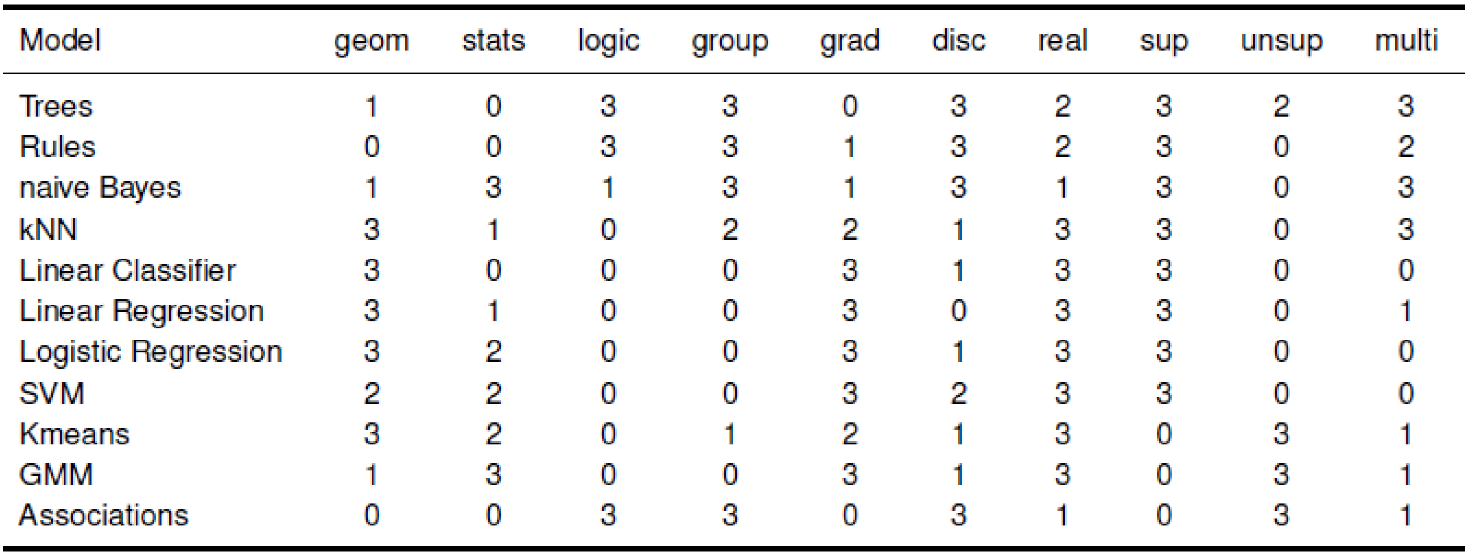
→The extent to which they can handle discrete and/or real-valued features;

→Whether they are used in supervised or unsupervised learning; and

→The extent to which they can handle multi-class problems.

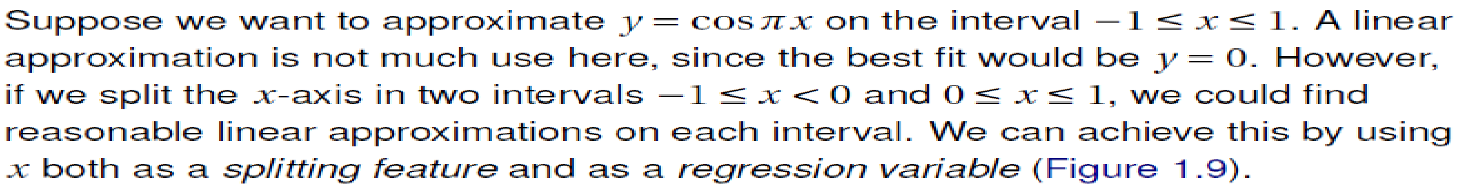
The first two properties could be expressed by discrete features with three and two values, respectively; or if the distinctions are more gradual, each aspect could be rated on some numerical scale. A simple approach would be to measure each property on an integer scale from 0 to 3, as in Table 1.4. This table establishes a data set in which each row represents an instance and each column a feature

