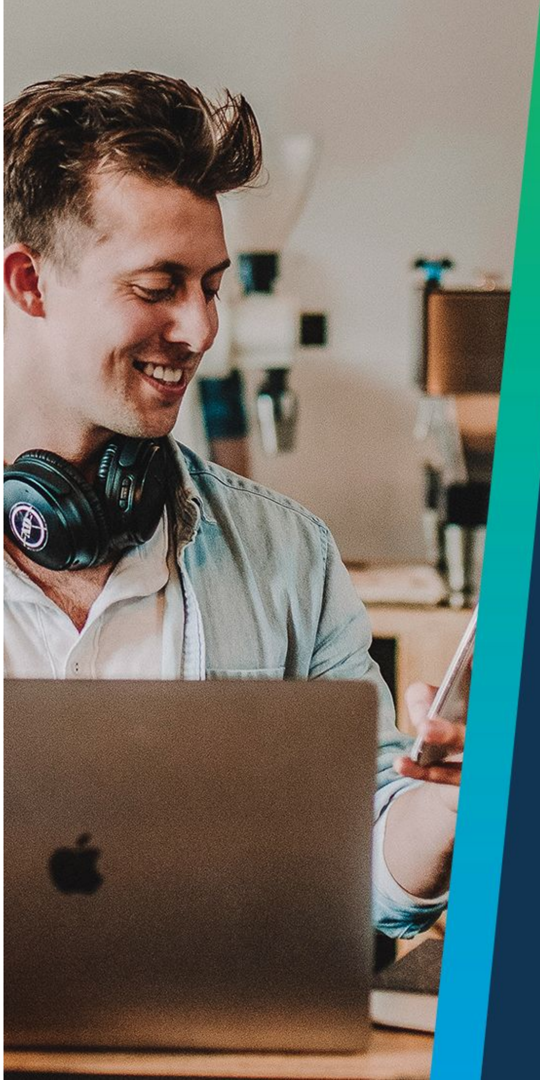


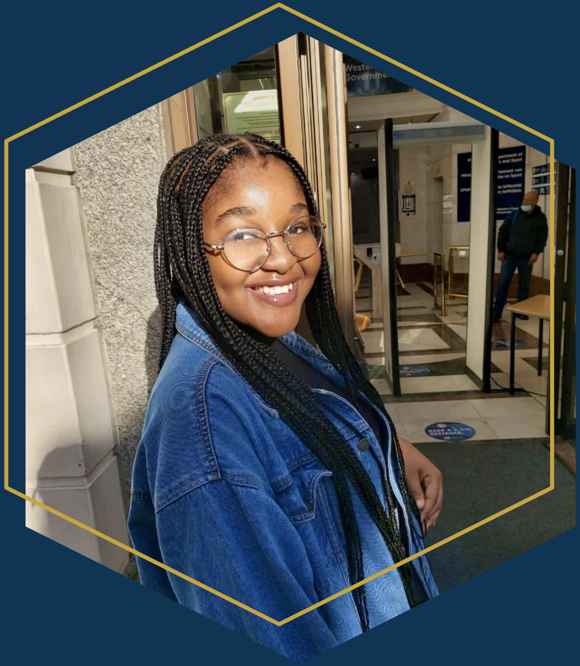
Data Science
Bootcamp

Hyperiondev

Logistic Regression & Model Evaluation

Welcome

Your Lecturer for this session



Sanana Mwanawina

Lecture - Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- No question is daft or silly - **ask them!**
- There are Q/A sessions midway and at the end of the session, should you wish to ask any follow-up questions.
- You can also submit questions here: hyperiondev.com/sbc4-ds-questions
- For all non-academic questions, please submit a query: hyperiondev.com/support
- Report a safeguarding incident: hyperiondev.com/safeguardreporting
- We would love your feedback on lectures: https://hyperionde.wufoo.com/forms/zsgv4m40ui4i0g/

Lecture – Code Repo

Go to: github.com/HyperionDevBootcamps

Then click on the “**C4_DS_lecture_examples**” repository, do view or download the code.

