به نام خداوند جان و خرد

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

یادگیری ماشین

تمرین شماره ۲

نام و نام خانوادگی: علی خرم فر

شماره دانشجویی: ۸۱۰۱۰۲۱۲۹

اردیبهشت‌ماه ۱۴۰۲

فهرست مطالب

_1 پاسخ سوال ۶ ..... ۱

۱-۱_ خروجی تخمین‌ها ۲

۱-۲_ نتیجه گیری ۴

۲_ پاسخ سوال ۷ ..... ۴

2-1_ مصورسازی نمایش داده‌ها ۴

۲-۲_ آموزش طبقه‌بند پارزن ۵

۲-۳_ خروجی طبقه‌بند ۶

خروجی با ۵۰ نمونه و Vn برابر ۱ ..... ۶

خروجی با ۵۰ نمونه و Vn برابر ۰.۱ ..... ۶

خروجی با ۵۰۰ نمونه و Vn برابر ۱ ..... ۶

خروجی با ۵۰۰ نمونه و Vn برابر ۰.۱ ..... ۶

مقایسه خروجی‌ها و نتیجه‌گیری ۷

۳_ پاسخ سوال ۸ ..... ۸

3-1_ EDA ۸

ارتباط بین ویژگی Area و Perimeter ۹

ارتباط بین ویژگی Area و Compactness ۹

ارتباط بین ویژگی Area و Kernel.Length ۱۰

ارتباط بین ویژگی Area و Kernel.Width ۱۰

ارتباط بین ویژگی Area و Asymmetry.Coeff ۱۱

ارتباط بین ویژگی Area و Kernel.Groove ۱۱

ارتباط بین ویژگی Perimeter و Compactness ۱۲

ارتباط بین ویژگی Perimeter و Kernel.Length ۱۲

ارتباط بین ویژگی Perimeter و Kernel.Width ۱۳

ارتباط بین ویژگی Perimeter و Asymmetry.Coeff ۱۳

ارتباط بین ویژگی Perimeter و Kernel.Groove ۱۴

ارتباط بین ویژگی Compactness و Kernel.Length ۱۴

ارتباط بین ویژگی Compactness و Kernel.Width ۱۵

ارتباط بین ویژگی Compactness و Asymmetry.Coeff ۱۵

ارتباط بین ویژگی Compactness و Kernel.Groove ۱۶
ارتباط بین ویژگی Kernel.Length و Kernel.Width ۱۶
ارتباط بین ویژگی Kernel.Length و Asymmetry.Coeff ۱۷
ارتباط بین ویژگی Kernel.Length و Kernel.Groove ۱۷
ارتباط بین ویژگی Kernel.Width و Asymmetry.Coeff ۱۸
ارتباط بین ویژگی Kernel.Width و Kernel.Groove ۱۸
ارتباط بین ویژگی Asymmetry.Coeff و Kernel.Groove ۱۹
هیستوگرام ویژگیArea ۲۰
هیستوگرام ویژگیPerimeter ۲۰
هیستوگرام ویژگیCompactness ۲۱
هیستوگرام ویژگیKernel.Length ۲۱
هیستوگرام ویژگیKernel.Width ۲۲
هیستوگرام ویژگیAsymmetry.Coeff ۲۲
هیستوگرام ویژگیKernel.Groove ۲۳
۲-۳_ پیش‌پردازش و نرمال‌سازی ۲۳
حذف مقادیر NULL ۲۳
نرمال‌سازی به روش min-max ۲۴
نرمال‌سازی به روش z-scoring ۲۴
۳-۳_ رگرسیون لاجستیک ۲۵
تقسیم داده‌ها به داده‌های آموزش و تست ۲۵
تکنیک ONE VS ALL ۲۶
3-4_ طبقه‌بندی با KNN ۲۷
۴_ پاسخ سوال ۹ ۲۸
4-1_ EDA ۲۸
داده‌های از دست رفته ۲۸
نمودارهای ScatterPlot ۲۹
نمودارهای Histogram ۳۰
نمودار BarPlot ۳۲
۴-۲_ همبستگی بین متغیرها ۳۳
۴-۳_ پیش پردازش داده‌ها ۳۴
Handling missing values ۳۵

Split train test ۳۵
۴-۴_ رگرسیون با ۱ ویژگی ۳۶
۴-۵_ رگرسیون با ۳ ویژگی ۳۷

۱_ پاسخ سوال ۶

تابع PDF زیر برای این مسئله ارائه شده است:

$$p(x) = \begin{cases} \frac{1}{2} & for\ 0 < x < 2 \\ 0 & otherwise \end{cases}$$

شکل ۱ تابع PDF ارائه شده

توزیع بالا به صورت پیوسته ارائه شده است. برای اینکه بتوانیم به کمک روش پارزن آن را تخمین بزنیم ابتدا تعداد مناسبی نمونه از آن تولید می‌کنیم. همچنین یک تابع برای تولید نمونه یکنواخت بین ۰ و ۱ برای تخمین پیاده‌سازی می‌کنیم که به تعداد نقاط نمونه تولید کند.

همانطور که میدانیم در روش Parzen حجم V ثابت است و می‌خواهیم ببینیم چند داده در آن می‌افتد. وقتی V کوچک می‌شود، K هم کوچک می‌شود. بجای اینکه K مستقیم محاسبه شود از تابع کرنل استفاده می‌کنیم. این تابع کرنل باید نامنفی بوده و انتگرال زیرش ۱ باشد که در این مسئله از نرمال استاندارد استفاده می‌شود.

سپس کرنل را که قرار است بر روی نمونه‌های PDF تنظیم کرده و تخمین را انجام دهیم به صورت تابع گوسی پیاده‌سازی می‌کنیم.

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi \left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n} \right)$$

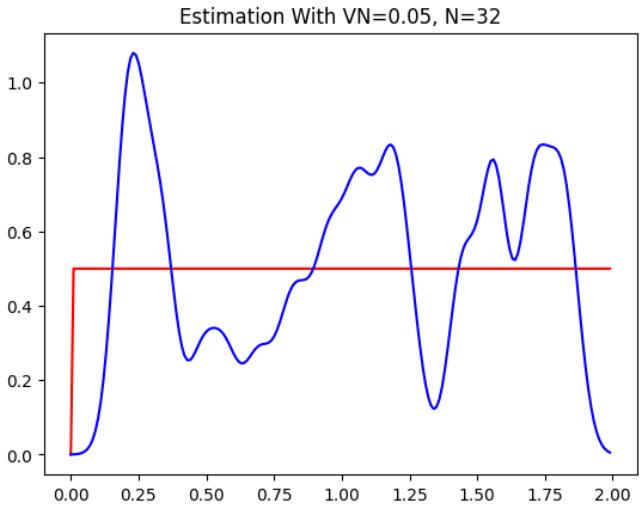
در این فرمول هر چه h بزرگتر باشد ، تعمیم بیشتری خواهیم داشت و لی جزئیات از دست می‌روند. و هرچه h کوچکتر باشد، بیش‌برازش بیشتر می‌شود و قدرت تعمیم را از دست می‌دهیم. در فرمول بالا وقتی n زیاد شود همگرایی به توزیع واقعی را خواهیم داشت.

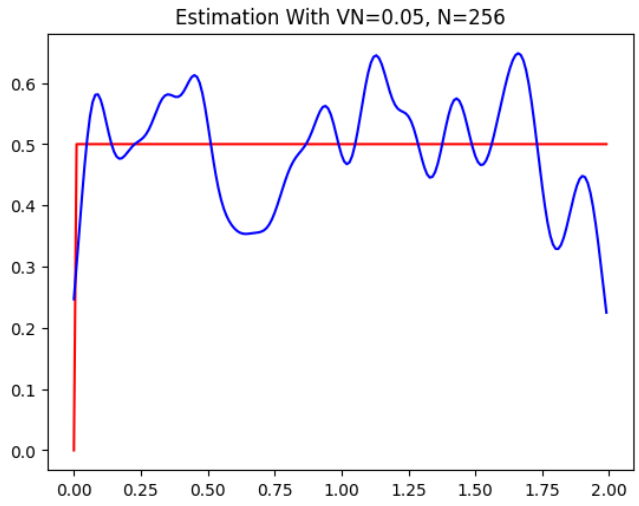
برای بررسی نتیجه نیز یک حلقه تشکیل داده که بر اساس مسئله برای پارامتر پنجره مختلف و تعداد نقاط مختلف نتایج تخمین را در یک لیست ذخیره می‌کند.

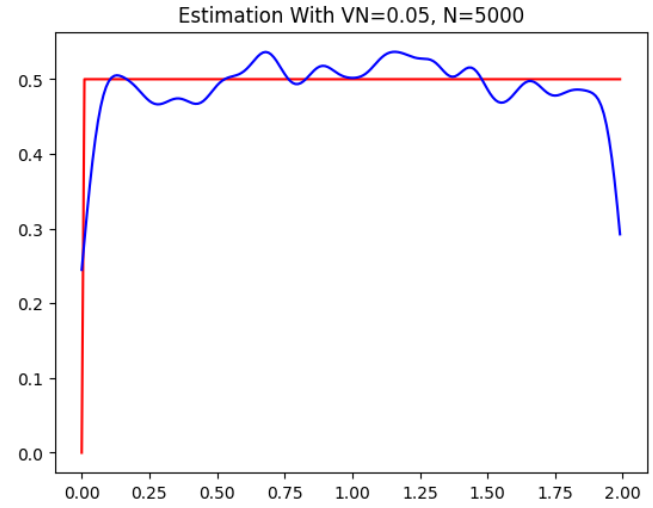
```
n_samples = [32, 256, 5000]
vn_values = [0.05, 0.2]
Results = {}
sample_points = {}
```

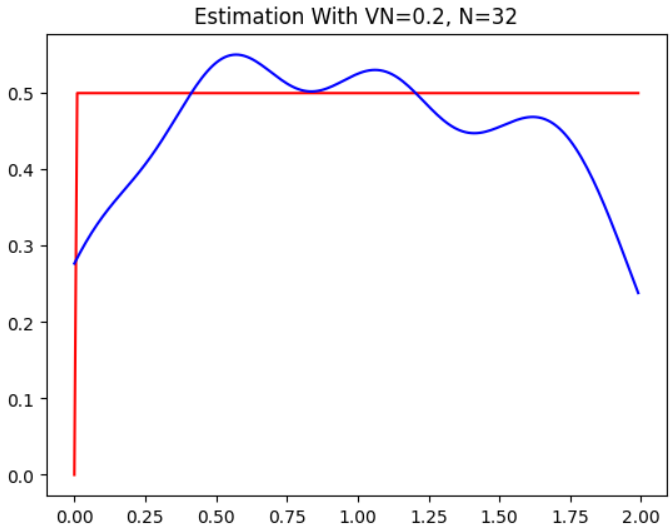
۱-۱_ خروجی تخمین‌ها

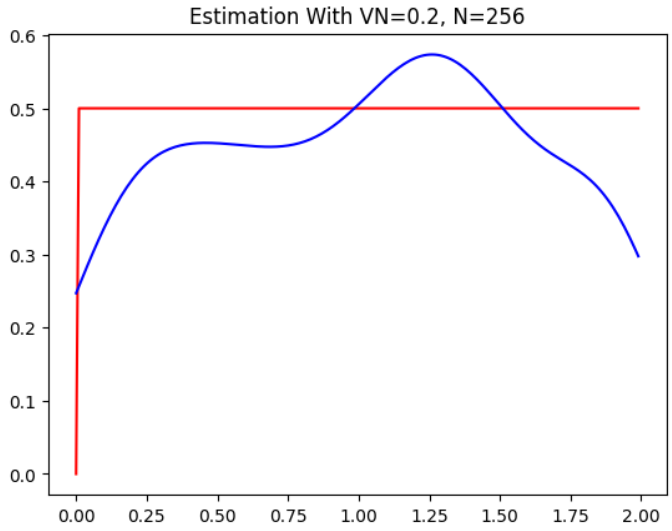
برای بررسی نتیجه تخمین و درک شهودی از تفاوت آن با PDF واقعی ارائه شده، نمودار هرکدام از نتایج را رسم می‌کنیم. نتایج به صورت زیر هستند:

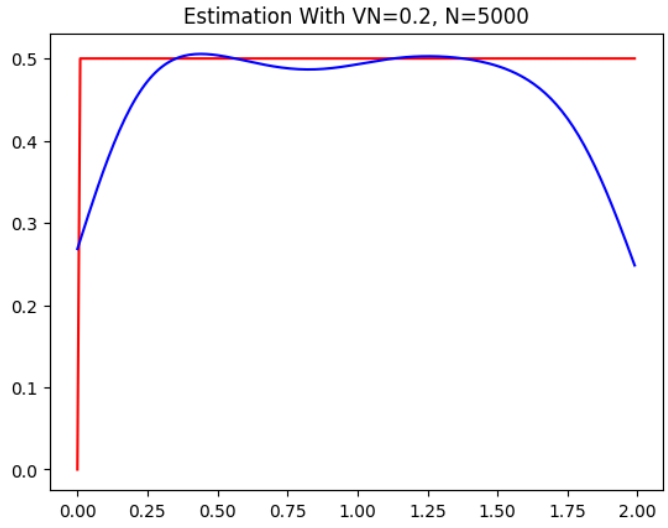





Estimation With VN=0.05, N=5000
0.5
0.4
0.3
0.2
0.1
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Estimation With VN=0.2, N=32
0.5
0.4
0.3
0.2
0.1
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Estimation With VN=0.2, N=256
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00



۲-۱_ نتیجه گیری

۳ نمودار اول تخمین‌های با VN=0.05 به ازای تعداد نقاط ۳۲، ۲۵۶ و ۵۰۰۰ بودند و ۳ نمودار دوم نیز برای VN=0.2 برای نقاط مربوطه هستند.

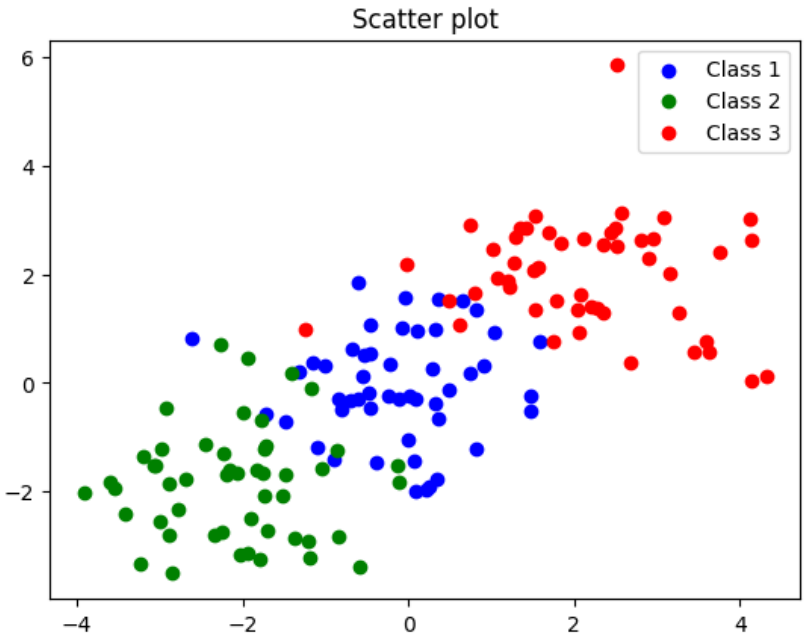
وقتی که تعداد نقاط کم در حد ۳۲ است، خروجی روش پارزن بسیار از PDF واقعی فاصله دارد و تخمین مناسبی ارائه نمی‌دهد. خصوصا وقتی که VN=0.05 است و لبه‌ها و اعوجاج‌های بیشتری مشاهده می‌شود و نتیجه جالب نیست. وقتی که تعداد نقاط را افزایش می‌دهیم و ۲۵۶ می‌رسد تخمین Smooth تر شده و به PDF نزدیک‌تر می‌شود. این مورد با ۵۰۰۰ بهبود چشمگیری پیدا می‌کند. با افزایش VN تخمین باز هم Smoothتر می‌شود ولی این مورد برای وقتی که تعداد نقاط بالاست باعث شده که خطا در برخی نواحی بیش از اندازه شود.

۲_پاسخ سوال ۷

ابتدا برای هر کلاس ۵۰ نمونه داده تولید می‌کنیم. باتوجه به اینکه داده‌ها ۲ بعدی هستند باید نرمال ۲بعدی برای تولید آن‌ها استفاده شود به این منظور از پکیج Numpy تابع multivariate_normal را فراخوانی می‌کنیم.

