امیربهنام - ۹۸۳۱۱۳۳

۱-

الف) داده های پرت و نویز هر دو شکلی از داده های نامنظم هستند که می‌توانند دقت و تجزیه و تحلیل داده ها را مختل کرده و بر آن تاثیر بگذارند. تفاوت اصلی بین این دو نوع داده در این است که داده های پرت مقداری است که به طور قابل توجهی با سایر مقادیر موجود در مجموعه داده متفاوت است در حالی که نویز نوسانات تصادفی در داده ها است که ممکن است از نظر مقدار یا اندازه قابل توجه نباشد. داده های پرت معمولاً مفید در نظر گرفته می‌شوند و ممکن است بینش های ارزشمندی را ارائه دهند در حالی که نویز عموماً به عنوان یک مسئله نامطلوب در نظر گرفته می‌شود که می‌تواند الگوهای واقعی در داده ها را مبهم کند.

ب) داده های پرت می‌تواند در تشخیص خطاها یا ناهنجاری ها در یک مجموعه داده مفید باشد. به عنوان مثال، در زمینه پزشکی، اطلاعات پرت در داده های بیماران می‌تواند به شناسایی بیماری ها یا شرایط نادری که نیاز به بررسی بیشتر دارند کمک کند. نقاط پرت در داده های مالی همچنین می‌تواند به کشف فرصت های سرمایه گذاری منحصر به فرد کمک کند.

ج) بله - اگر نویز به طور قابل توجهی با سایر نقاط داده در یک مجموعه داده متفاوت باشد ممکن است یک نقطه داده پرت باشد. با این حال، همه نویزها به عنوان داده های پرت در نظر گرفته نمی‌شوند. داده های پرت معمولاً یک رویداد شدید و نادر در نظر گرفته می‌شوند در حالی که نویز می‌تواند متناوب‌تر، تصادفی‌تر باشد و الگوی خاصی را منعکس نکند.

امیربهنام - ۹۸۳۱۱۳۳

۲- در زمینه داده کاوی و انبار داده و تحلیل و گزارش دهی و سایر اهداف هوش تجاری جمع آوری و سازمان دهی شده است. هدف اصلی تیم سفزن بزرگ و مرکزی از داده ها است که برای تجزیه و تحلیل آسان است. تجمیع و ذخیره و مدیریت داده های تاریخی از منابع متعدد در یک مکان واحد برای دسترسی و تجزیه

از سوی دیگر، پایگاه داده مجموعه ای از داده ها است که با روش خاص سازمان دهی شده اند تا امکان ذخیره سازی، بازیابی و پرس و جوی کارآمد داده ها را فراهم کنند.

شباهت ها :

۱- هر ۲ داده ها را به روش ساختار یافته ذخیره و سازماندهی می کنند.

۲- هر ۲ از SQL برای پرس و جو و گزارش استفاده می کنند.

۳- هر ۲ داده ها را به صورت الکترونیکی ذخیره می کنند و ظرفیت سازماندهی و بازیابی اطلاعات را دارند.

۴- هر ۲ به مدل سازی و طراحی داده نیاز دارند تا اطمینان حاصل شود که داده ها دقیق و قابل اعتماد هستند.

۵- هر ۲ می توانند از پشتیبان گیری داده ها و برنامه های بازیابی فاجعه برای محافظت در برابر از دست دادن داده ها پشتیبانی کنند.

تفاوت ها :

۱- هدف: یک پایگاه داده برای مدیریت و ذخیره داده های عملیاتی برای پشتیبانی از تراکنش های روزانه طراحی شده است، در حالی که یک انبار داده برای پشتیبانی از تصمیم گیری با ارائه بینش های معنادار از داده های تاریخی طراحی شده است.

۲- استفاده از داده (Data Usage): یک پایگاه داده برای پشتیبانی از تراکنش های با درگیری استفاده می شود، در حالی که انبارهای داده برای پشتیبانی از پرس و جوهای تحلیلی داده های تاریخی استفاده می شود.

۳- حجم داده (Data Volume): یک پایگاه داده معمولاً داده های محدود مربوط به یک برنامه خاص را ذخیره می کند، در حالی که یک انبار داده مقادیر زیادی از داده ها را از منابع متعدد ذخیره می کند که می تواند برای بینش های تحلیلی استفاده شود.

۴- ساختار داده (Data Structure): یک پایگاه داده دارای ساختار نرمال شده است، به این معنی که داده ها را در جداول با حداقل افزونگی ذخیره می کند تا مشکلات ذخیره سازی داده ها و ناسازگاری را کاهش دهد. از طرف دیگر، یک انبار داده ساختاری غیرعادی دارد، به این معنی که داده ها را به گونه ای ذخیره می کند که امکان دسترسی و تجزیه و تحلیل سریع را فراهم می کند، حتی با هزینه اضافی.

امیرپنام - ۹۸۳۱۱۳۳

۳- یکی از راه های تشخیص داده های پرت در یک مجموعه داده معمولی توزیع شده، استفاده از interquartile range (IQR) است. IQR معیاری برای تغییرپذیری است که گستردگی ۵۰ درصد میانی مجموعه داده را توصیف می‌کند و به عنوان تفاوت بین ۷۵ Percentile (Q۳) و ۲۵ Percentile (Q۱) داده ها محاسبه می‌شود.

سپس می‌توان از IQR برای تعیین آستانه ای برای تشخیص نقاط پرت استفاده کرد. به طور خاص، هر مشاهده ای که کمتر از $Q_1 - 1.5 \times IQR$ یا بالاتر از $Q_3 + 1.5 \times IQR$ باشد، ممکن است یک داده پرت در نظر گرفته شود. به طور خلاصه، فرآیند تشخیص نقاط پرت با استفاده از Percentile در یک مجموعه داده به طور معمول توزیع شده شامل مراحل زیر است:

گام ۱: ۲۵ Percentile و ۷۵ داده ها را محاسبه می‌کنیم.

گام ۲: IQR را به عنوان اختلاف بین Percentile ۷۵ و ۲۵ محاسبه می‌کنیم.

گام ۳: کران پایین ($Q_1 - 1.5 \times IQR$) و کران بالا ($Q_3 + 1.5 \times IQR$) را محاسبه می‌کنیم و نقاطی که خارج از مرزهای پایین و بالایی قرار دارند را به عنوان outlier شناسایی می‌کنیم.

امیربهنام - ۹۸۳۱۱۴۳

۴-

الف) فرآیند پاکسازی داده ها (Data cleansing) شامل شناسایی و تصحیح خطاها، ناسازگاری ها و نادرستی ها در یک مجموعه داده برای بهبود کیفیت و دقت داده ها است. این فرآیند شامل حذف داده های تکراری یا گمشده، تصحیح داده های نادرست یا متناقض و اعتبارسنجی داده ها در برابر منابع شناخته شده.

ب) Data visualization مهم است زیرا به کاربران اجازه می دهد از داده هایی که ممکن است با فاصله یا نگاه کردن به اعداد و جداول آشکار نشوند، بینش به دست آورند. این می تواند به شناسایی روندها، الگوها و نقاط پرت در داده برای تصمیم گیری کمک کند. همچنین، اهمیت دیگر آن این است که این فرآیند می تواند درک و تحلیل آسان تر الگوها، روندها و روابط درون داده ها را امکان پذیر سازد.

تفسیر و تبادل از چالش های اصلی Data visualization، انتخاب نوع مناسب تجسم برای استفاده با توجه به ماهیت، پیچیدگی و هدف داده ها است. زیرا تاثیر مهمی بر اثربخشی تجسم به عنوان ابزاری برای کاوش و ارتباط داده دارد. انتخاب تکنیک تجسم مناسب مستلزم در نظر گرفتن عوامل مختلفی مانند ماهیت داده ها، هدف تجسم، مخاطب مورد نظر و سطح جزئیات مورد نیاز برای تفسیر داده ها است.

ج) Data cleansing فرآیند مهم است زیرا دقت و سازگاری داده های مورد استفاده برای Data visualization را تضمین می کند. بدون Data cleansing، نمایش داده ها (Data visualization) ممکن است گمراه کننده یا نادرست باشد زیرا احتمال بیشتری وجود دارد که تجزیه و تحلیل داده ها تحت تاثیر خطاها و ناهماهنگی ها قرار گیرد که منجر به تصمیم گیری ضعیف می شود. بنابراین پاکسازی داده ها پیش نیازی برای نمایش موثر داده ها است زیرا تضمین می کند که visualizations به طور دقیق داده های اساس را منعکس می کنند.

