# EEP 596: AI and Health Care || Lecture 2

Dr. Karthik Mohan

Univ. of Washington, Seattle

Mar 31, 2022

# In-class Breakout (5 minutes)

Accuracy of Diagnosis

Faster Lab Tests (Imaging)

Reduce Costs in Healthcare

Preventative Diagnostic

Automated Transcribing of Doctor note

Better & Automated Diagnostic

Better Info. sharing

## Specific bottlenecks in health care

What are some specific bottlenecks in health care that you can think of where data analytics and AI can help? Think of the whole health care pipeline - from health care providers, to hospitals, to insurance to patients. What are some opportunities and what are some challenges ? Which challenges can data science help with and which challenges require policy changes or fixing other infrastructure issues?

# Next few Lectures: Recap of Linear Regression and Classification

- ML is a pre-requisite for this course. So recap will be high-level and quick!

Look up a reference on ML
Notes for each lecture
Summaries
Reference posted on discord
— ML

# Next few Lectures: Recap of Linear Regression and Classification

- ML is a pre-requisite for this course. So recap will be high-level and quick!
- If you haven't seen ML methods in the past, please connect with me

# Next few Lectures: Recap of Linear Regression and Classification

- ML is a pre-requisite for this course. So recap will be high-level and quick!
- If you haven't seen ML methods in the past, please connect with me
- We will also start off with applications in health care where **Regression** and **Classification** apply in this lecture and next lecture!

# Next few Lectures: Recap of Linear Regression and Classification

- ML is a pre-requisite for this course. So recap will be high-level and quick!
- If you haven't seen ML methods in the past, please connect with me
- We will also start off with applications in health care where **Regression** and **Classification** apply in this lecture and next lecture!
- Upcoming weeks will spotlight different facets of health care - preventative diagnostics, cancer classification, heart anomaly detetion, stress monitoring, EHR and more

Automation/Auto-Scribe

# Next few Lectures: Recap of Linear Regression and Classification

- ML is a pre-requisite for this course. So recap will be high-level and quick!
- If you haven't seen ML methods in the past, please connect with me
- We will also start off with applications in health care where **Regression** and **Classification** apply in this lecture and next lecture!
- Upcoming weeks will spotlight different facets of health care - preventative diagnostics, cancer classification, heart anomaly detetion, stress monitoring, EHR and more
- Suggestions for interesting health care angles to cover are welcome

create a survey & send out / Discord!

# Next few Lectures: Recap of Linear Regression and Classification

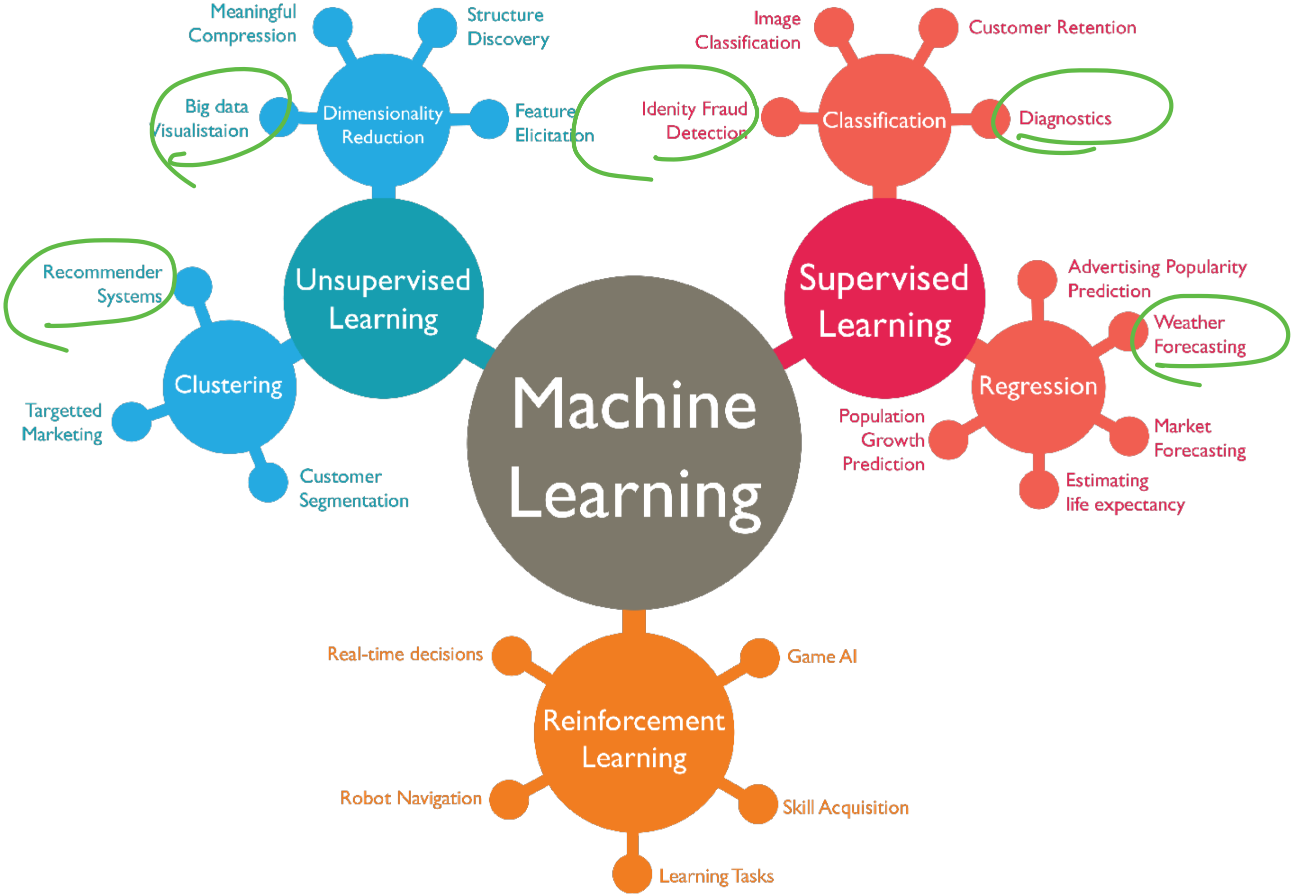
- ML is a pre-requisite for this course. So recap will be high-level and quick!
- If you haven't seen ML methods in the past, please connect with me
- We will also start off with applications in health care where **Regression** and **Classification** apply in this lecture and next lecture!
- Upcoming weeks will spotlight different facets of health care - preventative diagnostics, cancer classification, heart anomaly detetion, stress monitoring, EHR and more
- Suggestions for interesting health care angles to cover are welcome
- Guest lectures (about 4-6 planned this quarter) will shed light on state of health care and challenges from experts

