Big Bio-Data Analysis (Artificial Intelligence and Machine Learning)

15 July 2022

Machine Learning Algorithms

By

Richard Sserunjogi
Department of Computer Science
Makerere University, Uganda
sserurich@gmail.com

AFRICAN CENTERS OF EXCELLENCE IN BIOINFORMATICS & DATA INTENSIVE SCIENCE

Overview

- Linear Regression
  - Introduction to Linear Regression
  - Simple & Multi-Linear Regression
  - Regularization: Ridge & Lasso Regression
- Logistic Regression
  - Introduction to Logistic Regression (Binary Logistic Regression)
  - Odds Ratio
  - Maximum Likelihood Estimation
- Support Vector Machine
- K Nearest Neighbour

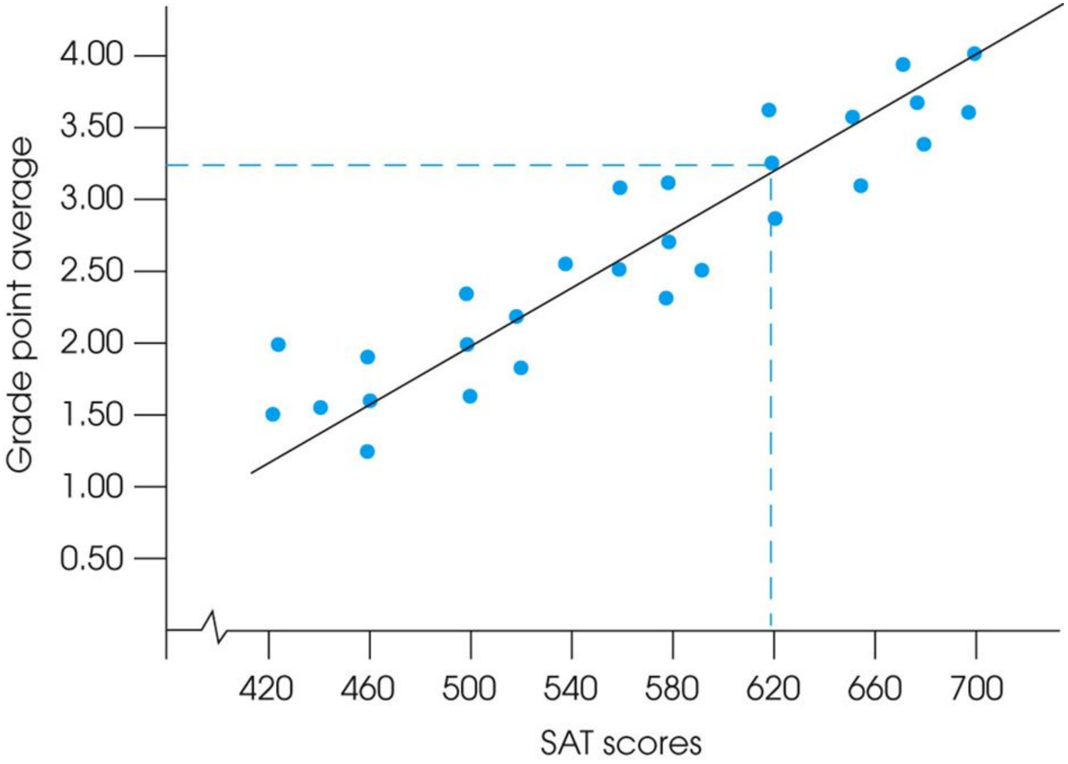
Introduction to Linear Regression

- Linear regression is one of the most well known and well understood algorithm in statistics and machine learning.
- Linear regression is studied as a model for understanding the relationship between input and output numerical variables.
- **Regression** is a statistical procedure that determines the equation for the straight line that best fits a specific set of data

Introduction to Linear Regression(cont'd)

- Any straight line can be represented by an equation of the form Y = βX + α, where b and α are constants.
- The value of β is called the slope constant and determines the direction and degree to which the line is tilted.
- The value of α is called the Y-intercept and determines the point where the line crosses the Y-axis.

Introduction to Linear Regression(cont'd)

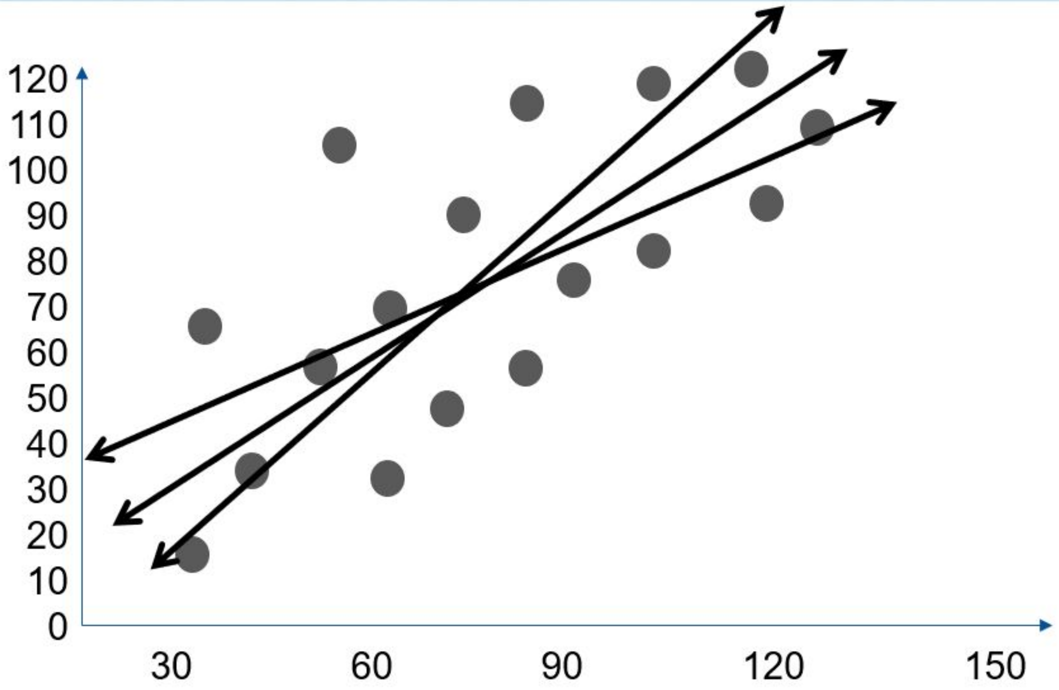


Simple Linear Regression (cont'd)

- Equation for our line: $\mathbf{y_i = \alpha + \beta x_i + \varepsilon_i}$
  - y: our response or dependent variable
  - α: the y-intercept (value of x when y=0)
  - β: the regression coefficient (slope of the line)
  - x: our predictor/independent variable value
  - ε: the error (assumed independent)
- Goal: We want to find α and β that minimize the sum of square residuals: $\mathbf{SS_{res} = \sum_i (y_i - (\alpha + \beta xi))2}$

Simple Linear Regression (cont'd)

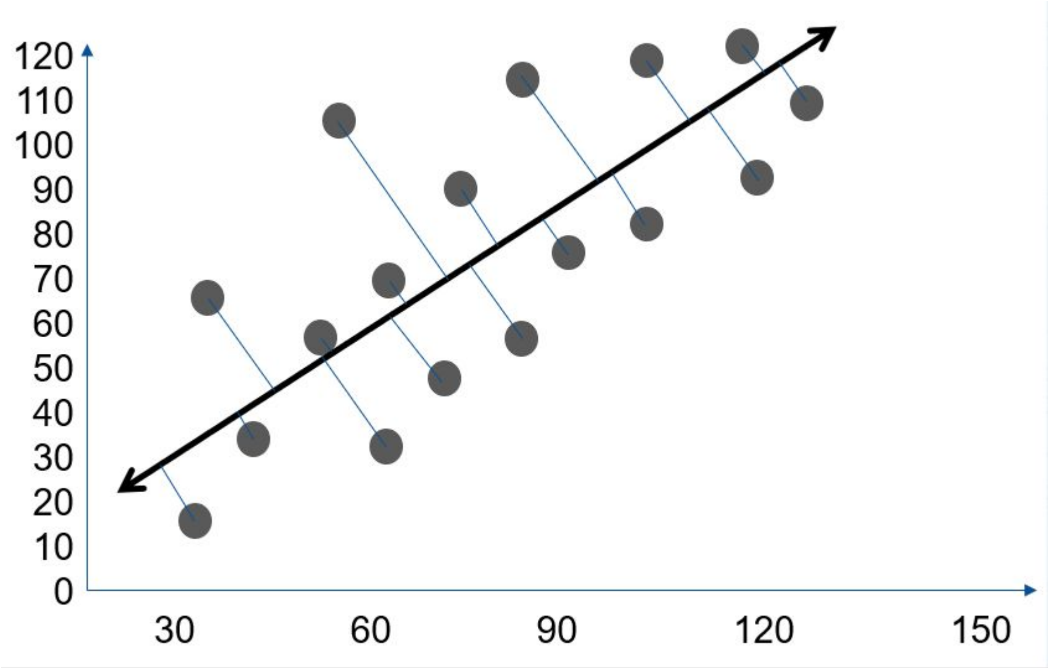
Fitting a line to data



- How well a set of data points fits a straight line can be measured by calculating the distance between the data points and the line.
- The total error between the data points and the line is obtained by squaring each distance and then summing the squared values.
- The regression equation is designed to produce the minimum sum of squared errors

