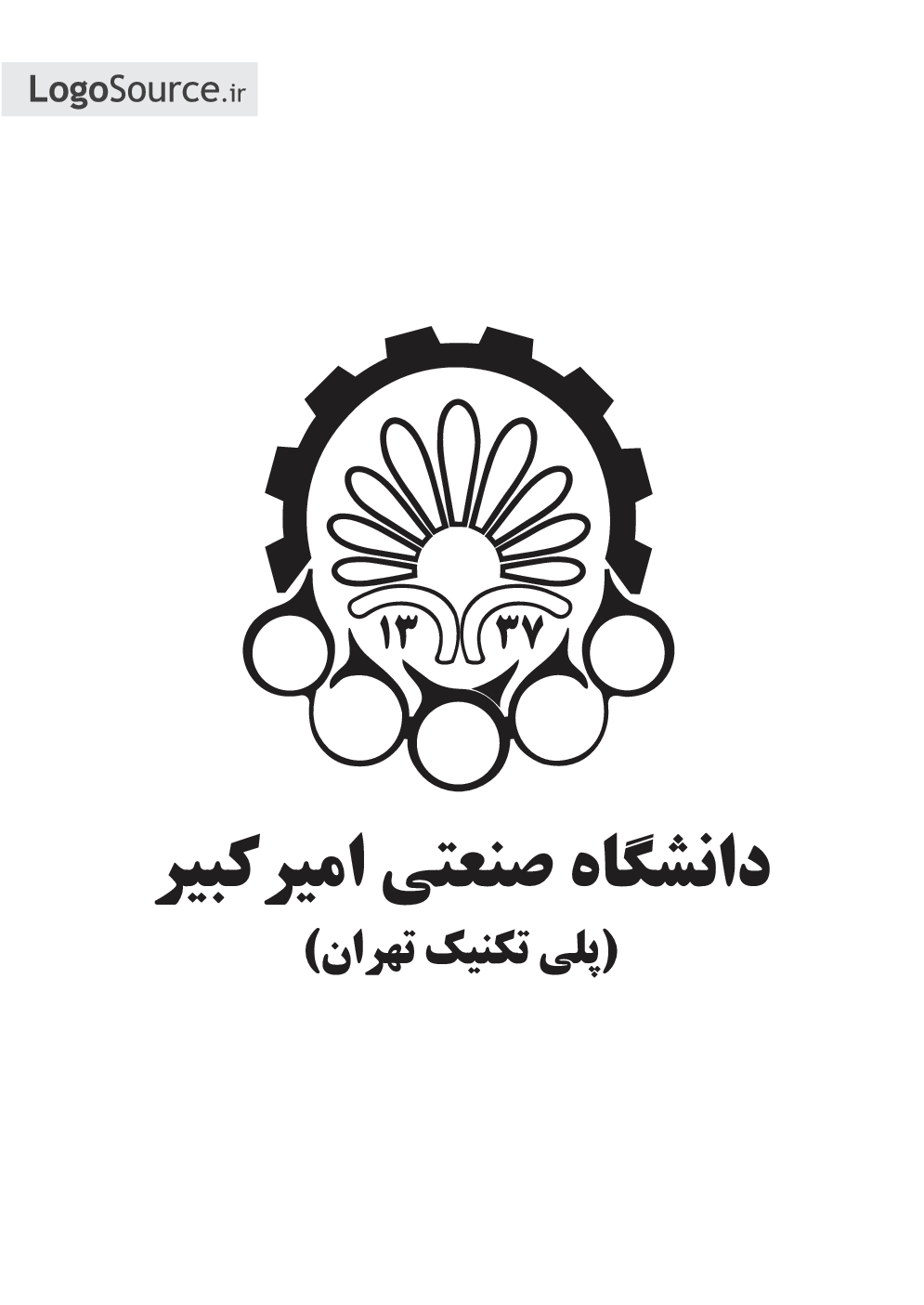
به نام کیمیاگر عالم



**یادگیری ماشین کاربردی**

عنوان

**تمرین اول (HW01)**

مدرس

**دکتر احسان ناظر فرد**

دانشجو

**امیرحسین بابائیان**

401131002

ترم بهار 02-01

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

# فهرست

[فهرست 2](#_Toc129544304)

[فهرست تصاویر 5](#_Toc129544305)

[سوال اول 7](#_Toc129544306)

[بخش a 7](#_Toc129544307)

[بخش b 7](#_Toc129544308)

[بخش c 7](#_Toc129544309)

[بخش d 7](#_Toc129544310)

[بخش e 8](#_Toc129544311)

[بخش f 8](#_Toc129544312)

[بخش g 9](#_Toc129544313)

[بخش h 9](#_Toc129544314)

[بخش i 10](#_Toc129544315)

[بخش j 11](#_Toc129544316)

[بخش k 11](#_Toc129544317)

[بخش l 12](#_Toc129544318)

[بخش m 12](#_Toc129544319)

[بخش n 12](#_Toc129544320)

[سوال دوم 13](#_Toc129544321)

[بخش a 13](#_Toc129544322)

[بخش b 13](#_Toc129544323)

[نمودار barplot 13](#_Toc129544324)

[نمودار violinplot 14](#_Toc129544325)

[نمودار histplot 14](#_Toc129544326)

[نمودار pieplot 14](#_Toc129544327)

[نمودار sunburst 15](#_Toc129544328)

[دیگر نمودار ها 15](#_Toc129544329)

[بخش c و d 15](#_Toc129544330)

[قدم اول – حذف ستون و بدست آوردن اطلاعات مفید 15](#_Toc129544331)

[قدم دوم – تبدیل رشته های KG و GB 16](#_Toc129544332)

[قدم سوم - labelencoder 16](#_Toc129544333)

[قدم چهارم – تقسیم کردن داده ها 17](#_Toc129544334)

[بخش e 17](#_Toc129544335)

[بخش f 18](#_Toc129544336)

[بخش g 18](#_Toc129544337)

[سوال سوم 19](#_Toc129544338)

[توضیحات دیتاست 19](#_Toc129544339)

[بخش a 20](#_Toc129544340)

[بخش b 22](#_Toc129544341)

[بخش c 22](#_Toc129544342)

[بخش d 23](#_Toc129544343)

[Ridge 23](#_Toc129544344)

[Lasso 23](#_Toc129544345)

[Elastic 23](#_Toc129544346)

[بخش e 23](#_Toc129544347)

[بخش f 24](#_Toc129544348)

[سوال چهارم 24](#_Toc129544349)

[بخش a 24](#_Toc129544350)

[بخش b 25](#_Toc129544351)

[بخش c 25](#_Toc129544352)

[بخش d 26](#_Toc129544353)

[بخش e 26](#_Toc129544354)

[سوال پنجم 26](#_Toc129544355)

[بخش a 27](#_Toc129544356)

[بخش b 27](#_Toc129544357)

[بخش c 28](#_Toc129544358)

# فهرست تصاویر

[تصویر 1) خروجی قطعه کد پاسخ بخش h سوال 1 10](#_Toc129544359)

[تصویر 2) خروجی 10 داده تصادفی 13](#_Toc129544360)

[تصویر 3) نمودار barplot 14](#_Toc129544361)

[تصویر 4) نمودار violonplot 14](#_Toc129544362)

[تصویر 5) نمودار histogram 14](#_Toc129544363)

[تصویر 6) نمودار pieplot 15](#_Toc129544364)

[تصویر 7) نمودار sunburst 15](#_Toc129544365)

[تصویر 8) خروجی پس از حذف ستون laptop\_ID 15](#_Toc129544366)

[تصویر 9) خروجی اطلاعات ویژگی ها 16](#_Toc129544367)

[تصویر 10) حذف رشته های زائد 16](#_Toc129544368)

[تصویر 11) خروجی پس از labelEncoding 17](#_Toc129544369)

[تصویر 12) داده های تست 17](#_Toc129544370)

[تصویر 13) خروجی MSE با درجه های مختلف 18](#_Toc129544371)

[تصویر 14) میانگین و انحراف از معیار MSE بر اساس تعداد همسابه ها 18](#_Toc129544372)

[تصویر 15) خروجی MSE، R2 و model score 18](#_Toc129544373)

[تصویر 16) جدول Hitmap 19](#_Toc129544374)

[تصویر 17) نمودار توزیع بر حسب سه ویژگی اول 20](#_Toc129544375)

[تصویر 18) نمودار توزیع بر حسب سه ویژگی دوم 20](#_Toc129544376)

[تصویر 19) نمودار توزیع بر حسب سه ویژگی سوم 20](#_Toc129544377)

[تصویر 20) نمودار هیت مپ تمامی ویژگی ها 21](#_Toc129544378)

[تصویر 21) نمودار ارتباط بین دما و رطوبت 21](#_Toc129544379)

[تصویر 22) نمودار میزان تغییر 4 ویژگی از ساعت 18 الی 23 21](#_Toc129544380)

[تصویر 23) مجموعه داده ها پس از تغییرات اعمال شده 22](#_Toc129544381)

[تصویر 24) میزان Correlation 22](#_Toc129544382)

[تصویر 25) داده های تست پس از اعمال StandardScaler 22](#_Toc129544383)

[تصویر 26) خروجی نمونه پس از جداسازی داده آموزش و تست 22](#_Toc129544384)

[تصویر 27) مقدار لیست آلفا 23](#_Toc129544385)

[تصویر 28) مقدار لیست l1\_ratios 23](#_Toc129544386)

[تصویر 29) خروجی پس از آموزش و تست در Ridge 23](#_Toc129544387)

[تصویر 30) خروجی پس از آموزش و تست در Lasso 23](#_Toc129544388)

[تصویر 31) خروجی پس از آموزش و تست در Elastic 23](#_Toc129544389)

[تصویر 32) نتیجه اعمال معیار های R2 و RMSE 23](#_Toc129544390)

[تصویر 33) اضافه کردن دیتاست digits 24](#_Toc129544391)

[تصویر 34) تقسیم بندی داده ها به داده آموزش و تست 25](#_Toc129544392)

[تصویر 35) داده های تست پس از اعمال min max scaler 25](#_Toc129544393)

[تصویر 36) خروجی Classification\_Report پس از آموزش و مدل سازی با 5 همسایه 25](#_Toc129544394)

[تصویر 37) ماتریس Confusion 25](#_Toc129544395)

[تصویر 38) خروجی Score در k=10 روش KFold 26](#_Toc129544396)

[تصویر 39) نمودار نسبت Accuracy نسبت به K بدون KFold 26](#_Toc129544397)

[تصویر 40) نمودار نسبت Accuracy به K با استفاده از KFold 26](#_Toc129544398)

[تصویر 41) بارگیری دیتاست iris از sklearn.datasets 27](#_Toc129544399)

[تصویر 42) جداسازی داده های آموزش و تست 80/20 27](#_Toc129544400)

[تصویر 43) خروجی داده های آموزش پس از استاندارد سازی 27](#_Toc129544401)

[تصویر 44) نمودار صحت در حالت خواسته شده 28](#_Toc129544402)

[تصویر 45) نمودار صحت پس از تغییر نسبت داده های تست و آموزش 28](#_Toc129544403)

[تصویر 46) تصویری از کد به همراه پیش بینی 28](#_Toc129544404)

# سوال اول

## بخش a

بله، می‌توان از الگوریتم K-Nearest Neighbors (KNN) برای مسئله دسته‌بندی چند کلاسه استفاده کرد. در این الگوریتم، با استفاده از سیستم رأی‌گیری اکثریتی بین k نزدیک‌ترین همسایه، کلاس نمونه ورودی تعیین می‌شود. به عبارت دیگر، با شناسایی k نزدیک‌ترین همسایه به نمونه ورودی و انجام رأی‌گیری برای تعیین کلاس، می‌توان برای مسائل دسته‌بندی چند کلاسه از KNN استفاده کرد.

## بخش b

Curse of dimensionality به این معنی است که با افزایش تعداد ابعاد داده‌ها، فضای داده‌ها به طور چشمگیری بزرگتر می‌شود. در الگوریتم KNN، عملکرد آن به طور مستقیم با تعداد ابعاد داده‌ها مرتبط است. افزایش تعداد ابعاد داده‌ها باعث می‌شود که نقاط به هم دور شده و در نتیجه کارایی الگوریتم KNN کاهش می‌یابد. این به این معنی است که با افزایش تعداد ابعاد، پیدا کردن همسایه‌های مناسب برای هر نقطه بسیار دشوارتر می‌شود. به عبارت دیگر، بالا رفتن تعداد ابعاد باعث کاهش دقت و کارایی الگوریتم KNN می‌شود.

## بخش c

اصولاً KNN برای وظایف یادگیری با ناظر استفاده می‌شود ولی می‌توان از آن در برخی وظایف یادگیری بدون ناظر نیز استفاده کرد. در این حالت، KNN معمولاً به عنوان یک الگوریتم خوشه‌بندی استفاده می‌شود. بدین منظور، ابتدا نزدیک‌ترین همسایه‌های هر نقطه را پیدا کرده و سپس با استفاده از این اطلاعات، نقاطی که به هم نزدیک هستند را در یک خوشه قرار می‌دهیم. به این ترتیب، می‌توانیم داده‌ها را بر اساس شباهت به یکدیگر گروه‌بندی کنیم، استفاده از KNN در وظایف یادگیری بدون ناظر به دلیل سادگی و کارایی آن، مخصوصاً برای داده‌هایی با تعداد بالا، بسیار مناسب است.

## بخش d

KNN الگوریتمی برای دسته‌بندی داده‌ها است که برای پیش‌بینی برچسب داده‌های جدید از نزدیک‌ترین همسایه‌های آن‌ها استفاده می‌کند. در مقابل، الگوریتم‌های دسته‌بندی دیگری مانند Naive Bayes و درخت تصمیم‌گیری، با استفاده از الگوهای آماری و احتمالاتی برچسب داده‌های جدید را پیش‌بینی می‌کنند.

