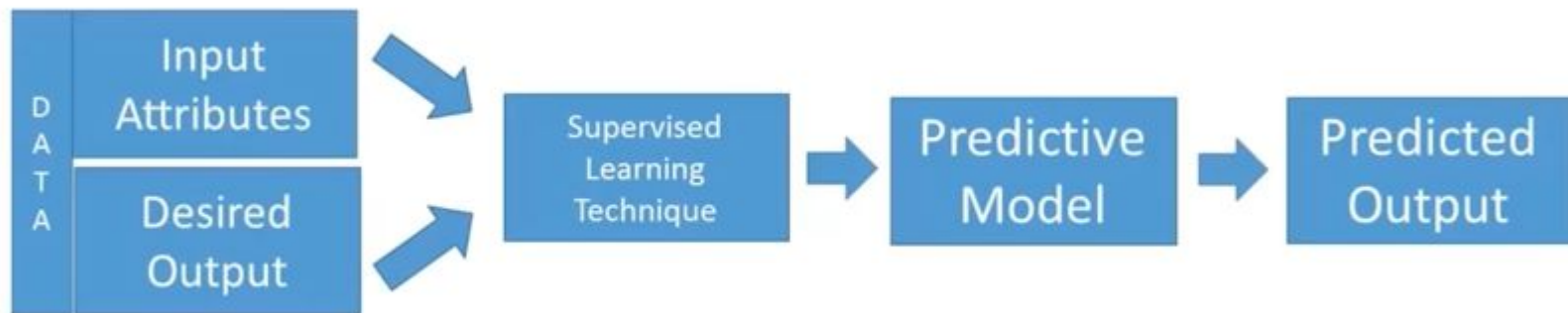

Lecture 7: *Classification*

AI/ML Foundation Course
with Python

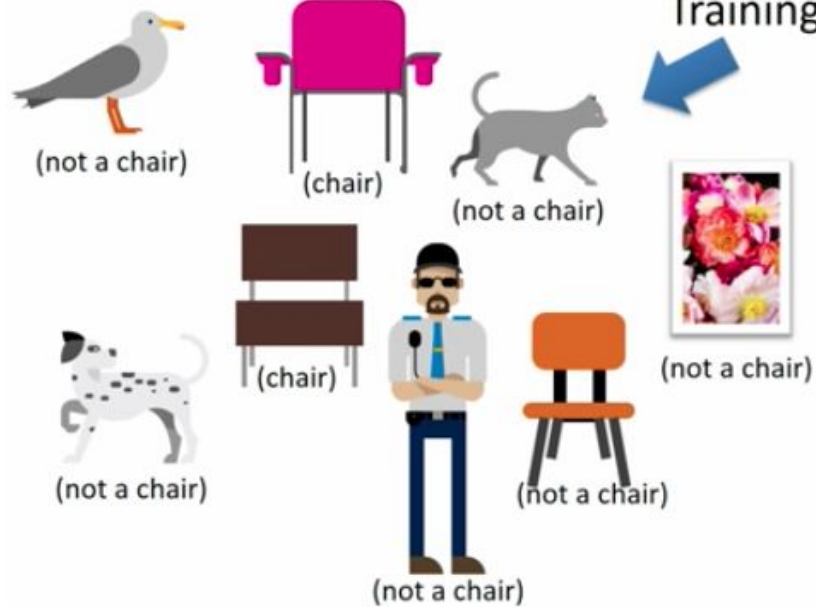
Copyright © 2018 Ankita Sinha. All rights reserved



Logistic Regression

A form of regression that allows the prediction of discrete variables by a mix of continuous and discrete predictors.

Training set



Test set



Logistic Regression

$$Y = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + E$$

Y = Dependent Variables

b_0 = Constant

b_i = Coefficient of Variable 1

X_1 = Independent variables

E = Error Term



Binary Logistic Regression

It is used when the dependent variable is dichotomous.



Multinomial Logistic Regression

Nominal (Google, Yahoo, CNN)

Ordinal (1,2,3,4,5,6,7,8,9)

It is used when the dependent or outcome variable has more than two categories.

When ? Why ?

- When the dependent variable has only two levels.
- Yes/No
- Male/Female
- Taken/Not Taken

Examples

- A catalog company wants to increase the proportion of mailings that result in sales.
- A doctor wants to accurately diagnose a possible cancerous tumor.
- A loan officer wants to know whether the next customer is likely to default.

Sample Size

1. Very small samples have much sampling errors.
2. Large samples decrease the chances of error.
3. Recommended sample size > 400
4. Recommended sample size for each group > 20


Assumptions

1. No assumptions about the distributions of the predictor variables.
2. Predictors do not have to be normally distributed.
3. Predictors variables do not have to be linearly related.
4. Predictors does not have to have equal variances within each group.

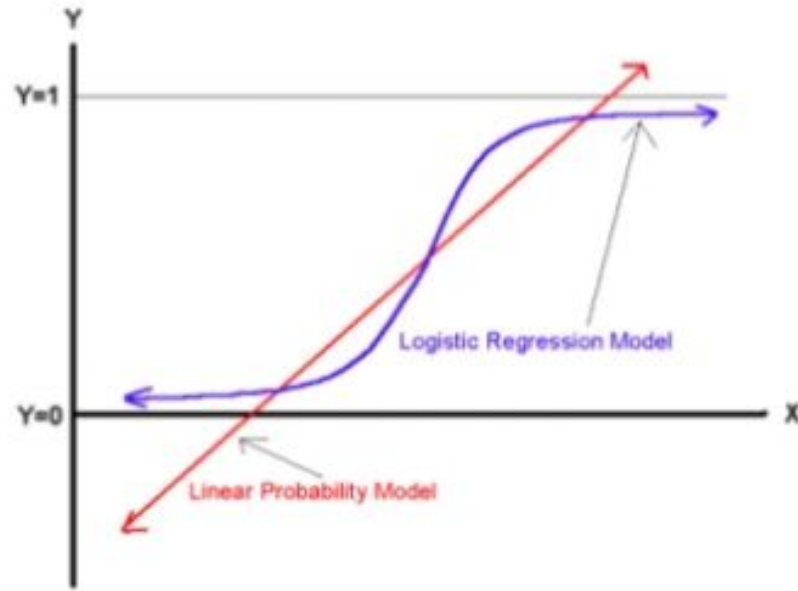


1. Measuring the Probability of Outcome

→ **The Probability of the outcome is measured by the odds of occurrence of an event.**

- 
- **P** ⇒ Probability of an Event
 - **$1-P$** ⇒ Probability of it not occurring
 - Odds of Success = **$P/1-P$**

Linear Model vs Logistic Model



Question : Determine whether household income and monthly mortgage will predict taking or decline the offer

Independent Variable : Household income and monthly mortgage

Dependent Variable: Take the offer or decline the offer.

