An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Logistic Regression

Presented by:
Shreyas Shukla

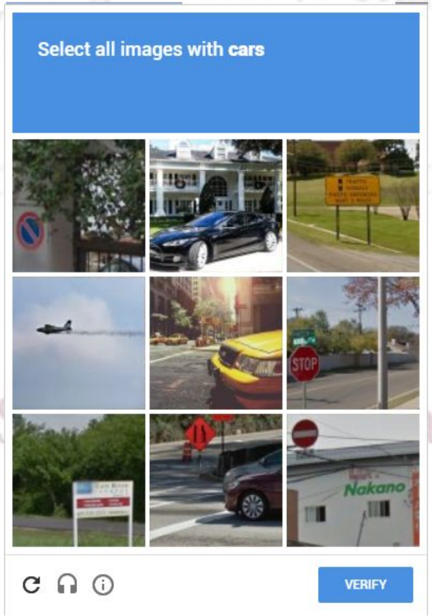
Logistic Regression

- A classification algorithm to predict categorical target labels.
- Allow us to predict a categorical label based on historical feature data.
- The categorical target column is two or more discrete class labels.

Classification algorithms predict a class or category label:

- Class 0: Car Image
- Class 1: Street Image
- Class 2: Bridge Image

You are helping Google label class data!



Keep in mind, any continuous target can be converted into categories through discretization.

- Class 0: House Price \$0-100k
- Class 1: House Price \$100k-200k
- Class 2: House Price >\$200k

- Classification algorithms also often produce a probability prediction of belonging to a class:
    - Class 0: 10% Probability
    - Class 1: 85% Probability
    - Class 2: 5% Probability
- Model reports back prediction of Class 1: image is a street.

- Also note our prediction ŷ will be a category, meaning we won’t be able to calculate a difference based on **y-ŷ.**
    - **Car Image - Street Image** does not make sense.
- We will need to discover a completely different set of error metrics and performance evaluation!

An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:

Let’s get started!

and
Ritvij Bharat Private Limited (RBPL)

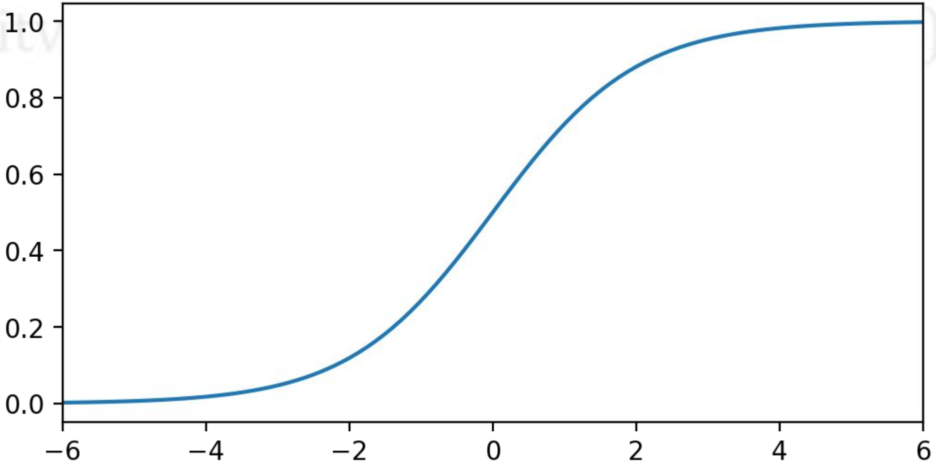
Presented by:
Shreyas Shukla

The Logistic Function

Logistic Regression transforms a Linear Regression into a classification model by using logistic function:

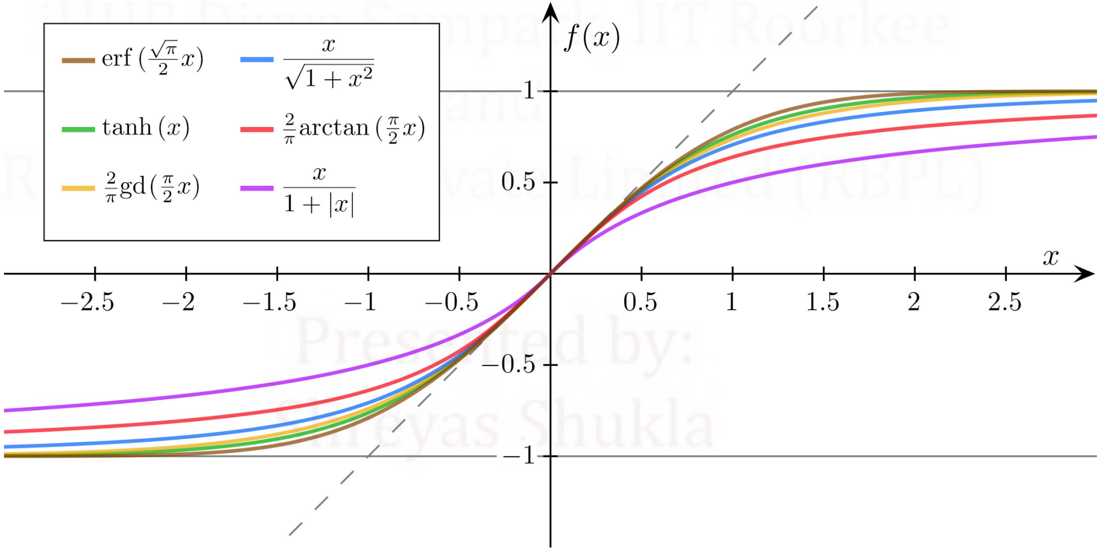
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Why logistic function versus a logarithmic function needed?

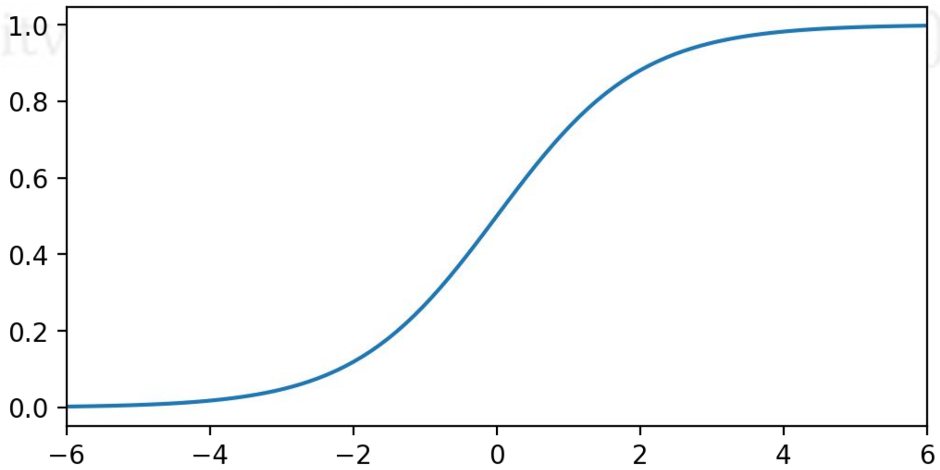


$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

There is a “family” of logistic functions.



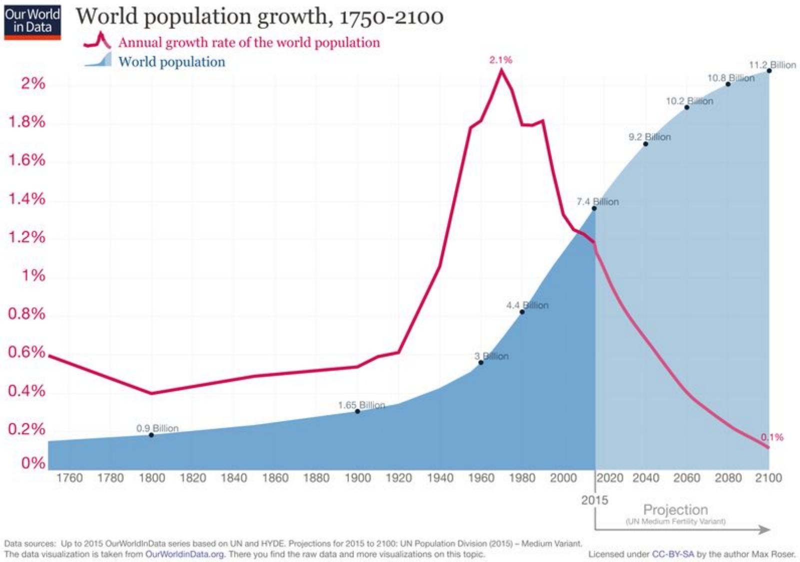
- “leveling off” behavior of the curve.
- Notice that **any** value of **x** will have an output range between 0 and 1.
- Many natural real world systems have a “carrying capacity” or a natural limiting factor.