امیرپنام - ۹۸۱۳۱۳۳

۵-

الف)

$$\text{cosine similarity}(G_1, G_2) = \frac{(G_1 \cdot G_2)}{(\|G_1\| \, \|G_2\|)}$$

$$(G_1 \cdot G_2) = (-3 \times 9) + (5 \times 20) + (8 \times 16) + (-2 \times 8) + (1 \times 2) + (2 \times 15) + (3 \times -6) + (-5 \times -15)$$

$$+ (10 \times 25) + (-1 \times -2) \Rightarrow (G_1 \cdot G_2) = 516$$

$$\|G_1\| = \sqrt{(-3)^2 + 5^2 + 8^2 + (-2)^2 + 1^2 + 2^2 + 3^2 + (-5)^2 + (10)^2 + (-1)^2}$$

$$\Rightarrow \|G_1\| = \sqrt{244} \Rightarrow \|G_1\| \simeq 15.56$$

$$\|G_2\| = \sqrt{9^2 + 20^2 + 16^2 + 8^2 + 2^2 + 10^2 + (-6)^2 + (-15)^2 + 25^2 + (-2)^2}$$

$$\Rightarrow \|G_2\| = \sqrt{1960} \Rightarrow \|G_2\| \simeq 44.86$$

$$\text{cosine similarity} = \frac{516}{15.56 \times 44.86} \Rightarrow \text{cosine similarity} \simeq 0.811$$

$$\text{correlation}(G_1, G_2) = \frac{n \sum G_1 \times G_2 - \sum G_1 \times \sum G_2}{\sqrt{[n \sum G_1^2 - (\sum G_1)^2][n \sum G_2^2 - (\sum G_2)^2]}}$$

$n = 10$

$$\sum (G_1) = -3 + 5 + 8 - 2 + 1 + 2 + 3 - 5 + 10 - 1 \Rightarrow \sum (G_1) = 18$$

$$\sum (G_2) = 9 + 20 + 16 + 8 + 2 + 10 - 6 - 15 + 25 - 2 \Rightarrow \sum (G_2) = 67$$

$$\sum (G_1^2) = (-3)^2 + 5^2 + 8^2 + (-2)^2 + 1^2 + 2^2 + 3^2 + (-5)^2 + (10)^2 + (-1)^2 \Rightarrow \sum (G_1^2) = 244$$

$$\sum (G_2^2) = 9^2 + 20^2 + 16^2 + 8^2 + 2^2 + 10^2 + (-6)^2 + (-15)^2 + 25^2 + (-2)^2 \Rightarrow \sum (G_2^2) = 1670$$

$$\sum (G_1 \times G_2) = (-3 \times 9) + (5 \times 20) + (8 \times 16) + (-2 \times 8) + (1 \times 2) + (2 \times 10) + (3 \times -6)$$

$$+ (-5 \times -15) + (10 \times 25) + (-1 \times -2) \Rightarrow \sum (G_1 \times G_2) = 516$$

$$\text{correlation}(G_1, G_2) = \frac{10 \times 516 - 18 \times 67}{\sqrt{(10 \times 244 - 18^2) \times (10 \times 1670 - 67^2)}}$$

$$\Rightarrow \text{correlation} = \frac{3954}{5059.07} \Rightarrow \text{correlation} \simeq 0.781$$

امیرهام - ۹۸۳۱۱۳۳

۵ -

الف - ادامه)

$$i(x;y) = H(x) + H(y) - H(x,y)$$

$$\Rightarrow i(G_1,G_2) = H(G_1) + H(G_2) - H(G_1,G_2)$$

$$H(G_1) = -\left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)\log_2\left(\frac{1}{10}\right) \Rightarrow H(G_1) = 3.32$$

به طور مشابه برای ۲ تا هم همین روند تکرار می‌شود و $H(G_1) = H(G_2)$ پس داریم:

$$\Rightarrow H(G_2) = 3.32$$

$$\Rightarrow H(G_1,G_2) = 6.107$$

$$\Rightarrow H(G_1) + H(G_2) - H(G_1,G_2) = 6.64 - 6.107 = 0.533$$

ب) بر اساس مقادیر هر معیار می‌توان گفت که cosine و correlation در نسبت با سایر معیارها نشان دهنده رابطه‌ی قوی بین G1 و G2 است اما برای mutual information تعداد بیشتری به دست آمده است که نشان دهنده رابطه‌ی ضعیف‌تر آنها است.

بنابراین می‌توان گفت که مستقل نیستند و درجاتی با هم ارتباط دارند.

2) نتایج به دست آمده با استفاده از معیارهای مختلف ممکن است به دلیل تفاوت در مفروضات اساسی و ویژگی‌های هر معیار متفاوت باشد. برای مثال، شباهت کسینوسی فقط بزرگی بردارها را در نظر می‌گیرد در حالی که correlation هم قدر و هم جهت آنها را در نظر می‌گیرد. از سوی دیگر mutual information، اطلاعات به اشتراک گذاشته شده توسط ۲ متغیر را بدون رابطه آنها جمع‌آوری می‌کند.

بنابراین، هنگام استفاده از معیارهای شباهت مختلف، به دست آوردن نتایج متفاوت متداول است و انتخاب معیار مناسب بر اساس ویژگی‌های داده‌ها و سوال تحقیق مهم است.

امیرپنام - ۹۸۳۱۱۳۳

۴- Aggregation فرآیندی در پیش پردازش داده است که در آن چندین ردیف از داده ها ترکیب شده و در یک ردیف خلاصه می شوند. این کار معمولاً به منظور ساده سازی داده ها و قابل مدیریت تر کردن آنها برای تجزیه و تحلیل انجام می شود. Aggregation می تواند شامل عملیات عددی مانند جمع آوری، میانگین گیری یا یافتن حداقل یا حداکثر مقدار یک ستون برای هر گروه از داده ها باشد.

مزایا:

۱- ساده سازی داده ها: با جمع آوری داده ها می توان میزان داده ها را کاهش داد و درک آن نیز ساده تر می شود.

۲- کاهش نویز: تجمیع داده ها می تواند به کاهش نویز درون داده ها با تجمیع اطلاعات اضافی و حذف موارد پرت کمک کند.

۳- تجزیه و تحلیل بهتر: جمع آوری داده ها امکان تجزیه و تحلیل بهتر اطلاعات را فراهم می کند و می تواند به کشف الگوهایی کمک کند که در غیر این صورت پنهان می شدند.

معایب:

۱- از دست دادن داده ها: یکی از نگرانی های اصلی هنگام برخورد با تجمیع (Aggregation) این است که ممکن است اطلاعات مهمی را در فرآیند از دست دهیم.

۲- نتایج غرضانه: اگر تجمیع با دقت انجام نشود، می تواند منجر به نتایج غرضانه شود.

۳- تجزیه و تحلیل ناقص: تجمیع ممکن است دیدگاه محدودی و نادرست از داده ها را ارائه دهد که ممکن است منجر به تجزیه و تحلیل ناقص شود.

Sampling در پیش پردازش داده ها فرآیند انتخاب زیرمجموعه ای نماینده از داده ها از یک مجموعه داده بزرگتر به منظور به حداقل رساندن هزینه ها، افزایش کارایی و بهبود دقت تجزیه و تحلیل است. این شامل انتخاب نقاط داده از یک جمعیت و استفاده از آنها برای استنتاج و پیش بینی در مورد کل جمعیت است.

مزایا:

۱- صرفه جویی در زمان و منابع: نمونه برداری به کاهش حجم داده ها کمک می کند که باعث صرفه جویی در زمان و منابع محاسباتی در پردازش داده ها می شود.

۲- بهره وری بهبود یافته: نمونه برداری می تواند داده ها را قابل مدیریت تر کند و کارایی مدل های ایجاد شده از داده ها را بهبود بخشد.

۳- ارزیابی دقیق: از نمونه گیری می توان برای ارزیابی دقت مدل ها با اعتبارسنجی نتایج از طریق داده های آزمایشی استفاده کرد.

امیرپنام - ۹۸۳۱۴۳

۶- ادامه)

Sampling - معایب :

۱- خطر سوگیری نمونه‌گیری : در صورتی که نمونه معرف مجموعه داده بزرگتر نباشد، سوگیری نمونه‌گیری می‌تواند رخ دهد که منجر به نتایج نادرست می‌شود.

۲- خطر undersampling (کم نمونه‌سازی) یا oversampling (بیش از حد نمونه‌گیری) : تکنیک‌های نمونه‌گیری نادرست می‌تواند منجر به کم نمونه‌سازی یا بیش از حد نمونه‌گیری از یک زیرگروه خاص در داده‌ها شود که منجر به نتایج نادرست شود.

۳- از دست دادن اطلاعات : نمونه‌برداری میزان داده‌های مورد استفاده برای تجزیه و تحلیل را کاهش می‌دهد که ممکن است باعث از دست رفتن اطلاعات ارزشمند شود.

امیرنام_۹۸۳۱۱۴۳

۷-

الف)

Feature selection: Feature selection (انتخاب ویژگی) تکنیک انتخاب و استراج مهمترین ویژگی ها در یک مجموعه داده است، در حالی که ویژگی های کم اهمیت را کنار می گذارند. به عبارت دیگر، به کاهش تعداد ویژگی های یک مجموعه داده کمک می کند و در عین حال تا حد امکان اطلاعات اصلی را حفظ می کند.

Feature Extraction: استخراج ویژگی شامل تبدیل داده های با ابعاد بالا به فضایی با ابعاد پایین تر است. این تکنیک شامل ایجاد ویژگی های جدید از ترکیب ویژگی های موجود است. این الگوها را در داده ها شناسایی و ترکیب می کند تا مجموعه ای کاهش یافته از ویژگی ها را ایجاد کند که حداکثر واریانس در داده ها را توضیح می دهد.

