LEAP

AI@PLAKSHA

Logistic Regression

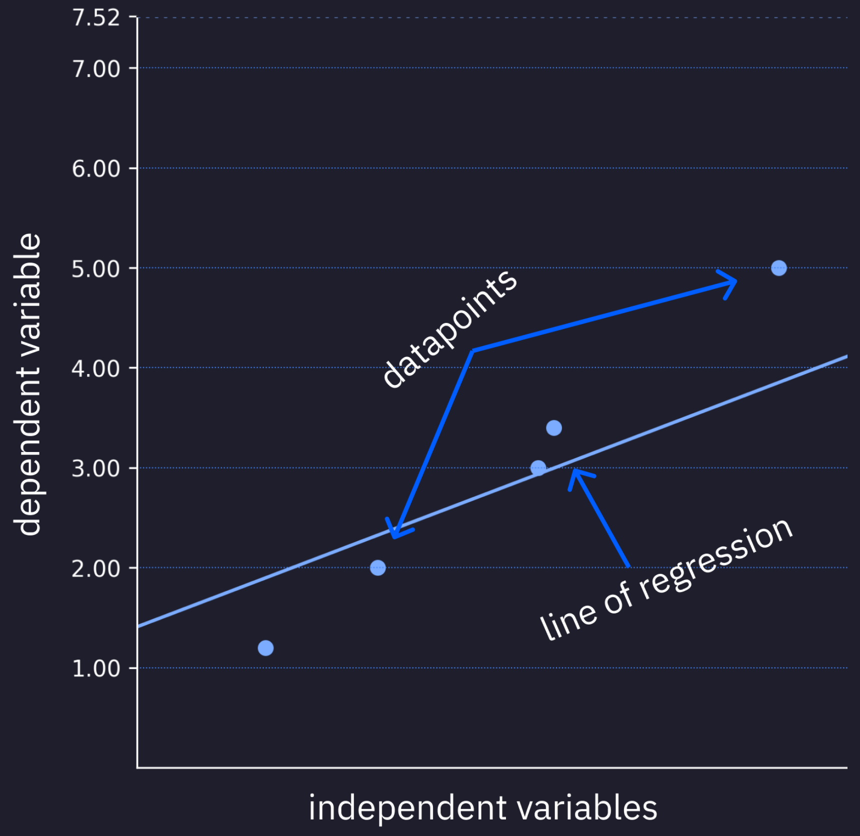
28 November 2023

Learning & Exploration in AI Practices

LEAP
AI@PLAKSHA

Review of Linear Regression

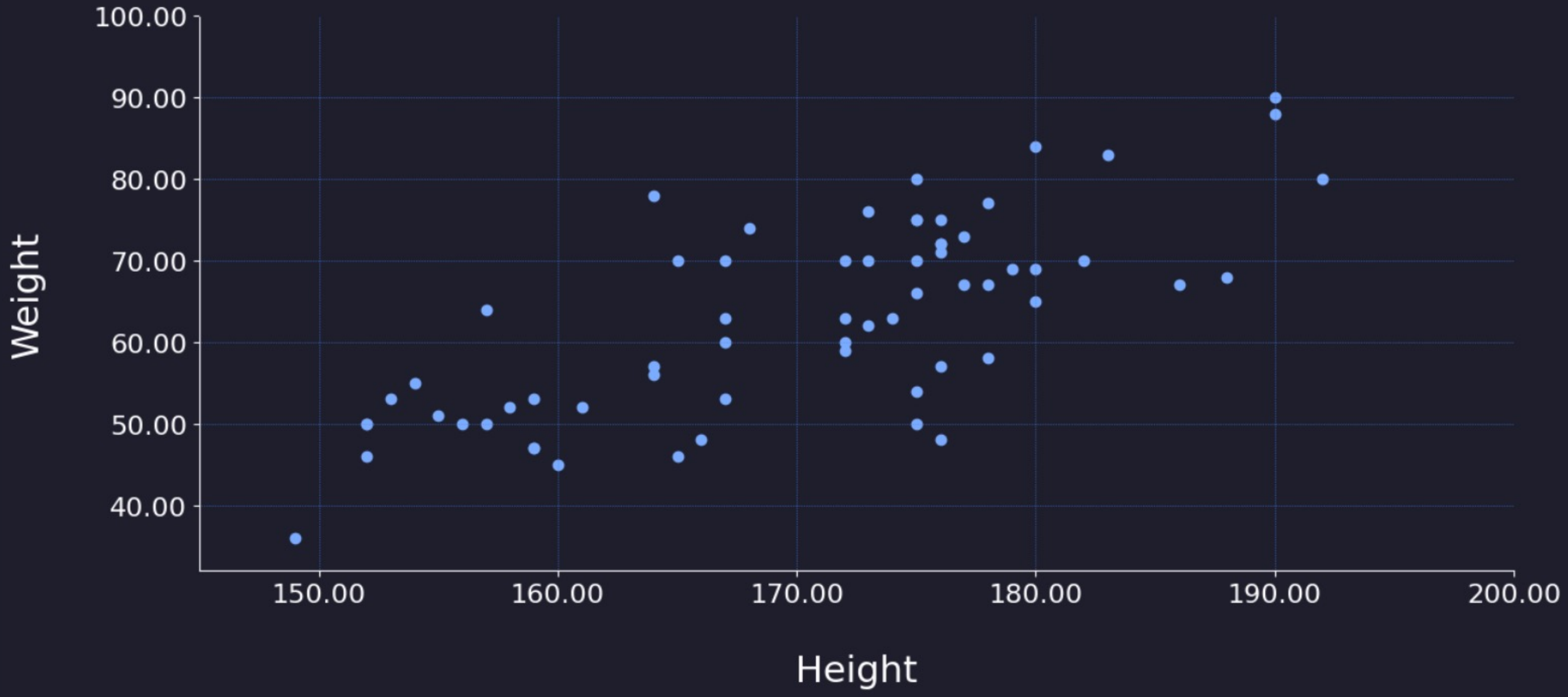
01



- Used to predict continuous values.
- Fit a line on the data
- This is done using the mean square error (MSE) method.
- Use the line to predict the dependent variable given the independent variables.