Wednesday [second half of lecture

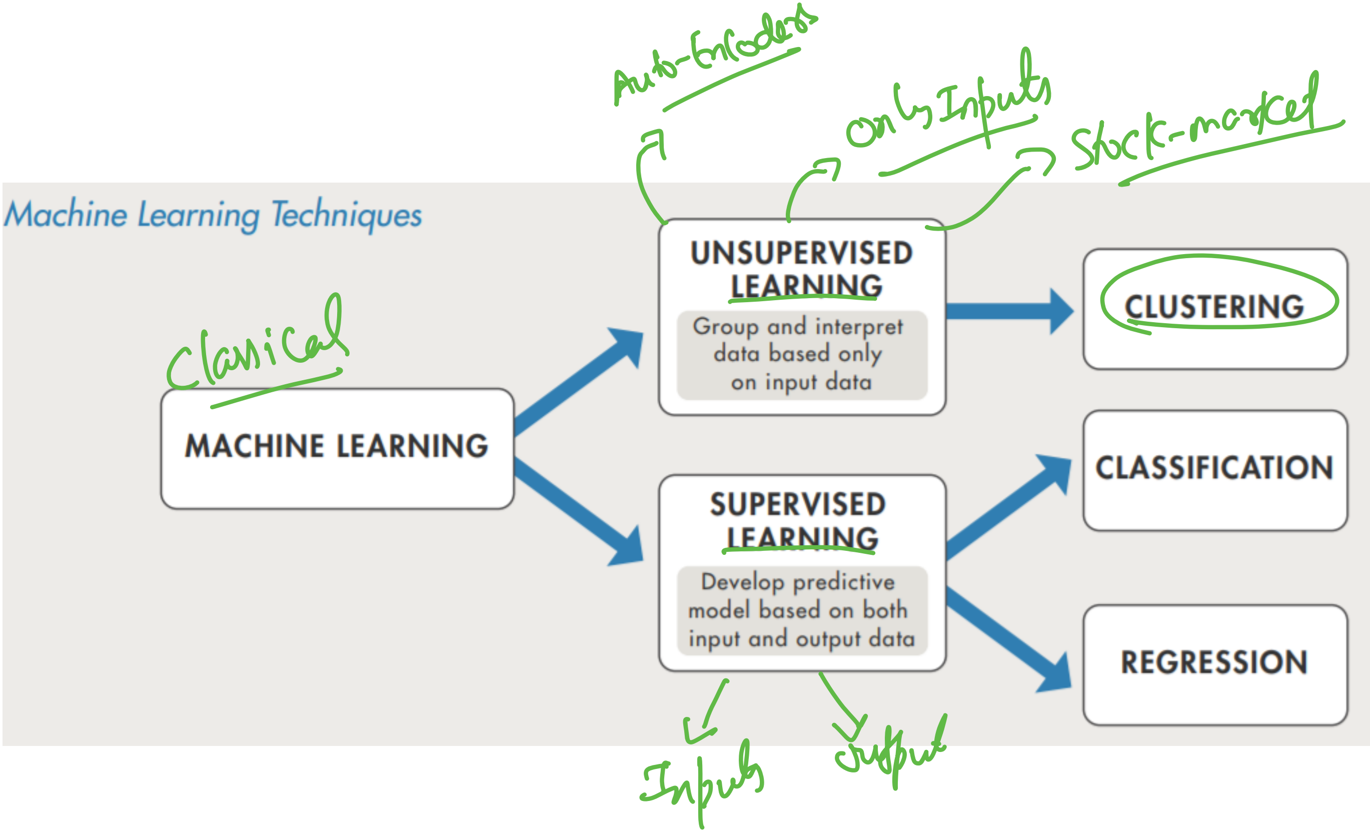
# Next few Lectures: Recap of Linear Regression and Classification

- ML is a pre-requisite for this course. So recap will be high-level and quick!
- If you haven't seen ML methods in the past, please connect with me
- We will also start off with applications in health care where **Regression** and **Classification** apply in this lecture and next lecture!
- Upcoming weeks will spotlight different facets of health care - preventative diagnostics, cancer classification, heart anomaly detetion, stress monitoring, EHR and more
- Suggestions for interesting health care angles to cover are welcome
- Guest lectures (about 4-6 planned this quarter) will shed light on state of health care and challenges from experts
- Any questions/thoughts/suggestions?

# What is Machine Learning?



# Supervised vs Unsupervised Learning

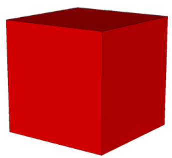


# Supervised Learning

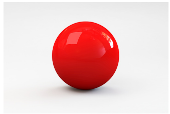
Target

Apple

Different



Cube

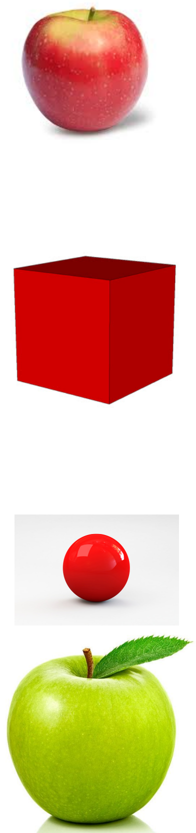


Ball

Apple

# Un-Supervised Learning

Bunch of Images/objects



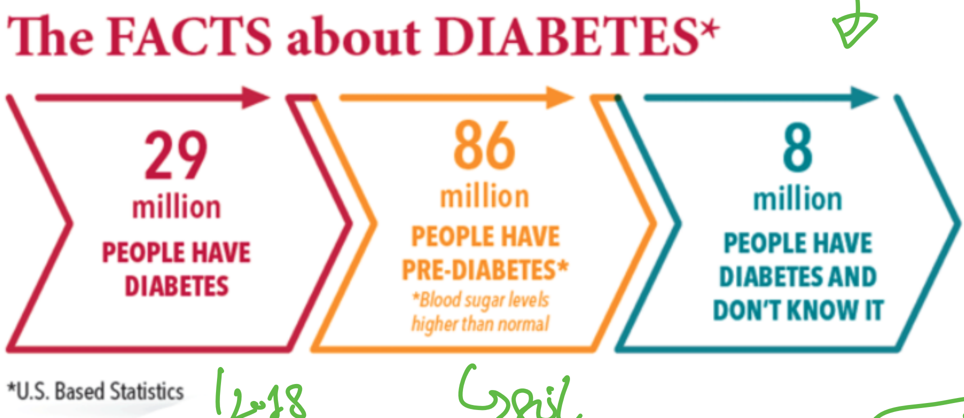
cluster

Similar!

Don't know what it is?

