Data Mining

HW1

Keivan Ipchi Hagh - 9831073

بخش تئوری

سوال اول

الف) داده noise می‌تواند ناشی از عوامل گوناگونی مانند اشتباه در مرحله data entry بوده و معمولا الگو خاصی ندارند. در صورتی که داده outlier نه تنها می‌تواند ناشی از خطا باشند، بلکه می‌تواند ناشی از الگو قشر کوچکی از جامعه آماری باشد که رفتار غیر معمولی دارد. به بیان دیگر، معمولا outlier داده ای است که معمولا دوست نداریم آن را داشته باشیم و سعی بر حدف آن داریم تا مدل دقیق تری داشته باشیم، در صورتی که داده noise لزوما داده بی معنی و بدون منطق و استفاده است.

ب) در مساءلی چون outlier detection، نیاز به داده outlier داریم تا بتوانیم همچین مس‌ءله ای را حل کنیم، پس در اینجا outlier ها نقش حیاتی ایفا می‌کنند زیرا می‌خواهیم آنها را پیشبینی کنیم! به طور کلی زمانی که این گونه داده ها مستند شده باشند که داده درست و legitimate هستند، برای ما خیلی مفید می‌توانند باشند.

ج) یک outlier می‌تواند noise هم باشد، ولی برعکس صادق نیست.

سوال دوم

در داده کافی، یک data warehouse سیستمی برای ذخیره داده های historical بوده که در گام های بعد بتوان بر روی آنها انواع تحلیل و برسی کرد.

یک database، مجموعه داده زنده و عملیاتی یک سیستم بوده که به طور مستقیم با کارکرد برنامه ارتباط دارد. در صورتی که یک data warehouse، معمولا به شکل passive کار کرد داشته و تیم های داده معمولا با آن کار کرده و تجزیه و تحلیل میکنند.

سوال سوم

در این روش ابتدا توضیع نرمال را بدست آورده و با استفاده از آن Q1 و Q3 را محاسبه می‌کنیم. سپس Inner Quantile Range یا همان IQR را محاسبه کرده که تفاضل Q1 و Q3 است. حال هر گونه داده ای که خارج بازه [Q1-1.5IQR, Q3+1.5IQR] باشد را outlier محسوب می‌کنیم.

سوال چهارم

الف) فرایند پاکسازی داده، به آن گفته می‌شود که داده های اشتباه، خراب، ناقص، فرمت اشتباه و یا تکراری را حذف و یا اصلاح کنیم. تا داده ما uniform و یکدست باشند.

ب) نمایش داده زمانی اهمیت دارد که بتوانیم با استفاده از آن درک شهودی بر داده داشته باشیم، به طوری که روابط و الگو های پیچیده را ساده سازی کنیم تا insight های معتبری از آن دریافت شود.

از چالش های آن استفاده از نمودار درست و یا نمایش داده های بیش از ۳ بعد می‌توان نام برد.

ج) پاکسازی داده یک requirement برای نمایش داده است، زیرا خطا در داده ها مثل فرمت آنها می‌تواند موجب ارور در مرحله نمایش شود. همچنین باعث می‌شود نمایش داده ما دقیق تر شود و داده تکراری و outlier ها نمایش را خراب نکنند.

سوال پنجم

الف)

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

Cosine Similarity) $\frac{(a.b)}{||a||.||b||} = \frac{516}{15.5*42.3} = 0.78$

Correlation) $SXY = \frac{\Sigma(xi-\bar{x})(yi-\bar{y})}{n-1} = \frac{395.4}{9} = 0.74 \quad r = \frac{S_{XY}}{S_X S_Y} = 0.74$

Mutual Information) $H(x) = -\sum \mathrm{p(xi)log2p(xi)}$

از آنجایی چه طبق فرمول نمیتوانیم اعداد منفی در لگاریتم داشته باشیم، همه اعداد دو بردار را scale کرده و سپس مقدار ثابت ۱۰ به هر دو اضافه می‌کنیم. این کار تغییری بر مقدار MI نمی‌گذارد زیرا این معیار را برای مقایسه دو بردار استفاده می‌کنیم.

$$G1)\ H(x) = -\sum p_i.\log_2 pi = \frac{-3}{18}\log_2\frac{-3}{18} + \cdots = 3.3212$$

$$G2)\ H(x) = -\sum p_i.\log_2 pi = \frac{9}{67}\log_2\frac{9}{67} + \cdots = 3.3213$$

$$H(G1;G2) = H(G1) + H(G2) - H(G1,G2) = 6.6425 - 6.36 = 0.2925$$

ب) با توجه به مقادیر بالای شباهت کسینوس و correlation می‌توان نتیجه گرفت که دو ژن وابسته هستند.

