Data Science

Dr. Muhammad Adeel Nisar

Assistant Professor – Department of IT,
Faculty of Computing and Information Technology,
University of the Punjab, Lahore

Slides Courtesy: Dr. Kamran Malik

Probabilities (recap)

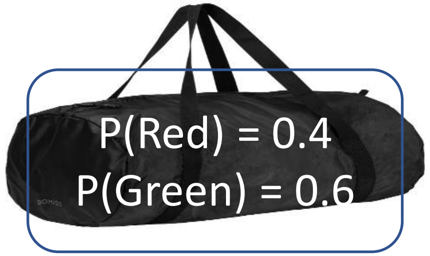
Probabilities (Recap)

- $P(X = 0) =$ 5/12
- $P(Y = 3) =$ 4/12
- $P(X = 1, Y = 2) =$ 2/12
- $P(Y = 2, X = 1) =$ 2/12
- $P(X = 1 \mid Y = 2) =$ 2/3
- $P(Y = 2 \mid X = 1) =$ 2/4

X	Y
0	0
0	1
1	0
1	2
2	3
2	0
2	3
1	3
1	2
0	3
0	2
0	0

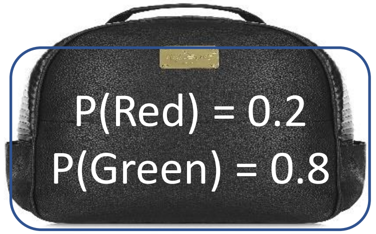
Probabilities (Recap)

P(bag1) = 0.3



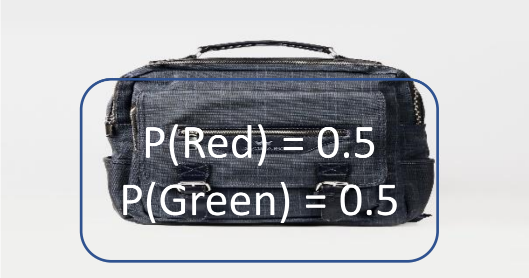
P(Red) = 0.4
P(Green) = 0.6

P(bag2) = 0.5



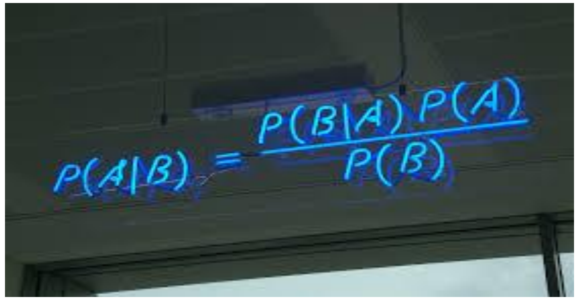
P(Red) = 0.2
P(Green) = 0.8

P(bag3)=0.2



P(Red) = 0.5
P(Green) = 0.5

- P(bag2) = 0.5
- P(Red|bag1) = 0.4
- P (Red) = 0.32
- P(bag1|Red) = $\frac{P(Red|bag1) \times P(bag1)}{P(Red)}$



Probabilities (Recap)

- $P(A, B)$ or $P(A \cap B) = ?$
- $P(A, B) = P(A \mid B) * P(B) = P(B \mid A) * P(A)$
- $P(A, B \mid C) = \cancel{P(A \mid B, C) * P(B \mid C) = P(B \mid A, C) * P(A \mid C)}$

$P(A \mid B, C) \times P(B \mid C)$

$P(C, B)$

- $P(A \cap B \cap C) = P(A, B, C) = P(A \mid B, C) \times \boxed{P(B, C)}$

$P(A \mid B, C) \times P(B \mid C) \times P(C)$

$\times P(C \mid B) \times P(B)$

$P(B, A, C) = P(B \mid A, C) \times P(A, C)$

$\times P(A \mid C) \times P(C)$

- $P(A \cap B \cap C \cap D) = ?$

$P(C, A, B)$

$P(C, B, A)$

Probabilities (Recap)