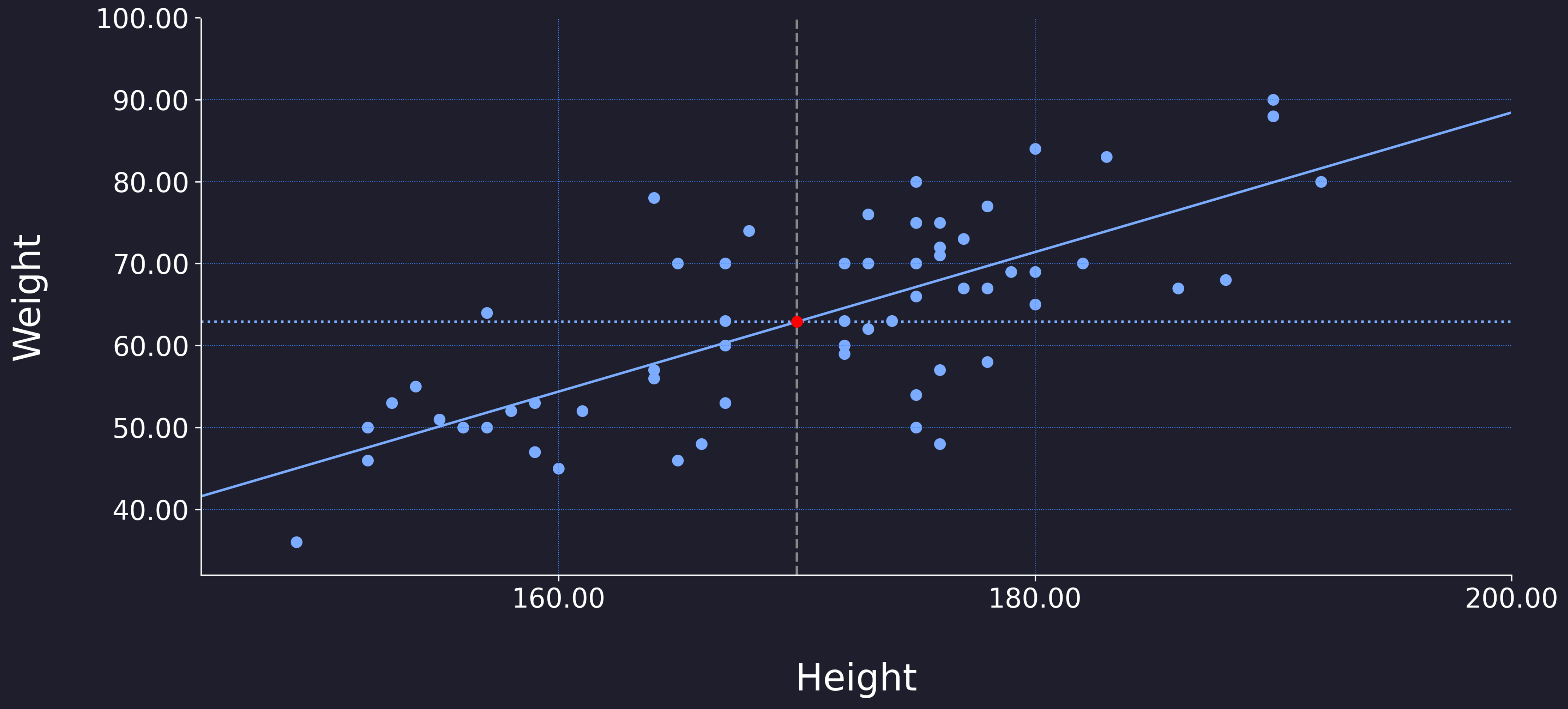
Review of Linear Regression

02



Review of Linear Regression

03



LEAP
AI@PLAKSHA

Logistic Regression

Logistic Regression is like linear regression except...

- It is used to predict **True or False** values, instead of predicting something continuous like ***height***.
- Classification.
- It fits data in an ***S-shaped curve***, instead of a line.

05

Why can’t we use linear regression for classification?

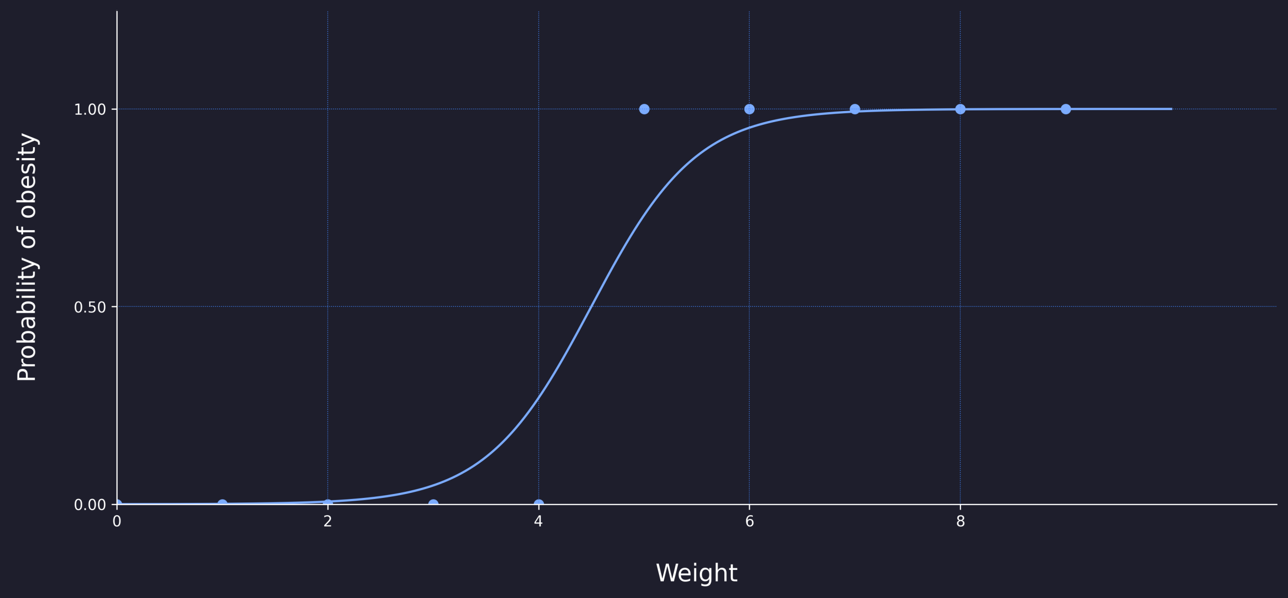
LEAP
AI@PLAKSHA

Linear regression is used to predict continuous values.

Why learn this?

Binary Classification.

