COMP30810
Intro to Text Analytics

Dr. Binh Thanh Le

thanhbinh.le@ucd.ie

Insight Centre for Data Analytics

School of Computer Science

University College Dublin

Today goals

Regression Model:

Linear Regression for predicting a **numeric** target feature (can also be used for categorical target feature); looks for linear relationship between descriptive and target feature

Classification Model:

Logistic Regression for predicting a **categorical** target feature; looks for linear separation between classes

Example

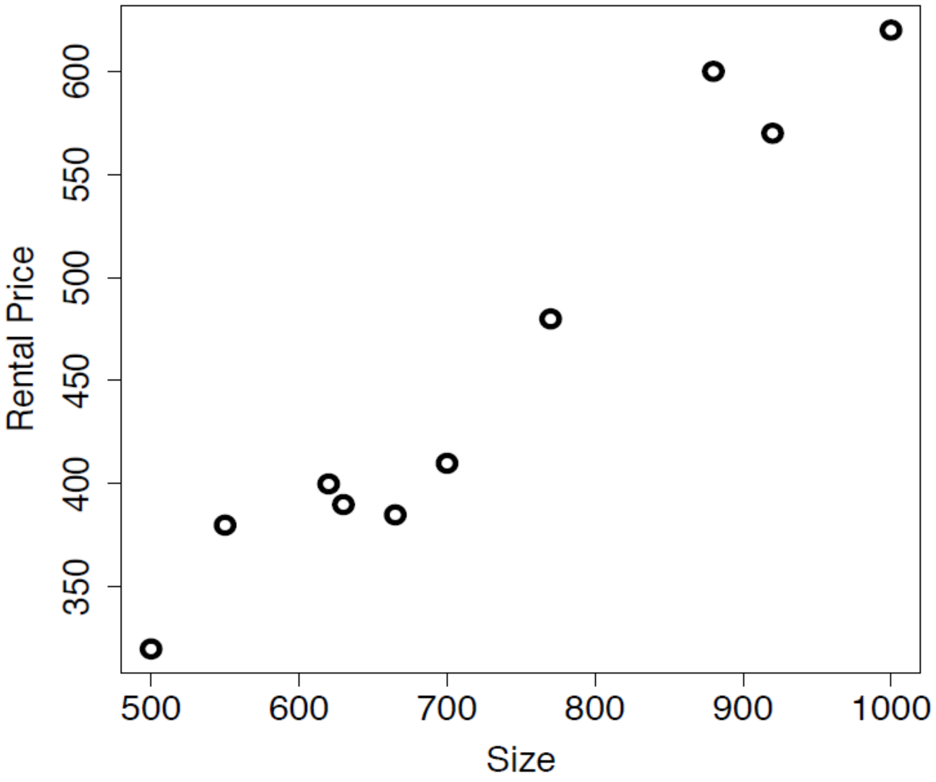
- **Table:** The **office rentals dataset**: a dataset that includes office rental prices and several descriptive features for 10 Dublin city-centre offices.

ID	SIZE	FLOOR	BROADBAND RATE	ENERGY RATING	RENTAL PRICE
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

Can we predict the rental price (target outcome), given the descriptive features for an office?

Simple Example

- **Table:** The **office rentals dataset**: a dataset that includes office rental prices and Size features for 10 Dublin city-centre offices.



ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

Regression VS Classification

- **Regression**: Predict the **RentalPrice** given the **Size** of an office

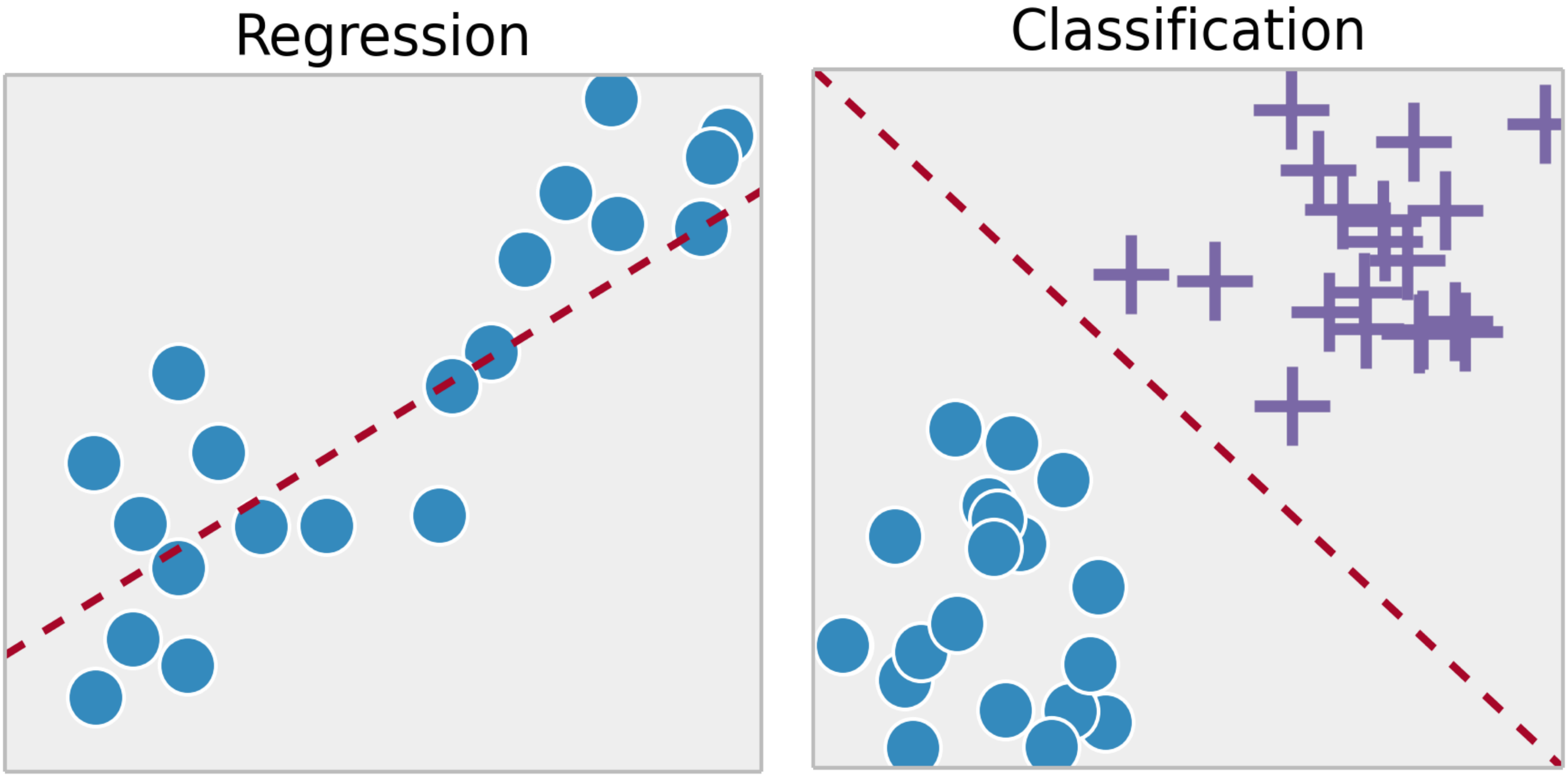
- **Classification:** Predict if the **RentalPrice** is High or Low given the Size of office

  (the focus is on predicting the **Probability(RentalPrice=High| Size)**)

  - We typically work with **two classes** and code one class with 0 and the other class with 1.

  - If **more than 2 classes**, we can use the one-vs-all formulation to create **n** binary classification problems (where **n** is the number of classes).

