COMPARISON OF ML ALGORITHMS

VARIOUS ML ALGORITHMS

PROS & CONS

Algorithm	Best at	Pros	Cons
Random Forest	Apt at almost any machine learning problem Bioinformatics	Can work in parallel Seldom overfits Automatically handles missing values No need to transform any variable No need to tweak parameters Can be used by almost anyone with excellent results	Difficult to interpret Weaker on regression when estimating values at the extremities of the distribution of response values Biased in multiclass problems toward more frequent classes
Linear regression	Baseline predictions Econometric predictions Modelling marketing responses	Simple to understand and explain It seldom overfits Using L1 & L2 regularization is effective in feature selection Fast to train Easy to train on big data thanks to its stochastic version	You have to work hard to make it fit nonlinear functions Can suffer from outliers
Support Vector Machines	Character recognition Image recognition Text classification	Automatic nonlinear feature creation Can approximate complex nonlinear functions	Difficult to interpret when applying nonlinear kernels Suffers from too many examples, after 10,000 examples it starts taking too long to train
K-nearest Neighbors	Computer vision Multilabel tagging	Fast, lazy training Can naturally handle	Slow and cumbersome in the predicting phase

	Recommender systems Spell checking problems	extreme multiclass problems (like tagging text)	Can fail to predict correctly due to the curse of dimensionality
Naive Bayes	Face recognition Sentiment analysis Spam detection Text classification	Easy and fast to implement, doesn't require too much memory and can be used for online learning Easy to understand Takes into account prior knowledge	Strong and unrealistic feature independence assumptions Fails estimating rare occurrences Suffers from irrelevant features
Logistic regression	Ordering results by probability Modelling marketing responses	Simple to understand and explain It seldom overfits Using L1 & L2 regularization is effective in feature selection The best algorithm for predicting probabilities of an event Fast to train Easy to train on big data thanks to its stochastic version	You have to work hard to make it fit nonlinear functions Can suffer from outliers
K-means	Segmentation	Fast in finding clusters Can detect outliers in multiple dimensions	Suffers from multicollinearity Clusters are spherical, can't detect groups of other shape Unstable solutions, depends on initialization

CALCULATIONS AND INTERPRETATION OF RESULTS

NAÏVE BAYES

NAÏVE BAYES ALGORITHM: THEORY AND CALCULATIONS

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

Likelihood

Class Prior Probability

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target. Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

$$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(Yes) = 9/14 = 0.64$$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$

Joint posterior probability for 8 independent variables is a logical extension of two variables, by taking product of all 8 conditional probabilities. Since python library or any statistical software would do the math, only demonstration of logic and a sample calculation is adequate.

PRESENTATION OF RESULTS: CONFUSION MATRIX

Learners are expected to build tables of likelihood for 8 variables. (Depending on time available they may do so only for one or two variables.)

- **True positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **True negatives (TN):** We predicted no, and they don't have the disease.
- **False positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **False negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Sample Confusion matrix (exact numbers to be populated from Diabetes case)

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

INTERPRETATION OF RESULTS: TESTING ACCURACY

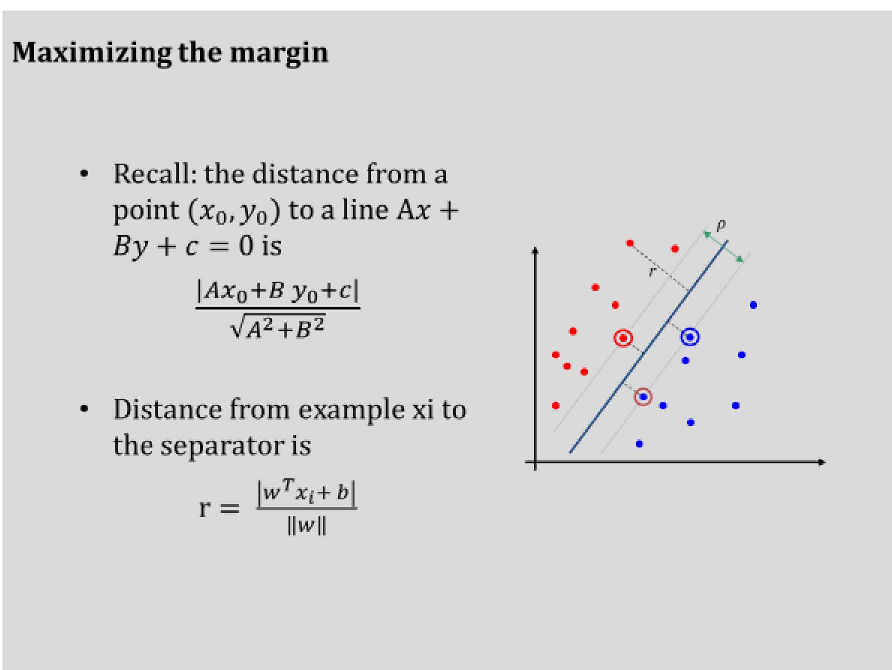
Accuracy is not a single number but depending on precise question being answered, can be from amongst :-