Feature Engineering: مهندسی ویژگی فرآیند ایجاد ویژگی های جدید از داده های خام با ترکیب، تبدیل یا دستکاری آنها است. این یک فرآیند خلاقانه و تکراری است که شامل ترکیبی از دانش حوزه، شهود و آزمایش ویژگی ها می شود تا عملکرد مدل بهتری حاصل شود.

تفاوت ها:

۱- Feature selection و Feature Extraction بر کاهش ابعاد مجموعه داده تمرکز دارد در حالی که Feature Engineering بر ایجاد ویژگی های جدید به ای بهبود عملکرد مدل تمرکز دارد.

۲- Feature selection فقط مرتبط ترین ویژگی ها را انتخاب می کند در حالی که Feature Extraction می تواند ویژگی های جدید را از ویژگی های موجود ایجاد کند.

۳- Feature Engineering شامل ایجاد دستی یا خودکار ویژگی های جدید است در حالی که Feature selection و Feature Extraction فقط شامل اصلاح یا انتخاب ویژگی های موجود است.

۴- Feature selection و Feature Extraction روش های داده محور هستند، در حالی که Feature Engineering یک رویکرد دانش محور است.

۵- Feature selection یک رویکرد ساده تر از Feature Extraction و Feature Engineering است، زیرا فقط شامل شناسایی ویژگی های مهم است. استخراج ویژگی و مهندسی ویژگی تکنیک های پیچیده ای هستند زیرا شامل ایجاد ویژگی های جدید که در مجموعه داده ها نیستند یا اصلاح ویژگی های موجود برای بهبود عملکرد مدل هستند.

۶- Feature selection و Feature extraction به طور کلی برای الگوریتم های یادگیری ماشین برای بهبود دقت و کارایی آنها اعمال می شود. از سوی دیگر Feature Engineering در زمینه های متعددی از تجزیه و تحلیل داده ها مانند بینایی کامپیوتر، پردازش زبان طبیعی و تجزیه و تحلیل سری های زمانی اعمال می شود.

امیر بهنام - ۹۸۳۱۱۳۳

۷-

الف- ادامه)

تفاوت ها:

۷- تفاوت اصلی بین این ۳ تکنیک این است که انتخاب ویژگی با انتخاب آموزنده ترین ویژگی هایی که برای کار مهم هستند مجموعه ویژگی را محدود می کند در حالی که استخراج ویژگی با ترکیب ویژگی های اصلی مجموعه جدیدی از ویژگی ها را ایجاد می کند. از سوی دیگر مهندسی ویژگی فرآیندی است برای اصلاح ویژگی ها یا ایجاد ویژگی های جدید برای بهبود عملکرد مدل.

ب)

تفاوت اصلی بین الگوریتم های کاهش بعد خطی و غیرخطی در نحوه تبدیل داده ها است. الگوریتم های کاهش بعد خطی از تبدیل های خطی داده ها استفاده می کنند و بر این معنی که با این فرض کار می کنند که داده ها را می توان با یک زیرفضای خطی کم بعد نشان داد. آنها محورهای خطی جدیدی را در داده ها پیدا می کنند که بیشترین تنوع را به خود اختصاص می دهند و داده ها را روی این محورها پخش می کند. از سوی دیگر الگوریتم های غیرخطی داده ها از تبدیل های غیرخطی استفاده می کنند و با این فرض کار می کنند که داده ها روی یک منیفولد غیرخطی یا نزدیک آن قرار دارند. آنها سعی می کنند روابط غیرخطی پیچیده ای را بین متغیرهایی که نمی توانند با تبدیل های خطی ضبط کنند و داده ها را در فضای کم بعدی در حالی که ساختار داده ها را حفظ می کنند ارائه می دهد.

به طور خلاصه، الگوریتم های کاهش ابعاد خطی ساده تر و سریع تر و به زمان محاسباتی کمتری نیاز دارند اما ممکن است پیچیدگی کامل داده ها را در بر نگیرد. الگوریتم های غیرخطی می توانند الگوهای پیچیده تری را در داده ها ثبت کنند اما ممکن تفسیر آنها سخت تر باشد.

PCA ← یک الگوریتم کاهش ابعاد خطی است که با یافتن جهت دارای بیشترین واریانس در داده ها و نمایش داده ها بر روی این جهات کار می کند. این امر ابعاد داده ها را کاهش می دهد و در عین حال بیشتر اطلاعات را حفظ می کند. پیش بینی حاصل، نمایش با ابعاد پایین تر از داده های اصلی است که در آن ویژگی های جدید (مولفه های اصلی) همبستگی ندارند و بیشتر تغییرات داده ها را به تصویر می کشند.

t-SNE ← یک الگوریتم کاهش ابعاد غیرخطی است که با حفظ شباهت های بین نقاط داده در یک فضای کم بعد کار می کند. سعی می کند نقشه ای از فضای با ابعاد بالا به فضای کم بعد پیدا کند که ساختار محلی داده ها را حفظ کند. در واقع می توان گفت که ابعاد داده ها را با به حداقل رساندن واگرایی بین شباهت های زوجی نقاط با ابعاد بالا و کم بعد کاهش می دهد.

امیرپهام - ۹۸۳۶۱۳۳

۸- ابتدا داده ها را مرتب می‌کنیم:

Q_1 Q_3

۱, ۳, ۲۵, ۲۶, ۲۷, ۲۷, ۲۷, ۲۸, ۲۹, ۳۳, ۳۶, ۳۹, ۴۱, ۴۹, ۷۰

median

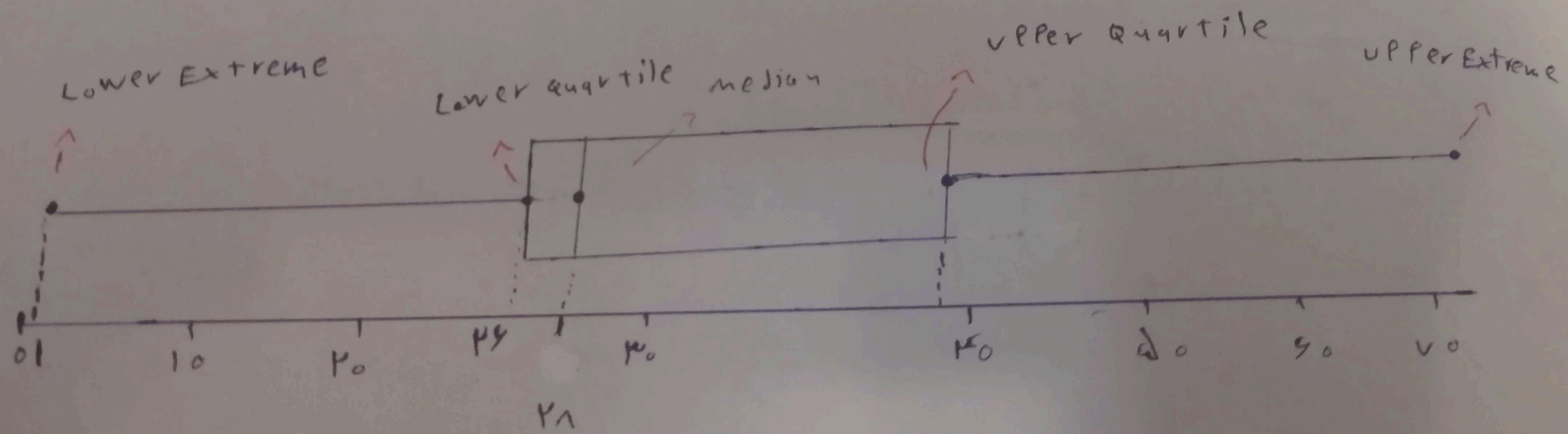
minimum = ۱

max = ۷۰

median = ۲۸

Lower Quartile = ۲۶

upper quartile = ۳۹



امیرپنام - ۹۸۳۶۱۳۳

۹-

الف) نمودار Q-Q یک تکنیک گرافیکی است که برای مقایسه ۲ توزیع احتمال استفاده می‌شود. نمودار Q-Q با ترسیم quantile های ۲ توزیع در مقابل یکدیگر کار می‌کند. به طور خاص، محور x نشان دهنده quartile های توزیع اول و محور y نشان دهنده quartile های توزیع دوم است. اگر ۲ توزیع یکسان باشند، نقاط روی نمودار Q-Q باید در امتداد یک خط مستقیم قرار گیرند. اگر توزیع ها متفاوت باشند، نقاط روی نمودار Q-Q در نقطه ای از خط مستقیم منحرف می‌شوند.

ب) ۳ نوع از نمودار Q-Q عبارت اند از:

۱- S-shaped : این نشان می‌دهد که ۲ توزیع شکل های متفاوت دارند، اما یکی از دم ها از نظر شکل مشابه با توزیع دیگر است. نقاط روی نمودار Q-Q از خط مورب در ابتدا و یا انتهای طرح منحرف می‌شوند اما سپس از خط وسط پیروی می‌کنند.

۲- concave (مقعر) : در نمودار مقعر Q-Q، نمودار به سمت بالا منحنی می‌شود که نشان می‌دهد quantile های یک توزیع به طور پیوسته بالاتر از quantile های توزیع دیگر است. این شکل نشان می‌دهد که یک توزیع دارای میانه بالاتری نسبت به دیگری است.

