LECTURE 3. CLASSIFICATION

AI & ML, Applications in Manufacturing

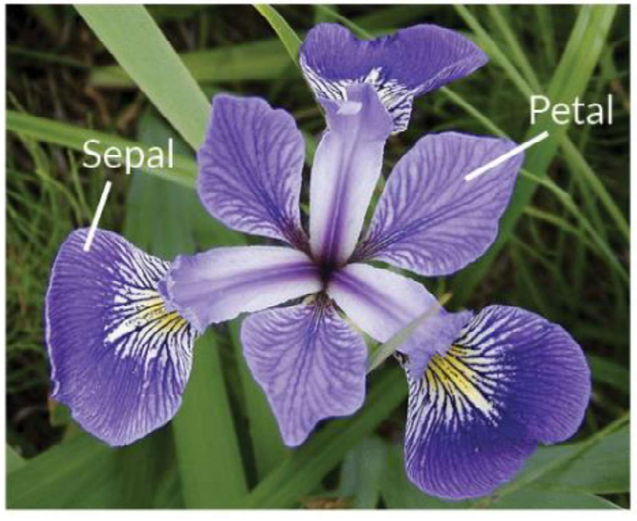
(MANU 465)

Ahmad Mohammadpanah

Ph.D., P.Eng.

AIntelligentManufacturing.com

What is classification?