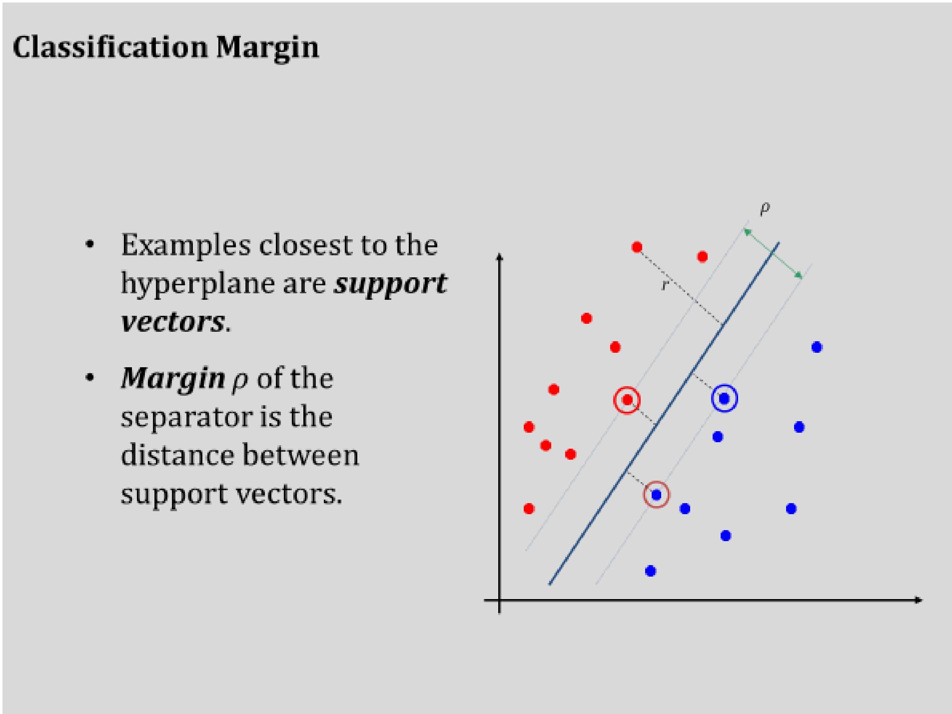
- **Accuracy:** Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  - FP/actual no = 10/60 = 0.17
- **Specificity:** When it's actually no, how often does it predict no?
  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
- **Precision:** When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91
- **Prevalence:** How often does the yes condition actually occur in our sample?
  - actual yes/total = 105/165 = 0.64

SUPPORT VECTOR MACHINES

THEORY AND CALCULATIONS

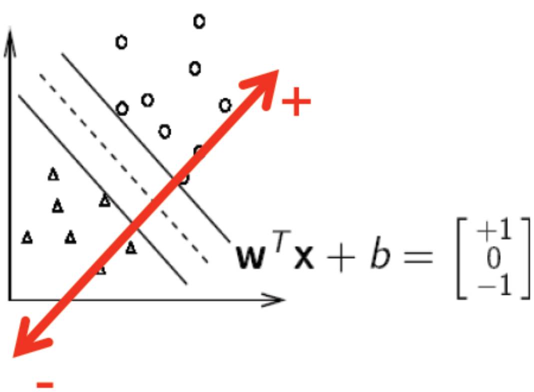
The algorithm is best explained using visualization and basic concepts in coordinate geometry.





