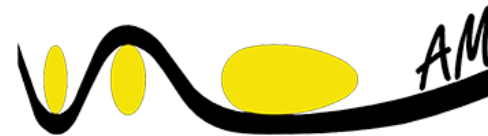


LECTURE 3. CLASSIFICATION

AI & ML, Applications in Manufacturing (MANU 465)

Ahmad Mohammadpanah

Ph.D., P.Eng.



AlntelligentManufacturing.com

What is classification?





Iris Versicolor



Iris Setosa

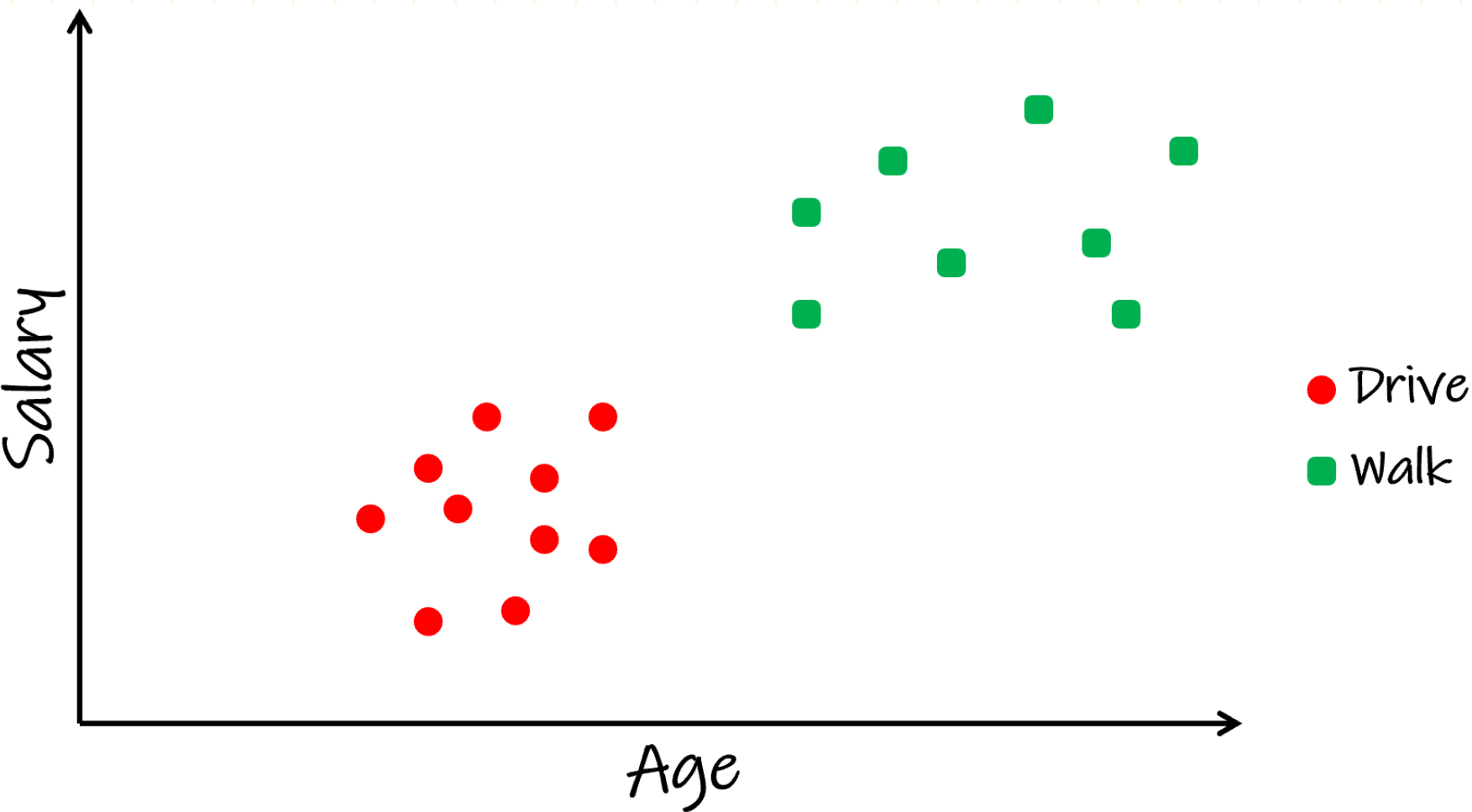


