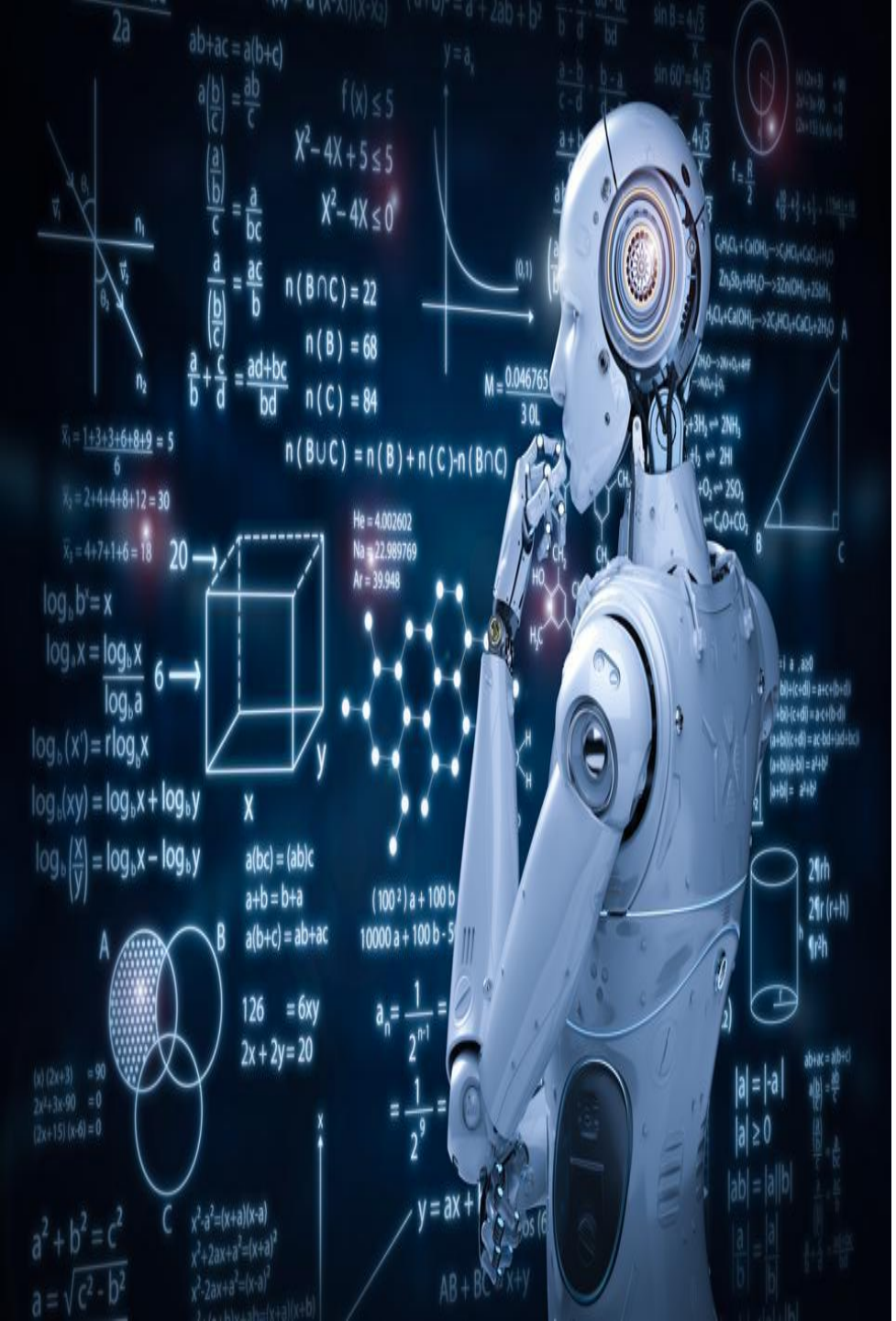


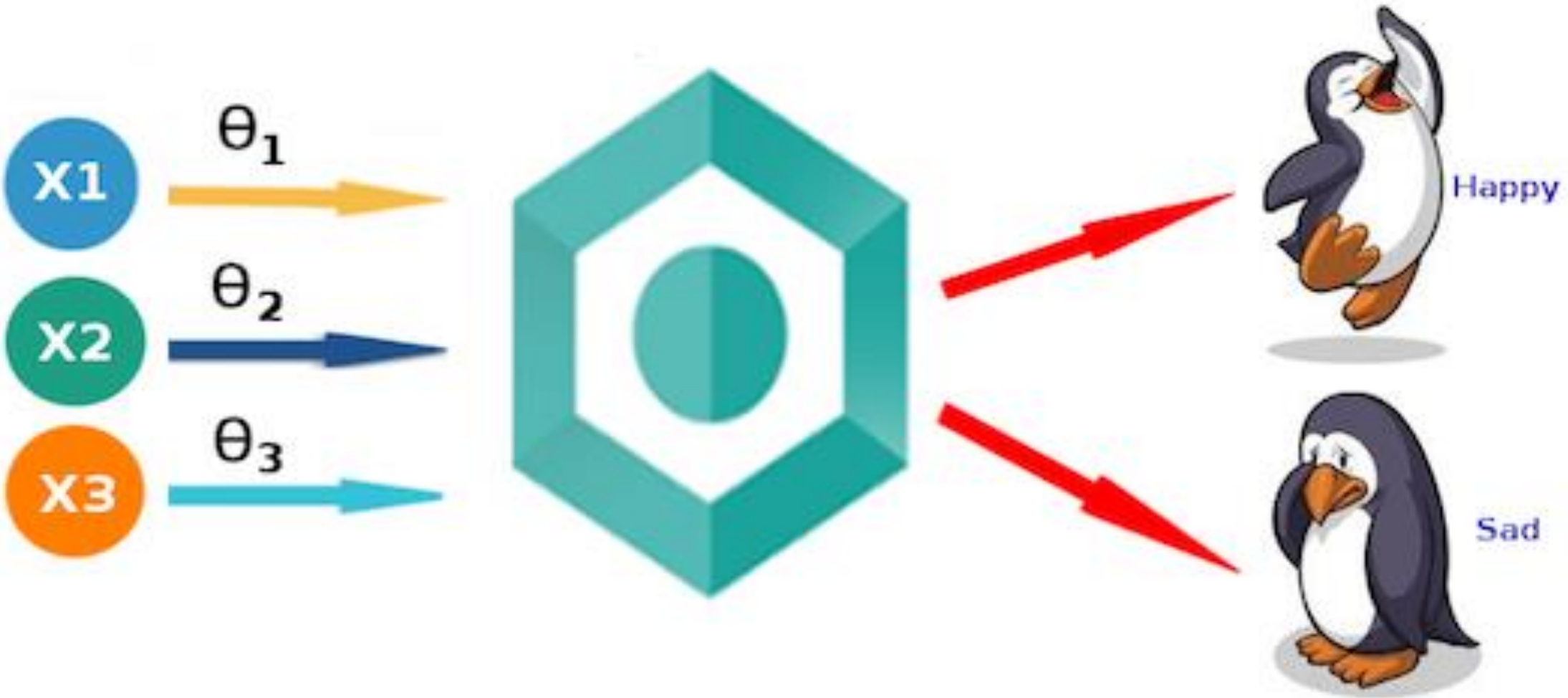
Practical Machine Learning

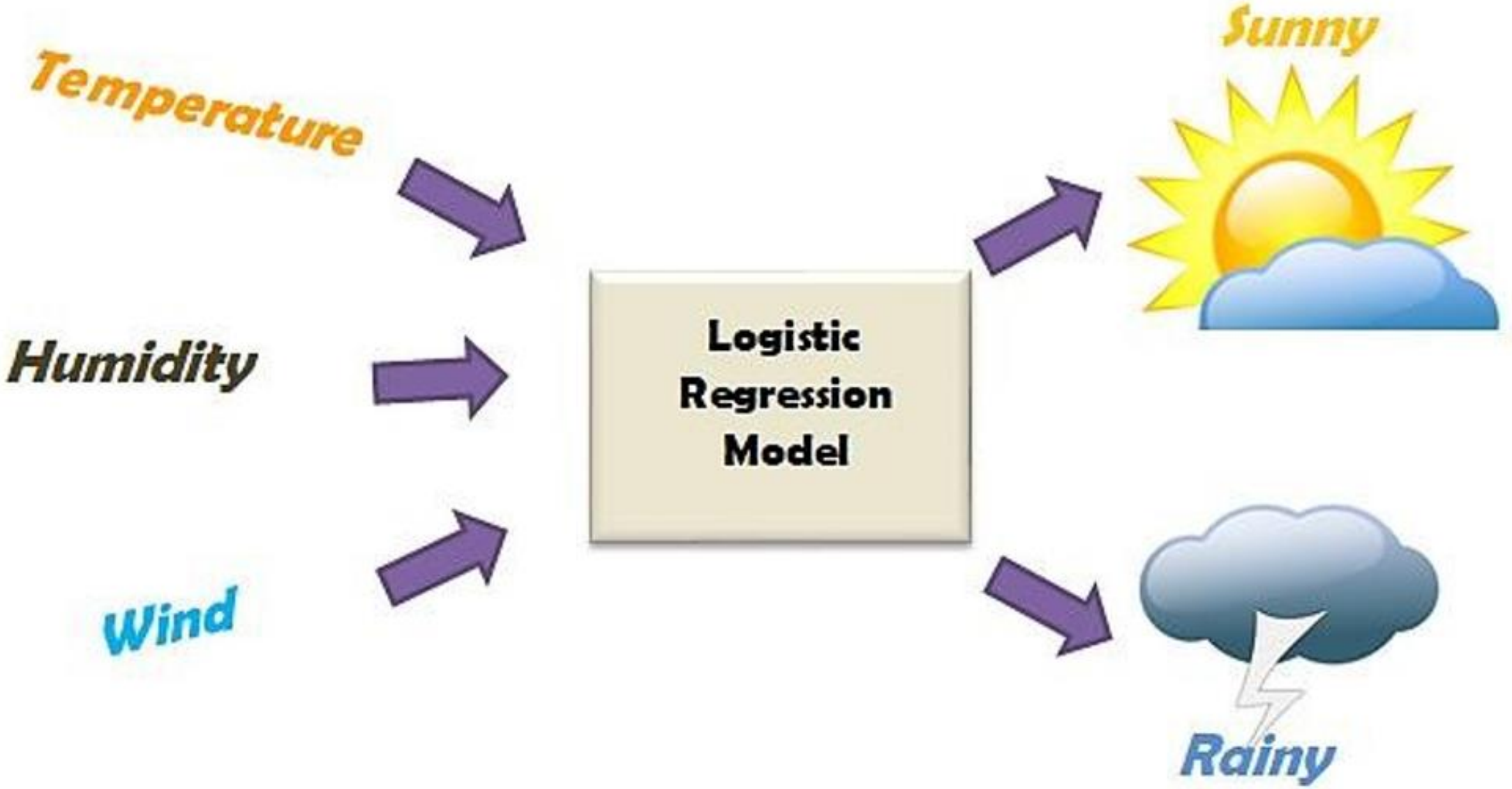
