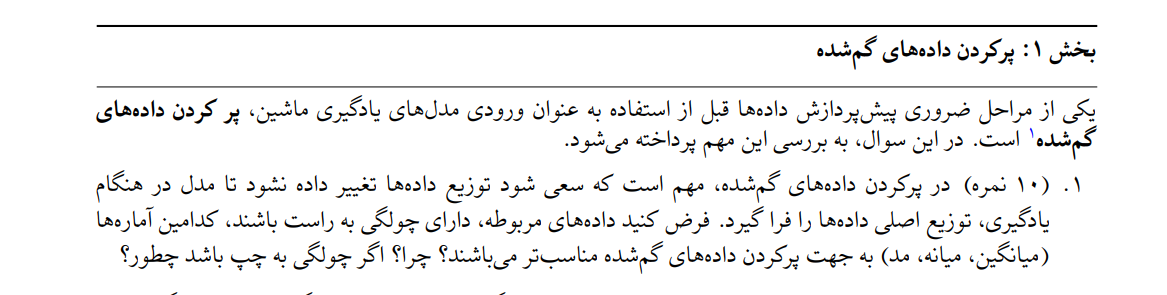
 **گزارش تمرین دوم مبانی داده‌کاوی و کاربردهای آن (۲۱۰۱۹)**

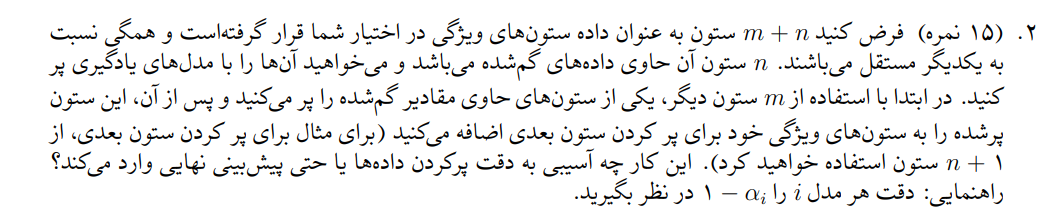
استاد: دکتر مانا مس‌کار  
دستیار آموزشی: محمدسبحان کسائی  
اعضای گروه: صبا عبدی (۴۰۱۱۰۴۲۷6)، آوا صدیقی (۴۰۱۱۱۰۱۵۹)



برای پر کردن داده‌ها[[1]](#footnote-1)، همانگونه که در صورت سوال ذکر شده است، باید از شاخصی استفاده کرد که توزیع و ویژگی‌های اساسی داده‌ها را تغییر ندهد.

می‌دانیم در داده‌هایی که چولگی به سمت راست دارند، یعنی همان چولگی مثبت را دارند، ترتیب مقادیر میانگین، میانه و مد بدین شکل است: . می‌دانیم که میانگین تا حدودی تحت تأثیر داده‌های پرت[[2]](#footnote-2) است. بنابراین اگر از میانگین برای پر کردن داده‌ها استفاده کنیم، میانگین کل داده‌ها از مقدار واقعی خود دور شده، یعنی بیشتر می‌شود، و چولگی داده‌ها را تشدید می‌کند. پس میانگین انتخاب مناسبی نمی‌باشد. اگر از مد استفاده کنیم، تجمیع داده‌ها در سمت چپ تشدید شده و توزیع را دچار مشکل می‌کند. علاوه بر این، ممکن است متناسب با نوع داده[[3]](#footnote-3) موجود نتوانیم از مد استفاده کنیم. بنابراین از میانه استفاده می‌کنیم. میانه تنها به ترتیب داده‌ها وابسته است و مقدار داده‌ها و فاصله آن‌ها از یکدیگر در آن تغییری ایجاد نمی‌کند؛ بنابراین اثر داده‌های پرت را تا حدودی از بین می‌برد و مرکزیت داده را حفظ می‌کند. لذا استفاده از میانه کمترین آسیب را به داده وارد می‌کند.

می‌دانیم در داده‌هایی که چولگی به سمت راست دارند، یعنی همان چولگی مثبت را دارند، ترتیب مقادیر میانگین، میانه و مد بدین شکل است: . مشابه حالت قبل، می‌دانیم که میانگین تا حدودی تحت تأثیر داده‌های پرت[[4]](#footnote-4) است. بنابراین اگر از میانگین برای پر کردن داده‌ها استفاده کنیم، میانگین کل داده‌ها از مقدار واقعی خود دور شده، یعنی کمتر می‌شود، و چولگی داده‌ها را تشدید می‌کند. پس میانگین انتخاب مناسبی نمی‌باشد. اگر از مد استفاده کنیم، تجمیع داده‌ها در سمت راست تشدید شده و توزیع را دچار مشکل می‌کند. علاوه بر این، ممکن است متناسب با نوع داده[[5]](#footnote-5) موجود نتوانیم از مد استفاده کنیم. بنابراین از میانه استفاده می‌کنیم. میانه تنها به ترتیب داده‌ها وابسته است و مقدار داده‌ها و فاصله آن‌ها از یکدیگر در آن تغییری ایجاد نمی‌کند؛ بنابراین اثر داده‌های پرت را تا حدودی از بین می‌برد و مرکزیت داده را حفظ می‌کند. لذا استفاده از میانه کمترین آسیب را به داده وارد می‌کند.