if all events are independent

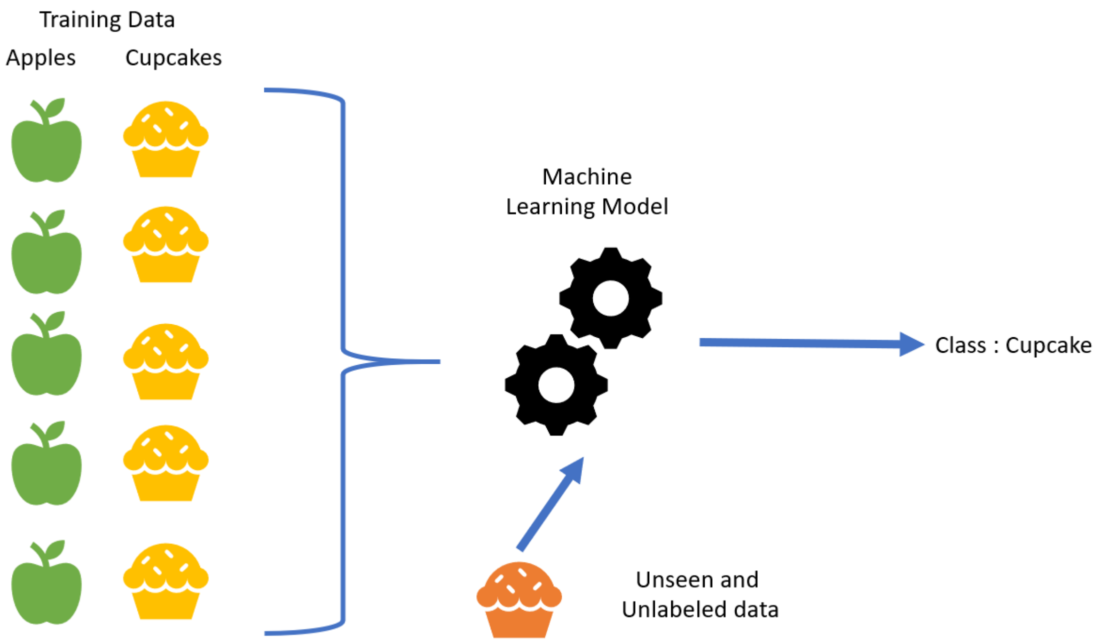
- P (A , B) = P(A) * P (B)
- P (A , B , C) = P(A) x P(B) x P(C)
- P (A , B , C, D) =
- P(A1,A2, . . ., An) =