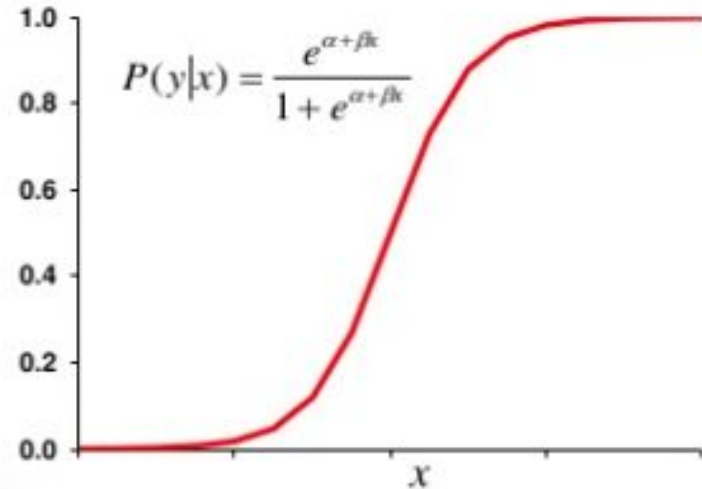
Logistic Function

A model that predicts probabilities between 0 and 1.

S-shaped Model

B = Log odds ratio

$$\log_e[P/(1-P)] = \beta_0 + \beta_1 X$$



Logistic Function

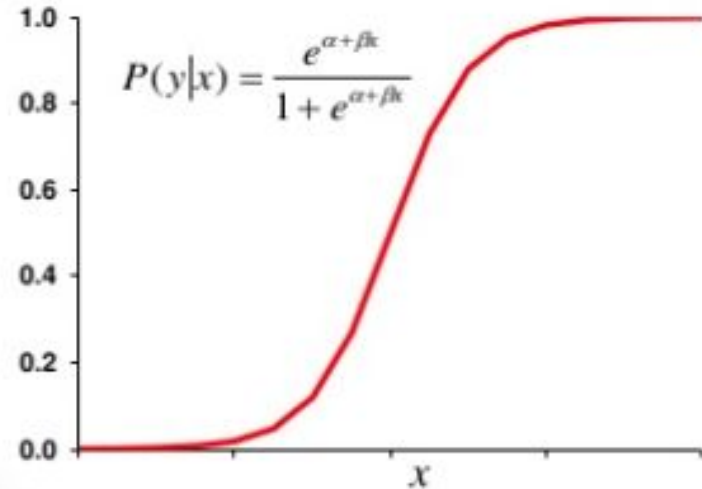
A model that predicts probabilities between 0 and 1.

S-shaped Model

β = log odds ratio associated with predictors

e^{β} = odds ratio

$$\ln\left(\frac{p}{1-p}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_i x_i$$



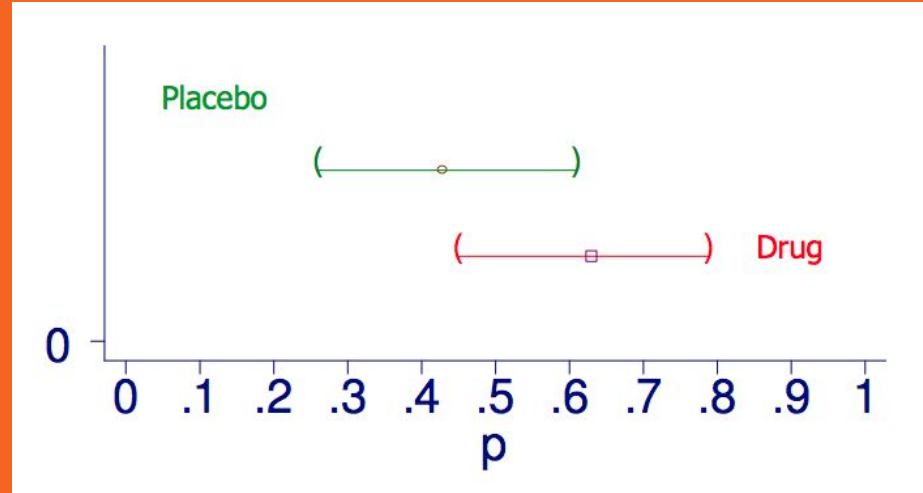
Question:

→ Whether a new drug is better than a placebo?

Relief?	Drug	Placebo
No	13	20
Yes	22	15
Total	35	35

Question:

→ Confidence Interval



Question:

→ Odds Ratio

$$\frac{\text{odds(Relief | Drug)}}{\text{odds(Relief | Placebo)}}$$

Question:

→ Odds Ratio

$$\frac{P(\text{Relief} \mid \text{Drug}) / [1 - P(\text{Relief} \mid \text{Drug})]}{P(\text{Relief} \mid \text{Placebo}) / [1 - P(\text{Relief} \mid \text{Placebo})]}$$

Question:

→ Odds Ratio

$$\frac{0.63/(1 - 0.63)}{0.45/(1 - 0.45)} = 2.26$$

The Drug is an estimated 2 ¼ times better than the placebo.

