به نام خدا

پروژه درس داده‌کاوی

استاد دانشپور

یاسمن پی‌خوش

مقدمه

دیابت یکی از شایع‌ترین بیماری‌های مزمن در جهان است که بر اثر افزایش سطح قند خون در بدن به وجود می‌آید. پیش‌بینی به‌موقع دیابت می‌تواند در جلوگیری از عوارض جدی آن مانند بیماری‌های قلبی، نارسایی کلیوی و آسیب به سیستم عصبی مؤثر باشد. تحلیل داده و داده‌کاوی و پیاده سازی مدل های یادگیری، در این زمینه به پزشکان و متخصصان بهداشت کمک می‌کند تا با استفاده از داده‌های موجود، بیماران در معرض خطر را شناسایی کنند و راهکارهای پیشگیری مناسب را ارائه دهند.

هدف پروژه

این پروژه دو هدف اصلی دارد:

۱. پیش‌بینی ابتلا به دیابت با استفاده از الگوریتم‌های طبقه‌بندی :

- الگوریتم‌های درخت تصمیم (Decision Tree) و KNN (K-Nearest Neighbors) به‌کار گرفته شده‌اند تا با استفاده از داده‌های بیماران، پیش‌بینی کنند که آیا فرد به دیابت مبتلا است یا خیر.

۲. خوشه‌بندی بیماران برای شناسایی گروه‌های پرخطر :

- با استفاده از الگوریتم‌های خوشه‌بندی K-means وHierarchical Clustering ، بیماران بر اساس ویژگی‌های سلامتی مانند سن، BMI و سطح HbA1c به گروه‌های مشابه تقسیم شده‌اند تا گروه‌های پرخطر شناسایی شوند.

در این پروژه از الگوریتم های بالا برای پیاده سازی مدل استفاده شده است.

داده‌های استفاده‌شده

معرفی مجموعه داده

مجموعه داده استفاده‌شده در این پروژه شامل اطلاعات مربوط به بیماران مختلف است. این مجموعه شامل ویژگی‌های زیر است:

- **سن :(Age)** اطلاعات مربوط به سن بیماران.
- **شاخص توده بدنی :(BMI)** میزان چاقی بیماران که مقدار طبیعی آن در واقعیت بین ۱۰ تا ۶۰ می‌باشد.

- **سطح HbA1c:** یک معیار برای ارزیابی کنترل قند خون طی ۲-۳ ماه گذشته.
- **سطح گلوکز خون (Blood Glucose Level):** مقدار گلوکز در خون بیماران که به صورت کلی مقدار مجاز آن بین ۷۰ تا ۳۰۰ میلی گرم بر دسی لیتر است.
- **تاریخچه استعمال دخانیات (Smoking History):** وضعیت استعمال دخانیات بیماران (مانند سابقه یا عدم سابقه استعمال دخانیات).
- **فشار خون بالا (Hypertension):** مشخص می‌کند که آیا بیمار دچار فشار خون بالا است یا خیر.
- **بیماری‌های قلبی (Heart Disease):** وضعیت وجود بیماری‌های قلبی در بیماران.
- **دیابت (Diabetes):** متغیر هدف (مشخص می‌کند بیمار دیابت دارد یا خیر).

مشکلات موجود در داده‌ها

- **مقادیر گمشده (Missing Values):** در داده‌های اولیه، برخی از ویژگی‌ها دارای مقادیر گمشده بودند:

```
Missing values before cleaning:
gender                 0
age                    2
hypertension           0
heart_disease          0
smoking_history        1
bmi                    0
HbA1c_level            1
blood_glucose_level    0
diabetes               0
dtype: int64
```

  - مقادیر گمشده در ویژگی‌های age و HbA1c_level وجود داشتند. این مقادیر با میانگین ویژگی مربوطه پر شدند.

- **داده‌های تکراری (Duplicate Rows):**

```
Duplicate rows before cleaning: 0
```

در مجموعه داده هیچ‌گونه داده تکراری وجود نداشت.

- **داده‌های پرت (Outliers):** برخی از مقادیر در ستون blood_glucose_level غیرمعمول بودند. به‌عنوان مثال، مقدار گلوکز بیشتر از ۳۰۰ یا کمتر از ۷۰ به‌عنوان داده پرت در نظر گرفته شد و حذف گردید.(در دیتاست مقادیر کمتر از ۷۰ وجود نداشت.)

مراحل پاک‌سازی و ذخیره داده‌ها

۱. مقادیر گمشده با میانگین هر ستون پر شدند.

۲. داده‌های پرت شناسایی و حذف شدند.

۳. داده‌های پاک‌سازی‌شده به یک فایل جدید با نام cleaned_modified_diabetes_prediction_dataset.csv ذخیره شدند .

پیش‌پردازش داده‌ها

مراحل انجام‌شده در پیش‌پردازش:

۱. **مدیریت مقادیر گمشده:**

- در داده‌های اولیه، برخی ویژگی‌ها دارای مقادیر گمشده بودند (مانند age، HbA1c_levelو smoking_history).
- برای رفع این مشکل، مقادیر گمشده با میانگین مقادیر موجود در هر ستون پر شدند.

```
# Handle missing values
# Fill missing BMI with the mean value
data['bmi'] = data['bmi'].fillna(data['bmi'].mean())
data = data[(data['bmi'] >= 10) & (data['bmi'] <= 60)]

# Fill missing HbA1c_level with the mean value
data['HbA1c_level'] = data['HbA1c_level'].fillna(data['HbA1c_level'].mean())

# Drop rows with any remaining missing values
data = data.dropna()
```

۲. **حذف داده‌های پرت:(Outliers)**

- مقادیر غیرعادی در ستون blood_glucose_level(بیش از ۳۰۰) شناسایی و حذف شدند.

```
# Handle outliers in blood_glucose_level
data = data[(data['blood_glucose_level'] <= 300)]
```

- برای ستون bmi، مقادیر باید بین ۱۰ تا ۶۰ باشد و سایر مقادیر، به عنوان داده‌ی پرت در نظر گرفته می‌شوند.

```
data = data[(data['bmi'] >= 10) & (data['bmi'] <= 60)]
```

۳. **حذف داده‌های تکراری:**

- بررسی داده‌ها نشان داد هیچ داده تکراری در مجموعه داده وجود ندارد.

```
# Check and remove duplicate rows
print(f"Duplicate rows before cleaning: {data.duplicated().sum()}")
data = data.drop_duplicates()
```

٤. **نرمال‌سازی ویژگی‌ها:**

- مقادیر ویژگی‌های عددی مانند age، bmi، HbA1c_level و blood_glucose_level به بازه‌ای نرمال تبدیل شدند تا مدل‌های یادگیری ماشین به داده‌های با مقیاس بزرگ حساس نباشند.

```
# Normalize numeric columns
scaler = MinMaxScaler()
columns_to_scale = ['age', 'bmi', 'HbA1c_level', 'blood_glucose_level']
data.loc[:, columns_to_scale] = scaler.fit_transform(data[columns_to_scale])
```

۵. **تعارضات و خطاهای منطقی:**

- در ستون age، به تعداد ۱۵۴۰ رکورد دارای سن منفی هستند. برای انجام پیش پردازش روی این قسمت، میانگین سن های مثبت انجام محاسبه و برای مقادیر منفی این مقدار میانگین جایگذاری شده است.(بعد از انجام دادن این مرحله به شکل های مختلف(حذف کردن سطرها)،در این روش مدل ها عملکرد بهتری داشتند.)

```
mean_age_positive = data.loc[data['age'] >= 0, 'age'].mean()

# Replace negative ages with the mean of positive ages
data.loc[:, 'age'] = data['age'].apply(lambda x: mean_age_positive if x < 0 else x)

# Fill missing ages with the mean of positive ages
data.loc[:, 'age'] = data['age'].fillna(mean_age_positive)
```

- در ستون gender، به تعداد ۱۹ سطر به جز دو مقدار male و female، مقادیر unknown و other وجود داشت که برای انجام پیش پردازش روی آن می‌توانیم سطرهای دارای این مقادیر را حذف کنیم و یا با مقدار مد در آن ستون جایگذاری کنیم. چون تعداد سطر های دارای این مقدار نسبت به کل داده ها کم است حذف آنها تاثیر زیادی نمی‌گذارد.

```
# Handle gender column
# Replace invalid genders with 'unknown'
valid_genders = ['male', 'female']
data['gender'] = data['gender'].apply(lambda x: x if x in valid_genders else 'unknown')
```

۶. تبدیل به بازه عددی:

در این قسمت از کد ستون هایی که دارای داده های اسمی هستند نیز به عدد تبدیل شدند:

```
# Encode categorical columns using one-hot encoding
data = pd.get_dummies(data, columns=['gender', 'smoking_history'], drop_first=True)
```

نتیجه پیش‌پردازش:

- داده‌های تمیز و نرمال‌شده به فایل جدیدی با نام cleaned_modified_diabetes_prediction_dataset.csvذخیره شدند.
- این مجموعه داده اکنون آماده استفاده در الگوریتم‌های طبقه‌بندی و خوشه‌بندی است.

