Amirkabir University of Technology
(Tehran Polytechnic)

Applied Machine Learning Course

By Dr. Nazerfard

CE5501 | Spring 2024

Assignment (2)

Name: Esmaeil Khosravi

S_ID: 402131046

Email: es.khosravi@aut.ac.ir

سوال ۱

قسمت a

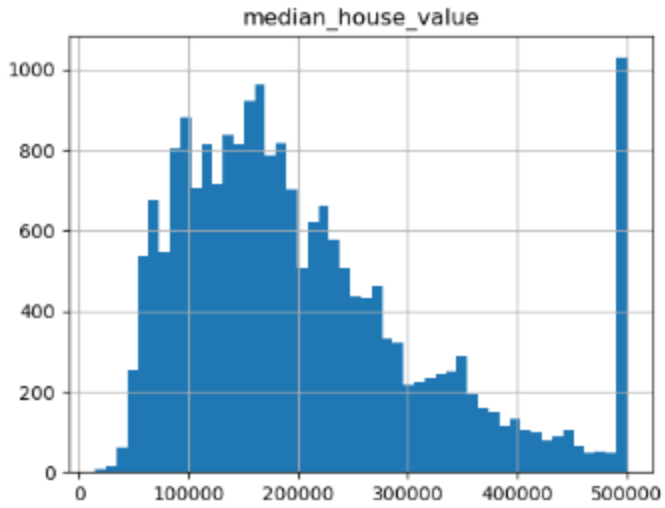
داده‌های مربوط به قیمت خانه‌های کالیفرنیا را از یک فایل **CSV** بارگیری می‌کند و یک نمونه از ۱۰ مورد تصادفی از داده را چاپ می‌کند.

قسمت b

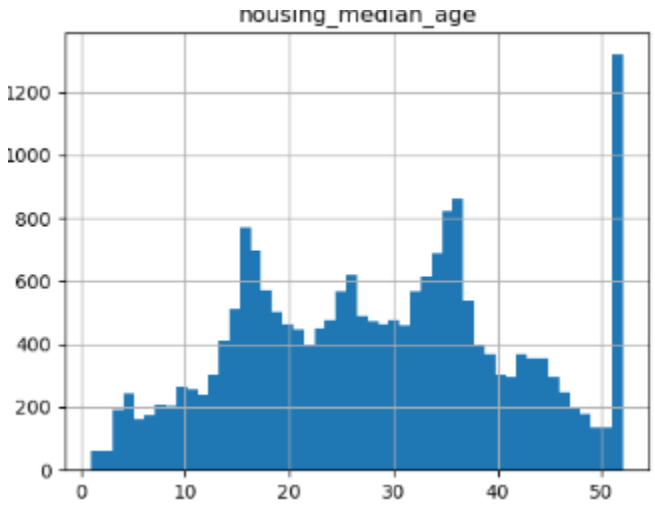
با استفاده از نمودارهای جفتی و هیستوگرام‌ها، داده‌ها را برای بررسی رابطه بین متغیرها و توزیع آن‌ها مشاهده می‌کند. همچنین نمودار های توزیع هر ویژگی نمایش داده شده اند.

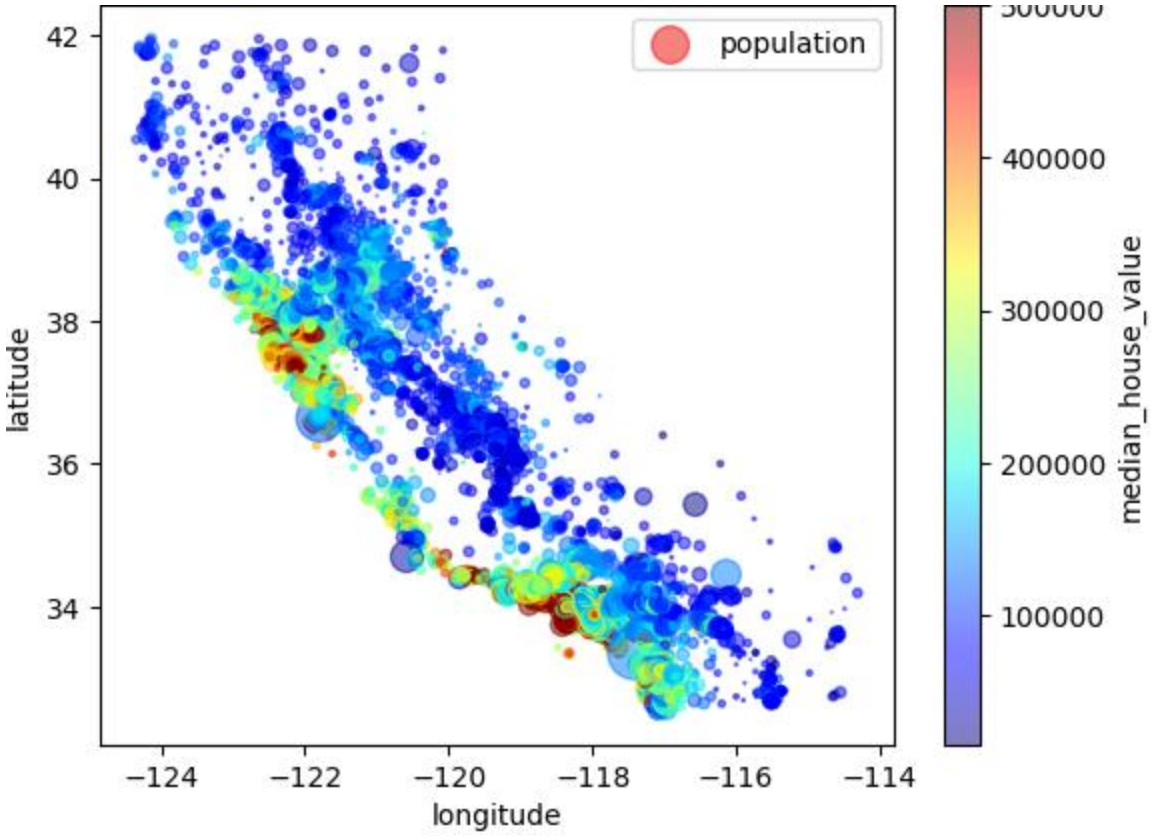
- نمودار **median_income** نشان میدهد که بیشتر درامد در بازه بین ۲۰ هزار دلار تا ۶۰ هزار دلار بوده است. (البته این داده ها به طور مقیاس بندی شده نشان داده شده اند.)