Regression VS Classification



Linear Regression: Linear Model

- Scatter plot shows linear relationship between **SIZE** and **RENTAL PRICE**
- This relationship can be approximately captured via a parameterized line
- The equation of a line can be written as:

$$\boldsymbol{y} = \boldsymbol{b} + \boldsymbol{m} * \boldsymbol{x}$$

where **m** *is the slope,*
b *is the bias (aka y-intercept)*

Or:

$$\boldsymbol{y} = \boldsymbol{w_0} + \boldsymbol{w_1} * \boldsymbol{x}$$

where $\boldsymbol{w_1}$ *is the slope,*
$\boldsymbol{w_0}$ *is the bias*

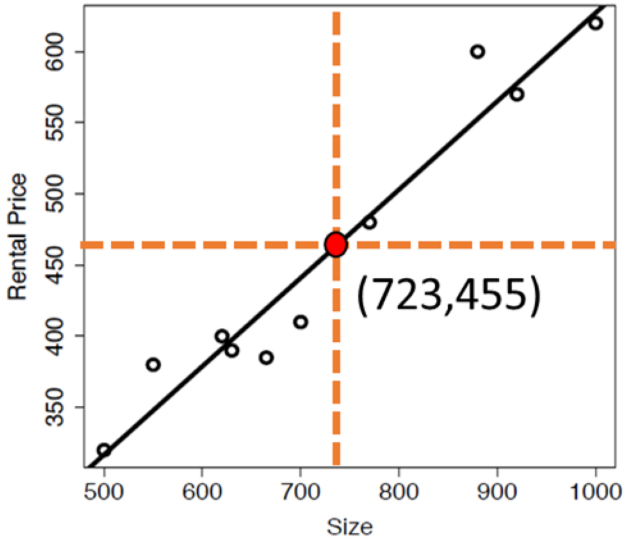
Linear Regression: Linear Model

$$\boldsymbol{y} = \boldsymbol{w_0} + \boldsymbol{w_1} * \boldsymbol{x}$$

where $\boldsymbol{w_1}$ *is the slope,*
$\boldsymbol{w_0}$ *is the bias*

From x and y vectors, calculate $\bar{x}, \bar{y}$ as the means of x and y

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620



$$w_1 = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2} = 0.62$$

$$w_0 = \bar{y} - w_1(\bar{x}) = 6.74$$

RENTAL PRICE = 6.47 + 0.62 x SIZE

Linear Regression: Linear Model

- What is the expected Rental Price for a new/test 730 square foot office?

RENTAL PRICE = 6.47 + 0.62 x SIZE

RENTAL PRICE = 6.47 + 0.62 x 730

= 459.07

This model is known as **simple linear regression** (1 descriptive feature, 1 target feature)

This allows us to extend the model to use more features

$$\boldsymbol{target\ feature} = \boldsymbol{w_0} + \boldsymbol{w_1} * \boldsymbol{feature_1} + \boldsymbol{w_2} * \boldsymbol{feature_2} + \dots + \boldsymbol{w_n} * \boldsymbol{feature_n}$$

$$w_i = \frac{\sum(x - \bar{x_i})(y - \bar{y})}{\sum(x - \bar{x_i})^2}$$

$$w_0 = \bar{y} - \left(\sum_{i=1..n} w_i\, \bar{x_i}\right)$$

Linear Regression: Assumptions

When you want to apply the LR for your data, you should check the five key assumptions:

- Linear relationship
- Multivariate normality
- No or little multicollinearity
- No auto-correlation
- Homoscedasticity

Read more:

https://www.statisticssolutions.com/assumptions-of-linear-regression/

https://github.com/justmarkham/DAT4/blob/master/notebooks/08_linear_regression.ipynb