طبقه‌بندی(Classification)

معرفی الگوریتم‌ها:

در این پروژه دو الگوریتم طبقه‌بندی برای پیش‌بینی دیابت به‌کار گرفته شده است:

۱. **درخت تصمیم** (Decision Tree) :

- یک مدل ساده و قابل تفسیر که بر اساس قوانین شرطی عمل می‌کند

در پیاده سازی این درخت تصمیم به روش زیر عمل شده است:

ابتدا فایل دیتاست داده های نمیز شده آپلود شده و سپس داده های آن به دو دسته test و train تقسیم شده‌اند.(همچنین متغیر هدف و ویژگی ها نیز مشخص شده‌اند.)

```
# Separate features (X) and target (y)
X = data.drop(columns=['diabetes'])  # 'diabetes' is the target column
y = data['diabetes']

# Split the dataset into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

سپس مدل تعریف شده و داده های test و train به مدل داده شده‌اند.

```
# ----------------------------
# Initialize models
# ----------------------------
decision_tree = DecisionTreeClassifier(random_state=42)
```

```
# ----------------------------
# Train and test Decision Tree
# ----------------------------
decision_tree.fit(X_train, y_train)
dt_predictions = decision_tree.predict(X_test)
```

و در نهایت معیار های ارزیابی مدل اعمال شده‌اند:

```
# Decision Tree evaluation
metrics['Decision Tree'] = {
    'Accuracy': accuracy_score(y_test, dt_predictions),
    'Precision': precision_score(y_test, dt_predictions),
    'Recall': recall_score(y_test, dt_predictions),
    'F1 Score': f1_score(y_test, dt_predictions),
    'Confusion Matrix': confusion_matrix(y_test, dt_predictions)
}
```

۲. K-Nearest Neighbors (KNN) :

- یک مدل مبتنی بر فاصله که پیش‌بینی را بر اساس داده‌های نزدیک به هر نمونه انجام می‌دهد.

به طور مشابه با قسمت پیاده سازی مدل درخت تصمیم گیری، مراحل زیر برای پیاده سازی مدل KNN طی شده است:

```
knn = KNeighborsClassifier(n_neighbors=5)  # Set the number of neighbors to 5
```

```
# ----------------------------
# Train and test KNN
# ----------------------------
knn.fit(X_train, y_train)
knn_predictions = knn.predict(X_test)
```

ارزیابی مدل‌ها:

الف) درخت تصمیم:(Decision Tree)

- تحلیل :
  - دقت بالا نشان می‌دهد که مدل عملکرد کلی خوبی در پیش‌بینی داده‌ها دارد.
  - حساسیت (۷۴.۲۸٪) نشان‌دهنده توانایی مدل در شناسایی موارد مثبت دیابت است، که در سناریوهایی مثل غربالگری اولیه بسیار مهم است.

ب) KNN

- تحلیل :
  - دقت بالاتر از Decision Tree نشان می‌دهد که KNN در پیش‌بینی کلی بهتر عمل می‌کند.
  - دقت مثبت بالا (۹۶.۲۴٪) نشان می‌دهد که مدل در کاهش موارد مثبت کاذب (FP) عملکرد خوبی دارد.
  - اما حساسیت کمتر (۶۲.۵۹٪) بیانگر این است که مدل توانایی کمتری در شناسایی تمام موارد مثبت دیابت دارد.

نتیجه‌گیری:

۱. Decision Tree :

- مناسب برای سناریوهایی که حساسیت (Recall) مهم‌تر است، مثلاً در غربالگری اولیه بیماران.
- به دلیل سادگی و قابل تفسیر بودن، می‌تواند در سیستم‌های تصمیم‌گیری اولیه استفاده شود.

۲. KNN :

- مناسب برای مواردی که دقت مثبت (Precision) اهمیت بیشتری دارد، مثلاً زمانی که کاهش هشدارهای نادرست (FP) ضروری است.
- برای داده‌هایی که مقیاس‌بندی‌شده هستند و ویژگی‌های عددی دارند، عملکرد خوبی نشان می‌دهد.

❖ نکته: برای ستون smoking_history در ابتدا مقادیر current,yes و ever یکسان در نظر گرفته شد و دقت برای دو طبقه بند به صورت زیر به دست آمد:

```
# Handle smoking column
# Replace 'current', 'yes', and 'ever' with 'current'
data['smoking_history'] = data['smoking_history'].replace(['current', 'yes', 'ever'], 'current')

# Replace 'not current' and 'former' with 'former'
data['smoking_history'] = data['smoking_history'].replace(['not current', 'former'], 'former')
```

```
Decision Tree Evaluation Metrics:
Accuracy: 0.9517734286286821
Precision: 0.7033132530120482
Recall: 0.7400950871632329
F1 Score: 0.7212355212355213
Confusion Matrix: [[26630   788]
 [  656  1868]]

KNN Evaluation Metrics:
Accuracy: 0.963195511321889
Precision: 0.8994382022471911
Recall: 0.634310618066561
F1 Score: 0.7439591078066915
Confusion Matrix: [[27239   179]
 [  923  1601]]
```

در حالتی که این مقادیر را یکسان در نظر نگیریم ارزیابی مدل به شکل زیر می‌شود:

```
Decision Tree Evaluation Metrics:
Accuracy: 0.9520406118495759
Precision: 0.7043576258452291
Recall: 0.7428684627575277
F1 Score: 0.7231006556112611
Confusion Matrix: [[26631   787]
 [  649  1875]]

KNN Evaluation Metrics:
Accuracy: 0.9624273595618196
Precision: 0.8972174900624645
Recall: 0.6259904912836767
F1 Score: 0.7374562427071178
Confusion Matrix: [[27237   181]
 [  944  1580]]
```

همانطور که از خروجی ها مشاهده می‌شود، با یکسان در نظر گرفتن بعضی از مقادیر، مدل درخت تصمیم در مقادیرaccuracy,precision,recall و f1score کاهش پیدا کرده و در ماتریس آشفتگی می‌بینیم که تعداد TPها کاهش ، FN ها افزایش،FPها افزایش و TNها کاهش پیدا کرده است. از آنجایی که در این مسئله پیدا کردن درست بیماری اهمیت زیادی دارد، به دنبال کاهش دادن FNها و افزایش دادن معیارrecall هستیم.پس برای درخت تصمیم گیری این پیش پردازش مناسب نبوده است.

اما برای طبقه بند KNN، با یکسان در نظر گرفتن این مقادیر مدل بهبود داشته است. در ماتریس آشفتگی هم با در نظر گرفتن این مقادیر به عنوان مقادیر یکسان داریم: TPها افزایش، FNها کاهش، TNها افزایش و FPها کاهش داشته است.

- ❖ برای اطمینان از اینکه نتایج تصادفی نیستند، از random_state استفاده شده است. راه های دیگری هم برای ارزیابی مثل Cross_validation وجود دارد. با تغییر ندادن پیش پردازش، در چندین بار اجرا، تغییری در نتایج ایجاد نشد.

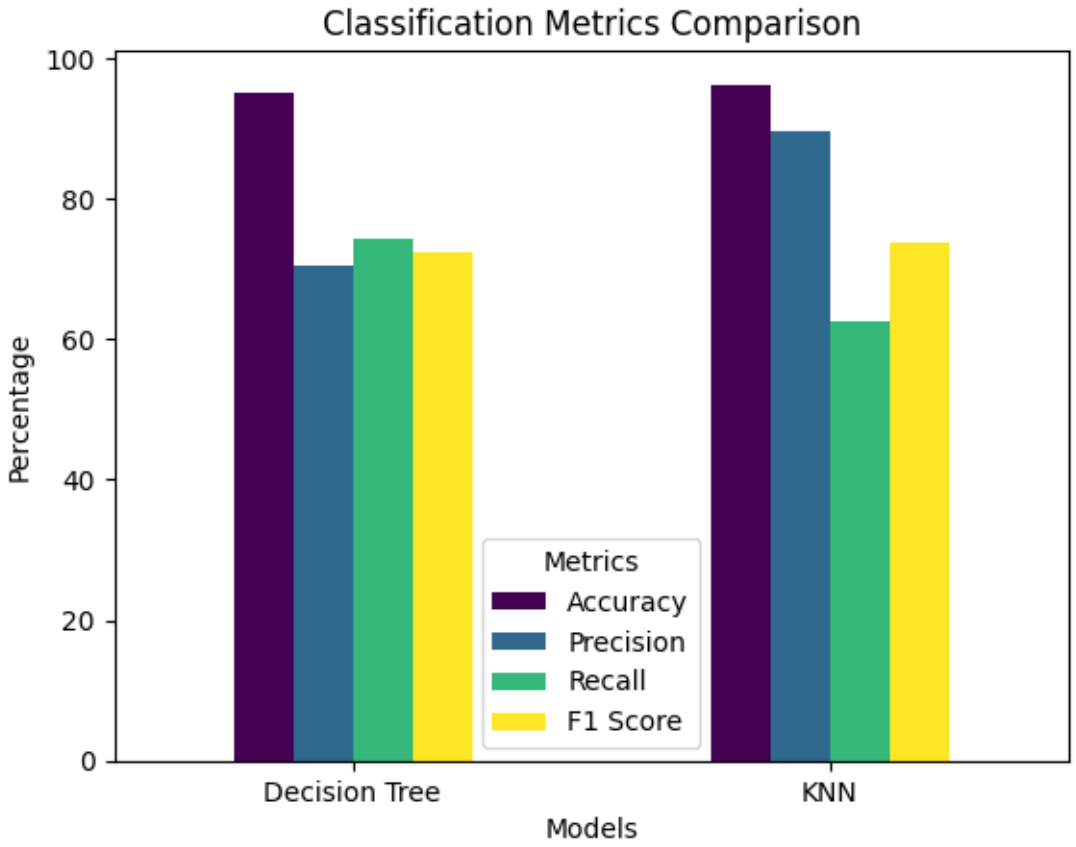
در مرحله بعدی برای بررسی نتایج دو مدل درخت تصمیم گیری و KNN آن هارا به صورت نموداری رسم می‌کنیم:

```python
# ----------------------------
# 1. Compare Classification Results
# ----------------------------
# Define classification metrics
classification_metrics = {
    'Decision Tree': {'Accuracy': 95.20, 'Precision': 70.43, 'Recall': 74.28, 'F1 Score': 72.31},
    'KNN': {'Accuracy': 96.24, 'Precision': 89.72, 'Recall': 62.59, 'F1 Score': 73.74}  # Updated metrics
}

# Convert to DataFrame for visualization
class_metrics_df = pd.DataFrame(classification_metrics).T
```

```python
# Plot classification metrics
plt.figure(figsize=(10, 6))
class_metrics_df.plot(kind='bar', colormap='viridis')
plt.title('Classification Metrics Comparison')
plt.ylabel('Percentage')
plt.xlabel('Models')
plt.xticks(rotation=0)
plt.legend(title='Metrics')
plt.savefig('classification_metrics_comparison.png')  # Save the classification metrics plot
plt.show()
```

خروجی:



تصویر ۱:"مقایسه متریک‌هایAccuracy ، Precision، Recallو F1 Score برای مدل‌های Decision Tree و KNN. Decision Tree حساسیت بیشتری دارد، در حالی که KNN دقت مثبت بالاتری ارائه می‌دهد".

تحلیل مقایسه متریک‌های طبقه‌بندی

در این مرحله براساس نمودار عملکرد دو مدل یادگیری ماشین **Decision Tree** و **K-Nearest Neighbors (KNN)** را بر اساس چهار متریک اصلی **(Accuracy, Precision, Recall, F1 Score)** را مقایسه می‌کنیم:

Accuracy

- **تعریف** :درصد نمونه‌هایی که به درستی طبقه‌بندی شده‌اند.
- **تحلیل** :
  - هر دو مدل دقت بالایی را نشان می‌دهند، اما مدل KNN با دقت ۹۶.۲۴٪ عملکرد بهتری نسبت به Decision Tree دارد.
  - این نتیجه نشان می‌دهد که KNN در تفکیک کلی داده‌ها عملکرد بهتری داشته و احتمال اشتباه در پیش‌بینی‌های آن کمتر است.
  - **کاربرد** :در مسائلی که خطای کلی اهمیت بالایی دارد، مانند سیستم‌های پیشنهادگر (Recommendation Systems)، مدل KNN انتخاب بهتری خواهد بود.

Precision .تعریف :درصد نمونه‌های پیش‌بینی‌شده مثبت که واقعاً مثبت هستند. این متریک نشان می‌دهد مدل تا چه اندازه توانسته است مثبت‌های کاذب (False Positives) را کاهش دهد.

- **تحلیل** :
  - مدل **KNN** با Precision ۸۹.۷۲٪ برتری قابل‌توجهی نسبت به Decision Tree دارد.
  - این عملکرد نشان می‌دهد که KNN در کاهش پیش‌بینی‌های اشتباه مثبت (FP) موفق‌تر است.
  - **کاربرد** :
    - در سناریوهایی که کاهش **هشدارهای اشتباه** اهمیت دارد، مانند سیستم‌های تشخیص تقلب (Fraud Detection) یا تشخیص اسپم(Spam Detection) ، استفاده از KNN مؤثرتر می‌باشد.

Recall

- **تعریف** :درصد نمونه‌های مثبت واقعی که به درستی شناسایی شده‌اند. این متریک توانایی مدل در شناسایی موارد مثبت را نشان می‌دهد.
- **تحلیل** :
  - مدل **Decision Tree**با Recall ٪۷۴.۲۸ عملکرد بهتری در شناسایی موارد مثبت واقعی دارد.
  - این نتیجه نشان می‌دهد که Decision Tree احتمال از دست دادن موارد مثبت واقعی (FN) را کاهش داده است.
  - **کاربرد** :
    - در مواردی که **شناسایی موارد مثبت بحرانی است**، مانند غربالگری بیماری‌ها، مدل Decision Tree انتخاب مناسب‌تری است.

F1 Score

- **تعریف** :میانگین هارمونیک Precision و .Recall این متریک توازن بین Precision و Recall را اندازه‌گیری می‌کند.
- **تحلیل** :
  - مدل‌ها از نظر امتیاز F1 بسیار نزدیک به هم هستند :
    - Decision Tree : ٪۷۲.۳۱
    - KNN : ٪۷۳.۷۴
  - این نزدیکی نشان می‌دهد که هر دو مدل به طور مشابه توانسته‌اند تعادل خوبی بین Precision و Recall ایجاد کنند.
  - **کاربرد** :

- در مواردی که توازن بین Precision و Recall اهمیت دارد (مانند تحلیل احساسات یا شناسایی خطاهای تولید)، هر دو مدل می‌توانند عملکرد قابل قبولی داشته باشند.

نتیجه‌گیری کلی

۱. **Decision Tree** :

- **مزیت** :حساسیت (Recall) بالاتر. مناسب برای کاربردهایی که کاهش موارد مثبت از دست رفته (FN)اهمیت دارد.
- **کاربرد** :در مسائل پزشکی، مانند غربالگری بیماری‌ها یا پیش‌بینی ریسک‌هایی که شناسایی موارد مثبت حیاتی است.

۲. **KNN** :

- **مزیت** Precision : بالاتر و Accuracy بهتر. مناسب برای کاربردهایی که کاهش مثبت‌های کاذب (False Positives) اهمیت دارد.
- **کاربرد** :در سیستم‌های تشخیص تقلب، شناسایی اسپم یا مواردی که دقت کلی مهم است.

جمع‌بندی

- **Decision Tree** مدل مناسبی برای شرایطی است که **حساسیت** بالاتر اهمیت بیشتری دارد و عدم شناسایی موارد مثبت هزینه بالایی دارد. پس در اینجا بهتر است که از این طبقه بند استفاده کنیم.
- **KNN** برای شرایطی که نیاز به **دقت بالا** و کاهش مثبت‌های کاذب داریم، مناسب‌تر است.

خوشه‌بندی(Clustering)

معرفی الگوریتم‌های خوشه‌بندی

در این پیاده‌سازی، دو الگوریتم زیر برای خوشه‌بندی داده‌ها استفاده شده است:

۱. K-means Clustering

- الگوریتم K-means یک روش یادگیری بدون نظارت است که داده‌ها را بر اساس نزدیکی به مراکز خوشه (Centroids) تقسیم می‌کند.
- تعداد خوشه‌ها از روی نمودار و روش Elbow انتخاب می‌شود. (در این پیاده سازی ۳ خوشه).
- این الگوریتم به دلیل سرعت و کارایی بالا برای داده‌های بزرگ مناسب است.

برای پیاده‌سازی مراحل زیر را انجام می‌دهیم:

فایل دیتاست داده‌های پیش پردازش شده را آپلود می‌کنیم.

