

**گزارش تکلیف دوم درس داده‌کاوی**

**استاد درس: دکتر احمدی**

تهیه کنندگان:

سید نیما محمودیان

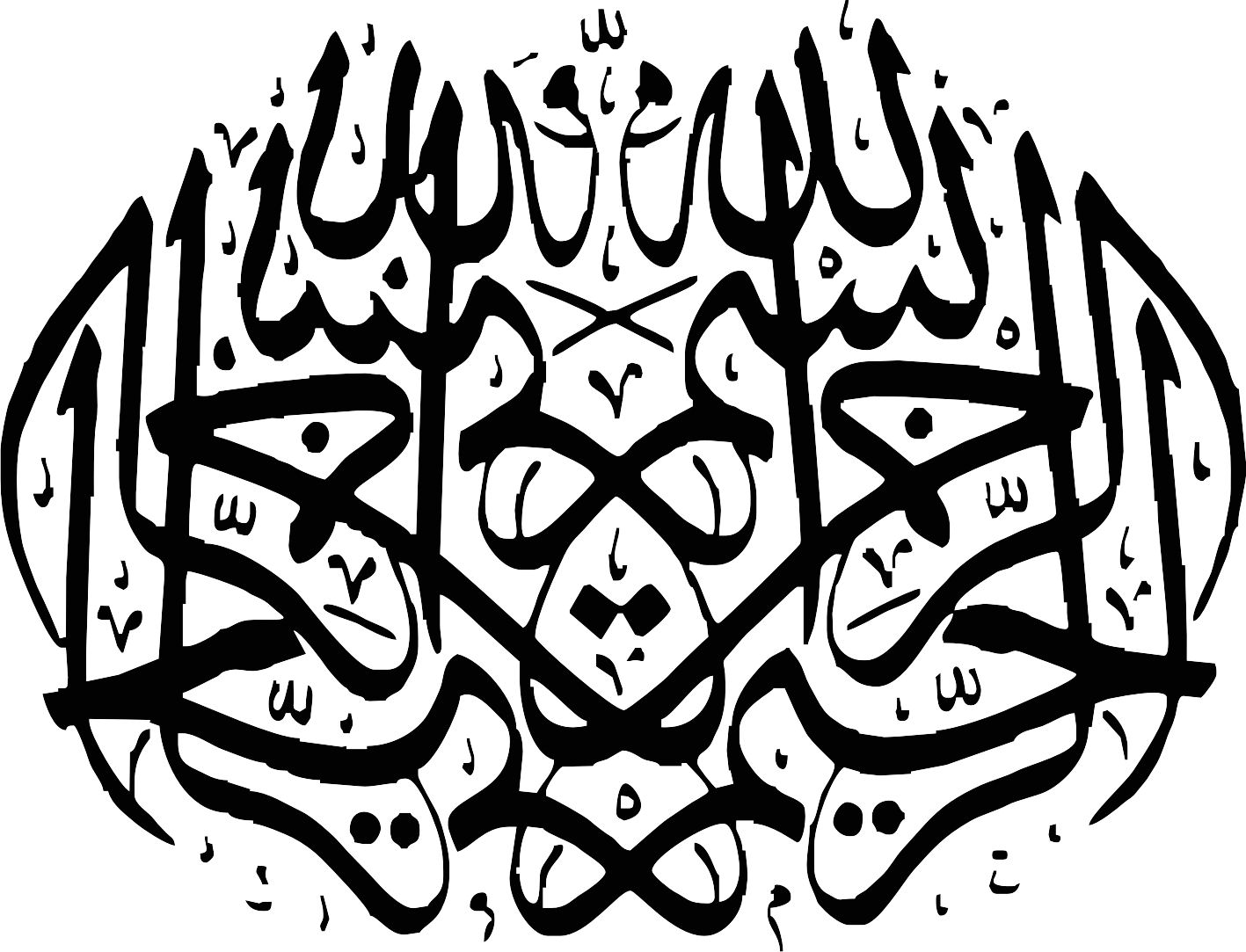
402125005

معصومه بهبهانی‌زاده

402191003

محسن فراغه

401225002



فهرست مطالب

[چکیده 1](#_Toc163856775)

[1-معرفی مجموعه داده 1](#_Toc163856776)

[2- فراخوانی داده‌ها 1](#_Toc163856777)

[1-2 سوال مسئله یک 1](#_Toc163856778)

[2-2 کدهای اولیه و فراخوانی داده‌ها 2](#_Toc163856779)

[3-2 تحلیل توصیفي داده‌ها 3](#_Toc163856780)

[4-2 پیش پردازش داده‌ها 5](#_Toc163856781)

[1-4-2پیدا کردن داده‌های تکراری 5](#_Toc163856782)

[2-4-2 پیدا کردن مقادیر گمشده 5](#_Toc163856783)

[3-4-2 پیدا کردن مقادیر پرت 8](#_Toc163856784)

[4-4-2 استاندارد سازی داده‌ها 16](#_Toc163856785)

[5-4-2 تقسیم بندی داده‌ها به دو دسته آموزشی و آزمایشی 17](#_Toc163856786)

[6-4-2 انتخاب ویژگی با روش جست و جو رو به جلو 19](#_Toc163856787)

[3-اجرای الگوریتم KNN 21](#_Toc163856788)

[2-3-پیدا کردن تعداد همسایه مناسب 21](#_Toc163856789)

[2-3-ارزیابی عملکرد KNN با روش Leave One Out 23](#_Toc163856790)

[4-دسته‌بندی بیزی 25](#_Toc163856791)

[5- مسئله دوم 27](#_Toc163856792)

[1-5 ترسيم در محور مختصات دو بعدى و تعیین برچسب داده ها 27](#_Toc163856793)

[2-5 تعیین مرز دو كلاس با روش نزديكترين همسايگي 28](#_Toc163856794)

[3-5 تعیین مرز دو كلاس با روش پرسپترون 30](#_Toc163856795)

فهرست شکل‌ها

[شکل 1:کد فراخوانی کتابخانه‌ها 2](#_Toc163857700)

[شکل 2:کد فراخوانی مجموعه داده 2](#_Toc163857701)

[شکل 3: خروجی کد df.nunique 4](#_Toc163857702)

[شکل 4: خروجی کد df.dtype 4](#_Toc163857703)

[شکل 5: خروجی کد df.nunique 5](#_Toc163857704)

[شکل 6:Pairplot 6](#_Toc163857705)

[شکل 7: نمودار Heatmap 7](#_Toc163857706)

[شکل 8: مشخص کردن داده‌های تکراری 7](#_Toc163857707)

[شکل 9: نمایش مقادیر گمشده 8](#_Toc163857708)

[شکل 10: نمایش تعداد مقادیر " ؟ " 8](#_Toc163857709)

[شکل 11: نمایش مقادیر " 0 " 9](#_Toc163857710)

[شکل 12: کد مربوط به نمایش مقادیر پرت 9](#_Toc163857711)

[شکل 13: جایگذاری دادههای پرت با مقادیر حد بالا و پایین 12](#_Toc163857712)

[شکل 14: نمایش مجدد تعداد مقادیر " 0 " 12](#_Toc163857713)

[شکل 15: جایگذاری مقادیر nan به جای " 0 " 13](#_Toc163857714)

[شکل 16: کد اجرا شده برای جایگذاری مقادیر گمشده 13](#_Toc163857715)

[شکل 17: کد اجرا شده برای نمایش نمودار مقایسه‌‌ای هر یک از روش‌ها 14](#_Toc163857716)

[شکل 18: نمودار ضخامت پوست 14](#_Toc163857717)

[شکل 19: نمودار انسولین 15](#_Toc163857718)

[شکل 20: نمودار گلوکز 15](#_Toc163857719)

[شکل 21: نمودار فشارخون 16](#_Toc163857720)

[شکل 22: نمودار BMI 16](#_Toc163857721)

[شکل 23: استاندارد سازی داده 17](#_Toc163857722)

[شکل 24: کد تقسیم داده‌ها به دو دسته آموزشی و آزمایشی 17](#_Toc163857723)

[شکل 25: نمایش نمودار میله‌ای داده‌ها 18](#_Toc163857724)

[شکل 26: کد متعادل سازی مجموعه داده 18](#_Toc163857725)

[شکل 27: خروجی پس از متعادل سازی مجموعه داده 19](#_Toc163857726)

[شکل 28: کد روش جست و جو رو به جلو 19](#_Toc163857727)

[شکل 29: ویژگی انتخابی از روش جست و جو رو به جلو 20](#_Toc163857728)

[شکل 30: ارائه مدل برای اعتبارسنجی 20](#_Toc163857729)

[شکل 31: گزارش حساسیت 20](#_Toc163857730)

[شکل 32: گزارش اختصاصیت 20](#_Toc163857731)

[شکل 33: کد پیدا کردن K بهینه 21](#_Toc163857732)

[شکل 34: نمودار معیار دقت بر اساس K 22](#_Toc163857733)

[شکل 35: ماتریس درهم‌ریختگی و کد ایجاد کننده آن 22](#_Toc163857734)

[شکل 36: محاسبه سه مقدار دقت، حساسیت و اختصاصیت به ترتیب 23](#_Toc163857735)

[شکل 37: ایجاد مدل و آموزش مدل 23](#_Toc163857736)

[شکل 38: نحوه انجام ارزیابی Leave One Out 24](#_Toc163857737)

[شکل 39: سه معیار دقت حساسیت و اختصاصیت حاصل از Leave One Out 24](#_Toc163857738)

[شکل 40: کد مربوط به دسته‌بندی بیزی 25](#_Toc163857739)

[شکل 41: ماتریس درهم ریختگی حاصل از دسته‌بندی بیزی 26](#_Toc163857740)

[شکل 42: سه معیار دقت حساسیت و اختصاصیت برای مدل دسته‌بندی بیزی 26](#_Toc163857741)

[شکل 43:کد‌های قسمت 1 27](#_Toc163857742)

[شکل 44:خروجی قسمت 1 28](#_Toc163857743)

[شکل 45: :کدهای روش NN 29](#_Toc163857744)

[شکل 46: خروجی روش NN 29](#_Toc163857745)

[شکل 47: کدهای روش پرسپترون 30](#_Toc163857746)

[شکل 48: خروجی روش پرسپترون 31](#_Toc163857747)

فهرست جدول‌ها

[جدول 1: نمای کلی مجموعه داده 3](#_Toc163857748)

[جدول 2: خروجی کد df.describe 4](#_Toc163857749)

[جدول 3: مقادیر پرت شاخص بارداری 9](#_Toc163857750)

[جدول 4: مقادیر پرت شاخص گلوکز 10](#_Toc163857751)

[جدول 5 : مقادیر پرت شاخص فشار خون 10](#_Toc163857752)

[جدول 6: مقادیر پرت شاخص ضخامت پوست 10](#_Toc163857753)

[جدول 7: بخشی از مقادیر پرت شاخص انسولین 10](#_Toc163857754)

[جدول 8: مقادیر پرت شاخص BMI 11](#_Toc163857755)

[جدول 9: مقادیر پرت شاخص DiabetesPedigreeFunction 11](#_Toc163857756)

[جدول 10: مقادیر پرت شاخص سن 11](#_Toc163857757)

[جدول 11: داده‌های استاندارد شده به روش مین مکس 17](#_Toc163857758)

# چکیده

در این مطالعه تحلیل بر روی مجموعه داده[[1]](#footnote-1) دیابت انجام شده است. داده­های مورد بررسی از [[2]](#footnote-2)UCI گرفته شده است. پس از معرفی مجموعه داده، فرآیند آماده سازی داده­ها و بررسی مقادیر گمشده و پرت را در دستور کار داریم، سپس به استاندارد سازی داده­ها می­پردازیم. در ادامه کار با استفاده از روش جست و جوی رو به جلو ویژگی­های کاربردی تر را از بین ویژگی­های موجود انتخاب می­کنیم. سپس معیارهای دقت، حساسیت و اختصاصیت را برای این انتخاب ویژگی­ها به دست می­آوریم. در ادامه با استفاده از روش KNN [[3]](#footnote-3) دسته بندی داده­ها را انجام می­دهیم. معیارهای نام برده در بالا را دوباره برای مدل ارائه شده محاسبه می­کنیم و تمامی مراحل را با استفاده از روش دسته­بندی بیزی در دستور کار خواهیم داشت.

