بسمه تعالی

نام و نام خانوادگی : امیرحسین امین نگارشی

درس : پروژه کارشناسی علوم کامپیوتر

موضوع پروژه: پیاده سازی آنومالی با روش های آماری

استاد : دکتر زمینی

**فهرست مطالب**

مقدمه...........................................................................................................................................................2

مفاهیم z-score ........................................................................................................................................7

مفاهیم IQR ..............................................................................................................................................20

مفاهیم GMM ...........................................................................................................................................25

مفاهیم Isolation Forest.......................................................................................................................33

نتایج و نتیجه گیری....................................................................................................................................41

**مقدمه**

من امیرحسین امین نگارشی هستم دانشجوی رشته علوم کامپیوتر ترم 8 و درس پروژه کارشناسی را با دکترزمینی دارم و موضوع پروژه من درباره پیدا کردن آنومالی با روش های آماری است من 5 روش آنومالی را در این پروژه بررسی می کنم این 5 روش عبارت اند از:

1. **روش‌های مبتنی بر توزیع نرمال (Z-Score)**
2. **روش IQR (Interquartile Range)**
3. **روش‌های مبتنی بر مدل‌های آماری (مانند Gaussian Mixture Models)**
4. **روش Isolation Forest (ترکیب آماری و یادگیری ماشین)**

من هرکدام از این موارد در بخشی جداگانه بررسی میکنم؛ در هر بخش ابتدا مفاهیم و تعاریف آن را توضیح می دهم و در نهایت کد آن را زده و خروجیش را میگیریم و نمایش می دهیم.

من این روش های آماری روی دیتاستی بررسی کردم که مربوط به نمرات دانش آموزان یک مدرسه در درس فیزیک است و این دیتاست 4 ستون دارد ، ستون های آن عبارت اند از:

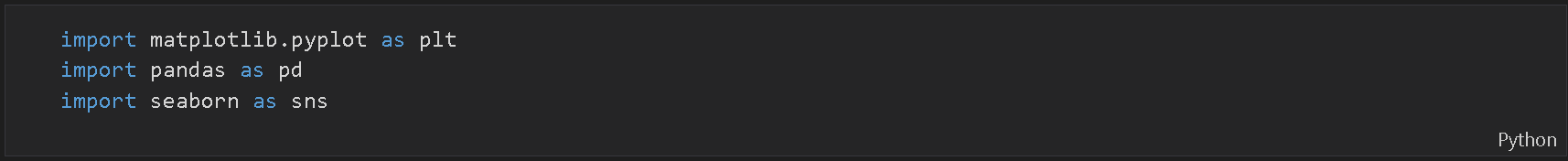
1. اسم آن هاست میشه Name
2. جنسیت آن هاست میشه Gender
3. شماره کلاس آن هاست میشه Class
4. نمره آن ها در این درس میشه Grade

این دیتاست در مجموع دارای 300 sample یا نمونه است و قرار این 4 روش روی این دیتاست آزمایش کرده تا بفهمیم چه قدر دقیق تعداد داده های پرت و خود داده های پرت را پیش بینی میکند.

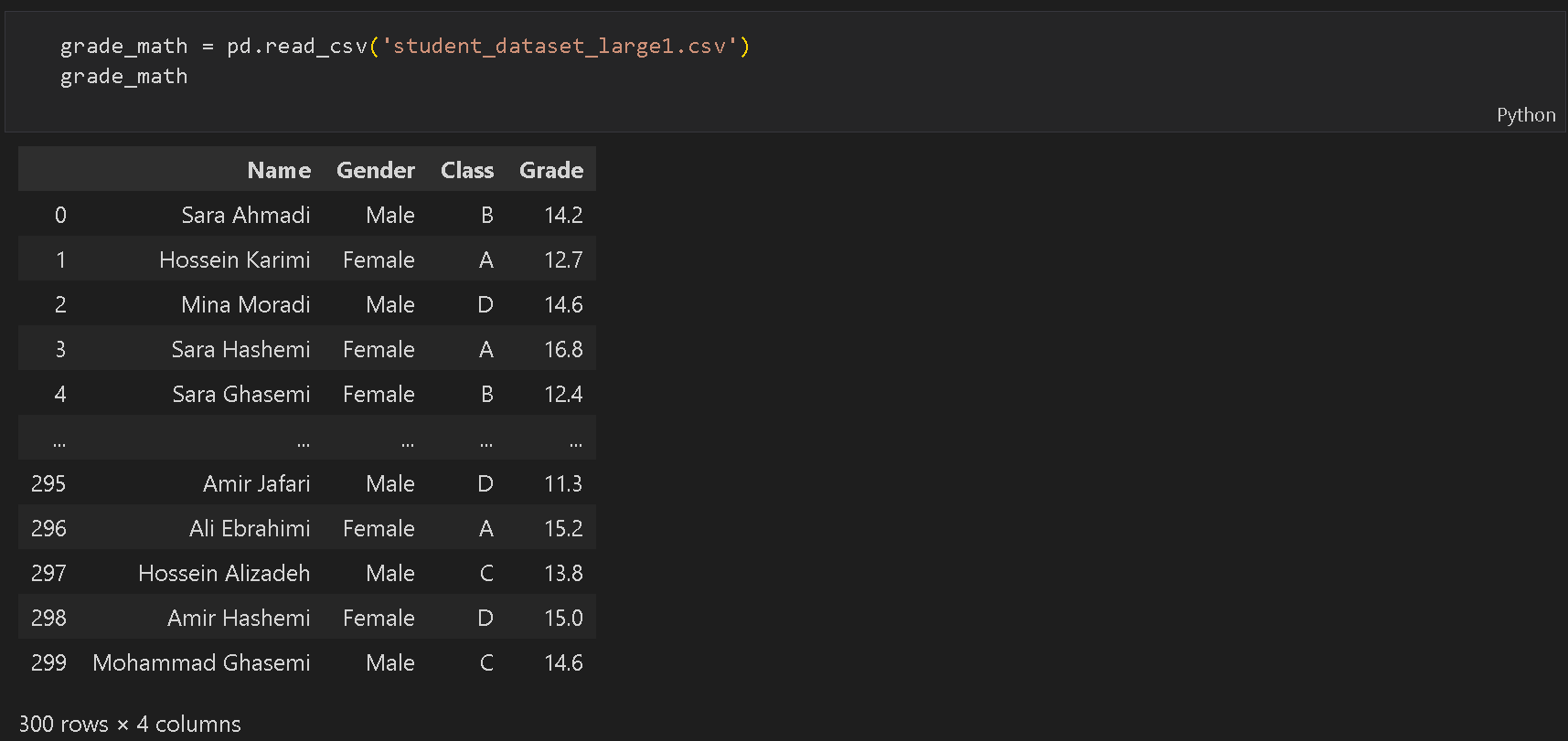
داده های پرت این دیتاست عبارت اند از : نمراتی دانش آموزانی که نمره شان از 9 کمتر شده یا بالای 17 شده اند.

کد های ابتدایی برای نمایش داده های پرت در صفحه بعد می توانید مشاهده کنید.

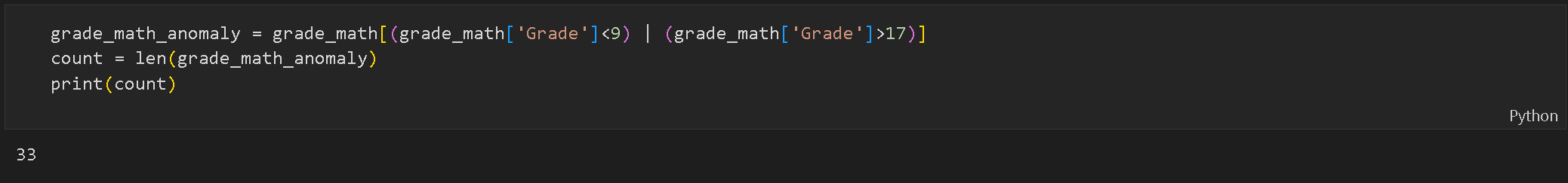
ابتدا library های مورد نیازمون مثل pandas,matplotlib,seaborn را import می کنیم.



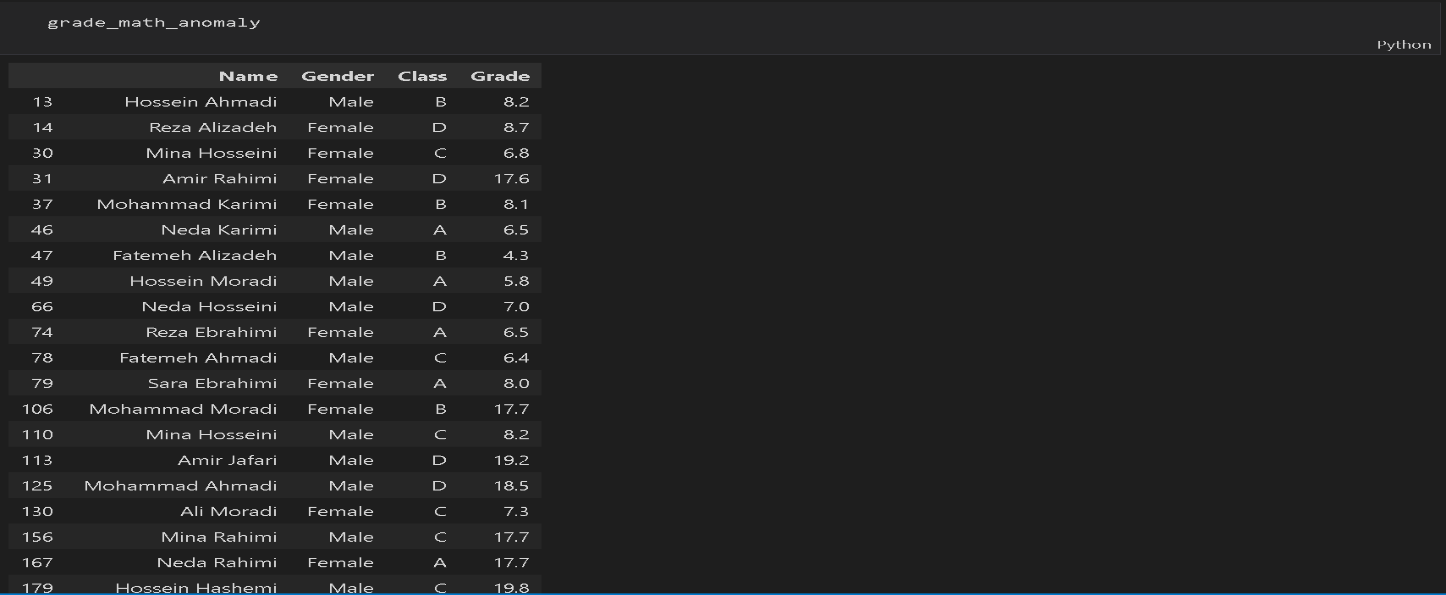
بعد از آن دیتاست را با کمک pandas و متد read\_csv می خوانیم.



