Introduction to Supervised Learning

What is Supervised Learning?

Supervised learning is a type of machine learning where a model is trained on a labeled dataset, meaning the input data is paired with the correct output.

The goal is for the model to learn the mapping between inputs and outputs so it can make accurate predictions on new, unseen data.

Linear Regression

Linear Regression

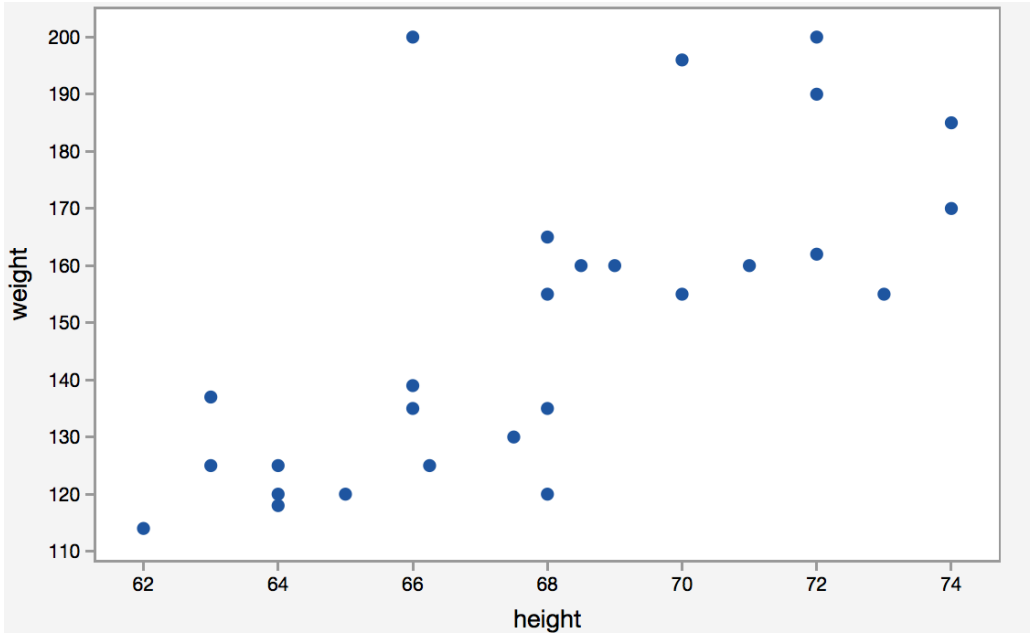
- Linear regression is perhaps one of the most well known and well understood algorithms in statistics and machine learning.
- Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y).
- More specifically, that y can be calculated from a linear combination of the input variables (x).

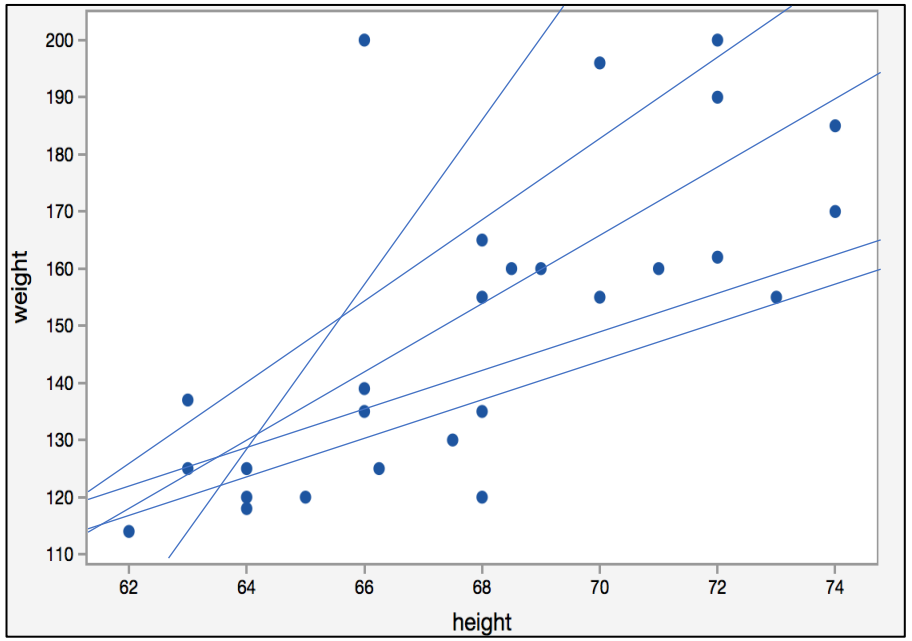
Linear Regression

- When there is a single input variable (x), the method is referred to as Simple Linear Regression.
- When there are multiple input variables, literature from statistics often refers to the method as Multiple Linear Regression.
- Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called Ordinary Least Squares.
- It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression.

Simple Linear Regression

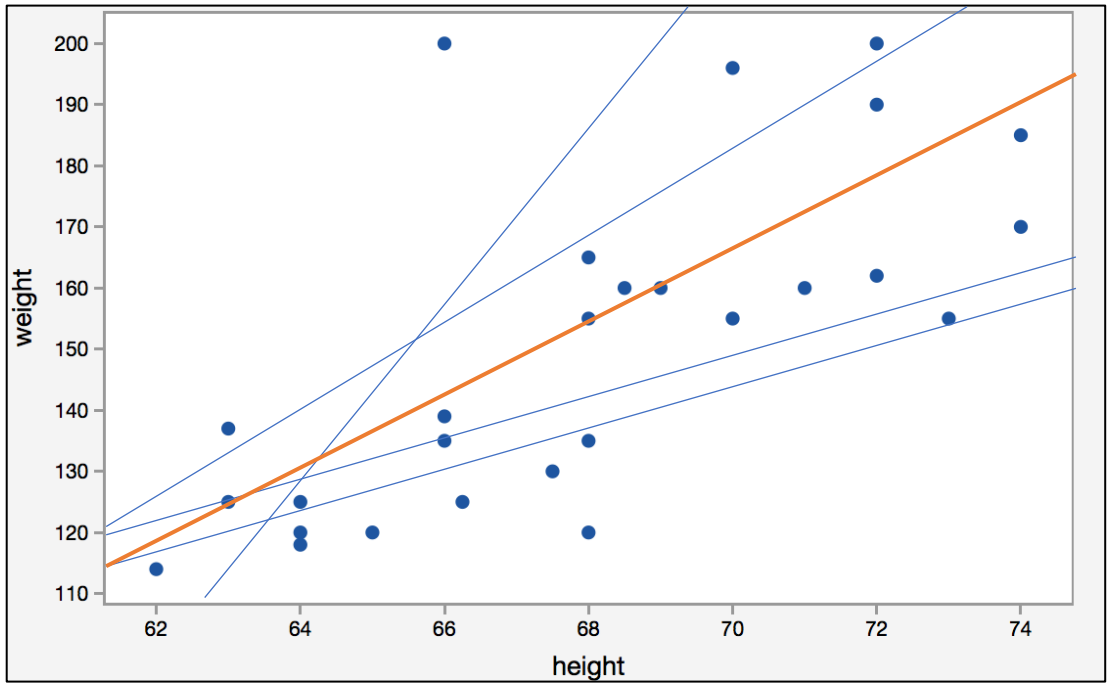
- Simple linear regression is useful for finding relationship between two variables.
- One is predictor or independent variable and other is response or dependent variable which is a quantitative variable.
- For example, relationship between height and weight.





Simple Linear Regression

Simple Linear Regression



Best Fit Line

Simple Linear Regression

- The core idea is to obtain a line that best fits the data.
- The best fit line is the one for which total prediction error (all data points) are as small as possible.
- The equation for this model for the population is as follows,

$$y = \beta_0 + \beta_1 x + \varepsilon$$

- The values β_0 and β_1 must be chosen so that they minimize the error.

