Day 12, Oct-14, 2024

Setting up Supervised Learning Problems - Part I

We, human causally make predictions like estimating and predicting the bus arrival time, raining or Sunny without any explicit mathematics.

But our Computer relies on mathematics

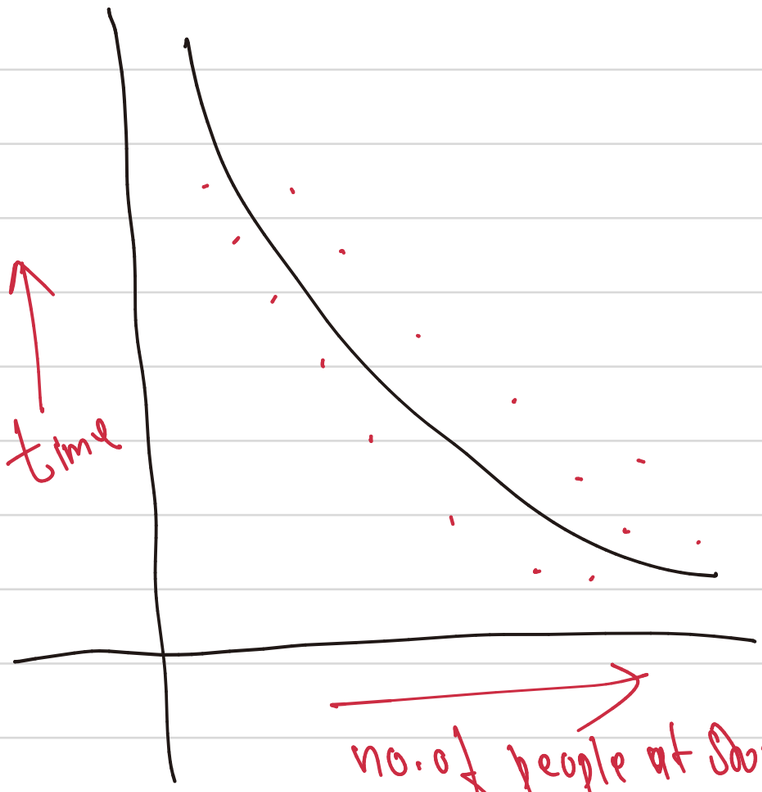
$x_i \in R^D$ where D is the Characteristics of envin

$x_i \Rightarrow$ features like people, weather

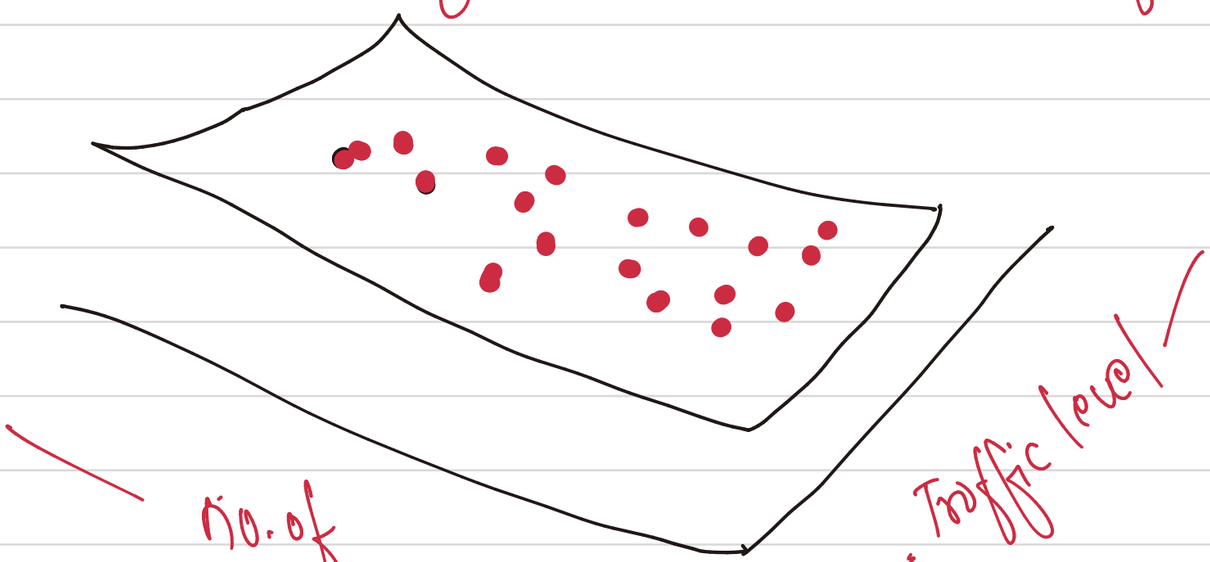
So, $\begin{pmatrix} x_1, y_1 \\ x_2, y_2 \\ \vdots \\ x_n, y_n \end{pmatrix}$