مسئله دوم مورد بررسی در این پژوهش، شامل بررسی و به دست آوردن مرز تصمیم بین دو کلاس با استفاده از روش­های نزدیکترین همسایگی و پرسپترون می­باشد.

# **1-معرفی مجموعه داده**

مجموعه داده مورد بررسی، اطلاعات 768 فرد را نمایش می­دهد، هدف مجموعه داده این است که بر اساس اندازه‌گیری‌های تشخیصی خاص موجود در مجموعه داده، پیش‌بینی تشخیصی اینکه آیا بیمار مبتلا به دیابت است یا خیر را انجام دهد. در جمع­آوری این مجموعه داده محدودیت­هایی نیز ایجاد شده است. این مجموعه داده اطلاعات زنان بالای 21 سال از میراث هندی پیما هستند. توضیحات هر یک از ویژگی مورد بررسی عبارتند از:

1. بارداری: تعداد دفعاتی که فرد باردار شده است.
2. گلوکز: غلظت کلوگز پلاسما در یک آزمایش دو ساعته
3. فشار خون: فشار خون دیاستولیک
4. ضخامت پوست: ضخامت چین­های پوستس سه سر بازو بر حسب میلی متر
5. انسولین: سرم دو ساعته انسولین
6. شاخص توده بدنی BMI[[4]](#footnote-4): وزن بر حسب کیلوگرم / قد بر حسب متر به توان دو
7. DiabetesPedigreeFunction: تابعی از سابقه خانوادگی فرد از ابتلا به دیابت
8. سن: سن افراد شرکت کننده

# 2- فراخوانی داده­ها

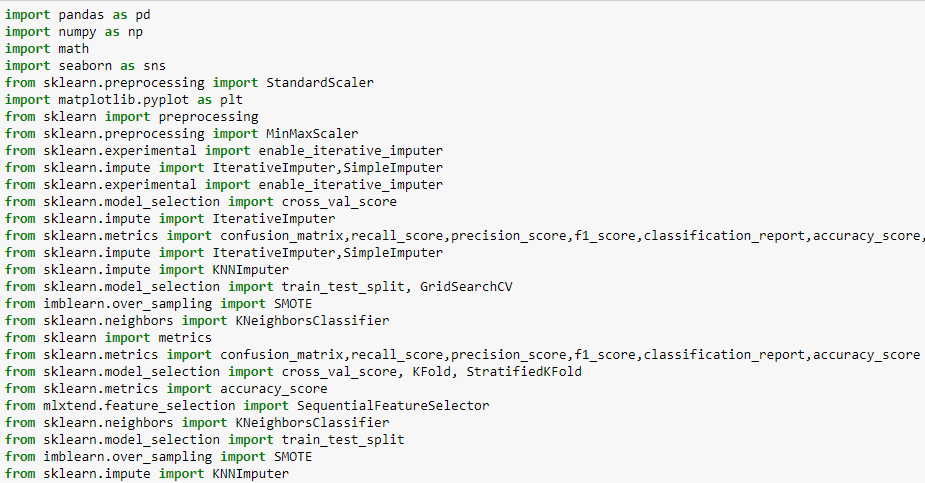
## 1-2 سوال مساله یک

## مجموعه داده pima-indian-diabetes از مخزن دادههای استاندارد دانشگاه کالیفرنیا ( UCI Machine Learning Repository)مورد نظر است. سوالات زیر را پاسخ دهید.

الف) بعد از پیش پردازش داده­ها، با استفاده از روش جست و جوی رو به جلو، بهترین ترکیب ویژگی­ها را به دست آورید. از روش نزدیکترین همسایگی جهت دسته بندی استفاده کنید. بدین منظور از روش Hold out استفاده کنید. مقدارهای دقت، حساسیت و اختصاصیت را گزارش کنید.

## 2-2 کدهای اولیه و فراخوانی داده‌ها

در ابتدا کتابخانه­های مورد نیاز را فراخوانی می­کنیم. در شکل 1 بخشی از آن­ها قابل مشاهده می­باشند.



شکل 1:کد فراخوانی کتابخانه­ها

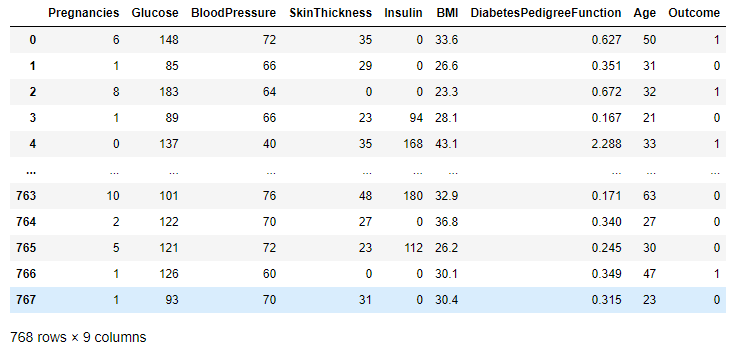
با توجه به این مهم، که برای بررسی این مجموعه داده فایل CSV[[5]](#footnote-5) مربوط را دانلود کرده­ایم، برای فراخوانی این مجموعه داده مطابق شکل 2 پیش خواهیم رفت.



شکل 2:کد فراخوانی مجموعه داده

در جدول 1 یک نمای کلی از مجموعه داده را نمایش می­دهیم:

جدول 1: نمای کلی مجموعه داده



## 3-2 تحلیل توصیفي داده­ها

در گذشته همانگونه که در شکل 2 مشاهده کردید، یک قالب داده تحت عنوان “df” از کل مجموعه داده­ای که در اختیار داشتیم، ایجاد کردیم. برای تسلط بیشتر بر روی مجموعه داده اقدامات زیر را انجام می­دهیم:

df.head : برای مشاهده پنج سطر اول از مجموعه داده استفاده می­شود.

df.tail: برای مشاهده پنج سطر آخر از مجموعه داده استفاده می­شود.

df.nunique: تعداد مقادیر منحصر به فرد را در هر ستون از چارچوب داده برمیگرداند. این به ما نشان می­دهد که در هر ستون چند مقدار مختلف وجود دارد و داده­ها چقدر متنوع هستند. کد و خروجی این بخش در شکل 3 نمایش داده شده است.

df.describe: تعداد داده­های هر ستون، مقادیر میانگین، انحراف معیار، کمترین و بیشترین مقدار هر ستون و چارک­ها را نمایش می­دهد. کد و جدول خروجی در جدول 2 قابل مشاهده است.

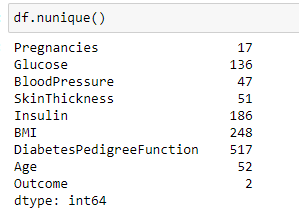
df.info : خلاصه­ای از داده­ها شامل فهرست، نام ستون­ها، انواع داده­ها، مقادیر غیرتهی، میزان استفاده از حافظه و سایر اطلاعات را چاپ می­کند. این یک نمای کلی از چارچوب داده و ویژگی­های آن به ما می­دهد. کد و خروجی این بخش در شکل 5 نمایش داده شده است

df.shape: تعداد ردیف­ها و ستون­ها را به ما برمی­گرداند.

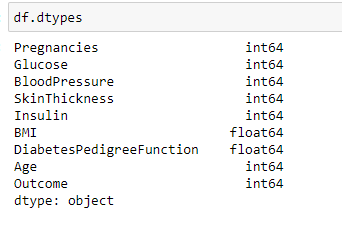
df.dtype: نوع هر داده را مشخص می­کند. کد و خروجی این بخش در شکل4 نمایش داده شده است.

در ادامه برای شناخت بهتر مجموعه داده و ارتباط دو به دو هر یک از ویژگی­ها با یکدیگر نمودارهای زیر را رسم کرده­ایم:

نمودار Pairplot نشان داده شده در شکل 6 که به عنوان ماتریس پراکندگی نیز شناخته می شود، ماتریسی از نمودارها می­باشد که امکان تجسم رابطه بین هر جفت متغیر در یک مجموعه داده را فراهم می­کند. هر دو نمودار هیستوگرام و پراکندگی را ترکیب می کند و یک نمای کلی منحصر به فرد از توزیع­ها و همبستگی­های مجموعه داده ارائه می­دهد.

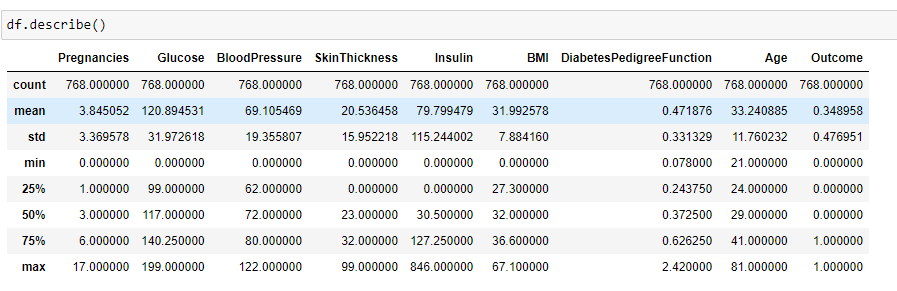


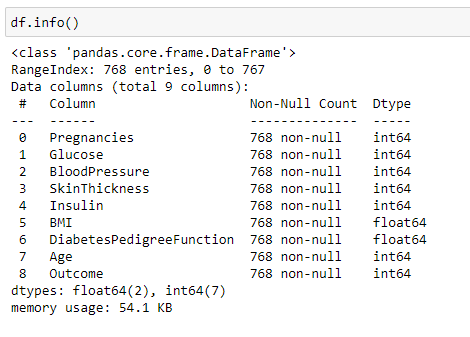
شکل 3: خروجی کد df.nunique



شکل 4: خروجی کد df.dtype

جدول 2: خروجی کد df.describe





شکل 5: خروجی کد df.nunique

نمودار Heatmap[[6]](#footnote-6) ترسیم شده، همانگونه که در شکل 7 پیداست، برای نشان دادن روابط بین دو متغیر استفاده می­شود که یکی در هر محور ترسیم شده است. با مشاهده چگونگی تغییر رنگ سلول­ها در هر محور، می­توانید مشاهده کنید که آیا الگوهایی در ارزش یک یا هر دو متغیر وجود دارد یا خیر.

