به نام خدا

گزارش پروژه پایانی داده کاوی

حانیه حسین نژاد - ۴۰۰۱۲۳۱۰۲۹

https://github.com/hany-1485/Datamining

این پروژه شامل پیش‌پردازش داده‌ها، تحلیل داده‌ها، ساخت مدل‌های یادگیری ماشین، و اجرای الگوریتم‌های خوشه‌بندی است که در ادامه، هر بخش را بررسی میکنیم :

---

1. بارگذاری داده‌ها

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,
ConfusionMatrixDisplay,accuracy_score ,precision_score, recall_score, f1_score,
classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
from sklearn.decomposition import PCA
import seaborn as sns
from sklearn.cluster import AgglomerativeClustering

dataset=pd.read_csv("/modified_diabetes_prediction_dataset.csv")
dataset.head()
```

- کتابخانه‌های موردنیاز برای تحلیل داده، مدل‌سازی، و خوشه‌بندی وارد می‌شوند.

- داده‌ها از یک فایل CSV به نام modified_diabetes_prediction_dataset.csv بارگذاری می‌شوند و اطلاعات اولیه‌ی آن بررسی می‌شود.

2.بررسی و پاک‌سازی داده‌ها

بررسی شکل و ستون‌ها:

```
dataset.shape
columns=dataset.columns
print(columns)
```

- تعداد ردیف‌ها و ستون‌ها و نام ستون‌ها بررسی می‌شود.

حذف مقادیر NaN و داده‌های تکراری:

```
dataset = dataset.dropna(subset=['diabetes'])
dataset = dataset.drop_duplicates()
dataset.shape
```

- داده‌هایی که ستون diabetes آن‌ها مقدار ندارد حذف می‌شوند.
- داده‌های تکراری نیز حذف می‌شوند.

بررسی و اصلاح مقادیر ستون‌ها:

```
dataset['smoking_history'].unique()
dataset['gender'].unique()
dataset['gender'] = dataset['gender'].map({'Male':0 ,'Female':
1,'Other':1,'unknown':1})
dataset['smoking_history'] =
dataset['smoking_history'].map({'never':0,'former':1,'not current':2, 'No Info':3,
'current':4 ,'yes':4, 'ever':5})
```

- مقادیر موجود در ستون‌های smoking_history و gender بررسی می‌شوند.
- مقادیر این ستون‌ها به مقادیر عددی تبدیل می‌شوند.

- مقادیر current و yes به یک معنا هستند بنابراین هردو به مقدار یکسان مپ میشوند.

پر کردن مقادیر گمشده:

```
dataset= dataset.fillna(value={
    'gender': dataset['gender'].mode(),
    'smoking_history': dataset['smoking_history'].mode(),
    'hypertension': dataset['hypertension'].mode(),
    'age': dataset['age'].mean(),
    'bmi': dataset['bmi'].mean(),
    'blood_glucose_level': dataset['blood_glucose_level'].mean(),
    'HbA1c_level': dataset['HbA1c_level'].mean(),
    'heart_disease': dataset['heart_disease'].mode()
})
```

- مقادیر گمشده برای ستون‌های مختلف بر اساس میانگین یا مد (بیشترین تکرار) پر می‌شوند.

3.مدیریت داده‌های پرت

بررسی و مدیریت مقادیر غیرمنطقی:

```
col_num = ['age', 'bmi','blood_glucose_level','HbA1c_level']
for col in col_num:
    mean_value = dataset[dataset[col] > 0][col].mean()
    dataset[col] = dataset[col].apply(lambda x: mean_value if x <= 0 else x)
```

- ستون‌های عددی بررسی می‌شوند و مقادیر نامعتبر (مثل مقادیر کمتر یا مساوی صفر) با میانگین داده‌های معتبر جایگزین می‌شوند.

شناسایی و حذف داده‌های پرت:

ابتدا نمودار جعبه ای مربوط به هرستون را رسم میکنیم و مشاهده میکنیم که در برخی از ستون ها داده پرت موجود است

```
dt = dataset.copy()
num=dt.select_dtypes(include=['number']).columns
num = num.drop({'diabetes','hypertension', 'heart_disease','smoking_history'})
mask = pd.Series(True, index=dt.index)
for i in num:
    max_limit = dt[i].mean() + 3*dt[i].std()
    min_limit = dt[i].mean() - 3*dt[i].std()
    mask &= (dt[i] < max_limit) & (dt[i] > min_limit)
new_dataset = dt[mask]
```

- برای ستون‌های عددی، مقادیر پرت شناسایی و حذف می‌شوند.

حذف داده‌های غیرمنطقی:

```
index_to_drop = new_dataset[((new_dataset['smoking_history'] > 5) |
(new_dataset['hypertension'] > 1) | (new_dataset['heart_disease'] > 1) |
(new_dataset['diabetes'] > 1))].index
new_dataset = new_dataset.drop(index_to_drop)
```

- مقادیر غیرمعتبر در ستون‌هایی که مقدار آن‌ها باید در بازه خاصی باشد مانند smoking_history یا diabetes حذف می‌شوند.

4.مدل‌سازی یادگیری ماشین

داده‌های ورودی و خروجی:

```
x=np.array(new_dataset.drop('diabetes',axis=1))
y=np.array(new_dataset['diabetes'])
```

- ویژگی‌های مستقل (x) و برچسب‌ها (y) تعریف می‌شوند.

تقسیم داده‌ها:

```
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.3,
random_state=2)
```

- داده‌ها به دو بخش آموزشی (۷۰٪) و تستی (۳۰٪) تقسیم می‌شوند.

مدل رگرسیون لجستیک

```
model_logistic = LogisticRegression()
model_logistic.fit(X_train, Y_train)
```

- مدل رگرسیون لجستیک آموزش داده می‌شود.