- نمودار **median_house_value** نشان میدهد که بیشتر قیمت خانه ها در بازه بین ۱۰۰هزار دلار تا ۳۰۰ هزار دلار بوده است.

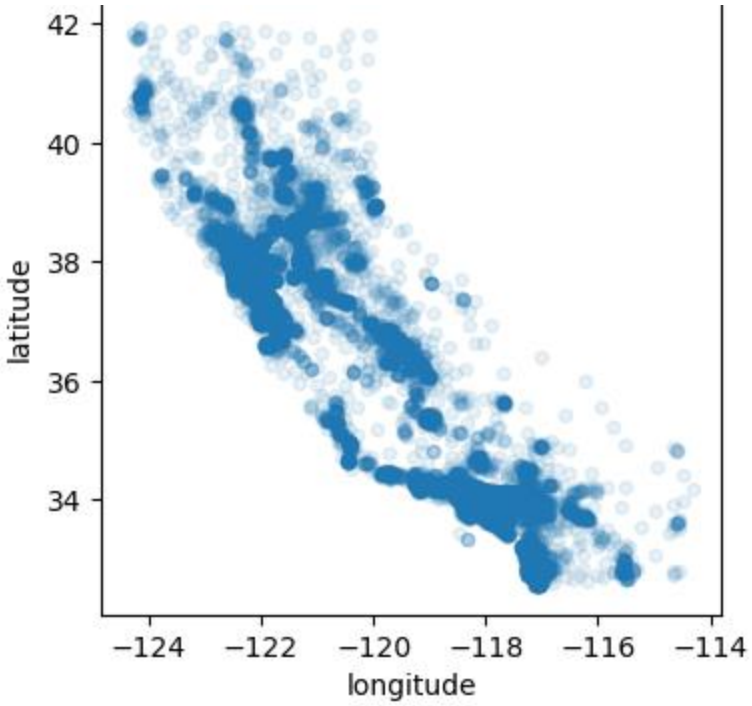


- نمودار **median_house_age** نشان میدهد که بیشتر عمر خانه ها در یک منطقه در بازه بین ۳۰ تا ۴۰ سال بوده است. همچنین تعداد خانه هایی که عمر ۵۰ ساله داشته اند نیز از بقیه بیشتر است.



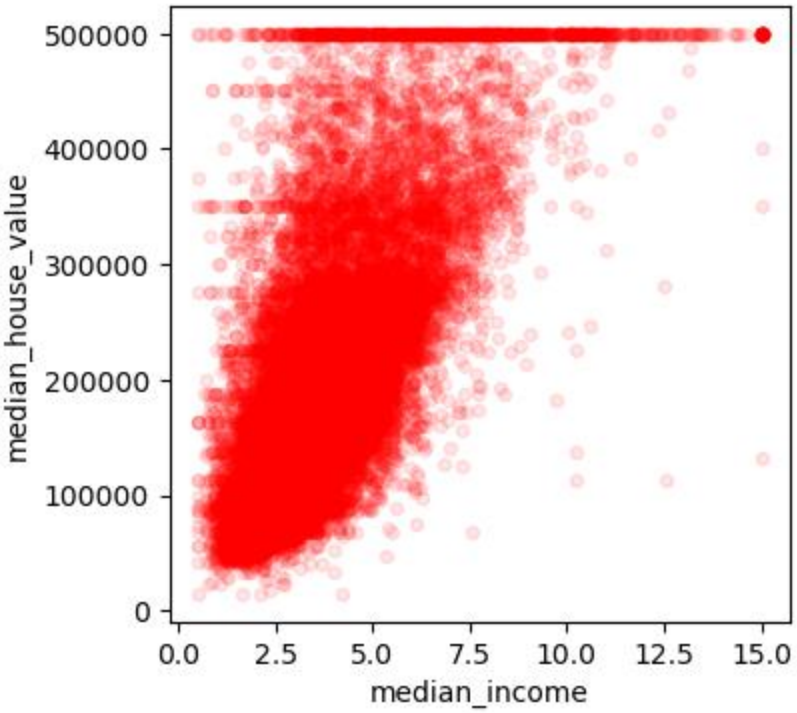


- در نمودار فوق عرض نشان دهنده عرض جغرافیایی و طول نشان دهنده ی طول جغرافیایی است. این نمودار نشان می دهد که قیمت خانه بسیار مرتبط با مکان آن بوده است به طوریکه مناطق با عرض جغرافیایی بیشتر قیمت بیشتری داشته اند. در نمودار فوق، دایره های با شعاع بیشتر تراکم بیشتر جمعیت را نشان می دهد. رنگ قرمز به معنای قیمت بیشتر و آبی قیمت کمتر را نشان می دهد.



نمودار فوق پراکندگی داد ها نسبت به مشخصات جغرافیایی طول و عرض را نشان می دهد. مناطق با تراکم جمعیت بالا پر رنگ تر هستند.(مناطقی مانند **(Bay Area, Los Angeles ,**

در ادامه نمودار های بیشتری در رابطه با همبستگی میان ویژگی ها نشان داده شده است که یک نمونه از آن در اینجا تحلیل میشود.



نمودار فوق یک همبستگی قوی بین دو ویژگی میانگین درامد و میانگین قیمت خانه ها را نشان می دهد. به طوریکه افراد با درآمد بالا خانه های گران تری خریده اند.