Day 7: Sep22 DBDA

Kiran Waghmare

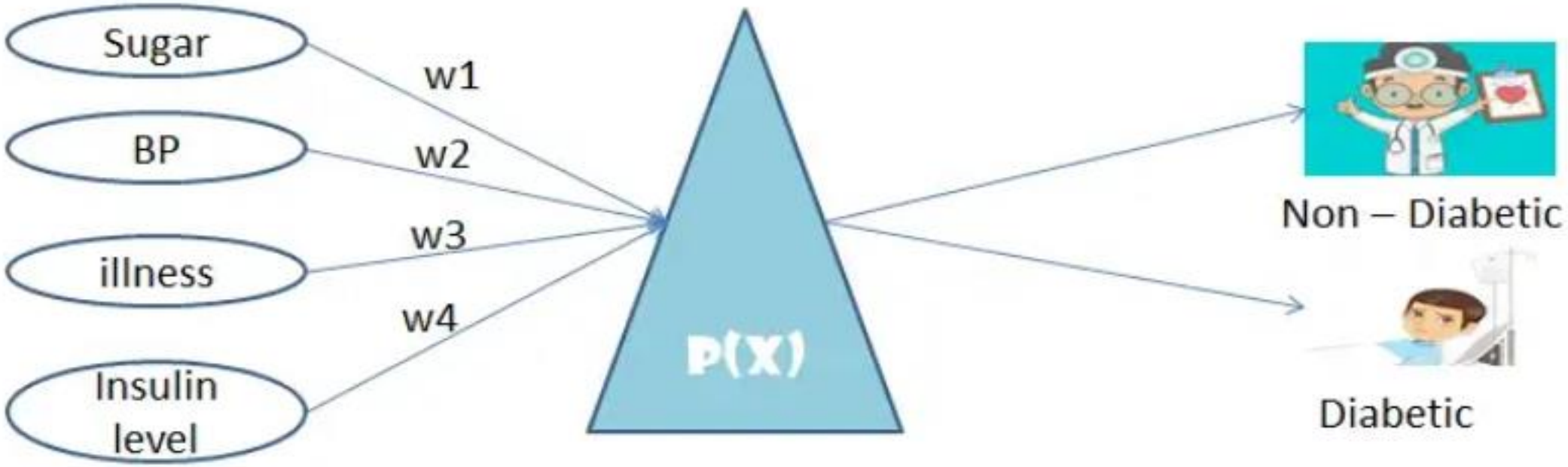
Agenda

- Logistic Regression
- Classification
- Measures for classification
- KNN


X1
X2
X3
θ1
θ2
θ3
Happy
Sad
@dataaspirant.com
Inputs: X1, X2, X3 || Weights: θ1, θ2, θ3 || Outputs: Happy or Sad