Question:

- Log Odds (Beta coefficients for the predictors)
- $T_x = 0$ if placebo
- $T_x = 1$ if drug

$$\log[\text{odds}(\text{Relief} | T_x)] = \log\left(\frac{P(\text{relief} | T_x)}{P(\text{no relief} | T_x)}\right) = \beta_0 + \beta_1 T_x$$

Question:

- Log Odds (Beta coefficients for the predictors)
- $T_x = 0$ if placebo
- $T_x = 1$ if drug

$$\log(\text{odds}(\text{Relief}|\text{Drug})) = \beta_0 + \beta_1$$

$$\log(\text{odds}(\text{Relief}|\text{Placebo})) = \beta_0$$

$$\log(\text{odds}(\text{Relief}|D)) - \log(\text{odds}(\text{Relief}|P)) = \beta_1$$

Question:

- Log Odds (Beta coefficients for the predictors)
- $T_x = 0$ if placebo
- $T_x = 1$ if drug

$$\log\left(\frac{\text{odds}(R | D)}{\text{odds}(R | P)}\right) = \beta_1$$

Question:

- Log Odds (Beta coefficients for the predictors)

$$\text{OR} = \exp(\beta_1) = e^{\beta_1}$$

So: $\exp(\beta_1)$ = odds ratio of relief for patients taking the Drug-vs-patients taking the Placebo.

Question:

$$\log \left(\frac{\Pr(Y = 1)}{\Pr(Y = 0)} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_r X_r$$

$$p/(1-p) = \exp(\beta_0 + \beta_1 X)$$

$$p = (1 - p) \cdot \exp(\beta_0 + \beta_1 X)$$

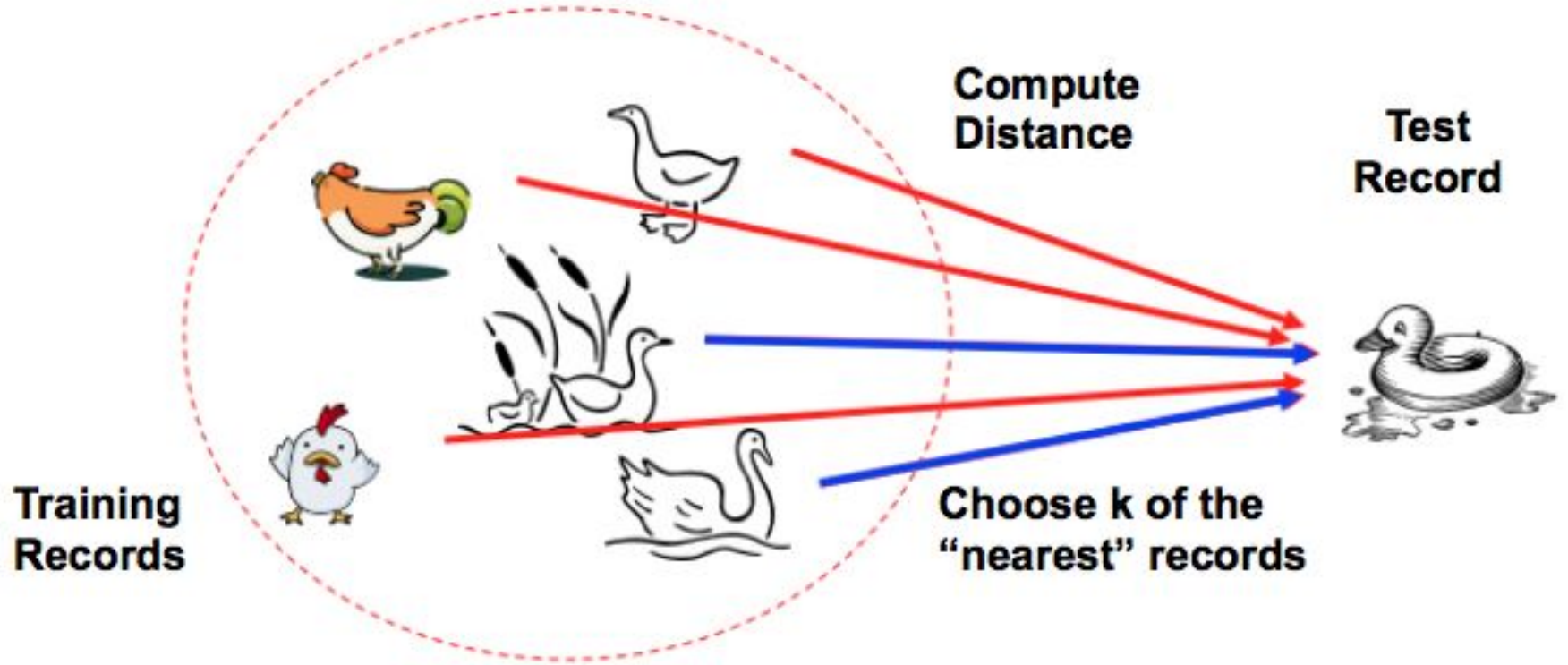
$$p = \exp(\beta_0 + \beta_1 X) - p \cdot \exp(\beta_0 + \beta_1 X)$$

