

Artificial Intelligence and Soft computing



K.N. TOOSI
University of
Technology

هوش مصنوعی و تحول دیجیتال



Hasan Ghasemzadeh
<http://wp.kntu.ac.ir/ghasemzadeh>

Artificial Intelligence

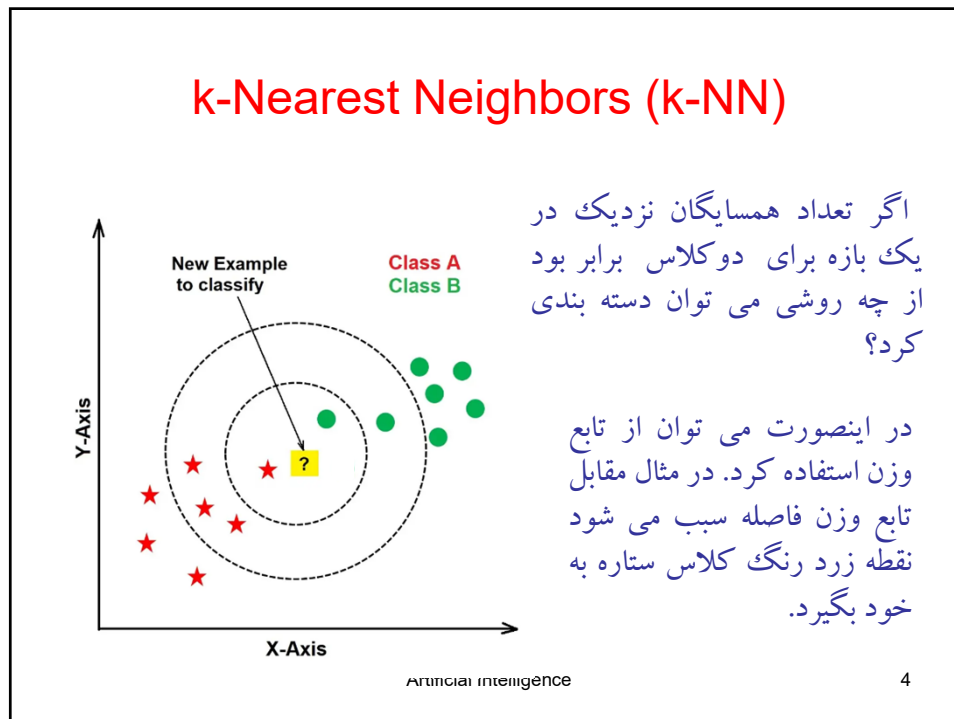
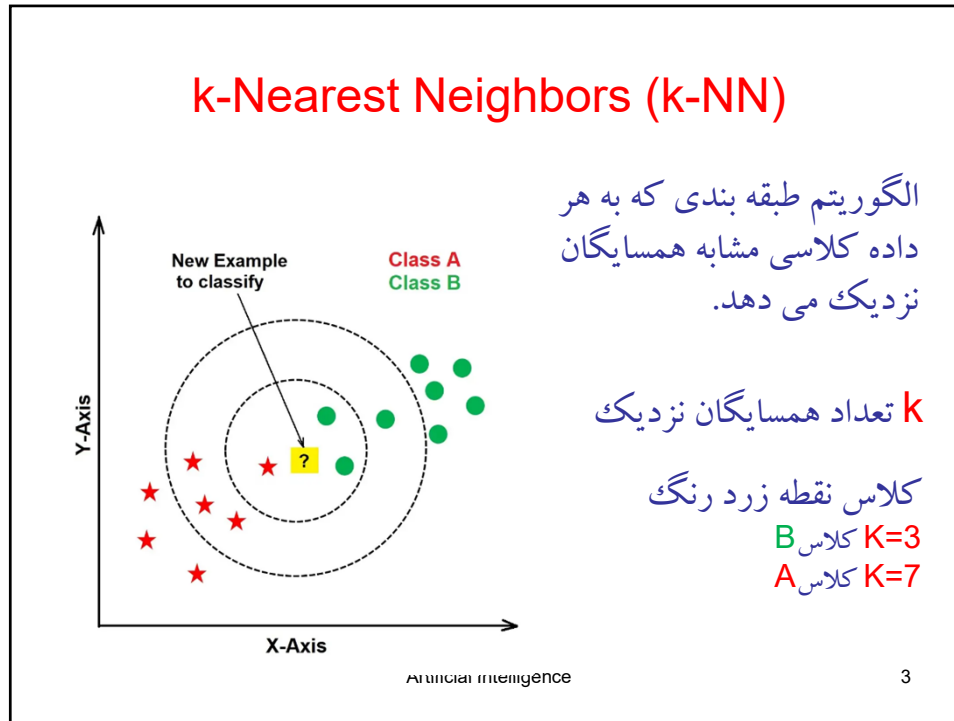
k-NN (k-Nearest Neighbors)