Cold-Start problem

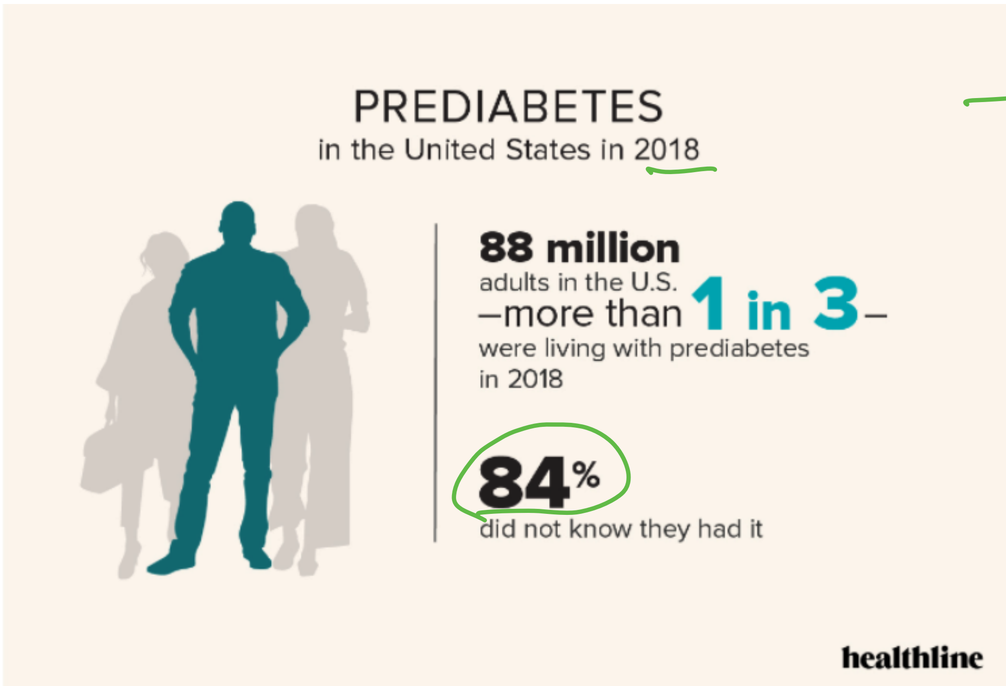
# Classification Case Study: Diabetes




2018

Issue with diagnostics/ pre-emption

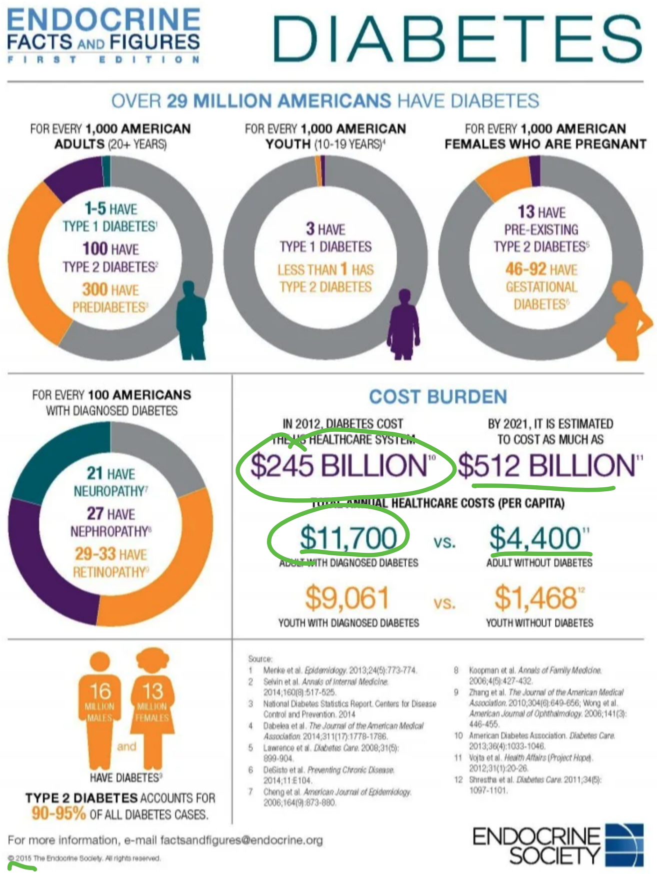
# Classification Case Study: Diabetes




Prevent Diabetes

Source: 2020 CDC report

# Classification Case Study: Diabetes




Early Diagnosis Can cut costs by half for Diabetes

# Classification Case Study: Diabetes

```
data.head(10)
```

History of Diabetes

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| **5** | 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 |
| **6** | 3 | 78 | 50 | 32 | 88 | 31.0 | 0.248 | 26 | 1 |
| **7** | 10 | 115 | 0 | 0 | 0 | 35.3 | 0.134 | 29 | 0 |
| **8** | 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| **9** | 8 | 125 | 96 | 0 | 0 | 0.0 | 0.232 | 54 | 1 |

Outcome → $\boxed{\text{TARGET}}$

Glucose → HbA1C → 3month sugar level in body

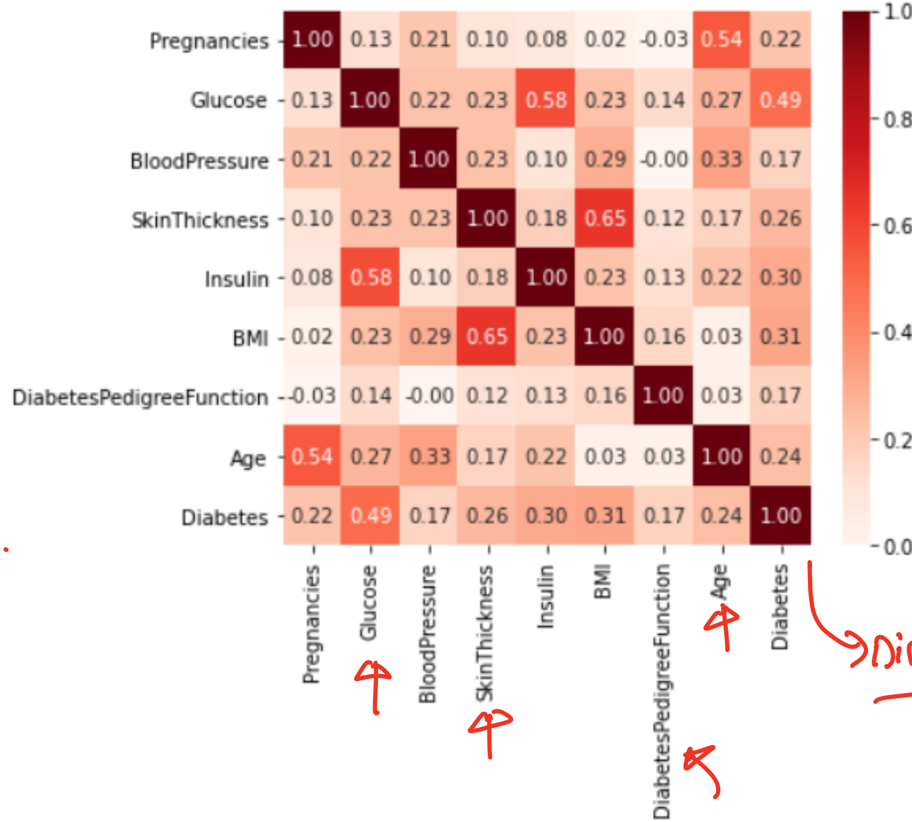
# Classification Case Study: Diabetes

| Feature | | | Classification rules | |
|---|---|---|---|---|
| Num. | Name | Class | TN | TP |
| 1 | Number of times pregnant | Numeric | [0.79,16.04] | [13.69,16.28] |
| 2 | Plasma glucose concentration | Numeric | [25.92,148.08] | n/a |
| 3 | Diastolic blood pressure | Numeric | [6.18,84.45] | [53.71,81.74] |
| 4 | Triceps skin fold thickness | Numeric | [8.33,52.15] | [15.39,27.88] |
| 5 | 2-h serum insulin | Numeric | [435.02,730.53] | [759.30,840.51] |
| 6 | Body mass index | Numeric | [36.43,37.96] | [31.75,58.41] |
| 7 | Diabetes pedigree function | Numeric | n/a | n/a |
| 8 | Age | Numeric | [68.45,75.98] | 34.29,41.01] |