$$p + p \cdot \exp(\beta_0 + \beta_1 X) = \exp(\beta_0 + \beta_1 X)$$

$$p \cdot \{1 + \exp(\beta_0 + \beta_1 X)\} = \exp(\beta_0 + \beta_1 X)$$

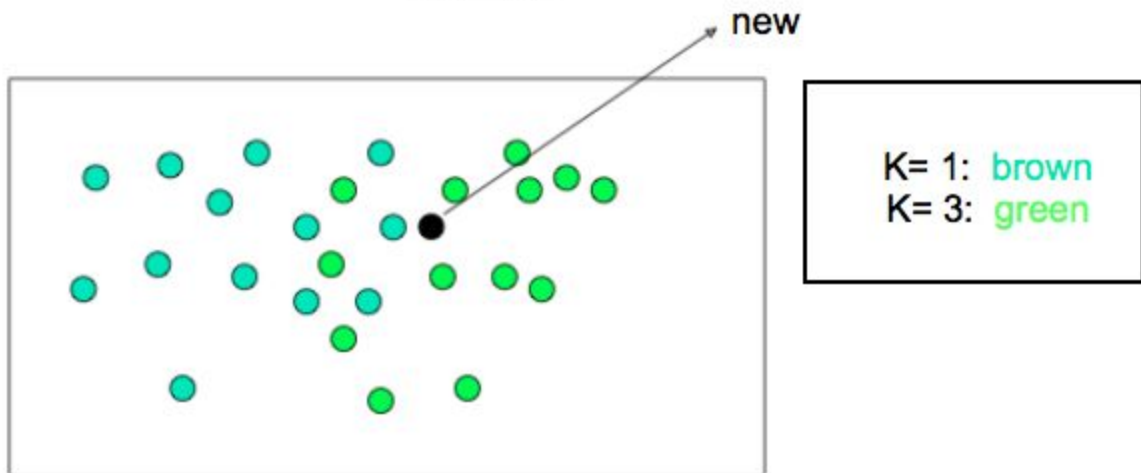
$$p = \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)}$$

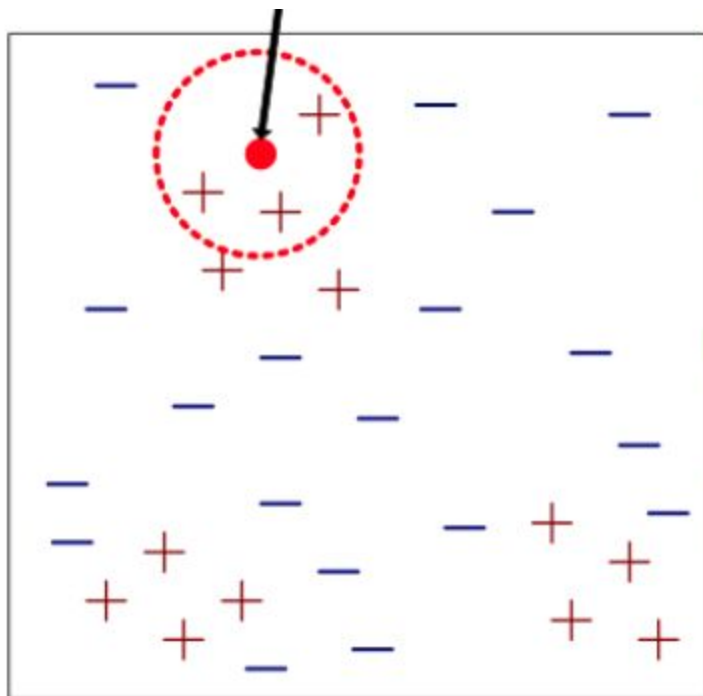
K-Nearest Neighbour



Majority vote within the k nearest neighbors

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$





The set of stored records

Distance Metric to compute
distance between records

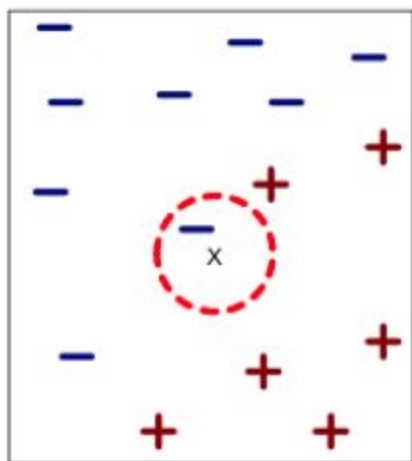
The value of k , the number of
nearest neighbors to retrieve

To classify an unknown record:

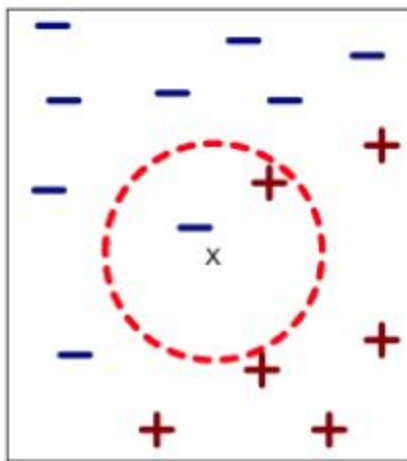
Compute distance to other
training records

Identify k nearest neighbors

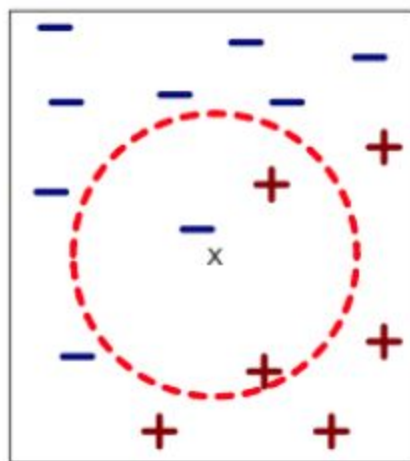
Use class labels of nearest
neighbors to determine the
class label of unknown record
(e.g., by taking majority vote)



(a) 1-nearest neighbor



(b) 2-nearest neighbor



(c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Compute distance between two points:

Euclidean distance

$$d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$$

Determine the class from nearest neighbor list

take the majority vote of class labels among the k-nearest neighbors

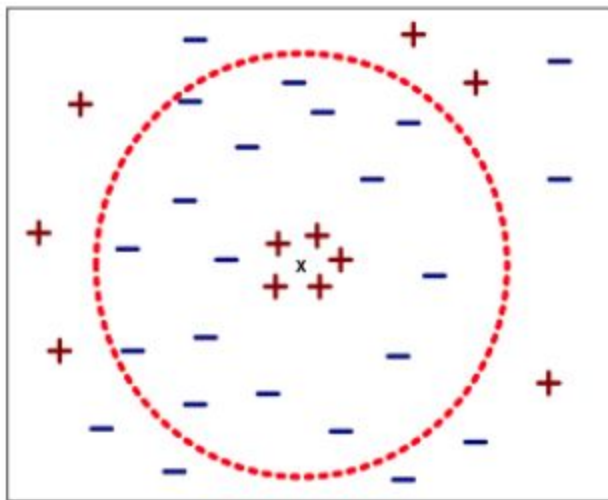
Weigh the vote according to distance

weight factor, $w = 1/d^2$

Choosing the value of k :

