VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

Department of Computer Science and Engineering

IV B.Tech – II Semester (Sections – C & D)

Machine Learning

UNIT-I – The ingredients of machine learning, Tasks – Sessions-1&2

Syllabus

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

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.

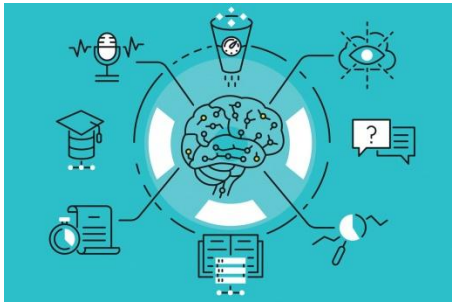
Binary classification and related tasks: Classification, Scoring and ranking, Class probability estimation

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Introduction To Machine Learning

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Background



1980
Mainframes to PC

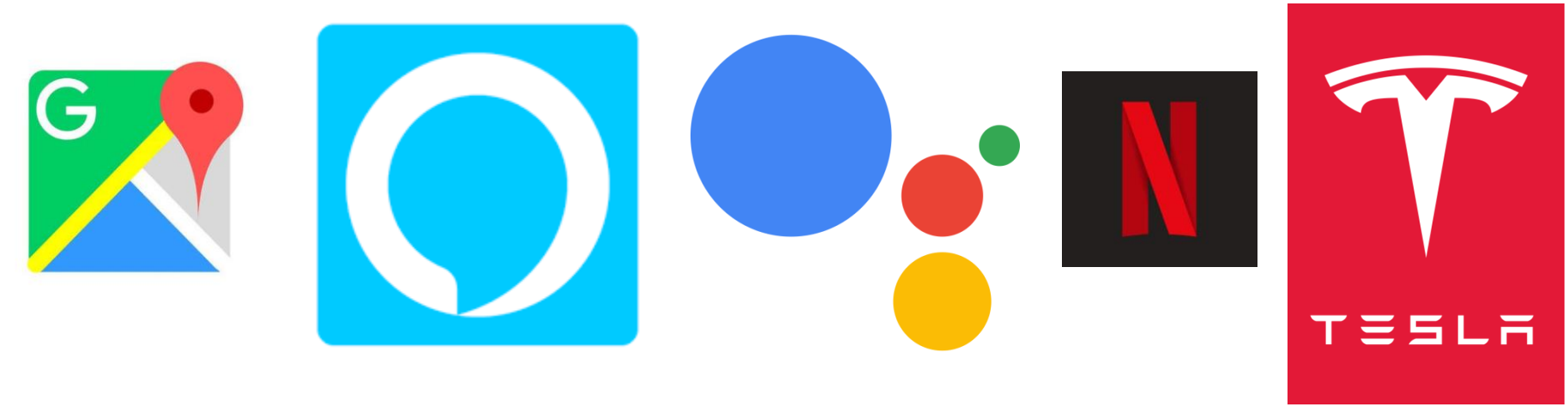
1990
Web-Intraction &Connection

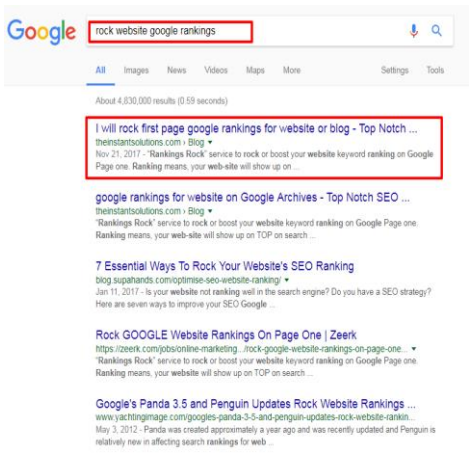
2000
Movable revolution
(Computing to half the world)

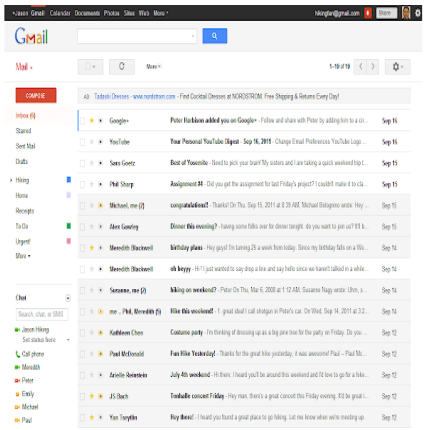
After 2000
Movable revolution to Intelligence