Simple Linear Regression (cont’d)

Fitting a line to data

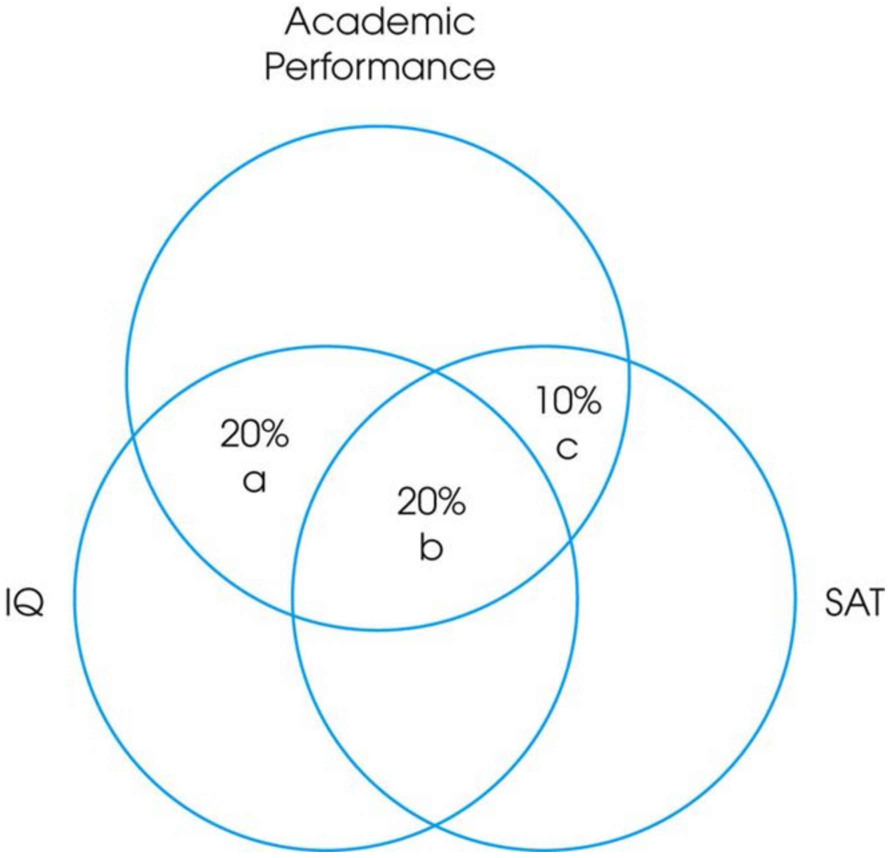


Select the “best” line that minimizes the sum of the distances of points to the fitted line –these distances are called the “residuals”

Multi-Linear Regression

- In the same way that simple linear regression produces an equation that uses values of X to predict values of Y, multiple regression produces an equation that uses two different variables (X_1 and X_2) to predict values of Y.
    - ***Multiple predictors***
- The equation is determined by a least squared error solution that minimizes the squared distances between the actual Y values and the predicted Y values.

Multi-Linear Regression (Cont’d)



- Predicting the variance in academic performance from IQ and SAT scores.
- The overlap between IQ and academic performance indicates that 40% of the variance in academic performance can be predicted from IQ scores.
- Similarly, 30% of the variance in academic performance can be predicted from SAT scores.
- However, IQ and SAT also overlap, so that SAT scores contribute an additional predication of only 10% beyond what is already predicted by IQ.

Multi-Linear Regression (cont'd)

General form of the equation: $\mathbf{y = \beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_nx_n+ \varepsilon}$

y: our response or dependent variable

β_0: the y-intercept coefficient

β_i : the regression coefficient (slope of the line)

x: our predictor/independent variable value

ε: the error (assumed independent)

Goal: ***We want to minimize the sum of square residuals for all predictors***

Multi-Linear Regression (cont'd)

General form of the equation: $\mathbf{y = \beta_0+\beta_1x_1+\beta_2x_2+\ldots+\beta_nx_n+ \varepsilon}$

y: our response or dependent variable

β_0: the y-intercept coefficient

β_i : the regression coefficient (slope of the line)

x: our predictor/independent variable value

ε: the error (assumed independent)

Goal: *We want to minimize the sum of square residuals for all predictors*

Multi-Linear Regression (cont'd)

Important Assumptions

- For a linear regression to be valid, the relationship between the response and predictor should be (nearly) linear.
- *Residuals* should be checked for the following:
  - Independence
  - Normally distributed
  - Equal variance (should occur on both sides of line)

Bias-Variance Trade-Off

- There are two critical characteristics of estimators to be considered i.e. **the bias and the variance**.
- **The bias** is the difference between the true population parameter and the expected estimator. It measures the accuracy of the estimates.
- **The Variance** measures the spread, or uncertainty, in the estimates.

Bias-Variance Trade-Off (cont'd)

- Both the bias & the variance are desired to be low, as large values result in poor predictions from the model.
- The OLS estimator has the desired property of being unbiased. However, it can have a huge variance. Specifically, this happens when:
    - ***The predictor variables are highly correlated with each other;***
    - ***There are many predictors***
- The solution to this is: **reduce variance at the cost of introducing some bias**. This approach is called **regularization**

Regularization:Ridge & LASSO Regression

Regularization revolves around modifying the loss function L; in particular, we add ***a regularization term that penalizes*** some specified properties of the model parameters.

$$L_{reg}(\beta) = L(\beta) + \lambda R(\beta),$$

where λ is a scalar that gives the weight (or importance) of the regularization term.

Fitting the model using the modified loss function L_{reg} would result in model parameters with desirable properties.

Regularization:Ridge & LASSO Regression

Ridge Regression (L2)

- Technique used when the data suffers from multicollinearity (independent variables are highly correlated)
- Regularization term penalizes the squares of the parameter magnitudes
- Penalty shrinks magnitude of all coefficients
- Larger coefficients strongly penalized because of the squaring

Regularization:Ridge & LASSO Regression

LASSO Regression (L1)

- Technique that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting model
- Regularization term penalizes the sum of the parameter absolute values (L1 penalty)
- Penalty selectively shrinks some coefficients & larger penalties result in coefficient values closer to zero.

Linear Regression : The Syntax

Import the class containing the regression method

```
from sklearn.linear_model import LinearRegression
```

Create an instance of the class

```
LR= LinearRegression()
```

Fit the instance on the data and then predict the expected value

```
LR= LR.fit(X_train, y_train)
y_predict= LR.predict(X_test)
```

Logistic Regression (LR)

- LR is a type of regression for binary prediction problems (the response classes are 0/1, TRUE/FALSE, etc.)
- It provides a smooth probabilistic transition between the two classes while also controlling for over-fitting
- Output values are forced between [0,1]

Logistic Regression (LR)