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

An Introduction to Machine Learning with Python Programming

11 Sep 2023 - 20 Oct 2023



Shreyas Shukla

An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

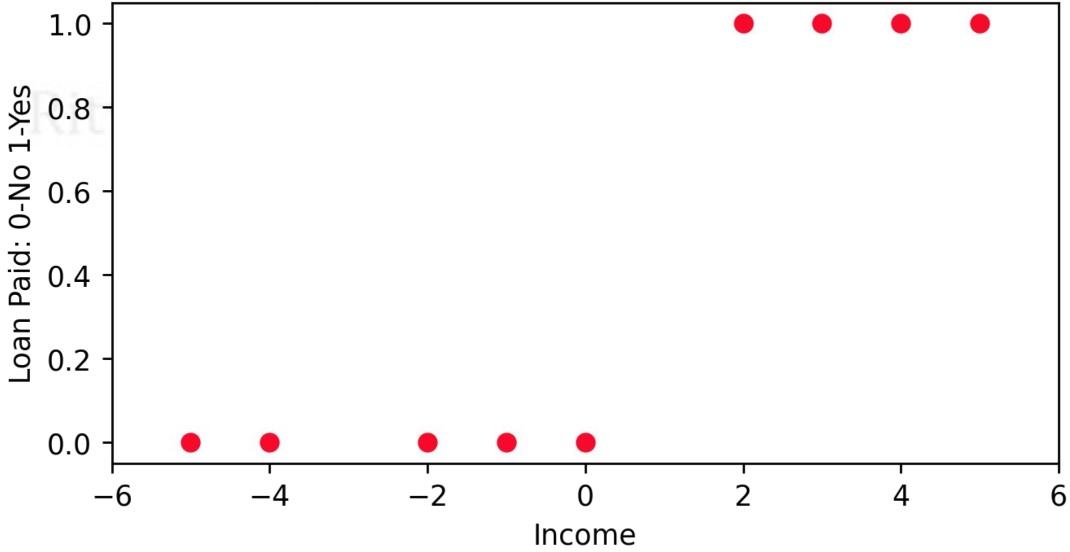
Linear to Logistic Intuition

Presented by:
Shreyas Shukla

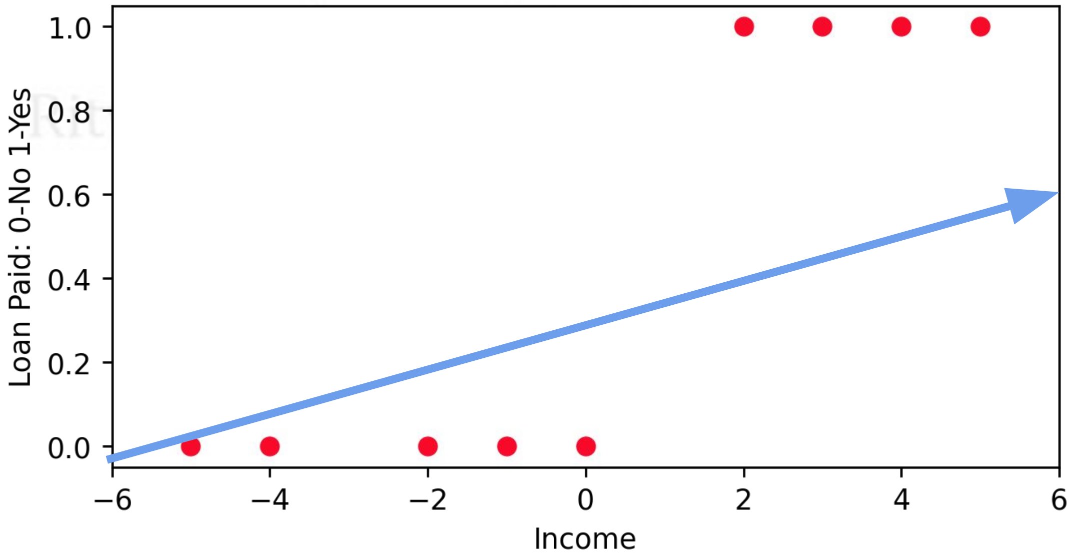
Imagine a dataset with a single feature (previous year's income) and a single target label (loan default)

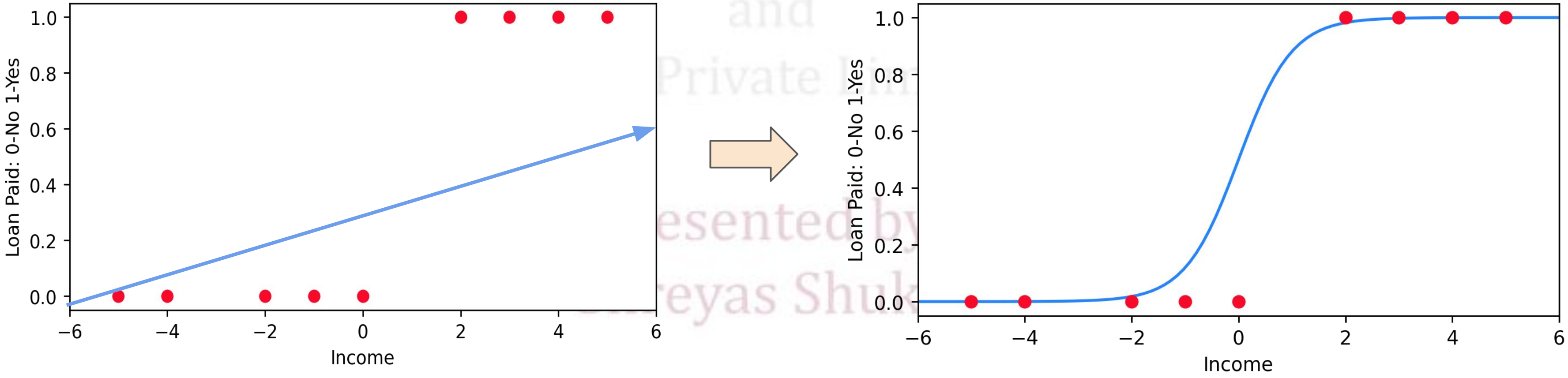
Income	Loan Paid
-5	0
-4	0
-2	0
-1	0
0	0
2	1
3	1
4	1
5	1

- Let’s plot income versus default
- People with negative income tend to default on their loans.
- What if we had to predict default status given someone’s income?

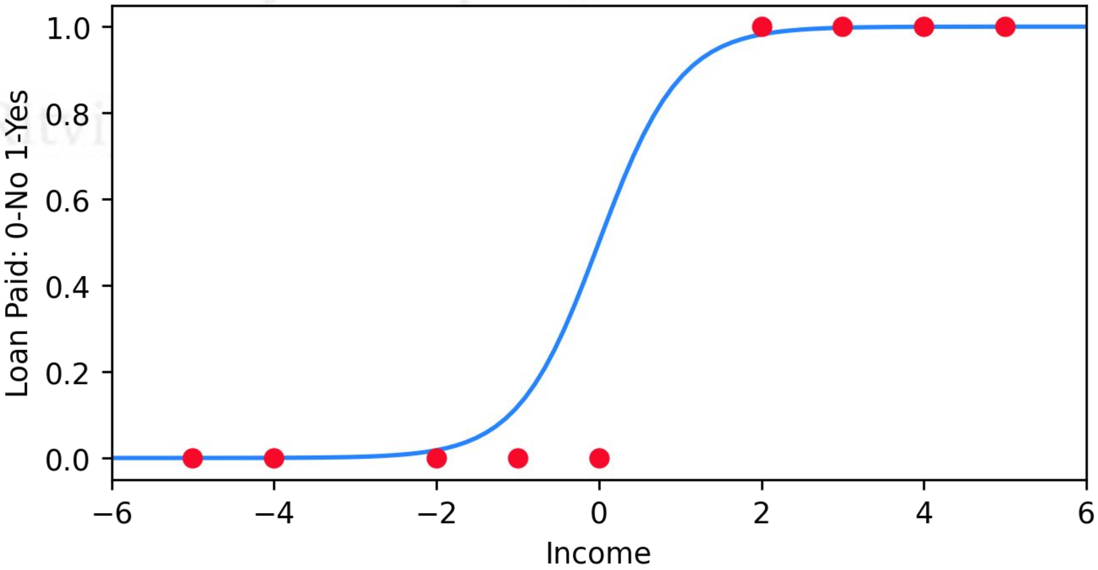


- Linear Regression would not work (recall Anscombe's quartet):
- Linear Regression easily distorted by only having 0 and 1 as possible y training values.
- Also would be unclear how to interpret predicted y values between 0 and 1.

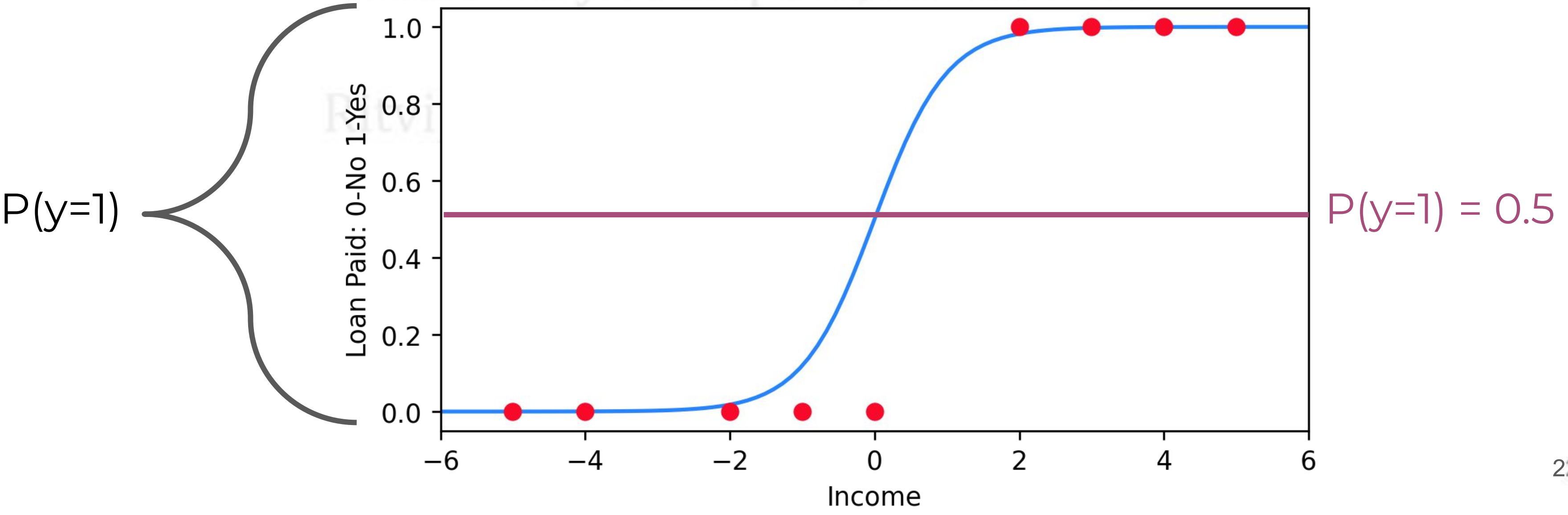



Loan Paid: 0-No 1-Yes
Income
1.0
0.8
0.6
0.4
0.2
0.0
−6
−4
−2
0
2
4
6

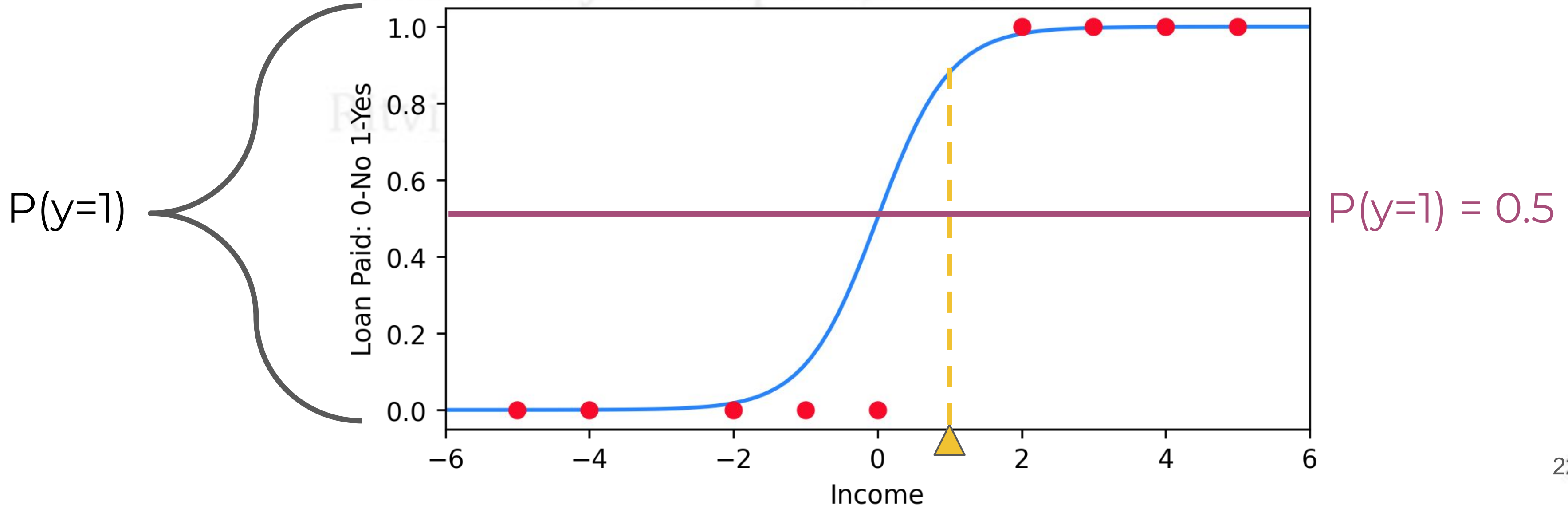
- what Logistic Regression would look like.?
- Treat the y-axis as a probability of belonging to a class:



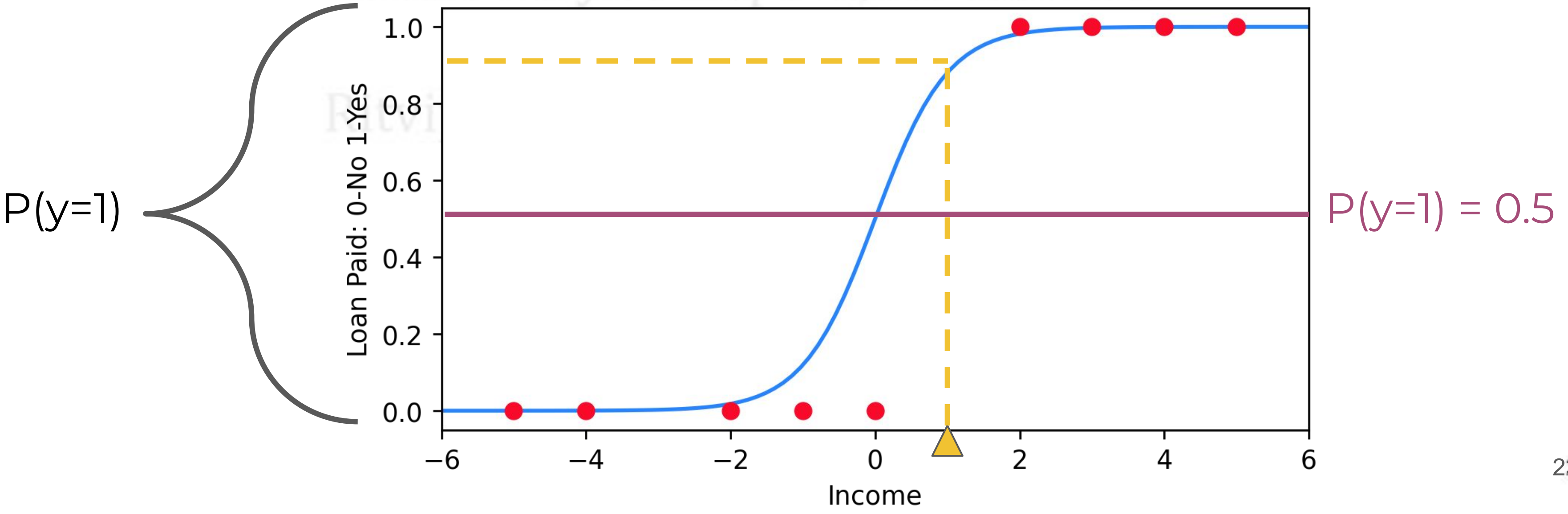
Treating P(y=1) >= 0.5 as a cut-off for classification:



For example, a new person with an income of 1:



- Predict a 90% probability of paying off loan, return prediction of Loan Paid = 1.
- But how do we actually create this Logistic Regression line?



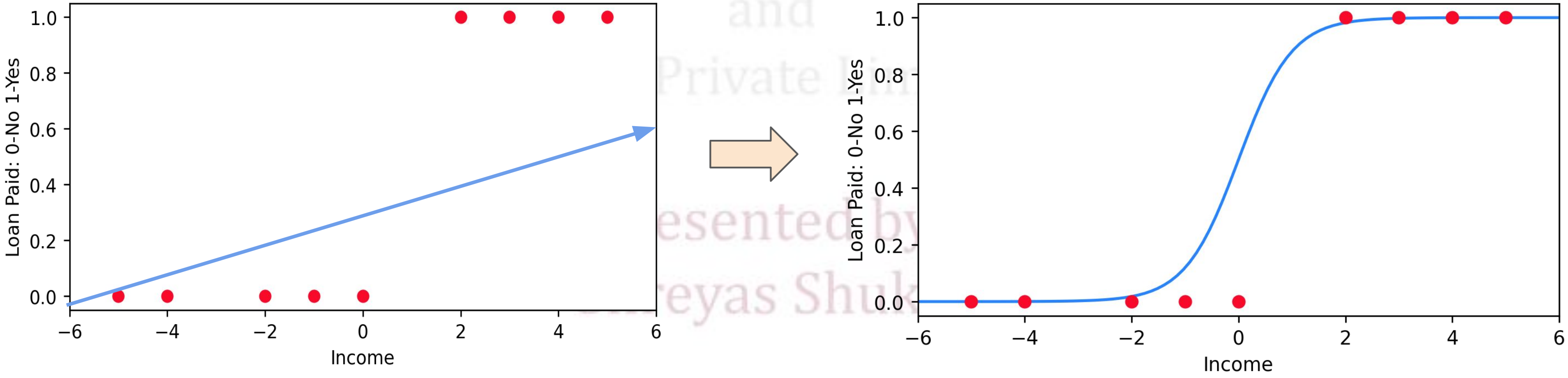
An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Linear to Logistic

Presented by:
Shreyas Shukla

Let’s go through the math of converting Linear Regression to Logistic Regression.



Linear Regression equation:

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

We also know the Logistic function transforms any input to be between 0 and 1

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

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

$$\hat{y} = \boxed{\sum_{i=0}^{n} \beta_i x_i}$$

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

In terms of the logistic function:

$$\hat{y} = \sigma(\beta_0 x_0 + \cdots + \beta_n x_n)$$

$$\hat{y} = \sigma\left(\sum_{i=0}^{n} \beta_i x_i\right)$$

- Writing it out fully

$$\hat{y} = \frac{1}{1 + e^{-\sum_{i=0}^{n} \beta_i x_i}}$$

- How do we interpret the coefficients and their relation to ŷ ?

- First understand the term **odds.**
- You may be familiar with from gambling **odds** which are often referred to in the sense of N to 1.
- But where does this actually come from?

The odds of an event with probability **p** is defined as the chance of the event happening divided by the chance of the event not happening:

$$\frac{p}{1-p}$$

- We can rearrange it to show that it is equivalent to modelling the log of the odds as a linear combination of the features.
- This will allow us to solve for the coefficients and feature x in terms of **log odds.**

$$\hat{y} = \frac{1}{1 + e^{-\sum_{i=0}^{n} \beta_i x_i}}$$

An Introduction to Machine Learning with Python Programming

11 Sep 2023 - 20 Oct 2023

C

iHUB Divya

Ritvij Bharat

$$\hat{y} = \frac{1}{1 + e^{-\sum_{i=0}^{n} \beta_i x_i}}$$

$$\hat{y} + \hat{y} e^{-\sum_{i=0}^{n} \beta_i x_i} = 1$$

by:

Shreyas Shukla

Solving for **log odds:**

$$\hat{y} + \hat{y}e^{-\sum_{i=0}^{n}\beta_i x_i} = 1$$

$$\hat{y}e^{-\sum_{i=0}^{n}\beta_i x_i} = 1 - \hat{y}$$

Solving for **log odds:**

$$\hat{y} + \hat{y}e^{-\sum_{i=0}^{n}\beta_i x_i} = 1$$

$$\hat{y}e^{-\sum_{i=0}^{n}\beta_i x_i} = 1 - \hat{y}$$

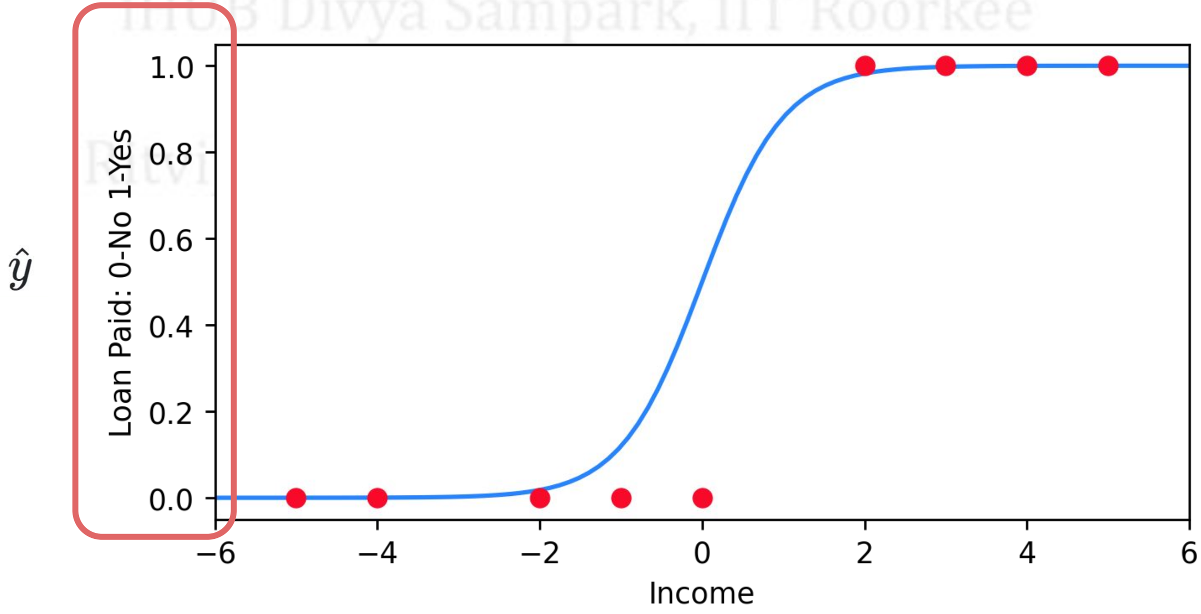
$$\frac{\hat{y}}{1-\hat{y}} = e^{\sum_{i=0}^{n}\beta_i x_i}$$

Solving for **log odds:**

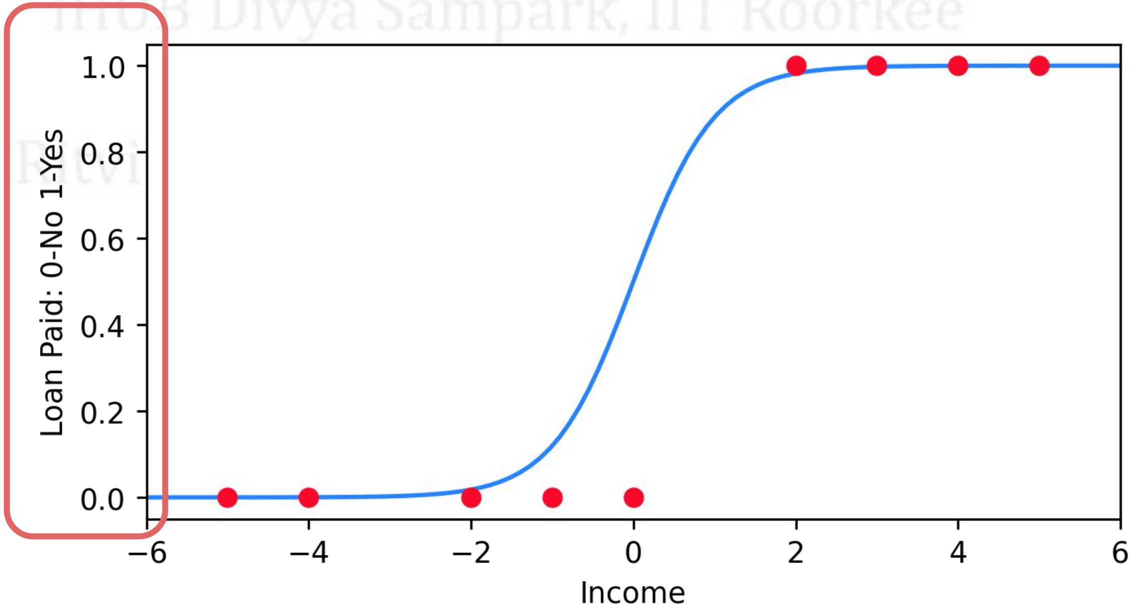
$$\frac{\hat{y}}{1-\hat{y}} = e^{\sum_{i=0}^{n} \beta_i x_i}$$

$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

How would curve look like in terms of log odds?