۲-۱_ مصورسازی نمایش داده‌ها

برای مصورسازی این داده‌ها در فضای دوبعدی از Scatter Plot استفاده می‌کنیم:



۲-۲_ آموزش طبقه‌بند پارزن

در متن سوال به طور دقیق مشخص نشده که از چه تابع کرنلی استفاده شود. باتوجه به این مورد ما از همان تابع کرنل سوال قبل استفاده می‌کنیم که به صورت نرمال استاندارد است.

نحوه طبقه‌بند پیاده‌سازی شده به این صورت است که هربار که یک نقطه جدید داده می‌شود، باتوجه به تخمین انجام شده برای ۳ کلاس قبلی، در آن پنجره پارزن ماکزیمم گرفته می‌شود و لیبلی که به کلاس مورد نظر داده می‌شود برابر با ماکزیمم خواهدبود. در اصل در آن هایپرکیوب خاص، باتوجه به تخمین انجام‌شده، بهترین کلاس انتخاب می‌شود.

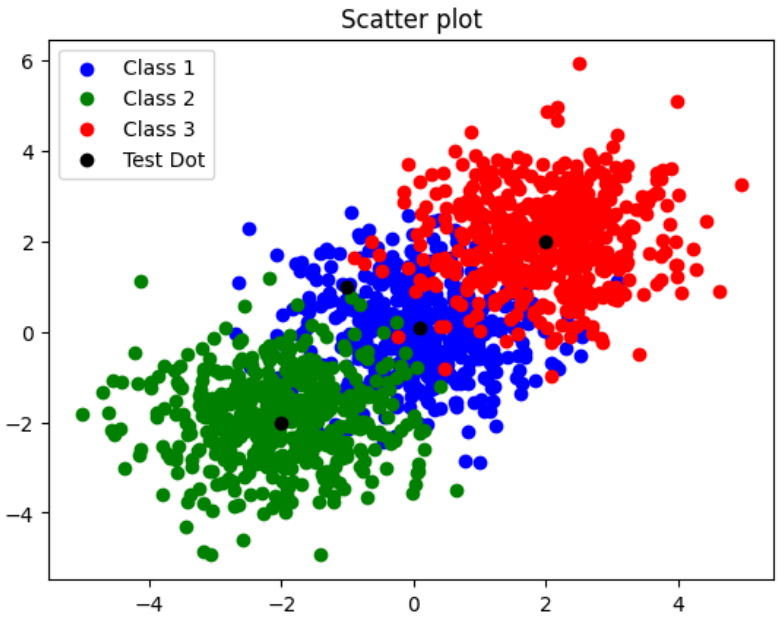
Classification using Parzen window method

A decision making using a classifier based on Parzen window estimation can be performed by simple majority voting method. Here, we check how it works. According to Professor Mireille Boutin [2], we pick the class such that $Prob(w_{i0}|x_0) \geq Prob(w_i|x_0) \forall i = 1, ..., c$ from Bayes' rule. In other words,

$$\begin{aligned} &\iff \rho(x_0|w_{i0})Prob(w_{i0} \geq \rho(x_0|w_i)Prob(w_i) \\ &\iff \rho(x_0, w_{i0}) \geq \rho(x_0, w_i) \\ &\iff \underbrace{\sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}_l - \mathbf{x}_0}{h_n}\right)}_{(\mathbf{x}_l \text{ in class } w_{i0})} \geq \underbrace{\sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}_l - \mathbf{x}_0}{h_n}\right)}_{(\mathbf{x}_l \text{ in class } w_i)}. \end{aligned}$$

۳-۲_ خروجی طبقه‌بند

برای این مسئله نقاط زیر در نظر گرفته شد که به رنگ مشکی در نمودار زیر مشخص هستند:



سعی شده که نقاطی باشند که طبقه‌بند به خوبی بررسی شود.

```
۱ [0.1, 0.1],
۲ [-2.0, -2.0],
۳ [2.0, 2.0],
۴ [-1.0, 1.0]
```

خروجی با ۵۰ نمونه و Vn برابر ۱

```
{1.0: [0, 1, 2, 0]}
```

خروجی با ۵۰ نمونه و Vn برابر ۰.۱

```
{0.1: [0, 1, 2, 2]}
```

خروجی با ۵۰۰ نمونه و Vn برابر ۱

```
{1.0: [0, 1, 2, 0]}
```

خروجی با ۵۰۰ نمونه و Vn برابر ۰.۱

{0.1: [0, 1, 2, 0]}

مقایسه خروجی‌ها و نتیجه‌گیری

طبقه‌بندی انجام شده برای نقاط اول تا سوم به خوبی انجام شده است. این نقاط را به نحوی انتخاب کرده بودیم که هرکدام در مرکز و میانگین توزیع هر کلاس باشند تا متوجه شویم طبقه‌بند به خوبی نقاط عمومی را تشخیص می‌دهد و Underfit نداریم.

نقطه‌ای که دارای چالش است نقطه چهارم است که در مرز بین ۳ کلاس قرار دارد و حالتی است که در کنار آن دو نقطه از کلاس ۱ وجود دارند. این مورد باعث شده زمانی که Vn برابر ۰.۱ باشد کلاس به اشتباه ۲ تشخیص داده شود. پس در اینجا با افزایش Vn دقت مدل افزایش یافته است.

زمانی که از ۵۰۰ نمونه استفاده کنیم، این مشکل پیش نیامد و با همان Vn کوچک، تخمین به درستی زده شد و دقت طبقه‌بند در این نقطه به خوبی سنجیده شد و توانست نقطه را هم با Vn کوچک و هم Vn بزرگ به خوبی تشخیص دهد.

این نتایج مربوط به اجرای بار اول نوت‌بوک بود. چندین بار دیگر هم اجرا شد و می‌توانیم موارد زیر را نتیجه بگیریم:

اندازه نمونه‌ها : برای این مسئله به خصوص، با ۵۰ نمونه طبقه‌بندی به خوبی انجام می‌شود و لزوما نیاز نیست در هر مسئله‌ای نمونه‌ها را به مقدار زیادی اضافه کنیم. تخمینگر پارزن در ۵۰ نمونه توانست نقاط را به خوبی طبقه‌بندی کند و کافی بود Vn به خوبی انتخاب شود و نیاز نیست محاسبات را بسیار زیاد کنیم. بدیهی است افزایش نمونه‌ها به افزایش دقت تخمین منجر می‌شود ولی باید این را در نظر بگیریم که با افزایش ابعاد افزایش داده‌های تست پیچیدگی محاسبات افزایش خواهدیافت. باید Tradeoff کنیم.

اندازه Vn : باتوجه به اجراهای متفاوتی که بررسی شد متوجه شدیم که اندازه Vn بستگی به نقاط تست دارند. اگر نقاط تست به دور از مرز بین کلاس‌ها باشند با تنظیم مقدار کوچک به خوبی جواب می‌دهد. این نقاط در نواحی که چگالی یک کلاس بالاست، با Vn کوچک به جزئیات بیشتری متمرکز می‌شود. این مورد باعث می‌شود که نویز حساس باشد. همچنین اگر تعداد داده‌ها کم باشد این مورد حساس‌تر می‌شود. در این سوال کلاس‌ها به صورت واضحی از هم تفکیک بودند و نویز خاصی وجود نداشت هردو مقادیر کم و زیاد این پارامتر طبقه‌بندی را به خوبی انجام می‌داد. فقط در چند اجرا در نقاطی که نویز در کنار نقاط تست بود مشکل مذکور مشاهده شد.

۳_ پاسخ سوال ۸

ابتدا فایل دیتاست مربوطه را در گوگل درایو بارگذاری می‌کنیم. از پکیج pandas برای مدیریت داده‌ها استفاده کرده و ۵ سطر اول را خروجی می‌گیریم:

```
data = pd.read_csv('/content/gdrive/MyDrive/Colab Notebooks/ML HW2/seeds.csv')

data.head()
```

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove	Type
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	1
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1

همانطور که مشخص است در این دیتاست ویژگی‌های مربوط به ۳ نوع گندم آورده شده و در ستون Typeنوع دانه گندم مشخص شده است. پس به طور کلی ۷ ویژگی در این دیتاست برای هر نوع دانه گندم ارائه شده است.

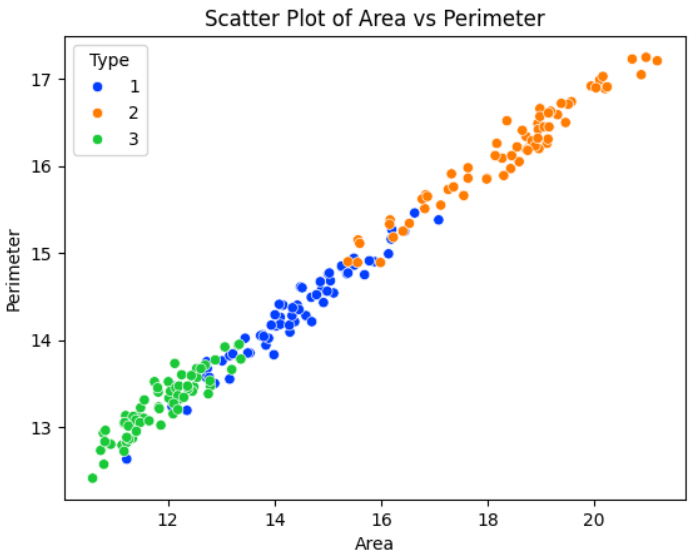
EDA _۳-۱

در این قسمت از مسئله قصد داریم که نمودارهای scatter plot را برای هر دو ویژگی ممکن رسم کنیم و ارتباط بین ویژگی‌ها را در این نمودارها بررسی و تحلیل کنیم. برای پیاده‌سازی این قسمت از مسئله می‌توانیم در یک حلقه هر جفت ویژگی را وارد کرده و نمودار آن را رسم کنیم. باید دقت کنیم که در این حلقه ستون اول بررسی نشود و هر کدام از ستون‌ها با ستونی غیر از خودش بررسی شود. همچنین نیازی به رسم نمودارهای تکراری نیست. نتایج خروجی به صورت زیر هستند. به طور کلی باتوجه به اینکه ۷ ویژگی داریم، ۲۱ ترکیب ممکن برای جفت ویژگی ممکن است:

به طور کلی نمودارهای از نوع Scatterplot همبستگی بین دو داده را به خوبی نشان می‌دهد و اگر که بتوانیم و ویژگی پیدا کنیم توانسته باشند در فضای ویژگی کلاس‌ها را به خوبی از هم تفکیک کنند میتوانیم از آن دو ویژگی برای مسئله طبقه‌بندی استفاده کنیم. به دلیل اینکه فضای ویژگی در این سوال ۲ بعدی است این نمودار درک شهودی نسبتا خوبی به ما خواهد داد.

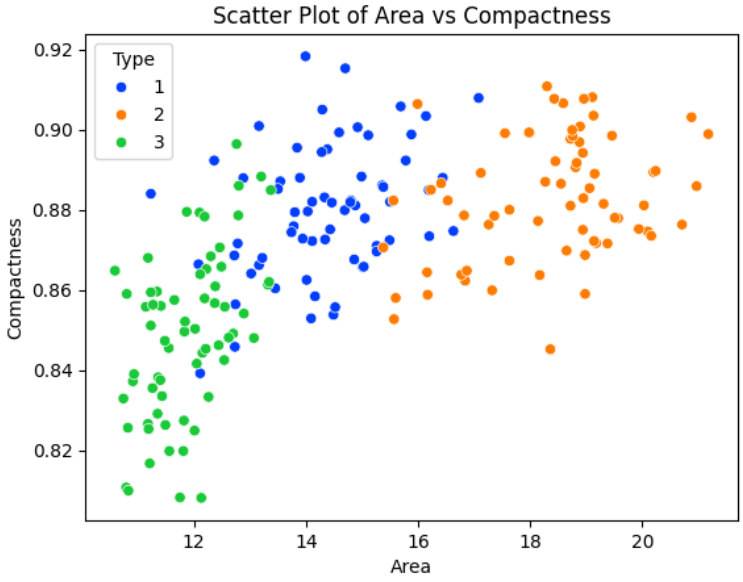
در ادامه به بررسی تک تک جفت ویژگی‌های ممکن خواهیم پرداخت و علاوه بر رسم نمودار Satterplot آن‌ها بررسی می‌کنیم که آیا تفکیک خوبی نسبت به کلاس‌ها صورت می‌پذیرد یا خیر. با کمک کتابخانه matplotlib نمودارها را رسم کرده و نقاط داده مربوط به هر نوع گندم را به یک رنگ اختصاص می‌دهیم تا قابل تشخیص باشند.

ارتباط بین ویژگی Area و Perimeter



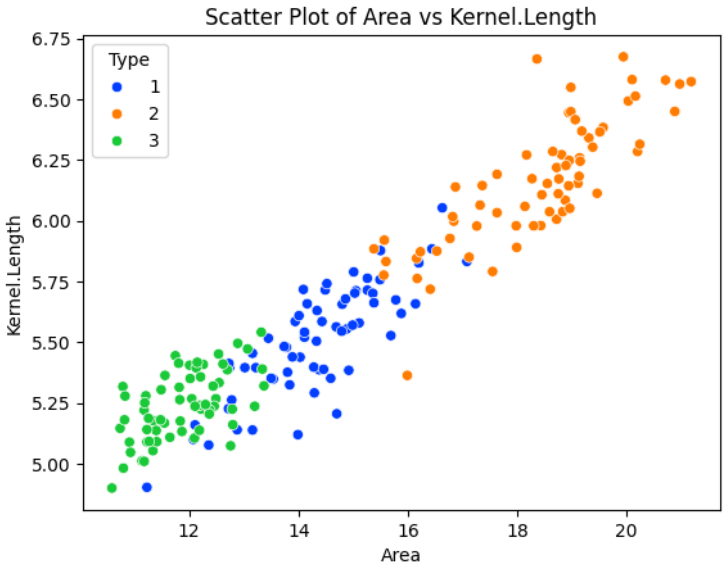
همانطور که در شکل بالا مشخص است این دو ویژگی به طور نسبی توانسته کلاس‌ها را به خوبی از هم تفکیک کنند. هرچند که در برخی نقاط نوع ۱ با نوع ۲ و۳ همپوشانی دارند و نویز داریم. به هر حال باید بقیه نمودار ها هم بررسی شوند تا بهترین ویژگی‌های ممکن برای طبقه‌بندی انتخاب شوند.

ارتباط بین ویژگی Area و Compactness



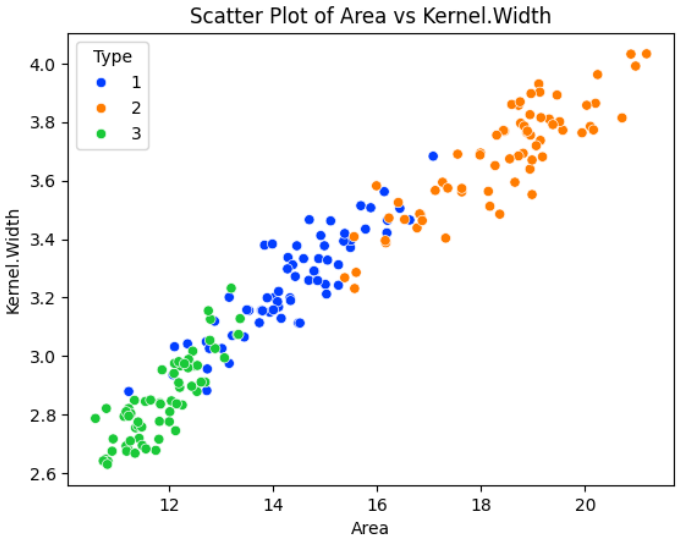
همانطور که در شکل بالا مشاهده می‌شود ترکیب این جفت ویژگی نتوانسته تفکیک خوبی نسبت به نمودار اولی که بررسی کردیم ارائه دهد و ایجاد یک طبقه‌بند برای حل این مسئله با کمک این ویژگی‌ها باعث می‌شود که خطای زیادی داشته باشیم زیرا که داده‌های نویز زیاد هستند. همچنین همبستگی خوبی بین داده‌های کلاس ۲ و ۱ وجود ندارد.