۳- curved (منحنی) : این نمودار نشان می‌دهد که ۲ توزیع دارای اشکال قابل توجه متفاوت هستند. نقاط روی نمودار Q-Q از خط مورب به شکل منحنی منحرف می‌شوند که نشان می‌دهد که quantile های ۲ توزیع نامتناسب هستند.

می‌توان موارد زیر را در مورد توزیع های آماری اولیه از روی Q-Q plot استنتاج کرد:

۱- تعیین میزان تفاوت بین ۲ توزیع

۲- تشخیص توزیع های دارای شکل های خاص و غیرمعمول

۳- تشخیص تغییرات شکل توزیع در طول زمان

۴- مقایسه شکل توزیع با توزیع های معروف مانند توزیع نرمال

همچنین از نمودار Q-Q plot می‌توانیم استنباط کنیم که آیا داده ها از توزیع خاصی پیروی می‌کنند یا خیر. اگر نقاط روی نمودار نزدیک به خط مورب باشند، نشان می‌دهد که داده ها از توزیع مربوطه پیروی می‌کنند. با این حال، اگر نقاط به طور قابل توجهی از خط مورب انحراف داشته باشند، نشان می‌دهد که داده ها از توزیع مربوطه پیروی نمی‌کنند. با مقایسه نمودارهای Q-Q ۲ مجموعه داده می‌توانیم ارزیابی کنیم که آیا آنها توزیع های مشابهی دارند یا خیر.

امیربهام - ۹۸۳۱۱۳۳

۱۰-

الف) در این تکنیک تبدیل خطی روی داده ها به اصطلاح انجام می شود و داده را بین ۲ مقدار ۰ و ۱ می برد و فرمول آن به صورت روبرو است

$$norm = \frac{x - min(x)}{max(x) - min(x)}$$

همچنین اگر بخواهیم از محدوده دلخواهی به جای محدوده استاندارد [۰,۱] داده ها را تنظیم کنیم، می توانیم از فرمول زیر استفاده کنیم:

$$x\text{-}norm = (x - min(x)) \frac{(new_max - new_min)}{max(x) - min(x)} + new_min$$

لازم به ذکر است که فرمول اول پاسخ جواب است و فرمول دوم برای توضیحات تکمیلی بود.

ب) در این روش میانگین و انحراف معیار داده ها را به یک مجموع محاسبه می شود و سپس با استفاده از فرمول زیر مقادیر هر داده به بازه ی مبتنی بر میانگین و انحراف تبدیل می شود و در واقع می توان تفاوت فردی با میانگین ۰ و انحراف معیار ۱ تولید می کند.

$$x\text{-}normalized = \frac{(x - mean(x))}{std(x)}$$

ج) با جابجایی نقاط اعشار مقادیر داده ها نرمال می شوند. برای اجرای این تکنیک هر مقدار داده را بر حداکثر مقدار مطلق تقسیم می کنیم. به عبارت دیگر این تکنیک داده ها را با تقسیم هر مقدار بر توان ۱۰ که حداکثر مقدار مطلق را کمتر از ۱ می کند، قیاس می کند. برای مثال اگر حداکثر مقدار مطلق در مجموعه داده ۱۰۰۰ باشد می توانیم همه مقدار را بر ۱۰۰۰ تقسیم کنیم تا آنها را در محدوده ۱- را کوچک کنیم.

$$x\text{-}norm = \frac{x}{10^k}$$

امیرپیام _ ۹۸۳۱۱۳۳

۱۱ -

الف) برای استخراج معادله نرمال با استفاده از روش مشتق جزئی باقیمانده را با معادله مجموع مجذور خطاها شروع می‌کنیم:

$$S = \sum (y - \hat{y})^2$$

که y مقدار واقعی و $\hat{y}$ مقدار پیش بینی شده و مجموع آن بر روی تمام نقاط داده است.

هدف یافتن ضرایبی است که مجموع مجذور خطاها را به حداقل برساند. این کار را می‌توانیم با گرفتن مشتق جزئی S با توجه به هر ضریب β_i انجام دهیم:

$$\frac{\partial S}{\partial \beta_i} = 2\sum (y - \hat{y})(-x_{ij})$$

جایی که β_i ضریب iام و x_{ij} تامین متغیر پیش بینی کننده است و مجموع آن روی تمام نقاط داده است.

در ادامه با قرار دادن هر مشتق جزئی برابر با صفر، سیستمی از معادلات را به دست می‌آوریم:

$$\sum (y - \hat{y}) x_{1j} = 0$$
$$\sum (y - \hat{y}) x_{2j} = 0$$
$$\sum (y - \hat{y}) x_{3j} = 0$$
$$\vdots$$
$$\sum (y - \hat{y}) x_{kj} = 0$$

که به صورت ماتریسی این معادلات را بازنویسی می‌کنیم:

$$\Rightarrow X^T(y - X\beta) = 0$$

با بسط معادله با ضریب ماتریسی، عبارت روبرو را به دست می‌آوریم:

$$\Rightarrow X^T y - X^T X \beta = 0 \Rightarrow X^T X \beta = X^T y \Rightarrow \beta = (X^T X)^{-1}(X^T y)$$

معادله نرمال

همچنین برای اینکه ثابت کنیم، معادله به دست آمده درست است، می‌توانیم آن را به معادله اصلی S جایگزین کنیم:

$$S = \sum (y - \hat{y})^2 \Rightarrow \sum (y - X\beta)^T (y - X\beta) \Rightarrow (y - X\beta)^T (y - X\beta)$$

$$\Rightarrow y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X \beta$$

امیرهام - ۹۸۳۱۱۳۳

۱۱ـ

الف ـ ادامه)

با گرفتن مشتق ε نسبت به β و صفر کردن آن به دست می‌آید:

$$\Rightarrow \frac{\partial \varepsilon}{\partial \beta} = -2X^T y + 2X^T X\beta = 0$$

با حل کردن β را داریم:

$$\Rightarrow \beta = (X^T X)^{-1}(X^T y)$$

و همانطور که قابل مشاهده است عبارت به دست آمده برابر با معادله نرمال است که به دست آوردیم.

ب)

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 1 & 1 \\ 1 & 3 \\ 1 & 2 \\ 1 & 6 \end{bmatrix} \qquad y = \begin{bmatrix} 5 \\ 6 \\ 3 \\ 6 \\ 3 \\ 10 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 1 & 3 & 2 & 6 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 1 & 1 \\ 1 & 3 \\ 1 & 2 \\ 1 & 6 \end{bmatrix}$$

$$\Rightarrow X^T X = \begin{bmatrix} 6 & 18 \\ 18 & 70 \end{bmatrix}$$

$$X^T y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 1 & 3 & 2 & 6 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 3 \\ 6 \\ 3 \\ 10 \end{bmatrix}$$

$$\Rightarrow X^T y = \begin{bmatrix} 33 \\ 121 \end{bmatrix}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} 6 & 18 \\ 18 & 70 \end{bmatrix}^{-1} \begin{bmatrix} 33 \\ 121 \end{bmatrix} \Rightarrow \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \frac{1}{96} \begin{bmatrix} 70 & -18 \\ -18 & 6 \end{bmatrix} \begin{bmatrix} 33 \\ 121 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \frac{1}{96} \begin{bmatrix} 132 \\ 132 \end{bmatrix} \Rightarrow y = \frac{132}{96} + \frac{132}{96} x \Rightarrow y = \underbrace{1.375}_{1.375} + \underbrace{1.375}_{} x$$

برای محاسبه مقدار خطای برازش از MSE استفاده می‌کنیم:

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 \Rightarrow \frac{1}{6} \times [\underbrace{(5-\ )^2}_{4.125} + \underbrace{(6-\ )^2}_{6.875} + \underbrace{(3-\ )^2}_{2.75} + \underbrace{(6-\ )^2}_{5.0} + \underbrace{(3-\ )^2}_{4.125} + \underbrace{(10-\ )^2}_{9.625}] \Rightarrow \underline{\underline{\ \ \ \

امیرحسام - ۹۸۳۱۱۳۳

۱۱-

ج)

$$x = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 4 & 16 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 2 & 4 \\ 1 & 6 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 5 \\ 6 \\ 3 \\ 6 \\ 3 \\ 15 \end{bmatrix}$$

$$x^T x = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 1 & 3 & 2 & 6 \\ 4 & 16 & 1 & 9 & 4 & 36 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 1 & 4 & 16 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 2 & 4 \\ 1 & 6 & 36 \end{bmatrix}_{6\times 3}$$

$$\Rightarrow x^T x = \begin{bmatrix} 6 & 18 & 70 \\ 18 & 70 & 324 \\ 70 & 324 & 1666 \end{bmatrix}$$

$$\Rightarrow x^T y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 1 & 3 & 2 & 6 \\ 4 & 16 & 1 & 9 & 4 & 36 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 3 \\ 6 \\ 3 \\ 15 \end{bmatrix}$$

$$\Rightarrow x^T y = \begin{bmatrix} 33 \\ 121 \\ 545 \end{bmatrix}$$

$$\Rightarrow B = (x^T x)^{-1} x^T y \Rightarrow \begin{bmatrix} \frac{2911}{890} & \frac{-1827}{890} & \frac{233}{890} \\ -\frac{1827}{890} & \frac{637}{445} & \frac{-171}{890} \\ \frac{233}{890} & \frac{-171}{890} & \frac{12}{445} \end{bmatrix} \begin{bmatrix} 33 \\ 121 \\ 545 \end{bmatrix}$$

$$\Rightarrow B = \begin{bmatrix} \frac{1981}{890} \\ \frac{334}{445} \\ \frac{39}{445} \end{bmatrix} \Rightarrow y = \frac{1981}{890} x^0 + \frac{334}{445} x + \frac{39}{445}$$