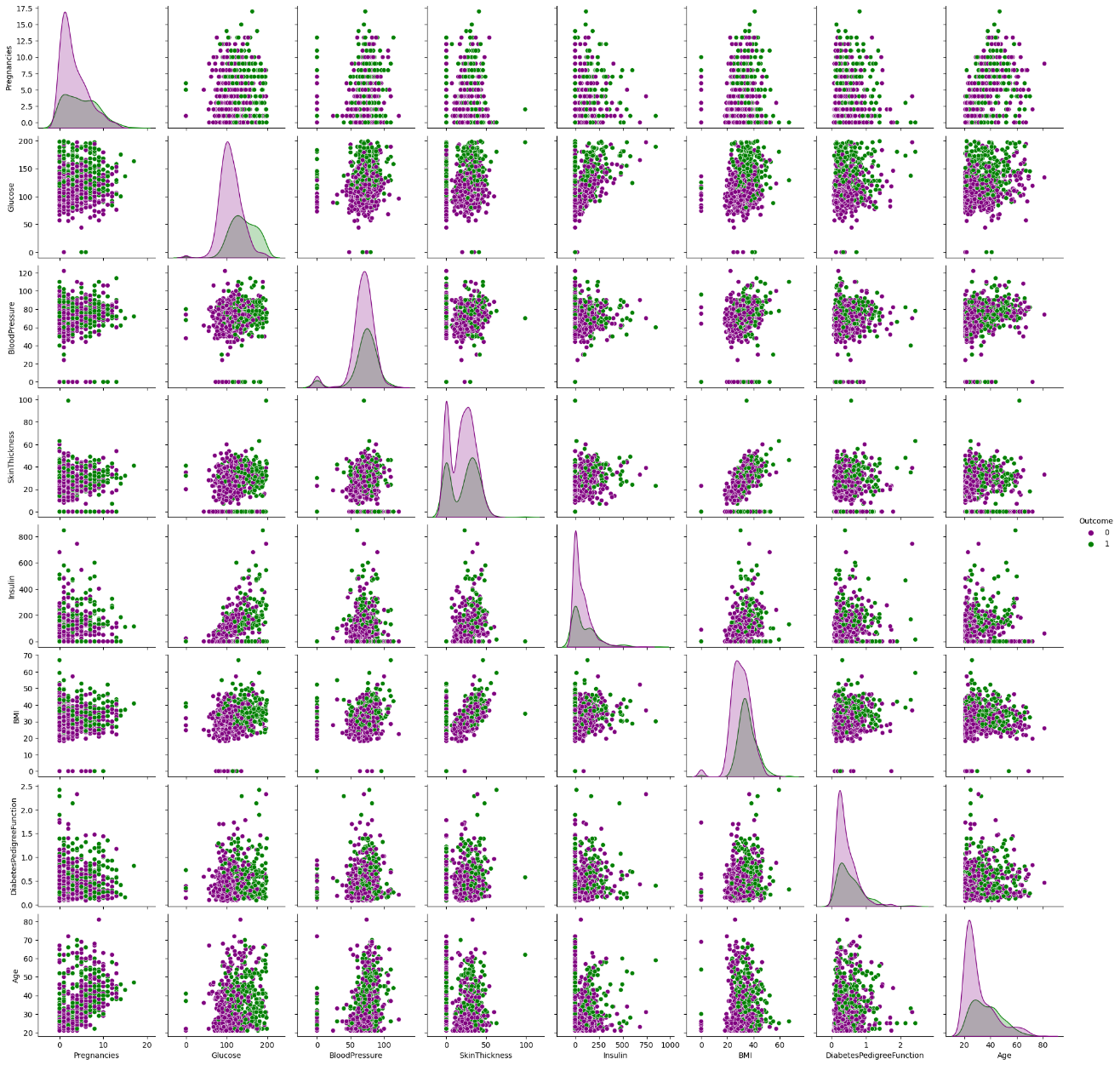
## 4-2 پیش پردازش داده­ها

### 1-4-2پیدا کردن داده­های تکراری

برای بررسی مجموعه داده باید اطمینان حاصل کنیم که هیچ داده تکراری در مجموعه داده ما وجود ندارد. اگر ستون­های کاملا مشابهی در مجموعه داده ما وجود دارد، باید حذف شوند. که در مجموعه داده مورد بررسی ما طبق کد دستوری در شکل8 ، هیچ دو ستون مشابهی موجود نمی­باشد

### 2-4-2 پیدا کردن مقادیر گمشده

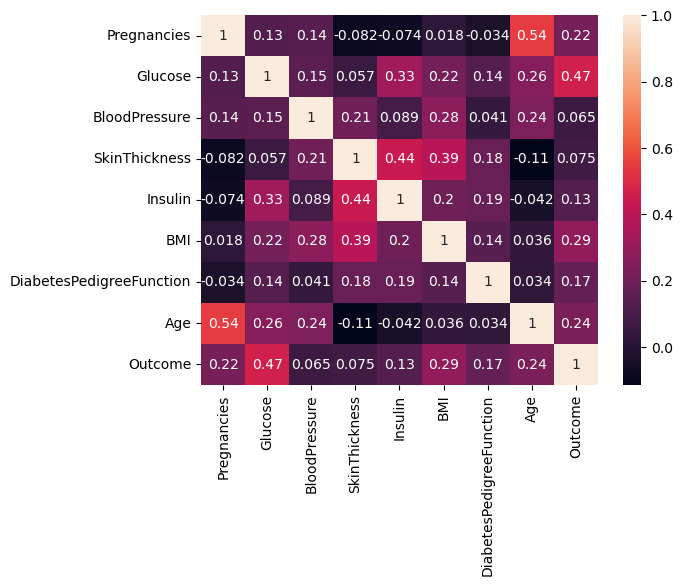
یکی از مهم­ترین چالش­هایی که در آماده سازی داده­ها با آن سر و کار داریم، داشتن مقادیر گمشده می­باشد. مقادیر گمشده ممکن است در پیش­بینی­های آینده یا در روند کار برای ما مشکلاتی را ایجاد کنند، از این رو شناسایی آن­ها حائز اهمیت است. کداجرا شده در شکل 9 تعداد داده­های گم شده را نمایش می­دهد:



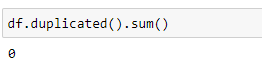
شکل 6:Pairplot

همانگونه که در شکل 10 مشاهده می­شود، هیچگونه مقدار گمشده­ای در این مجموعه داده موجود نمی­باشد. اما نکته قابل تامل در این بخش شناسایی داده­هایی با مقدار " ؟ " و یا " 0 " می­باشد. همان­طور که متوجه شده­اید با کد موجود در شکل 10 تنها مقادیری که وجود ندارند، مشخص می­شود. این در حالی است که گاهی وجود " 0 " برای یک شاخص معنی ندارد و حضور " ؟ " نیز در مجموعه داده به معنای عدم وجود مقدار است.

همانگونه که مشاهده می­شود برای شاخص­های بارداری، گلوکز، فشار خون، ضخامت پوست، انسولین، BMI و خروجی مجموعه داده، مقادیر صفر داریم. در این بین برای شاخص­های گلوکز، فشار خون، ضخامت پوست، انسولین و BMI ، مقدار صفر، مقداری نامعتبر و نشدنی است، پس نتیجه می­گیریم که این مقادیر در اصل ناموجود یا گمشده بوده­اند که با جای خالی گذاشتن، مقدارصفر را جایگزین کرده­اند.

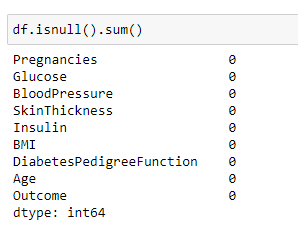


شکل 7: نمودار Heatmap

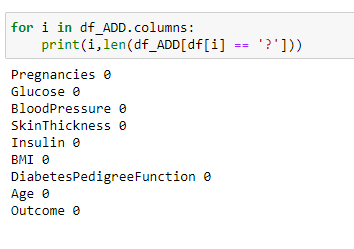


شکل 8: مشخص کردن داده¬های تکراری

پس از شناسایی داده­های گمشده، شناسایی داده­های پرت (نویز[[7]](#footnote-7)) را در دستور کار خواهیم داشت.



شکل 9: نمایش مقادیر گمشده



شکل 10: نمایش تعداد مقادیر " ؟ "

همانگونه که مشاهده می­شود، هیچگونه داده­ای با مقدار " ؟ " وجود ندارد. حال تعداد " 0 " را نیز در شکل 11 بررسی می­کنیم.

### 3-4-2 پیدا کردن مقادیر پرت

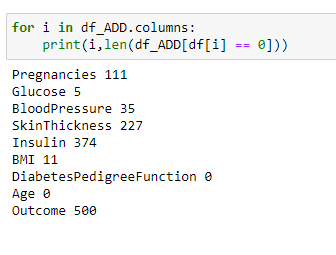
نویز، داده­هایی را شامل می­شود که به هر دلیلی دارای مقادیر نادرست یا غلط باشند. مانند عکس­های تار در دوربین ثبت تخلف، تفاوت در واحد پول چند ویژگی مختلف

داده­های پرت نیز به داده­هایی گفته می­شود که در فا­صله­ی غیرعادی از سایر مقادیر داده در یک نمونه­ی تصادفی از یک

جمعیت مشاهده می­شود.

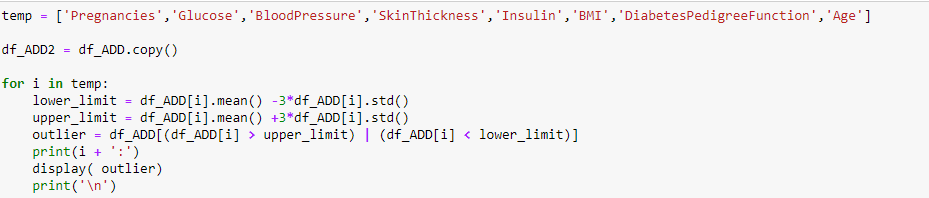
شناسایی نویزها و داده­های پرت امری حیاتی و تاثیرگذار در مبحث آماده سازی داده­ها می­باشد. از این رو باید در صدد رفع و شناسایی علت آن­ها باشیم. روش­های شناسایی این مشکلات عبارتند از:

* شناسایی نقطه پرت بر اساس مفهوم آماری پراکندگی
* استفاده از روش­های خوشه بندی
* گسسته کردن داده­ها



شکل 11: نمایش مقادیر " 0 "

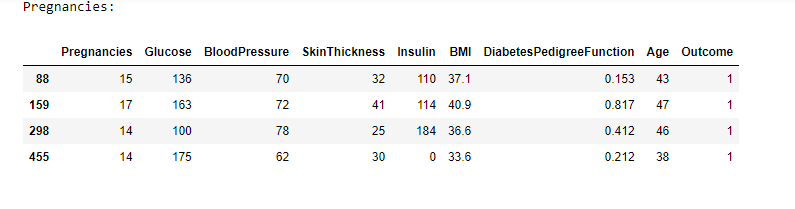
در این پژوهش، همانگونه که در شکل 12 قابل مشاهده است، برای شناسایی مقادیر پرت از روش شش سیگما [[8]](#footnote-8) ، استفاده کرده*­­*ایم. این روش مقادیر مثبت بیشتر از 3 سیگما و مقادیر منفی کمتر از 3 سیگما را داده پرت شناسایی می­کند.



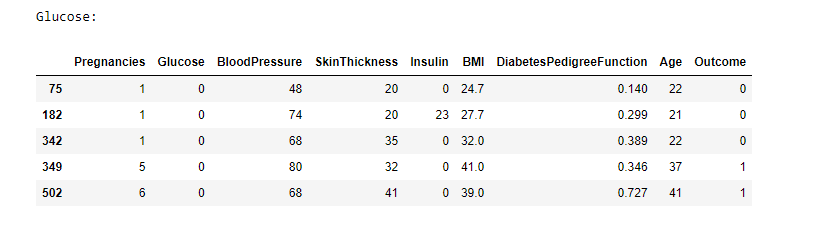
شکل 12: کد مربوط به نمایش مقادیر پرت

در جدول­های 4،3، 5، 6، 7، 8، 9 و 10 مقادیر پرت هر شاخص نمایش داده شده است:

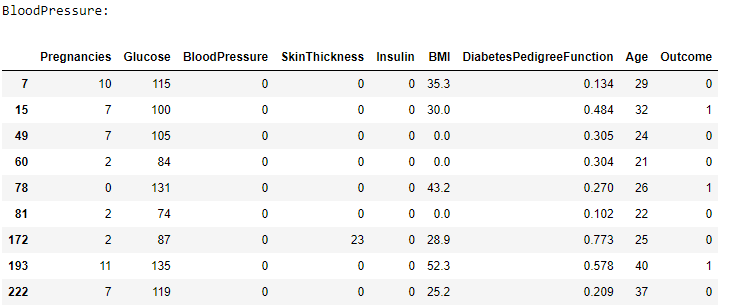
جدول 3: مقادیر پرت شاخص بارداری



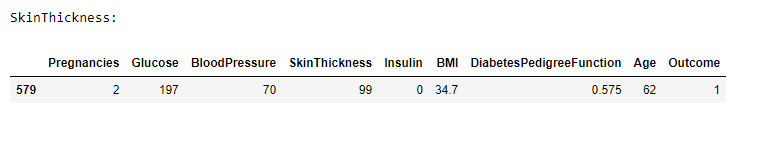
جدول 4: مقادیر پرت شاخص گلوکز



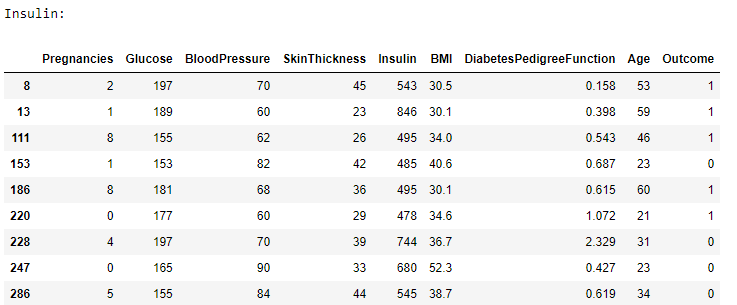
جدول 5 : مقادیر پرت شاخص فشار خون



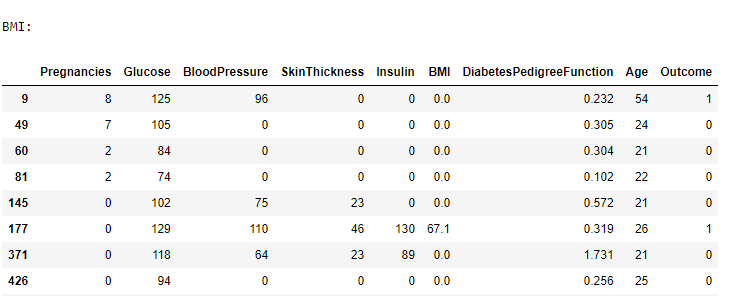
جدول 6: مقادیر پرت شاخص ضخامت پوست



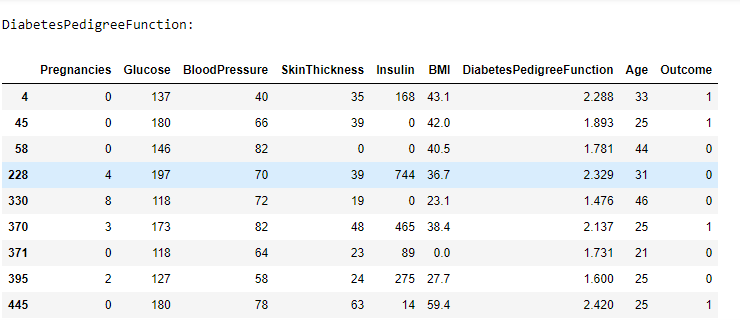
جدول 7: بخشی از مقادیر پرت شاخص انسولین



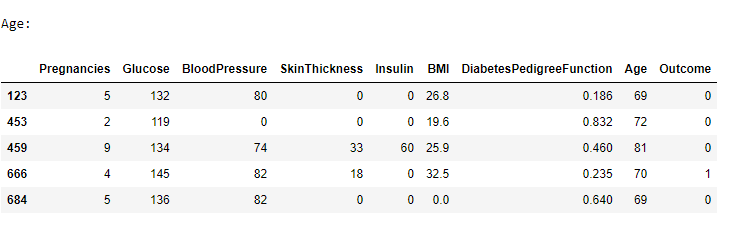
جدول 8: مقادیر پرت شاخص BMI



جدول 9: مقادیر پرت شاخص DiabetesPedigreeFunction



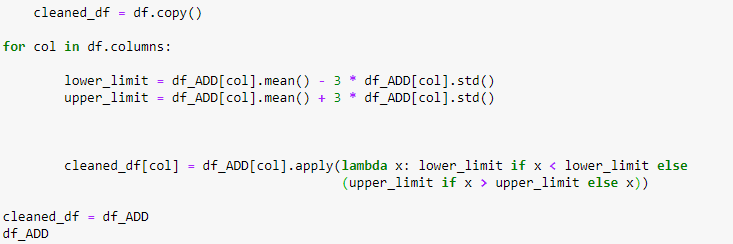
جدول 10: مقادیر پرت شاخص سن



برای رفع مشکل وجود داده­های پرت، چندین راه حل وجود دارد:

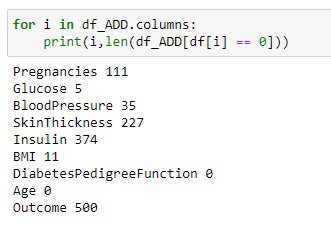
* حذف داده­های پرت
* جایگذاری داده­های پرت با مقدار میانگین
* جایگذاری داده­های پرت با مقادیر روی حد بالا و پایین

در این بخش، مطابق شکل 13، داده­های بیشتر از حد را با مقادیر روی حد بالا و کمتر را با مقادیر روی حد پایین جایگذاری می­کنیم.



شکل 13: جایگذاری دادههای پرت با مقادیر حد بالا و پایین

در این بخش به رفع مشکل داده­های گمشده می­پردازیم. با توجه به جایگذاری داده­های پرت با مقادیر حد بالا و پایین، در ابتدا مجددا تعداد صفرهای موجود در هر شاخص را بررسی می­کنیم، زیرا ممکن است تعدادی از صفرها حذف و یا به تعداد آن­ها افزوده شده باشد. شکل 14، این مهم را نمایش می­دهد.



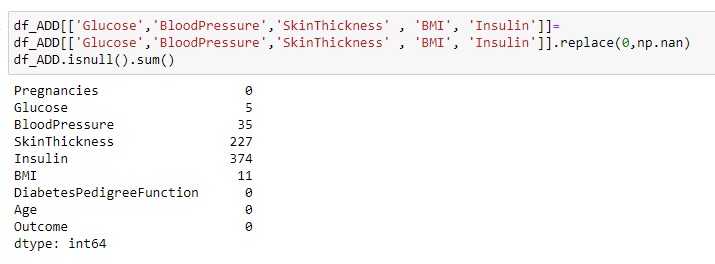
شکل 14: نمایش مجدد تعداد مقادیر " 0 "

با توجه به آنچه که از خروجی شکل 14 به دست آمده است، متوجه می­شویم که شاهد تغییر خاصی در مقادیر " 0 " نیستیم. حال مطابق شکل 15، تمامی مقادیر " 0 " را با مقدار nan جایگذاری می­کنیم.

از این بخش به بعد، باید به دنبال جایگذاری داده­های گمشده باشیم، روش­های مقابله با این مشکل عبارتند از :

* حذف مقادیر گمشده
* جایگذاری آن­ها با میانگین هر شاخص
* پر کردن داده­ها با مقادیر قبلی یا بعدی
* پر کردن با استفاده از مقادیر مشابه
* پر کردن با استفاده از مدل­های پیش­بینی

در این بخش، با استفاده از روش‌های Iterative ، Mean و Single به رفع این مشکل پرداخته ایم. روش IterativeImputer در ماشین لرنینگ یک روش پر کردن مقادیر گمشده در داده‌ها است.



شکل 15: جایگذاری مقادیر nan به جای " 0 "

این روش، بر اساس روشی تکراری عمل می‌کند. ابتدا الگوریتم یک مدل پیش‌بینی را انتخاب می‌کند، مانند یک مدل رگرسیون یا یک مدل شبکه عصبی. سپس مقادیر گم شده را با استفاده از این مدل پیش‌بینی می‌کند.

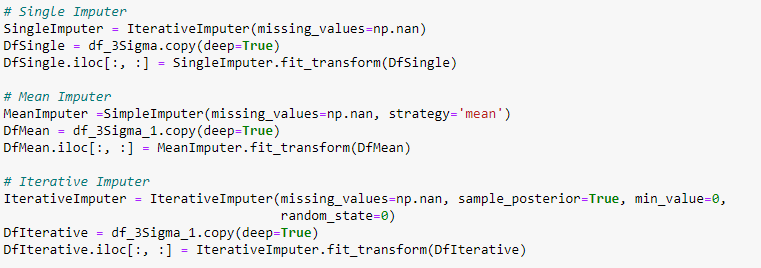
مرحله بعدی این است که مقادیر پیش‌بینی شده را به عنوان مقادیر جدید برای مقادیر گم شده استفاده کنیم و این مقادیر را در داده‌ها جایگزین کنیم. با این کار، مقادیر گم شده جایگزین شده و یک نسخه جدید از داده‌ها به دست می‌آید. این فرآیند تکرار می‌شود و هر بار مدل پیش‌بینی بر اساس داده‌های جدید آموزش داده می‌شود. این روند تا زمانی ادامه پیدا می‌کند که مقادیر گم شده به اندازه کافی پر شوند یا معیاری مانند تعداد تکرارها مشخص شود.

روش IterativeImputer برای پر کردن مقادیر گمشده از اطلاعات موجود در داده‌ها بهره می‌برد. با استفاده از روش تکراری، مدل پیش‌بینی تلاش می‌کند تا از روابط و الگوهای موجود در داده‌ها استفاده کند و مقادیر مفقود را تخمین بزند.

Single Imputer روشی است که مقادیر گمشده را با استفاده از یک مقدار واحد برای هر ویژگی در مجموعه داده به حساب می‌آورد.

Mean Imputer محاسبه‌گر میانگین نوعی از محاسبه‌گر منفرد است که مقادیر گمشده را با استفاده از مقدار میانگین ویژگی محاسبه می‌کند.

شکل­های 16 و 17 کدهای مربوط را نمایش می­دهند.

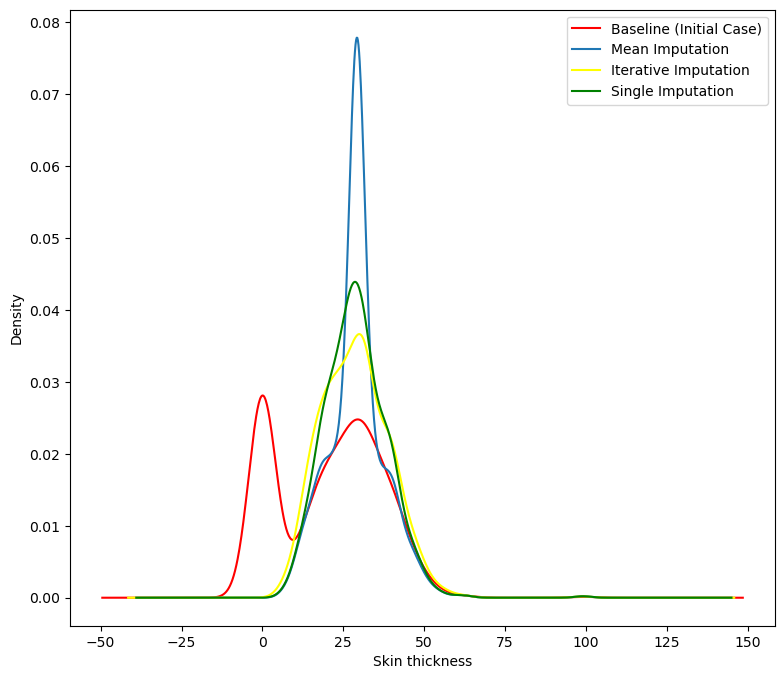


شکل 16: کد اجرا شده برای جایگذاری مقادیر گمشده



شکل 17: کد اجرا شده برای نمایش نمودار مقایسه¬ای هر یک از روش¬ها

در شکل 18 سه روش گفته شده برای پر کردن داده‌های گمشده را برای این ویژگی انجام دادیم، خط قرمز رنگ در نمودار خود داده اصلی هست. ما می‌خواهیم ببینیم کدوم یک از روش‌ها با این خط قرمز همپوشانی بیشتری دارد. در این بخش مشاهده می­شود که نمودار زرد رنگ تشابه بیشتری به نسبت بقیه نمودارها با نمودار اصلی دارد.