ارتباط بین ویژگی Area و Kernel.Length



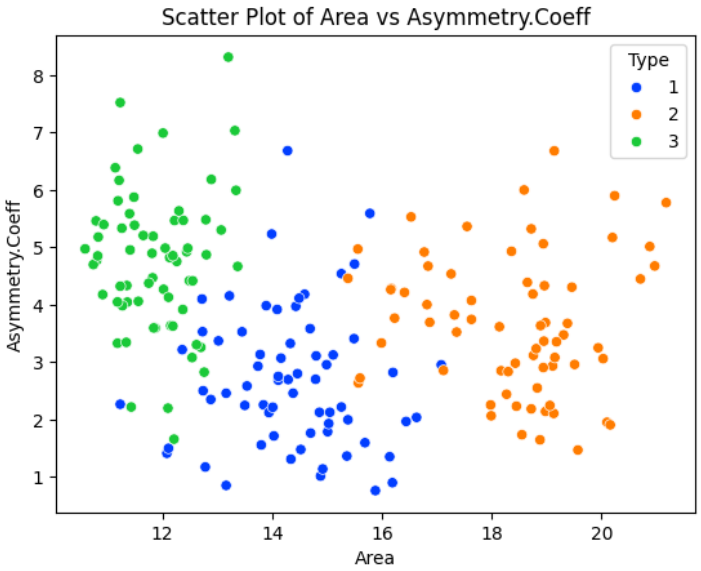
با بررسی نمودار بالا متوجه می‌شویم این دو ویژگی با هم همبستگی دارند. ولی برای تفکیک کلاس‌ها و تعیین مرز بین دو کلاس ۳ و ۲ مشکل خواهیم داشت زیرا که این دو کلاس در فضای این دو ویژگی با هم همپوشانی دارند.

ارتباط بین ویژگی Area و Kernel.Width



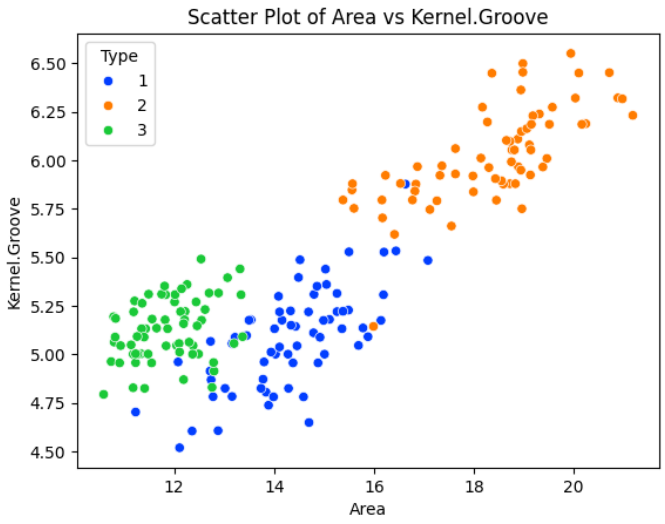
این دو ویژگی نیز با هم همبستگی دارند. ولی برای تفکیک کلاس‌ها و تعیین مرز بین دو کلاس ۳ و ۲ مشکل خواهیم داشت زیرا که این دو کلاس در فضای این دو ویژگی با هم همپوشانی دارند و برخی نقاط از نوع آبی در میانه نقاط سبز هستند.

ارتباط بین ویژگی Area و Asymmetry.Coeff



در فضای ویژگی این جفت ویژگی داده‌ها با هم همبستگی ندارند. ولی اگر هدف مسئله فقط طبقه‌بندی باشد یک تفکیک قابل قبولی بین داده‌های از کلاس‌های مختلف وجود دارد و این مورد کمک می‌کند که طبقه‌بندی به نسبت خوبی انجام شود. هرچند که این موردنیز بستگی به نوع طبقه‌بند دارد.

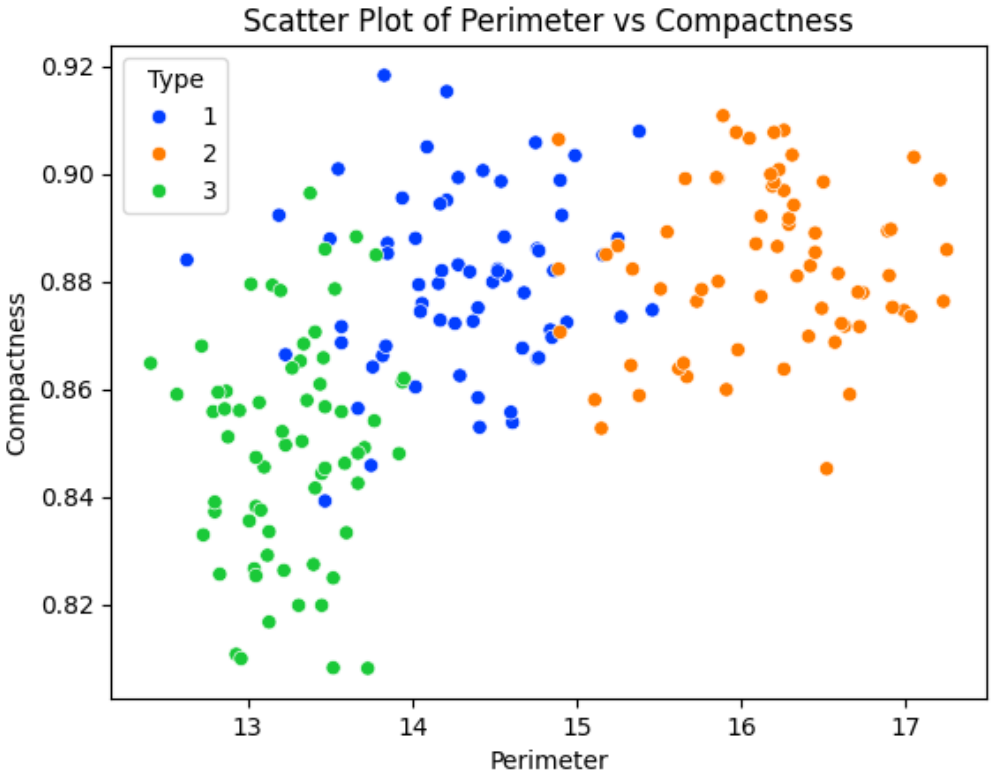
ارتباط بین ویژگی Area و Kernel.Groove



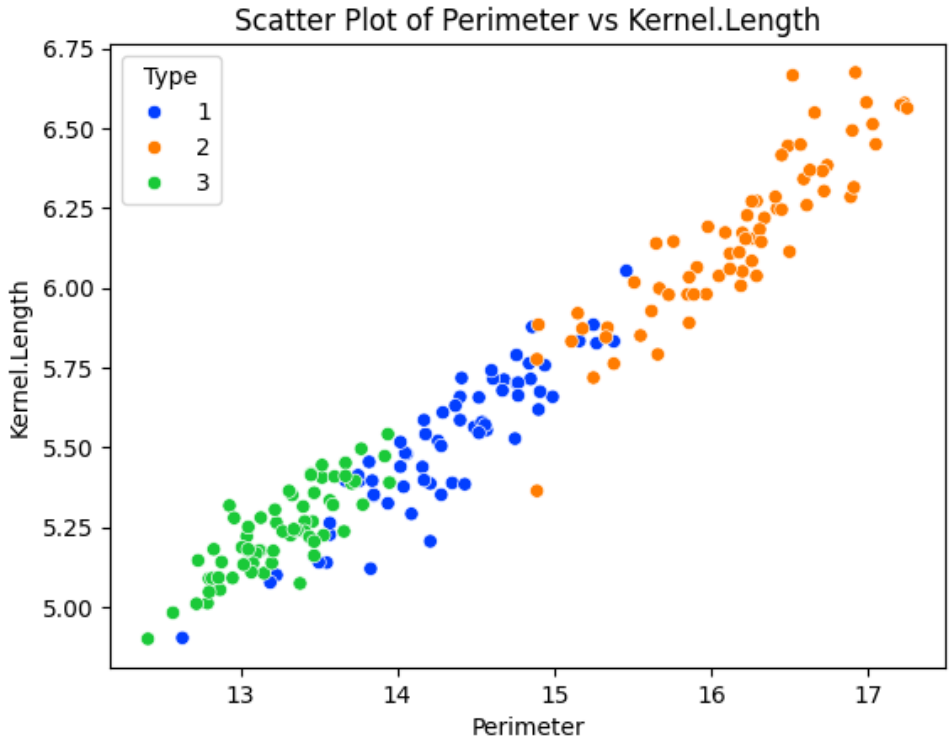
بین داده‌های دسته‌های ۲ و ۱ در این فضای ویژگی همبستگی وجود دارد. برخی داده‌های نویز از طبقه ۱ به رنگ آبی در اطراف نقاط دیگر ویژگی‌ها وجود دارند که این مورد باعث می‌شود در طبقه‌بندی دچار چالش شویم.

بقیه نمودارها هم بسیار مشابه این چند نمودار بررسی شده هستند و تحلیل مشابهی دارند.

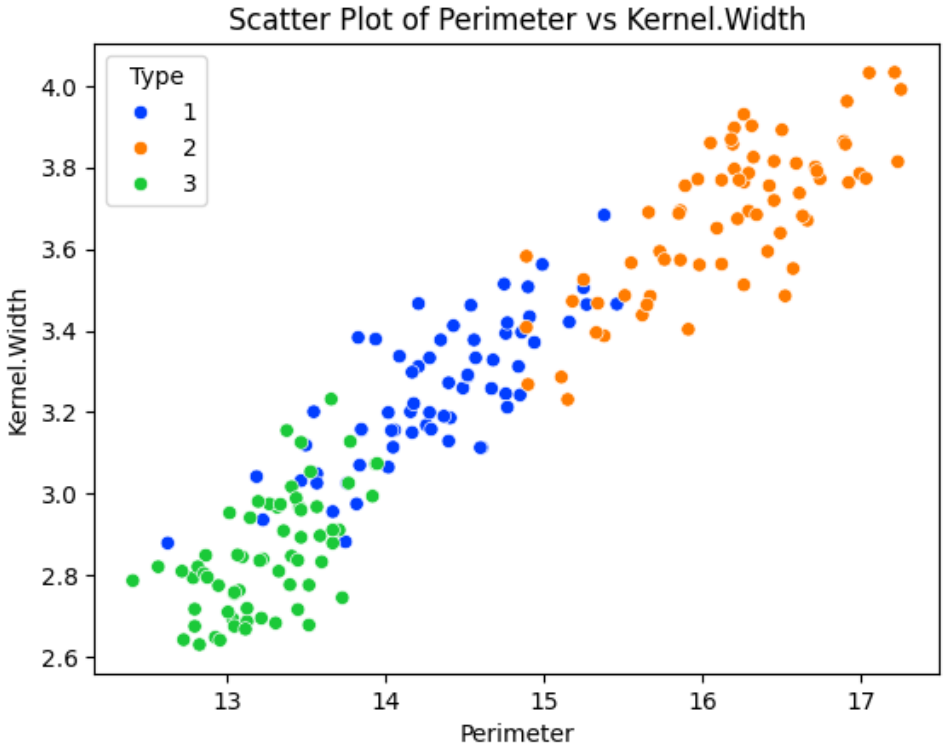
ارتباط بین ویژگی Perimeter و Compactness



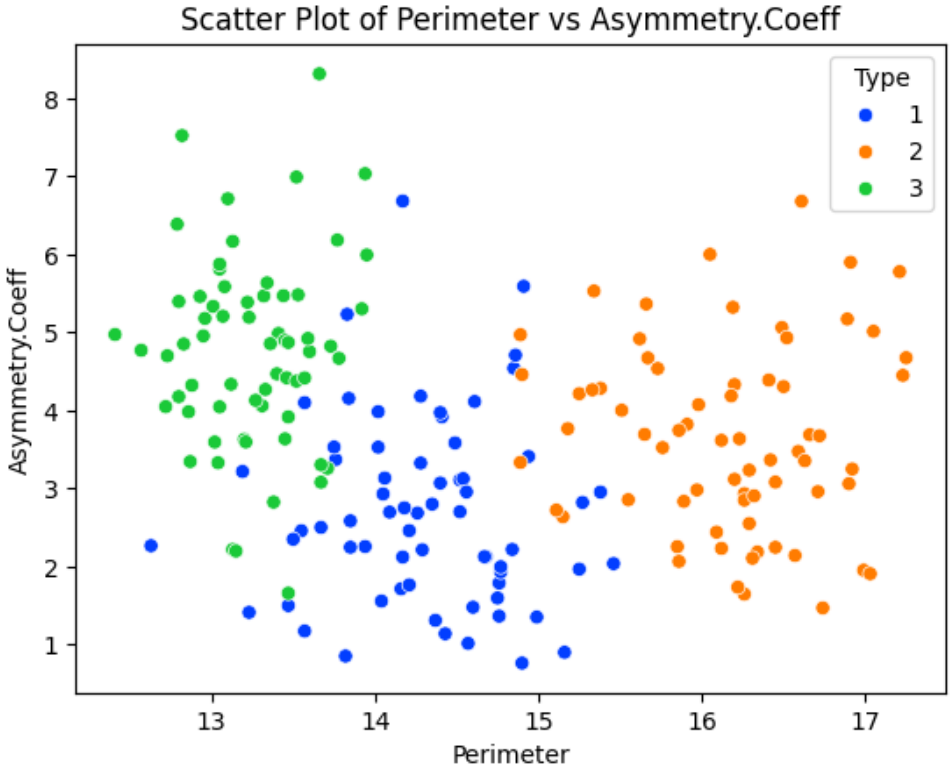
ارتباط بین ویژگی Perimeter و Kernel.Length



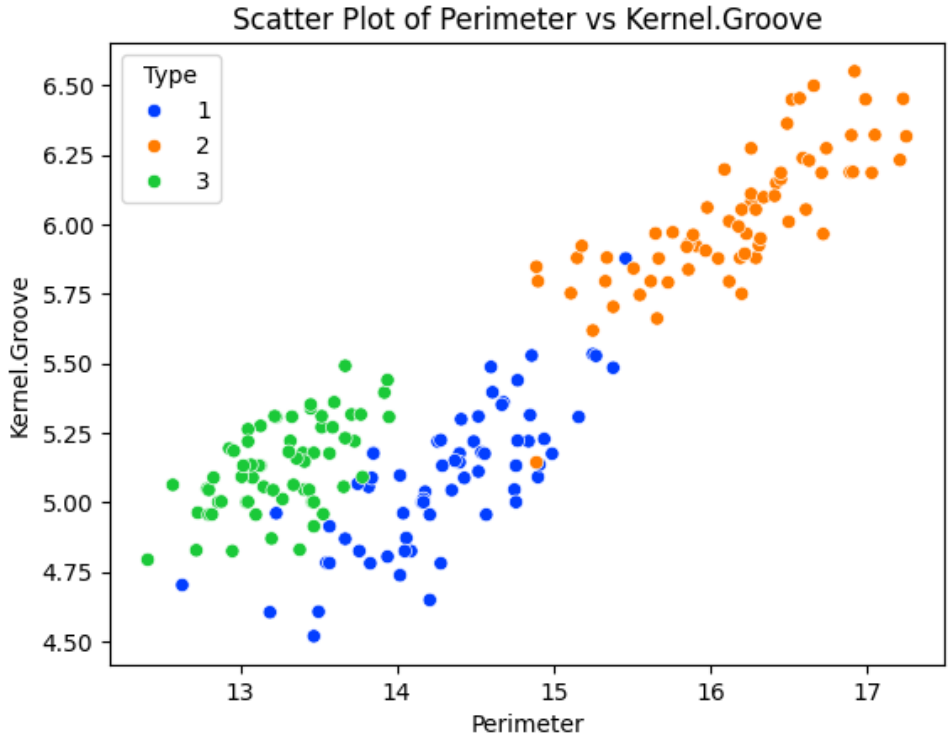
ارتباط بین ویژگی Perimeter و Kernel.Width



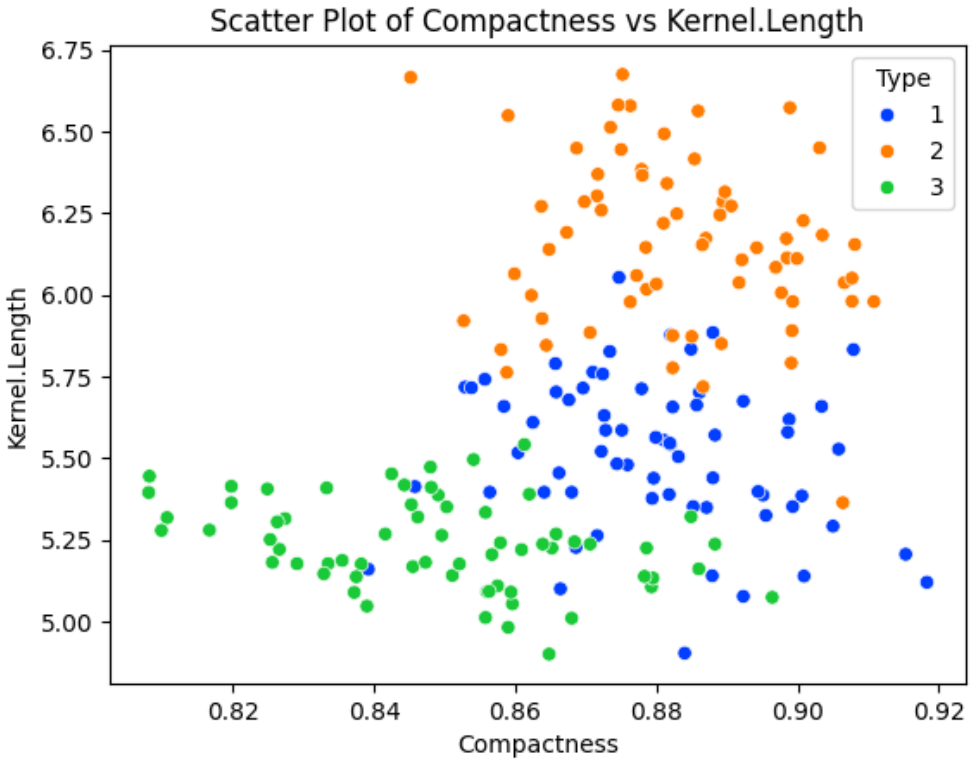
ارتباط بین ویژگی Perimeter و Asymmetry.Coeff