```
accuracy = accuracy_score(Y_test, y_logistic_predict)
precision = precision_score(Y_test, y_logistic_predict, average='binary')
recall = recall_score(Y_test, y_logistic_predict, average='binary')
f1 = f1_score(Y_test, y_logistic_predict, average='binary')
```

- معیارهای ارزیابی شامل دقت، دقت مثبت، بازخوانی، و F1-Score محاسبه می‌شوند.

- Accuracy: 0.94
- Precision: 0.52
- Recall: 0.33
- F1-Score: 0.40

مدل درخت تصمیم

```
model_DecisionTree = DecisionTreeClassifier(random_state=42)
model_DecisionTree.fit(X_train, Y_train)
```

- مدل درخت تصمیم آموزش داده می‌شود و معیارهای ارزیابی مشابه محاسبه می‌شوند.

- Accuracy: 0.95
- Precision: 0.63
- Recall: 0.66
- F1-Score: 0.64

برای مقایسه این دو مدل، یعنی رگرسیون لجستیک (Logistic Regression)و درخت تصمیم (Decision Tree)در دسته‌بندی داده‌ها، ما می‌توانیم چندین معیار را بررسی کنیم. این معیارها شامل دقت

(Accuracy)، دقت مثبت (Precision)، بازخوانی (Recall)، امتیاز F1 (F1-Score) ، و ماتریس سردرگمی (Confusion Matrix)می‌باشند.

مدل اول: رگرسیون لجستیک(Logistic Regression)

1.ماتریس سردرگمی(Confusion Matrix)

ماتریس سردرگمی برای رگرسیون لجستیک به‌صورت زیر است:

[[26582 579]

[1304 639]]

این ماتریس نشان می‌دهد:

- True Negatives (TN):تعداد نمونه‌هایی که به درستی به کلاس ۰ (منفی) نسبت داده شده‌اند: ۲۶,۵۸۲
- False Positives (FP):تعداد نمونه‌هایی که به اشتباه به کلاس ۱ (مثبت) نسبت داده شده‌اند: ۵۷۹
- False Negatives (FN)تعداد نمونه‌هایی که به اشتباه به کلاس ۰ (منفی) نسبت داده شده‌اند: ۱,۳۰۴
- True Positives (TP):تعداد نمونه‌هایی که به درستی به کلاس ۱ (مثبت) نسبت داده شده‌اند: ۶۳۹

2.دقت(Accuracy)

- دقت نشان‌دهنده درصد نمونه‌های درست پیش‌بینی‌شده است.
- برای مدل رگرسیون لجستیک، دقت برابر با ۰.۹۴ است.
- به این معنا که ۹۴ درصد از پیش‌بینی‌ها درست بوده‌اند.

3.دقت مثبت(Precision)

- دقت مثبت به این معنی است که از تمامی نمونه‌هایی که مدل به‌عنوان مثبت پیش‌بینی کرده، چه درصدی درست بوده‌اند.
- دقت مثبت برای این مدل برابر با ۰.۵۲ است.
- این نشان می‌دهد که ۵۲ درصد از نمونه‌های پیش‌بینی‌شده به‌عنوان مثبت، درست بوده‌اند.

4.بازخوانی(Recall)

- بازخوانی نشان‌دهنده این است که از تمامی نمونه‌های مثبت واقعی، چه درصدی به درستی شناسایی شده‌اند.

- بازخوانی برای مدل رگرسیون لجستیک برابر با ۰.۳۳ است.
- به این معنا که فقط ۳۳ درصد از نمونه‌های مثبت واقعی شناسایی شده‌اند.

5.(F1-Score)

- امتیاز F1 میانگینی از دقت و بازخوانی است و به‌ویژه زمانی مفید است که داده‌ها نامتوازن باشند.
- امتیاز F1 برای رگرسیون لجستیک برابر با ۰.۴۰ است.

6.گزارش دسته‌بندی(Classification Report) :

در گزارش دسته‌بندی، شما به‌طور دقیق می‌توانید مقادیر دقت، بازخوانی، و F1-Score را برای هر کلاس مشاهده کنید. در اینجا گزارش برای کلاس‌های ۰ و ۱ به‌صورت زیر است:

	precision	recall	f1-score	support
Class 0	0.95	0.98	0.96	27161
Class 1	0.52	0.33	0.40	1943
accuracy			0.94	29104
macro avg	0.74	0.65	0.68	29104
weighted avg	0.92	0.94	0.93	29104

مدل دوم: درخت تصمیم(Decision Tree)

1.ماتریس سردرگمی(Confusion Matrix)

ماتریس سردرگمی برای درخت تصمیم به‌صورت زیر است:

[[26402 759]

[[1282 661]

این ماتریس نشان می‌دهد:

- True Negatives (TN):تعداد نمونه‌هایی که به درستی به کلاس ۰ (منفی) نسبت داده شده‌اند: ۲۶,۴۰۲
- False Positives (FP):تعداد نمونه‌هایی که به اشتباه به کلاس ۱ (مثبت) نسبت داده شده‌اند: ۷۵۹

- False Negatives (FN):تعداد نمونه‌هایی که به اشتباه به کلاس ۰ (منفی) نسبت داده شده‌اند: ۶۶۱
- True Positives (TP):تعداد نمونه‌هایی که به درستی به کلاس ۱ (مثبت) نسبت داده شده‌اند: ۱,۲۸۲

2.دقت:(Accuracy)

- دقت مدل درخت تصمیم برابر با ۰.۹۵ است.
