Data Analysis Lab

Support Vector Machines
K-Nearest Neighbours

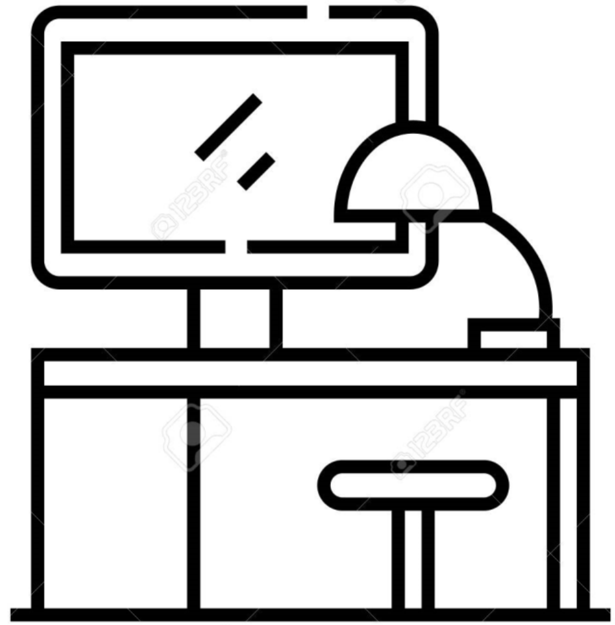
Fátima Leal

DEPARTAMENTO CIÊNCIA E TECNOLOGIA

UNIVERSIDADE PORTUCALENSE

Previous Lesson

- Classification metrics
- Naïve Bayes



Outline

- Support Vector Machines
- Practical Examples
- K-Nearest Neighbours
- Practical Examples
- Exercises

DEPARTAMENTO CIÊNCIA E TECNOLOGIA

Support Vector Machines

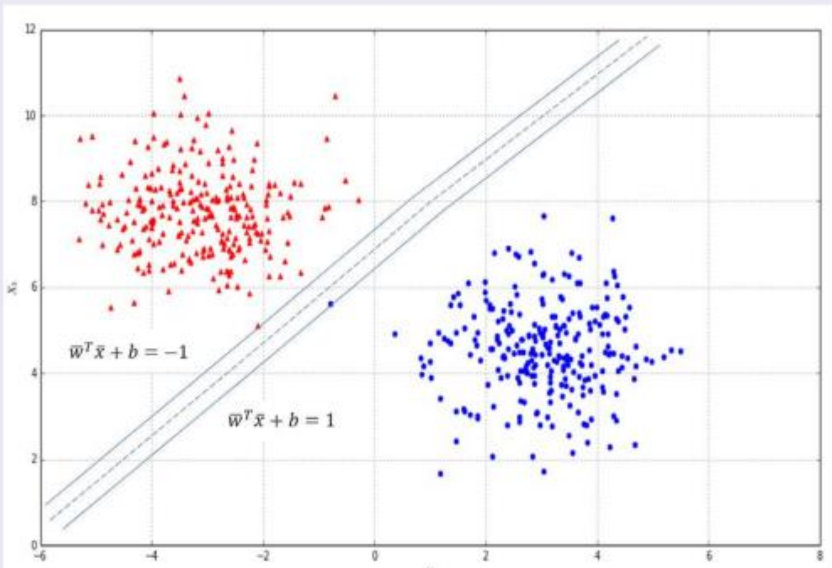
- **Support Vector Machines (SVM)** can work both linear and non-linear scenarios, allowing high performance in many different contexts.
- SVM probably represent the best choice for many tasks where it's not easy to find good separate hyperplane
- They can capture very high non-linear dynamics using a mathematical trick, without complex modifications in the algorithm

Support Vector Machines

- Let's consider a dataset of feature vectors we want to classify: $X = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_n\}$, where $\bar{x}_i \in \mathbf{R}^m$
- For simplicity, we assume it as a binary classification (in all the other cases, it's possible to use automatically the one-versus-all strategy) and we set our class labels as -1 and 1: $Y = \{y_1, y_2, ..., y_n\}$, where $y_n \in \{-1, 1\}$
- Our goal is to find the best separating hyperplane, for which the equation is:$\bar{w}^T\bar{x} + b = 0$, where $\bar{w} = W_1...W_m)$ and $\bar{x} = x_1...x_m)$
- In this way, our classifier can be written as: $\bar{y} = f(\bar(x)) = sgn(\bar{w}^T\bar{x} + b)$