قسمت **c**

در این قسمت پاکسازی داده ها انجام میشود. با استفاده از متد **describe** اطلاعات آماری مربوط به هر ویژگی نشان داده شده است. با استفاده از متد **info** معلوم میشود ویژگی **total_bedrooms** دارای داده ناموجود است. سه روش عمده برای برخورد با داده ناموجود وجود دارد:

- حذف داده های ناموجود
- حدس آنها با میانگین یا میانه یا مقدار تصادفی

- **حذف ویژگی دارای داده ناموجود**

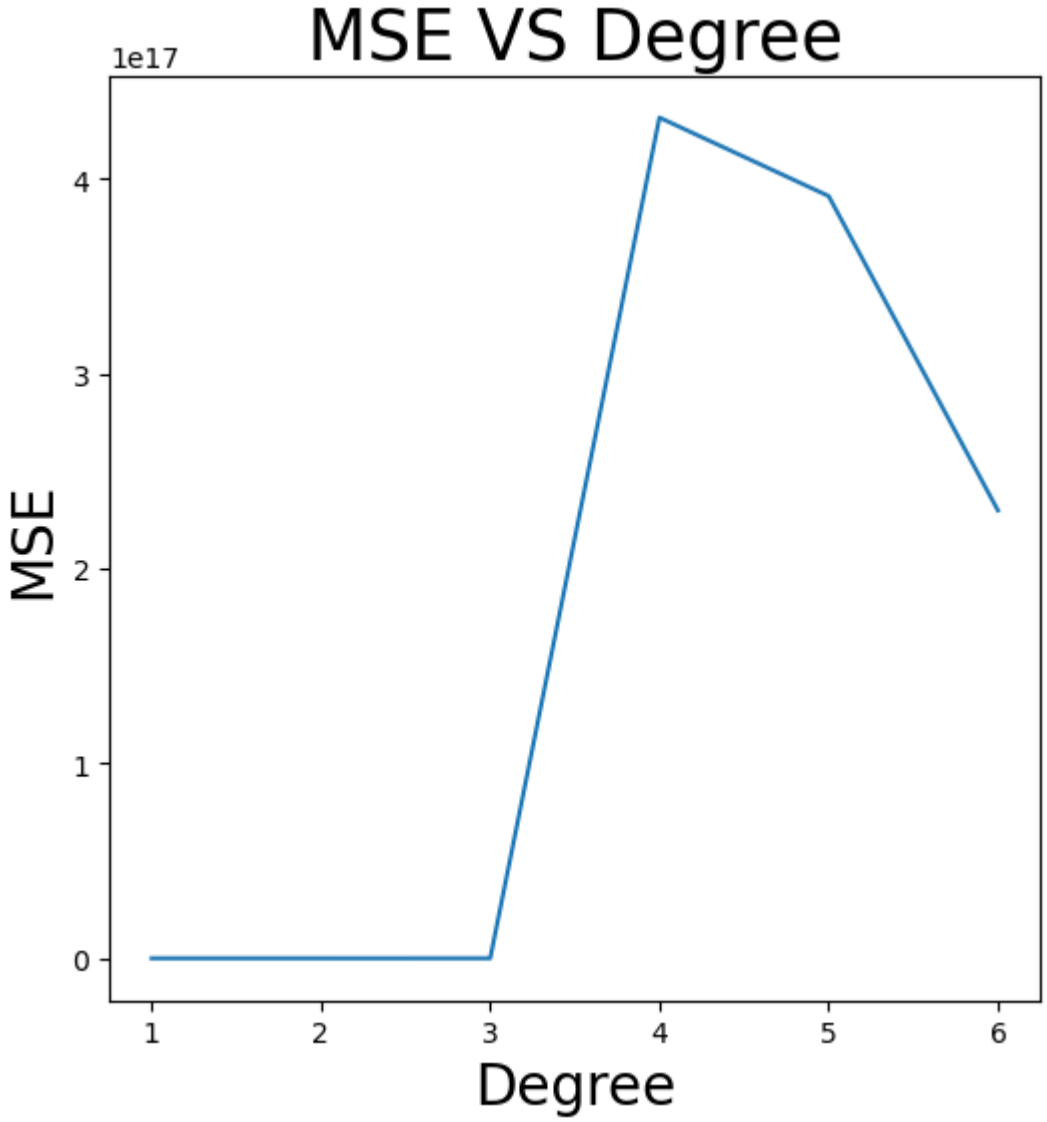
در این تمرین استفاده از هر سه روش نشان داده شده است. با متد duplicate معلوم میشود دیتاست دارای داده تکراری نیست. با استفاده از نمودار box plot وضعیت دادگان پرت نشان داده شده است یک تابع برای تشخیص بازه پرت نبودن داده ها به نام outlier_detect نوشته شده است که بر اساس متد IQR عمل میکند با استفاده از این تابع میتان دیتای پرت را فیلتر کرد.

قسمت d

در مرحله بعد با استفاد از یک پایپ لاین، تابع StandardScaler بر روی داد های عددی برای مقیاس دهی داده ها اعمال شده است و از تابع OneHotEncoder برای تبدییل داده متنی به نمایش برداری استفاده شده است. با استفاده از train_test_split داده ها به دو گروه ازمایشی و اموزشی تقسیم شده اند.

قسمت e

یک مدل رگرسیون چند جمله ای بر روی داده آموزشی اموزش دیده شده است و از متد k fold cross validation برای انتخاب بهترین داده آموزشی با کمترین میزان خطا Mean Square Error استفاده شده است . مقدار درجات ممکن برای مدل چندجمله ای تا مقدار ۷ در نظر گرفته شده است. در یک حلقه مدل های رگرسیون با درجه متفاوت تولیده شده و میزان خطای پیش بینی ان ها در یک لیست ذخیره شده است در ادامه با کمک این لیست نمودار MSE نسبت به درجه چندجمله ای رسم شده است. نتایج ارزیابی نشان می دهد بهترین مقدار برای درجه یک می باشد.



با تحلیل این نمودار مشخص میگردد که به ازای درجه ۴ مدل بدترین عملکرد را داشته است درحالیکه در درجات ۱ تا ۳ کمترین میزان خطا وجود داشته است که بهترین آن مربوط به یک است.