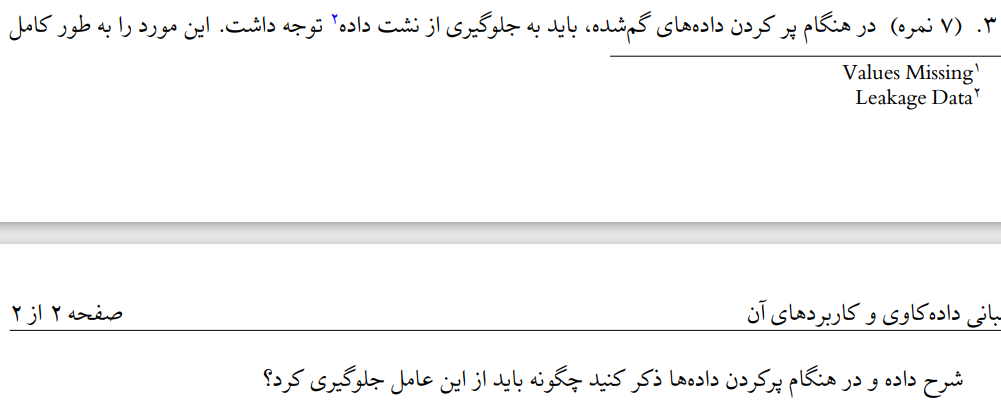
در هر دور، با استفاده از مدل یادگیری در دست، داده‌های یک ستون دارای داده‌های گم‌ شده[[6]](#footnote-6) را پر می‌کنیم. این عمل را باید n بار تکرار کنیم. این روش مشکلات متعددی را، مانند کاهش دقت و افزایش خطا، به وجود می‌آورد.

می‌دانیم مدل یادگیری هر مرحله دقتی محدود و کمتر از یک دارد. دقت هر مدل در این سوال، متناسب با مرحله‌ای که در آن است، برابر با است. بنابراین در هر مرحله، مقدار پیش‌بینی شده مساوی با برابر داده حقیقی است. با استفاده از داده‌های تخمینی هر مرحله در مراحل بعدی، دقت مدل‌ها و پیش‌بینی‌ها رفته‌رفته کاهش می‌یابند. به عبارتی، با گذشت چندین مرحله، به نوعی داده‌های واقعی از دست می‌روند و تخمین‌ها دیگر دقیق و بر مبنای داده‌های درست اولیه نخواهند بود و در نهایت دقت نهایی مدل برابر با می‌شود که بسیار پایین است.

در ادامه، به دلیل آنکه داده‌های تخمین زده شده در هر مرحله به مدل یادگیری مرحله بعد اضافه می‌شوند، ممکن است بین داده‌ها، به اشتباه، همبستگی به وجود بیاید. یعنی در مدل‌های یادگیری، شرایطی مبنی بر همبستگی داده‌ها اضافه شود که دقت مدل و داده‌های تخمینی را بیش از پیش بکاهد. علاوه بر این، شرط استقلال ویژگی‌ها نیز دیگر برقرار نخواهد بود.

علاوه بر این، مدل نهایی، به دلیل آنکه با تمام داده‌ها آموزش داده می‌شود، ممکن است به مشکل بیش‌برازش[[7]](#footnote-7) دچار شود و نتواند عملکرد مناسبی روی داده‌های جدید داشته باشد. بیش‌برازش به این علت رخ می‌دهد که ممکن است داده‌های هر مرحله به داده‌های ویژگی‌های قبل بسیار وابسته شده و استقلال و تأثیرگذاری خود را تا حدودی از دست بدهند.