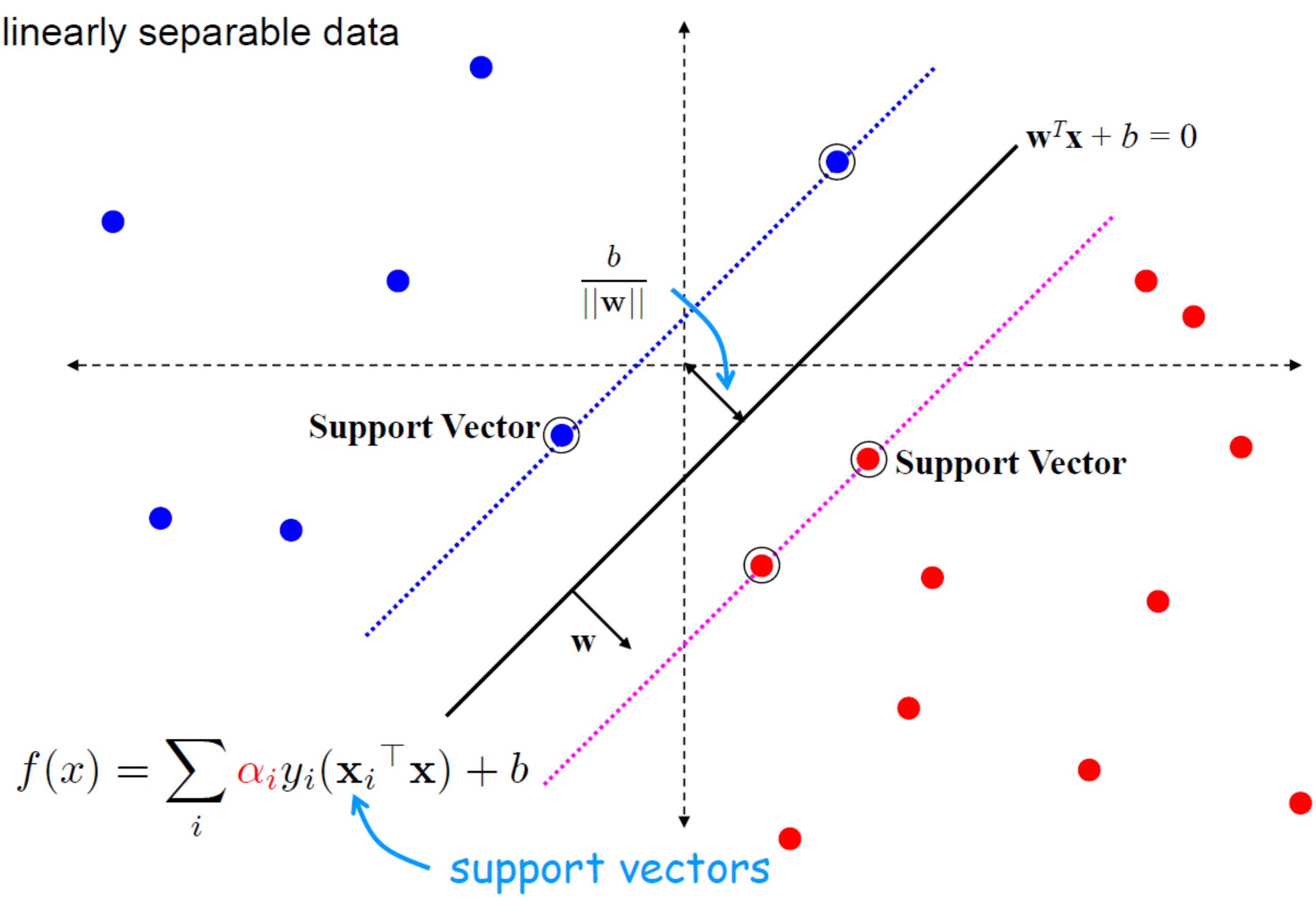
Support Vector Machines

- In a realistic scenario, the two classes are normally separated by a margin with two boundaries where a few elements lie. Those elements are called support vectors. For a more generic mathematical expression, it's preferable to renormalize our dataset so that the support vectors will lie on two hyperplanes with equations: $\bar{w}^T\bar{x} + b = -1$ and $\bar{w}^T\bar{x} + b = 1$