قسمت f

در این قسمت یک مدل چند جمله ای با درجه یک ساخته شده عملکرد آن در زیر نشان داد شده است.

```
Mean Squared Error (MSE): 13028197970.530788
Mean rooted Squared Error (MSE): 114141.131808523
R-squared (R2) score: -0.0022932917728970548
```

قسمت g

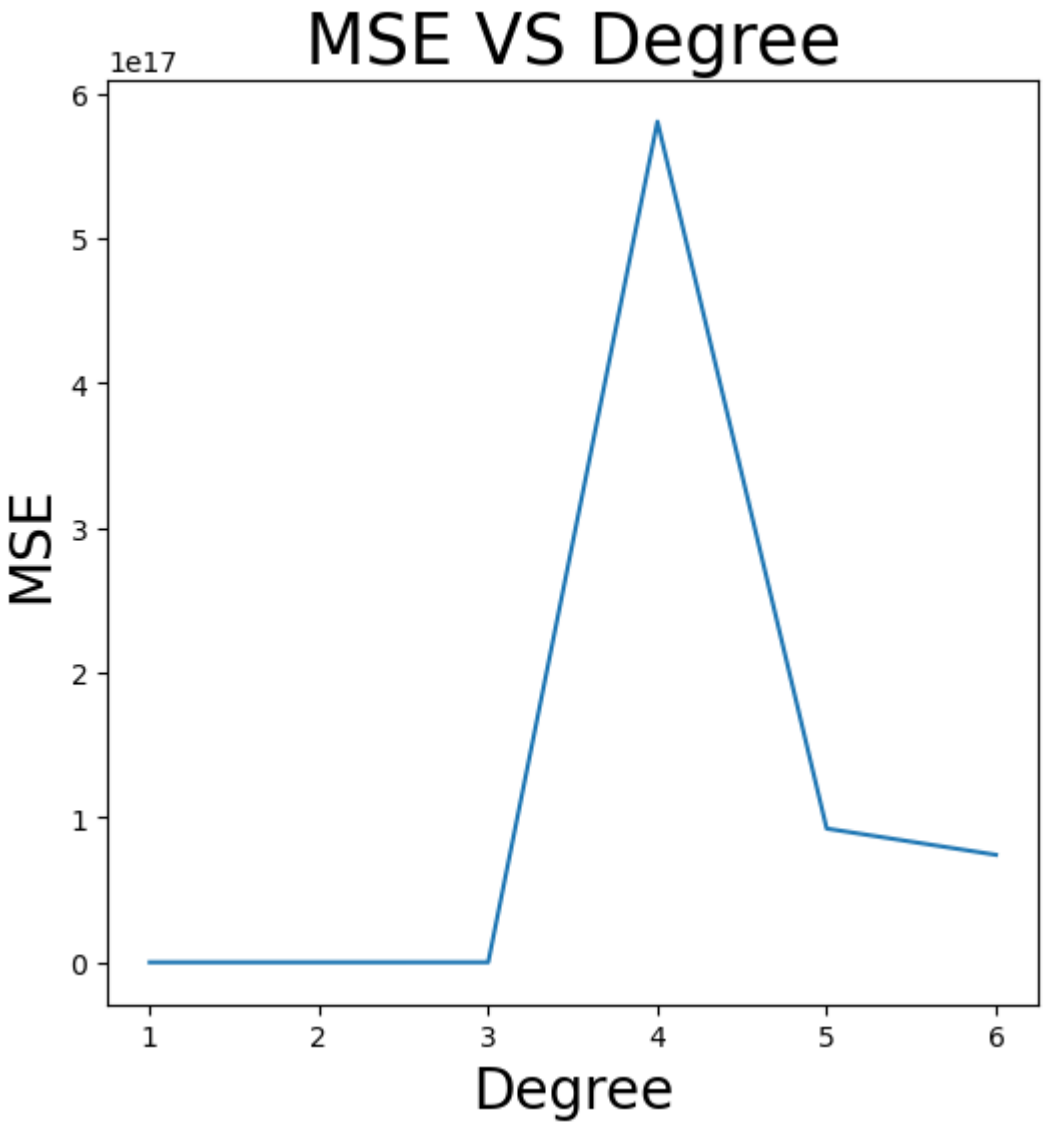
در دیتاست موردنظر تعداد کل اتاق در یک منطقه نشان داده شده است که شاید چندان مفید نباشد بنابراین می توان با تقسیم آن به تعداد کل اتاق ها خانه ها تعداد اتاق هر خانه را تعیین کرد. این عمل را می توان نسبت به ویژگی جمعیت و تعداد اتاق خواب ها هم انجام داد که در زیر نشان داده شده است.

```
[117]: data["rooms_per_household"] = data["total_rooms"]/data["households"]
       data["bedrooms_per_room"] = data["total_bedrooms"]/data["total_rooms"]
       data["population_per_household"]=data["population"]/data["households"]
```

این سه ویژگی جدید به دیتاست اضافه شده اند. حال مراحل **e** تا **f** با وجود ویژگی های جدید دوباره انجام شده اند.

قسمت h

تحلیل نمودار زیر نشان میدهد با اضافه کردن ویژگی های جدید و بهتر بالا، میزان خطای مدل برای درجات ۵ و ۶ کاهش یافته است که در این صورت میتوان از مدل های چندجمله ای با این درجات برای پیش بینی بهتر روابط غیرخطی میان ویژگی ها استفاده کرد.



سوال ۲

قسمت a

با استفاده از متد **read_csv** دیتا بارگذاری شده است. اطلاعات مربوط به ویژگی های آن با استفاده از متدهای **dtype** و **info** نشان داده شده است. ویژگی های **education,**

cigsPerDay, BPMeds, totChol, BMI, heartRate ,glucose دارای مقادیر نامجود هستند.