Iris Virginica



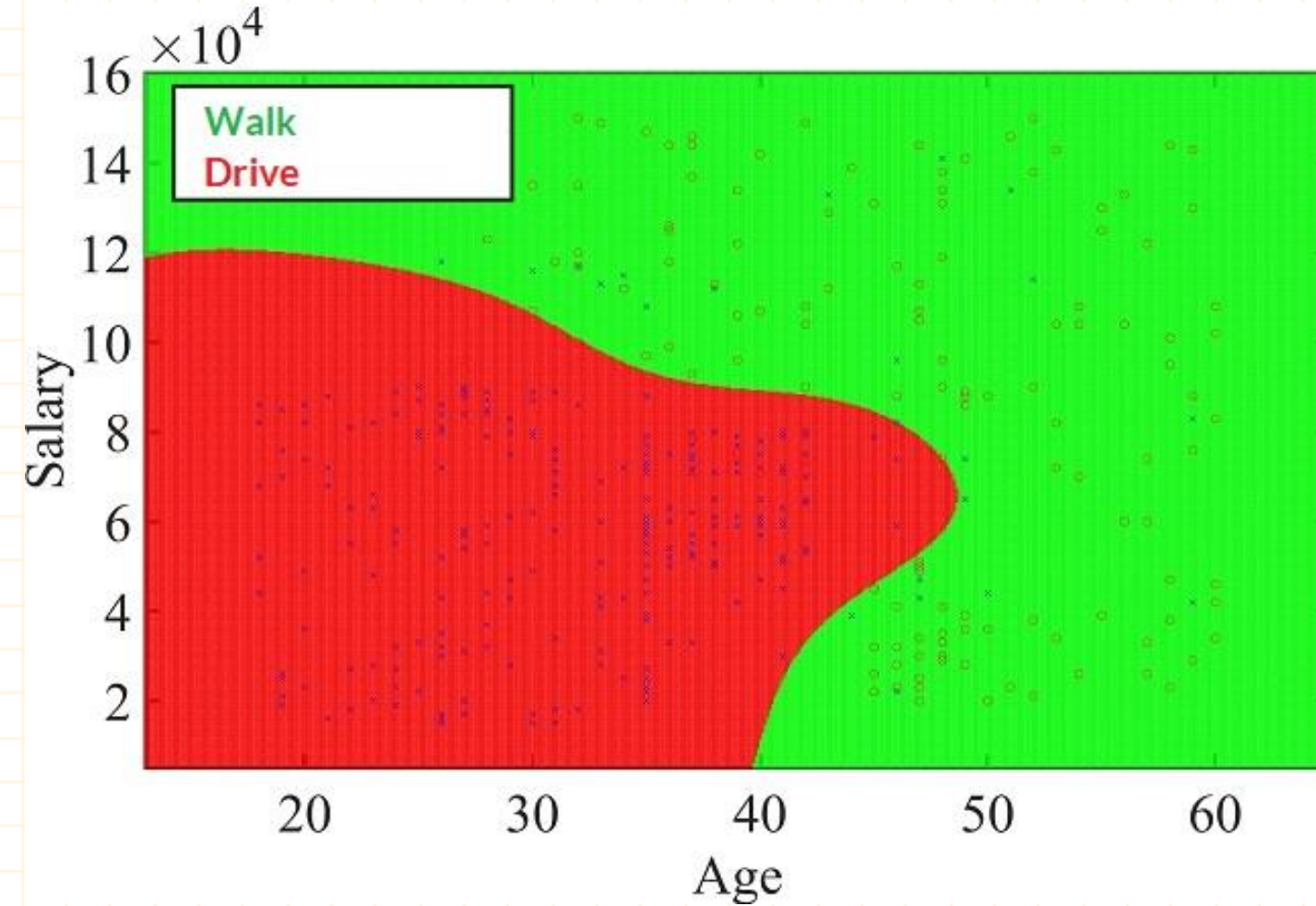
Suppose you gathered information on how people commute to UBC, their age and their salary. Can you use these data to predict if a new person will walk or drive based on his age and salary?

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:

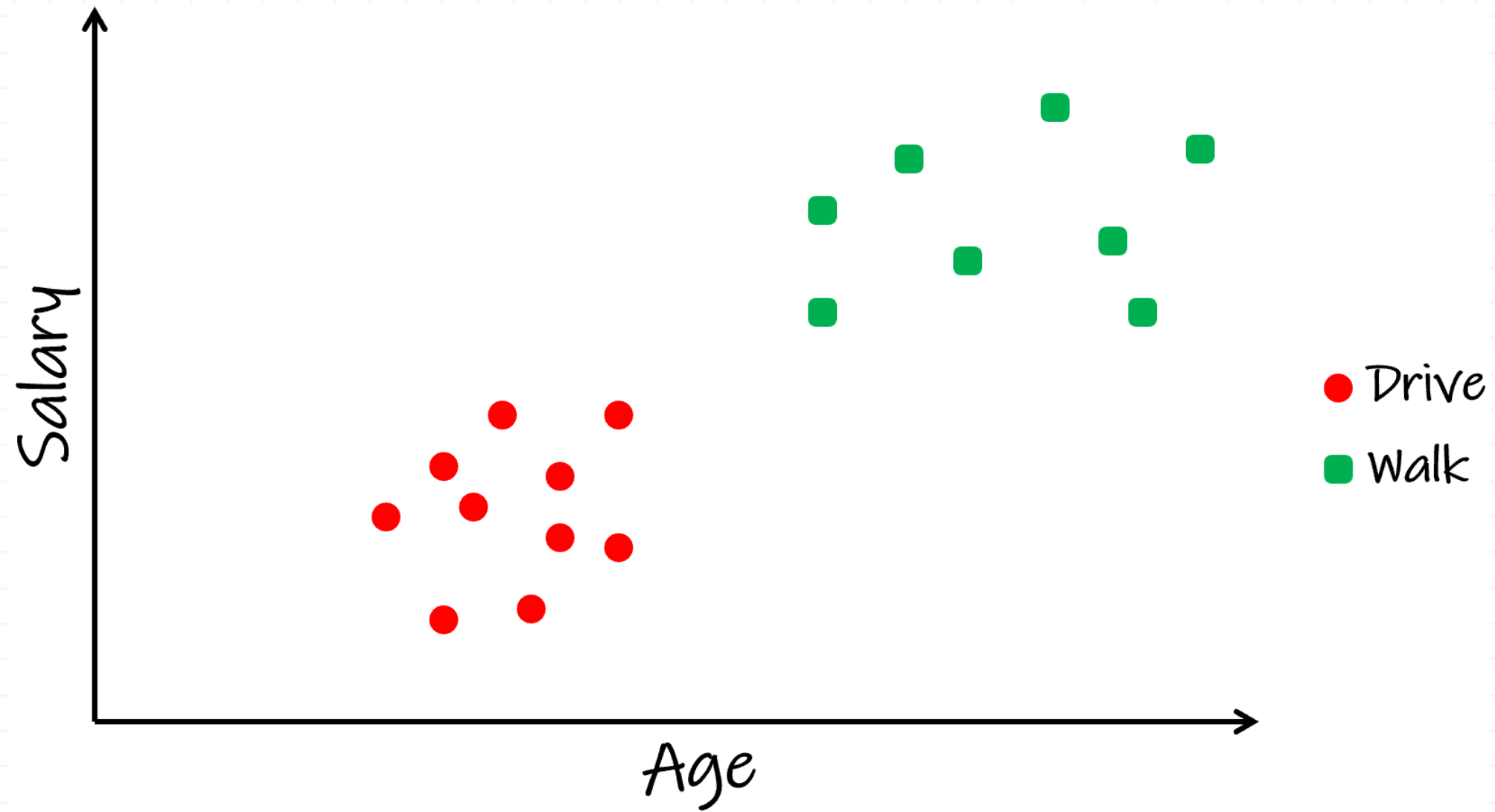
An algorithm for finding the boundary between classes.