Whether or not a person smokes

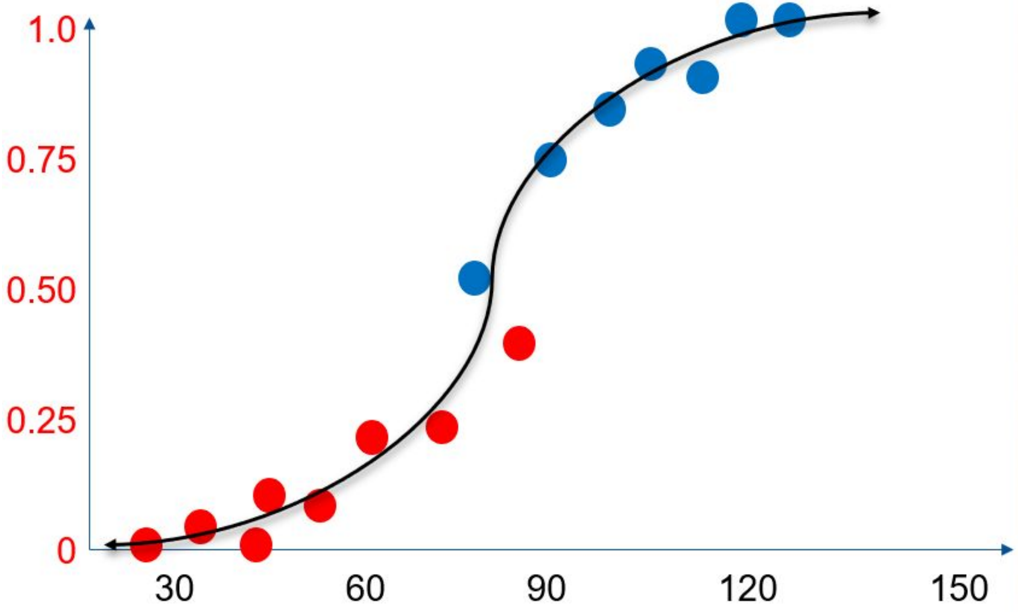
$$Y = \begin{cases} \text{Non} - \text{smoker} \\ \text{Smoker} \end{cases}$$

Binary Response

Success of a medical treatment

$$Y = \begin{cases} \text{Survives} \\ \text{Dies} \end{cases}$$

Logistic Regression (Cont.)



LR squeezes predictions to be probabilities between 0-1

New predictions can be assigned to one class or another using a cutoff (typically > 0.5)

Logistic Regression (cont’d)

Model Construction with Logistic Regression

- Given a set of binary class labels y∈{0,1}, we can provide the probability of a class for a *single* attribute using the equation:

$$P(y = 1|xj) = \frac{e^{b_0+b_1x_j}}{1+e^{b_0+b_1x_j}}.$$

- Above, ***b*** are regression coefficients, and $\boldsymbol{x_j}$ is the $\boldsymbol{j^{th}}$ predictor/feature.
- The more general form of the ”logit” function is:

Logit(p) = log(p/1-p) which is a sigmoid function

LR (Cont.): Odds Ratio

The logit function (not the regression coefficients) has a linear relationship with predictors.

This is hard to interpret so LR often provides “odds ratios” with results

Odds ratios are: prob. of success / prob. of failure

So an odds ratio of 1 is equivalent to a probability of 0.5

Odds Ratio Example

Example:

- The probability of passing is 0.8, the prob. of failing is 1 – 0.8 = 0.2.

  The odds of success are 0.8/0.2 = 4
  (e.g. “4-to-1 odds of passing”)

Logistic Regression (Cont.)

- With Linear Regression we tried to minimize the squares of the residuals, to get the best fitting line.
- But this doesn’t work for Logistic Regression.
    - Errors won’t be normally distributed (have 2 values)
- We use something called ***maximum likelihood*** to estimate what the β and α are.

Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) is an iterative process that estimates the best fitted equation.

- The iterative bit just means that we try lots of models until we get to a situation where tweaking the equation any further doesn’t improve the fit.
- MLE is kind of complicated, although the underlying assumptions are simple to understand, and very intuitive.
- The basic idea is that we find the coefficient value that makes the observed data most likely.

Maximum Likelihood Estimation

- MLE is a statistical method for estimating the coefficients of a model.
- The likelihood function (L) measures the probability of observing the particular set of dependent variable values (p_1, p_2, ..., p_n) that occur in the sample:
    - L = Prob (p_1* p_2* * * p_n)
- The higher the L, the higher the probability of observing the ps in the sample.

Non-binary variables

A lot of categorical variables are not binary though, what can we do with these?

Opinion poll responses

Ordinal Response

$$Y = \begin{cases} \text{Agree} \\ \text{Neutral} \\ \text{Disagree} \end{cases}$$

Which blood type does a person have, given the results of various diagnostic tests?

A
B
O
AB

Multinomial Logistic Regression

- Often we can recode them to a binary response.
  - You often see vote choice in Britain coded to Conservative or not (with the not category including Labour, the Liberal Democrats and everyone else).
- We could use something called **multinomial logistic regression**. This allows the dependent categorical variable to have more than two categories.

Logistic Regression : The Syntax

Import the class containing the regression method

```
from sklearn.linear_model import LogisticRegression
```

Create an instance of the class

```
LR= LogisticRegression(random_state=0)
```

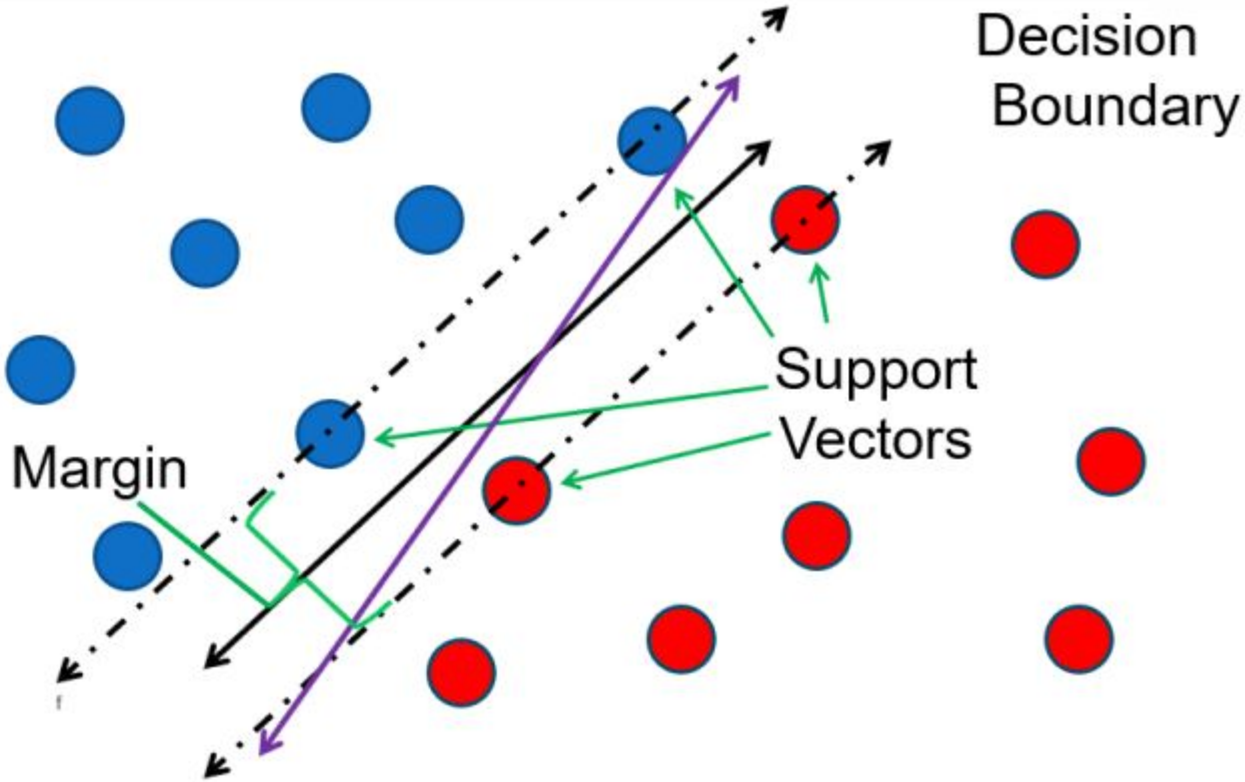
Fit the instance on the data and then predict the expected value

```
LR= LR.fit(X_train, y_train)
y_predict= LR.predict(X_test)
```

Support Vector Machine (SVM)

- Support vector machine (SVM) is a popular machine learning algorithm, often employed for classification or regression.
- The algorithm tries to find the optimal hyperplane (boundary) that can separate two classes of data.
- Here, **“optimal”** refers to the decision boundaries forming the widest margin between classes- the support vectors.

A Linear SVM Example



SVM Terminology

Support Vectors

- Support vectors are the data points, which are closest to the hyperplane.
- They define the separating line better by calculating margins.