① Regression → Respone is Continuous

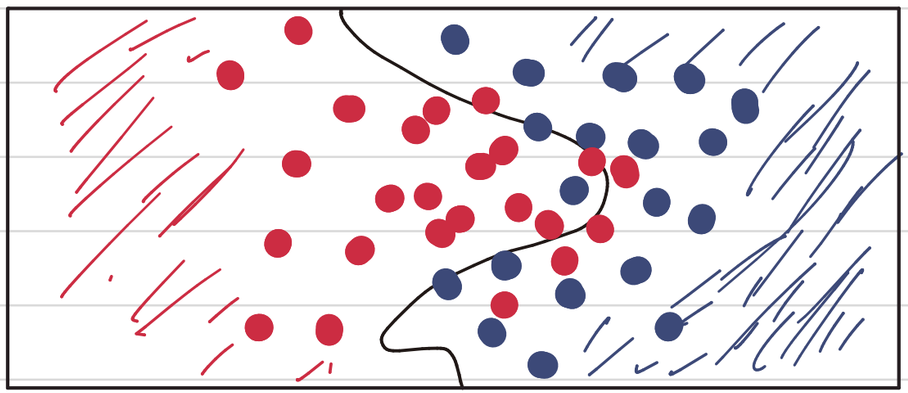
② Classification → Integer/Discrete or Classification



In - Higher Dimensional / Surface/Space



Classification example Blood Pressure → 1D



→ 2D Example of Blood Pressure prediction (Categorical)

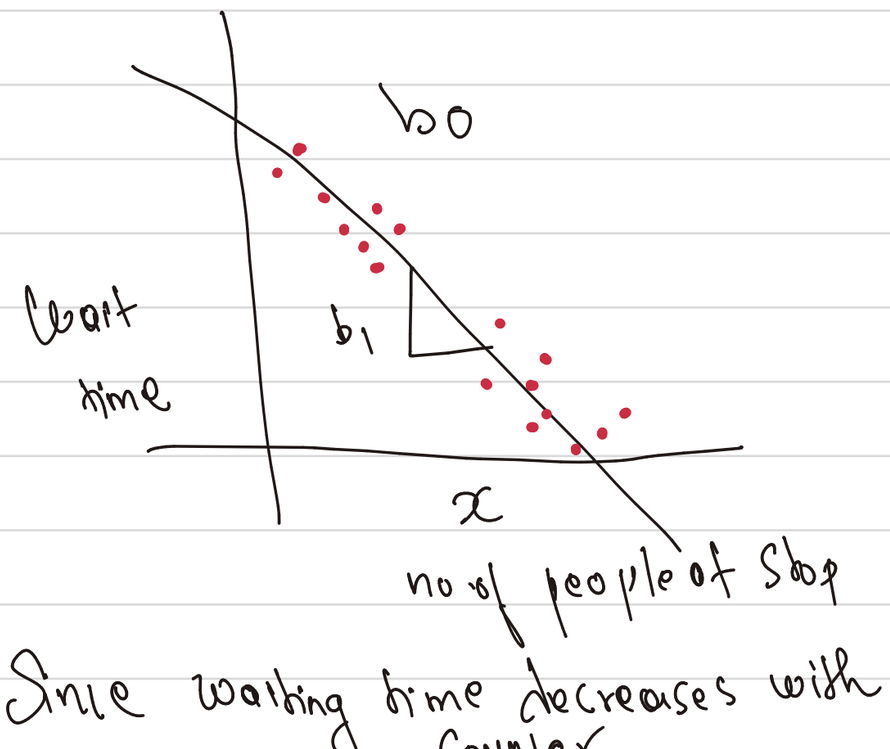
Linear Models

$$f_b(x) = b_0 + b_1 x$$

Eq for 'n' features.