If k is too small, sensitive to noise points

If k is too large, neighborhood may include points from other classes



Confusion Matrix:

ACTUAL CLASS	PREDICTED CLASS	
	Class=Yes	Class=No
Class=Yes	a	b
	c	d

- a: TP (true positive)
- b: FN (false negative)
- c: FP (false positive)
- d: TN (true negative)

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Cost Matrix

ACTUAL CLASS	PREDICTED CLASS		
	$C(i j)$	Class=Yes	Class=No
	Class=Yes	$C(\text{Yes} \text{Yes})$	$C(\text{No} \text{Yes})$
	Class=No	$C(\text{Yes} \text{No})$	$C(\text{No} \text{No})$

$C(i|j)$: Cost of misclassifying class j example as class i

	PREDICTED CLASS		
ACTUAL CLASS	$C(i j)$	Class=Yes	Class=No
	Class=Yes	$C(\text{Yes} \text{Yes})$	$C(\text{No} \text{Yes})$
	Class=No	$C(\text{Yes} \text{No})$	$C(\text{No} \text{No})$

$C(i|j)$: Cost of misclassifying class j example as class i

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS	C(i j)	+	-
	+	-1	100
	-	1	0

Model M1	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%
Cost = 3910

Model M2	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%
Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS		
		Class=Yes	Class=No
	ACTUAL CLASS		
	Class=Yes	a	b
	Class=No	c	d

Cost	PREDICTED CLASS		
		Class=Yes	Class=No
	ACTUAL CLASS		
	Class=Yes	p	q
	Class=No	q	p

Accuracy is proportional to cost if

1. $C(\text{Yes}|\text{No})=C(\text{No}|\text{Yes}) = q$

2. $C(\text{Yes}|\text{Yes})=C(\text{No}|\text{No}) = p$

$$N = a + b + c + d$$

$$\text{Accuracy} = (a + d)/N$$

$$\begin{aligned}\text{Cost} &= p(a + d) + q(b + c) \\ &= p(a + d) + q(N - a - d) \\ &= qN - (q - p)(a + d) \\ &= N[q - (q - p) \text{ Accuracy}]\end{aligned}$$

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a + c}$$

$$\text{Recall (r)} = \frac{a}{a + b}$$

$$\text{F - measure (F)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}$$

Precision is biased towards $C(\text{Yes}|\text{Yes})$ & $C(\text{Yes}|\text{No})$

Recall is biased towards $C(\text{Yes}|\text{Yes})$ & $C(\text{No}|\text{Yes})$

F-measure is biased towards all except $C(\text{No}|\text{No})$

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

True Positive Rate (TPR) (Sensitivity)

$$a/a+b$$

True Negative Rate (TNR) (Specificity)

$$d/c+d$$

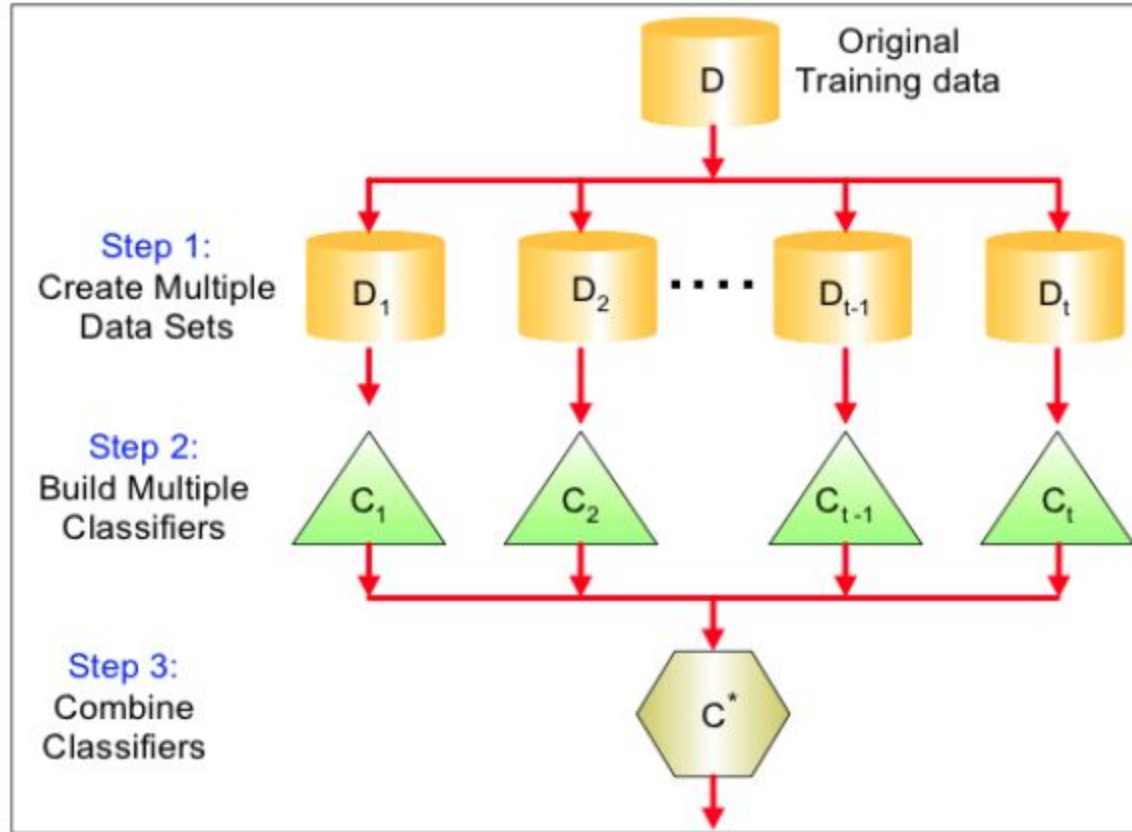
False Positive Rate (FPR)

$$c/c+d$$

False Negative Rate (FNR)