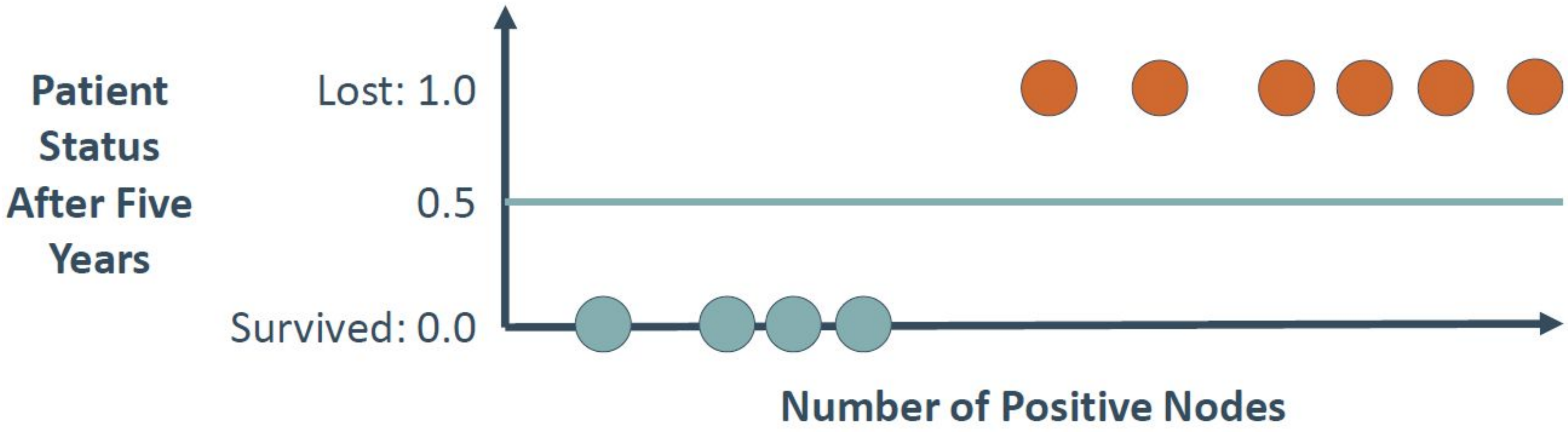
Hyperplane

- A hyperplane is a decision plane which separates between a set of objects having different class memberships

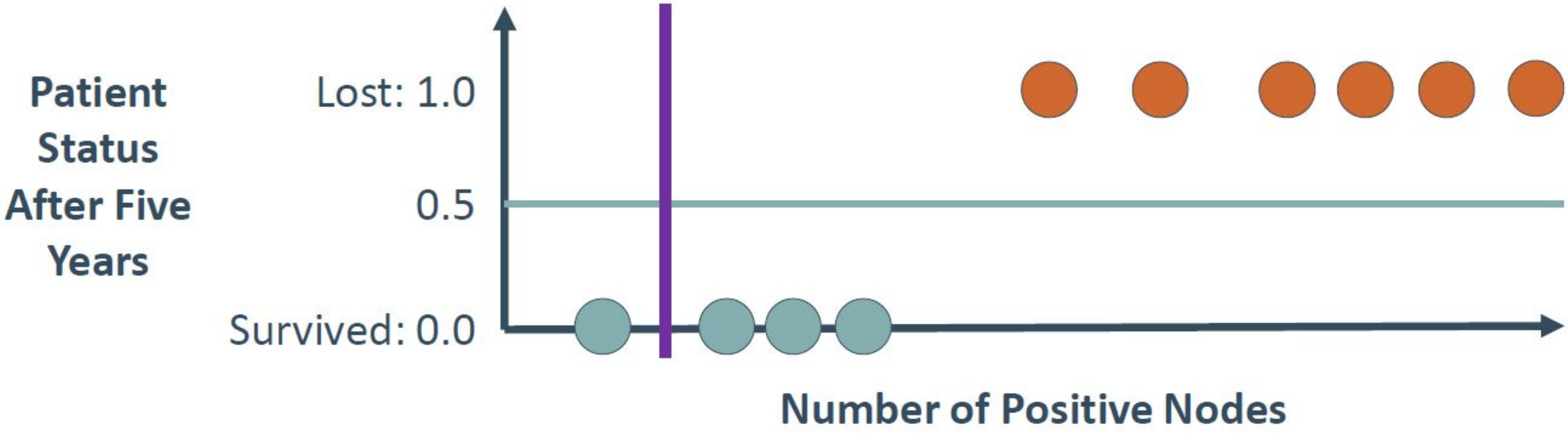
Margin

- A margin is a gap between the two lines on the closest class points.
- It is calculated as the perpendicular distance from the line to support vectors or closest points

Support Vector Machine (SVM)

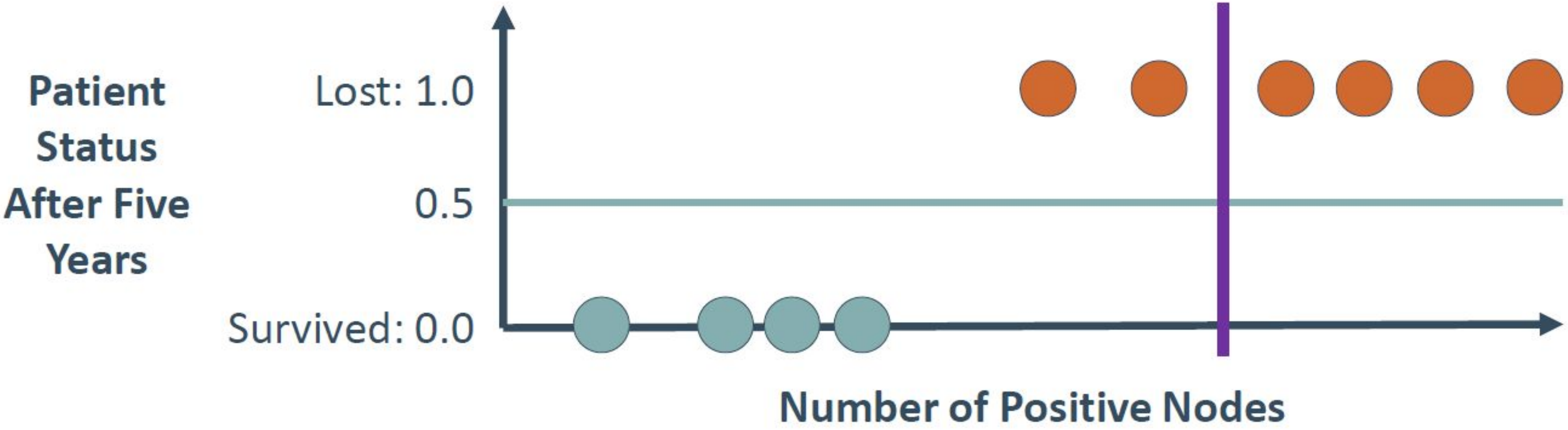


Support Vector Machine (SVM)



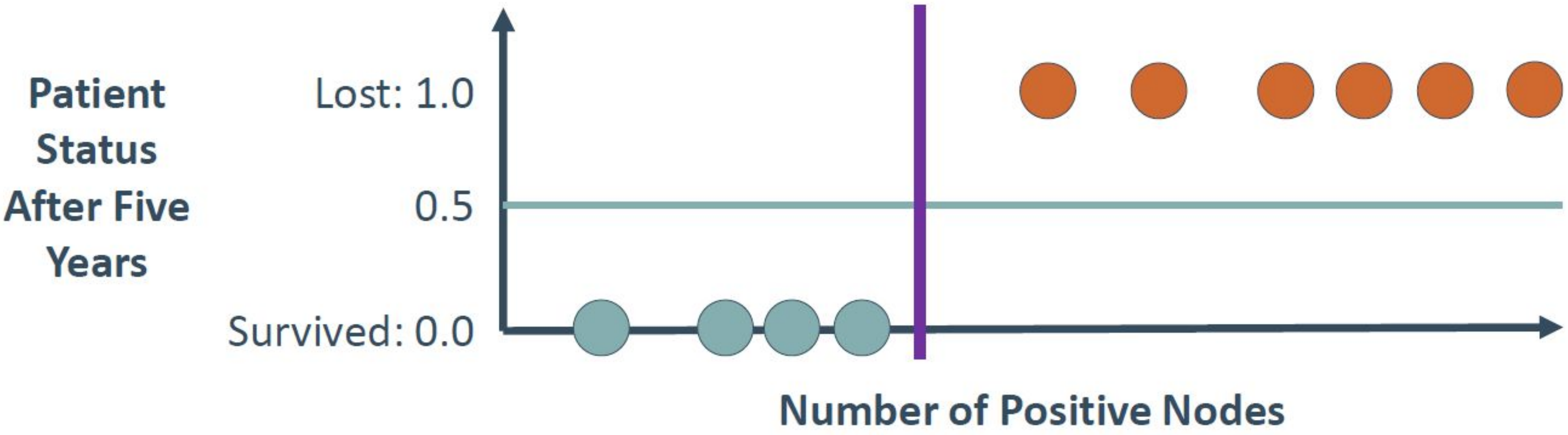
Three misclassifications

Support Vector Machine (SVM)



