# NUS Fintech Society

## Machine Learning Department
## Training Wing

Session 4: Supervised ML (3/10/2020)

# Content

- Intro to Machine Learning
- Supervised Learning
  - Linear Regression
  - Logistic Regression
  - Decision Tree
  - Naive Bayes
  - SVM
  - kNN

# 1. Intro to Machine Learning

# Activity Time

- Go to https://b.socrative.com/login/student/?feature=embed
- Room Name: NUSFINTECH
- Enter your name

## Facts about Machine Learning

# 2. Let’s have some fun!

# Activity Time

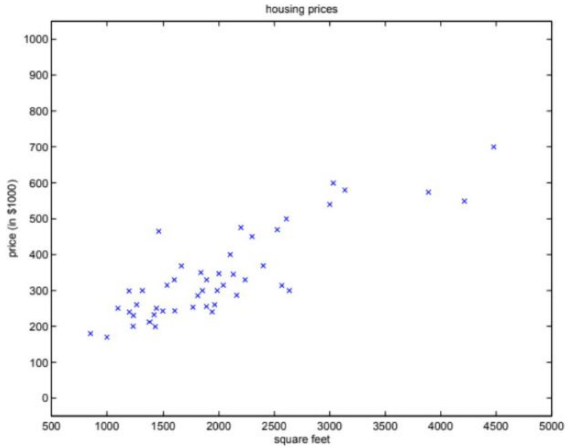
- Let's have some fun! We will randomly choose a name. If you are chosen, please share your screen and try our a photo of yourself on this website. (at your own discretion)
- https://www.how-old.net/

# 3. Linear Regression

# 2. Linear Regression

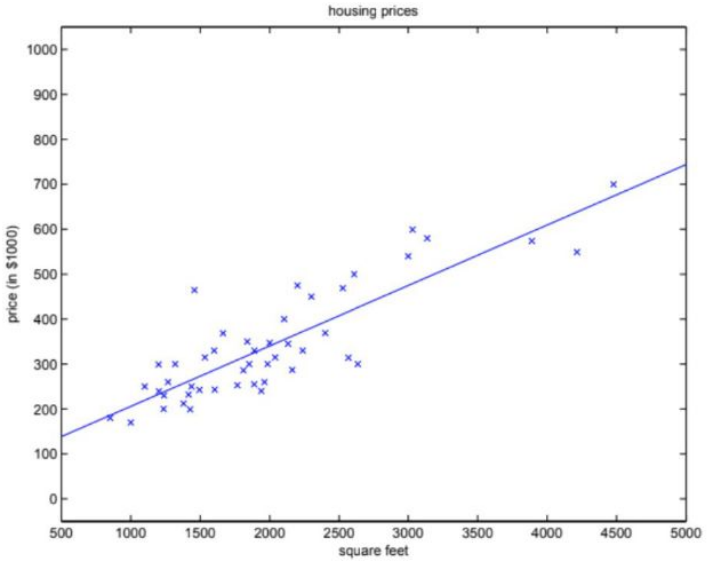
Linear Regression (In-Class A)
Objective:
Predict the housing prices (Target Variable) using

- the Input Variables (Housing Floor Area)



# 2. Linear Regression

Strategy: Use OLS (Ordinary Least Squares) to fit a line to minimise the squared errors



# Pros and Cons

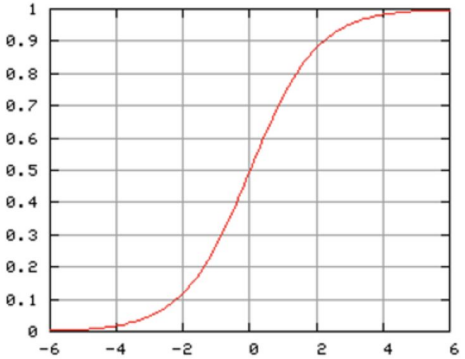
| Advantages | Disadvantages |
| --- | --- |
| Simple to implement<br>Easy to interpret | Sensitive to outliers |
| Overfitting can be reduced by regularization | Prone to under-fitting |