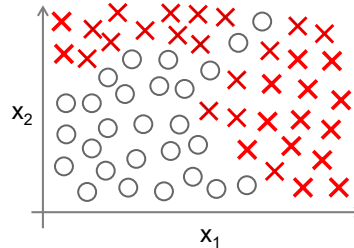
الگوریتم
همسایگان نزدیک

Artificial Intelligence

2



Applications of k-NN



- Recommendation Systems
 - Image Classification
 - Disease Prediction
 - k-NN is versatile and applicable in many real-world scenarios.
- | | |
|---|--|
| <ul style="list-style-type: none"> • Advantages: - Simple to implement - Effective in various cases | <ul style="list-style-type: none"> • 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 برآزش شده و تاثیر نوبه زیاد است

k اگر بزرگ باشد کم برآزش شده و تاثیر اطلاعات حساس از بین می رود

Distance Metrics in k-NN

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

Euclidean

Manhattan

Chebyshev

Minkowski

Cosine

Haversine

Hamming

Jaccard

Sørensen-Dice

Dynamic Time Warping

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.

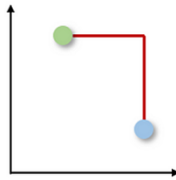
فاصله منهتن، در فضای دوبعدی بین دو نقطه A و B

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

فاصله منهتن در n بعد بین دو نقطه A و B

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

Manhattan



تعیین فواصل

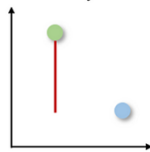
۳. فاصله چبیشف (Chebyshev Distance)

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



تعیین فواصل

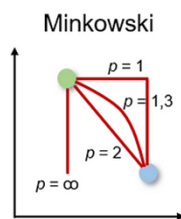
۴. فاصله مینکوفسکی (Minkowski Distance)

فاصله مینکوفسکی یک حالت تعمیم یافته از فاصله اقلیدسی و منهن است.

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

فاصله مینکوفسکی در n بعد بین دو نقطه A و B

در این فرمول، p یک پارامتر قابل تنظیم است:



- $p=1$: فاصله مینکوفسکی به فاصله منهن تبدیل می شود.
- $p=2$: فاصله مینکوفسکی به فاصله اقلیدسی تبدیل می شود.
- $p=\infty$: فاصله مینکوفسکی به فاصله چبیشف تبدیل می شود.

برای مقادیر بالاتر p ، فاصله به نقاطی که دورتر از هم هستند حساس تر می شود.

تعیین فواصل

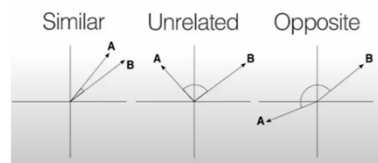
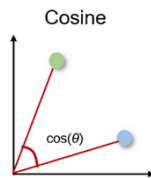
۵. فاصله کسینوسی (Cosine Similarity)

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

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

فاصله کسینوسی در n بعد بین دو نقطه A و B

هرچه قدر زاویه بین دو بردار کمتر باشد، شباهت کسینوسی به ۱ نزدیک تر خواهد بود. این روش به جای فاصله، شباهت را محاسبه می کند و برای داده های متنی و داده های با ابعاد بالا مناسب است.