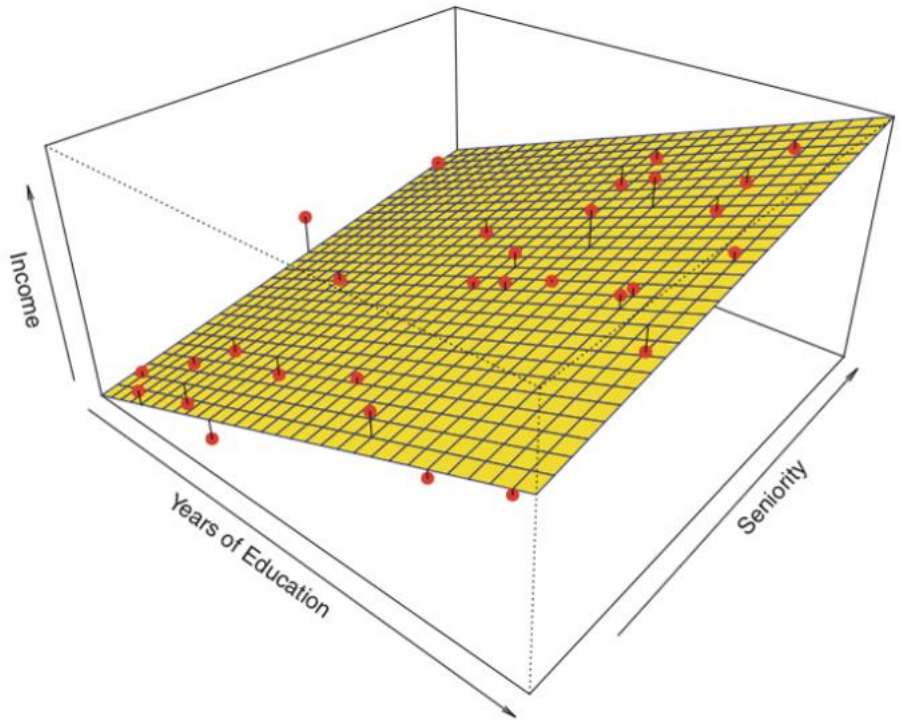
Multiple Linear Regression

- Consider here we have k independent variables.
- By minimizing this SSE, we can obtain the parameter estimations in here as well.
- The equation for this model is as follows,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k + \varepsilon$$

Multiple Linear Regression

- How this model is visualized.
- Consider a 2 variables with one response example.
- Here the Income is the response variable and Seniority and the Years Of Education are the independent variables.



Coefficient of Determination

- This R Squared Value is explaining the fraction of variation explained by the estimated model.
- In simple words how much of the variation of data, captured by this model.

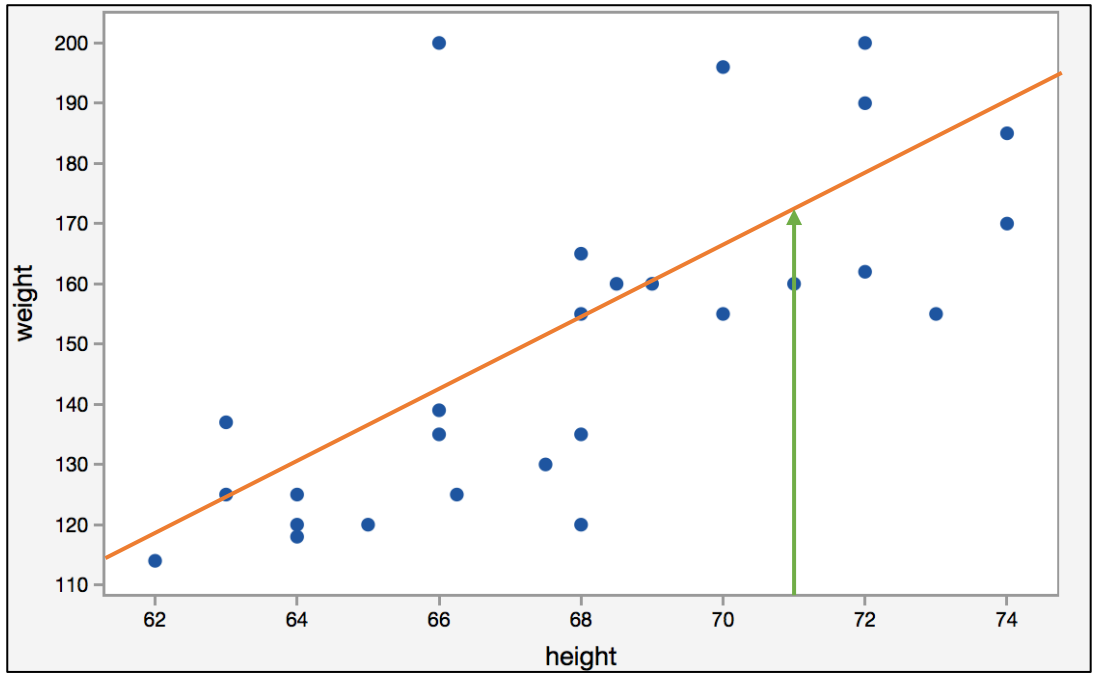
$$Total\ Sum\ of\ Squares\ (TSS) = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$Sum\ of\ Squares\ of\ Error\ (SSE) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$Coefficient\ of\ Determination = R^2 = \frac{TSS - SSE}{TSS}$$

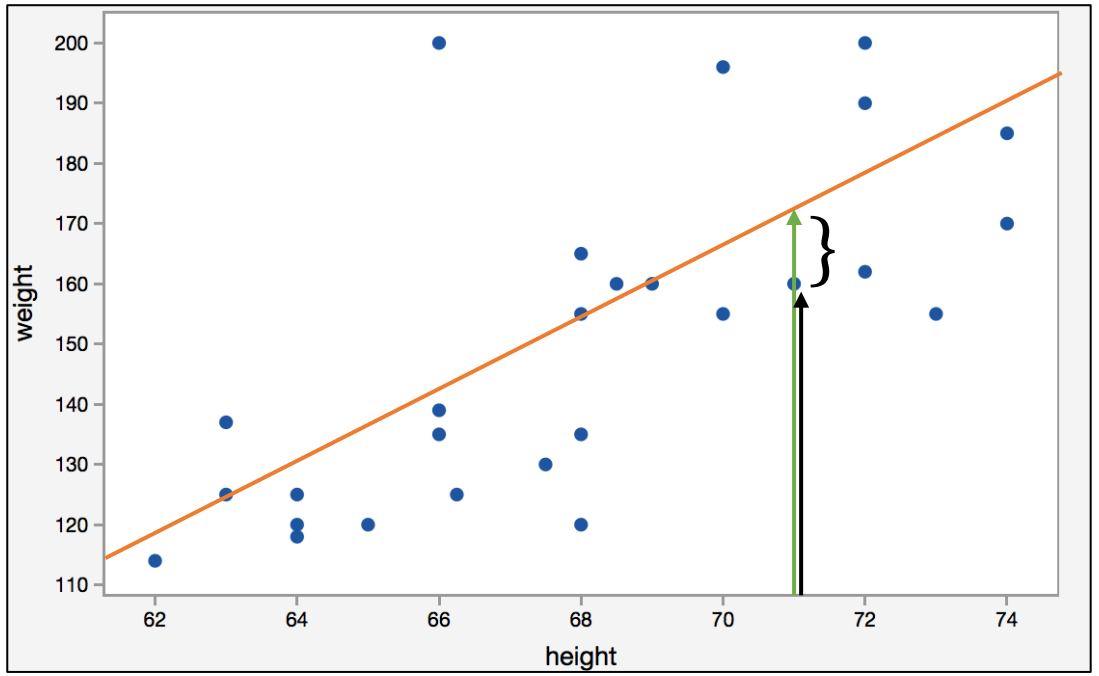
Predictions

- After fitting the model, the goal is to predict the response using independent data.



Predictions

- Always there is a prediction error.



