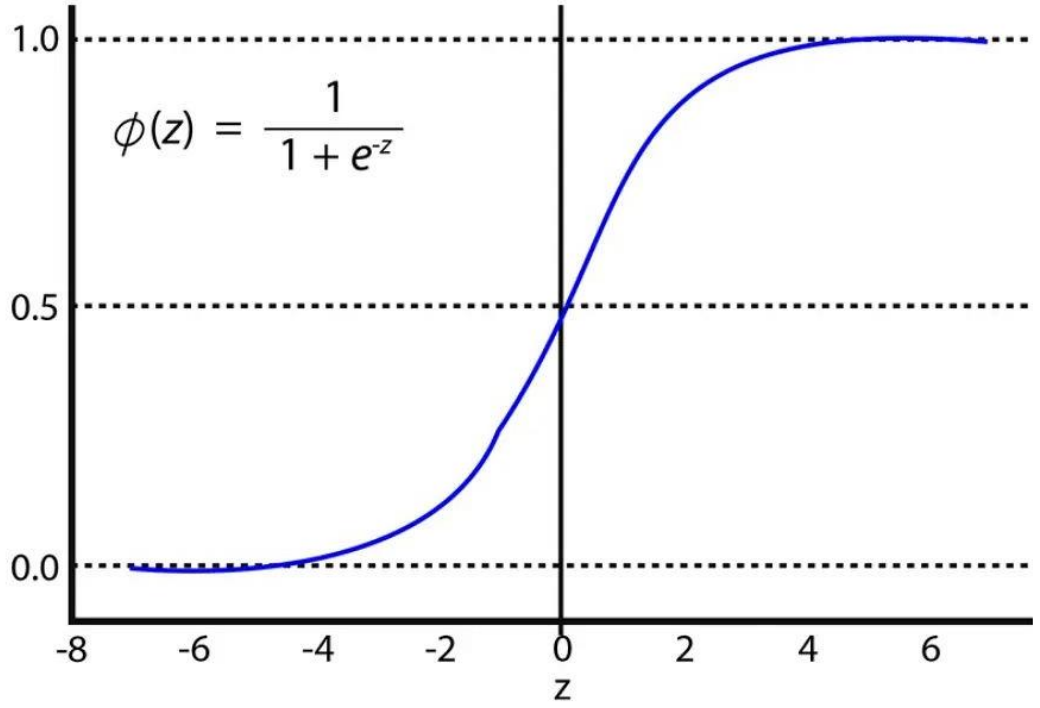
Logistic Regression

- Don't get confused by its name! It is a **classification**, not a regression algorithm.
- It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s).
- In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, It also known as logit regression.
- Since, it predicts the probability, its output values lies between 0 to 1 (as expected).



$$\Pr(Y = 1|X) = F(\beta_0 + \beta_1 X)$$

F is the cumulative standard ***logistic*** distribution function:

where $F(\beta_0 + \beta_1 X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$

Example: $\beta_0 = -3$, $\beta_1 = 2$, $X = .4$,
so $\beta_0 + \beta_1 X = -3 + 2*.4 = -2.2$ so
$\Pr(Y = 1|X=.4) = 1/(1+e^{-(-2.2)}) = .0998$

Why bother with logit if we have probit?

- Historically, logit is more convenient computationally
- In practice, logit and probit are very similar

Test for Trend - Logistic Regression Alternative

Logistic Model:

$$\Pr(Y=1|X_1,X_2,\ldots,X_p) = \frac{1}{1+e^{-(b_0+b_1X_1+b_2X_2+\ldots+b_pX_p)}}$$

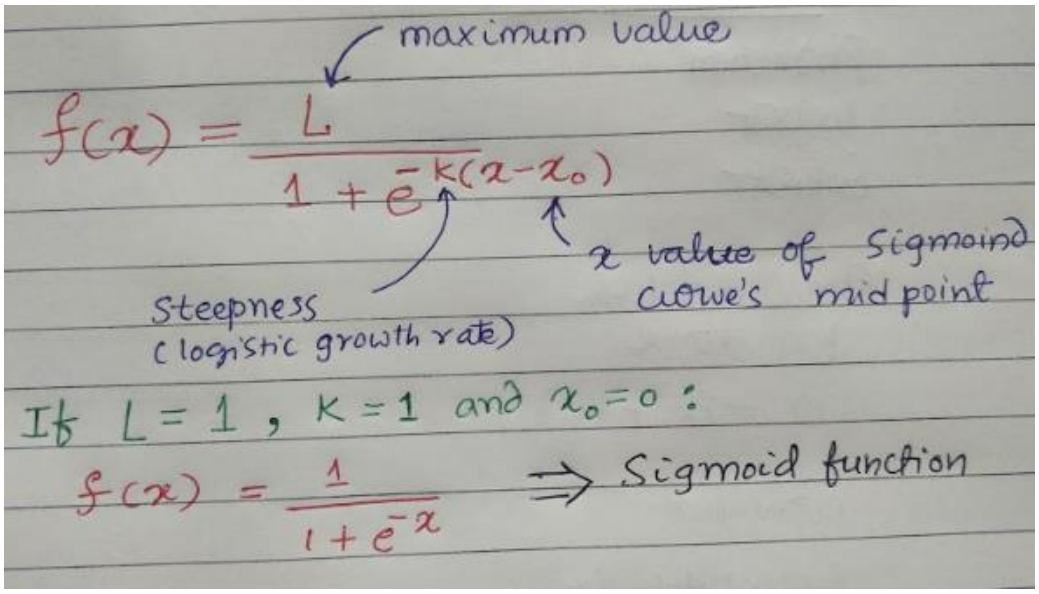
$$\text{logit } P = \ln\frac{P}{1-P} = b_0+b_1X_1+b_2X_2+\ldots+b_pX_p$$