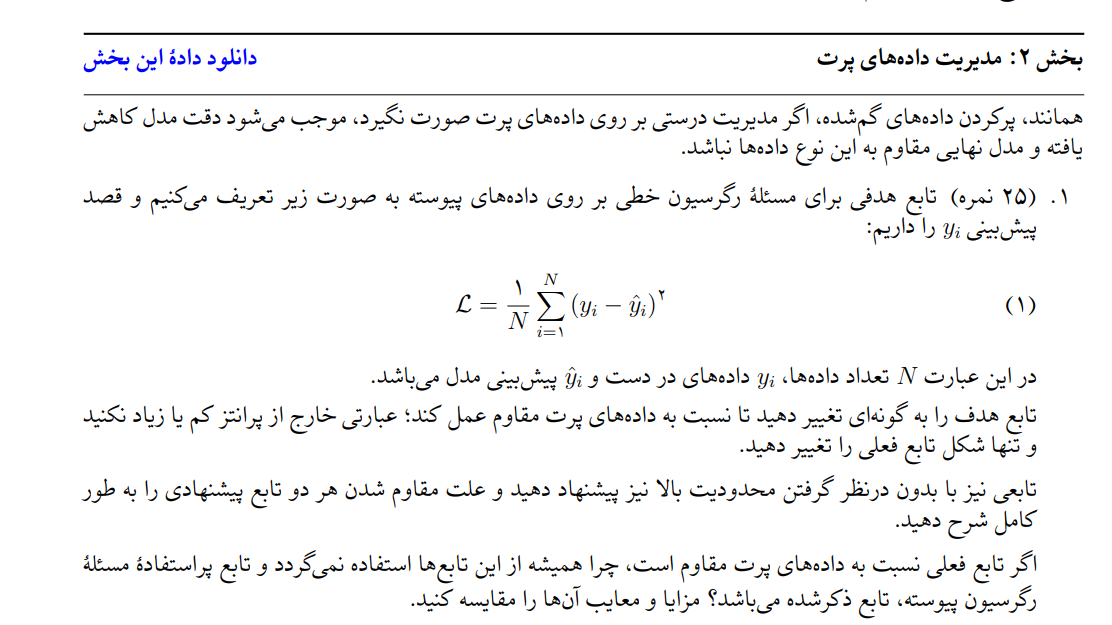




هنگامی که در تلاش برای توسعه یک مدل هستیم، برای آن دو بخش داده در نظر می‌گیریم؛ داده‌های آموزش[[8]](#footnote-8) و داده‌های تست. مدل ما با استفاده از داده‌های آموزش یاد می‌گیرد که چگونه تخمین بزند و ما با استفاده از داده‌های تست دقت آن را اندازه‌گیری می‌کنیم. حال اگر داده‌هایی که نباید در بخش آموزش باشند، مانند داده‌های تست، به نوعی در آن وجود داشته باشند، نشت داده رخ داده است. این یعنی مدل ما در ظاهر عملکردی بی‌نظیر دارد ولی در مواجهه با داده‌های واقعی، عملکرد خوبی از خود نشان نخواهد داد؛ یعنی تعمیم‌پذیری و دقت تخمین و ارزیابی مدل کاهش می‌یابد.

نشت داده می‌تواند از طرق مختلفی رخ دهد. اگر برای پر کردن مقادیر گم شده از آماره‌هایی مثل میانگین کل داده استفاده کنیم، یعنی برای محاسبه میانگین یا هر آماره‌ای کل داده را مبنا قرار دهیم، اطلاعات داده‌های تست، به دلیل یکسان بودن مقدار پر شده، وارد داده‌های آموزش می‌شوند. برای جلوگیری از این اتفاق باید ابتدا و پیش از انجام هر اقدامی، داده‌ها را به دو بخش آموزش و تست تقسیم کنیم. سپس از آماره‌های به دست آمده از هر بخش برای پر کردن مقادیر گم شده آن استفاده شود.

علاوه بر این می‌توان از روش‌های cross-validation و multiple imputation استفاده کرد. در cross-validation چندین Fold داریم. در ابتدا داده‌های مربوط به هر Fold را مشخص و پس از آن داده‌های بخش‌های آموزش و تست را تعیین می‌کنیم. سپس هر Fold را به طور جداگانه پردازش کرده و مقادیر گم‌ شده آن را پر می‌کنیم. در multiple imputation نیز ابتدا داده‌های آموزش و تست مشخص می‌شوند ولی به جای تخمین یک مقدار مشخص، چندین مقدار با احتمالاتی مشخص تخمین زده می‌شود و سپس با استفاده از مقداری که ترکیبی از تخمین‌ها است، داده‌های گم شده پر می‌شوند. لازم به ذکر است که ایده اصلی جلوگیری از نشت داده، همان مشخص کردن داده‌های آموزش و تست پیش از انجام هر تغییری بر روی داده‌ها است.



تابع هدف اولیه همان [[9]](#footnote-9)MSE است. در این تابع اختلاف مقادیر واقعی و مقادیر پیش‌بینی شده به توان دو می‌رسد. بنابراین در مواجهه با داده‌های پرت، مقدار این تابع بسیار افزایش پیدا می‌کند و کل تابع هدف، تحت تأثیر قرار می‌گیرد. علاوه بر این، یادگیری مدل نیز دچار مشکل می‌شود.