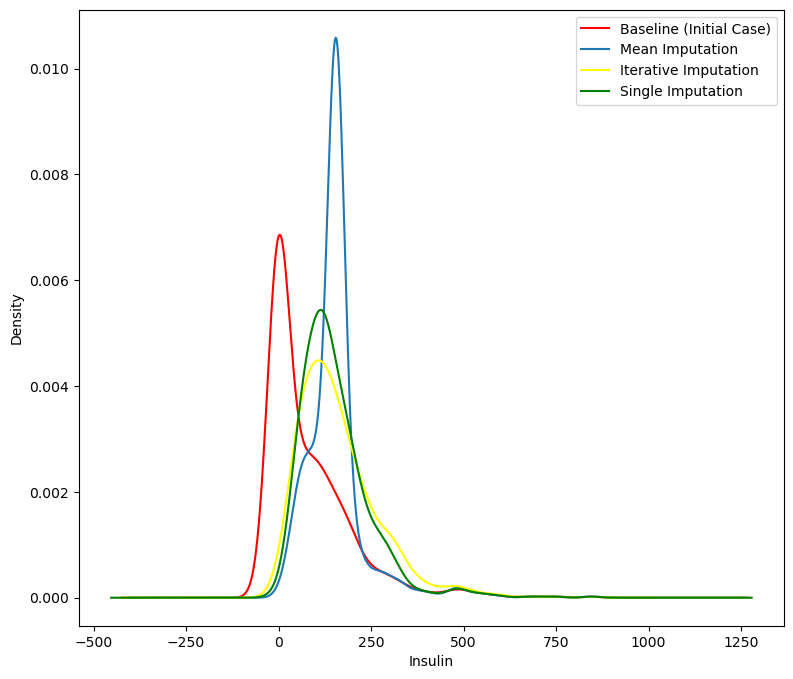


شکل 18: نمودار ضخامت پوست

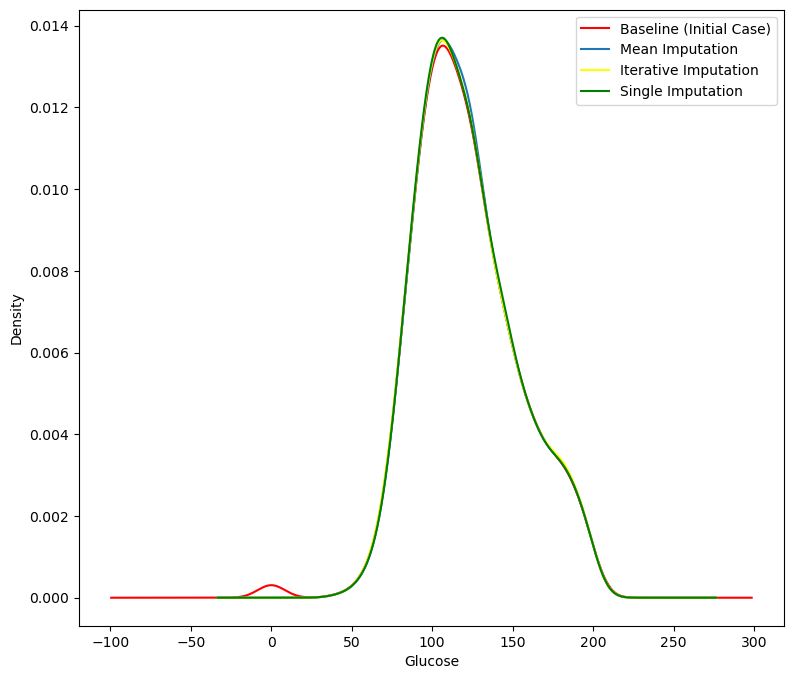
در شکل 19ویژگی انسولین را هم مانند قبل تحلیل می­کنیم. در این بخش نموداری که تشابه زیادی با نمودار قرمز رنگ داشته باشد موجود نیست. در شکل 20 برای نمودار گلوکز مشاهده می­کنیم که تمامی روش­های جایگذاری بر روی نمودار اصلی تقریبا منطبق شده­اند. این مهم به علت تعداد کم داده­های گمشده برای این ویژگی می­باشد. اما با توجه و دقت بیشتر مشاهده می­کنیم که نمودار زرد رنگ نزدیکی بیشتری به نمودار اصلی دارد. در شکل 21، نمودار فشارخون، تحلیل مانند قبل صورت می­گیرد. نزدیکترین نمودار به نمودار اصلی، نمودار زرد رنگ می­باشد.

با مشاهده نمودار شکل 22 برای نمودار BMI، متوجه می­شویم که این ویژگی نیز عملکردی مشابه ویژگی­های قبل دارد. نزیکترین نمودار به نمودار اصلی، نمودار زرد رنگ می­باشد.

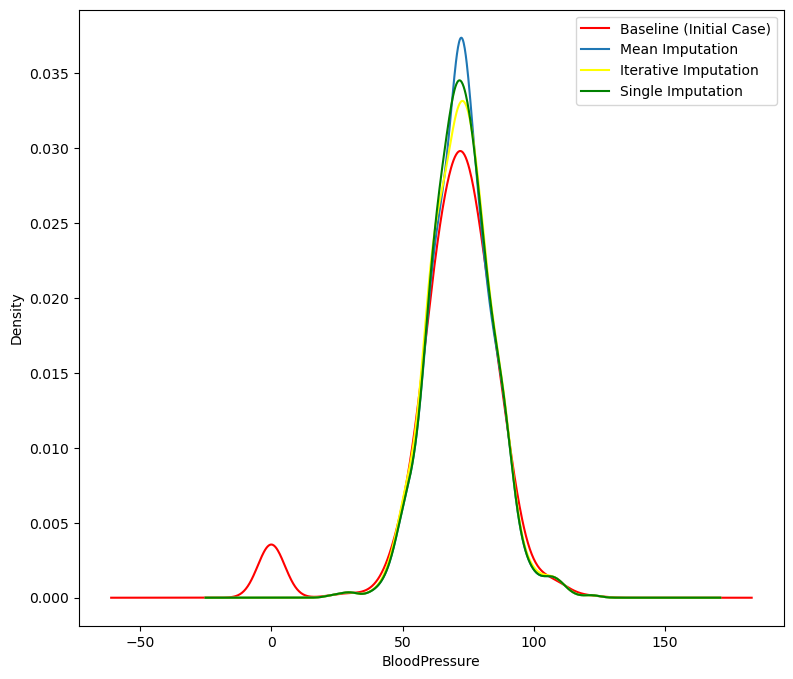
طبق مشاهدات صورت گرفته در این بخش، نزدیکترین مدل جایگذاری مقادیر گمشده به مجموعه داده اصلی روش IterativeImputer می­باشد.



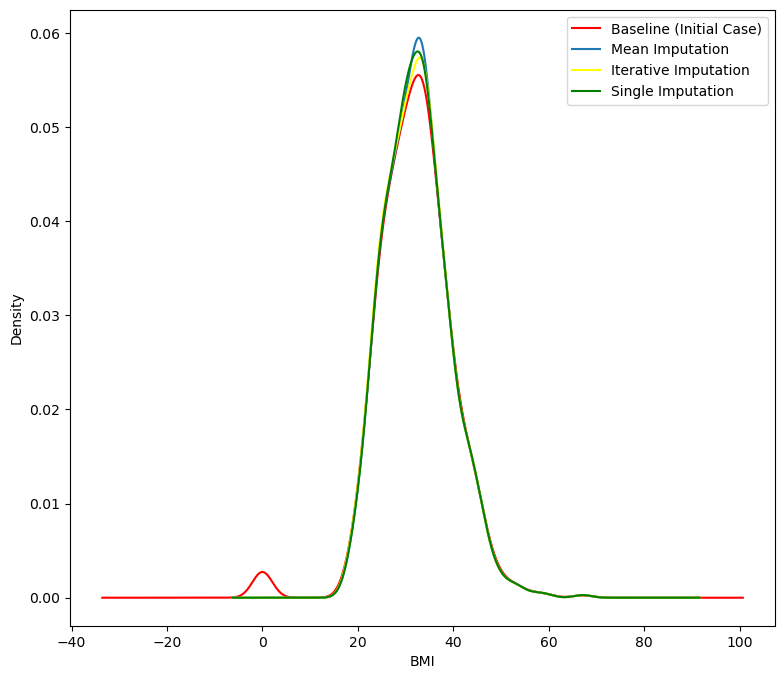
شکل 19: نمودار انسولین



شکل 20: نمودار گلوکز



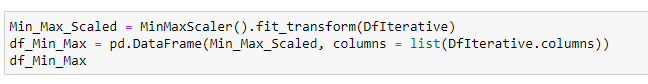
شکل 21: نمودار فشارخون



شکل 22: نمودار BMI

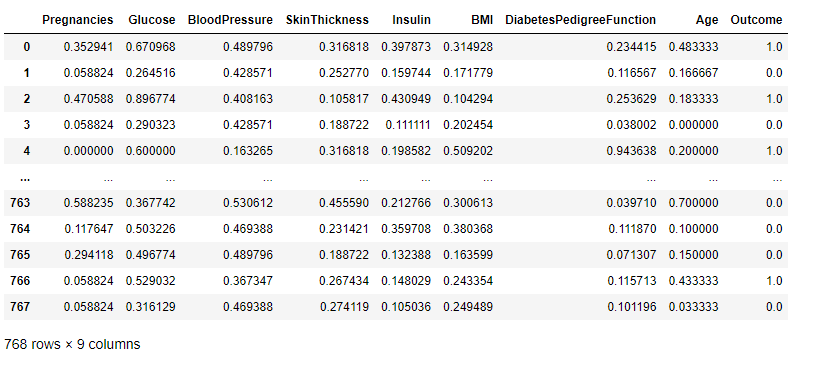
### 4-4-2 استاندارد سازی داد­ه­ها

در این بخش برای استانداردسازی داده ها از روش مین مکس استفاده کرده­ایم(شکل شماره 23). این روش، نوعی الگوریتم بهینه سازی است که برای یافتن بهترین راه حل ممکن برای یک مسئله معین با جست و جو در تمام راه حل­های ممکن و انتخاب راه حل با بالترین ارزش یا کمترین هزینه استفاده می­شود. با به حداقل رساندن حداکثر هزینه یا به حداکثر رساندن حداقل مقدار یک مجموعه معین از پارامترها کار می­کند. خروجی داده­ها در جدول 11 نمایش داده شده است.



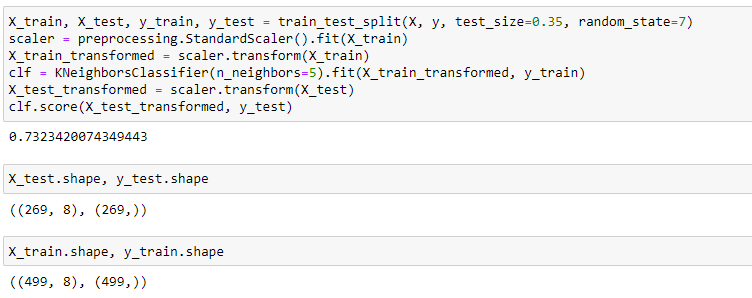
شکل 23: استاندارد سازی داده

جدول 11: داده‌های استاندارد شده به روش مین مکس



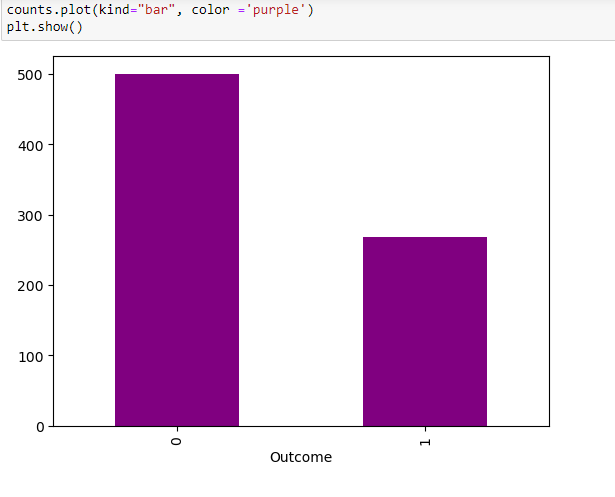
### 5-4-2 **تقسیم بندی داده­ها به دو دسته آموزشی و آزمایشی**

یکی از مهم­ترین کارها در پردازش داده، تقسیم بندی داده­ها به گروه­­های آموزشی و آزمایشی می­باشد. در شکل 24 ،35 درصد از داده­ها را به دسته آزمایشی و 65 درصد را به آموزشی تقسیم بندی می­کنیم.

******

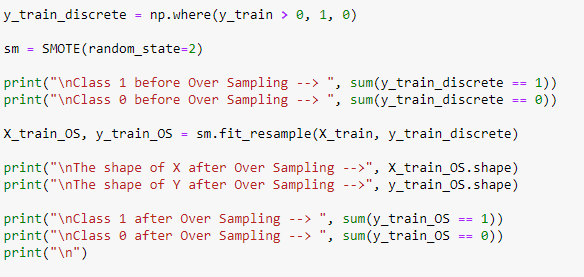
شکل 24: کد تقسیم داده¬ها به دو دسته آموزشی و آزمایشی

در این بخش می­خواهیم داده را متعادل کنیم. شکل شماره 25 تعداد داده­های موجود در هر کلاس را نمایش می­دهد. با توجه به نمودار مربوط خواهیم دید که داده­ها متعادل نیستند.

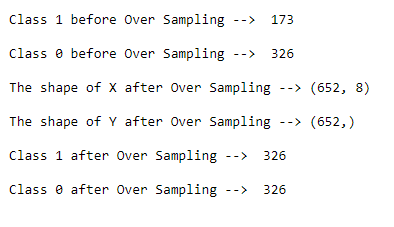
**

شکل 25: نمایش نمودار میله­ای داده­ها

همانگونه که می­دانیم برای متعادل سازی داده­ها، چندین روش وجود دارد. در شکل 26، از روش OverSampling برای متعادل سازی داده­ها استفاده کرده­ایم. نتایج خروجی آن در شکل 27 قابل مشاهده می­باشد.