ج) تفاوت اعداد بستگی به روش محاسبه متریک ها دارد. به گونه ای که متریک کسینوس تمرکز را بر روی اندازه بردار ها گذاشته در صورتی که correlation فرمولی خلاف این موضوع دارد. در نتیجه اعداد متفاوت خواهند بود اما می‌توان با در نظر گرفتن شمای کلی گفت همه متریک ها از جنبه های مختلف، اما یک پاسخ را می‌دهند.

سوال ششم

روش اول: aggregation

تعریف: در این روش، داده ها را از چندین منبع (برای مثال دیتابیس های مختلف) گرفته و ترکیب می‌کنیم تا داده جدید و کاربردی تولید کنیم. در این روش معمولا ETL های متعددی انجام شده و مرحله ای است که داده ها از data lake به data warehouse منتقل می‌شوند.

مزایا:

- حجم داده در دست را کاهش می‌دهد
- تعداد feature های موجود را کاهش داده و تحلیل آن را ساده تر می‌کند
- این کار می‌تواند موجب افزایش دقت مدل شود، زیرا noise کمی کاهش پیدا می‌کند و داده ها دقیق تر می‌شوند

معایب:

- نیاز به منابع سخت افزاری دارد و می‌تواند زمان بر باشد (یکی از bottleneck های صنعت)
- می‌تواند موجب از دست رفتن برخی داده ها شود.

روش دوم: sampling

تعریف: این روش یک نمونه داده کوچک تری نسبت به داده اصلی به ما می‌دهد. ادامه تعریف در قالب مزایا و معایب آمده.

مزایا:

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

- مخصوصا در زمینه big data می‌تواند باعث شود روی داده کمتری تحلیل کرده و در نتیجه سرعت را بالاتر ببرد
- زمانی که دسترسی به سخت افزار قوی نداریم، می‌تواند کمک کننده باشد
- می‌تواند نماینده الگو های داده اصلی باشد

معایب:

- نوع نمونه برداری بسیار مهم است تا داده نمونه خصوصیات داده اصلی را به ارث برده باشد، و گرنه تحلیل های روی این داده فاقد اعتبار خواهند بود.
- می‌تواند موجب بوجود آمدن bias رو داده بشود.
- این کار همیشه ممکن نیست، مخصوصا زمانی که داده کم یا داده خیلی پیچیده ای داشته باشیم.

سوال هفتم

الف) مفهوم Feature Selection آن است که طبق روش هایی بتوانیم مهم ترین و پررنگ ترین Feature ها را انتخاب کرده تا در مدل از آنها استفاده کنیم. هدف از این کار، کاهش تعداد ویژگی ها و حذف ویژگی های غیر مرتبط است و روش های گوناگونی برای این کار وجود دارد مانند high correlation filter و یا low variance filter.

در مفهوم Feature Extraction میخواهیم از دیتا خام ویژگی ها کارآمد را استخراج کنیم که فرایند از قضا زمان بری است زیرا نیاز به تحلیل ها و نمایش های مختلف دارد. در این بخش سعی داریم ویژگی هایی را انتخاب کنیم تا با هدف نهایی ما ارتباط خوبی داشته و بتوانند کمک کنند. برای مثال correlation مناسبی داشته باشند.

مفهوم Feature Engineering مقداری متفاوت است. به طوری که در اینجا میخواهیم ویژگی ها را با هم ترکیب کرده تا ویژگی جدید به کارآمد تری بوجود آوریم. این روش هم تعداد ویژگی ها را کاهش داده و از ویژگی ها بی ربط، یک مرتبط میسازد.