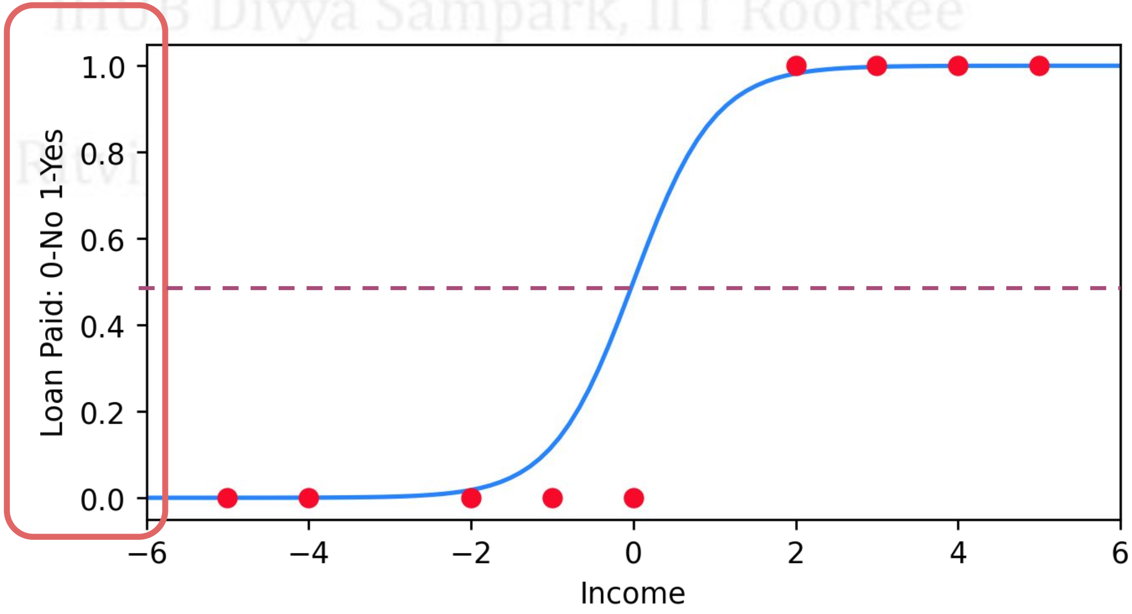


$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) \Leftarrow \hat{y}$$



For p=0.5

$$ln(\frac{0.5}{1-0.5}) = 0$$



For p=0.5, halfway point now at 0.

$$ln(\frac{0.5}{1-0.5}) = 0$$

As p ⟶ 1 then log odds becomes ∞

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

Class points now at infinity

$$\lim_{p \to 1} ln\left(\frac{p}{1-p}\right) = \infty$$

$$ln\left(\frac{0.5}{1-0.5}\right) = 0$$

$$\lim_{p \to 0} ln\left(\frac{p}{1-p}\right) = -\infty$$

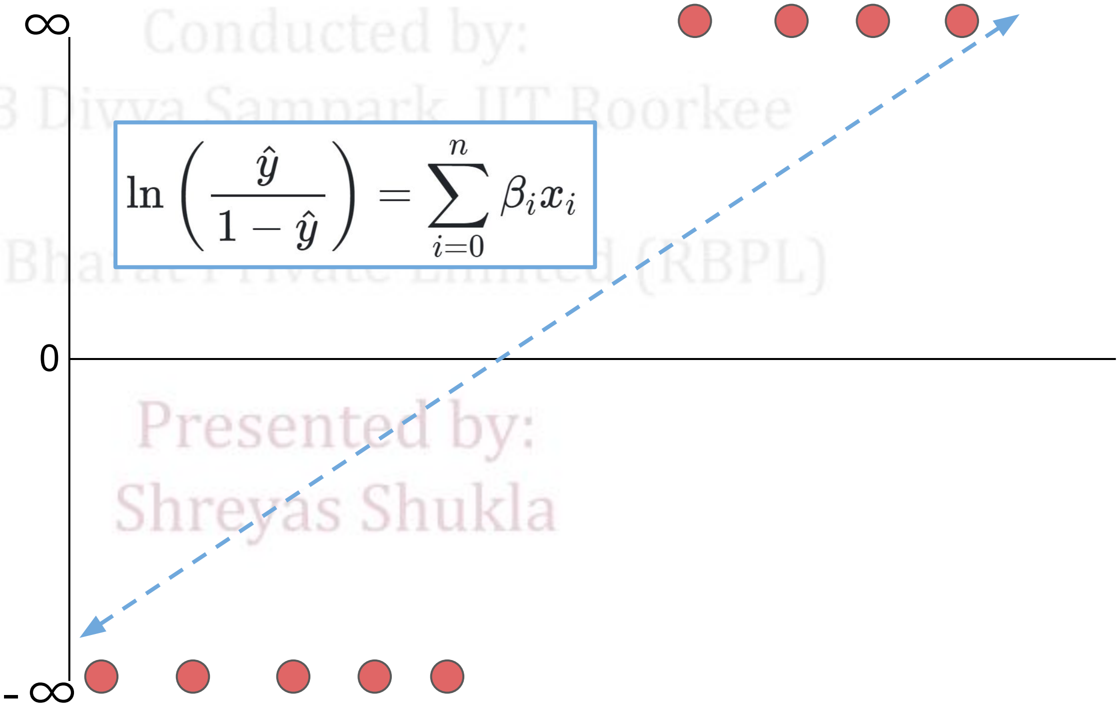
On log scale, logistic function is straight line

Coefficients in terms of change in log odds.

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$

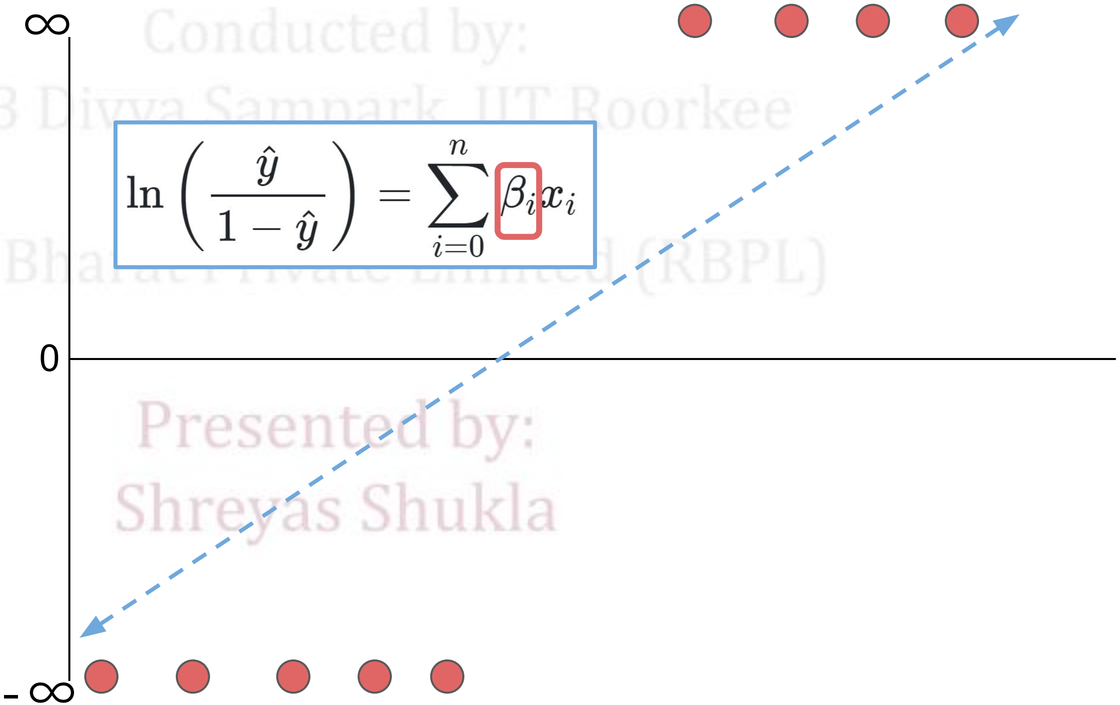


Is β simple to interpret? Not really...

$$\lim_{p \to 1} ln(\frac{p}{1-p}) = \infty$$

$$ln(\frac{0.5}{1-0.5}) = 0$$

$$\lim_{p \to 0} ln(\frac{p}{1-p}) = -\infty$$



Since the log odds scale is nonlinear, a **β** value can not be directly linked to “one unit increase” as it could in Linear Regression.

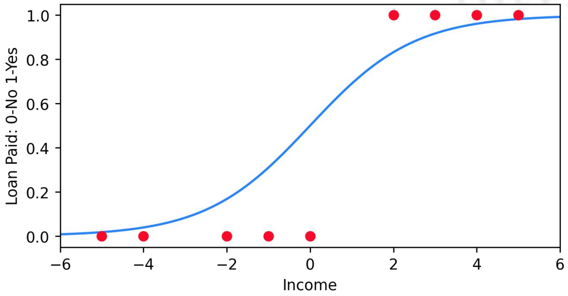
But there are some straightforward insights we can gain.