شکل 26: کد متعادل سازی مجموعه داده

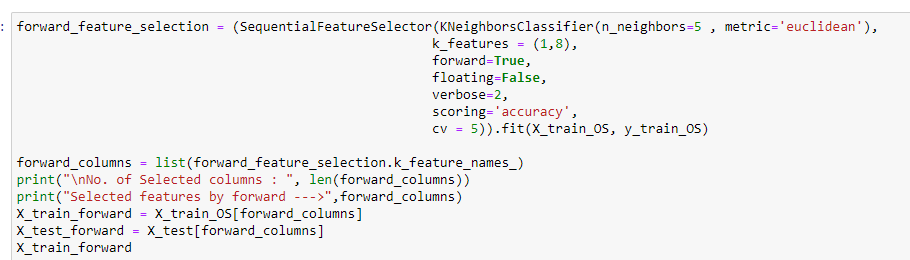


شکل 27: خروجی پس از متعادل سازی مجموعه داده

### 6-4-2 انتخاب ویژگی با روش جست و جو رو به جلو

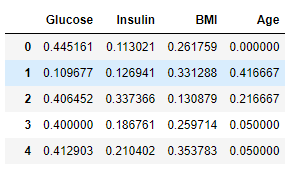
انتخاب ویژگی، به معنی انتخاب زیرمجموعه­ای از ویژگی­ها یا فاکتورهای مرتبط با مجموعه داده­ها می­باشد که بیشترین اطلاعات مفید را برای مدلسازی فراهم می­کنند. زمانی که از روش انتخاب ویژگی استفاده می­کنیم، تعداد ویژگی­های ورودی به مدل کاهش پیدا می­کند. دلیل استفاده از این روش این است که معمولاً مدل­های یادگیری ماشین برای مجموعه داده­ها با تعداد ویژگی­های بالا ، به دلیل ابعاد بالا و مواجه شدن با عملیات محاسباتی زیاد و در نهایت نتایج ضعیف، می­توانند با مشکل مواجه شوند. بنابراین، با کاهش تعداد ویژگی­ها، می­توانیم هر دو مشکل را حل کنیم. با این کار، میتوانیم از بار محاسباتی کمتری در مدلسازی استفاده کرده و در مقایسه با مجموعه داده­های اولیه، مدل ایجاد شده با دقت بیشتری پیش­بینی و پایداری داشته باشد .

روش رو به جلو در انتخاب ویژگی تکنیکی است برای انتخاب زیرمجموع­هایی از ویژگی­ها از داده­های اصلی که بیشترین ارتباط را برای کار پیش­بینی دارند. ایده این است که با مجموعه­ای خالی از ویژگی­ها شروع کنید و هر بار یک ویژگی را بر اساس میزان بهبود عملکرد مدل اضافه کنید. این فرآیند تا زمانی تکرار میشود که با افزودن ویژگی­های بیشتر، بهبود بیشتری حاصل نشود. شکل شماره 28 کد این بخش را نمایش می­دهد.



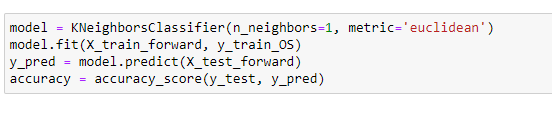
شکل 28: کد روش جست و جو رو به جلو

در نهایت با استفاده از این روش، ویژگی­های موجود در شکل 29 را خواهیم داشت:

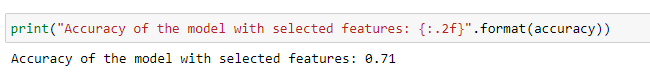


شکل 29: ویژگی انتخابی از روش جست و جو رو به جلو

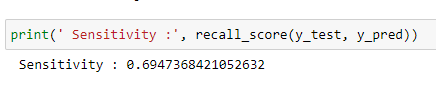
در ادامه برای اعتبار سنجی روش ارائه شده، معیارهای دقت در شکل 30، حساسیت در شکل 31و اختصاصیت در شکل 32 را گزارش می­کنیم:

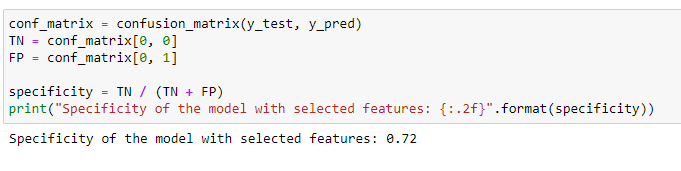


شکل 30: ارائه مدل برای اعتبارسنجی



شکل 31: گزارش دقت



شکل 31: گزارش حساسیت

شکل 32: گزارش اختصاصیت

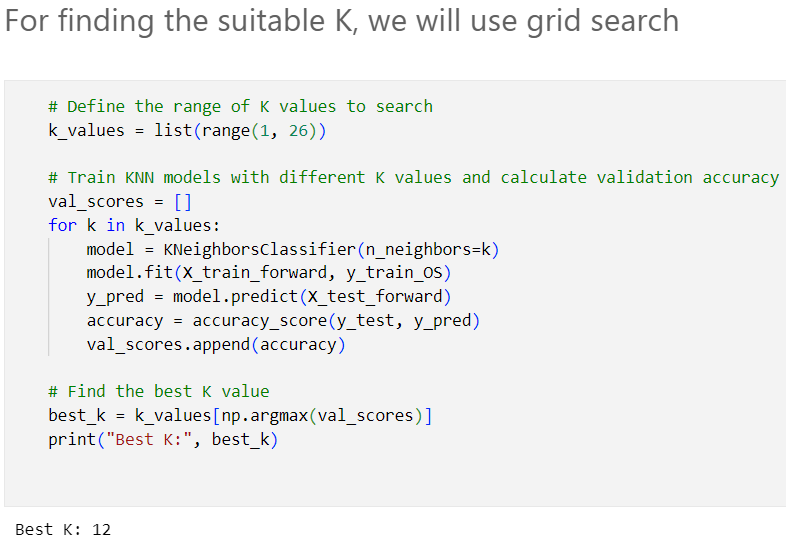
با توجه به معیارهای گزارش شده، میتوانیم نتیجه بگیریم، روش انتخاب ویژگی از سه منظر دقت، حساسیت و اختصاصیت، عملکرد مطلوب و قابل قبولی داشته است.

# 3-اجرای الگوریتم KNN

**صورت سوال**: حال از روش - kNN جهت دستهبندی استفاده کنید. 65 درصد دادهها را برای آموزش و الباقی را برای تست به کار ببرید. ابتدا کای مناسب را بر اساس دقت بدست آورید. سپس ماتریس در هم ریختگی را تشکیل داده و نتایج مندرج در آن را بر اساس معیارهای دقت، حساسیت و اختصاصیت شرح دهید. همچنین در ادامه با روش leave-one-out عملکرد روش KNN را ارزیابی نموده و معیارهای دقت، حساسیت و اختصاصیت را گزارش کنید.

## 2-3-پیدا کردن تعداد همسایه مناسب

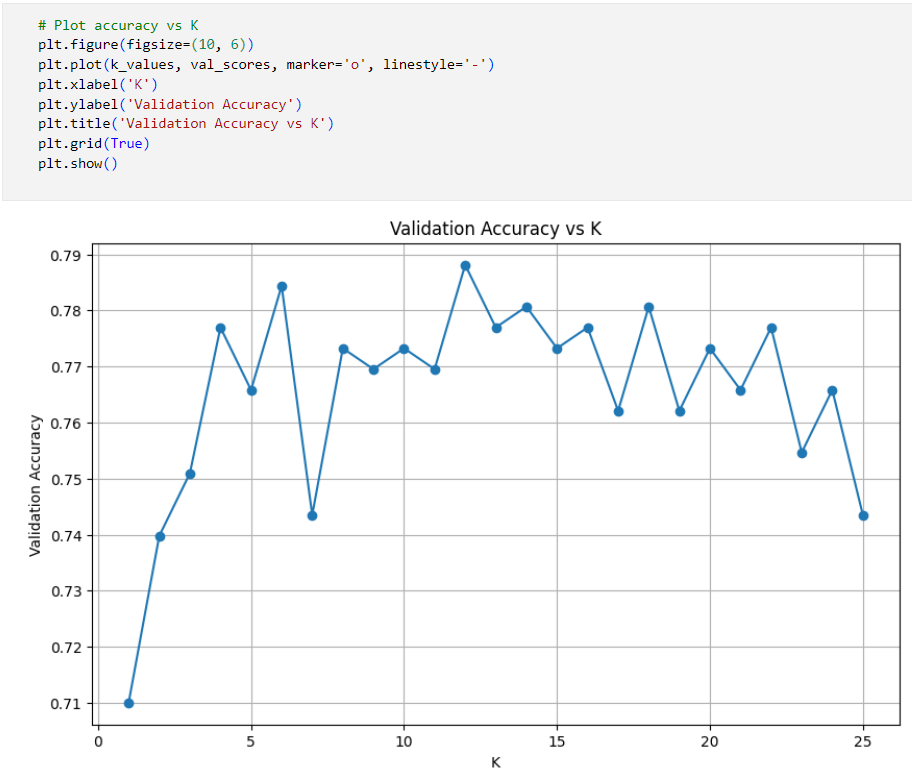
برای پیدا کردن K مناسب ابتدا محدوده‌ی مناسب برای K را تعریف می‌کنیم. حد بالای مقدار K برابر با است. 652 داده آموزشی داریم، ریشه دوم این عدد برابر با 25.53 است. بنابراین یک حد بالای مناسب برای جست و جو را می‌توان در نظر گرفت. برای پیدا کردن K مناسب از یک حلقه استفاده می‌کنیم. به طوری که ابتدا K را برابر 1 قرار می‌دهیم، مدلی را با این پارامتر آموزش می‌دهیم و تست می‌کنیم، مقدار دقت را محاسبه می‌کنیم و مقدار به دست آمده را در یک لیست ذخیره می‌کنیم. سپس با استفاده از تابع argmax() از کتابخانه نامپای، K متناظر با بیشترین مقدار دقت را نمایش می‌دهیم. شکل 33 کدی که وظیفه پیدا کردن K را به عهده دارد را نمایش می‌دهد.



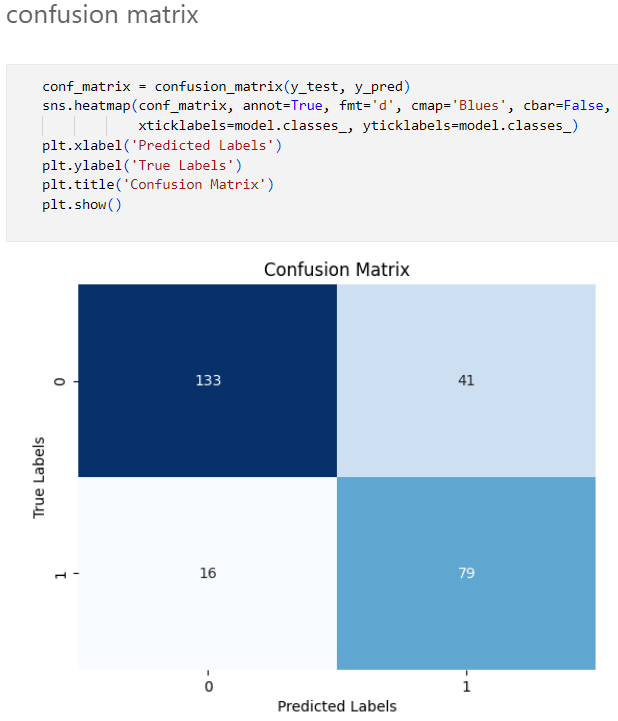
شکل 33: کد پیدا کردن K بهینه

با توجه به بررسی صورت گرفته، بهترین دقت را دارد. همچنین نمودار دقت بر اساس K را نیز رسم می‌کنیم. شکل 34 این نمودار را نمایش می‌دهد.