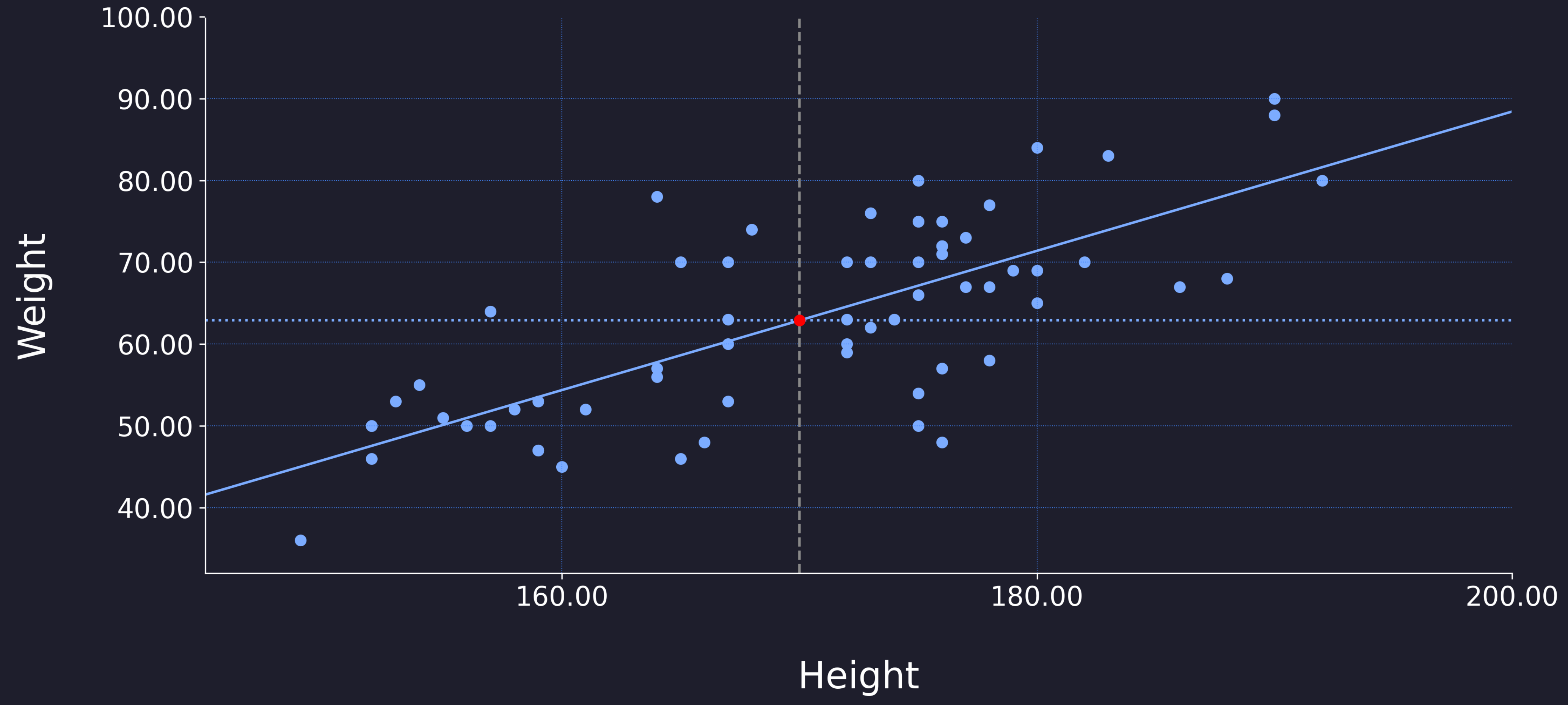
- Spam or not
- Pass or fail
- Buy or not
- Disease or not



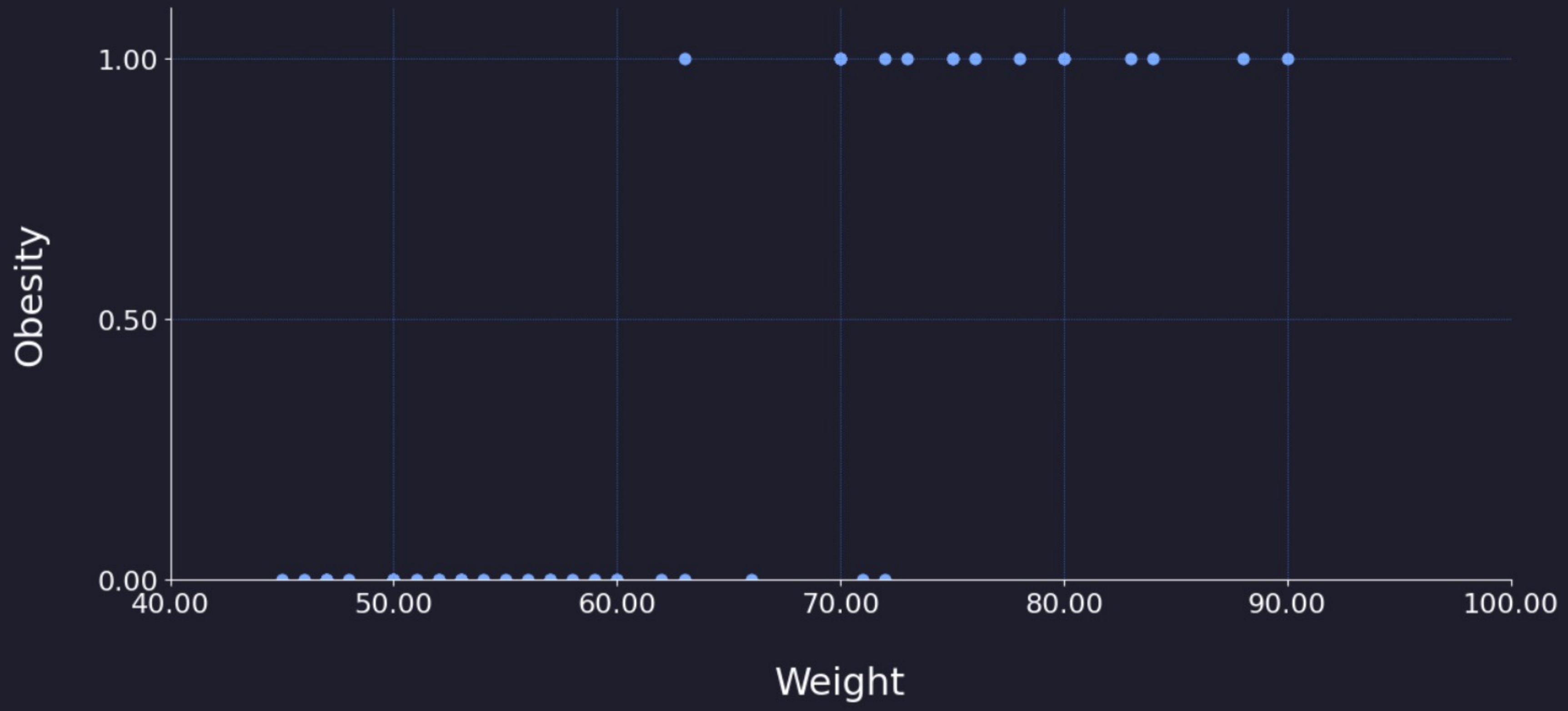
Interpretability.