در Naive Bayes، به دنبال پیدا کردن توزیع احتمال برای هر برچسب داده جدید هستیم، در حالی که در درخت تصمیم‌گیری، به دنبال یافتن یک راهبرد بهینه برای تقسیم داده‌ها به گروه‌های مختلف هستیم.

بنابراین، اصلی‌ترین تفاوت بین KNN و سایر الگوریتم‌های دسته‌بندی، در روشی است که برای پیش‌بینی برچسب داده‌های جدید استفاده می‌شود. در KNN از نزدیک‌ترین همسایه‌ها استفاده می‌شود، در حالی که در الگوریتم‌های دسته‌بندی دیگر از الگوهای آماری و احتمالاتی و یا روش‌های تقسیم داده‌ها استفاده می‌شود.

## بخش e

انتخاب معیار فاصله بر عملکرد الگوریتم KNN تأثیر زیادی دارد. از آنجایی که KNN بر پایه‌ی معیار فاصله کار می‌کند، انتخاب معیار مناسب به اهمیت بالایی برخوردار است.

معیار فاصله اقلیدسی معمولاً برای داده‌هایی که در فضای برداری هستند، مناسب است. این معیار بر اساس فاصله اقلیدسی بین دو نقطه در فضای n-بعدی محاسبه می‌شود.

معیار فاصله منهتن برای داده‌هایی که مختصاتشان با مقادیر نامتناهی یا بسیار بزرگ هستند، مفید است. در این معیار فاصله، فاصله بین دو نقطه برابر با جمع مقدار مطلق تفاضل بین مختصاتشان است.

معیار فاصله کسینوس برای داده‌هایی که مرتبط با زوایا و جهت هستند، مناسب است. در این معیار فاصله، فاصله بین دو نقطه برابر با کسینوس زاویه بین دو بردار است.

با توجه به مشخصات هر معیار فاصله، انتخاب معیار مناسب بسته به خصوصیات داده‌ها و مسئله‌ی مورد نظر در الگوریتم KNN بسیار مهم است. به عنوان مثال، اگر داده‌ها در فضای برداری هستند، معیار اقلیدسی مناسب است. اما اگر داده‌ها مرتبط با زوایا و جهت هستند، معیار کسینوس مناسب است.

## بخش f

فرایند انتخاب مقدار بهینه K در KNN به شرح زیر است:

1. جدول داده ها را به دو بخش تقسیم کنید: بخش آموزش و بخش تست.
2. از بخش آموزش استفاده کرده و مدل KNN را با مقادیر مختلف K آموزش دهید.
3. سپس با استفاده از بخش تست، عملکرد مدل را بررسی کنید و مقدار K بهینه را بر اساس عملکرد مدل در بخش تست انتخاب کنید.

متد های مختلفی برای انتخاب مقدار بهینه K وجود دارد که عبارتند از:

1. تعیین مقدار K بر اساس قوانین تجربی: در این روش، مقدار K به صورت دستی تعیین می شود و توسط کارشناسان مسئول انجام می شود. این روش ساده است اما ممکن است به دلیل عدم استفاده از داده های واقعی، بهینه نباشد.
2. تعیین مقدار K بر اساس تقسیم داده ها: در این روش، مقدار K بر اساس تعداد داده های آموزش و تست تقسیم بر دو تعیین می شود. این روش معمولاً به خوبی عمل می کند ولی ممکن است به صورت غیر قابل پیش بینی بهینه نباشد.
3. استفاده از روش کراس ولیدیشن: در این روش، داده ها به چند بخش تقسیم می شوند و مدل با استفاده از هر بخش به صورت مجزا آموزش داده می شود. سپس عملکرد مدل با استفاده از بخش تست بررسی می شود و مقدار K بهینه بر اساس عملکرد مدل بر روی بخش تست تعیین می شود. این روش معمولاً به خوبی عمل می کند و باعث افزایش دقت مدل می شود.

## بخش g

رگرسیون وزن‌دار محلی در الگوریتم KNN از رگرسیون KNN استاندارد به دو دلیل متمایز است. اولاً، تنها یک زیرمجموعه کوچکتر از همسایگان با وزن بیشتر برای پیش‌بینی استفاده می‌شود. دوماً، وزن‌ها بر اساس فاصله نسبی از نقطه داده به همسایگان آن محاسبه می‌شوند. از طرفی، رگرسیون KNN استاندارد تمام همسایگان را برای پیش‌بینی استفاده می‌کند و هیچ وزنی برای آن‌ها تعیین نمی‌شود.

مزیت استفاده از رگرسیون وزن‌دار محلی در الگوریتم KNN این است که دقت پیش‌بینی بهبود می‌یابد و الگوریتم قادر است به نواحی تخصصی نزدیک‌تر به هر نقطه توجه کند.

همچنین استفاده از وزن‌ها به این معنی است که نقاط داده‌ای که در فاصله نزدیک‌تری به نقطه تحت بررسی هستند، بیشترین تأثیر را در پیش‌بینی نقطه تحت بررسی دارند.

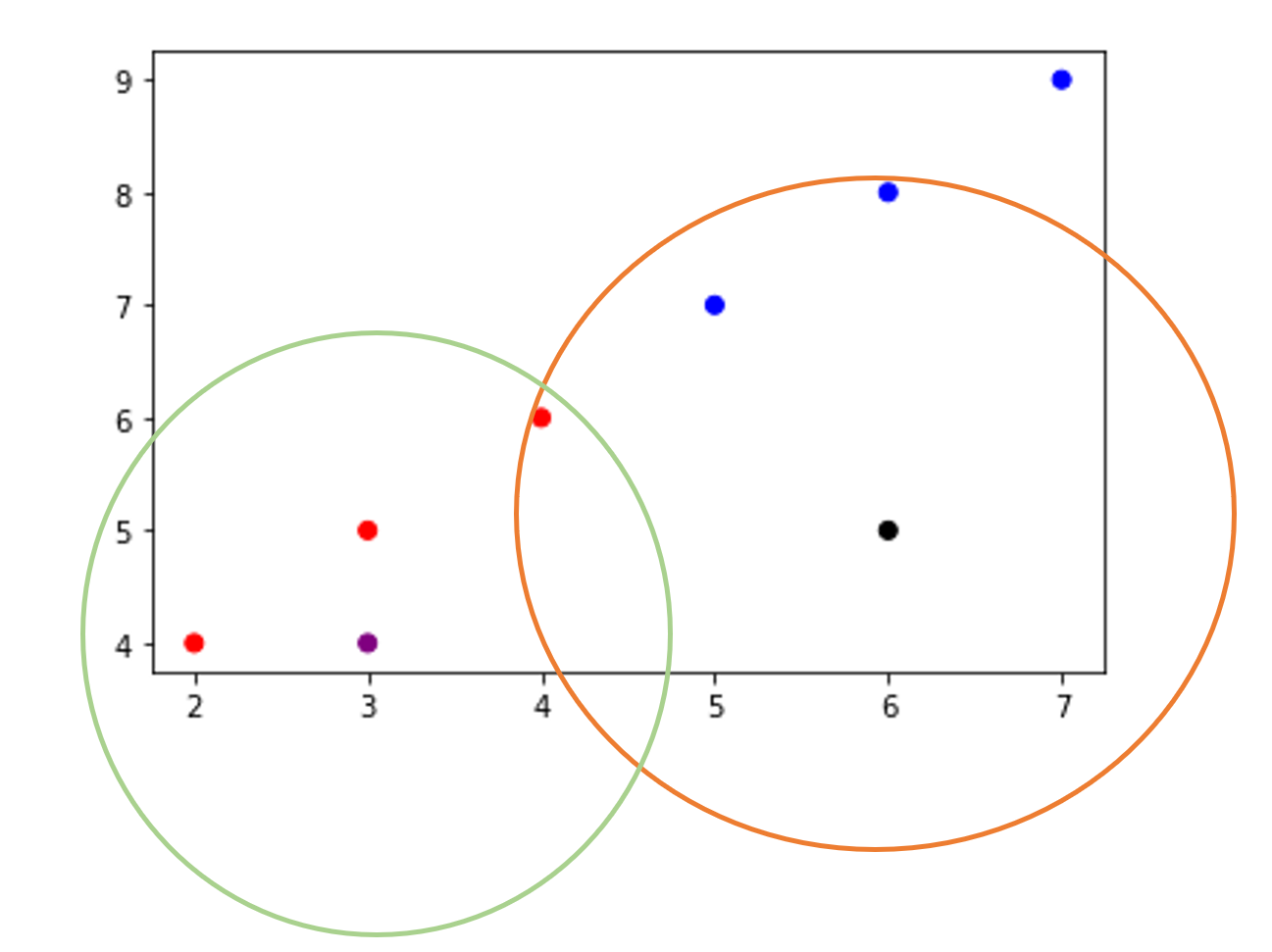
از معایب رگرسیون وزن‌دار محلی می‌توان به این نکته اشاره کرد که محاسبه وزن‌ها برای هر نقطه داده محاسبات بیشتری نیاز دارد و می‌تواند زمان پردازش را افزایش دهد. علاوه بر این، در برخی موارد، الگوریتم KNN با استفاده از تمام همسایگان نسبت به رگرسیون وزن‌دار محلی دقت بیشتری در پیش‌بینی داشته باشد.

برخی از کاربردهای رگرسیون وزن‌دار محلی در یادگیری ماشین شامل پیش‌بینی قیمت مسکن بر اساس ویژگی‌های مختلف، پیش‌بینی میزان آلودگی هوا بر اساس عوامل مختلف مانند دما و فشار هوا، تخمین بازدهی سهام بر اساس عوامل مختلف اقتصادی و تاریخی، پیش‌بینی خطر بیماری بر اساس عوامل مختلفی مانند سن، جنسیت و سابقه بیماری و ... است.

در کل، رگرسیون وزن‌دار محلی یک روش کارآمد برای پیش‌بینی دقیق‌تر در مواردی است که اطلاعات نزدیک به نقطه داده در نظر گرفته شده است و در مقایسه با رگرسیون KNN استاندارد، دارای دقت بالاتری می‌باشد.

## بخش h

برای نمایش نقاط روی صفحه یک قطعه کد نوشتم و تصویر مقابل را تولید کردم.



تصویر 1) خروجی قطعه کد پاسخ بخش h سوال 1

آبی رنگ ها دسته بندی B و قرمز رنگ ها دسته بندی A را نمایش می دهند هر کدام از نمونه های 7 و 8 به ترتیب بنفش و مشکی نمایش داده شده اند.

با در نظر گرفتن K=3 داریم که فاصله برای بنفش از کلاس A کمتر است پس در این دسته قرار می گیرید و همچنین برای مشکی نیز کلاس B فاصله کمتری دارد. (به ترتیب دایره های سبز و نارنجی گویا هستند.)

## بخش i

در یادگیری ماشین، پارامترهای مدل و هایپرپارامترهای الگوریتم یادگیری دو نوع پارامتر هستند که به طور کلی متفاوت عمل می‌کنند.

پارامترهای مدل مقادیری هستند که توسط الگوریتم یادگیری به‌صورت خودکار تنظیم می‌شوند و ارزش آن‌ها توسط داده‌های آموزش تعیین می‌شود. به‌عنوان مثال، در یک مدل خطی، ضرایب جمع و ضرب به‌صورت خودکار توسط الگوریتم تعیین می‌شوند و مقدار آن‌ها به‌صورت مستقیم توسط داده‌های آموزش تعیین می‌گردد.

هایپرپارامترها به‌عنوان پارامترهایی که در انتخاب الگوریتم یادگیری و تنظیم آن استفاده می‌شوند، تعریف می‌شوند. این پارامترها توسط کاربر مشخص می‌شوند و نمی‌توانند به‌صورت خودکار توسط الگوریتم تنظیم شوند. به‌عنوان مثال، در الگوریتم KNN، هایپرپارامترهایی مانند تعداد همسایگان و معیار فاصله مورد استفاده قرار می‌گیرند که باید توسط کاربر مشخص شوند.

از مزیت‌های پارامترهای مدل این است که مقادیر آن‌ها به‌صورت خودکار توسط الگوریتم یادگیری تنظیم می‌شوند و به‌صورت مستقیم به داده‌های آموزش بستگی دارند. از طرفی، هایپرپارامترها به‌عنوان پارامترهای کنترلی الگوریتم یادگیری عمل می‌کنند و به کاربر اجازه می‌دهند که بهترین مدل را از بین الگوریتم‌های یادگیری مختلف انتخاب کند.

بنابراین، پارامترهای مدل و هایپرپارامترها با یکدیگر تفاوت دیگری هم دارند که این تفاوت در تعیین مقدار آن‌هاست. پارامترهای مدل توسط الگوریتم یادگیری به‌صورت خودکار تنظیم می‌شوند و مقدار آن‌ها به‌صورت مستقیم توسط داده‌های آموزش تعیین می‌شود. اما هایپرپارامترها مقادیر ثابتی دارند و باید توسط کاربر به‌صورت دستی تعیین شوند.

استفاده از پارامترهای مدل باعث می‌شود که مدل به‌صورت خودکار به داده‌های آموزش تطبیق پیدا کند و دقت مدل در پیش‌بینی داده‌های تست بهبود یابد. اما استفاده از هایپرپارامترها به کاربر اجازه می‌دهد تا الگوریتم یادگیری را بهترین شکل ممکن تنظیم کند و از مدلی استفاده کند که دقت بالاتری دارد.

در کل، هایپرپارامترها می‌توانند تأثیر قابل‌توجهی بر روی دقت مدل داشته باشند، اما تعیین بهترین مقدار برای آن‌ها به دلیل نیاز به تست مجدد بر روی داده‌ها، می‌تواند زمان‌بر و پر هزینه باشد. در نتیجه، انتخاب هایپرپارامترها باید با دقت و بررسی دقیق انجام شود

## بخش j

۱. کیفیت و کمیت داده‌ها: این چالش بیانگر این موضوع است که داده‌ها باید کیفیت و کمیت خوبی داشته باشند تا مدل‌های یادگیری ماشین به‌صورت دقیق و قابل‌اعتماد کار کنند.

