# Logistic Regression

.Ratnakar
Fangdi Liu
Jiasong Liu

Instructor – Dr. Yasser Alginahi

October 6th, 2023
University of Windsor



University of Windsor

# Overview

- What is Logistic Regression?
- Why do we use Logistic Regression ?
- How does the algorithm work ?
- Some examples

# What is Logistic Regression?

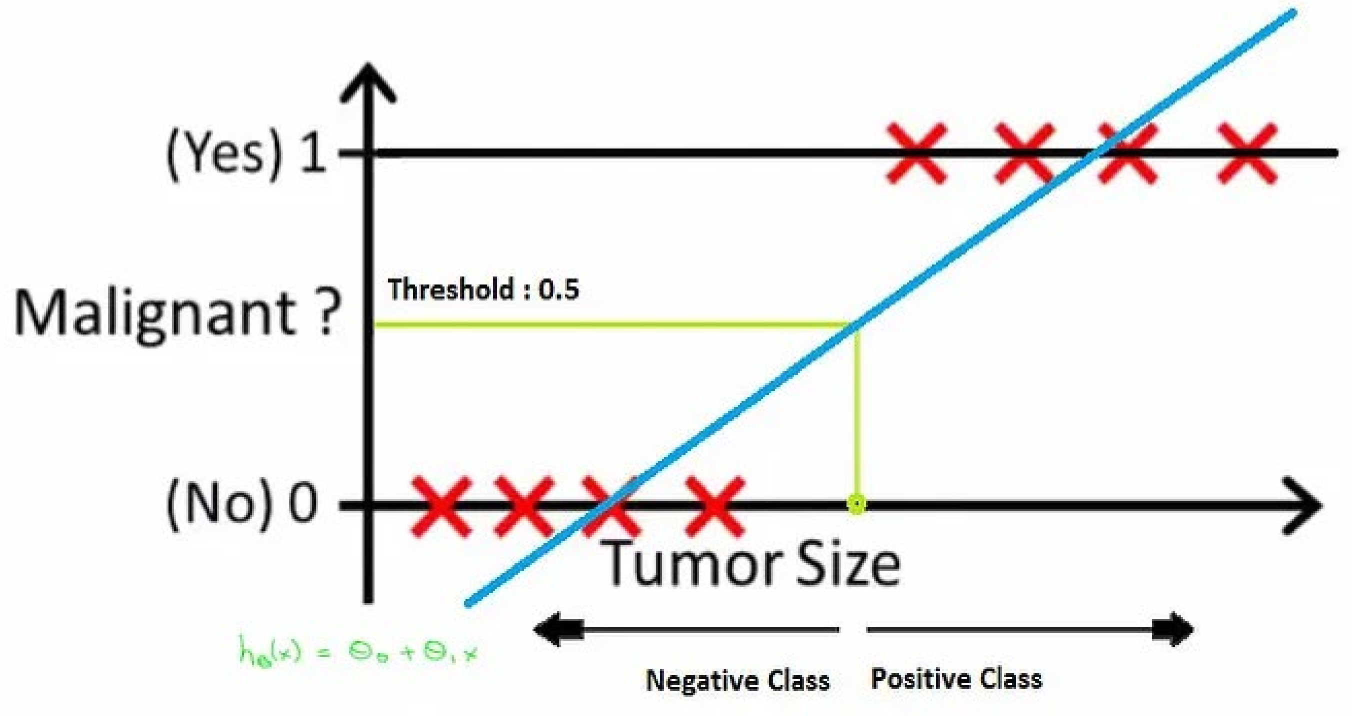
- Logistic regression is a statistical method used for predicting the probability of an event happening. It's often used in situations where the outcome you're interested in is binary, which means it has only two possible values.
- We want to learn about Logistic Regression as a method for Classification.
- Some examples are:
  - Spam or Ham email ?
  - Success or Failure ?

# Why use Logistic Regression ?

- What's wrong with Linear Regression ?
  - Linear regression's output is sensitive to outliers.
  - Need of an algorithm that absorbs the effects of outliers without impacting the final output.

