گزارش تمرین دوم مبانی داده‌کاوی و کاربردهای آن (۲۱۰۱۹)

استاد: دکتر مانا مس‌کار
دستیار آموزشی: محمدسبحان کسائی
اعضای گروه: صبا عبدی (۴۰۱۱۰۴۲۷۶)، آوا صدیقی (۴۰۱۱۱۰۱۵۹)

دانشکده مهندسی صنایع

بخش ۱: پرکردن داده‌های گم‌شده

یکی از مراحل ضروری پیش‌پردازش داده‌ها قبل از استفاده به عنوان ورودی مدل‌های یادگیری ماشین، **پر کردن داده‌های گم‌شده**[1] است. در این سوال، به بررسی این مهم پرداخته می‌شود.

۱. (۱۰ نمره) در پرکردن داده‌های گم‌شده، مهم است که سعی شود توزیع داده‌ها تغییر داده نشود تا مدل در هنگام یادگیری، توزیع اصلی داده‌ها را فرا گیرد. فرض کنید داده‌های مربوطه، دارای چولگی به راست باشند، کدامین آماره‌ها (میانگین، میانه، مد) به جهت پرکردن داده‌های گم‌شده مناسب‌تر می‌باشند؟ چرا؟ اگر چولگی به چپ باشد چطور؟

برای پر کردن داده‌ها[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: میانگین ≤ میانه ≤ مد. می‌دانیم که میانگین تا حدودی تحت تأثیر داده‌های پرت[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] 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: مد ≤ میانه ≤ میانگین. مشابه حالت قبل، می‌دانیم که میانگین تا حدودی تحت تأثیر داده‌های پرت[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] 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؛

[1] Imputation
[2] Outliers
[3] Data Type
[4] Outliers
[5] Data Type

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.

۲. (۱۵ نمره) فرض کنید $m+n$ ستون به عنوان داده ستون‌های ویژگی در اختیار شما قرار گرفته‌است و همگی نسبت به یکدیگر مستقل می‌باشند. n ستون آن حاوی داده‌های گم‌شده می‌باشد و می‌خواهید آن‌ها را با مدل‌های یادگیری پر کنید. در ابتدا با استفاده از m ستون دیگر، یکی از ستون‌های حاوی مقادیر گم‌شده را پر می‌کنید و پس از آن، این ستون پرشده را به ستون‌های ویژگی خود برای پر کردن ستون بعدی اضافه می‌کنید (برای مثال برای پر کردن ستون بعدی، از $n+1$ ستون استفاده خواهید کرد). این کار چه آسیبی به دقت پرکردن داده‌ها یا حتی پیش‌بینی نهایی وارد می‌کند؟ راهنمایی: دقت هر مدل i را $1-\alpha_i$ در نظر بگیرید.

در هر دور، با استفاده از مدل یادگیری در دست، داده‌های یک ستون دارای داده‌های گم شده[۶] 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 $1-\alpha_i$ است. بنابراین در هر مرحله، مقدار پیش‌بینی شده مساوی با $1-\alpha_i$ 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 $\prod_{i=1}^{n}(1-\alpha_i)$ 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[۷] 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.

۳. (۷ نمره) در هنگام پر کردن داده‌های گم‌شده، باید به جلوگیری از نشت داده[۲] توجه داشت. این مورد را به طور کامل شرح داده و در هنگام پرکردن داده‌ها ذکر کنید چگونه باید از این عامل جلوگیری کرد؟

هنگامی که در تلاش برای توسعه یک مدل هستیم، برای آن دو بخش داده در نظر می‌گیریم؛ داده‌های آموزش[۸] 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

[6] Missing values
[7] Overfitting
[8] train

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.

بخش ۲ : مدیریت داده‌های پرت **دانلود دادهٔ این بخش**

همانند، پرکردن داده‌های گم‌شده، اگر مدیریت درستی بر روی داده‌های پرت صورت نگیرد، موجب می‌شود دقت مدل کاهش یافته و مدل نهایی مقاوم به این نوع داده‌ها نباشد.

۱. (۲۵ نمره) تابع هدفی برای مسئلهٔ رگرسیون خطی بر روی داده‌های پیوسته به صورت زیر تعریف می‌کنیم و قصد پیش‌بینی y_i را داریم:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (1)$$

در این عبارت N تعداد داده‌ها، y_i داده‌های در دست و $\hat{y}_i$ پیش‌بینی مدل می‌باشد.

تابع هدف را به گونه‌ای تغییر دهید تا نسبت به داده‌های پرت مقاوم عمل کند؛ عبارتی خارج از پرانتز کم یا زیاد نکنید و تنها شکل تابع فعلی را تغییر دهید.

تابعی نیز با بدون درنظر گرفتن محدودیت بالا نیز پیشنهاد دهید و علت مقاوم شدن هر دو تابع پیشنهادی را به طور کامل شرح دهید.

