Soft computing

K.N. Toosi University of Technology

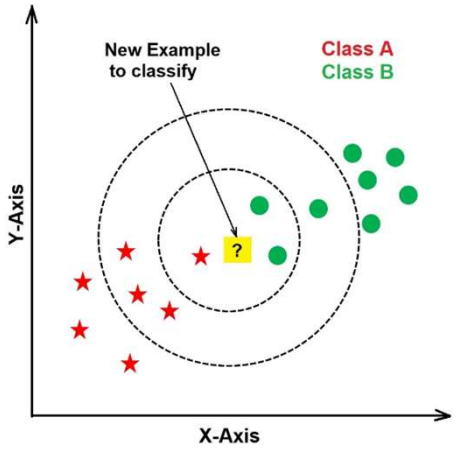
Soft Computing 1

k-NN (k-Nearest Neighbors)

الگوریتم
همسایگان نزدیک

Soft Computing 2

k-Nearest Neighbors (k-NN)

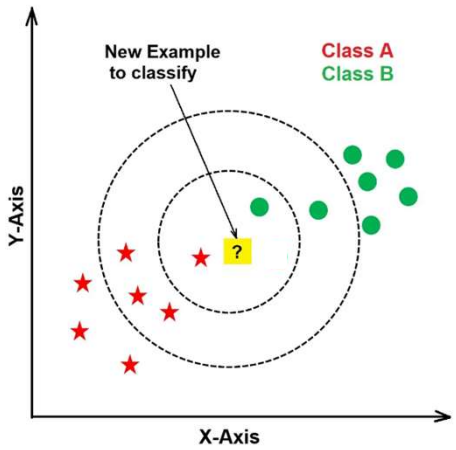


الگوریتم طبقه بندی که به هر داده کلاسی مشابه همسایگان نزدیک می دهد.

k تعداد همسایگان نزدیک

کلاس نقطه زرد رنگ
K=3 کلاس B
K=7 کلاس A

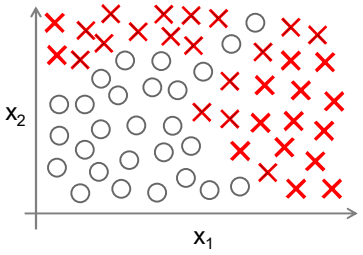
k-Nearest Neighbors (k-NN)



اگر تعداد همسایگان نزدیک در یک بازه برای دوکلاس برابر بود از چه روشی می توان دسته بندی کرد؟

در اینصورت می توان از تابع وزن استفاده کرد. در مثال مقابل تابع وزن فاصله سبب می شود نقطه زرد رنگ کلاس ستاره به خود بگیرد.

Applications of k-NN



- Recommendation Systems
- Image Classification
- Disease Prediction
- k-NN is versatile and applicable in many real-world scenarios.

- Advantages:
  - Simple to implement
  - Effective in various cases
- Disadvantages:
  - High memory requirement
  - Sensitive to scale of features

Steps of the k-NN Algorithm

- 1. Choose the value of k.
- 2. Calculate the distance between the new data point and all other points.
- 3. Find the k nearest points.
- 4. Classify based on the majority of nearest neighbors.

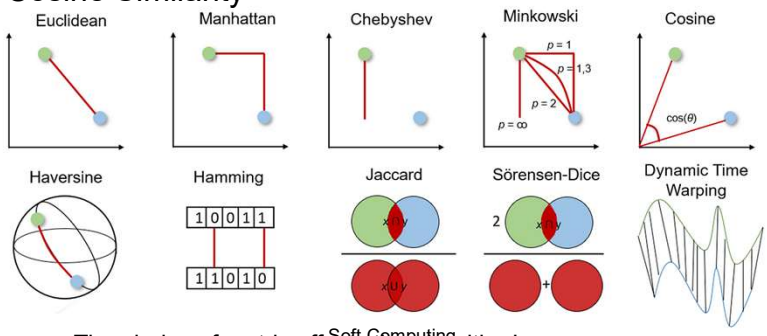
k اگر کوچک باشد بیش برازش شده وتاثیر نوفه زیاد است

k اگر بزرگ باشد کم برازش شده وتاثیر اطلاعات حساس از بین می رود

k افزایش یابد، میزان واریانس کاهش یافته ولی بایاس افزایش خواهد یافت.