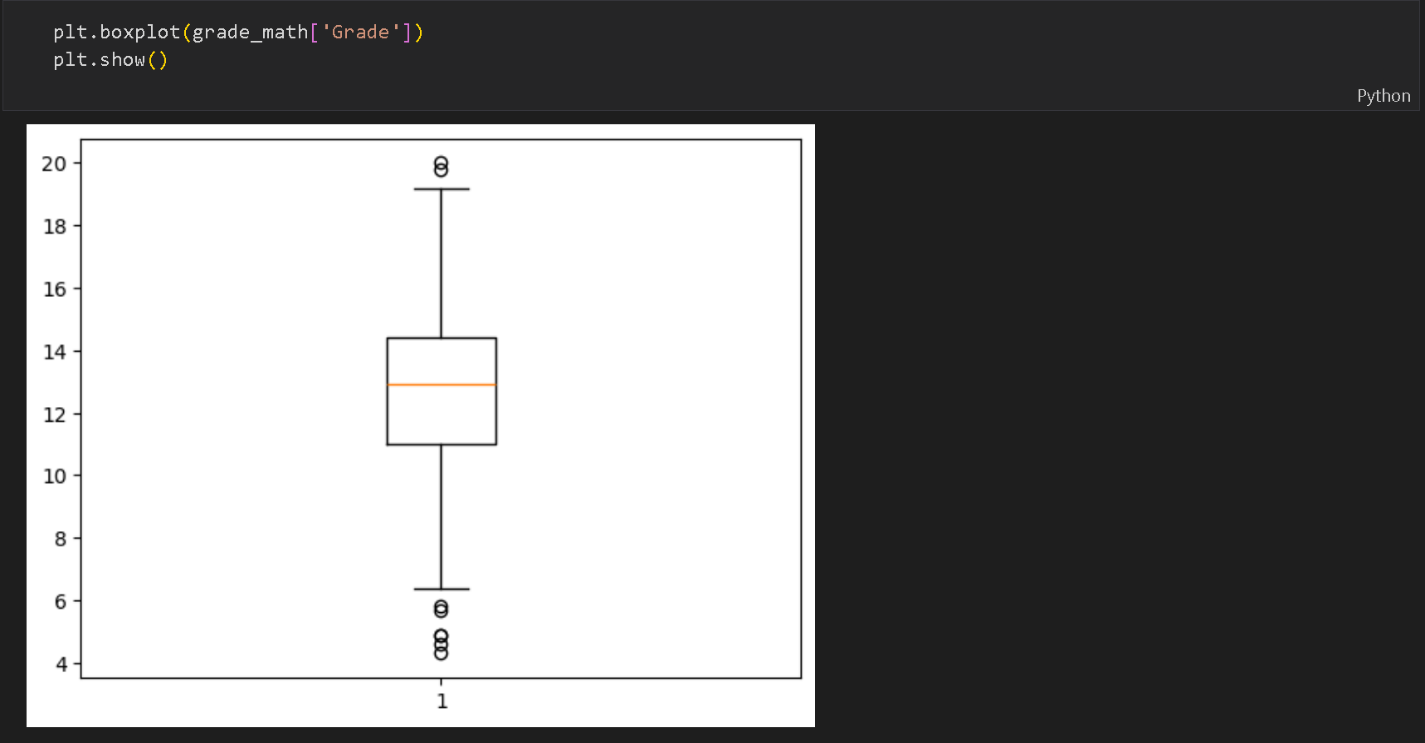
تعداد داده های پرت بدست می آوریم.



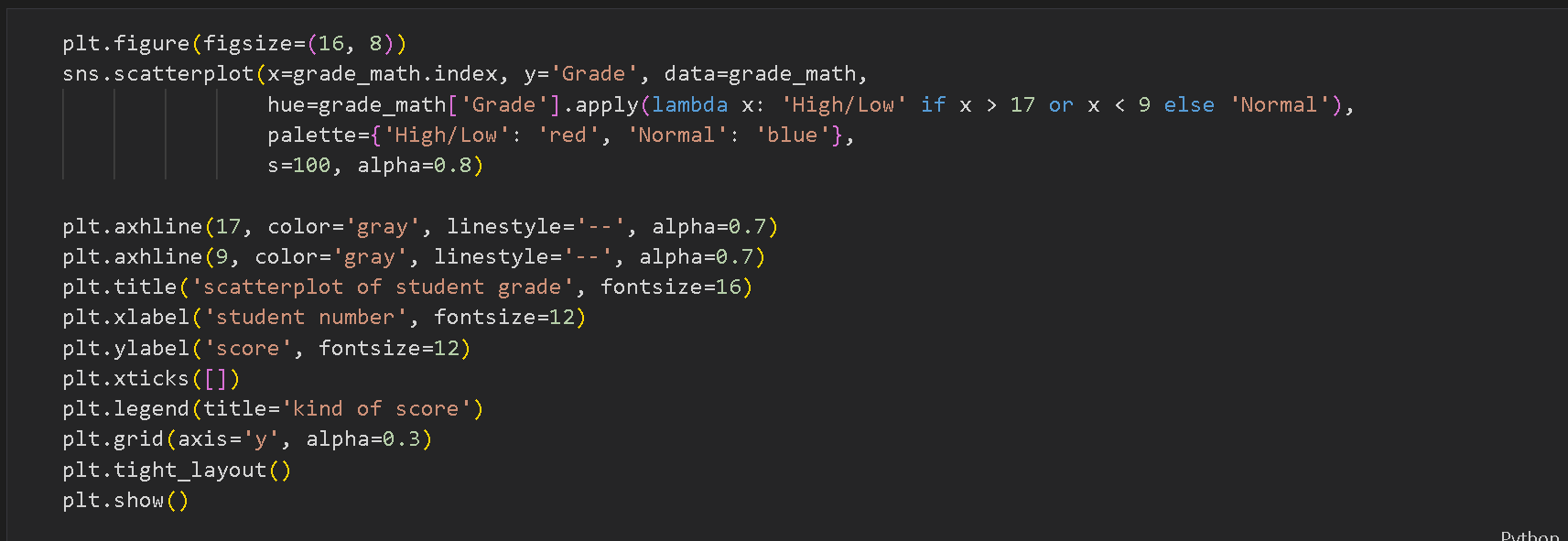
بعد از آن داده های پرت را نمایش می دهیم.



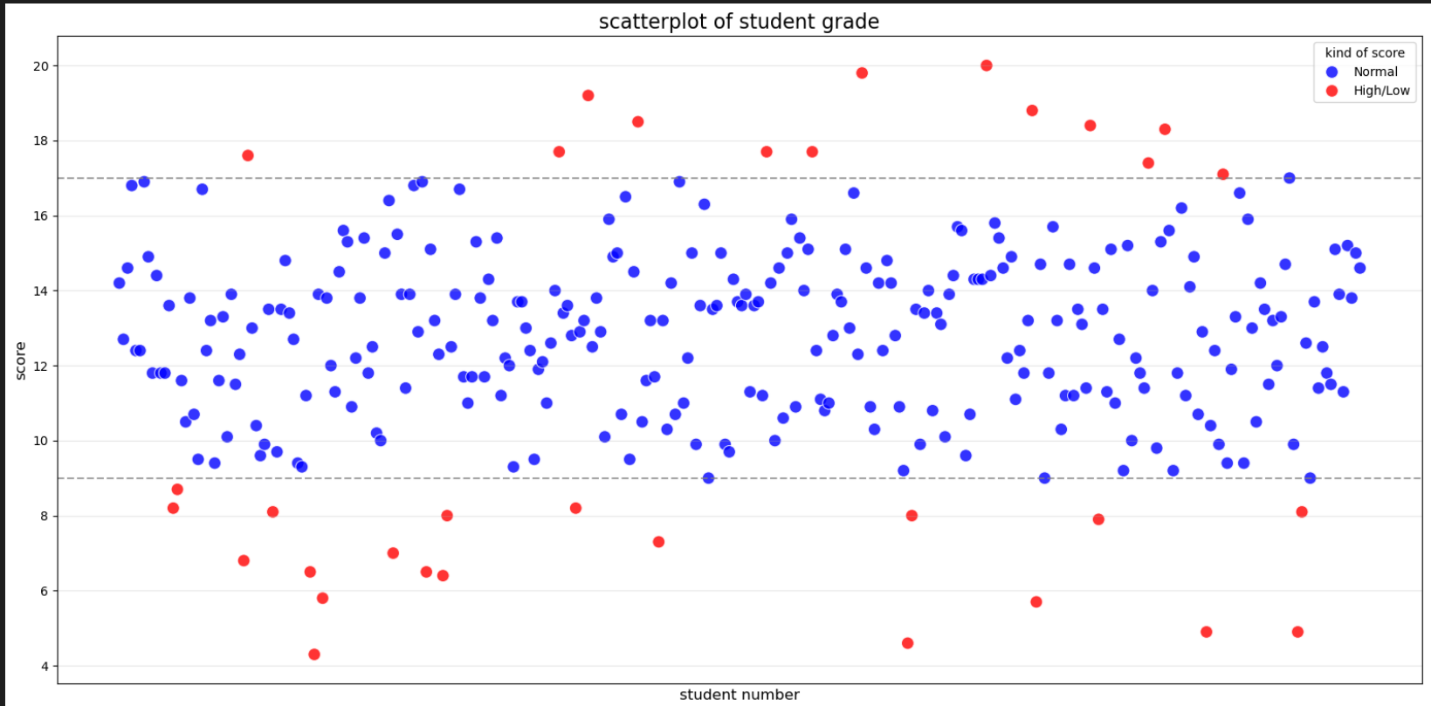
بعد از آن boxplot نمرات را کشیدیم.



و scatterplot با استفاده از seaborn و matplotlib می کشیم.



خروجی این scatterplot را هم می توانید در صفحه بعد مشاهده کنید. که این خروجی به وضوح نشان می دهد کدام داده ها آنومالی و کدام یک از آن ها داده های نرمال هستند.



**مفاهیم z-score**

**Z-score**  (یا نمره استاندارد) یک معیار آماری است که نشان میدهد یک داده چقدر از میانگین جامعه فاصله دارد و این فاصله بر حسب انحراف معیار اندازه گیری میشود. در سرمایه‌گذاری و معامله، Z-score اندازه‌گیری متغیر بودن ابزار است و می‌تواند توسط معامله‌گران برای کمک به تعیین نوسانات استفاده شود.

Z-score یک معیار آماری است که فاصله بین یک نقطه داده و میانگین یک مجموعه داده را کمی می کند. بر حسب انحرافات استاندارد بیان می شود. این نشان می دهد که یک نقطه داده چند انحراف استاندارد از میانگین توزیع دارد.

در [آمار](https://fa.wikipedia.org/wiki/%D8%A2%D9%85%D8%A7%D8%B1)، **نمره استاندارد** مضربی از [انحراف معیار](https://fa.wikipedia.org/wiki/%D8%A7%D9%86%D8%AD%D8%B1%D8%A7%D9%81_%D9%85%D8%B9%DB%8C%D8%A7%D8%B1) است که مقدار آن، نشان می‌دهد یک نمره خام (یعنی مقدار مشاهده شده یا یک نقطه از داده) چقدر بالاتر یا کمتر از [میانگین](https://fa.wikipedia.org/wiki/%D9%85%DB%8C%D8%A7%D9%86%DA%AF%DB%8C%D9%86) مقدار مشاهده یا اندازه‌گیری شده‌است. نمرات خام بالاتر از میانگین دارای نمره استاندارد مثبت و نمرات زیر میانگین دارای نمره استاندارد منفی هستند.

نمره استاندارد، با کسر [میانگین جمعیت](https://fa.wikipedia.org/wiki/%D9%85%DB%8C%D8%A7%D9%86%DA%AF%DB%8C%D9%86) از یک [نمره خام](https://fa.wikipedia.org/wiki/%D8%AF%D8%A7%D8%AF%D9%87_%D8%AE%D8%A7%D9%85) فردی و سپس تقسیم آن بر انحراف معیار [جمعیت](https://fa.wikipedia.org/wiki/%D8%AC%D8%A7%D9%85%D8%B9%D9%87_%D8%A2%D9%85%D8%A7%D8%B1%DB%8C) محاسبه می‌شود. این فرایند تبدیل نمره خام به یک نمره استاندارد، [نرمال سازی](https://fa.wikipedia.org/wiki/%D9%86%D8%B1%D9%85%D8%A7%D9%84_%D8%B3%D8%A7%D8%B2%DB%8C) نامیده می‌شود (اگرچه نمره استاندارد تنها راه نرمال سازی نیست).

نمرات استاندارد را معمولاً z-score می‌ نامند. این دو لفظ ممکن است به جای یک‌دیگر استفاده شوند. اصطلاحات دیگر شامل z-values، نمرات نرمال و متغیرهای استاندارد می‌باشد.

در [کنکور سراسری](https://fa.wikipedia.org/wiki/%DA%A9%D9%86%DA%A9%D9%88%D8%B1)، مبنای محاسبه رتبه داوطلبان، این نمره است.

**فرمول Z-score :**

* مقدار داده مورد نظر **X**:
* میانگین جامعه
* انحراف معیار جامعه

**تفسیر Z-score:**

* داده دقیقاً برابر میانگین است: **Z = 0**
* داده بالاتر از میانگین است (مثبت = بالاتر از میانگین) : **Z > 0**
* داده پایینتر از میانگین است (منفی = پایینتر از میانگین) : **Z < 0**
* مقدار مطلق Z-score () نشاندهنده میزان انحراف از میانگین است.