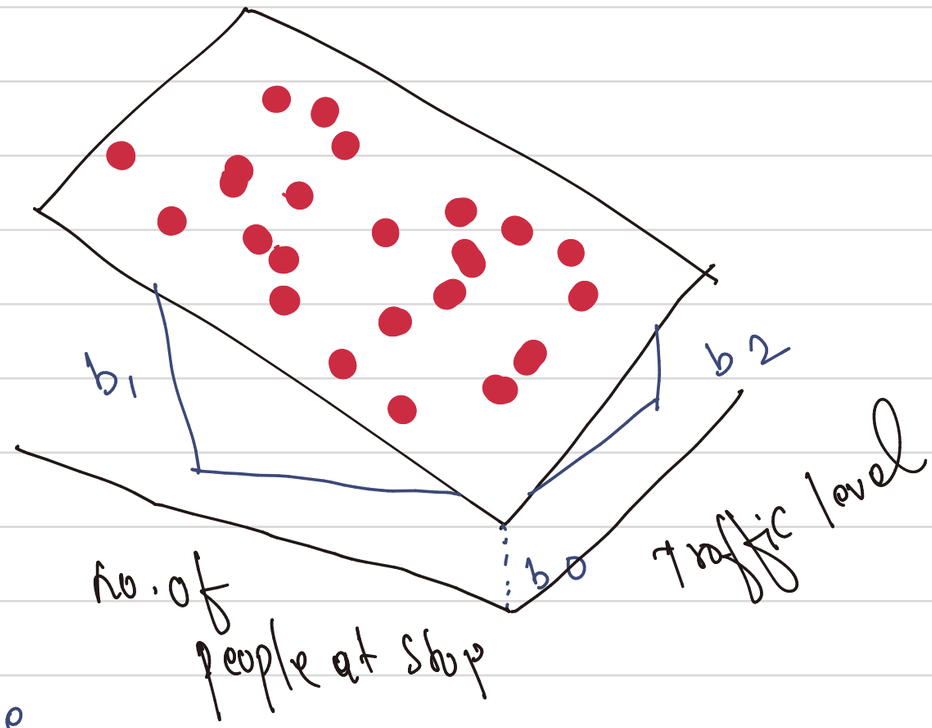
$$f_b(x) = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_D x_D$$

$$\boxed{f_b(x) \Rightarrow b^T x}$$



Since waiting time decreases with Counter

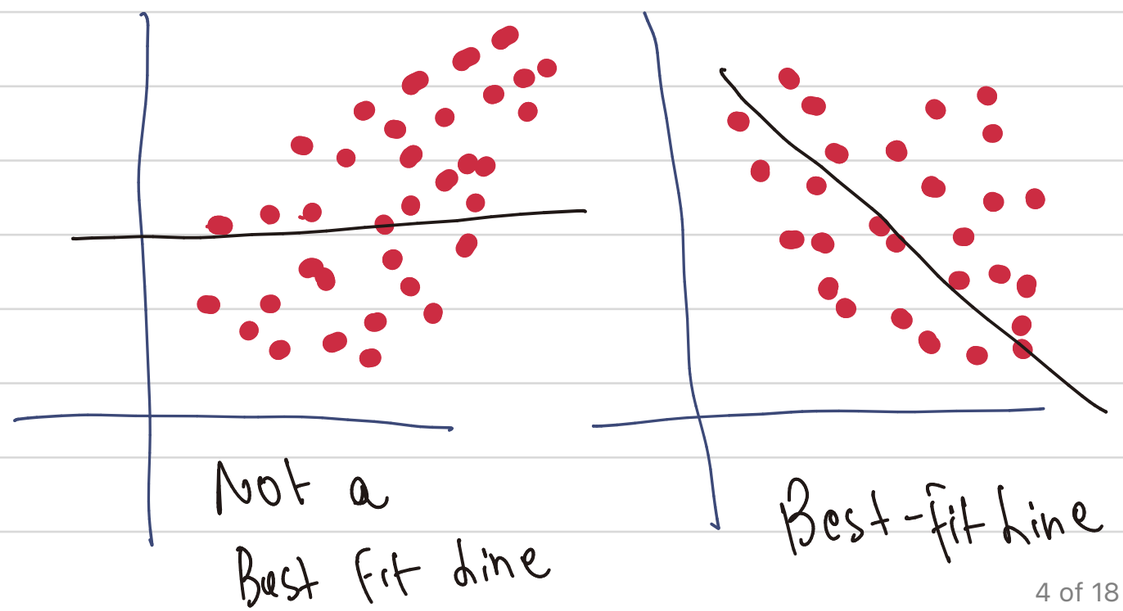
More Geometric Interpretation:



finding slope that fits the data well so we can use
① plimize a "loss" function.

For Linear Regression, a good choice is a Squared Error loss

$$\therefore L(b) = \sum_{i=1}^{N} \left(y_i - \vec{b}^{T} x\right)^2$$



For logistic Regression

$$f_b(x) \Rightarrow \frac{1}{1 + \exp(b^T x)}$$

$$\Rightarrow \frac{1}{1 + e^{-z}}$$

where $z = b_0 + b_1 x$.

for $z_n = b^T x$.