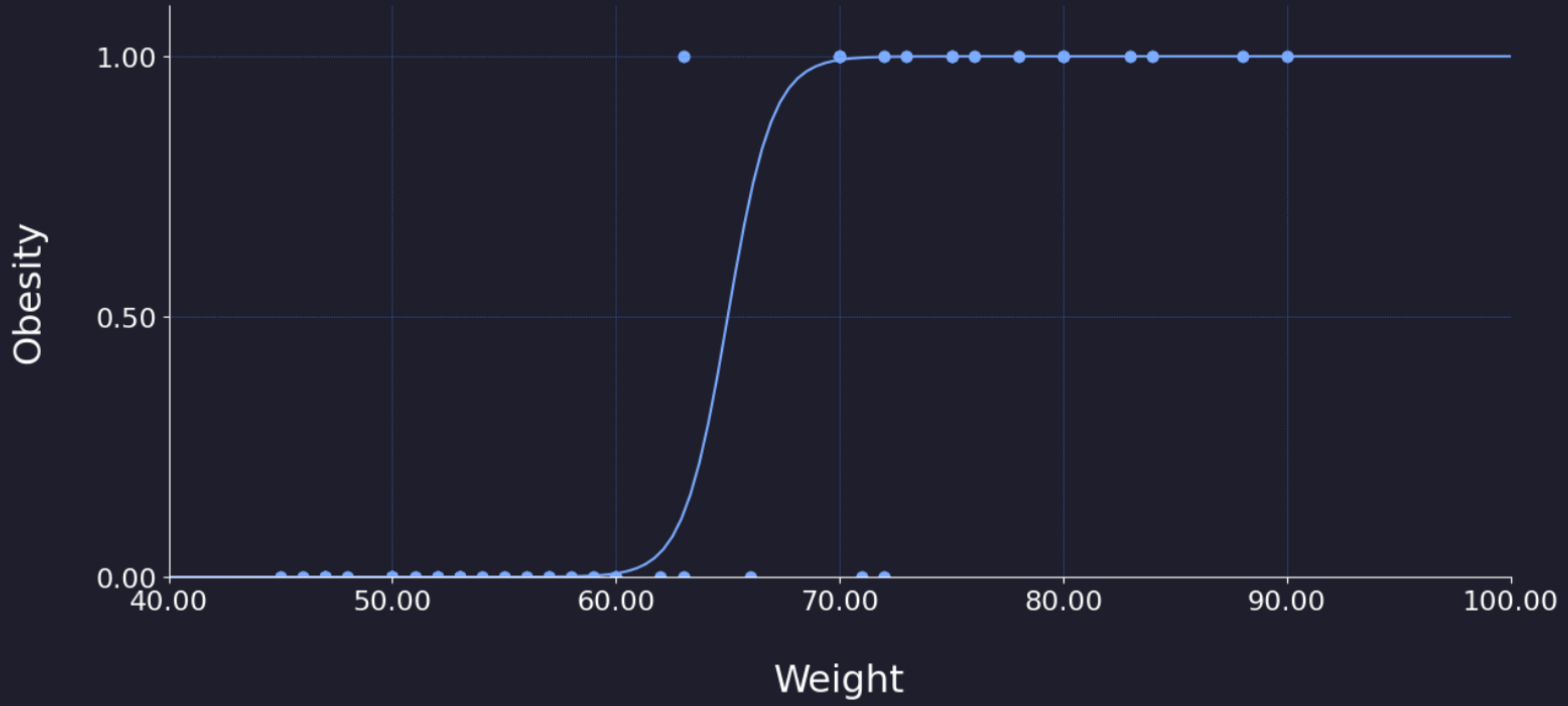
Ease of Implementation.

08


100.00
90.00
80.00
70.00
60.00
50.00
40.00
Weight
160.00
180.00
200.00
Height

Weight	Obese
50	0
90	1
70	1

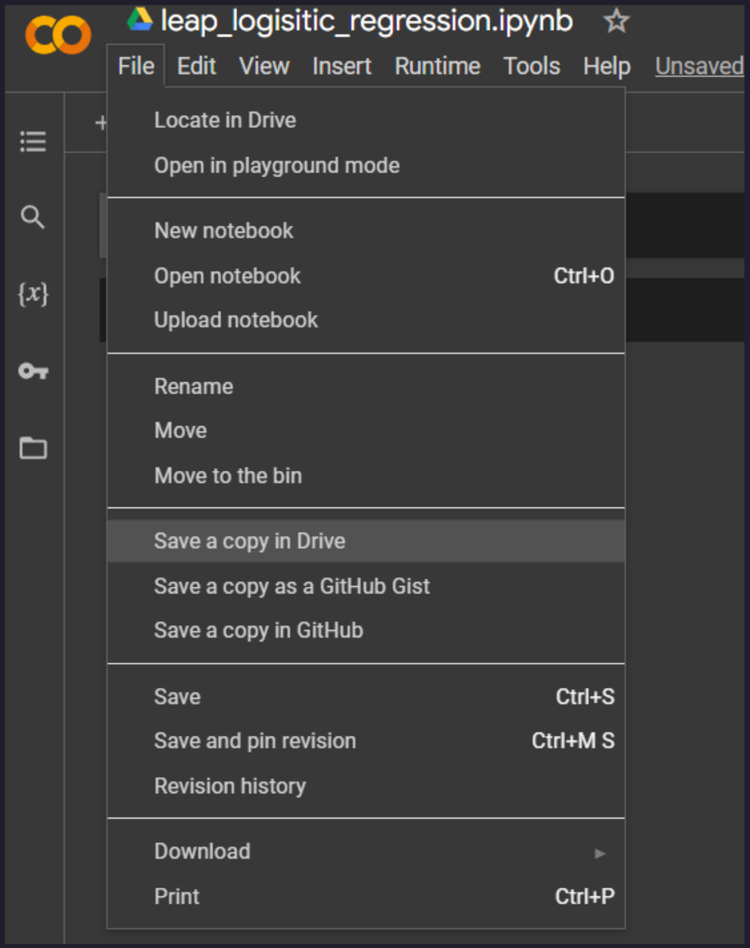




LEAP
AI@PLAKSHA

Colab Notebook

- Like before, go to **leap-ai.tech**
- Download the Colab Notebook
- Go to File->save a copy in Drive



Data Collection🚙

Will you accept the ride?

Data Collection

Features:

x_1: Distance from Plaksha (km)
x_2: Uber price (rs)

Output:

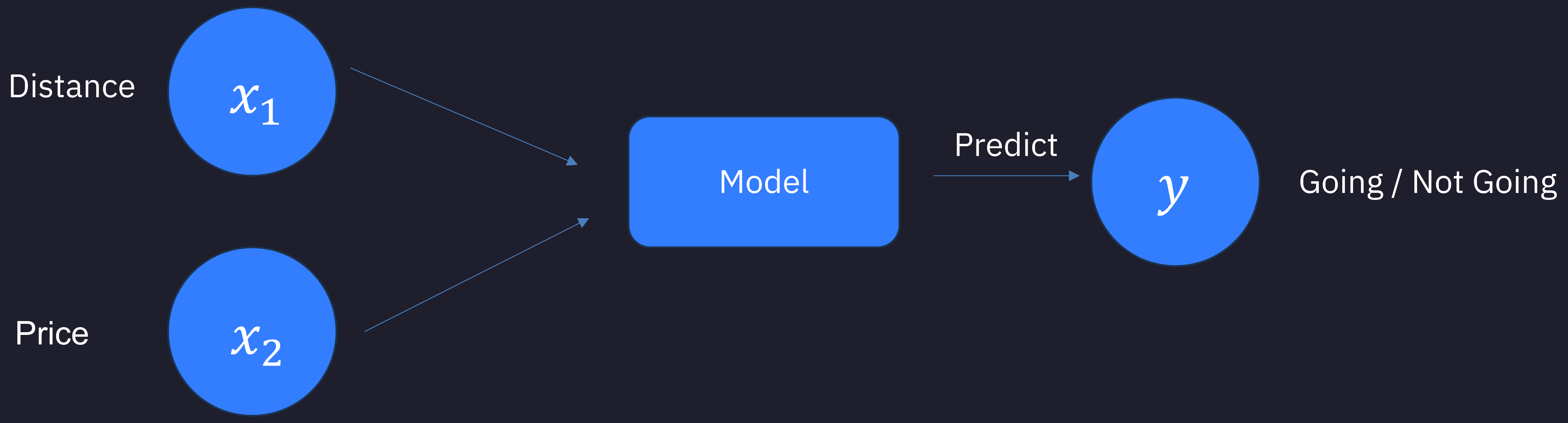
$$y: \begin{cases} 1, \text{you } \textit{take} \text{ the cab} \\ 0, \text{you } \textit{reject} \text{ the cab} \end{cases}$$

Examples:

$$\vec{x} = (x_1, x_2) = (5, 400)$$
$$y = 0$$

$$\vec{x} = (x_1, x_2) = (5, 200)$$
$$y = 1$$

Data Collection



15

Let’s Start

Building a binary classification model

Let’s understand our model’s workflow step by step

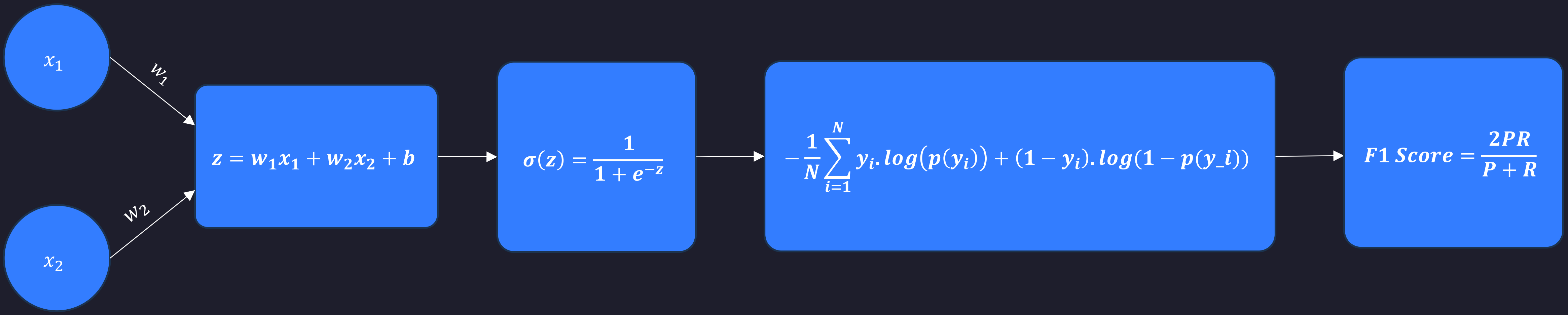
Input

Linear Model

Sigmoid Activation (σ)

Calculate & Minimize Cost

Evaluate & Predict



Features/Inputs for our model

Features are the variables on which our model will make the predictions. Basically, the inputs for the model
These will the features for our model

In our case,

$$x_1 = \text{Distance from Plaksha}$$

$$x_2 = \text{Price for the cab}$$

Input | **Linear Model** | **Sigmoid Activation (σ)** | **Calculate & Minimize Cost** | **Evaluate & Predict**

$x_1 \xrightarrow{w_1}$, $x_2 \xrightarrow{w_2}$

$$z = w_1x_1 + w_2x_2 + b$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$-\frac{1}{N}\sum_{i=1}^{N} y_i.log\big(p(y_i)\big) + (1-y_i).log(1-p(y\_i))$$

$$F1\ Score = \frac{2PR}{P+R}$$

Let's make this cleaner;

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$z = w.x + b$$

For linear regression, we would have used $z = w_1x_1 + w_2x_2 + b$ *or* $z = w.x + b$ for prediction but now what?

We need to transform the z to give us value between 0 and 1.

Guess

Σσ

21

AI@PLAKSHA

Sigma.. **Sigmoid Activation**

A mathematical function having a characteristic S-curve that takes real values to give values between 0 and 1. (useful to model probabilities)

$$\boldsymbol{\sigma(z) = \frac{1}{1 + e^{-z}}}, where\ \boldsymbol{z} = \boldsymbol{w}.\boldsymbol{x} + b$$

$$with\ w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Visualization

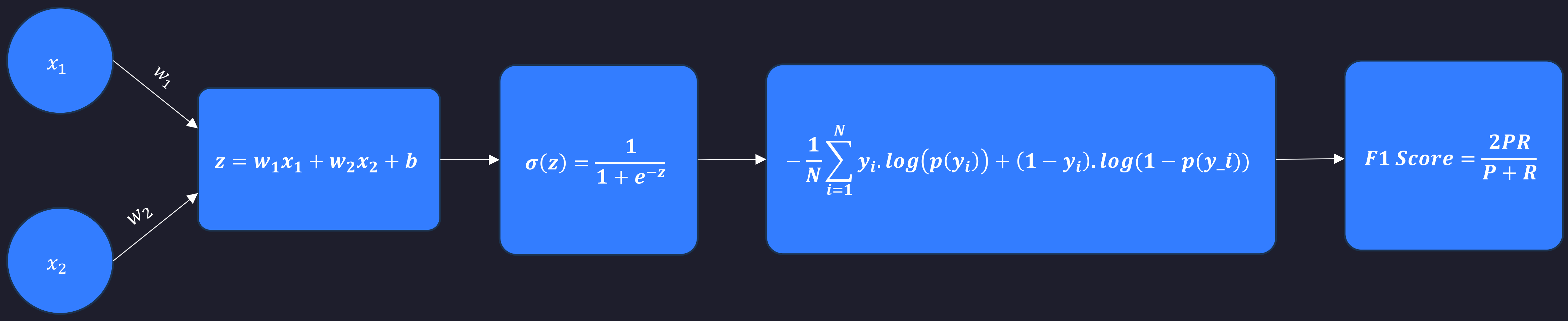
Input

Linear Model

Sigmoid Activation (σ)