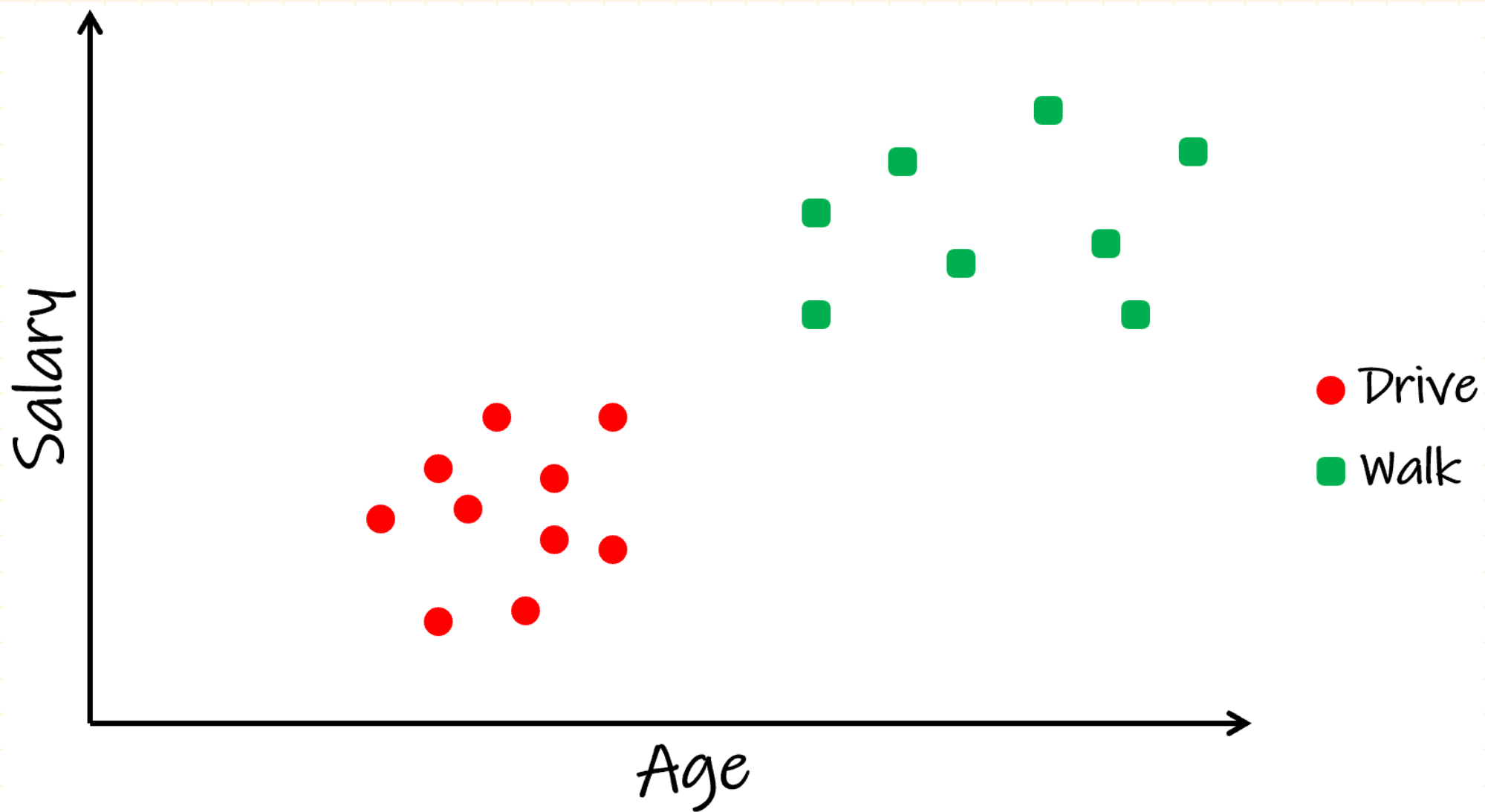
Some Common algorithms:

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

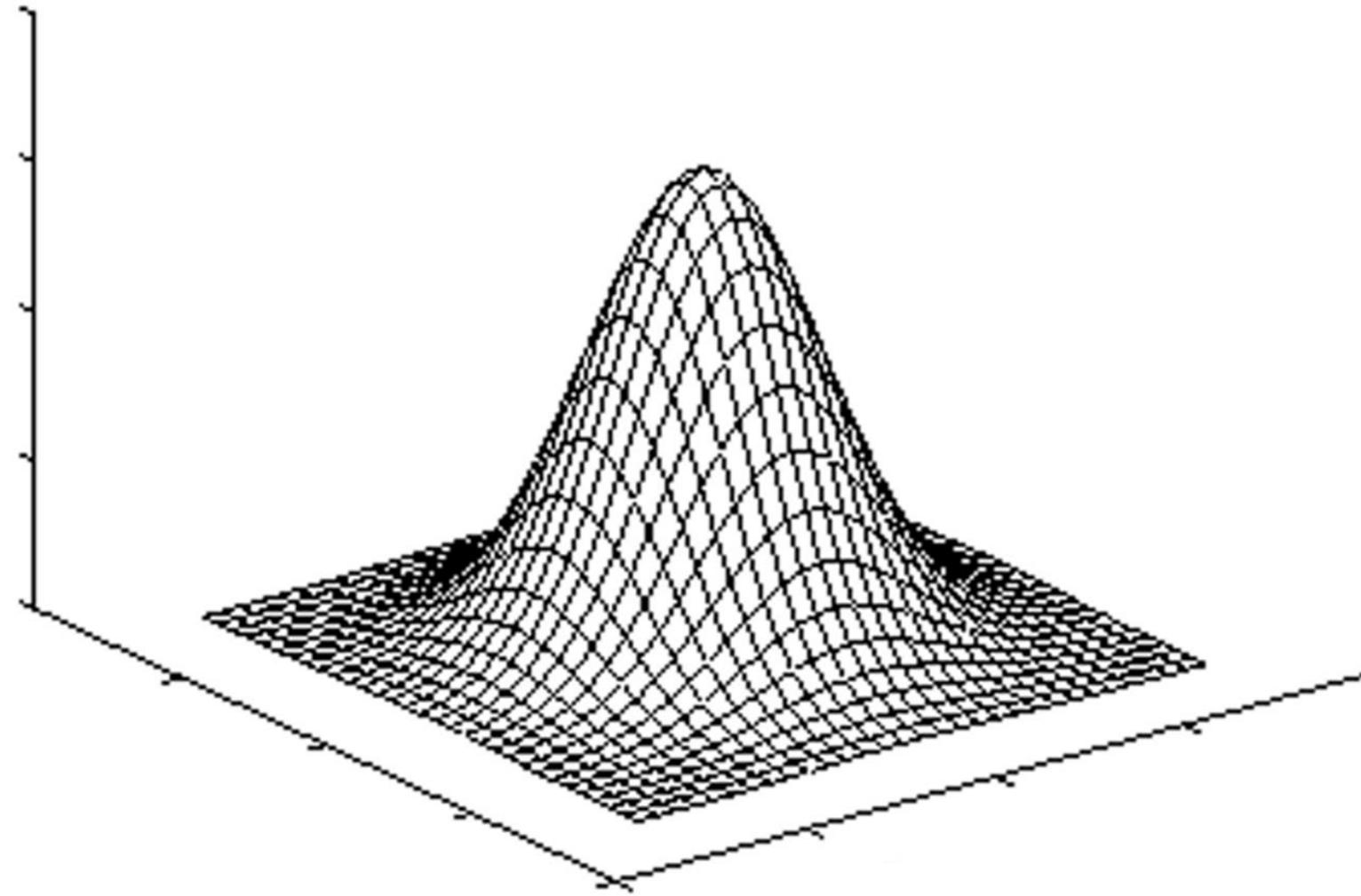
K-NN (K-Nearest Neighbor)



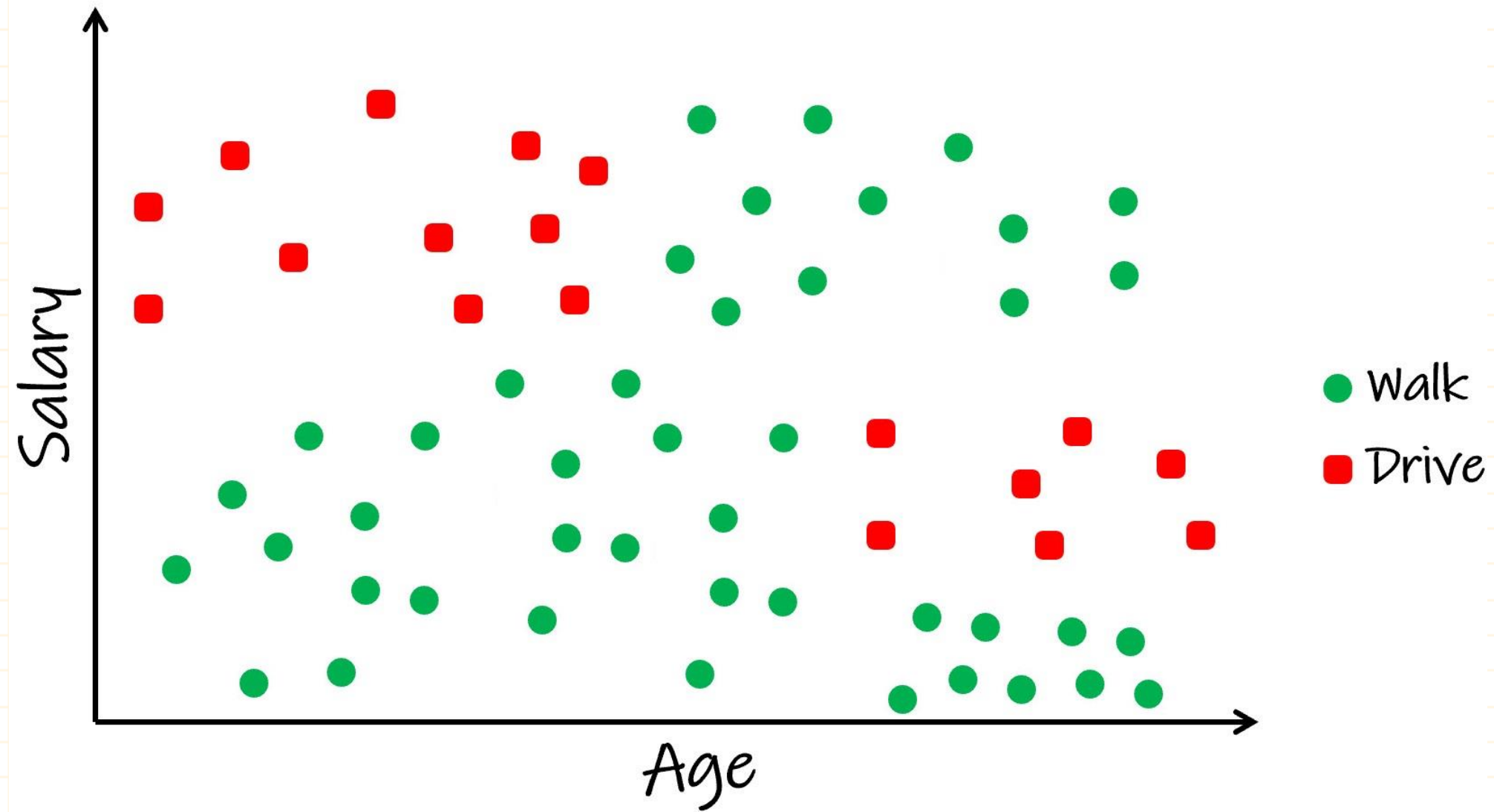
SVM (Support Vector Machine)