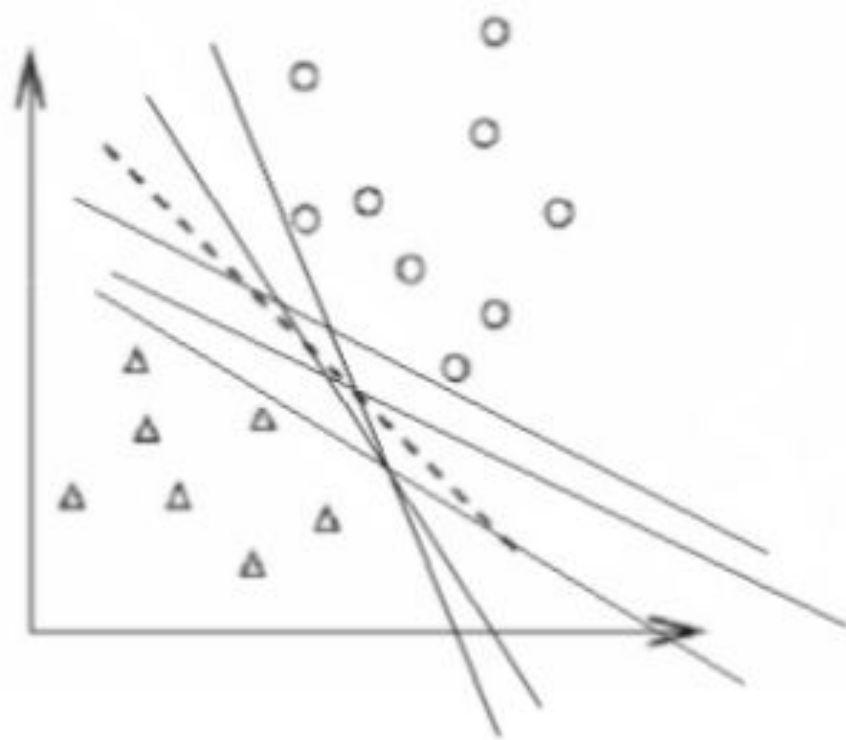
$$b/a+b$$

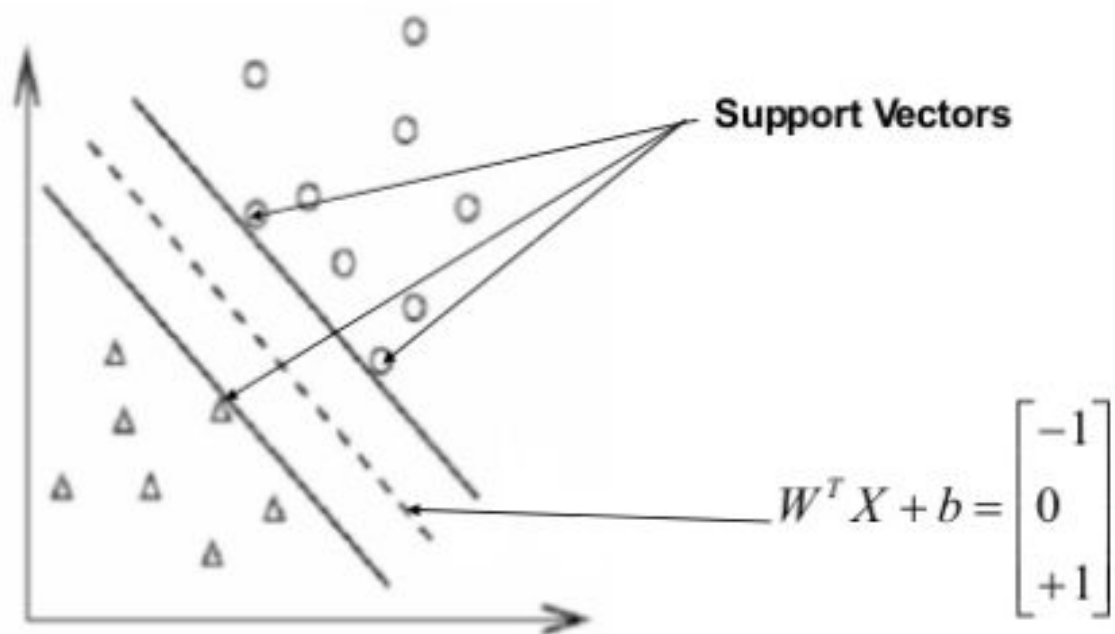
Ensemble Methods



Support Vector Machines

- **Currently considered one of the most efficient family of algorithms**
- **No Overfitting**
- **Computationally Efficient**





Cross Validation

- Split data into three sets
 - ◆ Training Set
 - ◆ Test Set
 - ◆ Validation/Hold-out Set
- Train classifiers with different parameters on training set
- Select the one with the smallest prediction error to check the test set.
- Estimate Performance on validation set.

Summary: Cross Validation

- Cross Validation selects model and assess classifier based on predicted estimates.**
- Each prediction estimates are obtained from separate training data sets.**
- A separate validation set is used for evaluating performance.**

K-fold Cross Validation

- Remove validation set and set it aside.
- Subdivide remaining data into k equally sized blocks.
- Cross validate for $k=1, \dots, K$
 - ◆ Remove block k from training data
 - ◆ Train classifier on remaining blocks
 - ◆ Estimate prediction error on block k
 - ◆ Select best classifier
- Once classifier is chosen, estimate its performance on validation set.

