MODULE-2

PART-A

1. What are the parameters? How can the parameters of a circle hypothesis be calculated in such a case? What if it is an ellipse? Why does it make more sense to use an ellipse instead of a circle? How can you generalize your code to K > 2 classes?Let us say our hypothesis class is a circle instead of a rectangle.

Answer:

In the case of a circle, the parameters are the center and the radius (see figure 2.3). We then need to find the tightest circle that includes all the positive examples as S and G will be the largest circle that includes all the positive examples and no negative example:

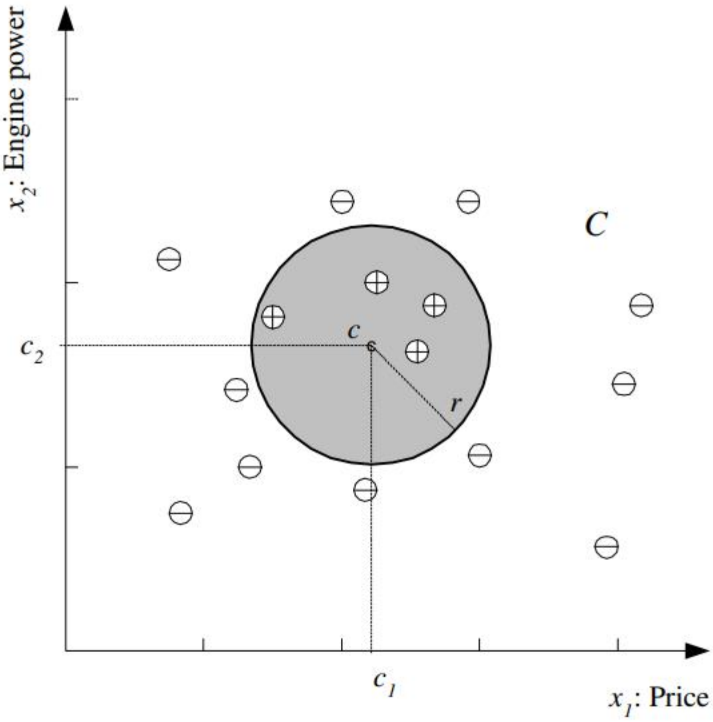


Figure 2.3 Hypothesis class is a circle with two parameters, the coordinates of its center and its radius.

It makes more sense to use an ellipse because the two axes need not have the same scale and an ellipse has two separate parameters for the widths in the two axes rather than a single radius. When there are K > 2 classes, we need a separate circle/ellipse for each class. For each class Ci , there will be one hypothesis which takesall elements of Ci as positive examples and instances of all Cj , j 6= i as negative examples.

2. What is the advantage of such a hypothesis class? Show that any class can be represented by such a hypothesis class with large enough m.Imagine our hypothesis is not one rectangle but a union of two or m > 1 rectangles

Answer:

In the case when there is a single rectangle, all the positive instances should form one single group; by increasing the number of rectangles, we get flexibility. With two rectangles for example (see figure 2.4), the positive instances can form two, possibly disjoint clusters in the input space. Note that each rectangle corresponds to a conjunction on the two input attributes and having multiple rectangles, corresponds to a disjunction. Any logical formula can be written as a disjunction of conjunctions. In the worst case (m = N), we can have a separate rectangle for each positive instance.

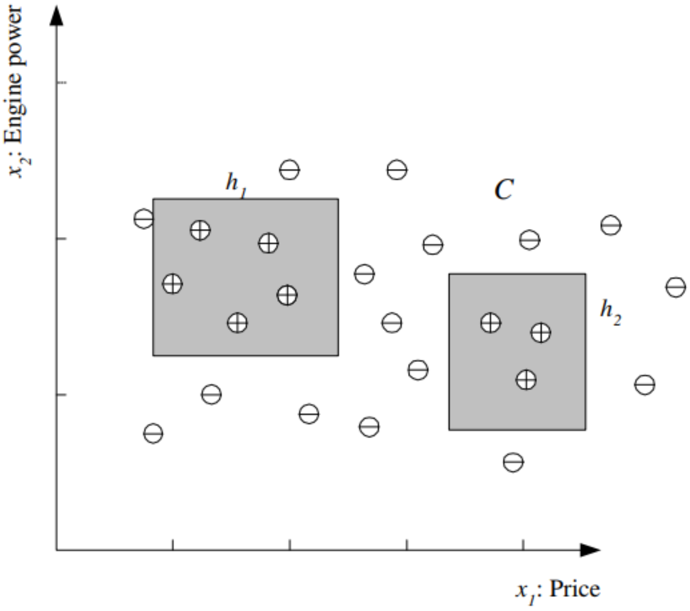


Figure 2.4 Hypothesis class is a union of two rectangles.

3. Choose a filtering algorithm that finds redundant instances?The complexity of most learning algorithms is a function of the training set.

Answer:

the time complexity of many learning algorithms is a function of the training set. There are numerous approaches to reduce this time complexity, and at various stages of training. In pre-training stage, dimensionality reduction can be done, while during training batch gradient updates can be done.

There are no standardized algorithms to filter out redundant instances mainly for two reasons: 1) due to reasons mentioned above, it is not recommended in most cases , 2) each dataset can have unique definition of redundancy , and more importantly, for the same dataset, two different learning tasks might have different definitions of redundancy and thus would require a need based filtering logic. One common approach to compare two instances is to find the similarity between them. You can try to find different similarity measures and pick one that suits your purpose. Visualization of data can often help in choosing exact thresholds for such techniques

4. Where should we choose x to learn with fewer queries?If we have a supervisor who can provide us with the label for any x,

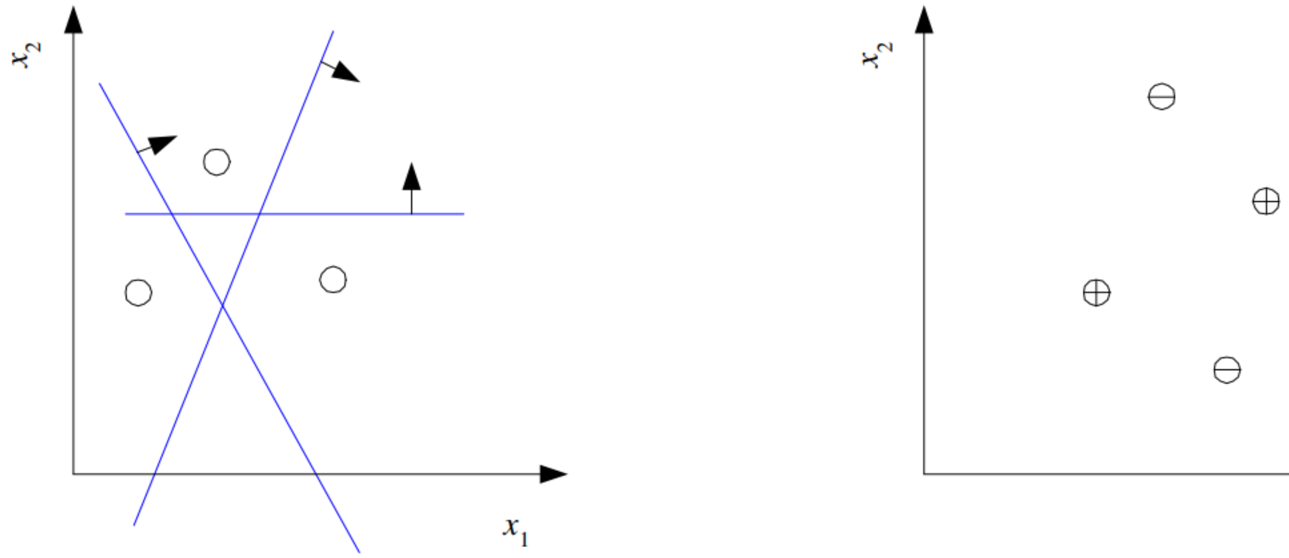
Answer:

The region of ambiguity is between S and G. It would be best to be given queries there so that we can make this region of doubt smaller. If a given instance there turns out to be positive, this means we can make S larger up to that instance; if it is negative, this means we can shrink G up until there.