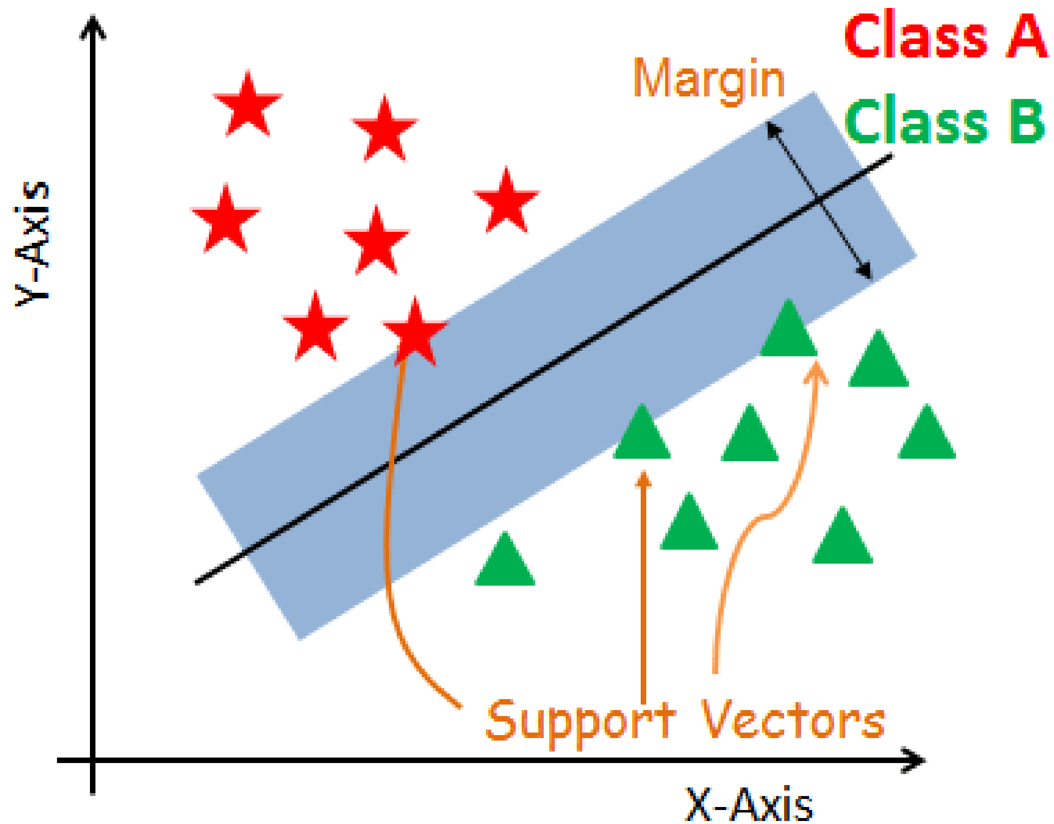


UPT DCT DEPARTAMENTO CIÊNCIA E TECNOLOGIA

Support Vector Machines



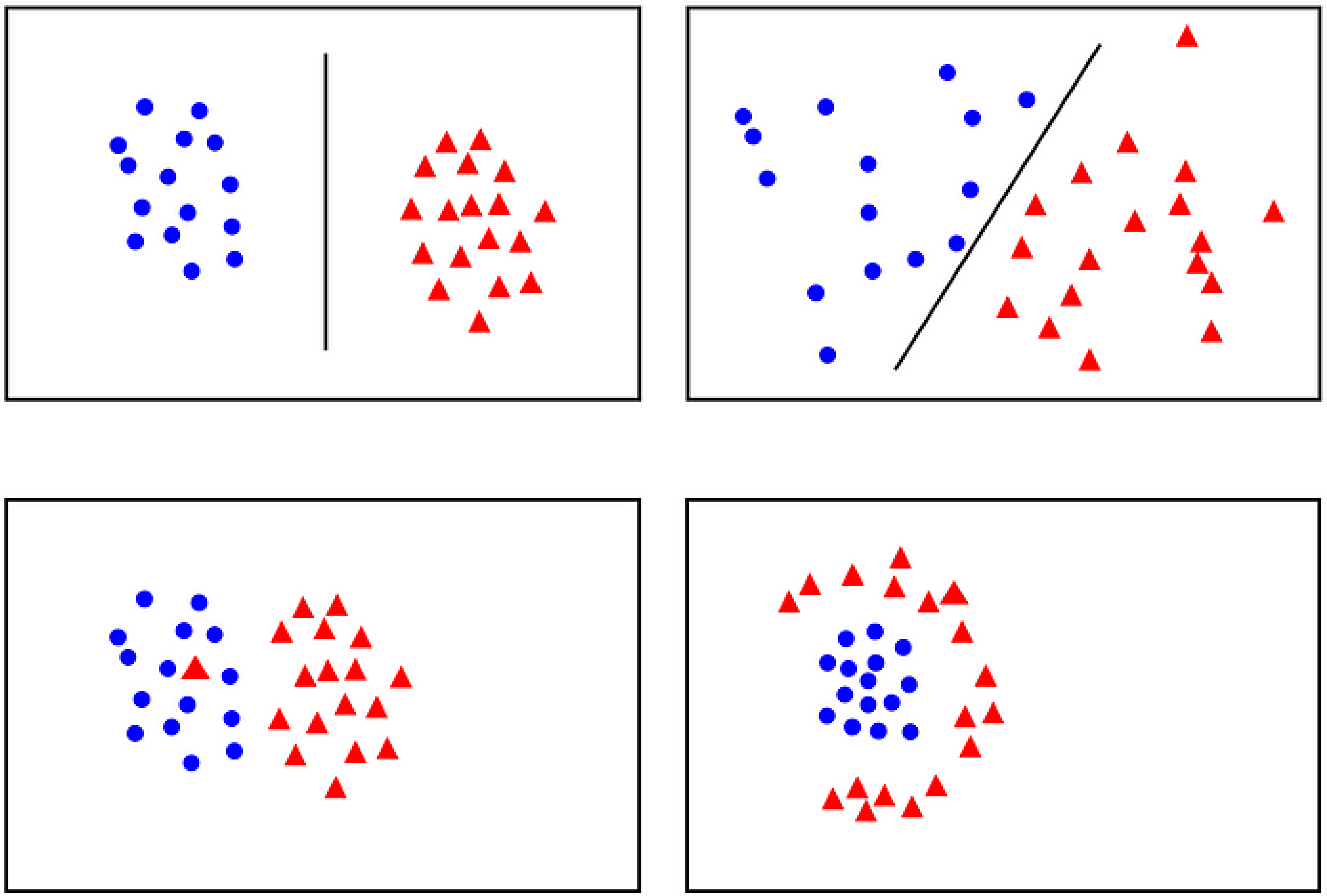
Support Vector Machines



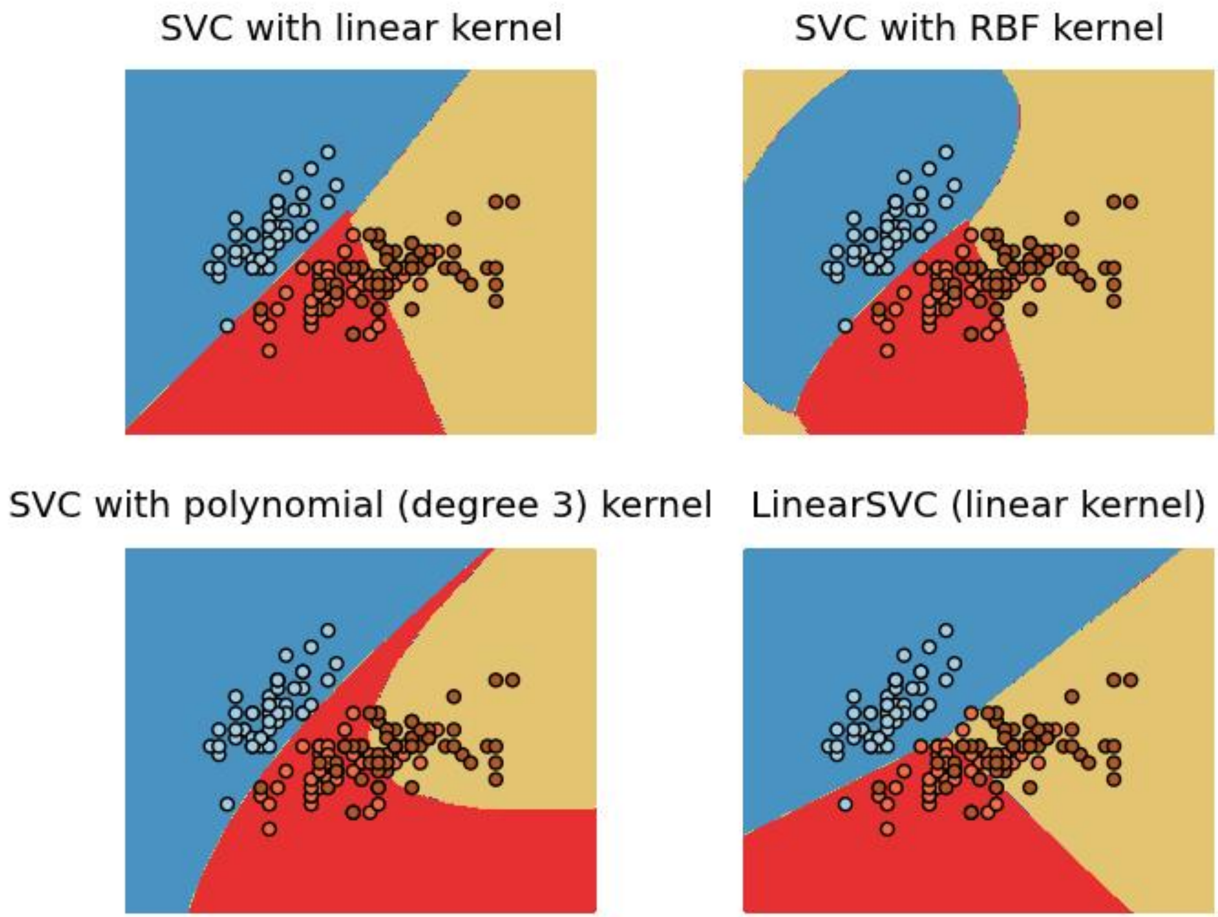
Support Vector Machines

linearly
separable

not
linearly
separable



Support Vector Machines



Support Vector Machines

- Python models
  - Predictons - Support Vector Regression (SVR)
  - Classifications - Support Vector Classification