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Applications

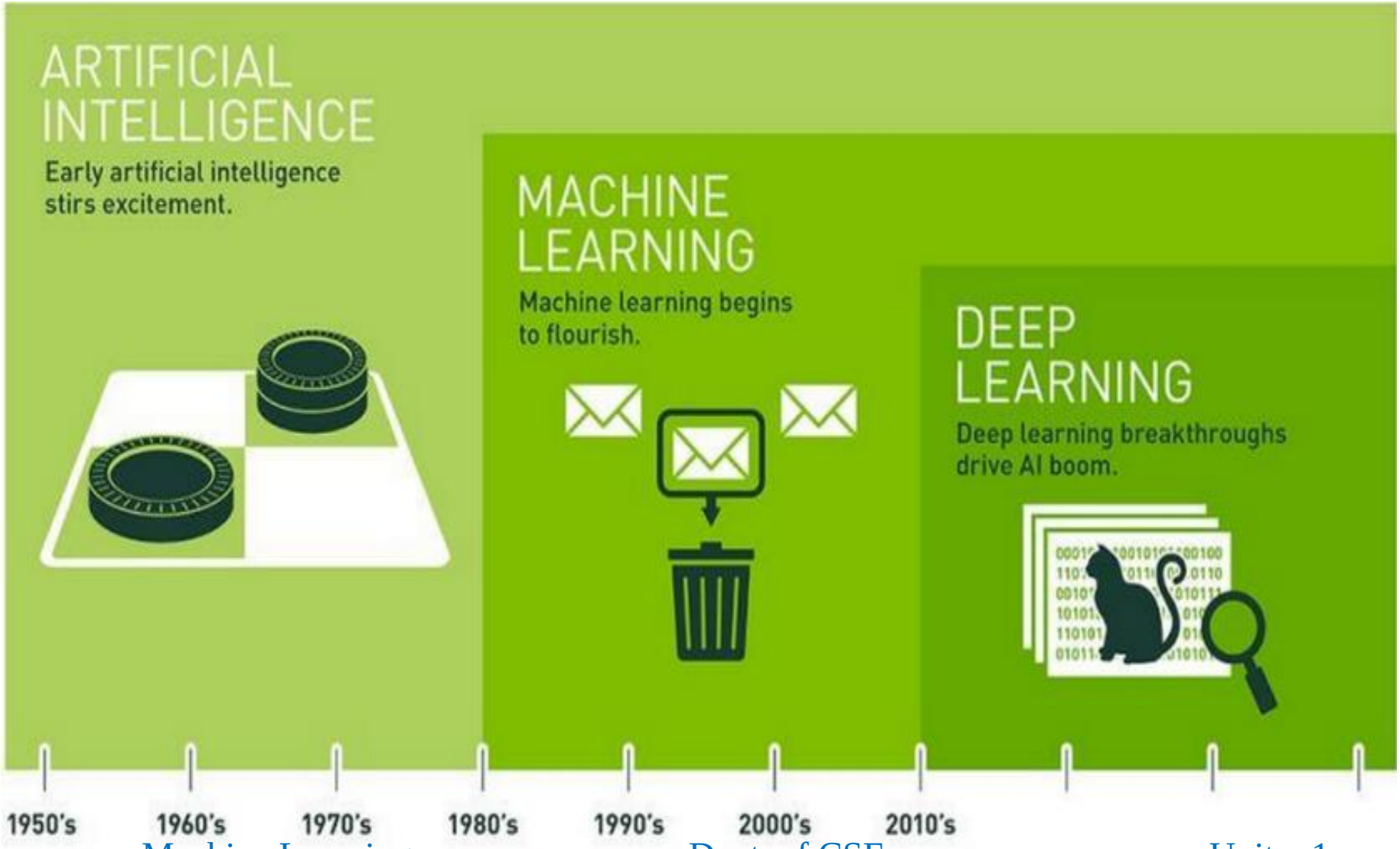




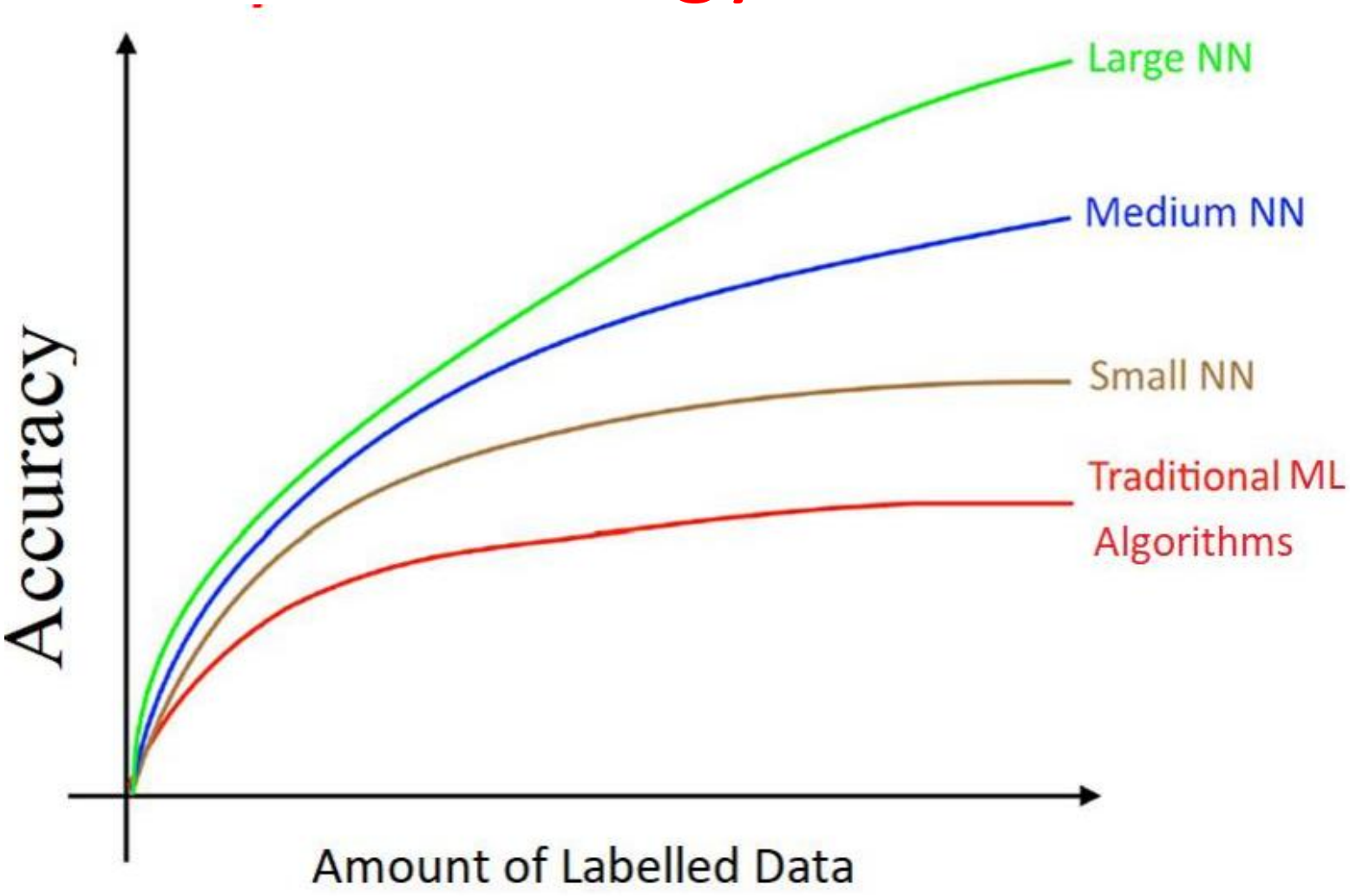


VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

What is Machine Learning



Technology Trends



Machine Learning Definition

- The field of study that gives computers the ability to learn with out being explicitly programmed.
- Machine learning is the systematic study of algorithms and systems that improve their knowledge or performance with experience.
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.
- Classify e-mails as spam or not spam
- Watching e-mails as spam or not spam
- Number of e-mails correctly classified as spam or not spam

Tasks in ML Example

- The most common machine learning tasks are predictive
  - Predicting a target variable from features.

- If our e-mails are described by word-occurrence features as in the text classification example, the similarity of e-mails would be measured using word-occurrence

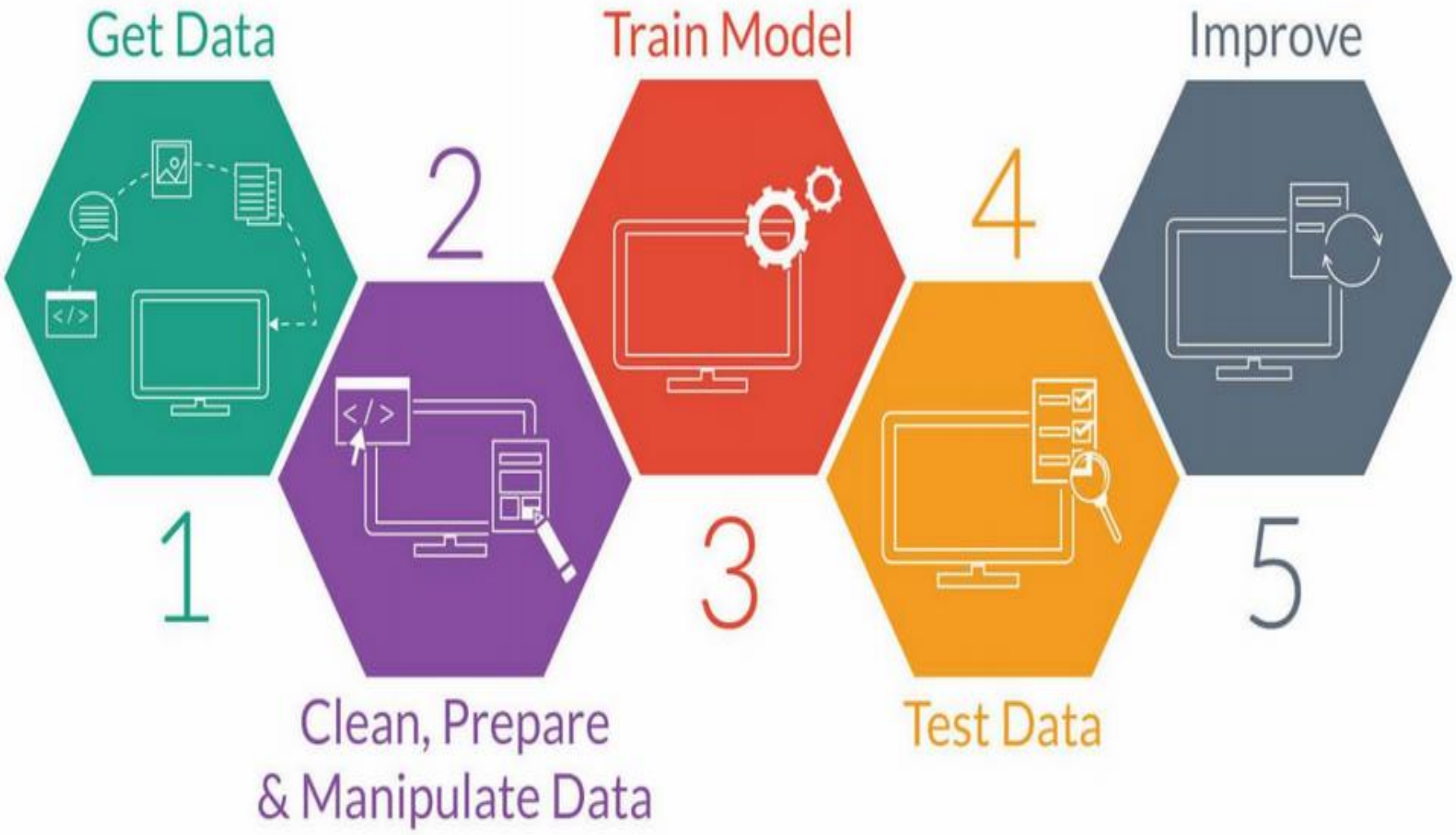
- Jaccard coefficient (JC)
  - Number of common words in two e-mails divided by the number of words occurring in either e-mail.

- Suppose that one e-mail contains 42 (different) words and another contains 112 words, and the two e-mails have 23 words in common, then their similarity would be $JC=\frac{23}{42+112-23}=\frac{23}{130}=0.18$

- We can then cluster e-mails into groups, such that the average similarity of an

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Tasks in ML



Ingredients of Machine Learning

- Feature
- Task
- Learning models
  - Supervised: Learns from Examples which provide desired outputs for given inputs
  - Unsupervised: Learns patterns in input data when no specific output values are given.
- Regression
- Examples

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Types of Learning

Supervised Learning	Unsupervised Learning
Input data is labelled	Input data is unlabelled
Uses training dataset	Uses just input dataset
Used for prediction	Used for analysis
Classification and regression	Clustering, density estimation and dimensionality reduction

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Quiz

- Given e-mail labbled as spam/not spam learn a spam filter.

  Supervised

- Given a set of news articles on web group them in to set of articles about same story.

  Un-Supervised

- Given customer data automatically discover market segments and group customers in to different market segments.

  Un-Supervised

- Given database of patients having diabetics or not. Learn classifier to say new patient having diabetics or not.