اگر تابع فعلی نسبت به داده‌های پرت مقاوم است، چرا همیشه از این تابع‌ها استفاده نمی‌گردد و تابع پراستفادهٔ مسئلهٔ رگرسیون پیوسته، تابع ذکرشده می‌باشد؟ مزایا و معایب آن‌ها را مقایسه کنید.

تابع هدف اولیه همان MSE[9] 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] استفاده کنیم: $L = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$. 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:

$$L = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}\frac{1}{2}(y_i - \hat{y}_i)^2 & if\ |y_i - \hat{y}_i| \le \delta \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2 & if\ |y_i - \hat{y}_i| > \delta\end{cases}$$

در این تابع، اگر مقدار اختلاف برابر یا کوچکتر از مقداری مانند δ باشد، تابع به *مانند* تابع هدف اصلی، تابع هدف در صورت سوال، محاسبه می‌شود. ولی اگر اختلاف از مقدار δ 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 δ 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 ناریب است و نتایج آن از دیگر تابع‌ها، دقیق‌تر است. علاوه بر این چون $(y_i - \hat{y}_i)^2$ 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.

[9] Mean Square Error
[10] Mean Absolute Error

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.

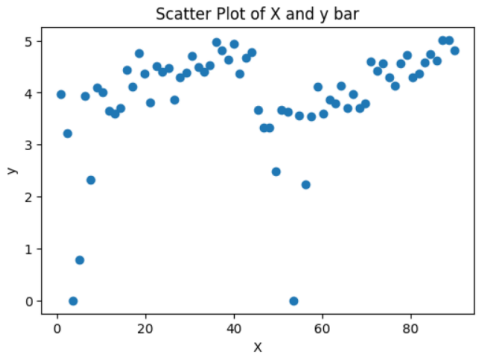
در نهایت، تابع Huber Loss، ترکیبی از دو تابع قبلی است؛ یعنی در مواجهه با اختلاف کوچک مانند MSE رفتار می‌کند و در مواجهه یا اختلاف زیاد مانند MAE رفتار می‌کند. این یعنی سرعت همگرایی مطلوبی دارد و در مواجهه با داده‌های پرت، کمتر تأثیر می‌پذیرد. اما در این تابع، محاسبات موجود پیچیده‌تر هستند و نیاز است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

۲. (۱۵ نمره) اگر تعداد داده‌های موجود به دلایلی کم باشند (مانند هزینهٔ بالایی نمونه‌گیری بیشتر)، باید سعی کرد تا از حذف داده‌ها پرهیز نمود و از داده‌های در دست بیشترین استفاده را داشت. در نگاه اول ممکن است بعضاً داده‌هایی پرت در نظر گرفته باشند و این خطا به علت عدم شناسایی درست توزیع داده‌ها باشد. با استفاده دو تکنیک پیاله‌کردن[۳] و تبدیل[۴] داده‌ها، مدل رگرسیون خطی بر روی داده‌ها برازش داده به‌طوری که معیار R^2 (معیاری برای بیان خوبی برازش بر روی داده‌ها می‌باشد که در این قسمت، کاری با مفهوم آن نداریم) برای مدل هر پیاله، بیشتر از ۶۰ درصد باشد. دقت کنید داده‌ای **نباید** حذف گردد.