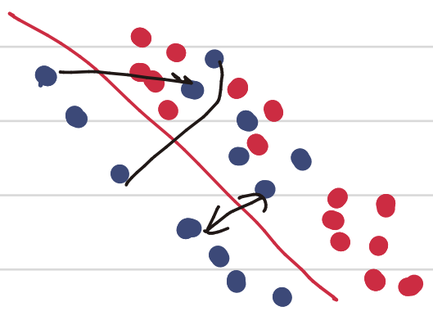
(Threshold = 0.5)

Threshold $\leq$ Outcome (1)

Threshold $>$ Outcome (0)

loss function in logistic Regression

$\longleftrightarrow$ is perpendicular and predicted error.



$$\left[\sum_{i=1}^{N} y_i \log\left(f_b(x_i)\right) + (1 - y_i) \log\left(1 - f_b(x_i)\right) \right]$$

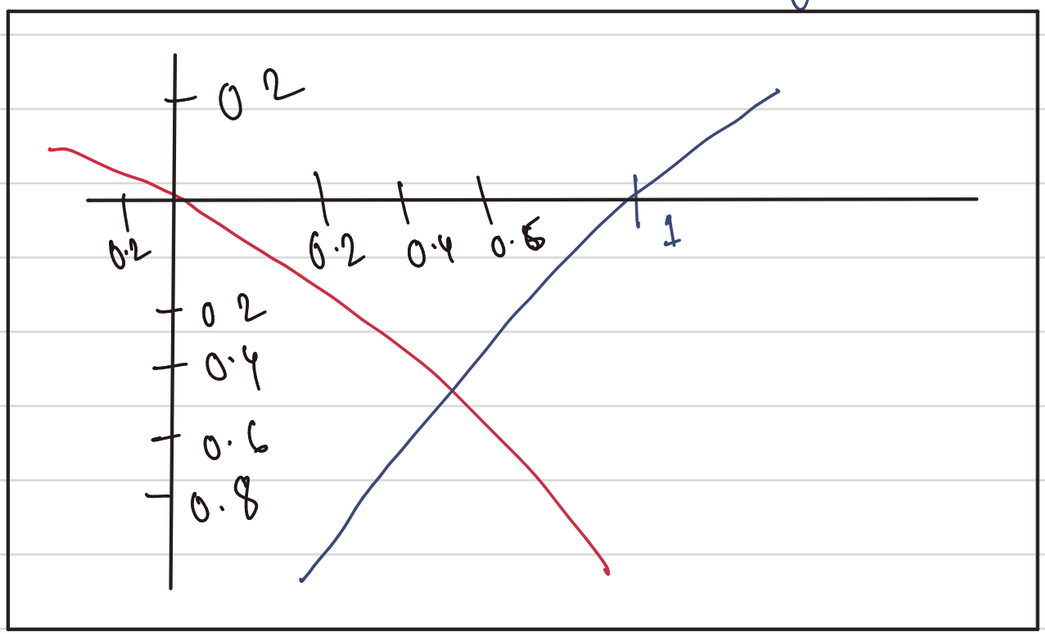
So, the above Loss function for Logistic Regression

$$\left[\sum_{i=1}^{N} y_i \log(f_b(x_i)) + (1-y_i)\log(1-f_b(x_i))\right]$$

when $y_i = 1$ red point on the graph

$y_i = 0$ blue point So it's a (binary Cross Entropy loss)

Graph for $\log(x), \log(1-x)$



So, for $y_i = 0$

$$\Rightarrow \sum_{i=1}^{N} 0 + (1-0)\log(1-f_b(x_i))$$

$$\Rightarrow (1-0)\log(1-f_b(x_i)$$

Interpretation:

if 1 is predicted with probability 1 then there is no loss and if 0 is predicted with probability 1 then there is loss.