# Classification Case Study: Diabetes

Features are not heavily correlated with each other

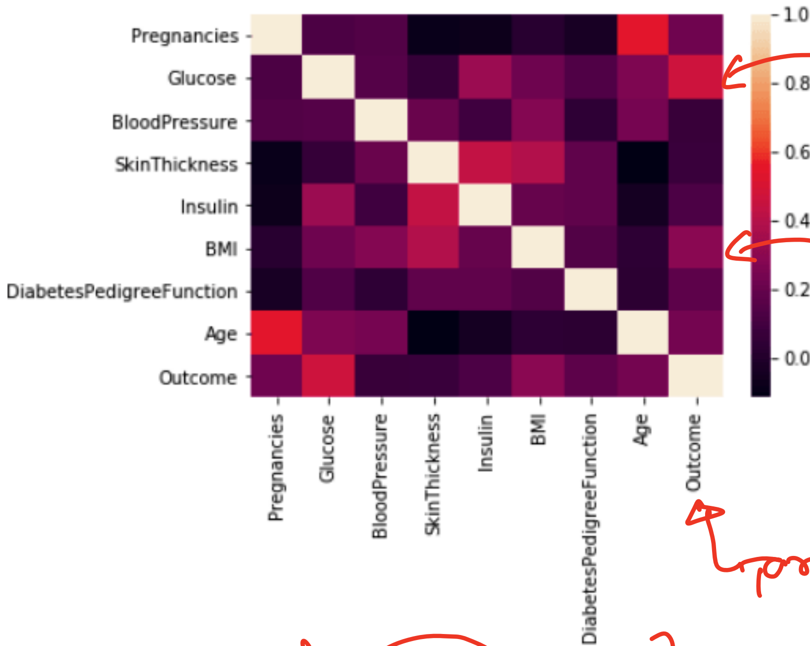
– This helps model learn from diff. features!




values in [0,1]

← Heat map / Correlation Matrix

Diagonals are 1!




Heat map pooled across the examples

target

Features

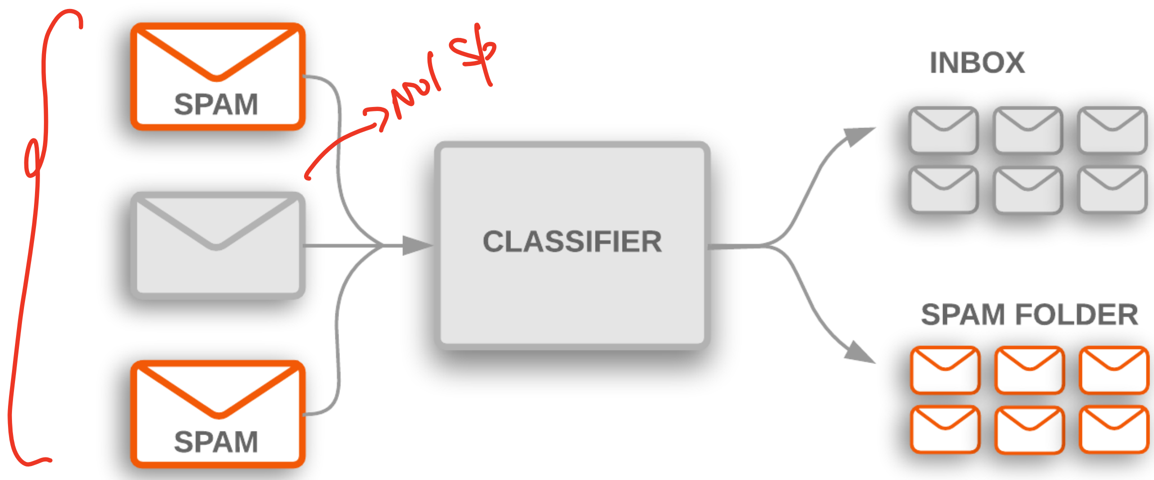
# Classification/Classifiers Recap!

Binary classification

### Pointers

Predict binary values from a set of features. Example: Has Diabetes/Doesn't have diabetes, given health profile of a patient. The health profile informs the features of the patient.

# Classification in Machine Learning



# Difference between Classification and Regression

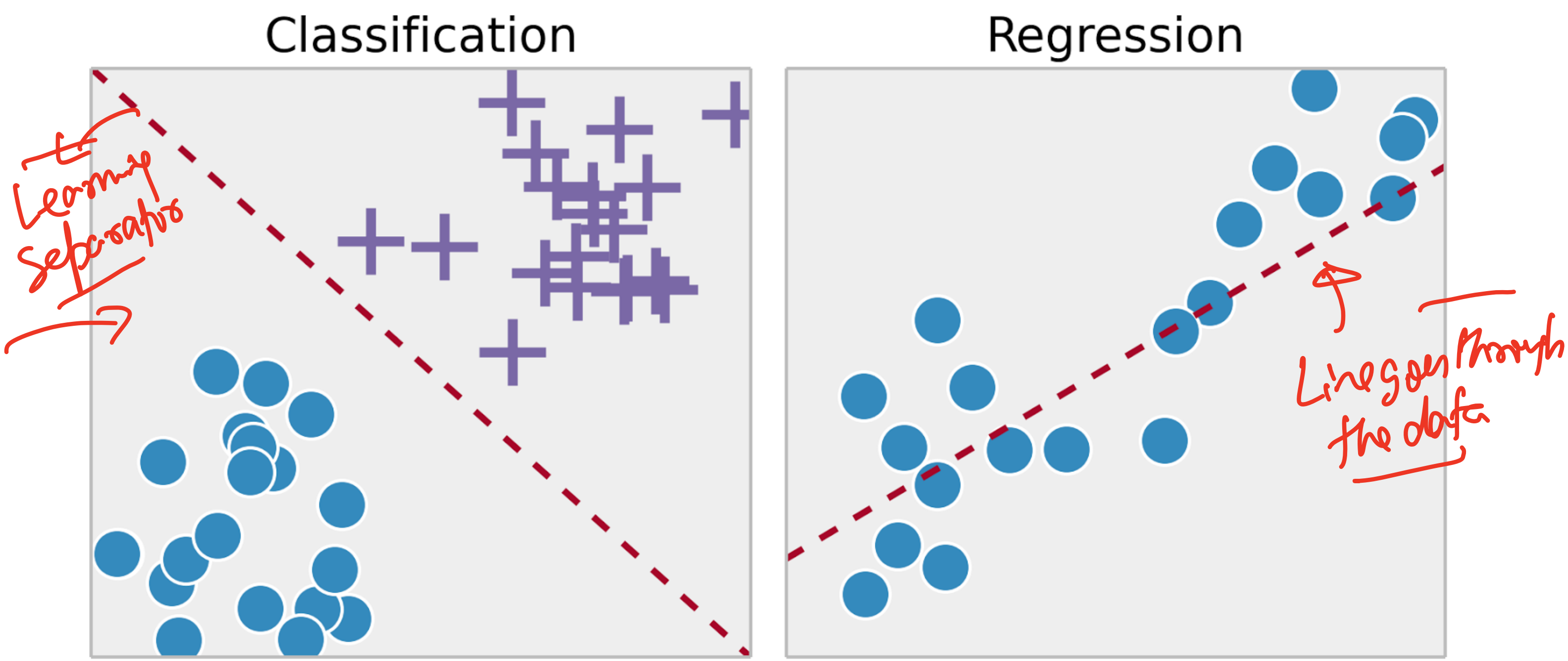
### Simple difference

The target type in Regression is **numeric** whereas that in classification is **categorical**

# Difference between Classification and Regression

### Simple difference

The target type in Regression is **numeric** whereas that in classification is **categorical**



# Types of Classification

**Binary vs Multi-class classification**

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

# Types of Classification

### Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

### Target is called Label

For binary classification, the convention is to label the target as positive or negative. Example: Positive for spam and negative for not-spam.