  Supervised

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Feature

Machine Learning Is All About using the right features to build the right models that achieve the right tasks

Category	Features
Housing Prices	No. of Rooms, House Area, Air Pollution, Distance from facilities, Economic Index city, Security Ranking etc.
Spam Detection	presence or absence of certain email headers, the email structure, the language, the frequency of specific terms, the grammatical correctness of the text etc.
Speech Recognition	noise ratios, length of sounds, relative power of sounds, filter matches
Cancer Detection	Clump thickness, Uniformity of cell size, Uniformity of cell shape, Marginal adhesion, Single epithelial cell size, Number of bare nuclei, Bland chromatin, Number of normal nuclei, Mitosis etc.
Cyber Attacks	IP address, Timings, Location, Type of communication, traffic details
Video Recommendations	Text matches, Ranking of the video, Interest overlap, history of seen videos, browsing patterns etc.
Image Classification	Pixel values, Curves, Edges etc.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Unit-1

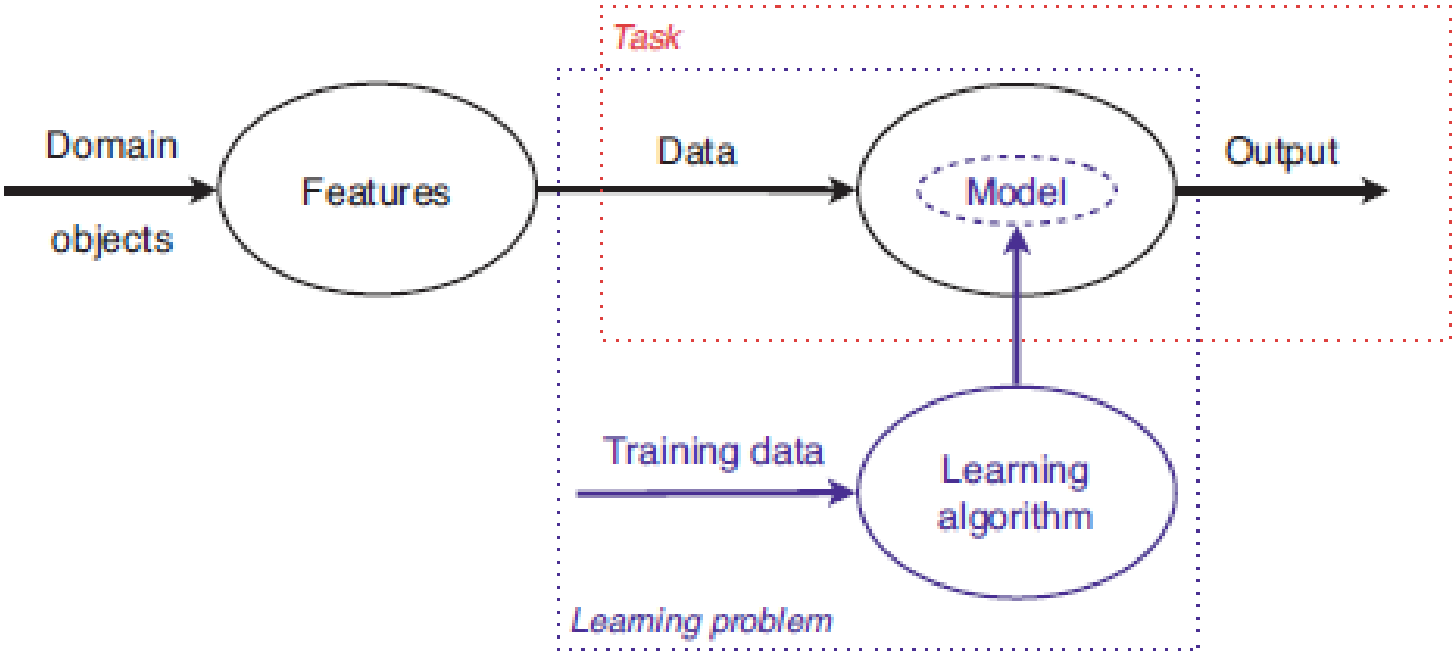
The ingredients of machine learning

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Introduction

- MACHINE LEARNING IS ALL ABOUT using the **right features** to build the right models that achieve the right tasks .
- In essence, **features** define a 'language' in which we describe the relevant objects in our domain, be they e-mails or complex organic molecules.
- A **task** is an abstract representation of a problem we want to solve regarding those domain objects: the most common form of these is classifying them into two or more classes

- Many of these tasks can be represented as a mapping from data points to outputs.
- This mapping or **model** is itself produced as the output of a machine learning algorithm applied to training data


Task
Domain
objects
Features
Data
Model
Output
Training data
Learning
algorithm
Learning problem

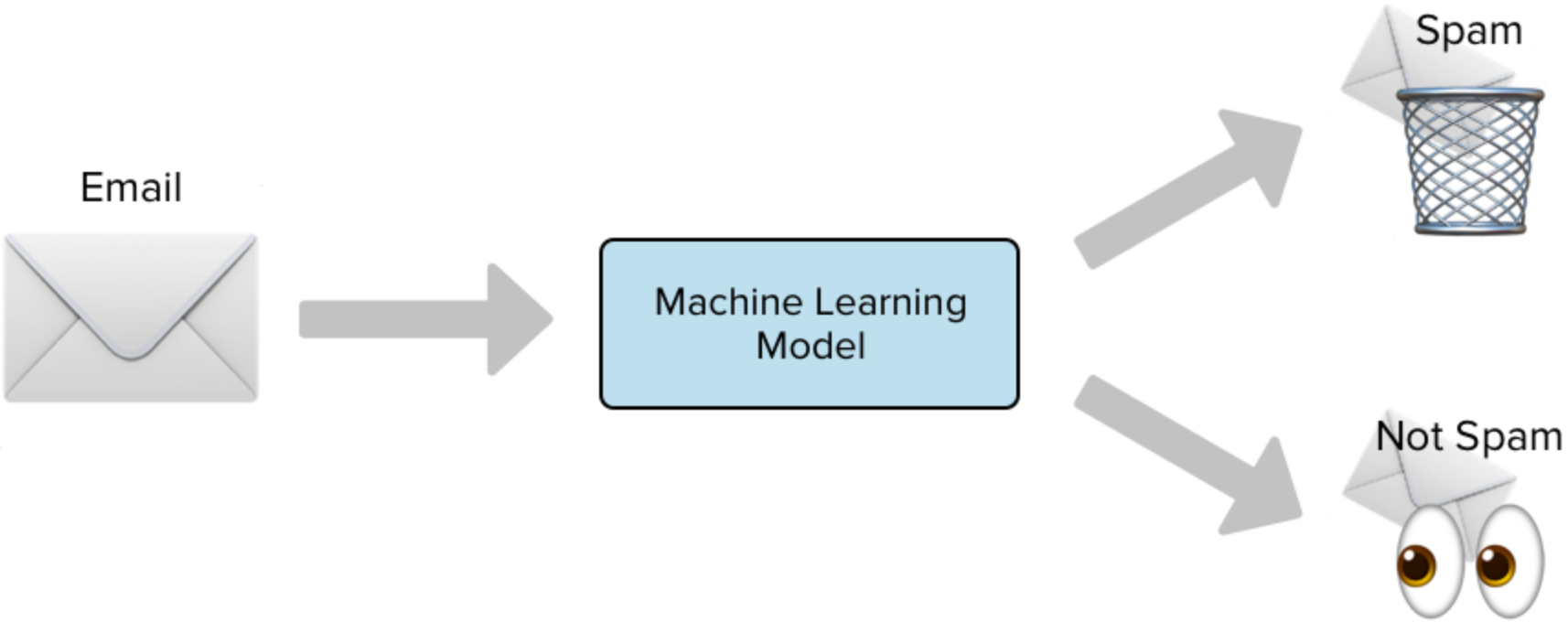
VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

1.1 Tasks: the problems that can be solved with machine learning

- Spam e-mail recognition was described in the Prologue .
- It constitutes a binary classification task, which is easily the most common task in machine learning
- One obvious variation is to consider classification problems with more than two classes.
- For instance, we may want to distinguish different kinds of ham e-mails, e.g., work-related e-mails and private messages.

- **SpamAssassin** is a widely used open-source spam filter.
- It calculates a score for an incoming e-mail, based on a number of built-in rules or 'tests' in SpamAssassin's terminology, and adds a 'junk' flag and a summary report to the e-mail's headers if the score is 5 or more.

```
-0.1 RCVD_IN_MXRATE_WL       RBL: MXRate recommends allowing
                             [123.45.6.789 listed in sub.mxrate.net]
 0.6 HTML_IMAGE_RATIO_02     BODY: HTML has a low ratio of text to image area
 1.2 TVD_FW_GRAPHIC_NAME_MID BODY: TVD_FW_GRAPHIC_NAME_MID
 0.0 HTML_MESSAGE            BODY: HTML included in message
 0.6 HTML_FONx_FACE_BAD      BODY: HTML font face is not a word
 1.4 SARE_GIF_ATTACH         FULL: Email has a inline gif
 0.1 BOUNCE_MESSAGE          MTA bounce message
 0.1 ANY_BOUNCE_MESSAGE      Message is some kind of bounce message
 1.4 AWL                     AWL: From: address is in the auto white-list
```


Email
Machine Learning Model
Spam
Not Spam

- Most current e-mail clients incorporate algorithms **to identify and filter** out spam e-mail, also known as junk e-mail or unsolicited bulk e-mail.
- Early spam filters relied on hand-coded pattern matching techniques such as regular expressions, but it soon became apparent that this is hard to maintain and offers insufficient flexibility – after all, one person's spam is another person's ham

- We could approach this as a combination of two binary classification tasks:
  - the first task is to distinguish between spam and ham,
  - and the second task is, among ham e-mails, to distinguish between work-related and private ones.
- However, some potentially useful information may get lost this way, as some spam e-mails tend to look like private rather than work-related messages.
- For this reason, it is often beneficial to view **multi-class classification** as a machine learning task in its own right.