- به این معنا که ۹۵ درصد از پیش‌بینی‌ها درست بوده‌اند.

3.دقت مثبت(Precision)

- دقت مثبت برای مدل درخت تصمیم برابر با ۰.۶۳ است.
- این نشان می‌دهد که ۶۳ درصد از نمونه‌های پیش‌بینی‌شده به‌عنوان مثبت، درست بوده‌اند.

4.بازخوانی:(Recall)

- بازخوانی برای مدل درخت تصمیم برابر با ۰.۶۶ است.
- به این معنا که ۶۶ درصد از نمونه‌های مثبت واقعی شناسایی شده‌اند.

(F1-Score).5

- امتیاز F1 برای مدل درخت تصمیم برابر با ۰.۶۴ است.

6.گزارش دسته‌بندی(Classification Report)

در گزارش دسته‌بندی برای مدل درخت تصمیم، شما می‌توانید مقادیر دقت، بازخوانی و F1-Score را مشاهده کنید. در اینجا گزارش برای کلاس‌های ۰ و ۱ به‌صورت زیر است:

	precision	recall	f1-score	support
not diabets	0.98	0.97	0.97	27161
diabets	0.63	0.66	0.64	1943
accuracy			0.95	29104
macro avg	0.81	0.81	0.80	29104
weighted avg	0.95	0.95	0.95	29104

مقایسه مدل‌ها:

۱. دقت:(Accuracy)

- درخت تصمیم دارای دقت بالاتری است (۰.۹۵ در مقایسه با ۰.۹۴ برای رگرسیون لجستیک).
- این نشان می‌دهد که درخت تصمیم پیش‌بینی‌های دقیق‌تری به‌طور کلی انجام داده است.

۲. دقت مثبت:(Precision)

- درخت تصمیم دارای دقت مثبت بهتری است (۰.۶۳ در مقایسه با ۰.۵۲ برای رگرسیون لجستیک).
- این به این معناست که درخت تصمیم نمونه‌های مثبت بیشتری را به درستی شناسایی کرده است.

۳. بازخوانی:(Recall)

- درخت تصمیم همچنین دارای بازخوانی بهتری است (۰.۶۶ در مقایسه با ۰.۳۳ برای رگرسیون لجستیک).
- این به این معناست که درخت تصمیم توانسته است نمونه‌های مثبت واقعی بیشتری را شناسایی کند.

۴. امتیاز:F1 (F1-Score)

- درخت تصمیم امتیاز F1 بالاتری دارد (۰.۶۴ در مقایسه با ۰.۴۰ برای رگرسیون لجستیک).
- این نشان‌دهنده تعادل بهتر در دقت و بازخوانی در مدل درخت تصمیم است.

نتیجه‌گیری:

- درخت تصمیم عملکرد بهتری نسبت به رگرسیون لجستیک در این مسئله دارد. دقت، دقت مثبت، بازخوانی و امتیاز F1 درخت تصمیم به وضوح بهتر است.
- رگرسیون لجستیک دقت بالاتری دارد، اما توانایی کمتری در شناسایی نمونه‌های مثبت (دیابت) دارد.
- درخت تصمیم به‌طور کلی می‌تواند مدل بهتری برای شناسایی نمونه‌های مثبت باشد، زیرا بازخوانی بالاتری دارد.

در نهایت، انتخاب مدل بستگی به اهداف دارد. اگر به دقت کلی بیشتر نیاز داریم، رگرسیون لجستیک مناسب‌تر است، اما اگر به شناسایی هر چه بیشتر نمونه‌های مثبت نیاز داریم، درخت تصمیم مدل بهتری است.

5.خوشه‌بندی:

خوشه‌بندی **KMeans** و **سلسله‌مراتبی** (Hierarchical Clustering)دو الگوریتم محبوب برای تقسیم داده‌ها به گروه‌ها (خوشه‌ها) هستند، اما روش‌ها و ویژگی‌های آن‌ها تفاوت‌های زیادی دارند. در ادامه، تفاوت‌های اصلی این دو الگوریتم را توضیح می‌دهیم.

1.نوع الگوریتم:

- **KMeans**
  - یک الگوریتم **غیر سلسله‌مراتبی** و **تخصیص خوشه‌ای** است که به **kخوشه** نیاز دارد. یعنی باید تعداد خوشه‌ها را از قبل مشخص کنید.
  - در این الگوریتم، ابتدا k مرکز (Centroids) به‌طور تصادفی انتخاب می‌شود و سپس داده‌ها به نزدیک‌ترین مرکز تعلق داده می‌شوند. این فرایند تکرار می‌شود تا جایی که مراکز به حالت بهینه برسند.
  - الگوریتم **تکراری** است و تا زمانی که تغییرات زیادی در تخصیص داده‌ها به خوشه‌ها مشاهده نشود، ادامه می‌یابد.
- **سلسله‌مراتبی:(Hierarchical Clustering)**
  - یک الگوریتم **سلسله‌مراتبی** است که به طور طبیعی یک درخت **Dendrogram**تولید می‌کند که می‌تواند برای انتخاب تعداد خوشه‌ها به کار رود.
  - این الگوریتم به دو روش **Agglomerative**(**افزایشی**) **Divisive**(**تفکیکی**)اجرا می‌شود :
    - **Agglomerative:**شروع با هر نقطه به عنوان یک خوشه و سپس خوشه‌ها با هم ترکیب می‌شوند.
    - **Divisive:**شروع با یک خوشه بزرگ و سپس آن را به بخش‌های کوچکتر تقسیم می‌کند.
  - در این الگوریتم، تعداد خوشه‌ها از پیش تعیین نمی‌شود و می‌توانید با استفاده ازDendrogram ، سطح دلخواه خوشه‌بندی را انتخاب کنید.