# Types of Classification

## Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

## Target is called Label

For binary classification, the convention is to label the target as positive or negative. Example: Positive for spam and negative for not-spam.

## Target Example in Diabetes

Example: Positive for has diabetes, negative for does not have diabetes

# Types of Classification

### Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

### Target is called Label

For binary classification, the convention is to label the target as positive or negative. Example: Positive for spam and negative for not-spam.

### Target Example in Diabetes

Example: Positive for has diabetes, negative for does not have diabetes
Example: Positive for high-risk of chronic diabetes, negative for ~~high~~-risk of chronic diabetes (as in the Programming Assignment)

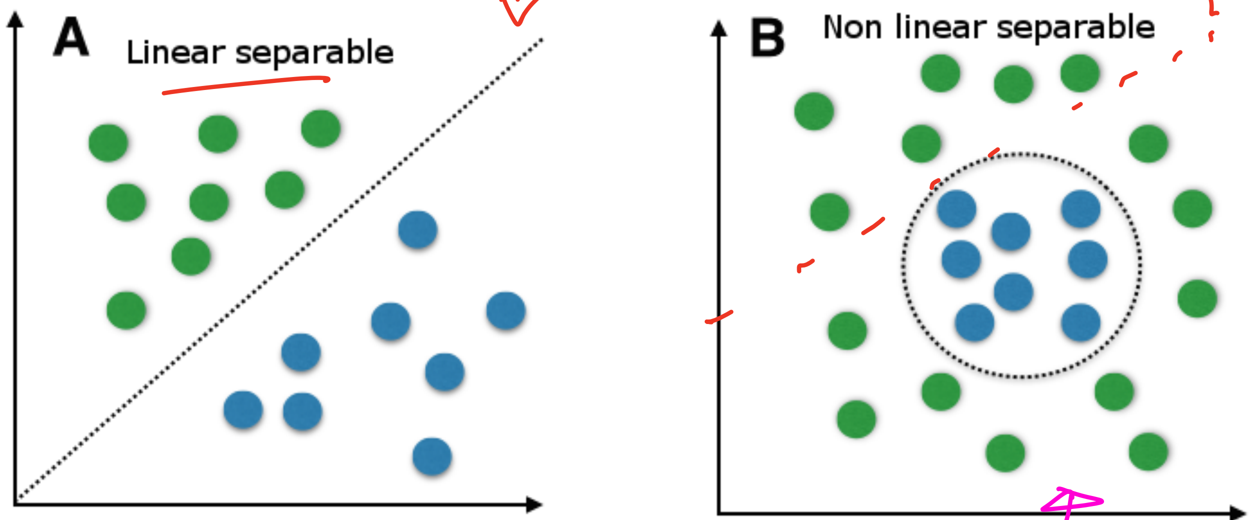
low

Prognosis

# Spam Classification Example

| Email excerpt | Type | Label |
|---|---|---|
| Could you please respond by tomorrow? | Not-spam | -1 |
| Congratulations!!! You have been selected... | Spam | +1 |
| Looking forward to your presentation... | Not-spam | -1 |
| ... | ... | ... |

# Linear Separability




Does not separate the blues from the greens

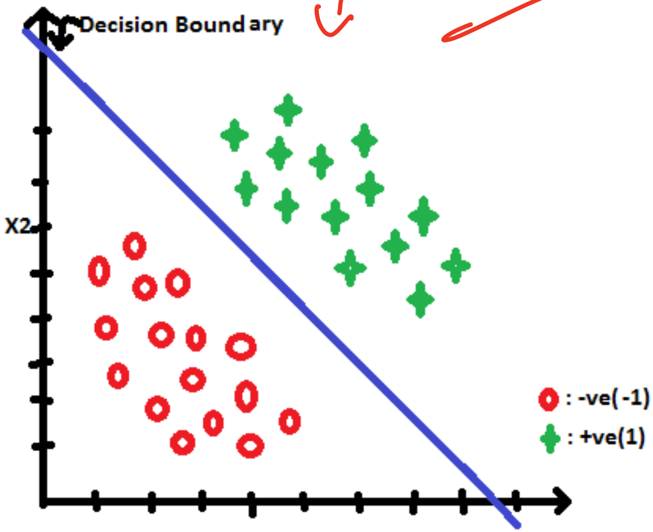
We can plot in 2 dimension (feature dimension, d=2)