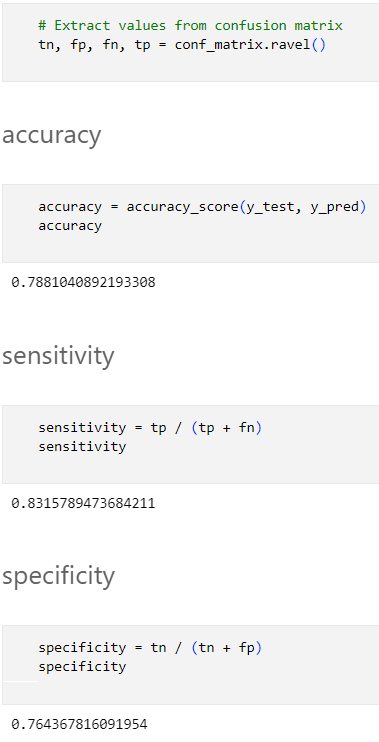
حال بار دیگر با پارامتر و همین دیتاست، مدل را آموزش و ارزیابی می‌کنیم. سپس ماتریس درهم‌ریختگی، و معیارهای دقت، حساسیت و اختصاصیت را با استفاده از فرمول‌های مربوطه محاسبه می‌کنیم. برای محاسبه دقت از تابع پیش‌ساخته accuracy\_score() در سایکیت لرن استفاده می‌کنیم. شکل 35 ماتریس درهم ریختگی و شکل 36 محاسبه سه معیار را نمایش می‌دهد.



شکل 34: نمودار معیار دقت بر اساس K



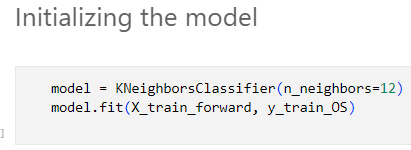
شکل 35: ماتریس درهم‌ریختگی و کد ایجاد کننده آن



شکل 36: محاسبه سه مقدار دقت، حساسیت و اختصاصیت به ترتیب

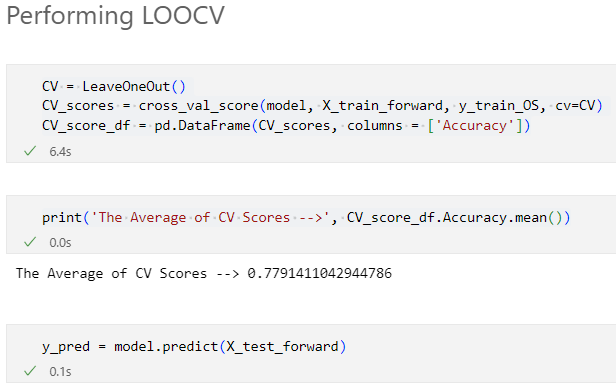
## 2-3-ارزیابی عملکرد KNN با روش Leave One Out

ابتدا یک مدل KNN با پارامتر ایجاد می‌کنیم و روی داده‌های آموزشی آن را آموزش می‌دهیم. شکل 37 نحوه انجام این کار را نمایش می‌دهد.



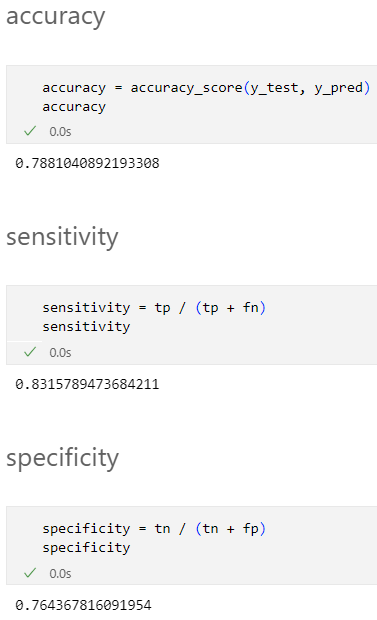
شکل 37: ایجاد مدل و آموزش مدل

حال مدل آموزش داده شده را با استفاده از روش Leave One Out ارزیابی می‌کنیم. ابتدا از کلاس LeaveOneOut() یک شی به نام CV ایجاد می‌کنیم. سپس با استفاده از تابع cross\_val\_score() ارزیابی را انجام می‌دهیم. این تابع، مدل ایجاد شده، دیتاست آموزشی و نحوه انجام ارزیابی را دریافت می‌کند و سپس داده‌ها را بر اساس نحوه انجام دریافت شده تقسیم می‌کند و مدل را ارزیابی می‌کند. سپس آرایه‌ی حاوی ارزیابی‌ها در یک دیتافریم ذخیره می‌شود و از آن میانگین گرفته می‌شود. شکل 38 نحوه کارکرد این کد را نمایش می‌دهد.



شکل 38: نحوه انجام ارزیابی Leave One Out

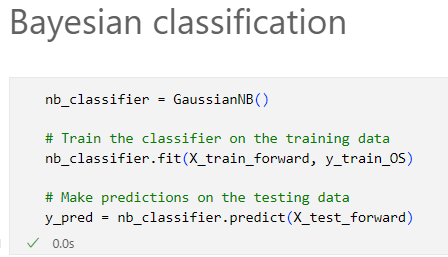
در ادامه سه معیار دقت، حساسیت و اختصاصیت این مدل محاسبه می‌شود. شکل 39 این سه مقدار را به ترتیب نمایش می‌دهد.



شکل 39: سه معیار دقت حساسیت و اختصاصیت حاصل از Leave One Out

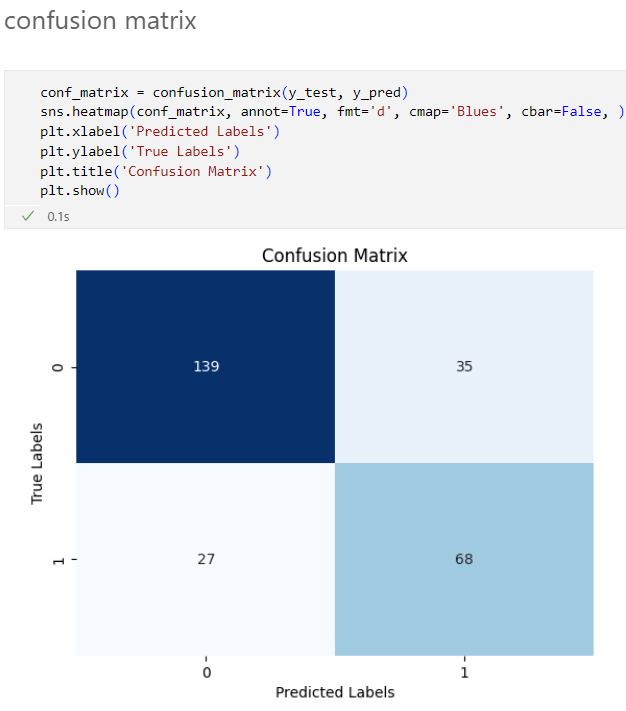
# 4-دسته‌بندی بیزی

برای انجام دسته‌بندی بیزی از همان دیتاست بخش قبلی استفاده می‌کنیم. با استفاده از کلاس GaussianNB() از ماژول scikit-learn مدل بیزی را آموزش می‌دهیم. برای این منظور ابتدا یک شی به نام nb\_classifier از این کلاس می‌سازیم و با پاس دادن دیتاست ویژگی‌ها و ستون لیبل متناظر آنها به عنوان آرگومان به متد fit()، مدل دسته‌بندی بیزی راآموزش می‌دهیم. در ادامه با استفاده از متد predict() و پاس دادن دیتاست تست، پیش‌بینی‌های مدل از این دیتاست را به دست می‌آوریم و در متغیر y\_pred ذخیره می‌کنیم. شکل 40 نحوه انجام این عمل را نمایش می‌دهد.

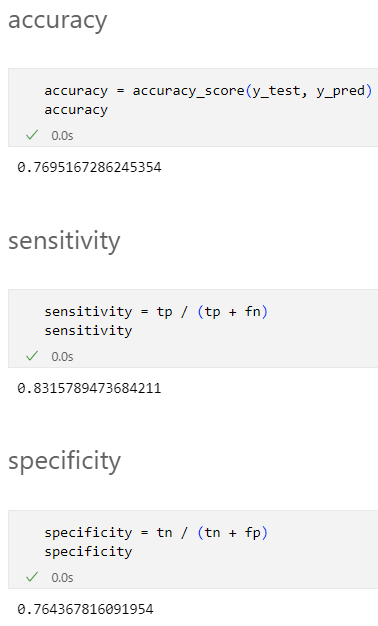


شکل 40: کد مربوط به دسته‌بندی بیزی

سپس همانند دو بخش قبلی به محاسبه ماتریس درهم‌ریختگی و معیارهای ارزیابی می‌پردازیم. شکل 41 ماتریس درهم‌ریختگی روش دسته‌بندی بیز را نشان می‌دهد. همچنین شکل 42 مقدار سه معیار دقت حساسیت و اختصاصیت را برای روش دسته‌بندی بیز نشان می‌دهد.



شکل 41: ماتریس درهم ریختگی حاصل از دسته‌بندی بیزی



شکل 42: سه معیار دقت حساسیت و اختصاصیت برای مدل دسته‌بندی بیزی

# 5- مسئله دوم

داده‌های زير را در نظر بگيريد)4,4) (3,4) (4,3) (3,3) (1,1) (0,1) (1,0) (0,0( داده ها فوق را در محور مختصات دو بعدى ترسيم نماييد. به صورت چشمى أنها را به دو كلاس تخصيص دهيد(برچسب داده ها را معين كنيد). ابتدا با كمک روش نزديكترين همسايگي، مرز دو كلاس را بيابيد. سپس با روش پرسپترون مرز دو کلاس را بیابید.

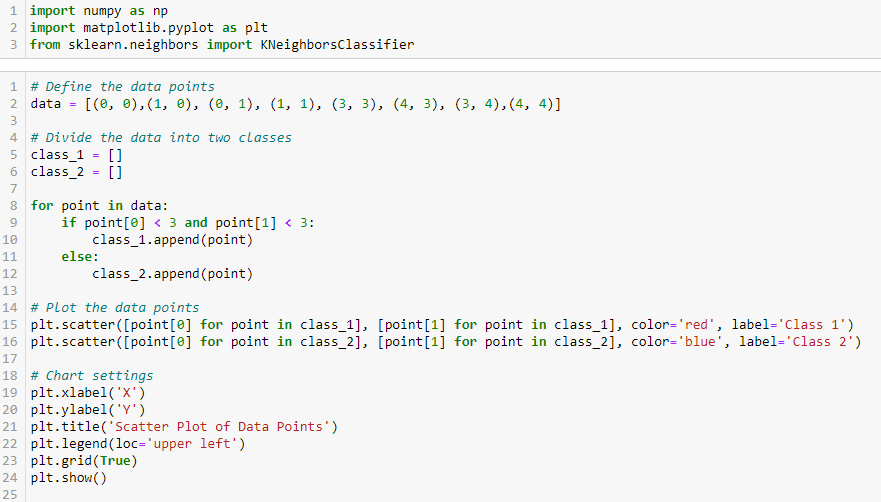
## 1-5 ترسيم در محور مختصات دو بعدى و تعیین برچسب داده ها

**تعریف داده‌ها:** داده‌ها به صورت یک لیست از جفت‌های مختصات (x,y) تعریف شده است.

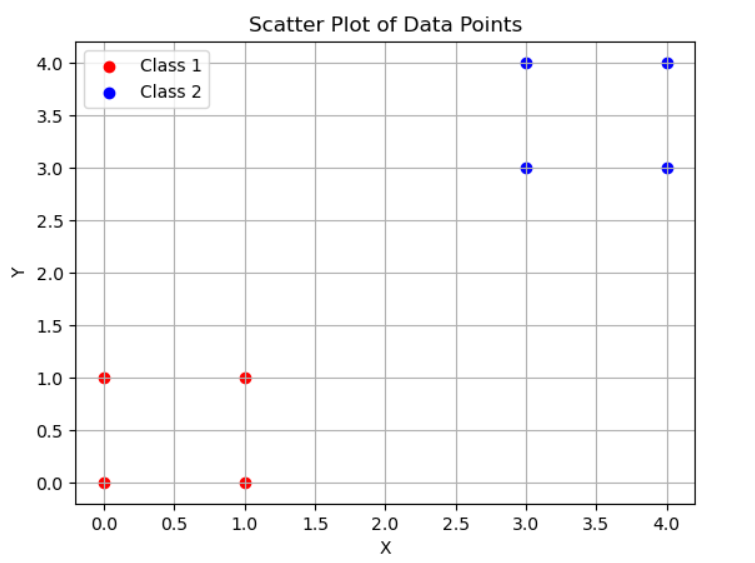
**تقسیم داده‌ها به دو کلاس:**  داده‌ها بر اساس موقعیتشان در محور مختصات دوبعدی به دو کلاس تقسیم می‌شوند. در اینجا، داده‌هایی که مختصات x و y آن‌ها کوچکتر 3 هستند در کلاس 1 و داده‌های دیگر در کلاس 2 قرار می‌گیرند.