2.نیاز به تعداد خوشه‌ها:

- **KMeans**:نیازمند این است که تعداد خوشه‌ها (k) از قبل مشخص شود. بنابراین، انتخاب k می‌تواند چالشی باشد و تأثیر زیادی بر نتایج خوشه‌بندی دارد.
- **سلسله‌مراتبی** :نیاز به مشخص کردن تعداد خوشه‌ها ندارد. به جای آن، الگوریتم یک درخت سلسله‌مراتبی (Dendrogram)می‌سازد که می‌تواند به شما در انتخاب بهترین تعداد خوشه‌ها کمک کند.

3.شیوه انجام خوشه‌بندی:

- **KMeans**:
  - الگوریتم به صورت **تخصیصی** عمل می‌کند، به این معنی که داده‌ها به نزدیک‌ترین مرکز خوشه نسبت داده می‌شوند.
  - پس از تخصیص داده‌ها به خوشه‌ها، مراکز خوشه‌ها بروزرسانی می‌شوند و این فرایند تکرار می‌شود.
- **سلسله‌مراتبی** :
  - الگوریتم به صورت **اتصال یا تفکیک** داده‌ها عمل می‌کند. یعنی شروع با خوشه‌های کوچک در روش Agglomerativeیا یک خوشه بزرگ در روش Divisive و سپس با توجه به شباهت‌ها خوشه‌ها را ادغام یا تقسیم می‌کند.

4.پیچیدگی محاسباتی:

- **KMeans**:پیچیدگی محاسباتی **O(n*k*d)**است که در آن n تعداد نقاط داده، kتعداد خوشه‌ها، و d تعداد ویژگی‌ها است. این الگوریتم معمولاً سریع‌تر از روش سلسله‌مراتبی است به‌ویژه در مجموعه‌های داده بزرگ.
- **سلسله‌مراتبی** :پیچیدگی محاسباتی **O(n^2)**یا **O(n^3)**بسته به نحوه پیاده‌سازی و اندازه داده‌ها است. این الگوریتم به‌ویژه برای مجموعه داده‌های بزرگ به شدت زمان‌بر است.

5.توزیع داده‌ها و حساسیت به خوشه‌ها:

- **KMeans**:
  - برای داده‌های **کروی یا کروی‌شکل** که خوشه‌ها به راحتی قابل تشخیص هستند، مناسب است.
  - این الگوریتم **حساس به مقدار اولیه k و مراکز اولیه** است. انتخاب بد برای k یا مراکز می‌تواند منجر به نتایج ضعیف شود.
- **سلسله‌مراتبی** :

- برای داده‌های **غیر کروی** و پیچیده‌تر نیز مناسب است و می‌تواند ساختارهای پیچیده‌تر از داده‌ها را شناسایی کند.
- این الگوریتم معمولاً به **مرکز اولیه** حساس نیست و از آنجا که به صورت مرحله به مرحله خوشه‌ها را می‌سازد، معمولاً عملکرد بهتری در شناسایی خوشه‌های مختلف دارد.

6.مزایا و معایب:

- **KMeans**:
  - **مزایا** :سریع‌تر، ساده‌تر، و مناسب برای مجموعه‌های داده بزرگ.
  - **معایب** :نیاز به انتخابk ، حساس به مراکز اولیه و اینکه ممکن است نتایج با خوشه‌های غیر کروی یا نابرابر در اندازه‌ها دقیق نباشد.
- **سلسله‌مراتبی** :
  - **مزایا** :نیازی به تعیین k ندارد، می‌تواند خوشه‌های غیر کروی را به خوبی شبیه‌سازی کند، و خروجی Dendrogramمی‌تواند به انتخاب تعداد خوشه‌ها کمک کند.
  - **معایب** :پیچیدگی محاسباتی بالا، و زمانی که داده‌ها خیلی زیاد باشند، می‌تواند به طور قابل توجهی کند باشد.

7.نحوه مقایسه خوشه‌ها:

- **KMeans**:هر داده به یکی از k خوشه‌ها تخصیص داده می‌شود و هیچ ساختار سلسله‌مراتبی ندارد.
- **سلسله‌مراتبی** :نتایج خوشه‌بندی به صورت یک درخت سلسله‌مراتبی نمایش داده می‌شود که می‌تواند برای تحلیل روابط بین خوشه‌ها مفید باشد.

8.استفاده در داده‌های واقعی:

- **KMeans**:معمولاً زمانی استفاده می‌شود که تعداد خوشه‌ها از پیش مشخص باشد و داده‌ها تمایل به داشتن خوشه‌های کروی یا منظم دارند.
- **سلسله‌مراتبی** :زمانی استفاده می‌شود که می‌خواهیم خوشه‌بندی دقیق‌تری از داده‌ها داشته باشیم و به راحتی از داده‌ها درخت ساختاری استخراج کنیم.

نتیجه‌گیری:

- **KMeans** سریع‌تر است و برای داده‌های بزرگ و خوشه‌هایی که تمایل به شکل‌های منظم دارند، مناسب است. اما انتخاب k از قبل یک چالش است.
- **سلسله‌مراتبی** زمان‌برتر است و مناسب برای داده‌هایی است که خوشه‌های پیچیده دارند. همچنین نیاز به انتخاب k ندارد و می‌توان از Dendrogram برای تصمیم‌گیری در مورد تعداد خوشه‌ها استفاده کرد.

1.استانداردسازی داده‌ها با استفاده ازStandardScaler

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(x)
```