سعی شد در تعاریف بالا تفاوت بین آنها توضیح داده شود. به طور کنی Feature Extraction در ابتدا انجام شده تا تعدادی ویژگی بدست آوریم. سپس Feature Engineering و Feature Selection انجام می‌شود تا ویژگی های برازنده را انتخاب کنیم.

سوال هشتم

ابتدا آنها را مرتب کرده و min، max، median، q1 و q3 را بدست می‌آوریم.

Sorted values: 1, 3, 25, 26, 27, 27, 27, 28, 29, 33, 36, 39, 41, 49, 70

Minimum: 1

Maximum: 70

Median: 28

First Quantile (Q1): 26

Third Quantile (Q3): 39

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

سوال نهم

الف) روش q-q- plot برای مقایسه توزیع دو ویژگی عددی است که معمولا برای این استفاده می‌شود تا توزیع ویژگی و همچنین تفاوت های بین آنها را بیابیم. در این روش ابتدا هر دو ویژگی را مرتب کرده و چارک ها را مشخص می‌کنیم. سپس هر دو ویژگی را روی نمودار می‌کشیم و با خط ۴۵ درجه y=x مقایسه می‌کنیم.

ب) این نمودار می‌تواند سه شکل داشته باشد:

- خط مورب: تمامی داده ها روی این خط بوده و توزیع مشابه/یکسان دارند که باعث شده توزیع روی یک خط بیفتد.

- خمیده به راست: معمولا زمانی این اتفاق می‌افتد که توزیع ها متفاوت بوده و میزان خمیدگی، میزان تفاوت این دو توزیع را نشان می‌دهد.

- خمیده به چپ: مانند خمیده به راست اما در جهت عکس

از این نمودار می‌توان فهمید آیا توزیع دو ویژگی مانند یکدیگر است و یا خیر (نوعی correlation). زمانی که خمیدگی زیاد باشد نشان دهنده آن است که توزیع دو ویژگی متفاوت است.

سوال دهم

الف) یک روش normalization و یا scaling بوده که داده را بین دو مقدار ۰ و ۱ میبرد. فرمول آن پایین تر آورده شده:

$$norm = \frac{x - \min(x)}{\max(x) - \min(x)}$$

ب) z-score هم یک روش دیگر برای standardization است که خروجی با میانگین ۰ و std برابر با ۱ تولید می‌کند. فرمول آن به شرح زیر است:

$$norm = \frac{x - mean(x)}{std(x)}$$

ج) یک روش هم مانند z-score خروجی با میانگین ۰ و std برابر با ۱ تولید می‌کند. فرمول آن به شرح زیر است:

$$r\_i = (y\_i - \hat{y}\_i) / (s\sqrt{(1 - h\_ii)})$$

سوال یازدهم

الف)

Loss function) $\sum(y - y^{hat})^2$

$Loss\ function\ to\ be\ minized\ using\ derivative) \frac{\partial s}{\partial B} = 2\sum(y - y^{hat})(-xiJ) = X^T(y - xB) = 0$

=> $B = (X^T X)^{-1}.(X^T y)$

Solving the problem using the above equations would result in:

- $X^T y = [33\ 121]$ and $X^T X = \begin{bmatrix} 6 & 18 \\ 18 & 70 \end{bmatrix}$ then Y = 1.15 + 1.15X

ب)

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

ج)

بخش عملی

سوال اول

با استفاده از دستور isna و یا info می‌توان به سادگی این مقادیر را بدست آورد. خروجی:

```
df.info()
```
✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   island             344 non-null    object 
 1   bill_length_mm     342 non-null    float64
 2   bill_depth_mm      342 non-null    float64
 3   flipper_length_mm  342 non-null    float64
 4   body_mass_g        342 non-null    float64
 5   sex                333 non-null    object 
 6   species            344 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