مثال : مثال:

فرض کنید میانگین نمرات یک کلاس  **70** با انحراف معیار  **10** باشد. اگر دانشآموزی نمره  **85** بگیرد،

Z-score او میشود:

یعنی نمره او  **1.5 انحراف معیار** بالاتر از میانگین است.

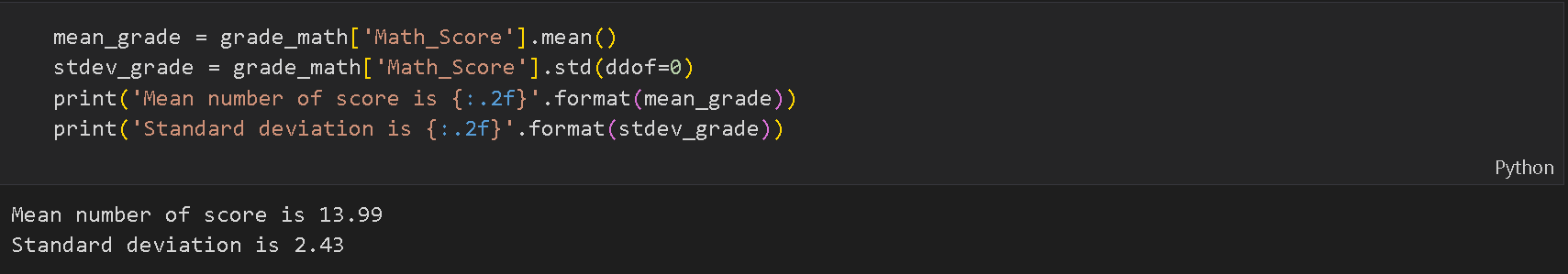
**کاربردهای Z-score:**

1. **مقایسه دادههای مختلف**: مثلاً مقایسه نمرات دو آزمون با میانگین و انحراف معیار متفاوت.
2. **تشخیص دادههای پرت (Outlier)**: معمولاً اگر ∣*z*∣>3، داده پرت در نظر گرفته میشود.
3. **تبدیل به توزیع نرمال استاندارد** (میانگین=0، انحراف معیار=1) برای محاسبات آماری.

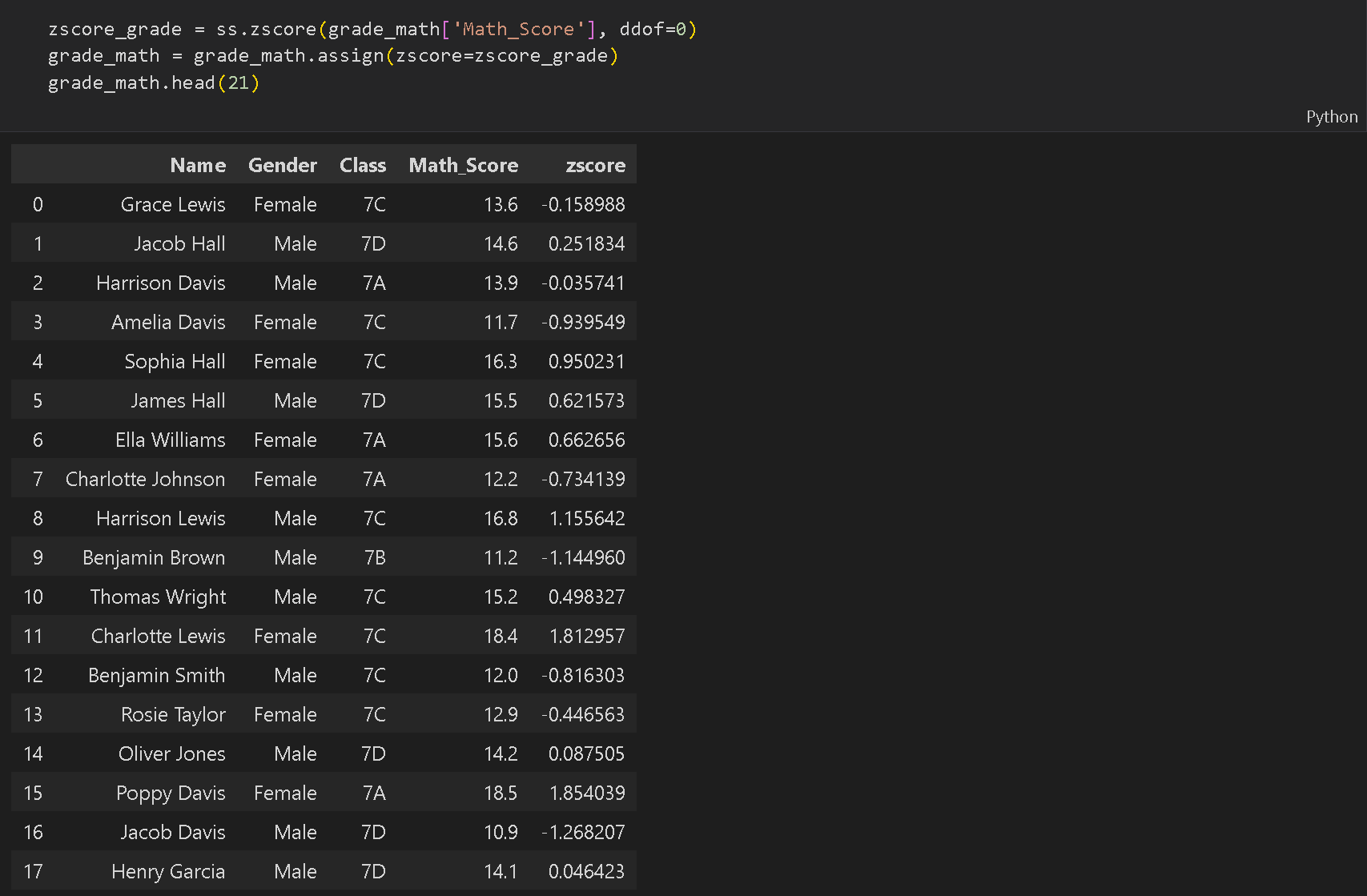
حال به بررسی کد زده برای تشخیص آنومالی احتمالی این دیتاست می پردازیم:

من چون در فایل جداگانه کد آن را زده ام ابتدا library های مدنظرم را import می کنم که تنها تفاوت در import کردن scipy است و این library را برای بتوانیم zscore برای تمام داده ها حساب کنیم import کنیم. بعد از آن هم دیتاستمون را با pandas می خونیم .



بعد از این کار ابتدا میانگین و انحراف معیار نمرات دانش آموزان این دبیرستان را حساب کرده کدو نتایج آن به شرح زیر است:

برای محاسبه zscore هرکدام از نمرات از متد zscore کتابخانه scipy استفاده کرده که ورودی دارد یکی داده هایی که که می خواهی zscore آن ها را پیاده کنیم و یکی ddof که آن را بعدتر توضیح می دهم و سپس میایم ستون zscore نمرات را اضافه کرده و آن را نمایش می دهیم کد و خروجی آن به شرح زیر است:



**نقش ddof در محاسبه انحراف معیار**

انحراف معیار (σ) می‌تواند به دو صورت محاسبه شود:

1. **انحراف معیار جامعه (Population Standard Deviation):**

N تعداد کل داده‌هاست.

1. **انحراف معیار نمونه (Sample Standard Deviation):**

xˉمیانگین نمونه و ddofتعداد درجات آزادی کسر شده از Nاست.

پارامتر ddof (Delta Degrees of Freedom)مشخص می‌کند که چه تعداد درجات آزادی از تعداد داده‌ها (N) کسر شود. به طور پیش‌فرض در scipy.stats.zscore مقدار ddof=0است، یعنی انحراف معیار جامعه محاسبه می‌شود.

**توضیح ddofدر تابع zscore**

در تابع scipy.stats.zscore، پارامتر ddofبه شما امکان می‌دهد نوع انحراف معیار (جامعه یا نمونه) را مشخص کنید:

* **ddof=0 (پیش فرض):** انحراف معیار جامعه محاسبه می‌شود. این برای زمانی مناسب است که داده‌های شما کل جامعه را تشکیل می‌دهند.
* **ddof =1 :** انحراف معیار نمونه محاسبه می‌شود. این برای زمانی مناسب است که داده‌های شما نمونه‌ای از یک جامعه بزرگ‌تر هستند و می‌خواهید تخمینی بی‌طرفانه از انحراف معیار جامعه داشته باشید.

**تفاوت در خروجی**

* وقتی ddof=0است، انحراف معیار با تقسیم بر Nمحاسبه می‌شود، که منجر به مقادیر بزرگ‌تر برای Z-score می‌شود.
* وقتی ddof=1است، انحراف معیار با تقسیم بر N−1محاسبه می‌شود، که باعث می‌شود انحراف معیار کمی بزرگ‌تر و در نتیجه Z-score ‌ها کمی کوچک‌تر شوند.

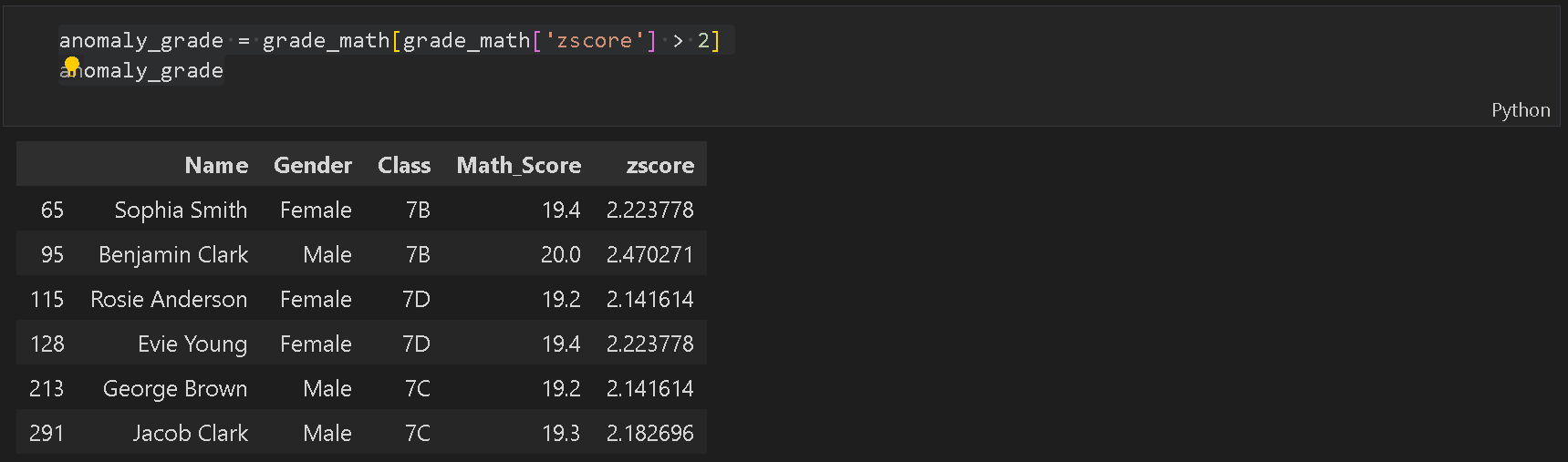
**چه زمانی از ddof=0یا ddof=1 استفاده کنیم؟**