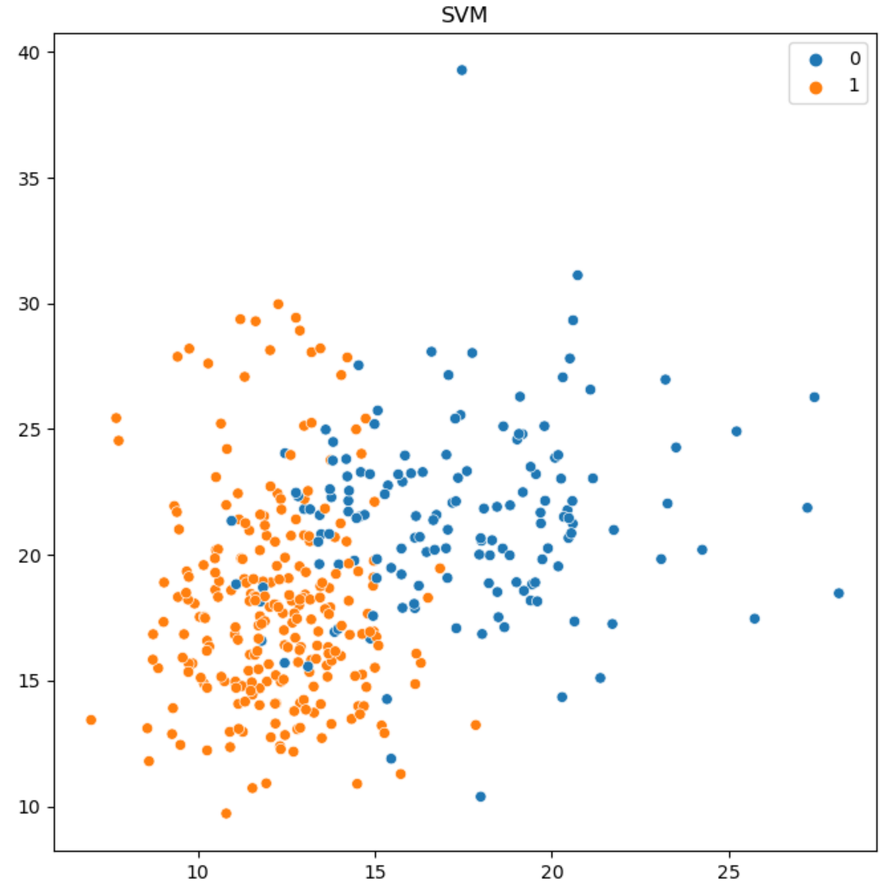
Module-sklearn.svm

Support Vector Machines - Example

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import svm
#Load dataset
cancer = datasets.load_breast_cancer()

# print the names of the 13 features (X)
print("Features: ", cancer.feature_names)
# print the label type of cancer('malignant' 'benign') (Y)
print("Labels: ", cancer.target_names)
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target,
                                                    test_size=0.3,random_state=109) # 70% training and 30% test
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel
#Train the model using the training sets
clf.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
# Model Accuracy: how often is the classifier correct?
print("----------Classification report-----------")
print(classification_report(y_test, y_pred))
```

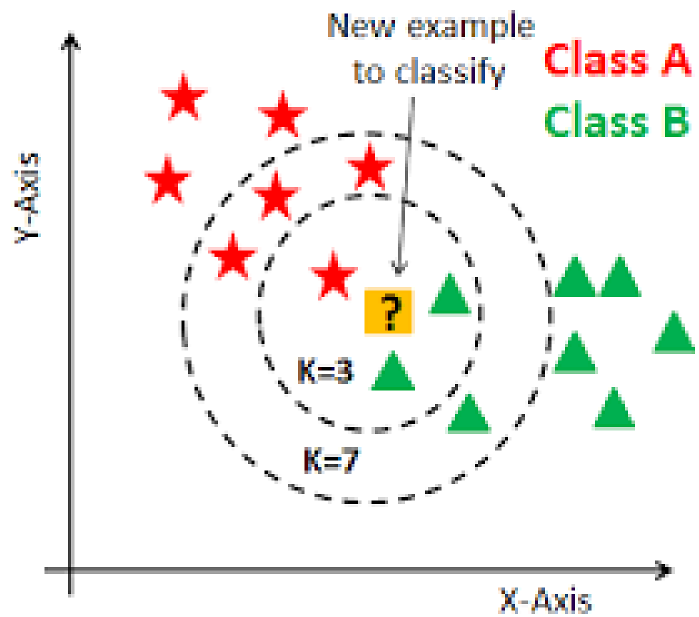
Support Vector Machines - Plot Example