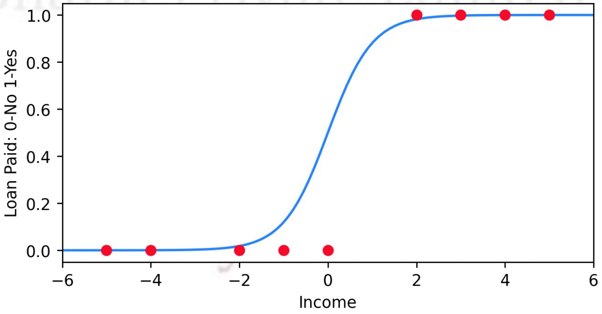
$$\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i$$

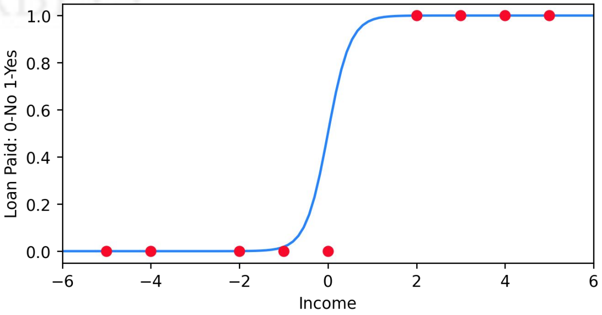
- Sign of Coefficient
    - Positive **β** indicates more likelihood of belonging to 1 class with increase in associated **x** feature.
    - Negative **β** indicates an decrease in likelihood of belonging to 1 class with increase in associated **x** feature.

- Magnitude of Coefficient
  - Harder to directly interpret magnitude of **β** directly, especially in discrete and continuous x feature values.
  - But we can use **odds ratio**, essentially comparing magnitudes against each other.
  - Comparing magnitudes of coefficients against each other can lead to insight over which features have the strongest effect on prediction output.

How we actually fit this curve?

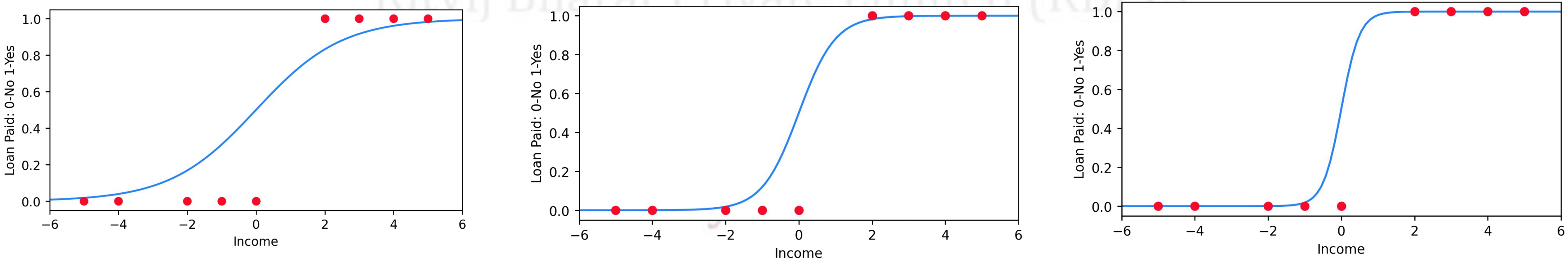




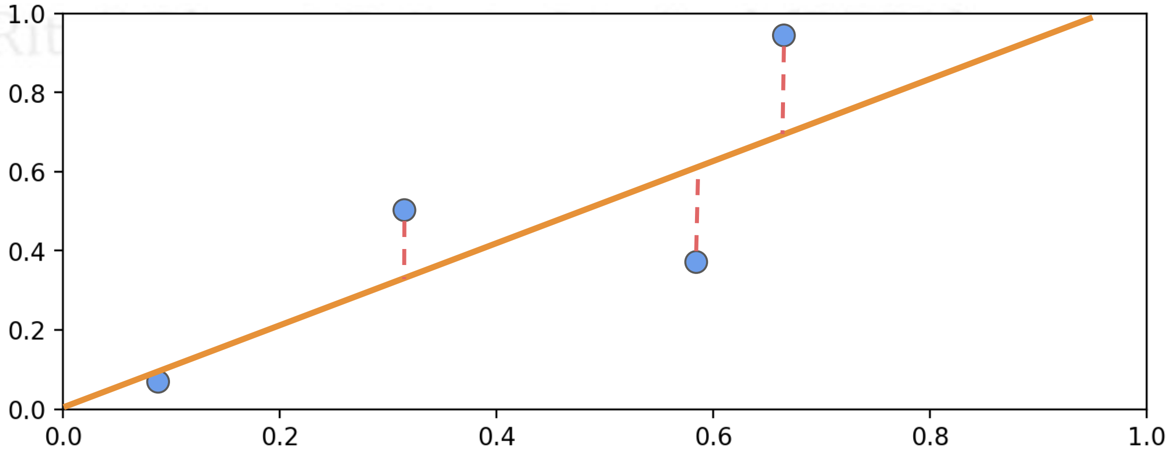


Finding the Best Fit

Here we have three different Logistic Regression curves with different **β** values.



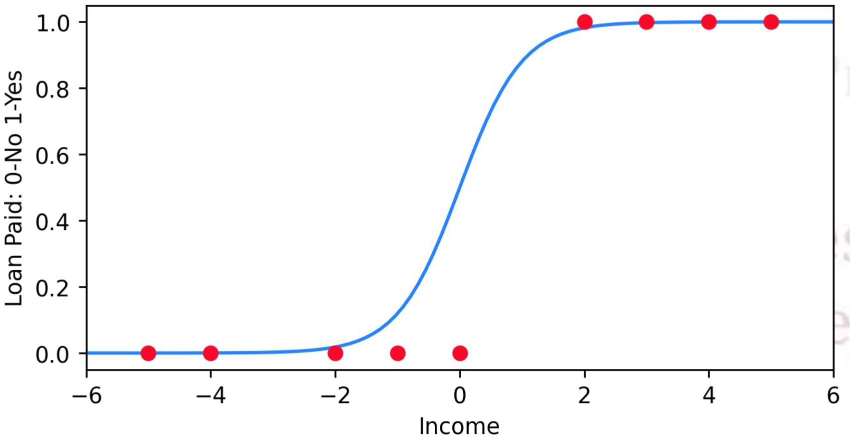
Recall that in Linear Regression we seek to minimize the Residual Sum of Squares (RSS).

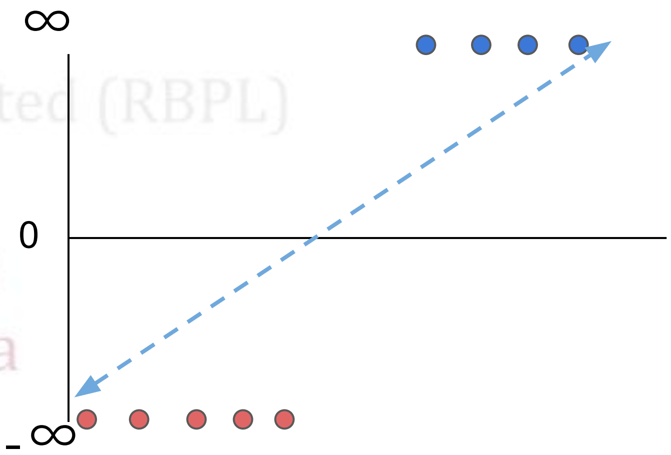


Unfortunately, even in log odds, targets are at infinity, making RSS unfeasible.

$$\boxed{\ln\left(\frac{\hat{y}}{1-\hat{y}}\right) = \sum_{i=0}^{n} \beta_i x_i}$$

The first step for maximum likelihood is to go from log odds back to probability.





$$\ln\left(\frac{p}{1-p}\right) = \ln(odds)$$

$$\ln\left(\frac{p}{1-p}\right) = \ln(odds)$$

$$\frac{p}{1-p} = e^{\ln(odds)}$$

$$p = (1-p)e^{\ln(odds)}$$

$$p = (1 - p)e^{\ln(odds)}$$

$$p = e^{\ln(odds)} - pe^{\ln(odds)}$$

$$p + pe^{\ln(odds)} = e^{\ln(odds)}$$

$$p(1 + e^{\ln(odds)}) = e^{\ln(odds)}$$

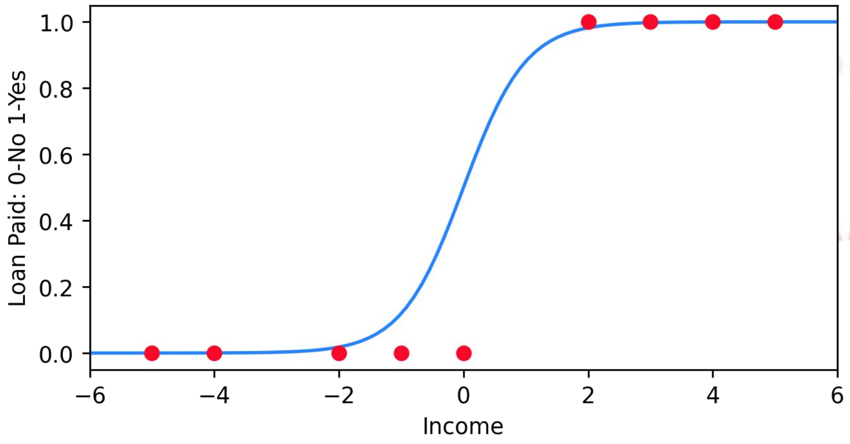
$$p = e^{\ln(odds)} - pe^{\ln(odds)}$$

$$p + pe^{\ln(odds)} = e^{\ln(odds)}$$

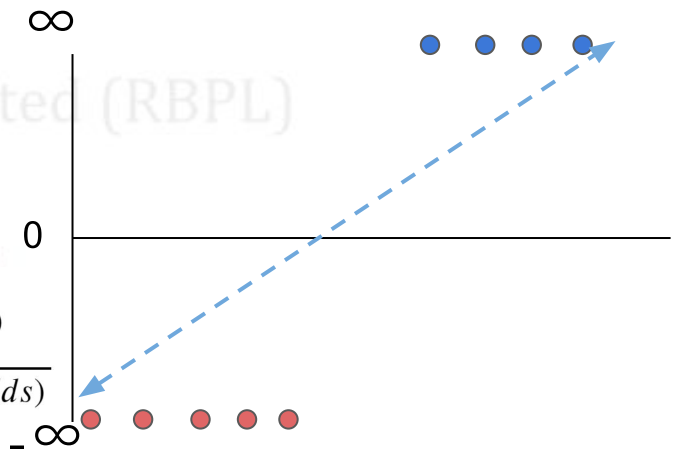
$$p(1 + e^{\ln(odds)}) = e^{\ln(odds)}$$

$$\boxed{p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}}$$

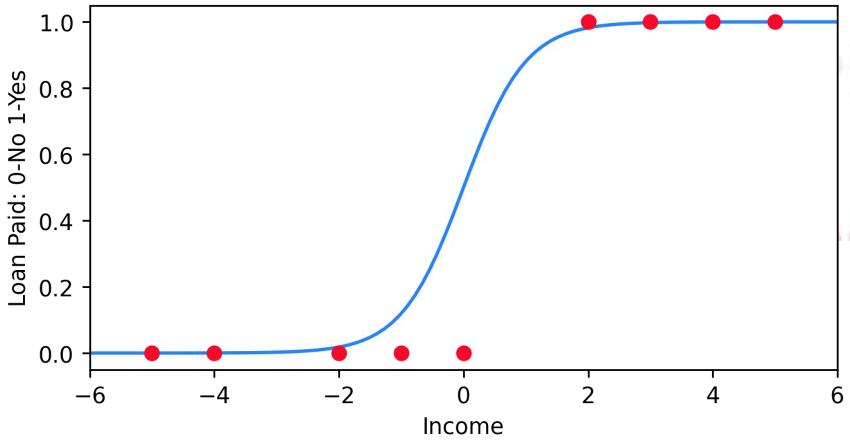
We can now convert ln(odds) into a probability.