d=100! → can't plot

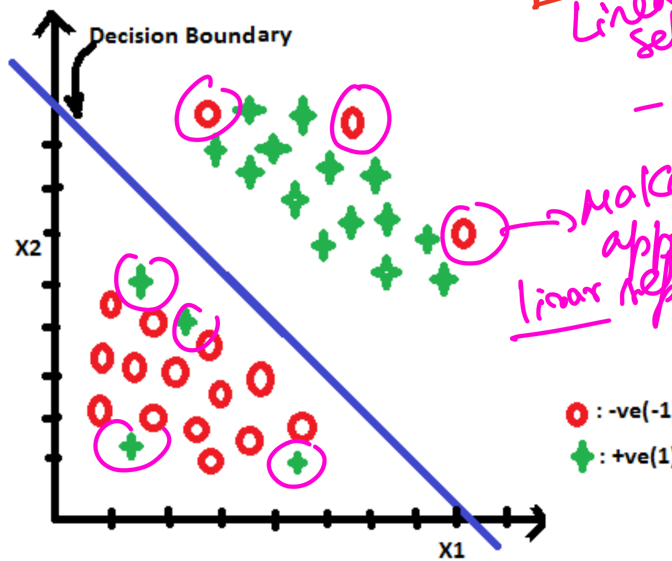
# Approximate Linear Separability

Exact Linear separability

Approximate Linear separability




Perfect Linearly separable



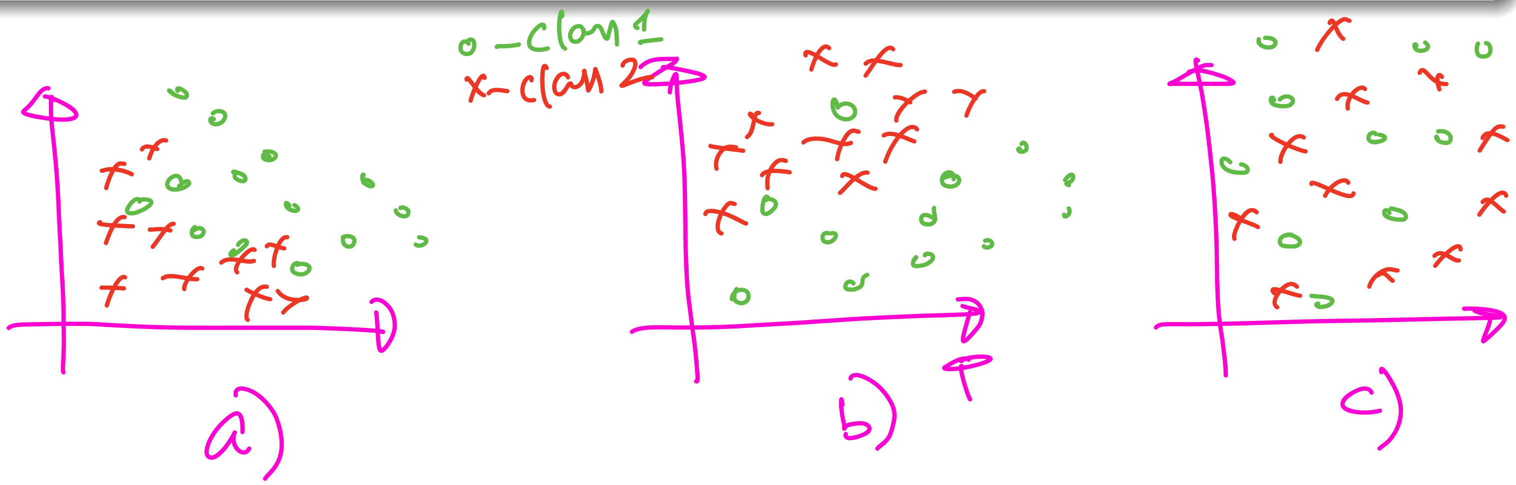

Almost Linearly Separable

Makes it approx. linear separable

# ICE #1

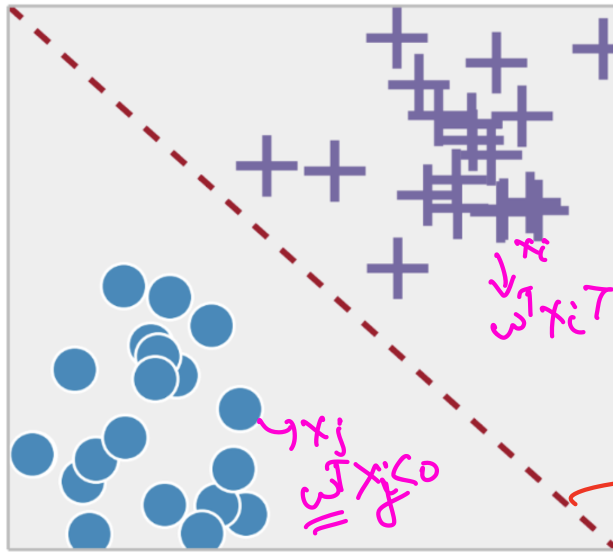
Which of the following data sets is the closest to being linearly separable?

pollev.com/karthikmohan088



# Logistic Regression

Basic Linear model for classification



$x_i$

$w^T x_i > 0$

$x_j$

$w^T x_j < 0$

$w^T x = 0$

weights

features

Have a dummy feature of 1

## LR fundamentals

- Linear Model
- Want score $w^T x^i > 0$ for $y_i = +1$ and $w^T x_i < 0$ for $y_i = -1$!
- If linearly separable data, above is feasible. Else, minimize error in separability!!

# Logistic Regression

Convert

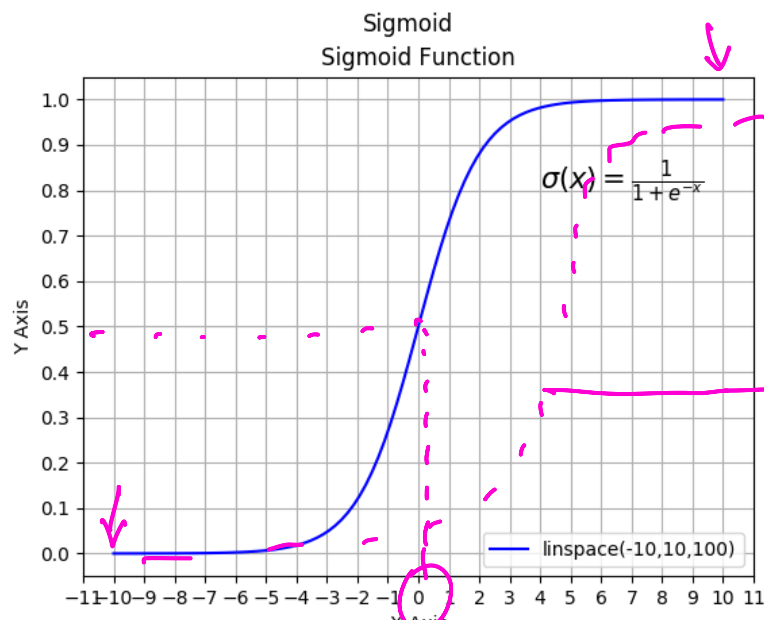
Score($w^TX$) → probability!

## Probability for a class

In LR, the score, $w^T x$ is converted to a probability through the sigmoid function. So we can talk about $P(\hat{y}^i = +1)$ or $P(\hat{y}^i = -1)$

↳ Has Diabetes ↳ does not have diabetes

## Sigmoid Function



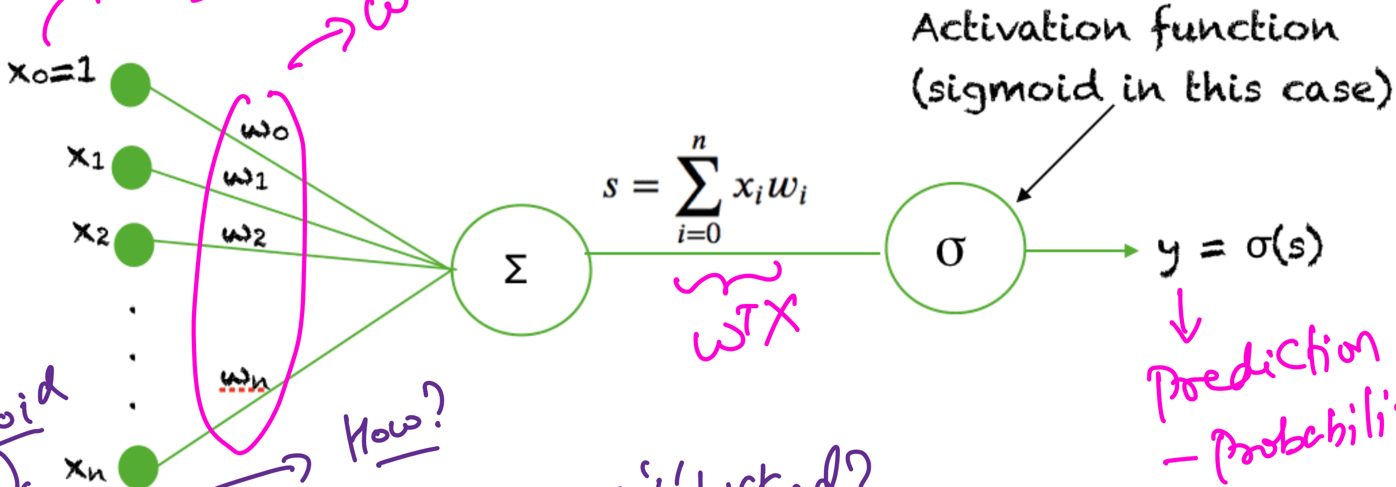

S shaped

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

# LR represented Graphically

Dummy feature of 1

$w \in \mathbb{R}^d$




$W^T X$

Prediction
– Probability

Sigmoid

How?