```
plt.figure(figsize=(8, 8))
sns.scatterplot(X_train[:, 0],
X_train[:, 1], hue=y_train)
plt.title("SVM")
plt.show()
```

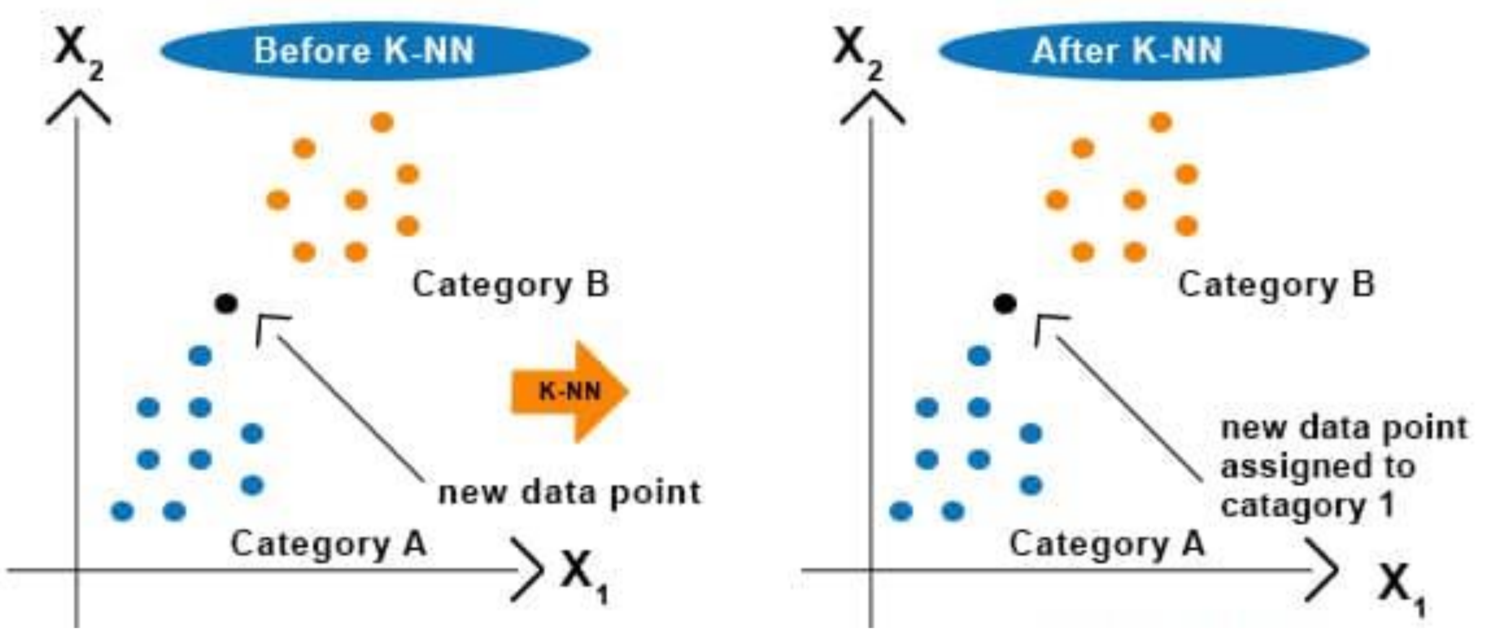


K-Nearest Neighbours

- Calculate the number of nearest neighbors.
- Calculate the distance of testing observations with all training data using Euclidean distance.

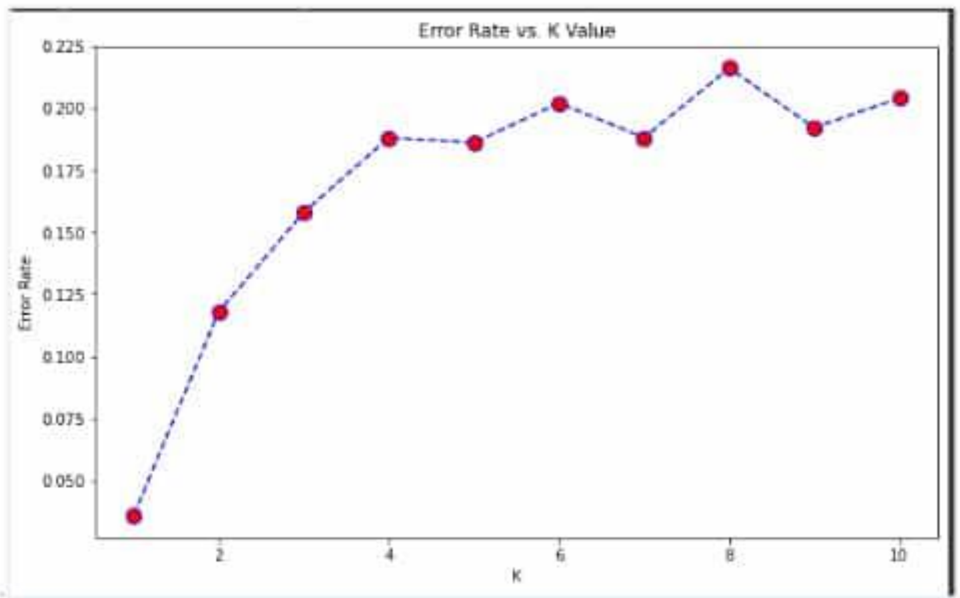


K-Nearest Neighbours



K-Nearest Neighbours

- How to choose the number of neighbours?
  - K value is initialized randomly & starts computing.
  - If you choose a small value of K, the decision boundaries will be unstable.
  - Derive a plot between error rate & K denoting values in a defined range.
  - Then choose the value for K which has less error rate.



K-Nearest Neighbours - Apps

- Recommending systems: Recommending ads for youtube and social media users, recommending products on any E-commerce websites.