Table 1.4. **THE MLM DATA SET**



Model	geom	stats	logic	group	grad	disc	real	sup	unsup	multi
Trees	1	0	3	3	0	3	2	3	2	3
Rules	0	0	3	3	1	3	2	3	0	2
naive Bayes	1	3	1	3	1	3	1	3	0	3
kNN	3	1	0	2	2	1	3	3	0	3
Linear Classifier	3	0	0	0	3	1	3	3	0	0
Linear Regression	3	1	0	0	3	0	3	3	0	1
Logistic Regression	3	2	0	0	3	1	3	3	0	0
SVM	2	2	0	0	3	2	3	3	0	0
Kmeans	3	2	0	1	2	1	3	0	3	1
GMM	1	3	0	0	3	1	3	0	3	1
Associations	0	0	3	3	0	3	1	0	3	1

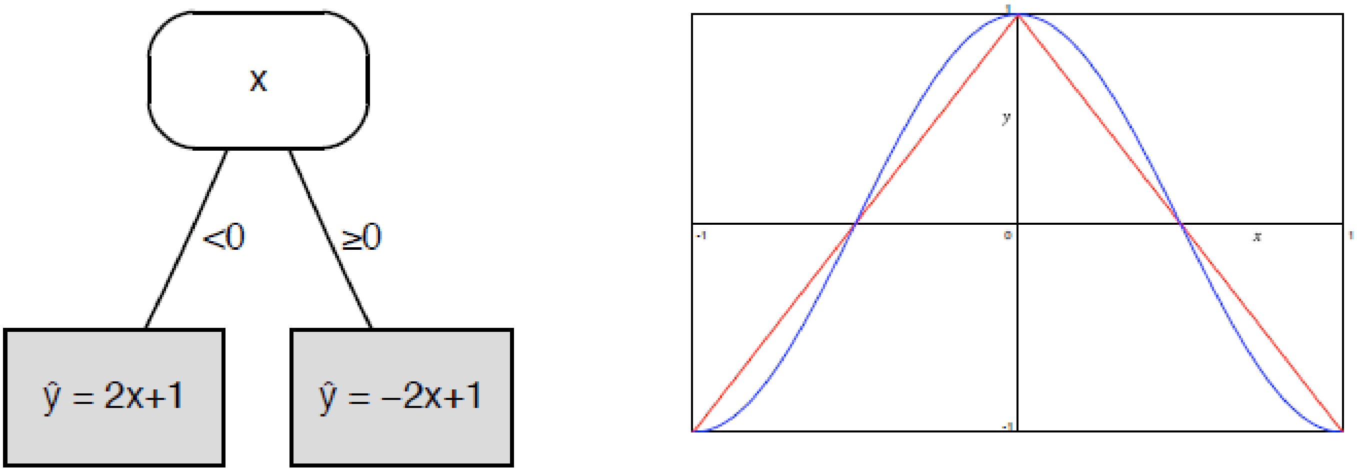
THE MANY USES OF FEATURES:



Suppose we want to approximate $y = \cos \pi x$ on the interval $-1 \leq x \leq 1$. A linear approximation is not much use here, since the best fit would be $y = 0$. However, if we split the x-axis in two intervals $-1 \leq x < 0$ and $0 \leq x \leq 1$, we could find reasonable linear approximations on each interval. We can achieve this by using x both as a *splitting feature* and as a *regression variable* (Figure 1.9).

A small regression tree

FIG:DIAGRAM FORREGRESSION TREE

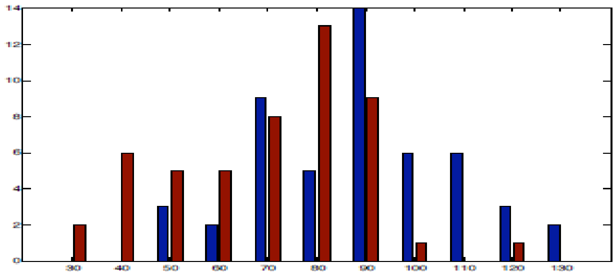


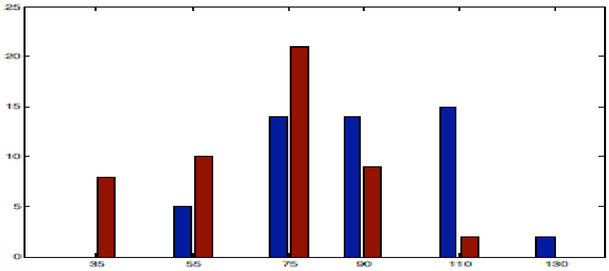
(left) A regression tree combining a one-split feature tree with linear regression models in the leaves. Notice how *x* is used as both a splitting feature and a regression variable.
(right) The function $y=COS\pi x$ on the interval -1≤X≤+1, and the piecewise linear approximation achieved by the regression tree.