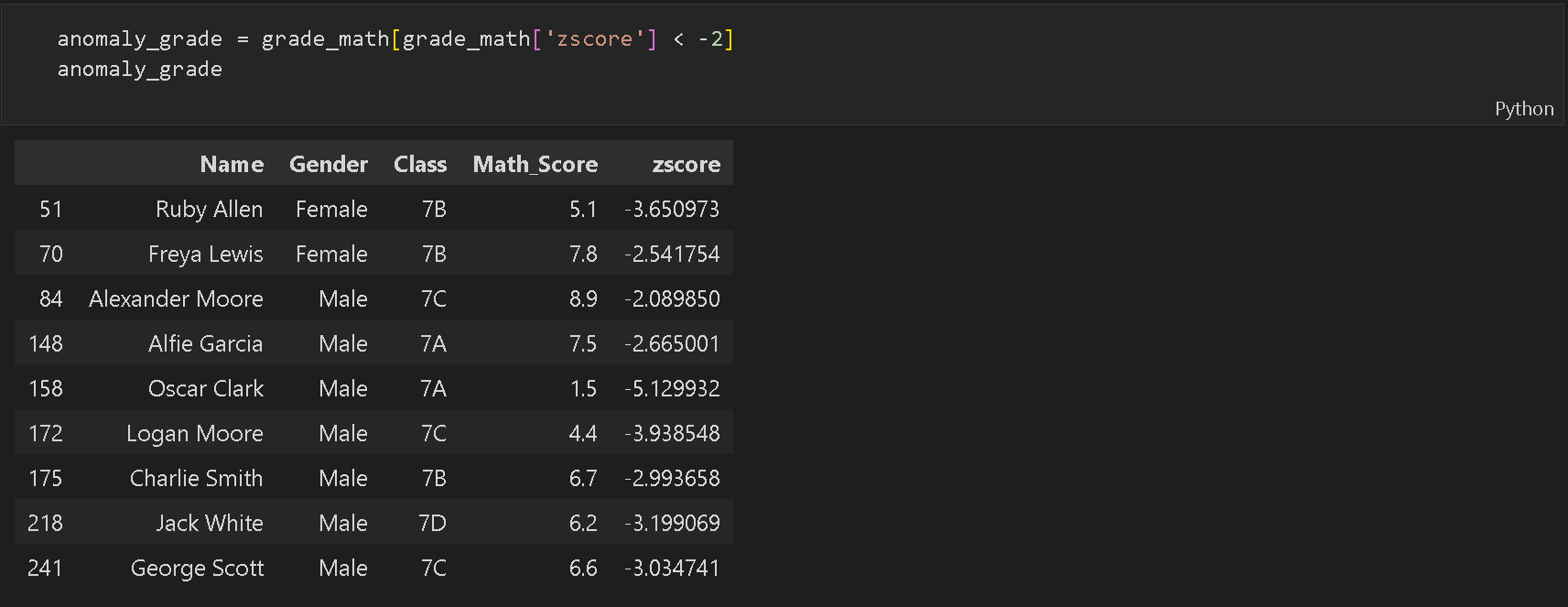
* **اگر داده‌ها کل جامعه را نشان می‌دهند :** از ddof=0استفاده کنید.
* **اگر داده‌ها نمونه‌ای از یک جامعه بزرگ‌تر هستند:** از ddof = 1 استفاده کنید تا تخمین بهتری از انحراف معیار جامعه داشته باشید.

**نکات تکمیلی**

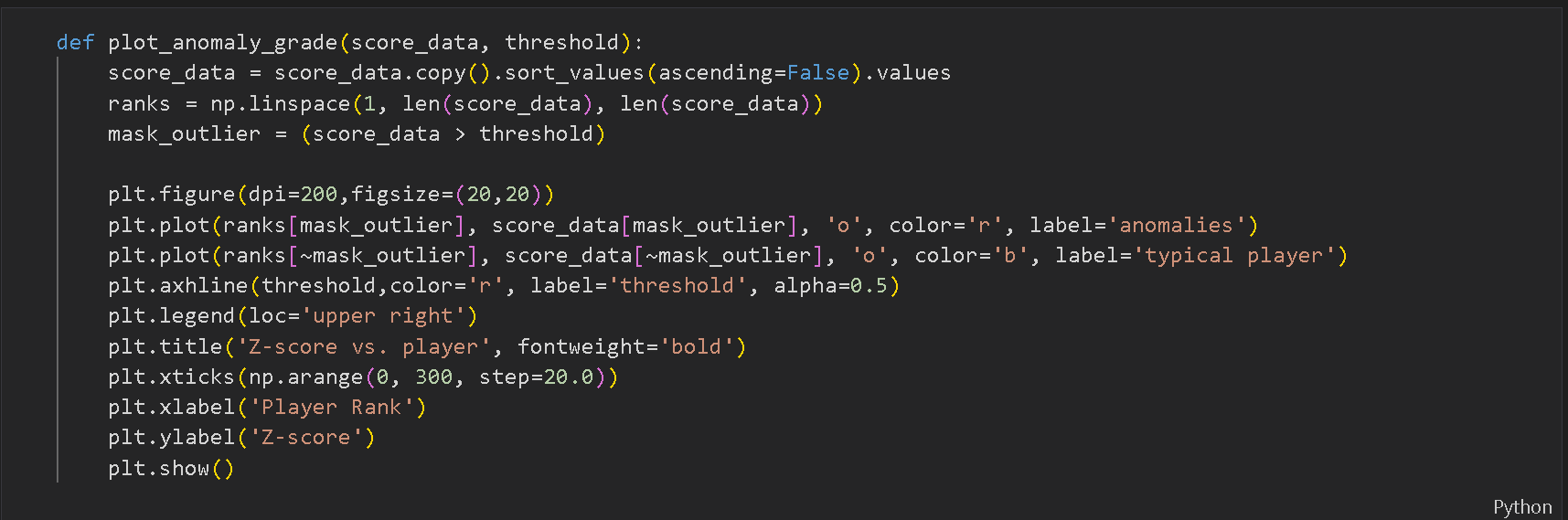
* در عمل، تفاوت بین ddof=0 و ddof =1برای داده‌های با حجم زیاد (تعداد داده‌های زیاد) کم می‌شود، زیرا N−1و N به هم نزدیک می‌شوند.
* در numpy.std(که در محاسبات zscore استفاده می‌شود)، رفتار پیش‌فرض ممکن است متفاوت باشد، بنابراین همیشه مقدار ddof را به صراحت مشخص کنید تا از نتایج غیرمنتظره جلوگیری شود.

اما برای پیدا کردن آنومالی ها با zscore باید یک threshold تعریف کنیم و اعداد بالای آن threshold را آنومالی در نظر بگیریم یعنی اگرz-score نمره بیشتر از threshold شد. آن داده آنومالی است. کدی که داده های آنومالی مشخص می کنند را در صفحه بعد خواهید دید.