۲. انتخاب ویژگی‌ها: انتخاب ویژگی‌های مناسب از داده‌ها می‌تواند به‌عنوان یک چالش مهم در یادگیری ماشین باشد. این موضوع به دلیل این است که تعداد بسیار زیادی ویژگی‌ها می‌توانند برای داده‌ها وجود داشته باشند و لازم است که از بین آن‌ها ویژگی‌هایی که مفیدتر هستند انتخاب شوند.

۳. پیچیدگی مدل: پیچیدگی مدل یک چالش مهم در یادگیری ماشین است. بعضی از مدل‌های یادگیری ماشین بسیار پیچیده هستند و به‌صورت خودکار قابل‌تنظیم نیستند. به‌عنوان نمونه، مدل‌های شبکه‌های عصبی عمیق چالش‌هایی مانند آموزش، تنظیم پارامتر و غیره دارند.

۴. پیچیدگی محیط: پیچیدگی محیط نیز یک چالش در یادگیری ماشین است. بعضی از محیط‌ها، مانند محیط‌هایی که دارای شرایط آموزش بسیار گسترده هستند، بسیار پیچیده هستند و برای مدل‌های یادگیری ماشین مشکل‌ساز هستند.

## بخش k

اگر مدل شما در داده‌های آموزشی عملکرد عالی داشته باشد، اما به نمونه‌های جدید به خوبی عمومی سازی نشود، ممکن است مشکلی وجود داشته باشد.

این مشکل به عنوان بیش‌برازش (Overfitting) شناخته می‌شود، به این معنی که مدل برای داده‌های آموزشی بسیار پیچیده شده و اطلاعات جزئی و نویز در آن داده‌ها را به یاد گرفته است، اما نتوانسته است الگوهای کلی و عمومی را درک کند.

ترجیحاً باید از روش‌های زیر استفاده کنید تا این مشکل را حل کنید:

۱. استفاده از داده‌های بیشتر: جمع‌آوری داده‌های بیشتری که شامل اطلاعات متنوعی از شرایط مختلف است، می‌تواند بهبودی در عملکرد عمومی مدل برای داده‌های جدید داشته باشد.

۲. استفاده از تکنیک‌های کاهش اندازه مدل: با کاهش اندازه مدل و از بین بردن بعضی از پارامترهای آن می‌توان از بیش‌برازش جلوگیری کرد.

۳. استفاده از روش‌های مانع‌سازی بیش‌برازش: مثل Dropout که با حذف برخی ویژگی‌ها در هر مرحله آموزش، از بیش‌برازش جلوگیری می‌کند.

## بخش l

اگر شما با استفاده از مجموعه داده آزمون، هایپرپارامترهای مدل خود را تنظیم کنید، ممکن است مشکلاتی بوجود آید. اگر هایپرپارامترها برای بهتر شدن عملکرد در داده‌های آزمون تنظیم شوند، احتمال اینکه مدل برای داده‌های دیگر به درستی کار نکند، بسیار بالا است. این مشکل به عنوان بیش‌برازش به داده آزمون شناخته می‌شود و ممکن است باعث شود که مدل شما برای داده‌های جدید به درستی کار نکند. بنابراین برای تنظیم هایپرپارامترها باید از یک مجموعه داده جداگانه به نام مجموعه اعتبارسنجی (Validation set) استفاده کرد.

## بخش m

در رگرسیون چند جمله ای، اگر بررسی منحنی‌های یادگیری، تفاوت بزرگی بین خطاهای آموزش و اعتبارسنجی وجود داشته باشد، این به این معناست که مدل در برابر بررسی داده‌های جدید (اعتبارسنجی) کمترین کارایی را از خود به نمایش گذاشته است. این امر به دلیل بیش‌برازش (overfitting) مدل رخ می‌دهد. برای رفع این مشکل، سه روش می‌توان به کار برد:

1. استفاده از روش‌های کاهش اندازه ویژگی (Feature Selection) و کاهش پیچیدگی یا ساده سازی مدل (Model Simplification)
2. استفاده از روش‌های جلوگیری از بیش‌برازش، مانند Regularization
3. افزایش تعداد داده‌های آموزش (Training Data)

## بخش n

1. رگرسیون ریج به جای رگرسیون خطی ساده (بدون هیچ گونه تنظیم) به دلیل جلوگیری از بیش‌برازش (overfitting) و کاهش وابستگی متغیرهای ورودی استفاده می‌شود. این روش از جمله تکنیک‌های رگولاریزاسیون است که به دلیل اضافه کردن جمله شامل مقدار مطلق همواره مثبت از پارامترهای مدل، می‌تواند باعث کاهش بیش‌برازش و بهبود کارایی مدل شود.
2. لاسو به جای رگرسیون ریج به دلیل اینکه می‌تواند متغیرهای ورودی را انتخاب کند و از بین بردارد که موجب کاهش ابعاد داده و افزایش کارایی مدل می‌شود. به عبارت دیگر، لاسو به عنوان یک الگوریتم انتخاب ویژگی (feature selection) مفید است.
3. الستیک نت به دلیل داشتن مزیت‌های رگرسیون ریج و لاسو به عنوان ترکیبی از دو روش، بهتر از هر دوی آن‌ها استفاده می‌شود. به این صورت که جمله رگولاریزاسیون در الستیک نت شامل یک ترم لاسو و یک ترم ریج می‌شود، به این ترتیب می‌تواند بهتر از رگرسیون ریج و لاسو با هم ترکیب شوند و کارایی مدل را بهبود بخشند.

# سوال دوم

فایل کد در پوشه ی Source Codes با عنوان P2.ipynb قرارداده شده است.

## بخش a

اندازه دیتا ست برابر است با 1383 در 13 یعنی 1383 سطر و 13 ستون که هر ستون یک فیچر است.

ده داده ی تصادفی نیز به شرح ذیل است:

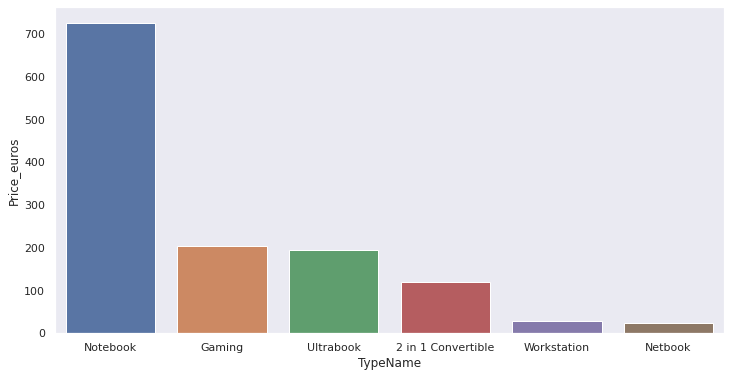


تصویر 2) خروجی 10 داده تصادفی

## بخش b

### نمودار barplot

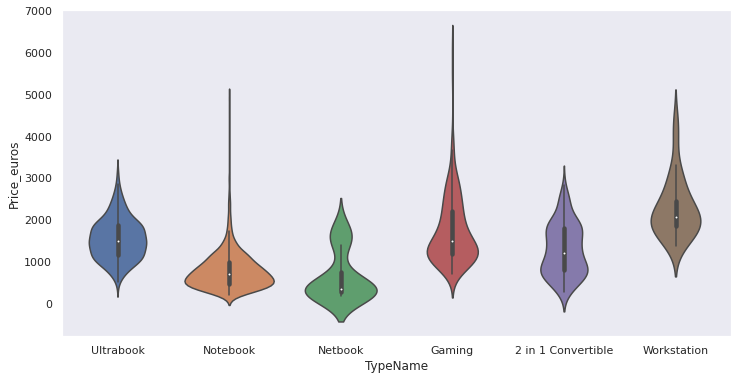
خب این نمودار تعداد بر حسب نوع لپ تاپ آورده شده است. (به Price\_euros توجه نشود.)



تصویر 3) نمودار barplot

### نمودار violinplot

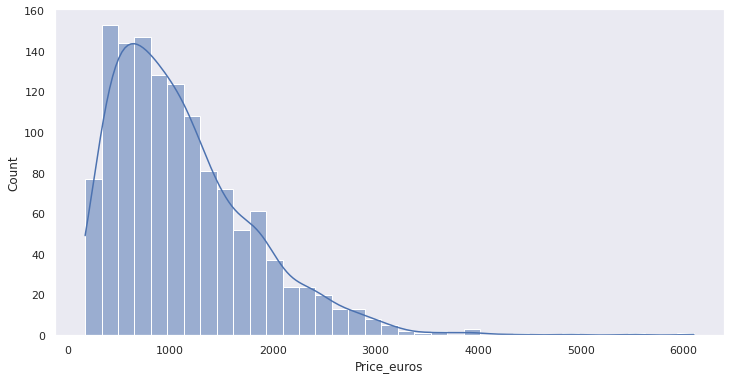
این نمودار مشابه نمودار جعبه ای عمل می کند با این تفاوت که می توان چندین توزیع را با یکدیگر مقایسه کرد.



تصویر 4) نمودار violonplot

### نمودار histplot

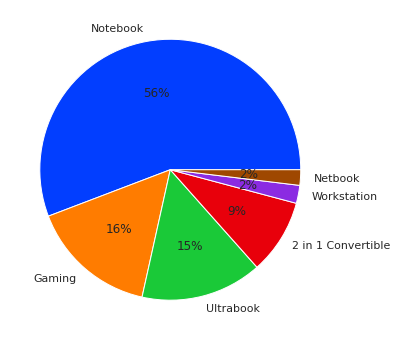
نمودار هیستوگرام داده ها تعداد بر مبنای قیمت را در این بخش داریم.



تصویر 5) نمودار histogram

### نمودار pieplot

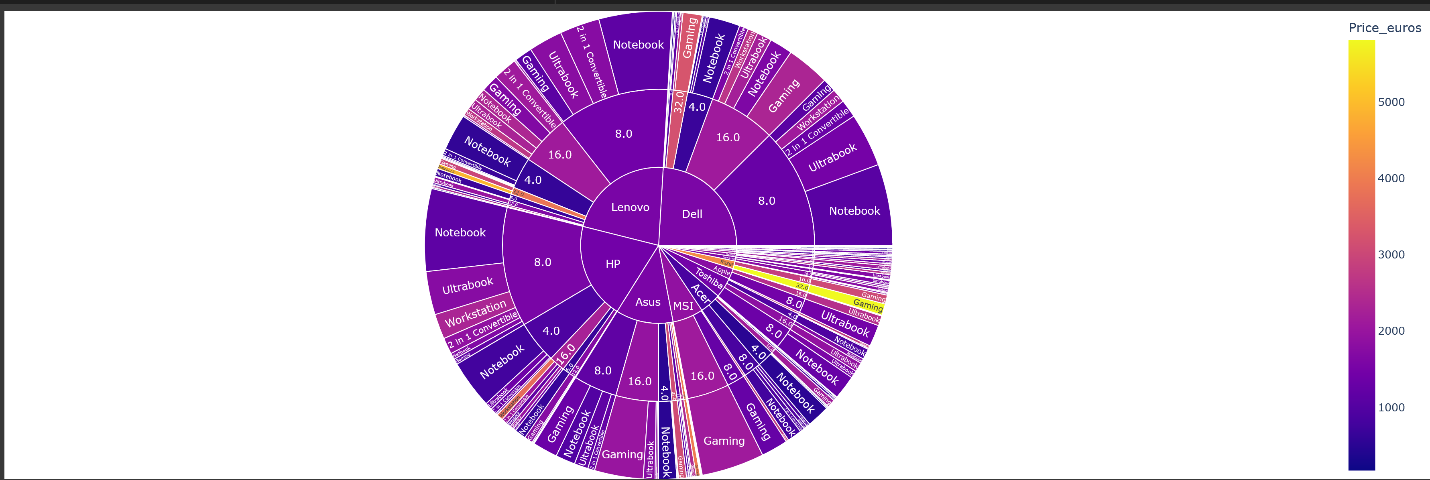
نمودار دایره ای تعداد هر نوع را در شکل ذیل مشاهده می کنید.



تصویر 6) نمودار pieplot

### نمودار sunburst

یکی از جذاب ترین نمودار ها که می توان رسم کرد نمودار sunburst است در این نمودار هر شرکت بر حسب میزان حضورش در بازار در گام اول تقسیم بندی شده و در قدم بعدی نیز بر حسب تولیدات متفاوتی که ارائه می کند.



تصویر 7) نمودار sunburst

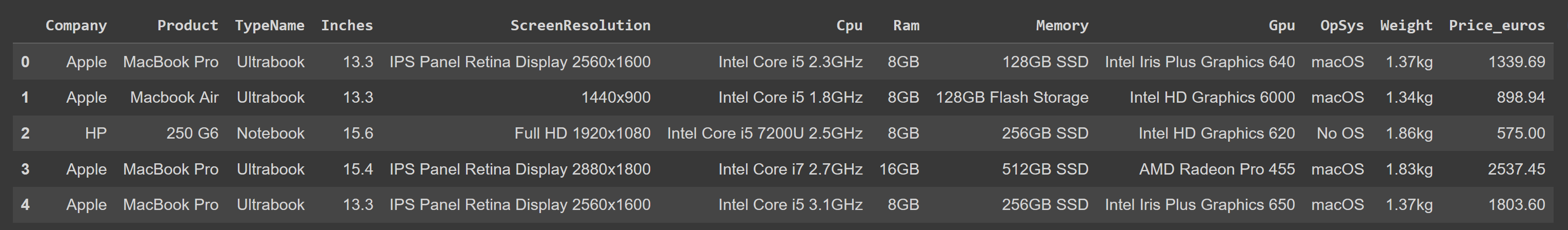
### دیگر نمودار ها

نمودار های دیگری نیز رسم شده است که در بخش کد قابل مشاهده می باشد.