Logistic Regression

Qualitative Responses

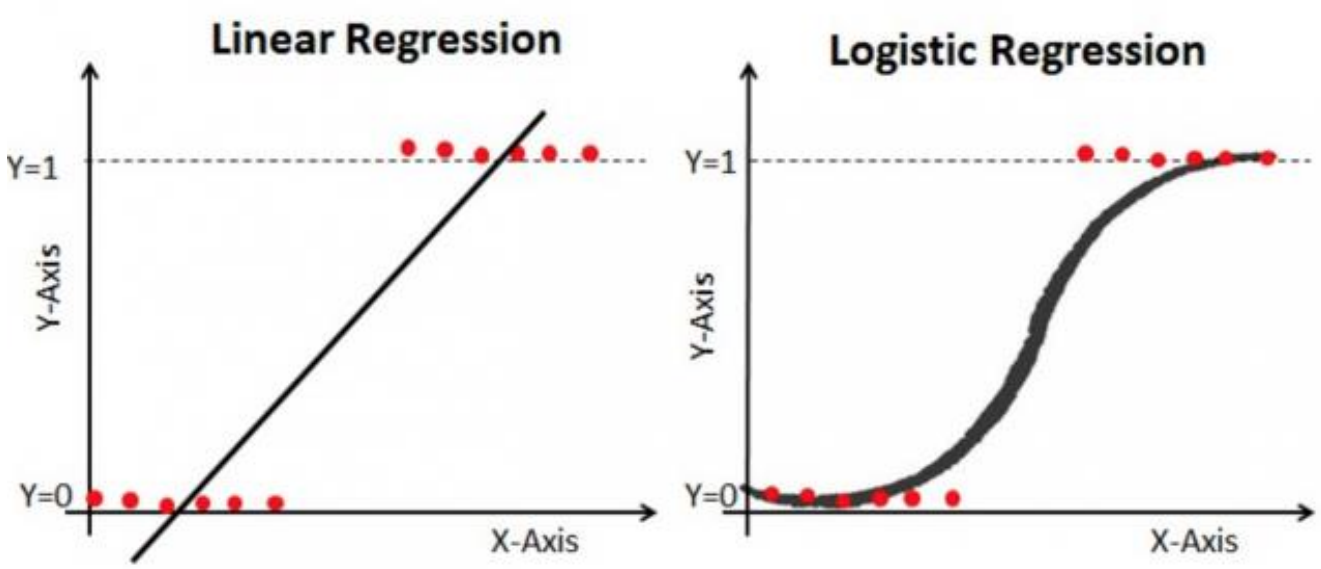
- What about the response variable is a categorical (Qualitative) variable?
- Ex:-
  - Yes/ No
  - True/ False
  - Good/ Bad
  - Pass/ Fail

  } Dichotomous Responses/ Binary Responses (Responses with two classes)

  - High/ Medium/ Low
  - Class 01/ Class 02/ Class 03/ Class 04
  - Red/ Green/ Blue

Logistic Regression

- Logistic regression is used when we have a dichotomous variable (Two levels categorical variable Ex- Yes/ No) as the response variable.
- In that case we cannot use linear regression for the predictions.



Logistic Regression

Logistic Regression

- Here what we can calculate as the output of the model is the probability of belonging to the category.
- Then we cannot work with this model, $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p$.
- We have to convert this mode such that the output is positive as well as the output should be lying between 0 and 1.
- The we have to deal with the Sigmoid function.

Logistic Regression

- The we have to deal with the Sigmoid function.

$$S(x) = \frac{1}{1 + e^{-x}}$$

- So we put above model inside to the sigmoid function and satisfy above discussed criteria.

$$P = S(y) = \frac{1}{1 + e^{-y}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p)}}$$

- So,

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p)}}$$

Logistic Regression

- Then we can show that,

$$log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p$$

- Here $log\left(\frac{P}{1-P}\right)$ is called as Logit.
- Parameters of the model will be estimated using Maximum Likelihood Estimation.
- When predicting some variable using this model a score will be given through this model and the probability should be found and then with that probability according to some cutoff value, the prediction will be done.

Logistic Regression

- Consider the example,

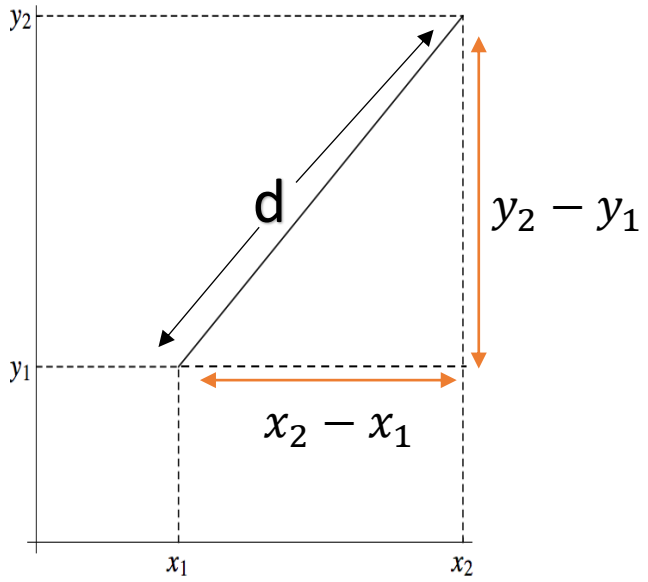
$$P \geq Some\ cutoff \longrightarrow Assign\ to\ class\ 01$$
$$P < Some\ cutoff \longrightarrow Assign\ to\ class\ 02$$

- Generally, this cutoff values is 0.5 in most cases.
- But this can be changed according to the situation.

K Nearest Neighbors (KNN)

Euclidean Distance

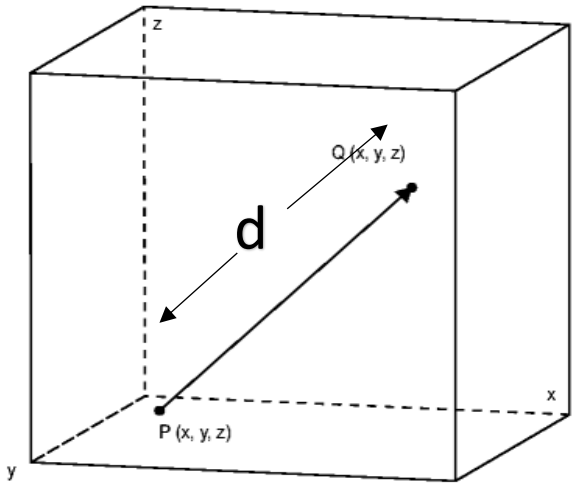
- It measures of the straight-line distance between two points in a Euclidean space, which is a space that obeys the principles of classical geometry.



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Euclidean Distance

- In a higher-dimensional Euclidean space, the formula extends similarly by considering the differences between the coordinates of the two points in each dimension and summing their squares, then taking the square root of the sum.



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

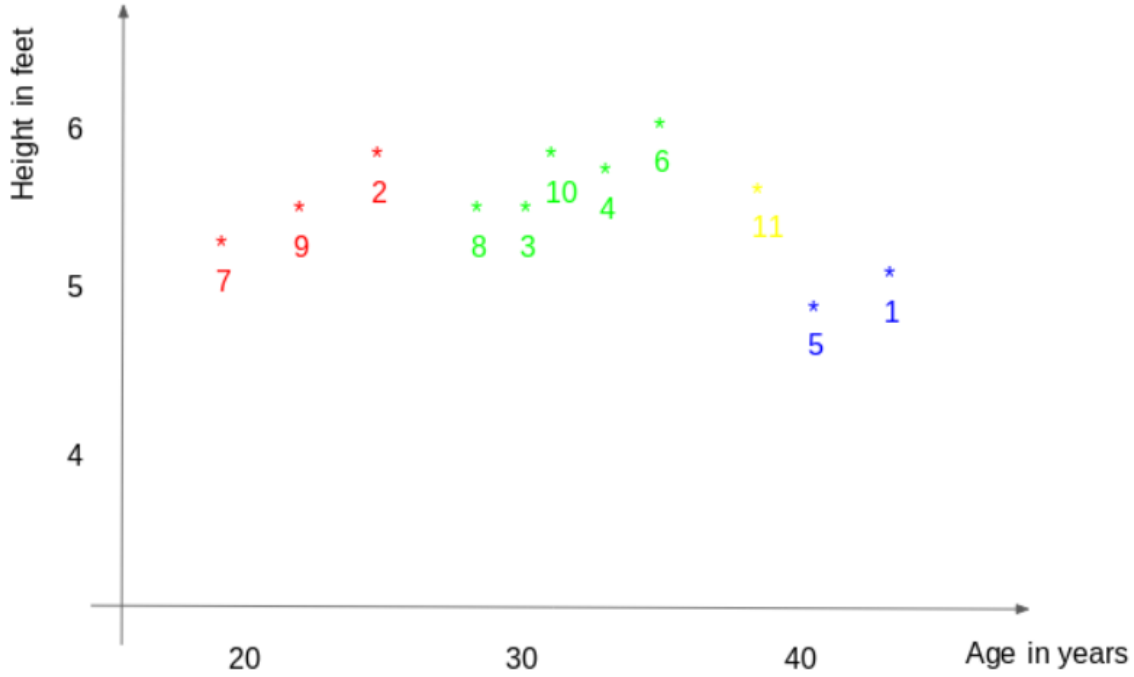
K- Nearest Neighbors

- Consider the following example.
- Think that we are going to fit a model using Height and Age for the response variable Weight.
- Think that we need to find the Weight for observation 11 using the given data.