Calculate & Minimize Cost

Evaluate & Predict



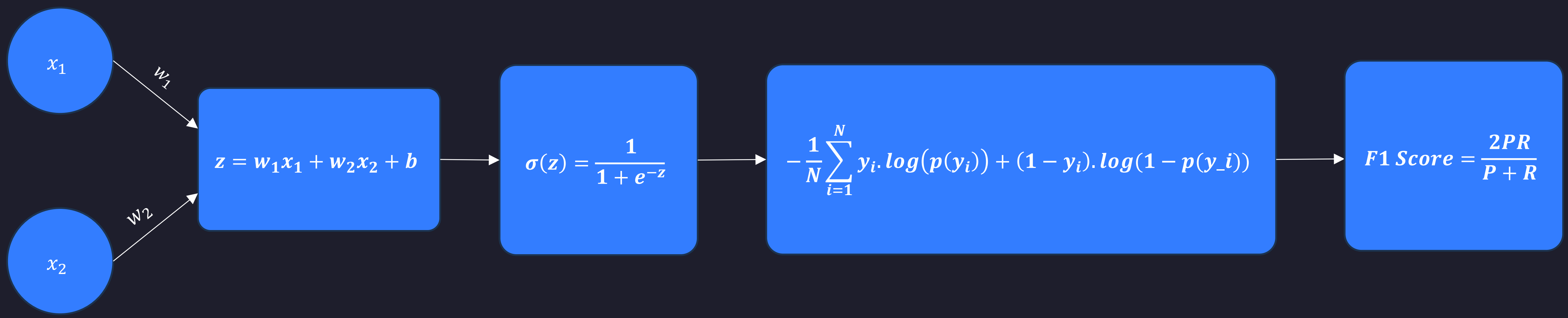
Input

Linear Model

Sigmoid Activation (σ)

Calculate & Minimize Cost

Evaluate & Predict



Binary Cross Entropy Loss 👨‍💻

Uncertainty rules!

Cost

Cost Function is a way to penalize the model for not doing its intended purpose.

It instructs the machine learning model how it should **learn.**

Using the right cost is integral for efficient training of any Machine Learning model.

What do we need?

What do we need for optimization of our model?
What is our goal?

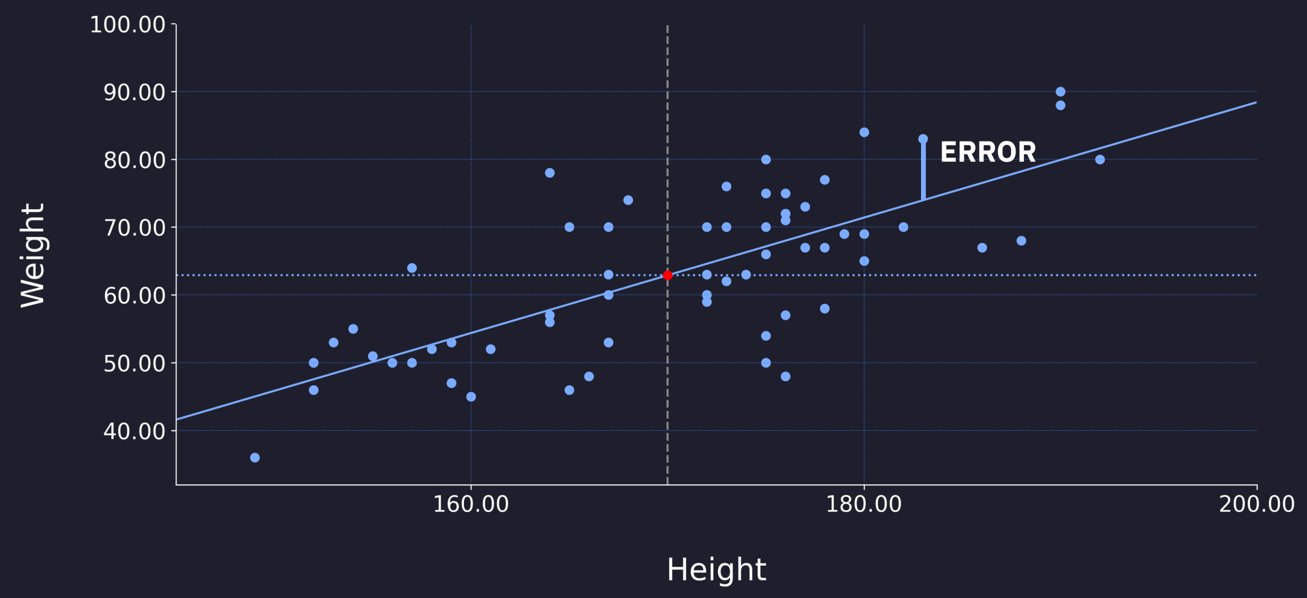
PROBABILITY!

The probability of predicting the right class!

RE: Mean Squared Error

Mean Squared Error was the error we used for the Linear Regression model.

Mean Squared Error tells the model about how far the ground truth is from the current model.



Will MSE Work?

Mean Squared Error does work, but it does not give the model a good metric to learn off on, it does not punish the model enough.

This is because MSE works on the distance of the ground truth from our model, but we need a loss to account for probability

This is where **Binary Cross Entropy Loss** comes into picture !

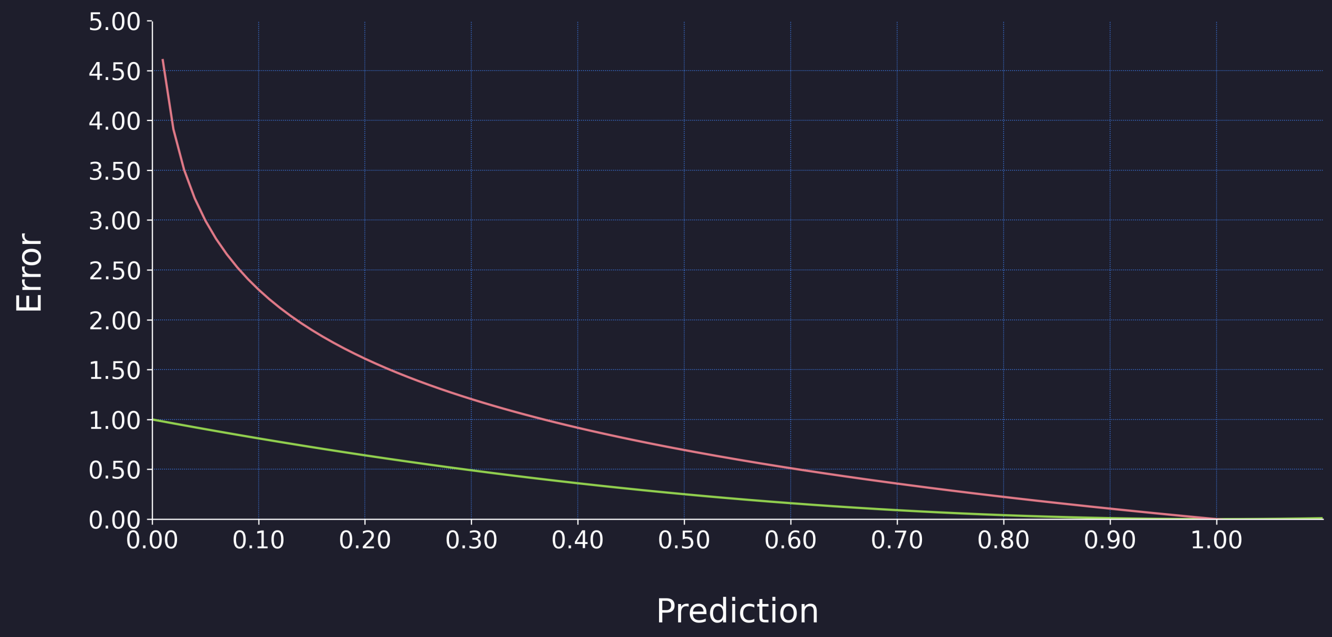
Binary Cross Entropy Loss

First, let's look at how profound the difference is.