## بخش c و d

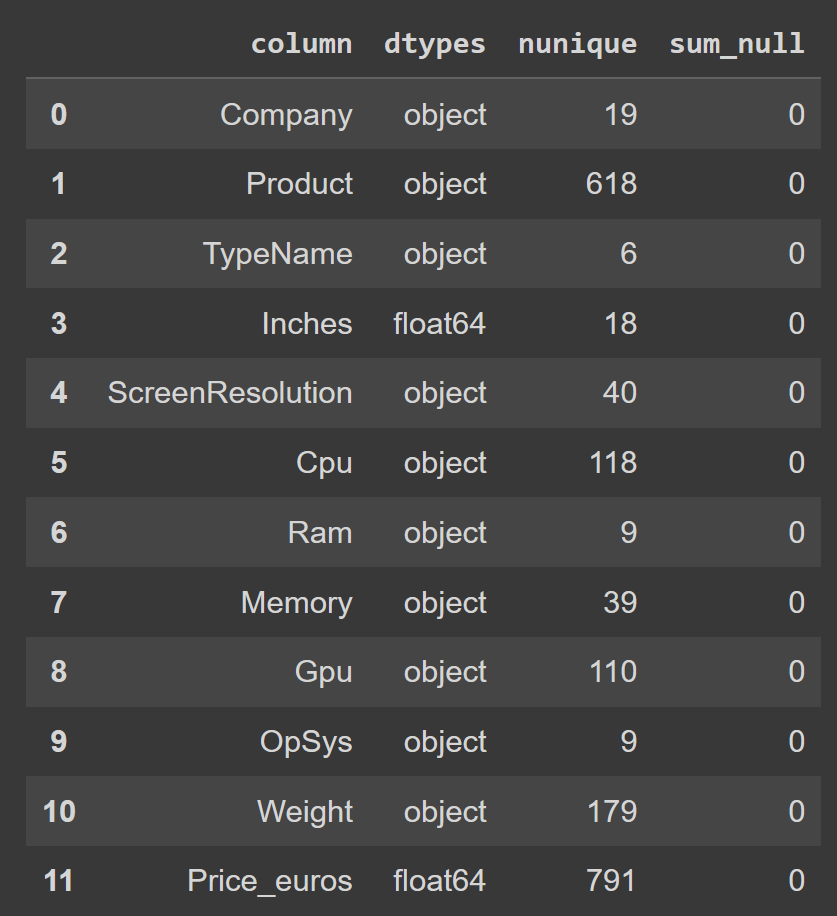
### قدم اول – حذف ستون و بدست آوردن اطلاعات مفید

در این قدم ستون laptop\_ID که بدون کار برد بود را حذف نمودیم.



تصویر 8) خروجی پس از حذف ستون laptop\_ID

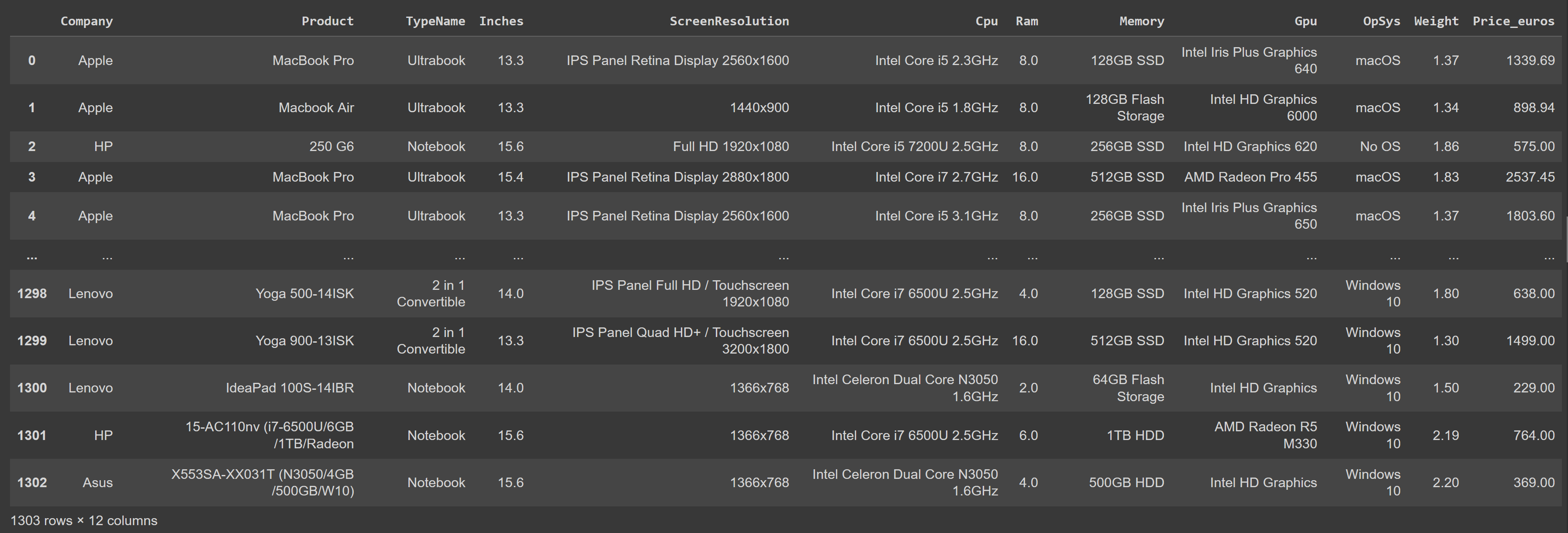
سپس داده ها را از نظر نوع بررسی نمودیم تا اطلاعات مربوط به هریک را بدست آوریم و بتوانیم تغییرات مد نظر را اعمال کنیم.



تصویر 9) خروجی اطلاعات ویژگی ها

### قدم دوم – تبدیل رشته های KG و GB

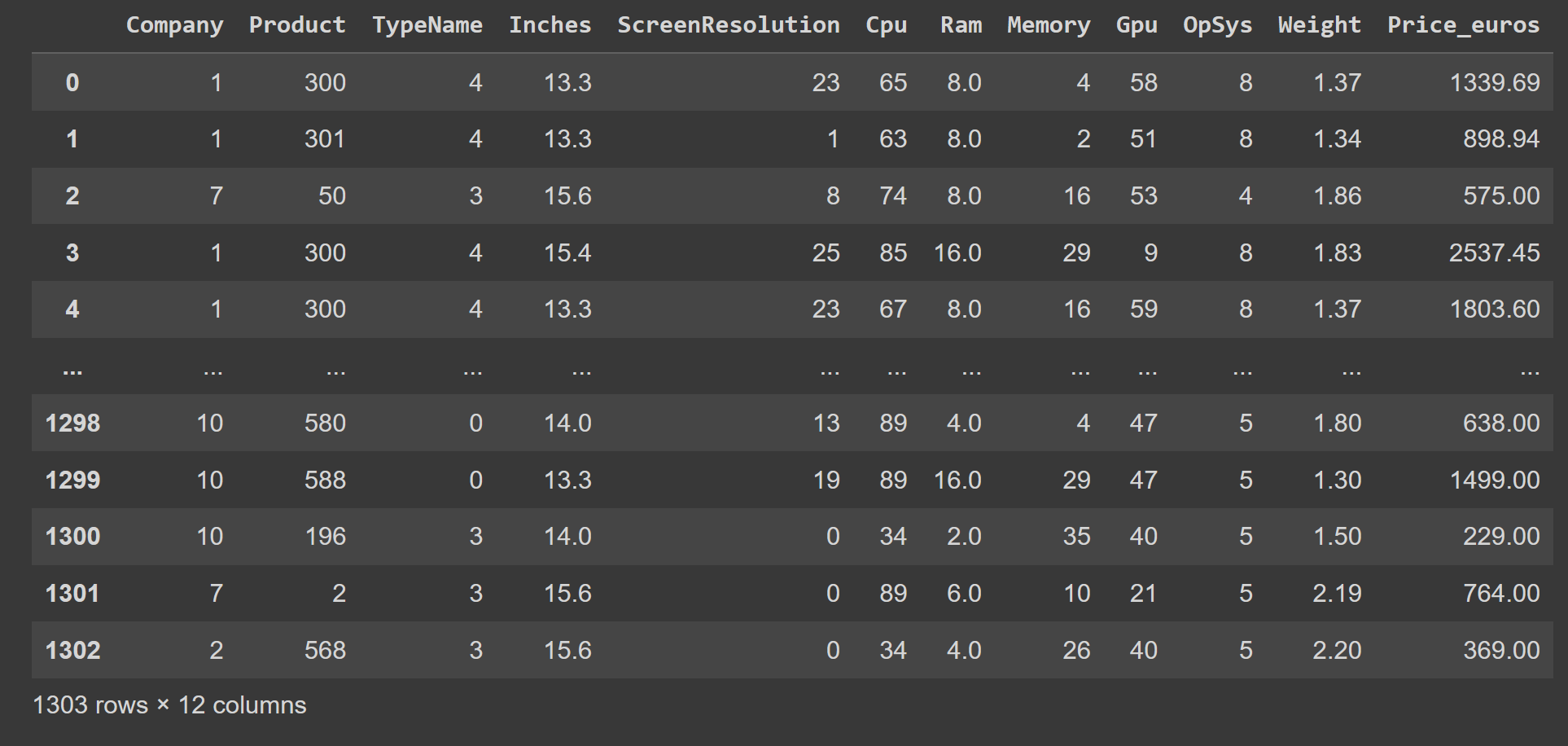
در این قدم داده های ستون هایی که عددی هستن اما قابل کار عددی رویشان نیست را تغییر می دهیم، مانند ستون weight و RAM.



تصویر 10) حذف رشته های زائد

### قدم سوم - labelencoder

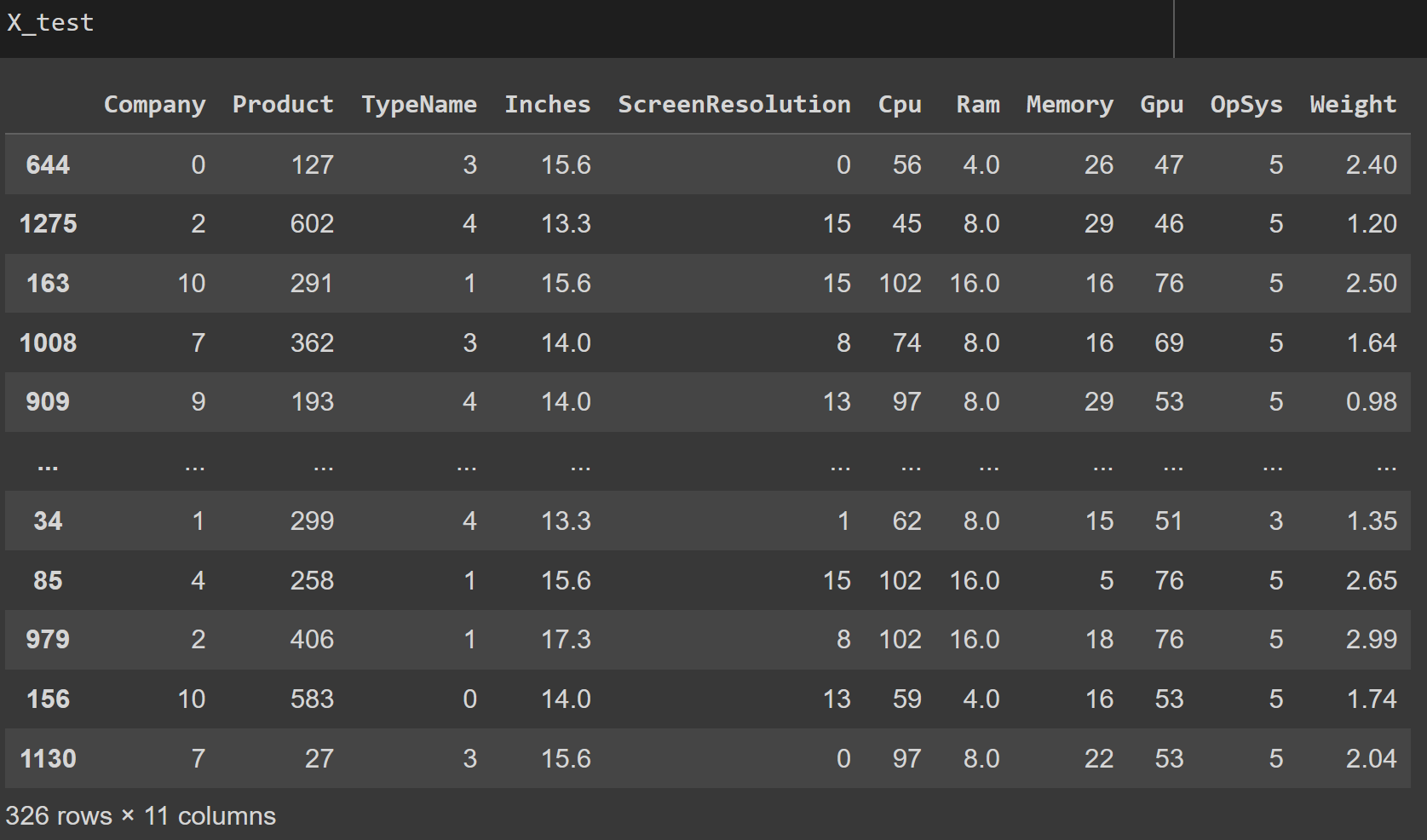
ستون هایی که محتوای آن ها شامل توضیحات متنی بود را به عدد تبدیل کردیم.