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

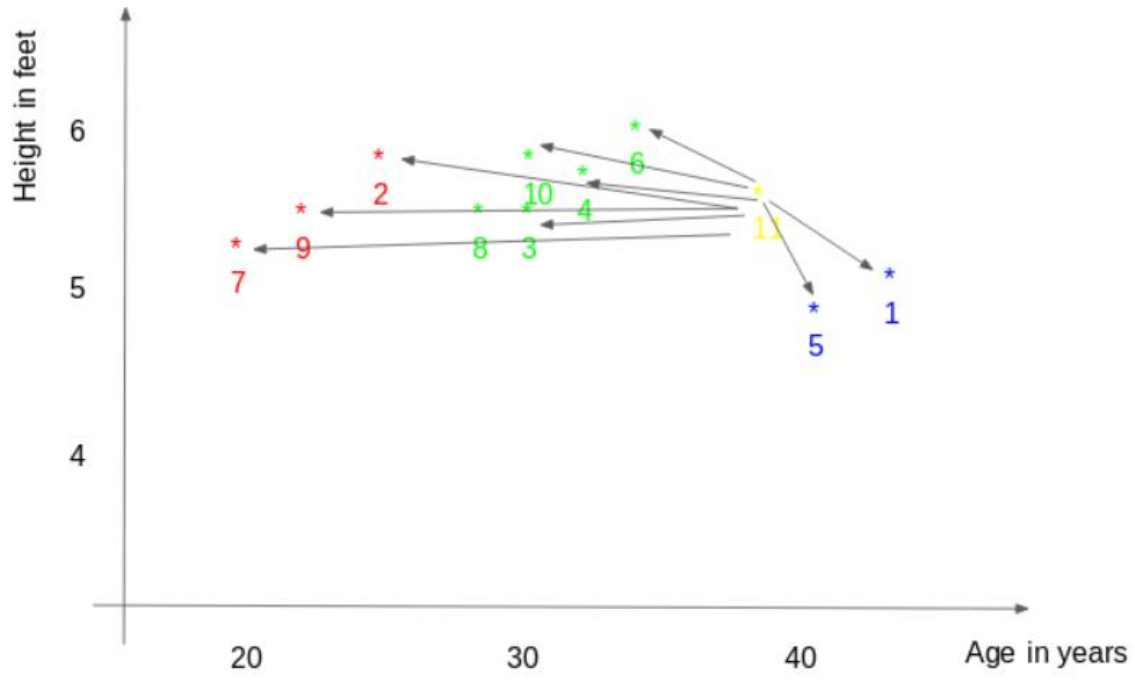
K- Nearest Neighbors

If we plot this,



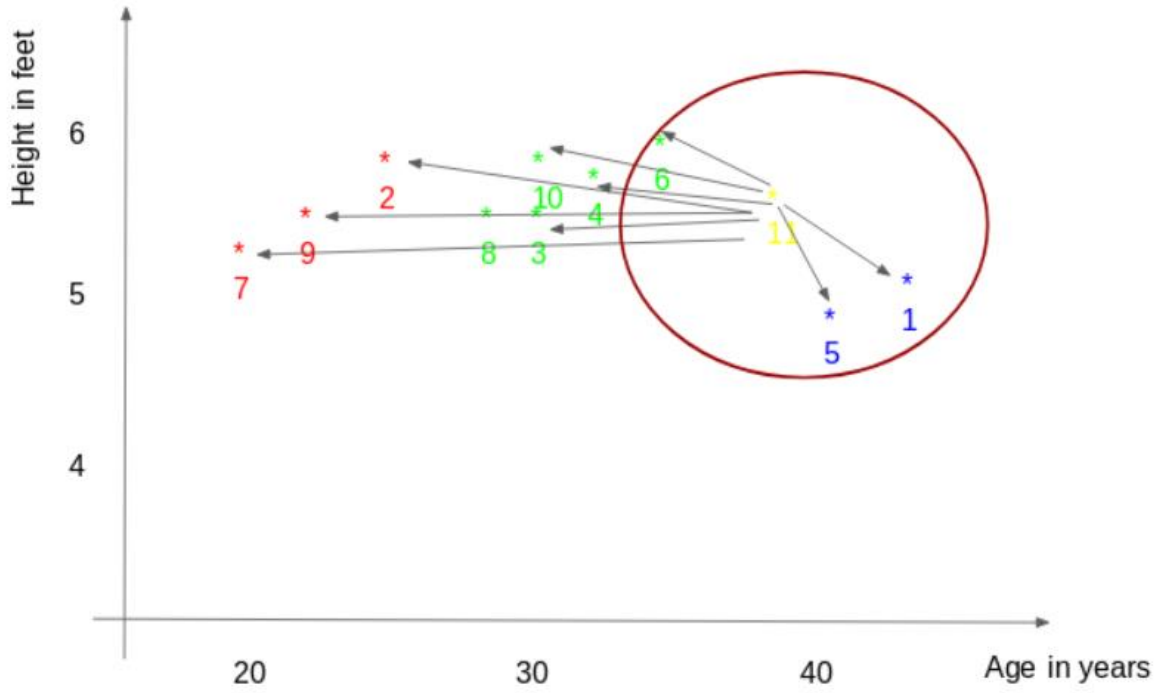
K- Nearest Neighbors

First, the distance between the new point and each training point is calculated.



K- Nearest Neighbors

- The closest k data points are selected (based on the distance).
- Consider here k=3, so in this example, points 1, 5, 6 will be selected if the value of k is 3.



K- Nearest Neighbors

- Then the average of these data points is the final prediction for the new point.
- Here, we have weight of ID11 is (77+72+60)/3 = 69.66 kg.
- If we have a classification case, the most frequent category will be chosen to assign.

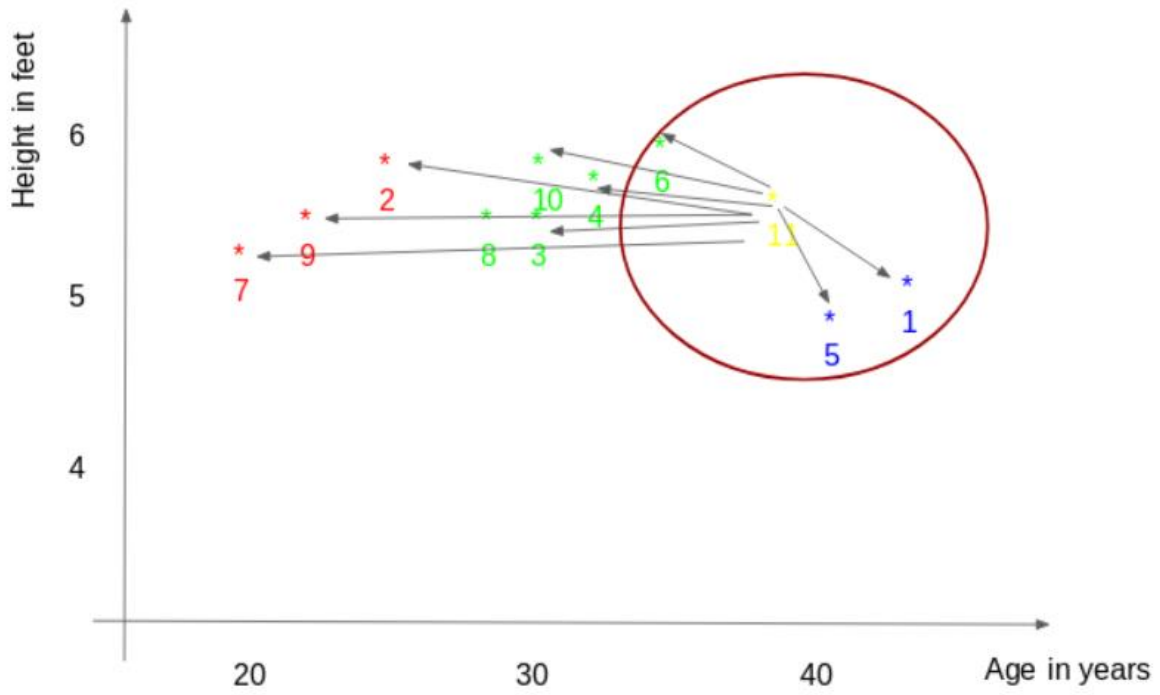
K- Nearest Neighbors

- Consider the following example.
- Think that we are going to fit a model using Height and Age for the response variable Weight.