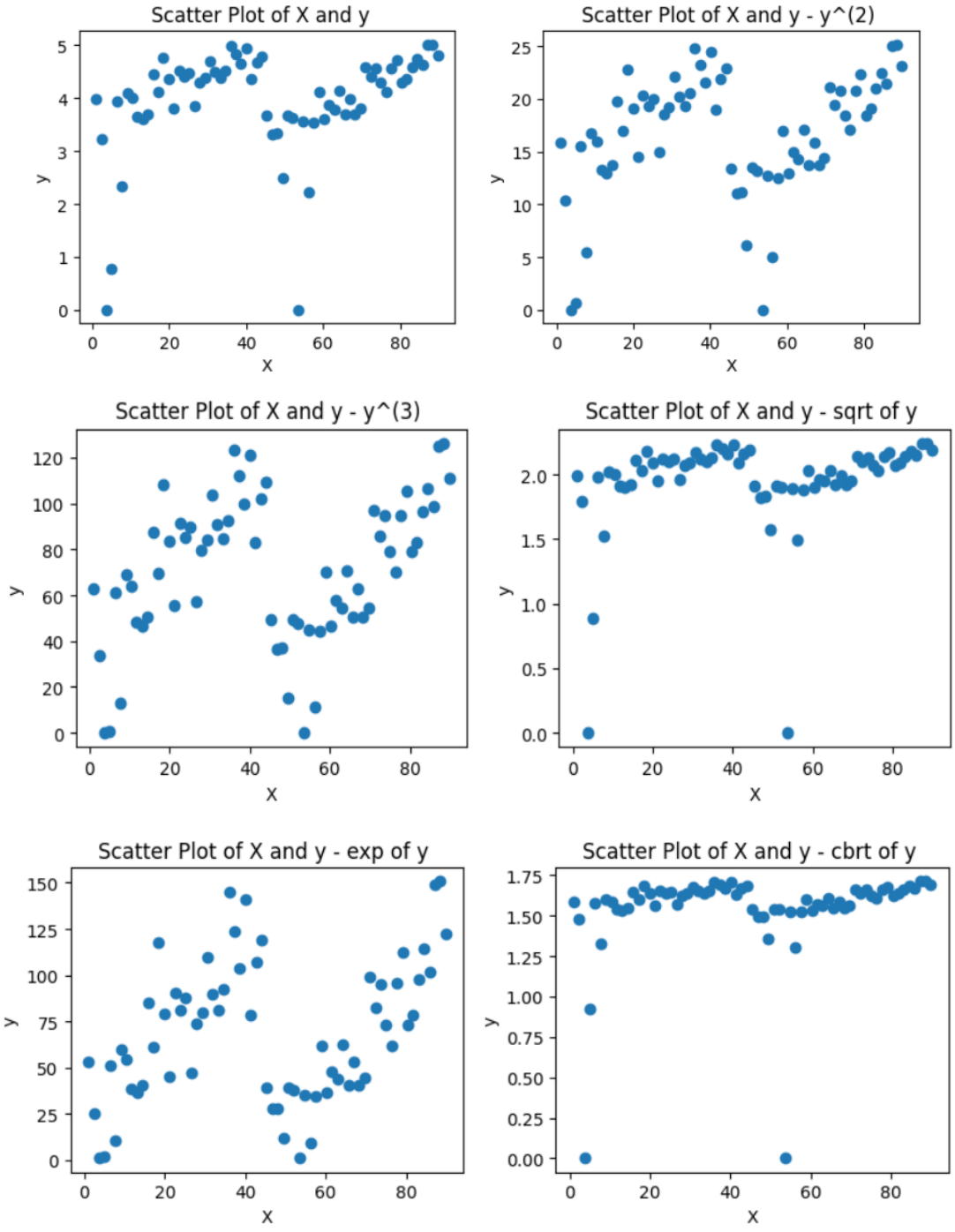
پس از فراخوانی کتابخانه‌های مورد استفاده، داده موردنظر را می‌خوانیم و اطلاعات آن را دریافت می‌کنیم. نوع داده float است و هیچ داده گم‌شده‌ای نداریم.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67 entries, 0 to 66
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   X       67 non-null     float64
 1   y       67 non-null     float64
dtypes: float64(2)
memory usage: 1.2 KB
```

پس از جدا کردن داده‌های X و $\hat{y}$، 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:

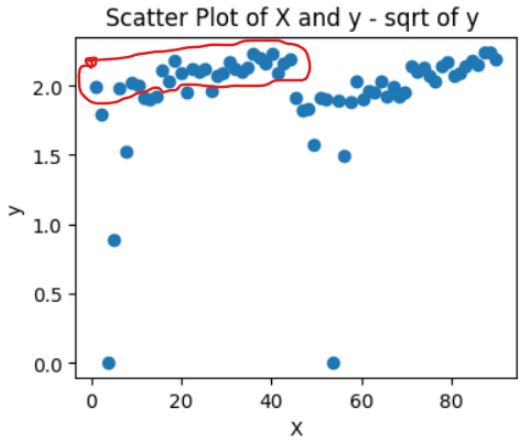


حال برای مشخص کردن پیاله‌ها، چندین تبدیل[11] معروف را روی داده‌ها، مقدار y، پیاده سازی می‌کنیم. این تبدیل‌ها به ترتیب توان دو، توان سه، ریشه دوم، نمایی exp و ریشه سوم هستند. در عکس زیر، این نمودارها کنار یکدیگر به نمایش درآمده‌اند.