(1)Decision value as strength

Decision function $f(x) = \text{sign}(w^T x_{new} + b)$



INTERPRETATION OF WEIGHTS

Magnitude of weight after due normalization of variables, helps in deciding feature importance. The interpretation of weights can be explained using OLS (Ordinary Least Squares) regression as an analogy.

Detailed working example can be used from :-

https://charlienewey.github.io/manually-calculating-an-svms-support-vectors/

INTERPRETATION OF RESULTS: TESTING ACCURACY

While exact values of results will vary between Naïve Bayes and SVM, the interpretation of testing accuracy is same. Previous section on Naïve Bayes accuracy refers.

HEALTHCARE APPLICATIONS

USING ML ALGORITHMS

The value of machine learning in healthcare is its ability to process huge datasets beyond the scope of human capability, and then reliably convert analysis of that data into clinical insights that aid physicians in planning and providing care, ultimately leading to better outcomes, lower costs of care, and increased patient satisfaction.

It has been estimated that big data and machine learning in pharma and medicine could generate a value of up to $100B annually, based on better decision-making, improved efficiency of research/clinical trials, and new tool creation for physicians, consumers, insurers, and regulators.

a. Disease identification / diagnosis - Creation of a platform to analyze data, and loop it back in real time to physicians to aid in clinical decision making is CDSS. A physician sees a patient and enters symptoms, data, and test results into the EMR, there's machine learning behind the scenes looking at everything about that patient, and prompting the doctor with useful information for making a diagnosis, ordering a test, or suggesting a preventive screening. In the long term, we will be able to incorporate bigger sets of data that can be analyzed in real time to provide all kinds of information to the provider and patient.
b. Show causal relationships in disease prognosis and help in predictions.
c. Patient risk profile – depending on various signs and symptoms and lifestyle factors.
d. Gather public health data and predict epidemic outbreaks.
e. Reduce 1-year mortality - Health systems can reduce 1-year mortality rates by predicting the likelihood of death within one year of discharge and then match patients with appropriate interventions, care providers, and support.

NEW SERVICES & PRODUCTS- CLINICAL

1. Application of ML classification algorithms on diabetes dataset is representative case study of classifying any clinical dataset into infected/not-infected categories. For example, likelihood of cancer, hypertension etc, through careful selection of variables relevant to predicting that particular disease.
2. Predict chronic disease - Machine learning can help hospital systems identify patients with undiagnosed or misdiagnosed chronic disease, predict the likelihood that patients will develop chronic disease, and present patient-specific prevention interventions.
3. If time-series data is available then diseases can be predicted based on ML from historic data.
4. New products can be developed by integrating ML algorithms into existing diagnostic solutions. For example, medical image classification uses similar ML classifiers to classify retinal images into diabetic retinopathy absent/present. Such diagnostic reports can be made more informative for radiologists by providing ML-provided insights.

NEW SERVICES & PRODUCTS- HEALTHCARE MANAGEMENT

a. Reduce readmissions - Machine learning can reduce readmissions in a targeted, efficient, and patient-centered manner. Clinicians can receive daily guidance as to which patients are most likely to be readmitted and how they might be able to reduce that risk.
b. Prevent hospital acquired infections (HAIs). Clinicians can monitor high risk patients and intervene to reduce that risk by focusing on patient-specific risk factors.

c. Reduce hospital Length-of-Stay (LOS). Health systems can reduce LOS and improve other outcomes like patient satisfaction by identifying patients that are likely to have an increased LOS and then ensure that best practices are followed.
d. Predict propensity-to-pay - Health systems can determine who needs reminders, who needs financial assistance, and how the likelihood of payment changes over time and after particular events.
e. Predict no-shows - Health systems can create accurate predictive models to assess, with each scheduled appointment, the risk of a no-show, ultimately improving patient care and the efficient use of resources.