Implementing k-NN in Python

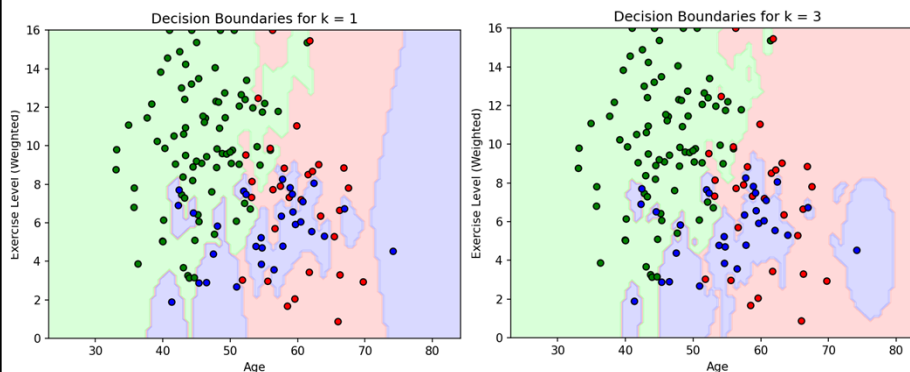
Example code using scikit-learn:

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

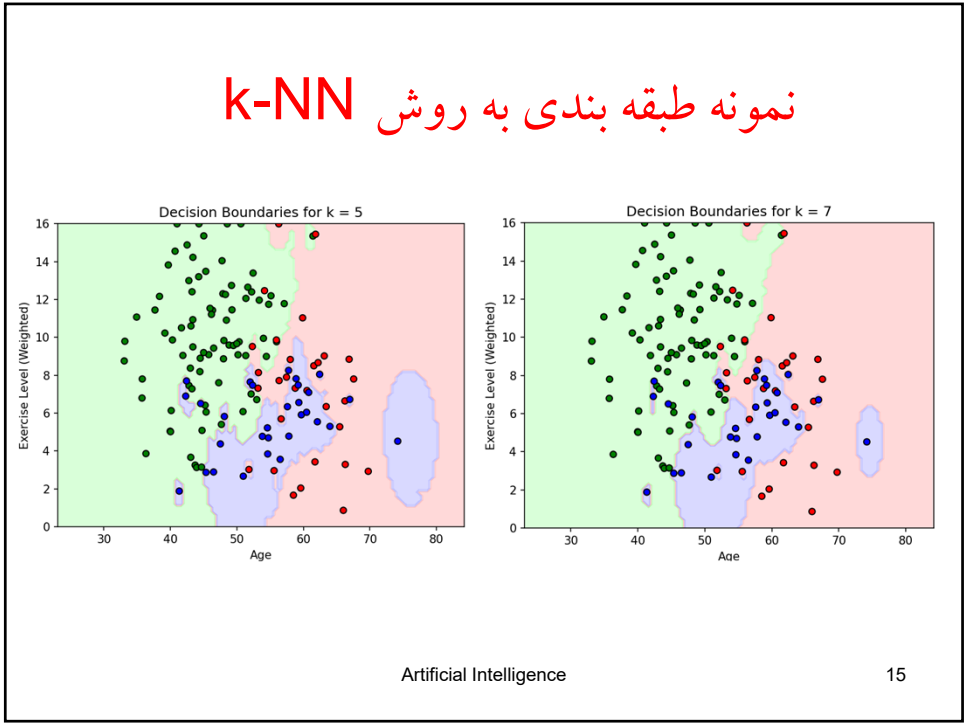
```
# Train k-NN model
knn = KNeighborsClassifier(n_neighbors=k, weights='distance')
knn.fit(X, y)
```