ID	Height	Age	Class
1	5	45	A
2	5.11	26	A
3	5.6	30	B
4	5.9	34	C
5	4.8	40	B
6	5.8	36	A
7	5.3	19	C
8	5.8	28	A
9	5.5	23	B
10	5.6	32	C
11	5.5	38	

K- Nearest Neighbors

- The closest k data points are selected (based on the distance).
- Consider here k=3, so in this example, points 1, 5, 6 will be selected if the value of k is 3.



K- Nearest Neighbors

- If we have a classification case, the most frequent category will be chosen to assign.
- 11th observation belongs to Class A.

K- Nearest Neighbors

- To select the optimum k for the algorithm, the hyperparameter optimization can be used.

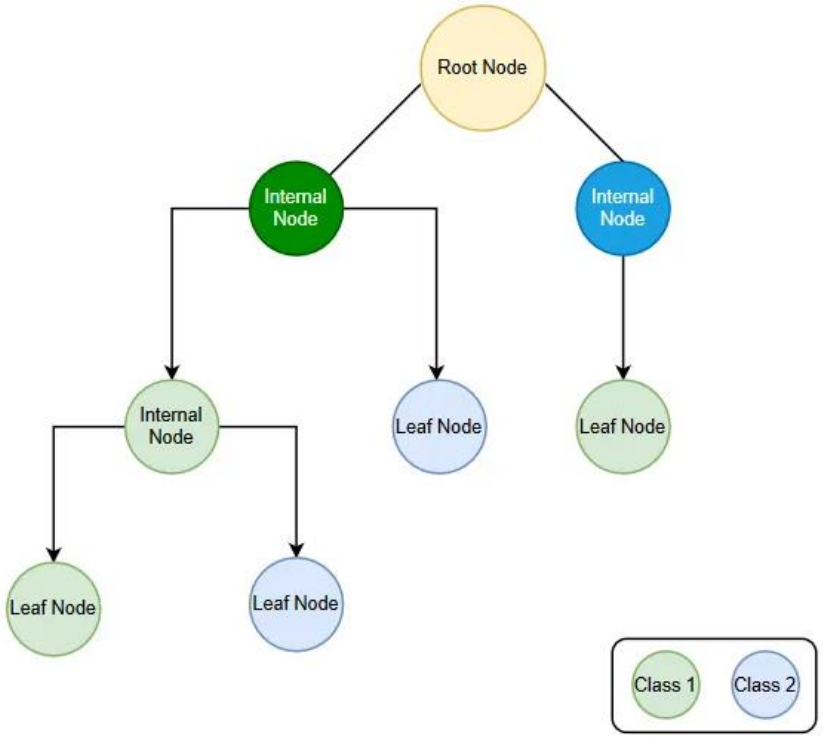
Classification and Regression Trees (CART)

Decision Trees

Decision Trees

- Decision tree is a non-parametric supervised learning technique, it is a tree of multiple decision rules, all these rules will be derived from the data features.
- It is one of most easy to understand & explainable machine learning algorithm.
- This ML algorithm is the most fundamental components of Random Forest, which are most popular & powerful ML algorithm.
- Two types
  - Classification trees
  - Regression trees

Structure of Classification Tree



- Each internal node represents a segment or region.
- With respect to tree analogy, segments or regions are nodes or leaves of the tree.

Structure of Classification Tree

- **Root Node**: This is the first node which is our training data set.
- **Internal Node**: This is the point where subgroup is split to a new sub-group or leaf node.
- **Leaf Node**: Final node from any internal node, this holds the decision.

About nodes in Decision Tree

- As mentioned before Decision tree is a tree like structure which will have nested nodes, the splitting of one node to another happens based on a threshold value of an attribute which we will discuss shortly in detail.
- Decision tree algorithm splits the training set (root node) to sub-groups - internal nodes & any internal node with final sub-group will be the leaf node.
- Which can be called as the Recursive partitioning.

About nodes in Decision Tree

- The splitting of node (root node) to sub-nodes happens based on purity, the Decision Tree algorithm split the node where it will find best homogeneity for the sub-nodes.
- If a Sub-node has all it’s class members then homogeneity will be higher.
- If the Sub-node has 5/5 class member distribution then homogeneity will be lowest and highest in case it is 8/2 or 9/1.

About nodes in Decision Tree

- To split a node Decision Tree algorithm needs best attribute & threshold value.
- Selection of best attribute & threshold value pair (f,t) happens based on below algorithms which will give you the purest nodes.

About nodes in Decision Tree

- The below algorithms helps to find the measurements of the best attributes:
  1. CART algorithm : **Gini Index**
  2. ID3 algorithm : **Information Gain**
  3. C4.5 algorithm : **Gain Ratio**

Gini Index

- The Gini Index is also known as Gini impurity. It is a measure of how mixed or impure a dataset is.
- The Gini impurity ranges between 0 and 1, where 0 represents a pure dataset and 1 represents a completely impure dataset(In a pure dataset, all the samples belong to the same class or category.
- On the other hand, an impure dataset contains a mixture of different classes or categories.

Gini Index

- In decision tree algorithms, a pure dataset is considered ideal as it can be easily divided into subsets of the same class.
- An impure dataset is more difficult to divide, as it contains a mixture of different classes.
- It means an attribute with a lower Gini index should be preferred.
- The lower the Gini impurity, the better the feature is for splitting the dataset.

Gini Index

Mathematically, The Gini Index is represented by

$$Gini\ impurity = 1 - \Sigma(p(i)^2)$$

Another commonly used formula is:

$$Gini\ impurity = \Sigma\left(p(i) * \left(1 - p(i)\right)\right)$$

- Where p(i) is the probability of a specific class and the summation is done for all classes present in the dataset.
- Both of these formulae are equivalent, and the result you get from these formulae would be the same.
- The first formula is more computationally efficient, therefore it is more commonly used.

Example - Gini Index

- let's consider a toy dataset with two classes "Yes" and "No" and the following class probabilities:
  - p(Yes) = 0.3 and p(No) = 0.7
- Using the first formula for Gini impurity:
  - Gini impurity = $1 — (0.3)^2 — (0.7)^2 = 0.42$

Using the second formula for Gini impurity:

Gini impurity = $(0.3 \times (1–0.3)) + (0.7 \times (1–0.7)) = 0.42$

Example - Gini Index

- As you can see, both formulae give the same result of 0.42
- In the example, the Gini impurity value of 0.42 represents that there is a 42% chance of misclassifying a sample if we were to randomly assign a label from the dataset to that sample.
- This means that the dataset is not completely pure, and there is some degree of disorder in it.

Example - Gini Index

- For example, if we have a dataset with two classes “Yes” and “No”, and the class probabilities are:
  - p(Yes) = 0.3 and p(No) = 0.7
- If we were to randomly pick a sample from the dataset, there would be a 30% chance that the sample belongs to the “Yes” class and a 70% chance that the sample belongs to the “No” class.
- If we were to randomly assign one of the class labels to the sample, there would be a ~~45%~~ 42% chance that we would assign the wrong label to the sample.

Example – Best Splitting

Age	Gender	Income	Credit Score	Buys_insurance
20	Male	High	Excellent	Yes
25	Female	High	Fair	No
30	Male	High	Excellent	Yes
35	Female	Medium	Excellent	Yes
40	Male	Low	Fair	No
45	Female	Low	Poor	No

- Let’s consider a toy dataset with the following features and class labels:
- The target class label is “Buys_insurance” and it can take two values “Yes” or “No”.