تصویر 11) خروجی پس از labelEncoding

### قدم چهارم – تقسیم کردن داده ها

داده ها را به بخش های مختلف train و test و validation تقسیم میکنیم، در تصویر ذیل می بینیم که صرفا داده های تست که 25 درصد کل داده ها هستند آورده شده است(تعداد داده های تست 326 می باشد.):

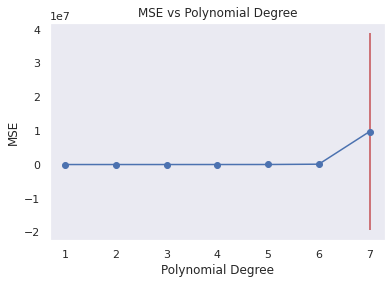


تصویر 12) داده های تست

## بخش e

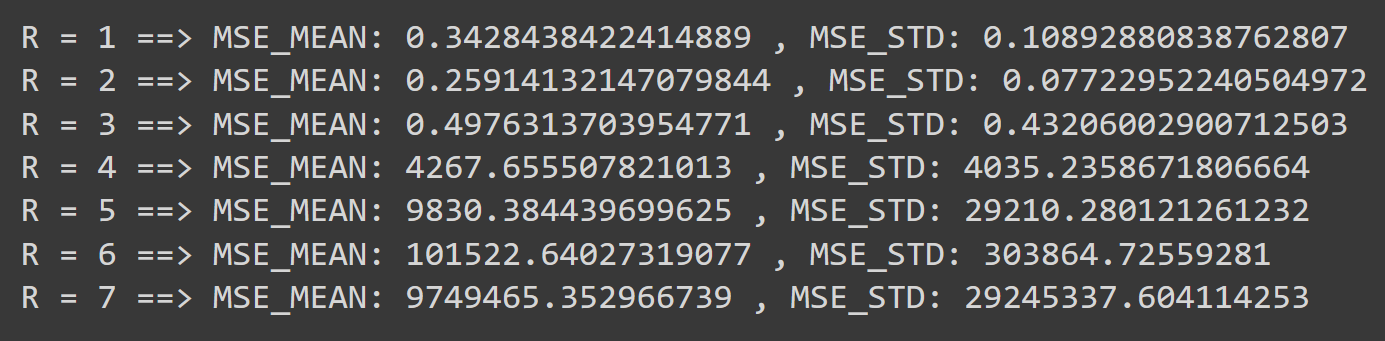
در این بخش ما با قرار دادن k=10 برای دسته بندی داده ها به دسته در kfold اقدام کردیم و برای رنج 1 تا 7 همسایه کار را اجرا نمودیم و براساس محاسبات انجام شده نمودار ذیل را به عنوان خروجی نمایش دادیم.

(این فرایند بر روی google colab نزدیک به 5 دقیقه زمان برد، در ابتدا برای 1 تا 10 تست کردیم که پس از 30 دقیقه با Execution Error مواجه شدیم.)



تصویر 13) خروجی MSE با درجه های مختلف

مقیاس نمودار 10 به توان 7 می باشد از این رو بایستی داده ها را به صورت ذیل مشاهده کرد.

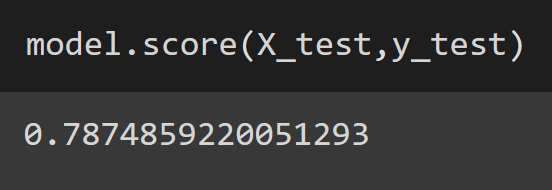
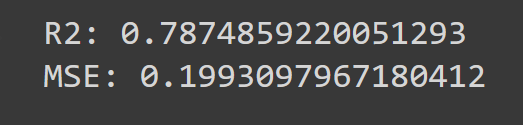


تصویر 14) میانگین و انحراف از معیار MSE بر اساس تعداد همسابه ها

بر این اساس K=2 را به عنوان عدد بهتر بر میگزینیم.

## بخش f

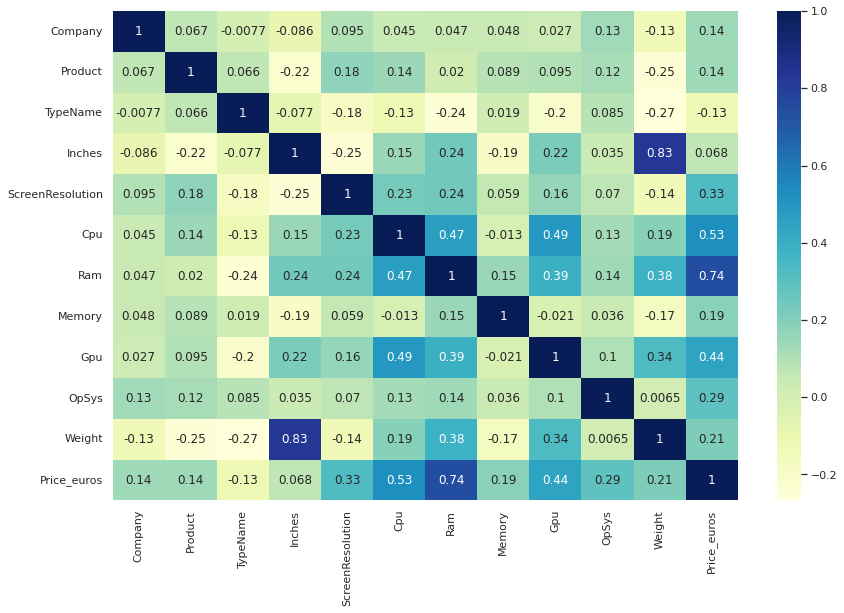
براساس داده های قبلی با استفاده از رگرسیون چندجمله ای داده ها را برای آموزش و تست دسته بندی نموده و در نهایت R2\_Score و MSE را محاسبه می کنیم که به شرح ذیل می باشد.



تصویر 15) خروجی MSE، R2 و model score

## بخش g

بر اساس هیت مپ تصویر مقابل در میابیم که بیشترین ارتباط با قیمت بین ویژگی ها مربوط به Ram می شود.



تصویر 16) جدول Hitmap

# سوال سوم

فایل کد در پوشه ی Source Codes با عنوان P3.ipynb برای بخش مصوری سازی و P3v2.ipynb برای بخش رگرسیون قرارداده شده است.

## توضیحات دیتاست

مجموعه داده کیفیت هوایی، داده‌هایی است که در مورد کیفیت هوای یک منطقه خاص و در یک بازه زمانی مشخص، جمع‌آوری شده‌اند. این داده‌ها ممکن است شامل اطلاعاتی درباره آلاینده‌های مختلف هوا باشند، مانند ذرات معلق، ازون، نیتروژن اکسید، دی اکسید کربن و سایر آلاینده‌های هوایی.

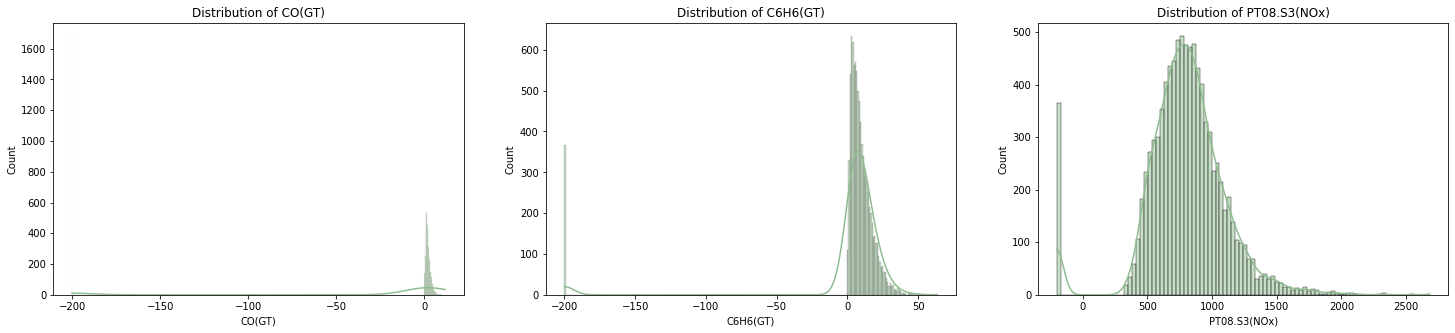
علاوه بر آلاینده‌های هوایی، این داده‌ها می‌توانند شامل اطلاعات هواشناسی مانند دما، رطوبت، سرعت و جهت باد، فشار هوا و تعداد ساعت آفتابی باشند. این داده‌ها به عنوان یک منبع ارزشمند برای بررسی روندهای آلاینده‌های هوایی و تأثیر آن‌ها بر سلامتی و محیط زیست استفاده می‌شوند.

از این داده‌ها برای تحلیل و پیش‌بینی کیفیت هوا، تعیین راهکارهای کاهش آلودگی هوا و تحلیل تأثیر آلودگی هوا بر سلامتی انسان و محیط زیست استفاده می‌شود. به عنوان مثال، می‌توان از این داده‌ها برای تحلیل تأثیر آلاینده‌های هوایی بر بیماری‌های تنفسی و قلبی، تعیین اثرات زیست محیطی سیاست‌های کاهش آلودگی هوا، و تعیین ارزش‌های اقتصادی آلودگی هوا استفاده کرد.

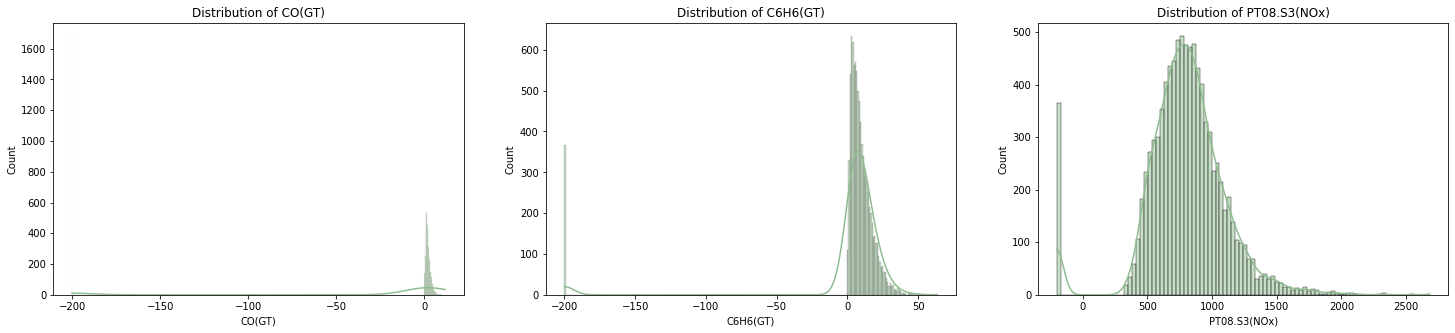
## بخش a

رسم نمودار ها در این بخش انجام شده است، توضیح کوتاه هر یک به عنوان کپشن جدول اضافه می شود که گزارش کار بیش از این طولانی نشود.

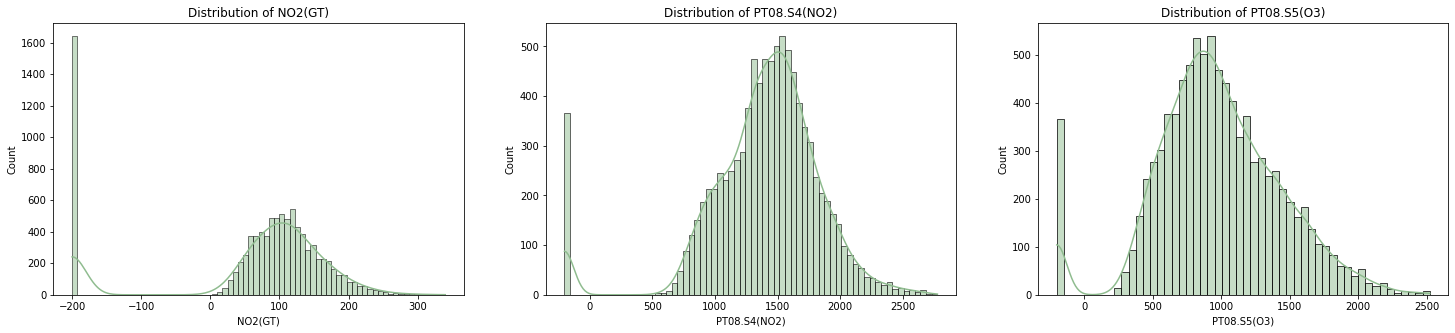
این نمودار جهت بررسی میزان نرمال بودن نسبت به یکدیگر گرفته شده اند.