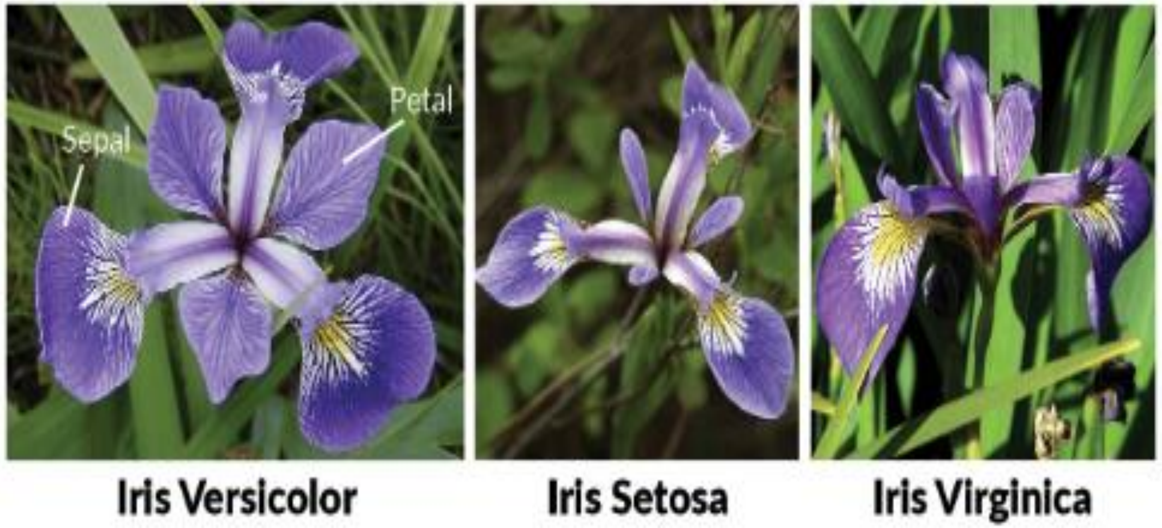
Conditionally independent

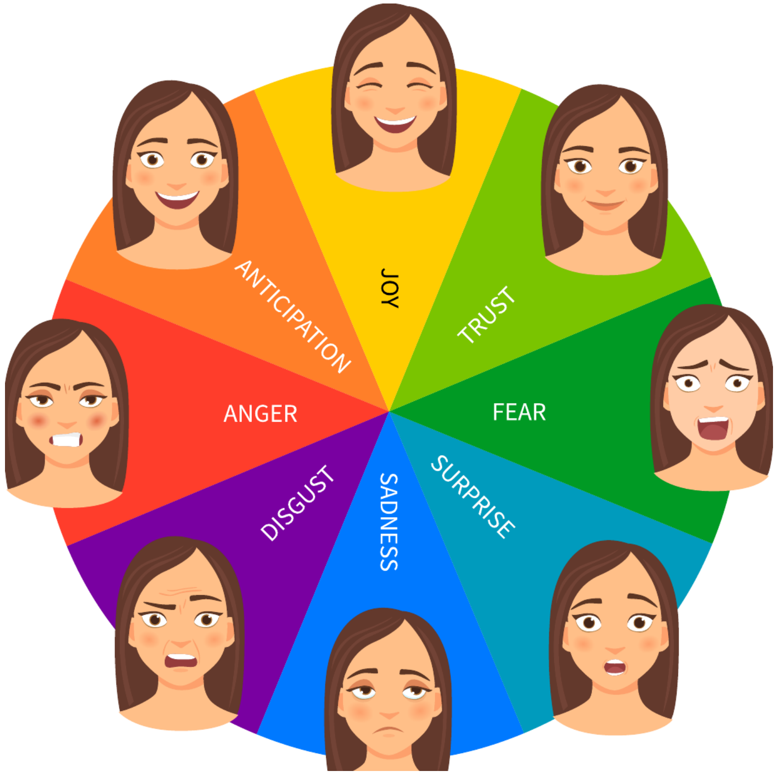
- P (A , B | C) = P(A | C) * P (B | C)
- P (A , B , C | D) = P(A|D) x P(B|D) x P(C|D)
- P(A1,A2, . . ., An | Z) = P(A1|Z) x . . . . x P(An|Z)

Classification

Classification Examples







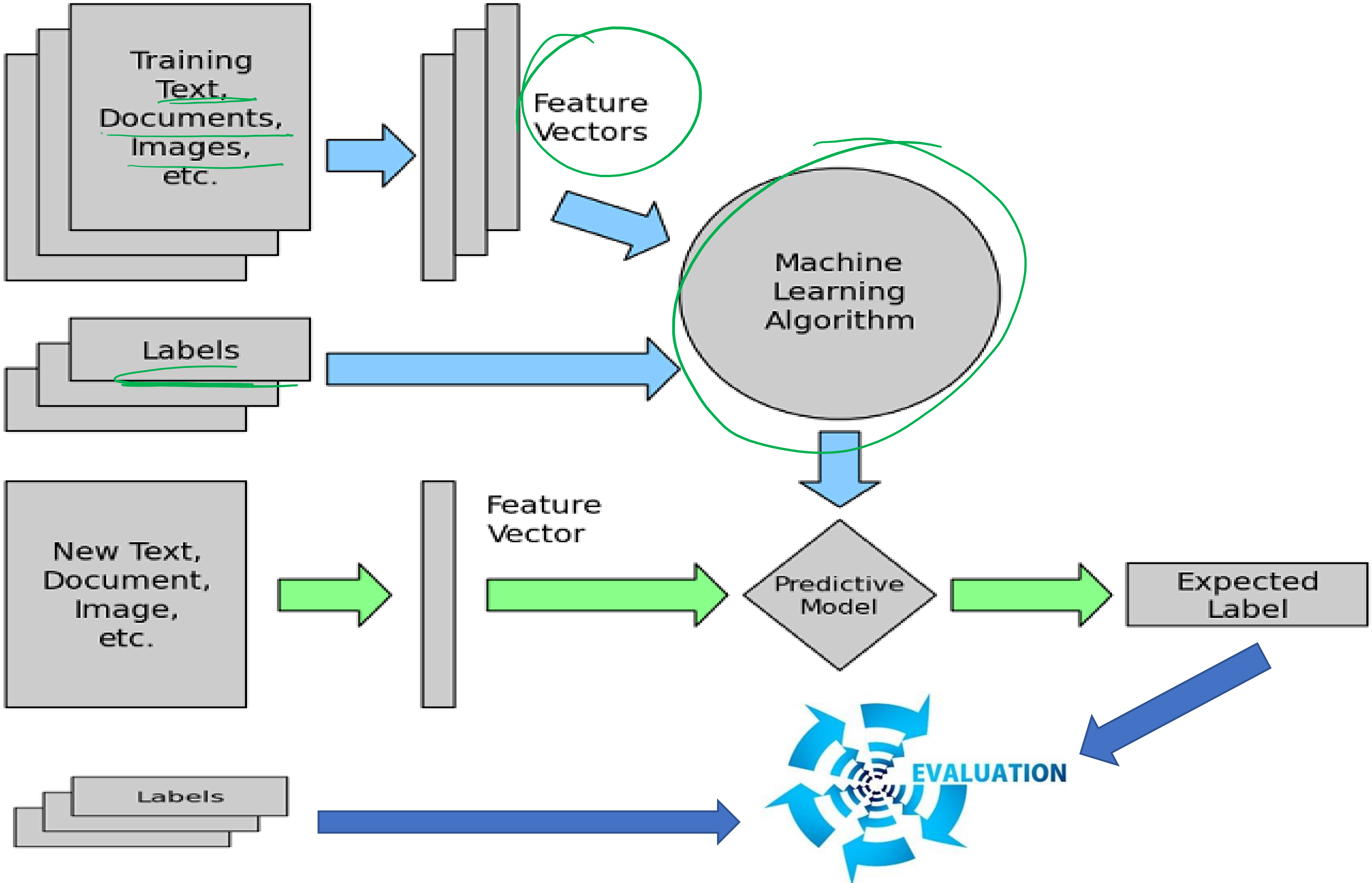
Classification Algorithms

- Classification algorithms are used when the output variable is categorical, which means there are two or more classes.
- Algorithms
  - Naïve Bayes
  - Logistic Regression
  - Support vector Machines
  - Random Forest
  - Decision Trees