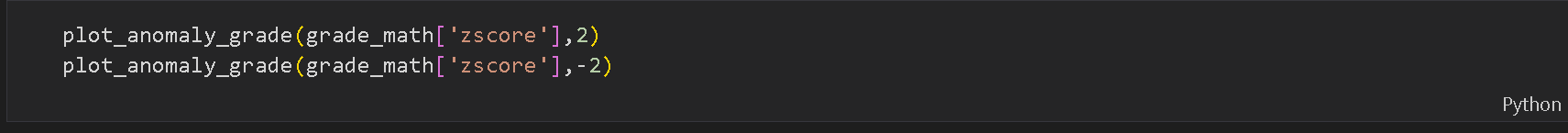


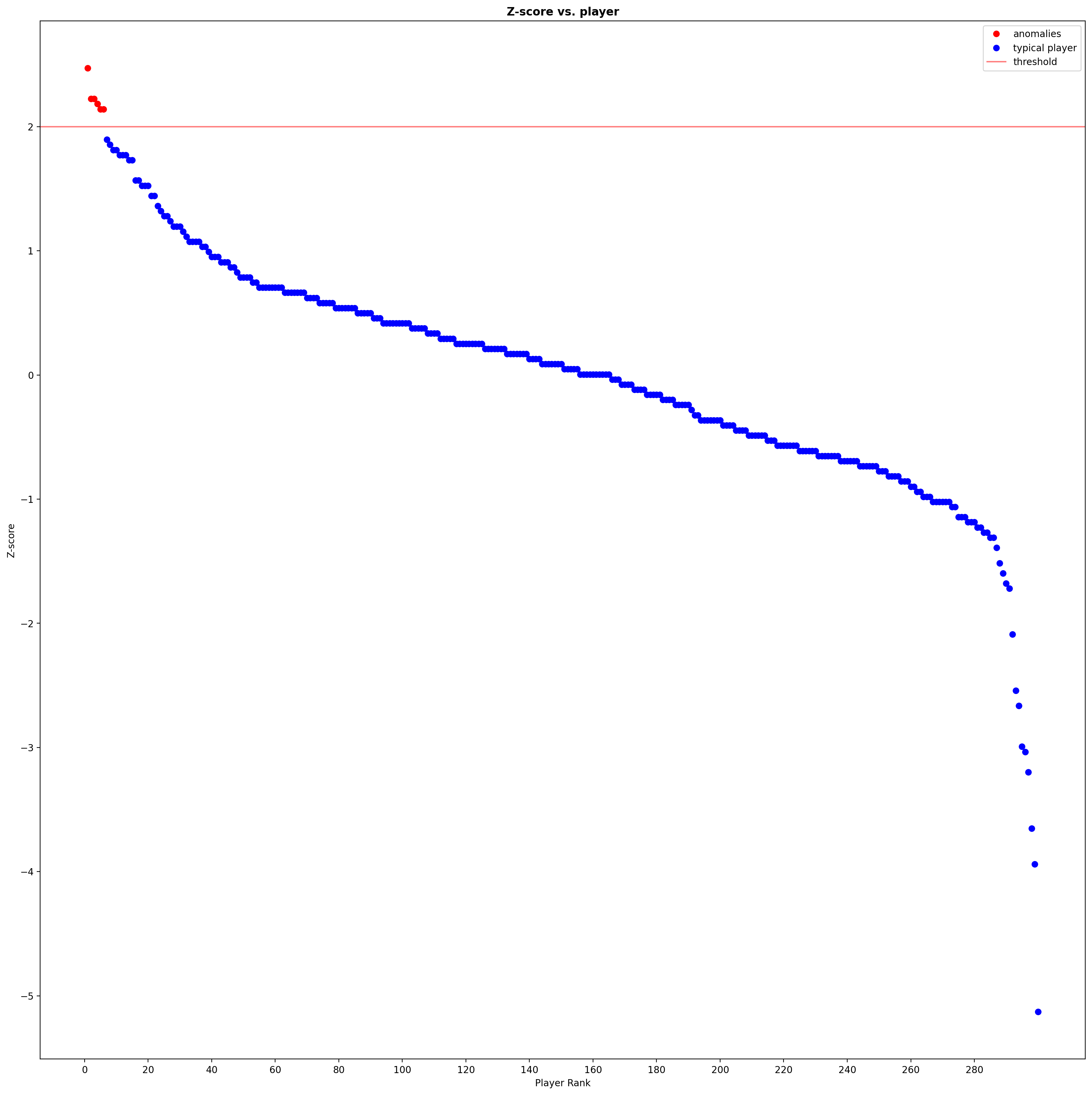


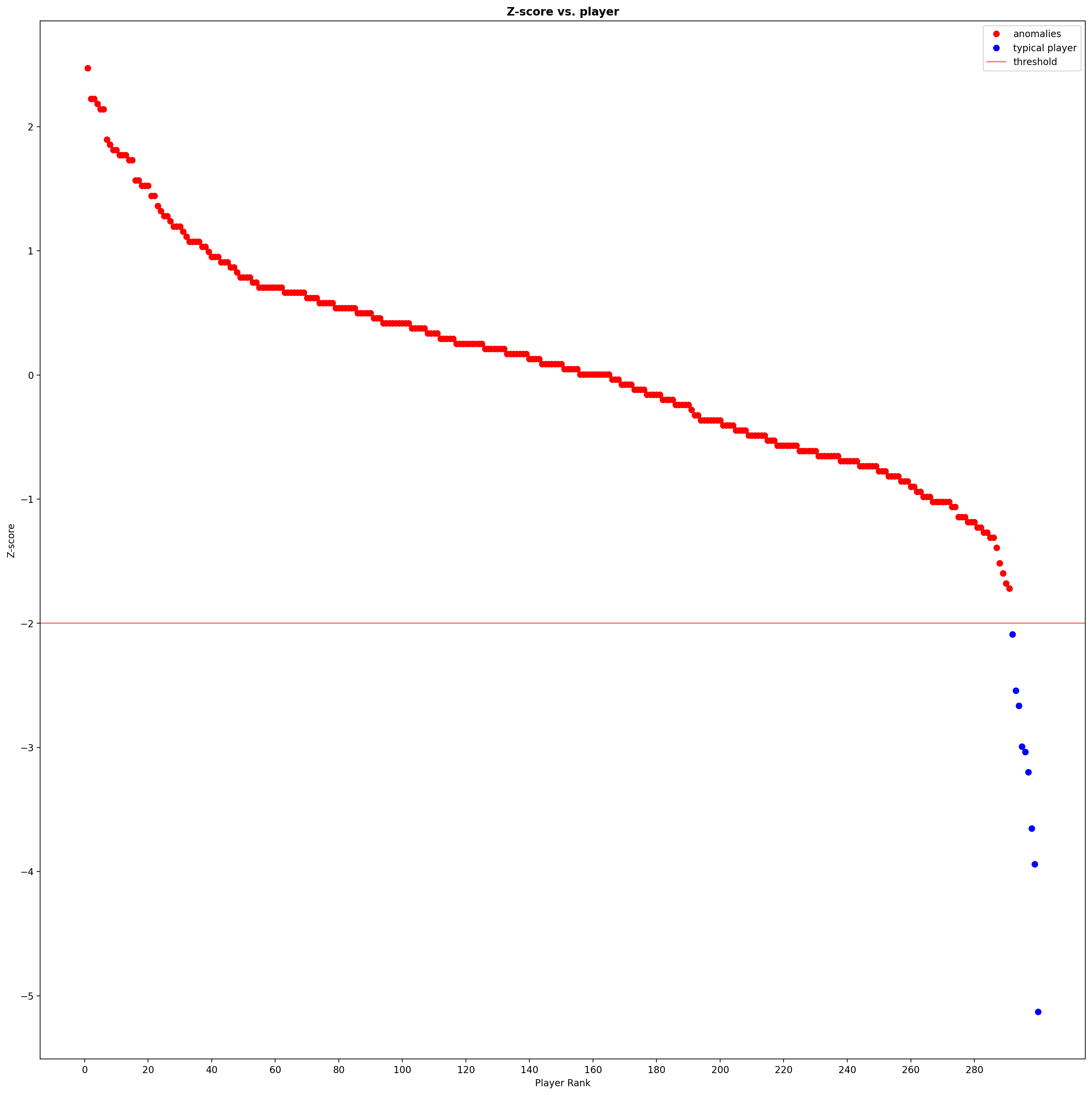
بعد از این می خواهم آنومالی ها را روی نمودار نشان بدهم برای این کار باید تابع تعریف کنیم که یک مقدار threshold و دیگری نمرات دانش آموزان را بگیرد بعد یک نسخه از داده‌ها می‌سازه (برای جلوگیری از تغییر داده اصلی) داده‌ها رو به ترتیب نزولی مرتب می‌کنه (بالاترین Z-score اول) و values داده رو به آرایه NumPy تبدیل می‌کنه. در خط بعدی یک آرایه به نام ranks تولید می‌کنه که رتبه (Rank) بازیکن‌ها رو مشخص می‌کنه. در خط سوم یک ماسک بولی (True/False) درست می‌کنه. اگر Z-score بازیکنی از آستانه (threshold) بیشتر باشه، اون بازیکن **آنومالی** حساب میشه. پس از این ها نمودار رسم می کنیم برای این با figure اندازه آن را مشخص می کنیم بعد با plot مشخص می کنیم دایره قرمز مربوط به آنومالی ها و بقیه را آبی رسم می کند بعد از آن خط افقی قرمزی که این همان آستانه (threshold) بین آنومالی و نرمال ها را می کشد. در خط های بعد نمودار ار تنظیم می کند اعمم تیتر نمودار و لیبل هرکدام از محور ها درجه بندی نمودار و... در آخر با show اون نمودار نمایش می دهد. کد این تابع تعریف شده را در زیر مشاهده می کنید



بعد از آن تابع را با آستانه "2" و (2-) صدا زده و کد و خرو جی آن به شرح زیر است:







Modified Z-score

**z-score اصلاح‌شده** (Modified Z-score) نسخه‌ای از z-score استاندارد است که برای داده‌هایی که توزیع نرمال ندارند یا در برابر داده‌های پرت (outliers) حساس هستند، بهبود یافته است. این معیار به‌ویژه در تحلیل داده‌های غیرنرمال یا داده‌هایی با پراکندگی بالا کاربرد دارد.

**تعریف z-score اصلاح‌شده**

z-score استاندارد به صورت زیر محاسبه می‌شود:

که در آن xمقدار داده، μ میانگین و σ انحراف معیار است.

اما z-score اصلاح‌شده، به جای میانگین و انحراف معیار، از **میانه (median)** و **میانگین انحراف مطلق از میانه (MAD)** استفاده می‌کند:

که در آن:

* **: Median** میانه داده‌ها.
* **MAD (Median Absolute Deviation):**میانه انحرافات مطلق داده‌ها از میانه، یعنی:

**چرا z-score اصلاح‌شده بهتر است؟**

1. **مقاومت در برابر پرت‌ها (Outliers)**:
   * z-score استاندارد به شدت تحت تأثیر داده‌های پرت قرار می‌گیرد، زیرا میانگین و انحراف معیار به مقادیر دور از مرکز حساس هستند.
   * z-score اصلاح‌شده با استفاده از میانه و MAD، که معیارهای مقاوم‌تری هستند، کمتر تحت تأثیر پرت‌ها قرار می‌گیرد.
2. **مناسب برای توزیع‌های غیرنرمال**:
   * z-score استاندارد فرض می‌کند داده‌ها توزیع نرمال دارند، اما z-score اصلاح‌شده برای داده‌هایی با توزیع‌های نامتقارن یا غیرنرمال نیز قابل استفاده است.
3. **دقت بیشتر در تحلیل داده‌های واقعی**:
   * در داده‌های واقعی که اغلب پرت‌ها یا توزیع‌های غیرنرمال دارند، z-score اصلاح‌شده نتایج دقیق‌تری برای شناسایی انحرافات ارائه می‌دهد.

**کاربردهای z-score اصلاح‌شده**

1. **شناسایی داده‌های پرت**:
   * در تحلیل داده‌ها، z-score اصلاح‌شده برای تشخیص داده‌های غیرعادی (outliers) در مجموعه‌های داده‌ای که توزیع نرمال ندارند، بسیار مفید است. معمولاً

اگر < 3.5 باشد، داده به عنوان پرت در نظر گرفته می‌شود.

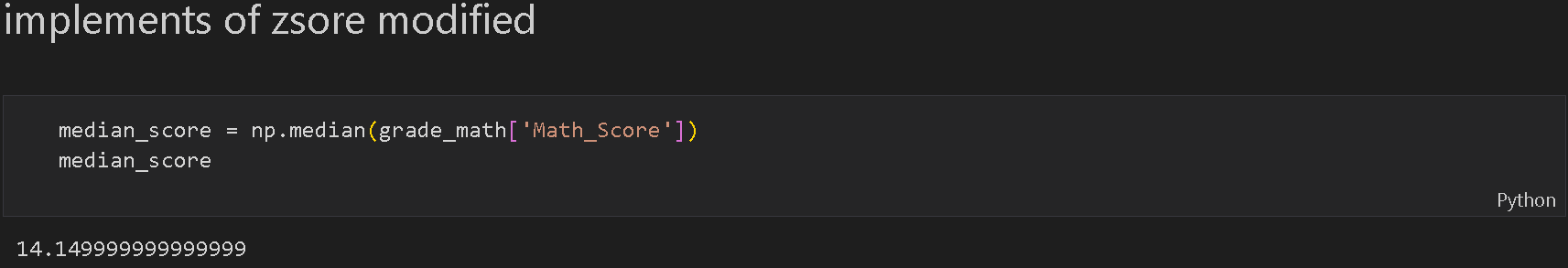
1. **تحلیل داده‌های مالی**:
   * در بازارهای مالی، داده‌ها اغلب پرت‌های زیادی دارند (مثل جهش‌های قیمتی). z-score اصلاح‌شده برای شناسایی این پرت‌ها یا تحلیل نوسانات مناسب است.
2. **کنترل کیفیت**:
   * در صنایع، برای شناسایی محصولات معیوب یا انحرافات در فرآیندهای تولید که داده‌ها ممکن است توزیع غیرنرمال داشته باشند.
3. **تحلیل داده‌های زیستی و پزشکی**:
   * داده‌های زیستی (مثل فشار خون یا سطح قند خون) ممکن است توزیع غیرنرمال داشته باشند. z-score اصلاح‌شده برای شناسایی مقادیر غیرعادی در این داده‌ها کاربرد دارد.

**تفاوت با z-score استاندارد**

* **حساسیت به پرت‌ها**: z-score استاندارد به پرت‌ها حساس است، در حالی که z-score اصلاح‌شده مقاوم‌تر است.
* **فرض توزیع**: z-score استاندارد فرض می‌کند داده‌ها توزیع نرمال دارند، اما z-score اصلاح‌شده این فرض را ندارد.
* **محاسبات**: z-score استاندارد از میانگین و انحراف معیار استفاده می‌کند، در حالی که z-score اصلاح‌شده از میانه و MAD بهره می‌برد.

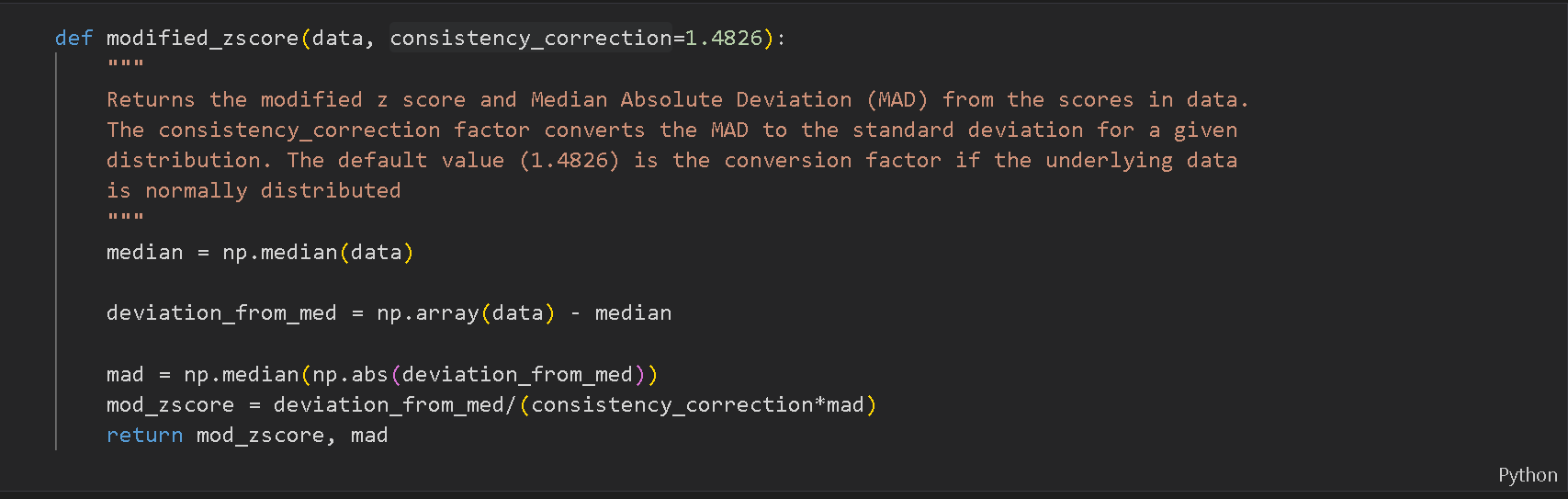
حال می خواهیم این Z-score اصلاح شده را پیاده سازی کنیم.