- Sometimes it is more natural to abandon the notion of discrete classes altogether and instead predict a **real number**.
- Perhaps it might be useful to have an assessment of an incoming e-mail's urgency on a sliding scale
- This task is called **regression** , and essentially involves learning a real-valued function from training examples labelled with true function values.

- Regression is a statistical measurement used to determine the strength of the relationship between one dependent variable (usually denoted by Y) and a series of other changing variables (known as independent variables).
- The two basic types of regression are **linear regression** and **multiple linear regression**

- **Linear regression** uses one independent variable to explain or predict the outcome of the dependent variable Y, **while multiple regression** uses two or more independent variables to predict the outcome.

- For example, construct such a training set by randomly selecting a number of e-mails from my inbox and labelling them with an urgency score on a scale of 0 (ignore) to 10 (immediate action required).
- This typically works by **choosing** a class of functions (e.g., functions in which the function value depends linearly on some numerical features) and **constructing** a function which minimises the difference between the predicted and true function values.

- Both classification and regression assume the availability of a training set of examples labelled with true classes or function values.
- Providing the true labels for a data set is often labour-intensive and expensive.
- Can we learn to distinguish spam from ham,or work e-mails from private messages, without a labelled training set?
- The answer is: yes, up to a point.
- The task of grouping data without prior information on the groups is called **clustering**
- Learning from unlabelled data is called **unsupervised** learning and is quite distinct from **supervised** learning , which requires labelled training data.

- A typical **clustering algorithm** works by assessing the similarity between instances and putting similar instances in the same cluster and 'dissimilar' instances in different clusters.
- There are many other patterns that can be learned from data in an unsupervised way.
- **Association rules** are a kind of pattern that are popular in marketing applications,and the result of such patterns can often be found on online shopping web sites

Looking for structure

- Like all other machine learning models, patterns are a manifestation of underlying structure in the data.
- Sometimes this structure takes the form of a single hidden or latent variable
- Like all other machine learning models, patterns are a manifestation of underlying
- **Matrix decomposition** can often reveal useful hidden structure.

- **Applications**

- Background Removal
- Topic Modeling
- Recommendations using Collaborative Filtering


CARVANA
Original Photo
Auto-editing
Manual editing

Dot product of Movie-A with User-X gives prediction for Movie-A by User-X

Movies

Users	*(Sparse matrix)*	Waking Life	Boyhood	Before Sunset
	Jesse	4.5		4.0
	Celine		3.5	5.0

≈

D (n_factors)

Users

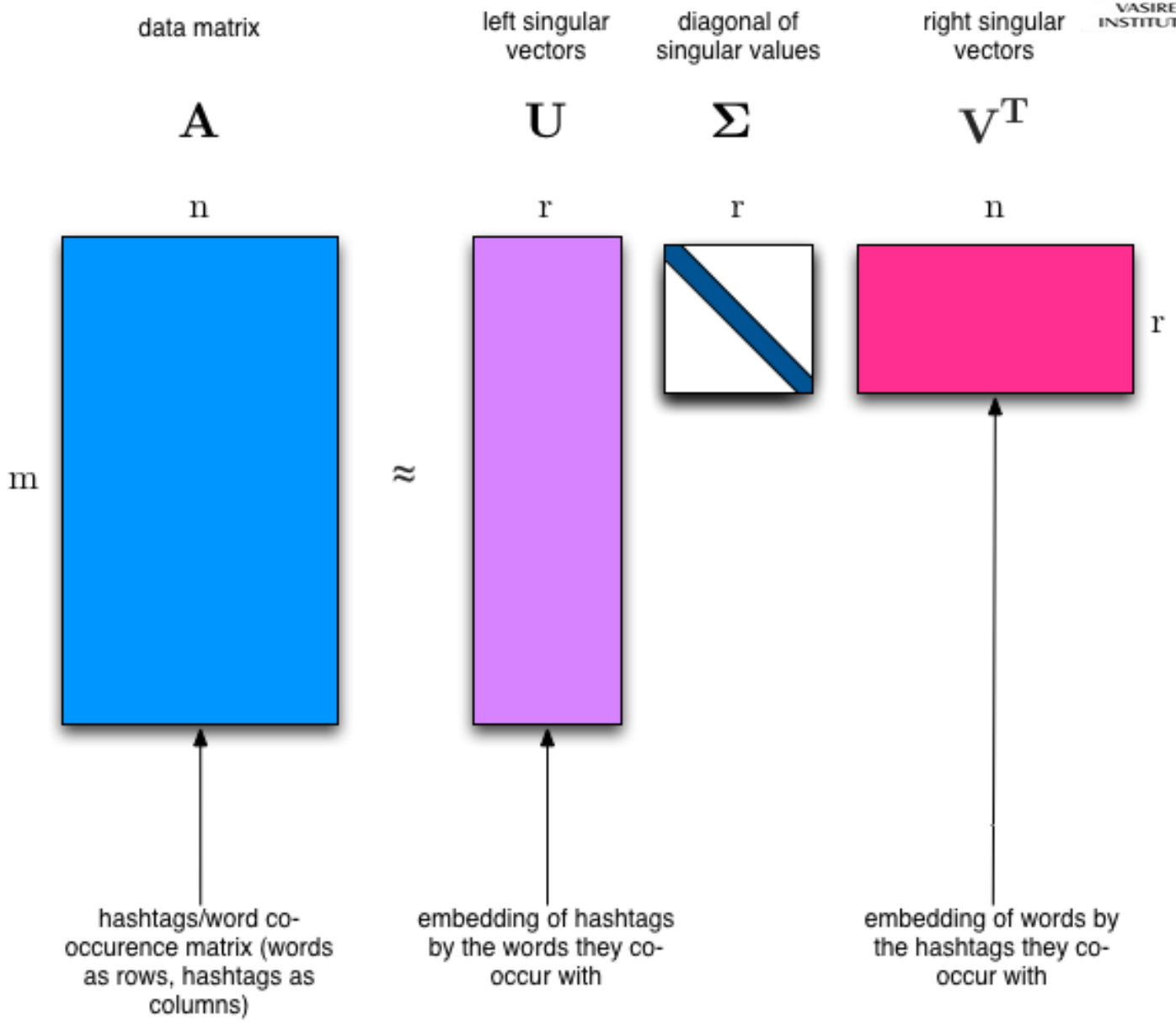
User embedding matrix

X

Movies

Movie embedding matrix

D (n_factors)


data matrix
left singular vectors
diagonal of singular values
right singular vectors
A
U
Σ
V^T
n
r
r
n
m
≈
r
hashtags/word co-occurence matrix (words as rows, hashtags as columns)
embedding of hashtags by the words they co-occur with
embedding of words by the hashtags they co-occur with

- We have already seen the distinction between supervised learning from labelled data and unsupervised learning from unlabelled data.
- We can similarly draw a distinction between whether the model output involves the target variable or not:
- we call it a **predictive model** if it does, and a **descriptive model** if it does not.

	Predictive model	*Descriptive model*
Supervised learning	classification, regression	subgroup discovery
Unsupervised learning	predictive clustering	descriptive clustering, association rule discovery

Table 1.1. An overview of different machine learning settings. The rows refer to whether the training data is labelled with a target variable, while the columns indicate whether the models learned are used to predict a target variable or rather describe the given data.

Evaluating performance on a task

- An important thing to keep in mind with all these machine learning problems is that they don't have a **'correct' answer**
- It is not like ordinary algorithms like **sorting algorithms**
- Accuracy
- No overfitting or underfitting

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Examples

- **Holdout**
- In this method, the dataset is *randomly* divided into three subsets:
- **Training set** is a subset of the dataset used to build predictive models.
- **Validation set** is a subset of the dataset used to assess the performance of the model built in the training phase. It provides a test platform for fine-tuning a model's parameters and selecting the best performing model. Not all modeling algorithms need a validation set.
- **Test set**, or unseen data, is a subset of the dataset used to assess the likely future performance of a model. If a model fits to the training set much better than it fits the test set, overfitting is probably the cause.

- **Cross-Validation**
- Cross-validation is a technique that involves partitioning the original observation dataset into a training set, used to train the model, and an independent set used to evaluate the analysis.
- The most common cross-validation technique is k-fold cross-validation, where the original dataset is partitioned into k equal size subsamples, called folds. The k is a user-specified number, usually with 5 or 10 as its preferred value.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Models: the output of machine learning

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Introduction

- Models form the central concept in machine learning as they are what is being learned from the data, in order to solve a given task.
- The tasks that machine learning aims to solve: classification, regression, clustering, association discovery, to name but a few.

- There is no simple way to classify machine learning algorithms
- For a given problem, the collection of all possible outcomes represents the **sample space or instance space(**set of all attributes**)**.
- The basic idea for creating a taxonomy of algorithms is that we divide the instance space by using one of three ways:
  - Using a Logical expression.
  - Using the Geometry of the instance space.
  - Using Probability to classify the instance space.

- i.e., there are three groups of models:
  - Geometric models,
  - probabilistic models, and
  - logical models.