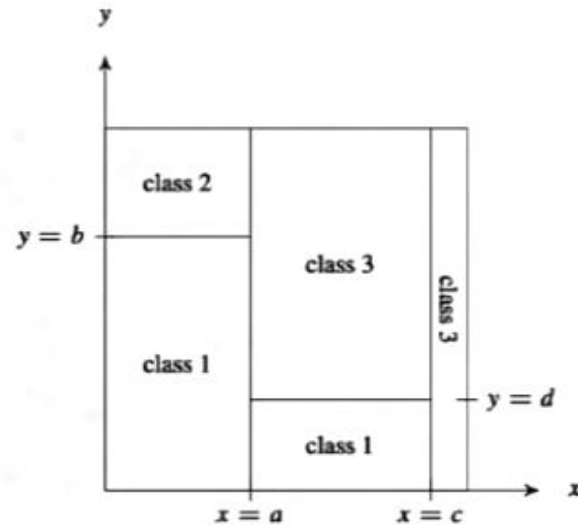
How to Split the Data

- Split at random
 - Split strategically
 - Sampling with replacement
 - Sampling without replacement
-
- ❖ Large Training Set → More accurate classifier
 - ❖ Small Training Set → Reflects variation between sample groups of sets

Overfitting and Underfitting

- **Adapting too closely to the idiosyncrasies of a sample set.**
- **Result : Small prediction error on training data but poor predictive performance on test data.**

Decision Trees: Classification



Decision Trees: Classification

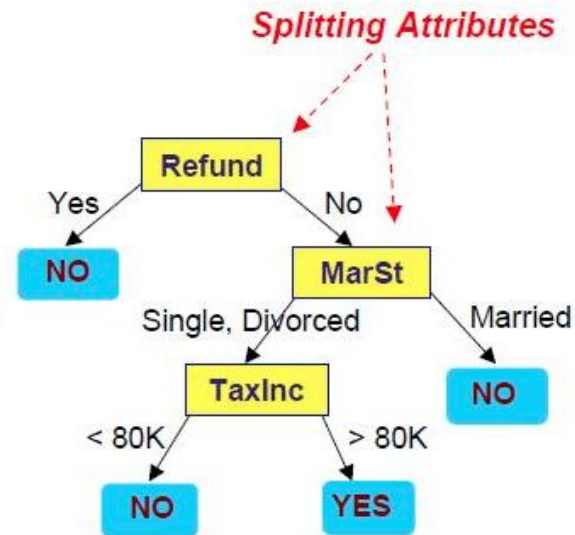
<i>Tid</i>	<i>Refund</i>	<i>Marital Status</i>	<i>Taxable Income</i>	<i>Cheat</i>
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

Decision Trees: Classification

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

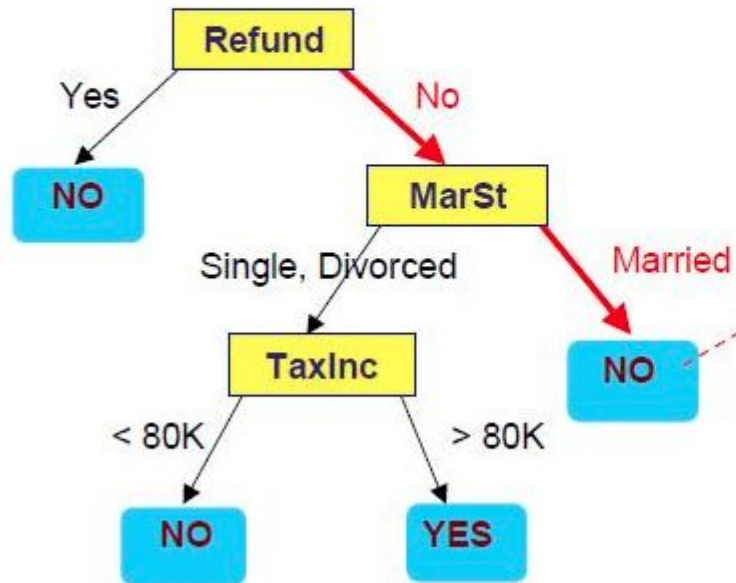
Training Data



Model: Decision Tree

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

Naive Bayes Classification Technique

The diagram shows the Naive Bayes formula with arrows pointing from descriptive labels to the corresponding terms in the equation:

$$P(c | x) = \frac{P(x | c) P(c)}{P(x)}$$

Labels and their corresponding terms:

- Likelihood points to $P(x | c)$
- Class Prior Probability points to $P(c)$
- Posterior Probability points to $P(c | x)$
- Predictor Prior Probability points to $P(x)$

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (c , *target*) given *predictor* (x , *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

How it Works?

1. *Convert the data set into frequency table*

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

How it Works?

2. *Convert the likelihood table by finding individual probabilities.*

Likelihood table				
Weather	No	Yes		
Overcast		4	$=4/14$	0.29
Rainy	3	2	$=5/14$	0.36
Sunny	2	3	$=5/14$	0.36
All	5	9		
	$=5/14$	$=9/14$		
	0.36	0.64		

How it Works?

3. Now use Naive Bayes equation to calculate the posterior probability for each class. The class with the highest posterior is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

$$P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} \mid \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

Random Forests Classification

- ★ Random forest classifier creates a set of decision trees from randomly selected subset of training set.
- ★ It then aggregates the votes from different decision trees to decide the final class of the test object.
- ★ *This works well because a single decision tree may be prone to noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results.*

Training Set

[X1, X2, X3, X4]

⇒

Labels

[L1, L2, L3, L4]

Random Forests Classification

- ★ Random forest classifier creates a set of decision trees from randomly selected subset of training set.
- ★ It then aggregates the votes from different decision trees to decide the final class of the test object.

Set of Decision Trees Created by Random Forest

[X1, X2, X3]

[X1, X2, X4]

[X2, X3, X4]

Random Forest classifier then collects votes and majority of class labels win.

