سوال اول:

- Dimension: ساختاری که برای دسته‌بندی داده‌ها مورد استفاده قرار می‌گیرد و کار را برای پیدا کردن و استخراج الگوی داده‌ها آسانتر می‌کند.
- Outlier: داده‌هایی هستند که از داده‌های معمول ما به دور هستند و به آنها شباهت ندارند. این داده می‌توانند بسیار مفید واقع شوند.
- Independent variable: متغیرهایی که مستقل هستند و به عنوان متغیرهایی ورودی مدل مورد استفاده قرار می‌گیرند، گفته می‌شود. از این متغیرها می‌توان، متغیرهای وابسته را بدست آورد.
- Dependent variable: ترجمه آن متغیر وابسته است و از روی متغیر مستقل قابل محاسبه و پیش‌بینی است. معمولا هدف ما پیدا کردن مقدار این متغیرها است بنابراین به آنها متغیر هدف نیز می‌گویند.
- Stratified sampling: در این نمونه‌برداری ابتدا داده‌ها را به دسته های کوچکتر می‌شکنیم و سپس از هر دسته تعدادی را انتخاب می‌کنیم (در هر دسته داده‌های مشابه قرار می‌گیرند).

سوال دوم:

- Decimal Scaling: با جابجایی نقطه اعشار، مقادیر داده‌ها نرمال می‌شود. برای اجرای این تکنیک، هر مقدار داده را بر حداکثر مقدار مطلق داده‌ها تقسیم می‌کنیم.

$$v_i' = \frac{v_i}{10^j}$$

- Min-Max Normalization: در این تکنیک نرمال‌سازی داده‌ها، تبدیل خطی روی داده‌های اصلی انجام می‌شود. حداقل و حداکثر مقدار از داده ها پیدا می‌شود و هر مقدار مطابق فرمول زیر جایگزین می‌شود.

$$\mathbf{v' = \frac{v - min(A)}{max(A) - min(A)}(new\_max(A) - new\_min(A)) + new\_min(A)}$$

- z-Score Normalization: در این تکنیک، مقادیر بر اساس میانگین و انحراف معیار داده A نرمال می‌شوند. فرمول استفاده شده به صورت زیر است:

$$v' = \frac{v - \overline{A}}{\sigma_A}$$

سوال سوم:

در الگوریتم ChiMerge ابتدا یک مرحله مقداردهی اولیه وجود دارد. در این مرحله با مرتب سازی نمونه‌ها و مجموعه‌ها براساس مقدارشان برای ویژگی‌های در حال گسسته سازی و سپس تشکیل گسسته سازی اولیه مقدار دهی اولیه می‌شوند. سپس در انتها یک فرآیند مبتنی بر ادغام از پایین به بالا است، که در آن فواصل از طریق آماره‌های x^2 محاسبه می‌شود و بازه‌های مجاور با کمترین مقدار فاصله ادغام می‌شوند تا وقتی که یک شرط خاتمه برآورده شود.

سوال چهارم:

$$Cosine\ similarity: cos(x,y) = \frac{x.y}{\sqrt{(x.x)(y.y)}}$$

$$Correlation: \rho_{x,y} = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})}}$$

$$Euclidean\ distance: d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$Manhattan\ distance: |x_1 - x_2| + |y_1 - y_2|$$

$$Bhattacharya\ distance = -\ln(BC(x,y))$$

- $x = [1,1,1,1]$ & $y = [2,2,2,2]$
  $\cos(x,y) = 1$
  $\rho_{x,y} = 0$
  $d(x,y) = 2$
- $x = [0,1,0,1]$ & $y = [1,0,1,0]$
  $\cos(x,y) = 0$
  $\rho_{x,y} = 1$
  $d(x,y) = 2$
  $J(x,y) = 0$
- $x = [1,1,0,1,0,1]$ & $y = [1,1,1,0,0,1]$
  $\rho_{x,y} = \frac{1}{4}$
  $Manhattan\ distance = 2$
  $Bhattacharya\ distance = -\ln 3$
- $x = [2,-1,2.0,-3]$ & $y = [-1,1,-1,0,0,-1]$
  $\cos(x,y) = 0$

$$\rho_{x,y} = 0$$

سوال پنجم:

روش کاهش داده‌ها ممکن است به توصیف فشرده‌ای از داده‌های اصلی دست یابد که از نظر کمیت بسیار کوچکتر است اما کیفیت داده‌های اصلی را حفظ می‌کند.