در این کد از StandardScaler برای مقیاس‌بندی داده‌ها استفاده می‌شود. در بسیاری از الگوریتم‌های یادگیری ماشین، به‌ویژه الگوریتم‌هایی مانندKMeans ، مقیاس‌بندی داده‌ها ضروری است. این کار باعث می‌شود که همه ویژگی‌ها (یا ستون‌ها) در همان مقیاس قرار بگیرند و از ایجاد مشکل به دلیل تفاوت مقیاس‌های داده‌ها جلوگیری شود.

- StandardScaler مقادیر را به‌گونه‌ای استاندارد می‌کند که میانگین هر ویژگی برابر با صفر و انحراف معیار آن برابر با یک باشد.
- این فرایند روی داده‌های Xاعمال شده و خروجی به متغیر X_scaledاختصاص می‌یابد.

2.تعریف محدوده k =تعداد خوشه‌ها

```
k_values = range(2, 7)
silhouette_scores = []
```

- در این بخش، یک محدوده k از ۲ تا ۶ تعریف می‌شود. این به این معنا است که ما در حال بررسی تعداد خوشه‌ها از ۲ تا ۶ برای خوشه‌بندی داده‌ها هستیم.
- لیستی به نام silhouette_scores نیز ایجاد می‌شود که در آن امتیازهای سیلوئت برای هر مقدار k ذخیره می‌شود.

3.اجرای خوشه‌بندی KMeans برای هر مقدار k و محاسبه امتیاز سیلوئت

```
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
```

```
cluster_labels = kmeans.fit_predict(x)
silhouette_avg = silhouette_score(x, cluster_labels)
silhouette_scores.append(silhouette_avg)
print(f"silhouette_score for k={k} : {silhouette_avg:.3f}")
```

- در اینجا یک حلقه for برای مقدارهای مختلف k اجرا می‌شود (از ۲ تا ۶).
- برای هر مقدار k ، یک مدل KMeans ساخته می‌شود با تعداد خوشه‌ها برابر با k.
- سپس از fit_predict() استفاده می‌شود تا داده‌ها را به خوشه‌های مختلف تقسیم کند و برچسب‌های خوشه‌ها را در cluster_labels ذخیره کند.
- برای ارزیابی کیفیت خوشه‌بندی، از silhouette_score استفاده می‌شود که معیار خوبی برای ارزیابی خوشه‌ها است. این امتیاز مقدار سیلوئت برای تعداد خوشه‌ها kرا محاسبه می‌کند.
- سپس این امتیاز در لیست silhouette_scores ذخیره می‌شود و برای هر مقدار k به طور جداگانه چاپ می‌شود.

نکته مهم در مورد:Silhouette Score

- Silhouette Scoreمعیاری است که کیفیت خوشه‌بندی را ارزیابی می‌کند. این امتیاز بین ۱- و ۱ است .
  - امتیاز نزدیک به ۱ نشان‌دهنده خوشه‌بندی خوب است، زیرا داده‌های هر خوشه به هم نزدیک‌تر هستند و از خوشه‌های دیگر فاصله بیشتری دارند.
  - امتیاز نزدیک به ۰ نشان‌دهنده وضعیت نامطمئن است، یعنی داده‌ها به وضوح در یک خوشه قرار نمی‌گیرند.
  - امتیاز منفی نشان‌دهنده خوشه‌بندی ضعیف است، زیرا داده‌ها بیشتر شبیه به داده‌های دیگر خوشه‌ها هستند تا خوشه خود.

4.یافتن بهترین مقدار k (Optimal k)

```
optimal_k = k_values[np.argmax(silhouette_scores)]
print(f"best k: {optimal_k}")
```

- پس از اجرای حلقه و محاسبه امتیازهای سیلوئت برای هر k ، حالا باید بهترین مقدار k (مقداری که بالاترین امتیاز سیلوئت را دارد) انتخاب شود.

- از np.argmax(silhouette_scores) برای یافتن اندیس بزرگترین امتیاز سیلوئت استفاده می‌شود و این اندیس به مقدار kمربوط به بهترین تعداد خوشه‌ها اشاره می‌کند.
- سپس مقدار optimal_k چاپ می‌شود که بهترین تعداد خوشه‌ها را نشان می‌دهد.

خوشه‌بندی:K-Means

```
scaler = StandardScaler()
X_normalized = scaler.fit_transform(x)
kmeans = KMeans(n_clusters=3, random_state=42)
new_dataset['KMeans_Cluster'] = kmeans.fit_predict(X_normalized)
```

- داده‌ها نرمال‌سازی می‌شوند و الگوریتم K-Means با ۳ خوشه اعمال می‌شود.

خوشه‌بندی سلسله‌مراتبی:

```
linked = linkage(X_sample, method='ward', metric='euclidean')
clustering = AgglomerativeClustering(n_clusters=3, metric='euclidean',
linkage='ward')
clusters = clustering.fit_predict(X_reduced)
```

- خوشه‌بندی سلسله‌مراتبی با استفاده از روش Ward انجام می‌شود.

6.تحلیل خوشه‌ها :

تحلیل خوشه‌ها در الگوریتمKMeans

در الگوریتمKMeans ، خوشه‌ها با توجه به ویژگی‌های داده‌ها به‌طور خودکار تقسیم‌بندی می‌شوند و میانگین‌ها و ویژگی‌های هر خوشه به ما کمک می‌کند تا جمعیت هدف و خصوصیات آن را شناسایی کنیم.

خوشه ۰(Cluster 0)

- **ویژگی‌های جمعیتی:**

- **جنسیت 64.4%** :از افراد این خوشه زن هستند که نشان‌دهنده یک جمعیت عمدتاً زن است.
- **سن** :میانگین سن ۴۴.۵۶ سال است، به این معنی که اعضای این خوشه نسبتاً جوان هستند.

- **ویژگی‌های سلامتی:**
  - این خوشه افراد بدون فشار خون بالا یا بیماری قلبی است.
  - سطح **HbA1c**متوسطی دارد (۵.۴۸)، که معمولاً برای افراد بدون دیابت معمول است.
  - سطح **گلوکز خون** در این خوشه ۱۳۶.۷۹ است که نشان‌دهنده احتمال ابتلا به دیابت است، اما این افراد بیشتر به عنوان پیش‌دیابتی محسوب می‌شوند.
  - **دیابت** :تنها ۶.۷۷٪ از اعضای این خوشه به دیابت مبتلا هستند که درصد پایینی است.
- **ویژگی‌های سبک زندگی:**
  - سطح **BMI**متوسط (۲۷.۹۳) است که نشان‌دهنده افرادی با وزن نسبتاً سالم، اما ممکن است برخی از آن‌ها کمی اضافه وزن داشته باشند.
  - تاریخچه سیگار کشیدن متوسط (۰.۳۲) که نشان‌دهنده شیوع کم سیگار در این گروه است.

خوشه ۱(Cluster 1)

- **ویژگی‌های جمعیتی:**
  - **جنسیت 55.1%** :از این خوشه زن هستند که باز هم نشان‌دهنده یک جمعیت زنانه است.
  - **سن** :میانگین سن این خوشه ۳۵.۲۹ سال است که نشان‌دهنده جمعیتی جوان‌تر است.
- **ویژگی‌های سلامتی:**
  - هیچ‌یک از اعضای این خوشه فشار خون بالا یا بیماری قلبی ندارند.
  - سطح **HbA1c**برابر ۵.۴۳ است که به احتمال زیاد این افراد هیچ‌کدام به دیابت مبتلا نیستند.
  - سطح **گلوکز خون** در این خوشه ۱۳۴.۰۶ است که نشان‌دهنده احتمال کمی برای ابتلا به دیابت است.
  - **دیابت** :تنها ۲.۹۹٪ از افراد دیابت دارند که درصد پایینی است.

- **ویژگی‌های سبک زندگی:**
  - **BMI**این خوشه به طور متوسط ۲۵.۵۳ است که به نظر می‌رسد اعضای این خوشه وزن متعادل دارند.
  - با تاریخچه سیگار کشیدن بالاتر (۳.۲۴)، این گروه ممکن است افرادی با عادات غذایی یا سبک زندگی ناسالم‌تر باشند.

خوشه ۲(Cluster 2)

- **ویژگی‌های جمعیتی:**
  - **جنسیت** :50.8% از اعضای این خوشه مرد هستند.
  - **سن** :میانگین سن ۶۳.۰۲ سال است که نشان‌دهنده جمعیت مسن‌تر است.
- **ویژگی‌های سلامتی:**
  - **فشار خون بالا** 71% :از اعضای این خوشه فشار خون بالا دارند، که نشان‌دهنده شیوع بالای بیماری‌های قلبی-عروقی است.
  - **بیماری قلبی** 37.6% :از اعضای این خوشه دچار بیماری قلبی هستند.
  - **سطح HbA1c** بالا (۵.۶۷) که نشان‌دهنده احتمال بالای دیابت است.
  - **سطح گلوکز خون** در این خوشه ۱۴۵.۲ است که نشان‌دهنده احتمال بالای ابتلا به دیابت است.
  - **دیابت** 23.2% :از اعضای این خوشه دیابت دارند.
- **ویژگی‌های سبک زندگی:**
  - **BMI**بالا (۲۹.۶۶) که احتمالاً به دلیل کم‌تحرکی یا سبک زندگی غیرسالم است.
  - تاریخچه سیگار کشیدن ۱.۶۵ است که نشان‌دهنده شیوع متوسط سیگار کشیدن در این گروه است.

تحلیل خوشه‌ها در الگوریتم Hierarchical Clustering

در این الگوریتم، خوشه‌ها به صورت سلسله‌مراتبی ساخته می‌شوند و ویژگی‌های میانگین هر خوشه به ما کمک می‌کند تا جمعیت و ویژگی‌های آن را بهتر بشناسیم. در ادامه، ویژگی‌های هر خوشه را در این الگوریتم بررسی می‌کنیم.

خوشه ۰(Cluster 0)

- **ویژگی‌های جمعیتی:**
  - **جنسیت** 50.8% :از اعضای این خوشه زن هستند.
  - **سن** :میانگین سن ۳۰.۶۱ سال است که نشان‌دهنده یک جمعیت جوان است.
- **ویژگی‌های سلامتی:**
  - تنها ۰.۲٪ از اعضای این خوشه فشار خون بالا دارند، که نشان‌دهنده این است که بیشتر این افراد سالم هستند.
  - 0.03%از اعضای این خوشه بیماری قلبی دارند.
  - سطح **HbA1c**پایین (۵.۳۷) است که احتمالاً هیچ‌کدام از این افراد به دیابت مبتلا نیستند.
  - سطح **گلوکز خون** 131.31است که به‌طور معمول نشان‌دهنده سطح گلوکز طبیعی است.
  - تنها ۱.۶۹٪ از این افراد دیابت دارند.
- **ویژگی‌های سبک زندگی:**
  - **BMI**متوسط (۲۴.۳۷) که نشان‌دهنده افراد سالم‌تر با وزن متعادل است.
  - تاریخچه سیگار کشیدن در این گروه متوسط (۲.۵۱) است.

خوشه ۱(Cluster 1)

- **ویژگی‌های جمعیتی:**
  - **جنسیت** 73.5% :از اعضای این خوشه زن هستند.
  - **سن** :میانگین سن ۵۴.۵ سال است که نشان‌دهنده جمعیت میانه‌سال است.
- **ویژگی‌های سلامتی:**
  - 11.9%از اعضای این خوشه فشار خون بالا دارند.
  - 0.39%از افراد بیماری قلبی دارند.
  - **HbA1c**متوسط (۵.۵۶) که ممکن است نشان‌دهنده افرادی با احتمال ابتلا به دیابت باشد.

- **سطح گلوکز خون** 140.19است که در مرز نرمال بودن قرار دارد.
- 9.5%از این خوشه دیابت دارند.

- **ویژگی‌های سبک زندگی:**
  - **BMI**بالا (۳۰.۱۷) که نشان‌دهنده افراد با اضافه وزن یا چاقی است.
  - تاریخچه سیگار کشیدن پایین‌تر از خوشه ۰.(0.96)

خوشه ۲(Cluster 2)

- **ویژگی‌های جمعیتی:**
  - **جنسیت** 28.4% :از اعضای این خوشه زن هستند.
  - **سن** :میانگین سن ۶۶.۵۲ سال است که این خوشه شامل افراد مسن‌تری است.
- **ویژگی‌های سلامتی:**
  - 43.3%از این خوشه فشار خون بالا دارند.
  - 69.8%از اعضای این خوشه دچار بیماری قلبی هستند.
  - **HbA1c**بالاتر (۵.۹۶) و **سطح گلوکز خون** بالاتر (۱۵۹.۹۷) است که این افراد معمولاً دچار دیابت هستند.
  - 35.99%از این افراد به دیابت مبتلا هستند.
- **ویژگی‌های سبک زندگی:**
  - **BMI**بالا (۲۹.۷۶) که نشان‌دهنده افزایش وزن و کم‌تحرکی است.
  - تاریخچه سیگار کشیدن ۲.۰۴ است که احتمالاً به دلیل سبک زندگی ناسالم است.

با مقایسه ویژگی‌های خوشه‌ها در هر دو الگوریتم، می‌توان نتیجه گرفت که هر خوشه نشان‌دهنده گروه‌های خاصی از افراد با ویژگی‌های جمعیتی، سلامتی و سبک زندگی متفاوت است:

۱. **خوشه ۰** :افراد جوان، سالم و با درصد پایین بیماری‌ها.

۲. **خوشه ۱ :**افراد میانه‌سال با احتمال ابتلا به دیابت و فشار خون بالا، ولی با سبک زندگی نسبتاً سالم‌تر از خوشه ۲.

۳. **خوشه ۲ :**افراد مسن با مشکلات سلامت جدی مانند فشار خون بالا، بیماری قلبی و دیابت.

برای برچسب‌گذاری هر خوشه، ابتدا باید ویژگی‌های مهم و متمایز هر خوشه را بررسی کنیم و سپس بر اساس ویژگی‌های اصلی و شاخصه‌های هر خوشه، یک برچسب مناسب برای آن اختصاص دهیم. برچسب‌ها باید به گونه‌ای باشند که توصیف دقیقی از ویژگی‌های جمعیتی و سلامتی هر خوشه ارائه دهند.

با توجه به تجزیه و تحلیل خوشه‌ها در الگوریتم‌های **KMeans** و **Hierarchical Clustering**، برچسب‌هایی مانند "خوشه سالم"، "خوشه پیش‌دیابتی" و "خوشه بیمار" برای هر گروه اختصاص میدهیم.

برچسب‌گذاری برای خوشه‌ها در الگوریتمKMeans

خوشه ۰:(Cluster 0)

- **ویژگی‌ها :**افراد جوان، سالم، بدون بیماری‌های فشار خون و بیماری‌های قلبی، با سطح گلوکز خون متوسط، و درصد پایین دیابت.
- **برچسب" :خوشه سالم "**
  - این خوشه نشان‌دهنده افرادی است که به طور کلی از نظر سلامتی وضعیت خوبی دارند. آن‌ها نسبتاً جوان هستند، و به نظر می‌رسد که هیچ‌کدام مشکلات عمده سلامتی ندارند.

خوشه ۱:(Cluster 1)

- **ویژگی‌ها :**افراد جوان‌تر (اما هنوز میانه‌سال)، بدون فشار خون بالا یا بیماری قلبی، اما احتمال ابتلا به دیابت بیشتر است. سطح گلوکز خون و HbA1c در مرز نرمال قرار دارد.
- **برچسب" :خوشه پیش‌دیابتی "**
  - این خوشه نشان‌دهنده افرادی است که هنوز به دیابت مبتلا نشده‌اند، اما احتمال ابتلا به آن در آینده برای آن‌ها بیشتر است. این افراد می‌توانند نیاز به تغییراتی در سبک زندگی خود داشته باشند.

خوشه ۲:(Cluster 2)

- **ویژگی‌ها :**افراد مسن با مشکلات سلامت جدی مانند فشار خون بالا، بیماری قلبی و دیابت BMI .بالا، سطح گلوکز خون و HbA1c بالا که نشان‌دهنده وضعیت دیابتی است.
- **برچسب" :خوشه بیمار "**
  - این خوشه نشان‌دهنده افرادی است که به وضوح مشکلات سلامتی جدی دارند، مانند بیماری‌های قلبی و دیابت. این افراد ممکن است به مراقبت‌های بهداشتی و درمانی خاص نیاز داشته باشند.

برچسب‌گذاری برای خوشه‌ها در الگوریتمHierarchical Clustering

خوشه ۰:(Cluster 0)

- **ویژگی‌ها :**افراد جوان، سالم، با سطح فشار خون و بیماری قلبی بسیار پایین. سطح HbA1c و گلوکز خون در حالت نرمال.
- **برچسب" :خوشه سالم "**
  - این خوشه شامل افراد سالم با وضعیت عمومی خوب است. آنان به احتمال زیاد هیچ‌گونه مشکلات عمده پزشکی ندارند.

خوشه ۱:(Cluster 1)

- **ویژگی‌ها :**افراد میانه‌سال با احتمال ابتلا به دیابت و فشار خون بالا BMI .بالا و سطح گلوکز خون در مرز نرمال، و درصد بیشتری از افراد دیابت دارند.
- **برچسب" :خوشه پیش‌دیابتی و اضافه وزن "**
  - این خوشه نشان‌دهنده افرادی است که در معرض خطر ابتلا به دیابت و فشار خون بالا هستند، و معمولاً دارای اضافه وزن یا چاقی هستند.

خوشه ۲:(Cluster 2)

- **ویژگی‌ها :**افراد مسن، با بیماری‌های قلبی و فشار خون بالا، سطح HbA1c و گلوکز خون بالا که به وضوح نشان‌دهنده وضعیت دیابتی و مشکلات سلامت مزمن است.
- **برچسب" :خوشه بیمار و مسن "**

- این خوشه شامل افراد مسن است که مشکلات سلامت جدی دارند، از جمله بیماری قلبی، فشار خون بالا و دیابت. آن‌ها نیاز به درمان‌های پزشکی و مراقبت‌های بهداشتی ویژه دارند.

نتیجه‌گیری

در نهایت، برچسب‌های مناسب برای خوشه‌های مختلف به شرح زیر است:

KMeans Clustering:

۱. **خوشه ۰**" :خوشه سالم"

۲. **خوشه ۱**" :خوشه پیش‌دیابتی"

۳. **خوشه ۲**" :خوشه بیمار"

Hierarchical Clustering:

۱. **خوشه ۰**" :خوشه سالم"

۲. **خوشه ۱**" :خوشه پیش‌دیابتی و اضافه وزن"

۳. **خوشه ۲**" :خوشه بیمار و مسن"

```
kmeans_labels = {0: 'not diabets', 2: 'diabets', 1: 'maybe diabets'}
```

```
hierarchica_labels = {1: 'maybe diabets',0: 'not diabets',2: 'diabets'}
```

- برچسب‌های خوشه‌ها مشخص می‌شوند.

```
sns.scatterplot(x=X_reduced[:, 0], y=X_reduced[:, 1],
hue=new_dataset['KMeans_Label'], palette='viridis')
```

- خوشه‌ها روی نمودار PCA مصورسازی می‌شوند.