Steps for creating a random forest classifier

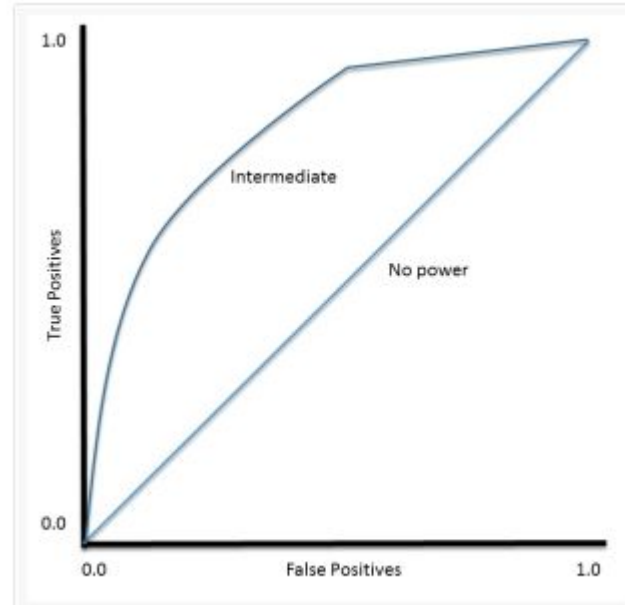
- ★ Import Classifier
- ★ Import Library
- ★ Train Model
- ★ Predict
- ★ Evaluate accuracy/performance of the model

Evaluating Performance of Classification Models

- ★ ROC Curve
 - Receiver Operating Characteristics Curve
 - A graph showing performance of a classification model
 - This curve plots two parameters
 - True Positive Rate and False Positive Rate
- ★ AOC (Area Under Curve)

Evaluating Performance of Classification Models

- ★ ROC Curve
 - Receiver Operating Characteristics Curve
 - A graph showing performance of a classification model
 - This curve plots two parameters
 - True Positive Rate and False Positive Rate



Summary

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F - \text{measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

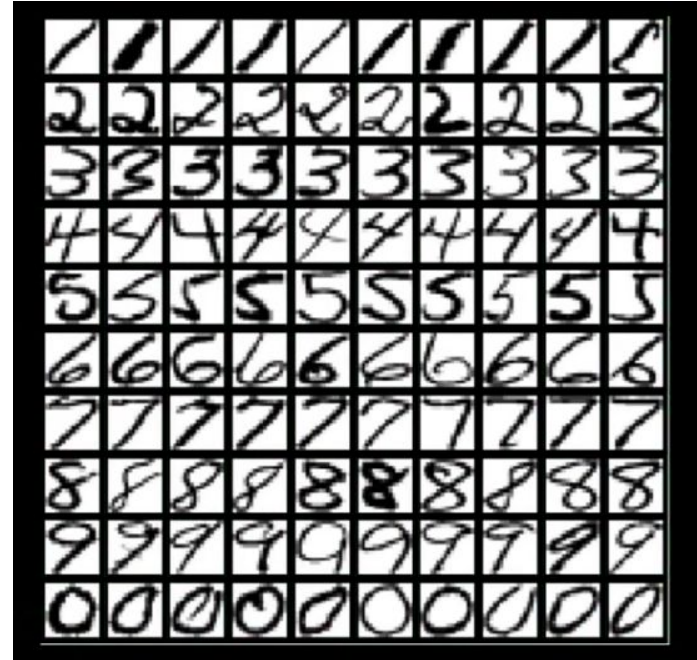
$$\text{Recall} = \frac{TP}{TP + FN}$$

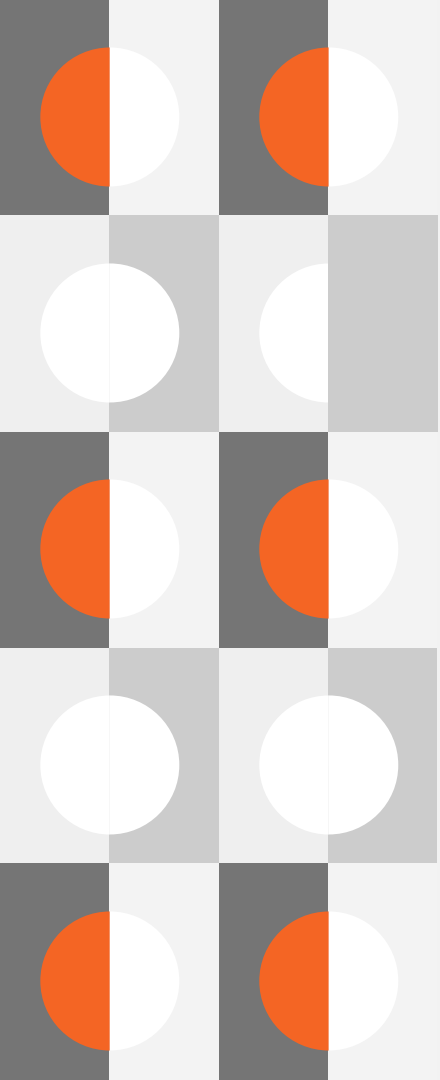
Hands on Lab : Blurred Image Prediction using Digits Dataset



0	0	0	0	0	76	39	0	0	0	0	0	0	0	0	0	0	0
0	0	9	185	248	244	238	238	134	71	0	0	0	0	0	0	0	0
0	0	16	222	242	170	170	232	255	245	80	0	0	0	0	0	0	0
0	0	0	31	43	0	0	37	173	255	238	30	0	0	0	0	0	0
0	0	0	0	0	0	0	0	19	216	255	128	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	202	255	161	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	202	255	161	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	202	255	95	0	0	0	0	0	0
0	0	0	0	0	0	0	0	71	255	255	37	0	0	0	0	0	0
0	65	239	239	239	129	0	25	211	255	174	2	0	0	0	0	0	0
49	240	205	221	255	249	127	193	255	229	46	0	0	0	0	0	0	0
131	247	50	47	197	255	255	255	255	177	0	0	0	0	0	0	0	0
131	255	204	69	198	255	255	255	255	234	142	8	0	0	0	49	26	
47	222	255	255	255	255	255	255	255	255	255	183	171	171	250	129		
0	6	101	184	237	171	142	101	139	237	243	255	255	255	175	51		
0	0	0	0	0	0	0	0	0	0	40	118	118	118	6	0		

1. Launch Jupyter
2. Open
LogisticRegression.ipynb
3. Import numpy,sklearn
and matplotlib





Thank
you