تصویر 17) نمودار توزیع بر حسب سه ویژگی اول

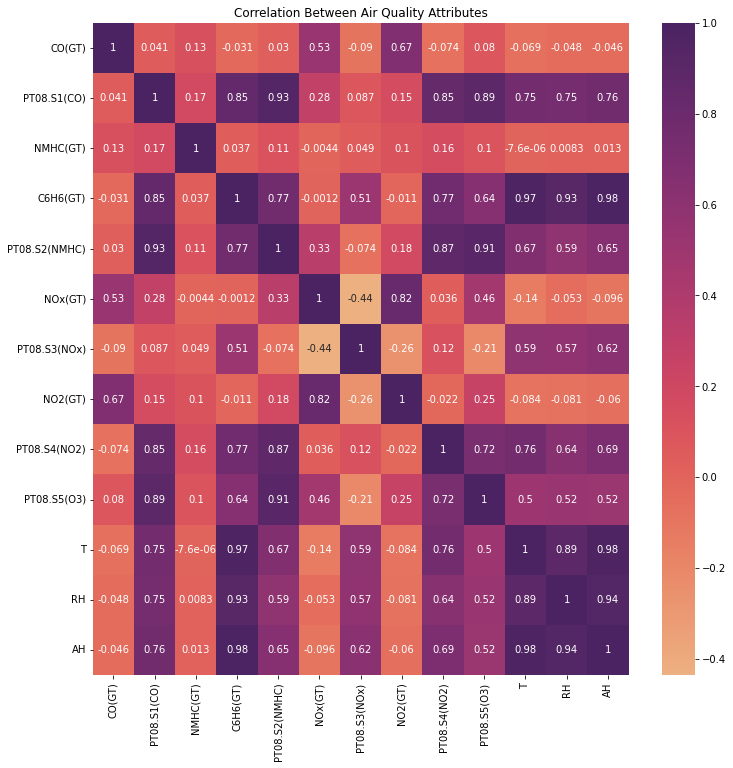


تصویر 18) نمودار توزیع بر حسب سه ویژگی دوم



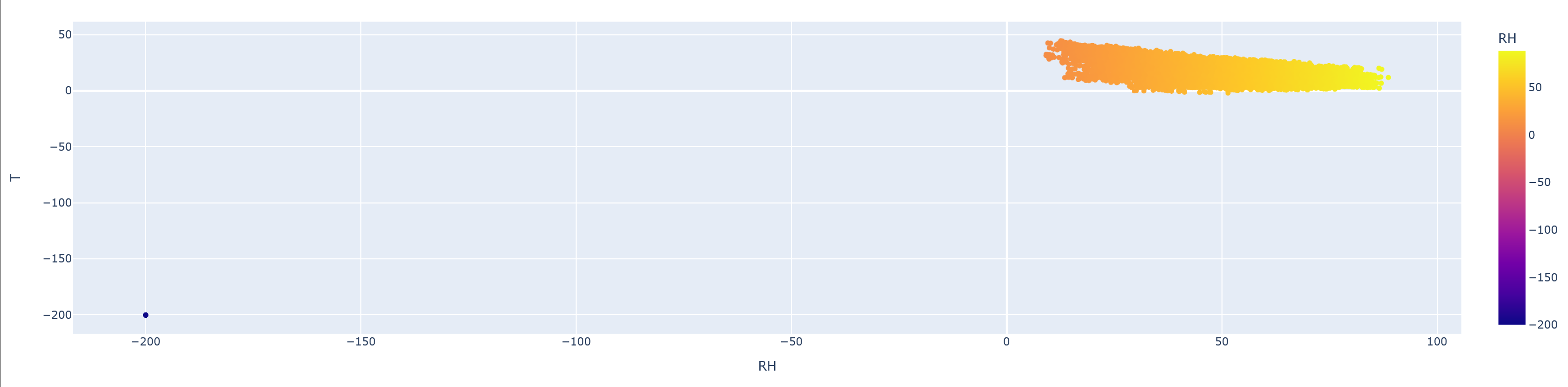
تصویر 19) نمودار توزیع بر حسب سه ویژگی سوم

از هیت مپ برای دریافت نزدیکی و ارتباط بین داده ها استفاده میکنیم:

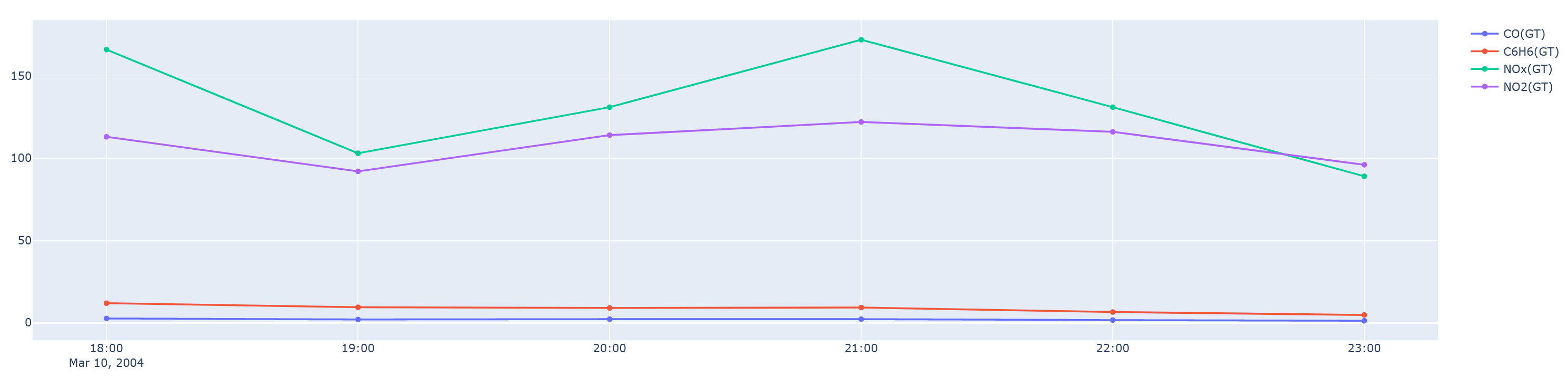


تصویر 20) نمودار هیت مپ تمامی ویژگی ها

این نمودار برای بررسی ارتباط میان رطوب و دما استفاده شده است ک ه یک داده پرت را مشاهده میکنیم.



تصویر 21) نمودار ارتباط بین دما و رطوبت



تصویر 22) نمودار میزان تغییر 4 ویژگی از ساعت 18 الی 23

## بخش b

دستور dropna اجرا شد برای حذف داده های دارای null اما هیچ سطری بابت این مورد حذف نشد، در این بخش ستون time و date با یکدیگر ترکیب شد و به عنوان یک datetimeObject ذخیره سازی شد.



تصویر 23) مجموعه داده ها پس از تغییرات اعمال شده

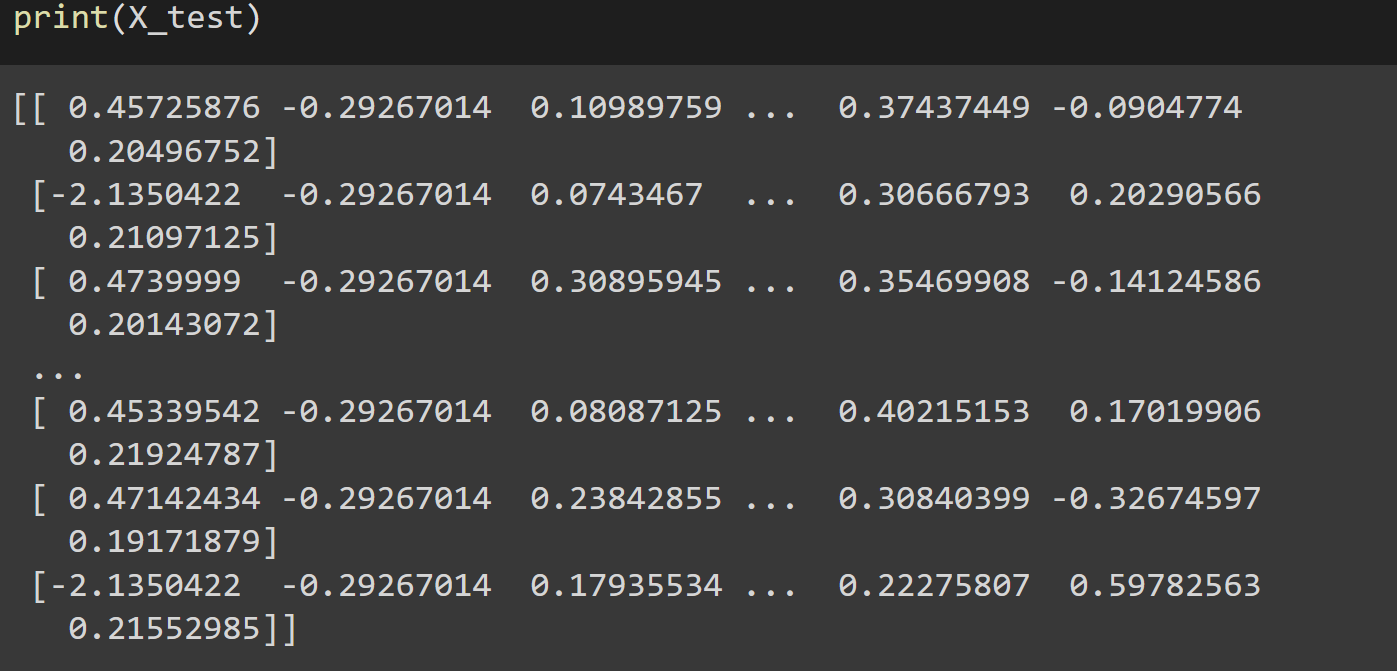
برای مهندسی ویژگی ها را برای بررسی می توانیم از هیت مپی که در بخش قبلی رسم کرده ایم استفاده کنیم.

بیشترین Correlation میان هدف ما با PT08.S2(NMHC) و PT08.S5(O3) با اعداد 0.93 و 0.89 ووجود دارد.

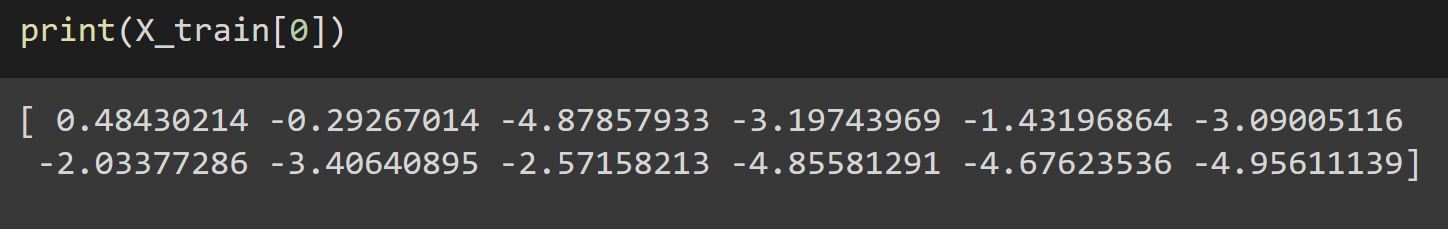


تصویر 24) میزان Correlation

## بخش c



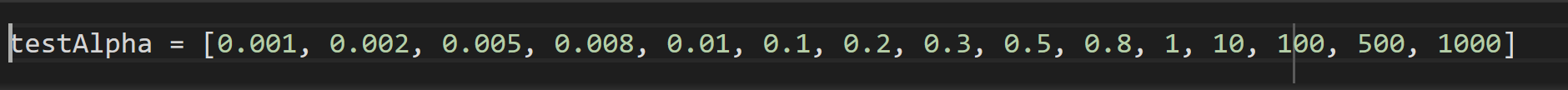
تصویر 25) داده های تست پس از اعمال StandardScaler



تصویر 26) خروجی نمونه پس از جداسازی داده آموزش و تست

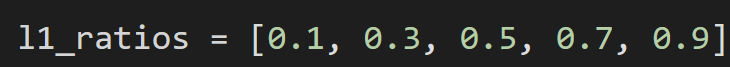
## بخش d

برای همه مقدار آلفا به شرح ذیل قرار داده شده است:



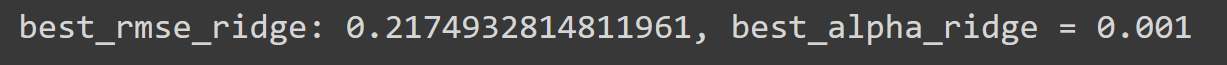
تصویر 27) مقدار لیست آلفا

برای رگرسیون الاستیک مقدار l1\_ratio به صورت ذیل قرار داده شده است:



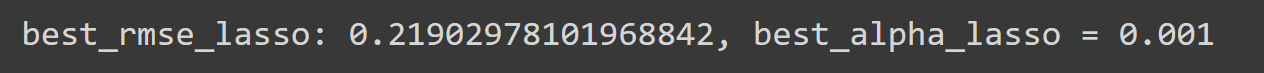
تصویر 28) مقدار لیست l1\_ratios

### Ridge



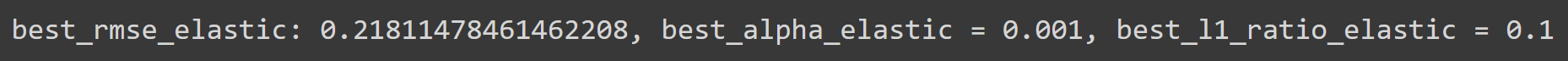
تصویر 29) خروجی پس از آموزش و تست در Ridge

### Lasso



تصویر 30) خروجی پس از آموزش و تست در Lasso

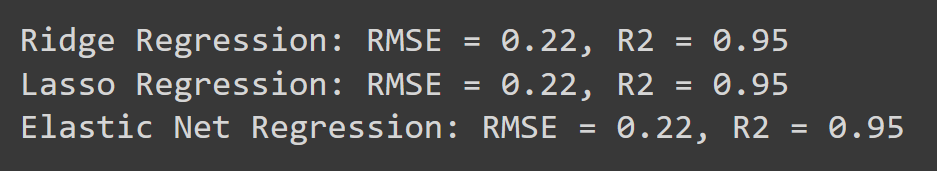
### Elastic



تصویر 31) خروجی پس از آموزش و تست در Elastic

## بخش e

با بهترین پارامتر های انتخاب شده هر سه مدل را آموزش دادیم و تست کردیم و معیار های خواسته شده را اعمال نمودیم:



تصویر 32) نتیجه اعمال معیار های R2 و RMSE

## بخش f

سه مدل رگرسیون Ridge ، Lasso و Elastic Net با هدف پیش‌بینی دقیقتر یک متغیر وابسته در دسترس قرار گرفت. نتایج بررسی سه مدل به شرح زیر است:

Ridge Regression: RMSE = 0.22, R2 = 0.95

Lasso Regression: RMSE = 0.22, R2 = 0.95

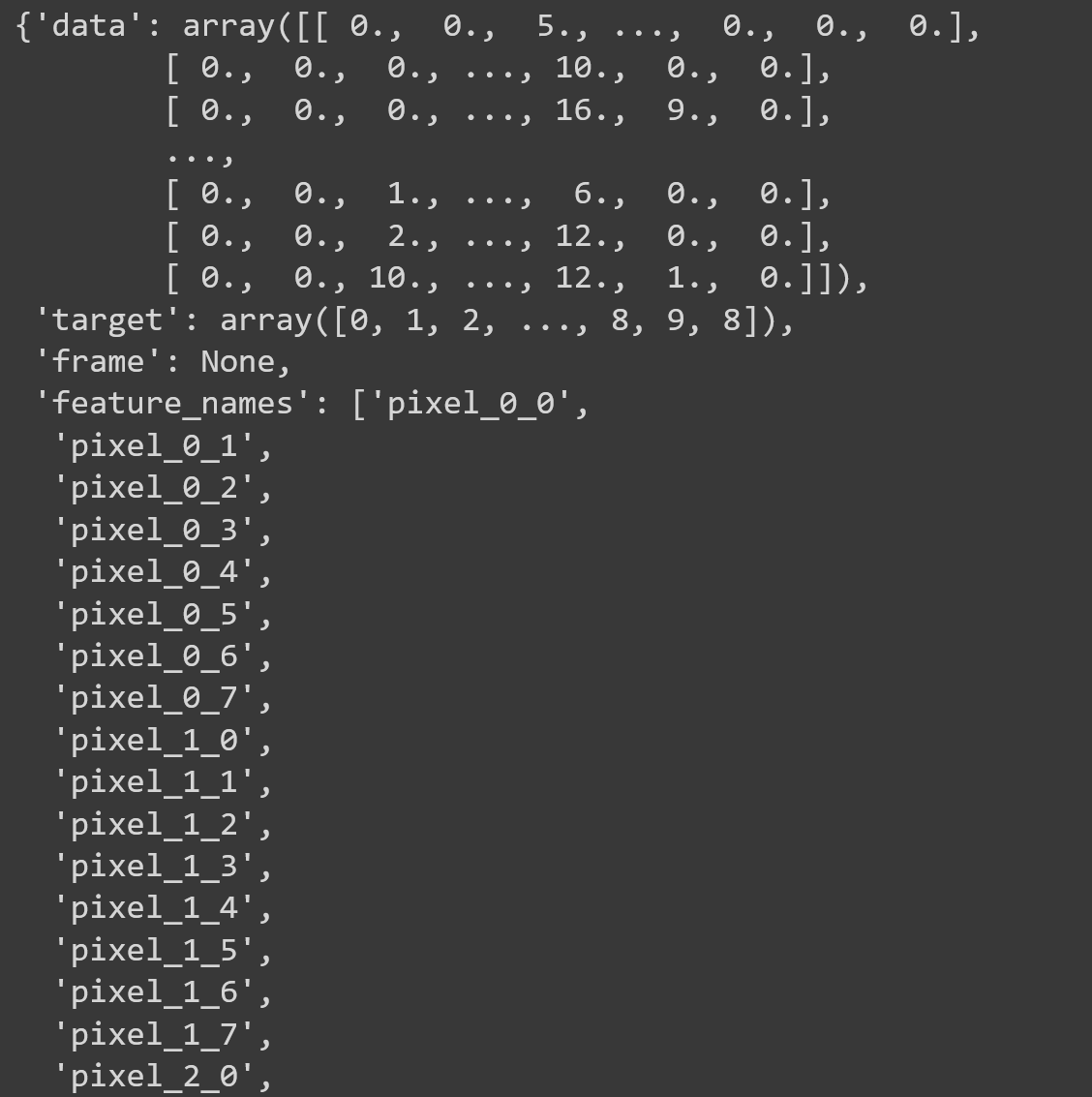
Elastic Net Regression: RMSE = 0.22, R2 = 0.95

با توجه به نتایج بدست آمده، مدل رگرسیون Ridge عملکرد بهتری نسبت به دو مدل دیگر از خود نشان داده است. در واقع، مقادیر RMSE و R2 برای این مدل به ترتیب کمتر و بیشتر از دو مدل دیگر است، که نشان دهنده دقت بیشتر و توانایی بیشتر مدل Ridge در پیش‌بینی متغیر وابسته است.

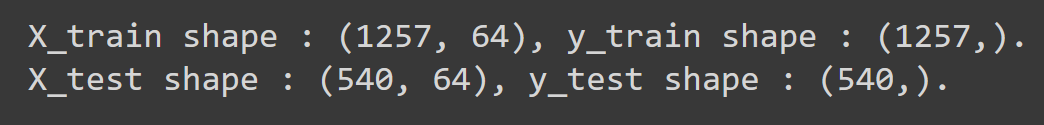
# سوال چهارم

فایل کد در پوشه ی Source Codes با عنوان P4.ipynb قرارداده شده است.

## بخش a

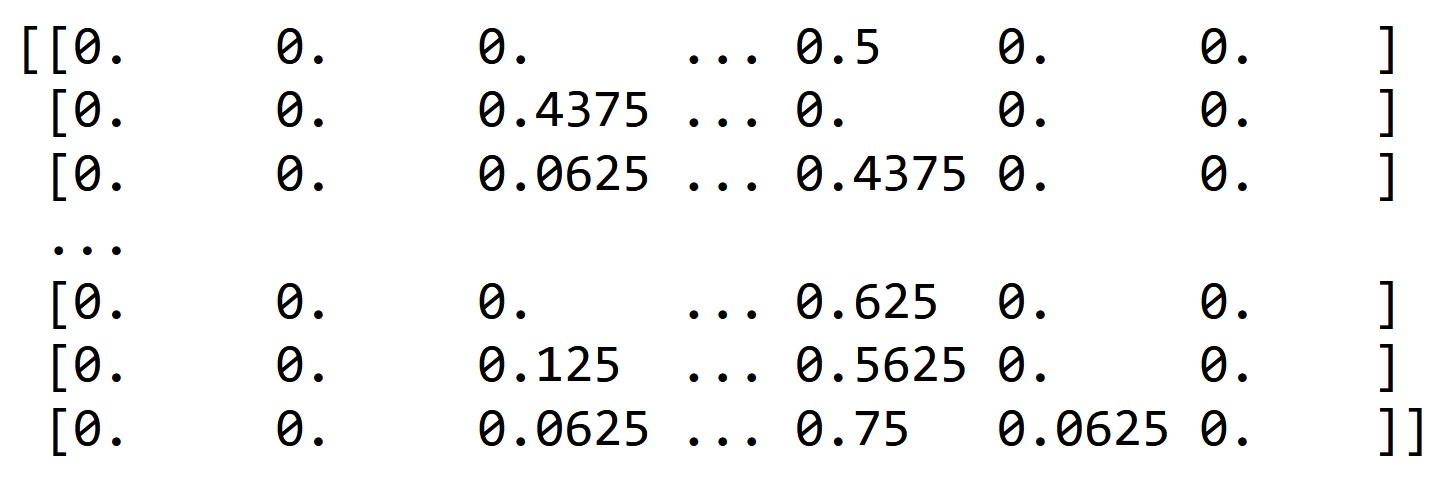


تصویر 33) اضافه کردن دیتاست digits



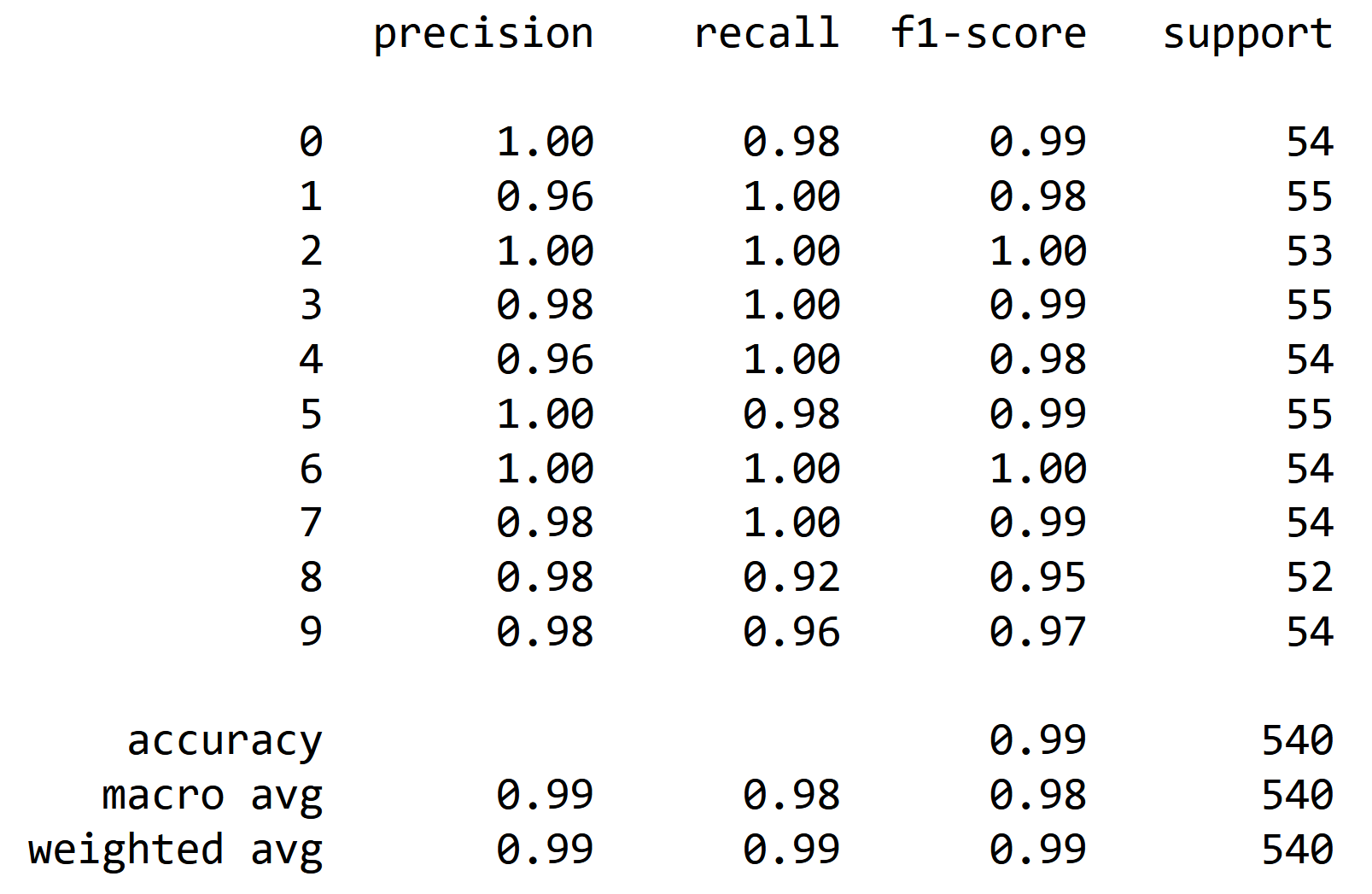
تصویر 34) تقسیم بندی داده ها به داده آموزش و تست