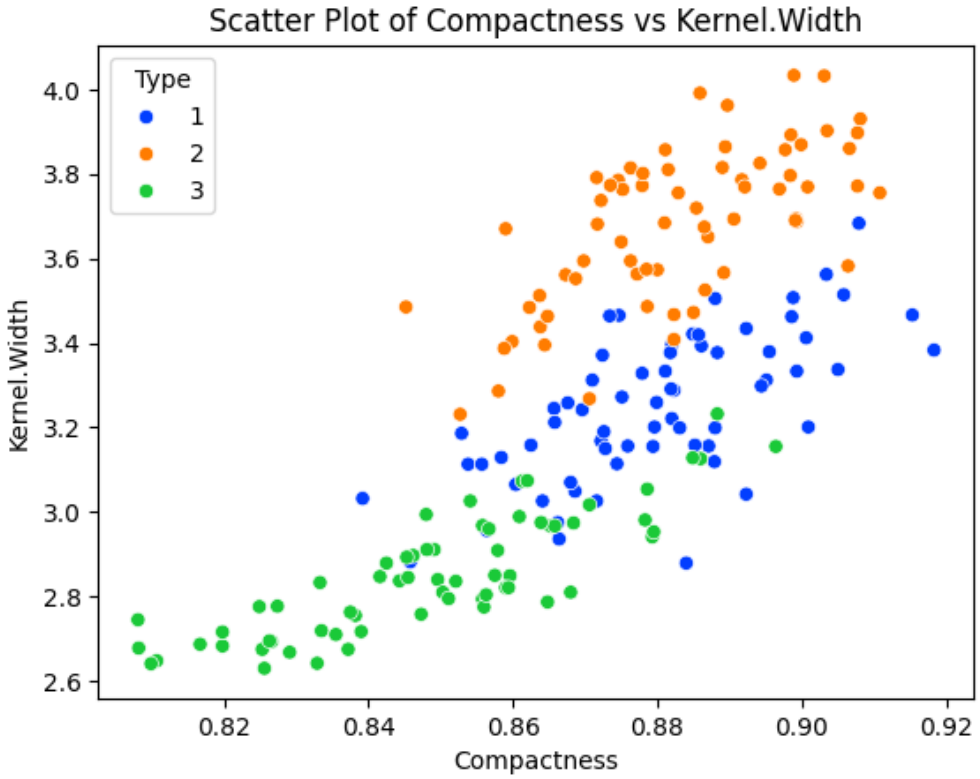
ارتباط بین ویژگی Perimeter و Kernel.Groove



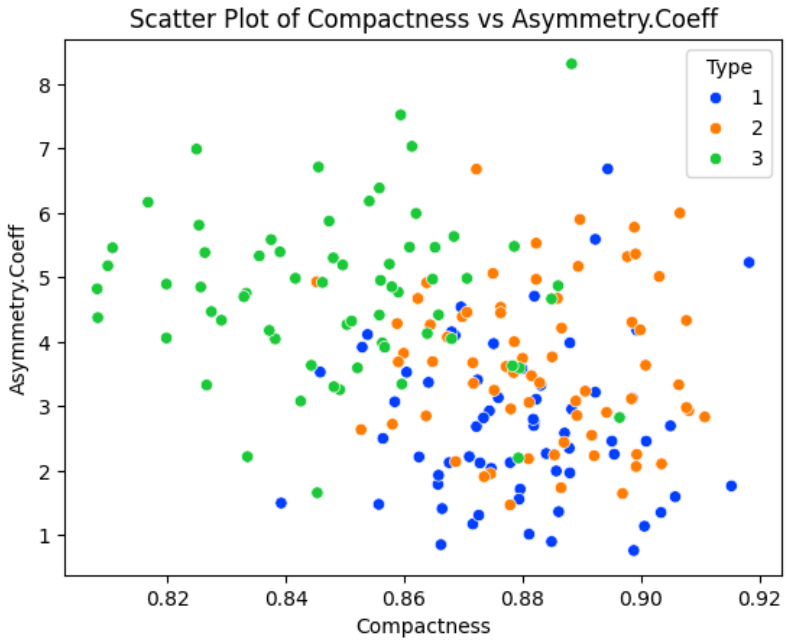
ارتباط بین ویژگی Compactness و Kernel.Length



ارتباط بین ویژگی Compactness و Kernel.Width

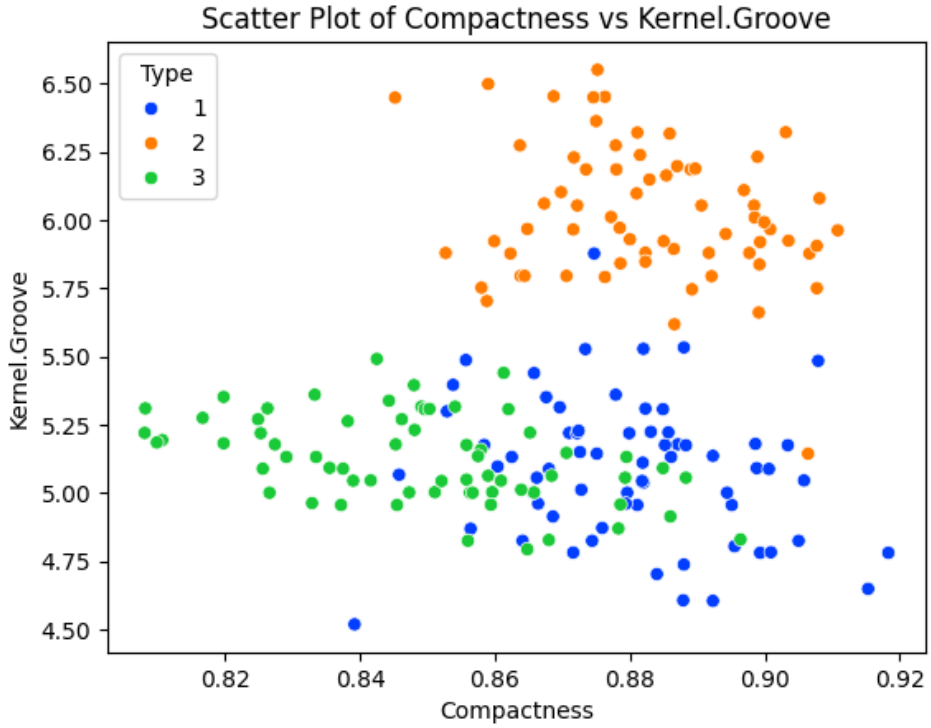


ارتباط بین ویژگی Compactness و Asymmetry.Coeff

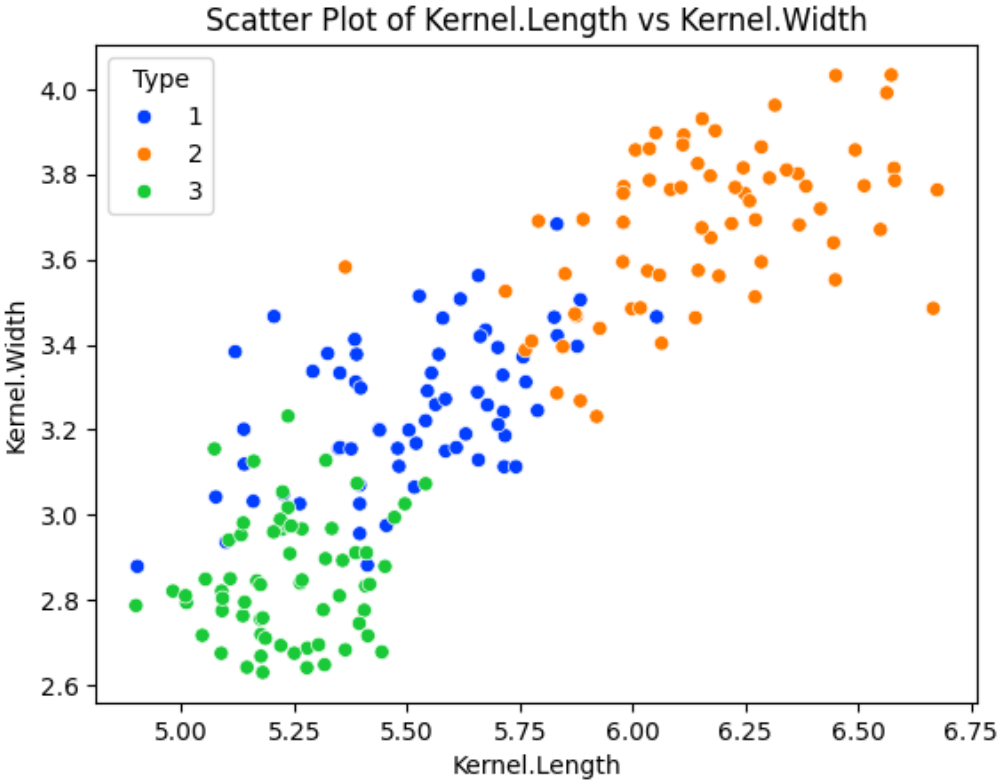


این نمودار تحلیل متفاوتی داشت. باتوجه به اینکه این جفت ویژگی همبستگی ندارند و هیچگونه تفکیکی بین کلاس‌ها وجود ندارد استفاده از آن‌ها برای طبقه‌بندی اشتباه است.

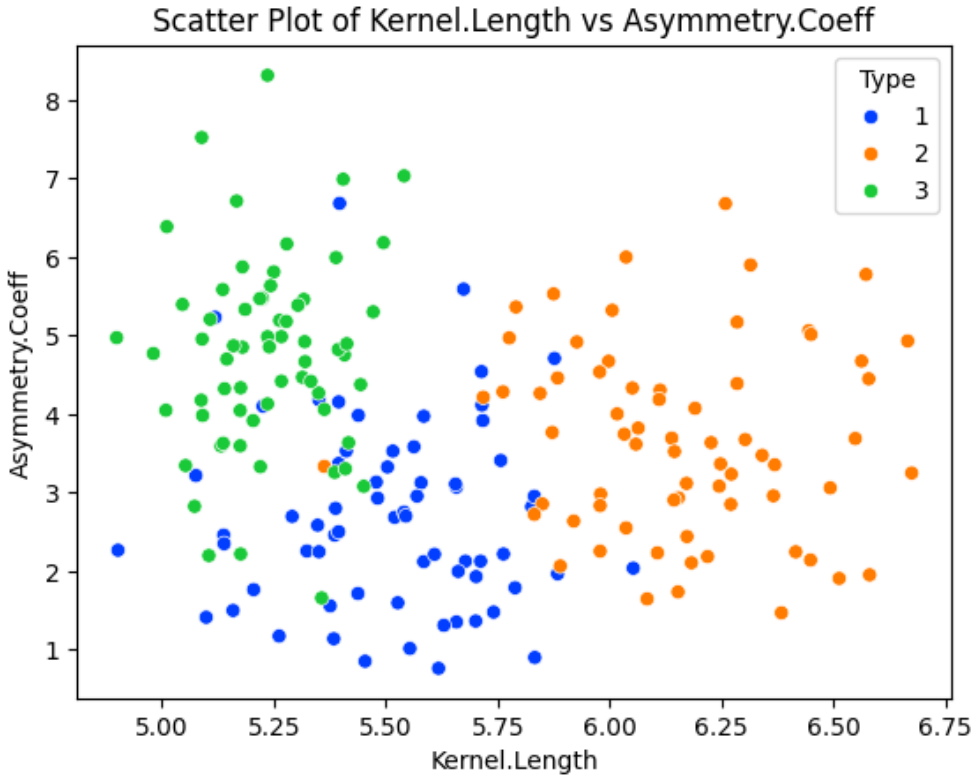
ارتباط بین ویژگی Compactness و Kernel.Groove



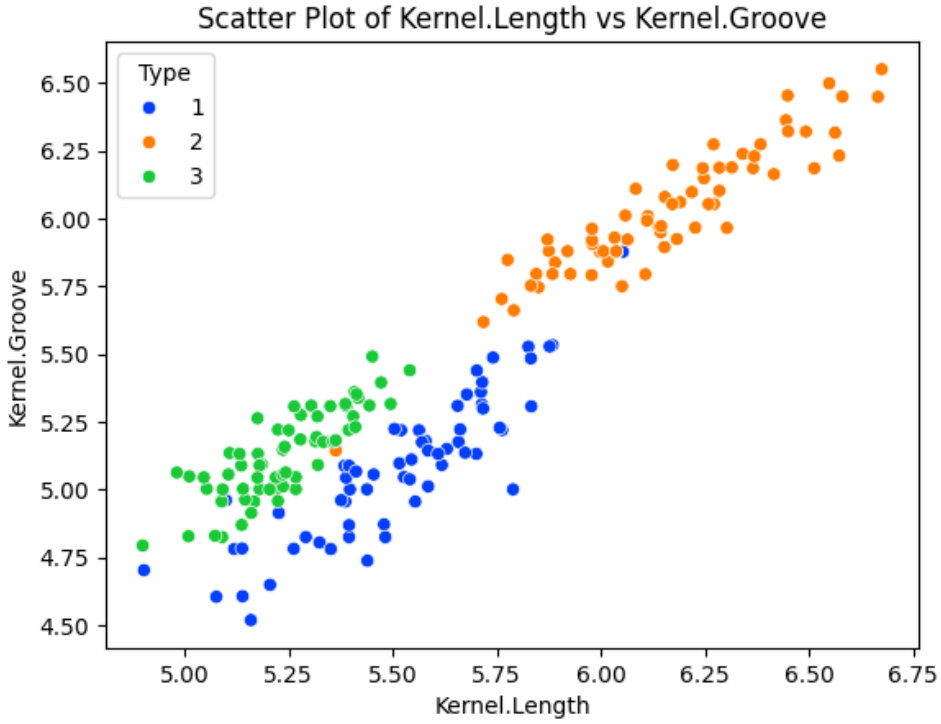
ارتباط بین ویژگی Kernel.Length و Kernel.Width



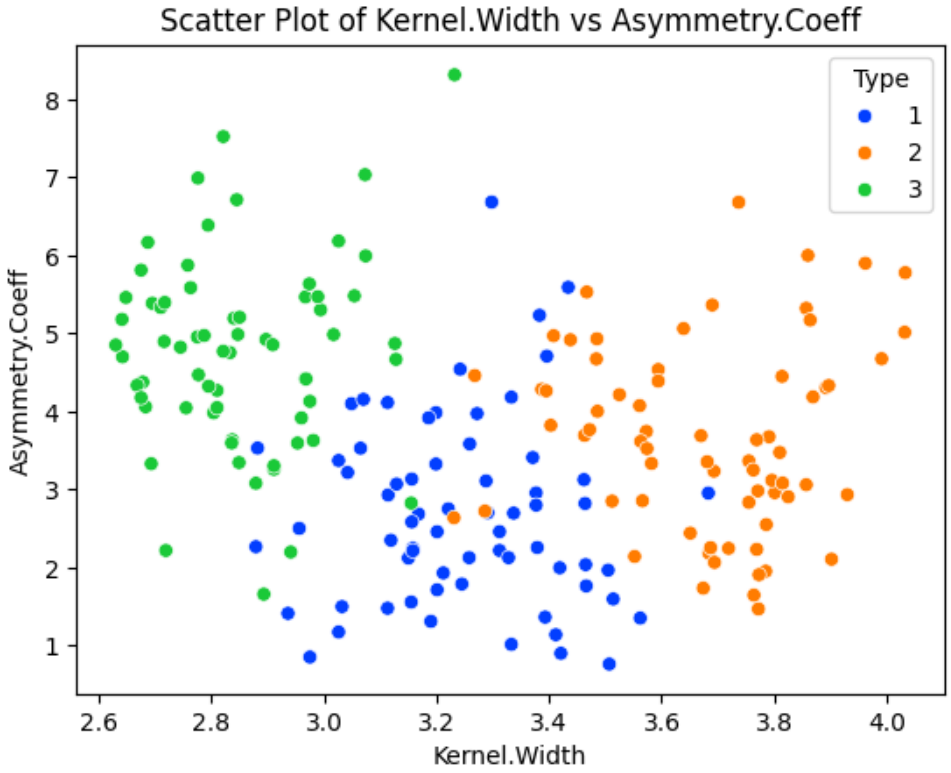
ارتباط بین ویژگی Kernel.Length و Asymmetry.Coeff