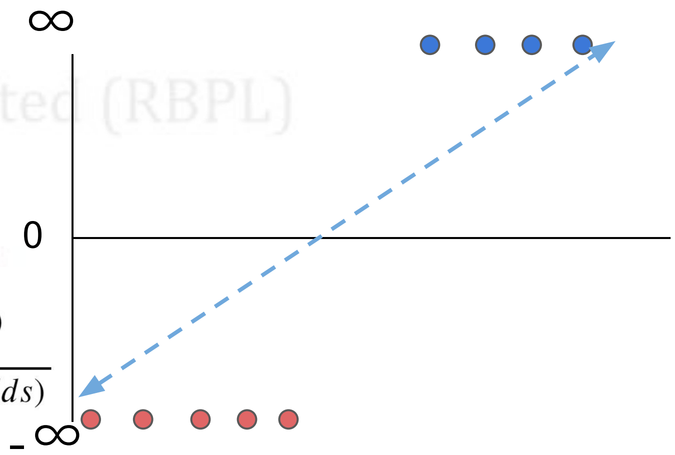
$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$



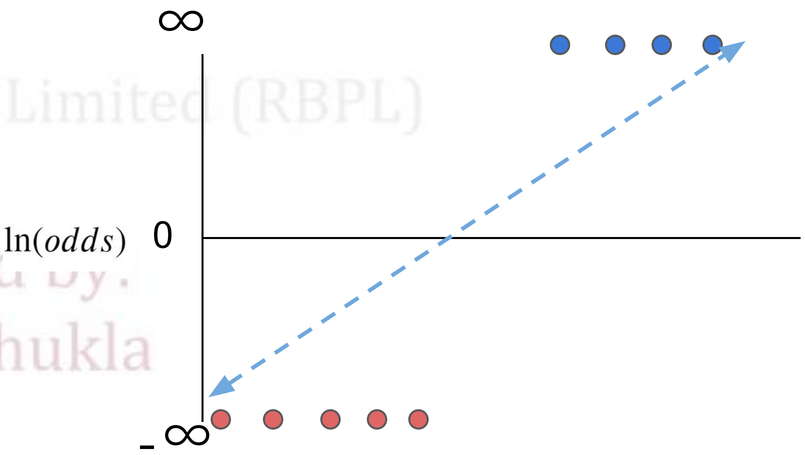
Let’s now explore the idea behind **maximum likelihood.**



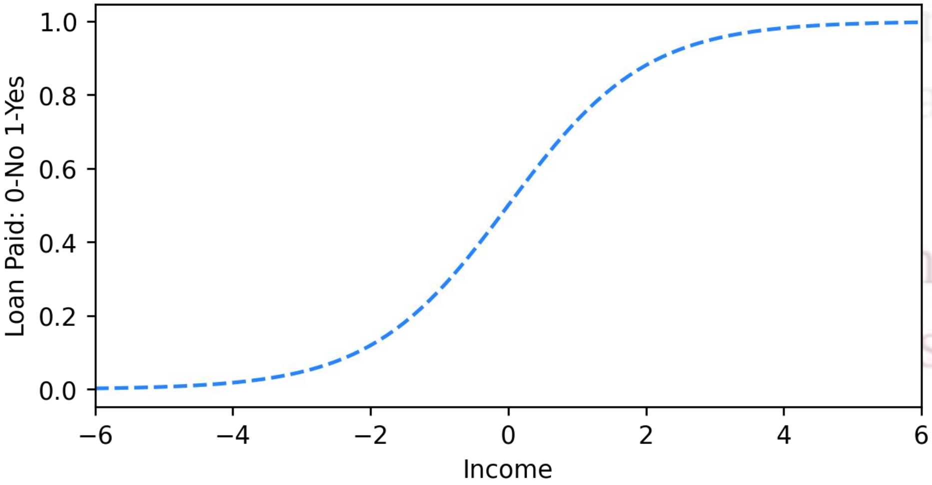
$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

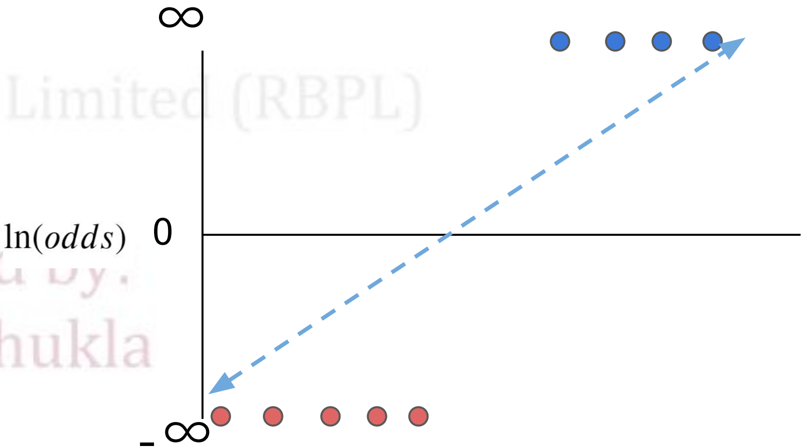


We choose a line in the log(odds) axis and project the points on to the line:

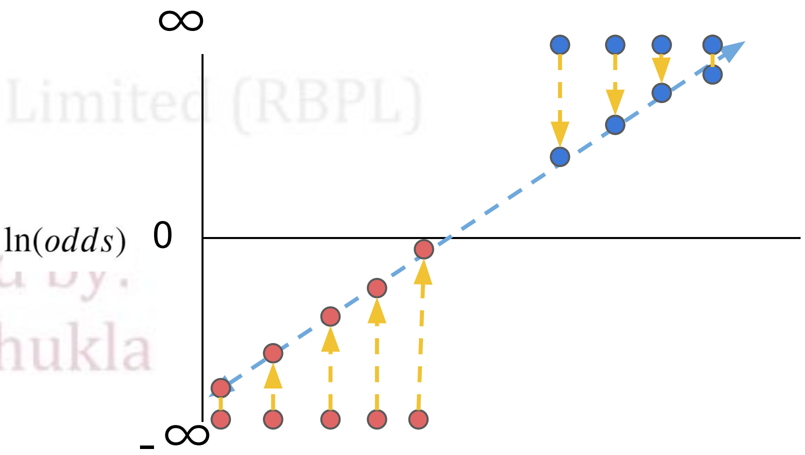


We also know this line has a form on the probability y-axis.



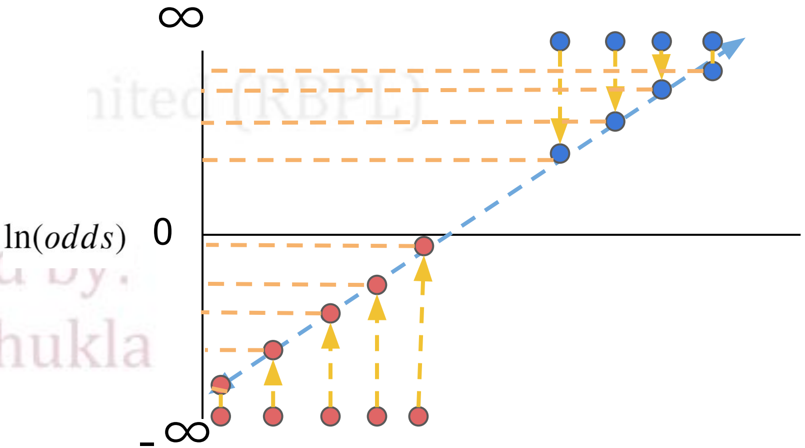


We choose a line in the log(odds) axis and project the points on to the line:

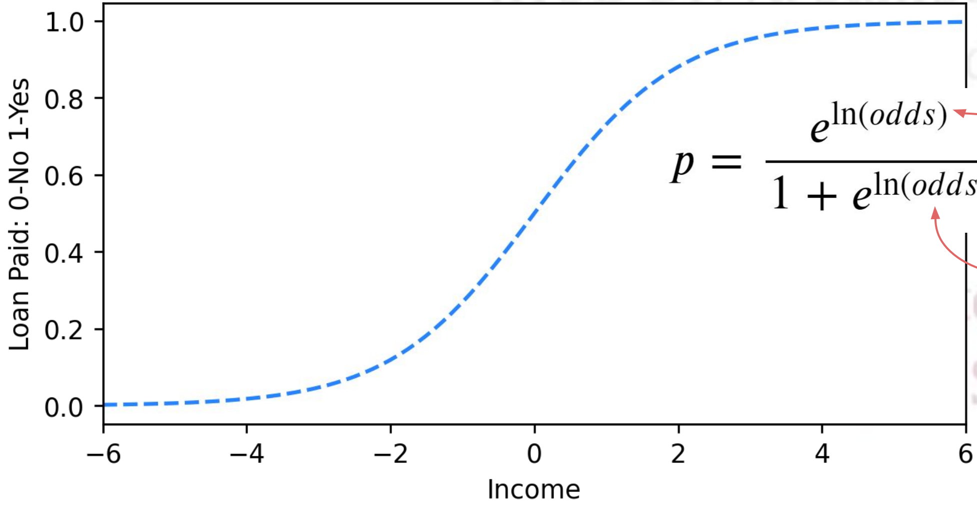


Plot these values as probabilities on the logistic regression model.

$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

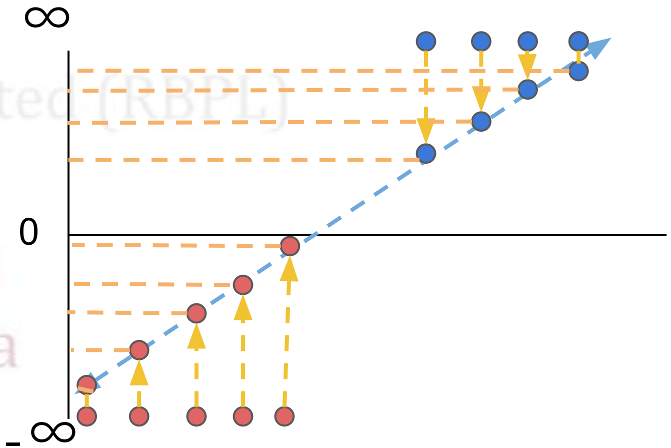


Plot these values as probabilities on the logistic regression model.