ابتدا میانه این نمرات را محاسبه کرد چون تعداد زوج است میانگین اون دو عدد وسط میشه میانه

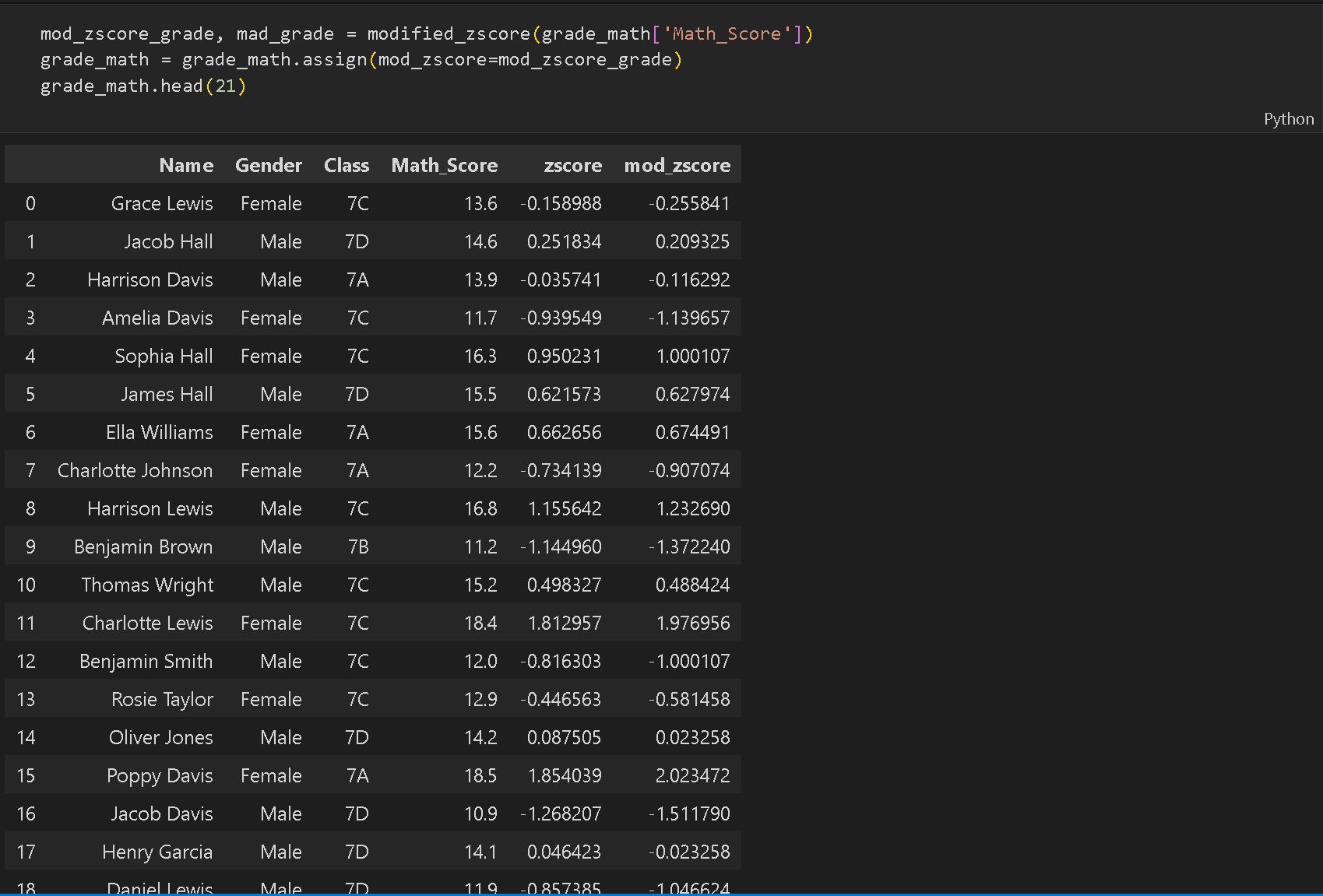


برای z-score اصلاح شده هیچ متدی در library های پایتون نیست پس باید خودمان از صفر متدش را بنویسیم . حال متد Z-score اصلاح شده تعریف می کنیم. این تابع دارای دو ورودی است یک نمرات و یکی هم consistency\_correction یک ثابت تصحیح که MAD را به انحراف معیار تبدیل می‌کند. مقدار پیش‌فرض آن 1.4826 است، که برای داده‌های با توزیع نرمال مناسب است . در این تابع اول میانه را حساب می کند بعد میاد اختلاف هر نمره را با میانه حساب می کند و np.abs(deviation\_from\_med) مقدار مطلق انحرافات از میانه را محاسبه می‌کند در آخر MAD را محاسبه و mod\_zscore را return می کند

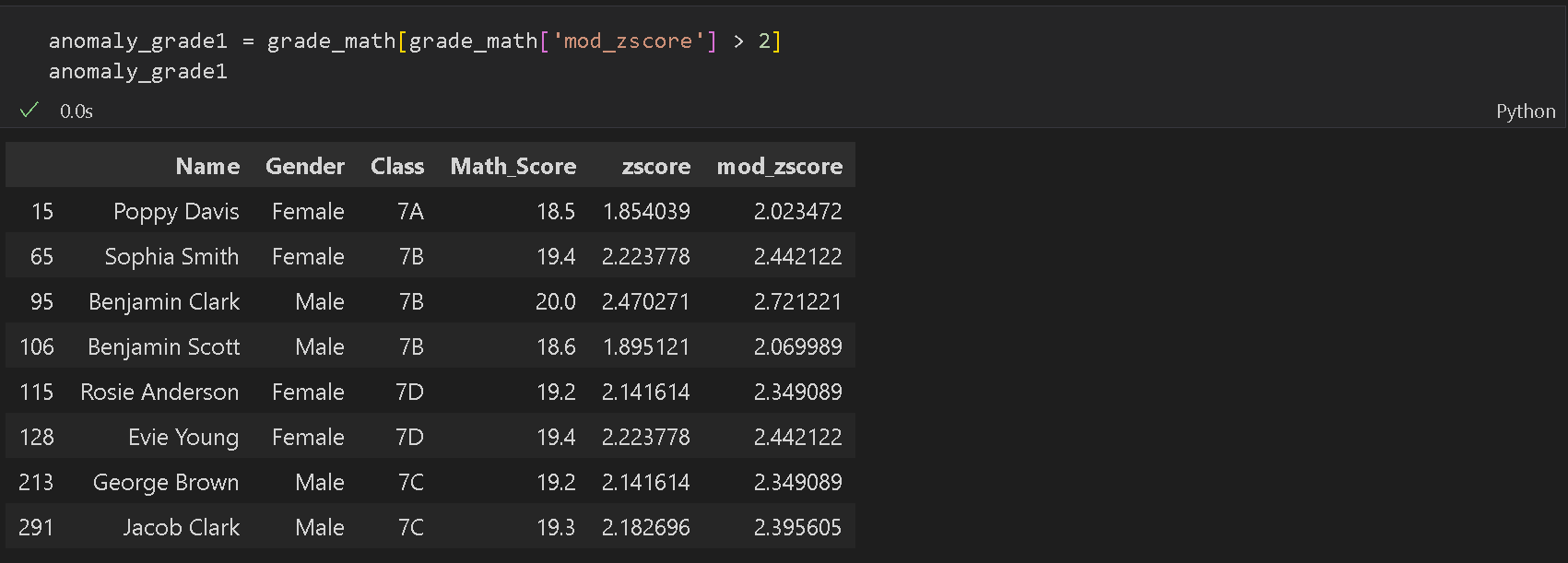
MAD معیاری مقاوم برای پراکندگی داده‌هاست و برخلاف انحراف معیار، به داده‌های پرت حساس نیست.



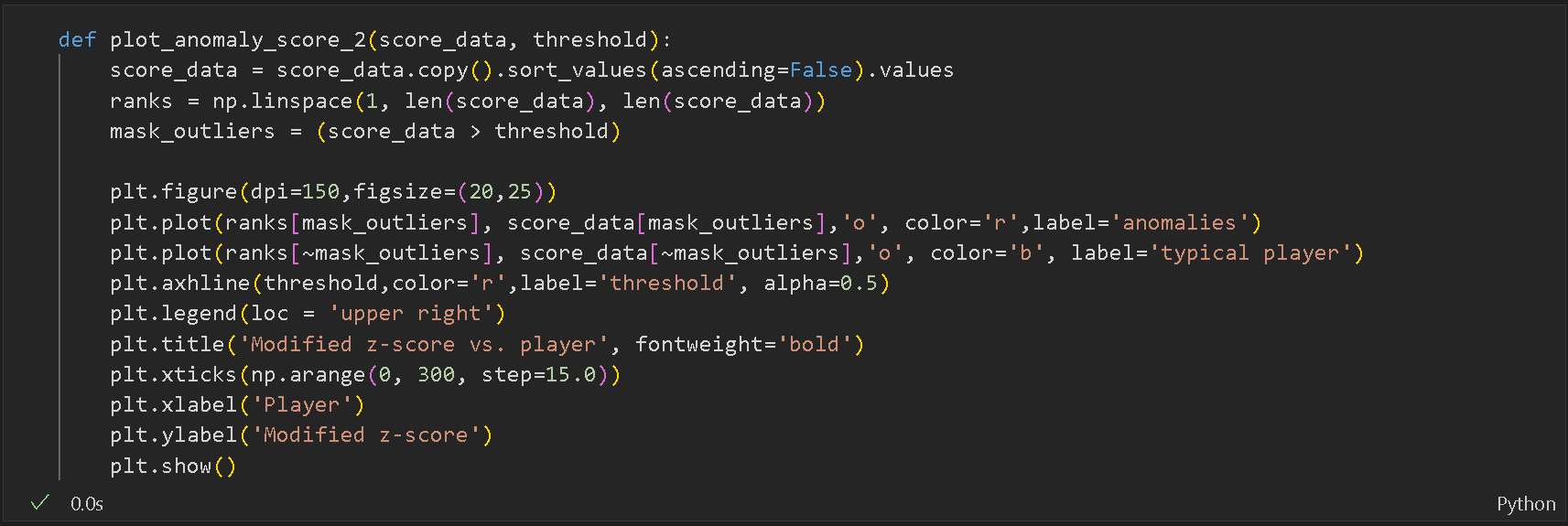
حال تابع را صدا زده و یک ستون مربوط به z\_score اصلاح شده به دیتاست اضافه کرده. کد و خروجی آن را در صفحه بعد خواهید دید



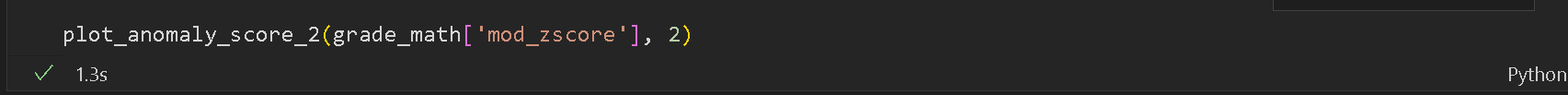
همانطور که می دانید برای z\_score اصلاح شده هم باید یک آستانه تعریف و اگر z\_score اصلاح شده بیشتر از آستانه بود آن را آنومالی در نظر بگیر کد آن که داده های آنومالی را با آستانه 2 بدست آمده را در زیر مشاهده می کنید.

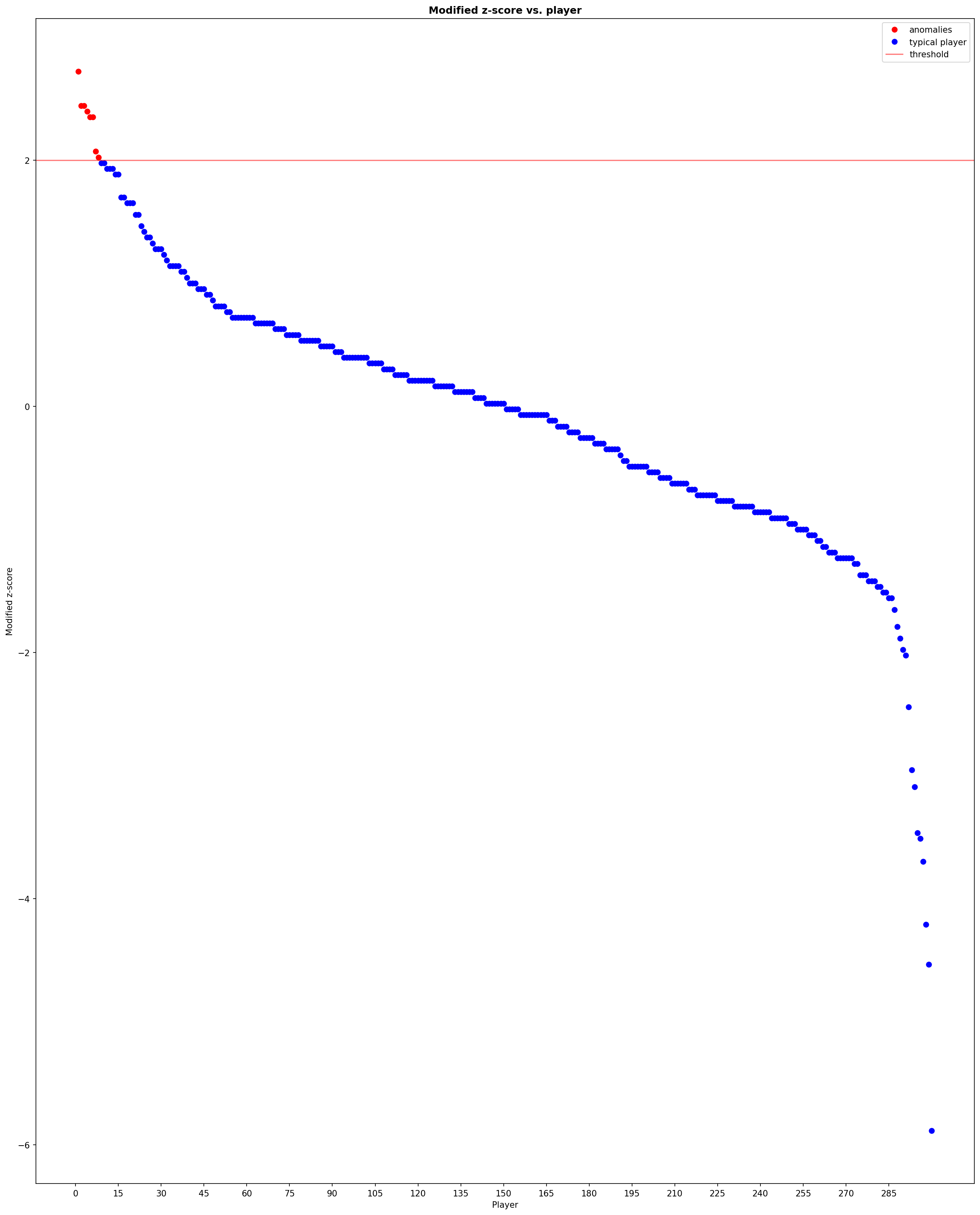


بعد از آن داده های آنومالی را در نمودار نمایش می دهیم. من این نمودار با همان تابعی که قبلا تعریف کردم برای نمایش آنومالی های z\_score استفاده شد استفاده می کنیم.