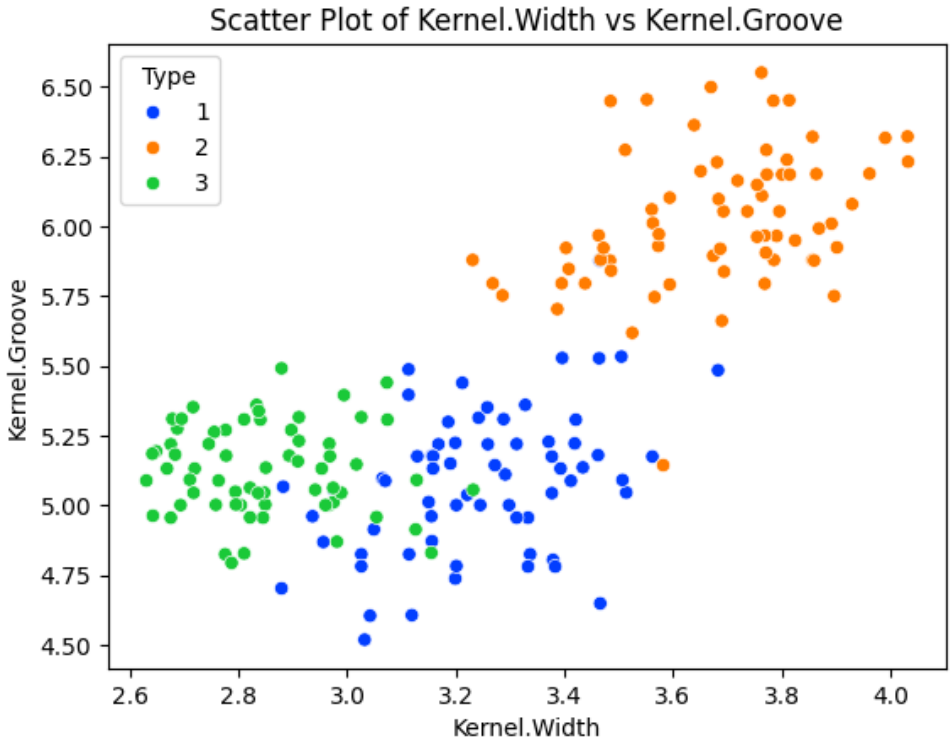
ارتباط بین ویژگی Kernel.Length و Kernel.Groove



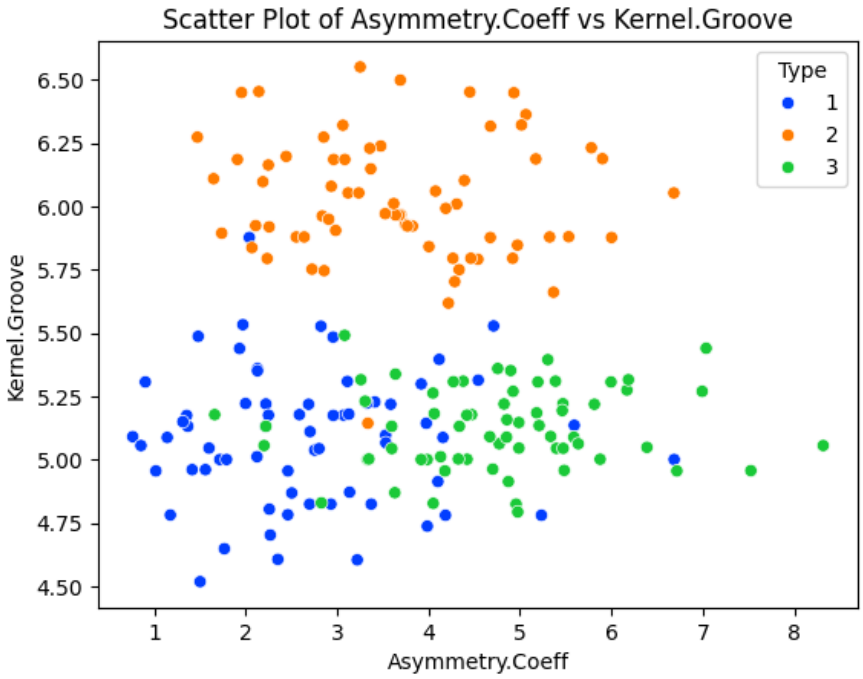
ارتباط بین ویژگی Kernel.Width و Asymmetry.Coeff



ارتباط بین ویژگی Kernel.Width و Kernel.Groove



ارتباط بین ویژگی Asymmetry.Coeff و Kernel.Groove



باتوجه به نمودارهای بررسی شده و اینکه هر نوع از شرایطی که ممکن است در نمودارها پدیدار شود را بررسی کردیم به نظر می‌رسد جفت ویژگی‌های زیر برای یک طبقه‌بندی مناسب باشند:

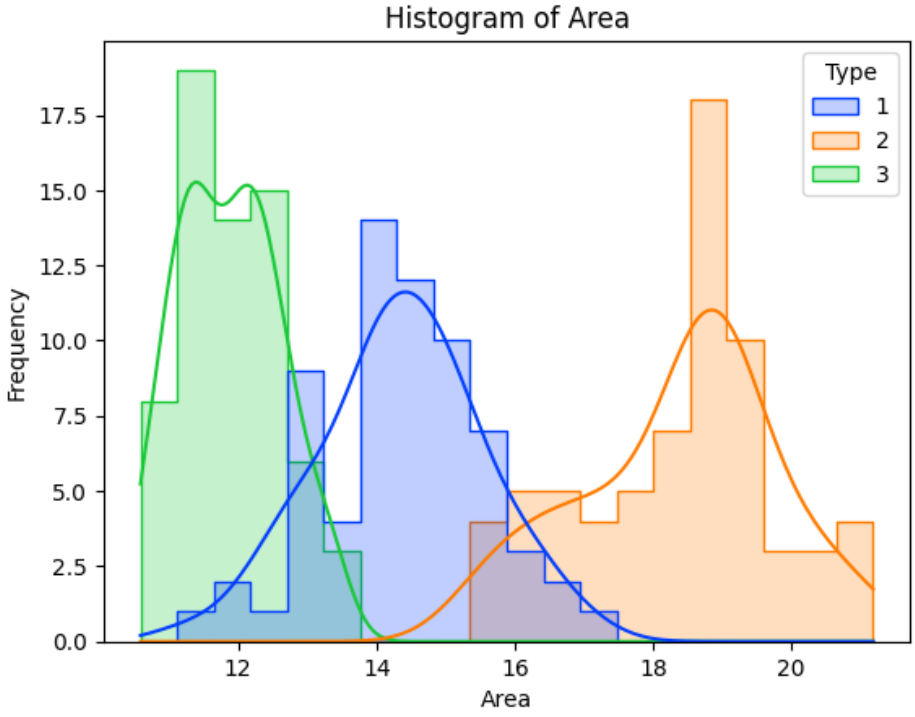
ارتباط بین ویژگی Area و Perimeter

ارتباط بین ویژگی Area و Kernel.Length

ارتباط بین ویژگی Kernel.Length و Kernel.Groove

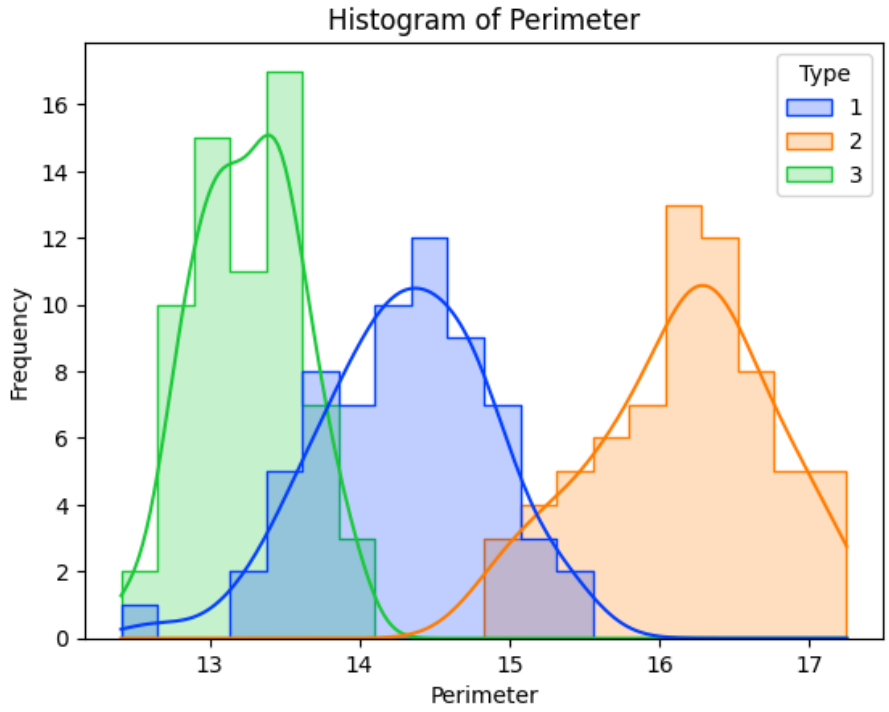
در مرحله بعدی نمودار هیستوگرام هر ویژگی را بر اساس نوع دانه گندم رسم می‌کنیم. بررسی هیستوگرام داده‌ها به ما کمک می‌کند که توزیع داده‌ها را بررسی کرده و پراکندگی آن‌ها را متوجه شویم. برای بررسی دقیق‌تر برای هر کلاس ۳ نمودار هیستوگرام را با ۳ رنگ متفاوت برای هر کلاس رسم می‌کنیم تا به طور دقیق تفاوت بین آن‌ها مشهود باشد:

هیستوگرام ویژگی Area

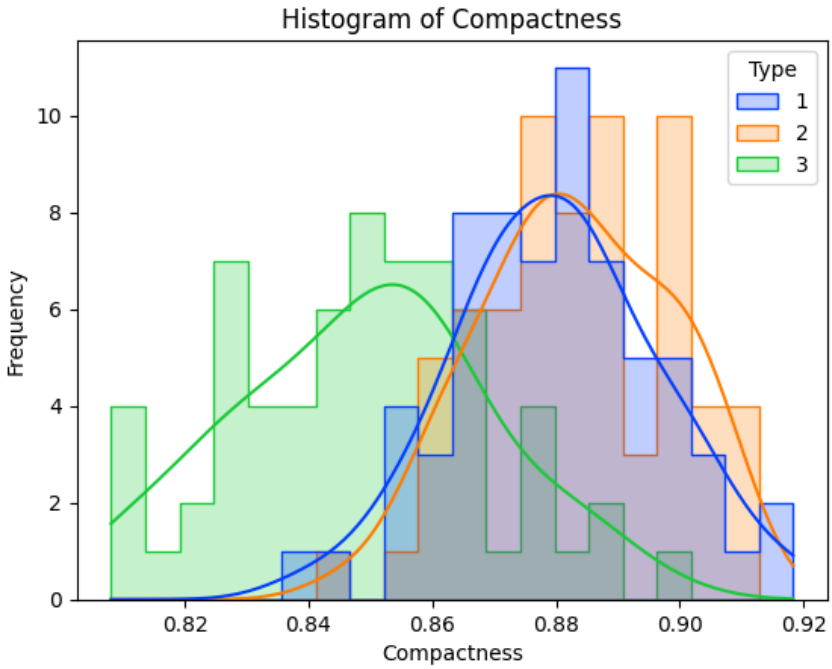


نمودار بالا نشان می‌دهد که ویژگی مورد نظر برای هر دسته توزیع نسبتا مشابهی دارد و از طرفی مقادیر ویژگی در هر کلاس متفاوت هستند هرچند که در برخی مقادیر همپوشانی دارند.

هیستوگرام ویژگی Perimeter

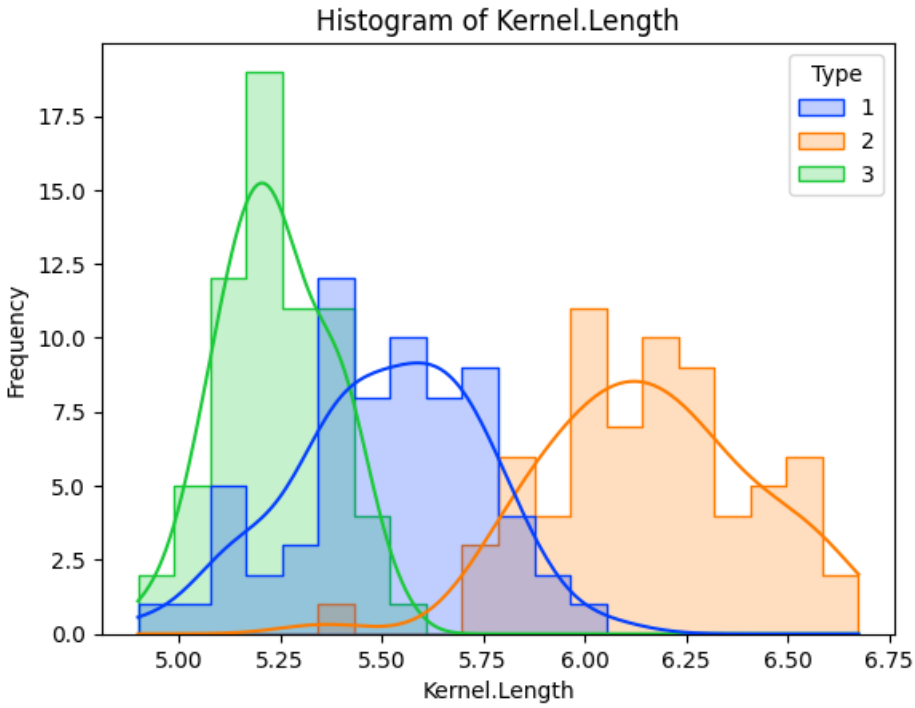


هیستوگرام ویژگی Compactness

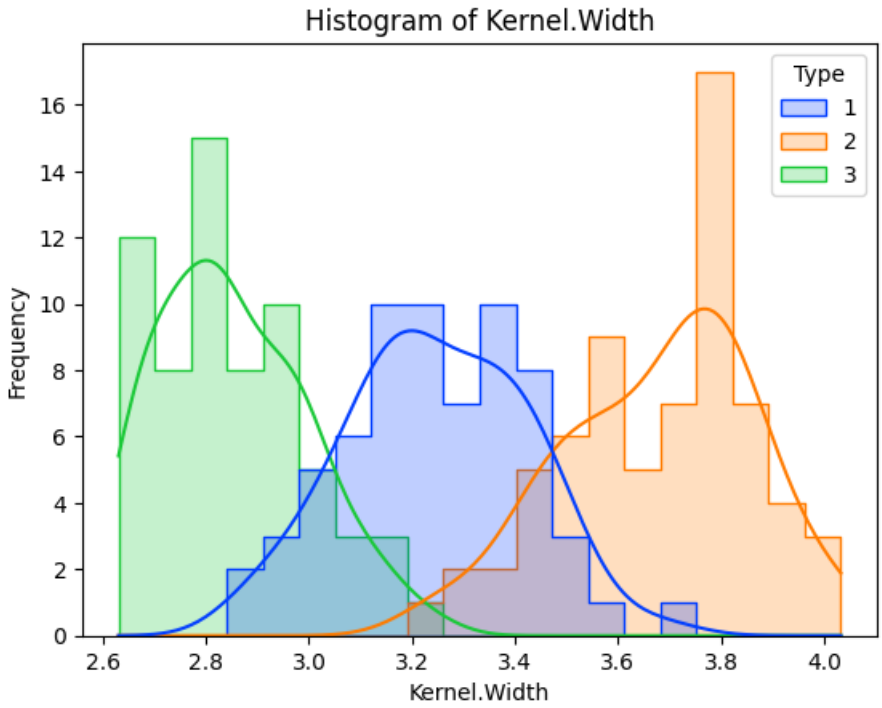


نمودار هیستوگرام بالا نشان می‌دهد که توزیع این ویژگی برای همه کلاس‌ها نسبتا مشابه بوده و همپ.شانی نسبتا زیادی با هم دارند و از هم مجزا نیستند.