قسمت b

پیش پردازش

داده تکراری

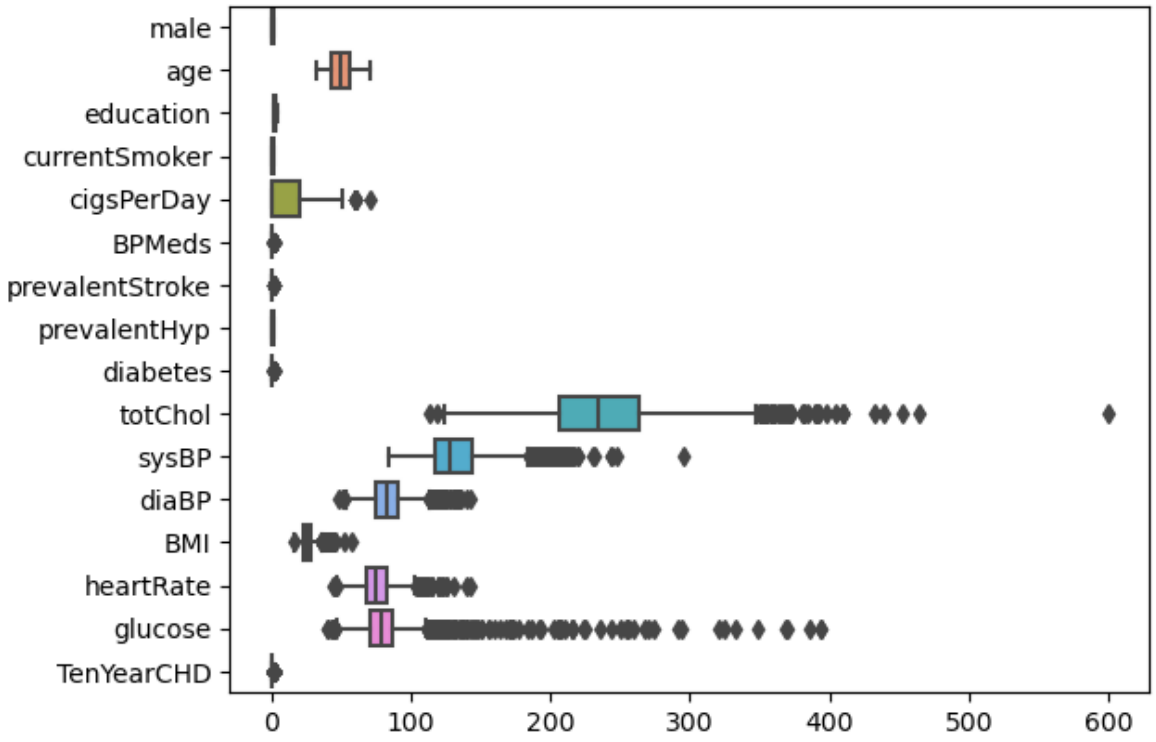
با استفاده از متد **duplicated** معلوم میشود دیتاست دارای داده تکراری نیست.

داده ناموجود

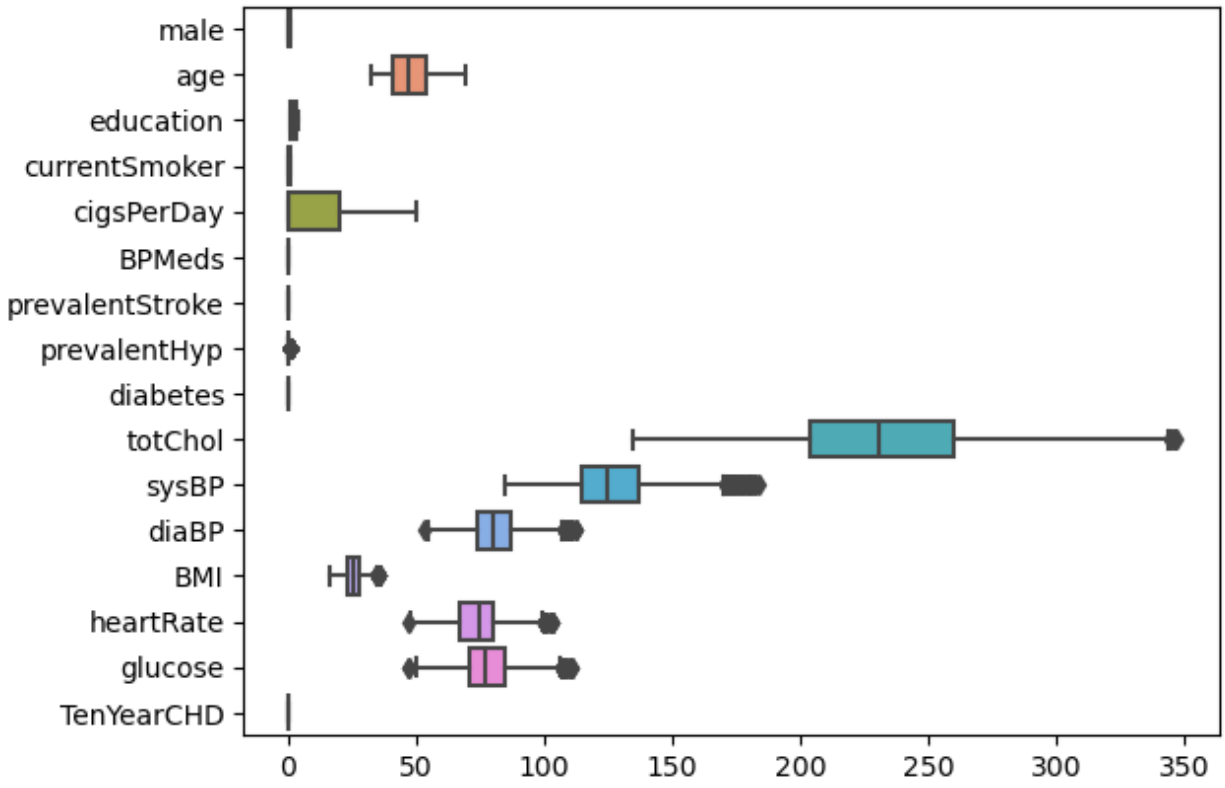
با استفاده از متد **dropna** داده های دارای داده ناموجود حذف شده اند. روش هایی دیگری مثل تخمین آنها با نقادیر آماری مانند میانگین یا میانه یا مقادیر تصادفی نیز می تواند بکارگرفته شود. نمودار های توزیع ویژگی های دیتاست نیز با متد **hist** ترسیم شده است.