حال این تابع را صدا زده و کد و خروجی آن را در زیر مشاهد ه می کنید.





**مفاهیم IQR**

IQR یا دامنه بین‌چارک‌ها (Interquartile Range) معیاری آماری است که برای اندازه‌گیری پراکندگی داده‌ها استفاده می‌شود. این معیار فاصله بین چارک اول (Q1) و چارک سوم (Q3) را نشان می‌دهد و به صورت زیر محاسبه می‌شود:

**چارک اول**  25٪ :**(Q1)** از داده‌های مرتب‌شده که کمتر از این مقدار هستند.

**چارک سوم** 75٪ :**(Q3)** از داده‌های مرتب‌شده که کمتر از این مقدار هستند.

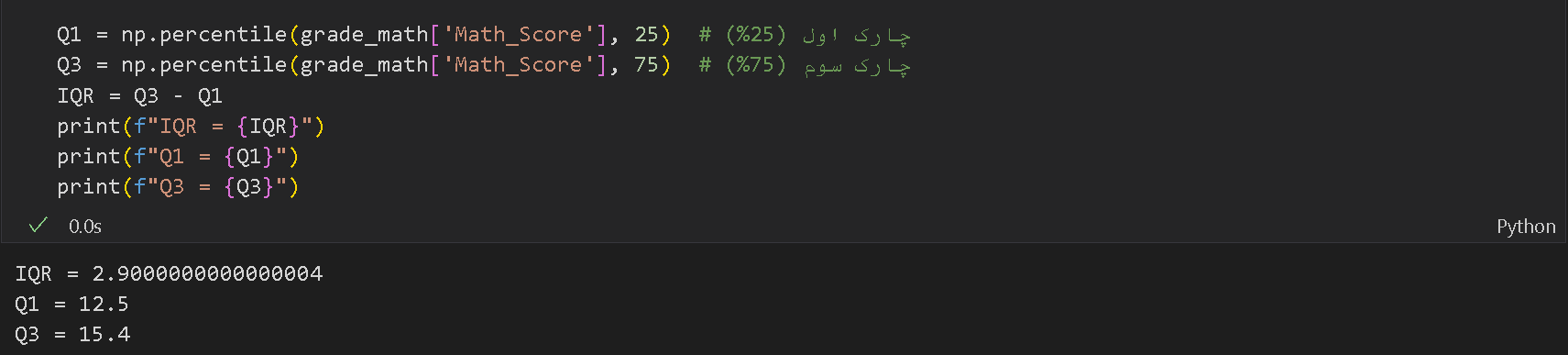
**کاربردهای IQR:**

1. **اندازه‌گیری پراکندگی داده‌ها :** IQR نشان‌دهنده گستردگی داده‌های میانی (50٪ میانی داده‌ها) است و نسبت به انحراف معیار کمتر تحت تأثیر داده‌های پرت (Outliers) قرار می‌گیرد.
2. **شناسایی داده‌های پرت :** از IQR برای تشخیص داده‌های پرت استفاده می‌شود. معمولاً داده‌هایی که خارج از بازه زیر باشند، پرت محسوب می‌شوند:
3. **تحلیل توصیفی داده‌ها :** IQR در کنار میانگین یا میانه برای توصیف توزیع داده‌ها به کار می‌رود، به‌ویژه در داده‌های غیرنرمال.
4. **مقایسه توزیع‌ها:** IQR برای مقایسه پراکندگی بین مجموعه‌های داده‌ای مختلف استفاده می‌شود.
5. **رسم نمودار جعبه‌ای (Boxplot):** IQR بخش اصلی جعبه در نمودار جعبه‌ای را تشکیل می‌دهد که توزیع داده‌ها را به‌صورت بصری نشان می‌دهد.

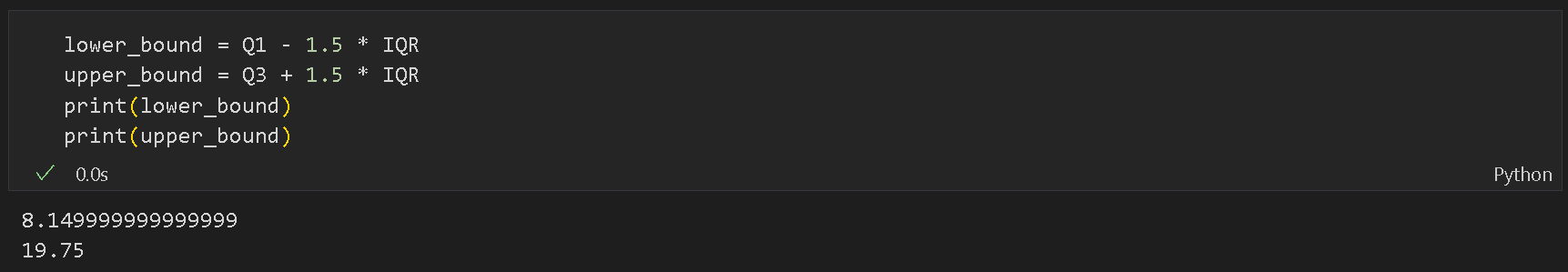
**مزایا:**

* مقاومت در برابر داده‌های پرت.
* ساده و قابل‌فهم برای تحلیل داده‌ها.

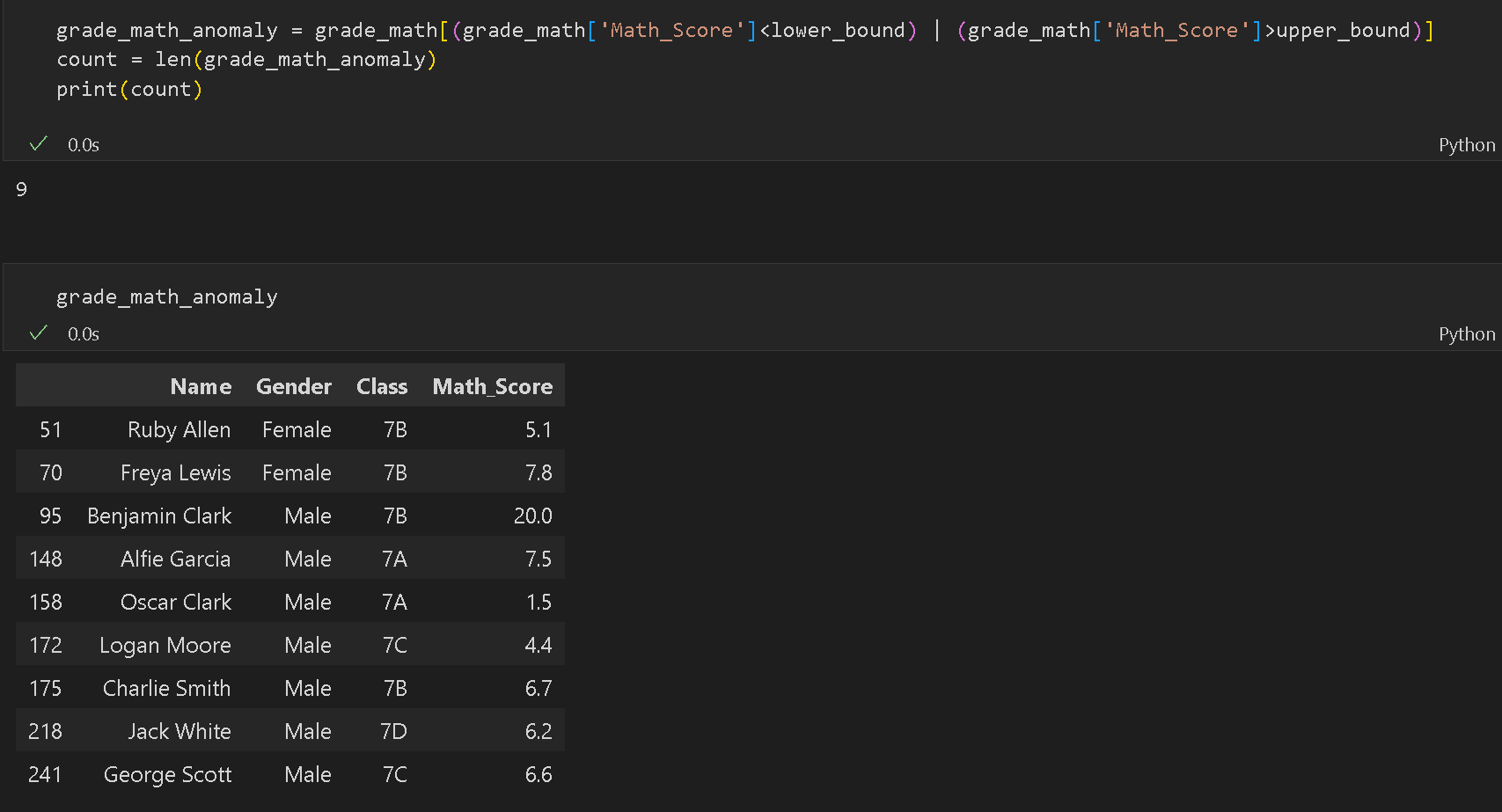
بعد از تعریف IQR باید کد مربوط به تشخیص آنومالی را با IQR بزنیم ابتدا library های موردنیازمون را import کرده که با library های z-score یکی است. پس از آن داده را با کتابخانه pandas می خونیم. در مرحله بعد Q1 و Q3 را با استفاده از کتابخانه numpy می سازیم بعد از طریق این دوتا IQR بدست آوردیم بعد همه این ها را print می کنیم کد و خروجی آن را در زیر مشاهده می کنید.