# Why use Logistic Regression ?

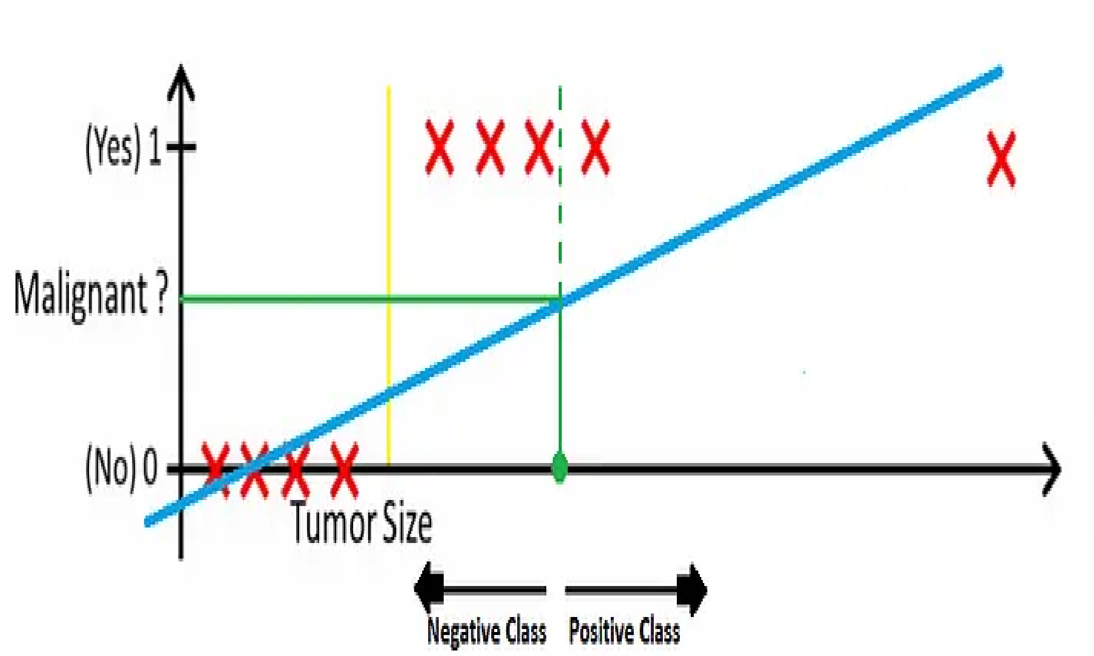
- The example uses linear regression to classify if a tumor is Malignant or not.
- Outcomes in linear regression can be greater than 1 and less than 0.
- Best Fit Line (BFL) extends above 1 and below 0.




[Source: https://tinyurl.com/ydzyhyt2]

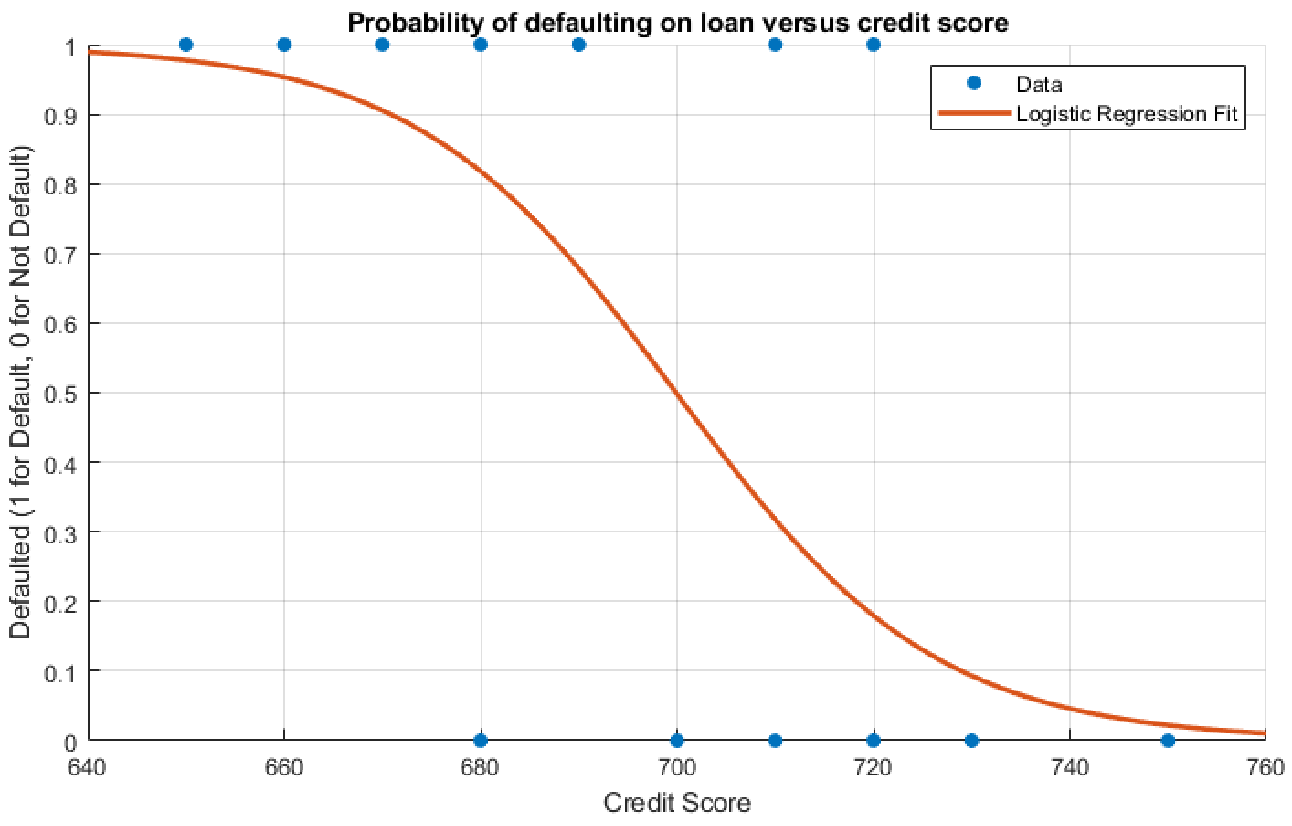
# Why use Logistic Regression?

- Best fit line (BFL) changes due to outliers.
- The new BFL gives us wrong prediction.
- Logistic Regression gives us output between 0 and 1 by “Squashing” the BFL around the edges.