داده پرت

با استفاده از نمودار **boxplot** میتوان مشخص نمود هر ویژگی داده پرت دارد یا نه. این نمودار برای همه ویژگی ها رسم شده است. سپس برای حذف داده های پرت یک تابع به نام **detect_outlier** نوشته شده است که بر اساس روش **interquartile Range (IQR)** یک بازه برای تشخیص محدوه پرت بودن برای یک ویژگی تعیین میکند. این تابع بر همه ویژگی های دیتاست اعمال شده است و با کمک آن دیتاست فیلتر شده بدون داده پرت بدست آمده است. تصویر زیر وضعیت دیتاست را از نظر داده ها پرت نشان می دهد.



تصویر زیر بعد از اعمال تابع به دیتاست را نشان می دهد.



مقیاس دهی به داده ها **feature scaling**

با استفاده از متد **MinMaxScaler** مقادیر دادها به بازه صفر تا یک مقیاس شده اند که این کار یادگیری مدل را آسانتر میکند.

قسمت **C**

در این قسمت مرتبط ترین ویژگی ها در تشخیص بیماری با استفاده از کلاس **SelectKBest** انجام شده است نحوه کار این الگوریتم به شرح زیر است:

SelectKBest یک الگوریتم یادگیری ماشین است که برای انتخاب **K** ویژگی برتر از میان مجموعه داده ای بزرگ استفاده می شود. این الگوریتم با محاسبه یک امتیاز برای هر ویژگی و سپس انتخاب **K** ویژگی با بالاترین

امتیاز عمل می کند. امتیاز هر ویژگی می تواند به روش های مختلفی محاسبه شود، اما رایج ترین روش ها عبارتند از:

Mutual Information این روش میزان وابستگی بین یک ویژگی و کلاس هدف را اندازه گیری می کند.

Gainاین روش میزان کاهش انتروپی را که یک ویژگی به هنگام اضافه شدن به مدل ایجاد می کند، اندازه گیری می کند.

Chi-Squaredاین روش میزان وابستگی بین یک ویژگی و کلاس هدف را با استفاده از آزمون کای دو اندازه گیری می کند. پس از محاسبه امتیازات، **K**ویژگی برتر انتخاب می شوند و بقیه حذف می شوند. این می تواند به طور قابل توجهی ابعاد مجموعه داده را کاهش دهد و به بهبود عملکرد مدل یادگیری ماشین کمک کند.

SelectKBest اغلب به عنوان بخشی از فرآیند پیش پردازش داده در یادگیری ماشین استفاده می شود. این می تواند برای انتخاب ویژگی هایی که برای یک کار خاص مرتبط تر هستند، یا برای حذف ویژگی هایی که ممکن است باعث نویز یا بیش برازش شوند، استفاده شود.

مرتبط ترین ویژگی های انتخاب شده با کمک الگوریتم فوق موارد زیر هستند:

cigsPerDay

BPMeds

prevalentStroke

prevalentHyp

diabetes

totChol

sysBP

diaBP

BMI

heartrate

قسمت d

با استفاده از متد **train_test_split** دیتاست به مجموعه آموزشی و آزمایشی با اندازه ۶۰ درصد تقسیم شده است.

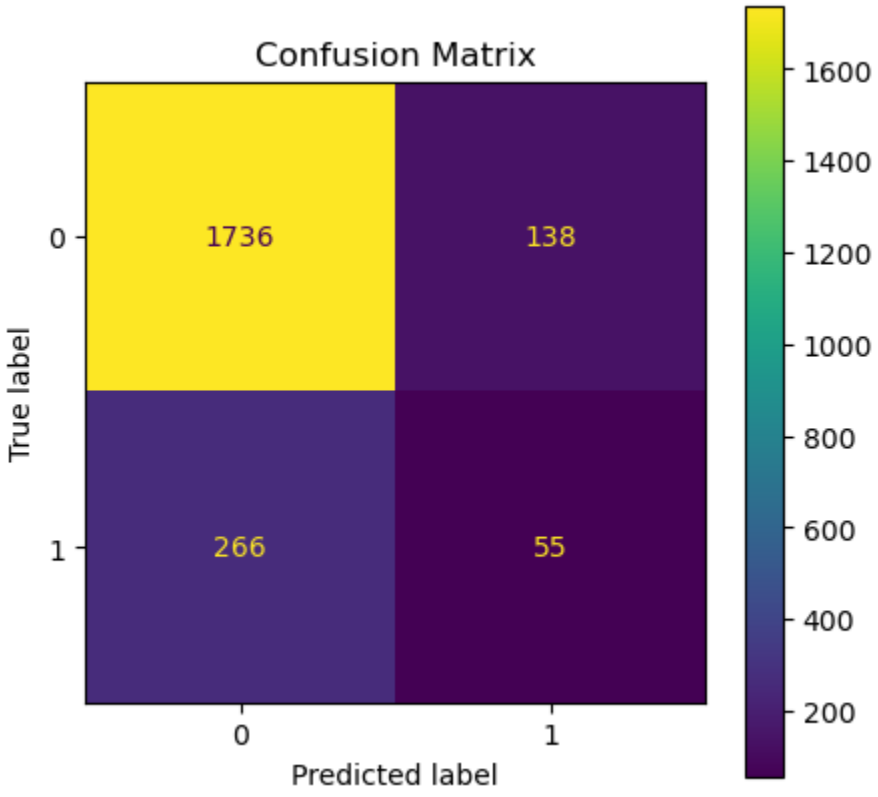
قسمت e

یک مدل **KNN** با اندازه پارامتر **k = 3** روی داده اموزشی **train** شده است.

قسمت f

نحوه عملکرد مدل با استفاده از **confusion_matrix** و **accuracy** و **R2_score** ارزیابی شده است.