| Data | Prob | t | Pred |
|---|---|---|---|
| $x^{(1)}$ | 0.7 | 0.5 | 1 |
| $x^{(2)}$ | 0.7 | 0.8 | 0 |
| $x^{(3)}$ | 0.6 | 0.5 | 1 |

How is 't' picked?

minimizing metric on validation set

$y > t \rightarrow$ Predict +ve class

$y < t \rightarrow$ Predict –ve class

# Logistic Regression

prob. > t → predict 1

else 0

→ sigmoid

## LR Prediction

$$\hat{y}_i = \frac{1}{1 + e^{-\underbrace{\hat{w}}_{\text{weight}}{}^T \underbrace{x^i}_{\text{feature}}}}$$

## LR Loss

Assume that $y_i = 0$ or $y_i = 1$ (i.e. the negative class has a label 0). Then the binary cross-entropy loss applies to LR:

$$\min_w \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!
6. Logistic Regression uses the log-loss or cross-entropy loss whereas Linear Regression uses the quadratic loss

# Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!
6. Logistic Regression uses the log-loss or cross-entropy loss whereas Linear Regression uses the quadratic loss
7. Logistic Regression loss can be derived as a MLE - So its well grounded in statistics.

Maximum Likelihood Estimate

# Evaluating Classifiers!

### ICE #2

Let's say you are tasked with predicting risk of lung cancer for patients. You create a classifier which has 95% accuracy on patients who actually have low risk of lung cancer. Should you be happy with the classifier?

a) Yes

b) No

c) Maybe!

d) Something's fishy!

pollev.com/karthikmohan088

# Evaluating Classifiers!

observational study vs a Randomized Control Trial / study

ICE #3

Let's say you are tasked with predicting risk of lung cancer for patients. Your data set is obtained from patients who volunteer for the study and hence you end up having a lot of patients with risk for lung cancer. You create a classifier which has 90% accuracy on patients who actually have high risk of lung cancer. Should you be happy with the classifier?

a) Yes
b) No
c) Maybe!
d) Something's fishy!

pollev.com/karthikmohan088

# Evaluating classifiers

### Class imbalance

The above data set is an example of class imbalance. What can go wrong here?

# Evaluating classifiers

### Class imbalance

The above data set is an example of class imbalance. What can go wrong here?

### Better metric than accuracy

100 cancer patient
900 non-cancer patient

Consider the **confusion matrix** for above cancer classification example with the trivial classifier (predict everything as not-cancer).

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives | 0 | 100 |
| Negatives | 0 | 900 |

90% Accuracy seems good but needs careful attention

# Evaluating classifiers

## Better metric than accuracy

Consider the confusion matrix for above cancer classification example with the trivial classifier (predict everything as not-cancer). → Predictions

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives | ✓ 0 | 100 ✗ |
| Negatives | ✗ 0 | 900 ✓ |

← Confusion matrix

100
900

Truth/ True Labels

Positives = Patients with cancer
Negatives = Patients without cancer

# Evaluating classifiers

### Better metric than accuracy

Consider the confusion matrix for above cancer classification example with the trivial classifier (predict everything as not-cancer).

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives | 0 | 100 |
| Negatives | 0 | 900 |

### Better metric than accuracy

Accurcay is how many data points the classifier got right divided by the total data points. What's accuracy here?

$$\text{Accuracy} = \frac{\text{Sum of Diagonals}}{\text{Total Data points}}$$

# Evaluating classifiers

## Better metric than accuracy

Consider the confusion matrix for above Cancer classification example with the trivial classifier (predict everything as not-cancer).

| | Predicted Positive | Predicted Negatives |
|---|---:|---|
| Positives (P) | 0 | 100 |
| Negatives (N) | 0 | 900 |

# Evaluating classifiers

### Better metric than accuracy

Consider the confusion matrix for above Cancer classification example with the trivial classifier (predict everything as not-cancer).

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives (P) | 0 | 100 |
| Negatives (N) | 0 | 900 |

### Accuracy, Precision, Recall and F1-score

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives (P) | TP | FN |
| Negatives (N) | FP | TN |

# Evaluating classifiers

### Better metric than accuracy

Consider the confusion matrix for above Cancer classification example with the trivial classifier (predict everything as not-cancer).

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives (P) | 0 | 100 |
| Negatives (N) | 0 | 900 |

# Evaluating classifiers

## Better metric than accuracy

Consider the confusion matrix for above Cancer classification example with the trivial classifier (predict everything as not-cancer).

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives (P) | 0 | 100 |
| Negatives (N) | 0 | 900 |

→ Recall

## Accuracy, Precision, Recall and F1-score

**Precision (Pr)** = TP/(TP + FP) → looking at column

**Recall (R)** = TP/(TP + FN) = TP/P

**F1-score** = $\frac{2 \times Pr \times R}{Pr + R}$ } Harmonic Mean between P & R

**Accuracy (Acc)** = $(TP + TN)/(P + N)$

# ICE #4

## More Confusion!

Let's say we computed a **Confusion Matrix** for another Cancer Classifier on a different data set and we obtained:

| | Predicted Positive | Predicted Negatives |
|---|---|---|
| Positives (P) | 50 | 50 |
| Negatives (R) | 100 | 400 |

## Metrics!

**Accuracy**, **Pr**, **R** and **F1** are as follows:

a) $75\%, 0.2, 0.5, 0.285$

b) $80\%, 0.3, 0.4, 0.285$ ?

c) $80\%, 0.5, 0.3, 0.1875$

d) $75\%, 0.3, 0.5, 0.1875$

# Programming Assignment 1: Diabetes Classification

### Kaggle Contest

- **Description:** You get to work on the Diabetes data set and make predictions using your favorite classifiers

DT → Decision Tree

Interpretability!

People in medicine use Decision Tree charts to make decisions

# Programming Assignment 1: Diabetes Classification

## Kaggle Contest

- **Description:** You get to work on the Diabetes data set and make predictions using your favorite classifiers
- **Programming componen~~n~~t:** A starter Jupyter notebook will be provided

# Programming Assignment 1: Diabetes Classification

## Kaggle Contest

- **Description:** You get to work on the Diabetes data set and make predictions using your favorite classifiers
- **Programming componennt:** A starter Jupyter notebook will be provided
- **Kaggle component:** Submit your predictions on a “held out” test data set for a fun peer learning experience

# Programming Assignment 1: Diabetes Classification

## Kaggle Contest

- **Description:** You get to work on the Diabetes data set and make predictions using your favorite classifiers
- **Programming componennt:** A starter Jupyter notebook will be provided 1
  ↳ ~~Library~~ Modules
- 2 **Kaggle component:** Submit your predictions on a "held out" test data set for a fun peer learning experience
- 3 **Report component:** Consolidate your learnings, insights, graphs in one place

# Programming Assignment 1: Diabetes Classification

## Kaggle Contest

- **Description:** You get to work on the Diabetes data set and make predictions using your favorite classifiers
- **Programming componennt:** A starter Jupyter notebook will be provided
- **Kaggle component:** Submit your predictions on a “held out” test data set for a fun peer learning experience
- **Report component:** Consolidate your learnings, insights, graphs in one place
- Assigned Sunday morning and due next Sunday night