Temperature
Humidity
Wind
Logistic Regression Model
Sunny
Rainy

LOGISTIC REGRESSION MODELLING



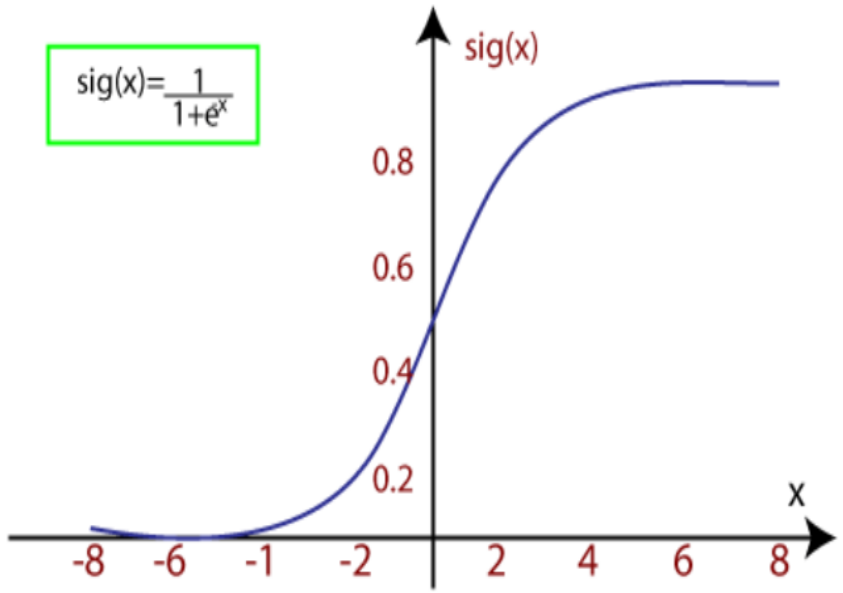
W1,w2,w3,w4 – Amount of each individual medical problem
P(x) – Probability Calculation

Logistic Regression

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

- f(x)= Output between the 0 and 1 value.
- x= input to the function
- e= base of natural logarithm.

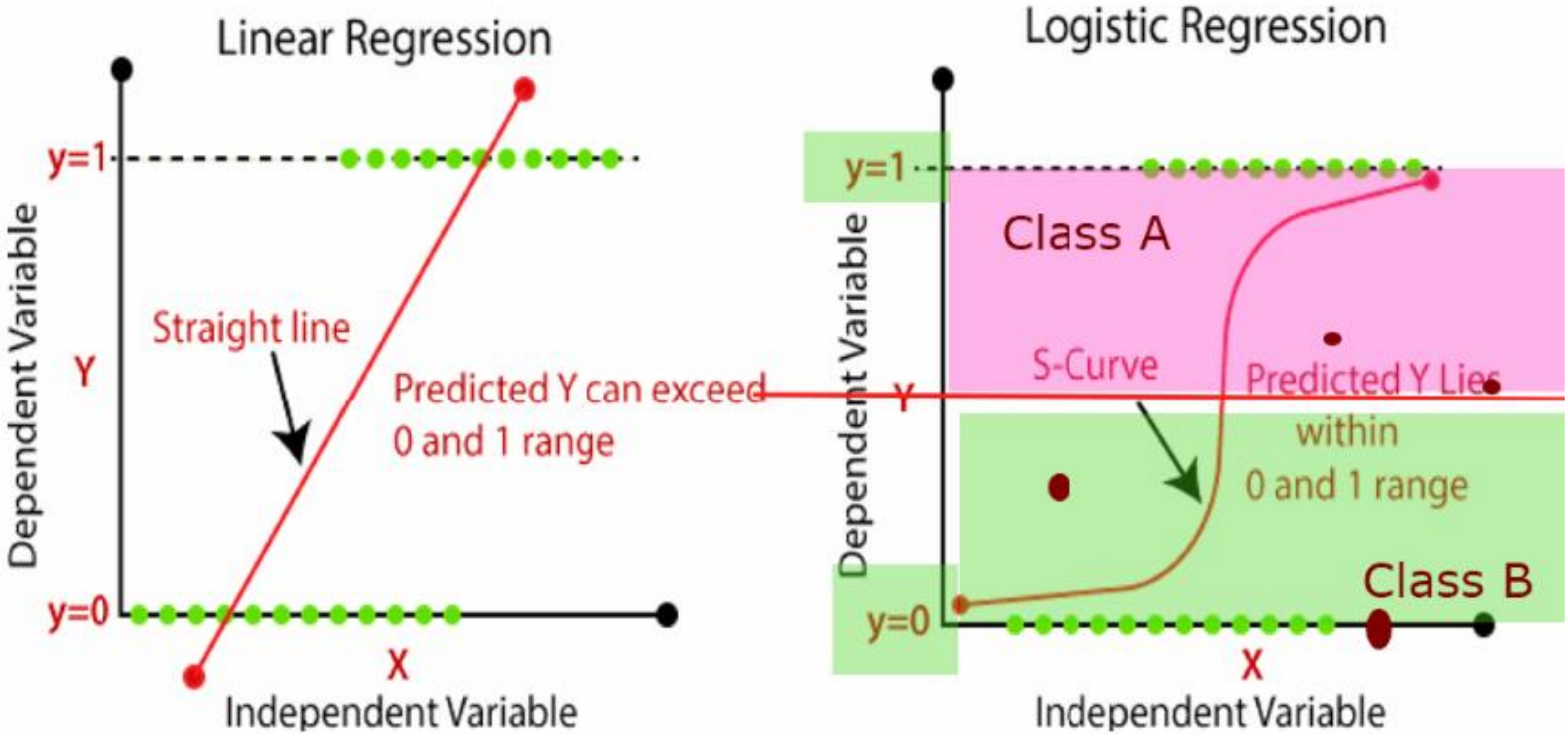
When we provide the input values (data) to the function, it gives the S-curve as follows