Distance Metrics in k-NN

- Euclidean Distance (most common)
- Manhattan Distance
- Chebyshev
- Minkowski
- Cosine Similarity



The choice of metric affects the algorithm's accuracy

تعیین فواصل

۱. فاصله اقلیدسی (Euclidean Distance)

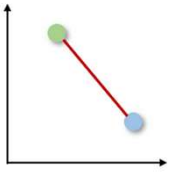
$$d(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

فاصله اقلیدسی در دو بعد بین دو نقطه A و B

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

فاصله اقلیدسی در n بعد بین دو نقطه A و B

Euclidean



این روش به نقاط دور حساس است، به این معنی که داده‌های پرت می‌توانند تأثیر بیشتری داشته باشند.

تعیین فواصل

۲. فاصله منهتن (Manhattan Distance)

فاصله منهتن، که به عنوان فاصله شهری نیز شناخته می‌شود، مجموع قدر مطلق تفاوت مختصات بین دو نقطه است.

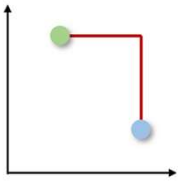
فاصله منهتن، در فضای دوبعدیبین دو نقطه A و B

$$d(A,B) = |x_2 - x_1| + |y_2 - y_1|$$

فاصله منهتن در n بعد بین دو نقطه A و B

$$d(A,B) = \sum_{i=1}^{n} |x_i - y_i|$$

Manhattan



Soft Computing 9

تعیین فواصل

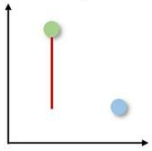
۳. فاصله چبیشف (Chebyshev Distance)

فاصله چبیشف یا L_∞ حداکثر فاصله‌ای است که در محورهای مختصات مختلف بین دو نقطه وجود دارد

$$d(A,B) = \max(|x_i - y_i|)$$

این فاصله در شرایطی مفید است که بخواهیم تنها بزرگ‌ترین تفاوت بین مختصات را در نظر بگیریم.

Chebyshev



Soft Computing 10

تعیین فواصل

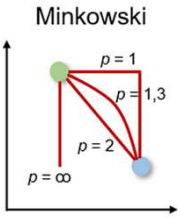
۴. فاصله مینکوفسکی (Minkowski Distance)

فاصله مینکوفسکی یک حالت تعمیم‌یافته از فاصله اقلیدسی و منهتن است.

فاصله مینکوفسکی در n بعد بین دو نقطه A و B

$$d(A,B) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

در این فرمول، p یک پارامتر قابل تنظیم است:



p=1: فاصله مینکوفسکی به فاصله منهتن تبدیل می‌شود.
p=2: فاصله مینکوفسکی به فاصله اقلیدسی تبدیل می‌شود.
p= ∞ : فاصله مینکوفسکی به فاصله چبیشف تبدیل می‌شود.

برای مقادیر بالاتر p، فاصله به نقاطی که دورتر از هم هستند حساس‌تر می‌شود.

تعیین فواصل

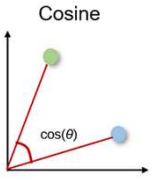
۵. فاصله کسینوسی (Cosine Similarity)

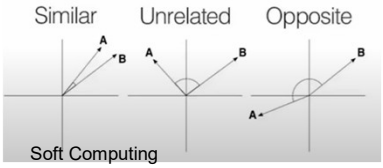
فاصله کسینوسی در واقع یک معیار برای محاسبه شباهت بین دو بردار است، که در آن از زاویه بین دو بردار استفاده می‌شود. این معیار برای داده‌های با ابعاد بالا و در شرایطی که طول بردارها اهمیت ندارد، کاربرد دارد.

فاصله کسینوسی در n بعد بین دو نقطه A و B

$$\frac{A \cdot B}{||A|| \times ||B||} = \cos(\theta) = \text{Similarity}(A,B)$$

هرچقدر زاویه بین دو بردار کمتر باشد، شباهت کسینوسی به ۱ نزدیک‌تر خواهد بود. این روش به جای فاصله، شباهت را محاسبه می‌کند و برای داده‌های متنی و داده‌های با ابعاد بالا مناسب است.