  The outcome of the transformation of the instance space by a machine learning algorithm using the above techniques should be **exhaustive** (cover all possible outcomes) and **mutually exclusive** (non-overlapping).

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

1. Logical models

1.1 Logical models - Tree models and Rule models

- **Logical models** use a logical expression to divide the instance space into segments and hence construct grouping models.
- A **logical expression** is an expression that returns a Boolean value, i.e., a True or False outcome.
- Once the data is grouped using a logical expression, the data is divided into homogeneous groupings for the problem we are trying to solve.
- For example, for a **classification problem**, all the instances in the group belong to one class.

- There are mainly two kinds of logical models: **Tree models** and **Rule models**.

- Rule models consist of a collection of implications or IF-THEN rules.
- For tree-based models, the **'if-part'** defines a segment and the **'then-part'** defines the behaviour of the model for this segment.
- Rule models follow the same reasoning.

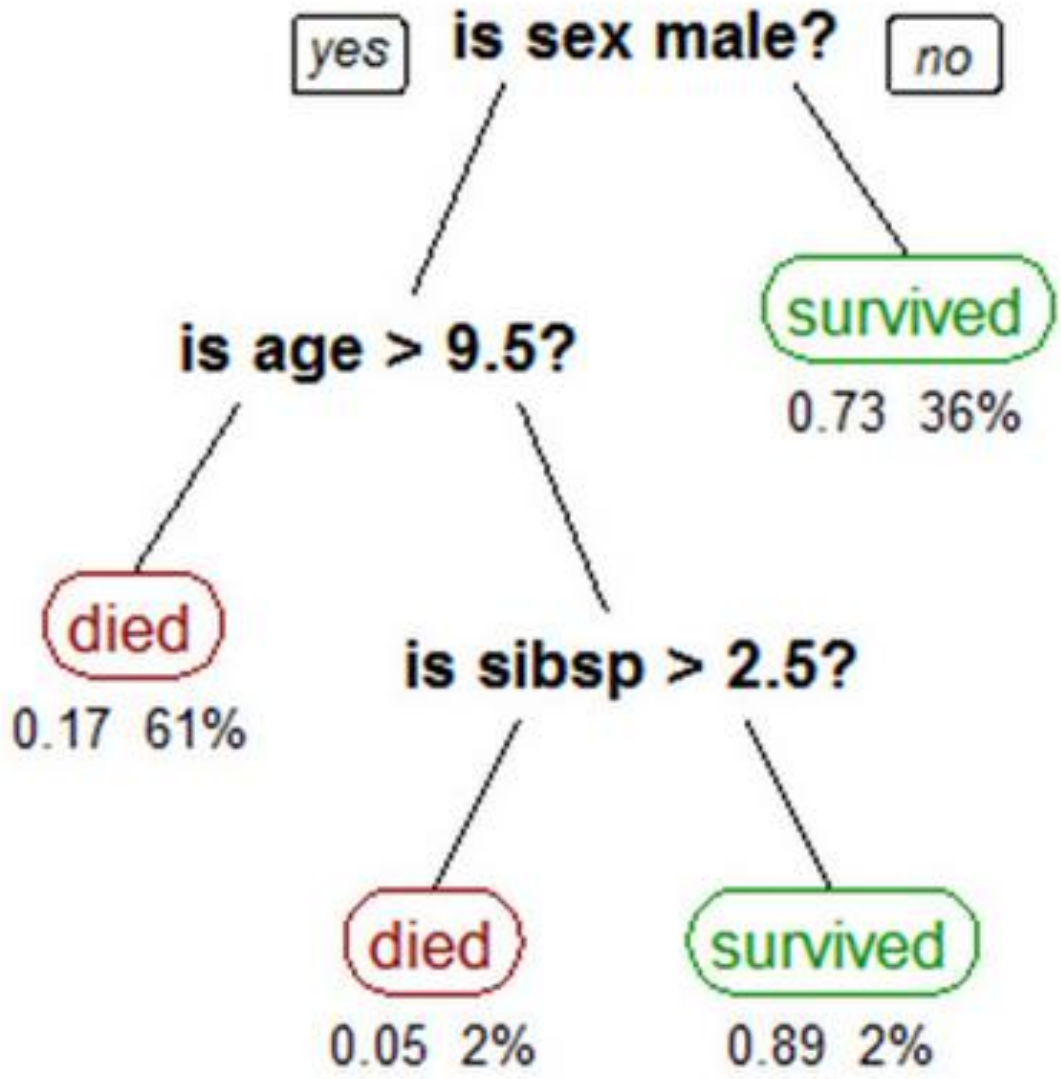
- Tree models can be seen as a particular type of rule model where the **if-parts** of the rules are organized in **a tree structure**.

- But the difference is
- The representation. For tree modelthere's a branching structure at each point that tells you whether to go left or right. For rule based, it's a sequence of logical predicates that are executed in order (e.g. If X is true and Y or Z are false it's a rabbit).

- Both Tree models and Rule models use the same approach to supervised learning.
- The approach can be summarized in two strategies:
  - we could first find the body of the rule (the concept) that covers a sufficiently homogeneous set of examples and then find a label to represent the body.
  - Alternately, we could approach it from the other direction, i.e., first select a class we want to learn and then find rules that cover examples of the class.

Example: tree-based model

- The tree shows survival numbers of passengers on the Titanic ("sibsp" is the number of spouses or siblings aboard).
- The values under the leaves show the probability of survival and the percentage of observations in the leaf.
- The model can be summarized as: Your chances of survival were good if you were (i) a female or (ii) a male younger than 9.5 years with less than 2.5 siblings.


yes
is sex male?
no
is age > 9.5?
survived
0.73 36%
died
0.17 61%
is sibsp > 2.5?
died
0.05 2%
survived
0.89 2%

1.2 Logical models and Concept learning

- To understand logical models further, we need to understand the idea of **Concept Learning**.
- Concept Learning involves learning logical expressions or concepts from examples.

What Is Concept Learning?

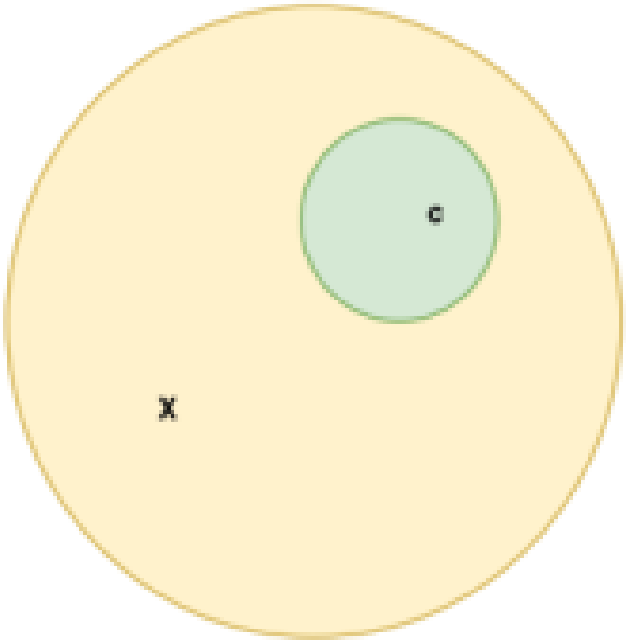
- "The problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples." — Tom Michell
- Much of human learning involves acquiring general concepts from past experiences.
- For example, humans identify different vehicles among all the vehicles based on specific sets of features defined over a large set of features.
- This special set of features differentiates the subset of cars in a set of vehicles. This set of features that differentiate cars can be called a concept.

- Similarly, machines can learn from concepts to identify whether an object belongs to a specific category by processing past/training data to find a hypothesis that best fits the training examples.

- A Formal Definition for Concept Learning is "***The inferring of a Boolean-valued function from training examples of its input and output.***"

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Target Concept:



- The set of items/objects over which the concept is defined is called the set of instances and denoted by X.
- The concept or function to be learned is called the target concept and denoted by c.
- It can be seen as a boolean valued function defined over X and can be represented as **c: X -> {0, 1}**.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

A Concept Learning Task – Enjoy Sport

Training Examples

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	YES
2	Sunny	Warm	High	Strong	Warm	Same	YES
3	Rainy	Cold	High	Strong	Warm	Change	NO
4	Sunny	Warm	High	Strong	Warm	Change	YES

ATTRIBUTES

CONCEPT

- A Concept Learning Task called "Enjoy Sport" as shown above is defined by a set of data from some example days. Each data is described by six attributes.
- The task is to learn to predict the value of Enjoy Sport for an arbitrary day based on the values of its attribute values.
- The problem can be represented by a **series of hypotheses**. Each hypothesis is described by a conjunction of constraints on the attributes.
- The training data represents a set of positive and negative examples of the target function.
- In the example above, each hypothesis is a vector of six constraints, specifying the values of the six attributes – Sky, AirTemp, Humidity, Wind, Water, and Forecast.
- The training phase involves learning the set of days (as a conjunction of attributes) for which Enjoy Sport = yes.