Example – Best Splitting

- We want to determine the best feature to use as the root node for the decision tree.
- To do this, we will calculate the Gini impurity for each feature and select the feature with the lowest Gini impurity.
- First, we will calculate the Gini impurity of the target class label “Buys_insurance” using the formula:
  - Gini impurity = $1 — (p(Yes))^2 - (p(No))^2$
- Where p(Yes) and p(No) are the class probabilities.

Example – Best Splitting

- In this dataset, the class probabilities are:
  - p(Yes) = 3/6 = 0.5 p(No) = 3/6 = 0.5
- so,
  - Gini impurity = 1 — $(0.5)^2$ — $(0.5)^2$ = 0.5
- Now, we will calculate the Gini impurity for each feature using the formula:

$$Gini\ impurity\ =\ 1 - \sum_{i=1}^{N} (p(Feature = Valuei))^2$$

Example – Gini impurity for feature “Gender”

Gender	Buys_insurance
Male	Yes
Female	No
Male	Yes
Female	Yes
Male	No
Female	No

Example – Weighted Gini impurity for feature “Gender”

Example – Gini impurity for feature “Gender”

- **Gender = Male** , Target has {Yes = 2, No = 1}
- $Gini\ Impurity(Male) = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = 0.444$

- **Gender = Female**, Target has : {Yes = 1, No = 2}
- $Gini\ Impurity(Female) = 1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right) = 0.444$

- **Weighted Gini impurity for feature “Gender”** $= \left(\frac{3}{6} * 0.444 + \frac{3}{6} * 0.444\right) = 0.444$

Example – Gini impurity for feature “Income”

Income	Buys_insurance
High	Yes
High	No
High	Yes
Medium	Yes
Low	No
Low	No

Example – Gini impurity for feature “Income”

_

Example – Weighted Gini impurity for feature “Income”

- **Income = High** , Target has {Yes = 2, No = 1}
- $Gini\ Impurity(High) = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = 0.444$

- **Income = Medium**, Target has : {Yes = 1, No = 0}
- $Gini\ Impurity(Medium) = 1 - \left(\left(\frac{1}{1}\right)^2 + \left(\frac{0}{1}\right)^2\right) = 0$ (Pure)

- **Income = Low**, Target has : {Yes = 0, No = 2}
- $Gini\ Impurity(Low) = 1 - \left(\left(\frac{0}{2}\right)^2 + \left(\frac{2}{2}\right)^2\right) = 0$ (Pure)

- **Weighted Gini impurity for feature “Income”** $= \left(\frac{3}{6} * 0.444 + \frac{1}{6} * 0 + \frac{2}{6} * 0\right) = 0.222$

Example – Gini impurity for feature “Credit Score”

Credit Score	Buys_insurance
Excellent	Yes
Fair	No
Excellent	Yes
Excellent	Yes
Fair	No
Poor	No

Example – Gini impurity for feature “Credit Score”

_

Example – Gini impurity for feature “Credit Score”

- **Credit Score = Excellent**, Target has {Yes = 3, No = 0}
- $Gini\ Impurity(Excellent) = 1 - \left(\left(\frac{3}{3}\right)^2 + \left(\frac{0}{3}\right)^2\right) = 0$ (Pure)

- **Credit Score = Fair**, Target has : {Yes = 0, No = 2}
- $Gini\ Impurity(Fair) = 1 - \left(\left(\frac{0}{2}\right)^2 + \left(\frac{2}{2}\right)^2\right) = 0$ (Pure)

- **Credit Score = Poor**, Target has : {Yes = 0, No = 1}
- $Gini\ Impurity(Poor) = 1 - \left(\left(\frac{0}{1}\right)^2 + \left(\frac{1}{1}\right)^2\right) = 0$ (Pure)

- **Weighted Gini impurity for feature “Credit Score”** $= 0$ (Pure)

Example – Gini impurity for feature “Age”

AGE	BUYS_INSURANCE
20	Yes
25	No
30	Yes
35	Yes
40	No
45	No

Example – Gini impurity for feature “Age”

Example – Gini impurity for feature “Age”

AGE IN ASCENDING ORDER	BUYS_INSURANCE	CONSECUTIVE AVERAGE AGES
20	Yes	(20+25)/2 = 22.5
25	No	(25+30)/2 = 27.5
30	Yes	(30+35)/2 = 32.5
35	Yes	(35+40)/2 = 37.5
40	No	(40+45)/2 = 42.5
45	No	

Example – Gini impurity for feature "Age"

- **Threshold = 22.5**
- **≤ 22.5 → {Yes = 1, No = 0}** $Gini\ Impurity(T \leq 22.5) = 0$ (Pure)
- **> 22.5 → {Yes = 2, No = 3}** $Gini\ Impurity(T > 22.5) = 1 - \left(\left(\frac{2}{5}\right)^2 + \left(\frac{3}{5}\right)^2\right) = 0.48$
- **Weighted Gini impurity** $\boldsymbol{for\ T = 22.5} = \left(\frac{1}{6} * 0 + \frac{5}{6} * 0.48\right) = 0.40$

- **Threshold = 27.5**
- **≤ 27.5 → {Yes = 1, No = 1}** $Gini\ Impurity(T \leq 27.5) = 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 0.5$
- **> 27.5 → {Yes = 2, No = 2}** $Gini\ Impurity(T > 27.5) = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 0.5$
- **Weighted Gini impurity** $\boldsymbol{for\ T = 27.5} = \left(\frac{2}{6} * 0.5 + \frac{4}{6} * 0.5\right) = 0.50$

Example – Gini impurity for feature “Age”

- **Impurity Values for Average Age**
- 22.5 → 0.40
- 27.5 → 0.50
- 32.5 → 0.444
- 37.5 → **0.25** (lowest impurity)
- 42.5 → 0.40

- We can select the Age Threshold value as 37.5 which gives the lowest Gini Impurity value for feature “Age”

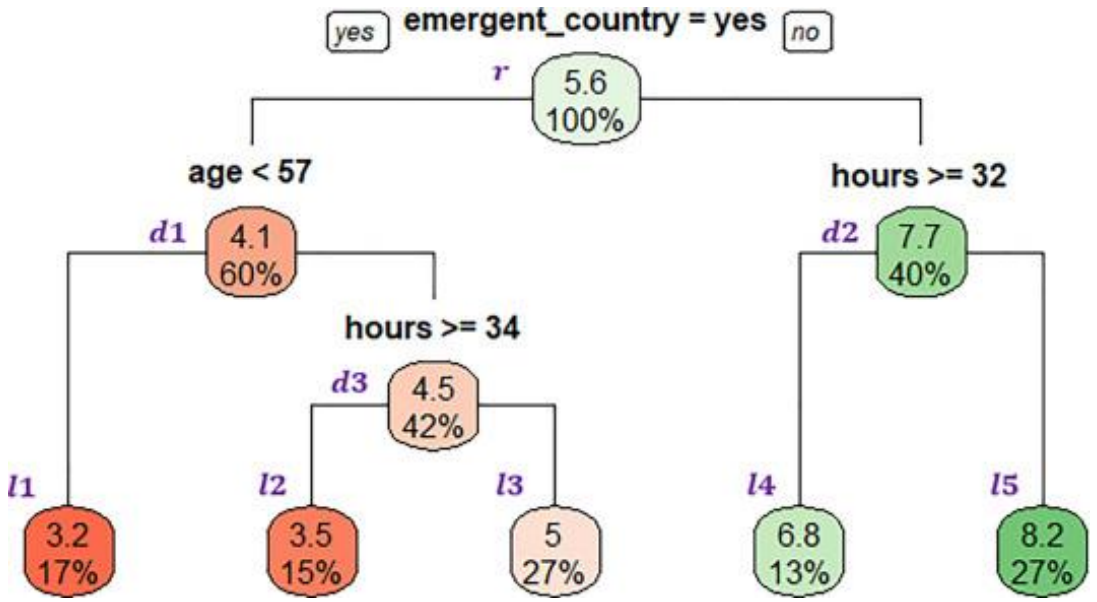
Example – Best Splitting

From the above calculations, we can see that the feature “Credit Score” has the lowest Gini impurity of 0. Implicating it is the purest feature in the dataset.

So, we can select “Credit Score” as the root node for the decision tree.

