Machine Learning

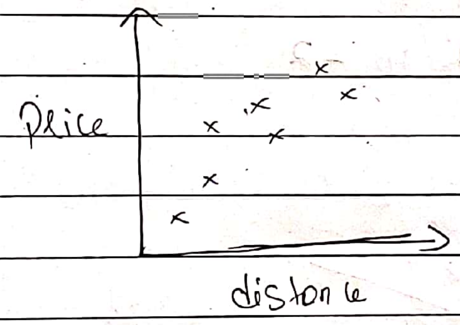
Supervised learning

Topic 1

Linear Regression

-> very intuitive supervised learning Algorithm and a regression technique

-> Ex:- Suppose we have data where we have car prices and its respective distance travelled.
If we plot it in a 2D graph



we can see there is a linear relationship between the price and distance travelled by car.

If you want to train a model, then it takes specific car model, it's price and it's distance travelled.

Mathematical function for car price model

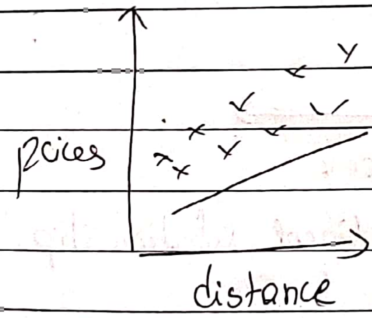
$$F(x) = A_1 x + A_0$$

Here A_1 & A_0 are weights. Weights are which, determine how are ~~ar~~ linear function behaves on different inputs. Determining the right weights is what we call learning.

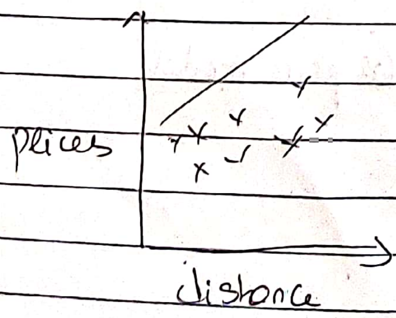
Let say if the value of A_1 is 5 & A_0 is 0.5 then it would be

good fit.

Let say if we change A_0 to -2



If we think the relation is steeper & shift it up increase the value of A_1 (as it determines the slope value) as much as 10, then it would be



Training:-

How do we actually learn the weight's for any model?
How's do we know that our labels are correct for cars in our dataset?

Training and evaluating a machine learning model is done using something called cost function.
In supervised learning cost function is the deviation between the predicted labels and true labels.

If the deviation is small, then cost is small, which means our prediction is more accurate.

Most common cost function for linear regression is least-squares cost function.

1. The cost function would be

$$C(x) = \frac{1}{2} \sum_{i=1}^{n} (F(x_i) - Y_i)^2$$

The deviation between predicted output $F(x_i)$ and true label (Y_i) is called residual.
The least squares cost is trying to minimize the sum of squares of residuals.

Here in $F(x)$ is our mathematical function with weights.
The weight A_1 & A_0 which produce optimal model are the weights which minimize cost function.

How do we compute optimal weights?

These are two ways:-
1) Analytical
2) Numerical

1) Analytical:- In Analytical we find for closed-form expression which for optimal value.
In this particular case, we can calculate gradients with respect weights of cost function and set those gradients too. Then solve the weights that achieve 0 gradient.

This solution is good but not models have well formed gradient expressions that allows us to solve for global optimum.

2) Numerical:- For the above problem, numerical method involve a step-wise update procedure where the weights are iteratively brought close to their optimal value.
Here we compute gradient with respect to weights and then apply following update for each weight A

$$A_{new} = A_{old} - \alpha \cdot \frac{\partial C}{\partial A}$$

we continue applying these iterations until these weights converge to global optimum. This numerical procedure is called gradient descent.

In this α is called hyperparameter. It determines how quickly updates are made to our weights.

When does linear Fit Fit?

When the data has linear relationship between each other.