5. Show that the VC dimension of a line is 3. Assume our hypothesis class is the set of lines, and we use a line to separate the positive and negative examples, instead of bounding the positive examples as in a rectangle, leaving the negatives outside

Answer:

As we see in figure 2.5 below, for all possible labeling of three points, there exist a line to separate positive and negative examples. With four points, no matter how we place these four points in two dimensions, there is at least one labeling where we cannot draw a line such that on one side lie all the positives and on the other lie all the negatives.



All possible labelings of three points can be separated using a line.

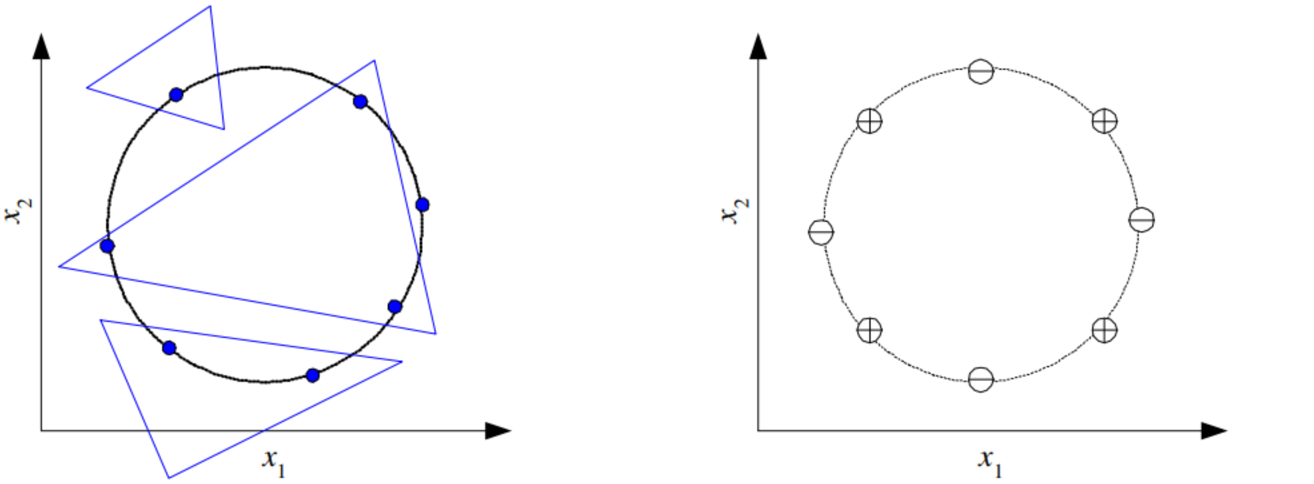
These four points cannot be separa

Figure 2.5 With a line, we can shatter three points but not

6. Show that the VC dimension of the triangle hypothesis class is in two dimensions. Hint: For best separation, it is best to place the seven points equidistant on a circle.

Answer:

As we can see in figure 2.6, for all possible labeling of seven points, we can draw a triangle to separate the positive and negative examples. We cannot do the same when there are eight points



These seven points can be separated using a triangle no matter how they are labeled.

These eight points with this labeling cannot b separated using a triangle.

Figure 2.6 A triangle can shatter seven points but not eight.

7. List out an error function that not only minimizes the number of mis-classifications but also maximizes the margin.Assume that our Hypothesis class is the set of lines.

Answer:

10. Which of these are limitations of the backpropagation algorithm?

Answder:

(a) local minima problem

(b) slow convergence

(c) scaling

PART-B

1. Explain in detail about Logistic Regression?

Answer:

○ Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

2. Explain in detail about BLUE assumptions

Answers:

The **Gauss Markov theorem** tells us that if a certain set of assumptions are met, the ordinary least squares estimate for regression coefficients gives you the *best linear unbiased estimate (BLUE)* possible.

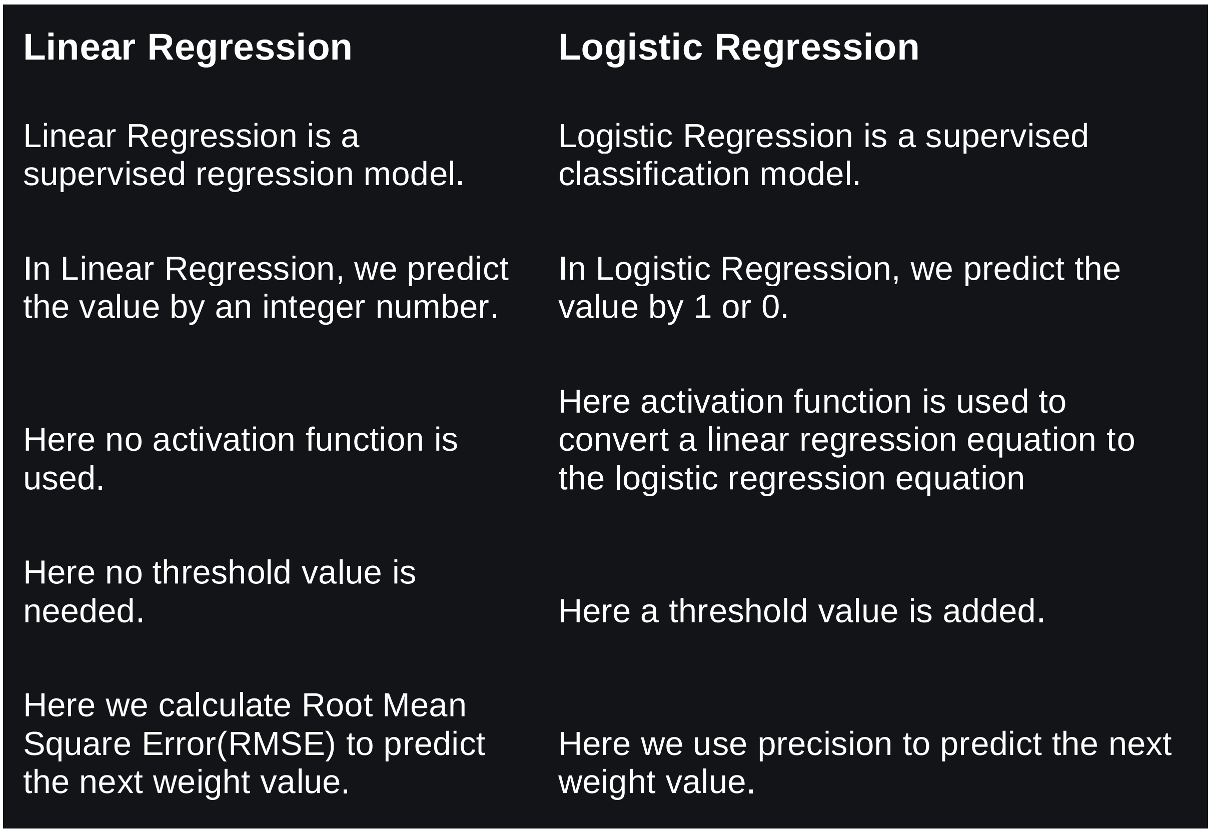
Gauss Markov Assumptions

There are five Gauss Markov assumptions (also called *conditions*):