where $P = \Pr(Y=1|X_1,X_2,\ldots,X_p)$

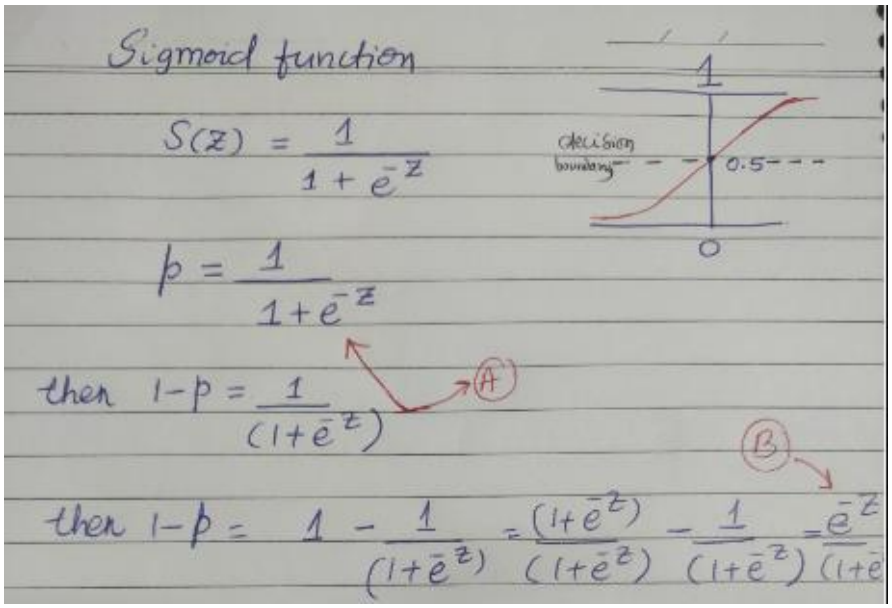
X = independent variable or predictor

Y = dichotomous dependent or outcome variable

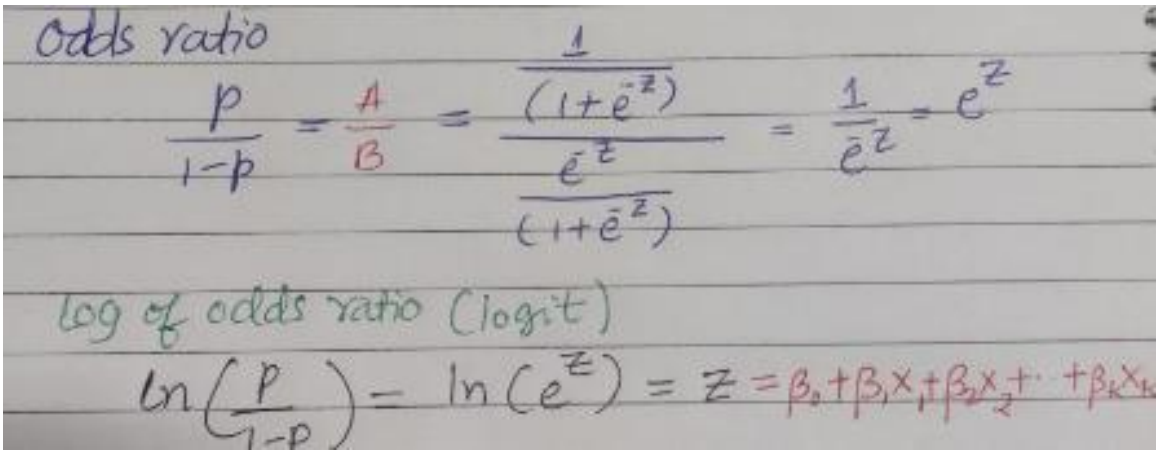
Sigmoid function is a special case of Logistic function and lies between 0 – 1. (Squashing)



Use sigmoid function to model the probability of dependent variable being 1 or 0 (binary classification).



Now we will see how to derive the log of odds ratio [p/(1-p)]



The logistic regression assigns each row a probability of bring True and then makes a prediction for each row where that prbability is >= 0.5 i.e**. 0.5 is the default threshold**.

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('d:Social_Network_Ads.csv')
df
```

Out[2]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [3]:
```
x = df.iloc[:,2:-1]
y = df.iloc[:,-1]
```

In [4]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=4)
```

In [5]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

In [6]:
```
lr.fit(x_train,y_train)
```

Out[6]:
```
LogisticRegression()
```

In [7]:
```
ypre = lr.predict(x_test)
```

In [8]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,ypre)
```

Out[8]:
```
0.7125
```