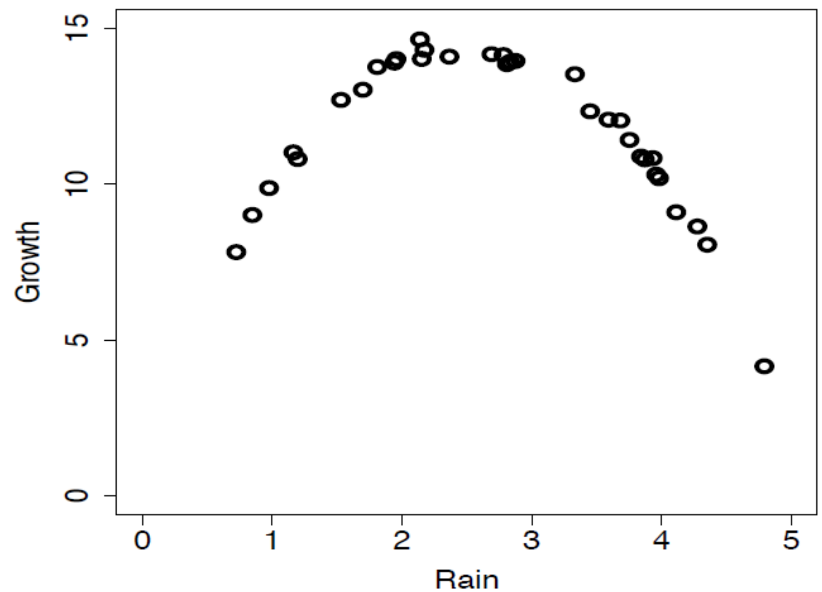
What if the relationship is not linear?

Solution1:

Create new features that capture nonlinear polynomials of original features

– E.g., original descriptive feature: RAIN. Create a new feature (quadratic polynomial): $RAIN^2$

$$GROWTH = w0 + w1*RAIN + w2*RAIN^2$$



Solution2: **Create feature interactions**

– E.g., original descriptive features: SALARY, HOUSE_PRICE. Create a new feature: the ratio of the two features SALARY/HOUSE_PRICE

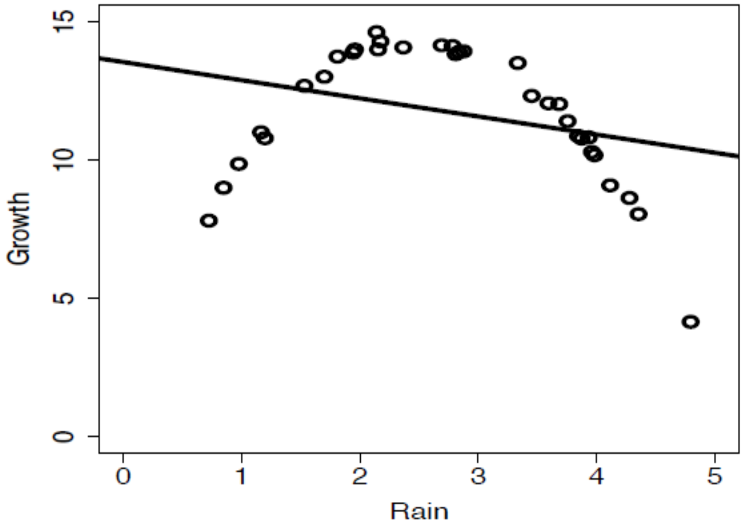
Finally: Build linear regression model with original features + new (derived) features that aim to capture non-linear behavior

What if the relationship is not linear?