```
cleaned_file_path = 'cleaned_modified_diabetes_prediction_dataset.csv'
data = pd.read_csv(cleaned_file_path)
```

سپس ویژگی های مناسب را برای خوشه بندی جدا می‌کنیم:

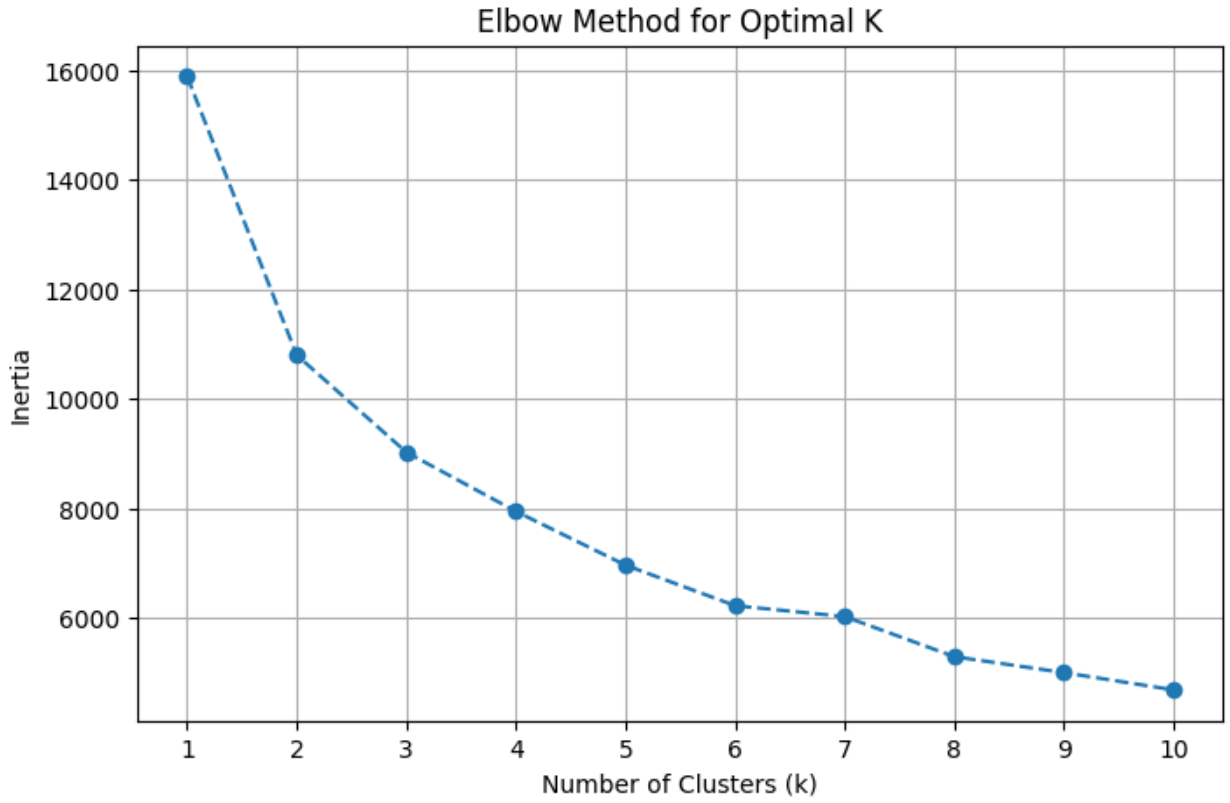
```
features = ['age', 'bmi', 'HbA1c_level', 'blood_glucose_level']
X = data[features]
```

در مرحله بعدی از نمودار Elbow استفاده می‌کنیم تا تعداد مناسب خوشه هارا پیدا کنیم:

```
# ---------------------
# 1. Elbow Method for Optimal K
# ---------------------
print("Determining optimal number of clusters using the Elbow method...")
inertia_values = []
k_values = range(1, 11)  # Test k from 1 to 10
```

وسپس نمودار را نمایش می‌دهیم:

```python
# Plot Elbow curve
plt.figure(figsize=(8, 5))
plt.plot(k_values, inertia_values, marker='o', linestyle='--')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_values)
plt.grid()
plt.savefig('elbow_method_plot.png')  # Save the Elbow plot
plt.show()
```



Optimal number of clusters (k) chosen: 3

نمودار Elbow نشان‌دهنده رابطه بین تعداد خوشه‌ها (k) و مقدار خطای مربعات فاصله ها است. wcss معیاری برای مجموع مربعات فاصله نقاط از مراکز خوشه‌ها است؛ هرچه این مقدار کمتر باشد، داده‌ها بهتر خوشه‌بندی شده‌اند.

۱. **الگوی کاهش:**

- برای k = 1، wcss بیشترین مقدار (حدود ۱۶۰۰۰) را دارد؛ چون تمامی نقاط در یک خوشه قرار گرفته‌اند.
- با افزایش تعداد خوشه‌ها(k) ، مجموع خطا به سرعت کاهش می‌یابد. این کاهش نشان می‌دهد که هر چه تعداد خوشه‌ها بیشتر شود، داده‌ها بهتر تقسیم‌بندی می‌شوند.
- اما از k = 3 به بعد، نرخ کاهش به تدریج کمتر می‌شود و نمودار به حالت "صاف‌تر" نزدیک می‌شود.

۲. **انتخاب k بهینه(Optimal K)** :

- بر اساس روشElbow ، مقدار بهینه kجایی است که نرخ کاهش به طور قابل ملاحظه‌ای کاهش یابد و نمودار زاویه (elbow) مشخصی ایجاد کند.
- در این نمودار، نقطه تغییر زاویه در k= 3 مشاهده می‌شود. این به این معنی است که افزایش تعداد خوشه‌ها به بیش از ۳، بهبود معناداری در کاهش wcss ایجاد نمی‌کند.

۳. **نتیجه‌گیری:**

- مقدار بهینه خوشه (k) برای این داده‌ها برابر با **۳** است.
- انتخاب k= 3 باعث تعادل میان کاهش wcss و پیچیدگی مدل می‌شود. استفاده از تعداد خوشه‌های بیشتر ممکن است به overfitting منجر شود یا تحلیل داده‌ها را پیچیده‌تر کند.

خروجی:

- **Optimal number of clusters (k) chosen: 3**

این مقدار k به معنای این است که داده‌ها به بهترین شکل ممکن در ۳ خوشه دسته‌بندی می‌شوند.

حالا از آنجا که تعداد داده‌ها زیاد است،از MiniBatch برای سریع تر شدن مدل استفاده می‌کنیم:

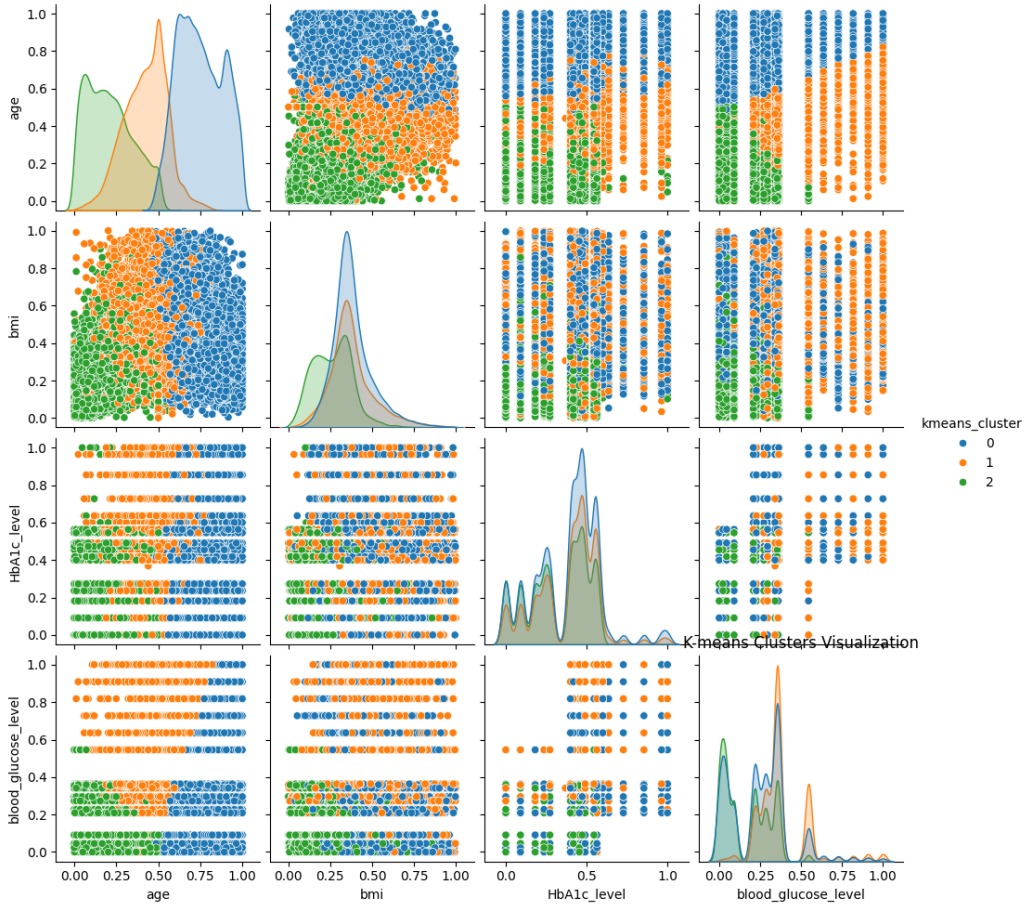
```
# ---------------------
# 2. K-means Clustering
# ---------------------
print("Performing K-means clustering with optimal k...")
kmeans = MiniBatchKMeans(n_clusters=optimal_k, random_state=42, batch_size=100)
data['kmeans_cluster'] = kmeans.fit_predict(X)
```

سپس خوشه هارا نمایش می‌دهیم:

```python
# Visualize K-means Clusters
sns.pairplot(data, hue='kmeans_cluster', vars=features, palette='tab10')
plt.title("K-means Clusters Visualization")
plt.savefig('optimized_kmeans_clusters.png')  # Save the plot
plt.show()
```

و نتایج خوشه بندی K-Means را ذخیره می‌کنیم:

```python
# Save K-means results
data[['kmeans_cluster']].to_csv('kmeans_clusters.csv', index=False)
print("K-means clustering results saved.")
```



تصویر ۲: "توزیع بیماران در خوشه‌های الگوریتم K-means Clustering بر اساس ویژگی‌های سن، شاخص توده بدنی، HbA1c، و سطح گلوکز خون. خوشه ۰ (آبی) نمایانگر بیماران پرخطر با مقادیر بالای HbA1c و گلوکز خون است".

تحلیل:

تحلیل نمودار Pairplot مربوط به خوشه‌بندی:K-means

نمودار Pairplot داده‌های مربوط به متغیرهای انتخاب شده برای خوشه‌بندی را به صورت توزیع و پراکندگی در بین سه خوشه (cluster) نشان می‌دهد. این تحلیل بر اساس متغیرهای مشخص شده و خوشه‌های برچسب‌گذاری شده (۰، ۱، ۲) انجام شده است.

رنگ‌ها و خوشه‌ها:

- هر خوشه با یک رنگ (نارنجی، سبز، آبی) مشخص شده است.
- نقاط داده در هر خوشه به طور مجزا توزیع شده‌اند که نشان‌دهنده جداسازی مناسب K-means است.

تحلیل ترکیب متغیرها:(Scatterplots)

- **محورهای متقاطع (Scatterplots)** :
  - پراکندگی نقاط در هر ترکیب از ویژگی‌ها مثلاً age و bmi نشان می‌دهد که خوشه‌ها بر اساس ترکیب ویژگی‌ها از هم تفکیک شده‌اند.
  - برخی خوشه‌ها در ویژگی‌های مشخص مانند bmi و blood_glucose_level همپوشانی دارند، اما در ترکیب با دیگر ویژگی‌ها بهتر جدا شده‌اند.

تحلیل توزیع تک متغیرها(Diagonals) :

- **age** :
  - خوشه‌ها از نظر سنی به خوبی جدا شده‌اند.
  - خوشه سبز (۲) شامل گروهی با سنین کمتر است، در حالی که خوشه آبی (۰) گروهی با سنین بالاتر را پوشش می‌دهد.
- **bmi** :
  - توزیع شاخص توده بدنی (BMI) بین خوشه‌ها همپوشانی بیشتری دارد.
  - خوشه آبی (۰) بیشتر شامل مقادیر میانگین bmi است.
- **HbA1c_level** :

- خوشه‌ها به خوبی جدا شده‌اند و خوشه آبی (۰) مقادیر بالاتر این شاخص را شامل می‌شود.

- **blood_glucose_level** :
  - خوشه‌ها بر اساس سطح گلوکز خون نیز به خوبی جدا شده‌اند، به‌ویژه خوشه آبی (۰) که سطح گلوکز بالاتری دارد.

همبستگی ویژگی‌ها با خوشه‌ها:

- **age** و **bmi** :
  - خوشه‌های آبی و نارنجی پراکندگی بیشتری نشان می‌دهند، در حالی که خوشه سبز محدود به گروه سنی پایین‌تر با مقادیر BMI مختلف است.
- **HbA1c_level** و **blood_glucose_level** :
  - خوشه‌ها کاملاً جدا شده‌اند؛ خوشه آبی دارای سطح HbA1c و گلوکز خون بالاتری است.

ویژگی خاص خوشه‌ها:

- خوشه آبی (۰) : افرادی با مقادیر بالای HbA1c_level و blood_glucose_level و معمولاً در گروه سنی بالاتر.
- خوشه سبز (۱) : افراد جوان‌تر با مقادیر نسبتاً کم blood_glucose_levelو HbA1c_level.
- خوشه نارنجی (۲) : ترکیب متعادلی از مقادیر bmiو age، اما در میانه طیف HbA1c_level.

---

۲. **Hierarchical Clustering**

- خوشه‌بندی سلسله‌مراتبی، داده‌ها را در یک ساختار درختی (دندروگرام) سازماندهی می‌کند.
- این روش به طور مکرر داده‌ها را بر اساس شباهت به یکدیگر ترکیب کرده یا تقسیم می‌کند.
- با توجه به حجم داده‌ها، خوشه‌بندی سلسله‌مراتبی روی یک نمونه کوچک (۱۰۰۰ داده) اجرا شده است.

برای پیاده‌سازی این الگوریتم، مراحل زیر را طی می‌کنیم:

در ابتدا با استفاده از Sample به صورت رندوم به تعداد ۱۰۰۰ داده برای الگوریتم استفاده می‌کنیم و برای مطمئن شدن از اینکه نتایج شانسی نیستند، از random_state استفاده می‌کنیم.سپس با استفاده از تابع Linkage و با متد ward برای نحوه ی محاسبه فاصله، ماتریس پیوند Z را ایجاد می‌کنیم.

```
# ----------------------------
# 3. Hierarchical Clustering
# ----------------------------
print("Performing Hierarchical clustering on a sample of data...")

# Random sampling to reduce data size
sampled_data = X.sample(n=1000, random_state=42)

# Perform hierarchical clustering
Z = linkage(sampled_data, method='ward')
```

در مرحله بعدی نمودار dendrogram را رسم می‌کنیم:

```
# Plot the dendrogram
plt.figure(figsize=(10, 7))
dendrogram(Z, truncate_mode='level', p=5)
plt.title('Hierarchical Clustering Dendrogram (Sampled Data)')
plt.xlabel('Data Points')
plt.ylabel('Distance')
plt.savefig('optimized_hierarchical_dendrogram.png')
plt.show()
```

در نهایت برچسب‌های خوشه‌ای (Cluster) را به داده‌های نمونه‌گیری شده اختصاص می‌دهیم و نتایج را فایل جداگانه‌ای ذخیره می‌کنیم:

```
# Assign clusters to sampled data
sampled_data['hierarchical_cluster'] = fcluster(Z, t=3, criterion='maxclust')

# Save Hierarchical Clustering results for the sample
sampled_data['hierarchical_cluster'].to_csv('hierarchical_clusters_sampled.csv', index=False)
print("Hierarchical clustering results (sampled) saved.")
```

تحلیل خوشه‌ها

برای هر یک از الگوریتم‌های خوشه‌بندی، مشخصات آماری و میانگین ویژگی‌های خوشه‌ها محاسبه شده و خوشه‌های پرخطر شناسایی شده‌اند:

۱. K-means Clustering

تحلیل ویژگی‌های میانگین هر خوشه نشان می‌دهد که خوشه‌ها به شکل زیر قابل تفسیر هستند:

```
K-means Cluster Characteristics:
                     age       bmi  HbA1c_level  blood_glucose_level
kmeans_cluster
0               0.748565  0.380230     0.385146             0.246320
1               0.423555  0.376155     0.387197             0.380731
2               0.211132  0.263157     0.323364             0.163905
```

- **خوشه ۰ :**بیماران مسن‌تر با مقادیر متوسط در همه ویژگی‌ها.
- **خوشه ۱ :**بیماران مسن‌تر با BMI و HbA1c بالا و سطوح بالای گلوکز خون. این خوشه ممکن است بیماران پرخطر را شامل شود.
- **خوشه ۲ :**بیماران جوان‌تر با BMI و HbA1c کمتر و سطح گلوکز خون پایین‌تر.
- نمودار توزیع ویژگی‌ها در هر خوشه، تفکیک واضحی بین گروه‌های مختلف را نشان می‌دهد.

---

۲. Hierarchical Clustering

تحلیل ویژگی‌های خوشه‌های سلسله‌مراتبی بر اساس دندروگرام و مشخصات آماری نشان می‌دهد.

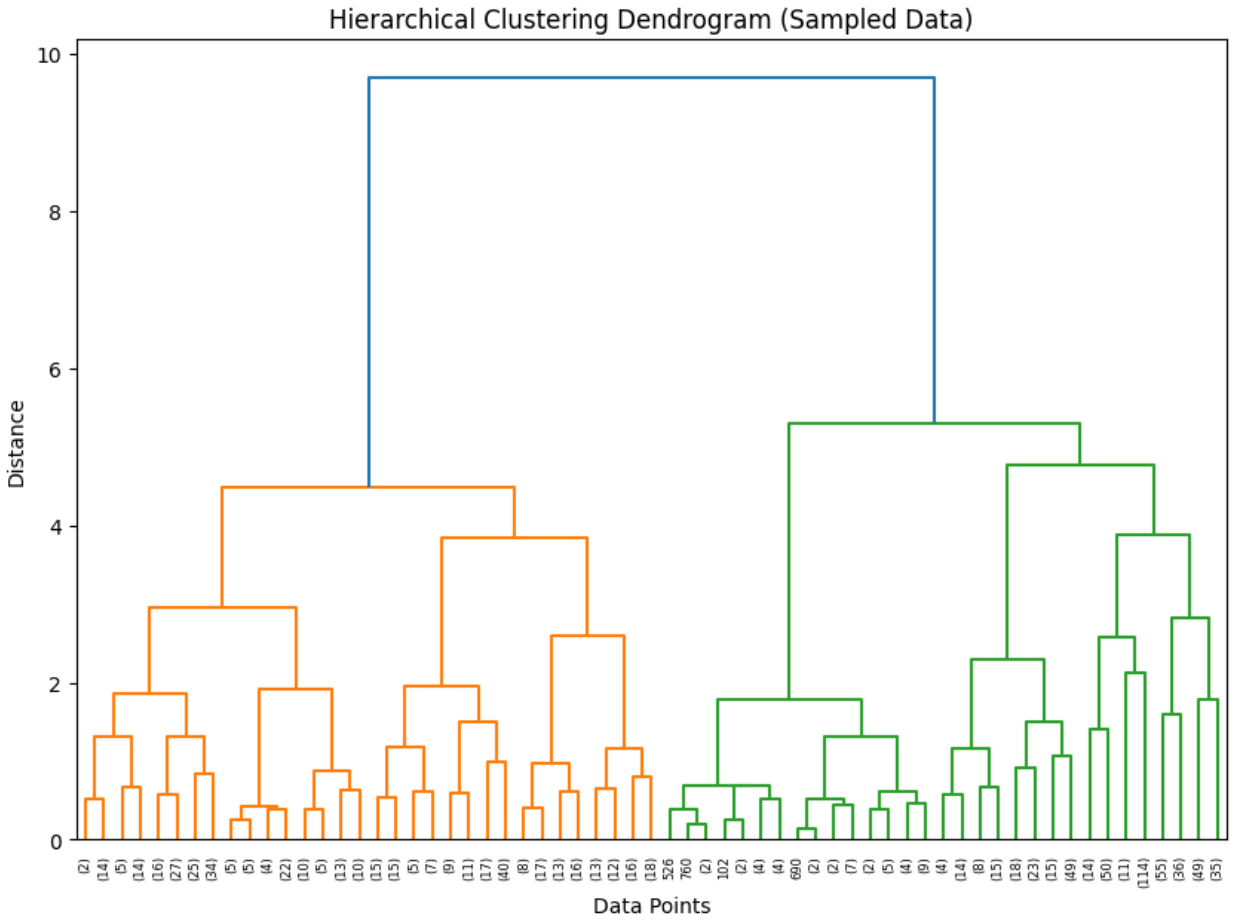
```
Hierarchical Cluster Characteristics (Sampled Data):
                          age       bmi  HbA1c_level  blood_glucose_level
hierarchical_cluster
1                    0.274723  0.290562     0.344757             0.222686
2                    0.800659  0.387662     0.588395             0.765957
3                    0.680031  0.392723     0.368699             0.252977
```

- **خوشه ۱ :**بیماران جوان‌تر با سطح پایین HbA1c و گلوکز خون.
- **خوشه ۲ :**بیماران مسن‌تر با مقادیر بالایBMI ، HbA1cو گلوکز خون، که نشان‌دهنده گروه پرخطر است.
- **خوشه ۳ :**بیماران مسن‌تر با BMIو HbA1c پایین‌تر.
- دندروگرام خوشه‌بندی سلسله‌مراتبی ساختار خوشه‌ها را نشان می‌دهد و بر اساس ارتفاع پیوندها، تعداد سه خوشه نهایی انتخاب شده است.

شناسایی خوشه‌های پرخطر

- بیماران حاضر در خوشه‌هایی با مقادیر بالای **HbA1cو گلوکز خون**)خوشه ۱ در K-means و خوشه ۲ در (Hierarchical Clustering احتمالاً در معرض خطر بیشتری برای عوارض دیابت هستند.

- این خوشه‌ها می‌توانند به‌عنوان اهداف اصلی برای مداخله‌های پزشکی و درمانی در نظر گرفته شوند.



تصویر ۳:"دندروگرام خوشه‌بندی سلسله‌مراتبی با استفاده از نمونه داده‌ها، ساختار سه خوشه اصلی را نشان می‌دهد. برش در سطح فاصله حدود ۴، سه خوشه متمایز را ایجاد می‌کند".

تحلیل دندروگرام خوشه‌بندی سلسله‌مراتبی

این دندروگرام ساختار سلسله‌مراتبی خوشه‌بندی داده‌ها را با استفاده از نمونه‌ای از داده‌ها (۱۰۰۰ نمونه) به تصویر می‌کشد. در این ساختار، هر گره نشان‌دهنده ادغام خوشه‌ها بر اساس معیار فاصله یا شباهت است. ارتفاع خطوط افقی در دندروگرام معیاری برای فاصله بین خوشه‌ها در هنگام ادغام است؛ هر چه این خطوط بلندتر باشند، شباهت بین خوشه‌های ادغام‌شده کمتر است.

ویژگی‌ها و جزئیات تحلیل:

۱. **ساختار سلسله‌مراتبی خوشه‌بندی**:

- فرآیند خوشه‌بندی از سطح داده‌های منفرد آغاز شده و به تدریج با ادغام داده‌های مشابه، خوشه‌های بزرگ‌تر تشکیل می‌شود.
- در سطوح پایین‌تر، داده‌هایی که شباهت بیشتری دارند سریع‌تر ادغام می‌شوند، در حالی که خوشه‌هایی با تفاوت‌های بیشتر در سطوح بالاتر ادغام می‌شوند.

۲. **سه خوشه اصلی**:

- با برش دندروگرام در سطح فاصله حدود ۴ (معیار مشخص برای تقسیم‌بندی)، **سه خوشه اصلی** شناسایی می‌شوند :
  - **خوشه نارنجی** :
    - این خوشه شامل داده‌هایی است که از نظر ویژگی‌ها به یکدیگر نزدیک‌تر هستند و احتمالاً مشخصات همگن‌تری دارند.
    - فاصله کمتر بین زیرخوشه‌ها نشان‌دهنده شباهت بالای داده‌ها در این خوشه است.
  - **خوشه سبز** :
    - این خوشه شامل داده‌هایی است که پراکندگی بیشتری دارند و ممکن است تنوع بیشتری در ویژگی‌های آن‌ها وجود داشته باشد.
    - خطوط بلندتر در ساختار این خوشه حاکی از زیرخوشه‌های متنوع‌تر است.
- **خوشه‌های دیگر** :ممکن است داده‌های باقی‌مانده در گروه‌های کوچک‌تر تقسیم شده باشند، اما برش دندروگرام در این سطح به ساده‌تر شدن تحلیل کمک می‌کند.

۳. **فاصله‌های بیشتر و ادغام‌های نهایی**:

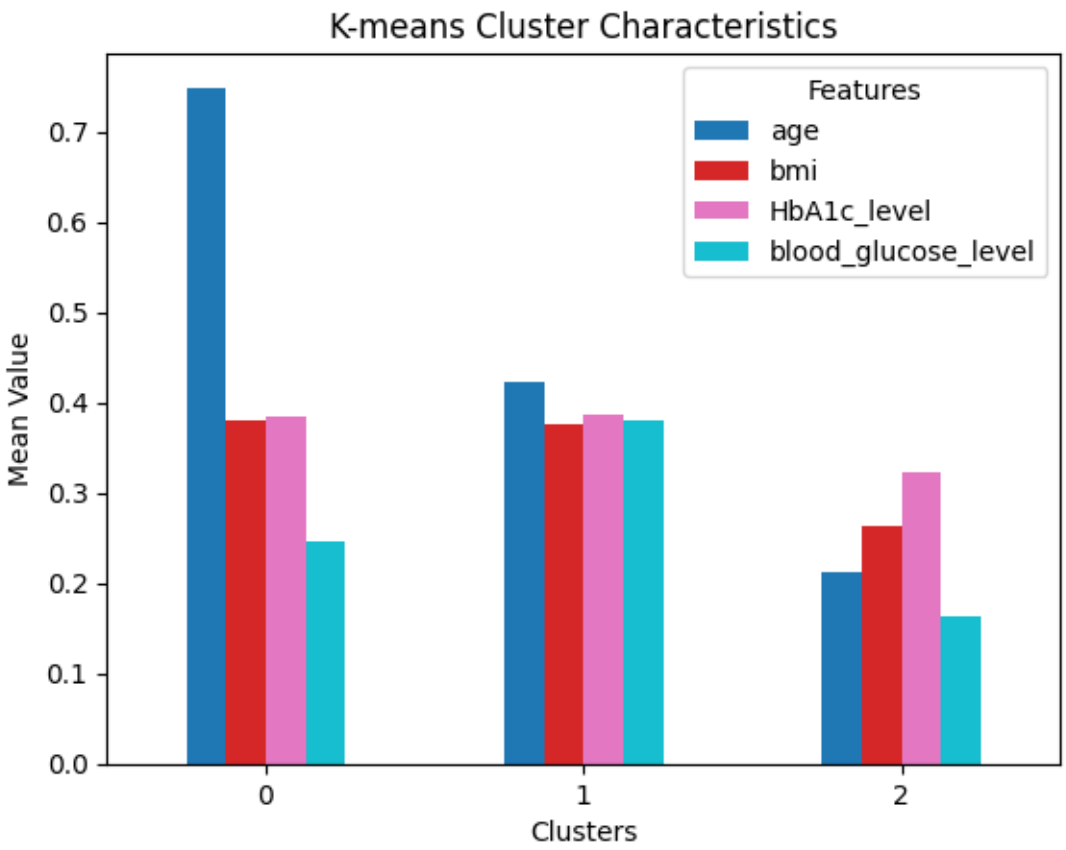
- خطوط بلند در انتهای دندروگرام (نزدیک به ارتفاع ۱۰) نشان‌دهنده فاصله زیاد بین خوشه‌های اصلی است. این فاصله‌های بیشتر حاکی از آن است که خوشه‌های اصلی دارای تفاوت‌های بنیادین هستند.
- این موضوع نشان می‌دهد که داده‌ها در دو یا سه گروه مجزا از هم دسته‌بندی می‌شوند که شباهت کمتری بین آن‌ها وجود دارد.

٤. کاربرد فاصله‌ها در خوشه‌بندی:

- خوشه‌بندی سلسله‌مراتبی این امکان را فراهم می‌کند که داده‌ها با شباهت‌های بسیار نزدیک‌تر در خوشه‌های پایین‌تر و داده‌های متفاوت‌تر در خوشه‌های بالاتر گروه‌بندی شوند.
- فاصله‌های بیشتر در دندروگرام کمک می‌کند تا تصمیم‌گیری بهتری در مورد تعداد خوشه‌ها صورت گیرد.

نتیجه‌گیری:

این دندروگرام ساختار سلسله‌مراتبی داده‌ها را به خوبی نمایش می‌دهد و نشان می‌دهد که داده‌ها در سه خوشه اصلی به بهترین نحو دسته‌بندی می‌شوند. خوشه نارنجی داده‌هایی با همگنی بالاتر را نشان می‌دهد، در حالی که خوشه سبز تنوع بیشتری را شامل می‌شود. فاصله‌های طولانی‌تر در ارتفاع‌های بالاتر نشان‌دهنده تفاوت‌های بیشتر بین خوشه‌های اصلی است و برش در سطح فاصله ۴ تعادل مناسبی بین تعداد خوشه‌ها و شباهت داده‌ها ارائه می‌دهد. این تحلیل می‌تواند برای درک بهتر گروه‌های داده و شناسایی الگوهای موجود در مجموعه داده‌ها استفاده شود.



تصویر ۴:"میانگین ویژگی‌های هر خوشه در الگوریتم K-means Clustering. خوشه ۱ نشان‌دهنده بیماران پرخطر با مقادیر بالای HbA1c و گلوکز خون است، در حالی که خوشه ۲ نمایانگر بیماران کم‌خطر است."

تحلیل مشخصات خوشه‌هایK-means

این نمودار، میانگین مقادیر ویژگی‌های کلیدی هر خوشه در الگوریتم K-means Clustering را نشان می‌دهد. این تحلیل شامل بررسی چهار ویژگی اصلی است :**سن**(age) ، **شاخص توده بدنی**(BMI) ، **سطحHbA1c** ، و **سطح گلوکز خون** .(blood_glucose_level) هر خوشه مشخصه‌های متفاوتی از گروه‌های بیماران را نشان می‌دهد که در ادامه به صورت تخصصی بررسی می‌شود:

خوشه ۲ (کم‌خطر):

- **ویژگی‌ها** :
  - **میانگین سنی پایین** :(0.3) این خوشه شامل بیماران جوان‌تر است.
  - **BMIپایین‌تر** :شاخص توده بدنی این گروه تقریبا ۰.۳ است.
  - **HbA1cپایین‌تر** :میانگین HbA1c این خوشه کمتر از سایرین است که نشان‌دهنده کنترل بهتر قند خون در این گروه است.
  - **سطح گلوکز خون پایین‌تر** :مقدار گلوکز خون در این خوشه کمترین مقدار را دارد.
- **تحلیل** :
  - این خوشه نمایانگر گروهی از افراد جوان است که در وضعیت سلامتی بهتری قرار دارند. این افراد ممکن است کمتر در معرض خطر بیماری‌های مرتبط با قند خون یا چاقی باشند.
  - **کاربرد** :تمرکز بر حفظ وضعیت فعلی این گروه از طریق کنترل‌های دوره‌ای و تشویق به سبک زندگی سالم.

خوشه ۰ (خطر متوسط):

- **ویژگی‌ها** :
  - **میانگین سنی بالا (۰.۷)**: این خوشه شامل بیماران مسن‌تر است که میانگین سنی بیشتری نسبت به خوشه‌های دیگر دارند.
  - **BMI متوسط** :این گروه در محدوده میانی شاخص توده بدنی قرار دارد.
  - **HbA1c و گلوکز خون متوسط** :مقادیر این دو شاخص در این خوشه نیز در محدوده متوسط است.
- **تحلیل** :

- این خوشه ممکن است شامل گروهی از بیماران باشد که در حال حاضر در معرض خطر متوسط هستند. احتمالاً این افراد به دلایلی مانند سبک زندگی ناسالم یا شروع مشکلات متابولیک، نیازمند نظارت و مداخله در مراحل ابتدایی هستند.
- **کاربرد** :برنامه‌ریزی برای پیشگیری از افزایش خطر در این گروه، از طریق آموزش تغذیه و ورزش.

خوشه ۱ (پرخطر):

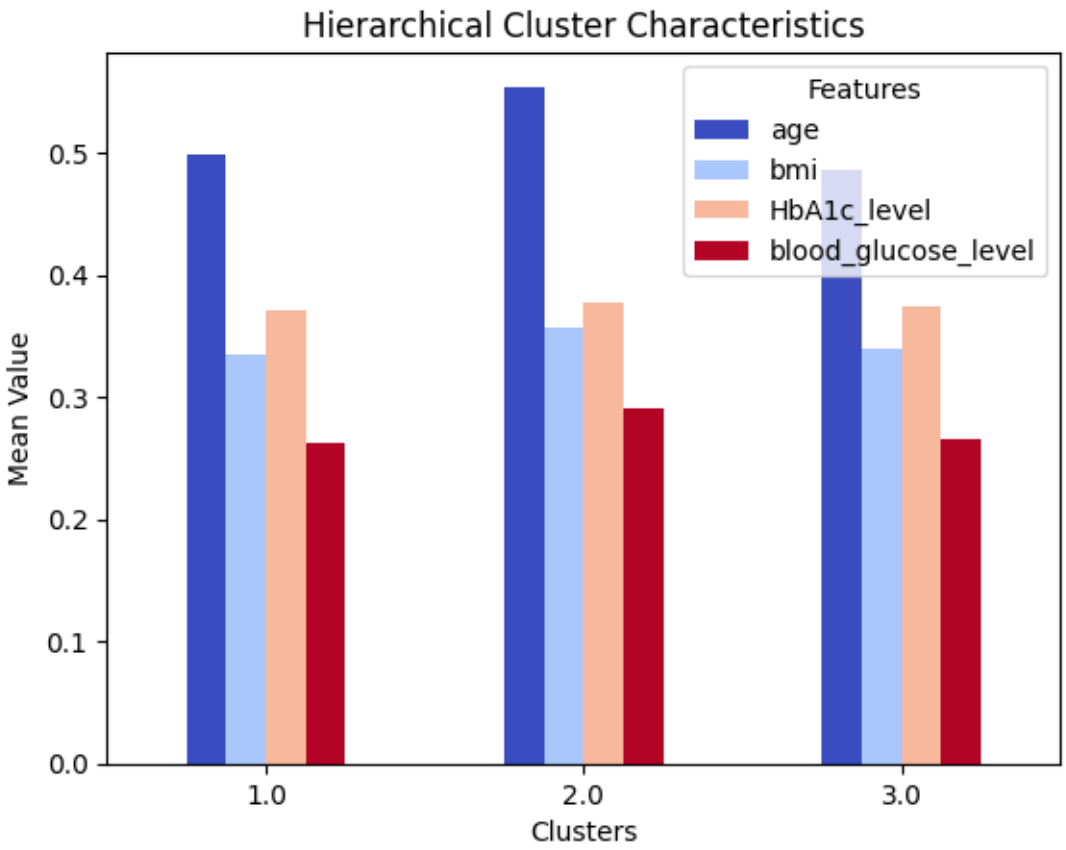
- **ویژگی‌ها** :
  - **میانگین سنی بالا (۰.۵)** : این خوشه نیز مانند خوشه ۰ شامل بیماران مسن‌تر است.
  - **BMIبالا** :شاخص توده بدنی در این خوشه بیشترین مقدار را دارد.
  - **HbA1cو گلوکز خون بالا** :این خوشه دارای بیشترین مقادیر HbA1c و گلوکز خون است که نشان‌دهنده وضعیت نامطلوب در این گروه است.
- **تحلیل** :
  - این خوشه گروه پرخطری را شامل می‌شود که احتمالاً در معرض بیماری‌هایی مانند دیابت و مشکلات مرتبط با چاقی قرار دارند. این بیماران ممکن است نیاز به مداخلات درمانی فوری و کنترل دقیق داشته باشند.
  - **کاربرد** :طراحی برنامه‌های درمانی اختصاصی برای کاهش ریسک در این گروه، از جمله کنترل دقیق قند خون، رژیم‌های غذایی خاص، و درمان‌های پزشکی.

جمع‌بندی:

۱. **خوشه ۲ (کم‌خطر)** :گروهی جوان‌تر با مقادیر متوسط شاخص‌ها. نظارت و پیشگیری برای جلوگیری از ورود به گروه پرخطر ضروری است

۲. **خوشه ۰ (خطر متوسط)** :گروهی مسن‌تر .ولی در وضعیت سلامتی خوب. تمرکز باید بر پیشگیری از زوال وضعیت آن‌ها باشد.

۳. **خوشه ۱ (پرخطر)** :گروهی مسن‌تر با شاخص‌های متابولیک بالا که نیاز به مداخلات درمانی فوری دارند.



تصویر ۵ : "میانگین ویژگی‌های هر خوشه در خوشه‌بندی سلسله‌مراتبی. خوشه ۳ بیماران مسن‌تر با BMI متوسط و خوشه ۲ بیماران با سطح بالای گلوکز خون را نشان می‌دهد."

تحلیل مشخصات خوشه‌های سلسله‌مراتبی

تحلیل:

این نمودار میانگین مقادیر چهار ویژگی اصلی **سن**(age) ، **شاخص توده بدنی**(BMI) ، **سطح** HbA1c و **سطح گلوکز خون** را برای هر یک از خوشه‌های حاصل از خوشه‌بندی سلسله‌مراتبی نشان می‌دهد. این تحلیل به تفکیک سه خوشه اصلی (۱، ۲ و ۳) ارائه شده و تفاوت‌های بین خوشه‌ها به طور دقیق‌تر بررسی می‌شود.

خوشه ۱: بیماران کم‌خطر

- **ویژگی‌ها** :
  - **میانگین سنی نسبتاً کمتر** :(0.49) این خوشه شامل بیماران جوان‌تر است.
  - **BMIکمی کمتر از خوشه‌های دیگر** :شاخص توده بدنی در این گروه کمتر است که می‌تواند نشانه‌ای از وزن سالم‌تر باشد.
  - **HbA1cو گلوکز خون مشابه خوشه‌های دیگر** :وضعیت این دو شاخص در حد متعادل باقی مانده است.
- **تحلیل** :
  - این خوشه نمایانگر گروهی از بیماران است که در وضعیت متابولیک نسبتاً سالم‌تری قرار دارند و در دسته کم‌خطر قرار می‌گیرند.
  - **کاربرد** :هدف اصلی در این گروه، حفظ وضعیت فعلی و پیشگیری از افزایش ریسک در آینده است.

خوشه ۲: بیماران با خطر متوسط به بالا

- **ویژگی‌ها** :
  - **میانگین سنی بالا (۰.۵۵) :** این خوشه شامل بیماران مسن‌تر است.
  - **BMIمتوسط** :این گروه از نظر شاخص توده بدنی در سطح متوسط قرار دارد.
  - **HbA1cمتوسط** :مقدار HbA1c نشان می‌دهد که وضعیت قند خون این بیماران در محدوده کنترل‌شده اما نه بهینه قرار دارد.
  - **سطح گلوکز خون بالاتر** :افزایش گلوکز خون در این خوشه نشان می‌دهد که این گروه ممکن است به دلیل وضعیت قند خون کنترل‌نشده در معرض خطر بیشتری باشد.
- **تحلیل** :

- این خوشه نمایانگر گروهی از بیماران است که نیاز به توجه بیشتری در مدیریت گلوکز خون دارند، چراکه سطح بالای گلوکز ممکن است نشانه‌ای از کنترل نامناسب یا پیشرفت بیماری باشد.

خوشه ۳: بیماران با خطر متوسط

- **ویژگی‌ها** :
  - **میانگین سنی کمی پایین‌تر (۰.۴۸)** : این خوشه شامل بیماران نسبتاً جوان‌تر از خوشه ۱ است.
  - **BMI** : شاخص توده بدنی این گروه تقریبا نزدیک به خوشه ۱ است.
  - **HbA1cمشابه خوشه ۲** : وضعیت HbA1c این خوشه تقریبا مشابه خوشه ۲ باقی مانده است.
  - **سطح گلوکز خون متوسط** : گلوکز خون این گروه نیز در محدوده متوسط است.

- **تحلیل** :
  - این خوشه نمایانگر گروهی از بیماران است که ممکن است در مراحل ابتدایی مشکلات متابولیک باشند یا بیماری‌های مرتبط با قند خون آن‌ها تحت کنترل نسبی باشد.
  - **کاربرد** :این گروه نیاز به نظارت منظم دارند تا از ورود به وضعیت پرخطر جلوگیری شود.

الگوهای شناسایی شده

۱. **میانگین سنی** :
   - بیماران مسن‌تر در خوشه ۲ قرار دارند و بیماران جوان‌تر در خوشه ۳ دیده می‌شوند.

۲. **سطح گلوکز خون** :
   - خوشه ۲ دارای سطح گلوکز خون بالاتری است که ممکن است نشان‌دهنده گروهی از بیماران با دیابت کنترل‌نشده یا در مراحل پیشرفته بیماری باشد.

۳. BMI:

- شاخص توده بدنی در خوشه‌ها نسبتاً نزدیک است، اما خوشه ۱ کمی کمتر از بقیه خوشه‌ها است که نشان‌دهنده وضعیت وزنی بهتر این گروه است.

نتیجه‌گیری

- **خوشه ۳ (خطر متوسط)** :
  - بیماران با وضعیت سلامتی کنترل‌شده اما نیازمند نظارت منظم.
- **خوشه ۲ (خطر متوسط به بالا)** :
  - بیماران با وضعیت گلوکز خون نامناسب که نیاز به مداخلات درمانی دقیق‌تر دارند.
- **خوشه ۱ (کم‌خطر)** :
  - بیماران سالم‌تر که هدف اصلی، حفظ وضعیت متعادل آن‌ها است.

❖ در سلول آخر کد، نمودارها برای مقایسه بهتر رسم شده‌اند.