سوال دوم

با دستور dropna و چندین آرگومان مرتبط می‌توان به سادگی ردیف هایی که شامل حتی یک مقدار NaN هستند را پاک کرد. تعداد ردیف قبل این کار در **سوال اول** نمایش داده شد. بعد این کار:

```
df.dropna(axis = 0, how = 'any').info()
```
✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 333 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   island             333 non-null    object 
 1   bill_length_mm     333 non-null    float64
 2   bill_depth_mm      333 non-null    float64
 3   flipper_length_mm  333 non-null    float64
 4   body_mass_g        333 non-null    float64
 5   sex                333 non-null    object 
 6   species            333 non-null    object 
dtypes: float64(4), object(3)
memory usage: 20.8+ KB
```

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

سوال سوم

```
for col in ['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g']:
    df[[col]] = SimpleImputer(missing_values = np.nan, strategy = 'mean').fit_transform(df[[col]])
```

[26] ✓ 0.0s

```
for col in ['sex', 'island', 'species']:
    df[[col]] = SimpleImputer(missing_values = np.nan, strategy = 'most_frequent').fit_transform(df[[col]])
```

[27] ✓ 0.0s

```
df.info()
```

[28] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   island             344 non-null    object
 1   bill_length_mm     344 non-null    float64
 2   bill_depth_mm      344 non-null    float64
 3   flipper_length_mm  344 non-null    float64
 4   body_mass_g        344 non-null    float64
 5   sex                344 non-null    object
 6   species            344 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

سوال چهارم

```
island_mapping = {
    'Biscoe': 0,
    'Dream': 1,
    'Torgersen': 2
}
sex_mapping = {
    'female': 0,
    'male': 1
}
species_mapping = {
    'Adelie': 0,
    'Chinstrap': 1,
    'Gentoo': 2
}

for col, mapping in [('sex', sex_mapping), ('island', island_mapping), ('species', species_mapping)]:
    encoder = LabelEncoder()
    encoder.fit(list(island_mapping.values()))
    df[col] = encoder.transform([mapping[val] for val in df[col]])
```

[48] ✓ 0.0s

```
df
```

[49] ✓ 0.0s

	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	species
0	2	39.10000	18.70000	181.000000	3750.000000	1	0
1	2	39.50000	17.40000	186.000000	3800.000000	0	0
2	2	40.30000	18.00000	195.000000	3250.000000	0	0
3	2	43.92193	17.15117	200.915205	4201.754386	1	0
4	2	36.70000	19.30000	193.000000	3450.000000	0	0
...	...	...	...	...	...	...	...
339	1	55.80000	19.80000	207.000000	4000.000000	1	1
340	1	43.50000	18.10000	202.000000	3400.000000	0	1
341	1	49.60000	18.20000	193.000000	3775.000000	1	1
342	1	50.80000	19.00000	210.000000	4100.000000	1	1
343	1	50.20000	18.70000	198.000000	3775.000000	0	1

344 rows × 7 columns

سوال پنجم

این روش برای افزایش حجم داده استفاده می‌شود مخصوصا زمانی که داده زیادی در دسترس نیست. برای مثال در پردازش تصویر، این عمل با transformation های گوناگونی نظیر rotate، translate و scale انجام می‌شود. هدف این کار افزایش تنوع داده برای مدل حین آموزش است.

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

نکته مهم آن است که این عمل فقط روی داده آموزشی انجام شده و نه داده تستی! زیرا میخواهیم مدل generalize تر شود، پس اگر موقع آموزش استفاده نشود فایده ای هم برای ما نخواهد داشت.

سوال ششم

در روش upsample تعداد داده آموزشی را زیاد می‌کنیم و یا در تصویر اندازه آن را بزرگ می‌کنیم. در صورتی که در downsample تعداد و یا ابعاد کاهش پیدا می‌کند. معمولا downsample زمانی استفاده می‌شود که داده بیش از اندازه مورد نیاز داریم که نمیخواهیم در آموزش از آنها استفاده کنیم. پس بخشی را لحاظ نمی‌کنیم. تکنیک های متعددی برای این کار ها وجود دارد تا توزیع خراب نشود.

سوال نهم

وجه اشتراک هر دو روش در آن است که سعی بر برطرف کردن داده imbalanced دارند. هر دو مدل، روش های upsampling و oversampling را ترکیب کرده تا به این هدف برسند.

در SMOTEENN، ابتدا کلاس با اقلیت داده را oversampling کرده تا نمونه های جدید تولید شود. سپس undersampling به داده با تعداد زیاد کرده تا متعادل شوند که با استفاده از KNN این را اعمال می‌کند.

در SMOTETomek، به روشی مشابه کار می کنیم، اما به جای حذف نمونه هایی که توسط KNN به عنوان داده نویزدار یا ضعیف طبقه بندی شده اند، نمونه هایی را که به هر دو کلاس اقلیت و اکثریت تعلق دارند حذف می‌شوند.

سوال هشتم

برای SMOTETomek:

```
temp_df = df[['island', 'species']].copy()
temp_df.groupby(by = "island", as_index = False).agg({"species": "count"})
```

[93] ✓ 0.0s

	island	species
0	0	168
1	1	124
2	2	52

```
x, y = SMOTETomek(random_state = 0).fit_resample(temp_df.drop(columns = ["species"]), temp_df["species"])
Counter(y)
```

[94] ✓ 0.0s

```
Counter({0: 152, 2: 152, 1: 152})
```

برای SMOTEENN:

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

```
temp_df = df[['island', 'species']].copy()
temp_df.groupby(by = "island", as_index = False).agg({"species": "count"})
```

[98] ✓ 0.0s

	island	species
0	0	168
1	1	124
2	2	52

```
x, y = SMOTEENN(random_state = 0).fit_resample(temp_df.drop(columns = ["species"]), temp_df["species"])
Counter(y)
```

[99] ✓ 0.0s

```
Counter({0: 152})
```

سوال نهم

```
df.describe()
```

[102] ✓ 0.0s

	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	species
count	344.000000	344.000000	344.000000	344.000000	344.000000	344.000000	344.000000
mean	0.662791	43.921930	17.151170	200.915205	4201.754386	0.520349	0.918605
std	0.726194	5.443643	1.969027	14.020657	799.613058	0.500313	0.893320
min	0.000000	32.100000	13.100000	172.000000	2700.000000	0.000000	0.000000
25%	0.000000	39.275000	15.600000	190.000000	3550.000000	0.000000	0.000000
50%	1.000000	44.250000	17.300000	197.000000	4050.000000	1.000000	1.000000
75%	1.000000	48.500000	18.700000	213.000000	4750.000000	1.000000	2.000000
max	2.000000	59.600000	21.500000	231.000000	6300.000000	1.000000	2.000000

```
pd.DataFrame(
    data = StandardScaler().fit_transform(df.drop(columns = ["species"])),
    columns = [col for col in df.columns if col != "species"]
).describe()
```

[108] ✓ 0.0s

	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
count	3.440000e+02	3.440000e+02	3.440000e+02	3.440000e+02	3.440000e+02	3.440000e+02
mean	8.262125e-17	-1.156697e-15	4.131062e-16	-8.262125e-16	8.262125e-17	-6.454785e-18
std	1.001457e+00	1.001457e+00	1.001457e+00	1.001457e+00	1.001457e+00	1.001457e+00
min	-9.140204e-01	-2.174858e+00	-2.060444e+00	-2.065333e+00	-1.880837e+00	-1.041561e+00
25%	-9.140204e-01	-8.548868e-01	-7.889322e-01	-7.796428e-01	-8.162745e-01	-1.041561e+00
50%	4.650279e-01	6.035444e-02	7.569585e-02	-2.796522e-01	-1.900612e-01	9.600978e-01
75%	4.650279e-01	8.422188e-01	7.877425e-01	8.631834e-01	6.866374e-01	9.600978e-01
max	1.844076e+00	2.884265e+00	2.211836e+00	2.148873e+00	2.627899e+00	9.600978e-01

Data Mining

HW1

Keivan Ipchi Hagh - 9831073

سوال دهم