We are assuming the actual value (ground truth) to be y=1

x-axis is the prediction [0,1]
y-axis is the loss/penalty

The green graph is MSE
The red graph is BCE



Binary Cross Entropy Loss

Binary Cross Entropy Loss is a **loss to maximize probability**, so taking

$$L = y^{\hat{y}} + (1 - y)^{1-\hat{y}}$$

$$L = \hat{y} \cdot \log(y) + (1 - \hat{y}) \cdot \log(1 - y)$$

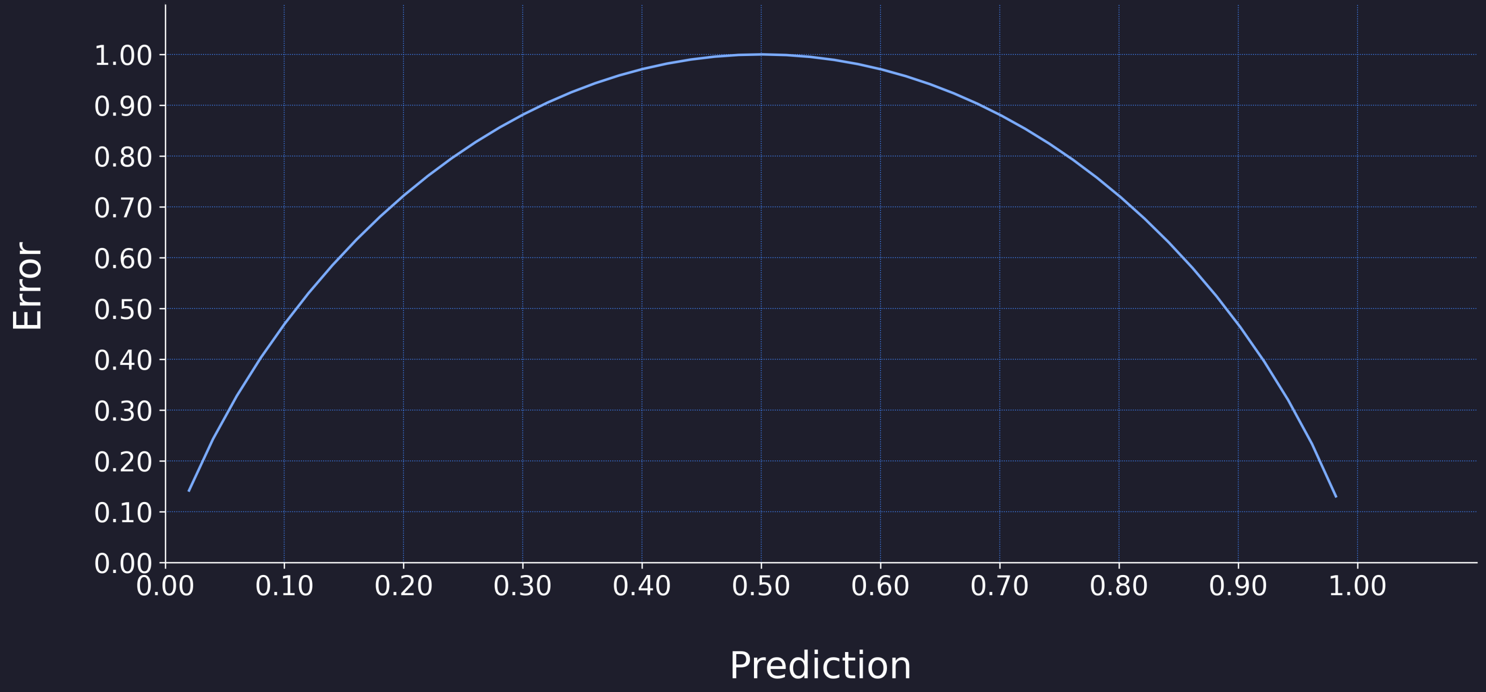
$$L = -(\hat{y} \cdot \log(y) + (1 - \hat{y}) \cdot \log(1 - y))$$

Now as you can see, this function can also be called a Negative Log Likelihood Loss

What is Life? Why am I here?

Entropy is the mathematical measure of uncertainty.

$$E = -p\log(p) - (1-p)\log(1-p)$$



Cross Entropy ⚔

It is the entropy between 2 probability distributions.

It tells us how well does one distribution models another distribution.

It refers to the **uncertainty of the modeled probability distribution being equal to the target probability distribution**, a high uncertainty of this gives a high cross entropy value and vice versa.

Cost

Now finally since we now intuitively know that the loss we have works.

We will convert this to a **cost function.**

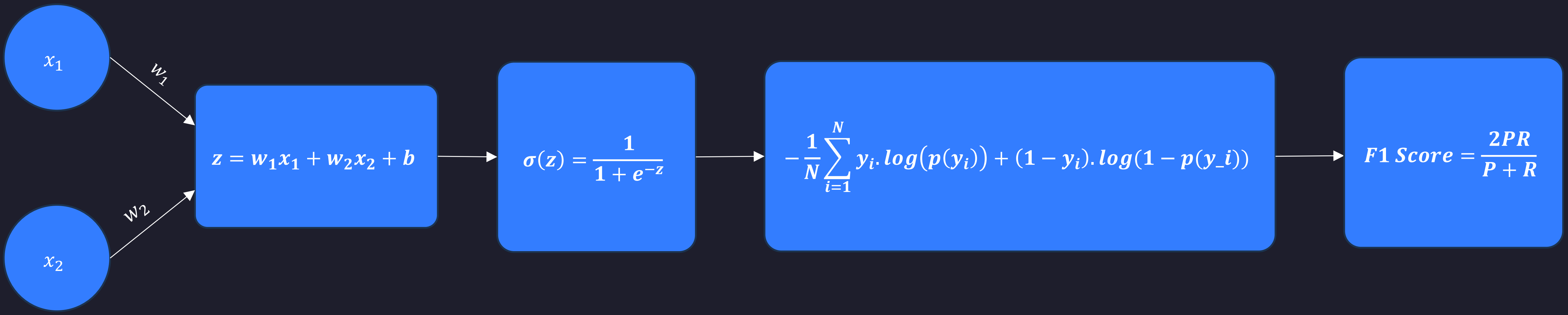
$$\boldsymbol{Cost} = -\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{N}} \hat{y} \cdot \log_2(\boldsymbol{y}) + (\mathbf{1} - \hat{\boldsymbol{y}}) \cdot \boldsymbol{log}(\mathbf{1} \; - \hat{\boldsymbol{y}})$$

Now let's start coding this.