$\log(x)$ and $\log(1-x)$ for $x=0$

$\log(0)$ and $\log(1-0)$

$\log(0)$ and $\log(1)$

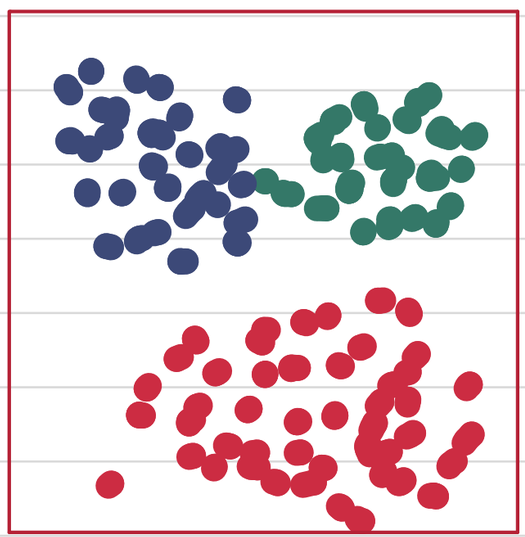
When model gets confident and totally wrong it gets penalty.

When model not gets confident and is correct also the model gets penalty (it could be L1 and L2).

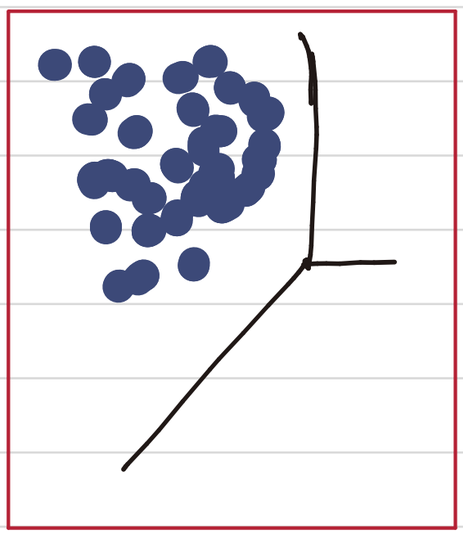
Coding Part See the Code.

⇒ We do prediction using the features data & the features are $x_1, x_2, x_3 \cdots x_n$ and finding it's slope m or b_1 or Coefficient $b_1, b_2 \cdots b_n$ and we have y-intercept or bias b_0 → this is a singe variable (error).

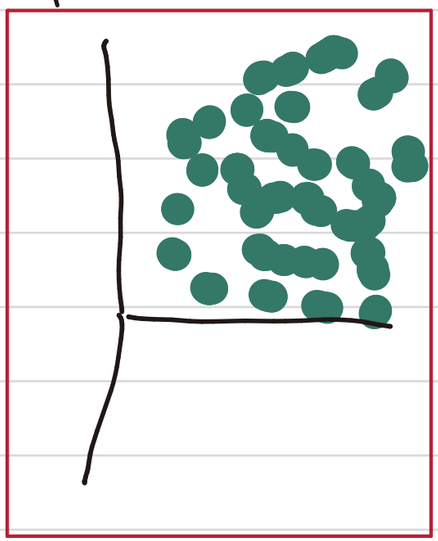
Clustered or Predictions



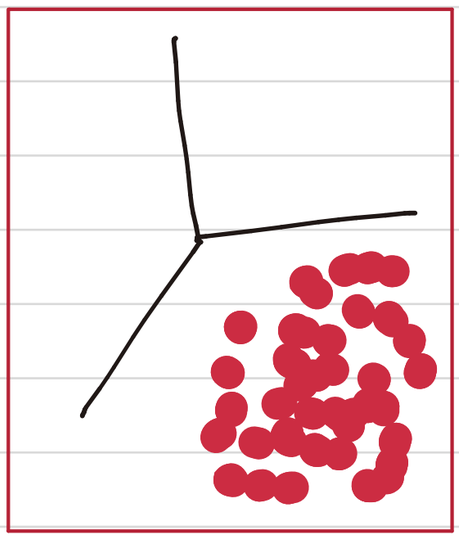
Prediction 1



Prediction 2



Prediction 3



for Linear Regression Decision Boundary is linear (But Not for Complex Process)

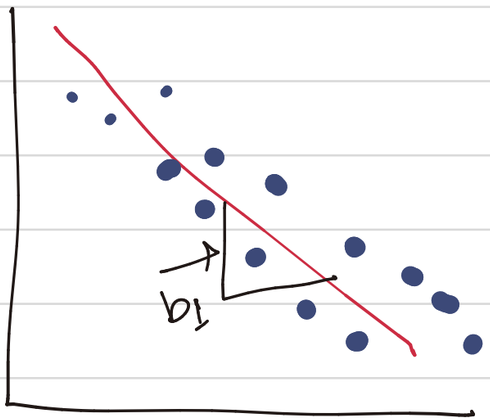
Sparse linear Models:

A model that knows how to ignore the relevant features will always do better than that tries to use them all. This is the main idea behind using sparsity in linear Regression to fit the model because we can have 'n-' features' and it is likely that our model will learn noise too.

