گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین
عرفان کرمی – ۹۸۲۲۲۰۷۹

۱. پس از دریافت اطلاعات کلی از دیتاست متوجه شدم که در دیتاست داده گمشده وجود ندارد و از این نظر نیازی به انجام پردازش نیست.

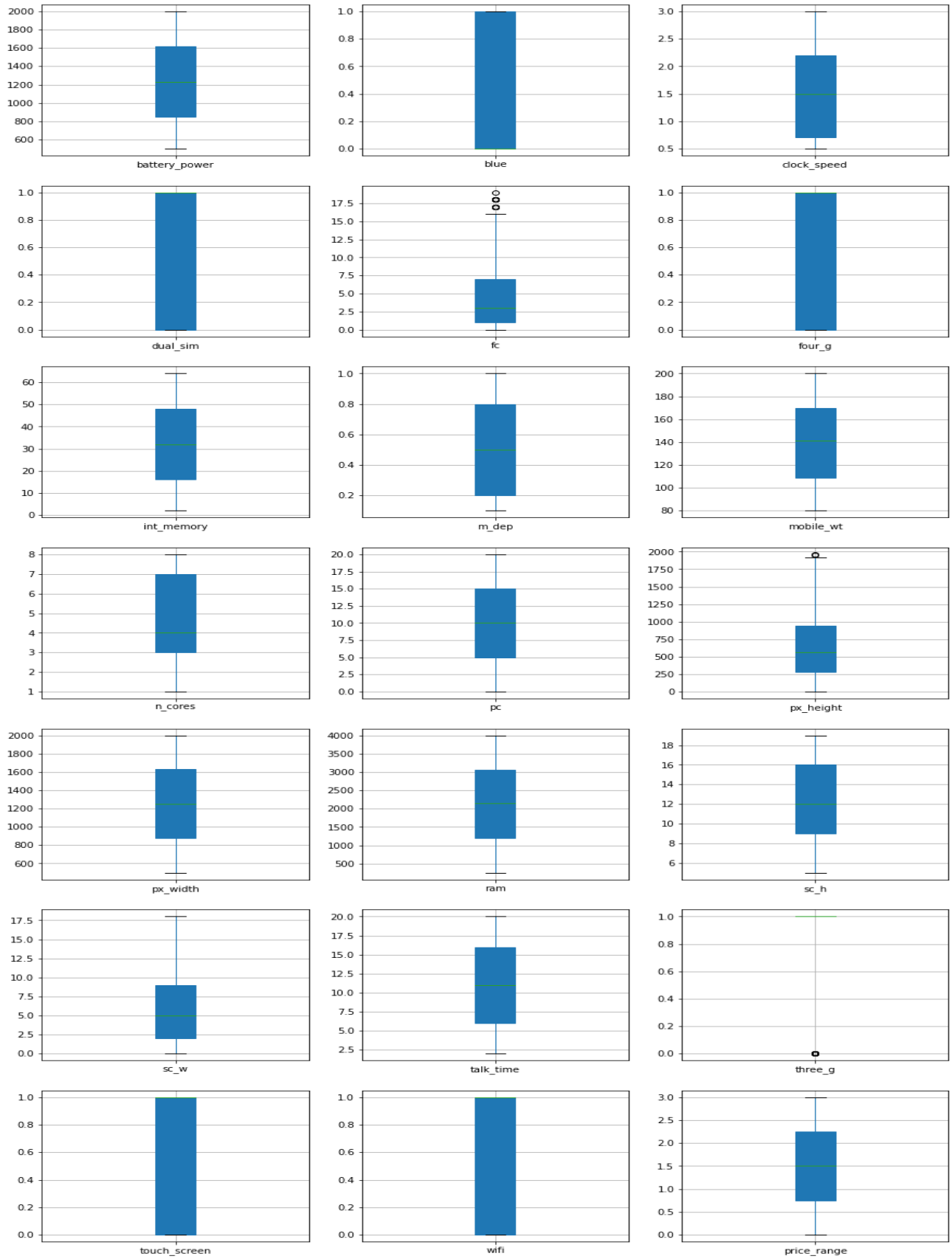
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  2000 non-null   int64
 1   blue           2000 non-null   int64
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64
 4   fc             2000 non-null   int64
 5   four_g         2000 non-null   int64
 6   int_memory     2000 non-null   int64
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64
 9   n_cores        2000 non-null   int64
 10  pc             2000 non-null   int64
 11  px_height      2000 non-null   int64
 12  px_width       2000 non-null   int64
 13  ram            2000 non-null   int64
 14  sc_h           2000 non-null   int64
 15  sc_w           2000 non-null   int64
 16  talk_time      2000 non-null   int64
 17  three_g        2000 non-null   int64
 18  touch_screen   2000 non-null   int64
 19  wifi           2000 non-null   int64
 20  price_range    2000 non-null   int64
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

برای یافتن داده های پرت یا همان (outlier) از متد IQR استفاده میکنیم . به این منظور ابتدا نموار جعبه ای را برای هر فیچر رسم میکنیم و وجود یا عدم وجود داده های پرت را به صورت بصری نشان میدهیم.

شکل صفحه بعد این نمودار را به تفکیک هر فیچر نشان میدهد.

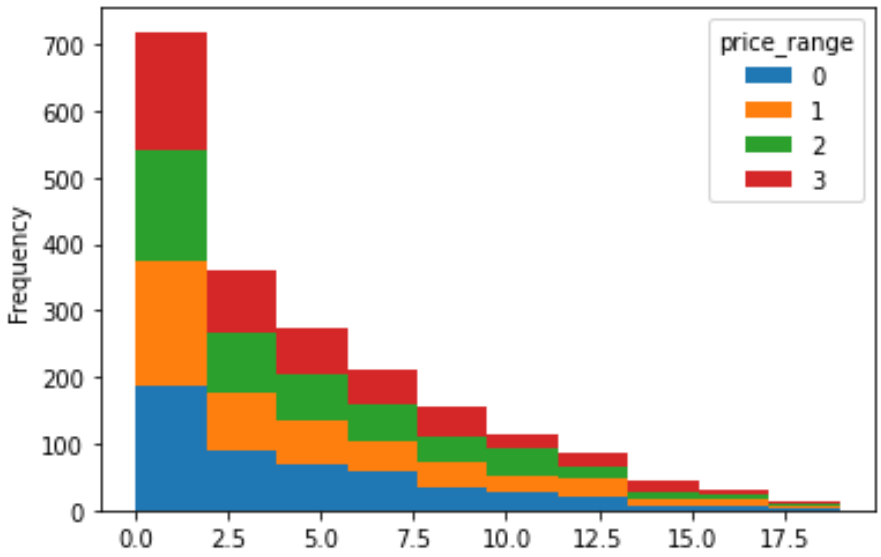
گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین

عرفان کرمی – ۹۸۲۲۲۰۷۹



همانگونه که از نمودار های جعبه ای هم مشخص است تنها سه فیچر tree_g, px_height, fc حاوی داده های پرت هستند که اگر دقت کنیم tree_g یک داده کتگوریکال دومقداری است و به همین دلیل داده هایی که متد IQR شناسایی کرده واقعا داده پرت نیستند.

برای ستون fc که همان تعداد پیکسل های دوربین جلو که بر اساس مگاپیکسل است هیستوگرام را رسم کرده ایم:



تصویر به خوبی نشان میدهد که در این ستون ما درگیر مشکل long tail هستیم که میتواند عملکرد الگوریتم های مختلف ما را تحت الشعاع قرار دهد پس حذف داده های پرت که داده های سمت راست هستند و سپس اسکیل کردن آنها میتواند این مشکل را برطرف کند. از طرفی به خوبی مشهود است که نسبت داده ها در رنج های مختلف قیمتی به ازای هر ستون تقریبا یکسان است پس در واقع الگوی یکسانی وجود دارد و احتمالا همبستگی بین رنج قیمت و مگاپیکسل دوربین جلو ناچیز است پس مجددا حذف این داده ها اشکالی را ایجاد نمیکند.