## بخش b

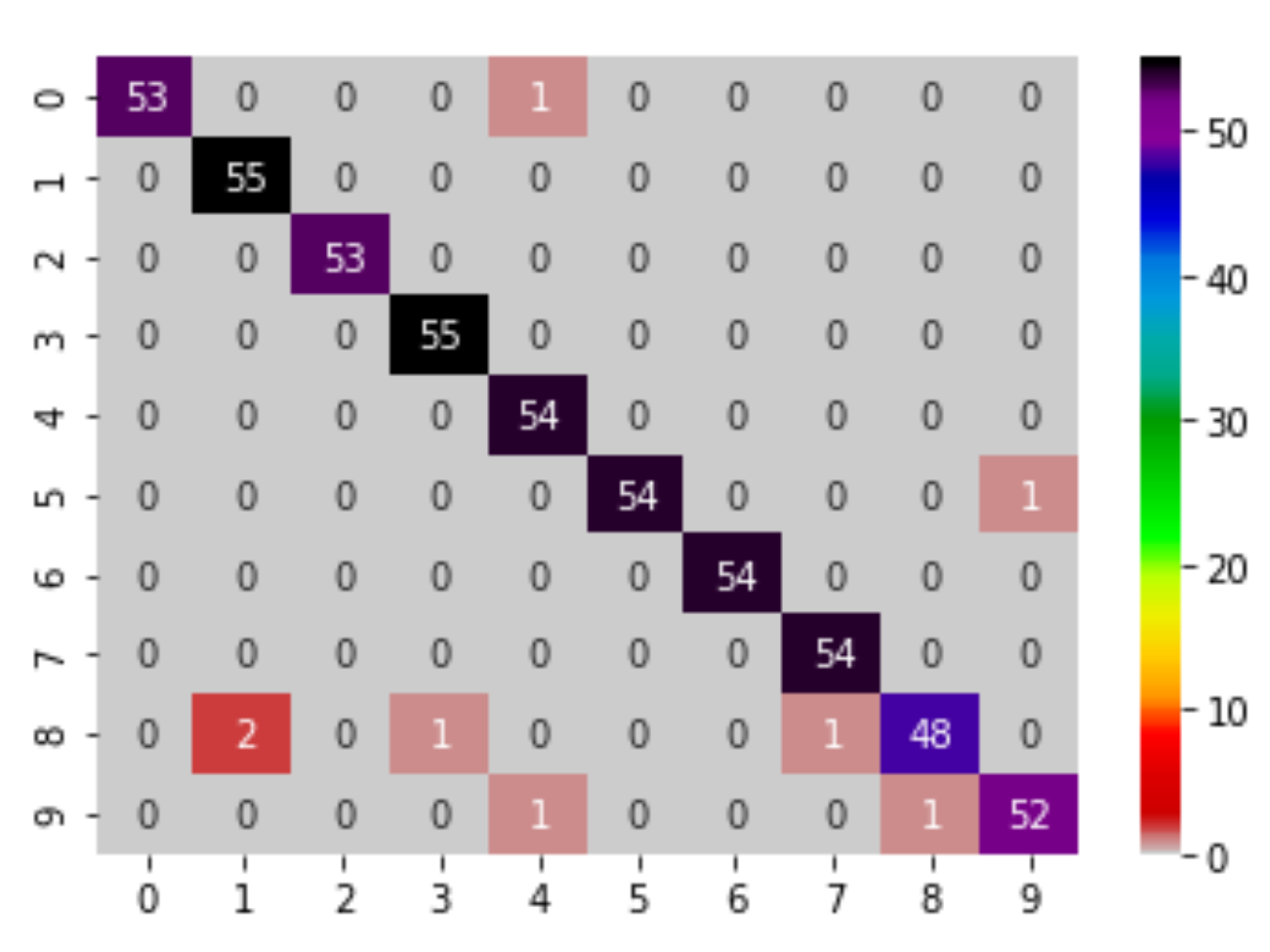


تصویر 35) داده های تست پس از اعمال min max scaler

## بخش c

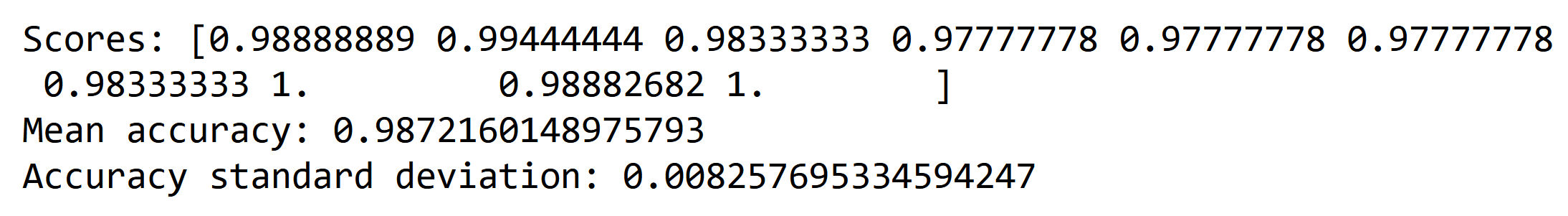


تصویر 36) خروجی Classification\_Report پس از آموزش و مدل سازی با 5 همسایه



تصویر 37) ماتریس Confusion

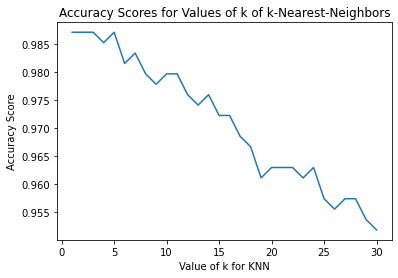
## بخش d



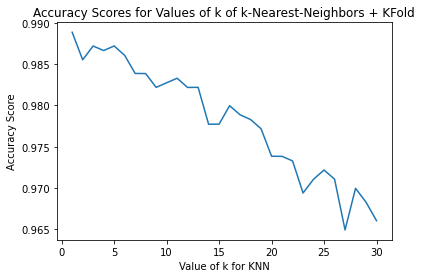
تصویر 38) خروجی Score در k=10 روش KFold

## بخش e

خب در این بخش قطعه کدی نسبتا طولانی نوشتیم که در آن بازه ی یک تا سی را برای k اختصاص دادیم و نتیجه صحت را در لیستی ذخیره نمودیم، الگوریتم را با این روش اجرا نمودیم و نتجه ذیل در قالب نمودار قابل دسترس است:



تصویر 39) نمودار نسبت Accuracy نسبت به K بدون KFold



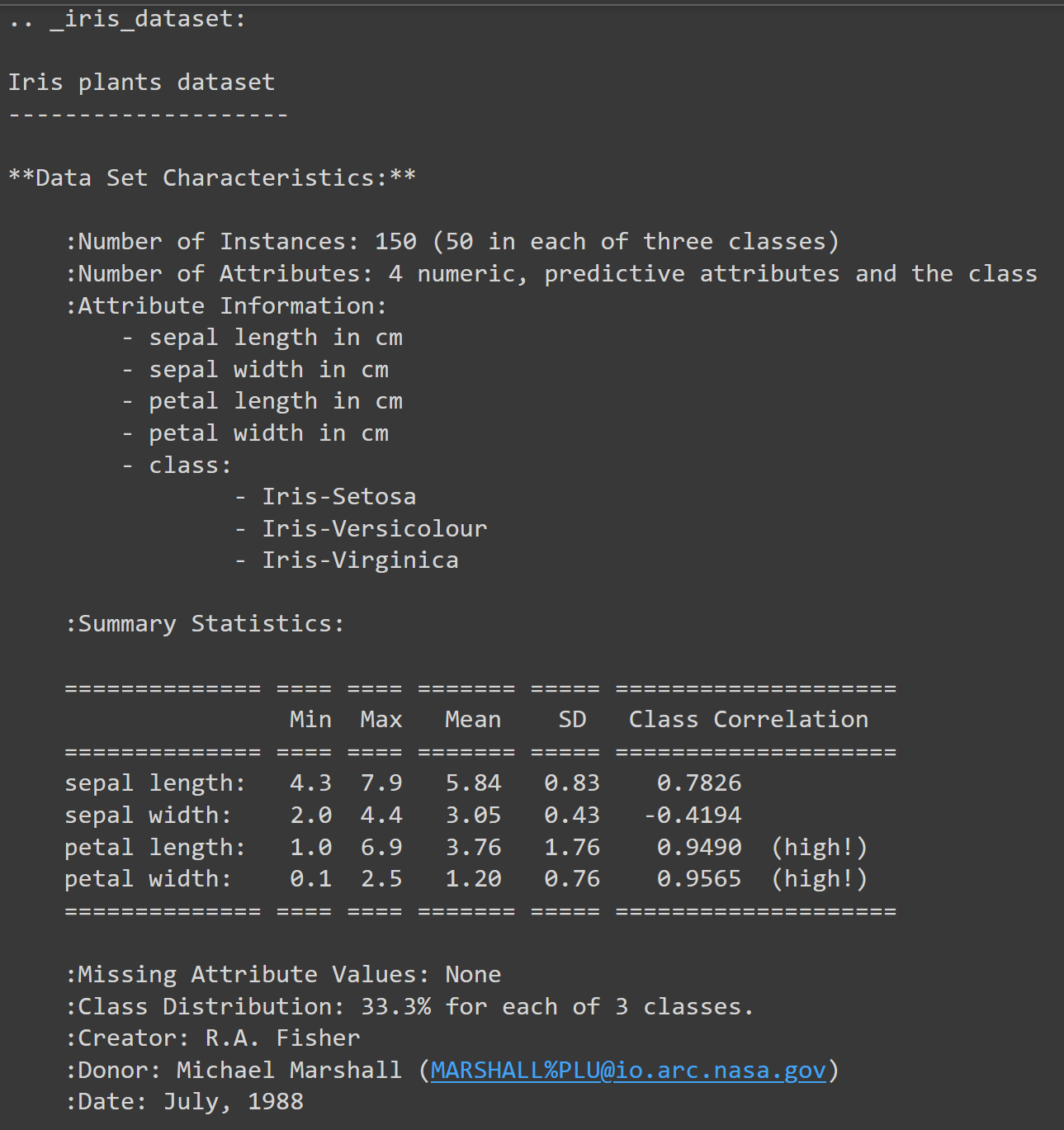
تصویر 40) نمودار نسبت Accuracy به K با استفاده از KFold

# سوال پنجم

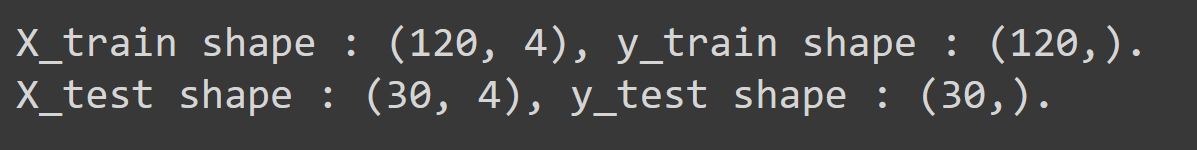
فایل کد در پوشه ی Source Codes با عنوان P5.ipynb قرارداده شده است.

فایل کد من به دلیل آماده سازی برای آموزش در کلاس پایتون کامل تر از خواسته های این سوال می باشد و در بخش های این سوال صرفا پاسخ آن بخش ها آورده شده است.

## بخش a



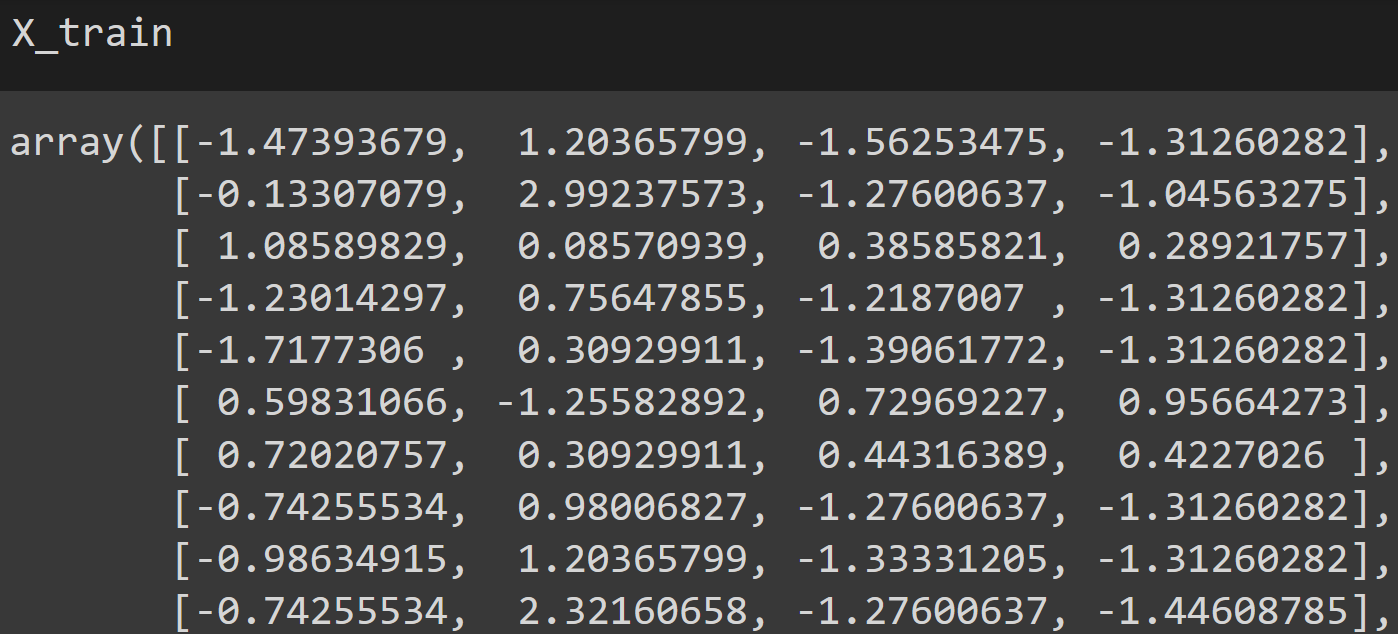
تصویر 41) بارگیری دیتاست iris از sklearn.datasets



تصویر 42) جداسازی داده های آموزش و تست 80/20

## بخش b

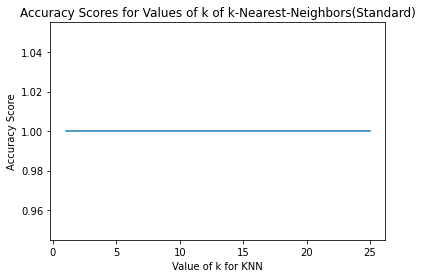
در این بخش ما داده های آموزش و تست را استاندارد میکنیم، خروجی بخشی از داده های آموزش در تصویر قابل مشاهده است.



تصویر 43) خروجی داده های آموزش پس از استاندارد سازی

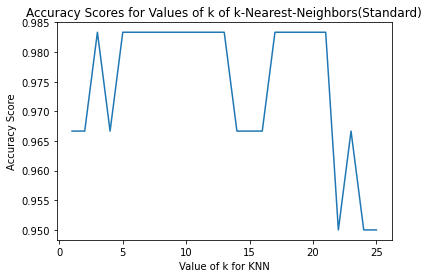
## بخش c

خب آموزش و بررسی انجام شد و تماما پاسخ با حالت خواسته شده 1 می باشد از این رو k را نمی توان یافت و مشاهده میکنیم که احتمالا ما دارای خطایی هستیم که یکی از دلایل آن کمبود تعداد داده ها می باشد، حال داده های آموزش و تست را به 60/40 تغییر می دهیم تا شاید تغییری حاصل شود و نتایج بهتری جهت نمایش بدست آید.



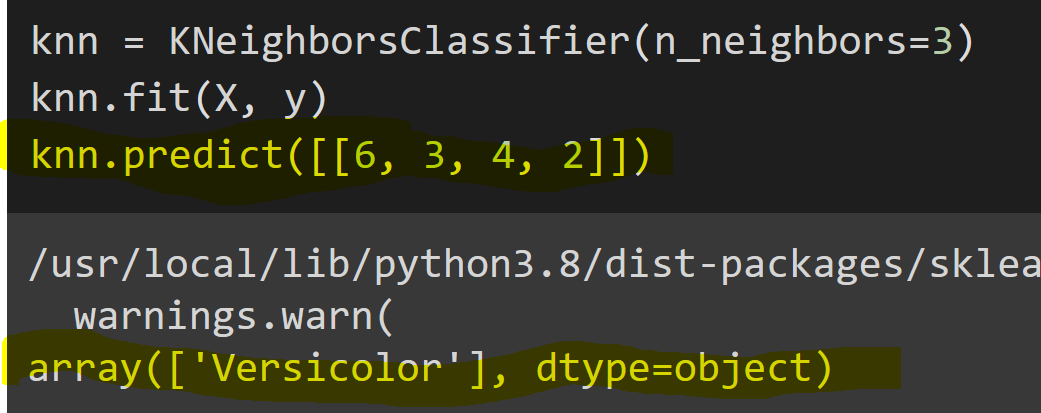
تصویر 44) نمودار صحت در حالت خواسته شده

در ادامه فعالیتی انجام می گردد که خواسته نشده است.



تصویر 45) نمودار صحت پس از تغییر نسبت داده های تست و آموزش

عدد 3 را به عنوان k برگزیده انتخاب میکنیم و سپس آموزش می دهیم.



تصویر 46) تصویری از کد به همراه پیش بینی