# Training the Logistic Regression Model

or Any Model

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $y$ |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

# Example: 70 : 10 : 20 Train-Val-Test data split

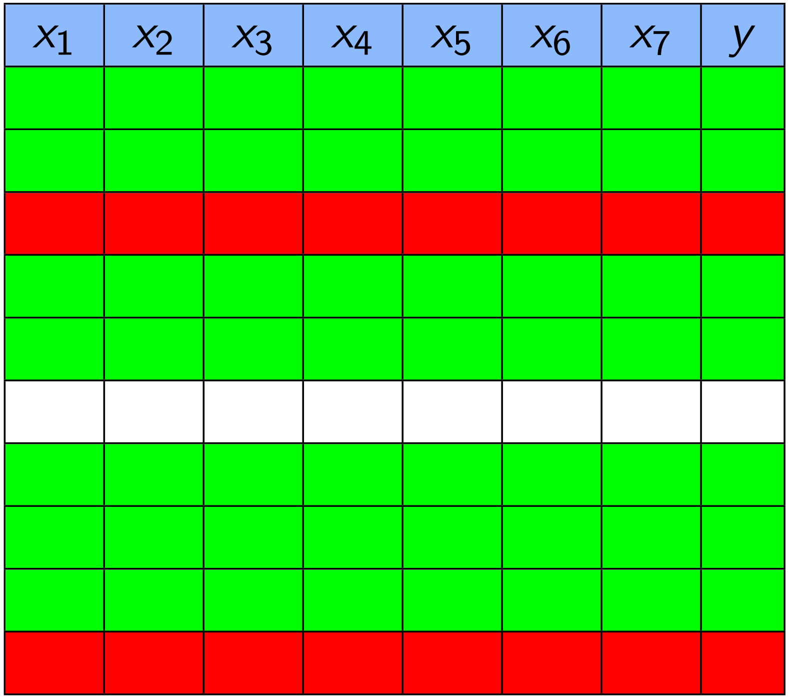
Train set

Choose 70% train data at random

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $y$ |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

# Example: 70 : 10 : 20 Train-Val-Test data split

Add 20% test data at random



# Example: 70 : 10 : 20 Train-Val-Test data split

## Remainder becomes validation data

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $y$ |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Depends on your data set

70 : 10 : 20

80 : 10 : 10

85 : 5 : 10

very little train data

Hyper-parameters:-

Fine-tuned on validation only!

# The phenomenon of Overfitting

### Overfitting

Overfitting is when your model performs great on training data but doesn't match up on test data. To account for overfitting, we also have a validation data set.

# The phenomenon of Overfitting

### Overfitting

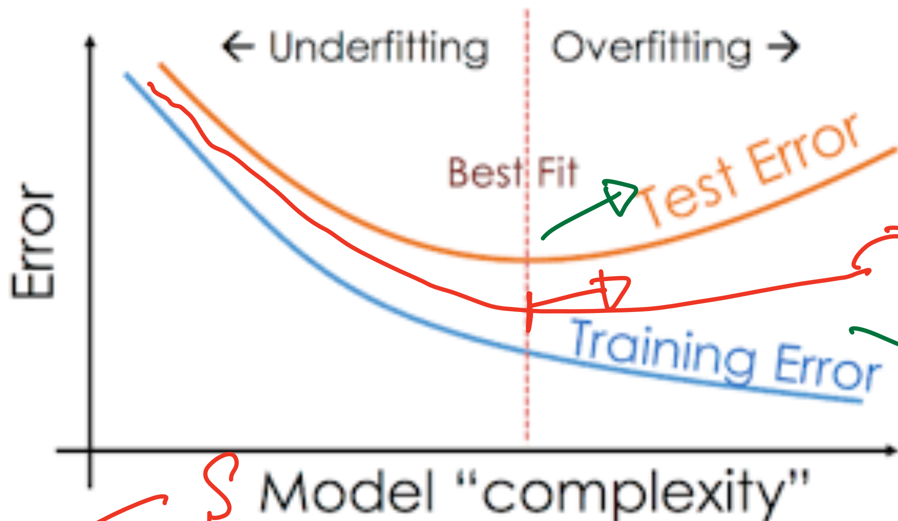
Overfitting is when your model performs great on training data but doesn't match up on test data. To account for overfitting, we also have a validation data set.

### Overfitting

When you have 90% accuracy on your training data for predicting diabetes but 70% on Kaggle contest in programming 1!

- Regularization
- Feature selection

# The figure to remember for over-fitting!




validation Error

Hyper-param

Template

Linear vs Non-Linear model

#features

Hyper-param tuning

Regularization

↑Reg. → Lower model complexity

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!

Overcoming over fitting

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)

Data weight label

$N$ $(x) \in \mathbb{R}^d$ $w \in \mathbb{R}^d$

Linear Algebra

$$\underset{N \times d}{X} \; \underset{d \times 1}{w} = \underset{N \times 1}{y}$$

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!
- ICE #0: Find a solution for $\underbrace{1^T w = 1}$ where $w \in \mathcal{R}^2$.

$w_1 + w_2 = 1$

$\left\{ \begin{array}{l} [0, 1] \\ [1, 0] \\ (\alpha, \beta) \quad \alpha + \beta = 1 \end{array} \right.$

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!
- ICE #0: Find a solution for $1^T w = 1$ where $w \in \mathcal{R}^2$.
- Unique solution when $\underline{N > d}$ and when $X$ has full rank!

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!
- ICE #0: Find a solution for $1^T w = 1$ where $w \in \mathcal{R}^2$.
- Unique solution when $N > d$ and when $X$ has full rank!
- Over-fitting happens when number of data points comparable to the number of attributes/features (order)

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!
- ICE #0: Find a solution for $1^T w = 1$ where $w \in \mathcal{R}^2$.
- Unique solution when $N > d$ and when $X$ has full rank!
- Over-fitting happens when number of data points comparable to the number of attributes/features (order)
- `Solution A` to overfitting: Increase number of examples so that $N >> d$

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!
- ICE #0: Find a solution for $1^T w = 1$ where $w \in \mathcal{R}^2$.
- Unique solution when $N > d$ and when $X$ has full rank!
- Over-fitting happens when number of data points comparable to the number of attributes/features (order)
- `Solution A` to overfitting: Increase number of examples so that $N >> d$
- `Solution B`: Decrease number of features so that $d << N$

Feature Selection Strategies

# Understanding over-fitting better

- Idea is that there maybe many solutions that fit the data - So pick the solution wisely!
- Consider the linear system $Xw = y$. This system is under-determined when $N < d$ (number of examples ¡ feature dimension)
- Infinitely many solutions when $N < d$!
- ICE #0: Find a solution for $1^T w = 1$ where $w \in \mathcal{R}^2$.
- Unique solution when $N > d$ and when $X$ has full rank!
- Over-fitting happens when number of data points comparable to the number of attributes/features (order)
- Solution A to overfitting: Increase number of examples so that $N >> d$
- Solution B: Decrease number of features so that $d << N$
- Solution C: Regularization! (Perhaps accomplish B as well along the way)

Dropout ↳ Lasso → Also does feature selection!

# Next Lecture

More on over-fitting, Decision Trees Classifiers, Random Forests and other important ML details recap!