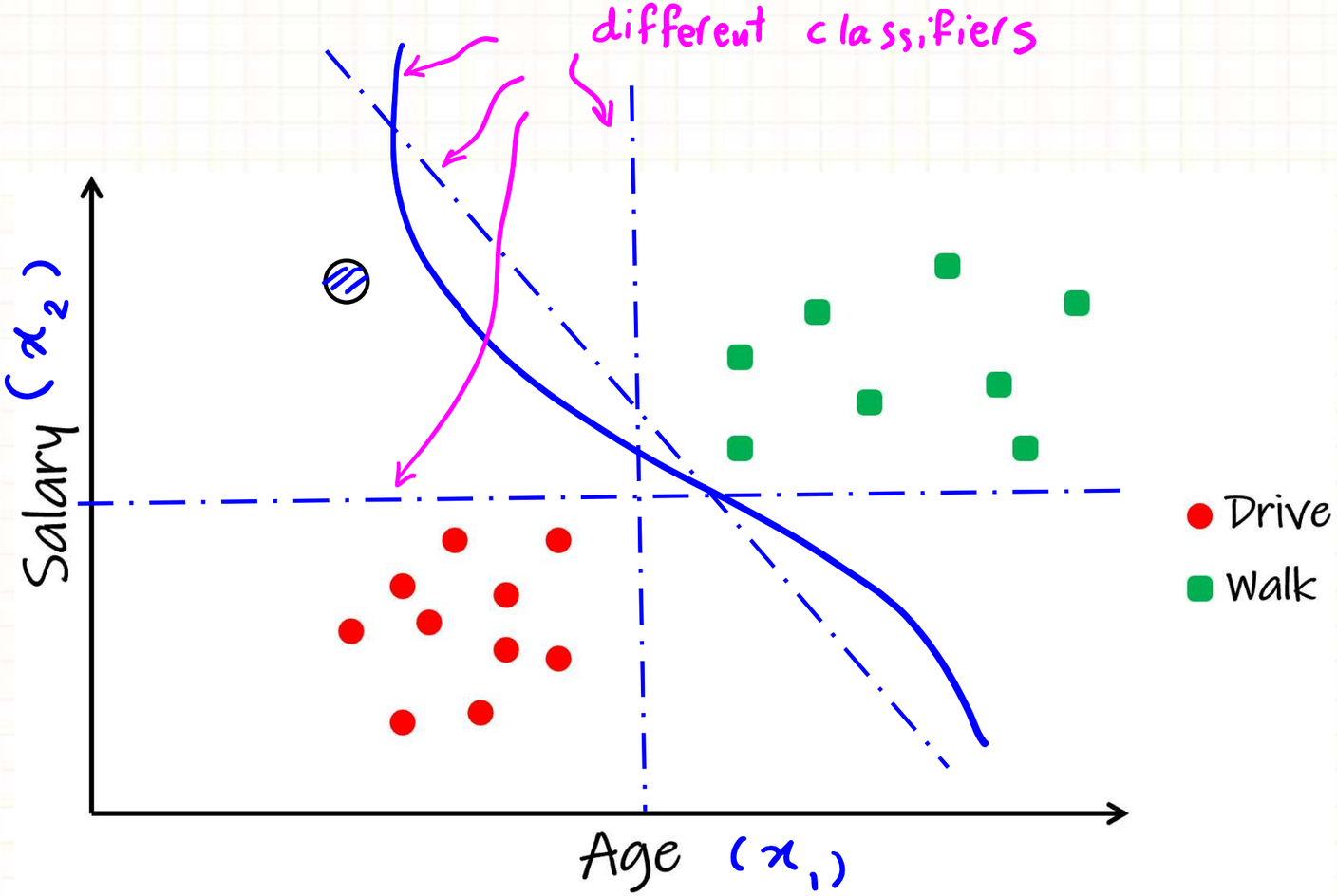
Iris Versicolor

Iris Setosa

Iris Virginica

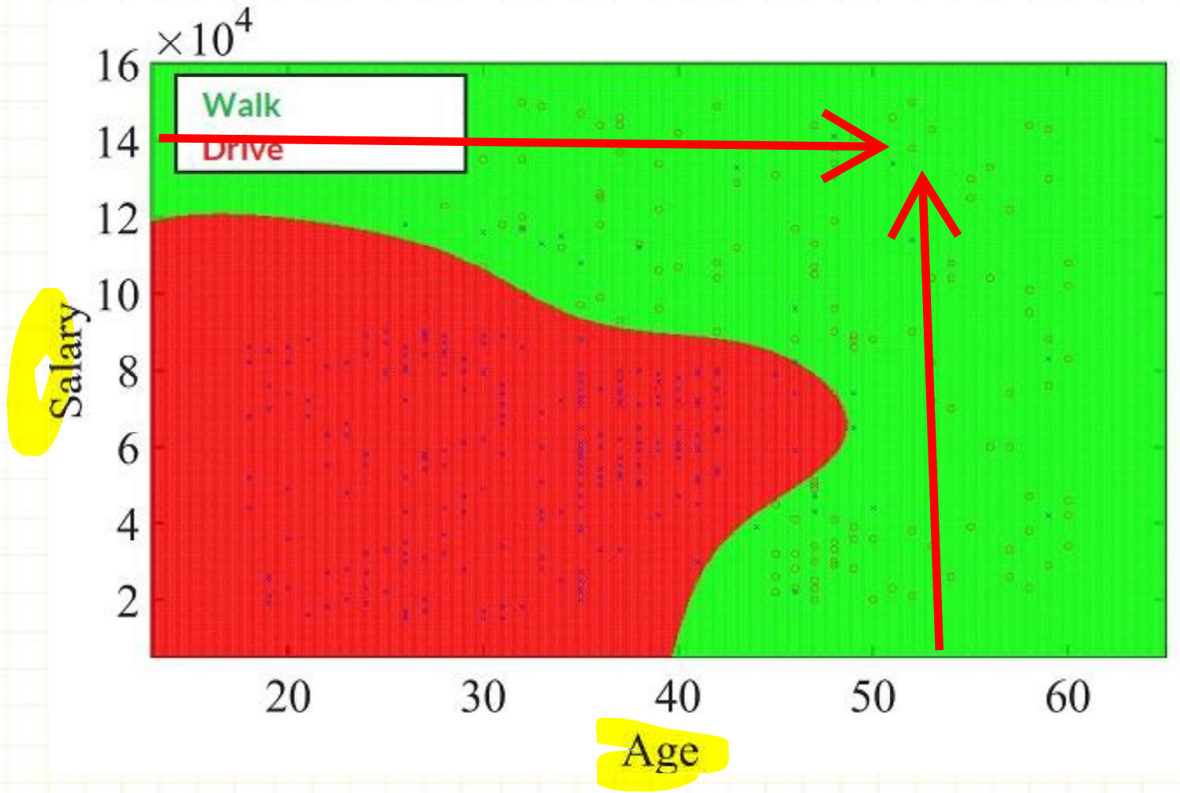
Suppose you gathered information on how people commute to UBC, their age and their salary. Can you use these data to predict if a new person will walk or drive based on his age and salary?

People	Age	Salary	Commute
1	21	25k	walk
2	23	100k	walk
3	19	50k	Drive
4	.	45	walk
5	32	.	walk
.	.	.	.
3000	24	75k	Drive



Classifier:

An algorithm for finding the boundary between classes.



Some Common algorithms:

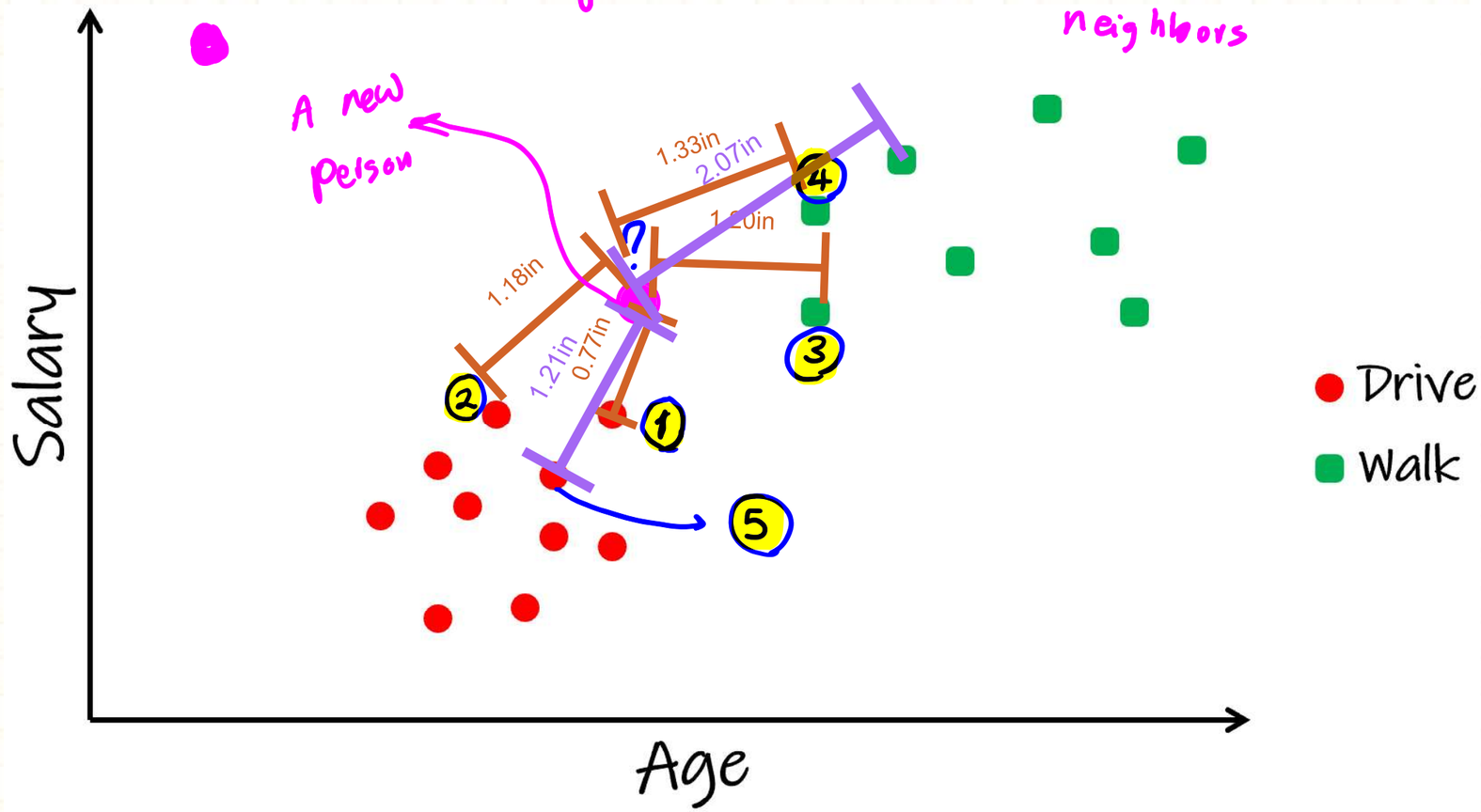
- K-Nearest Neighbors (K-NN)
- Support Vector Machine (SVM)
- Kernel SVM
- Naive Bayes
- Decision Tree
- Random Forest