**ترسیم داده‌ها:** با استفاده از تابع plt.scatter() داده‌ها در محور مختصات دوبعدی نمایش داده می‌شوند. داده‌های کلاس 1 با رنگ قرمز و داده‌های کلاس 2 با رنگ آبی نمایش داده شده‌اند.

**تنظیمات نمودار:** محور x و y به ترتیب به "X" و "Y" تنظیم شده‌اند. همچنین عنوان نمودار به "Scatter Plot of Data Points" تغییر یافته است. همچنین با استفاده از plt.legend() برچسب‌های کلاس‌ها به درستی به نمودار اضافه شده‌اند و با فراخوانی plt.grid(True) خطوط شبکه در نمودار نیز نمایش داده شده‌اند. و در نهایت با استفاده از plt.show() نمودار نمایش داده می‌شود. کد ها و خروجی های این بخش در شکل های 43 و 44 قرار گرفته شده است.



شکل 43:کد‌های قسمت 1



شکل 44:خروجی قسمت 1

## 2-5 تعیین مرز دو كلاس با روش نزديكترين همسايگي

**تعریف داده‌ها و برچسب‌ها:** در این قسمت، داده‌ها به صورت نقاطی در فضای دو بعدی تعریف شده‌اند. همچنین برچسب‌های متناظر با هر نقطه نیز تعیین شده‌اند که نشان می‌دهد هر نقطه به کدام دسته تعلق دارد. به عنوان مثال، اگر اولین چهار نقطه به یک دسته و آخرین چهار نقطه به دسته دیگر تعلق داشته باشند، برچسب‌های متناظر با این نقاط به ترتیب ۱ و ۲ است.

**ساخت مدل: KNN** در این بخش، یک مدل KNN با استفاده از کتابخانه scikit-learn ایجاد می‌شود. این مدل تعیین کننده‌ی تعداد همسایه‌هایی است که برای هر نقطه برای پیش‌بینی برچسب استفاده می‌شود. در اینجا k=1 است، بنابراین تنها نزدیک‌ترین نقطه برای هر نقطه برای پیش‌بینی برچسب در نظر گرفته می‌شود.

**آموزش مدل:** مدل KNN با استفاده از داده‌های آموزشی و برچسب‌هایش آموزش داده می‌شود. این به این معناست که مدل با داده‌های آموزشی تنظیم می‌شود تا بتواند برچسب‌های نقاط جدید را پیش‌بینی کند.

**نمایش مرز تصمیم و پیش‌بینی برچسب نقاط جدید:** بعد از آموزش مدل، می‌توان با استفاده از مدل برچسب‌های نقاط جدید را پیش‌بینی کرد. این کار با فراخوانی تابع predict انجام می‌شود که بر اساس مدل آموزش دیده، برچسب‌های نقاط جدید را پیش‌بینی می‌کند.

برای نمایش مرز تصمیم، محدوده‌ای از فضای داده‌ها (با استفاده از حداقل و حداکثر مقادیر x و y) مشخص شده و برای هر نقطه در این محدوده، برچسب پیش‌بینی شده توسط مدل رسم می‌شود. این کار باعث تشکیل یک مرز تصمیم بین دسته‌ها می‌شود که می‌تواند به صورت یک خط یا حتی یک منحنی نمایش داده شود.

این بخش از کد در واقع مربوط به ایجاد یک شبکه‌ی دقیق از نقاط برای ایجاد مرز تصمیم است. برای این کار، محدوده‌ای از مقادیر x و y را ایجاد می‌کند که در آن محدوده، اطراف داده‌ها همگرا شده‌اند. سپس یک شبکه از نقاط روی این محدوده ایجاد می‌شود. با استفاده از این شبکه، برای هر نقطه از آن، مقدار پیش‌بینی شده توسط مدل KNN محاسبه می‌شود. این مقادیر سپس برای رسم مرز تصمیم استفاده می‌شوند.

**خط13 کد:** این خط کمینه و بیشینه مقادیر x را مشخص می‌کند و سپس یک واحد اضافه می‌کند تا مرزهای داده را افزایش دهد. **خط14 کد** هم این کار را برای مقادیر y انجام می‌دهد.

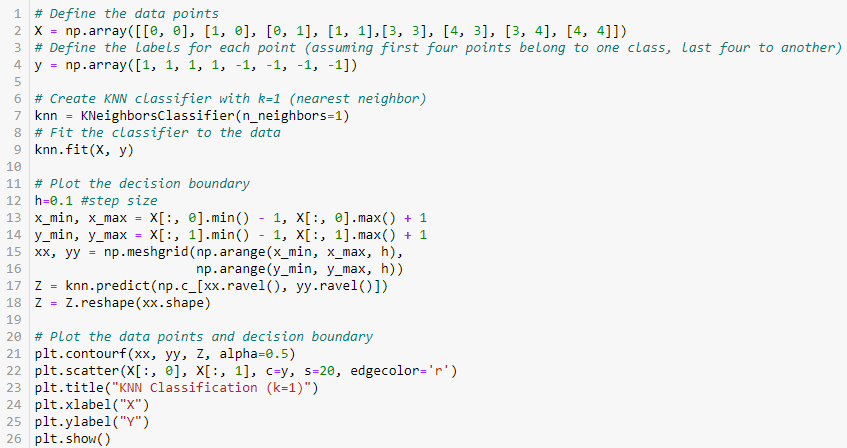
**خط15 و 16 کد:** این تابع یک شبکه از نقاط را ایجاد می‌کند که در آن مقادیر x و y از x\_min تا x\_max و y\_min تا y\_max با گام 0.1 قرار می‌گیرند.

**خط17 کد:** برای هر نقطه از شبکه، مقدار پیش‌بینی شده توسط مدل KNN محاسبه می‌شود.

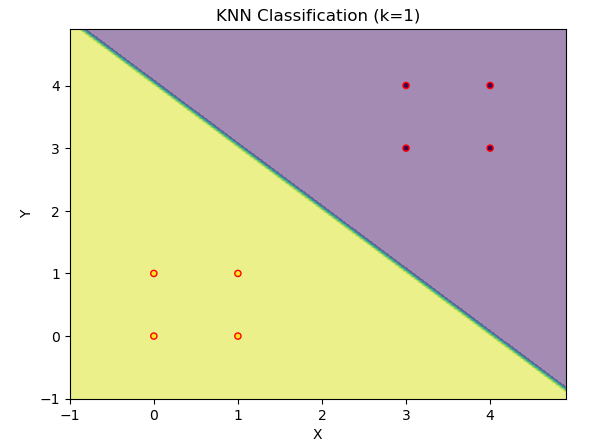
**خط18 کد:** شکل Z را با استفاده از ابعاد شبکه xx تغییر می‌دهد تا درست متناظر با آن شود.

در نهایت، یک نمودار با استفاده از کتابخانه matplotlib ایجاد می‌شود که داده‌ها به همراه مرز تصمیم بر روی آن نمایش داده می‌شود. این نمودار به ما کمک می‌کند تا دسته‌بندی و مرز تصمیم را به صورت بصری درک کنیم.

کد ها و خروجی های این بخش در شکل های 45 و 46 قرار گرفته شده است.



شکل 45: :کدهای روش NN



شکل 46: خروجی روش NN

## 3-5 تعیین مرز دو كلاس با روش پرسپترون

**تعریف داده‌ها و برچسب‌ها: X** یک ماتریس است که هر سطر آن نقطه‌ای در فضای دوبعدی است و y یک بردار است که برچسب متناظر با هر نقطه را نشان می‌دهد. به عبارت دیگر، برچسب‌های دسته‌بندی برای هر نقطه ارائه شده است.

**مقداردهی اولیه وزن‌ها و انحراف:**

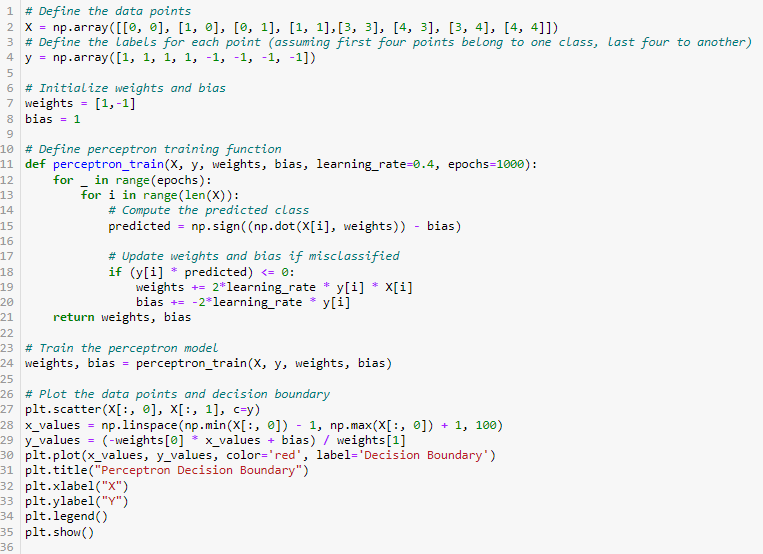
Weights یک بردار بر اساس لیبل های موجود تعریف میکنیم که با ابعاد برابر تعداد ویژگی‌ها (در اینجا ۲) است.

Bias به صورت رندوم 1 قرار می دهیم.

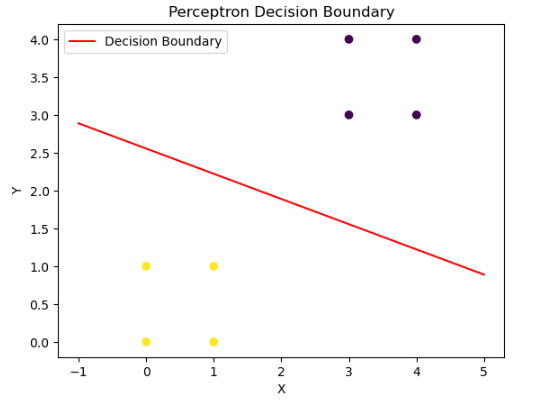
**تعریف تابع آموزش پرسپترون:** یک تابع با نام perceptron\_train تعریف شده است که مسئول آموزش مدل پرسپترون بر روی داده‌ها است.این تابع برای تعداد مشخصی از تکرارها (epochs) روی داده‌ها حلقه می‌زند و وزن‌ها و انحراف را بر اساس نقاط داده و برچسب‌هایشان به‌روزرسانی می‌کند.(به طور رندوم و با انجام تکرار های مختلف نرخ یادگیری 0.4 دارای بیشترین دقت که 80 درصد است را میدهد.)

**آموزش مدل پرسپترون:** مدل پرسپترون با استفاده از تابع آموزش perceptron\_train آموزش داده می‌شود. در هر مرحله از آموزش، وزن‌ها و انحراف به‌روزرسانی می‌شوند تا مدل بتواند به درستی داده‌ها را دسته‌بندی کند.

**نمایش مرز تصمیم:** مرز تصمیم بین دو دسته با استفاده از وزن‌ها و انحراف محاسبه شده و در نهایت روی نمودار نشان داده می‌شود (با استفاده از فرمول موجود). این مرز تصمیم به صورت یک خط در فضای داده نمایش داده می‌شود تا بتوانیم داده‌ها و دسته‌بندی آن‌ها را بصری‌سازی کنیم. کد ها و خروجی های این بخش در شکل های 47 و 48 قرار گرفته شده است.

****

شکل 47: کدهای روش پرسپترون



شکل 48: خروجی روش پرسپترون

1. Data Set [↑](#footnote-ref-1)
2. 2 مخزن داده­های استاندارد دانشگاه کالیفرنیا [↑](#footnote-ref-2)
3. K-Nearest­\_Neighbor [↑](#footnote-ref-3)
4. 4Body mass index  [↑](#footnote-ref-4)
5. 6Comma-Separated Values [↑](#footnote-ref-5)
6. نقشه حرارتی [↑](#footnote-ref-6)
7. Noise [↑](#footnote-ref-7)
8. Six Sigma [↑](#footnote-ref-8)