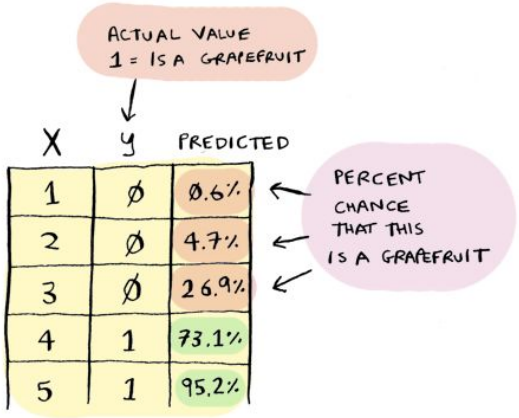
# 3. Logistic Regression

# 3. Logistic Regression

- Prediction function splits out a percentage
- A sigmoid function makes the prediction function output a percentage
- Sigmoid function produces

$$P(t) = \frac{1}{1 + e^{-t}}$$



# Pros and Cons

| Advantages | Disadvantages |
| --- | --- |
| Predicted parameters give inference about the importance of each feature | Requires moderate or no multicollinearity between independent variables |
| Very efficient when the dataset has features that are linearly separable | Cannot solve non linear problems |

# 4. Decision Tree

# 4. Decision Tree

Objective:
Given a dataset with features and labels, divide them into “pure” subsets based on each feature-label pair.

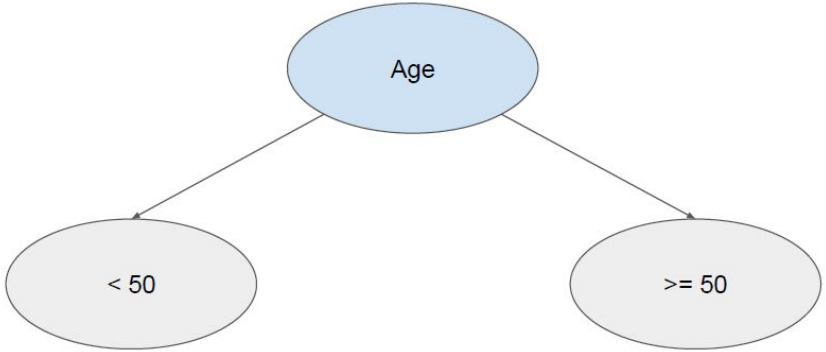
3 with diabetes, 2 without

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| **0** | 88 | 141 | 183 | 1 |
| **1** | 67 | 100 | 175 | 1 |
| **2** | 21 | 71 | 0 | 0 |
| **3** | 58 | 86 | 71 | 0 |
| **4** | 39 | 160 | 175 | 1 |

New row: Outcome?

| | Age | Insulin | Glucose |
|---|---|---|---|
| **5** | 40 | 102 | 165 |

# 4. Decision Tree

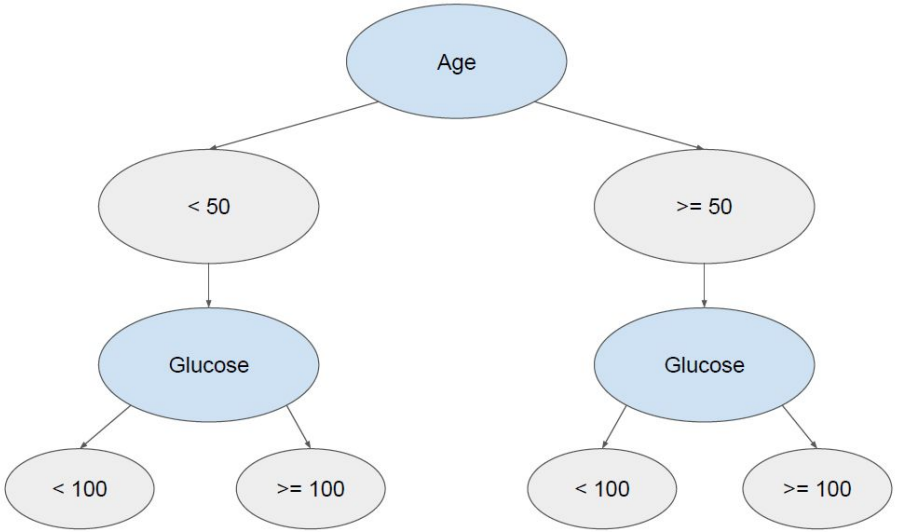



| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| **2** | 21 | 71 | 0 | 0 |
| **4** | 39 | 160 | 175 | 1 |

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| **0** | 88 | 141 | 183 | 1 |
| **1** | 67 | 100 | 175 | 1 |
| **3** | 58 | 86 | 71 | 0 |

Not pure. Continue splitting by features!