پس داده های پرت را برای دو ستون fc و px_height حذف میکنیم. پس از حذف ۱۸ +۲ =۲۰ داده پرت شناسایی و حذف شد .

گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین
عرفان کرمی – ۹۸۲۲۲۰۷۹

۲. برای بررسی فیچرها و اطلاعات آماری مربوط به آنها اقدامات مختلفی میتوانیم انجام دهیم که به مرور هز کدام را به طور کامل بررسی و تفسیر میکنیم.

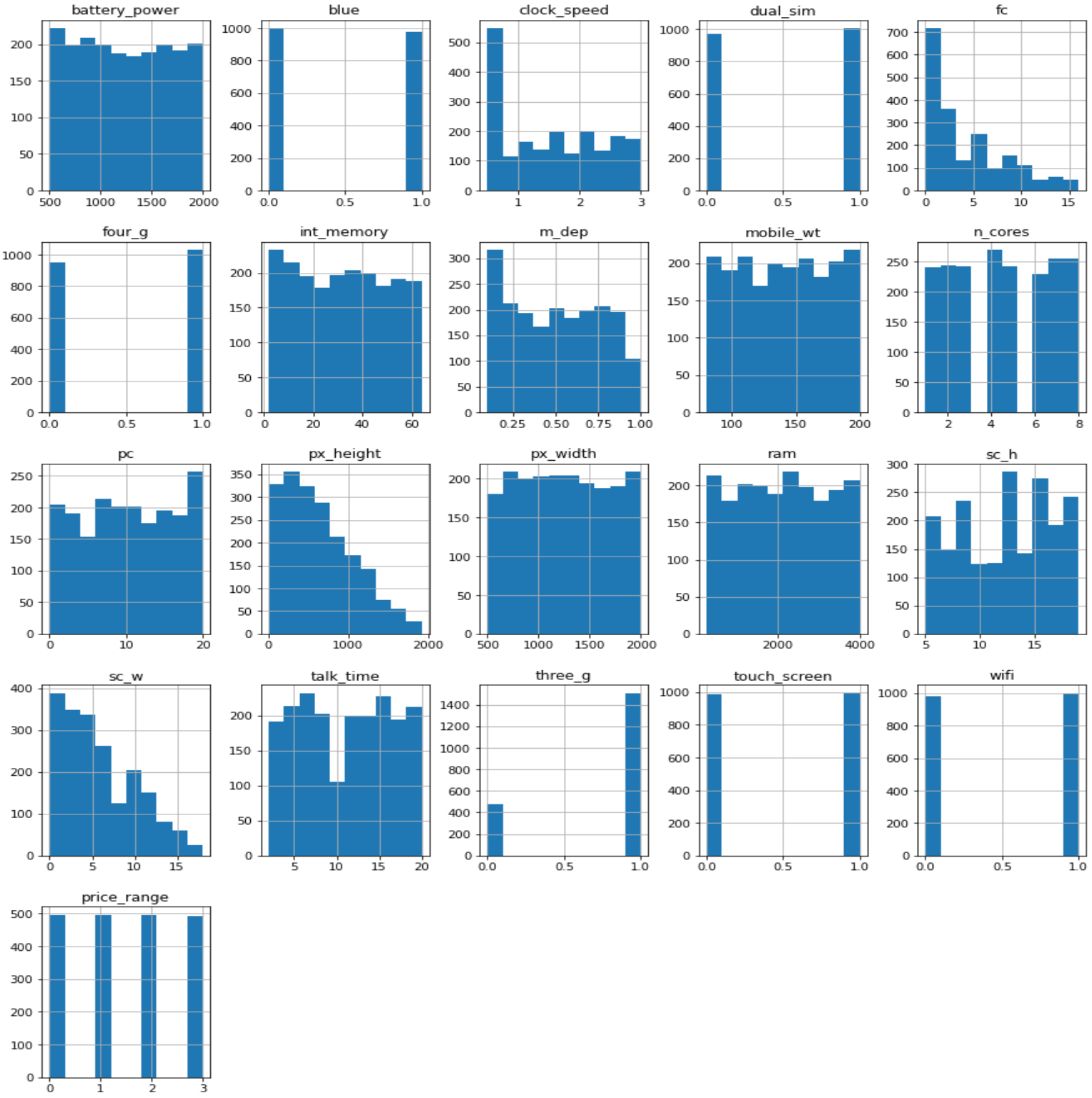
	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc
count	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000
mean	1236.402020	0.494444	1.521162	0.509091	4.189899	0.520707	32.055051	0.502222	140.168182	4.527273	9.836364
std	439.568762	0.500095	0.817236	0.500044	4.168422	0.499697	18.126820	0.288722	35.385004	2.291263	6.023408
min	501.000000	0.000000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000	1.000000	0.000000
25%	849.500000	0.000000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000	3.000000	5.000000
50%	1224.000000	0.000000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000	4.000000	10.000000
75%	1614.000000	1.000000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000	7.000000	15.000000
max	1998.000000	1.000000	3.000000	1.000000	16.000000	1.000000	64.000000	1.000000	200.000000	8.000000	20.000000

	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
count	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000	1980.000000
mean	641.183838	1249.646465	2125.135859	12.316667	5.773737	11.003535	0.760606	0.501515	0.505051	1.497980
std	439.957961	431.606730	1084.556984	4.209307	4.358066	5.458048	0.426821	0.500124	0.500101	1.117863
min	0.000000	500.000000	256.000000	5.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000
25%	281.750000	874.000000	1209.750000	9.000000	2.000000	6.000000	1.000000	0.000000	0.000000	0.000000
50%	561.500000	1247.000000	2146.500000	12.000000	5.000000	11.000000	1.000000	1.000000	1.000000	1.000000
75%	942.000000	1629.250000	3066.500000	16.000000	9.000000	16.000000	1.000000	1.000000	1.000000	2.000000
max	1920.000000	1998.000000	3998.000000	19.000000	18.000000	20.000000	1.000000	1.000000	1.000000	3.000000

در بالا اطلاعات آماری کلی در رابطه با فیچرها آماده است حال ابتدا یک هیستوگرام از فیچر ها رسم، سپس ماتریس همبستگی را برای داده ها به دست می آوریم هر یک را به طور کلی تر تفسیر میکنیم.