Quiz

Go to **joinmyquiz.com**

Input **Linear Model** **Sigmoid Activation (σ)** **Calculate & Minimize Cost** **Evaluate & Predict**

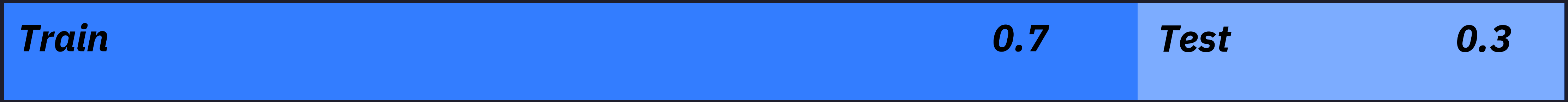


Evaluation 👨‍🔬

How do we know we are successful?

Testing?

- All available data is not used for training.
- Testing always on *unseen, 'new' data.*

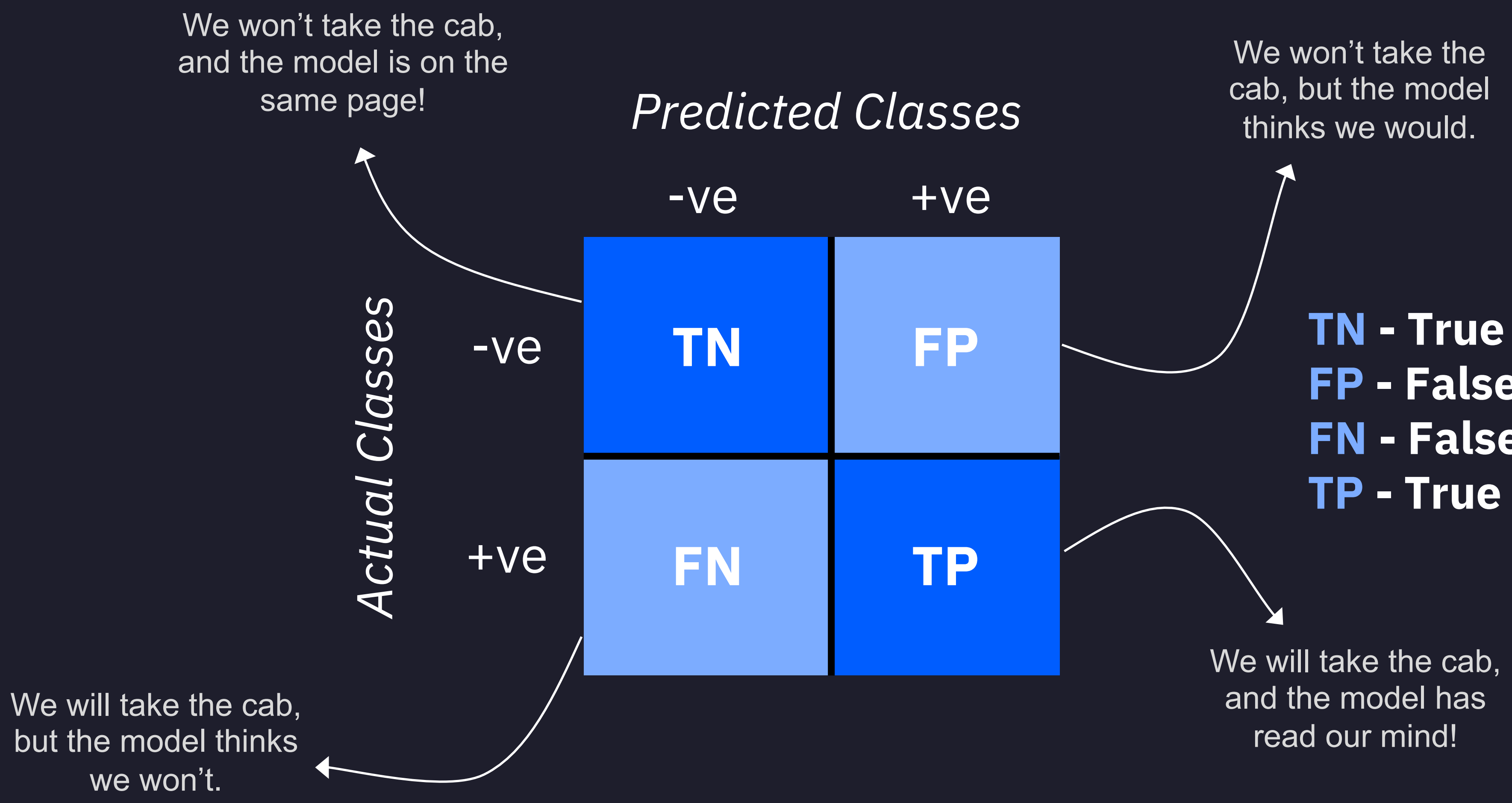


Accuracy

$$\frac{correct\ predictions}{total\ number\ of\ predictions}$$

- Why do you think this is useful?

Confusion Matrix



TN - True Negative
FP - False Positive
FN - False Negative
TP - True Positive

AI@PLAKSHA

Precision & Recall

- Precision:
  - *What proportion of positive identifications were actually correct?*
  - *When the model thought we would take the cab, how many times was it correct?*

$$\frac{TP}{TP + FP}$$

- Recall:
  - *What proportion of actual positives were identified correctly?*
  - *When we would take the cab, how many times was the model on the same page?*

$$\frac{TP}{TP + FN}$$

LEAP AI@PLAKSHA

The best Precision and Recall?

- Best case?
- Prioritizing one might lead to a worsening other.
- *The FP and FN tug.*

$$\frac{TP}{TP + FP}$$

$$\frac{TP}{TP + FN}$$

F1 Score

- *A way to combine Precision and Recall.*
- It is the **harmonic mean** of the Precision and Recall.
- *A low Precision or Recall has a high impact in F1 Score.*

F1 Score

$$\frac{2\,\text{Precision} \ * \ \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{\text{TP}}{\text{TP} \ + \frac{1}{2}(FP + FN)}$$

A Note on Bias

- Despite our best try, all low scores. Especially accuracy.

  - Imbalanced data
  - “Confounding variables”
  - Bad data collection

Ranking up.

- F1 score is **sensitive** to changes in the classification threshold.

- Pre-process, Sample, Feature Select. Boost!

- *Consider that you might not need a high score everywhere.*

And we are done!

P.S. Check out the Post Workshop Content!

Bibliography

- https://developers.google.com/machine-learning/crash-course/classification
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://scikit-learn.org/stable/modules/model_evaluation.html
- https://www.ibm.com/topics/logistic-regression
- https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

LEAP

AI@PLAKSHA