- Thus, the problem can be formulated as:
- Given instances X which represent a set of all possible days, each described by the attributes:
  - Sky – (values: Sunny, Cloudy, Rainy),
  - AirTemp – (values: Warm, Cold),
  - Humidity – (values: Normal, High),
  - Wind – (values: Strong, Weak),
  - Water – (values: Warm, Cold),
  - Forecast – (values: Same, Change).
- Try to identify a function that can predict the target variable Enjoy Sport as yes/no, i.e., 1 or 0.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

1.3 Concept learning as a search problem and as Inductive Learning

- We can also formulate Concept Learning as a **search problem**.
- We can think of Concept learning as searching through a set of predefined space of potential hypotheses to identify a hypothesis that best fits the training examples.
- Concept learning is also an example of **Inductive Learning**. Inductive learning, also known as **discovery learning**, is a process where the learner discovers rules by observing examples

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

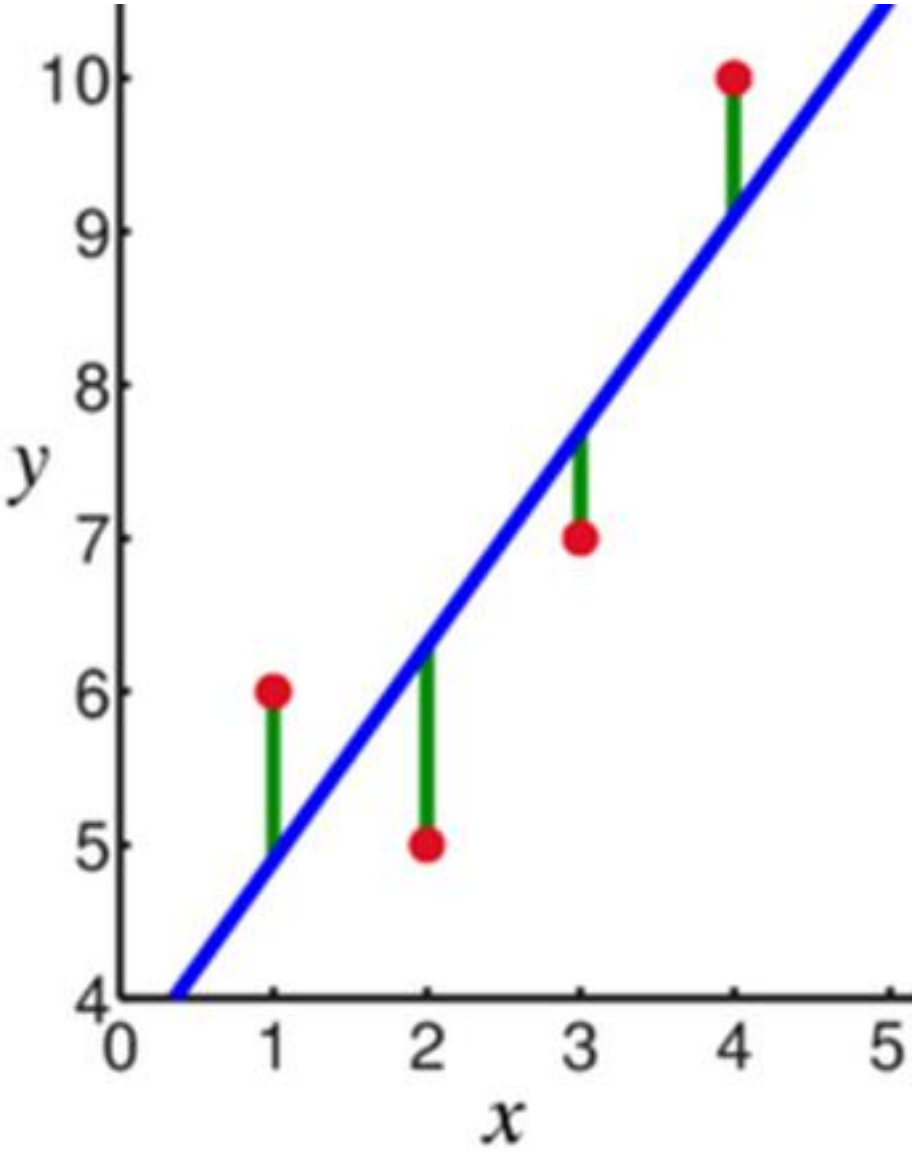
2. Geometric models

- Models that define **similarity** by considering the geometry of the instance space.
- In Geometric models, **features** could be described as points in two dimensions (*x*- and *y*-axis) or a three-dimensional space (*x*, *y,* and *z*).
- Even when features are not intrinsically geometric, they could be modelled in a geometric manner (for example, temperature as a function of time can be modelled in two axes).

- In geometric models, there are two ways we could impose similarity.
- We could use geometric concepts like **lines or planes to segment (classify)** the instance space. These are called **Linear models**.
- Alternatively, we can use the geometric notion of distance to represent similarity.
- In this case, if two points are close together, they have similar values for features and thus can be classed as similar. We call such models as **Distance-based models**.

2.1 Linear models

- Linear models are relatively simple. In this case, the function is represented as a linear combination of its inputs.
- Thus, if x_1 and x_2 are two scalars or vectors of the same dimension and a and b are arbitrary scalars, then $ax_1 + bx_2$ represents a linear combination of x_1 and x_2.
- In the simplest case where $f(x)$ represents a straight line, we have an equation of the form $f(x) = mx + c$ where c represents the intercept and m represents the slope.


10
9
8
7
6
5
4
y
0
1
2
3
4
5
x

- Linear models are **parametric**, which means that they have a fixed form with a small number of numeric parameters that need to be learned from data.
- For example, in $f(x) = mx + c$, m and c are the parameters that we are trying to learn from the data.
- This technique is different from tree or rule models, where the structure of the model is not fixed in advance.

- Linear models are **stable**, i.e., small variations in the training data have only a limited impact on the learned model.
- In contrast, **tree models tend to vary more with the training data**, as the choice of a different split at the root of the tree typically means that the rest of the tree is different as well.

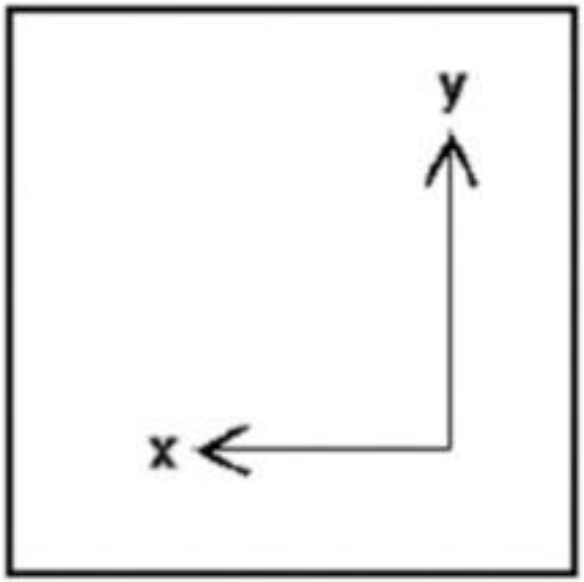
- As a result of having relatively few parameters, Linear models have **low variance and high bias**.
- This implies that **Linear models are less likely to overfit the training data** than some other models.
- However, they are more likely to underfit.
- For example, if we want to learn the boundaries between countries based on labelled data, then linear models are not likely to give a good approximation.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

2.2 Distance-based models

- **Distance-based models** are the second class of Geometric models.
- Like Linear models, distance-based models are based on the geometry of data.
- As the name implies, distance-based models work on the concept of distance.
- In the context of Machine learning, the concept of distance is not based on merely the physical distance between two points.
- Instead, we could think of the distance between two points considering the **mode of transport** between two points.

- Travelling between two cities by plane covers less distance physically than by train because a plane is unrestricted.
- , in chess, the concept of distance depends on the piece used – for example, a Bishop can move diagonally.
- Thus, depending on the entity and the mode of travel, the concept of distance can be experienced differently.
- The distance metrics commonly used are **Euclidean**, **Minkowski**, **Manhattan**, and **Mahalanobis**.



Manhattan

Euclidean

- Distance is applied through the concept of **neighbours and exemplars**.
- Neighbours are points in proximity with respect to the distance measure expressed through exemplars.
- Exemplars are either **centroids** that find a centre of mass according to a chosen distance metric or **medoids** that find the most centrally located data point.
- The most commonly used centroid is the arithmetic mean, which minimises squared Euclidean distance to all other points.

- Examples of distance-based models include the **nearest-neighbour** models, which use the training data as exemplars – for example, in classification.
- The **K-means clustering** algorithm also uses exemplars to create clusters of similar data points.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

3. Probabilistic models

- Probabilistic models see features and target variables as random variables.
- The process of modelling represents and **manipulates the level of uncertainty** with respect to these variables.
- There are two types of probabilistic models: **Predictive and Generative**.
- Predictive probability models use the idea of a **conditional probability** distribution $P(Y \mid X)$ from which Y can be predicted from X.

- Generative models estimate the **joint distribution** *P* (*Y*, *X*).
- Once we know the joint distribution for the generative models, we can derive any conditional or marginal distribution involving the same variables.
- Thus, the generative model is capable of creating new data points and their labels, knowing the joint probability distribution.
- The joint distribution looks for a relationship between two variables.
- Once this relationship is inferred, it is possible to infer new data points.