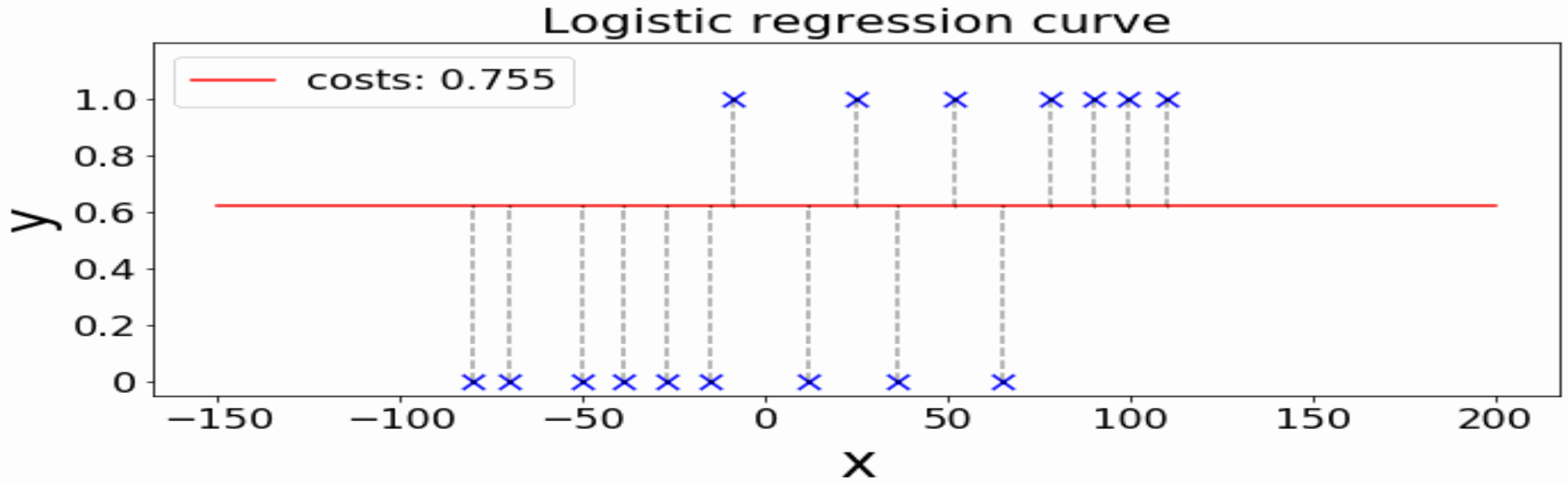


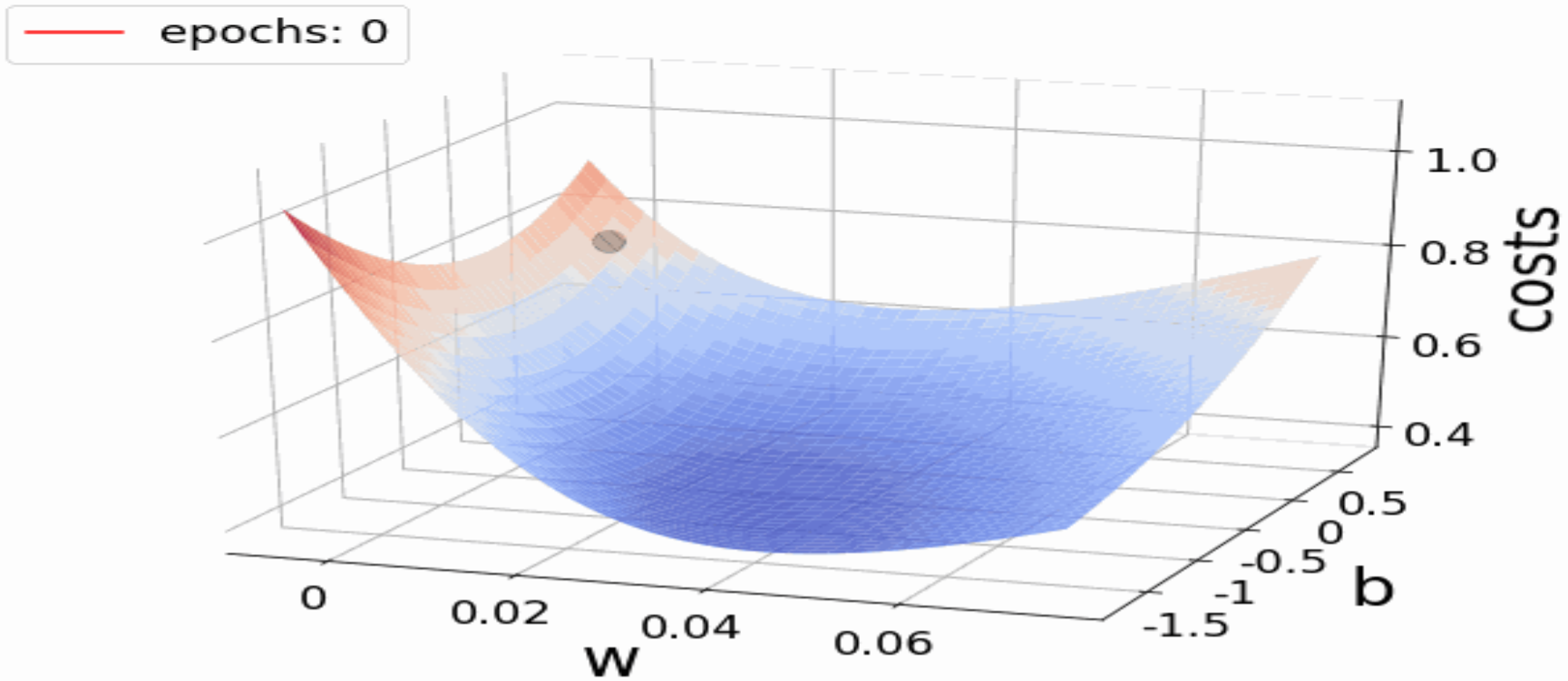
- It uses the concept of threshold levels, values above the threshold level are rounded up to 1, and values below the threshold level are rounded up to 0.