A linear model using original features:

13.510 - 0.667 * RAIN

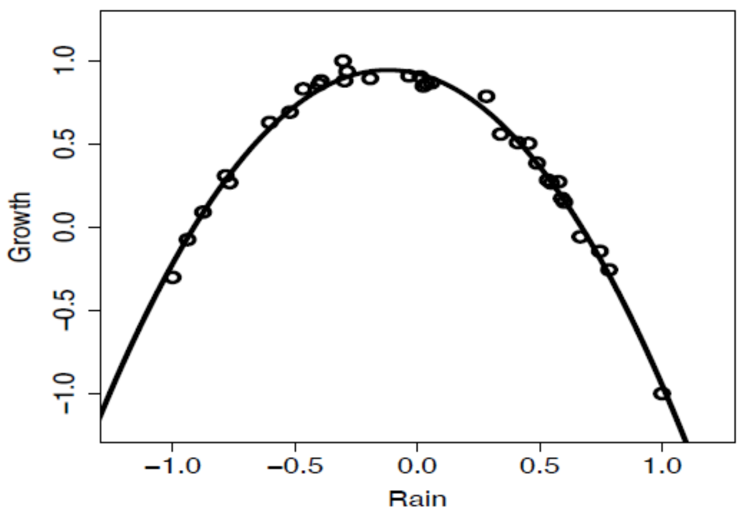
(This model has a large error!)



A linear model using original features and quadratic features:

0.3707 + 0.8475 * RAIN + 1.717 * RAIN^2

(This model fits the data better and has lower error)



Logistic Regression

- We assume only 2 classes (binary classification); can extend to many classes with one-vs-all approach

- We model the probability of class membership, e.g., if **p(PriceClass = High| Size) > 0.5**, then predict class **High**, else predict class **Low**

- We use a logistic function to make sure predictions are in the [0,1] interval (proper probabilities)

Linear regression:

$$p(PriceClass = High|\, Size) = \boldsymbol{w\_0} + \boldsymbol{w\_1} * Size$$

Logistic regression:

$$p(PriceClass = High|\, Size) = \boldsymbol{logistic}(\boldsymbol{w\_0} + \boldsymbol{w\_1} * Size)$$

Logistic Regression

logistic function

$$Logistic(x) = \frac{1}{1+e^{-x}} \qquad \mathbf{logistic(x)} = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}} \tag{8}$$

where x is a numeric value and e is **Euler's number** and is approximately equal to 2.7183.

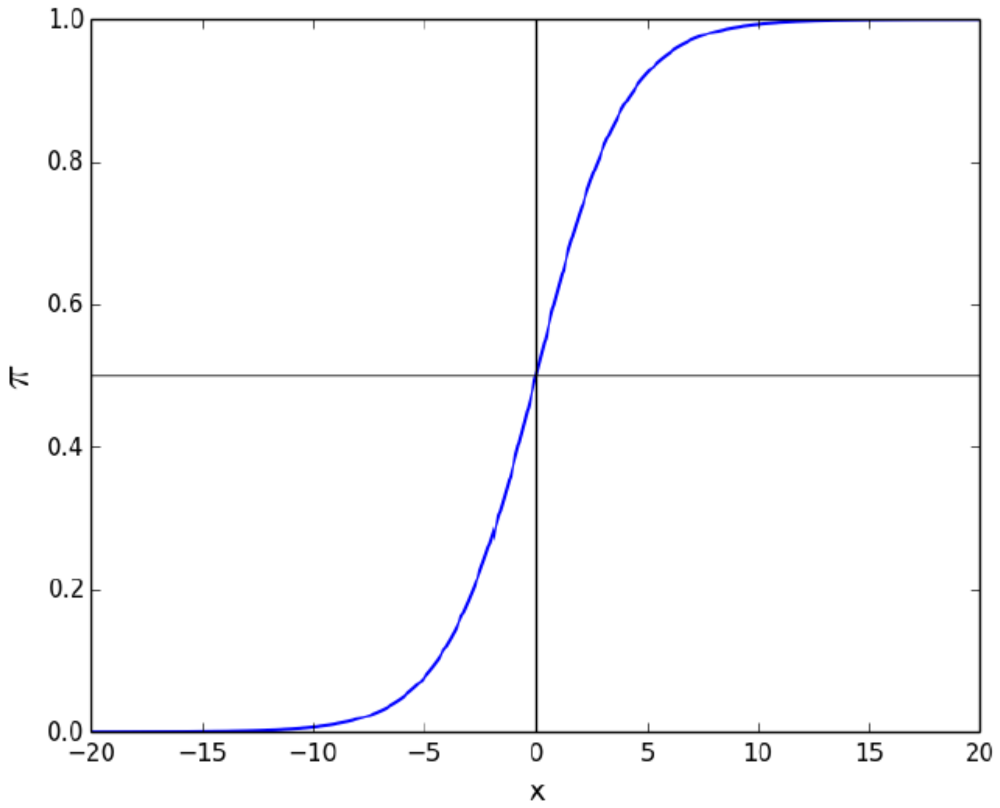
When performing logistic regression, we use the following form:

$$\pi = \Pr(y = 1 \mid x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

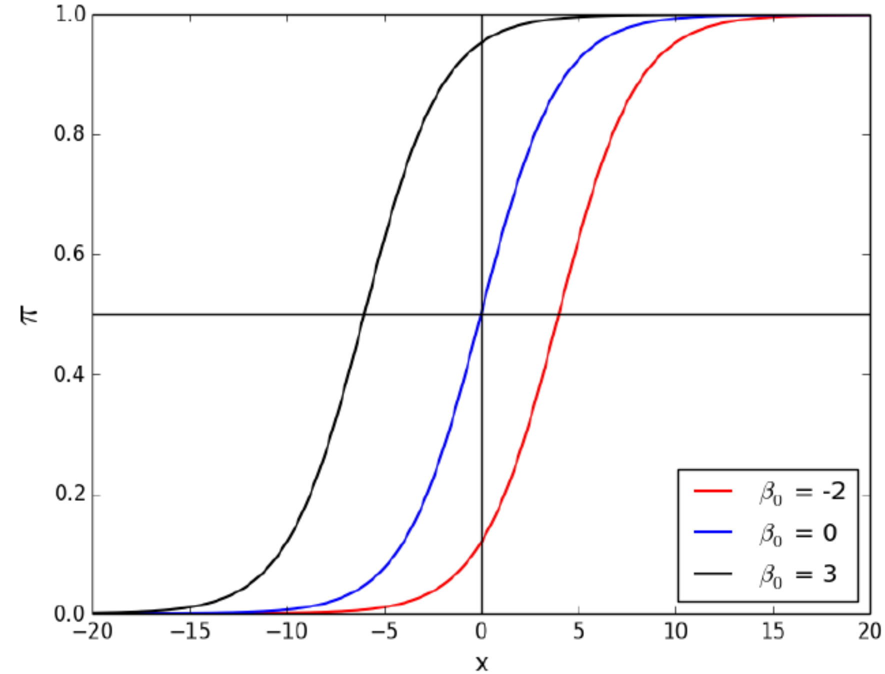
Logistic Regression

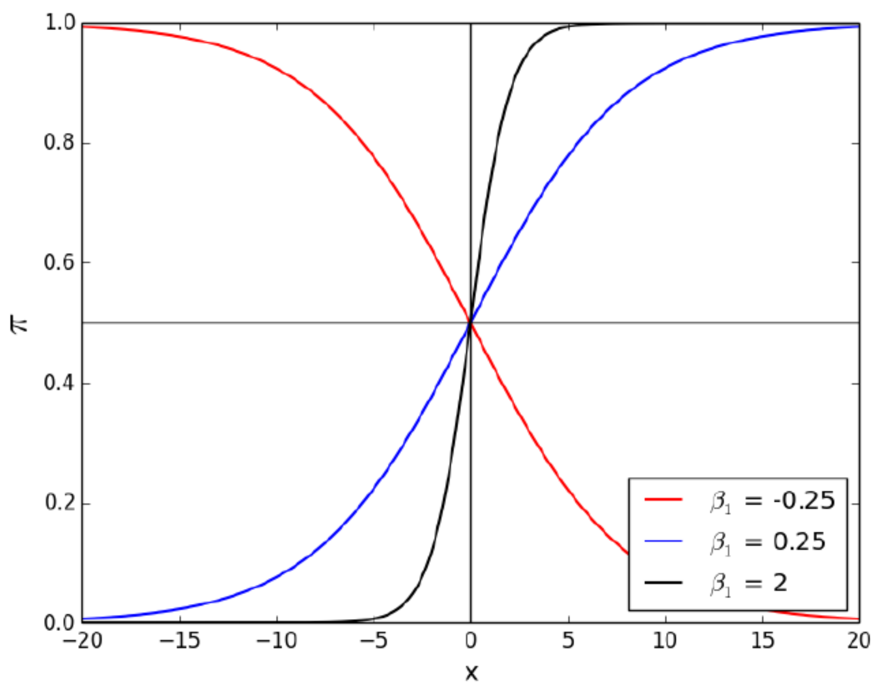
- *The logistic function takes on an "S" shape, where y is bounded by [0,1]*

