

به نام خدا

درس مباحث ویژه

ترم بهمن 1403

علیرضا حسن زاده

علی اکبر مکاریان

مدرس: محمد احمدزاده

---

## گزارش تمرین خوشه‌بندی با الگوریتم‌های K-Means و KNN

---

### موضوع تمرین:

در این تمرین، یک مجموعه داده فرضی از مشخصات سفینه‌های فضایی طراحی شد تا الگوریتم‌های خوشه‌بندی K-Means و KNN روی آن پیاده‌سازی شوند.

### مرحله 1: تولید دیتاست مصنوعی

در این مرحله با استفاده از کتابخانه‌های pandas و numpy، یک دیتاست شامل ۱۰۰ نمونه و ۲۰ ویژگی ساخته شد. موضوع دیتاست مربوط به سفینه‌های فضایی در یک پایگاه فرضی بود.

```
import pandas as pd
```

```
import numpy as np
```

```
import random
```

```
(42)np.random.seed
```

```
(42)random.seed
```

```
num_samples = 100
```

```

} = features
,Weight_ton": np.random.uniform(10, 200, num_samples)"
,Crew_Count": np.random.randint(1, 10, num_samples)"
Fuel_Type": np.random.choice(["Hydrogen", "Plasma", "Ion"], "
,num_samples)
,Prep_Time_days": np.random.randint(30, 500, num_samples)"
,Max_Speed_kmph": np.random.uniform(10000, 50000, num_samples)"
,Length_m": np.random.uniform(20, 100, num_samples)"
,Width_m": np.random.uniform(10, 50, num_samples)"
,Height_m": np.random.uniform(8, 40, num_samples)"
,Armor_Level": np.random.randint(1, 10, num_samples)"
,Auto_Launch": np.random.choice(["Yes", "No"], num_samples)"
Country": np.random.choice(["USA", "Russia", "China", "India", "
,"Germany"], num_samples)
,Energy_Consumption": np.random.uniform(50, 500, num_samples)"
}" for _ in (1000,9999)Model_Number": [f"X{random.randint"
,range(num_samples)]
,Temp_Tolerance": np.random.uniform(-150, 500, num_samples)"
Mission_Type": np.random.choice(["Combat", "Research", "
,"Transport"], num_samples)
,Oxygen_Status": np.random.randint(70, 100, num_samples)"
,Successful_Flights": np.random.randint(0, 20, num_samples)"
,Planet_Landing_Cap": np.random.randint(0, 5, num_samples)"
,Noise_Level": np.random.uniform(20, 90, num_samples)"
,Repairability_Score": np.random.randint(1, 10, num_samples)"

```

```
{
```

```
df = pd.DataFrame(features)
```

---

مرحله 2: ذخیره در فرمت‌های Excel و CSV

```
df.to_csv("spaceships_dataset.csv", index=False)
```

```
df.to_excel("spaceships_dataset.xlsx", index=False)
```

---

مرحله 3: پیش‌پردازش داده‌ها (تبدیل ویژگی‌های متنی به عددی)

```
from sklearn.preprocessing import LabelEncoder
```

```
()df_numeric = df.copy
```

```
{ } = label_encoders
```

```
:for column in df_numeric.select_dtypes(include="object").columns
```

```
()le = LabelEncoder
```

```
df_numeric[column] = le.fit_transform(df_numeric[column])
```

```
label_encoders[column] = le
```

---

مرحله 4: خوشه‌بندی با K-Means

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters=3, random_state=42)
```

```
df_numeric["KMeans_Cluster"] = kmeans.fit_predict(df_numeric)
```

---

مرحله 5: پیش‌بینی خوشه با الگوریتم KNN

```
from sklearn.neighbors import KNeighborsClassifier
```

```
X = df_numeric.drop("KMeans_Cluster", axis=1)
```

```
y = df_numeric["KMeans_Cluster"]
```

```
# تقسیم داده‌ها به دو نیمه برای آموزش و تست
```

```
[50:]X_train, X_test = X[:50], X
```

```
[50:]y_train, y_test = y[:50], y
```

```
knn = KNeighborsClassifier(n_neighbors=3)
```

```
knn.fit(X_train, y_train)
```

```
# پیش‌بینی خوشه برای داده‌های تست
```

```
df_numeric.loc[50:, "KNN_Predicted"] = knn.predict(X_test)
```

---

## مرحله 6: رسم نمودار خروجی با matplotlib

در اینجا برای نمایش بهتر داده‌ها، از ویژگی‌های وزن و سرعت استفاده شده:

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(10, 6))
```

```
plt.scatter(df_numeric["Max_Speed_kmph"], df_numeric["Weight_ton"],  
c=df_numeric["KMeans_Cluster"], cmap="viridis")
```

```
plt.title("K-Means Clustering of Spaceships")
```

```
plt.xlabel("Max Speed (km/h)")
```

```
plt.ylabel("Weight (ton)")
```

```
plt.savefig("clustering_plot.png")
```

```
plt.show()
```