K-NN (K-Nearest Neighbor)

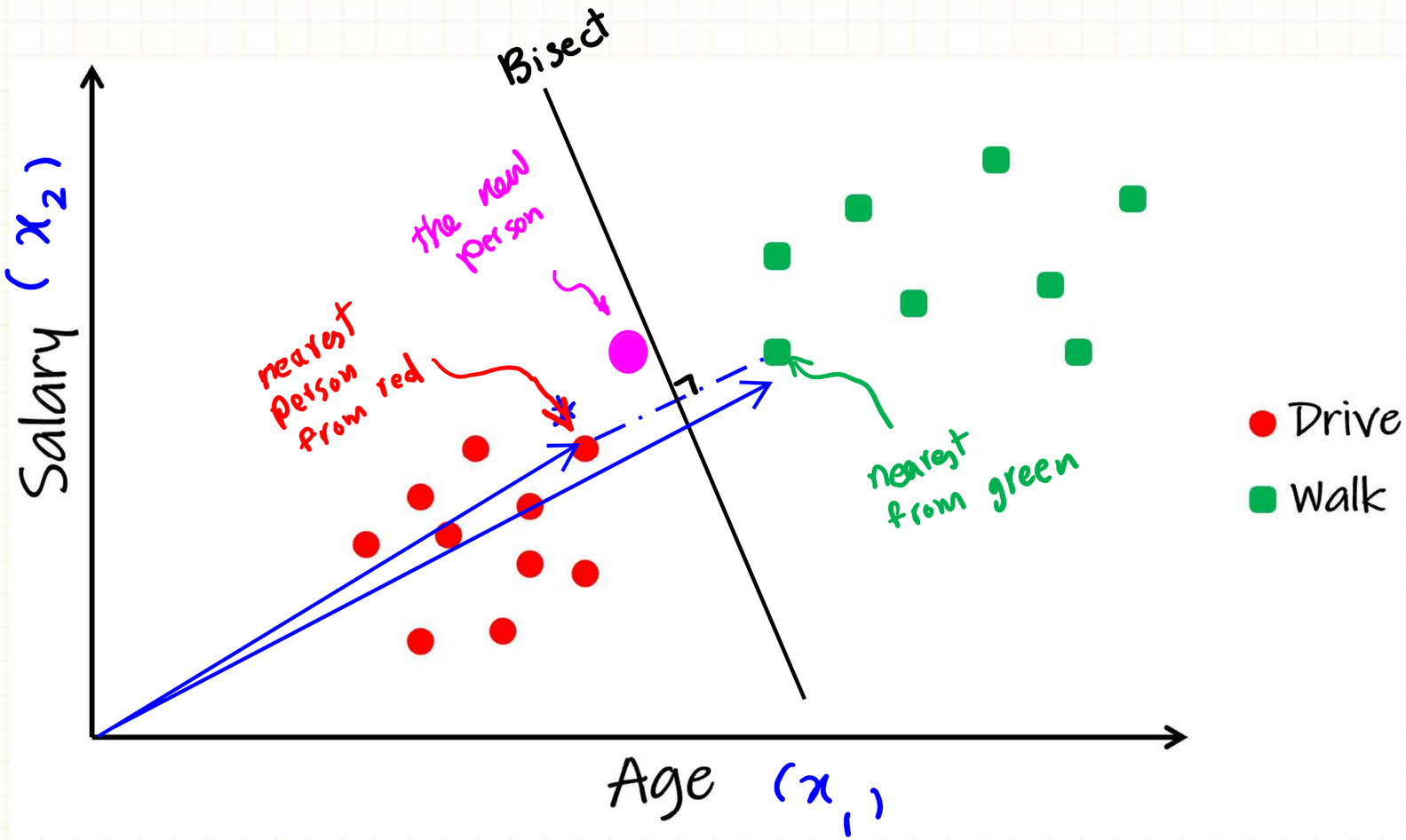
* k = 5 → { 3 Drive, 2 Walk } → the new person Drives.