- **Naïve Bayes** is an example of a probabilistic classifier.
- The Naïve Bayes algorithm is based on the idea of **Conditional Probability.**
- **Conditional probability is based on finding the** probability that something will happen, *given that something else* has already happened
- The task of the algorithm then is to look at the evidence and to determine the likelihood of a specific class and assign a label accordingly to each entity.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Grouping and Grading

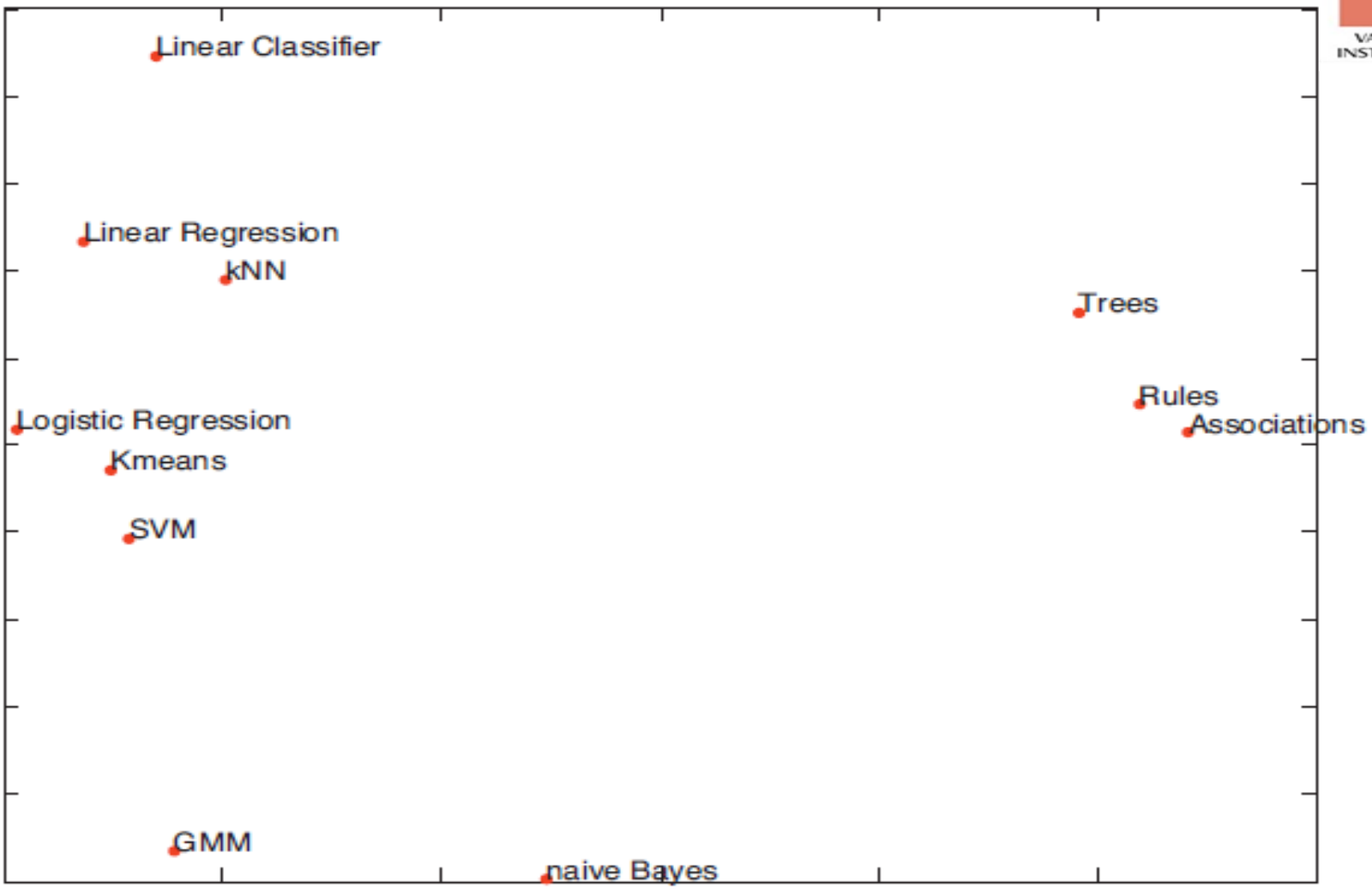
- We have looked at three general types of models: geometric models, probabilistic models and logical models.
- The key difference between these models is the way they handle the **instance space**.
- Grouping models do this by breaking up the instance space into groups or ***segments, the*** number of which is determined at training time

- What grouping models do at this finest resolution is often something very simple, such as assigning the **majority class** to all instances that fall into the segment.
- The main emphasis of training a grouping model is then on determining the right segments so that we can get away with this very simple **labelling** at the local segment level.

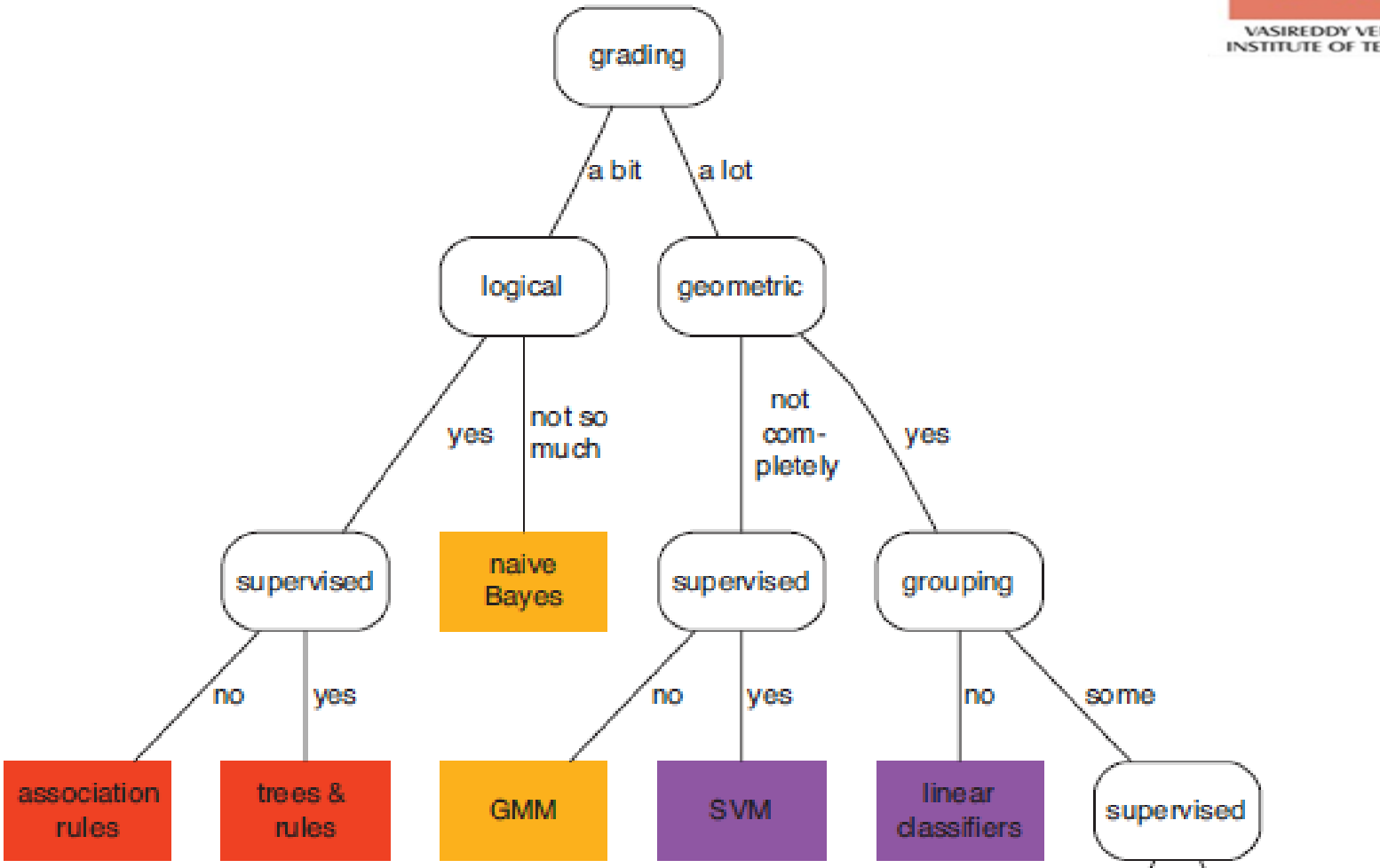
- A good example of grouping models are the tree-based models we have just considered.
- They work by repeatedly splitting the instance space into smaller subsets

- Grading models, on the other hand, do not employ such a notion of segment.
- Rather than applying very simple, local models, they form one global model over the instance space.
- Consequently, grading models are (usually) able to distinguish between arbitrary instances, no matter how similar they are

- **Support vector machines** and **other geometric** classifiers are examples of grading 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 dimension roughly ranges from grading models on the left to grouping models on the right.



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).

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Features: the workhorses of machine learning

Example 1.7, p.39

The MLM data set

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, p.39

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 MLM data set describing properties of machine learning models.