Feature construction and transformation

Class-sensitive discretisation:

(left) Artificial data depicting a histogram of body weight measurements of people with (blue) and without (red) diabetes, with eleven fixed intervals of 10 kilograms width each. **(right)** By joining the first and second, third and fourth, fifth and sixth, and the eighth,ninth and tenth intervals, we obtain a discretisation such that the proportion of diabetes cases increases from left to right. This discretisation makes the feature more useful in predicting diabetes.





B)BINARY CLASSIFICATION AND RELATED TASKS

Binary classification and related tasks are

- **Classification**
  -Assessing classification performance
  -Visualising classification performance

- **Scoring and ranking**
  -Assessing and visualising ranking performance
  -Tuning rankers
- **Class probability estimation**
  -Assessing class probability estimates

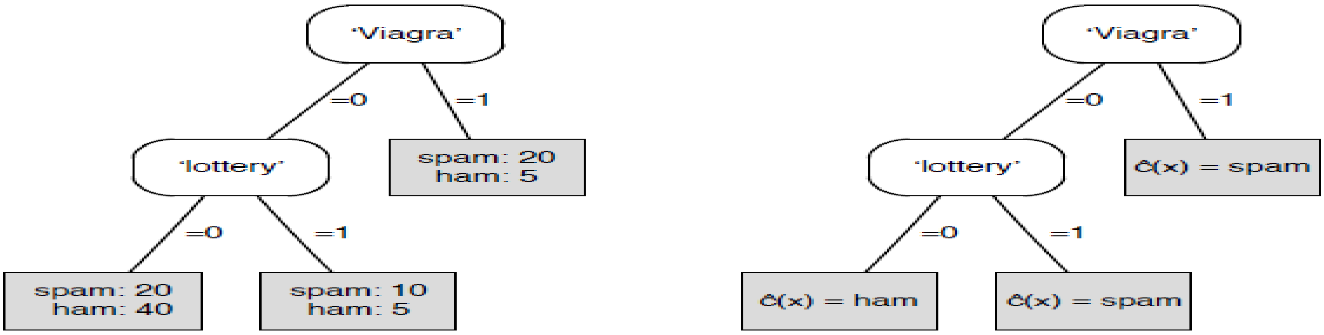
1) CLASSIFICATION

A *classifier* is a mapping $\hat{c} : \mathscr{X} \to \mathscr{C}$, where $\mathscr{C} = \{C_1, C_2, \ldots, C_k\}$ is a finite and usually small set of *class labels*. We will sometimes also use C_i to indicate the set of examples of that class.

We use the 'hat' to indicate that $\hat{c}(x)$ is an estimate of the true but unknown function $c(x)$. Examples for a classifier take the form $(x, c(x))$, where $x \in \mathscr{X}$ is an instance and $c(x)$ is the true class of the instance (sometimes contaminated by noise).

Learning a classifier involves constructing the function $\hat{c}$ such that it matches c as closely as possible (and not just on the training set, but ideally on the entire instance space $\mathscr{X}$).

Decision Tree Diagram For Classification:



(left) A feature tree with training set class distribution in the leaves. **(right)** A decision tree obtained using the majority class decision rule.

→Assessing classification performance

(left) A two-class contingency table or confusion matrix depicting the performance of the decision tree in Figure 2.1. Numbers on the descending diagonal indicate correct predictions, while the ascending diagonal concerns prediction errors. **(right)** Acontingency table with the same marginals but independent rows and columns.