نمونه طبقه بندی به روش k-NN

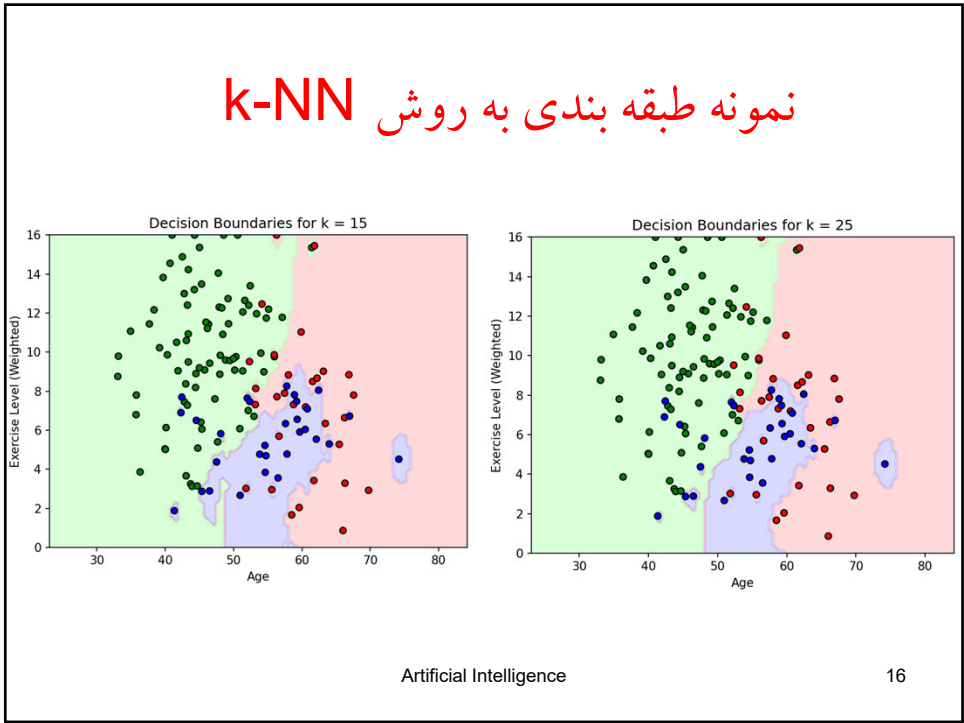
فایل: 9 kNN.py

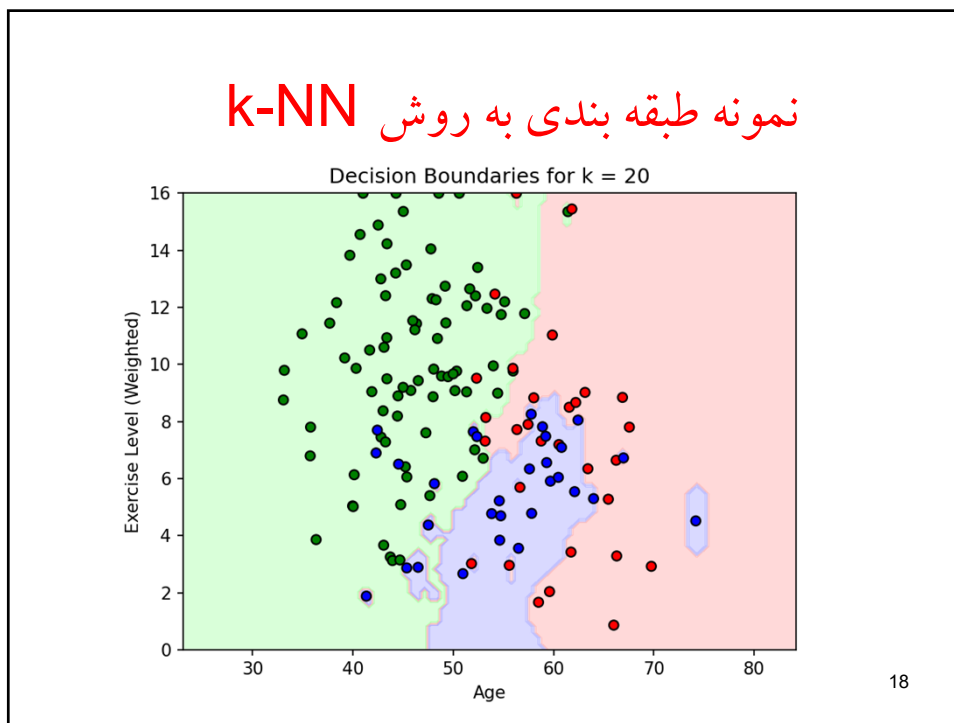
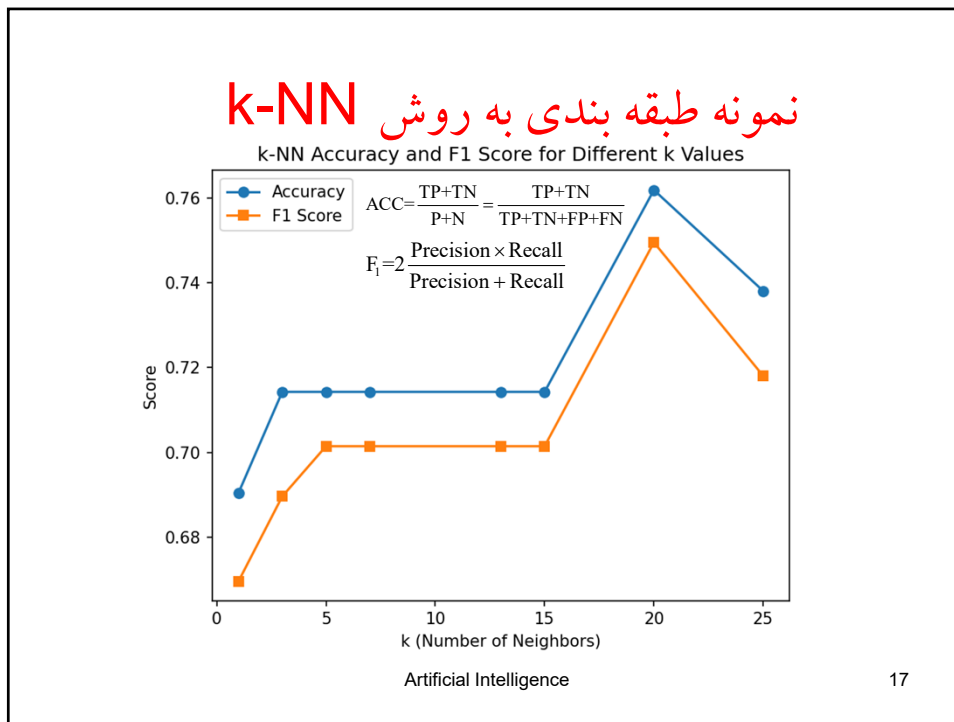


نمونه طبقه بندی به روش k-NN

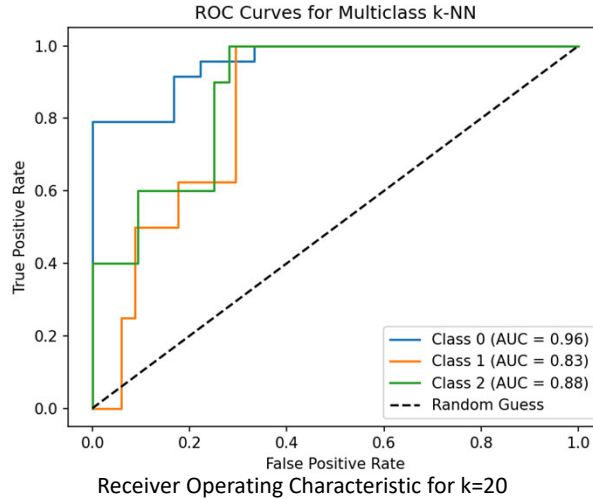


نمونه طبقه بندی به روش k-NN





نمونه طبقه بندی به روش k-NN



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.

Artificial Intelligence

20