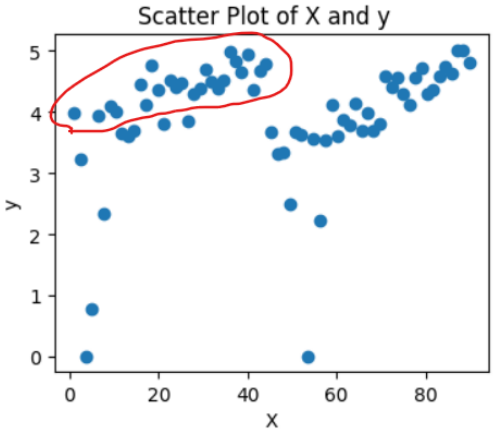


[11] transformation

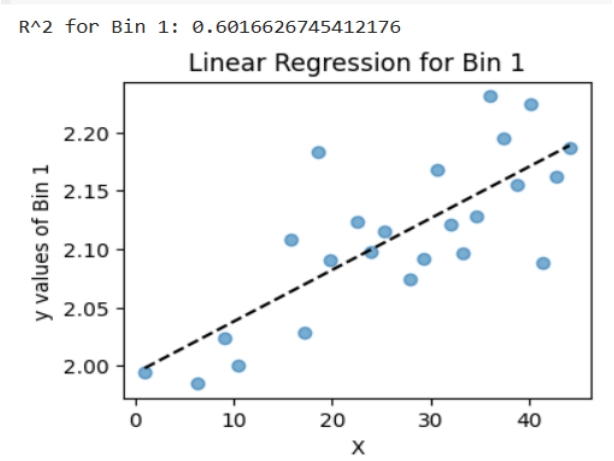
با آزمون و خطای بسیار، در نهایت به ۴ پیاله رسیدیم. پیاله اول[۱۲] با استفاده از ریشه دوم به دست آمده است و به شرح زیر است:



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

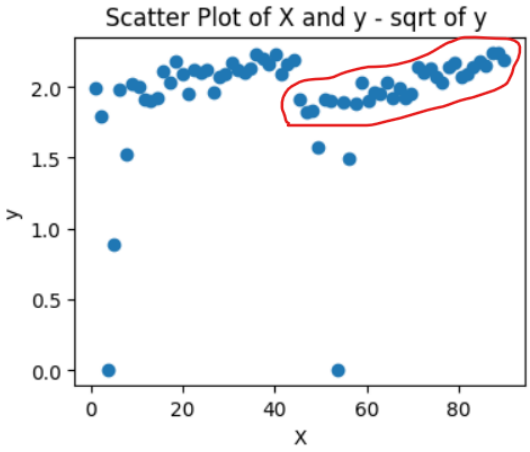


دقت مدل رگرسیون خطی این پیاله برابر با ۶۰.۱۶ درصد می‌باشد.

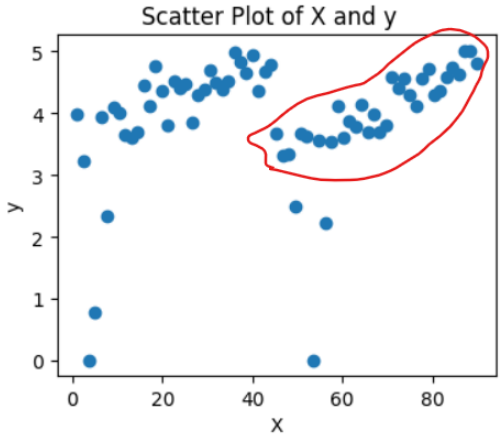


[12] Bin1

پیاله دوم[13] نیز با استفاده از ریشه دوم به دست آمد و به شرح زیر است:

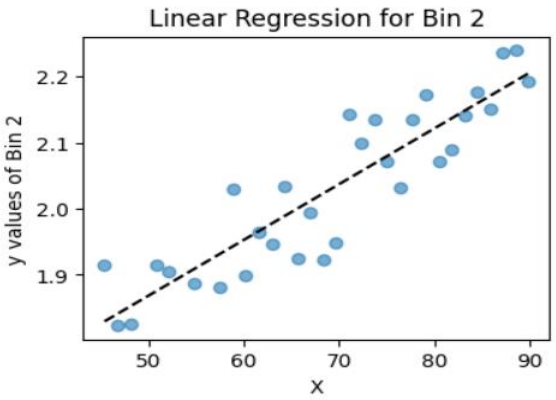


این بخش از داده در نمودار اصلی به شرح زیر است:



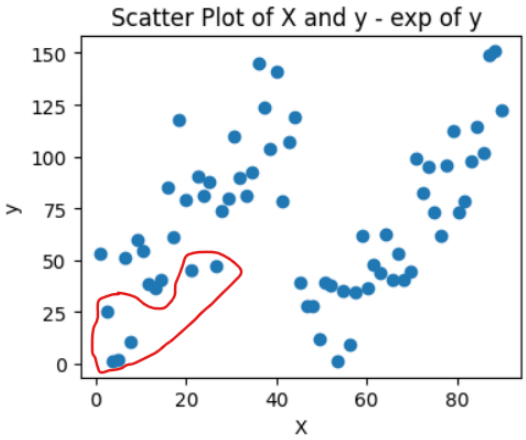
دقت مدل رگرسیون این پیاله نیز برابر با ۸۱.۳۳ است:

```
R^2 for Bin 2: 0.8133437584761423
```

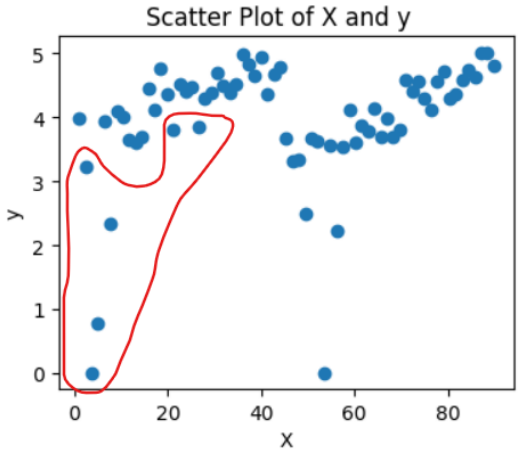


[13] Bin2

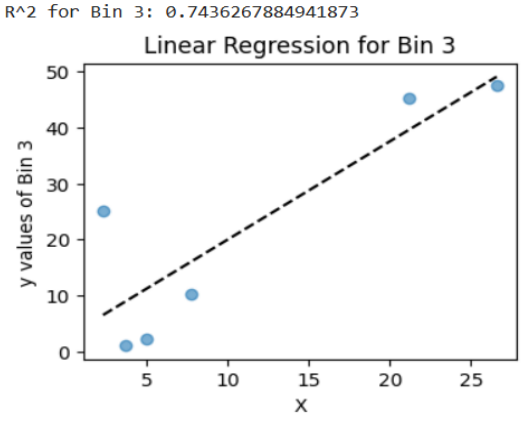
پیاله سوم[14] با استفاده از تبدیل exponential به دست آمده است و به این شرح است:



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

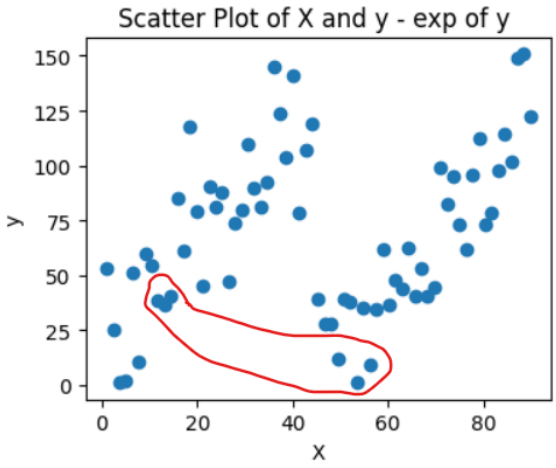


دقت این مدل برابر با ۷۴.۳۶ است:

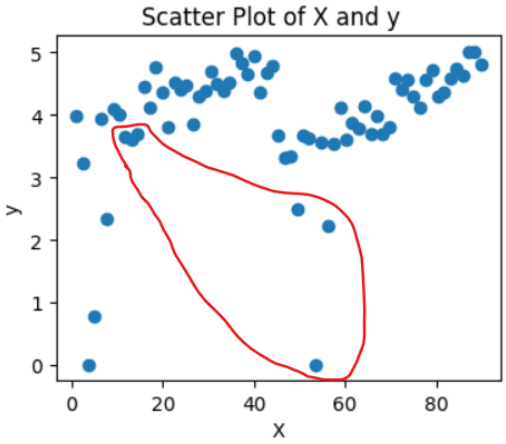


[14] Bin3

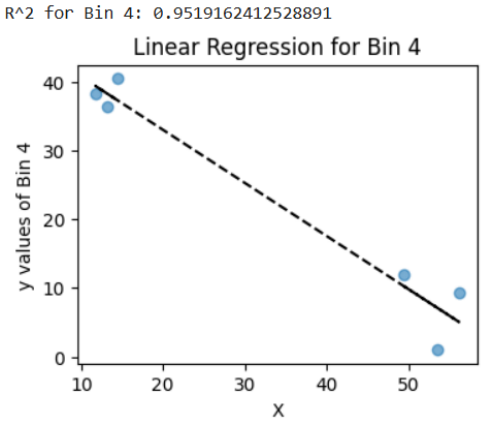
پیاله چهارم[15] نیز با استفاده از تبدیل نمایی یا همان exp به دست آمد:



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



دقت این مدل نیز برابر با ۹۵.۱۹ است:



[15] Bin4

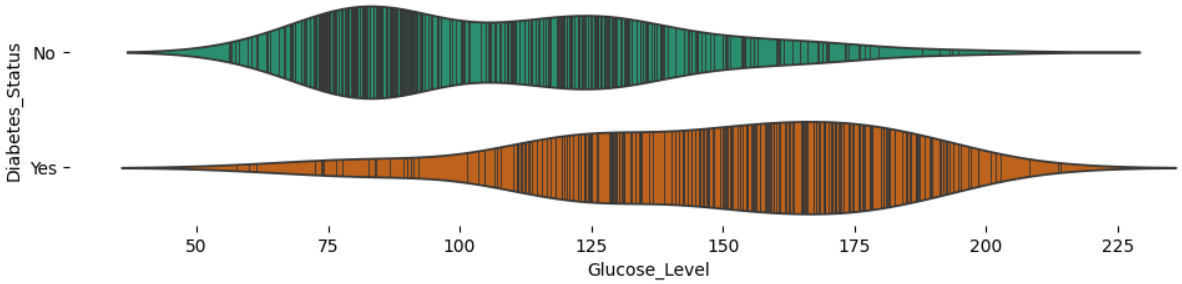
بخش ۳: دیابت و جمع‌بندی!

دانلود داده‌های این بخش

۱. (۱۸ نمره) داده‌های دردست، مربوط به میزان قند خون و وضعیت دیابت افراد می‌باشد. هدف پیدا کردن مقادیر مرزی برای تبدیل داده‌های میزان قند خون به سه دستهٔ احتمال دیابت «کم»، «متوسط» و «زیاد» می‌باشد. با استفاده از الگوریتم «Chi Merge»، مقادیر مرزی خواسته‌شده را بدست آورده و داده‌ها را برچسب بزنید. مقادیر موجود در هر بازه را تحلیل کنید؛ آیا نتایج منطقی‌اند؟

در این سوال ابتدا به کسب شهود و اطلاعات درست از داده می‌پردازیم وکتابخانه‌های مورد نیاز برای این مکار را نصب می‌کنیم.

نمودارهایی که جمینی در گوگل کولب پیشنهاد می‌دهد تا برای درک بهتر داده‌ها از آن‌ها استفاده کنیم را بررسی می‌کنیم. برای نمونه:



داده‌ها کاملا مطابق انتظارمان رفتار می‌کنند. این نشان می‌دهد که افرادی که به دیابت مبتلا هستند اگرچه بطورکلی از قندخون بالاتری برخوردار هستند اما می‌توانند گاهی سطح قند خونی حتی کمتر از بازهٔ نرمال داشته باشند و این به دلیل آن است که دیابت انواع مختلفی دارد و در هرکدام از این انواع (نوع یک و نوع دو) سطح قند خون می‌تواند میزان متفاوتی داشته باشد. اگرچه رایج‌ترین نوع دیابت همان نوع دو است که به دلیل مصرف زیاد قند ایجاد می‌شود و در آن سطح قند خون بالا می‌رود.

با استفاده از الگوریتم کای مرج، سعی می‌کنیم مقادیری مرزی (بازه‌هایی) برای سه احتمال ابتلا به دیابت کم، متوسط و زیاد بیابیم. برای این کار آمارهٔ مربع کای را محاسبه می‌کند و سپس بازه‌های نزدیک را تا رسیدن به ۳ دسته با هم ادغام می‌کند. هر بازه مرز مشخصی دارد و بر آن اساس، مقدار هر ردیف به یک دسته اختصاص داده می‌شود و در نهایت یه ستون جدید با برچسب‌های طبقه‌بندی‌شده به داده‌هایمان اضافه می‌شود.