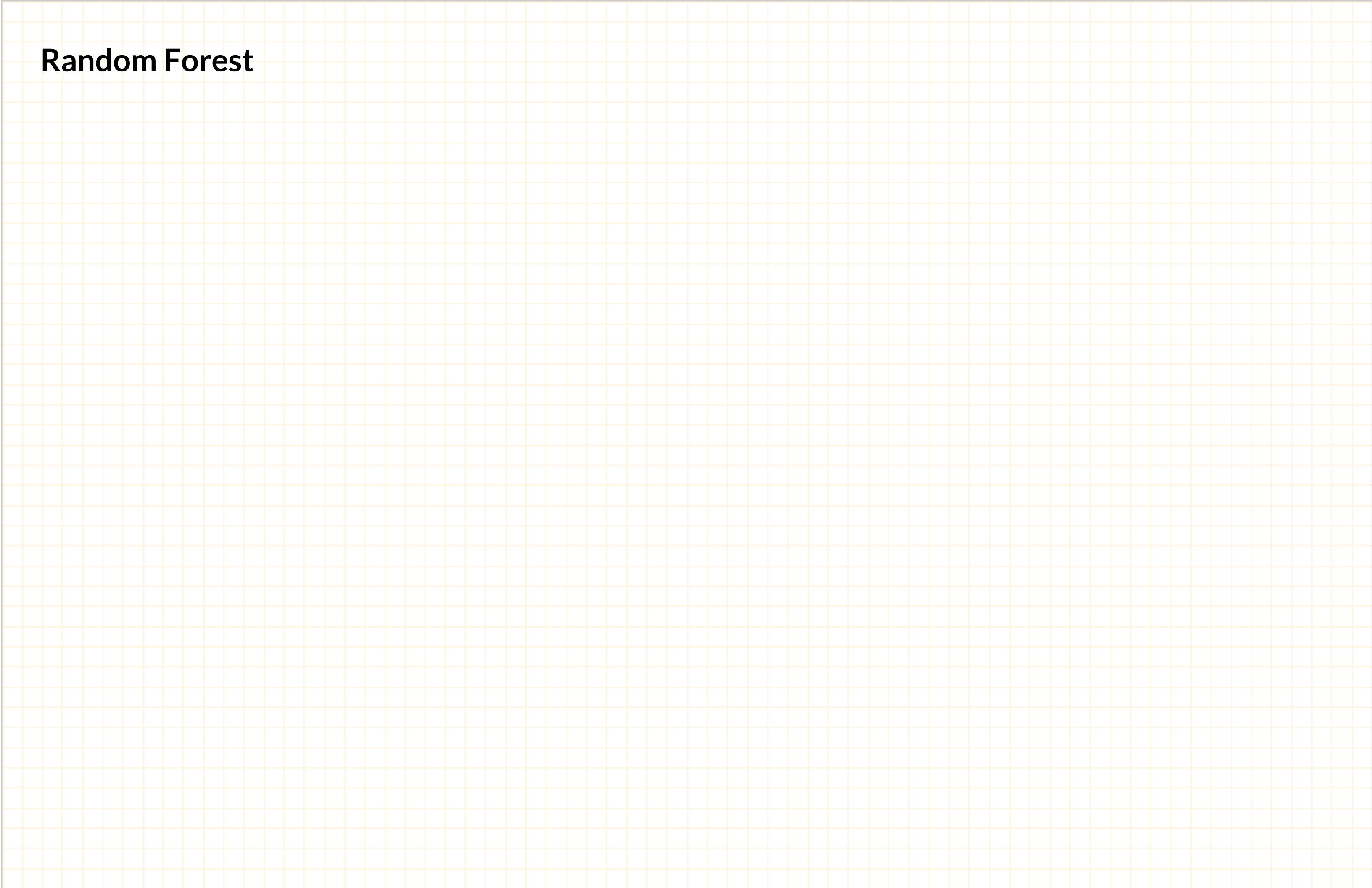
Kernel SVM



Decision Tree

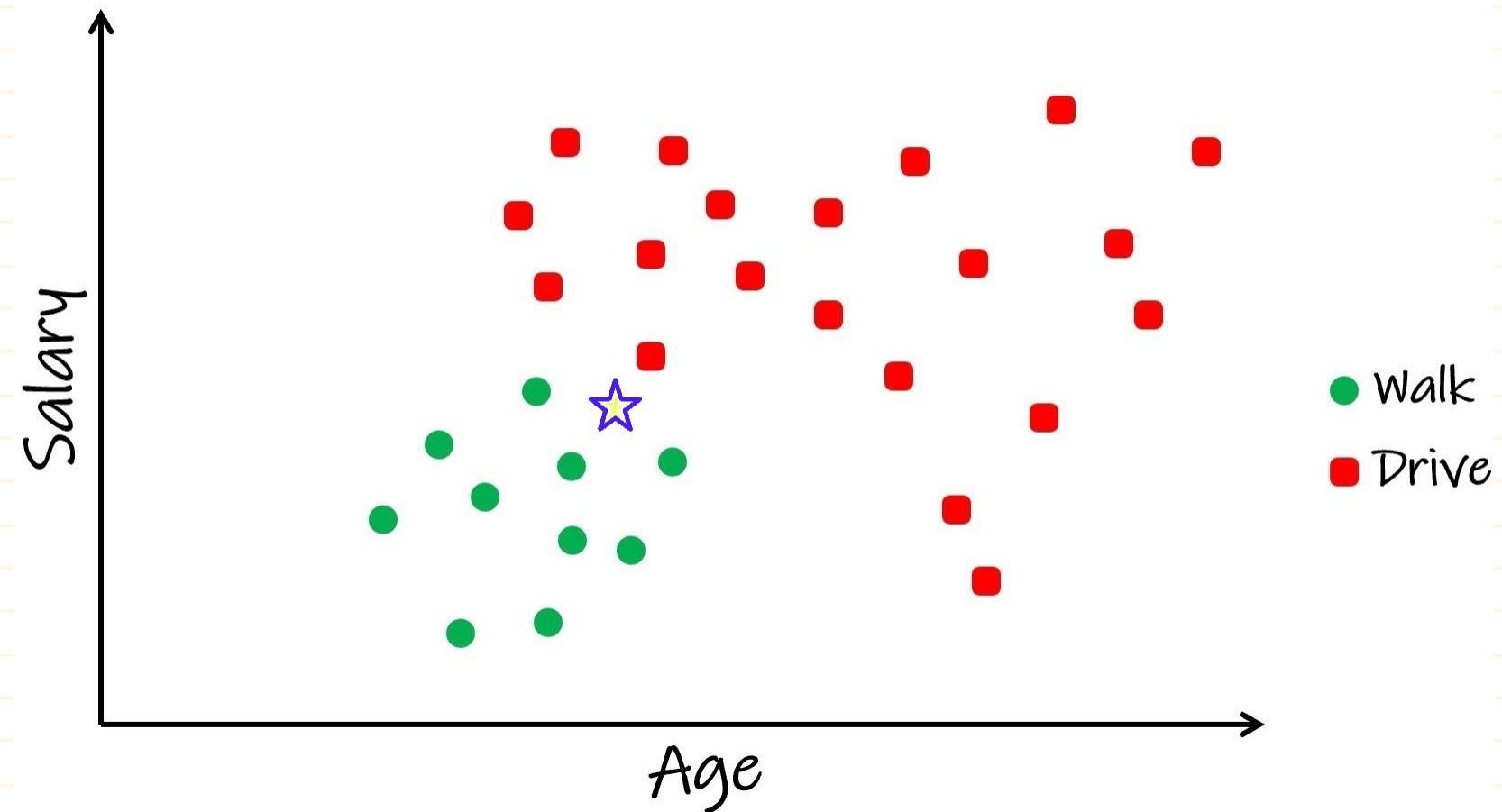


Random Forest



Naïve Bayes

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?