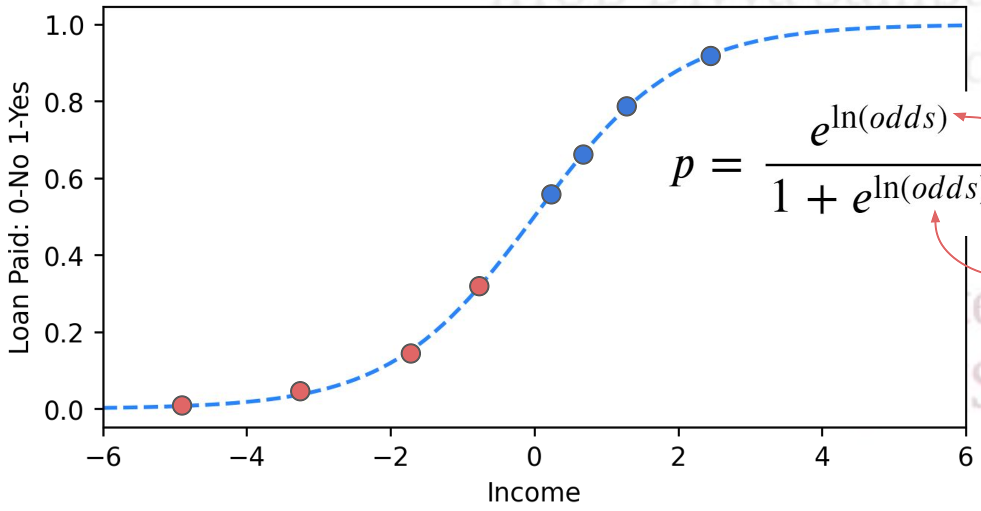


$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

$\ln(odds)$

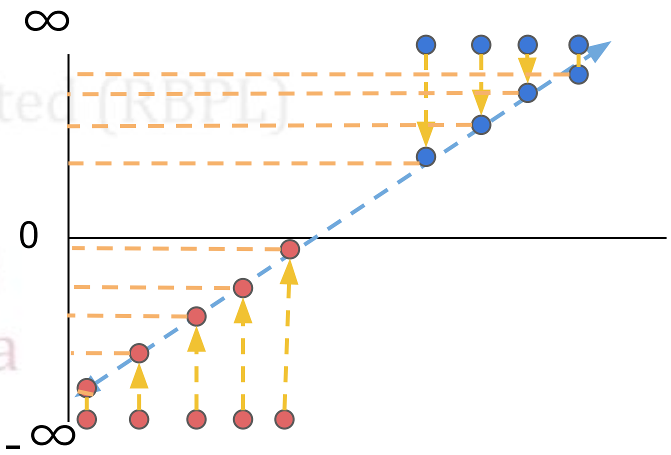


We now measure the likelihood of these probabilities.

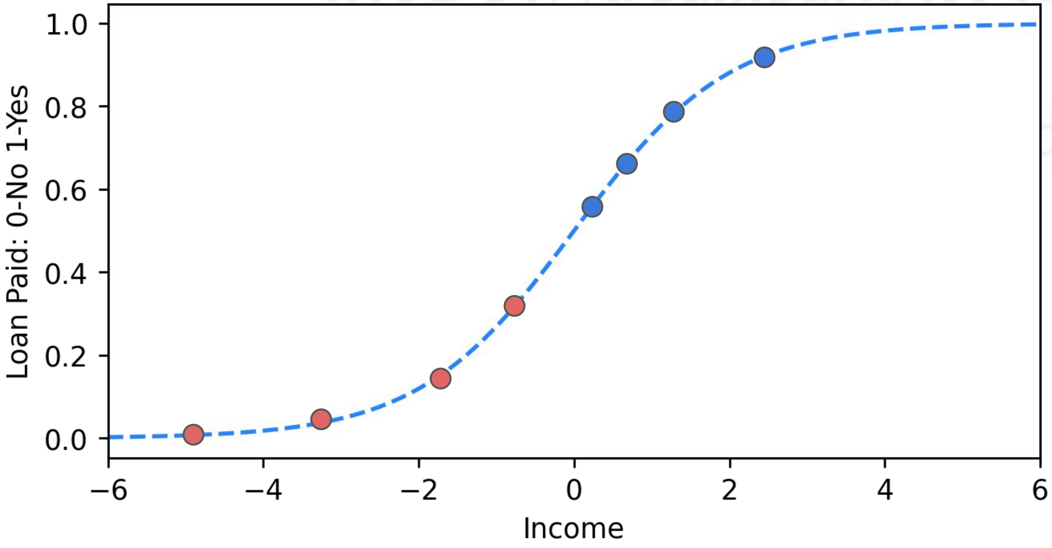


$$p = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}$$

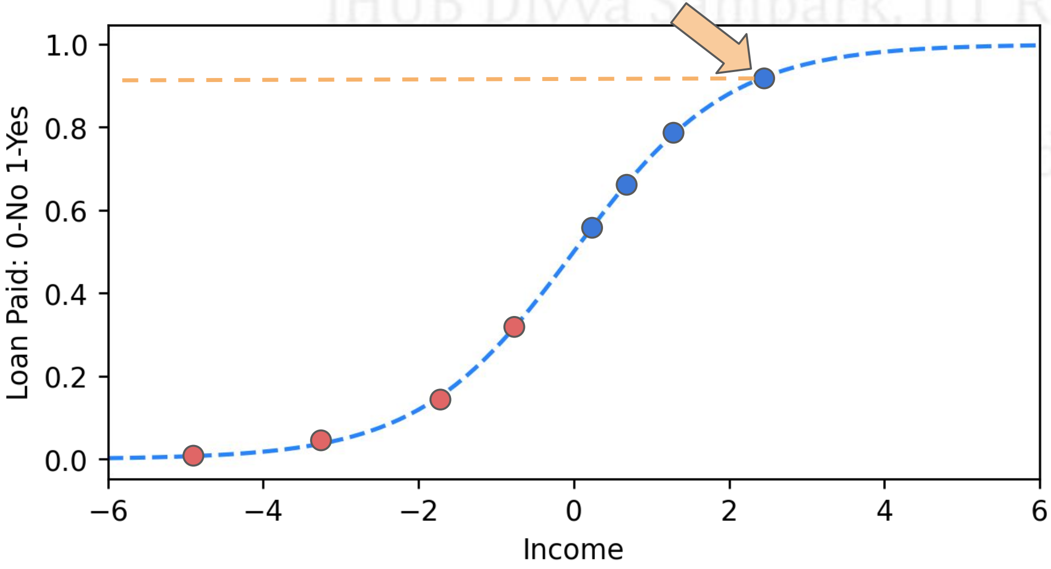
$\ln(odds)$



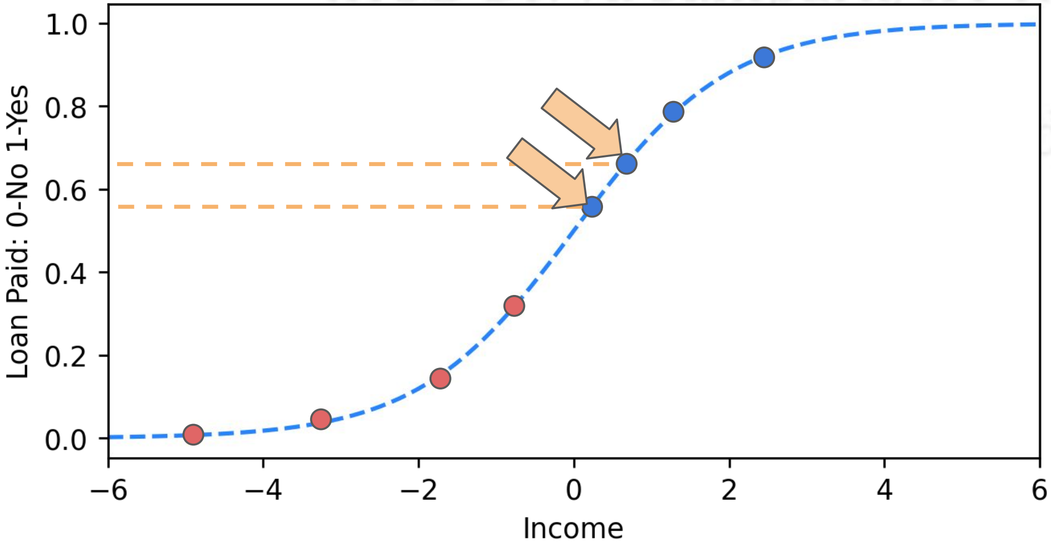
Likelihood = Product of probabilities of belonging to class 1



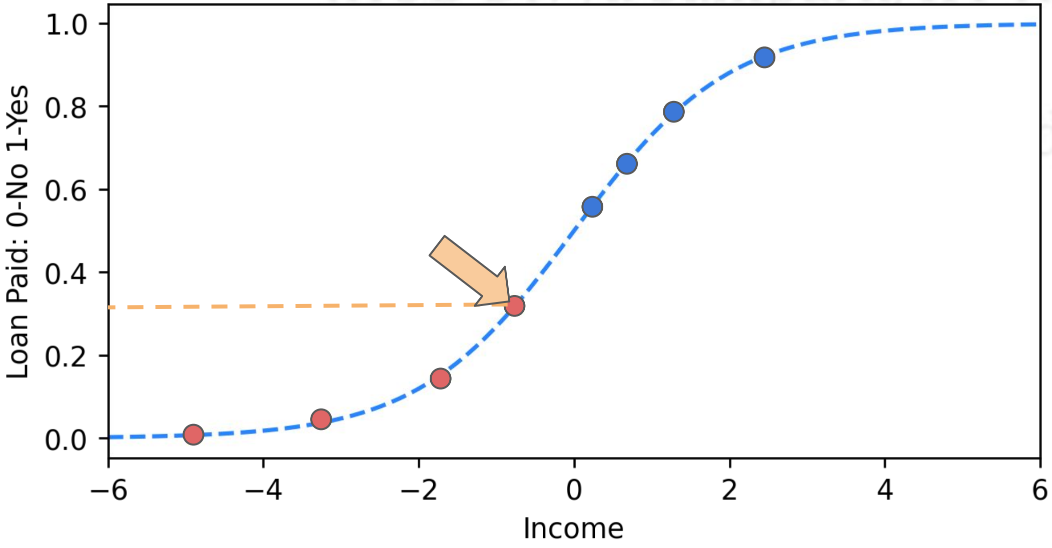
Likelihood = 0.9



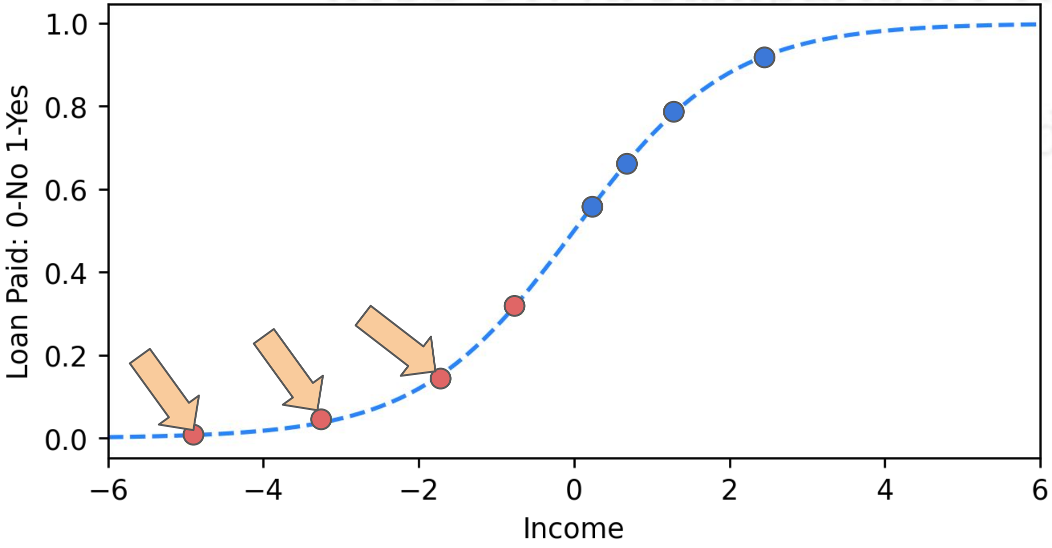
$$\text{Likelihood} = 0.9 \times 0.8 \times 0.65 \times 0.52 \times$$