# 4. Decision Tree




| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 2 | 21 | 71 | 0 | 0 |

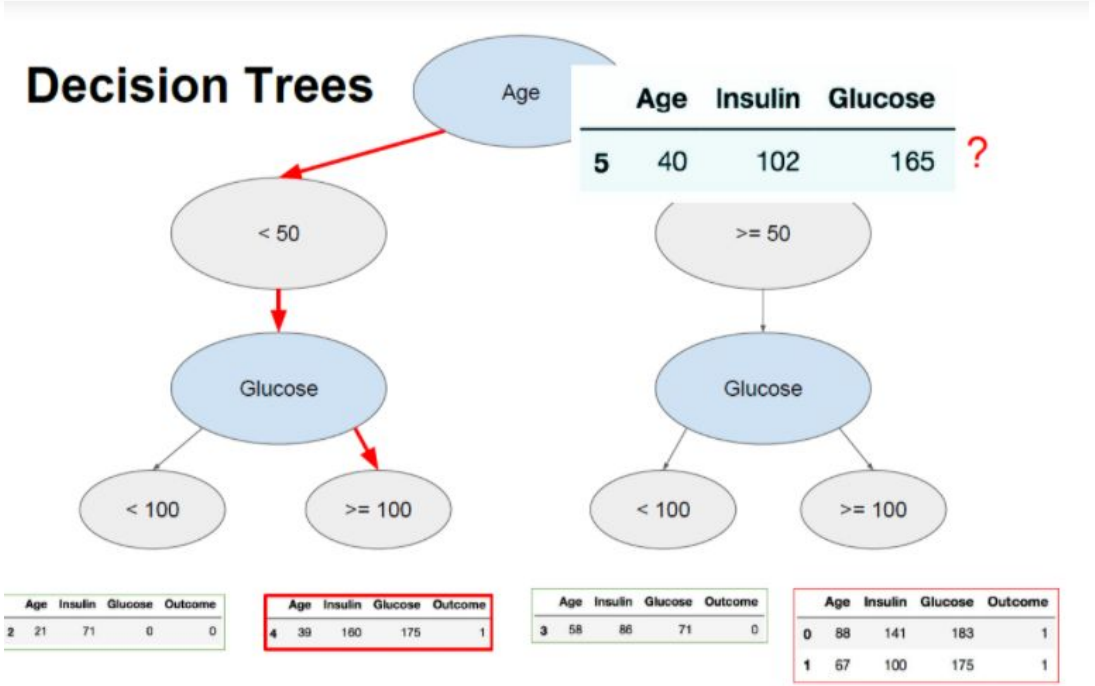
| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 4 | 39 | 160 | 175 | 1 |

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 3 | 58 | 86 | 71 | 0 |

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 0 | 88 | 141 | 183 | 1 |
| 1 | 67 | 100 | 175 | 1 |

Pure subsets. Stop splitting!

# 4. Decision Tree




| | Age | Insulin | Glucose |
|---|---|---|---|
| 5 | 40 | 102 | 165 |

?