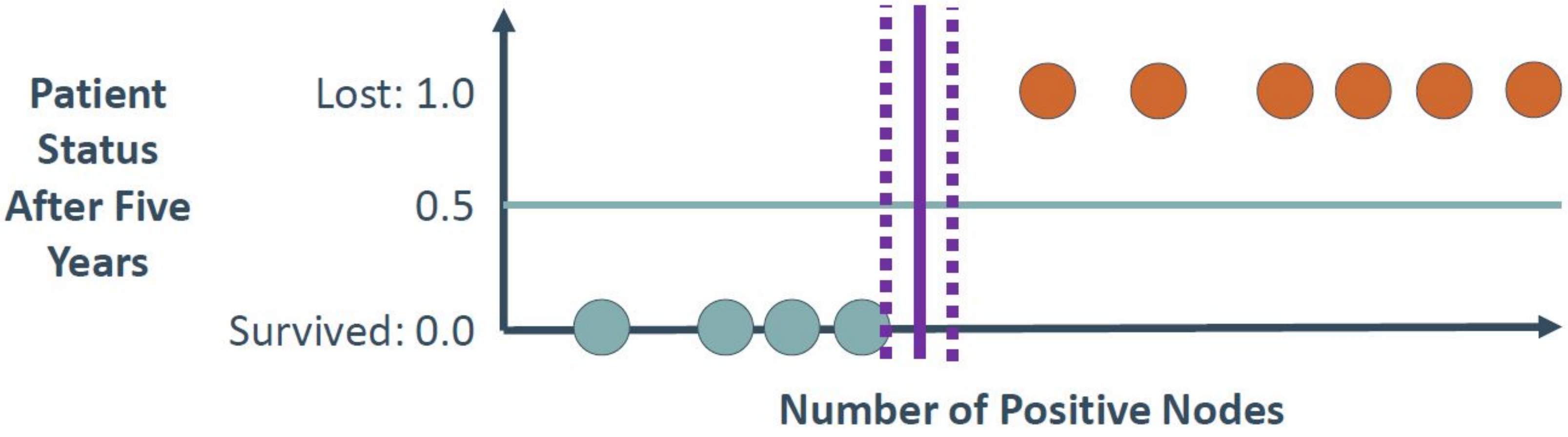
Two misclassifications

Support Vector Machine (SVM)



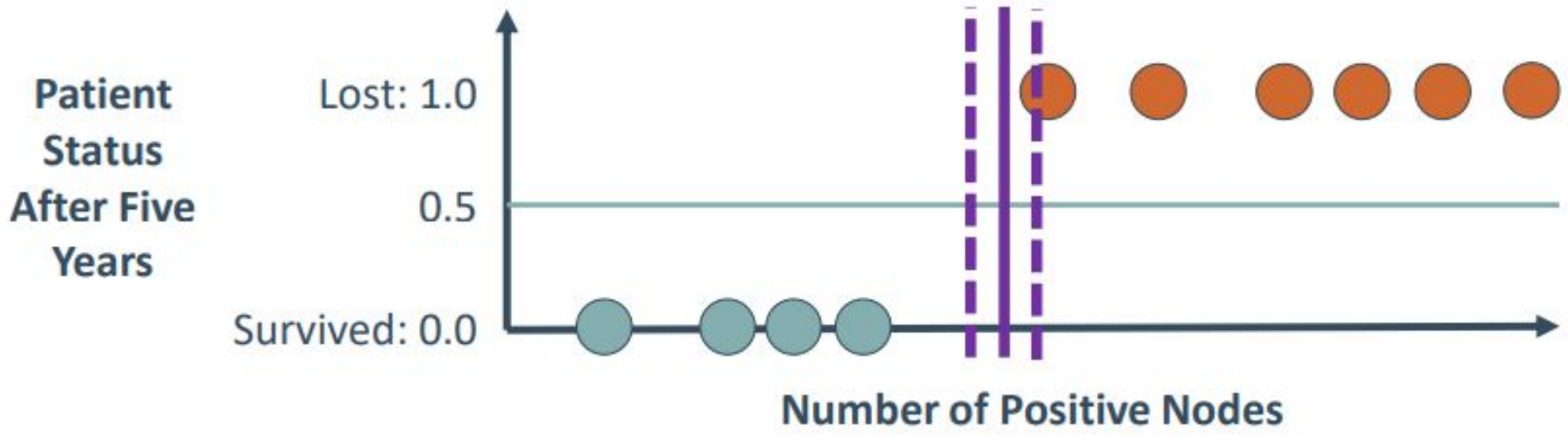
No misclassifications

Support Vector Machine (SVM)



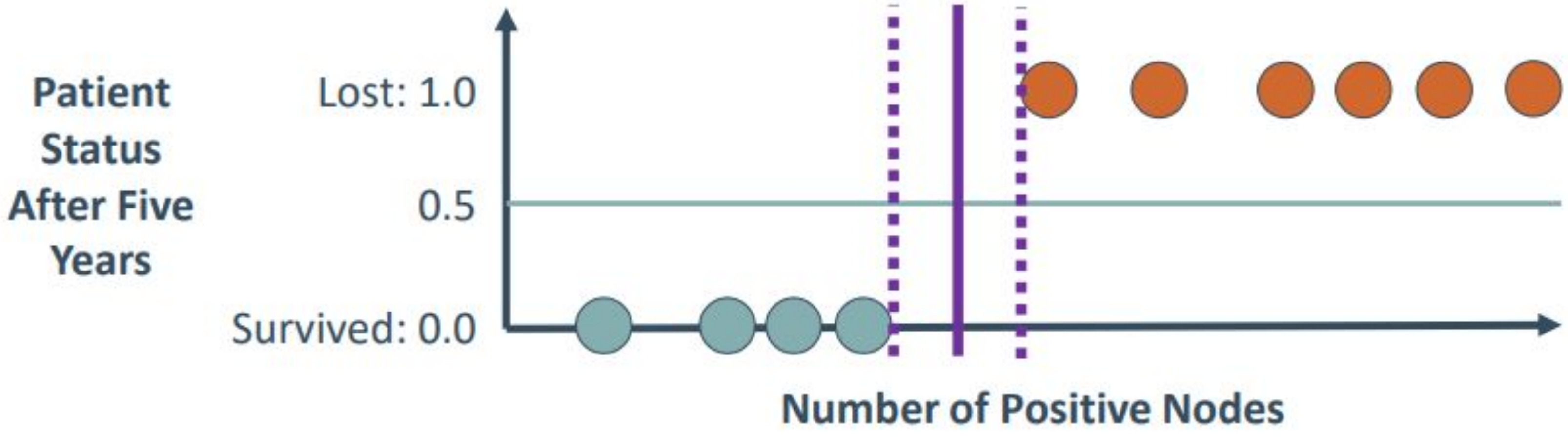
No misclassifications—but is this the best position?

Support Vector Machine (SVM)