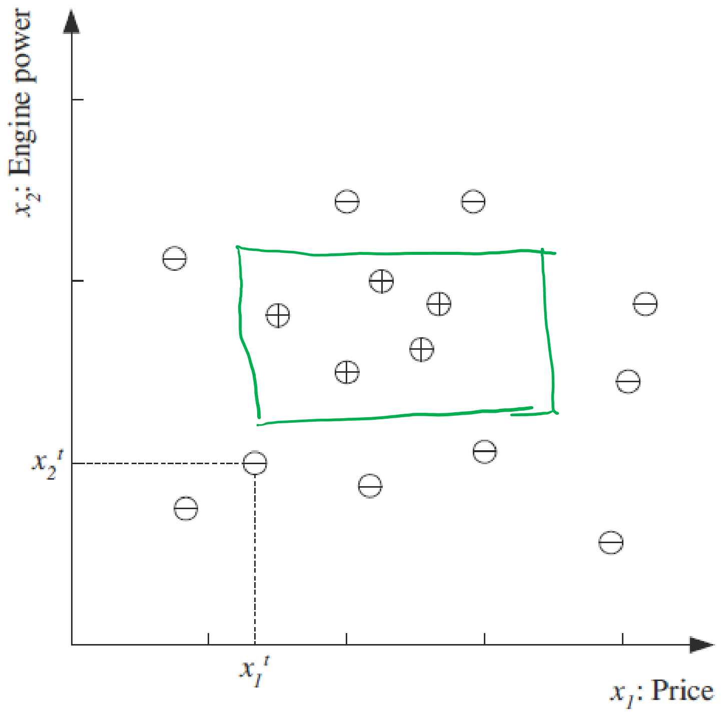
How to Perform Classification

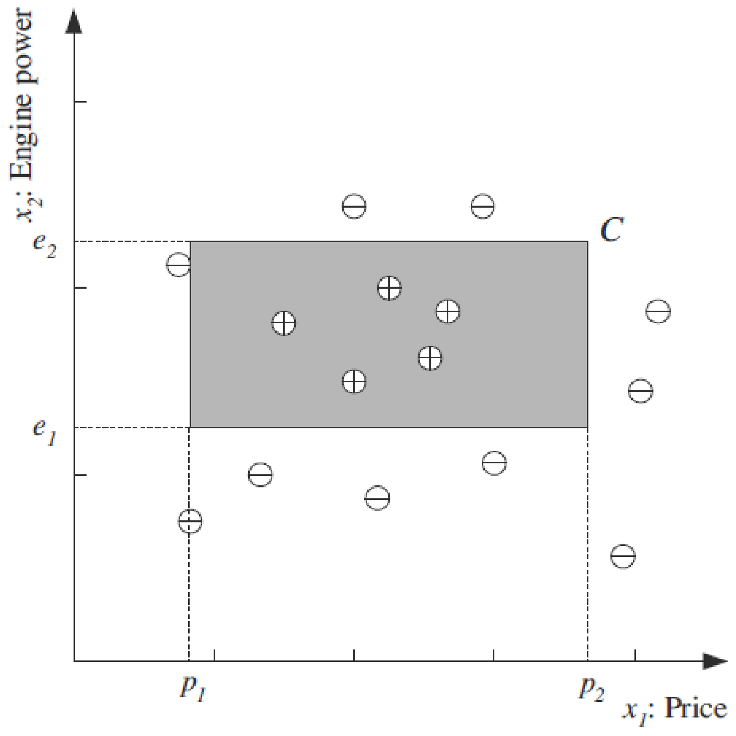
Training Data

Testing Data



Classification of a Family Car

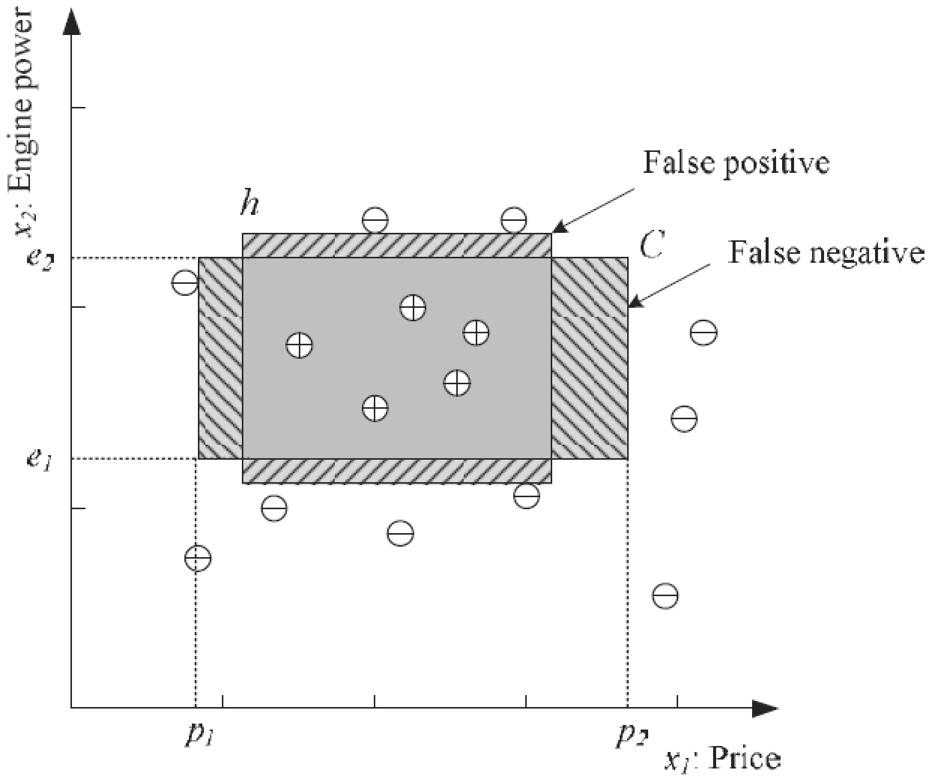


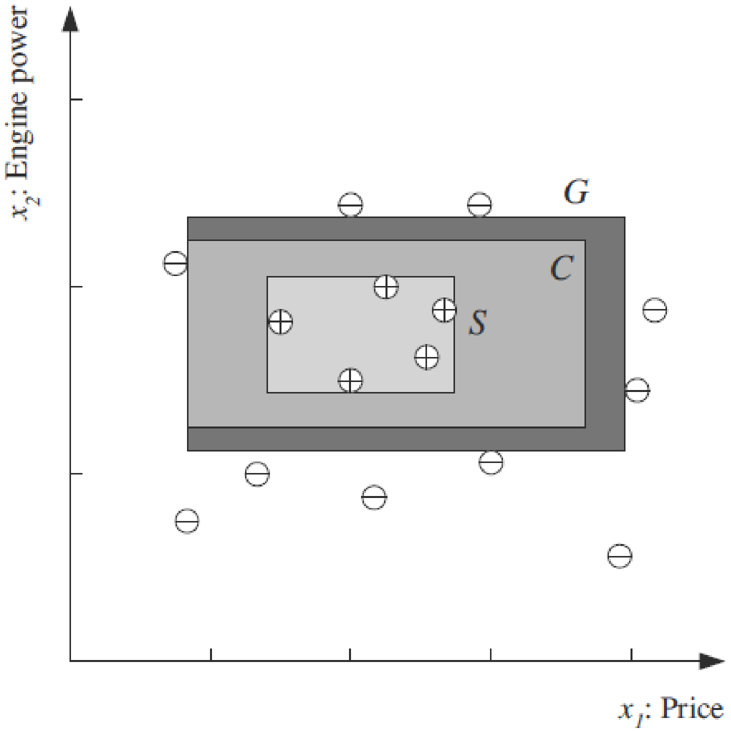


$(p_1 \leq \text{price} \leq p_2)$ AND $(e_1 \leq \text{engine power} \leq e_2)$

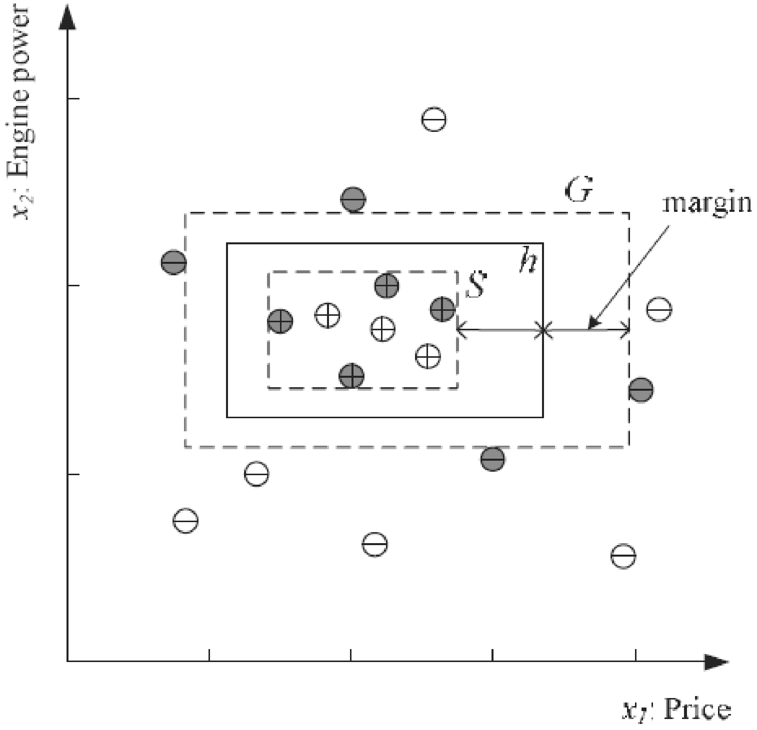
$$h(\boldsymbol{x}) = \begin{cases} 1 & \text{if } h \text{ classifies } \boldsymbol{x} \text{ as a positive example} \\ 0 & \text{if } h \text{ classifies } \boldsymbol{x} \text{ as a negative example} \end{cases}$$

$$E(h|\mathcal{X}) = \sum_{t=1}^{N} 1(h(\mathbf{x}^t) \neq r^t)$$





x_2: Engine power
G
C
S
x_1: Price



x_2: Engine power
G
margin
h
S
x_1: Price

Evaluation of a Classifier

FC : Family Car
NFC: Not a Family Car

Car	Actual	Predicted
1	FC	FC
2	FC	NFC
3	NFC	FC
4	FC	FC
5	NFC	FC
6	NFC	NFC
7	FC	NFC
8	FC	FC
9	NFC	FC
10	FC	FC
11	NFC	NFC
12	NFC	NFC

Model Evaluation Measures

- **True positives (TP):** data points predicted/labeled as positive that are actually positive