Contingency table

	Predicted ⊕	*Predicted* ⊖	
Actual ⊕	**30**	**20**	50
Actual ⊖	**10**	**40**	50
	40	60	100

	⊕	⊖	
⊕	**20**	**30**	50
⊖	**20**	**30**	50
	40	60	100

→Visualising classification performance:

Degrees of freedom for above topic

contains 9 values, however some of them depend on others: e.g., marginal sums depend on rows and columns, respectively. Actually, we need only 4 values to determine the rest of them. Thus, we say that this table has **4 degrees of freedom**. In general table having (*k* Å1)2 entries has *k*2 degrees of freedom.In the following, we assume that *Pos*, *Neg* , **TP** and **FP** are enough to reconstruct whole table.

The following contingency table:

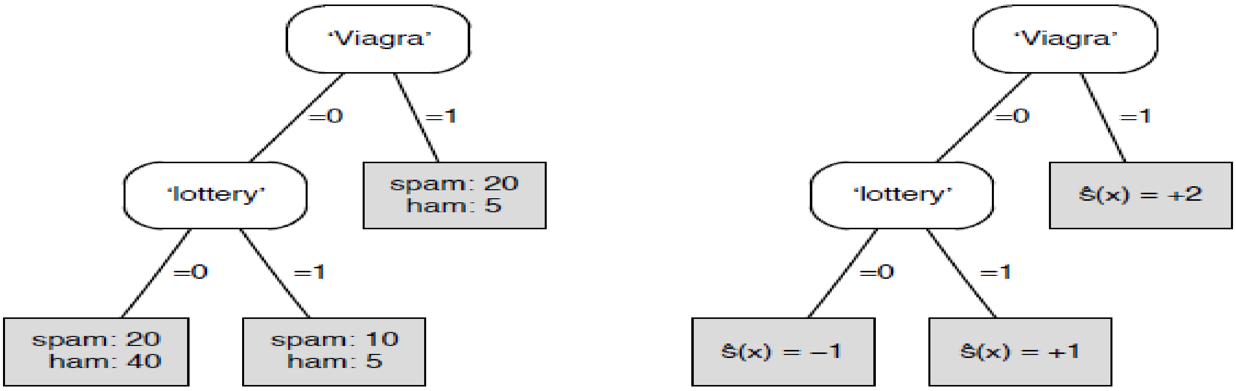
	Predicted ⊕	*Predicted* ⊖	
Actual ⊕	TP	FN	Pos
Actual ⊖	FP	TN	Neg
	0	0	0

2) SCORING AND RANKING

A *scoring classifier* is a mapping $\hat{\mathbf{s}}: \mathscr{X} \rightarrow \mathbb{R}^k$, i.e., a mapping from the instance space to a k-vector of real numbers.
The boldface notation indicates that a scoring classifier outputs a vector $\hat{\mathbf{s}}(x) = (\hat{s}_1(x), \ldots, \hat{s}_k(x))$ rather than a single number; $\hat{s}_i(x)$ is the score assigned to class C_i for instance x.
This score indicates how likely it is that class label C_i applies.

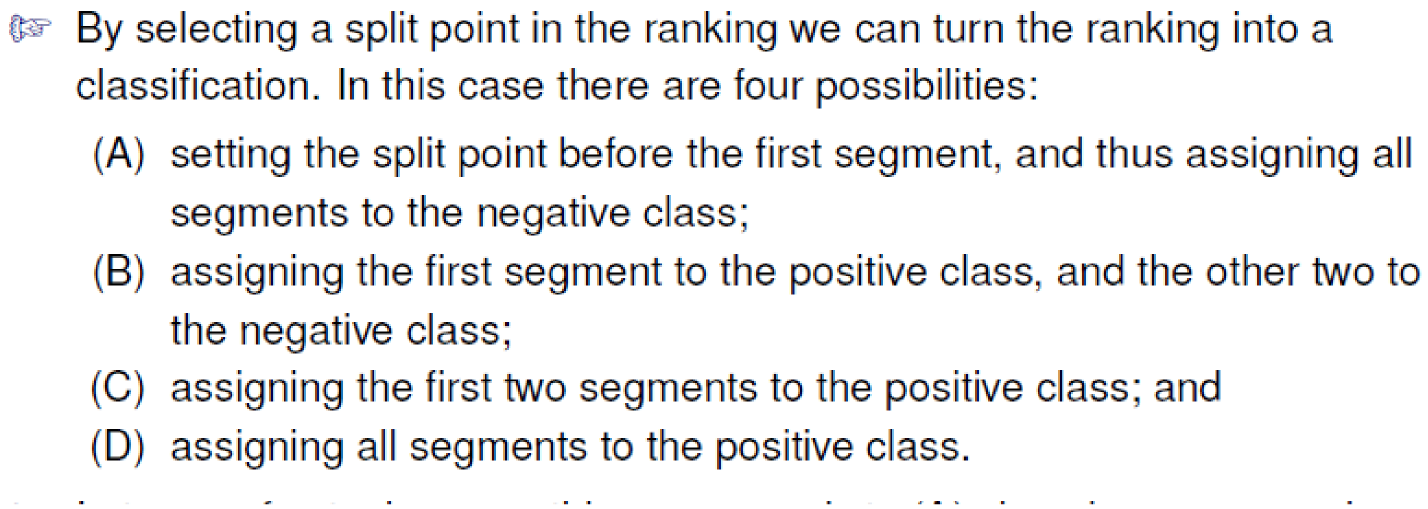
If we only have two classes, it usually suffices to consider the score for only one of the classes; in that case, we use $\hat{s}(x)$ to denote the score of the positive class for instance x.