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 2 | 21 | 71 | 0 | 0 |

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 4 | 39 | 160 | 175 | 1 |

| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 3 | 58 | 86 | 71 | 0 |

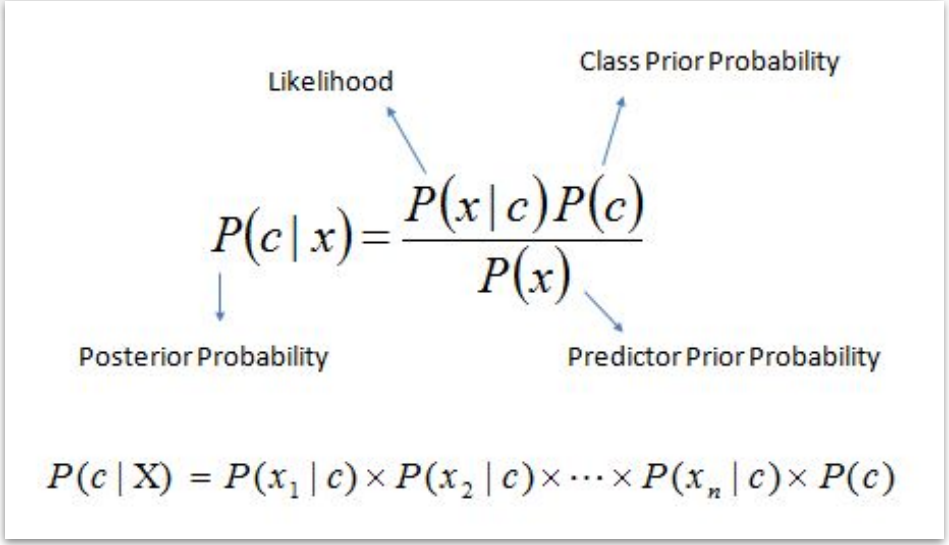
| | Age | Insulin | Glucose | Outcome |
|---|---|---|---|---|
| 0 | 88 | 141 | 183 | 1 |
| 1 | 67 | 100 | 175 | 1 |

# Pros and Cons

| Advantages | Disadvantages |
| --- | --- |
| Less effort for data preparation | Unstable |
| Intuitive | Calculations can get very complex |

# 5. Naive Bayes

# 5. Naive Bayes




C = Happy/Sad/Angry

X = “Oh yeah! I just won the lottery.”

X1 = “Oh”, X2 = “yeah”, X3 = “I”, X4 = “just”, X5 = “won”, X6 = “the”,

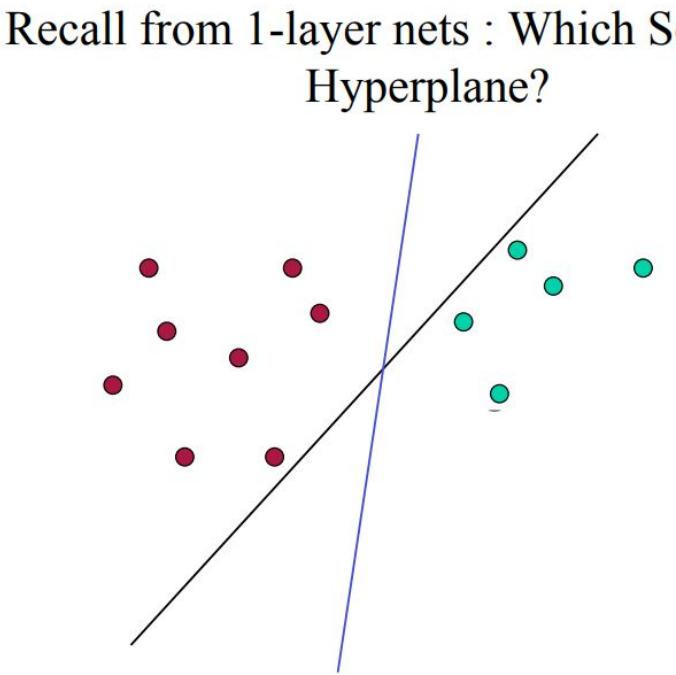
X7 = “lottery”

# Pros and Cons

| Advantages | Disadvantages |
| --- | --- |
| Better performance when independent assumption holds true | Almost impossible to have independent predictors in real life |
| Less training period | Zero Frequency |