برای محاسبه مقدار خطای برازش از MSE استفاده می‌کنیم:

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 \Rightarrow MSE = \frac{1}{6} [(5 - 10.48)^2 + (6 - 38.07)^2 + (3 - 30.06)^2 + (6 - 22.37)^2 + (3 - 10.4)^2 + (15 - 84.07)^2]$$

$$\Rightarrow MSE = \frac{1}{6} [30.3 + 1069.29 + 0.0036 + 267.97 + 54.76 + 5830.07]$$

$$\Rightarrow MSE = 1167.52$$

امیر بهنام – ۹۸۳۱۱۳۳ – تمرین اول داده کاوی – قسمت پیاده سازی

در ابتدا کتاب خانه های مورد نیاز را import می کنیم و سپس فایل Csv را به صورت زیر بار گذاری می کنیم :

```
df = pd.read_csv(r'C:\Users\dr. Behnam\Documents\Penguins\penguins.csv' , names=['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm'
```

سوال اول:

با استفاده از isna().sum() تعداد ردیف های که دارای NaN هستند را برای هر ستون پیدا می کنیم.

```
import pandas as pd
import matplotlib
from matplotlib import colors
from matplotlib import pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import PCA

df = pd.read_csv(r'C:\Users\dr. Behnam\Documents\Penguins\penguins.csv' , names=['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm'
print(df.isna().sum())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python +

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\dr. Behnam> & "C:/Users/dr. Behnam/AppData/Local/Programs/Python/Python38-32/python.exe" "c:/Users/dr. Behnam/Penguin.py"
island              0
bill _length_mm     2
bill_depth_mm       2
lipper_length_mm    2
body_mass_g         2
sex                11
species             0
```

در ستون island و species صفر , در ستون bill_length_mm , bill_depth_mm, body_mass_g و lipper_length_mm دو و در ستون sex یازده مقدار NaN وجود دارد.

سوال دوم:

ابتدا با استفاده از دستور df.shape[0] تعداد سطر ها را قبل از حذف مشخص می کنیم. سپس با استفاده از دستور df.dropna(how ='any') سطر های که حاوی مقدار NaN هستند را حذف می کنیم و در نهایت با استفاده از دستور df.shape[0] تعداد سطر ها را بعد از حذف را مشخص می کنیم.

تعداد سطر ها قبل از حذف برابر ۳۴۴ و بعد از حذف برابر ۳۳۳ است.

و همانطور که در شکل زیر مشخص شده است تمامی مقادیر NaN حذف شده اند.

```
df = pd.read_csv(r'C:\Users\dr. Behnam\Documents\Penguins\penguins.csv' , names=['island', 'bill _length_mm', 'b
#print(df.isna().sum())

number_of_rows_before_deletion = df.shape[0]
df = df.dropna(how ='any')
number_of_rows_after_deletion = df.shape[0]
print(df.isna().sum())
print('Number of rows before deletion:', number_of_rows_before_deletion)
print('Number of rows after deletion:', number_of_rows_after_deletion)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python Debug Console

```
island             0
bill _length_mm    0
bill_depth_mm      0
lipper_length_mm   0
body_mass_g        0
sex                0
species            0
dtype: int64
Number of rows before deletion: 344
Number of rows after deletion: 333
PS C:\Users\dr. Behnam>
```

سوال ۳ :

در ابتدا ۲ متغیر به نام های number_columns و string_columns تعریف می کنیم. سپس با استفاده از SimpleImputer مقادیر عددی را با ‘mean’ و مقادیر غیر عددی را با ‘most_frequnt’ جایگذاری می کنیم. در ادامه با استفاده از دستور fit_transform این جایگذاری را انجام می دهیم. سپس با استفاده از df.isnull().sum() چک می کنیم که مقادیر NaN برای هر سطر صفر باشد و در نهایت مجموعه داده را چاپ می کنیم.

```
21  string_columns = df.select_dtypes(include=['object']).columns.tolist()
22  number_columns = df.select_dtypes(include=[pd.np.number]).columns.tolist()
23  imputer = SimpleImputer(strategy='mean')
24  imputer_str = SimpleImputer(strategy='most_frequent')
25  df[string_columns] = imputer_str.fit_transform(df[string_columns])
26  df[number_columns] = imputer.fit_transform(df[number_columns])
27  print(df.isnull().sum())
28  print(df)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
island             0
bill _length_mm    0
bill_depth_mm      0
lipper_length_mm   0
body_mass_g        0
sex                0
species            0
dtype: int64
        island  bill _length_mm  bill_depth_mm  lipper_length_mm  body_mass_g      sex    species
0    Torgersen         39.10000       18.70000        181.000000   3750.000000     male     Adelie
1    Torgersen         39.50000       17.40000        186.000000   3800.000000  female     Adelie
2    Torgersen         40.30000       18.00000        195.000000   3250.000000  female     Adelie
3    Torgersen         43.92193       17.15117        200.915205   4201.754386     male     Adelie
4    Torgersen         36.70000       19.30000        193.000000   3450.000000  female     Adelie
..         ...              ...            ...               ...           ...     ...        ...
339      Dream         55.80000       19.80000        207.000000   4000.000000     male  Chinstrap
340      Dream         43.50000       18.10000        202.000000   3400.000000  female  Chinstrap
341      Dream         49.60000       18.20000        193.000000   3775.000000     male  Chinstrap
342      Dream         50.80000       19.00000        210.000000   4100.000000     male  Chinstrap
343      Dream         50.20000       18.70000        198.000000   3775.000000  female  Chinstrap

[344 rows x 7 columns]
PS C:\Users\dr. Behnam>
```

به عنوان نمونه در سطر ۴ تعدادی داده ی NaN وجود داشت که با میانگین و متداول ترین مقدار جایگزین شد و در شکل فوق قابل مشاهده است.

سوال ۴ :

در ابتدا ۳ ستونی که به صورت String هستند (sex , island , species) را مشخص می کنیم و برای هر کدام یک LabelEncoder تعریف می کنیم و با استفاده از fit_transform عملیات را روی هر ستون انجام می دهیم و در نهایت آنها را چاپ می کنیم و در شکل زیر نتیجه را مشاهده می کنیم:

```python
string_columns_for_label_encoding= ['sex', 'island', 'species']
le_sex = LabelEncoder()
le_island = LabelEncoder()
le_species = LabelEncoder()
df['sex'] = le_sex.fit_transform(df['sex'])
df['island'] = le_island.fit_transform(df['island'])
df['species'] = le_species.fit_transform(df['species'])
print(df[string_columns_for_label_encoding])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
ython.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launch
c:\Users\dr. Behnam\Penguin.py:22: FutureWarning: The pandas.np module is deprecat
umpy directly instead.
  number_columns = df.select_dtypes(include=[pd.np.number]).columns.tolist()
     sex  island  species
0      1       2        0
1      0       2        0
2      0       2        0
3      1       2        0
4      0       2        0
..   ...     ...      ...
339    1       1        1
340    0       1        1
341    1       1        1
342    1       1        1
343    0       1        1