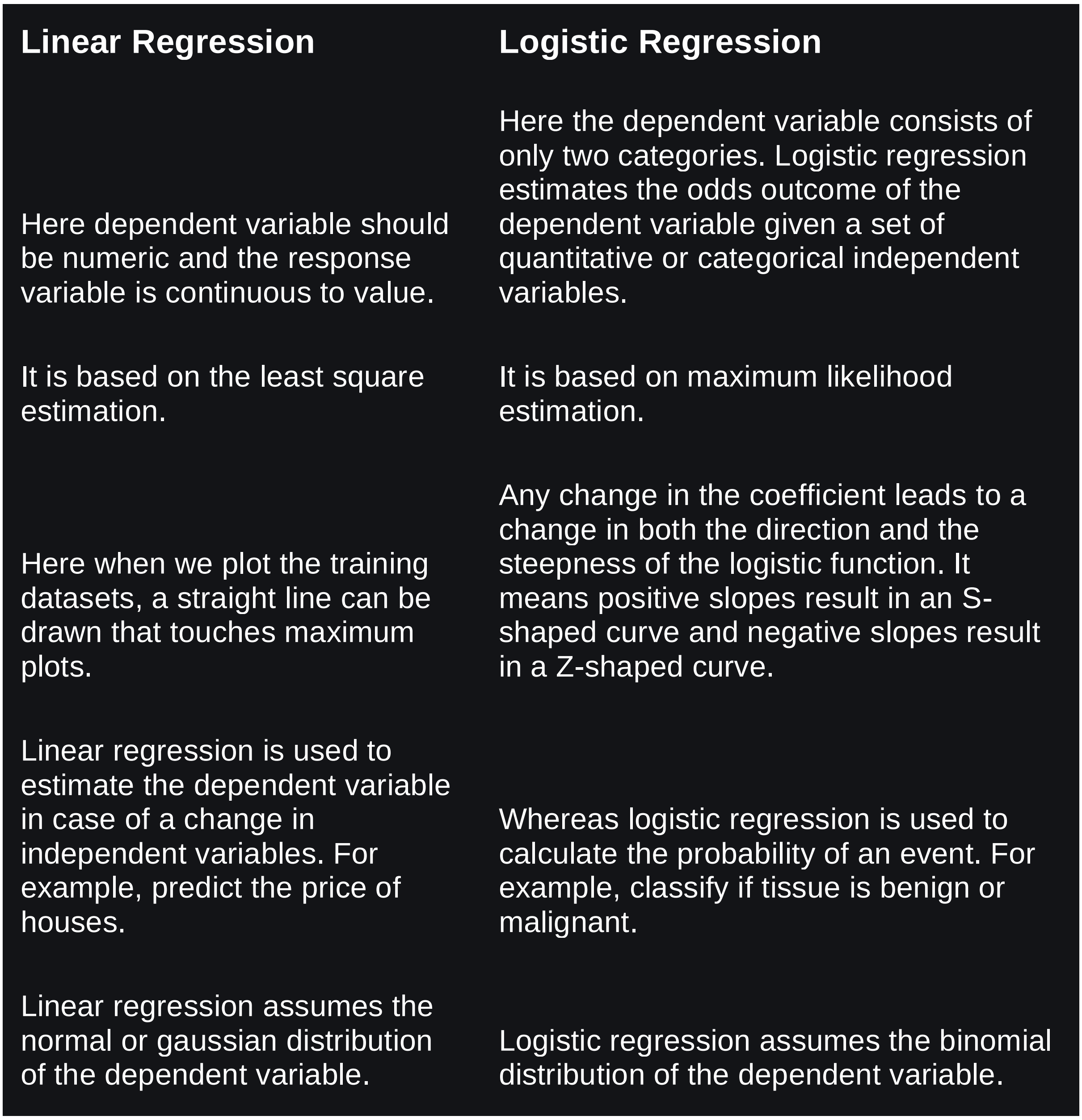
1. **Linearity**: the parameters we are estimating using the OLS method must be themselves linear.
2. **Random**: our data must have been randomly sampled from the population.
3. **Non-Collinearity**: the regressors being calculated aren't perfectly correlated with each other.
4. **Exogeneity**: the regressors aren't correlated with the error term.
5. **Homoscedasticity**: no matter what the values of our regressors might be, the error of the variance is constant.

3. **What is difference between Linear Regression and Logistic Regression in detail with examples ?**

Answer:



Linear Regression	Logistic Regression
Linear Regression is a supervised regression model.	Logistic Regression is a supervised classification model.
In Linear Regression, we predict the value by an integer number.	In Logistic Regression, we predict the value by 1 or 0.
Here no activation function is used.	Here activation function is used to convert a linear regression equation to the logistic regression equation
Here no threshold value is needed.	Here a threshold value is added.
Here we calculate Root Mean Square Error(RMSE) to predict the next weight value.	Here we use precision to predict the next weight value.



Linear Regression	Logistic Regression
Here dependent variable should be numeric and the response variable is continuous to value.	Here the dependent variable consists of only two categories. Logistic regression estimates the odds outcome of the dependent variable given a set of quantitative or categorical independent variables.
It is based on the least square estimation.	It is based on maximum likelihood estimation.
Here when we plot the training datasets, a straight line can be drawn that touches maximum plots.	Any change in the coefficient leads to a change in both the direction and the steepness of the logistic function. It means positive slopes result in an S-shaped curve and negative slopes result in a Z-shaped curve.
Linear regression is used to estimate the dependent variable in case of a change in independent variables. For example, predict the price of houses.	Whereas logistic regression is used to calculate the probability of an event. For example, classify if tissue is benign or malignant.
Linear regression assumes the normal or gaussian distribution of the dependent variable.	Logistic regression assumes the binomial distribution of the dependent variable.

4. What are the types of Linear Regression and explain them with Examples?

Answers:

Linear Regression is generally classified into two types:

1. Simple Linear Regression
2. Multiple Linear Regression

1. Simple

In Simple Linear Regression, we try to find the relationship between **a single independent variable** (input) and **a corresponding dependent variable (output)**. This can be expressed in the form of a straight line.

The same equation of a line can be re-written as:

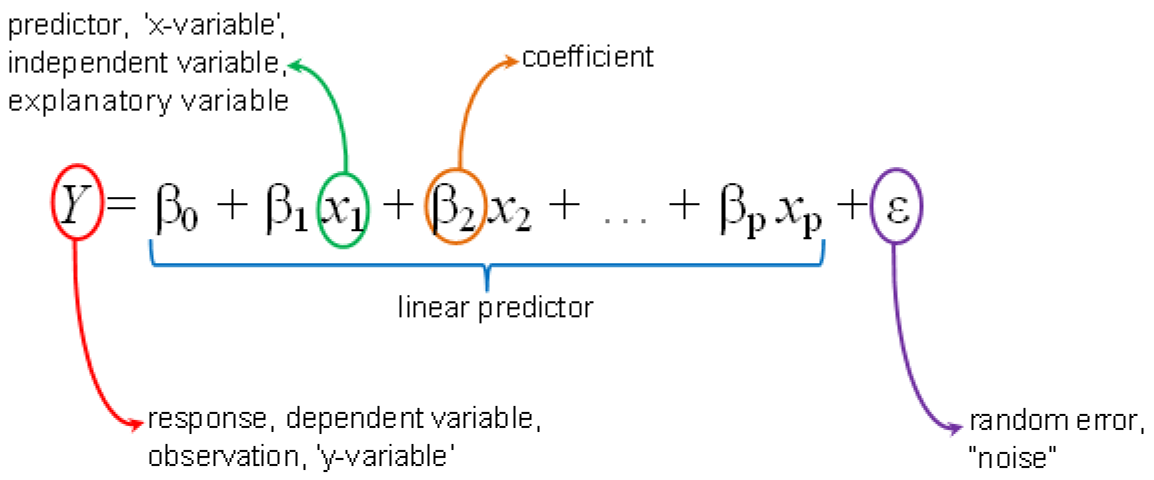
$$Y = \beta_0 + \beta_1 X + \epsilon$$

1. **Y** represents the output or dependent variable.
2. **β0 and β1** are two unknown constants that represent the intercept and coefficient (slope) respectively.
3. **ε** (Epsilon) is the error term.