choose the number of neighbors

find the nearest neighbors

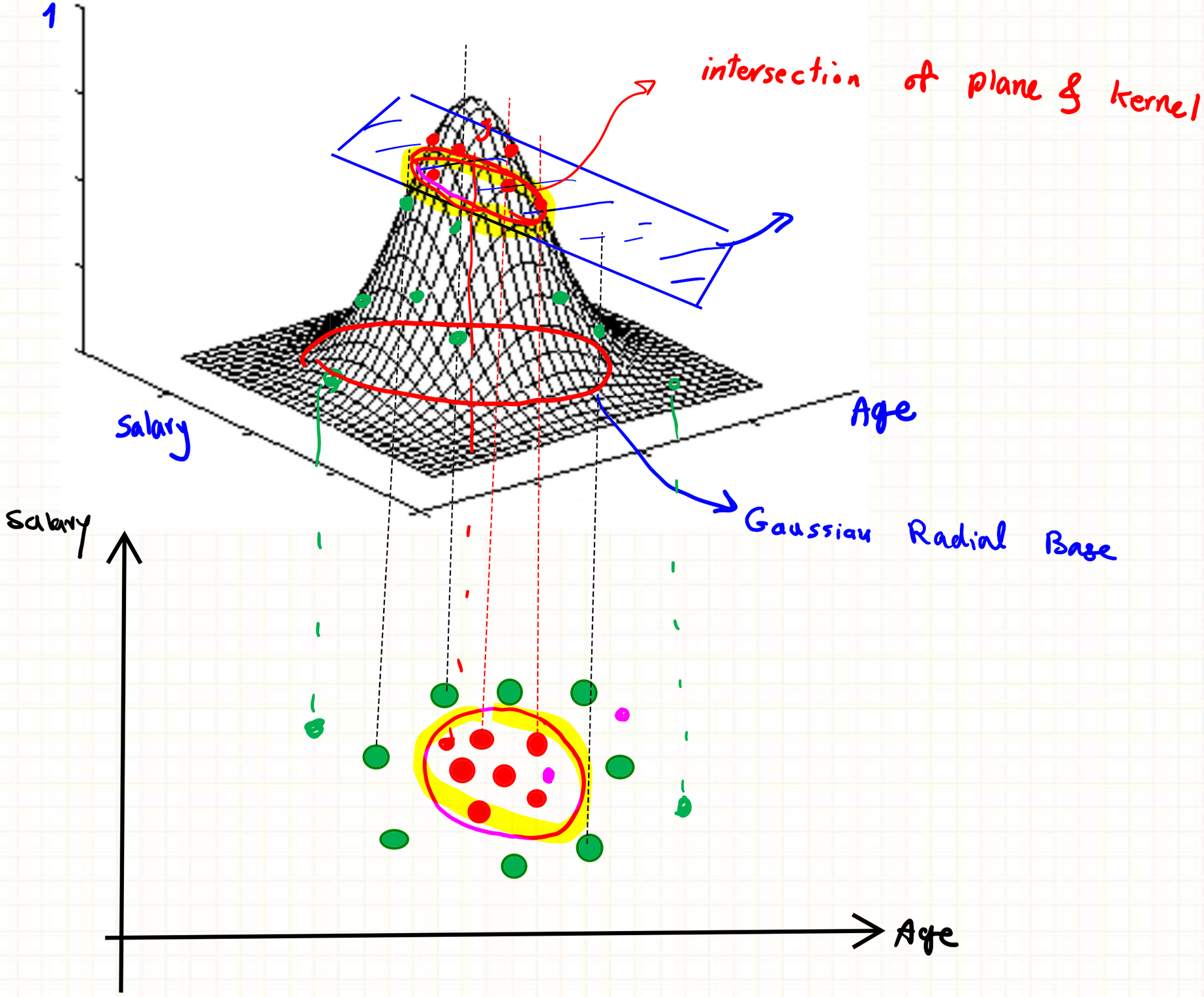


SVM (Support Vector Machine)