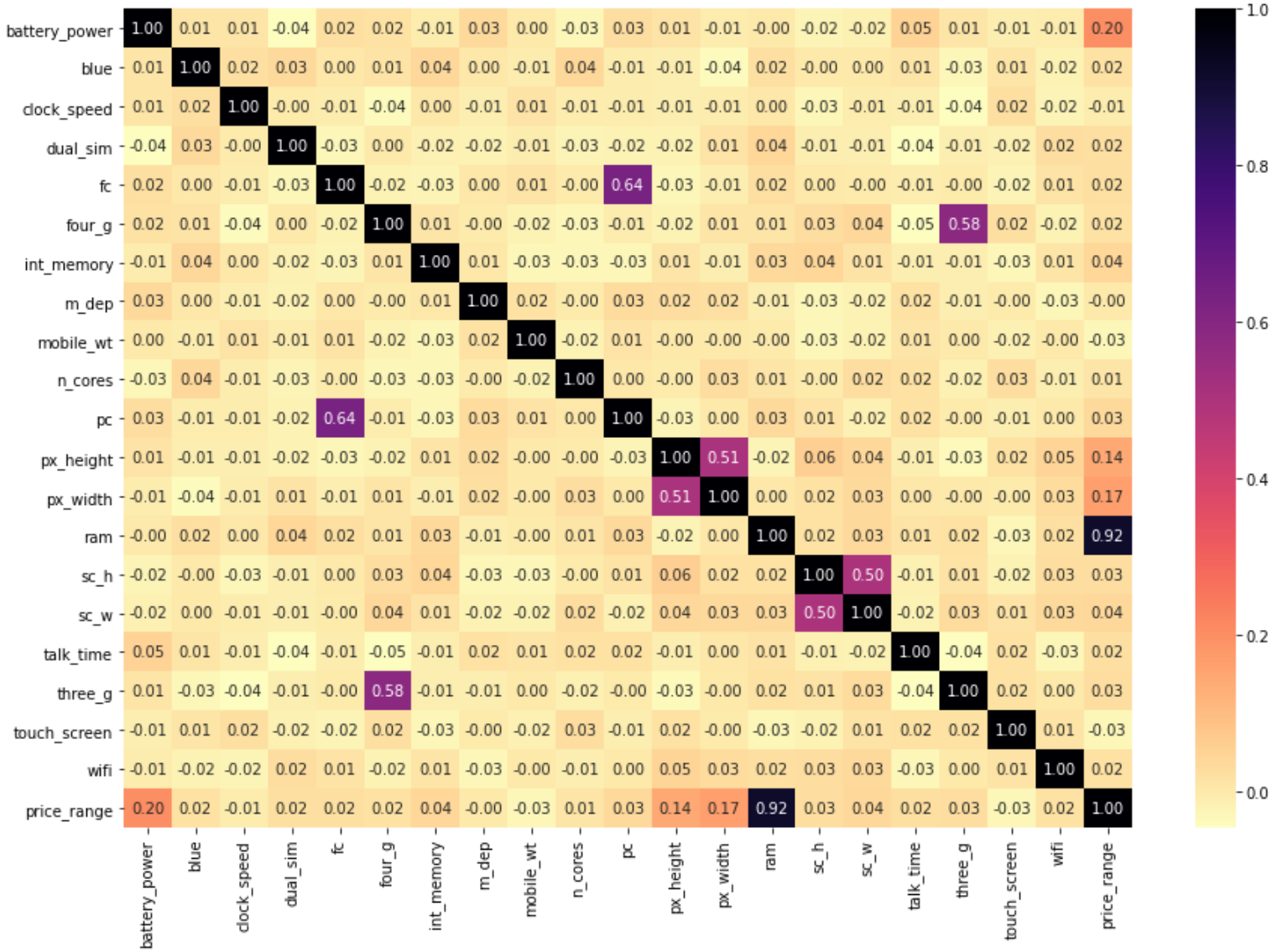
گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین

عرفان کرمی – ۹۸۲۲۲۰۷۹



گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین

عرفان کرمی – ۹۸۲۲۲۰۷۹

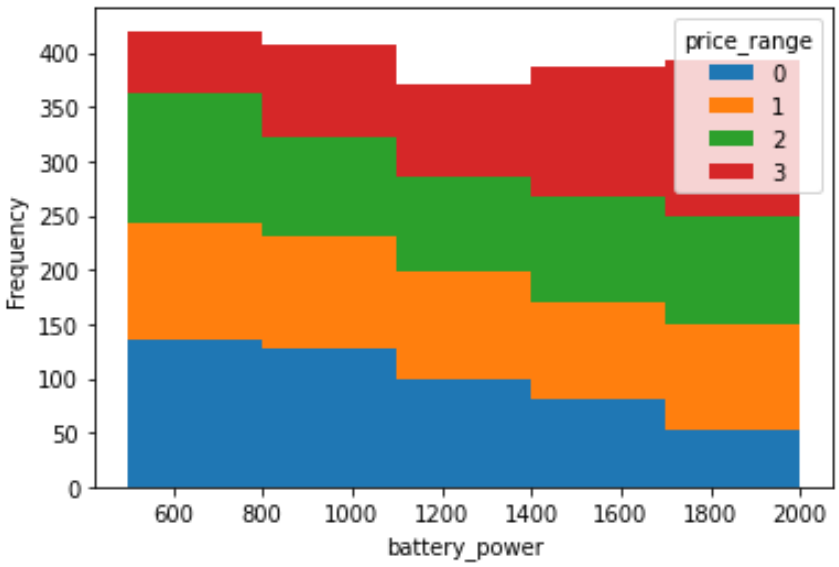


نمودار هیستوگرام نشان میدهد که اکثر فیچرهای دیتاست تا حدودی بالانس هستند و مشکل long tail برای آنها وجود ندارد و این مسئله خبر خوبی است چرا که داده های نامتوازن عملکرد الگوریتم ها را تحت الشعاع قرار میدهند و آنها را تضعیف میکنند در یکی دو مورد البته این مشکل وجود دارد که با استفاده از scaling میتوان با آن مقابله و اثر آن را کاهش داد.

ماتریس کواریانس هم نشان میدهید که مرتبط ترین فیچر ها که همبستگی زیادی با بازه قیمت دارند عبارتند از battery_power, px_height,px_width,ram از طرفی با استفاده از همبستگی بین سایر فیچر میتوان پیشبینی کرد که ترکیب کردن بعضی از این فیچر ها میتواند ویژگی های بهتری را ایجاد کند که این موارد را هم انجام داده و مجددا مارتیس کواریانس را رسم میکنیم.

برای دید بهتر نسبت به فیچر ها بعضی از آنها را بررسی میکنیم.

Battery_power



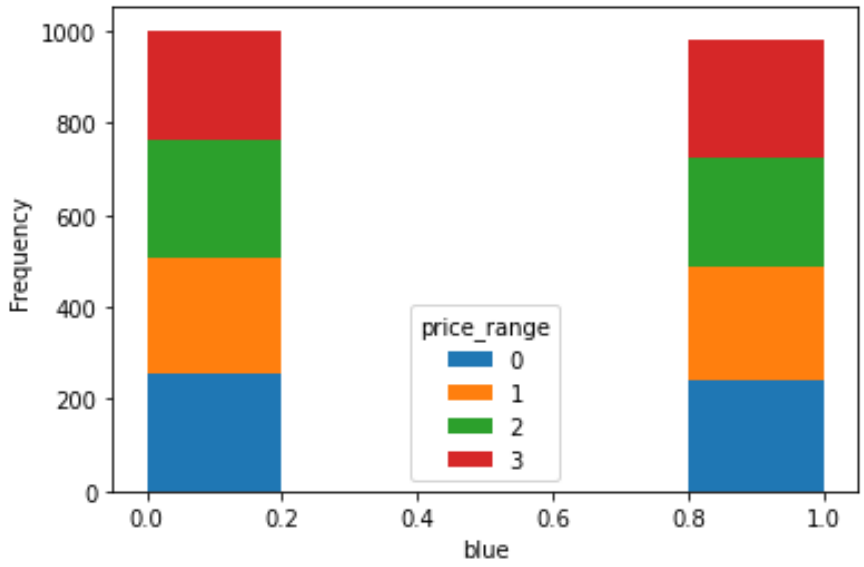
این نمودار به خوبی نشان میدهد که داده های ما برای این فیچر تا حد زیادی بالانس هستند و از طرفی با افزایش ظرفیت باتری تعداد موبایل هایی با قیمت متوسط تقریبا ثابت میماند اما تعداد موبایل های گران قیمت افزایش و تعداد موبایل های ارزان قیمت کاهش می یابد. از این نمودار میتوان تفسیر کرد که احتمالا یکی همبستگی مثبت و قابل توجهی میان ظرفیت باتری و قیمت موبایل وجود دارد. این موضوع در شکلی که از ماتریس کوواریانس رسم کردیم هم کاملا مشهود است.

گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین

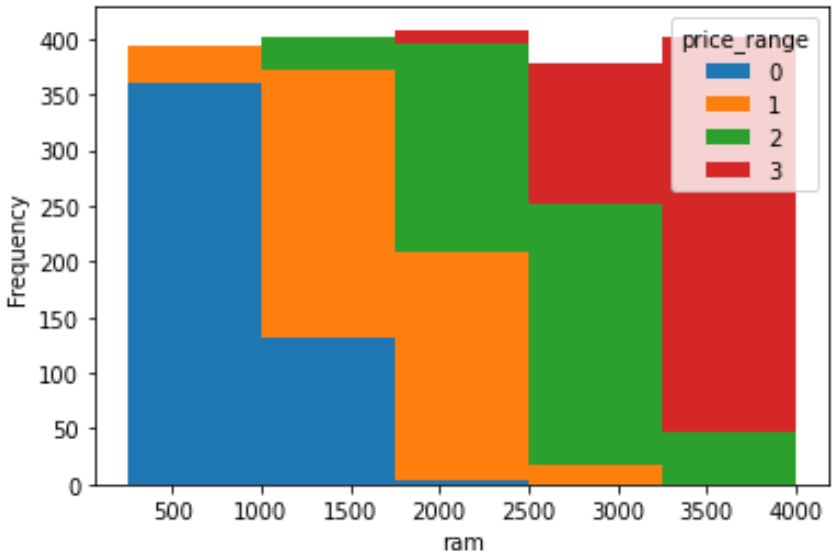
عرفان کرمی – ۹۸۲۲۲۰۷۹

BLUE

با رسم نمودار هیستوگرام بهتر متوجه میشویم که آبی بودن یا نبود از فیچر های بی تاثیر بر روی قیمت تلفن همراه است و میتوان آنرا از دیتاست حذف کرد.



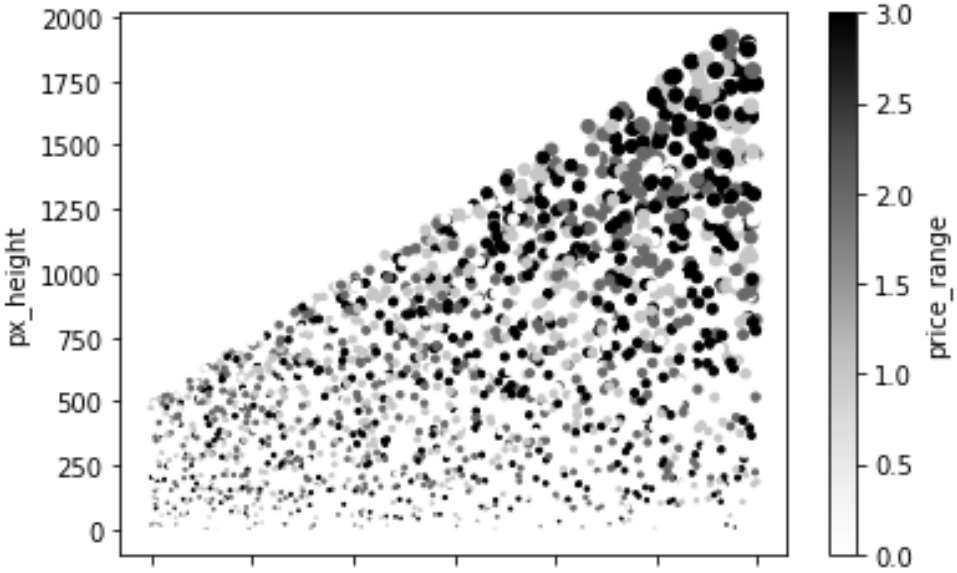
RAM



این نمودار و نیز ماتریس همبستگی به خوبی نشان میدهد که ram یکی از مهمترین ویژگی هایی است که میتواند در پیشبینی بازه قیمت به خوبی مورد استفاده قرار بگیرد.

در ادامه سعی میکنیم با ترکیب کردن برخی فیچرها، فیچرهایی را تولید کنیم که همبستگی بیشتری با تارگت داشته باشند.

حاصلضرب px_height, px_height

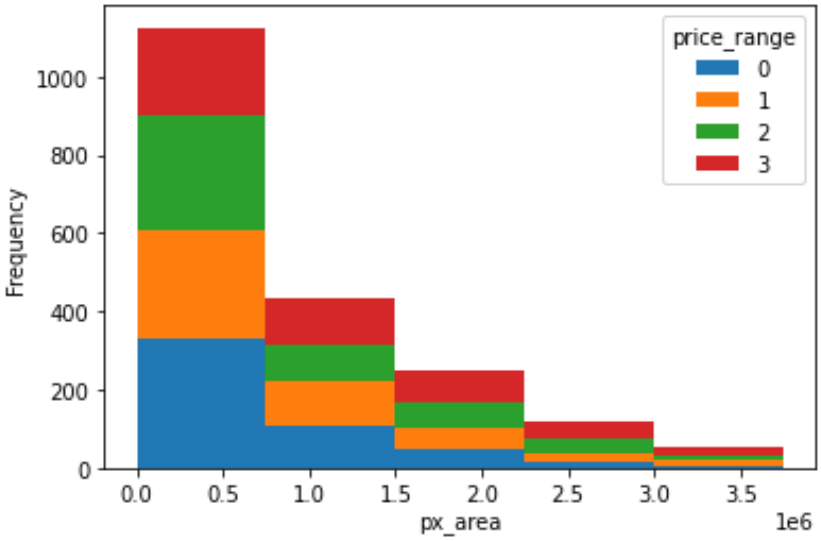


در این شکل به خوبی مشهود است که هرچه این حاصلضرب بیشتر شود رنج قیمتی افزایش می یابد. این فیچر جدید را **px_area** مینامیم.

اگر به کوریلیشن ها دقت کنیم میبینیم که ترکیب کردن این دو ویژگی سبب تولید ویژگی ای شده که همبستگی بیشتری با تارگت دارد.

```
px_height     0.144277
px_width      0.165132
px_area       0.172349
```

این را به صورت بصری هم میتوانیم ببینیم.



با کمی دقت میتوانیم متوجه شویم نسبت موبایل هایی با تارگت ۳ به موبایل هایی با تارگت ۰ به مرور افزایش می یابد.

حاصلضرب sc_h, sc_w

این فیچر جدید را area مینامیم.

```
sc_h    0.025641
sc_w    0.038076
area    0.040990
```

این فیچر کمی همبستگی بیشتری دارد و میتوانیم از آن به جای دو فیچر اولیه استفاده کنیم.

بررسی مقدار همبستگی و شکل هیستوگرام داده ها نکاتی را درباره برخی ویژگی ها به ما ارائه میدهد برخی از این اطلاعات و دید ها را در بخش بعد در قالب آزمون های آماری بررسی میکنیم.

۳. در این بخش ۵ آزمون آماری را مطرح و آنها را تفسیر میکنیم.