تفسیر ماتریس درهم ریختگی به صورت زیر است:



کلاس صفر: سالم بودن

کلاس یک: ابتلا به بیماری

۱۷۳۶ داده که عضو کلاس صفر بوده اند به درستی عضو کلاس صفر تشخیص داده شده اند

۱۳۸ داده که عضو کلاس صفر بوده اند، به اشتباه عضو کلاس یک تشخیص داده شده اند

۲۶۶ داده که عضو کلاس یک بوده اند به اشتباه عضو کلاس صفر تشخیص داده شده اند

۵۵ داده که عضو کلاس یک بوده اند به درستی عضو کلاس یک تشخیص داده شده اند

Accuracy: میزان صحت مدل ۸۲ درصد بوده است.

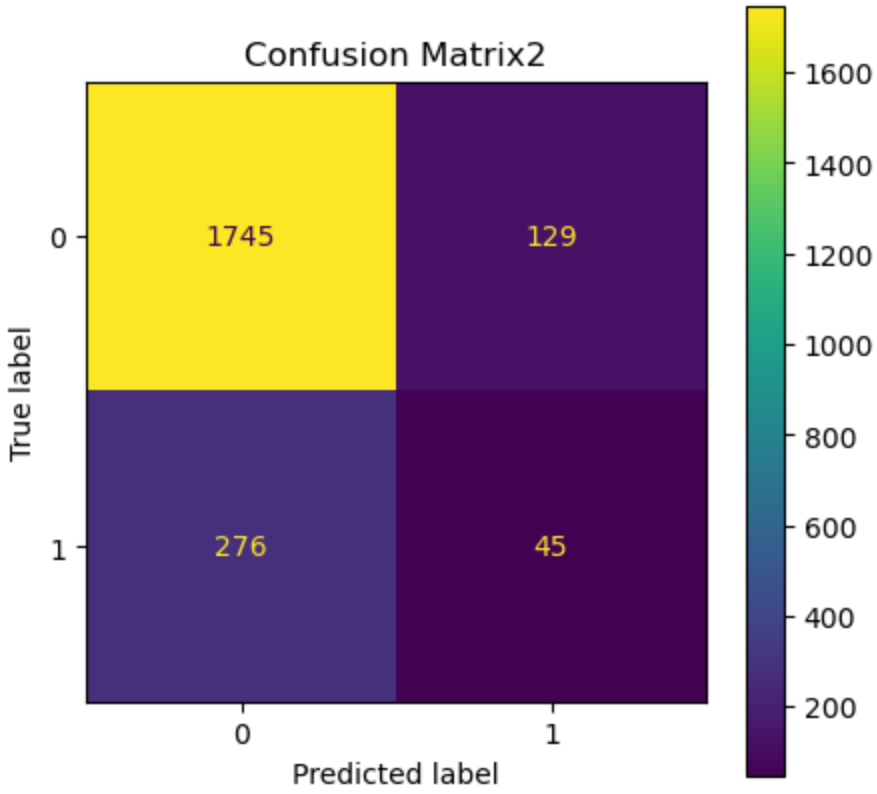
R2_score: میزان این شاخص برای مدل برابر منفی ۰.۴۷ بوده است.

قسمت g

در این قسمت یک مدل جدید **KNN** با شاخص فاصله **manhatan** ایجاد شده است.ارزیابی مدل جدید نشان می دهد از نظر شاخص های ذکر شده عملکرد مدل تغییر محسوسی نداشته است. تصویر ماتریس درهم ریختگی بعد از اعمال شاخص متفاوت در زیر آمده است.

Accuracy: میزان صحت مدل جدید ۸۱ درصد بوده است.

R2_score: میزان این شاخص برای مدل جدید برابر منفی ۰.۴۷ بوده است.


Confusion Matrix2
1745
129
276
45
0
1
True label
0
1
Predicted label
1600
1400
1200
1000
800
600
400
200

سوال ۳

قسمت a

بعد از فراخوانی کتابخانه های موردنیاز، **(pandas, hazm, ...)** دیتاست با استفاده از متد **read_csv()** در یک دیتافریم به نام **data** ذخیره می‌شود.

قسمت b

در این مرحله پیش‌پردازش داده‌ها شروع می‌شود. با استفاده از متد های مختلف کتابخانه **hazm** می توان نرمالسازی متن را انجام داد. برخی از نرمالسازی ها در ادامه امده است.

remove_diacritics

اِعراب را از متن حذف می‌کند.

remove_specials_chars

برخی از کاراکترها و نشانه‌های خاص را که کاربردی در پردازش متن ندارند حذف می‌کند.

decrease_repeated_chars

تکرارهای زائد حروف را در کلماتی مثل سلامممممم حذف می‌کند و در مواردی که نمی‌تواند تشخیص دهد دست کم به دو تکرار کاهش می‌دهد.

persian_number

اعداد لاتین و علامت ٪ را با معادل فارسی آن جایگزین می‌کند.

unicodes_replacement

برخی از کاراکترهای خاص یونیکد را با معادلِ نرمال آن جایگزین می‌کند. غالباً این کار فقط در مواردی صورت می‌گیرد که یک کلمه در قالب یک کاراکتر یونیکد تعریف شده است.

حذف کلمات توقف نیز با استفاده از لیستی که این کتابخانه اماده دارد انجام میشود. یک تابع برای حذف کلمات توقف، ریشه یابی کلمات[1] و چک کردن اینکه همه کاراکتر ها الفبا یا عدد باشند، در این تابع انجام میشود. این تابع به دیتافریم اعمال شده و متن نرمال شده به دیتافریم اضافه میشود.

قسمت c

در این قسمت دیتاست به دادهای آموزشی و آزمایشی تقسیم میشود. برای آموزش یک مدل دسته بند روی داده متنی، ورودی مدل باید به قالب عددی و به صورت برداری باشد تا مدل بتواند روی آن پردازش انجام دهد. در این قسمت از روش bag_of_words برای نمایش کلمات متن به صورت بردار استفاده شده است.