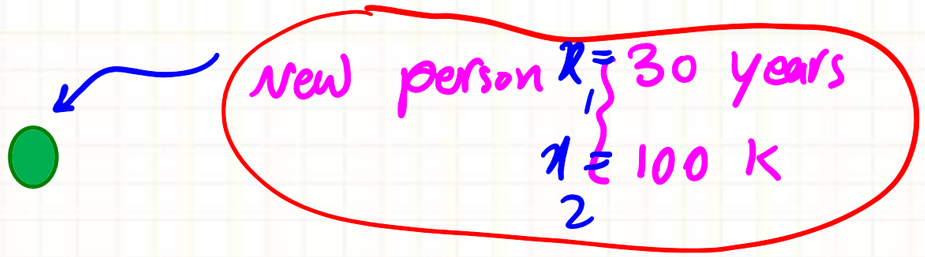


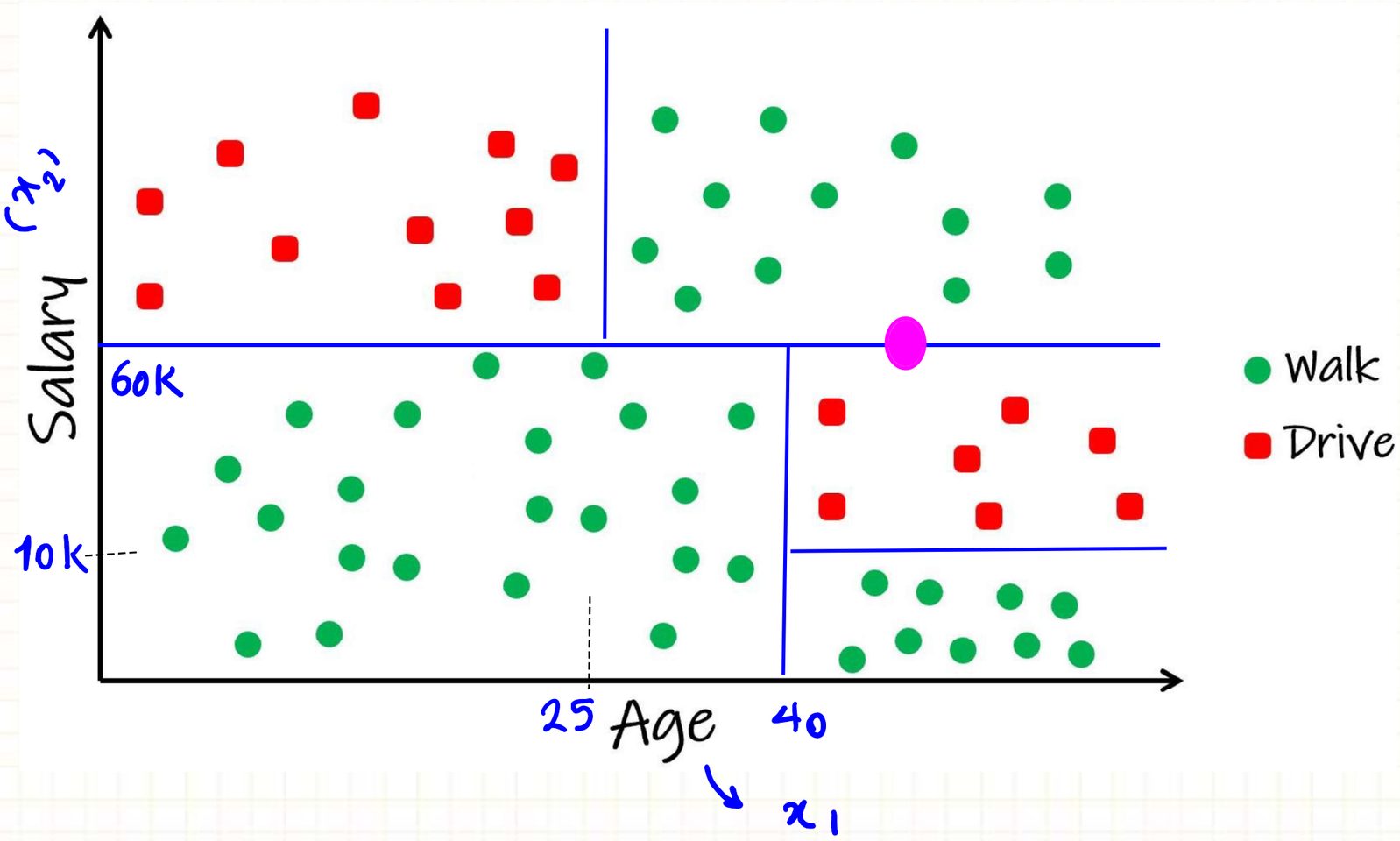
Kernel SVM

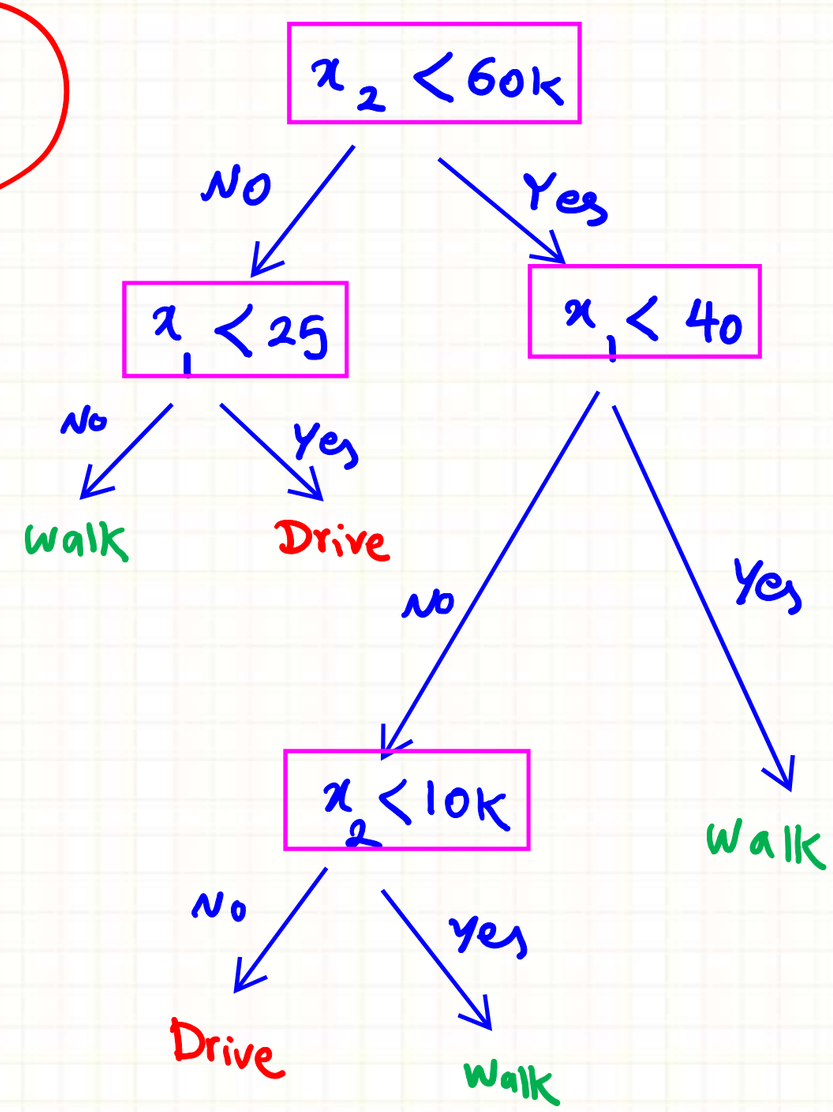
↳ Gaussian distribution



Decision Tree





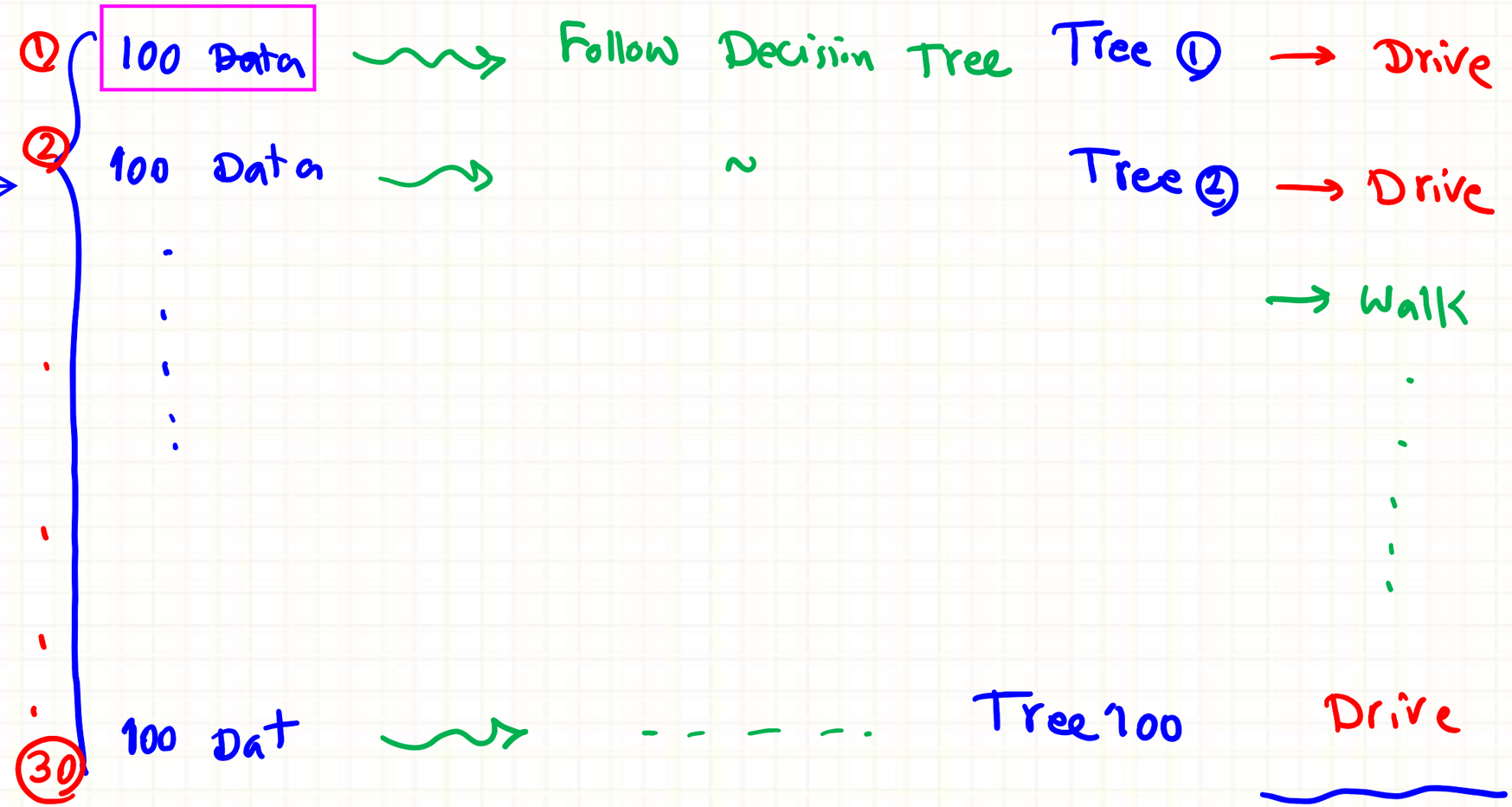


Random Forest

total Data

3000

Divide the data to 100 batches, then make a tree for each