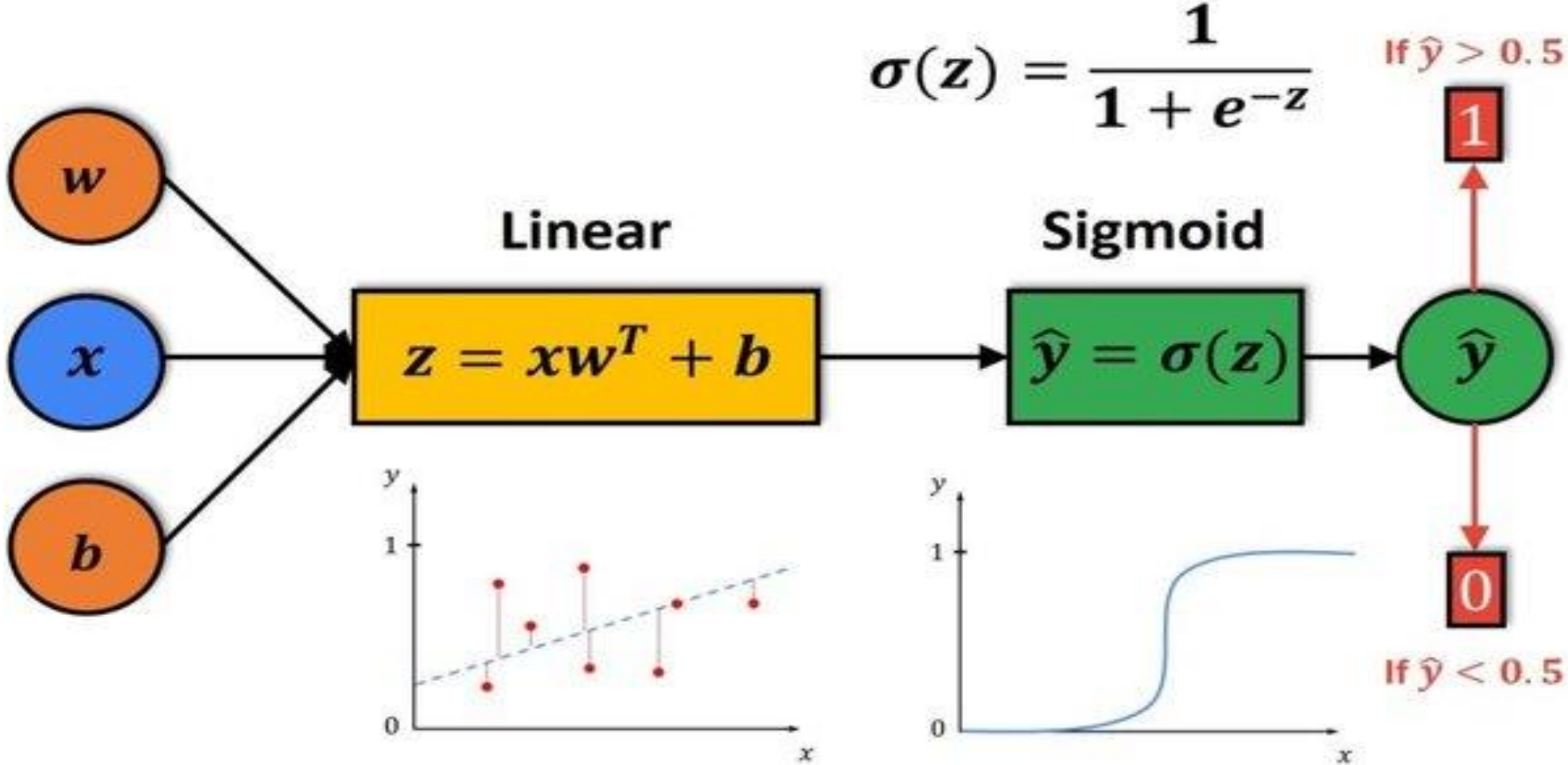
There are three types of logistic regression:

- **Binary(0/1, pass/fail)**
- **Multi(cats, dogs, lions)**
- **Ordinal(low, medium, high)**


Linear Regression
y=1
y=0
Dependent Variable
Y
Straight line
Predicted Y can exceed
0 and 1 range
X
Independent Variable
Logistic Regression
y=1
y=0
Dependent Variable
Y
Class A
S-Curve
Predicted Y Lies
within
0 and 1 range
Class B
X
Independent Variable


Logistic regression curve
costs: 0.755
y
x
1.0
0.8
0.6
0.4
0.2
0
−150
−100
−50
0
50
100
150
200


epochs: 0
costs
w
b
0
0.02
0.04
0.06
1.0
0.8
0.6
0.4
0.5
0
-0.5
-1
-1.5


$\sigma(z) = \frac{1}{1+e^{-z}}$
If $\hat{y} > 0.5$
1
w
x
b
Linear
$z = xw^T + b$
Sigmoid
$\hat{y} = \sigma(z)$
$\hat{y}$
0
If $\hat{y} < 0.5$
y
1
0
x
y
1
0
x

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code Commit

```
X_test=sc.fit_transform(X_test)
```

In [23]:
```
#X_train
```

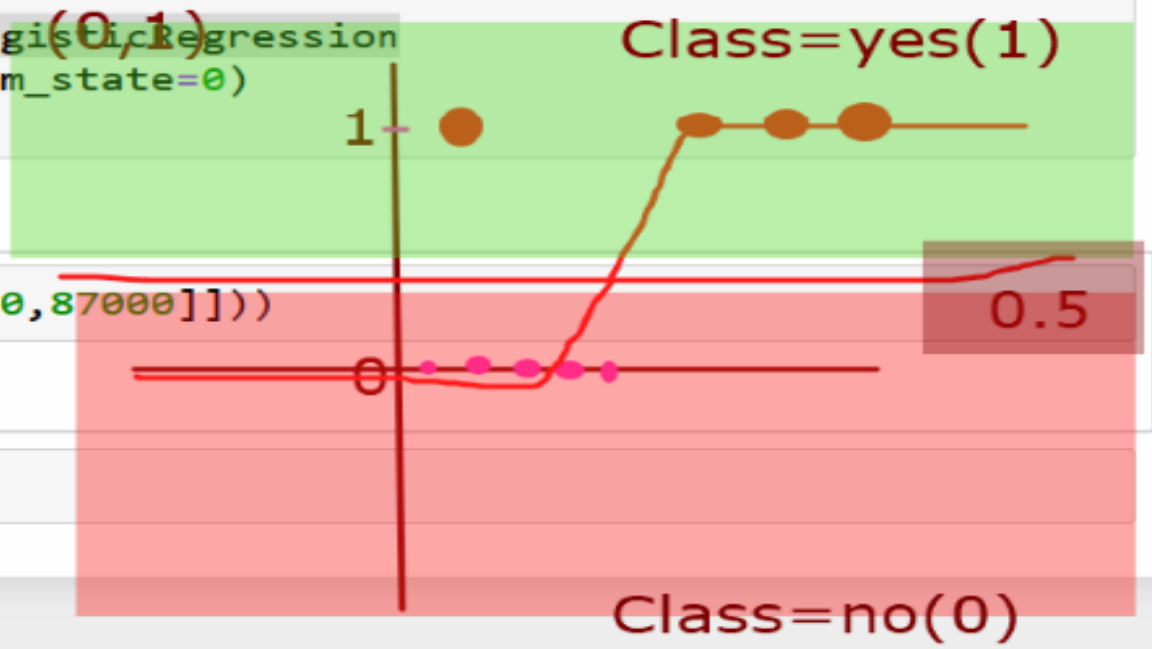
In [13]:
```
#Logistic Model
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression(random_state=0)
classifier.fit(X_train,y_train)
```

Out[13]:
```
LogisticRegression(random_state=0)
```

In [24]:
```
classifier.predict(sc.transform([[20,87000]]))
```

Out[24]:
```
array([1], dtype=int64)
```

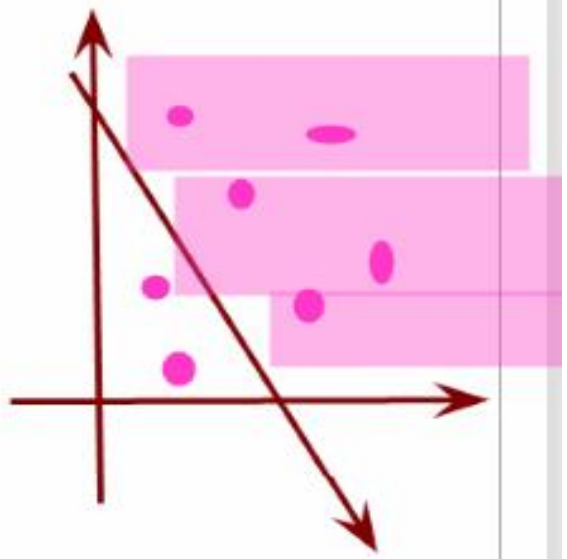
In [ ]:



File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code Commit

```
avoided as value-mapping will have precedence in case its length matches with *
x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a
single row if you intend to specify the same RGB or RGBA value for all points.
```



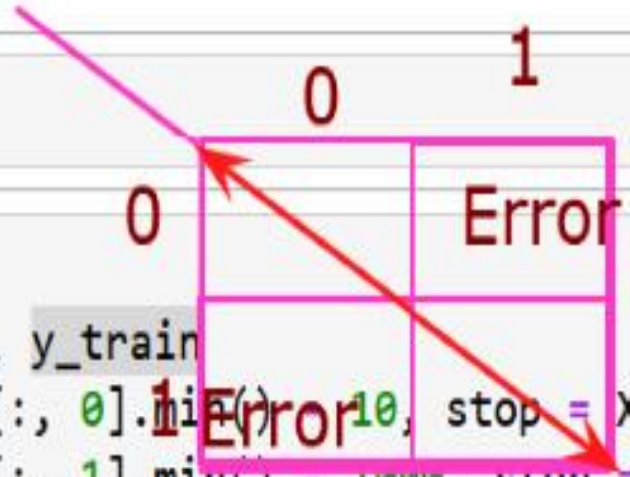
In [17]:
```
#Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
cm=confusion_matrix(y_test,y_pred)
cm
```

Out[17]:
```
array([[52,  6],
       [ 3, 19]], dtype=int64)
```

min=Error

max=Accuracy

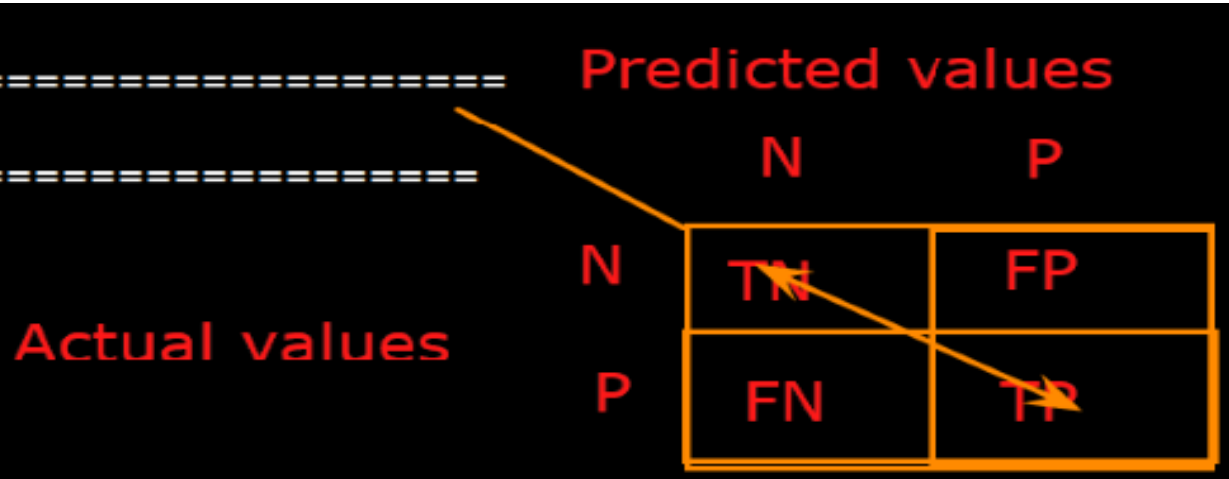
In [ ]:



In [14]:
```
from matplotlib.colors import ListedColormap
X_set, y_set = sc.inverse_transform(X_train), y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 10, stop = X_set[:, 0
                     np.arange(start = X_set[:, 1].min() - 1000, stop = X_set[:,
plt.contourf(X1, X2, classifier.predict(sc.transform(np.array([X1.ravel(), X2.ra
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
```

```
==================================================
Day 7: Classification
==================================================
Date: 09/01/2023
Topics:
------------------
    -Logistic Regression
    -Classification concept

    A,B,C,D
    A={A+,A,A-}
```



Population=P + N

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

P: positive case
N:negatives case
TP:correct predicted true values
TN:correctly predicted false values
FP:wrongly predicted positive values
FN:wrongly predicted negative values

Problem Statement

- Titanic dataset
- **Explore:** How does each feature relate to whether a person survives/alives?
- Do the EDA in more detail than usual and explain the results!
  - Splitting: 80-20, stratify: y, random_state = 0
- **Preprocessing:**
  - * Drop decks
  - * Fill in the missing value using a simple imputer
  - * One hot encoding: sex, alone
  - * Ordinal encoding: class
  - * Binary encoding: embark town
- **Model selection:**
  - * Evaluation metrics used: F1_score
  - Logistic Regression