Implementing k-NN in Python

Example code using scikit-learn:

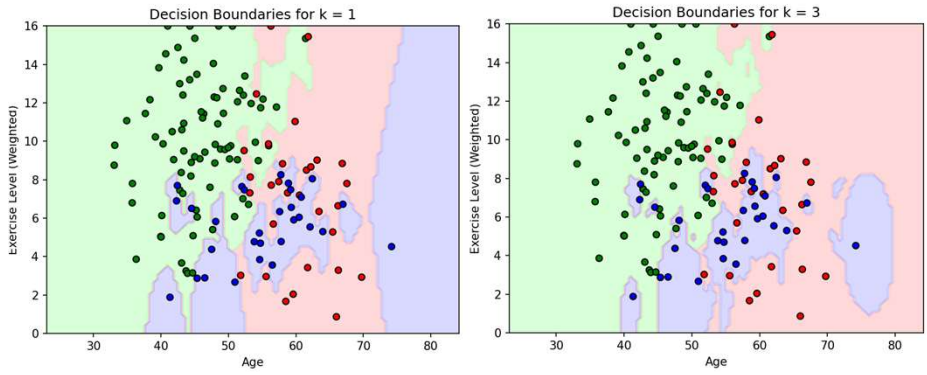
- from sklearn.neighbors import KNeighborsClassifier
- knn = KNeighborsClassifier(n_neighbors=3)
- knn.fit(X_train, y_train)
- y_pred = knn.predict(X_test)

```
# Train k-NN model
knn = KNeighborsClassifier(n_neighbors=k, weights='distance')
knn.fit(X, y)
```