# 6. Support Vector Machine(SVM)

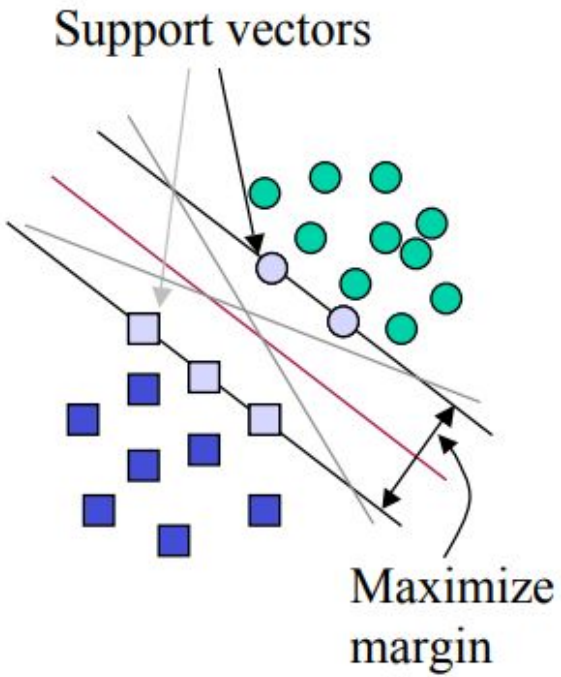
# 6. SVM




- In general, lots of possible solutions for a,b,c (an infinite number!)
- Support Vector Machine (SVM) finds an optimal solution

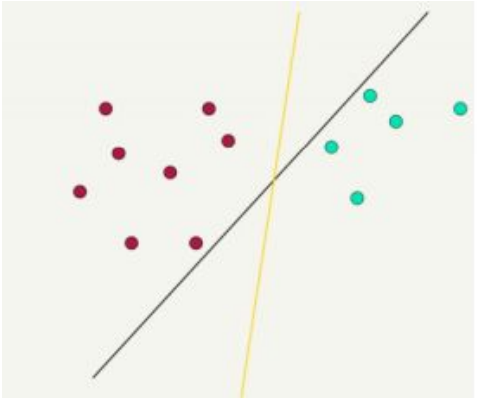
# 6. SVM

- SVMs maximize the margin (Winston terminology: the ‘street’) around the separating hyperplane.
- The decision function is fully specified by a (usually very small) subset of training samples, the support vectors.
- This becomes a Quadratic programming problem that is easy to solve by standard methods



# 6. SVM

- Lots of possible solutions for a,b,c.
- Some methods find a separating hyperplane, but not the optimal one (e.g., neural net)
- But: Which points should influence optimality? – All points?
- Linear regression
- Neural nets – Or only “difficult points” close to decision boundary
- Support vector machines



# Pros and Cons

| Advantages | Disadvantages |
| --- | --- |
| Works well then there is a clear margin of separation between classes | Not suitable for large data sets |
| Works well in high dimensional spaces | Does not perform well then there is more noise |

# 7. K- Nearest Neighbours(KNN)

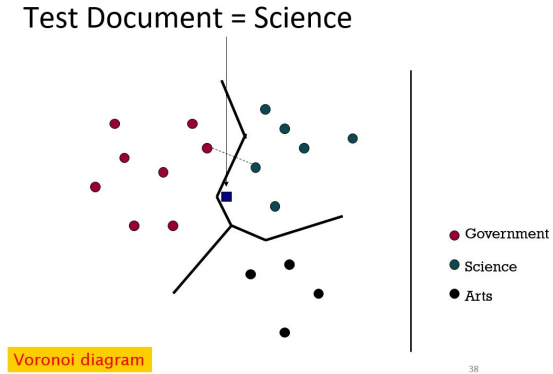
# 7. KNN

- KNN= K-nearest neighbour (based on Euclidean distance for example)

To classify a document d:

-Define k-neighborhood as the k nearest neighbors of d

-Pick the majority class label in the k-neighborhood



# 7. KNN

- Using only the closest example (1NN) subject to errors due to:
- A single atypical example.
- Noise (i.e., an error) in the category label of a single training example.
- More robust: find the k examples and return the majority category of these k
- k is typically odd to avoid ties; 3 and 5 are most common

# Pros and Cons

| Advantages | Disadvantages |
| --- | --- |
| No training period | Does not work well with large datasets |
| Easy to implement | Does not work well with high dimensions |

# THANKS!

**And much thanks to these ppl :D**

- https://www.saedsayad.com/naive_bayesian.htm
- NUS CS3244
- Hackwagon