[344 rows x 3 columns]
PS C:\Users\dr. Behnam>
```

سوال ۵ :

افزایش داده ها تکنیکی برای ایجاد نمونه های جدید داده های افزوده شده از داده های اصلی با اعمال تبدیل های مختلف است. هدف از افزایش داده ها افزایش داده های موجود و کاهش اضافه برازش با قرار دادن مدل در معرض ورودی های متنوع تر است. افزایش داده ها با اعمال تبدیل هایی مانند چرخش، چرخش، برش، زوم، تغییر روشنایی، کنتراست، اشباع، اضافه کردن نویز و غیره به داده های اصلی کار می کند. به عنوان مثال، چرخش یک تصویر به اندازه چند درجه، چرخاندن آن به صورت افقی یا عمودی، یا برش بخشی از تصویر برای ایجاد تصویر جدید. این تبدیل‌ها اساساً تصاویر جدیدی ایجاد می‌کنند که تغییرات دنیای واقعی مجموعه داده اصلی را شبیه‌سازی می‌کنند.

افزایش داده به طور کلی در مرحله آموزش یک مدل یادگیری ماشینی که در مجموعه آموزشی اعمال می شود. داده های افزوده شده برای آموزش مدل یادگیری ماشینی، بهبود دقت آن، و قوی تر کردن آن در برابر تغییرات در توزیع داده های ورودی استفاده می شود. دلیل عدم استفاده از افزایش داده در مجموعه آزمایشی این است که می‌خواهیم عملکرد مدل یادگیری ماشین را روی داده‌های اصلی بدون هیچ گونه اختلالی ارزیابی کنیم. اگر

تقویت داده‌ها را در مجموعه تست اعمال کنیم، تصاویر جدیدی را معرفی می‌کنیم که مدل قبلاً هرگز ندیده است و اندازه‌گیری عملکرد واقعی مدل بر روی داده‌های تست را دشوار می‌کند.

به طور خلاصه، افزایش داده‌ها تکنیکی برای ایجاد نمونه‌های داده‌های افزوده‌شده جدید با اعمال تبدیل‌های مختلف بر روی داده‌های اصلی است تا مدل را در معرض ورودی‌های متنوع‌تر قرار دهد، کاهش بیش‌برازش و افزایش داده‌های موجود. معمولاً در مرحله آموزش استفاده می شود و برای ارزیابی عملکرد واقعی مدل در مجموعه تست اعمال نمی شود.

سوال ۶:

Upsampling:

در روش Upsampling، تعداد نمونه‌های کلاس کمتر با تکثیر نمونه‌های این کلاس افزایش داده شده و در نهایت تعداد نمونه‌های تمامی کلاس‌ها برابر می‌شود. با این کار، جانب‌داری کلاس‌ها حذف شده و همچنین بهبود دقت مدل ممکن است دست یابد.

Downsampling:

در روش downsampling، تعداد نمونه‌های کلاس بیشتر با حذف یک یا چند نمونه از این کلاس کاهش داده می‌شود تا تعداد کلاس‌ها برابر شود. با این کار، جانب‌داری کلاس‌ها کاهش می‌یابد اما ممکن است برای کاهش حجم نمونه‌های داده برای الگوریتم‌ها مناسب باشد.

Resampling Combination (یک روش ترکیبی):

روش دیگری که میتوانیم برای اصلاح غیرتعادلی داده ها مورد استفاده قرار دهیم به این صورت است که ابتدا با روش upsampling نمونه های کلاس کمتر را تکثیر کرده و سپس از نتیجه حاصل، با استفاده از روش downsampling حجم داده ها را کاهش دهیم. به این ترتیب، جانب‌داری کلاس کمتر کاهش می‌یابد و در ضمن، حجم نمونه‌های داده هم در حد معقولی باقی می‌ماند.

سوال ۷:

SMOTE-TOMEK و SMOTE-ENN دو تکنیکی هستند که برای مدیریت مجموعه داده های نامتعادل استفاده می شوند. این تکنیک ها عملیات خاصی را برای افزایش نقاط داده کلاس اقلیت یا نمونه برداری از نقاط داده کلاس اکثریت برای متعادل کردن مجموعه داده اعمال می کنند.

SMOTE-TOMEK:

در این رویکرد، SMOTE (تکنیک نمونه برداری بیش از حد اقلیت مصنوعی) با پیوندهای TOMEK ترکیب می شود. پیوندهای TOMEK به شناسایی نقاطی که از نظر طراحی ضعیف هستند و در نزدیکی یک مرز بین طبقاتی قرار دارند کمک می کند. نقاط داده ای را که در نزدیکی مرزهای کلاس قرار دارند شناسایی می کند و آنها را حذف می کند. پیوندهای TOMEK نقاط متضاد را که نمونه هایی هستند که به هر دو کلاس اکثریت و اقلیت تعلق دارند حذف می کنند و مجموعه ای از پیوندهای tomek را ایجاد می کنند. سپس SMOTE برای نمونه برداری بیش از حد از نقاط داده کلاس اقلیت استفاده می شود. به این ترتیب، تکنیک SMOTE به تولید نقاط داده مصنوعی جدید با درون یابی بین نقاط داده ورودی کمک می کند.

SMOTE-ENN:

در این رویکرد، SMOTE با ویرایش نزدیکترین همسایه (ENN) ترکیب می شود. ویرایش نزدیکترین همسایه با حذف نقاط داده اضافی به کاهش نویز و تمیز کردن داده های آموزشی کمک می کند. سپس الگوریتم SMOTE برای افزایش نقاط داده اقلیت اعمال می شود. به این ترتیب، SMOTE با ایجاد نمونه های مصنوعی مشابه در کلاس اقلیت، داده ها را بیش از حد نمونه برداری می کند. قانون نزدیکترین همسایه ویرایش شده نقاطی را از کلاس اقلیت حذف می کند که توسط نزدیکترین همسایه ها به اشتباه طبقه بندی شده اند.

نقطه مشترک بین این دو روش این است که هر دوی آنها با تولید نمونه های اضافی از کلاس اقلیت با استفاده از روش SMOTE و با حذف نمونه های اضافی در هر دو کلاس اکثریت و اقلیت، مشکل مجموعه داده های نامتعادل را حل می کنند. همچنین، هر دو روش به بهبود دقت مدل‌های یادگیری ماشینی با افزایش مقدار داده‌های موجود برای دسته نادر در مجموعه داده ما کمک می‌کنند.

سوال ۸:

در ابتدا تعداد سطر ها را با استفاده از دستور df.shape[0] همانند سوال دوم چهاپ می کنیم. سپس تعداد سطر هایی که شامل Chinstrap (می توانستیم از ۲ ژن پنگوئن دیگر برای این کار استفاده کنیم و این انتخاب دلخواه بوده است) هستند را در داخل یک لیست ذخیره می کنیم و ۹۰٪ آنها را حذف می کنیم و تعداد سطر ها را پس از حذف چاپ می کنیم. در ادامه feature ها و label ها را از هم جدا می کنیم و در نهایت هم با استفاده از smotetomekو smoteenn غیر متعادل بودن دیتاست را هندل می کنیم. در شکل زیر نتیجه را مشاهده می کنیم:

```
print(f"Number of Rows Before The Process Of Removing Data: {df.shape[0]}")
# I Selected Equal To 1 Instead Of Chinstrap Because In The Previous Step We Label Encoded Each Class And Chinstrap Wa
Rows_With_Chinstrap = df[df.species == 1].index.tolist()
Remove_Rows_With_Chinstrap = Rows_With_Chinstrap[:int(0.9 * len(Rows_With_Chinstrap))]
df.drop(Remove_Rows_With_Chinstrap , inplace=True)
print(f"Number of Rows After Removing 90% Of Rows Which Contains Chinstrap: {df.shape[0]}")
X = df.drop('species', axis=1)
y = df['species']

smote_tomek = SMOTETomek(sampling_strategy='auto')
X_resampled_1, y_resampled_1 = smote_tomek.fit_resample(X, y)
print(f"How The Csv File And The Data Looks After Handling And Balancing The Data With SMOTETomek: {X_resampled_1.shap

smote_enn = SMOTEENN(sampling_strategy='auto')
X_resampled_2, y_resampled_2 = smote_enn.fit_resample(X, y)
print(f"How The Csv File And The Data Looks After Handling And Balancing The Data With SMOTEENN:: {X_resampled_2.shape
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER                                   Python Debug Console

\.vscode\extensions\ms-python.python-2022.16.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '65300' '--' 'c:\Users\dr.
entation.py'
c:\Users\dr. Behnam\DataAugmentation.py:24: FutureWarning: The pandas.np module is deprecated and will be removed from pandas in a futur
t numpy directly instead.
  number_columns = df.select_dtypes(include=[pd.np.number]).columns.tolist()
Number of Rows Before The Process Of Removing Data: 344
Number of Rows After Removing 90% Of Rows Which Contains Chinstrap: 283
How The Csv File And The Data Looks After Handling And Balancing The Data With SMOTETomek: (450, 6)
How The Csv File And The Data Looks After Handling And Balancing The Data With SMOTEENN:: (358, 6)
```

تعداد سطر های Chinstrap در داخل فایل ۶۸ تا بود که ۹۰٪ آن برابر ۶۲.۱ می شود که ۶۲-۳۴۴ برابر با ۲۸۳ می شود. در ادامه هم ۲ تکنیک ذکر شده مشاهده می شوند که داده ها را افزایش و هندل کرده اند.

سوال ۹:

در ابتدا یک Scaler می سازیم , سپس با استفاده از دستور get. ستون های می گیریم و آنها را fit می کنیم , در ادامه مقدار واریانس و میانگین هر ستون را قبل از نرمالسازی چاپ می کنیم و سپس داده ها را نرمال سازی می کنیم و در نهایت میانگین و واریانس داده ها بعد از نرمال سازی را به دست می آوریم. این مراحل در شکل زیر قابله مشاهده است:

```python
scaler = StandardScaler()
scaler.fit(df.get(['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm', 'body_mass_g ' , 'sex']))

print("Variance Before " + str(scaler.scale_))
print("Mean Before :" + str(scaler.mean_))

scaled_data = scaler.transform(df.get(['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm', 'body_m
scaler.fit(scaled_data)

print("Variance After  :" + str(scaler.scale_))
print("Mean After  :" + str(scaler.mean_))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python Debug Console

```
c:\Users\dr. Behnam\Penguin.py:22: FutureWarning: The pandas.np module is deprecated and will be removed from pandas in a future v
umpy directly instead.
  number_columns = df.select_dtypes(include=[pd.np.number]).columns.tolist()
Variance Before [7.25137759e-01 5.43572526e+00 1.96616328e+00 1.40002638e+01
 7.98449983e+02 4.99585753e-01]
Mean Before :[6.62790698e-01 4.39219298e+01 1.71511696e+01 2.00915205e+02
 4.20175439e+03 5.20348837e-01]
Variance After  :[1. 1. 1. 1. 1. 1.]
Mean After  :[ 8.26212483e-17 -1.15669748e-15  4.13106242e-16 -8.26212483e-16
  8.26212483e-17 -6.45478503e-18]
PS C:\Users\dr. Behnam>
```