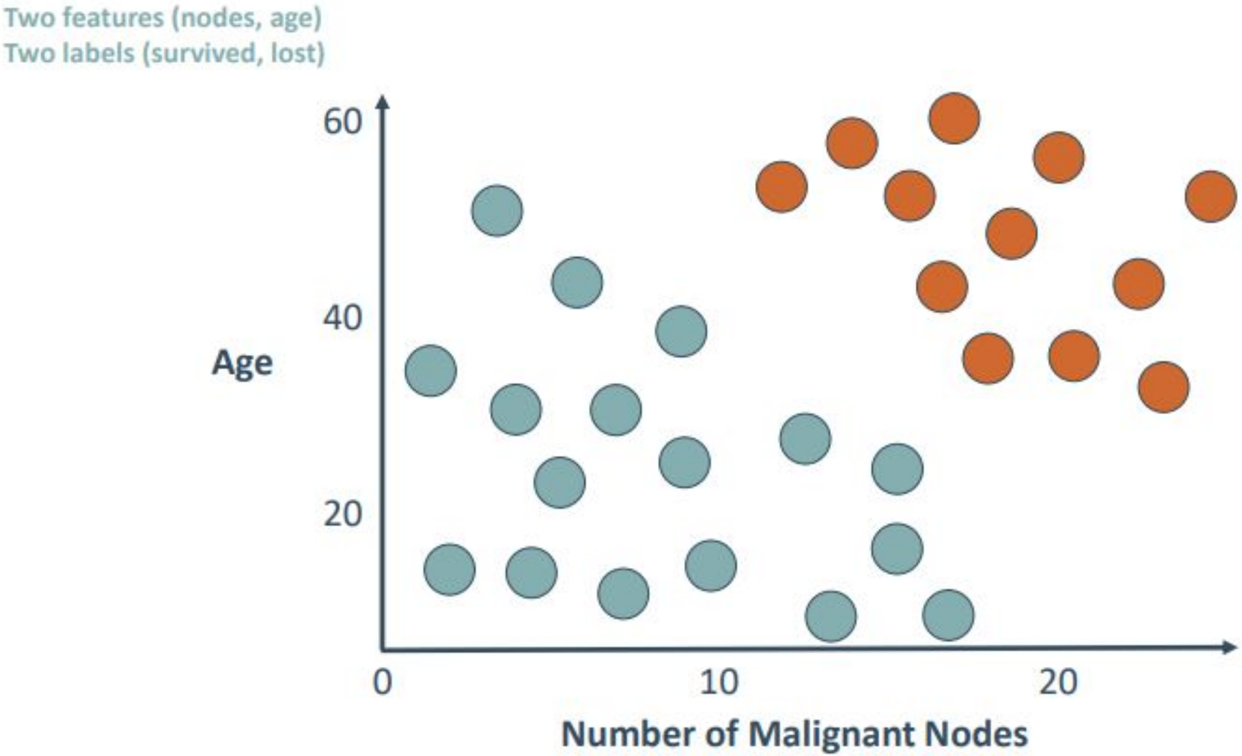
No misclassifications—but is this the best position?

Support Vector Machine (SVM)

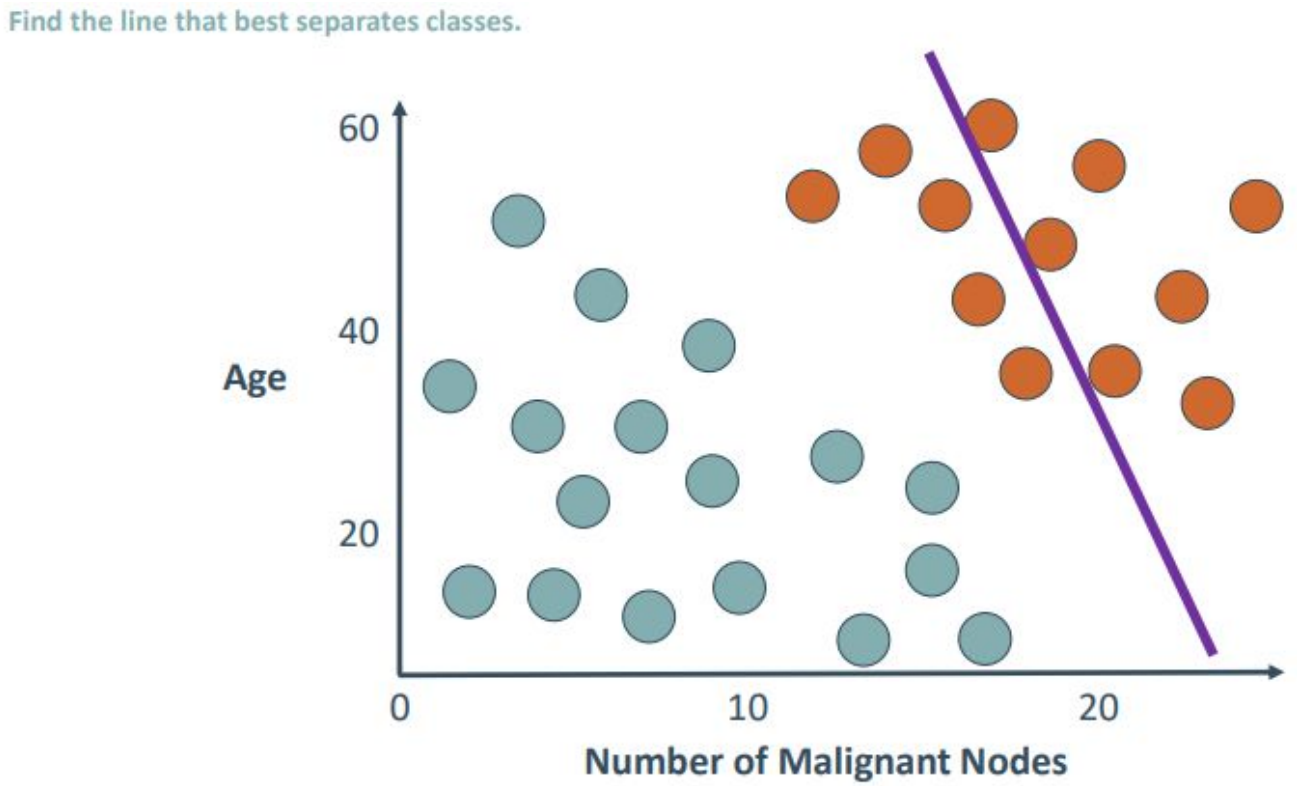


Maximize the region between classes.