روش های مختلفی برای این کار وجود دارد که به شرح زیر هستند:

- Data Cube Aggregation
- Dimension reduction: زمانی که به داده‌ای برخورد می‌کنیم که اهمیت کمتری دارد، فقط از ویژگی مورد نیاز برای تحلیل خود استفاده می‌کنیم. با این کار اندازه داده‌ها را کاهش می‌دهیم زیرا ویژگی های منسوخ یا اضافی را حذف می‌کنیم.
  - Step-wise Forward Selection
  - Step-wise Backward Selection
  - Combination of forwarding and Backward Selection
- Data Compression: تکنیک فشرده‌سازی داده‌ها با استفاده از مکانیزم‌های مختلف رمزگذاری، حجم فایل‌ها را کاهش می‌دهد.
  - Lossless Compression
  - Lossy Compression
- Numerosity Reduction: در این روش، داده‌های واقعی با مدل‌های ریاضی یا نمایش کوچکتر آنها جایگزین می‌شوند.
- Discretization & Concept Hierarchy Operation
  - Top-down discretization
  - Bottom-up discretization

سوال ششم:

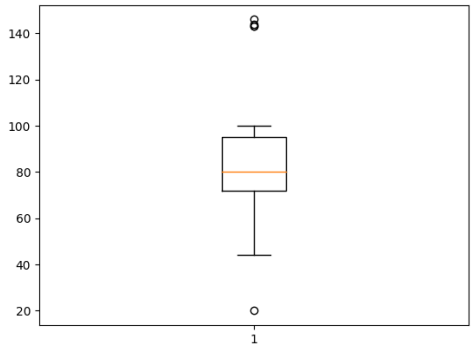
به هر چیزی که بتوانیم برای یک داده محاسبه کنیم، ویژگی گفته می‌شود. در استخراج ویژگی یا feature extraction تمام عوامل قابل محاسبه برای داده‌ها را بدست می‌آوریم. در انتخاب ویژگی یا feature selection از بین مجموعه ویژگی‌های موجود، یک زیرمجموعه ویژگی انتخاب می‌شود که معمولاً این زیرمجموعه از ویژگی‌ها، از همه مفیدتر هستند. به عبارت دیگر ابتدا استخراج ویژگی انجام شده و سپس از نتیجه‌ی آن برای انتخاب ویژگی استفاده می‌شود.

تبدیل موجک یک تکنیک پردازش سیگنال است که سیگنال‌های خطی را تبدیل می‌کند. هنگامی که این روش اعمال می‌شود، بردار داده به یک بردار عددی متفاوت متشکل از ضرایب موجک تبدیل می‌شود.

موجک تبدیل در کاهش داده‌ها نیز مفید است. اگر بخش کوچکی از قوی ترین ضرایب موجک را ذخیره کنیم، آنگاه می‌توان تقریب فشرده‌ی داده‌های اصلی را به دست آورد.

سوال هفتم:

نمودار جعبه‌ای به شکل زیر می‌باشد:



- Min = 44
- Max = 144
- Outliers = 20, 146

سوال هشتم:

- noise به طور پیش فرض نامطلوب است، زیرا مقدار اصلی ویژگی را تحریف می‌کند. outlierها به طور بالقوه می‌توانند درست باشند، و حتی شناسایی آنها می‌تواند هدف اصلی برخی از وظایف داده کاوی باشد. بنابراین، outlierها به طور بالقوه می‌توانند جالب و یا مطلوب باشند، اما نویز (طبق تعریف) اینطور نیست.
- وجود نویز در ویژگی‌ها می تواند داده‌ها را تصادفی‌تر یا غیرعادی‌تر به نظر برساند. بنابراین، ممکن است برخی از نمونه‌ها در داده های پر نویز به صورت outlier ظاهر شوند.
- outlierها می‌توانند اشیاء درست باشند که به نظر می‌رسد به مجموعه داده تعلق ندارند. آنها معمولاً به عنوان نویز طبقه بندی نمی‌شوند.

- نویز در داده‌ها می‌تواند به طور تصادفی برخی از مقادیر درست را غیرعادی و یا برخی نقاط پرت را به عنوان اشیاء درست جلوه دهد.

سوال نهم:

- مقادیر ممکن بین ۱- تا ۱ یا بین ۰ تا ۱ است.
- خیر، ممکن است دو داده ضریبی از یکدیگر باشند و برابر نباشند.
- اگر میانگین دو داده برابر صفر باشد آنگاه می‌تواند گفت Cosine Similarity و Correlation آنها باهم برابرند.