- KNN is used in politics whether the voter will vote or will not vote candidate.
- Other applications of KNN include video recognition, image recognition, and handwriting detection.
- Python
  - KNeighboursClassifier
  - KNeigboursRegressor

Module-sklearn.neighbors

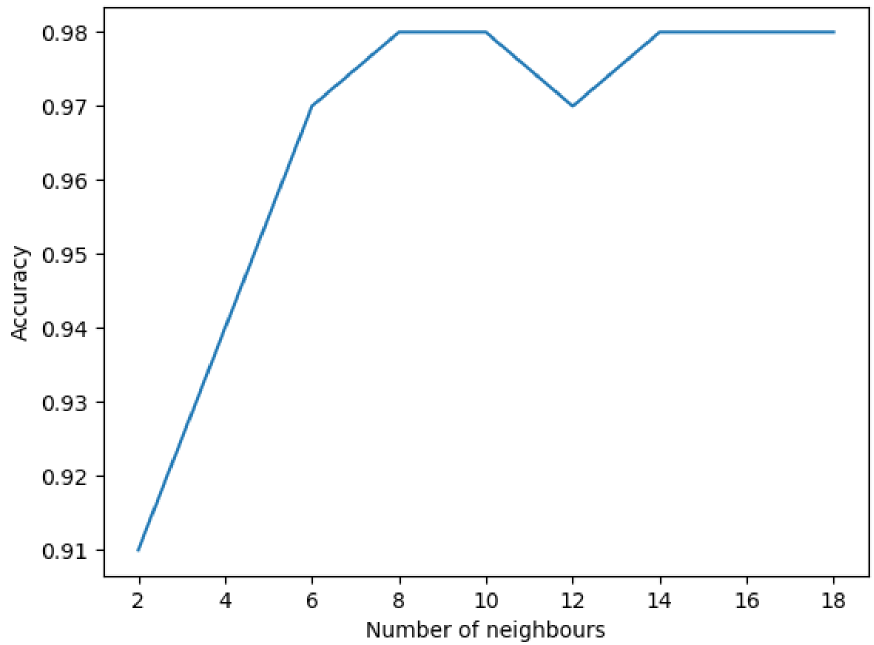
K-Nearest Neighbours - Example

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.neighbors import KNeighborsClassifier

#Load dataset
cancer = datasets.load_breast_cancer()

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target,
                                                    test_size=0.3,random_state=109) # 70% training and 30% test
#Create the Classifier
clf = KNeighborsClassifier(n_neighbors=8)
#Train the model using the training sets
clf.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)

print("----------Classification report-----------")
print(classification_report(y_test, y_pred))
```