قسمت d

آموزش یک مدل

در این قسمت در یک حلقه for ، یک مدل KNN با استفاده از داده های آموزشی، با پارامتر های مختلف K (نشان دهنده تعداد همسایگی داده پیش بینی شونده است) از ۱ تا ۲۰، آموزش دیده شده است و با استفاده از داده آزمایشی عمل دسته بندی روی آن انجام شده است. در هر بار، صحت یا accuracy مدل در یک لیست ذخیره شده است. بهترین k برای دسته بندی یک با دقت حدود ۸۹ درصد می باشد.

---

[1] lemmatization

سوال ۴

قسمت های a و b

در مرحله اول کتابخانه های لازم برای کار با این دیتاست فراخوانی می‌شوند. در مرحله بعدی، یک تابع به نام **load_images**برای استخراج و بارگذاری تصاویر نوشته شده است این تابع با استفاده از متدهای کتابخانه **openCv** مانند **cv2.imread** تصاویر را از فایل مقصد خوانده و عمل تغییر سایز را با استفاده از متد **resize** کتابخانه **openCv** انجام می دهد. بعد از آن با **label** تصاویر را با استفاده از اسم تصاویر مشخص کرده و تصاویر را در یک لیست به نام **images** و **label** ها را در یک لیست به نا **labels** اضفه میکند.

قسمت c

در این قسمت دیتاست با استفاده از متد **train_test_split** به دو مجموعه اموزشی و تست با نسبت های ۸۰ به ۲۰ درصد تقسیم می‌شود.

قسمت d

در این مرحله قرار است یک مدل **KNN** برای دسته بند برای دسته بندی تصاویر ساخته شود. ابتدا مدل **KNeighborsClassifier** از کتابخانه **sklearn** فراخوانی میشود با پارامتر **k = 5** یک مدل دسته بند ساخته شده و داده اموزشی برای اموزش این مدل استفاده میشود. مدل ساخته شده برای پیش بینی داده تست استفاده شده و کارایی آن از نظر شاخص های **accuracy**، **precision**، **recall** و **F1-score** ارزیابی شده است. در این مرحله از تابع **classification_report** استفاده شده است.

```
Classification Report:
              precision    recall  f1-score   support

         Cat       0.55      0.72      0.62      2515
         Dog       0.59      0.40      0.47      2485

    accuracy                           0.56      5000
   macro avg       0.57      0.56      0.55      5000
weighted avg       0.57      0.56      0.55      5000
```

شاخص دقت **precision** به این سوال پاسخ می دهد که چه نسبتی از مثبت پیش‌بینی شده واقعا مثبت بوده است در این مورد، این مدل ۵۵ درصد گربه ها و ۵۹ درصد سگ هایی که پیش بینی کرده با برچسب واقعی آن دیتا مطابق بوده است.

شاخص بازخوانی یا **recall** به این سوال پاسخ میدهد "چه نسبتی از مثبت ها به درستی به عنوان مثبت دسته‌بندی شده‌اند ؟" این مدل در مورد دسته گربه ۷۲ درصد و در درسته سگ ۴۰ درصد از مثبت ها را به عنوان مثبت دسته بندی کرده است.

شاخص **F1** عددی بین ۰ تا ۱ است و معیار **F1** که در واقع ترکیب متعادلی بین معیارهای دقت و صحت است، می‌تواند در مواردی که هزینه‌ی **False Positive** و **False Negative** متفاوت است به کار رود.

Accuracy یا صحت اساسی‌ترین معیار اندازه‌گیری کیفیت یک دسته بند است. صحت یعنی نسبت نتایج واقعی به کل موارد بررسی شده. این مدل دارای صحت ۵۶ درصد است یعنی ۵۶ درصد کل پیش بینی ها درست بوده است که برای یک مدل دسته بند مقدار قابل قبولی نیست

قسمت e

در این قسمت قرار است از **Cross-Validation** برای ارزیابی عملکرد مدل **KNN** استفاده شود.

Cross-Validation یا اعتبار سنجی متقابل یک روش برای ارزیابی یک مدل در ماشین لرنینگ و همچنین آزمایش نحوه عملکرد آن است. از **CV** اصولا در فعالیت های کاربردی یادگیری ماشین استفاده می شود. این کار درمقایسه و انتخاب یک مدل مناسب برای مسئله مدل‌سازی پیش‌بینی‌کننده خاص کمک می‌کند. در این موزد از از فاصله اقلیدسی که حالت پیش فرض است برای سنجش فاصله میان داده‌ها استفاده میشود.

[33]:
```
print("Average Accuracy:", avg_accuracy, "±", std_accuracy)
print("Average Precision:", avg_precision, "±", std_precision)
print("Average Recall:", avg_recall, "±", std_recall)
print("Average F1-score:", avg_f1_score, "±", std_f1_score)
```

```
Average Accuracy: 0.55832 ± 0.009594665184361565
Average Precision: 0.5670058647050586 ± 0.010783270604225176
Average Recall: 0.55832 ± 0.00959466518436156
Average F1-score: 0.5434821376191263 ± 0.010408495244943944
```

نتایج ارزیابی مدل با روش فوق در عکس بالا نشان داده شده است که نشان میدهد این مدل از نظر این شاخص ها عملکرد خوبی ندارد به طوریکه میانگین شاخص صحت آن در داده های اموزشی و ازمایشی مختلف، برابر ۵۵ درصد و انحراف معیار آن تقریبا ناچیز بوده است. برای شاخص دقت عدد ۵۶ درصد بدست امده است. برای شاخص بازخوانی و امتیاز **f1** هم نتیجه به ترتیب ۵۵ درصد و ۵۴ درصد بوده است.

قسمت f

با تحلیل نتایج بدست آمده به نظر می رسد استفاده از مدل های دیگر مانند شبکه های عصبی به دقت و صحت بالاتری می توان دست یافت و نتیجه آن که مدل **KNN** برای دسته بندی تصاویر خوب عمل نمیکند زیرا با توجه با آنکه تنها دو دسته (سگ و گربه)برای دسته بندی وجود دارد حتی با دسته بندی شانسی نیز دقت ۵۰ درصد قابل دستیابی است.