آزمون اول : از بخش قبل به نظر می رسد که میانگین کیفیت دوربین اصلی از دوربین جلو بیشتر باشد.پس ما سعی میکنیم فرض خود را رد کنیم پس فرض صفر را اینگونه قرار میدهیم که میانگین کیفیت آنها با هم برابر است یعنی

گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین
عرفان کرمی – ۹۸۲۲۲۰۷۹

$H_0: \mu_{fc} = \mu_{pc}$
$H_1: \mu_{fc} \neq \mu_{pc}$

با استفاده از آزمون Paired Student's t-test این فرضیات را مورد ارزیابی قرار میدهیم.

```
stat=53.885, p=0.000
reject null hypothesis
```

نتیجه آزمون فرض نشان میدهد فرض صفر رد میشود و همچنین با توجه به مقدار satat این یعنی حدس نخست ما مبنی بر بیشتر بودن میانگین کیفیت دوربین جلو بیشتر بوده است.

آزمون دوم: با توجه به قسمت قبل حدس میزنیم که آبی بودن یا نبودن موبایل و بازه قیمتی مستقل اند. پس فرض صفر را مبنی بر وابسته بودن این دو متغیر قرار میدهیم.

$H_0 =$ the two samples are independent
$H_1 =$ the two samples are dependent

با استفاده از آزمون Chi-Squared Test این فرضیات را مورد ارزیابی قرار میدهیم.

```
stat=0.948, p=0.814
Probably independent
```

نتیجه آزمون فرض نشان میدهد که حدس اولیه ای که داشتیم صحیح است و فرض صفر رد میشود پس آبی بودن یا نبودن موبایل واقعا تاثیری بر روی بازه قیمتی آن ندارد و با اطمینان خاطر میتوان ستون مربوط به آنرا از دیتاست حذف کرد.

آزمون سوم: با توجه به شکل نمودار هیستوگرام حدس میزنیم که فیچر battery_power دارای توزیع نرمال نباشد. این موضوع را با یک آزمون فرض آماری بررسی میکنیم. پس فرض ها را به صورت زیر قرار میدهیم.

$H_0 =$ the $battery\ power\ feature\ has\ gaussian\ distribution$
$H_1 =$ the $battery\ power\ feature\ has\ not\ gaussian\ distribution$

برای بررسی از آزمون Shapiro-Wilk Test استفاده میکنیم.

```
stat=0.952, p=0.000
Probably not Gaussian
```

نتیجه آزمون فرض نشان میدهید که حدس ما درست بوده است و فرض صفر رد میشود و توزیع فیچر battery_power غیر نرمال است.

آزمون چهارم: با توجه به نمودار هیستوگرام حدس میزنیم که میانگین برای جامعه داده های فیچر px_width برابر 190 px باشد پس با استفاده از یک آزمون فرض این مورد را بررسی میکنیم.

$H_0: \mu_{px\_width} = 190$
$H_1: \mu_{px\_width} \neq 190$

این موضوع را با استفاده از آزمون T-test انجام میدهیم.

```
t stat : 108.21499277595966 , p_value : 0.0
reject null hypothesis
```

این نشان میدهد حدس ما نادرست بوده است.

آزمون پنجم: با توجه به نمودار هیستوگرام به نظر میرسد که توزیع آماری دو فیچر px_width و px_height یکسان نیست پس با طرح یک آزون فرض این موضوع را بررسی میکنیم.

$H_0 =$ the $px\ width\ \ and\ px\ height\ features\ have\ same\ distribution$
$H_1 =$ the $px\ width\ \ and\ px\ height\ features\ have\ same\ distribution$

این موضوع را با استفاده از آزمون Kruskal-Wallis H Test بررسی میکنیم.

```
stat=1315.482, p=0.000
Probably different distributions
```

بررسی خروجی نشان میدهد حدس ما درست بوده و فرض صفر رد میشود پس میتوان گفت این دو فیچر دارای توزیع های آماری متفاوتی هستند.

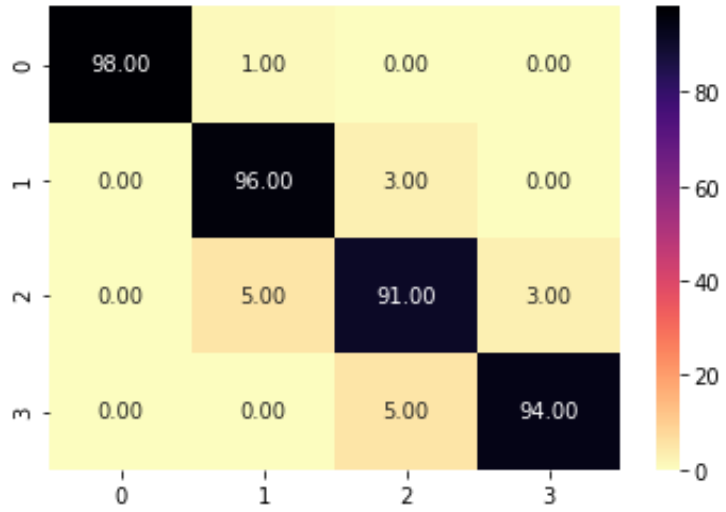
۴. در این مرحله قبل از هر چیز با استفاده از نمونه گیری طبقه ای داده هایمان را به دو قسمت train و test تقسیم بندی کرده و سپس از تعدادی از مدل های مرسوم برای کلاسبندی استفاده میکنیم. این مدل ها عبارتند از :

$$Logistic\ regression, SVM, DecisionTree, Random\ forest, MLP$$

کارایی هر کدام بر روی داده های تست به تفکیک آورده خواهد شد.

رگرسیون لجستیک

شکل زیر ماتریس درهمریختگی را برای مدل بر روی داده های تست نشان میدهد. از شکل به خوبی مشخص است که مدل کارایی نسبتا خوبی را بر روی داده های تست داشته است.



حال معیار های مختلف را برای این ماتریس به دست می آوریم.

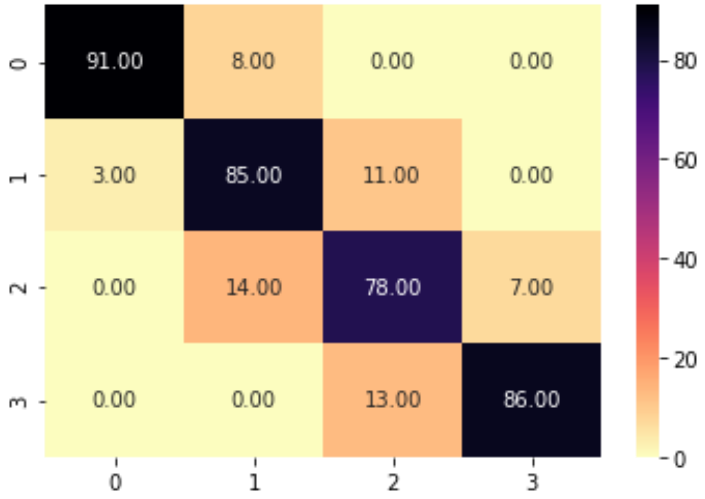
	precision	recall	f1-score	support
0	1.00	0.99	0.99	99
1	0.94	0.97	0.96	99
2	0.92	0.92	0.92	99
3	0.97	0.95	0.96	99
accuracy			0.96	396
macro avg	0.96	0.96	0.96	396
weighted avg	0.96	0.96	0.96	396

این داده ها به صورت کمی نشان میدهند که عملکرد مدل بر روی داده های تست نسبتا خوب بوده است و موفق شده ایم مدل مناسبی را آموزش دهیم. همچنین عملکرد مدل بر روی کلاس های مختلف هم تقریبا یکسان بوده است. حدس زده میشود که دلیل این امر توازن بین دادهای train و test و عدم سوگیری داده ها باشد که علت آن استفاده از نمونه گیری طبقه ای تصادفی در تقسیم داده ها است.(گزارش این توازن تقسیم در نوتبوک آمده است)

این مدل از رگرسیون لجستیک که در پکیج سایکیت لرن وجود دارد از رویکرد OvA استفاده میکند.