Soft Computing 13

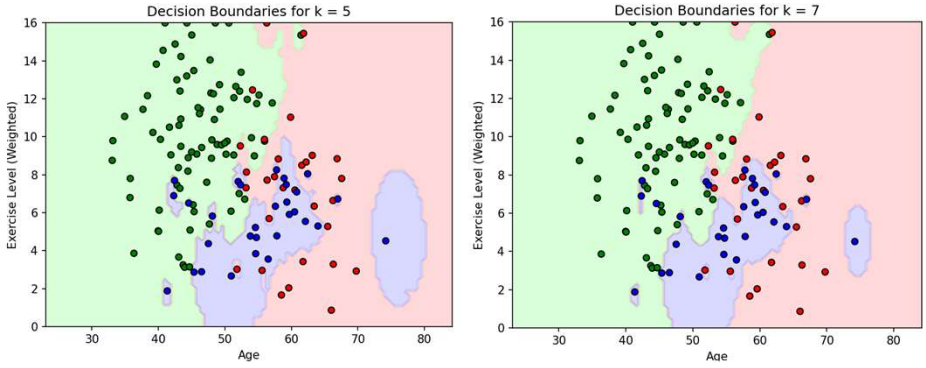
نمونه طبقه بندی به روش k-NN

فایل: 9 kNN.py



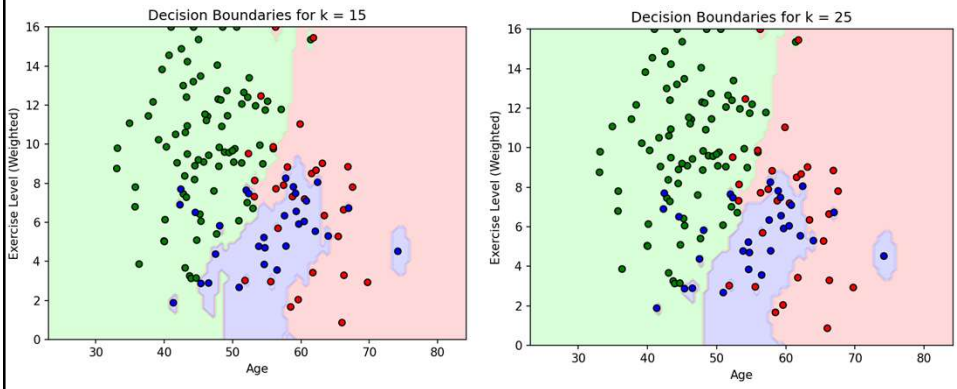
Soft Computing 14

نمونه طبقه بندی به روش k-NN



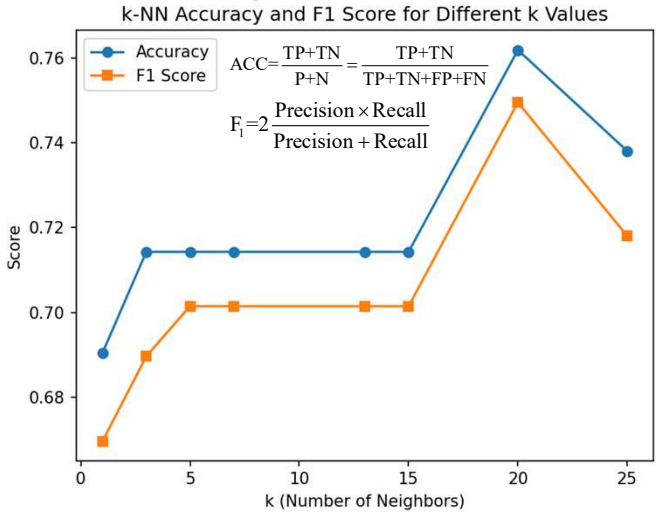
Soft Computing 15

نمونه طبقه بندی به روش k-NN



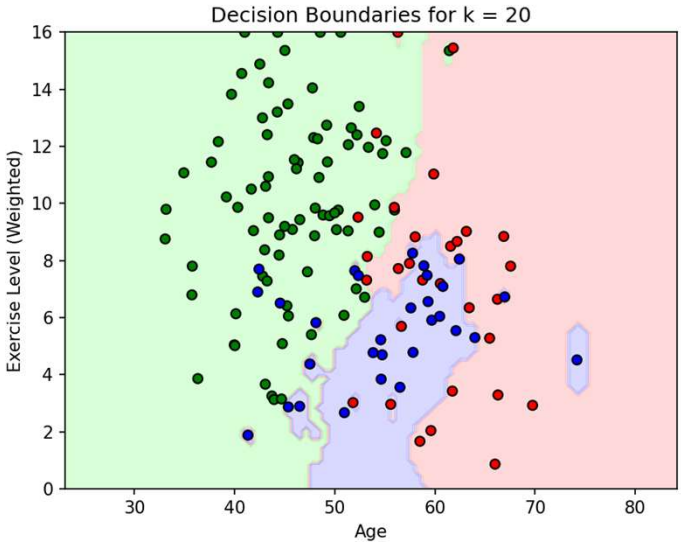
Soft Computing 16

نمونه طبقه بندی به روش k-NN



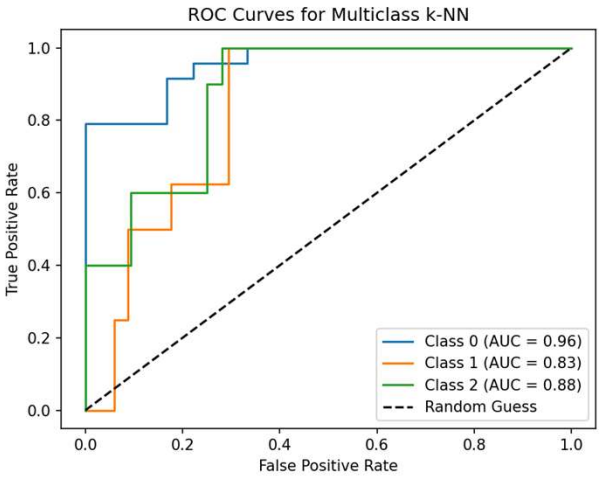
Soft Computing 17

نمونه طبقه بندی به روش k-NN



18

نمونه طبقه بندی به روش k-NN



Receiver Operating Characteristic for k=20

Soft Computing 19

تمرین برنامه نویسی

تمرین نهم : یک برنامه به زبان پایتون بنویسید که یک فایل داده را خوانده و به روش همسایگان نزدیک بر حسب ویژگیها داده ها را طبقه بندی نماید.

۱- فایل داده را بخوانید

۲- داده ها را طبقه بندی نمایید

۳- سه نقطه با ویژگی جدید را تعیین کنید در کدام کلاس داده هستند.

Soft Computing 20