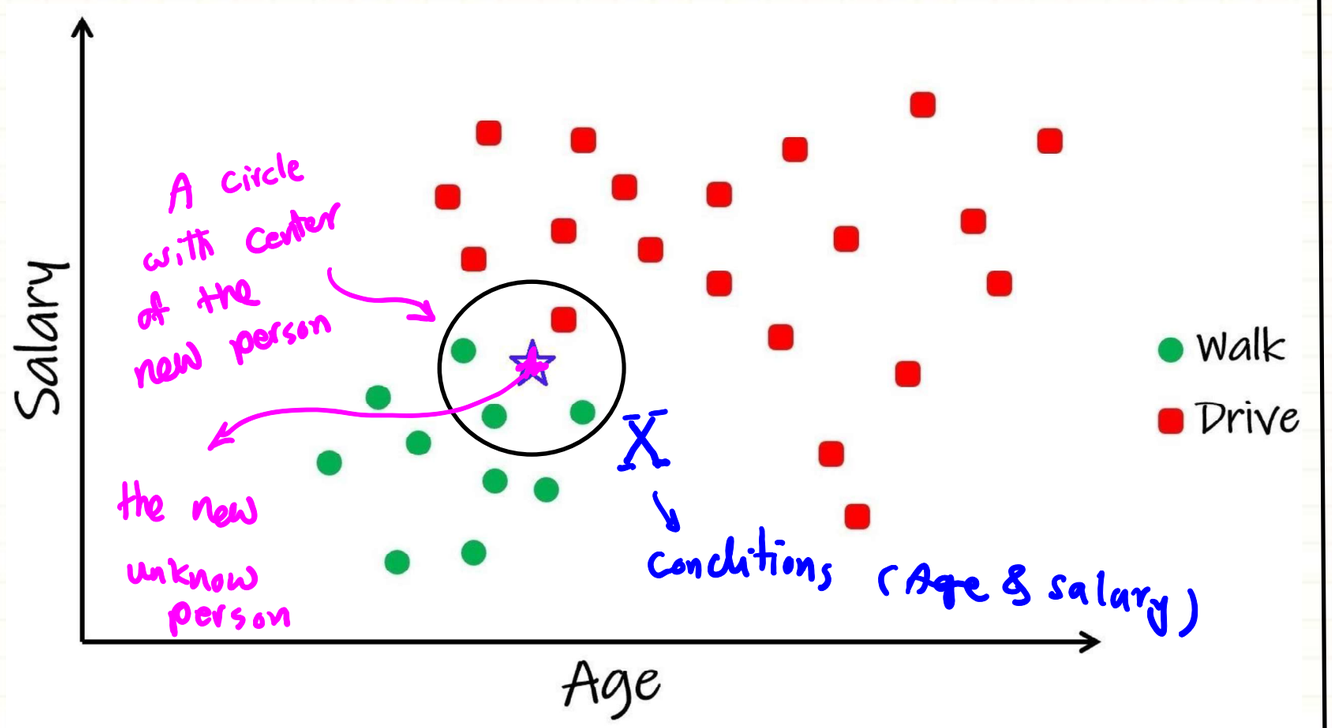


Naïve Bayes

$$P(A \mid B) = \frac{P(A)\,P(B \mid A)}{P(B)}$$

probability of event A | Given | Condition

Let's get back to our Salary-Age and Commute to UBC scenario, to see how Naïve Bayes predict the status of a new person. Does the new person (☆) walk or drive?