در ابتدا باید تابع هدف را به گونه‌ای تغییر دهیم که تنها شکل آن عوض شود و عبارتی اضافه یا کم نشود. در این راستا بهتر است از تابع MAE[[10]](#footnote-10) استفاده کنیم: . در این تابع، مقدار واقعی اختلاف در نظر گرفته می‌شود و به دلیل آنکه این اختلاف به توان دو نمی‌رسد، داده‌های پرت نمی‌توانند بر تابع هدف فعلی، برعکس تابع هدف قبلی، تأثیر زیادی بگذارند. بنابراین مدل نسب به داده‌های پرت مقاوم‌تر می‌شود.

در ادامه، به دلیل اینکه می‌توانیم شکل تابع را نیز تغییر دهیم، بهتر است از تابع Huber Loss یا Smooth MAE استفاده کنیم. تابع ذکر شده به این صورت است:

*در این تابع، اگر مقدار اختلاف برابر یا کوچکتر از مقداری مانند باشد، تابع به* مانند *تابع هدف اصلی، تابع هدف در صورت سوال، محاسبه می‌شود. ولی اگر اختلاف از مقدار بیشتر باشد، تابع هدف به حالت نوشته شده محاسبه می‌گردد. بنابراین در مواجهه با داده‌های پرت، مقدار آن‌ها با استفاده از رابطه موجود، کاهش یافته و تابع هدف به مقدار کمتری تحت تأثیر قرار می‌گیرد. چرا که اختلاف موردنظر در ضرب شده و سپس مقدار مشخصی از آن کم می‌شود. بنابراین مقدار کمتری در مقایسه با اختلاف اولیه وارد تابع هدف می‌شود و این یعنی تابع هدف در مقابل داده‌های پرت مقاوم شده است.*

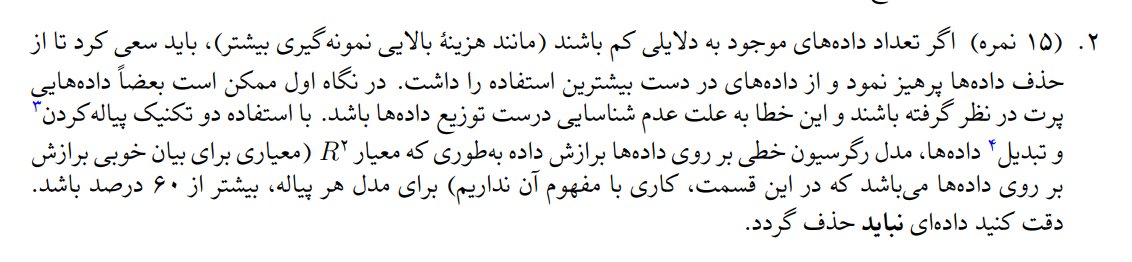
*تابع* MSE *به دلیل ویژگی‌هایی که دارد، بسیار پراستفاده است. از این وِیژگی‌ها می‌توان به سادگی ریاضیاتی و مشتق‌پذیری اشاره کرد. این ویژگی موجب می‌شود که بتوان روش‌های مبتنی بر گرادیان را آسان‌تر پیاده‌سازی کرد و مشائل رگرسیون خطی را حل نمود. علاوه بر این، تابع* MSE *ساده و تفسیرپذیر است. به عنوان مثال با کاهش مقدار تابع می‌توان دریافت که واریانس خطا کم شده و برآوردها دقیق‌تر هستند. در ادامه این تابع در مواجهه با نویزهایی که از تابع نوزیع نرمال پیروی می‌کنند بهترین عملکرد را دارد. در نهایت، بسیاری از مدل‌های کلاسیک و کتابخانه‌های فعلی بر پایه این تابع طراحی شده توسعه داده شده‌اند.*

*حال مزایا و معایب هر یک از روش‌ها را به ترتیب بیان می‌کنیم.*

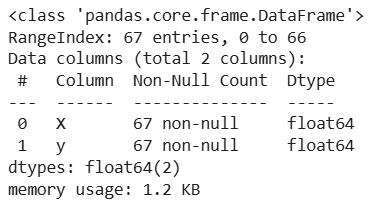
*در رابطه با* MSE*، اگر داده‌ها از توزیع نرمال پیروی کنند، این تابع بهترین برآوردکننده نااریب است و نتایج آن از دیگر تابع‌ها، دقیق‌تر است. علاوه بر این چون مشتق‌پذیر است، حل مسائل بهینه‌سازی با استفاده از آن راحت‌تر است و روش‌هایی چون حداقل مربعات آسان‌تر پیاده‌سازی می‌شوند. اما باید در نظر داشت که این تابع از داده‌‎های پرت تأثیر زیادی می‌پذیر و همین مورد ممکن است موجب تمرکز مدل بر روی داده‌های پرت شود.*