درخت تصمیم

شکل زیر ماتریس درهمریختگی را برای مدل بر روی داده های تست نشان میدهد. از شکل به خوبی مشخص است که کارایی مدل نسبت به مدل رگرسیون لجستیک به طور ملموسی کاهش یافته است علاوه بر آن کارایی مدل بر روی کلاس های مختلف هم تفاوت قابل توجهی را نشان میدهد که البته دلیلش را نمیدانم!



حال معیار های مختلف را برای این ماتریس به دست می آوریم.

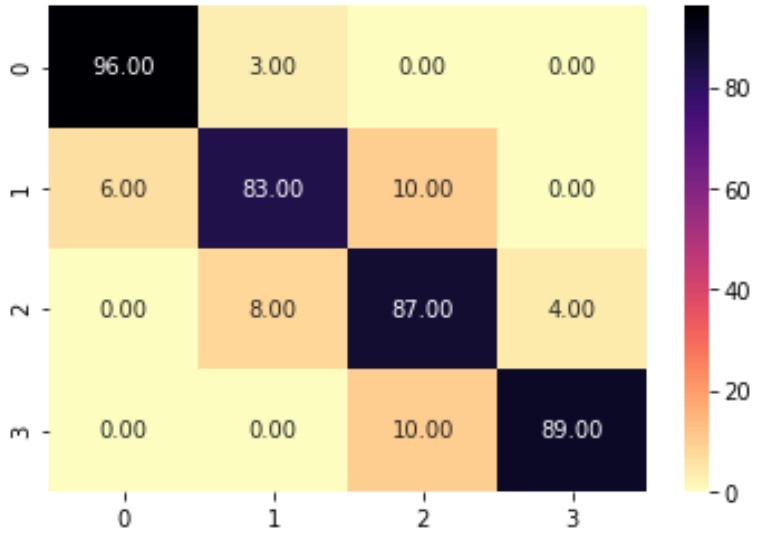
	precision	recall	f1-score	support
0	0.97	0.92	0.94	99
1	0.79	0.86	0.83	99
2	0.76	0.79	0.78	99
3	0.92	0.87	0.90	99
accuracy			0.86	396
macro avg	0.86	0.86	0.86	396
weighted avg	0.86	0.86	0.86	396

همانطور که مشخص است دقت مدل ۸۶ درصد است که افت حدود ۱۰ درصدی را نسبت به رگرسیون لجستیک نشان میدهد.

باید ذکر کنیم که درخت تصمیم به طور عادی از کلاسبندی چندکلاسه پشتیبانی میکند و نیازی به استفاده از رویکرد های OvO یا OvA نیست.

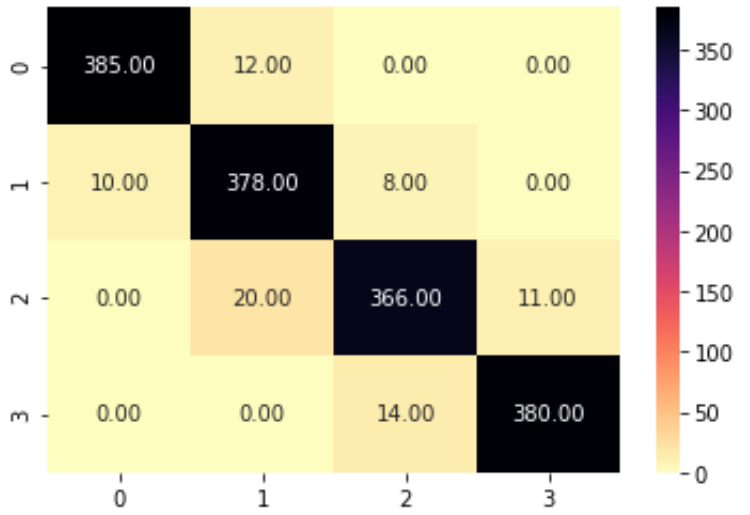
جنگل تصادفی

در جنگل تصادفی از تجمیع و voting دسته ای از درخت های تصمیم استفاده میشود و طبعا مانند آن از کلاسبندی چندکلاسه پشتیبانی میکند. در شکل زیر ماتریس درهمریختگی برای مدل آموزش دیده آن بر روی داده های تست آمده است:

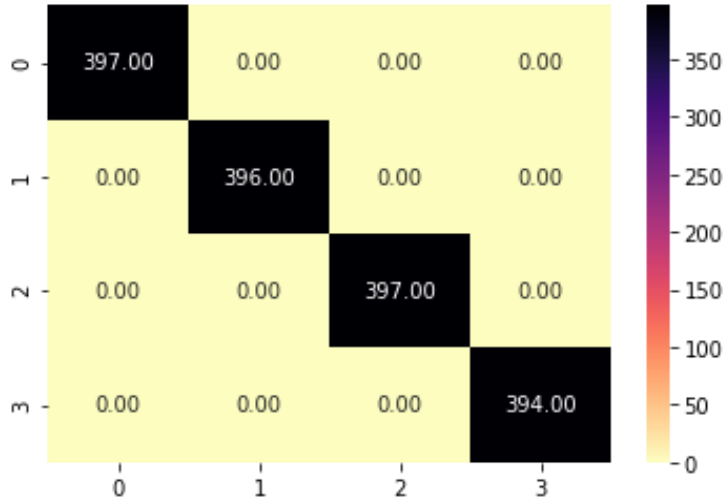


همانطور که مشهود است عملکرد آن بر روی داده های تست کمی از عمکرد درخت تصمیم بدتر بوده است .البته دلیل آن اورفیت شدن بیشتر جنگل تصادفی به دلیل پیچیده تر بودن مدل است. اگر عملکرد این دو را بر روی داده های train درنظر بگیریم این مورد به خوبی مشهود خواهد بود.

شکل زیر عملکرد درخت تصمیم بر روی داده های train است.



و شکل زیر هم عملکرد جنگل تصادفی را بر روی داده های train نشان میدهد.



اورفیت شدن مدل برای جنگل تصادفی نسبت به درخت تصمیم کاملا مشهود است.

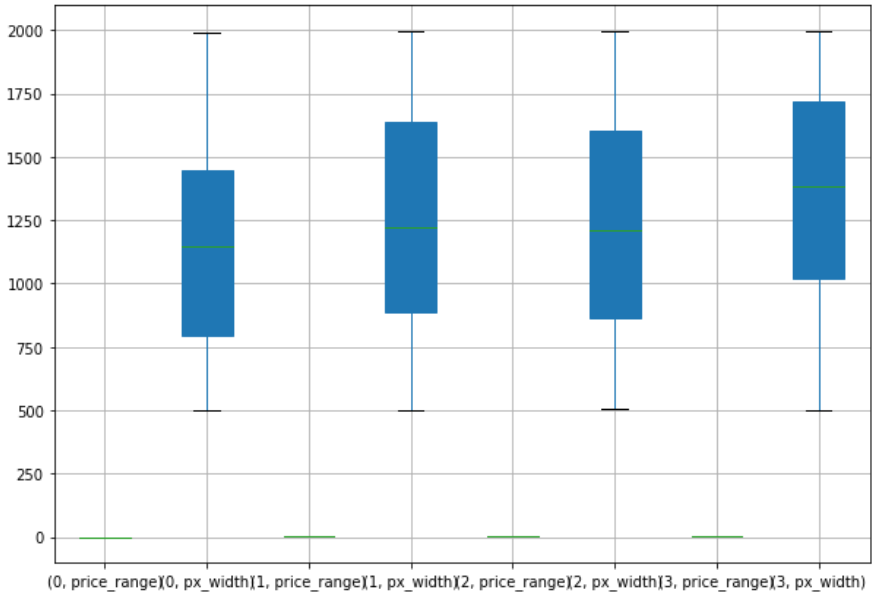
در زیر گزارش کلی از عملکرد مدل آمده است.