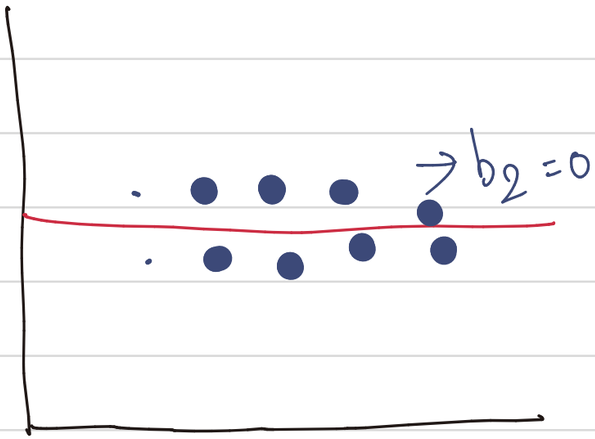
$$f_b(x) = b_0 + b_1 x_1 + \dots + b_n x_n \Rightarrow b^T x$$

but what we make the assumption that many of the b_d are exactly 0. Generally, we imagine that response don't change at all we

Change Some of the Inputs, all else held equal



Variable 1
with the relevant feature



Variable 2 (Slope = 0)
with the irrelevant features

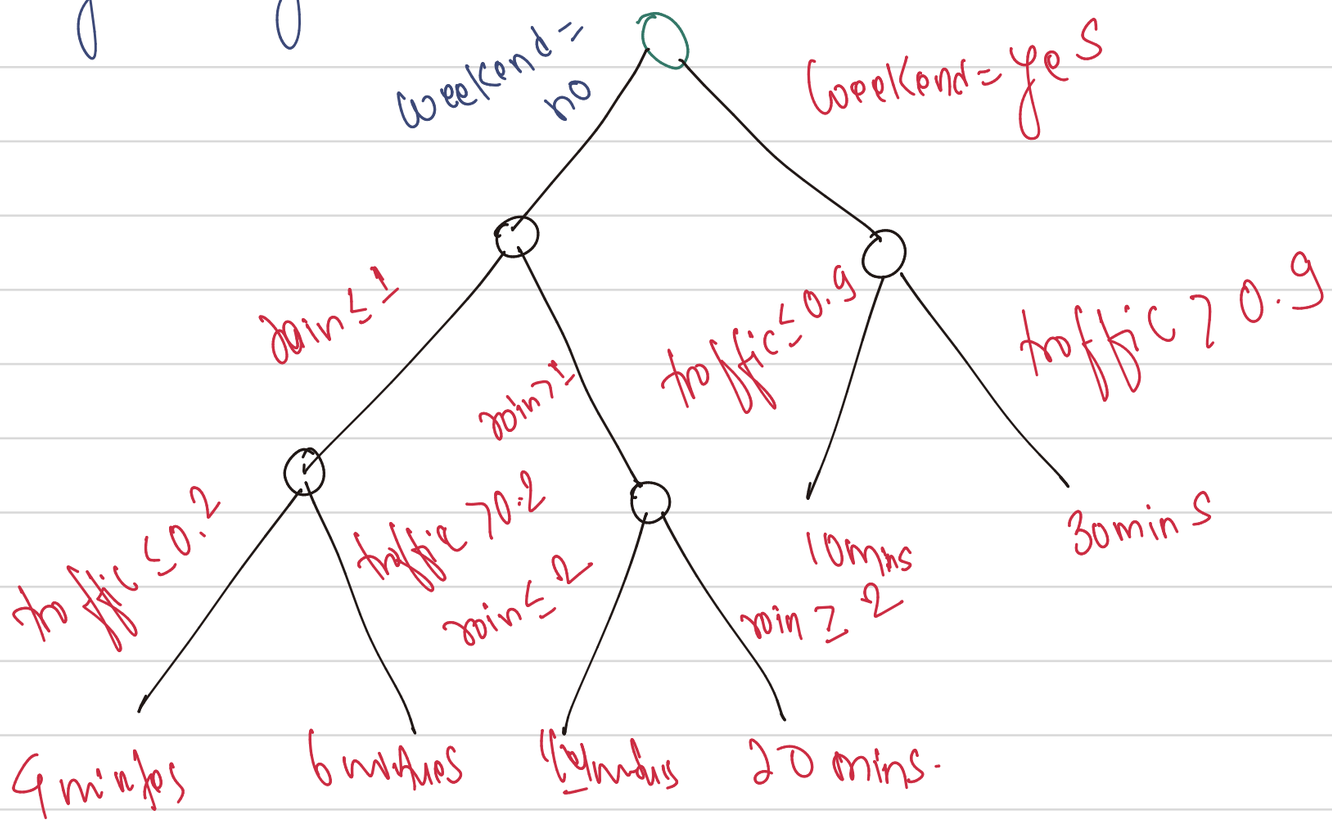
features has no contribute

Example Decision Boundary in Above Current Blood Pressure does the Same Level of Sparsity.