Classification with SVMs

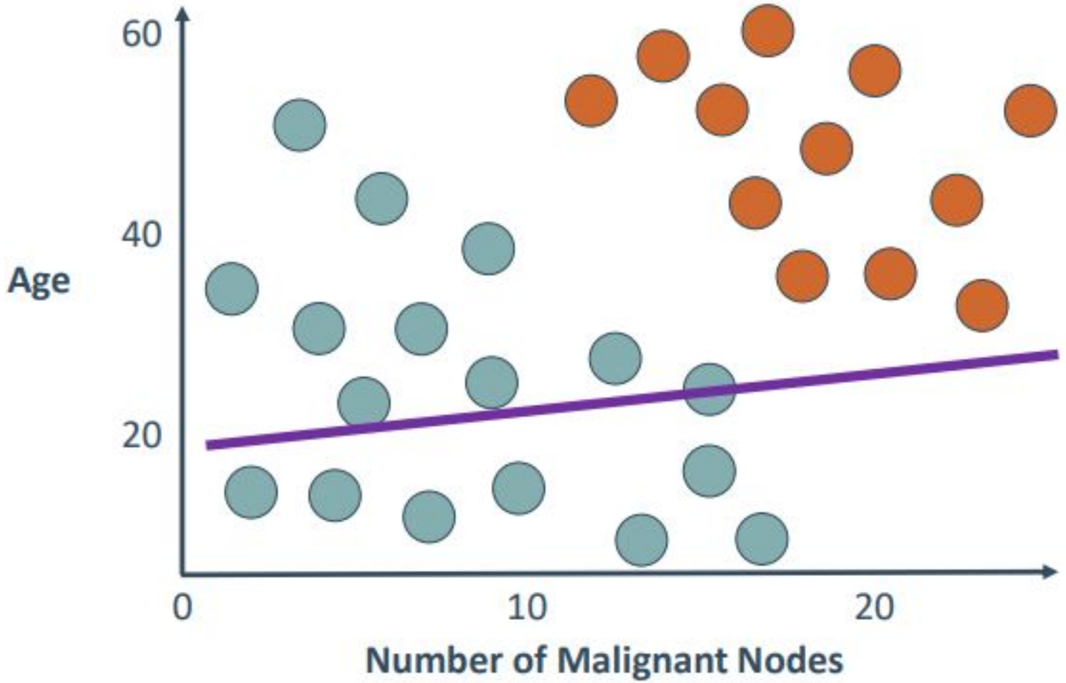


Classification with SVMs



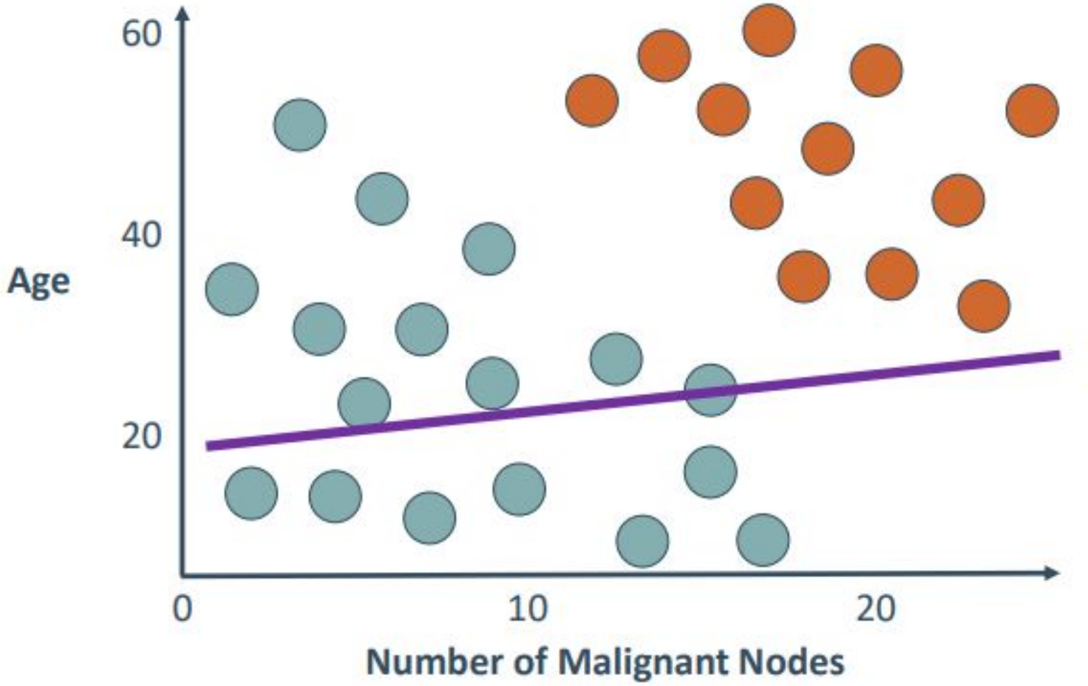
Classification with SVMs

Find the line that best separates classes.

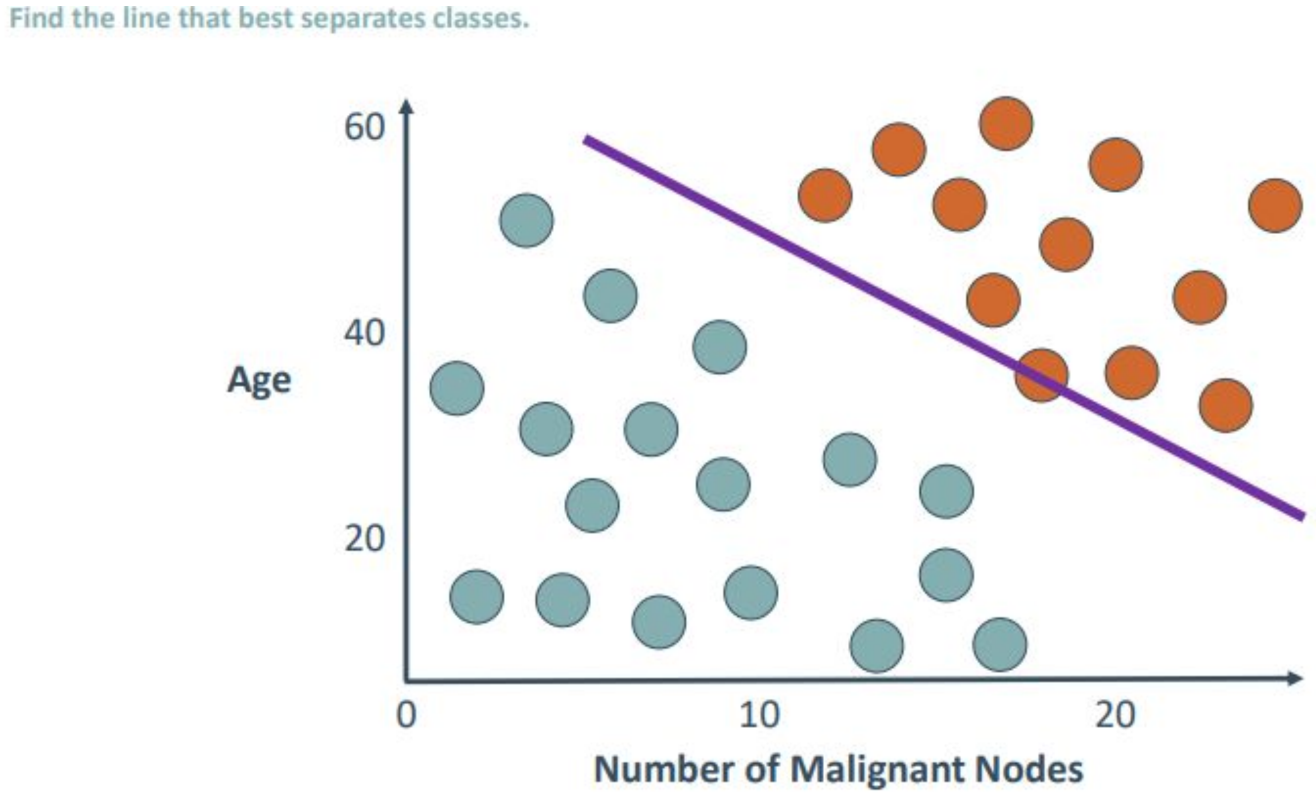


Classification with SVMs

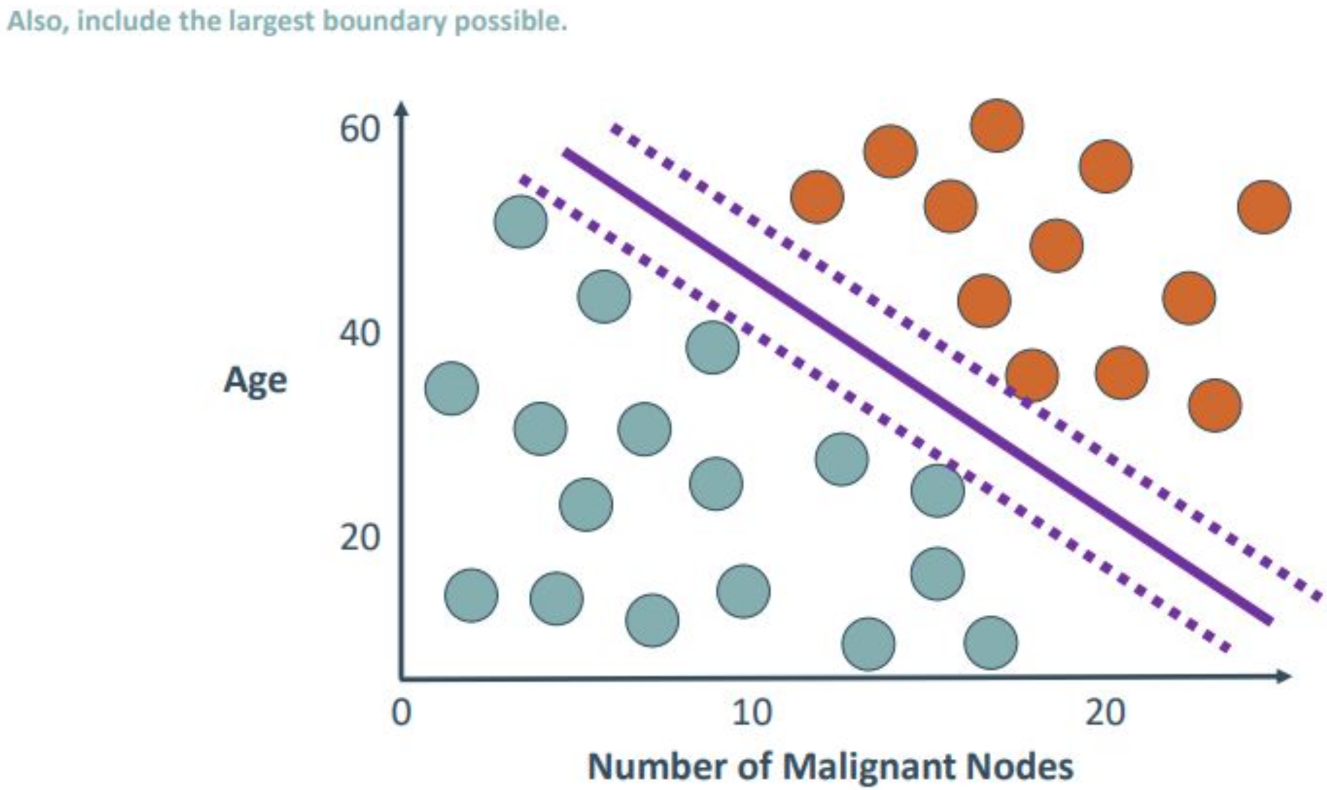
Find the line that best separates classes.