	precision	recall	f1-score	support
0	0.94	0.98	0.96	99
1	0.83	0.82	0.82	99
2	0.77	0.81	0.79	99
3	0.96	0.88	0.92	99
accuracy			0.87	396
macro avg	0.87	0.87	0.87	396
weighted avg	0.87	0.87	0.87	396

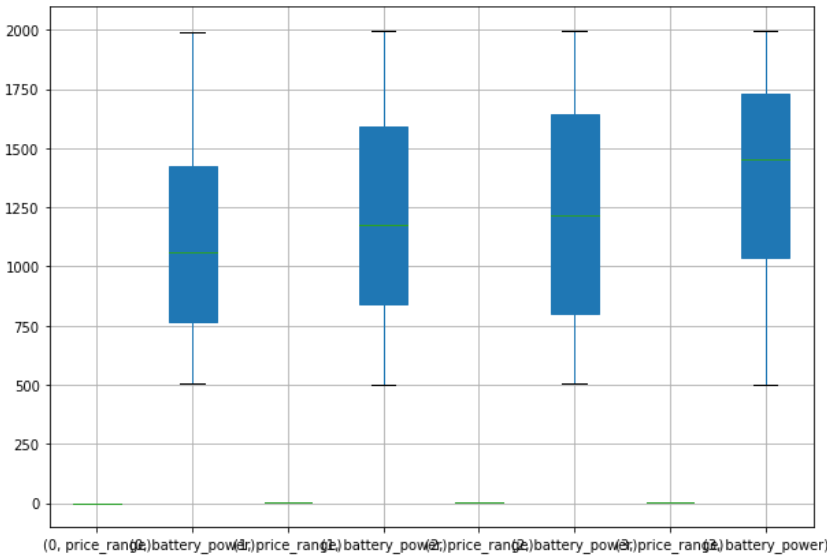
میتوانیم ببینیم که عملکرد جنگل تصادفی بر روی داده های کلاس ۱ و ۲ به طور ملموسی از داده های کلاس ۰و۴ بدتر است که البته دلیل آنرا نمیدانم.

۵. در قسمت قبل موارد خواسته شده در این بخش آورده شد. البته یکی از دلایلی که احتمالا میتوان برای عملکرد نسبتا بد تر مدل ها بر روی داده های کلاس ۱و۲ آورد این است که به دلیل نحوه توزیع این لیبل ها جدا کردن و تمایز آنها سخت تر است مثلا نمودار جعبه ای تفکیک شده بر حسب لیبل ها برای فیچر px_width را میتوانیم ببینیم:

گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین
عرفان کرمی – ۹۸۲۲۲۰۷۹

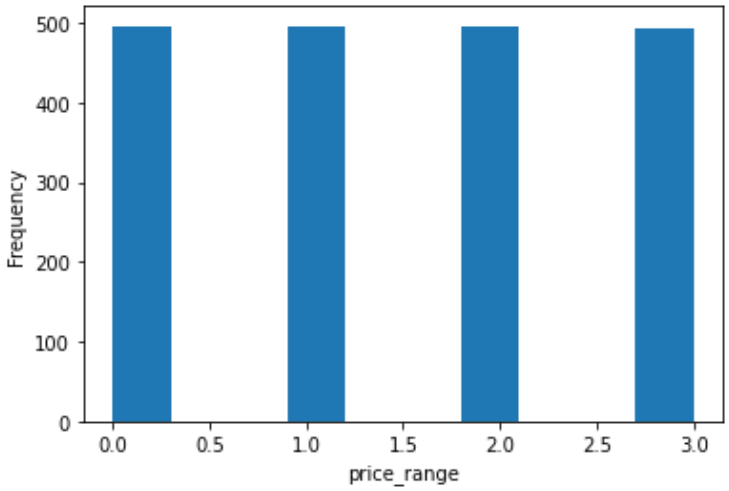


از شکل به خوبی مشخص است داده های کلاس ۱و۲ تقریبا بازه های منحصر به فردی ندارند و کاملا مشترک هستند همین کار را برای battery_power دیگر هم انجام میدهیم.



چیزی که در قسمت قبل گفتیم برای این فیچر هم تا حد زیادی صدق میکند و همین جداسازی این دو کلاس را نسبت به دو تای دیگر مشکل خواهد کرد.

۶. بله داده ها متوازن میباشند این را هم در داده های اصلی میتوانیم نشان دهیم و هم با استفاده از نمونه گیری طبقه ای داده ها به گونه ای تقسیم شده اند که داده های train و test هم متوازن باشند.



نمودار هیستوگرام متوازن بودن داده ها را در دیتاست اصلی نشان میدهد.

در صورتی که داده ها متوازن نباشند سه راه حل اصلی که میتوان استفاده کرد عبارتند از:

- حذف تعدادی از داده ها به گونه ای که داده های باقی مانده متوازن شوند
- استفاده از تکرار داده ها برای کلاسی که دارای داده کمتر است
- استفاده از شبکه های مولد تخاصمی به منظور ساخت داده های جدید و برقراری توازن میان داده ها

۷. برای این قسمت ما از ۲ ، scaler مختلف استفاده و نتایج را بررسی میکنیم.

این دو عبارتند از min_max scaler و standard scaler

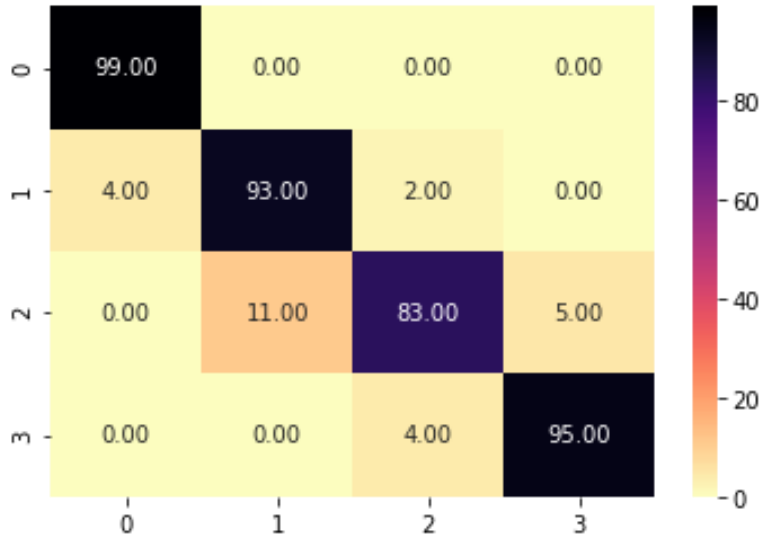
نتایج به دست آمده با استفاده از رگرسیون لجستیک

در دو شکل زیر میتوانید نتیجه رگرسیون لجستیک با استفاده از این دو scaler مختلف را ببینید:

Min_max scaler

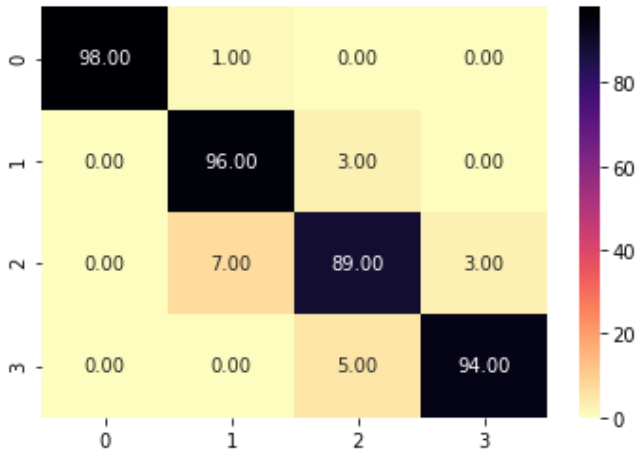
گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین

عرفان کرمی – ۹۸۲۲۲۰۷۹



	precision	recall	f1-score	support
0	0.96	1.00	0.98	99
1	0.89	0.94	0.92	99
2	0.93	0.84	0.88	99
3	0.95	0.96	0.95	99
accuracy			0.93	396
macro avg	0.93	0.93	0.93	396
weighted avg	0.93	0.93	0.93	396

standard scaler



	precision	recall	f1-score	support
0	1.00	0.99	0.99	99
1	0.92	0.97	0.95	99
2	0.92	0.90	0.91	99
3	0.97	0.95	0.96	99
accuracy			0.95	396
macro avg	0.95	0.95	0.95	396
weighted avg	0.95	0.95	0.95	396

به وضوح مشخص است که نتیجه هر دو از نتیجه بر روی داده اصلی بدترند اما مدلی که با استفاده از داده ای که بر روی آن standard scaler اعمال شده بود از دیگری بهتر بود. اما نکته ای که وجود دارد این است که زمان train برای این داده های اسکیل شده بسیار کمتر از حالت ترین کردن با داده های اصلی بود.

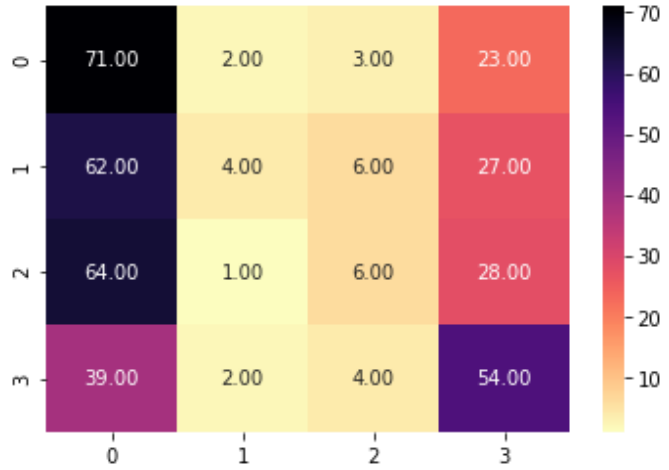
** نتایج درخت تصمیم با استفاده از داده های اسکیل شده همانند درخت تصمیم در حالت کار با داده های اصلی بود. جزئیات در نوتبوک آمده است

۸. داده ها از همان ابتدا به نسبت ۲۰-۸۰ جدا شده بودند و تمام نتایج بر روی داده های تست گزارش شده اند.

۹. با توجه به اینکه برای pov برابر با ۰,۹۹ فقط یک ستون را تولید می کند، بنابراین برای pov های کوچکتر نتیجه یکسان خواهد بود و خروجی ۱ ستون خواهد بود بنابراین نیازی به بررسی جداگانه آنها نیست.

پس این را بررسی میکنیم:

گزارش کار دیتاست ۱ – پروژه اول درس یادگیری ماشین
عرفان کرمی – ۹۸۲۲۲۰۷۹

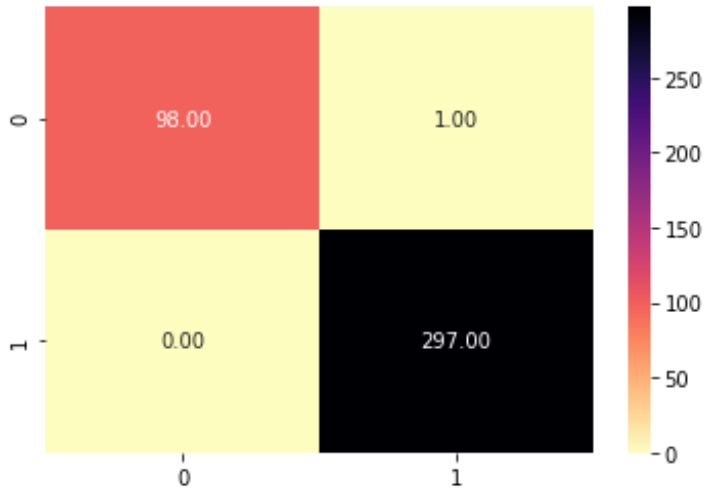


شکل به وضوح نشان میدهد که عملکرد به شدت افت پیدا کرده است. و دقت به حدود ۳۳ درصد رسیده است. اما دلیل اصلی که از pca استفاده میشود این است که اولا pca میتواند به عنوان یک متد استخراج ویژگی مورد استفاده قرار بگیرد و از طرفی خیلی از اوقات تعداد فیچر ها خیلی زیاد است (مخصوصا در پروژه های پردازش متن و پردازش تصویر) در نتیجه استفاده از این تکنیک میتواند تعداد فیچر ها و در نتیجه زمان ترین و مصرف حافظه را کاهش دهد.

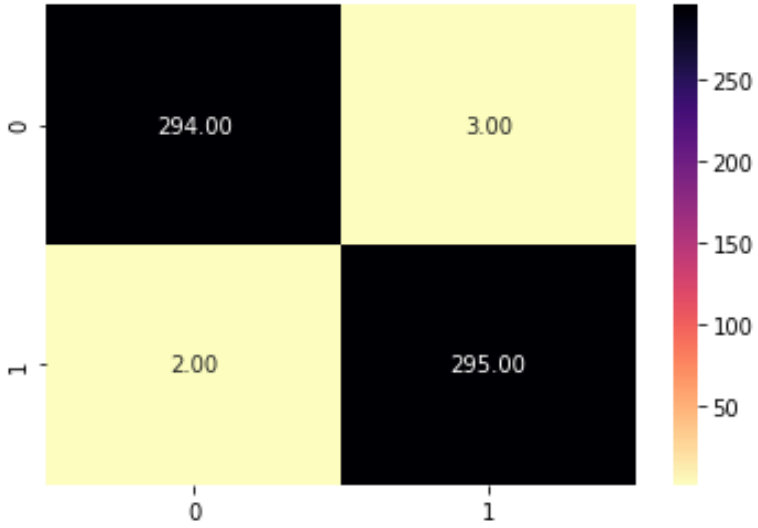
مثلا در همین مثال اگر pov را قرار دهیم ۰,۹۹۹۹۹۹۹ آنگاه دقت به ۰,۹۴ درصد میرسد و ۴ ستون استخراج میشود پس مصرف حافظه و زمان پردازش به طور چشمگیر کاهش می باید اما دقت تا حد خوبی حفظ شده است.

۱۰.حال لیبل ها را تغییر و مدل رگرسیون لجیتسک را با استفاده از آنها ترین میکنیم و نتایج را بررسی میکنیم:

در کمال تعجب میبینیم که نتایج بسیار بهتر شده اند . حدس میزنم دلیل این است که کلاس های ۱و۲ خیلی سخت تر تفکیک میشدند حال با کلاس ۳ ترکیب و از کلاس ۰ که از بقیه راحت ترتفکیک میشد جدا شده اند.



اگر دقت را محاسبه کنیم به عدد ۹۹ درصد میرسیم که دقت بالایی است. حال با تکنیک اورسمپلینگ دیتاست را متعادل میکنیم.



که اگر محاسبه کنیم همچنان دقت ۹۹ درصد بر روی داده های تست وجود دارد.