2. Multiple Linear Regression

In Multiple Linear Regression, we try to find the relationship between **2 or more independent variables (inputs)** and the corresponding dependent variable (output). The independent variables can be continuous or categorical.

The equation that describes how the predicted values of y is related to **p independent variables** is called as **Multiple Linear Regression equation :**


predictor, 'x-variable', independent variable, explanatory variable
coefficient
$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon$
linear predictor
response, dependent variable, observation, 'y-variable'
random error, "noise"

5. Explain Basic Decision Tree Algorithm?

Answer: Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.

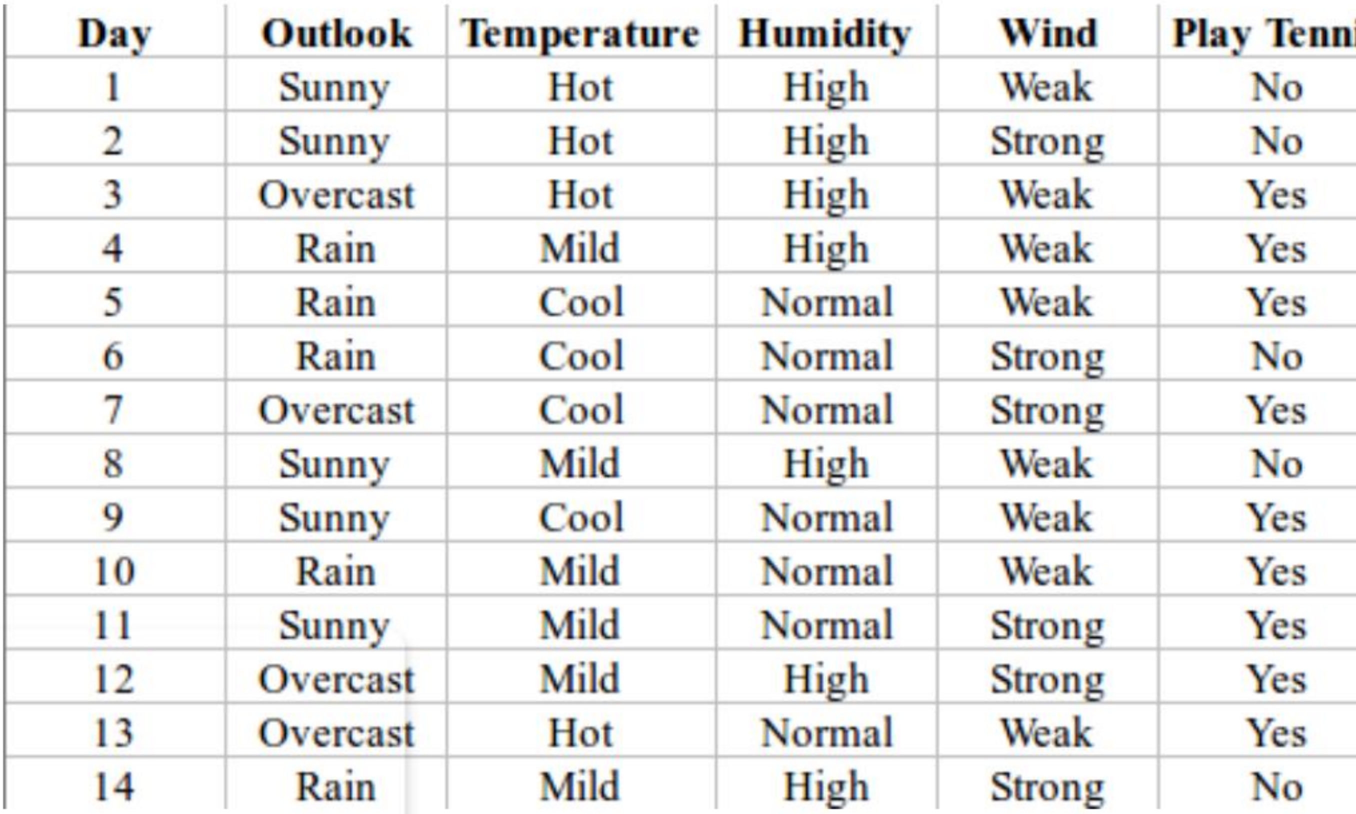
The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Steps in ID3 algorithm:

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates **Entropy(H)** and **Information gain(IG)** of this attribute.

3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.



Day	Outlook	Temperature	Humidity	Wind	Play Tenni
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Complete entropy of dataset is:

```
H(S) = - p(yes) * log2(p(yes)) - p(no) * log2(p(no))
     = - (9/14) * log2(9/14) - (5/14) * log2(5/14)
     = - (-0.41) - (-0.53)
     = 0.94
```

First Attribute - Outlook

```
Categorical values - sunny, overcast and rain
H(Outlook=sunny) = -(2/5)*log(2/5)-(3/5)*log(3/5) =0.971
H(Outlook=rain) = -(3/5)*log(3/5)-(2/5)*log(2/5) =0.971
H(Outlook=overcast) = -(4/4)*log(4/4)-0 = 0

Average Entropy Information for Outlook -
I(Outlook) = p(sunny) * H(Outlook=sunny) + p(rain) * H(Outlook=rain) +
p(overcast) * H(Outlook=overcast)
= (5/14)*0.971 + (5/14)*0.971 + (4/14)*0
= 0.693

Information Gain = H(S) - I(Outlook)
                 = 0.94 - 0.693
                 = 0.247
```

Second Attribute - Temperature

```
Categorical values - hot, mild, cool
H(Temperature=hot) = -(2/4)*log(2/4)-(2/4)*log(2/4) = 1
H(Temperature=cool) = -(3/4)*log(3/4)-(1/4)*log(1/4) = 0.811
H(Temperature=mild) = -(4/6)*log(4/6)-(2/6)*log(2/6) = 0.9179
Average Entropy Information for Temperature -
I(Temperature) = p(hot)*H(Temperature=hot) + p(mild)*H(Temperature=mild) +
p(cool)*H(Temperature=cool)
= (4/14)*1 + (6/14)*0.9179 + (4/14)*0.811
= 0.9108

Information Gain = H(S) - I(Temperature)
                 = 0.94 - 0.9108
                 = 0.0292
```

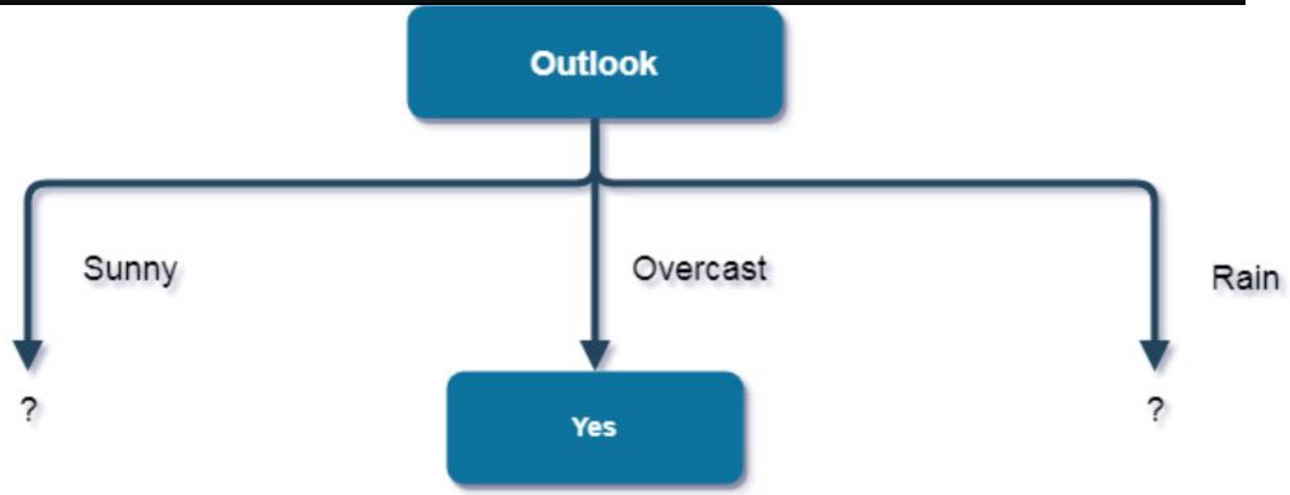
Third Attribute - Humidity

```
Categorical values - high, normal
H(Humidity=high) = -(3/7)*log(3/7)-(4/7)*log(4/7) = 0.983
H(Humidity=normal) = -(6/7)*log(6/7)-(1/7)*log(1/7) = 0.591
```

```
Average Entropy Information for Humidity -
I(Humidity) = p(high)*H(Humidity=high) + p(normal)*H(Humidity=normal)
= (7/14)*0.983 + (7/14)*0.591
= 0.787

Information Gain = H(S) - I(Humidity)
                 = 0.94 - 0.787
                 = 0.153
```

Fourth Attribute - Wind

```
Categorical values - weak, strong
H(Wind=weak) = -(6/8)*log(6/8)-(2/8)*log(2/8) = 0.811
H(Wind=strong) = -(3/6)*log(3/6)-(3/6)*log(3/6) = 1

Average Entropy Information for Wind -
I(Wind) = p(weak)*H(Wind=weak) + p(strong)*H(Wind=strong)
= (8/14)*0.811 + (6/14)*1
= 0.892

Information Gain = H(S) - I(Wind)
                 = 0.94 - 0.892
                 = 0.048
```



First Attribute - Temperature

```
Categorical values - hot, mild, cool
```

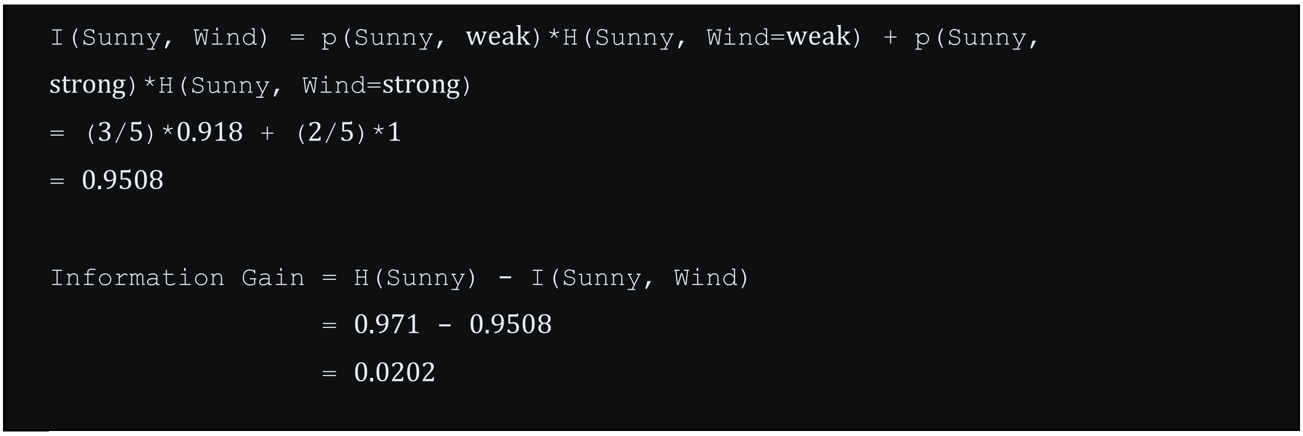
```
H(Sunny, Temperature=hot) = -0-(2/2)*log(2/2) = 0
H(Sunny, Temperature=cool) = -(1)*log(1)- 0 = 0
H(Sunny, Temperature=mild) = -(1/2)*log(1/2)-(1/2)*log(1/2) = 1
Average Entropy Information for Temperature -
I(Sunny, Temperature) = p(Sunny, hot)*H(Sunny, Temperature=hot) + p(Sunny,
mild)*H(Sunny, Temperature=mild) + p(Sunny, cool)*H(Sunny,
Temperature=cool)
= (2/5)*0 + (1/5)*0 + (2/5)*1
= 0.4

Information Gain = H(Sunny) - I(Sunny, Temperature)
                 = 0.971 - 0.4
                 = 0.571
```

Second Attribute - Humidity

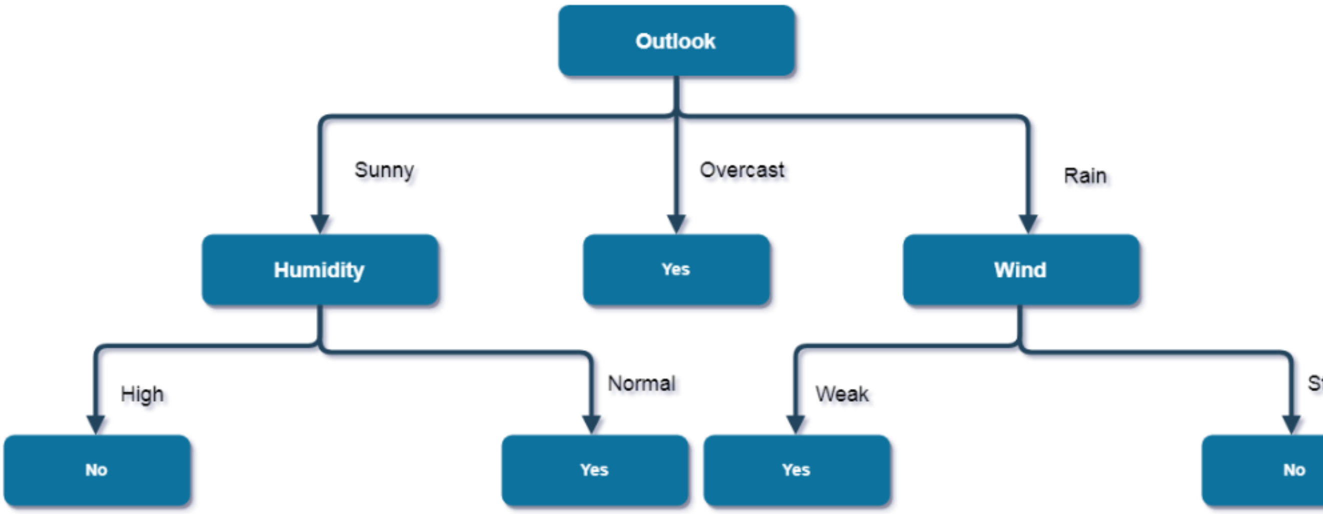
```
Categorical values - high, normal
H(Sunny, Humidity=high) = - 0 - (3/3)*log(3/3) = 0
H(Sunny, Humidity=normal) = -(2/2)*log(2/2)-0 = 0

Average Entropy Information for Humidity -
I(Sunny, Humidity) = p(Sunny, high)*H(Sunny, Humidity=high) + p(Sunny,
normal)*H(Sunny, Humidity=normal)
= (3/5)*0 + (2/5)*0
= 0

Information Gain = H(Sunny) - I(Sunny, Humidity)
                 = 0.971 - 0
                 = 0.971
```

Third Attribute - Wind

```
Categorical values - weak, strong
H(Sunny, Wind=weak) = -(1/3)*log(1/3)-(2/3)*log(2/3) = 0.918
H(Sunny, Wind=strong) = -(1/2)*log(1/2)-(1/2)*log(1/2) = 1

Average Entropy Information for Wind -
```



```
I(Sunny, Wind) = p(Sunny, weak)*H(Sunny, Wind=weak) + p(Sunny,
strong)*H(Sunny, Wind=strong)
= (3/5)*0.918 + (2/5)*1
= 0.9508

Information Gain = H(Sunny) - I(Sunny, Wind)
                 = 0.971 - 0.9508
                 = 0.0202
```



6. Explain how Hypothesis Search is carried out in Decision Tree Learning?

Answer:

- ID3 searches the space of possible decision trees: doing hill-climbing on information gain.
- It searches the *complete* space of all finite discrete-valued functions. All functions have at least one tree that represents them.
- It maintains only one hypothesis (unlike Candidate-Elimination). It cannot tell us how many other viable ones there are.
- It does not do back tracking. Can get stuck in local optima.
- Uses all training examples at each step. Results are less sensitive to errors.

7.Explain ID3 Algorithm with an example

Answer: refer Q5

8. Explain in detail about Information gain and Gini Index with Example

Answers:

The Information Gain

To build a decision tree, we need to decide which feature to check at which node. For instance, let's suppose that we have two unused features: and , both binary. We also have five objects, two of which are positive:

$$S: \quad \begin{bmatrix} a & b & class \\ \hline 0 & 0 & positive \\ 0 & 1 & positive \\ 1 & 0 & negative \\ 1 & 1 & positive \\ 0 & 0 & negative \end{bmatrix}$$

Which feature should we test to add a new node? The information gain can help us decide. It's the expected amount of information we get by inspecting the feature. Intuitively, the feature with the largest expected amount is the best choice. That's because it will reduce our uncertainty the most on average.

Gini index:

The Gini Index or Gini Impurity is calculated by subtracting the sum of the squared probabilities of each class from one. It favours mostly the larger partitions and are very simple to implement. In simple terms, it calculates the probability of a certain randomly selected feature that was classified incorrectly.
The Gini Index varies between 0 and 1, where 0 represents purity of the classification and 1 denotes random distribution of elements among various classes. A Gini Index of 0.5 shows that there is equal distribution of elements across some classes.

9. Define Residual and Explain how it can be handled in Linear Regression
Answers:

A residual is a measure of how far away a point is vertically from the regression line. Simply, it is the error between a predicted value and the observed actual value.

$$Residual\ (\epsilon) = \ y - \ \hat{y}$$

Residual Equation

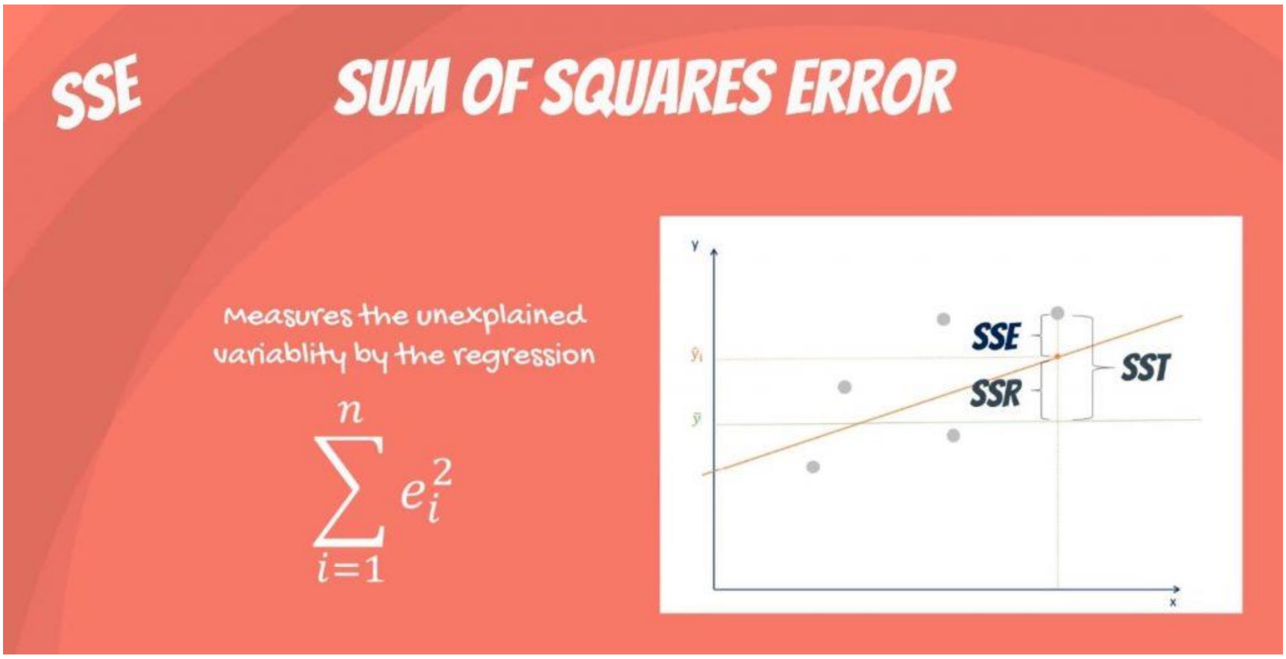
10. Why do we square the residuals instead of using modulus?
Answer:

Moreover in mathematical terms, the squared function is differentiable everywhere, while the absolute error is not differentiable at all the points in its domain(its derivative is undefined at 0). This makes the squared error more preferable to the techniques of mathematical optimization. To optimize the squared error, we can compute the derivative and set its expression equal to 0, and solve. But to optimize the absolute error, we require more complex techniques having more computations. Actually, we use the Root Mean Squared Error instead of Mean squared error so that the unit of RMSE and the dependent variable are equal and results are interpretable.

11. What is the Importance of SSE in Linear Regression

Answer:

The last term is the **sum of squares error**, or **SSE**. The error is the difference between the *observed* value and the *predicted* value.



We usually want to minimize the error. The smaller the error, the better the estimation power of the **regression**. Finally, I should add that it is also known as **RSS** or **residual sum of squares**. Residual as in: remaining or unexplained.

12. Explain the normal form equation of the Linear Regression.

Answer:

Normal Equation is an analytical approach to Linear Regression with a Least Square Cost Function. We can directly find out the value of θ without using Gradient Descent.

Following this approach is an effective and a time-saving option when are working with a dataset with small features.

Normal Equation is a follows :

$$\theta = \left(X^T X\right)^{-1} \cdot \left(X^T y\right)$$

In the above equation,
θ : hypothesis parameters that define it the best.
X : Input feature value of each instance.
Y : Output value of each instance.

Maths Behind the equation –
Given the hypothesis function

$$h(\theta) = \theta_0 x_0 + \theta_1 x_1 + \dots \theta_n x_n$$

where,
n : the no. of features in the data set.
x0 : 1 (for vector multiplication)

$$h(\theta) = \theta_0 x_0 + \theta_1 x_1 + \dots \theta_n x_n$$

Notice that this is dot product between θ and x values. So for the convenience to solve we can write it as :

$$\mathrm{J}(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} \frac{1}{2}[h_\Theta(x^{(i)}) - y^{(i)}]^2$$

The motive in Linear Regression is to minimize the cost function :

J(Theta) = frac{1}{2m} sum_{i = 1}^{m} frac{1}{2} [h_{Theta}(x^{(i)}) - y^{(i)}]^{2}
where,
xi : the input value of iih training example.
m : no. of training instances
n : no. of data-set features
yi : the expected result of ith instance

Let us representing cost function in a vector form.

$$\begin{bmatrix} h_\theta\left(x^0\right) \\ h_\theta\left(x^1\right) \\ \ldots \\ h_\theta\left(x^m\right) \end{bmatrix} - \begin{bmatrix} y^0 \\ y^1 \\ \ldots \\ y^m \end{bmatrix}$$

we have ignored 1/2m here as it will not make any difference in the working. It was used for the mathematical convenience while calculation gradient descent. But it is no more needed here.

$$\begin{bmatrix} \theta^T\left(x^0\right) \\ \theta^T\left(x^1\right) \\ \ldots \\ \theta^T\left(x^m\right) \end{bmatrix} - y$$

$$\begin{bmatrix} \theta_0\begin{pmatrix} 0 \\ x_0 \end{pmatrix} + \theta_1\begin{pmatrix} 0 \\ x_1 \end{pmatrix} + \ldots.\theta_n\begin{pmatrix} 0 \\ x_n \end{pmatrix} \\ \theta_0\begin{pmatrix} 1 \\ x_0 \end{pmatrix} + \theta_1\begin{pmatrix} 1 \\ x_1 \end{pmatrix} + \ldots\theta_n\begin{pmatrix} 1 \\ x_n \end{pmatrix} \\ \ldots \\ \theta_0\begin{pmatrix} m \\ x_0 \end{pmatrix} + \theta_1\begin{pmatrix} m \\ x_1 \end{pmatrix} + \ldots.\theta_n\begin{pmatrix} m \\ x_n \end{pmatrix} \end{bmatrix} - y$$

xij : value of jih feature in iih training example.

This can further be reduced to Xtheta - y
But each residual value is squared. We cannot simply square the above expression. As the square of a vector/matrix is not equal to the square of each of its values. So to get the squared value, multiply the vector/matrix with its transpose. So, the final equation derived is

$$(X\theta - y)^T(X\theta - y)$$

Therefore, the cost function is

So, now getting the value of θ using derivative

$$\frac{\partial J_\theta}{\partial \theta} = \frac{\partial}{\partial \theta}\left[(X\theta - y)^T (X\theta - y)\right]$$

$$\frac{\partial J_\theta}{\partial \theta} = 2X^T X\theta - 2X^T y$$

$$\text{Cost}'(\theta) = 0$$

So, this is the finally derived Normal Equation with θ giving the minimum cost value.

$$2X^T X\theta = 2X^T y$$

$$\left(X^T X\right)^{-1}\left(X^T X\right)\theta = \left(X^T X\right)^{-1} \cdot \left(X^T y\right)$$

$$\theta = \left(X^T X\right)^{-1} \cdot \left(X^T y\right)$$

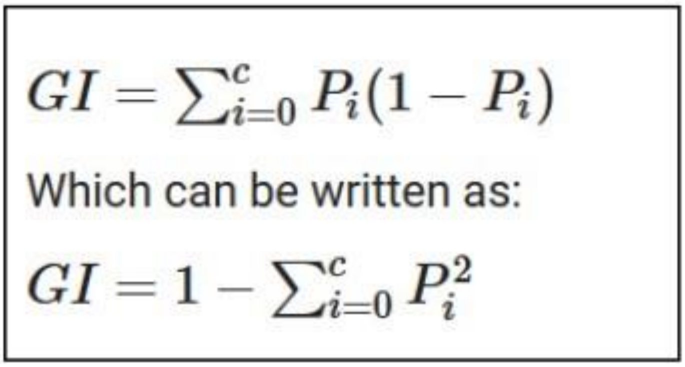
13. Explain in detail about CART Algorithm

Answer:

CART Algorithm:

This algorithm can be used for both classification & regression. CART algorithm uses Gini Index criterion to split a node to a sub-node. It start with the training set as a root node, after successfully splitting the root node in two, it splits the subsets using the same logic & again split the sub-subsets, recursively until it finds further splitting will not give any pure sub-nodes or maximum number of leaves in a growing tree or termed it as a Tree pruning.

How to calculate Gini Index?



$$GI = \sum_{i=0}^{c} P_i(1 - P_i)$$

Which can be written as:

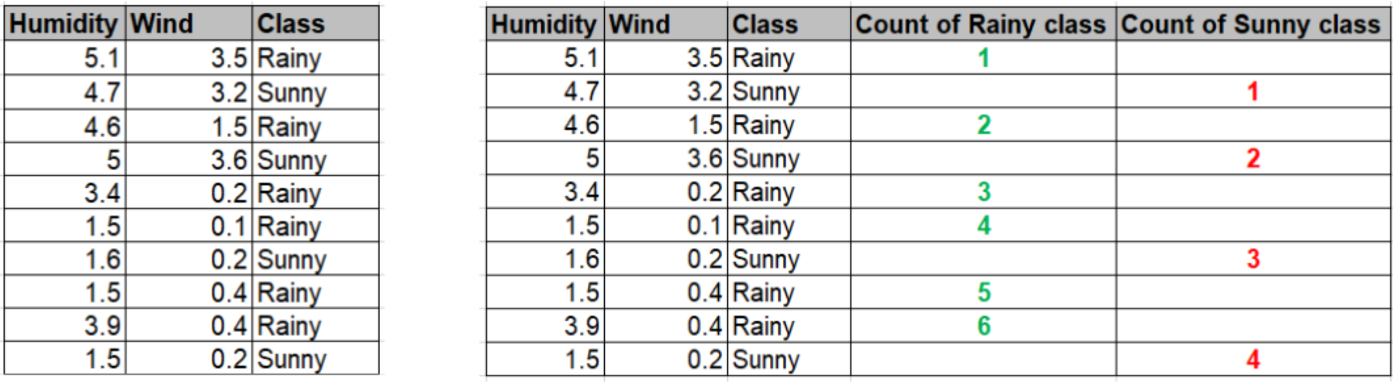
$$GI = 1 - \sum_{i=0}^{c} P_i^2$$

Image 2: Formula of Gini Index

In Gini Index, P is the probability of class **i** & there is total **c** classes.

Considering you have only two predictor/attributes: Humidity & Wind

Class: Rainy & Sunny



Humidity	Wind	Class
5.1	3.5	Rainy
4.7	3.2	Sunny
4.6	1.5	Rainy
5	3.6	Sunny
3.4	0.2	Rainy
1.5	0.1	Rainy
1.6	0.2	Sunny
1.5	0.4	Rainy
3.9	0.4	Rainy
1.5	0.2	Sunny

Humidity	Wind	Class	Count of Rainy class	Count of Sunny class
5.1	3.5	Rainy	1	
4.7	3.2	Sunny		1
4.6	1.5	Rainy	2	
5	3.6	Sunny		2
3.4	0.2	Rainy	3	
1.5	0.1	Rainy	4	
1.6	0.2	Sunny		3
1.5	0.4	Rainy	5	
3.9	0.4	Rainy	6	
1.5	0.2	Sunny		4