SCORING TREE



(left) A feature tree with training set class distribution in the leaves.
(right) A scoring tree using the logarithm of the class ratio as scores; spam is taken as the positive class.

→ Assessing and visualising ranking performance:



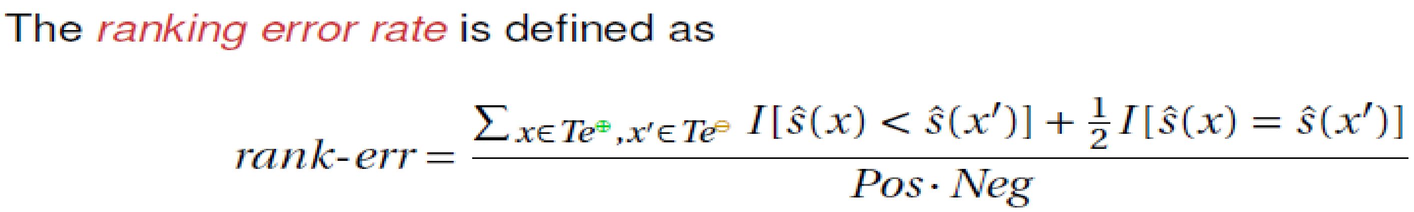
☞ By selecting a split point in the ranking we can turn the ranking into a classification. In this case there are four possibilities:

(A) setting the split point before the first segment, and thus assigning all segments to the negative class;

(B) assigning the first segment to the positive class, and the other two to the negative class;

(C) assigning the first two segments to the positive class; and

(D) assigning all segments to the positive class.