```
مقادیر مرزی برای هر بازه:
بازه کم: 56.16933597616912 تا 109.94933241070594
بازه متوسط: 110.0723407446404 تا 151.52238945061097
بازه زیاد: 151.66361041562976 تا 214.11163042563825
     Glucose_Level Glucose_Category
0        56.169336               کم
1        56.331494               کم
2        56.607751               کم
3        57.242250               کم
4        57.626245               کم
..             ...              ...
628     202.718770             زیاد
629     208.301678             زیاد
630     209.795287             زیاد
631     213.694088             زیاد
632     214.111630             زیاد

[633 rows x 2 columns]
```

۲. (۴۰ نمره) با توجه به آنچه تاکنون آموخته‌اید، دادهٔ مربوطه را تمیز کنید. این تمیزکاری‌ها شامل:

- اصلاح نام ستون‌ها
- اصلاح نوع داده‌ها[۵]
- یکدست کردن مقادیر ستون‌ها
- مدیریت داده‌های پرت، گم‌شده و ...

می‌باشند. شایان ذکر است اگر موارد دیگری وجود دارد، باید مدیریت آن‌ها نیز صورت گیرد و تنها مثال‌هایی برای راهنمایی آورده شد.

ابتدا جدول را آپلود و مشاهده می‌کنیم تا شهودی از کلیات جدول داشته باشیم. اقدامات اولیه شامل اصلاح نام ستون، اصلاح نوع داده‌ها، یکدست کردن مقادیر ستون را انجام می‌دهیم. یونیک هر ستون را می‌گیریم تا ببینیم که چه چیزهایی تکرار می‌شود. سپس بررسی را جزئی‌تر ادامه می‌دهیم.

۱. کشورها

بخش کشور خیلی خیلی مقادیر بهمریخته و تکراری هستند که نیاز به یکدست کردن دارد. فقط ایالات متحده به ۴۴ روش مختلف نوشته شده است! داده‌هایی که در کشوری هستند اما برای کشور دیگری کار می‌کنند برای درآمد کشوری که ساکن هستند محاسبه شده‌اند. بریتانیای بزرگ مجموعا یکپارچه شده است و انگلستان از آن مجزا نیست. (چراکه به‌جز مواردی که تحقیقات دربارهٔ خود بریتانیا باشد، این کشورها یک مجموعهٔ واحد به حساب می‌آیند.) U.A. امارات متحدهٔ عربی است اما UA در واقع اوکراین است.

۲. جنسیت

جنسیت را به ۳ حالت Male، Female و Other تبدیل کردیم. سایر شامل آن‌هایی که مایل نبودند جنسیت را اعلام کنند و داده‌های NaN می‌باشد.

۳. سایر اصلاحات ابتدایی

```
[76] df_cleaned['Total_Experience'] = df_cleaned['Total_Experience'].replace({
        '3 - 4 years': '2 - 4 years',
        '4 - 4 years': '2 - 4 years',

    })

    df_cleaned['Total_Experience'] = df_cleaned['Total_Experience'].fillna('1 year or less')
    df_cleaned['Total_Experience'].unique()
```

```
array(['8 - 10 years', '11 - 20 years', '31 - 40 years',
       '41 years or more', '5-7 years', '2 - 4 years', '21 - 30 years',
       '1 year or less'], dtype=object)
```

۴. بررسی موارد مشکوک

ابتدا تجربهٔ کاری‌های حوزهٔ تخصصی‌ای که وارد نشده را به مقدار «1 year or less» آپدیت می‌کنیم.

سپس با بررسی سن‌های مرزی مثل زیر ۱۸ سال، یا تحصلات بالا مثل PhD و ترکیب این موارد سعی می‌کنیم موارد عجیب را پیدا کنیم.

مواردی که با سن بسیار کم، مدارک بسیار بالا دارند یا سابقه کار بسیار زیاد دارند بر اساس سابقهٔ کاریشان و مدارکشان، سنشان را تخمین می‌زنیم و پر می‌کنیم. دقت شده است که تخمین‌ها از استدلال‌های کافی برخوردار باشند!!

2245	1206934	25-34	130000.0	12000.0	USD	USA	2 - 4 years	2 - 4 years	PhD	Male
2264	12087un	under 18	220000.0	5000.0	USD	USA	2 - 4 years	1 year or less	PhD	Male
4924	1464025	25-34	105000.0	8000.0	USD	USA	2 - 4 years	2 - 4 years	PhD	Male

چون سابقهٔ کاری ایشان کم است احتمالا سنشان حدود ۲۵ تا ۳۴ سال است.

```
filtered_df = df_cleaned[(df_cleaned['Age'] == '18-24') & (df_cleaned['Education'] == 'PhD')]
filtered_df
```

	ID	Age	Annual_Salary	Additional_Compensation	Currency	Country	Total_Experience	Field_Experience	Education	Gender
15302	2467018	18-24	72800.0	0.0	USD	USA	2 - 4 years	2 - 4 years	PhD	Female

برای اجتناب از دستکاری غیرضروری داده‌ها و با توجه به اینکه فقط یک نفر است و سابقهٔ کار کمی دارد (می‌تواند سابقه کارش مربوط به دورهٔ دکتری باشد) می‌توان پذیرفت که با جهشی خواندن و دکتری مستقیم خواندن توانسته در ۲۴ سالگی مدرک دکتری خود را اخذ کند!!

	ID	Age	Annual_Salary	Additional_Compensation	Currency	Country	Total_Experience	Field_Experience	Education	Gender
	898818	under 18	29120.0	0.0	USD	USA	2 - 4 years	2 - 4 years	High School	Female
	23392un	under 18	31200.0	0.0	USD	USA	2 - 4 years	2 - 4 years	High School	Female
	841918	under 18	32552.0	0.0	CAD	CANADA	2 - 4 years	2 - 4 years	High School	Male
	16282un	under 18	34320.0	0.0	USD	USA	1 year or less	1 year or less	High School	Female
	27973un	under 18	40000.0	0.0	USD	USA	1 year or less	1 year or less	High School	Female
	2463un	under 18	45000.0	1200.0	USD	USA	21 - 30 years	2 - 4 years	Master's degree	Female
	2573518	under 18	100200.0	12000.0	USD	USA	21 - 30 years	21 - 30 years	Some college	Female
	1076318	under 18	118000.0	0.0	USD	USA	31 - 40 years	11 - 20 years	Master's degree	Female

از آنجایی که افراد زیر ۱۸ سال Additional دریافت نمی‌کنند و با توجه به مبالغ حقوق و سوابق کاری، می‌توان گفت که سن سه مورد آخر نادرست است و می‌بایست اصلاح شود.

سن فرد آخر را بین ۵۵ تا ۶۴ قرار داده‌ایم و سن دو فرد دیگر را بین ۴۵ تا ۵۴.

حاصل اصلاحات فوق به شرح زیر است:

```
[78] df_cleaned[(df_cleaned['Age'] == 'under 18') & (df_cleaned['Education'] == 'PhD')]
     df_cleaned.loc[df_cleaned['ID'] == '12087un', 'Age'] = '25-34'
     df_cleaned.loc[df_cleaned['ID'] == '2518418', 'Age'] = '55-64'
     df_cleaned.loc[df_cleaned['ID'] == '1076318', 'Age'] = '55-64'
     df_cleaned.loc[df_cleaned['ID'] == '2573518', 'Age'] = '45-54'
     df_cleaned.loc[df_cleaned['ID'] == '2463un', 'Age'] = '45-54'
     df_cleaned.loc[df_cleaned['ID'] == '16282un', 'Education'] = 'High School'
```

خط آخر مربوط به تحصلات خالی برای فردیست که درآمد کم (در اندازهٔ درآمد افراد زیر ۱۸ سال و محصل) دارد و زیر ۱۸ سال سن دارد و احتمالا تحصیلات او نیز دبیرستان است.

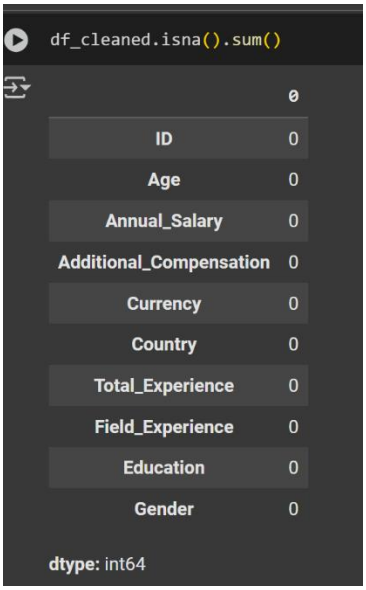
25109	823654	45-54	260000.0	78000.0	USD	USA	31 - 40 years	31 - 40 years	High School	Mal
26425	942954	45-54	264000.0	500000.0	USD	USA	31 - 40 years	31 - 40 years	College degree	Mal
8741	1840264	55-64	265000.0	0.0	USD	USA	31 - 40 years	31 - 40 years	Master's degree	Othe

از آنجایی که با همچین درآمدری، نباید اینهمه کمک از دولت دریافت کند و با در نظر گرفتن مدرک تحصیلی اش، احتمالی وجود دارد که رقم درآمدش را اشتباه وارد کرده باشد. (شاید یک صفر اضافه) اما چون شواهد کافی نیست، این مورد را بدون تغییر باقی می‌گذاریم.

حال مجددا خانه‌های خالی را بررسی می‌کنیم و تنها خانه‌ای که هنوز مقدار نامشخص دارد خانهٔ تحصیلات است.

```
df_known = df_cleaned[df_cleaned['Education'].notna()]
df_missing = df_cleaned[df_cleaned['Education'].isna()]

from sklearn.preprocessing import LabelEncoder

df_model = df_cleaned.copy()
le_age = LabelEncoder()
le_experience = LabelEncoder()

df_model['Age_encoded'] = le_age.fit_transform(df_model['Age'].astype(str))
df_model['Experience_encoded'] = le_experience.fit_transform(df_model['Total_Experience'].astype(str))

features = ['Age_encoded', 'Annual_Salary', 'Additional_Compensation', 'Experience_encoded']

from sklearn.ensemble import RandomForestClassifier

X_train = df_model.loc[df_model['Education'].notna(), features]
y_train = df_model.loc[df_model['Education'].notna(), 'Education']

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

X_missing = df_model.loc[df_model['Education'].isna(), features]
predicted_education = model.predict(X_missing)

df_cleaned.loc[df_cleaned['Education'].isna(), 'Education'] = predicted_education
```

برای حدس تحصسلات افراد از الگوریتم فوق استفاده می‌کنیم. در این الگوریتم اثر سن، درآمد، سابقهٔ کار و مقداری که از دولت یارانه دریافت می‌کنند دیده شده است و به روی پیشبینی اعمال شده است.



در نهایت می‌بینیم که همهٔ داده‌های نامشخص نیز مدیریت شده‌اند و با مقادیر معقولی تخمین زده شده اند. از آنجایی که تعداد ردیف‌ها حدود ۲۷۰۰ تاست و تعداد مقادیر تخمین زده شده کمتر از ۲۰۰ تاست، می‌توان ادعا کرد که حتی در صورت عدم صحت حدس‌هایمان، شاهد اثر سهمگینی به روی رفتار داده‌ها نخواهیم بود.

در نهایت هم فایل را ذخیره می‌کنیم و دانلود می‌کنیم تا از داده‌های تمیز و آماده استفاده کنیم.