[Source: https://tinyurl.com/ydzyhyt2]

# Logistic Regression



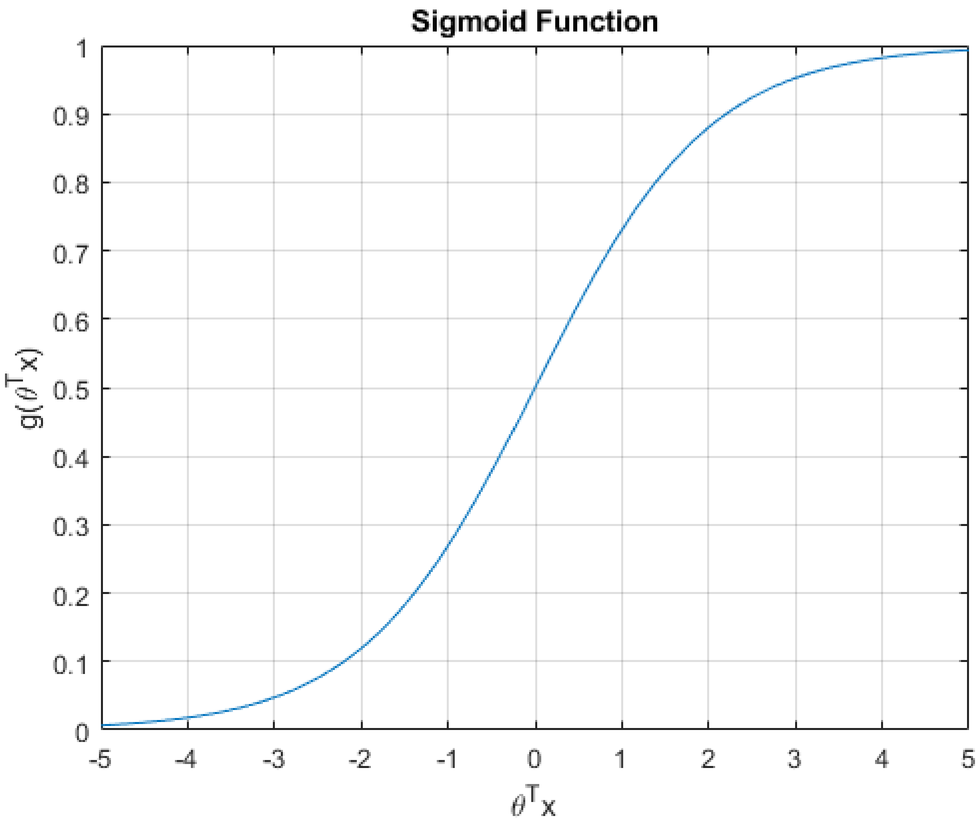

| Credit Score | 650 | 660 | 660 | 670 | 680 | 680 | 680 | 690 | 690 | 700 | 700 | 710 | 710 | 720 | 720 | 720 | 730 | 730 | 750 | 750 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Defaulted | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

# The Logistic Regression Algorithm

Predicted Value $0 \leq h_\theta(x) \leq 1$

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$h_\theta(x) = P(y = 1|x; \theta)$$




Hypothesis:

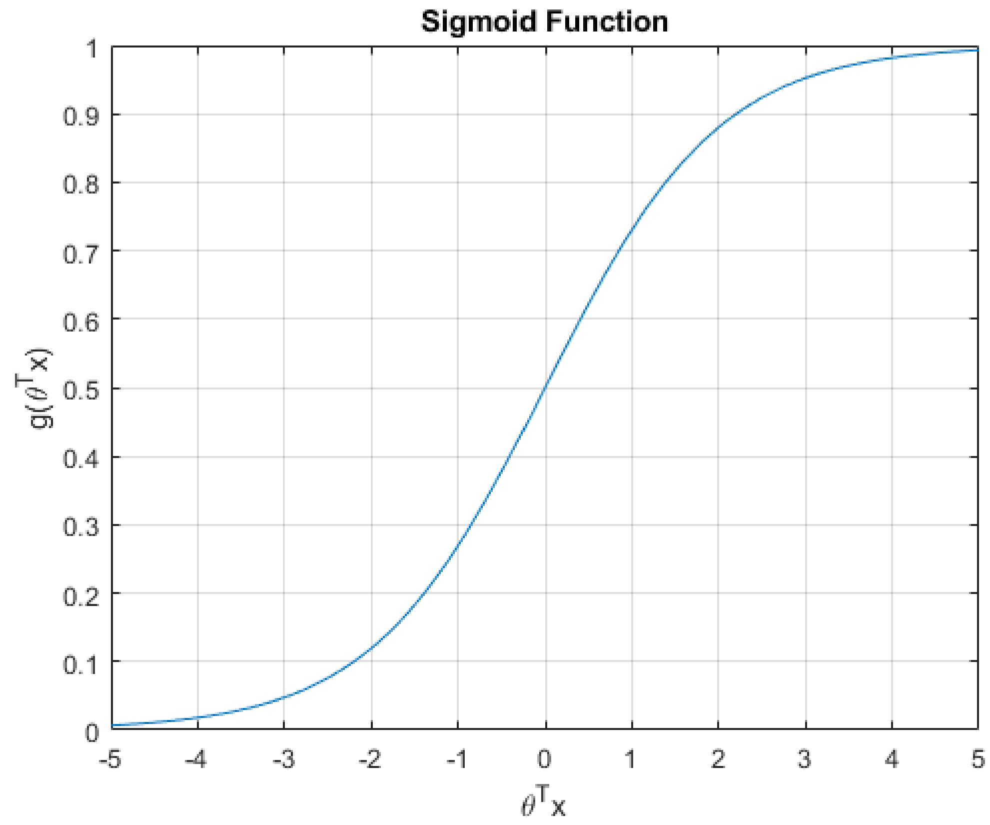
$$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$\theta = [\theta_0, \theta_1, \theta_2, \ldots, \theta_n]$ is parameters in Logistic Regression