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.

Implement Classifiers in Python

KNN

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

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

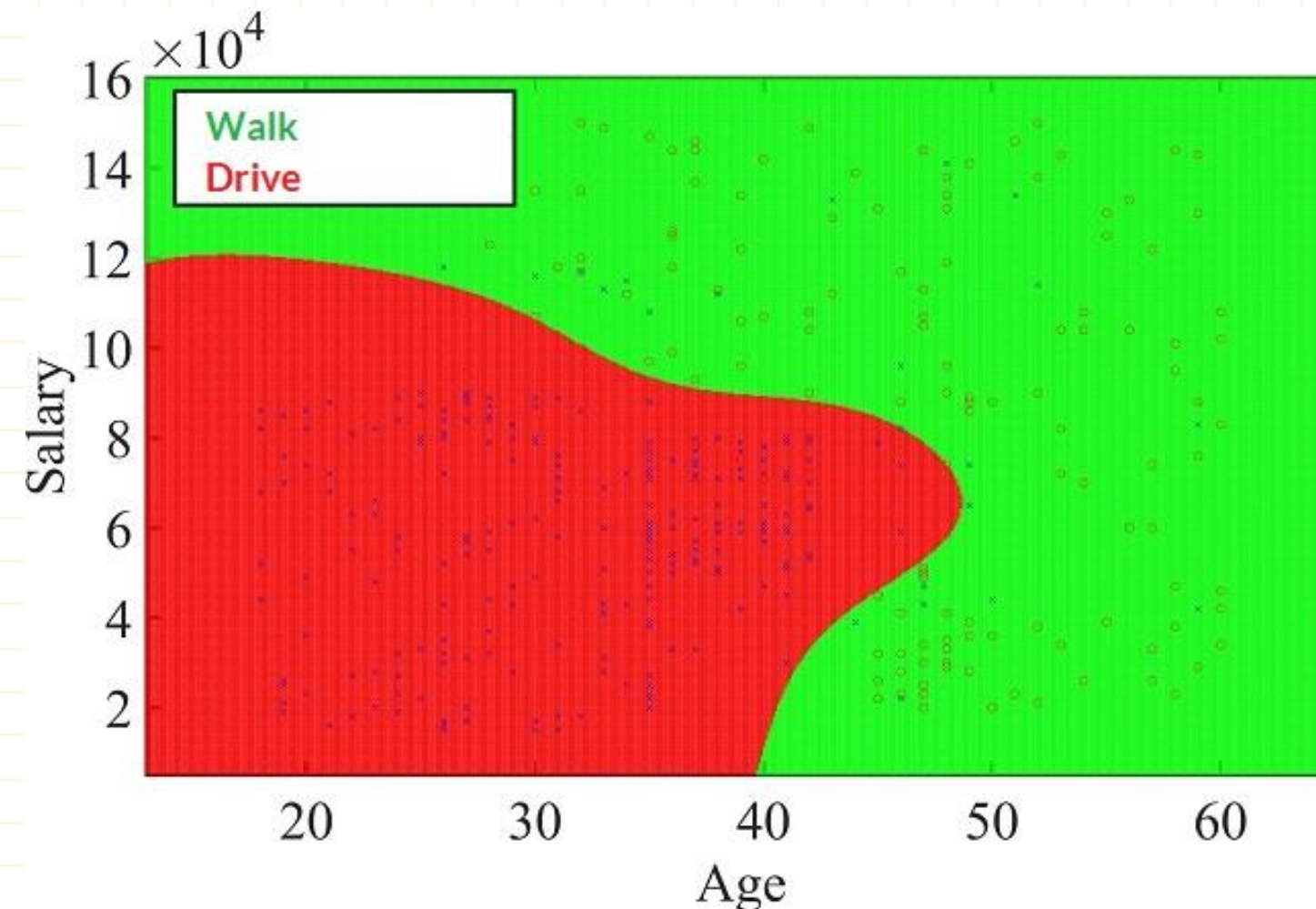
```
from sklearn.tree import DecisionTreeClassifier  
Treeclassifier = DecisionTreeClassifier()  
Treeclassifier.fit(X, y)
```

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 7-12](#)