we try to find the probability if the new person will "Drive" or "walk" given its condition X (Age, salary)

$$P(\text{Drive} \mid X) = \frac{P(\text{Drive})\,P(X \mid \text{Drive})}{P(X)} = ?$$

$$P(\text{Walk} \mid X) = \frac{P(\text{Walk})\,P(X \mid \text{Walk})}{P(X)} = ?$$

given

total # of people = 30

total # of Drivers = 20

" " " Walker = 10

$$P(\text{Drive}) = \frac{20}{30}$$

$$P(\text{walk}) = \frac{10}{30}$$

$$P(X) = \frac{\text{total \# of people inside circle}}{\text{total \# of People}}$$

$$P(X) = \frac{4}{30}$$

$$P(X \mid \text{Drive}) = \frac{\text{total \# of Driver inside circle}}{\text{total \# of Driver}} = \frac{1}{20}$$

$$P(X \mid \text{walk}) = \frac{3}{10}$$

$$P(\text{Drive} \mid X) = \frac{P(\text{Drive})\,P(X \mid \text{Drive})}{P(X)} = 25\%$$

$$P(\text{Walk} \mid X) = \frac{P(\text{walk})\,P(X \mid \text{walk})}{P(X)} = 75\%$$

How to build a classifier:

Step 1. Data Standardization

Step 2. Splitting Data into "Training set" and "Test set".

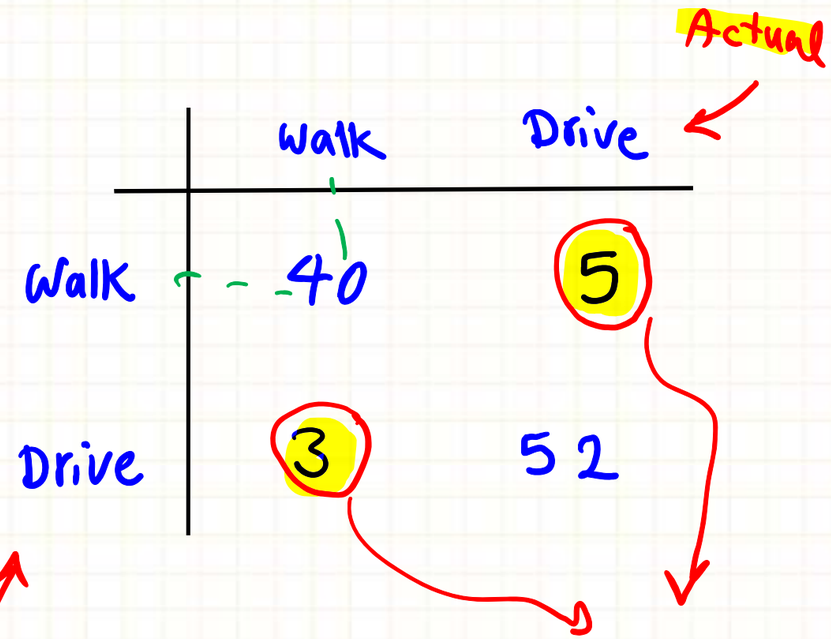
Step 3. Build the model based on the training set.

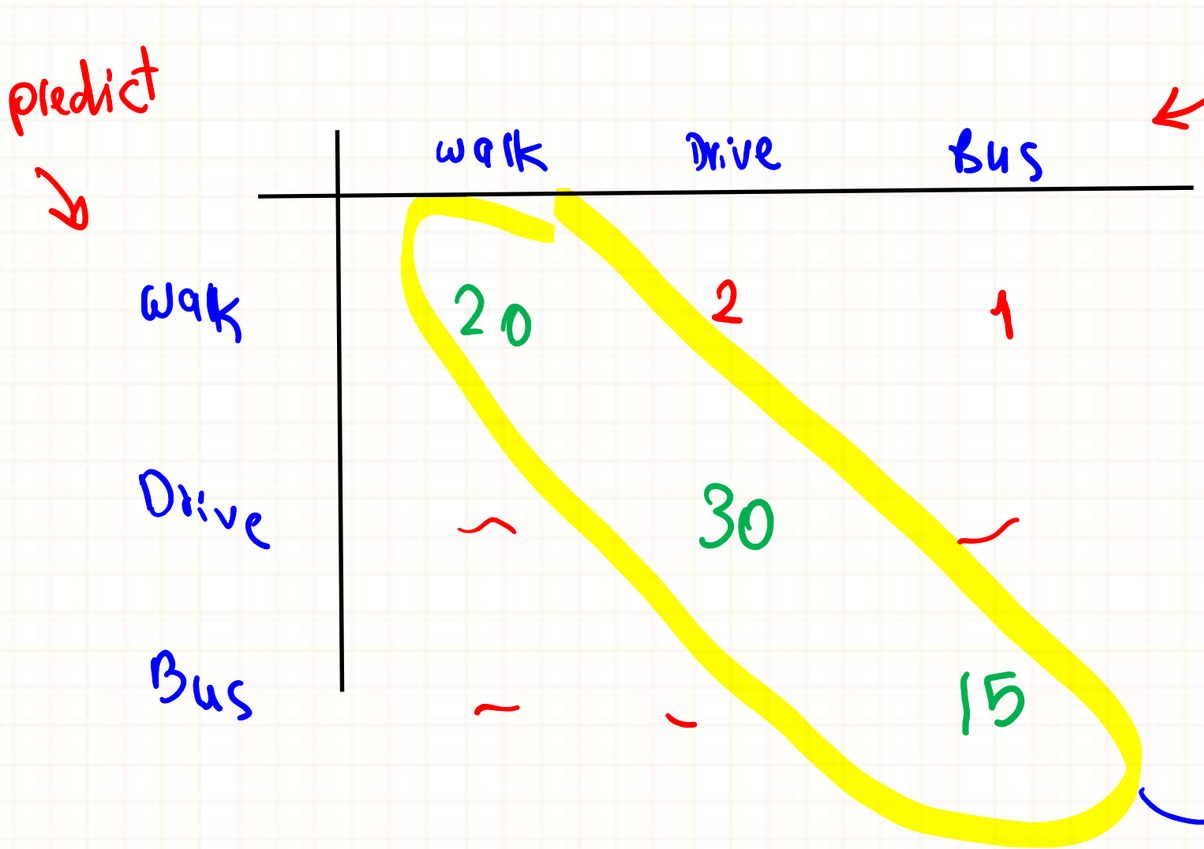
Step 4. Test the model accuracy based on test set (Confusion Matrix).