# Sigmoid function

- $g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

- Range between 0 and 1
- S–shaped curve
- Smoothness and continuity



# Decision Boundary (Threshold)

Basis for predictions

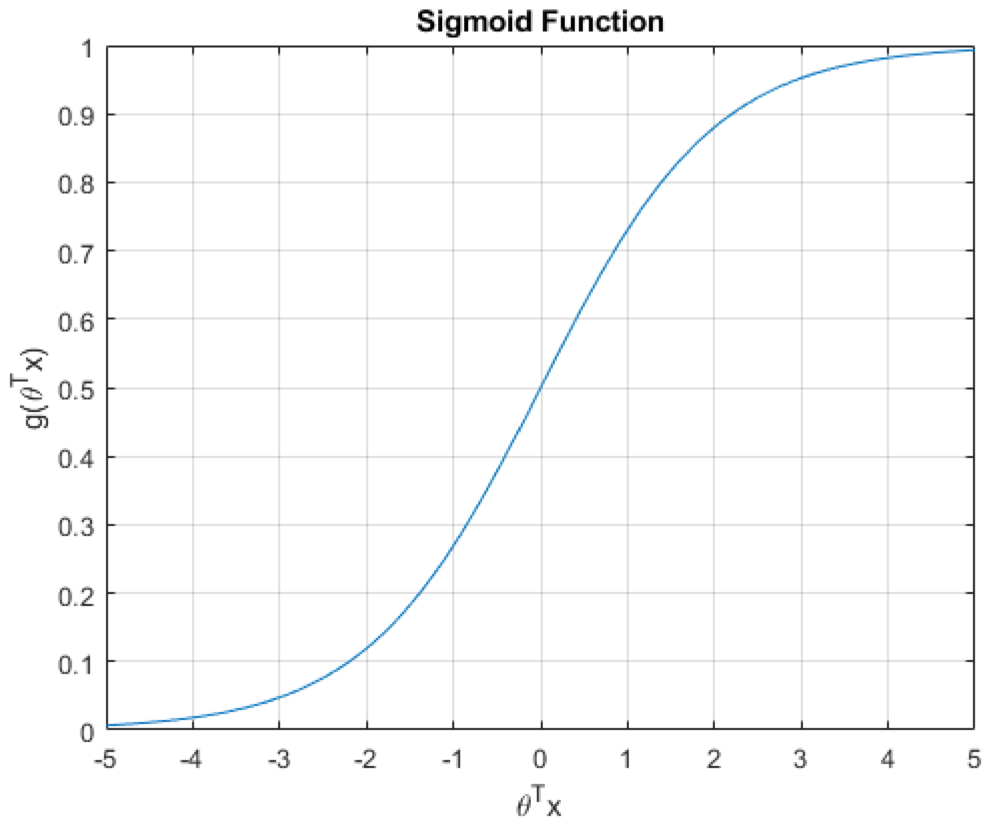
$if\ h_\theta(x) \geq 0.5, \qquad predict\ \ y = 1$