Elastic Net in the Sparsity, So the linear_model.ElasticNet(alpha = 1e-1
use shot ↑ - - - - - -)

Tree-Based Models

→ Tree-based models fit a different class of curves. To motivate them, consider making a prediction for the bus time arrival problem using the following diagram:



We can use some logic to do either regression or classification

for Regression each "leaf" at the bottom of the tree is a continuous prediction. For classification, we associate leaves with probabilities of different classes"

It turns out that we can train these models using Squared Error & Cross-Entropy (χ) losses before, though the details are beyond the scope of these notes.

So, we can infer that these rules are equivalent to drawing curves that are piecewise constant over subsets of the input space. Let's convince ourselves using some pictures. First, notice that a tree with a single

Split is exactly a "curve" that takes two values depending upon the split point.

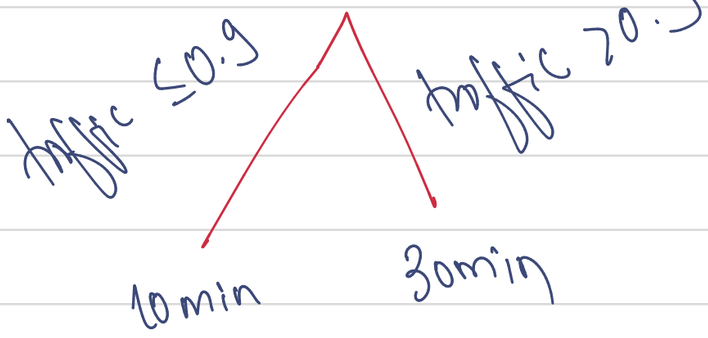
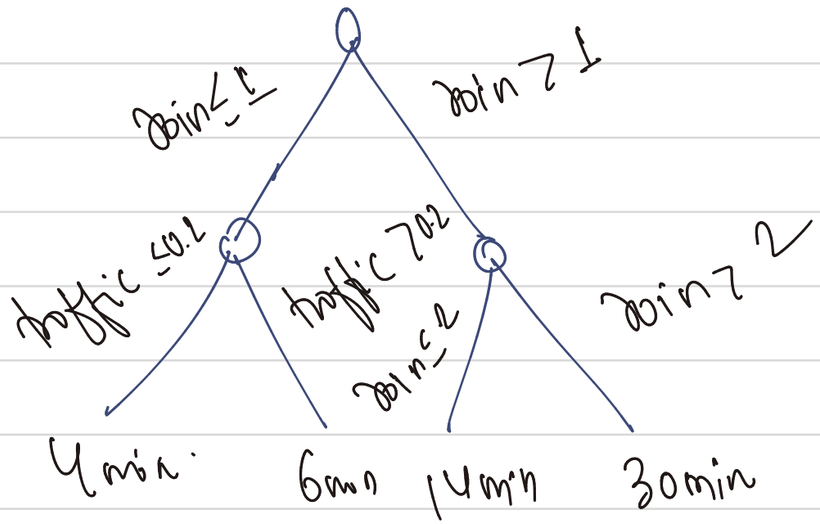


Fig. Curve illustrations

if we split the same variable deeper, it creates more steps.

What if we had two variables? Depending on the order, of the splits, we create different axis-aligned partitions.

Let's take more complex-



These tree-based methods arises the decision-tree and give rise to a new method we call it a random forest.

It does not work well as the complexity (decision-tree) so the collection of optimized decision-tree are often called Random-Forest.

The Random Forest and Gradient Boosted Decision Trees.

Random forest vs Gradient Bousted Decision Trees

→ Creates multiple decision trees independently focus on reducing variance(avoids overfitting by averaging)

→ used for both classification and Regression

→ builds trees Sequentially

→ Each tree corrects the error made by previous one (boosting)

→ focusing on reducing bias (Improves model accuracy progressively)

So, the tree-base methods have non-linear decision boundary as Compared to linear Models. Let's understand and Summarize.

Baselines.