- **False positives (FP):** data points predicted/labeled as positive that are actually negative
- **True negatives (TN):** data points predicted/labeled as negative that are actually negative
- **False negatives (TN):** data points predicted/labeled as negative that are actually positive

Model Evaluation Measures

Confusion Matrix

		Actual	
		Positive (FC)	**Negative** (NFC)
Predicted	**Positive** (FC)	TP	FP
	Negative (NFC)	FN	TN

Car	Actual	Predicted
1	FC	FC
2	FC	FC
3	NFC	FC
4	FC	FC
5	NFC	FC
6	NFC	NFC
7	FC	NFC
8	FC	FC
9	FC	FC
10	FC	NFC
11	NFC	NFC
12	NFC	NFC

Model Evaluation Measures

<table>
<tr><td rowspan="2" colspan="2">Confusion Matrix</td><td colspan="2">Actual</td></tr>
<tr><td>Positive
(FC)</td><td>Negative
(NFC)</td></tr>
<tr><td rowspan="2">Predicted</td><td>Positive
(FC)</td><td>5</td><td>2</td></tr>
<tr><td>Negative
(NFC)</td><td>2</td><td>3</td></tr>
</table>

Car	Actual	Predicted
1	FC	FC
2	FC	FC
3	NFC	FC
4	FC	FC
5	NFC	FC
6	NFC	NFC
7	FC	NFC
8	FC	FC
9	FC	FC
10	FC	NFC
11	NFC	NFC