DEPARTAMENTO CIÊNCIA E TECNOLOGIA

K-Nearest Neighbours - Example

```
import matplotlib.pyplot as plt

xvalues= [2, 4, 6, 8, 10, 12, 14, 16, 18] #number of neighbours
values = [0.91, 0.94, 0.97, 0.98, 0.98, 0.97, 0.98, 0.98, 0.98] #accuracy
plt.plot(xvalues, values)
plt.xlabel('Number of neighbours')
plt.ylabel('Accuracy')
plt.show()
```



Exercises

- Using the datasets travel insurance, UniversalBank and TaxInfo
- Do an exploratory analysis of the data set
- Divide the dataset in training and test
- Apply the naive bayes, SVM and k-NN classification models
- Obtain and plot the confusion matrix
- Obtain the classification report and analyse the best number of neighbours
- Start to employ the models which you have learnt to your project dataset

References

- Bonaccorso, G. (2020). Mastering Machine Learning Algorithms: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work. Packt Publishing Ltd.
- Bonaccorso, G. (2018). Machine Learning Algorithms: Popular algorithms for data science and machine learning. Packt Publishing Ltd.
- Lee, W. M. (2019). Python machine learning. John Wiley & Sons.
- Hauck, T. (2014). scikit-learn Cookbook. Packt Publishing Ltd.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). Introduction to linear regression analysis. John Wiley & Sons.
- Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

UNIVERSIDADE PORTUCALENSE

Do conhecimento à prática.