$$\pi = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$



Changing the β_0 value shifts the function horizontally.





Changing the β_1 value changes the slope of the curve

Example for Text Analytics – Ham/Spam SMS

```
ham Go until jurong point, crazy.. Available only in bugis n great world la
ham Ok lar... Joking wif u oni...
spam    Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005.
ham U dun say so early hor... U c already then say...
ham Nah I don't think he goes to usf, he lives around here though
spam    FreeMsg Hey there darling it's been 3 week's now and no word back!
ham Even my brother is not like to speak with me. They treat me like aids p
ham As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)'
spam    WINNER!! As a valued network customer you have been selected to rec
spam    Had your mobile 11 months or more? U R entitled to Update to the la
ham I'm gonna be home soon and i don't want to talk about this stuff anymor
spam    SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and s
spam    URGENT! You have won a 1 week FREE membership in our £100,000 Prize
ham I've been searching for the right words to thank you for this breather.
ham I HAVE A DATE ON SUNDAY WITH WILL!!
```

Download at: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

SMS Spam Collection Data Set

Download: Data Folder, Data Set Description

Abstract: The SMS Spam Collection is a public set of SMS labeled messages that have been collected for mobile phone spam research.

Data Set Characteristics:	Multivariate, Text, Domain-Theory	**Number of Instances:**	5574	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	N/A	**Date Donated**	2012-06-22
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	200580

Example for Text Analytics – Ham/Spam SMS

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model.logistic import LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_score

df = pd.read_csv('SMSSpamCollection', delimiter='\t',header=None)

X_train_raw, X_test_raw, y_train, y_test = train_test_split(df[1],df[0])

vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train_raw)
classifier = LogisticRegression()
classifier.fit(X_train, y_train)

X_test = vectorizer.transform( ['URGENT! Your Mobile No 1234 was awarded a Prize'] )
predictions = classifier.predict(X_test)
print('URGENT! Your Mobile No 1234 was awarded a Prize',' is predicted as:', predictions)


X_test = vectorizer.transform( [ 'Hey honey, whats up?'] )
predictions = classifier.predict(X_test)
print('Hey honey, whats up?',' is predicted as:', predictions)
```

```
URGENT! Your Mobile No 1234 was awarded a Prize  is predicted as: ['spam']
Hey honey, whats up?  is predicted as: ['ham']
```

Summary

- Linear Regression - Linear Model : This is Regression model. The output will be the predicted values of Target feature.

$$y = w_0 + w_1 * x$$

where w_1 *is the slope,*
w_0 *is the bias*

- In case you require the output in [0,1], you should apply the Logistic Regression. The $\boldsymbol{logistic}()$ function will turn your output to [0,1]
  - ➢ Predicted probability for each class
  - ➢ Based on the threshold (default=0.5), predicted labels can be provided

$$\pi = \Pr(y = 1 \mid x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$