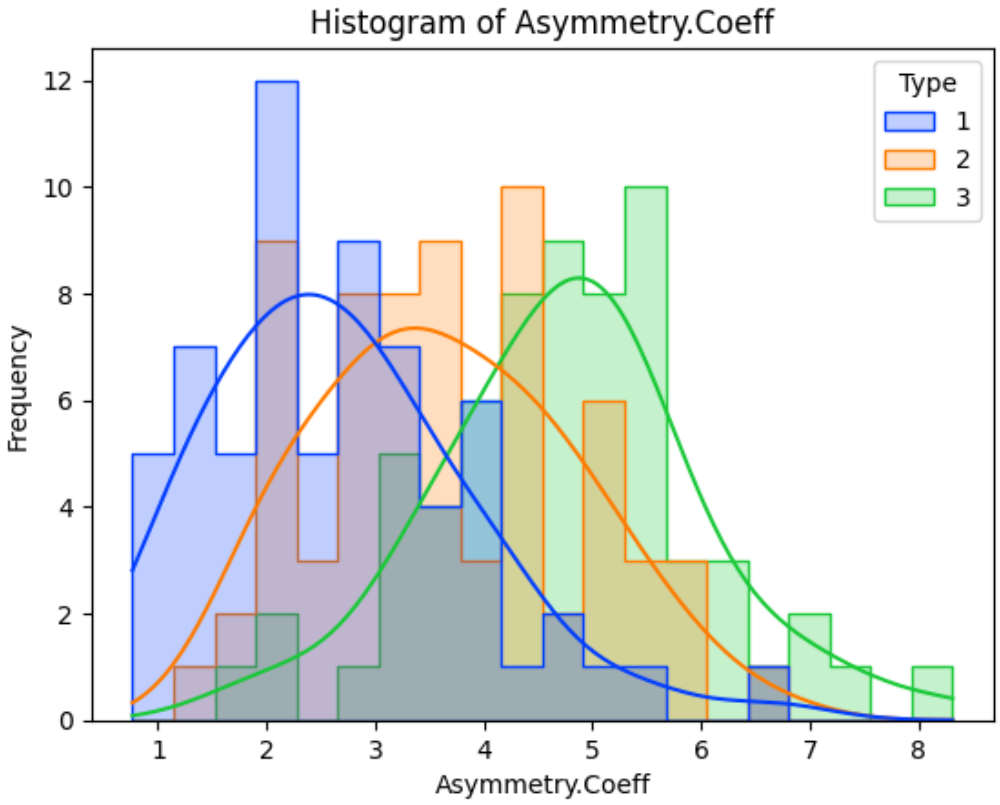
هیستوگرام ویژگی Kernel.Length



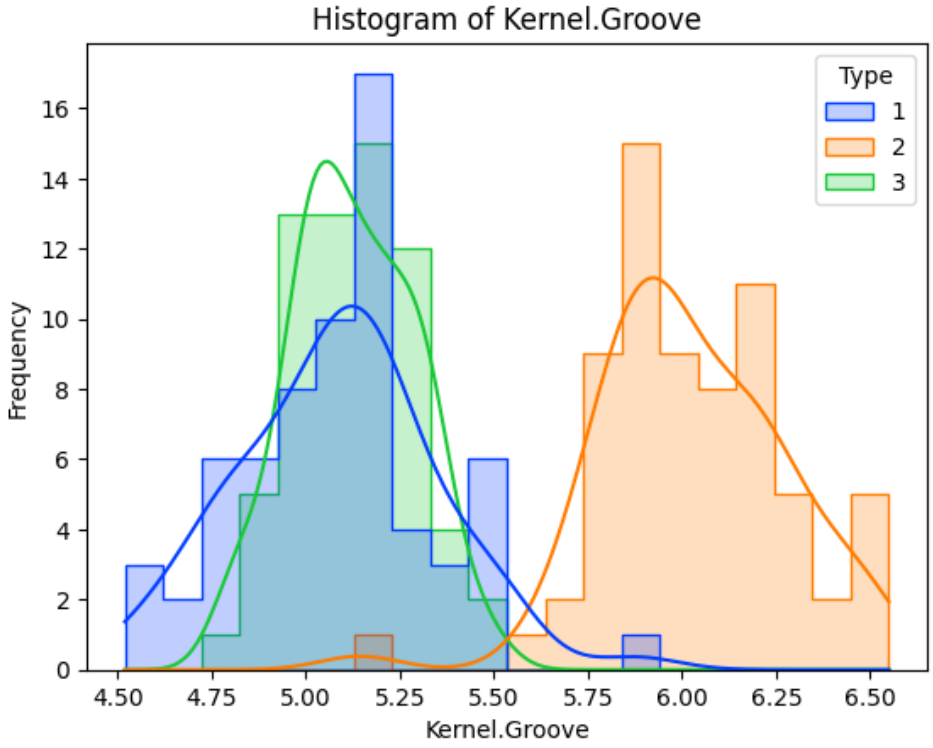
هیستوگرام ویژگی Kernel.Width



هیستوگرام ویژگی Asymmetry.Coeff



هیستوگرام ویژگی Kernel.Groove



با بررسی نمودارهای بالا متوجه شدیم که توزیع‌هایی که از هم فاصله به نسبتا خوبی دارند مثلا در ویژگی Area همان هایی هستند که در نمودار Scatterplot هم بین کلاس‌ها تفکیک به نسبت خوبی نشان دادند. پس استفاده از آن‌ها کمک می‌کند که طبقه‌بندی خوبی داشته باشیم. زیرا که پراکندگی داده‌ها در آن ویژگی‌ها بسته به نوع کلاس متفاوت است و این تفاوت همان موضوعی است که به پیش‌بینی داده‌هایی که کلاس آن‌ها مشخص نیست کمک می‌کند.

۲-۳_ پیش‌پردازش و نرمال‌سازی

در این مرحله لازم است تا پیش‌پردازش‌های لازم بر روی داده‌ها انجام شود تا پس از آن عمل اصلی مثل طبقه‌بندی را اعمال کنیم. برای نمونه مقادیری که نوع آن‌ها مشخص نیست در مرحله آموزش و تست قابل استفاده نیستند و باید حذف شوند. همچنین داده‌ها مقایس‌های گوناگونی دارند و برای اینکه تاثیر آن‌ها بسیار متفاوت از هم نباشد بهتر است که نرمال‌سازی هم روی داده‌ها انجام دهیم.

حذف مقادیر NULL

باتوجه به تصویر زیر مقادیر خالی وجود ندارد پس نیازی به حذف هیچ سطری نیست.

```python
#Checking for missing Values
missing_values_count = [0] * len(data.loc[0])
for row in data:
    for i, value in enumerate(row):
        if value is None or (isinstance(value, str) and value.strip() == ""):
            missing_values_count[i] += 1


print("Missing values per column:", missing_values_count)
```

```
Missing values per column: [0, 0, 0, 0, 0, 0, 0, 0]
```

نرمال‌سازی به روش min-max

با کمک این روش تمامی مقادیر به بازه ۰ تا ۱ تبدیل می‌شوند:

```python
# Min-Max Scaling
min_values = data.iloc[:, :-1].min()
max_values = data.iloc[:, :-1].max()
Scaled_data = (data.iloc[:, :-1] - min_values) / (max_values - min_values)

Scaled_data.head()
```

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove
0	0.440982	0.502066	0.570780	0.486486	0.486101	0.192837	0.345150
1	0.405099	0.446281	0.662432	0.368806	0.501069	0.033497	0.215165
2	0.349386	0.347107	0.879310	0.220721	0.503920	0.256149	0.150665
3	0.306893	0.316116	0.793103	0.239302	0.533856	0.197870	0.140817
4	0.524079	0.533058	0.864791	0.427365	0.664291	0.078133	0.322994

نرمال‌سازی به روش z-scoring

با کمک این روش تمام داده‌ها به نرمال استاندارد تبدیل می‌شوند. کافی است مقادیر از میانگین کم شده و تقسیم بر انحراف معیار شوند:

```python
#  Z scoreing
mean_values = Scaled_data.iloc[:, :-1].mean()
std_dev_values = Scaled_data.iloc[:, :-1].std()
Standard_data = (Scaled_data.iloc[:, :-1] - mean_values) / std_dev_values
```

```python
Standard_data.head()
```

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff
0	0.116870	0.186327	0.008124	0.270178	0.122825	-1.004836
1	-0.013269	-0.019710	0.441229	-0.200974	0.178333	-1.822590
2	-0.215325	-0.385998	1.466100	-0.793859	0.188906	-0.679910
3	-0.369436	-0.500463	1.058724	-0.719467	0.299923	-0.979005
4	0.418242	0.300792	1.397490	0.033475	0.783638	-1.593511

۳-۳_ رگرسیون لاجستیک

تقسیم داده‌ها به داده‌های آموزش و تست

قبل از اعمال مدل نیاز است که داده‌ها به دو قسمت آموزش و تست تقسیم‌بندی شوند تا بتوانیم پس از آموزش مدل آن را ارزیابی کنیم. بدیهی است داده‌های train و test اشتراکی با هم ندارند. برای انجام این کار یک تابع پیاده سازی شد. پس از اعمال این تابع بر روی دیتاست ورودی ۴ متغیر جدید خواهیم داشت:

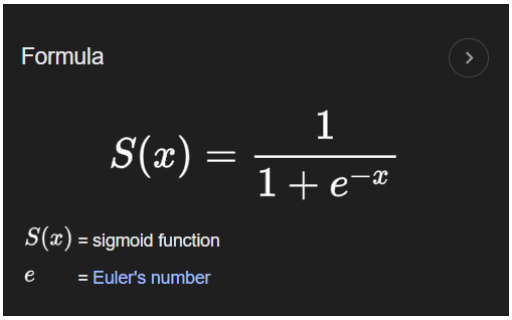
```
x_train, x_test, y_train, y_test
```

x_train برابر است با ویژگی‌هایی که قصد داریم با آن‌ها مدل را آموزش دهیم. X_test ویژگی‌هایی است که با آن‌ها مدل را ارزیابی کرده و آزمایش می‌کنیم. y_train وy_test نیز لیبل target هستند که در اینجا ۳ کلاس مختلف داریم. در این سوال از نسبت ۸۰ به ۲۰ برای تقسیم بندی استفاده کردیم و در نهایت تعداد داده‌های آموزش و تست برابر مقادیر زیر شد:

```
Train set size: 159
Test set size: 40
```

پس از این مرحله توابعی که برای آموزش مدل هستند را پیاده سازی کردیم.

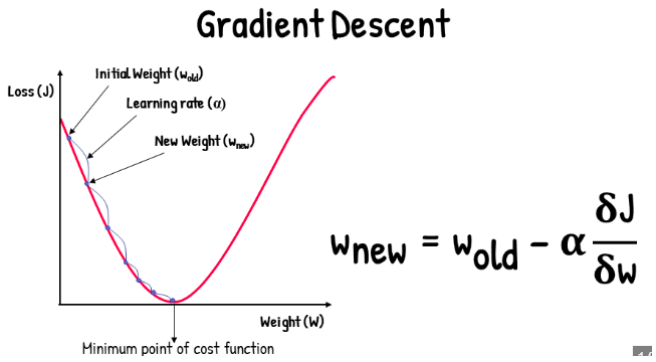
برای تبدیل مقادیر به احتمال از تابع سیگموید استفاده می‌کنیم که مقادیر ورودی را به بازه‌ی ۰ تا ۱ خواهدبرد.



همچنین برای تابع هزینه، تابع Cross Entropy را پیاده سازی کردیم.

$$J(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right],$$

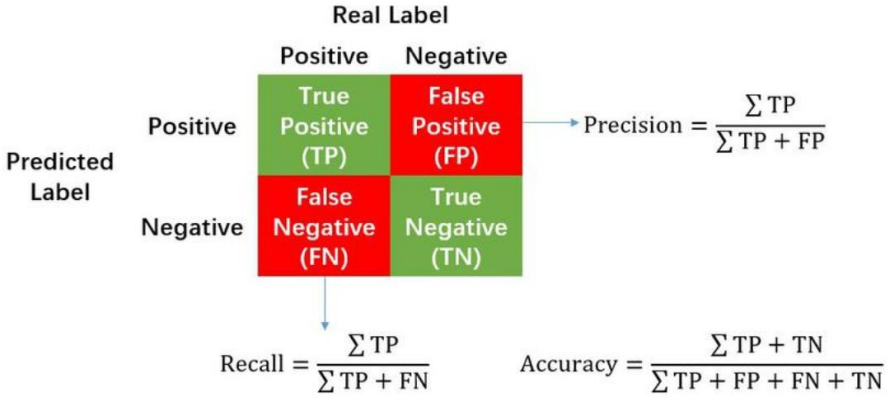
همچین برای کاهش خطا از گردیان کاهشی استفاده شد که این مورد نیز به صورت تابع پیاده‌سازی شد. در این تابع به تعداد ورودی در یک حلقه، هربار گرادیان محاسبه شده و بر اساس alpha مقادیر تتا بروزرسانی می‌شوند.



در حل این سوال مقدار alpha یا learning rate را برابر ۰.۱ در نظر گرفتیم.

تکنیک ONE VS ALL

در این مسئله قصد داریم ۳ کلاس را از هم تفکیک کرده و طبقه‌بندی کنیم. باتوجه به اینکه طبقه‌بندی باینری نیست، باید از این تکنیک استفاده کرده و مسئله را به چند طبقه‌بندی باینری تبدیل کنیم. به این صورت که هربار یک کلاس به عنوان کلاس ۱ در نظر گرفته شده و دو کلاس بعدی ۰ در نظر گرفته می‌شوند. این تکنیک پیاده سازی شد و سپس نتایج زیر برای این طبقه‌بندی حاصل شد:



مقادیر زیر در خروجی حاصل شد:

Accuracy: 0.9583333333333334

Recall: [0. 0.92307692 1]

F1 Score: [0. 0.96 0.95652174]

همچین ماتریس آشفتگی به صورت زیر است:

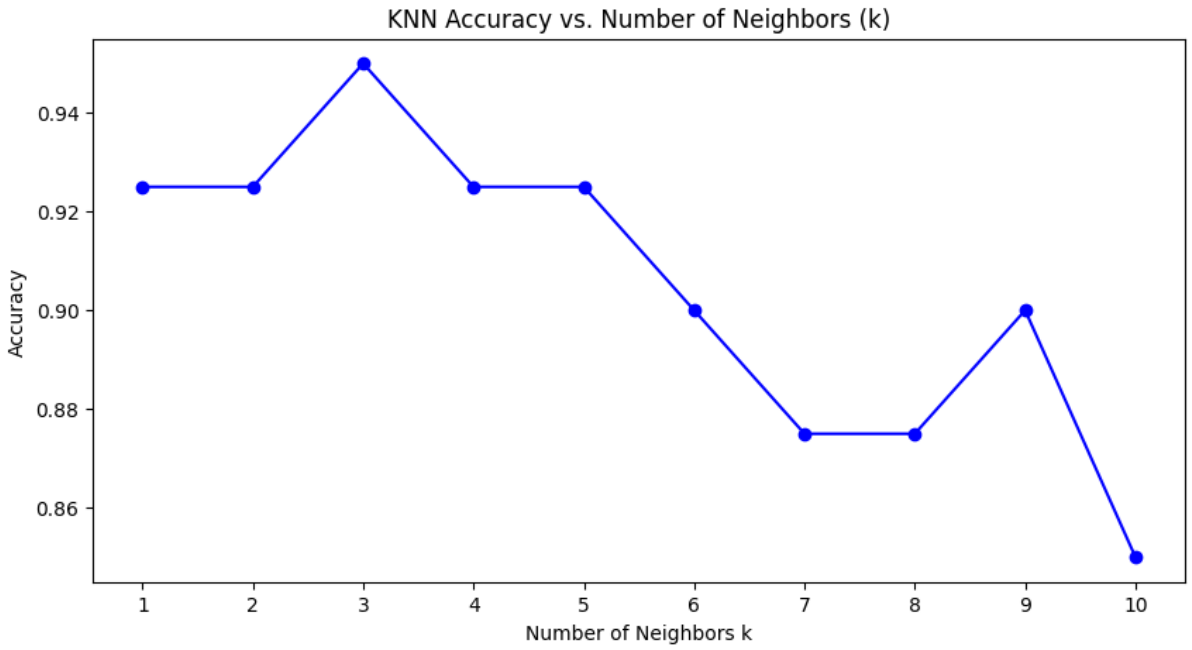
```
Confusion Matrix:
 [[ 0.  0.  0.]
 [ 0. 12.  1.]
 [ 0.  0. 11.]]
```

بررسی‌های زیادی شد ولی مشکل اینجاست که دقت کلاس اول صفر است و این یعنی هیچ مقداری را نتوانسته درست تشخیص دهد که اصلا نتیجه خوبی نیست. یا مشکل از کد است یا اینکه ویژگی‌هایی که باعث شده‌اند کلاس ۱ و ۲ شبیه هم باشند علت این موضوع است.