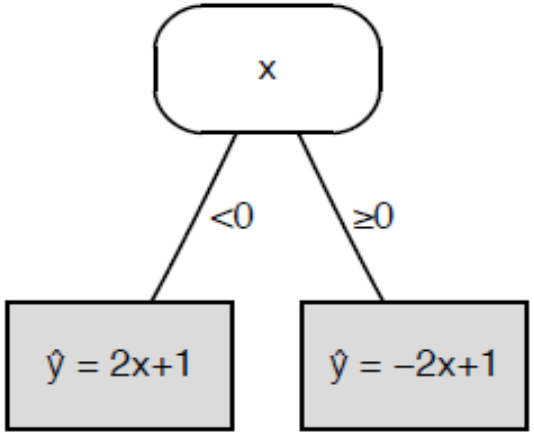
Example 1.8, p.41

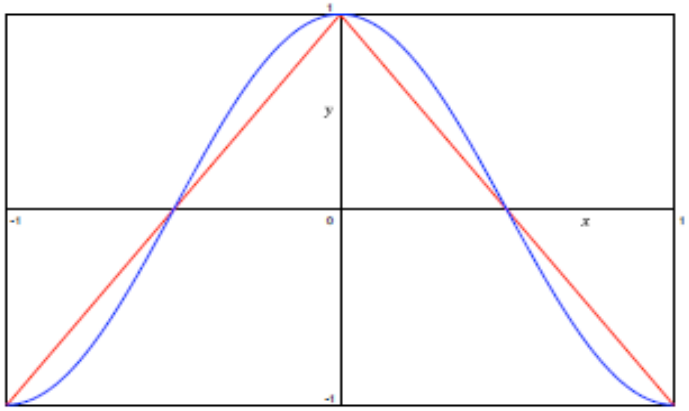
Two uses of features

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).

Figure 1.9, p.41

A small regression tree

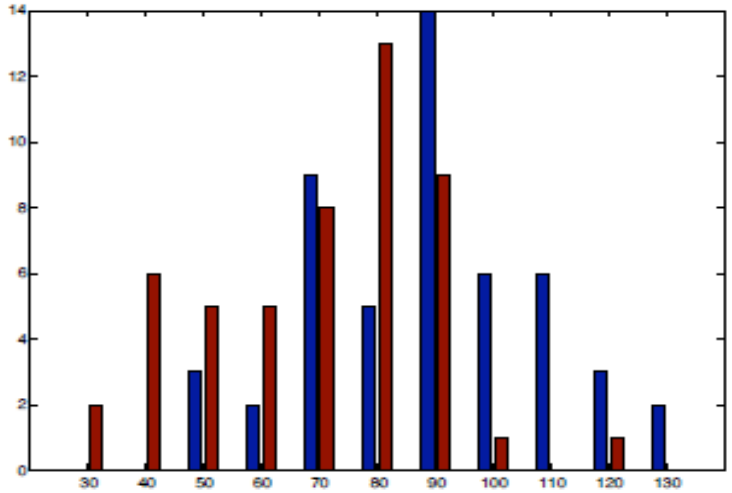


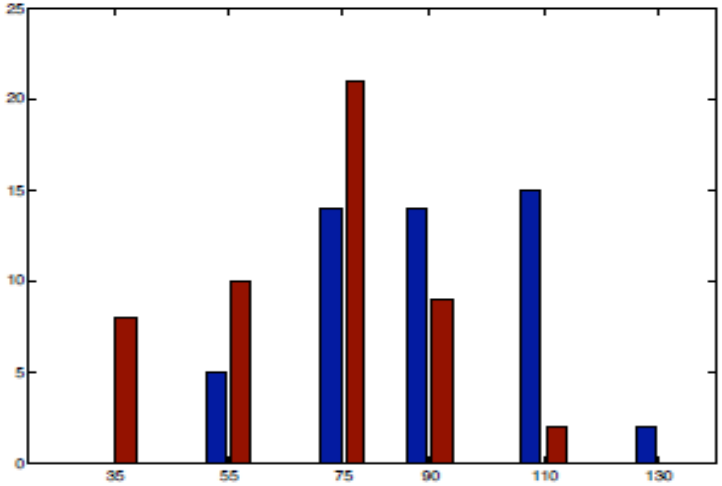


(left) A regression tree combining a one-split feature tree with linear regression models in the leaves. Notice how x is used as both a splitting feature and a regression variable. **(right)** The function $y = \cos \pi x$ on the interval $-1 \leq x \leq 1$, and the piecewise linear approximation achieved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

Figure 1.10, p.42

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.

Example 1.9, p.43

The kernel trick

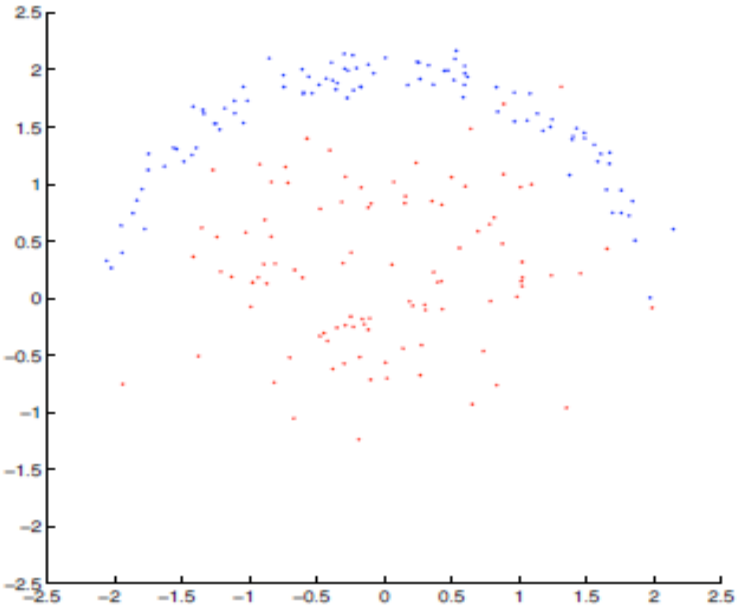
Let $\mathbf{x}_1 = (x_1, y_1)$ and $\mathbf{x}_2 = (x_2, y_2)$ be two data points, and consider the mapping $(x, y) \mapsto (x^2, y^2, \sqrt{2}xy)$ to a three-dimensional feature space. The points in feature space corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ are $\mathbf{x}'_1 = (x_1^2, y_1^2, \sqrt{2}x_1 y_1)$ and $\mathbf{x}'_2 = (x_2^2, y_2^2, \sqrt{2}x_2 y_2)$. The dot product of these two feature vectors is

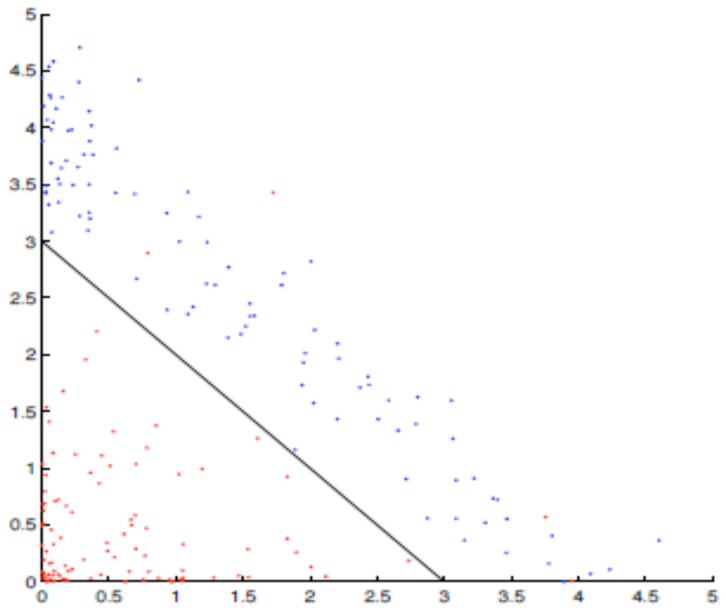
$$\mathbf{x}'_1 \cdot \mathbf{x}'_2 = x_1^2 x_2^2 + y_1^2 y_2^2 + 2x_1 y_1 x_2 y_2 = (x_1 x_2 + y_1 y_2)^2 = (\mathbf{x}_1 \cdot \mathbf{x}_2)^2$$

That is, by squaring the dot product in the original space we obtain the dot product in the new space *without actually constructing the feature vectors*! A function that calculates the dot product in feature space directly from the vectors in the original space is called a *kernel* – here the kernel is $\kappa(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 \cdot \mathbf{x}_2)^2$.

Figure 1.11, p.43

Non-linearly separable data





(left) A linear classifier would perform poorly on this data. **(right)** By transforming the original (x, y) data into $(x', y') = (x^2, y^2)$, the data becomes more 'linear', and a linear decision boundary $x' + y' = 3$ separates the data fairly well. In the original space this corresponds to a circle with radius $\sqrt{3}$ around the origin.

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Binary classification and related tasks

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

https://www.cse.iitk.ac.in/users/se367/10/presentation_local/Binary%20Classification.html

https://www.edureka.co/blog/classification-algorithms/

https://www.edureka.co/blog/classification-in-machine-learning/

VVIT
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Binary classification and related tasks

Contd..