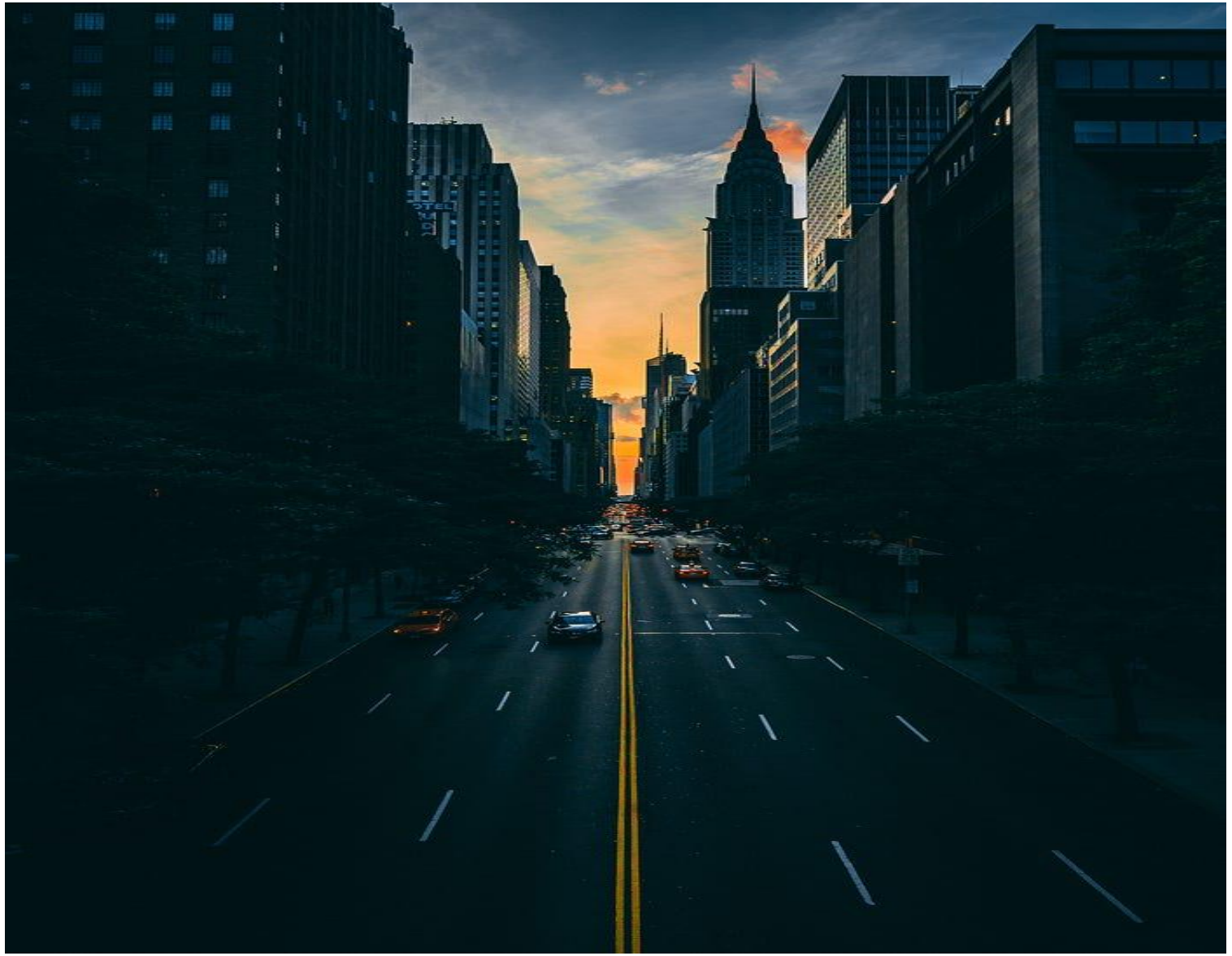
Structure of Regression Tree

- Regression can also be performed using Decision Trees/
- For the best splits **Reduction in Variance** method can be used.



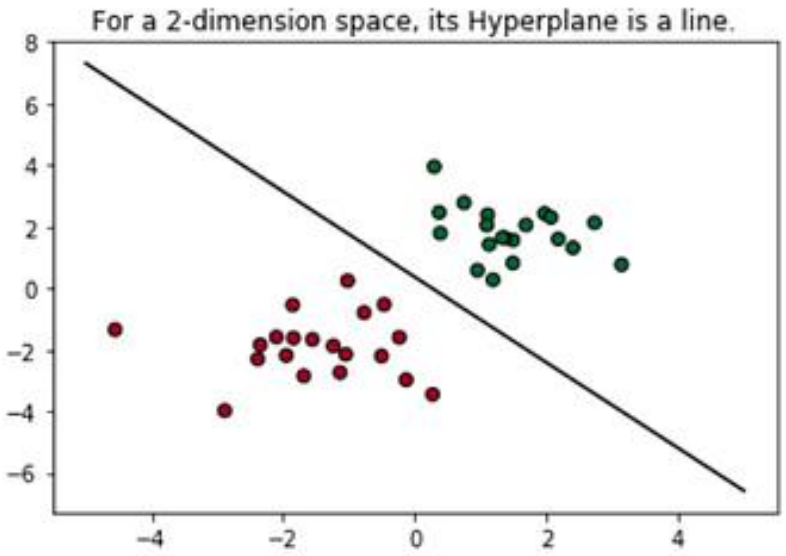
Support Vector Machines

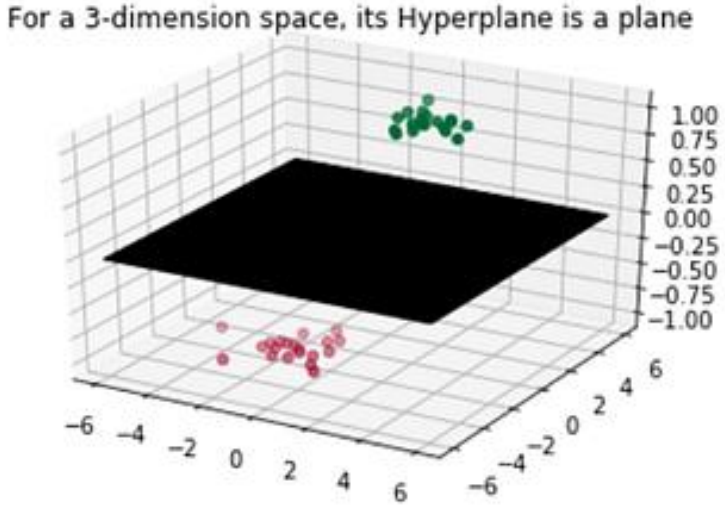
Look at This Image



Support Vector Machines

- Back to the ML world, those cars, building, pedestrians all become dots now. Red dots stand for objects on the left side of the street; green dots stand for those on the right.
- The street becomes dashed and solid lines.
- Support Vector Machine (the “road machine”) is responsible for finding the decision boundary to separate different classes and maximize the margin.
- Margins are the (perpendicular) distances between the line and those dots closest to the line.



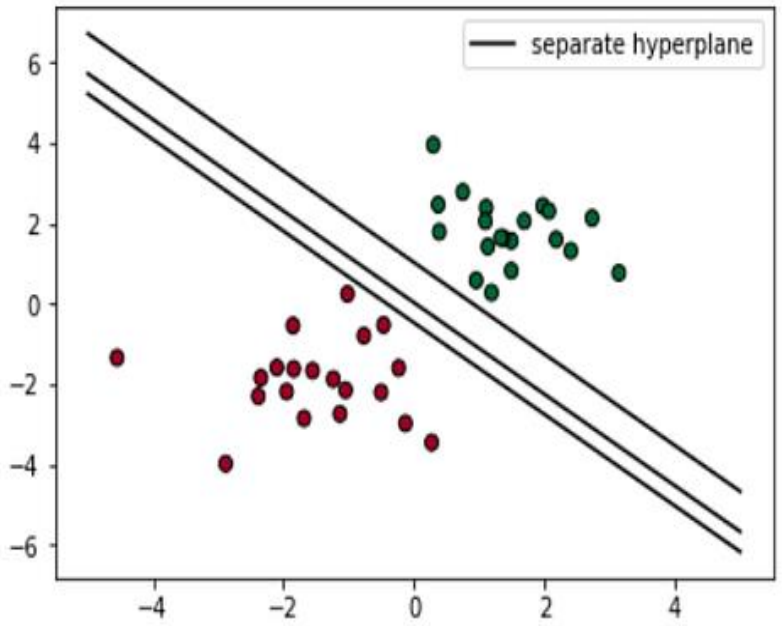


What is a Hyperplane?