۴-۳_ طبقه‌بندی با KNN

در آخرین بخش از این مسئله، طبقه‌بندی را با روش KNN انجام می‌دهیم. در این روش هر داده با k نزدیک ترین همسایه خود مقایسه شده و رای اکثریت گرفته می‌شود و لیبل داده جدید را اینگونه تعیین می‌کنیم.

برای اینکه بررسی کنیم چه مقدار از پارامتر k بهترین دقت را می‌دهد، در یک حلقه دقت‌های مختلف را به ازای K از ۱ تا ۱۰ بررسی کردیم که نمودار زیر نتایج این بررسی را نشان می‌دهد



پس افزایش K همیشه باعث افزایش دقت نمی‌شود و باید مقدار مناسبی تعیین شود تا مدل دچار Overfit یا Underfit نشود. برای نمونه در این مسئله بهترین k برابر ۳ است و دقت مدل در این مقدار ماکزیمم است.

۴_پاسخ سوال ۹

ابتدا دیتاست را بارگذاری کرده و چند سطر اول آن را به همراه ستون‌ها بررسی می‌کنیم. در این دیتاست دو ستون اول که شمارنده و ID هستند تاثیر خاصی ندارند و نیاز نیست که برای پردازش داده‌ها این موارد را جز ویژگی‌ها محسوب کنیم.

```
data_info = data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Unnamed: 0           2240 non-null   int64  
 1   ID                   2240 non-null   int64  
 2   Year_Birth           2240 non-null   int64  
 3   Education            2240 non-null   object 
 4   Marital_Status       2240 non-null   object 
 5   Income               2017 non-null   float64
 6   Kidhome              2240 non-null   int64  
 7   Teenhome             2240 non-null   int64  
 8   Dt_Customer          2240 non-null   object 
 9   Recency              2240 non-null   int64  
 10  MntCoffee            2035 non-null   float64
 11  MntFruits            2240 non-null   int64  
 12  MntMeatProducts      2240 non-null   int64  
 13  MntFishProducts      2240 non-null   int64  
 14  MntSweetProducts     2240 non-null   int64  
 15  MntGoldProds         2227 non-null   float64
 16  NumWebVisitsMonth    2040 non-null   float64
 17  Complain             2240 non-null   int64  
 18  NumPurchases         2240 non-null   int64  
 19  UsedCampaignOffer    2240 non-null   int64  
dtypes: float64(4), int64(13), object(3)
memory usage: 350.1+ KB
```

در این دیتاست به طور کلی ۲۰ ستون وجود دارد که در شکل بالا انواع آن به همراه نام و TYPE داده قابل مشاهده است.

EDA _۴-۱

داده‌های از دست رفته

Missing Values

```
data_missing = data.isnull().sum()
data_missing
```

```
Unnamed: 0           0
ID                   0
Year_Birth           0
Education            0
Marital_Status       0
Income             223
Kidhome              0
Teenhome             0
Dt_Customer          0
Recency              0
MntCoffee          205
MntFruits            0
MntMeatProducts      0
MntFishProducts      0
MntSweetProducts     0
MntGoldProds        13
NumWebVisitsMonth  200
Complain             0
NumPurchases         0
UsedCampaignOffer    0
dtype: int64
```

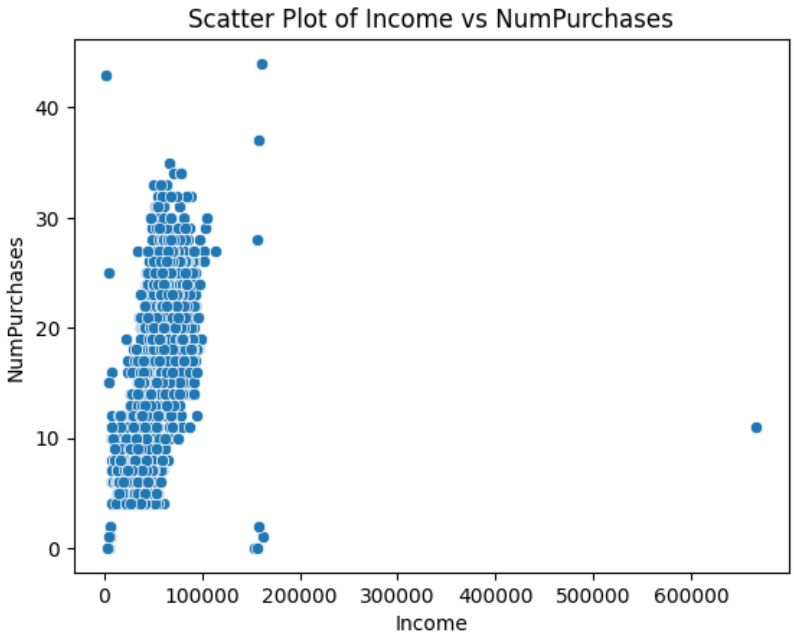
باتوجه به بررسی‌های انجام شده ۴ ستون دارای مقادیر null هستند. برای اینکه درصد هر کدام مشخص شود مقادیر null تقسیم بر تمامی مقادیر می‌شوند. نتیجه به صورت زیر است:

	Missing Percentage
Income	9.955357
MntCoffee	9.151786
MntGoldProds	0.580357
NumWebVisitsMonth	8.928571

نمودارهای ScatterPlot

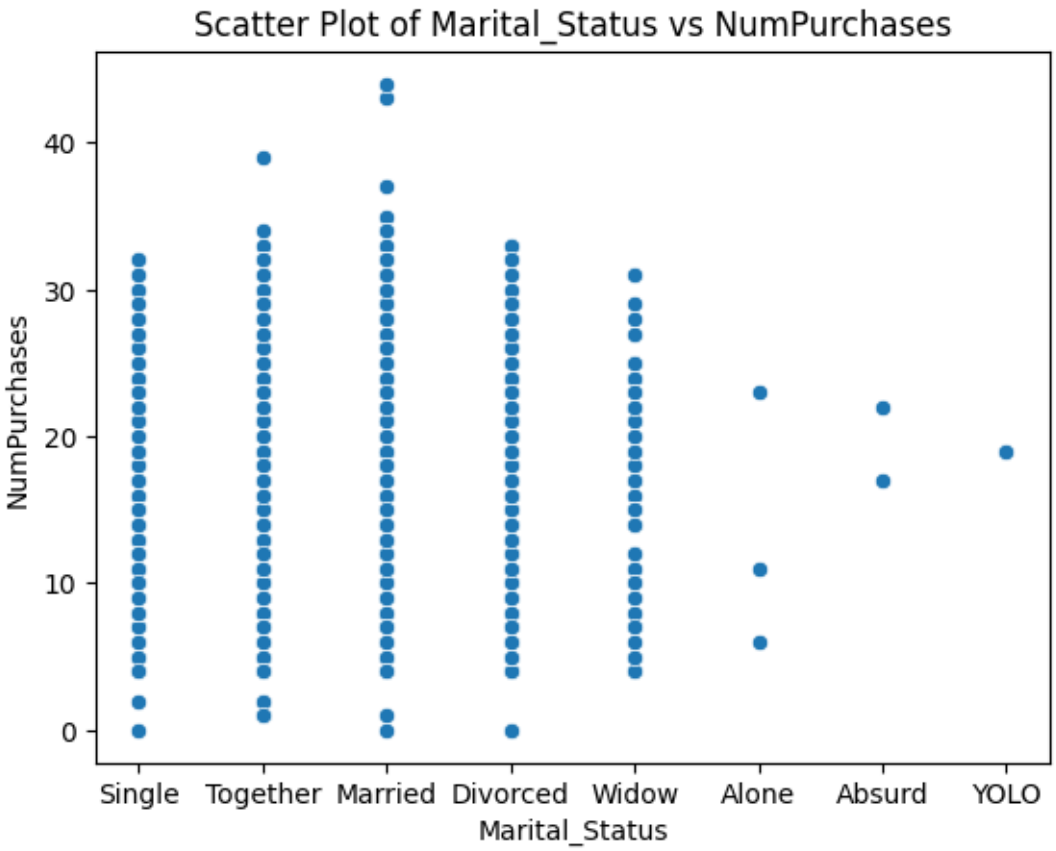
در این بخش نیز به مانند سوال قبلی نمودارهای ScatterPlot برای هر جفت از ویژگی‌ها رسم شد ولی به دلیل تعداد بالا (حدود۱۵۰) خروجی در نوت بوک ذخیره شده است.

نکته‌ای که حائز اهمیت است این است که باتوجه به اینکه بسیاری از متغیرها از نوع Categorical هستند، نمودار Scatterplot بهترین نمودار برای توصیف ارتباط بین آن‌ها نیست. ولی به هرحال طبق خواسته سوال تمامی آن‌ها رسم شدند. در زیر برخی از آن‌ها را مشاهده می‌کنیم:



مثلا در نمودار بالا ارتباط بین تعداد خرید و درآمد مشخص شده است. هرچند که بخاطر داده‌های Outlier ارتباط به خوبی نشان داده نشده است.

حتی در داده‌هایی که Numericalنیستند هم می‌توان به نتایجی دست یافت:

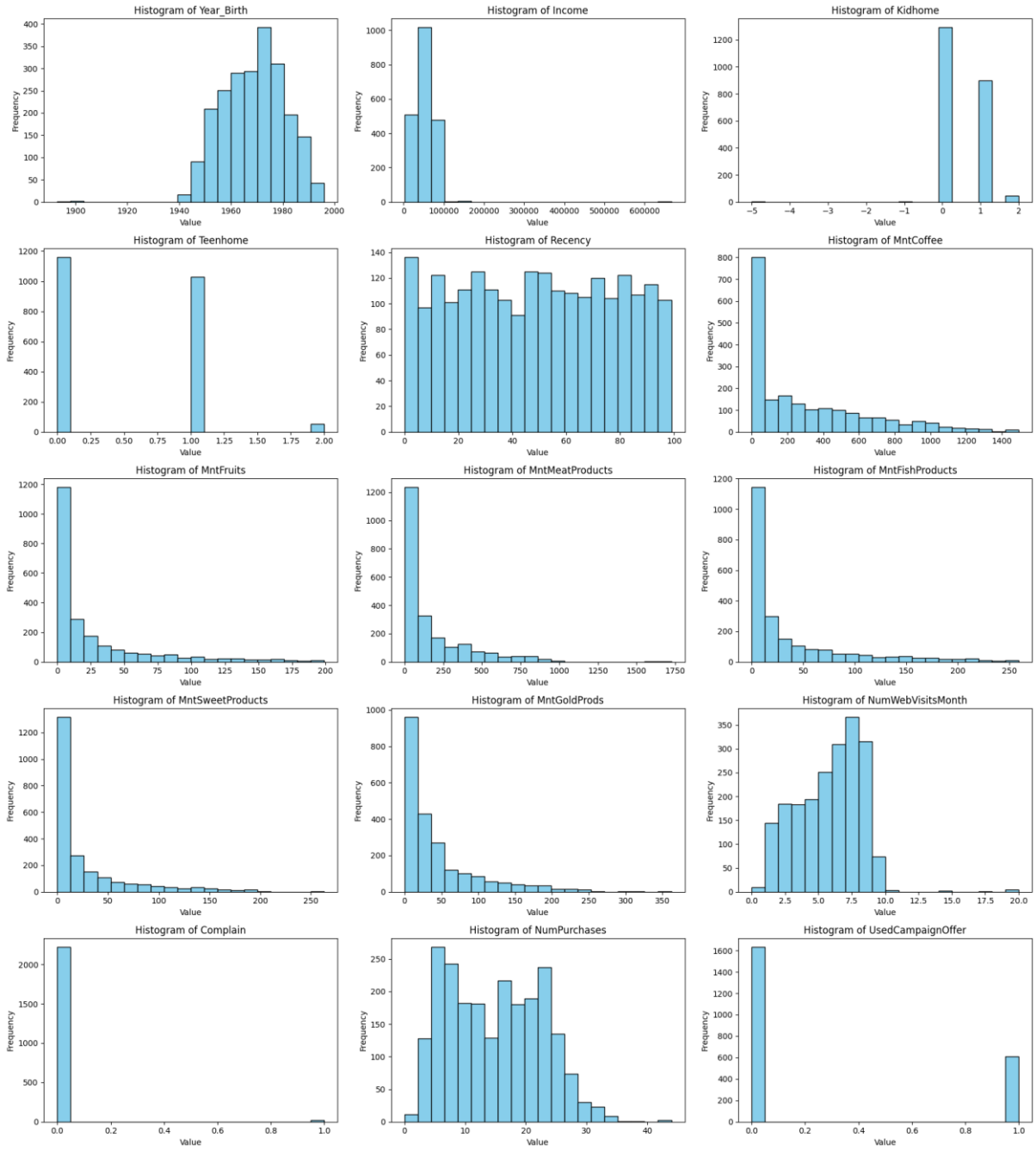


مثلا در نمودار بالا متوجه می‌شویم که وضعیت تاهل افراد بر تعداد خرید آن‌ها تاثیرگذار است.

نمودارهای Histogram

در این مسئله برای بررسی توزیع ویژگی‌های مختلف نیز نمودار Histogram آن‌ها رسم شد.

برای اینکه از نظر بصری بتوانیم بهتر مشاهده کنیم، نمودارها به صورت Subplot هستند. این نوع نمودار برای داده‌های Numerical بهتر است . به همین دلیل ابتدا این نوع داده‌ها را جدا کردیم و سپس برای آن نمودار رسم کردیم.



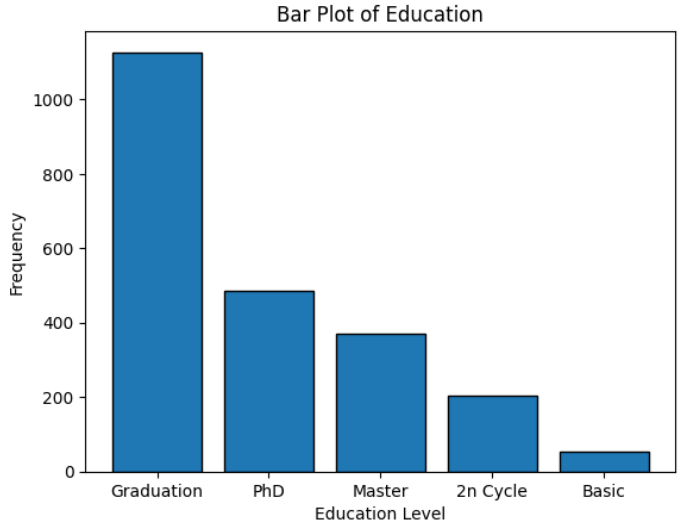
برای نمونه نمودار تاریخ تولد نشان می‌دهد که نرمال نزدیک است.

و یا برای نمودار درآمد مشخص است که رنج بیشتر مردم درآمد حدود ۱۰۰۰۰ دارند. نکته‌ای که باید توجه شود این است که Outlier ها تاثیر زیادی در نمودارها دارند و باید این مورد را در نظر گرفت.

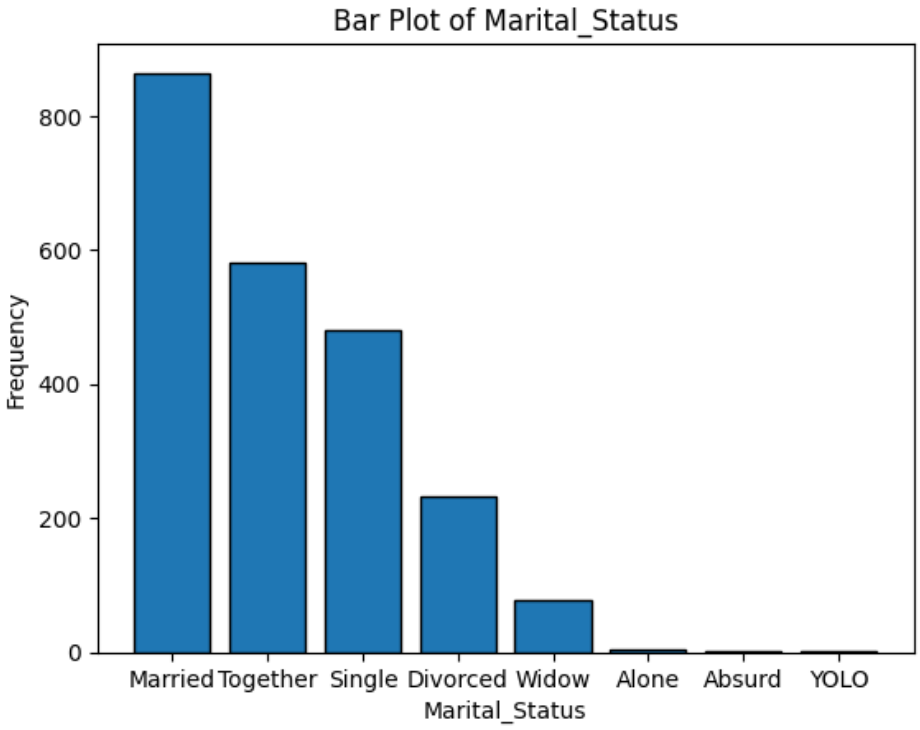
حال برای داده‌های Categorical نیز بجای Histogram می‌توانیم از Barplot استفاده کنیم.