سوال دهم:

در نمودار quantile، ما بین داده‌ها و یک ویژگی آنها نتیجه‌گیری کنیم ولی در نمودار quantile-quantile دو مجموعه داده مجزا با یکدیگر مقایسه می‌شوند.

سوال یازدهم:

برای داده‌های عددی می‌توانیم از فاصله Minkowski استفاده کنیم که فرمول آن همانند زیر است:

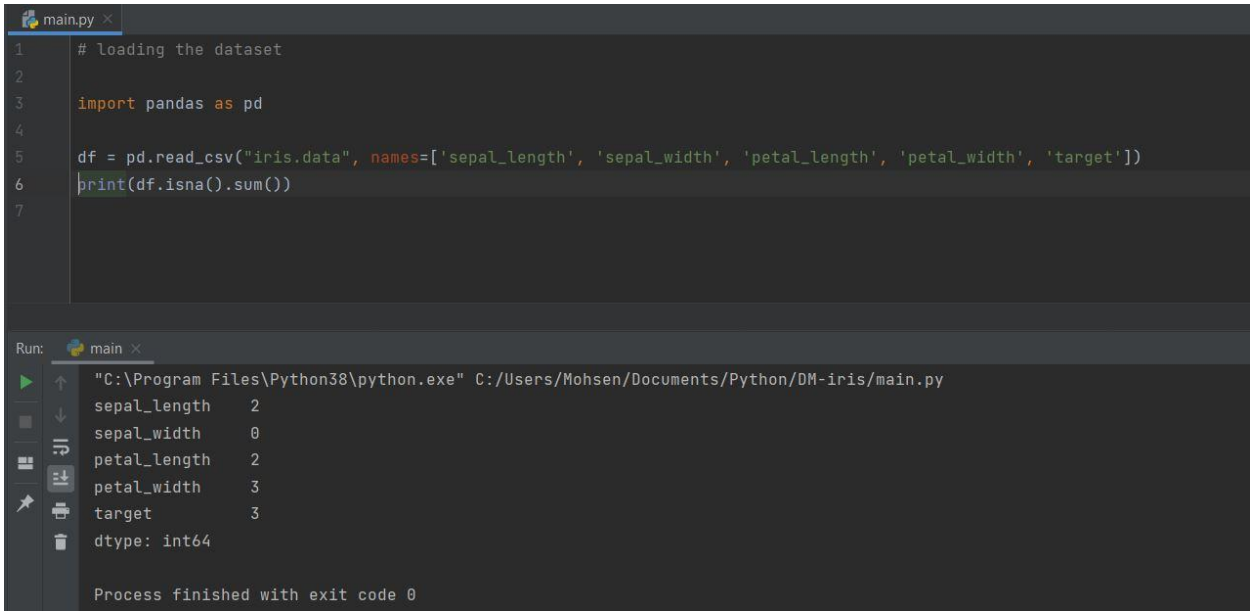
$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}.$$

برای داده‌های اسمی باید ابتدا آنها را به داده‌های باینری تقسیم کنیم و سپس از روش های دیگر محاسبه فاصله همانند جاکارد استفاده کنیم.