Classification with SVMs



Classification with SVMs



Linear SVM: The Syntax

Import the class containing the classification method.

from sklearn.svm import LinearSVC

Create an instance of the class

LinSVC = LinearSVC(penalty='l2', C=10.0)

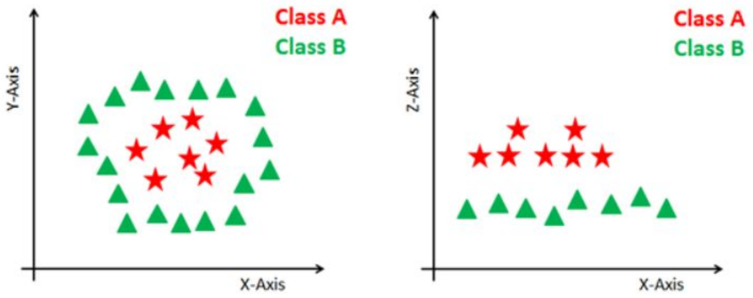
Fit the instance on the data and then predict the expected value

LinSVC = LinSVC.fit (X_train , y_train)

y_predict = LinSVC.predict (X_test)

Dealing with non-linear & inseparable planes

- Some problems can't be solved using linear hyperplane, as shown in the figure (left-hand side).
- In such situation, SVM uses a kernel trick to transform the input space to a higher dimensional space as shown on the figure (right-hand side)

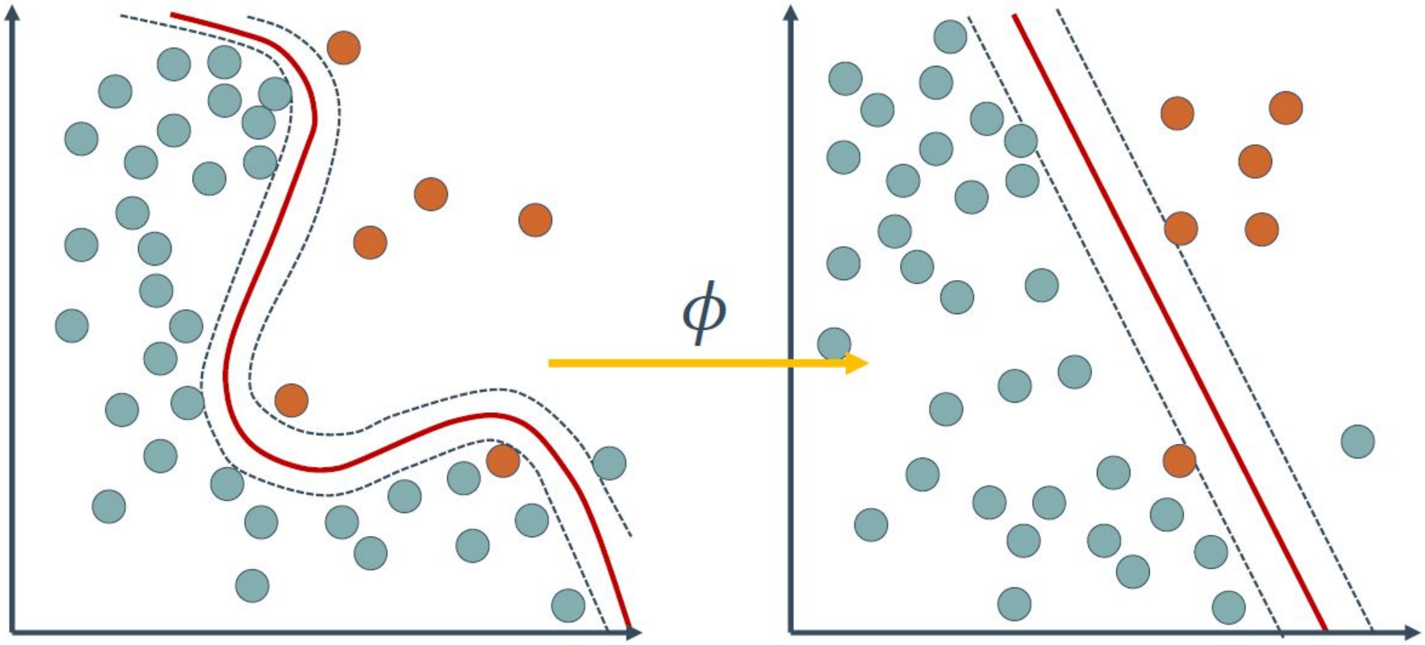


The data points are plotted on the x-axis and z-axis.

Now you can easily segregate these points using linear separation

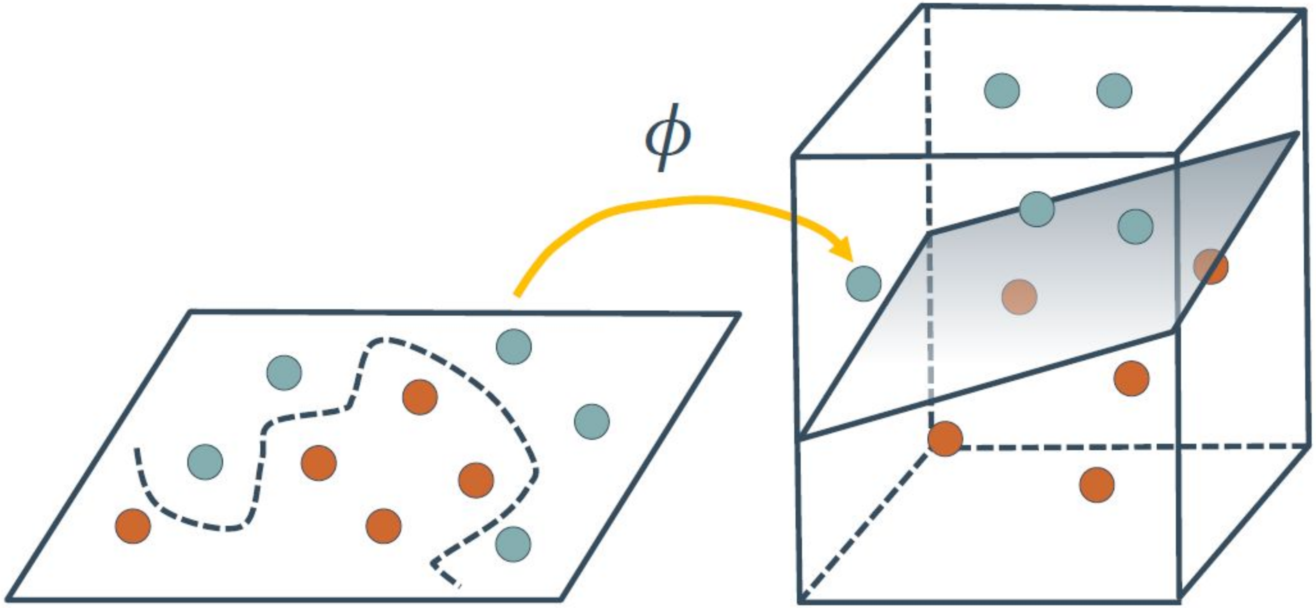
Non Linear Decision Boundaries with SVM

Non-linear data can be made linear with higher dimensionality.



The Kernel Trick

Transform data so it is linearly separable.



- SVM Gaussian Kernel (**Radial Basis Function Kernel)**
- Linear Kernel
- Polynomial Kernel
- etc

Advantages of SVM

- SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm.
- SVM use less memory because they use a subset of training points in the decision phase
- SVM works well with a clear margin of separation and with high dimensional space.

Disadvantages of SVMs

- SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes.
- It works poorly with overlapping classes
- Sensitive to the type of kernel used.

TAKE HOME ASSIGNMENT

- RESEARCH & READ ABOUT
    - THE K-NEAREST NEIGHBOUR ALGORITHM
    - NAÏVE BAYES ALGORITHM

References

- https://machinelearningmastery.com/linear-regression-for-machine-learning/
- Linear Regression Wiki page
- Lecture notes by Tom Mitchel
- Lecture notes by Dan Veltri
- Lecture notes by Jeff Witmer
- Hands-On Machine Learning with Scikit-Learn & TensorFlow
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- https://en.wikipedia.org/wiki/Maximum_likelihood_estimation
- https://en.wikipedia.org/wiki/Multinomial_logistic_regression
- https://people.umass.edu/biep540w/pdf/Whitlock%20Schluter%20Ch%2017%20regression.pdf

References

- Intel Academy Machine Learning Course
- https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

Thank you

If you have any questions feel free to email me:

sserurich@gmail.com