Objectives

- Understanding error
- Evaluating our models
- Learn about what Logistic regression is

Regression Model

$$Y = \text{Sales} = 2.939 + 0.046\ TV + 0.189\ Radio + (-0.001)\ Billboard$$

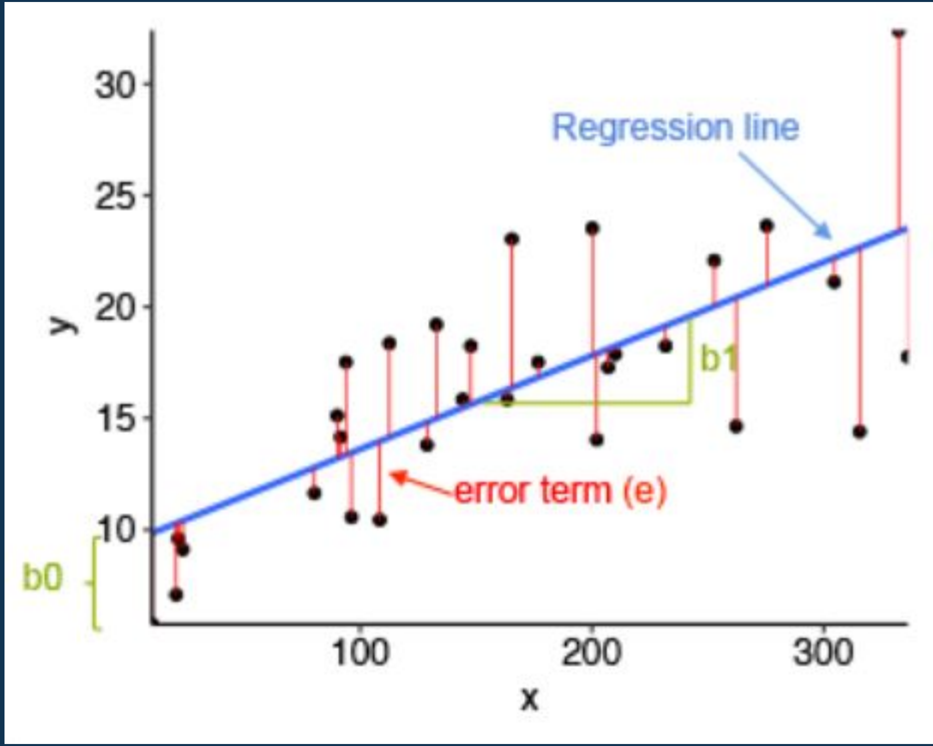
- This is an approximate model i.e., $Y \approx \beta 0 + \beta 1 Xi$
  So really what we obtain is the approximate number of Sales

$$\text{Sales} \approx 2.939 + 0.046\ TV + 0.189\ Radio + (-0.001)\ Billboard$$

or $$\hat{Y} = 2.939 + 0.046\ TV + 0.189\ Radio + (-0.001)\ Billboard$$

- A minimised difference is still a difference: after linear regression, there is still some difference between observed values for Y, and values for Y predicted by f. If your data, when plotted, does not seem to fall along a straight line, linear regression is not the right model for the problem. But even if it does, the straight line is only a model of the data, hence the use of the symbol "≈".

Regression Model



✓ $\hat{y} = mx + b$

✓ $y = mx + b + error$

error = actual - predicted
error = y - (mx + b)
y= mx + b + error

How we make predictions

★ Run your regression in Python

SIMPLE LINEAR REGRESSION: Sales = $\beta 0 + \beta 1$ TV Budget

```
[0.04753664]
7.032593549127695
```

$\hat{Y} = 7.0326 + 0.0475$ TV Budget $= 7.0326 + 0.0475(100) = 11.783$

Or use:

```
reg.predict(np.array([[100]]))
```

To get the same result.