نمودار BarPlot

همانطور که گفته شد برای داده‌هایی که عددی نیستند بهتر است از این نوع نمودار استفاده کنیم.
برای نمونه نمودار تحصیلات به صورت زیر است :

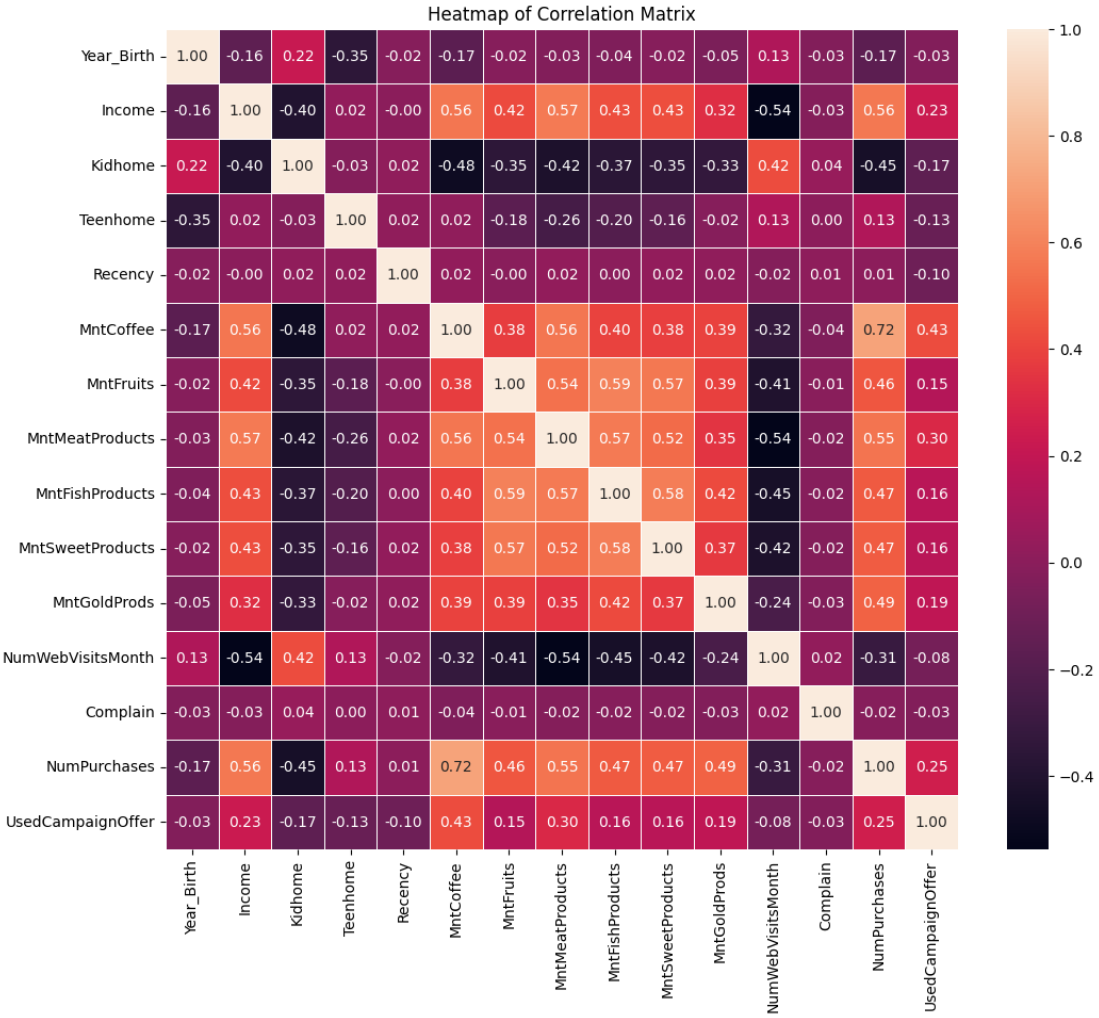


میتوانیم متوجه شویم که بیشتر افراد فارغ التحصیل کارشناسی هستند. افراد با سواد ابتدایی تعداد کمی از مراجعه کنندگان را تشکیل می‌دهند. و یا برای نمودار وضعیت تاهل می‌توانیم متوجه شویم که افرادی که مجرد نیستند بسیار بیشتر هستند:



۲-۴_ همبستگی بین متغیرها

برای نمایش همبستگی بین متغیرها از نمودار Heatmap استفاده می‌کنیم. این نمودار بر اساس Correlation Coefficent بین متغیرها تشکل می‌شود و هر خانه از آن مقداری بین -۱ تا +۱ میگیرد. این مورد برای اینکه بعدا بتوانیم یک مدل رگرسیون روی داده‌ها تنظیم کنیم خیلی مهم است و به ما کمک می‌کند که ویژگی‌های بهتری را برای آن انتخاب کنیم:

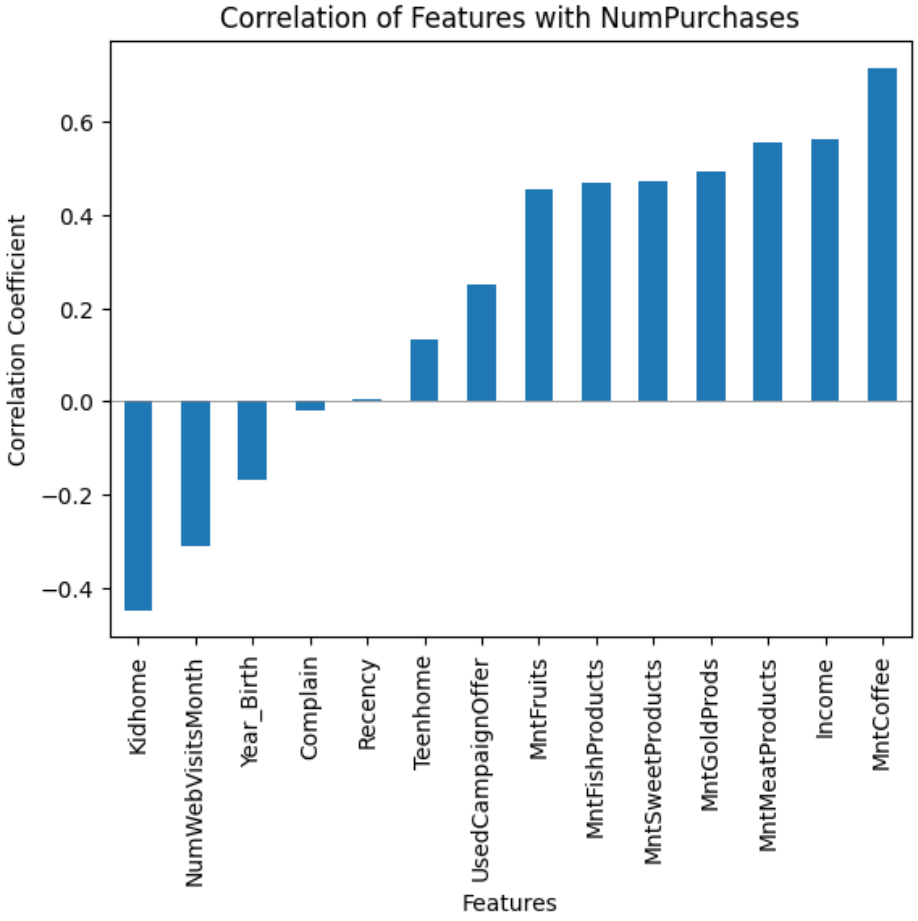


هرچند که وابستگی بین متغیرها و متغیر TARGET که NumPurchases است به صورت عادی نیز خروجی گرفته شد:

```
Kidhome          -0.447073
NumWebVisitsMonth  -0.309666
Year_Birth       -0.168304
Complain         -0.020583
Recency           0.005740
Teenhome           0.133163
UsedCampaignOffer   0.251386
MntFruits          0.455461
```

```
MntFishProducts     0.469454
MntSweetProducts    0.472876
MntGoldProds        0.493939
MntMeatProducts     0.554229
Income              0.562603
MntCoffee           0.715164
```

اگر که مقادیر بالا را در یک نمودار رسم کنیم نتیجه زیر حاصل می‌شود:



نمودار بالا نشان می‌دهد که سه ویژگی mntcoffee و income و mntmeatproducts به ترتیب بیشترین همبستگی را با متغیر TARGET که NumPurchases است را دارند. در مرحله بعد از این متغیرها برای رگرسیون می‌توان استفاده کرد.

۳-۴_ پیش پردازش داده‌ها

قبل از اینکه مدل را آموزش داده و تست کنیم نیاز است که داده‌ها را آماده کنیم. در این مسئله ابتدا مقادیر nan را مدیریت کرده و سپس داده‌ها را به دو قسمت آموزش و تست تقسیم میکنیم.

Handling missing values

برای داده‌هایی که null هستند می‌توان تصمیمات مختلفی گرفت. باتوجه به قسمت قبلی سوال می‌دانیم که درصدی از ۴ متغیر miss هستند:

	Missing Percentage
Income	9.955357
MntCoffee	9.151786
MntGoldProds	0.580357
NumWebVisitsMonth	8.928571

برای داده mntGoldProds به دلیل اینکه درصد آن کم است می‌توانیم کل سطر را حذف کنیم.

ولی دیگر داده‌ها درصد زیادی دارند و باید مقادیر را مدیریت کنیم.

برای نمونه Income را بهتر است با میانه آن جایگزین کنیم. دلیل استفاده نکردن از میانگین این است که این متغیر دارای Outlier است که تاثیر زیادی بر میانگین می‌گذارد.

برای دیگر داده‌ها نیز می‌توانیم بر حسب همین موضوع میانگین یا میانه بگیریم.

Split train test

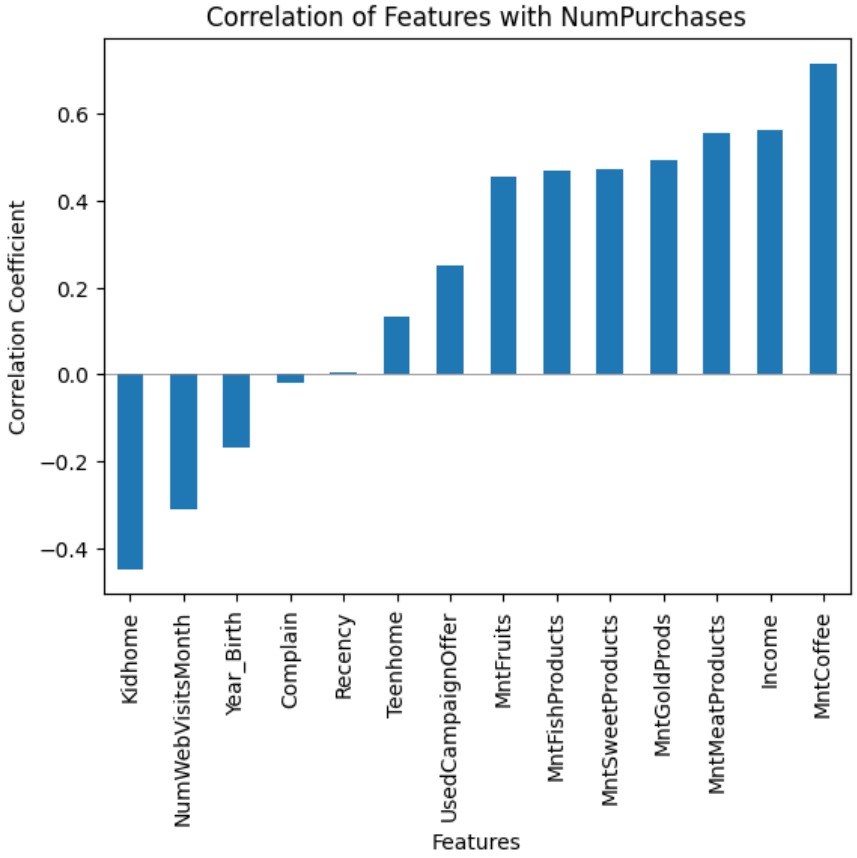
قبل از اعمال مدل نیاز است که داده‌ها به دو قسمت آموزش و تست تقسیم‌بندی شوند تا بتوانیم پس از آموزش مدل آن را ارزیابی کنیم. بدیهی است داده‌های train و test اشتراکی با هم ندارند. برای انجام این کار یک تابع پیاده سازی شد. پس از اعمال این تابع بر روی دیتاست ورودی ۴ متغیر جدید خواهیم داشت:

```
x_train, x_test, y_train, y_test
```

x_train برابر است با ویژگی‌هایی که قصد داریم با آن‌ها مدل را آموزش دهیم. X_test ویژگی‌هایی است که با آن‌ها مدل را ارزیابی کرده و آزمایش می‌کنیم. y_trainو y_test نیز لیبل target هستند که در اینجا ۳ کلاس مختلف داریم. در این سوال از نسبت ۸۰ به ۲۰ برای تقسیم بندی استفاده کردیم

۴-۴_ رگرسیون با ۱ ویژگی

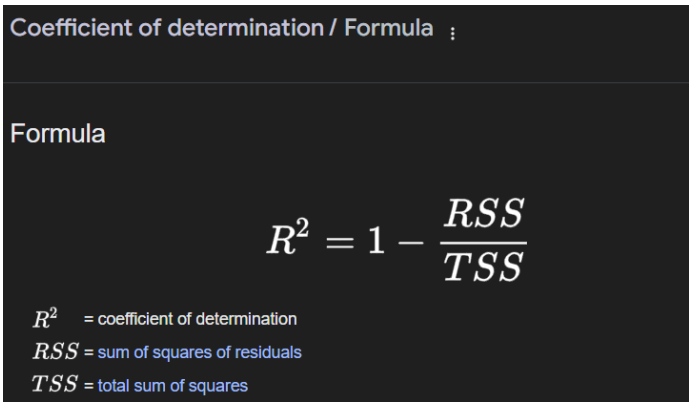
باتوجه به بررسی انجام شده در قسمت قبلی، ویژگی که بیشترین CC یا ضریب همبستگی با Target که همان NumPurchases داشت، MntCoffee بود پس این ویژگی را برای مدل رگرسیون انتخاب می‌کنیم.



پس از آموزش مدل مقادیر زیر برای خطا و R2 Score بدست آمد. در این مسئله از این دو مقدار برای ارزیابی مدل استفاده می‌کنیم. این سوال مثل قبلی نیست که بتوانیم دقت را با توجه به تعداد تشخیص نادرست اعلام کنیم. مدل رگرسیون در این سوال با پارامتر RMSE که به نحوی خطای تخمین را محاسبه می‌کند محاسبه می‌شود. هرچه که این مقدار کمتر باشد.، مدل ما بهتر است.:

$$RMSE = \sqrt{\frac{\Sigma (y_i - \hat{y}_i)^2}{N - P}}$$

پارامتر دیگر نیز R2 است که نشان دهنده این است که متغیر مستقل ما به چه میزان توسط متغیر وابسته قابل توصیف است. هرچه که این مقدار بیشتر باشد، مدل ما بهتر است.



در بخش اول با ۱ ویژگی نتیجه زیر حاصل شد:

RMSE: 5.515693934346293, R2: 0.5060615818164325

۵-۴_ رگرسیون با ۳ ویژگی

در بخش دوم با ۳ ویژگی که نحوه انتخاب آن‌ها در قسمت قبل توضیح داده‌شد، نتیجه زیر حاصل شد:

RMSE: 5.332397723894167, R2: 0.5383449857788027

همانطور که مشاهده می‌شود، علاوه بر اینکه R2 افزایش یافت RMSE هم کاهش یافت. پس مدل بهبود یافته است. یعنی پیش‌بینی‌هایی که مدل انجام می‌دهد، به مقدار واقعی نزدیک تر شده‌اند.

دلیل انتخاب ۳ ویژگی همانطور که در قسمت قبل از ذکر شد باتوجه به ضریب همبستگی و رابطه خطی بین ویژگی‌ها و متغیر TARGET است که نمودار آن را در شکل صفحه قبل مشاهده می‌کنیم.

این ۳ متغیر با بیشترین ضریب همبستگی :

MntMeatProducts 0.554229

Income 0.562603

MntCoffee 0.715164