Step 5. Use the model to predict new inputs.

What is a Confusion Matrix ?

on Test set (100 people)





$$\text{Model Accuracy} = \frac{40+52}{100} = 92\%$$

Implement Classifiers in Python

KNN

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 3)
classifier.fit(X, y)
```

(X_{train}, Y_{train})

SVM

```
from sklearn.svm import SVC
SVMclassifier = SVC(kernel = 'rbf')
SVMclassifier.fit(X, y)
```

NaiveB

```
from sklearn.naive_bayes import GaussianNB
NBclassifier = GaussianNB()
NBclassifier.fit(X, y)
```

Tree

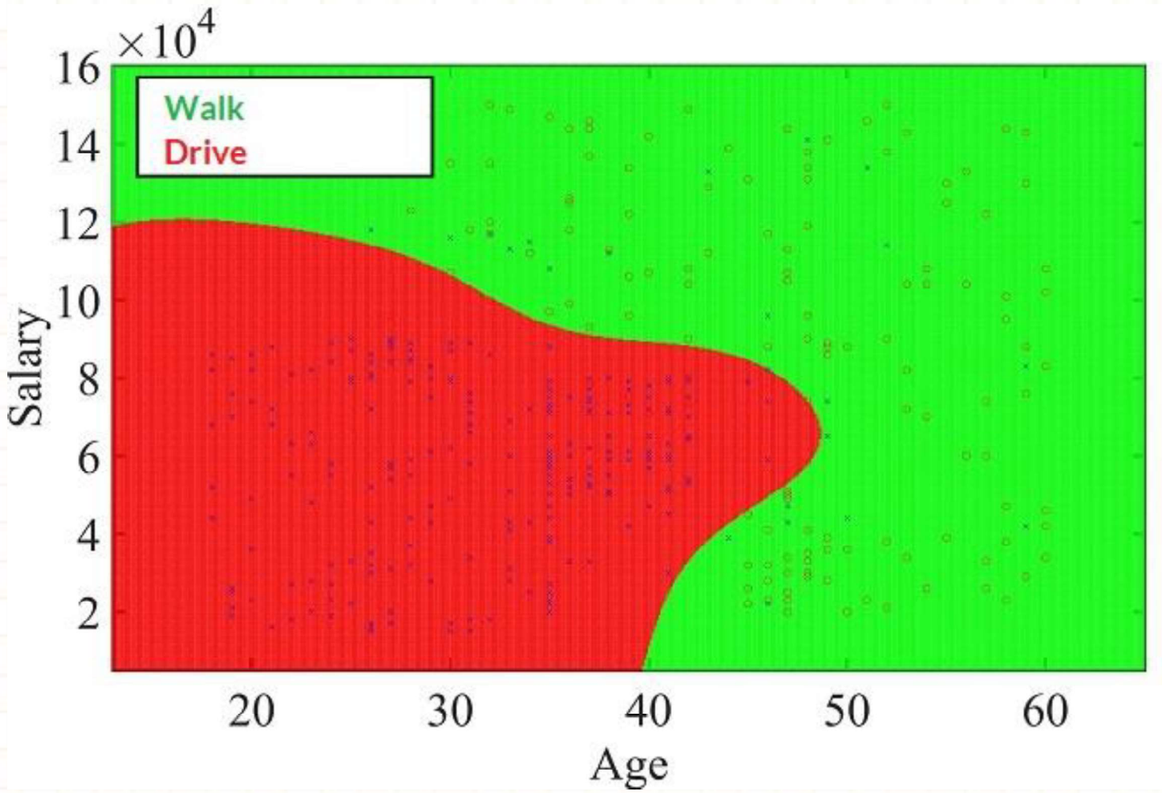
```
from sklearn.tree import DecisionTreeClassifier
Treeclassifier = DecisionTreeClassifier()
Treeclassifier.fit(X, y)
```

Forest

```
from sklearn.ensemble import RandomForestClassifier
Forestclassifier = RandomForestClassifier(n_estimators = 10)
Forestclassifier.fit(X, y)
```

Summary:

- Each classifier might generate a different prediction model.
- The boundary might be linear or non-linear.
- The input can be more than 2 parameters (age, salary, weight, …), or categorical/logistic/binary parameters (gender, nationality, …). More than 3 parameters will be just impossible to visualize!



- You will learn how to apply these classification concepts and build a classification model in Python, in Tutorial 2, and Canvas/Examples 7-1 1