12	NFC	NFC

Model Evaluation Measures

- **Accuracy:** Closeness of a measured value to a standard or known value ((TP + TN)/(TP+TN+FN+FP)) = **8/12** = **0.67**
- **Recall:** Ability of a classification model to identify all relevant instances (TP / (TP + FN)) = **5/7**
- **Precision:** Ability of a classification model to return only relevant instances (TP / (TP + FP)) = **5/7**
- **F1 score:** A single metric that combines recall and precision using the harmonic mean (2 (Precision * Recall)/(Precision + Recall)) **= ?**

Example 2

Images	Actual Label	Predicted Label
Image1	Happy	Sad
Image2	Sad	Sad
Image3	Sad	Happy
Image4	Happy	Happy
Image5	Happy	Happy
Image6	Sad	Happy
Image7	Sad	Happy
Image8	Happy	Happy
Image9	Happy	Happy
Image10	Sad	Sad

Confusion Matrix		Actual	
		Positive	Negative
Predicted	Positive		
	Negative		

- Accuracy = ?
- Recall = ?
- Precision = ?
- F1-Score = ?

Example 3

Confusion Matrix		Actual	
		Positive	Negative
Predicted	Positive		
	Negative		

- Accuracy = ?
- Recall = ?
- Precision = ?
- F1-Score = ?