$if\ h_\theta(x) < 0.5, \qquad predict\ \ y = 0$

Which means

$if\ \theta^T x \geq 0, \qquad predict\ \ y = 1$

$if\ \theta^T x < 0, \qquad predict\ \ y = 0$




Where $\theta^T x$ is called Decision Boundary here

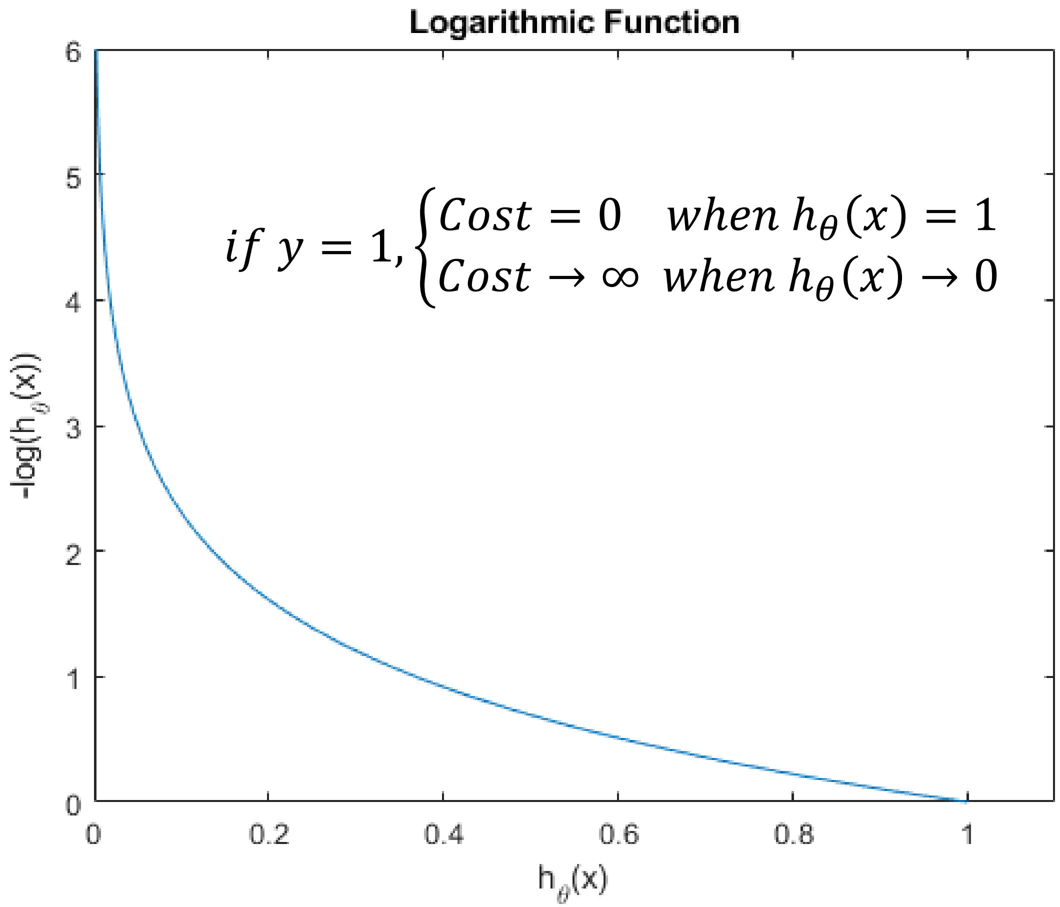


University of Windsor

# Maximum Likelihood Estimation (MLE)

To measure how accurate the Logistic Regression is, define Cost Function:

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$$



# Maximum Likelihood Estimation (MLE)

Simplified cost function:

$Cost\ (h_\theta(x), y) = -ylog\big(h_\theta(x)\big) - (1-y)log(1-h_\theta(x))$

For a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$,

Loss Function $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$

$$= -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}log\left(h_\theta(x^{(i)})\right) + (1-y^{(i)})\log(h_\theta(x^{(i)})))$$

# Maximum Likelihood Estimation (MLE)

To get right $\theta$ value, define $\theta_j$,

$Repeat\{\theta_j \coloneqq \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)\}$, $\alpha$ is learning rate

$$\frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$Repeat\{\theta_j \coloneqq \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}\}$$

Until $\theta$ is less than the set value or the cycle reaches upper limit, take $\theta = \theta_j$

# An Example:
# Prediction if people own property

- On the right is a table contains 3 features: "age", "yearly salary" and "own property"
- "Age" and "yearly salary" are independent variables
- "Own property" is a categorical dependent variable
- Use the "age" and "yearly salary" of a person to predict if they "own property".

| Age | Yearly Salary | Own property |
|---|---|---|
| 16 | 24000 | 0 |
| 18 | 29000 | 0 |
| 55 | 60000 | 1 |
| 23 | 48000 | 0 |
| 29 | 50000 | 0 |
| 66 | 75000 | 1 |
| 9 | 1500 | 0 |
| 12 | 2000 | 0 |
| 13 | 2000 | 0 |
| 40 | 55000 | 1 |
| 6 | 500 | 0 |
| 24 | 48000 | 1 |
| 22 | 36000 | 0 |
| 22 | 39000 | 1 |
| 21 | 35000 | 0 |