سوال ۱۰:

در ابتدا یک PCA می سازیم و تعداد Component ها را وارد می کنیم و سپس feature ها را fit و transform می کنیم و در ادامه feature ها را چاپ می کنیم.

```python
pca = PCA(n_components=6)
pca.fit(df.get(['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm', 'body_mass_g' , 'sex']))
features = pca.transform(df.get(['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm', 'body_mass_g' ,
print(features)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python Debug Consol

```
c:\Users\dr. Behnam\Penguin.py:22: FutureWarning: The pandas.np module is deprecated and will be removed from pandas in a future vers
irectly instead.
  number_columns = df.select_dtypes(include=[pd.np.number]).columns.tolist()
[[-2.26751055  1.26617456 -0.04859122 -0.872059   -0.60096994  0.45236073]
 [-2.07892687 -0.52810927  0.84250778 -0.95544949  0.11518465  0.56089195]
 [-2.14353973 -0.40919839  1.10331118 -0.73317112  0.26448695 -0.35807252]
 ...
 [-0.37997769  1.29443141  0.59806168  0.63524885 -0.73176807  0.0491286 ]
 [ 0.39844116  1.66930133  0.81971706  0.38429929  0.07467784 -0.53864661]
 [-0.53618254  0.00374146  1.47013351  0.89928838  0.56835346  0.0637368 ]]
PS C:\Users\dr. Behnam>
```

از انجایی که می خواهیم کاهش بعد از ۶ به ۳ داشته باشیم تعداد Component ها را به ۳ تغییر می دهیم:

```python
#PCA

pca = PCA(n_components=3)
pca.fit(df.get(['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm', 'body_mass_g' , 'sex']))
features = pca.transform(df.get(['island', 'bill _length_mm', 'bill_depth_mm', 'lipper_length_mm', 'body_mass_g' , 's
print(features)
```

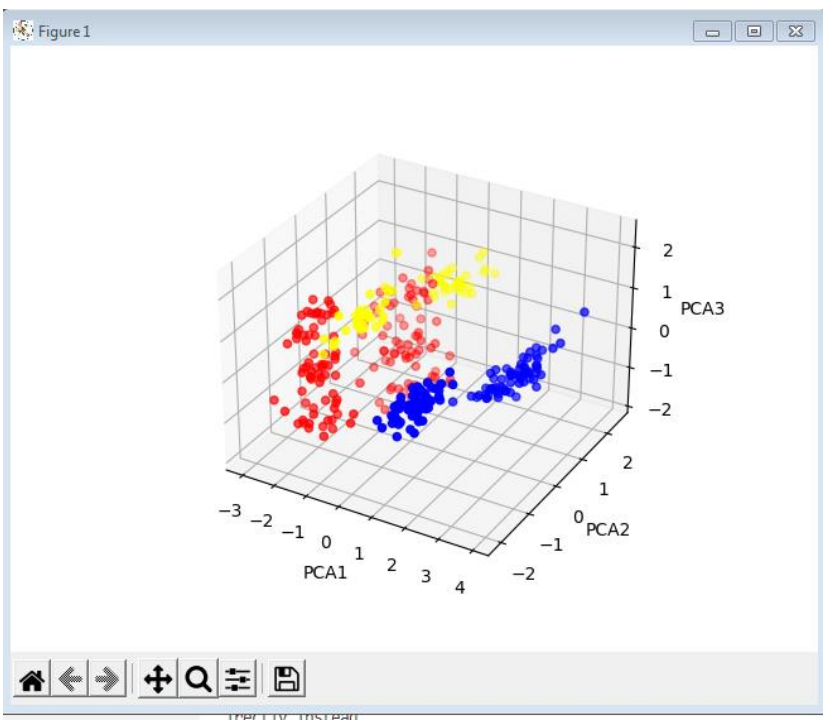
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python Debug Console

```
irectly instead.
  number_columns = df.select_dtypes(include=[pd.np.number]).columns.tolist()
[[-2.26751055  1.26617456 -0.04859122]
 [-2.07892687 -0.52810927  0.84250778]
 [-2.14353973 -0.40919839  1.10331118]
 ...
 [-0.37997769  1.29443141  0.59806168]
 [ 0.39844116  1.66930133  0.81971706]
 [-0.53618254  0.00374146  1.47013351]]
```

سوال ۱۱:

در ابتدا یک figure می سازیم و سپس ۳ رنگ دلخواه را انتخاب می کنیم و در ادامه یک شکل ۳ بعدی را رسم می کنیم , در شکل های زیر کد و خروجی را مشاهده می کنیم:

```python
fig = plt.figure()
colors = ['red', 'yellow', 'blue']
colormap = matplotlib.colors.ListedColormap(colors)
ax = fig.add_subplot(111, projection='3d')
ax.scatter(features[:, 0], features[:, 1], features[:, 2], c=df.species ,cmap=colormap )
ax.set_xlabel('PCA1')
ax.set_ylabel('PCA2')
ax.set_zlabel('PCA3')
plt.show()
```



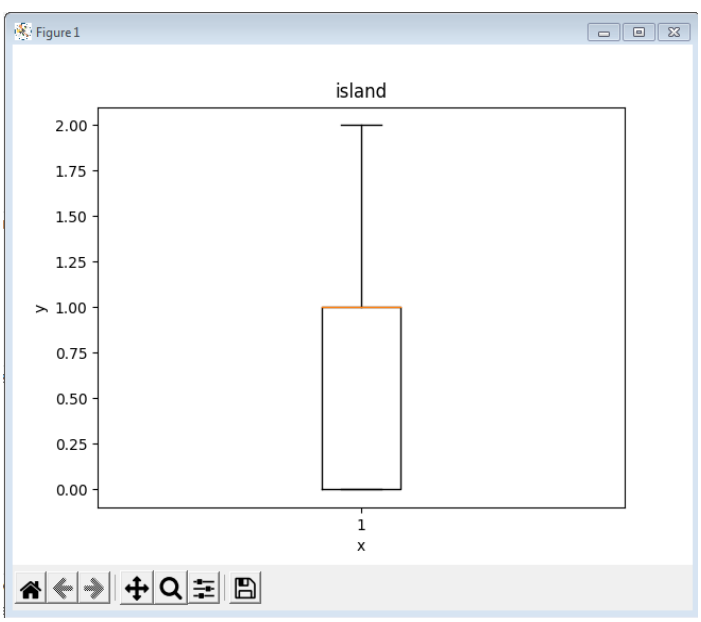
سوال ۱۲ :

برای هر یک از ویژگی ها نمودار Boxplot را رسم می کنیم . این کار را قبل از گام نرمال سازی و در واقع بعد از Label Encoding انجام می دهیم. کد مورد استفاده به صورت زیر است:

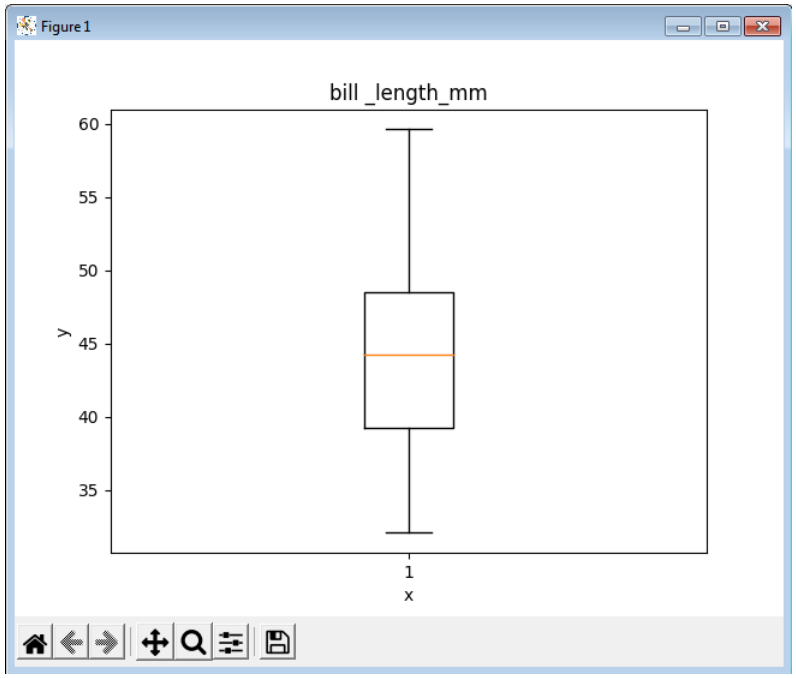
برای اجرا هر کدام از نمودار ها آنها را از حالت کامنت خارج می کنیم.