*در رابطه با* MAE*، این تابع عملکر بهتری در مواجهه با داده‌های پرت دارد و زمانی که داده‌ها نویز زیادی داشته باشند، نتایج بهتری نسبت به* MSE *ارائه می‌دهد. اما این تابع در نقطه صفر مشتق نداشته و می‌تواند روش‌های گرادیان محور را دجار مشکل کند. علاوه بر این، همگرایی به مقدار بهینه در این تابع کندتر اتفاق می‌افتد.*

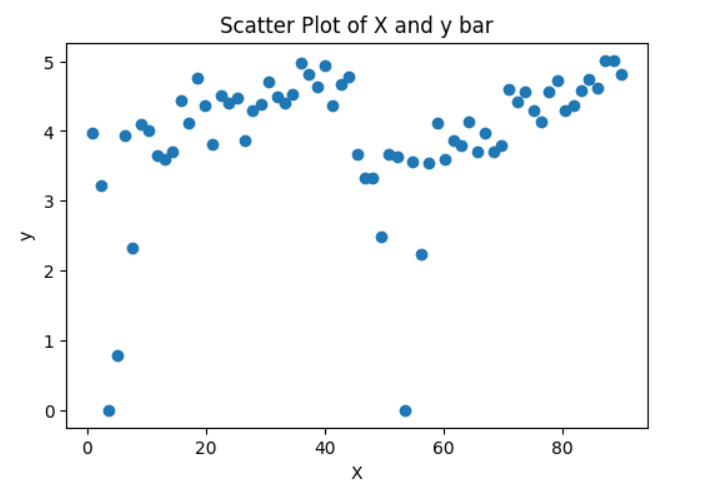
*در نهایت، تابع* Huber Loss*، ترکیبی از دو تابع قبلی است؛ یعنی در مواجهه با اختلاف کوچک مانند* MSE *رفتار می‌کند و در مواجهه یا اختلاف زیاد مانند* MAE *رفتار می‌کند. این یعنی سرعت همگرایی مطلوبی دارد و در مواجهه با داده‌های پرت، کمتر تأثیر می‌پذیرد. اما در این تابع، محاسبات موجود پیچیده‌تر هستند و نیاز است تا مقدار مناسبی برای تعیین شود.*

**

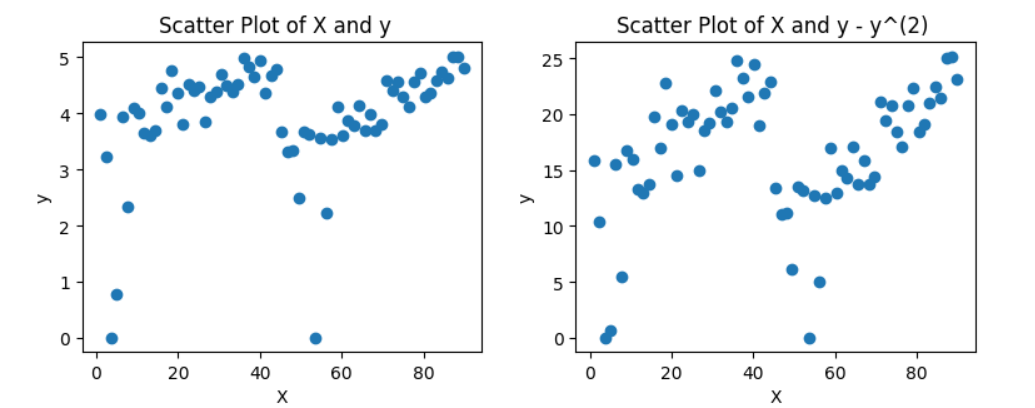
*پس از فراخوانی کتابخانه‌های مورد استفاده، داده مورددنظر را می‌خوانیم و اطلاعات آن را دریافت می‌کنیم. نوع داده* float *است و هیچ داده گم‌شده‌ای نداریم.*