# Sample code

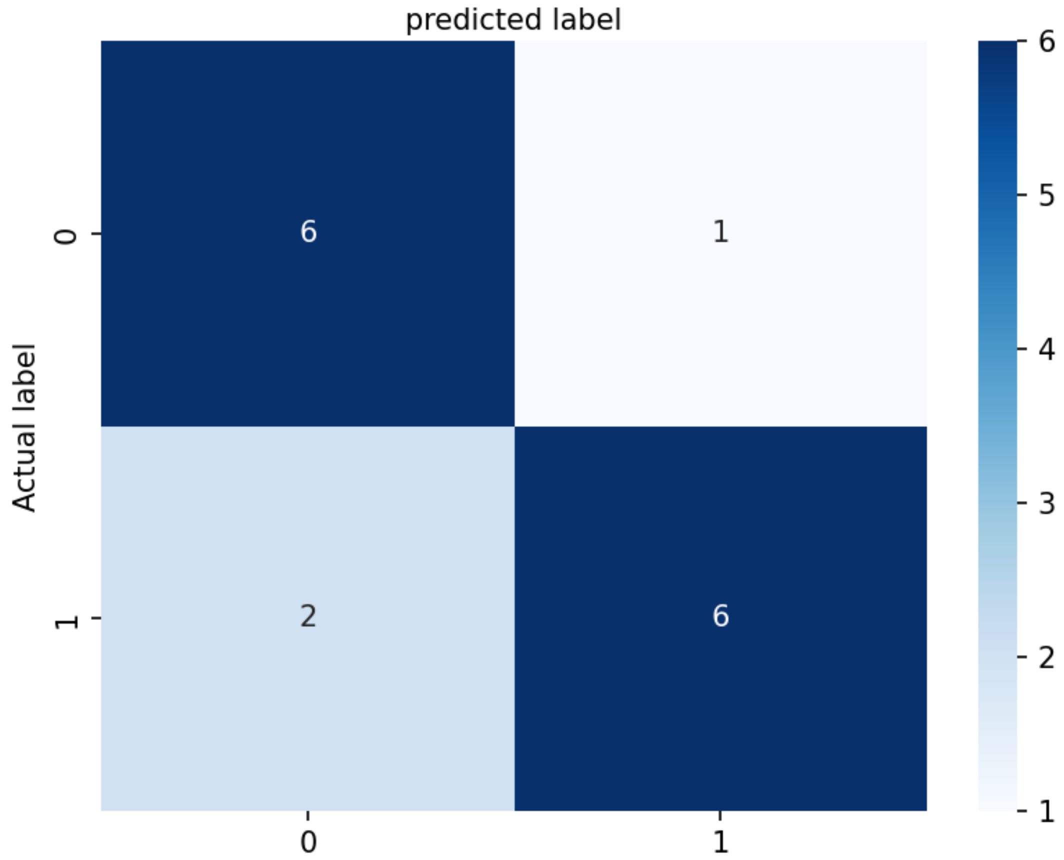
```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

# selecting features

col_names = ['age', 'yearly_salary', 'own_property']

sample_data = pd.read_csv("./sample_data.csv",
header=None, names=col_names)


feature_col = ['age', 'yearly_salary']

X = sample_data[feature_col]

y = sample_data.own_property # label


# split data

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=26)
```

```
# choose model and training
logreg = LogisticRegression(solver='lbfgs')

logreg.fit(X_train, y_train)

# set the threshold to 0.5 (default is 0.5)
Threshold = 0.5

# probability prediction
y_pred = np.where(logreg.predict_proba(X_test)[:, 1]
> Threshold, 1, 0)
```

# Evaluate Model Performance

**Confusion Matrix**

- On the right is the confusion matrix for evaluating our model's performance.
- We have 6 true positives and true negatives.
- We have 1 false positive and 2 false negatives.



# Evaluate Model Performance

**Precision, Recall, F1-score**

- On the right is a table contains the scores of the model from different measurements for 2 classes.
- Our model classifies into 2 classes: “has property” and “Does not have property”.
- Since we are interested in class “has property”. We will focus on the scores from row 2 of the table.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Does not have property | 0.75 | 0.86 | 0.80 | 7 |
| Has property | 0.86 | 0.75 | 0.80 | 8 |
| Accuracy | | | 0.80 | 15 |
| Macro avg | 0.80 | 0.80 | 0.80 | 15 |
| Weighted avg | 0.81 | 0.80 | 0.80 | 15 |

# Evaluate Model Performance

## **Precision, Recall, F1-score**

- Precision is given by:
  - $Precision = \frac{TP}{TP+FP} = \frac{6}{6+1} = 0.86$
- Recall is given by:
  - $Recall = \frac{TP}{TP+FN} = \frac{6}{6+2} = 0.75$
- F1-score is given by:
  - $F1\ score = \frac{2*(Precision*recall)}{Precision+recall} = \frac{2*(0.857*0.75)}{0.857+0.75} = 0.80$

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Does not have property | 0.75 | 0.86 | 0.80 | 7 |
| Has property | 0.86 | 0.75 | 0.80 | 8 |
| Accuracy | | | 0.80 | 15 |
| Macro avg | 0.80 | 0.80 | 0.80 | 15 |
| Weighted avg | 0.81 | 0.80 | 0.80 | 15 |

# Evaluate Model Performance

**Precision, Recall, F1-score**

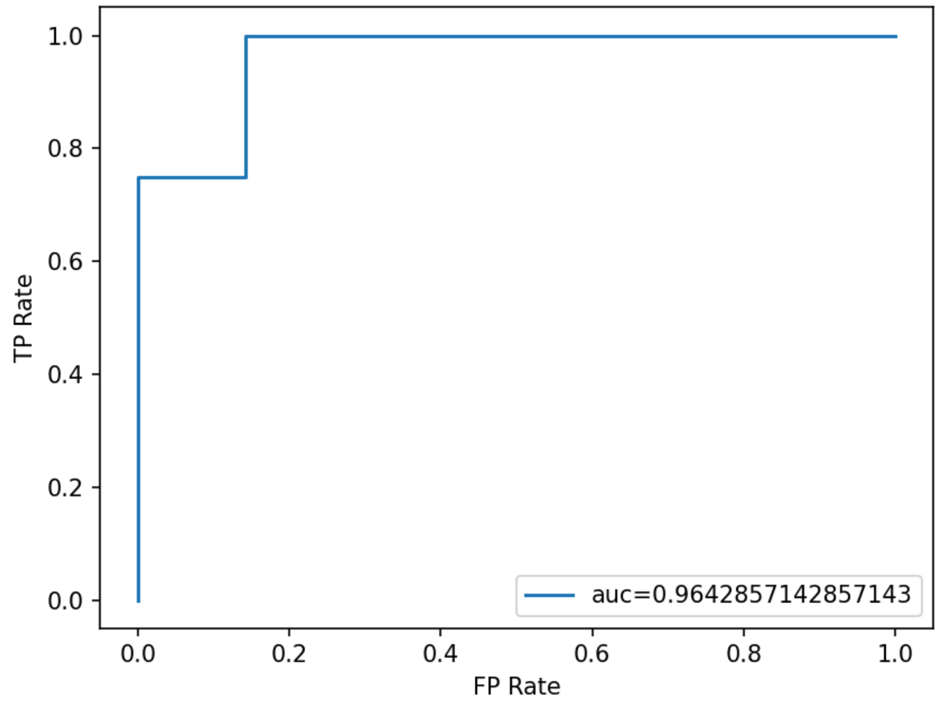
- The Precision score is 0.86 which is relatively good.
- The recall score is 0.75 which could be better. We could optimize it by modifying threshold.
- F1-score is 0.80 means the class distribution is not very imbalanced.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Does not have property | 0.75 | 0.86 | 0.80 | 7 |
| Has property | 0.86 | 0.75 | 0.80 | 8 |
| Accuracy | | | 0.80 | 15 |
| Macro avg | 0.80 | 0.80 | 0.80 | 15 |
| Weighted avg | 0.81 | 0.80 | 0.80 | 15 |