```
#plt.boxplot(df['island'])
#plt.boxplot(df['bill _length_mm'])
#plt.boxplot(df['bill_depth_mm'])
#plt.boxplot(df['lipper_length_mm'])
#plt.boxplot(df['body_mass_g'])
#plt.boxplot(df['sex'])
plt.boxplot(df['species'])
plt.xlabel('x')
plt.ylabel('y')
#plt.title('island')
#plt.title('bill_lenth_mm')
#plt.title('bill_depth_mm')
#plt.title('lipper_length_mm')
#plt.title('body_mass_g')
#plt.title('sex')
plt.title('species')
plt.show()
```

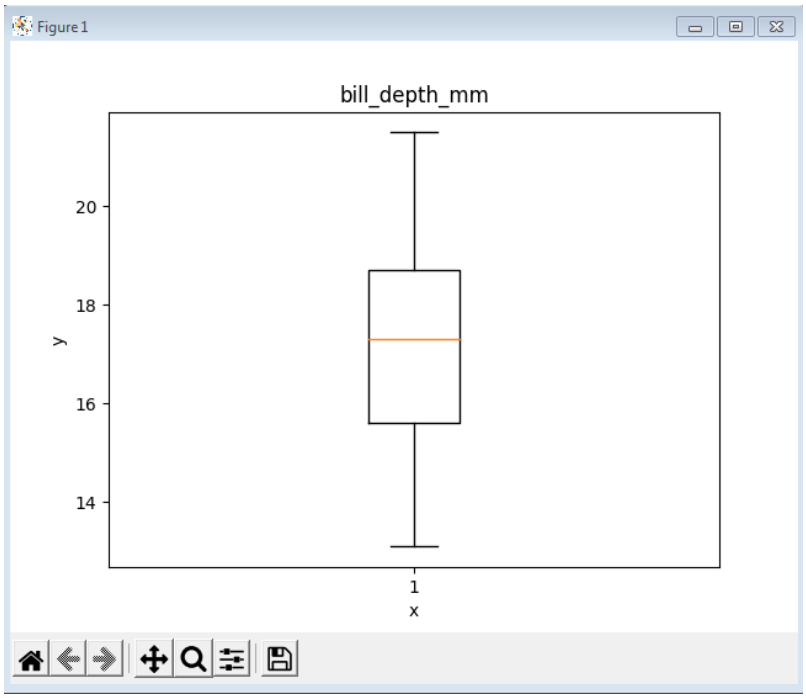
برای Island :



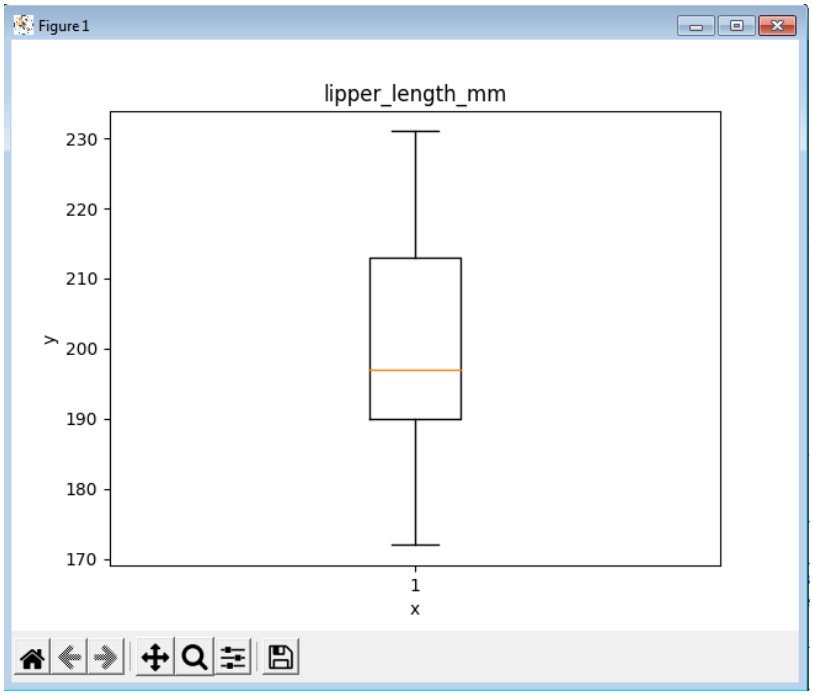
برای bill _length_mm :



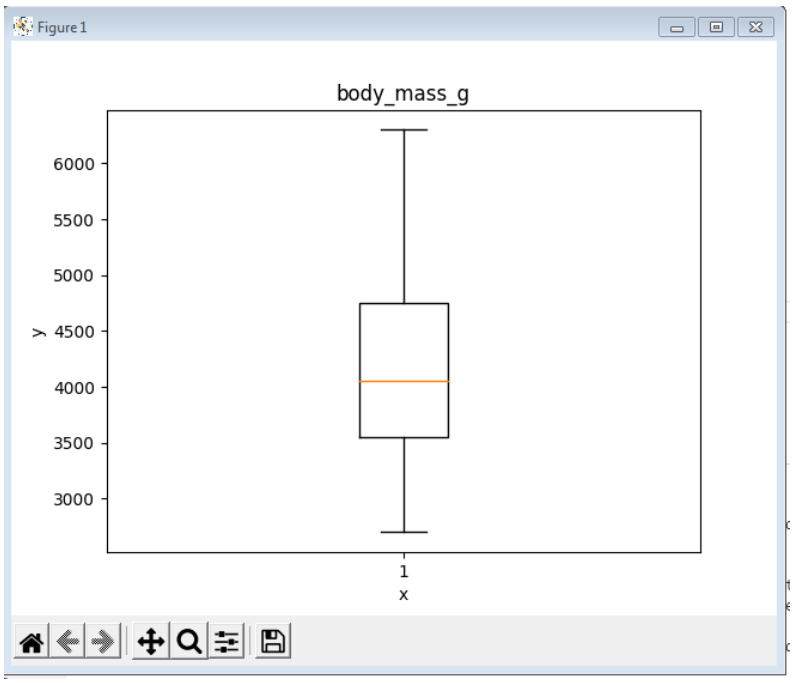
برای bill_depth_mm :



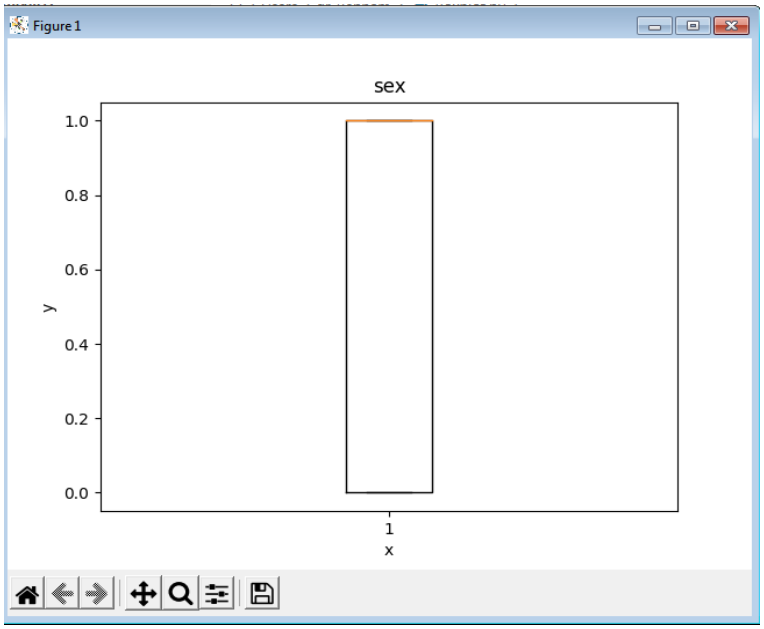
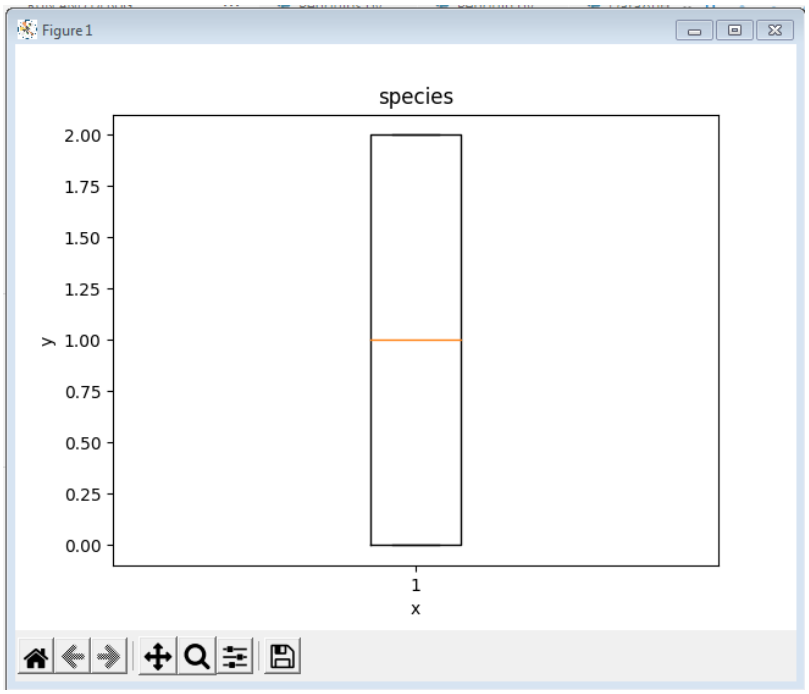
برای lipper_length_mm:



برای body_mass_g :



برای sex :



برای species: