Page No.
Date : / /

Linear Regression:

Problem statement:

We have a collection of labelled examples $\{(x_i, y_i)\}_{i=1}^{N}$

where N = size of collection.

x_i = D-dimensional feature vector

y_i = real-valued target ($y \in \mathbb{R}$).

We want to build a model $f_{\vec{w},b}(\vec{x})$ as linear comb^n of features of example $\vec{x}$:

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

The model is parameterized by two values $\vec{w}$, and b.

to predict

We will use the model, the unknown 'y' for given $\vec{x}$ like:

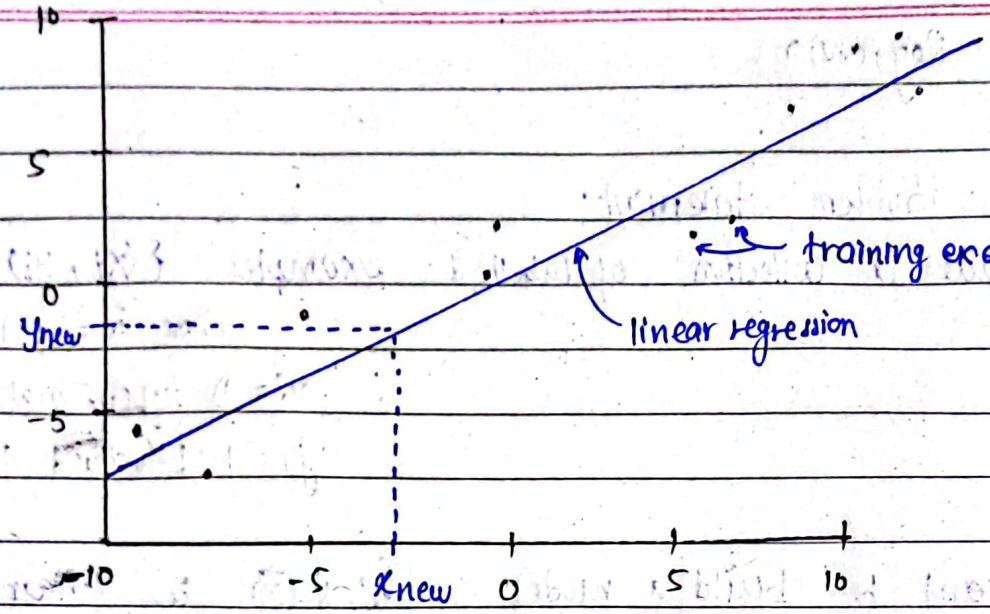
$$y \leftarrow f_{\vec{w},b}(\vec{x})$$

Two models parameterized by two different pairs $(\vec{w}, b)$ will likely produce two different predictions when applied to same example.

So we want to find the optimal values for $\vec{w}$ and b: $(\vec{w}^*, b^*)$

and obviously the optimal values of parameters define the model that makes most accurate predictions.

The hyperplane in Linear regression is chosen to be as close to all training examples as possible.



We can use this line to predict the value of target "ynew" for a new unlabelled input example "xnew".

If the line was far from the training datas, the 'ynew' would have fewer chances to be correct.

Q2:

To get the latter requirement satisfied, the optimization procedure which we use to find the optimal values for $\vec{w}^*$ and b^* tries to minimize the following expression:

$$\frac{1}{N}\sum_{i=1}^{N}\left(f_{\vec{w},b}(\vec{x}_i) - y_i\right)^2$$

In mathematics the expression we minimize/maximize is called objective (function).

The expression $\left(f_{\vec{w},b}(\vec{x}_i) - y_i\right)^2$ in the above objective is called loss fun^n.

Page No.
Date : / /

this particular choice of loss fun^n is called squared error loss.

All model-based learning algos have a loss fun^n and what we do to find the best model is: we try to minimize the objective known as the cost fun^n.

In linear regression, the cost fun^n is given by the average loss, also called the 'emperical risk'.

The average loss/emperical risk for a model, is the average of all penalties obtained by applying the model to training data.

Note: When designing a ML Algo, many decisions are made that can significantly impact its performance and behavior. For linear regression, the choice of using a quadratic loss fun^n (squared error loss) instead of absolute loss or cubic loss is not arbitary.

The quadratic loss is used because it has a continuous derivative, making it smooth and easier to work mathematically. (particularly when using linear algebra to find optimal solutions. In contrast, the absolute error lacks a continuous derivative, leading to challenges in optimization.

Additionally, the decision to use linear models for simplicity is more beneficial as linear models are less prone to overfitting compared to other complex models.

Page No.
Date: / /

Decision Tree Learning:

Problem Statement:

Like previously, we have a collection of labelled examples ; labels belong to set $\{0,1\}$. We want to build a decision tree that would allow us to predict the class given a feature vector.

Solution:

There are various formulations of decision tree learning algo. Among which ID3 is one.

The optimization criterion, in this case, is the average log-likelihood:

$$\frac{1}{N}\sum_{i=1}^{N}\left[y_i \ln f_{ID3}(x_i) + (1-y_i)\ln\left(1-f_{ID3}(x_i)\right)\right],$$

where,

f_{ID3} → Decision Tree

Note: Looks very similar to logistic regression. However, contrary to the logistic regression learning algo → which builds a parametric model $f_{w,b}$ by finding optimal solution to the optimization criterion,

the ID3 algo optimizes it approximately by constructing a non-parametric model.

$$f_{ID3}(x) \overset{def}{=} Pr(y=1|x)$$

For Logistic regression: https://tinyurl.com/5n7uadj8

Page No.
Date : / /

Learning ID3 (Iterative Dechotomiser 3) Algo using example qn:

Example Dataset: (The example dataset of COVID-Test patients)

ID	Fever	Cough	Breathing Issues	Infected?
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES
6	NO	YES	NO	NO
7	NO	YES	YES	YES
8	YES	YES	NO	NO
9	NO	YES	NO	NO
10	NO	NO	NO	NO
11	NO	YES	YES	YES
12	YES	YES	YES	YES

We will calculate 2 things "Entropy" and "Information Gain"

Entropy of dataset = measure of disorder in target feature of the dataset.

We denote our dataset as 'S' and Entropy with 'H'.

So,

$$H(S) = -\sum_{i=1}^{n} p_i \times \log_2(p_i)$$

where: n = total no of ~~the column~~ classes in target column.

p_i = probability of class 'i'.

So back to the calculation:

Information gain of the dataset:

(1) Entropy of whole system:

$$H(+6,-6) = -\frac{6}{12}\log_2\frac{6}{12} - \frac{6}{12}\log_2\frac{6}{12} = \frac{1}{2}\cdot 1$$

(2) Entropy of all attributes:

Entropy Fever: $H(+4,-2) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.9182$

Entropy Cough: $H(+5,-4) = -\frac{5}{9}\log_2\frac{5}{9} - \frac{4}{9}\log_2\frac{4}{9} = 0.9910$

Entropy Breathing Issues: $H(+6,-0) = -\frac{6}{6}\log\frac{6}{6} - 0$

Information Gains (IG):

Page No.
Date: / /

$$IG\{Fever\} = Entropy(whole\ data) - \frac{6}{12} \times 0.9182$$

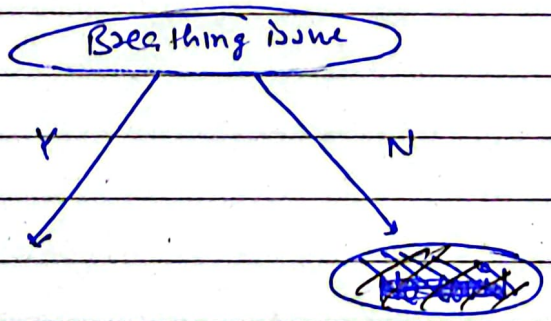
$$= 1 - \frac{6}{12} \times 0.9182 = 0.5409$$

$$IG\{cough\} = Entropy(whole\ data) - \frac{9}{12} \times 0.9910$$

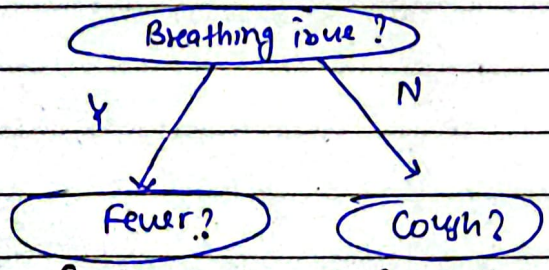
$$= 0.25675$$

$$IG\{breathing\_issue\} = 1 - 0 = 1$$

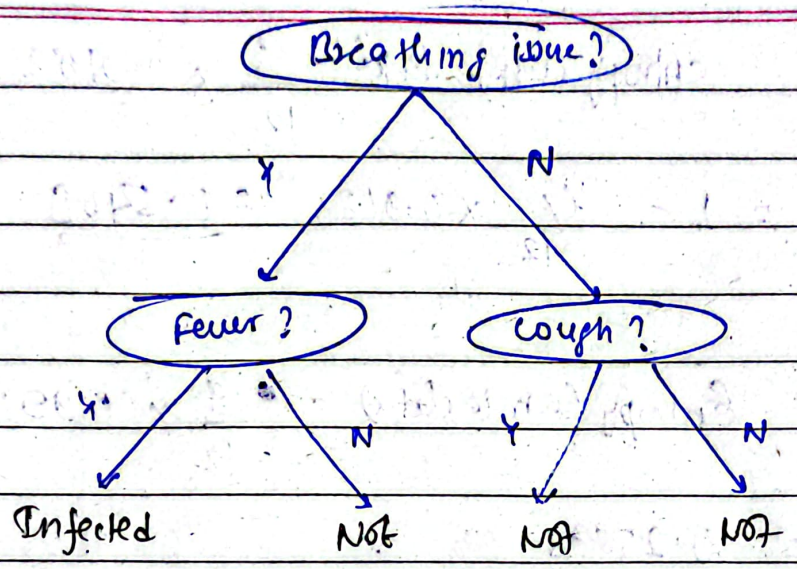
Here, IG{breathing_issue} == highest so, Breathing - issue = root node



Now, we select the next highest IG. ie. IG{Fever} and fix it in left.



and next parameter for right of rootnode will be cough since its only one left unused



For all Fever Yes and Breathing issue = Yes
and other than that none of it is infected