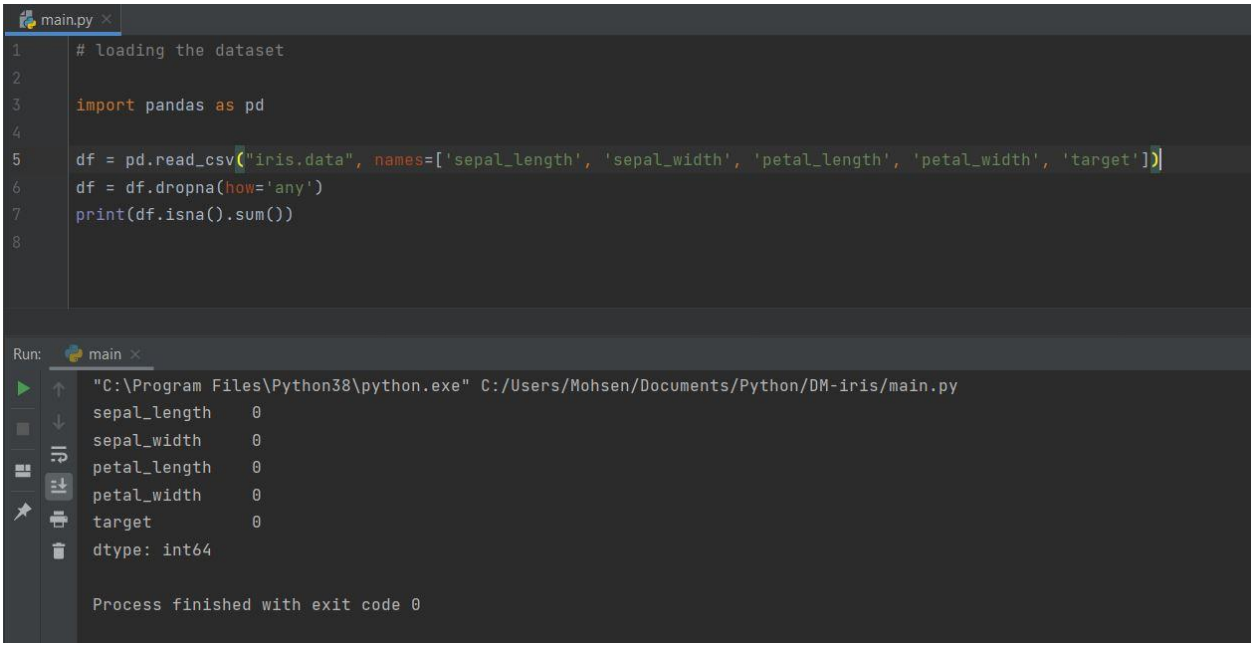
بخش پیاده‌سازی:

تمامی اسکرین‌شات ها در پوشه Screenshots قرارداده شده‌اند.

ابتدا تعداد ردیف‌های دارای مقدار NaN را برای هر ویژگی پیدا می‌کنیم:



سپس آنها را از مجموعه حذف می‌کنیم:



در ادامه داده‌های غیرعددی را کد می‌کنیم:

main.py

```python
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# loading the dataset
df = pd.read_csv("iris.data", names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'target'])
# missing values
df = df.dropna(how='any')
# label encoding
le = LabelEncoder()
le.fit(["Iris-setosa", "Iris-versicolor", "Iris-virginica"])
df['target'] = le.transform(df['target'])
print(df)

```

Run: main

```
"C:\Program Files\Python38\python.exe" C:/Users/Mohsen/Documents/Python/DM-iris/main.py
['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
     sepal_length  sepal_width  petal_length  petal_width  target
0             5.1          3.5           1.4          0.2       0
1             4.9          3.0           1.4          0.2       0
2             4.7          3.2           1.3          0.2       0
3             4.6          3.1           1.5          0.2       0
4             5.0          3.6           1.4          0.2       0
..            ...          ...           ...          ...     ...
153           6.7          3.0           5.2          2.3       2
154           6.3          2.5           5.0          1.9       2
155           6.5          3.0           5.2          2.0       2
157           6.2          3.4           5.4          2.3       2
158           5.9          3.0           5.1          1.8       2

[150 rows x 5 columns]
```

در بخش بعدی داده‌ها را نرمال می‌کنیم:

main.py

```python
# normalization
scaler = StandardScaler()
scaler.fit(df.get(['sepal_length', 'sepal_width', 'petal_length', 'petal_width']))
print('mean before normalization: ' + str(scaler.mean_))
print('variance before normalization: ' + str(scaler.scale_))
scaled_data = scaler.transform(df.get(['sepal_length', 'sepal_width', 'petal_length', 'petal_width']))
scaler.fit(scaled_data)
print('mean after normalization: ' + str(scaler.mean_))
print('variance after normalization: ' + str(scaler.scale_))
df['sepal_length'] = scaled_data[:, 0]
df['sepal_width'] = scaled_data[:, 1]
df['petal_length'] = scaled_data[:, 2]
df['petal_width'] = scaled_data[:, 3]

```

Run: main

```
"C:\Program Files\Python38\python.exe" C:/Users/Mohsen/Documents/Python/DM-iris/main.py
mean before normalization: [5.84333333 3.054      3.75866667 1.19866667]
variance before normalization: [0.82530129 0.43214658 1.75852918 0.76061262]
mean after normalization: [-4.73695157e-16 -6.63173220e-16  3.31586610e-16 -2.84217094e-16]
variance after normalization: [1. 1. 1. 1.]

Process finished with exit code 0
```

در انتها الگوریتم PCA را اجرا می‌کنیم و داده را پلات می‌کنیم:

نمودارهای جعبه‌ای به ترتیب ویژگی‌ها در پوشه Screenshots قرارداده شده‌اند.