How we make predictions

★ Run your regression in Python

MULTIPLE LINEAR REGRESSION:

```
Coefficients:  [ 0.04576465  0.18853002 -0.00103749]
Intercept:  2.938893694594085
```

$\hat{Y}$ = 2.939 + 0.046 TV + 0.189 Radio + (-0.001) Billboard
$\hat{Y}$ = 2.939 + 0.046(100) = 7.539 which is 7539 units

Or use the following to get the same result:

```
reg.predict(np.array([[100,0,0]]))
```

Evaluating our models

★ We need to split our dataset into training and testing data
★ We can interpret a useful statistical measure called R-squared

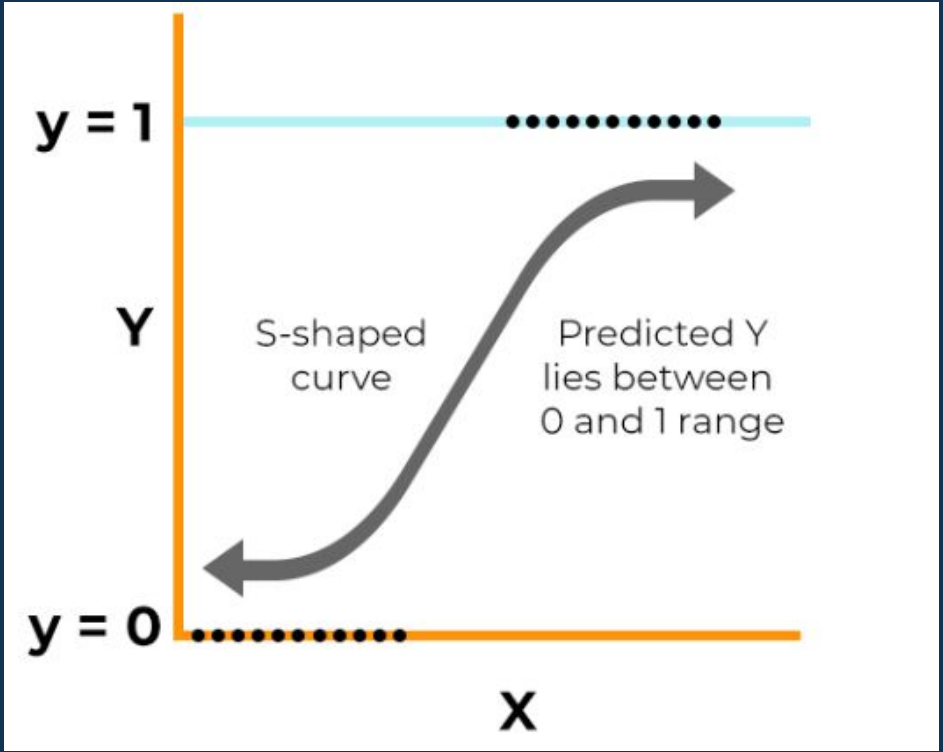
Classification Problems

- ★ When linear regression models were discussed previously, we assumed that our dependent variables **Y** is a **continuous numerical variable**. However it very common in machine learning problems instead to be dealing with **categorical variables.** These variables take on distinct non-continuous values which will correspond to a specific set of categories.
- ★ Predicting categorical variables is called **classification.** Classification problems are very common, perhaps even more so than for problems suited for regression.

Logistic Regression

- One approach to classification is logistic regression, which is a common way to do binary logistic regression, which is classifying into two categories. It works by using the logistic function, also known as the sigmoid function. This is an S-shaped curve that maps input values to x output values y.
- Logistic regression is similar to linear regression, however the output is not continuous along a line, but a value between 0 and 1.
- That value can then be interpreted as the probability of that the instance belongs to a certain category.

Logistic Regression



$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any

Hyperiondev

Thank you for joining us