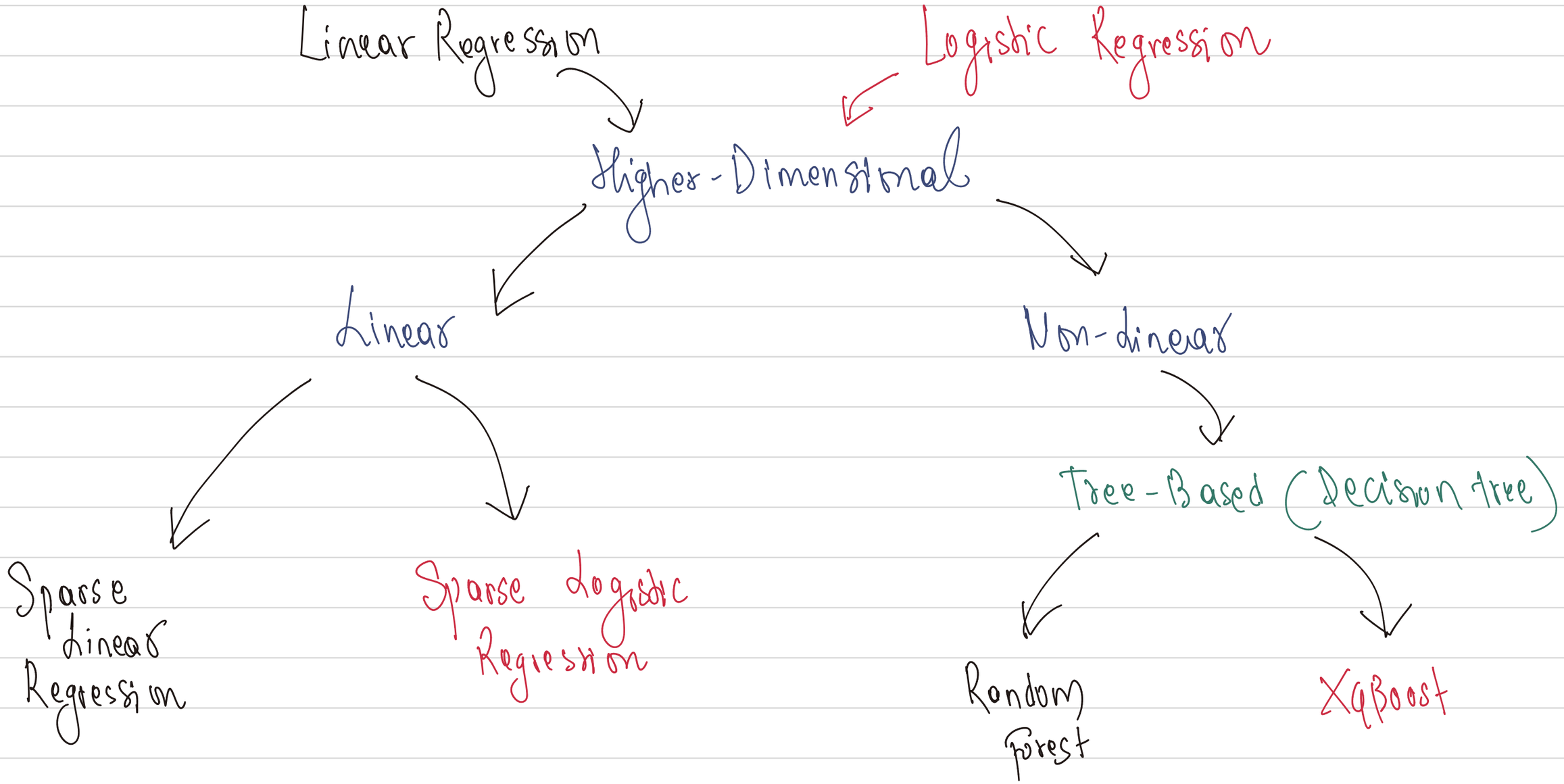


Fig. Hierarchial Tree showing the linear and Non-linear Model for Regression & Classification

→ Quick and less features use logistic and linear Regression

→ Above figure Black is Regression and Red is Classification

→ Tree-Method Based Method are slower to train but handle Complex decisions and can be difficult to interpret.

	Strengths	Weakness
Linear / Logistic Regression	- No tuning parameters → very fast to train	→ Unstable when many features to pick from → Can only fit linear (curves) boundaries

Sparse linear/ Logistic Regression

→ Stable when every/many features to pick up

→ Can only fit linear curves

Tree-based Classification & Regression

→ Can fit nonlinear functions of inputs

→ Harder to interpret