# Evaluate Model Performance

**AUC-ROC Curve**

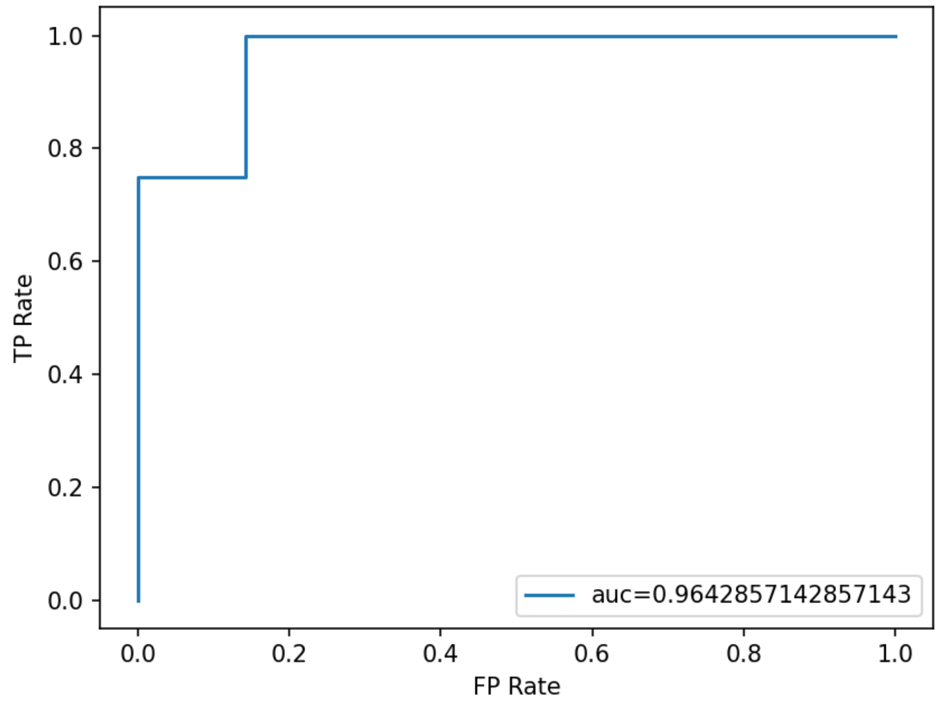
- Performance measurement for classification problem at different thresholds.
- Receiver Operating Characteristic (ROC) is probability curve



# Evaluate Model Performance

### AUC-ROC Curve

- Area under the *ROC Curve* (AUC) is the area under the curve. It measures the separability.
- Our model's AUC score is 0.9643 which means it distinguish our positive and negative classes well. Thus, it has good performance.

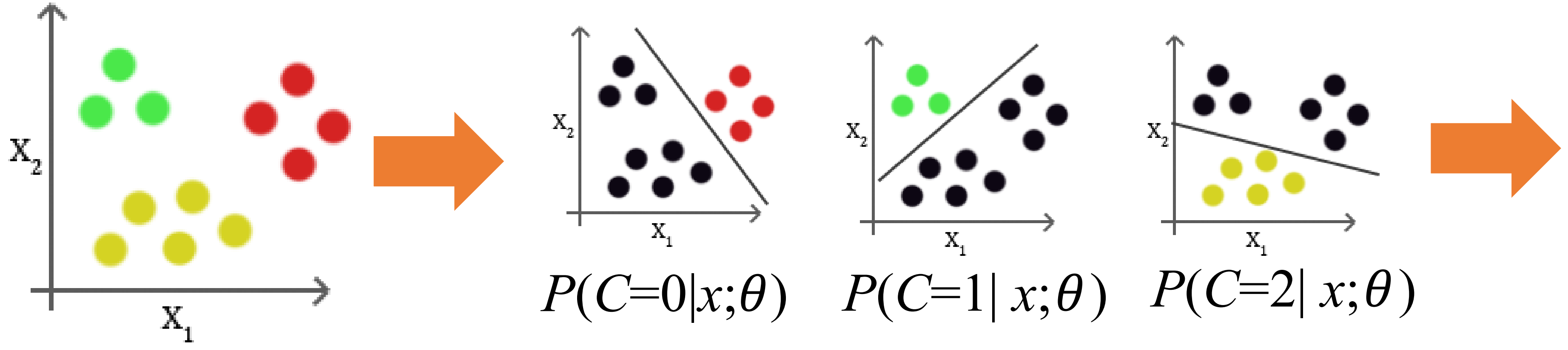


# Logistic Regression for Multiclass classification

- Multi-classes (more than 2 dependent features) problems can be solved by Logistic regression.
- Common approach include **Leave One Out** (One-vs-rest) and **Softmax Regression** (Multinomial approach)
- https://tinyurl.com/6rw854w7