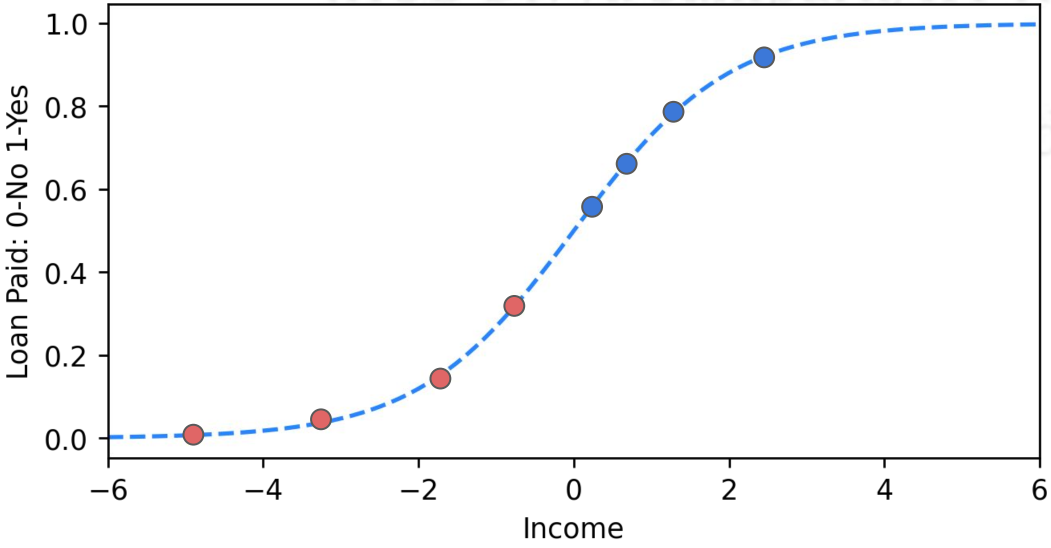
Likelihood = 0.9 × 0.8 × 0.65 × 0.55 × (1-p) ×



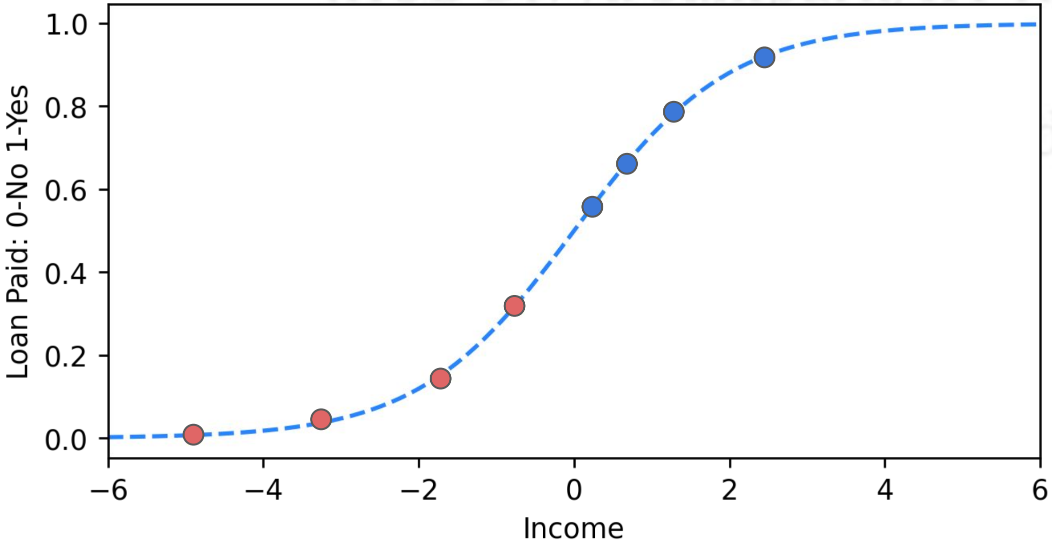
Likelihood = 0.9 × 0.8 × 0.65 × 0.55 × (1-0.3) × (1-0.2) × (1-0.08) × (1-0.02)



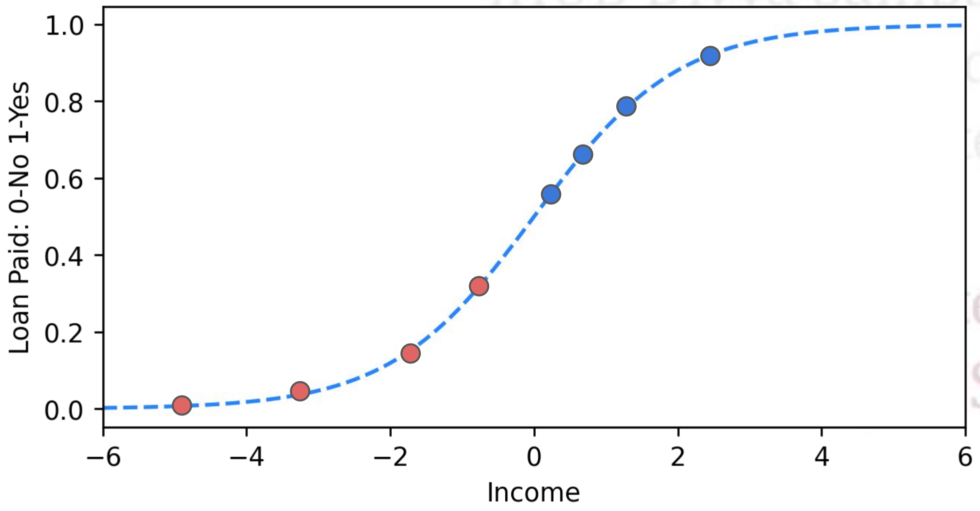
Likelihood = 0.129

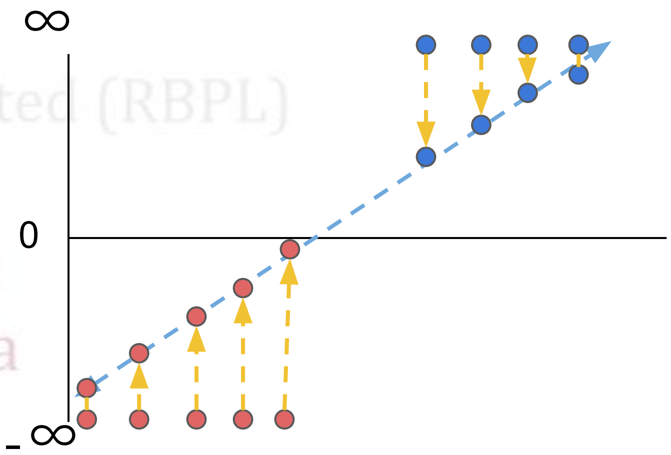


Note in practice we actually maximize the **log** of the likelihoods. (e.g. ln(0.9)×ln(0.8)×…)

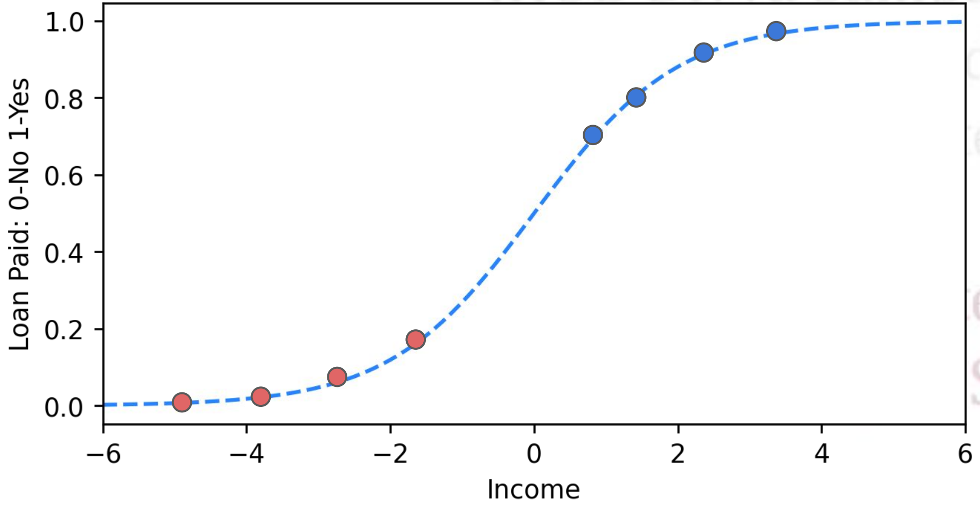


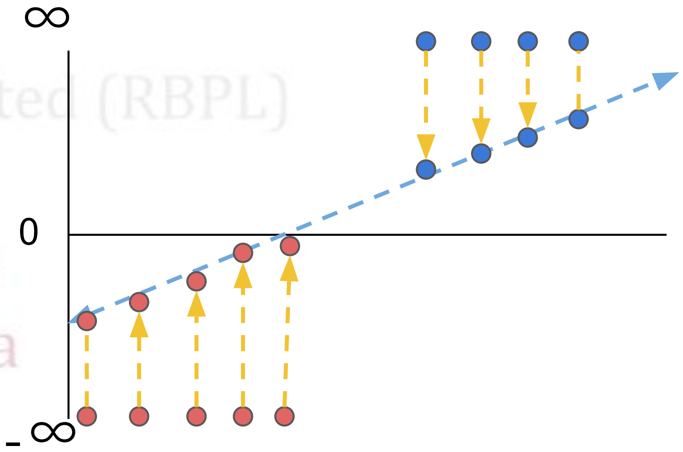
There is some set of coefficients that will maximize these log likelihoods.





Choose best coefficient values in log odds terms that creates maximum likelihood.





An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Let’s explore Logistic Regression with Python!

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Presented by:
Shreyas Shukla