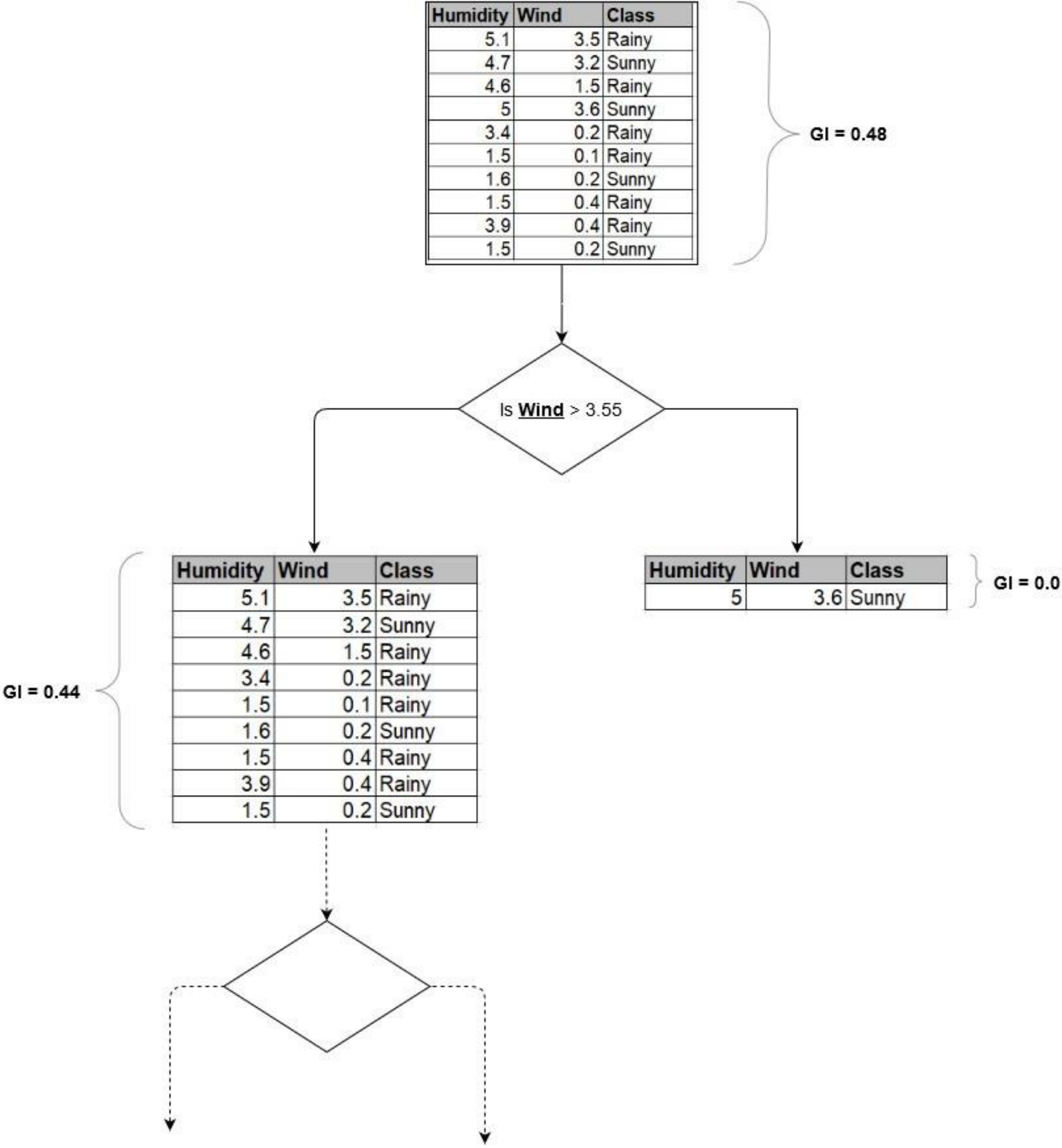
Image 3: Data & it’s distribution by class

GI = 1 — ((num of observations from Feature_1/total observation)^2 + (num of observations from Feature_2/total observation)^2)

GI = 1-((6/10)^2 + (4/10)^2) => 1-(0.36+0.16) => 1–0.52 => 0.48

So, the Gini index for the first/initial set is 0.48

Basic idea on how the Node split happens:



Humidity	Wind	Class
5.1	3.5	Rainy
4.7	3.2	Sunny
4.6	1.5	Rainy
5	3.6	Sunny
3.4	0.2	Rainy
1.5	0.1	Rainy
1.6	0.2	Sunny
1.5	0.4	Rainy
3.9	0.4	Rainy
1.5	0.2	Sunny

Humidity	Wind	Class
5.1	3.5	Rainy
4.7	3.2	Sunny
4.6	1.5	Rainy
3.4	0.2	Rainy
1.5	0.1	Rainy
1.6	0.2	Sunny
1.5	0.4	Rainy
3.9	0.4	Rainy
1.5	0.2	Sunny

Humidity	Wind	Class
5	3.6	Sunny

Image 4: Node splitting on Gini Index

Based on attribute “wind” (f) & threshold value “3.55” (t) the CART algorithm created nodes/subsets which would give a pure subsets to right side of the above flow (ref: image 4).

14. How do you learn a class from examples to perform Supervised Learning

Answer:

https://youtu.be/PopqmTZeVu8

15. Explain the difference between Multi-class and Multi-Label Classification

Answer:

Multiclass Classification:

Multiclass classification is the problem of classification in machine learning where our task is to classify between more than two classes. As in binary classification, we only classify between 2 classes in Multiclass, we classify between more than two classes.

Also, Read – 200+ Machine Learning Projects Solved and Explained.

For example, if we classify e-mails as spam and not as spam, it is not the problem of multiclass classification because it is the problem of binary classification. But if we classify between cars like sedan, SUV and hatchback, it is the problem of multiclass classification.

Multilabel Classification:

Multilabel classification is a classification problem in machine learning where the task is to classify the labels of each instance where the labels can be from 0 to n number of classes. For example, think of a facial recognition system what to do if it recognizes multiple people in an image.

It will attach each person with a tag and then it will recognize the faces of all the people in the photo. Here, people are classes, and the recognition system will attach every tag to every class in this kind of problem.

16How do Classification and Regression differ?

Answer:

Difference between Regression and Classification

Regression Algorithm	Classification Algorithm
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete value.
The task of the regression algorithm is to map the input value (x) with the continuous output variable(y).	The task of the classification algorithm is to map the input value(x) with the discrete output variable(y).
Regression Algorithms are used with continuous data.	Classification Algorithms are used with discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, which can divide the dataset into different classes.
Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classification problems such as Identification of spam emails, Speech Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into Binary Classifier and Multi-class Classifier.

Comparison between Classification and Regression:

Parameter	CLASSIFICATION	REGRESSION
Basic	The mapping function is used for mapping values to predefined classes.	Mapping Function is used for the mapping of values to continuous output.

Parameter	CLASSIFICATION	REGRESSION
Involves prediction of	Discrete values	Continuous values
Nature of the predicted data	Unordered	Ordered
Method of calculation	by measuring accuracy	by measurement of root mean square error
Example Algorithms	Decision tree, logistic regression, etc.	Regression tree (Random forest), Linear regression, etc

17. What are the five popular algorithms we use in Machine Learning?

Answers:

- Linear Regression.
- Logistic Regression.
- Decision Tree.
- Naive Bayes.
- kNN.

18. Explain the importance of Pruning?

Answer:

Pruning or post-pruning

As the name implies, pruning involves cutting back the tree. After a tree has been built (and in the absence of early stopping discussed below) it may be overfitted. The CART algorithm will repeatedly partition data into smaller and smaller subsets until those final subsets are homogeneous in terms of the outcome variable. In practice this often means that the final subsets (known as the *leaves* of the tree) each consist of only one or a few data points. The tree has learned the data exactly, but a new data point that differs very slightly might not be predicted well.

Early stopping or pre-pruning

An alternative method to prevent overfitting is to try and stop the tree-building process early, before it produces leaves with very small samples. This heuristic is known as *early stopping* but is also sometimes known as pre-pruning decision trees.

At each stage of splitting the tree, we check the cross-validation error. If the error does not decrease significantly enough then we stop. Early stopping may underfit by stopping too early. The current split may be of little benefit, but having made it, subsequent splits more significantly reduce the error.

19. What is a model selection in Machine Learning?

Answer:

Firstly, the order of our polynomial is set before the training process begins, and we call these special parameters in our model "**hyperparameters**". Secondly, **the process of figuring out the values of these hyperparameters is called hyperparameter optimization and is a part of model selection**. Thirdly, as mentioned in the previous post, machine learning is mostly concerned with prediction, which means that **we define the 'best' model as the one that generalizes the best on future data**, i.e. which model would perform the best on data it wasn't trained on?

To figure this out, we usually want to come up with some kind of **evaluation metric**. Then we divide our training dataset into 3 parts: a **training**, a **validation** (sometimes called **development**), and a **test** dataset. Then we train our model on the training dataset, perform model selection on the validation dataset, and do a final evaluation of the model on the test dataset. This way, we can determine the model with the lowest generalization error. **The generalization error refers to the performance of the model on unseen data**, i.e. data that the model hasn't been trained on