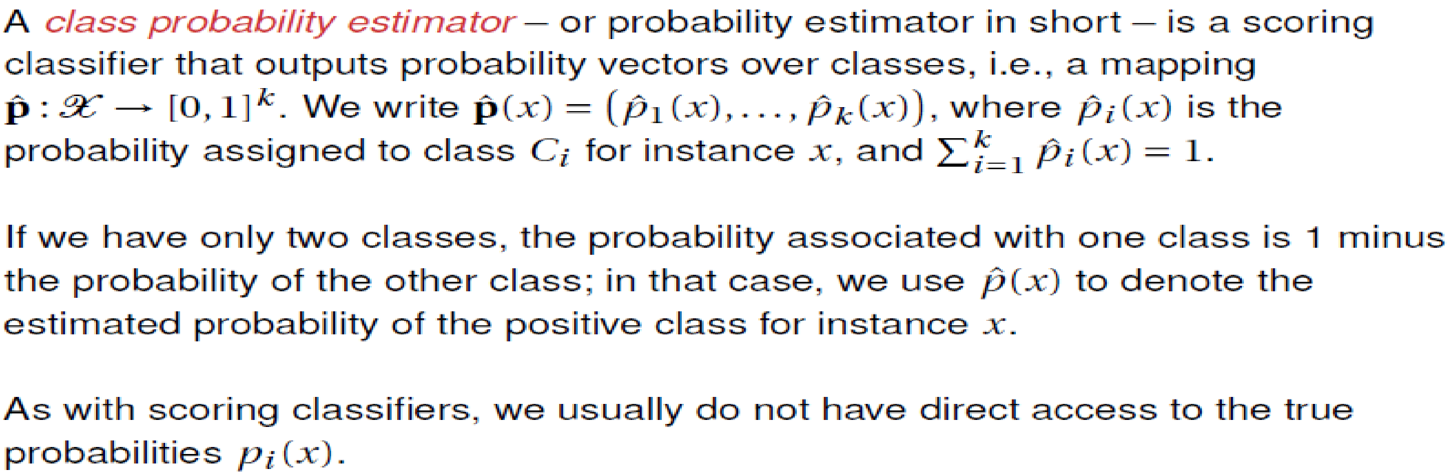
The *ranking error rate* is defined as

$$rank\text{-}err = \frac{\sum_{x\in Te^{\oplus}, x'\in Te^{\ominus}} I[\hat{s}(x) < \hat{s}(x')] + \frac{1}{2} I[\hat{s}(x) = \hat{s}(x')]}{Pos \cdot Neg}$$

→Tuning rankers:

You have carefully trained your Bayesian spam filter, and all that remains is setting the decision threshold. You select a set of six spam and four ham e-mails and collect the scores assigned by the spam filter. Sorted on decreasing score these are 0.89 (spam), 0.80 (spam), 0.74 (ham), 0.71 (spam), 0.63 (spam), 0.49 (ham), 0.42 (spam), 0.32 (spam), 0.24 (ham), and 0.13 (ham).

3.CLASS PROBABILITY ESTIMATION

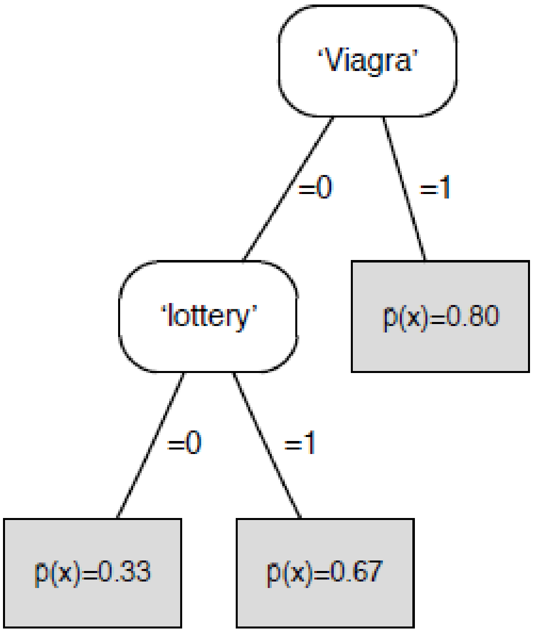


A *class probability estimator* – or probability estimator in short – is a scoring classifier that outputs probability vectors over classes, i.e., a mapping $\hat{\mathbf{p}} : \mathscr{X} \rightarrow [0,1]^k$. We write $\hat{\mathbf{p}}(x) = \big(\hat{p}_1(x), \ldots, \hat{p}_k(x)\big)$, where $\hat{p}_i(x)$ is the probability assigned to class C_i for instance x, and $\sum_{i=1}^{k} \hat{p}_i(x) = 1$.

If we have only two classes, the probability associated with one class is 1 minus the probability of the other class; in that case, we use $\hat{p}(x)$ to denote the estimated probability of the positive class for instance x.

As with scoring classifiers, we usually do not have direct access to the true probabilities $p_i(x)$.

PROBABILTIY ESTIMATION TREE:



A probability estimation tree derived from the feature tree in Figure 1.4.

→Assessing class probability estimates:

It requires mean and squared probability form is

We can define the *squared error* (*SE*) of the predicted probability vector $\hat{\mathbf{p}}(x) = (\hat{p}_1(x), \ldots, \hat{p}_k(x))$ as

$$\mathrm{SE}(x) = \frac{1}{2} \sum_{i=1}^{k} (\hat{p}_i(x) - I[c(x) = C_i])^2$$

and the *mean squared error* (*MSE*) as the average squared error over all instances in the test set:

$$\mathrm{MSE}(Te) = \frac{1}{|Te|} \sum_{x \in Te} \mathrm{SE}(x)$$