Images	Actual Label	Model 1	Model 2
Patient1	Not Corona	Not Corona	Not Corona
Patient2	Not Corona	Not Corona	Not Corona
Patient3	Not Corona	Not Corona	Not Corona
Patient4	Not Corona	Not Corona	Not Corona
Patient5	Not Corona	Not Corona	Not Corona
Patient6	Not Corona	Not Corona	Corona
Patient7	Not Corona	Not Corona	Corona
Patient8	Not Corona	Not Corona	Corona
Patient9	Corona	Not Corona	Corona
Patient10	Corona	Not Corona	Corona

Naïve Bayes Classifier

(Classification Algorithms)

Naïve Bayes Classifier

$$P(x \mid c)$$

$$P(x_1, x_2, \ldots, x_n \mid c)$$

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Naïve Bayes Classifier

- Assumption

Conditional Independence: Assume the feature probabilities x are independent given the class *c.*

$$P(x \mid c)$$

$$P(x_1, x_2, \ldots, x_n \mid c)$$

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Naïve Bayes Classifier

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Naïve Bayes Classifier

Bayes Rule: $$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid x)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(x \mid c) P(c)}{P(x)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Dropping the denominator

Naïve Bayes Classifier

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = P(n) =

P(x1=1|y) = P(x1=0|y)=

P(x1=1|n) = P(x1=0|n) =

P(x2=1|y) = P(x2=0|y) =

P(x2=1|n) = P(x2=0|n) =

P(x3=1|y) = P(x3=0|y) =

P(x3=1|n) = P(x3=0|n) =

P(x4=1|y) = P(x4=0|y) =

P(x4=1|n) = P(x4=0|n) =

P(x5=1|y) = P(x5=0|y) =

P(x5=1|n) = P(x5=0|n) =

P(x6=1|y) = P(x6=0|y) =

P(x6=1|n) = P(x6=0|n) =

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Naïve Bayes Classifier

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = 1/2 P(n) = 1/2

P(x1=1|y) =2/3 P(x1=0|y) =1/3

P(x1=1|n) = 1/3 P(x1=0|n) = 2/3

P(x2=1|y) = 2/3 P(x2=0|y) = 1/3

P(x2=1|n) = 2/3 P(x2=0|n) = 1/3

P(x3=1|y) = 0 P(x3=0|y) = 1

P(x3=1|n) = 1/3 P(x3=0|n) = 2/3

P(x4=1|y) = 2/3 P(x4=0|y) = 1/3

P(x4=1|n) = 2/3 P(x4=0|n) = 1/3

P(x5=1|y) = 2/3 P(x5=0|y) = 1/3

P(x5=1|n) = 2/3 P(x5=0|n) = 1/3

P(x6=1|y) = 2/3 P(x6=0|y) = 1/3

P(x6=1|n) = 1/3 P(x6=0|n) = 2/3

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Naïve Bayes Classifier

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = 1/2 P(n) = 1/2

P(x1=1|y) =2/3 P(x1=0|y) =1/3

P(x1=1|n) = 1/3 P(x1=0|n) = 2/3

P(x2=1|y) = 2/3 P(x2=0|y) = 1/3

P(x2=1|n) = 2/3 P(x2=0|n) = 1/3

P(x3=1|y) = 0 P(x3=0|y) = 1

P(x3=1|n) = 1/3 P(x3=0|n) = 2/3

P(x4=1|y) = 2/3 P(x4=0|y) = 1/3

P(x4=1|n) = 2/3 P(x4=0|n) = 1/3

P(x5=1|y) = 2/3 P(x5=0|y) = 1/3

P(x5=1|n) = 2/3 P(x5=0|n) = 1/3

P(x6=1|y) = 2/3 P(x6=0|y) = 1/3

P(x6=1|n) = 1/3 P(x6=0|n) = 2/3

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Naïve Bayes Classifier (smoothing)

Naïve Bayes Classifier (Smoothing)

- A solution would be **Laplace smoothing** , which is a technique for smoothing categorical data.
- A small-sample correction, or **pseudo-count**, will be incorporated in every probability estimate.
- Consequently, no probability will be zero.
- This is a way of regularizing Naive Bayes, and when the pseudo-count is zero, it is called Laplace smoothing.
- While in the general case it is often called **Lidstone smoothing**.