# Leave One Out (One-vs-rest)

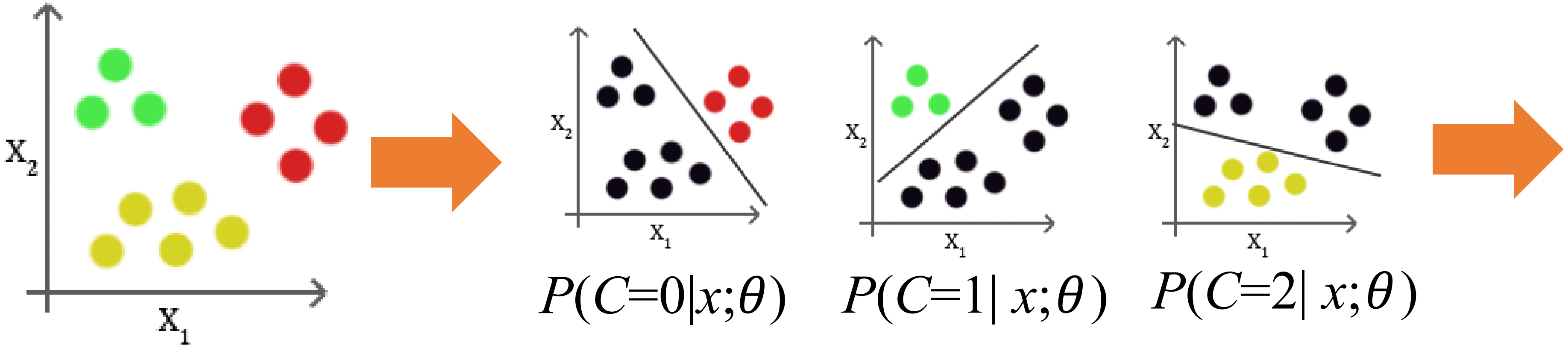
- Build logistic regression models for each class and select the class with the highest probability predicted by the models.




Output the class with the highest probability

# Leave One Out (One-vs-rest)

- In the example below, we have 3 different classes $C$ given input $x$ and coefficient $\theta$. We output the class with highest probability $P$.




Output the class with the highest probability

# Softmax Regression (Multinomial approach)

- $x$ is input data and $\theta$ is coefficients (weights and bias).
- $K$ is the maximum number of classes. $e$ is Euler's number.
- $P$ is the probability of each class output by the model.

$$\frac{1}{\sum_{j=1}^{K} e^{\theta^{(j)T}x}} \begin{bmatrix} e^{\theta^{(1)T}x} \\ e^{\theta^{(2)T}x} \\ \dots \\ e^{\theta^{(K)T}x} \end{bmatrix} = \begin{bmatrix} P(C=0|x;\theta) \\ P(C=1|x;\theta) \\ \dots \\ P(C=K|x,\theta) \end{bmatrix}$$

# Softmax Regression (Multinomial approach)

- Compute the probability of the classes for each row of the dataset
- Probabilities output by each different model amounting to 1
- Output the class with the highest probability

$$\frac{1}{\sum_{j=1}^{K} e^{\theta^{(j)T}x}} \begin{bmatrix} e^{\theta^{(1)T}x} \\ e^{\theta^{(2)T}x} \\ \dots \\ e^{\theta^{(K)T}x} \end{bmatrix} = \begin{bmatrix} P(C = 0|x;\theta) \\ P(C = 1|x;\theta) \\ \dots \\ P(C = K|x,\theta) \end{bmatrix}$$

# Reference Video

[Source: https://tinyurl.com/4k9rreuv]

# Conclusions

- Logistic regression is a statistical method used for predicting the probability of an event happening.
- Linear regression is not enough to classify outcomes because of varying BFL due to outliers. That is why we need Logistic regression.
- Logistic regression uses sigmoid function to get an output between 0 and 1.
- Logistic Regression output's 1 if above threshold, and 0 if below.
- To measure the performance of logistic regression models, we use Confusion matrix, Precision, Recall and F1-score.
- AUC-ROC curve measures how well the model distinguish between classes.
- There are 2 approaches for multiclass classification: Leave one Out and Softmax Regression.

# References

[1] Sarang Narkhede, “Understanding Logistic Regression,” *Medium*, May 17, 2018. https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102 (accessed Oct. 06, 2023).

[2] Saishruthi Swaminathan, “Logistic Regression — Detailed Overview,” Medium, Mar. 15, 2018. https://tinyurl.com/529x54ts

[3] M. P. LaValley, “Logistic Regression,” *Circulation*, vol. 117, no. 18, pp. 2395–2399, May 2008, doi: https://doi.org/10.1161/circulationaha.106.682658.

[4] “Unsupervised Feature Learning and Deep Learning Tutorial,” *ufldl.stanford.edu*. https://tinyurl.com/2xeu39uh