- Hyperplane is an (n minus 1)-dimensional subspace for an n-dimensional space.
- For a 2-dimension space, its hyperplane will be 1-dimension, which is just a line.
- For a 3-dimension space, its hyperplane will be 2-dimension, which is a plane that slice the cube.

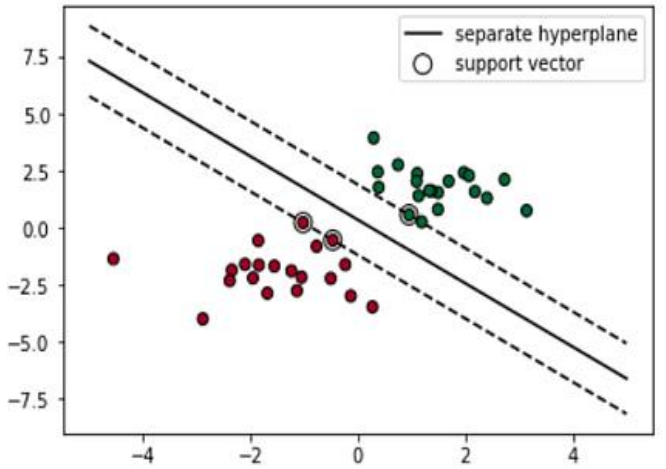
What is a Separating Hyperplane?

- Assuming the label y is either 1 (for green) or -1 (for red), all those three lines below are separating hyperplanes.



What is margin?

- Let’s say we have a hyperplane — line X
- Calculate the perpendicular distance from all those 40 dots to line X, it will be 40 different distances
- Out of the 40, the smallest distance, that’s the margin



SVM in Linear Separable Cases

- Obviously, infinite lines exist to separate the red and green dots in the example above.
- SVM needs to find the optimal line with the constraint of correctly classifying either class:
    1. Follow the constraint: only look into the separate hyperplanes(e.g. separate lines), hyperplanes that classify classes correctly
    2. Conduct optimization: pick up the one that maximizes the margin

SVM in Linear Non-Separable Cases

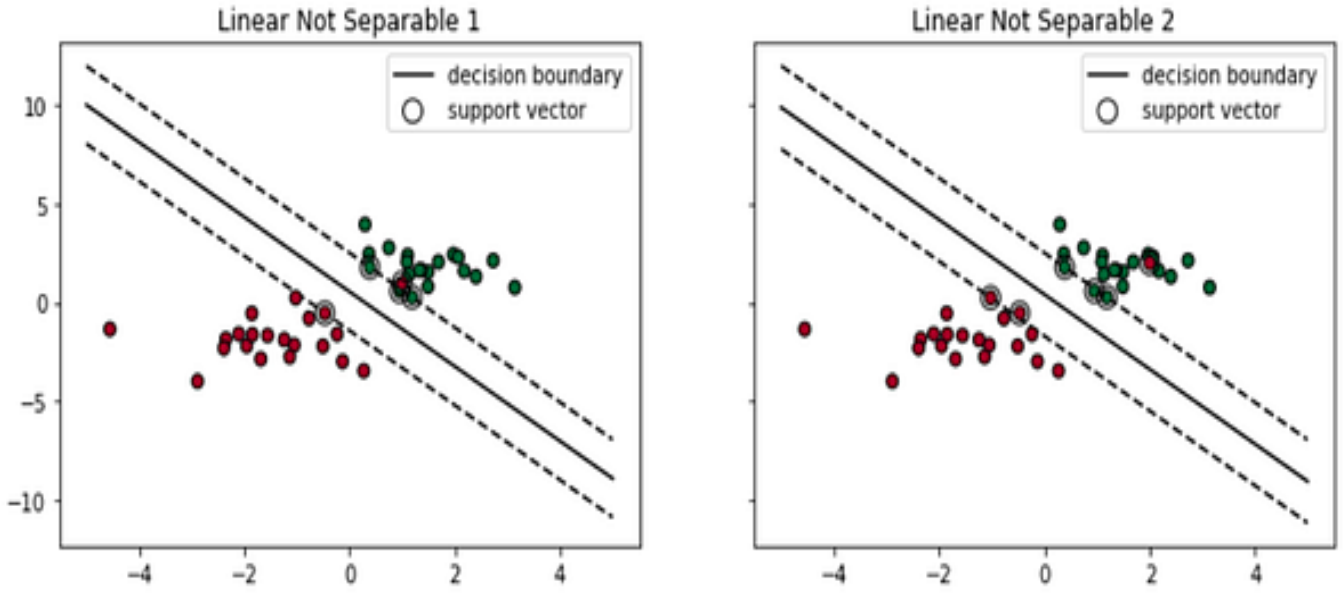
- In the linearly separable case, SVM is trying to find the hyperplane that maximizes the margin, with the condition that both classes are classified correctly.
- But in reality, datasets are probably never linearly separable, so the condition of 100% correctly classified by a hyperplane will never be met.

SVM in Linear Non-Separable Cases

- SVM address non-linearly separable cases by introducing two concepts:
  1. Soft Margin: try to find a line to separate, but tolerate one or few misclassified dots (e.g. the dots circled in red)
  2. Kernel Trick: try to find a non-linear decision boundary

Soft Margin

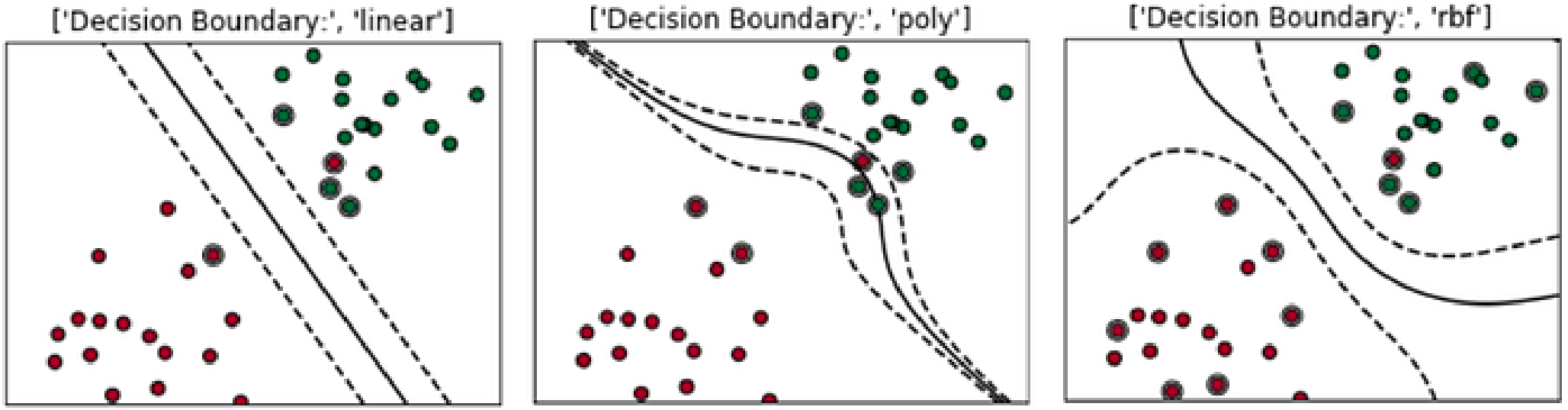
- Two types of misclassifications are tolerated by SVM under soft margin:
  1. The dot is on the wrong side of the decision boundary but on the correct side/ on the margin (shown in left)
  2. The dot is on the wrong side of the decision boundary and on the wrong side of the margin (shown in right)


Linear Not Separable 1
decision boundary
support vector
10
5
0
−5
−10
−4
−2
0
2
4
Linear Not Separable 2
decision boundary
support vector
−4
−2
0
2
4

Soft Margin

Soft Margin

- Applying Soft Margin, SVM tolerates a few dots to get misclassified and tries to balance the trade-off between finding a line that maximizes the margin and minimizes the misclassification.



Kernel Trick

- What Kernel Trick does is it utilizes existing features, applies some transformations, and creates new features.
- Those new features are the key for SVM to find the nonlinear decision boundary.