```
pd.DataFrame(
    data = PCA(n_components = 3).fit_transform(df),
    columns = ["pca_1", "pca_2", "pca_3"]
)
```

31] ✓ 0.0s

	pca_1	pca_2	pca_3
0	-2.267511	1.266175	-0.048591
1	-2.078927	-0.528109	0.842508
2	-2.143540	-0.409198	1.103311
3	-0.604432	1.186353	0.705630
4	-2.594861	-0.183224	0.554844
...	...	...	...
339	0.448975	2.107021	1.421763
340	-1.012636	-0.547222	0.739321
341	-0.379978	1.294431	0.598062
342	0.398441	1.669301	0.819717
343	-0.536183	0.003741	1.470134

344 rows × 3 columns

Data Mining

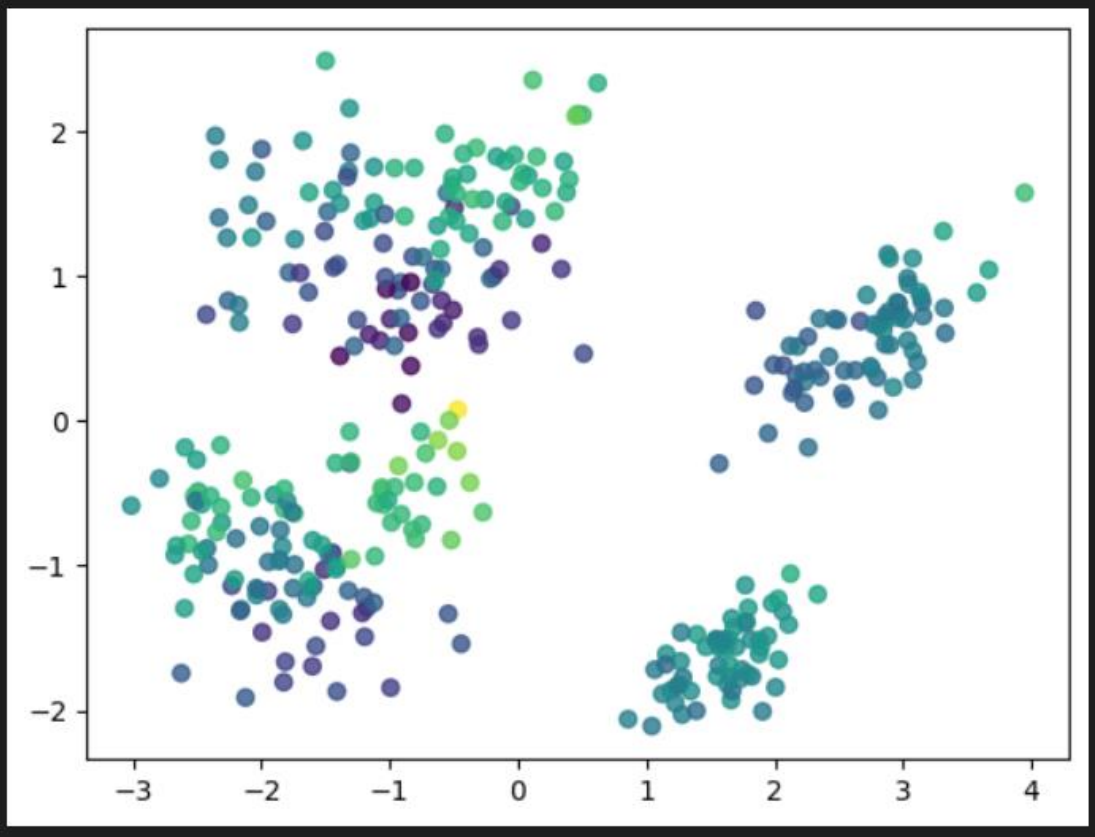
HW1

Keivan Ipchi Hagh - 9831073

سوال یازدهم

```
plt.scatter(
    x = pca_df["pca_1"],
    y = pca_df["pca_2"],
    c = pca_df["pca_3"],
    cmap = "viridis",
    alpha = 0.8,
)
plt.show()
```

[164] ✓ 0.2s



Data Mining

HW1

Keivan Ipchi Hagh - 9831073

سوال دوازدهم

```
fig, ax = plt.subplots(figsize = (10, 5))
df.boxplot(ax = ax, notch = True, bootstrap = 5000)
plt.show()
```

[173] ✓ 2.4s