Naïve Bayes Classifier (after Smoothing)

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = P(n) =

P(x1=1|y) = P(x1=0|y)=

P(x1=1|n) = P(x1=0|n) =

P(x2=1|y) = P(x2=0|y) =

P(x2=1|n) = P(x2=0|n) =

P(x3=1|y) = P(x3=0|y) =

P(x3=1|n) = P(x3=0|n) =

P(x4=1|y) = P(x4=0|y) =

P(x4=1|n) = P(x4=0|n) =

P(x5=1|y) = P(x5=0|y) =

P(x5=1|n) = P(x5=0|n) =

P(x6=1|y) = P(x6=0|y) =

P(x6=1|n) = P(x6=0|n) =

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Naïve Bayes Classifier (Smoothing)

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = 1/2 P(n) = 1/2

P(x1=1|y) =3/5 P(x1=0|y) =2/3

P(x1=1|n) = 2/5 P(x1=0|n) = 3/5

P(x2=1|y) = 3/5 P(x2=0|y) = 2/5

P(x2=1|n) = 3/5 P(x2=0|n) = 2/5

P(x3=1|y) = 1/5 P(x3=0|y) = 4/5

P(x3=1|n) = 2/5 P(x3=0|n) = 3/5

P(x4=1|y) = 3/5 P(x4=0|y) = 2/5

P(x4=1|n) = 3/5 P(x4=0|n) = 2/5

P(x5=1|y) = 3/5 P(x5=0|y) = 2/5

P(x5=1|n) = 3/5 P(x5=0|n) = 2/5

P(x6=1|y) = 3/5 P(x6=0|y) = 2/5

P(x6=1|n) = 2/5 P(x6=0|n) = 3/5

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Programming Assignment (example)

Naïve Bayes Classifier

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = P(n) =

P(x1=1|y) = P(x1=0|y)=

P(x1=1|n) = P(x1=0|n) =

P(x2=1|y) = P(x2=0|y) =

P(x2=1|n) = P(x2=0|n) =

P(x3=1|y) = P(x3=0|y) =

P(x3=1|n) = P(x3=0|n) =

P(x4=1|y) = P(x4=0|y) =

P(x4=1|n) = P(x4=0|n) =

P(x5=1|y) = P(x5=0|y) =

P(x5=1|n) = P(x5=0|n) =

P(x6=1|y) = P(x6=0|y) =

P(x6=1|n) = P(x6=0|n) =

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Naïve Bayes Classifier

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

```
X_Train = np.array([[1,0,0,1,1,1],
                    [1,1,0,1,1,1],
                    [0,1,0,0,0,0],
                    [0,0,1,0,0,1],
                    [1,1,0,1,1,0],
                    [0,1,0,1,1,0]])
print(X_Train)
```

```
yes_train = X_Train[:3,:]
print(yes_train)
```

```
[[1 0 0 1 1 1]
 [1 1 0 1 1 1]
 [0 1 0 0 0 0]]
```

```
yes_train.shape
```

```
yes_one_Prob = yes_train.sum(axis=0)/yes_train.shape[0]
print(yes_one_Prob)
```

```
[0.66666667 0.66666667 0.         0.66666667 0.66666667 0.66666667]
```

```
yes_zero_Prob = 1 - yes_one_Prob
print(yes_zero_Prob)
```

```
[0.33333333 0.33333333 1.         0.33333333 0.33333333 0.33333333]
```

Naïve Bayes Classifier

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

P(y) = 1/2　　P(n) = 1/2

P(x1=1|y) =2/3　　P(x1=0|y) =1/3

P(x1=1|n) = 1/3　　P(x1=0|n) = 2/3

P(x2=1|y) = 2/3　　P(x2=0|y) = 1/3

P(x2=1|n) = 2/3　　P(x2=0|n) = 1/3

P(x3=1|y) = 0　　P(x3=0|y) = 1

P(x3=1|n) = 1/3　　P(x3=0|n) = 2/3

P(x4=1|y) = 2/3　　P(x4=0|y) = 1/3

P(x4=1|n) = 2/3　　P(x4=0|n) = 1/3

P(x5=1|y) = 2/3　　P(x5=0|y) = 1/3

P(x5=1|n) = 2/3　　P(x5=0|n) = 1/3

P(x6=1|y) = 2/3　　P(x6=0|y) = 1/3

P(x6=1|n) = 1/3　　P(x6=0|n) = 2/3

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} P(x \mid c) P(c)$$

Programming Assignment (OCR)

Programming Assignment

- Implement an OCR system that distinguishes between the images of digits
- Four files (trainX, trainY, testX and testY)

```
• from keras.datasets import mnist
• (x_train, y_train), (x_test, y_test) = mnist.load_data()
• x_train=x_train.reshape(60000,784)
• x_test=x_test.reshape(10000,784)
```

Programming Assignment

```python
%matplotlib inline
import numpy as np
from matplotlib import pyplot as plt
```

```python
img2=np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,
               0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,
               0,0,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,
               0,0,0,0,0,0,0,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,
               1,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0,0,0,0,0,0,0,1,0,
               0,0,1,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,
               0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
               0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
               0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
               0,0,0,0],dtype=np.uint8)
```

Programming Assignment

```
1  img2.shape
```

```
(256,)
```

```
img = np.reshape(img2,(16,16),order='F')
```

```
1  img.shape
```

```
(16, 16)
```

```
1  plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x5cc1780>
```