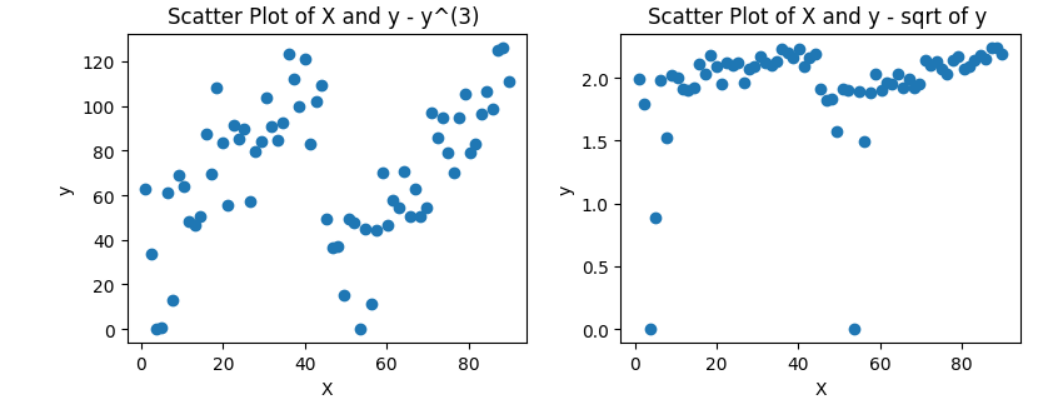
**

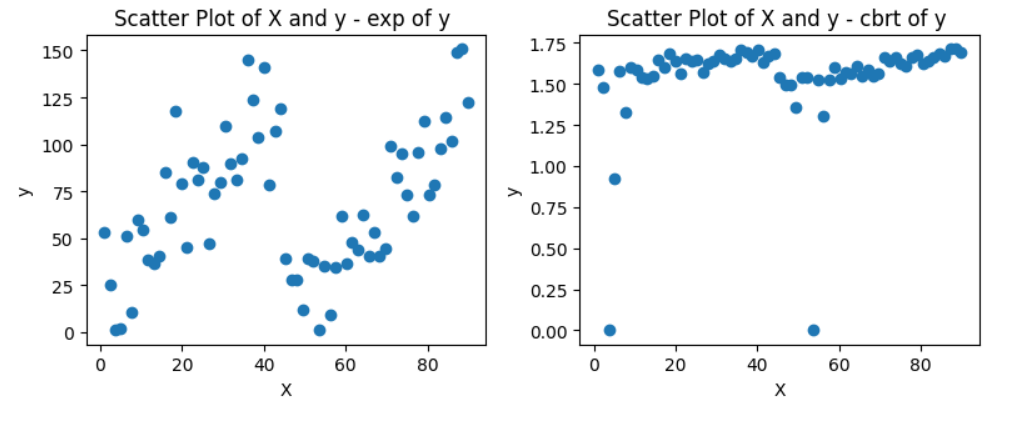
*پس از جدا کردن داده‌های* X *و ،* scatter plot *این دو سری را رسم می‌کنیم تا دیدی کلی از داده‌ها به دست آوریم و داده‌های پرت را تا حدودی بشناسیم. نمودار به این شکل است:*



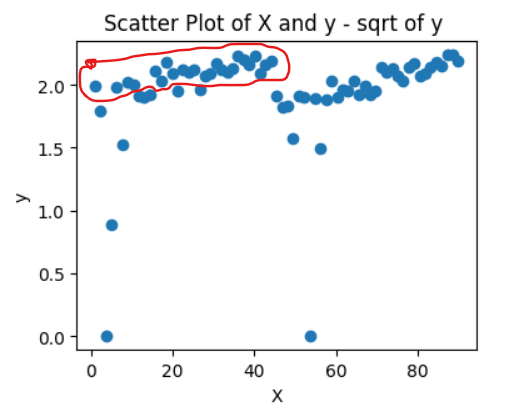
*حال برای مشخص کردن پیاله‌ها، چندین تبدیل[[11]](#footnote-11) معروف را روی داده‌ها، مقدار* y*، پیاده سازی می‌کنیم. این تبدیل‌ها به ترتیب توان دو، توان سه، ریشه دوم، نمایی* exp *و ریشه سوم هستند. در عکس‌ زیر، این نمودارها کنار یکدیگر به نمایش درآمده‌اند.*

**

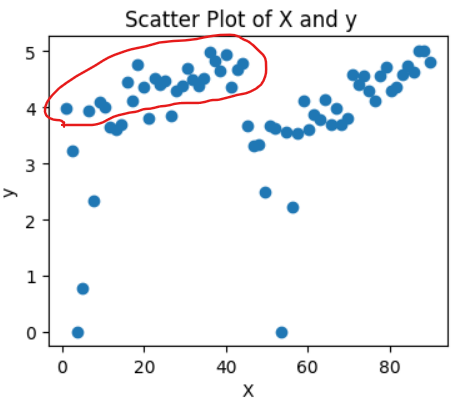
**



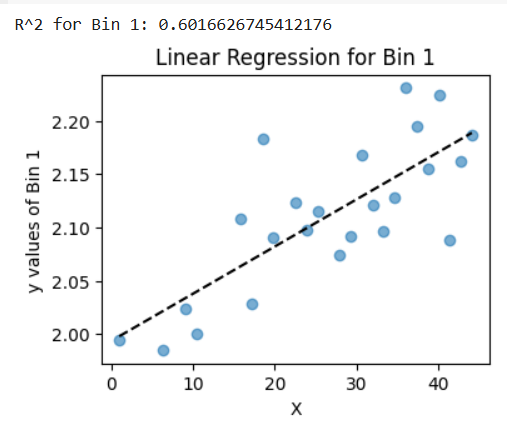
با آزمون و خطای بسیار، در نهایت به 4 پیاله رسیدیم. پیاله اول[[12]](#footnote-12) با استفاده از ریشه دوم به دست آمده است و به شرح زیر است:



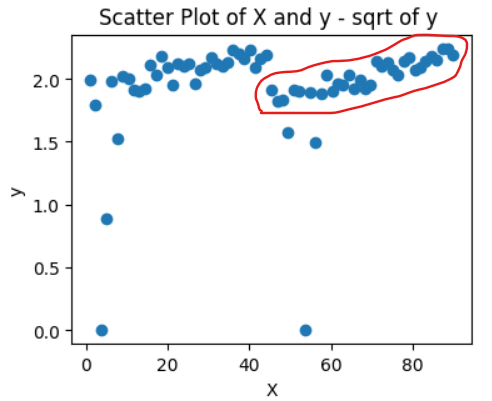
این بخش از داده در نمودار اصلی به شرح زیر است:



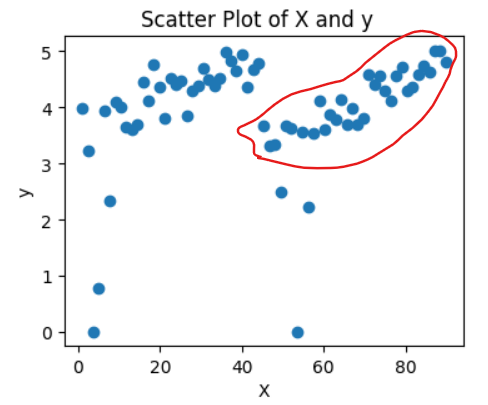
دقت مدل رگرسیون خطی این پیاله برابر با 60.16 درصد می‌باشد.



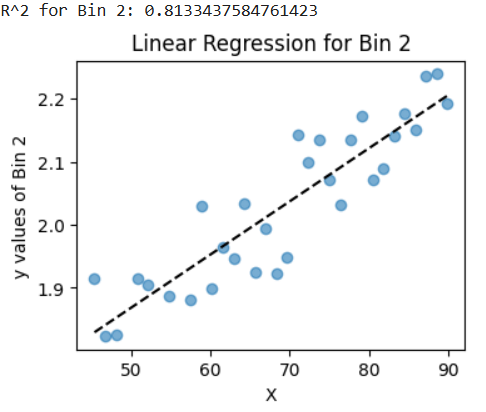
پیاله دوم[[13]](#footnote-13) نیز با استفاده از ریشه دوم به دست آمد و به شرح زیر است:



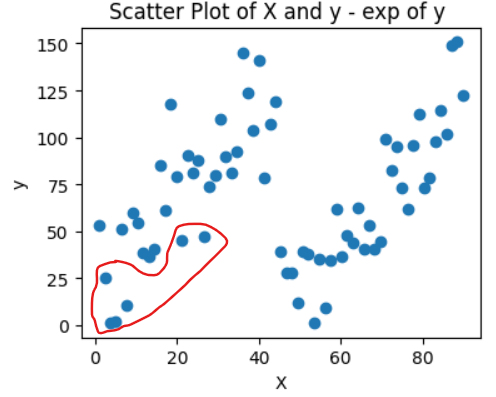
این بخش از داده در نمودار اصلی به شرح زیر است:



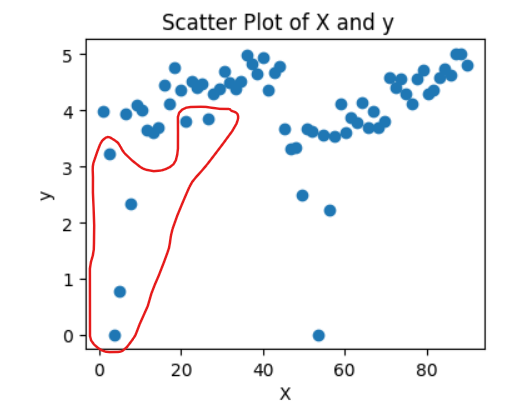
دقت مدل رگرسیون این پیاله نیز برابر با 81.33 است:



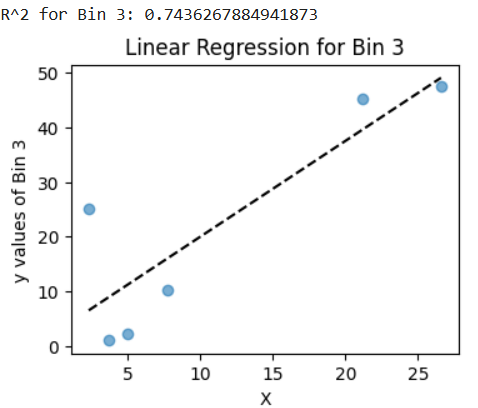
پیاله سوم[[14]](#footnote-14) با استفاده از تبدیل exponential به دست آمده است و به این شرح است:



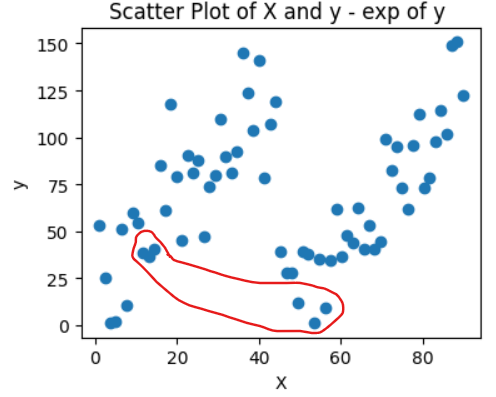
این بخش از داده در نمودار اصلی به شرح زیر است:



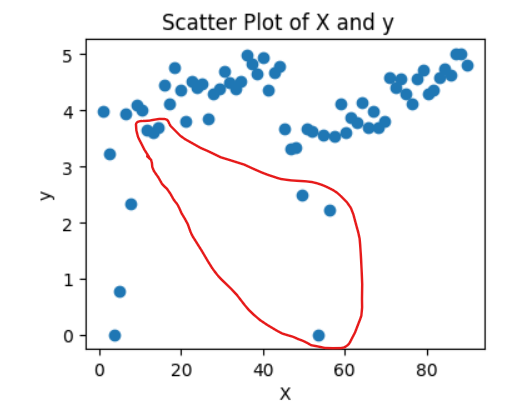
دقت این مدل برابر با 74.36 است:



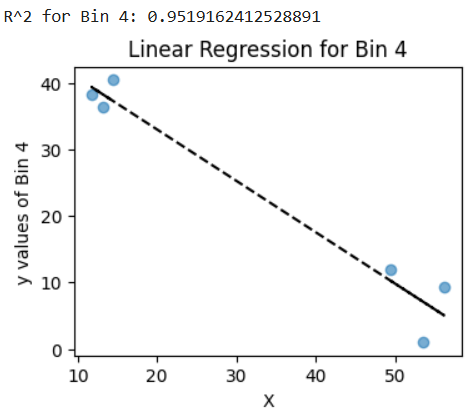
پیاله چهارم[[15]](#footnote-15) نیز با استفاده از تبدیل نمایی یا همان exp به دست آمد:

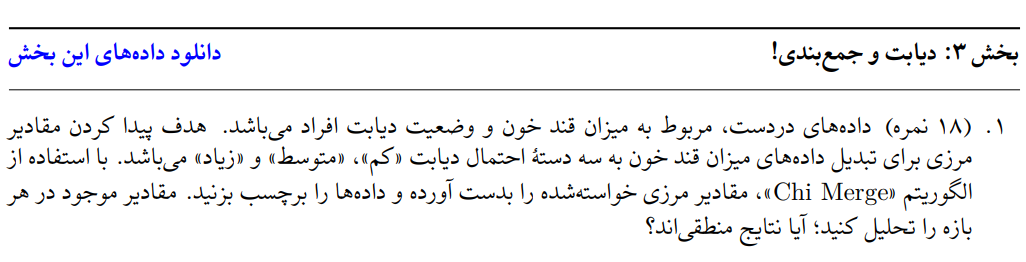


این بخش از داده در نمودار اصلی به شرح زیر است:



دقت این مدل نیز برابر با 95.19 است:





در این سوال

1. Imputation [↑](#footnote-ref-1)
2. Outliers [↑](#footnote-ref-2)
3. Data Type [↑](#footnote-ref-3)
4. Outliers [↑](#footnote-ref-4)
5. Data Type [↑](#footnote-ref-5)
6. Missing values [↑](#footnote-ref-6)
7. Overfitting [↑](#footnote-ref-7)
8. train [↑](#footnote-ref-8)
9. Mean Square Error [↑](#footnote-ref-9)
10. Mean Absolute Error [↑](#footnote-ref-10)
11. transformation [↑](#footnote-ref-11)
12. Bin1 [↑](#footnote-ref-12)
13. Bin2 [↑](#footnote-ref-13)
14. Bin3 [↑](#footnote-ref-14)
15. Bin4 [↑](#footnote-ref-15)