برای تشخیص آنومالی با IQR باید بازه درست با همین چارک ها و IQR که ابتدای بازه فرمولش و انتهای بازه آن این است اگر داده ما خارج از این بازه بیفتد آنومالی است پس یک lower bound (ابتدای بازه) و upperbound (انتهای بازه) محاسبه کرد و print می کنیم کد و خروجی آن به شرح زیر است

**

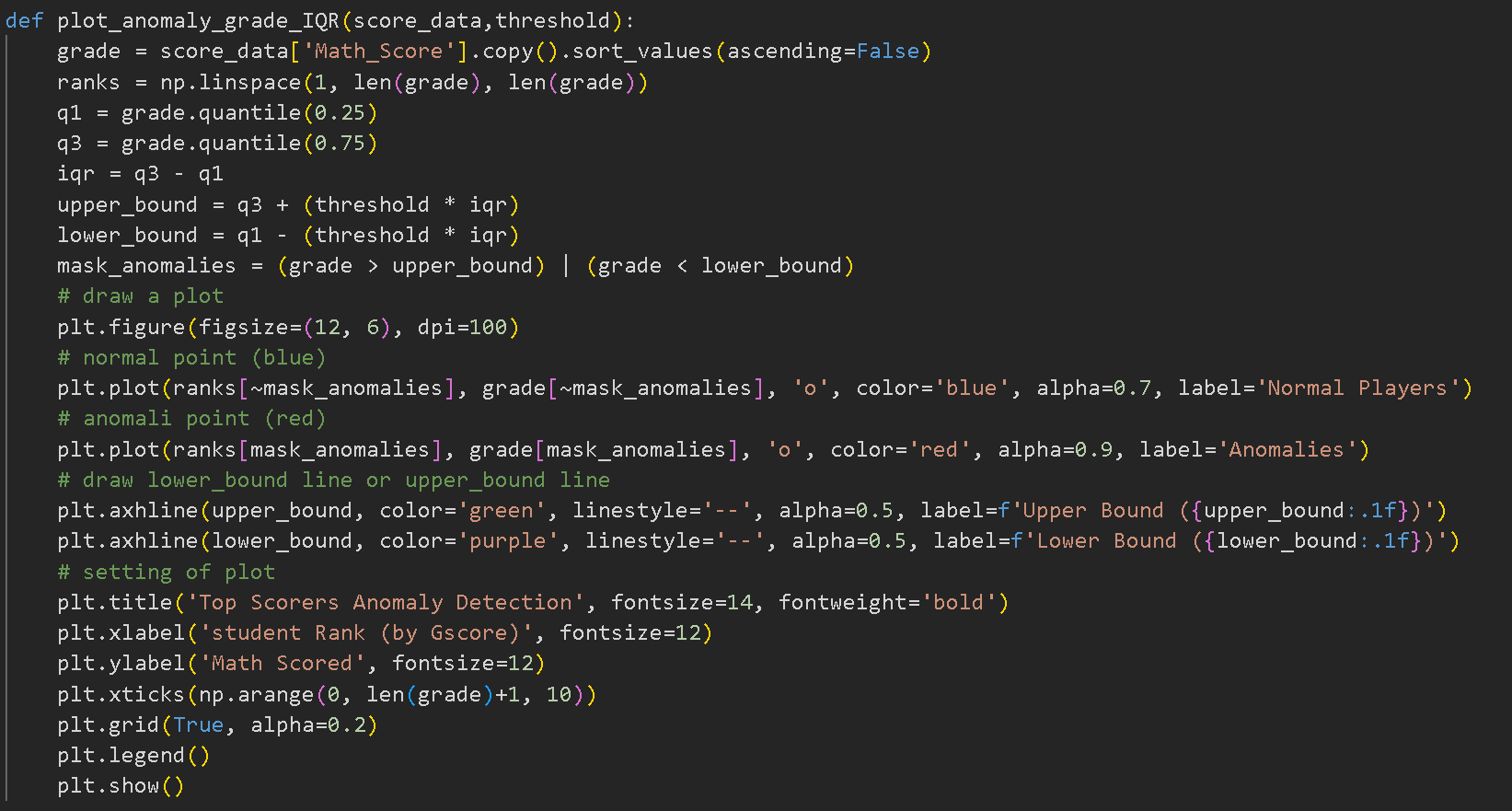
*در مرحله بعد باید داده های آنومالی را تشخیص دهد در ستون نمرات دیتاستمان باید شرط بزاریم و بگیم نمراتی که در این بازه نبودند را* return *کن بعد ازآن تعداد آنومالی ها به همراه خود داده های آنومالی را در خروجی صفحه بعد به همراه کد آن ها می توانید مشاهده کنید.*

**

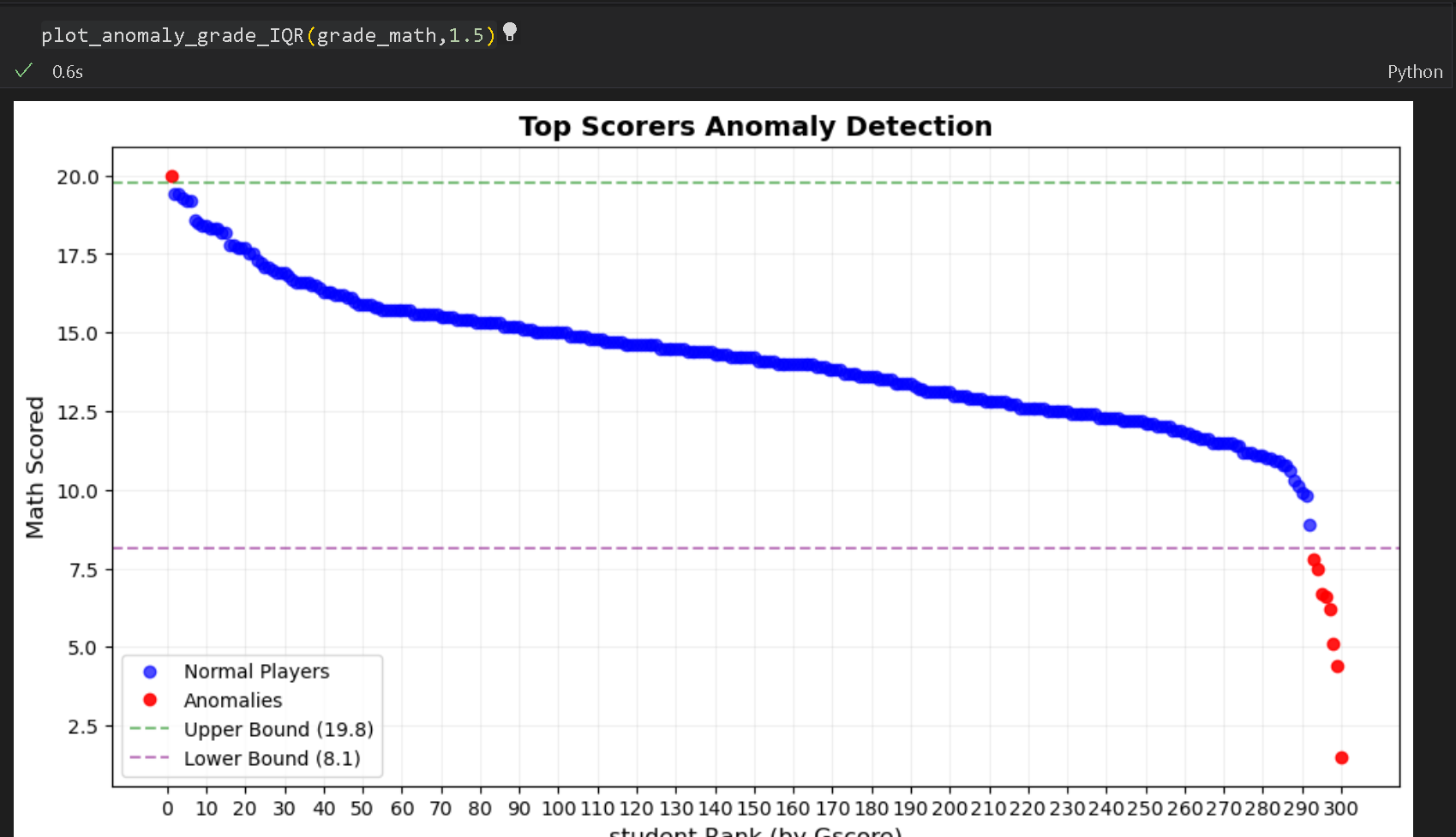
*حال در انتها می خواهیم تابعی برای نشون دادن آنومالی ها با نمودار نقطه ای* (scatter plot)*نشان دهیم.*

*تابعی که می نویسیم مثل توابع قبلی رسم باید نمرات و ضریب* IQR *(که پیش فرض 1.5) را بگیرد و* rank *تعریف کنیم بعد* Q1 *و*Q3 *و* IQR *و* upper\_bound *و* lower\_bound *بدست می آوریم.*

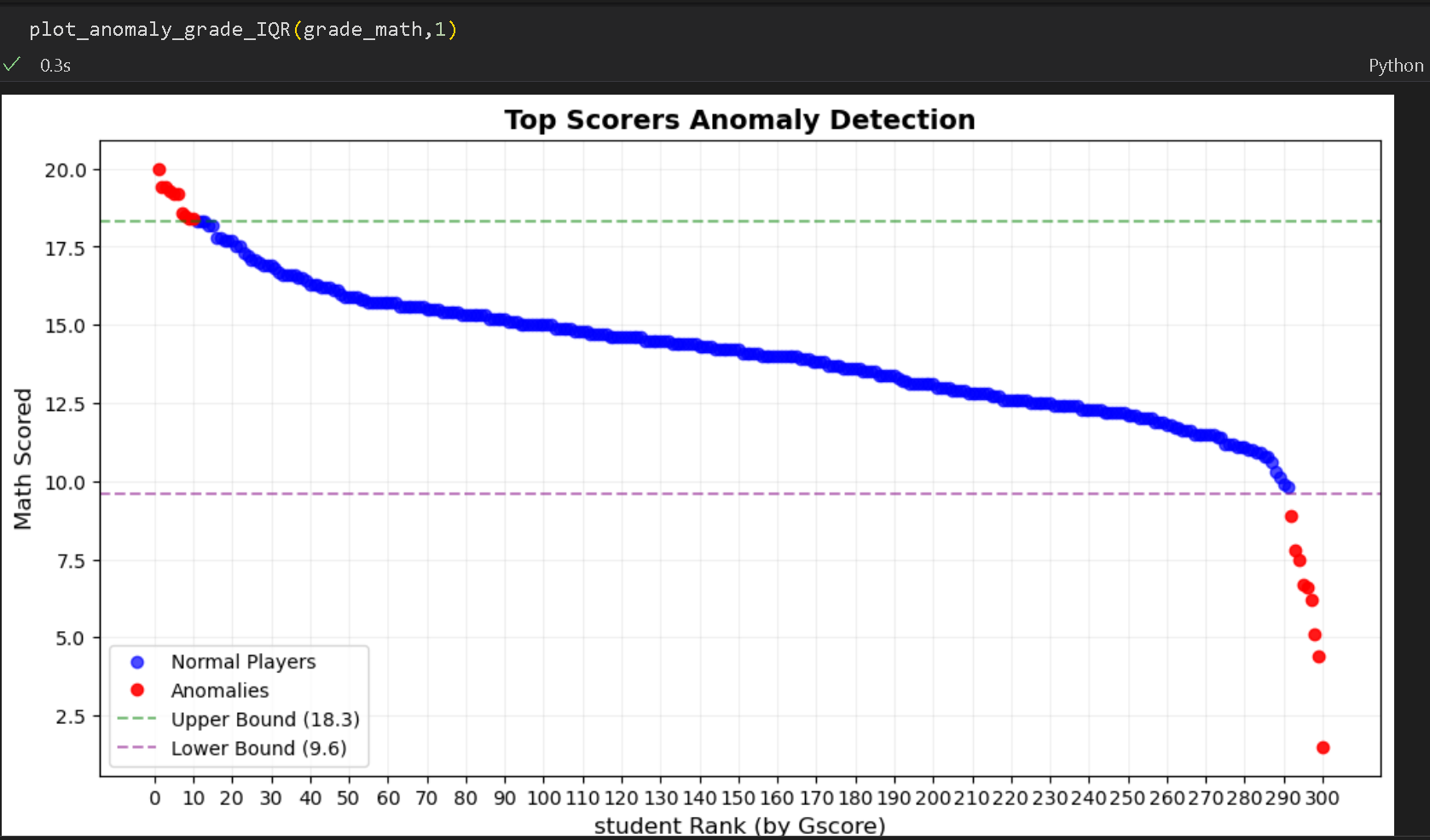
Mask anomalies *را تعریف کردیم یعنی داده های آنومالیمان را بدست می آوریم با آن حال نمودار را می کشیم ابتدا با* figure *اندازه صفحه خروجی را تنظیم کرده پس ازآن داده های نرمال با آبی در خط بعد داده های آنومالی را با قرمز مشخص کرده در خطوط بعد خط که* lower\_bound *با بنفش و* upper\_bound *با سبز* *مشخص می کند در انتهای آن جزییات نمودار را تنظیم می کنیم مثل تیتر نمودار یا* label *محورهای* x *و* y *و... در آخر با* show *آن را نمایش می دهیم. کد آن را در صفحه بعد می بینیم.*

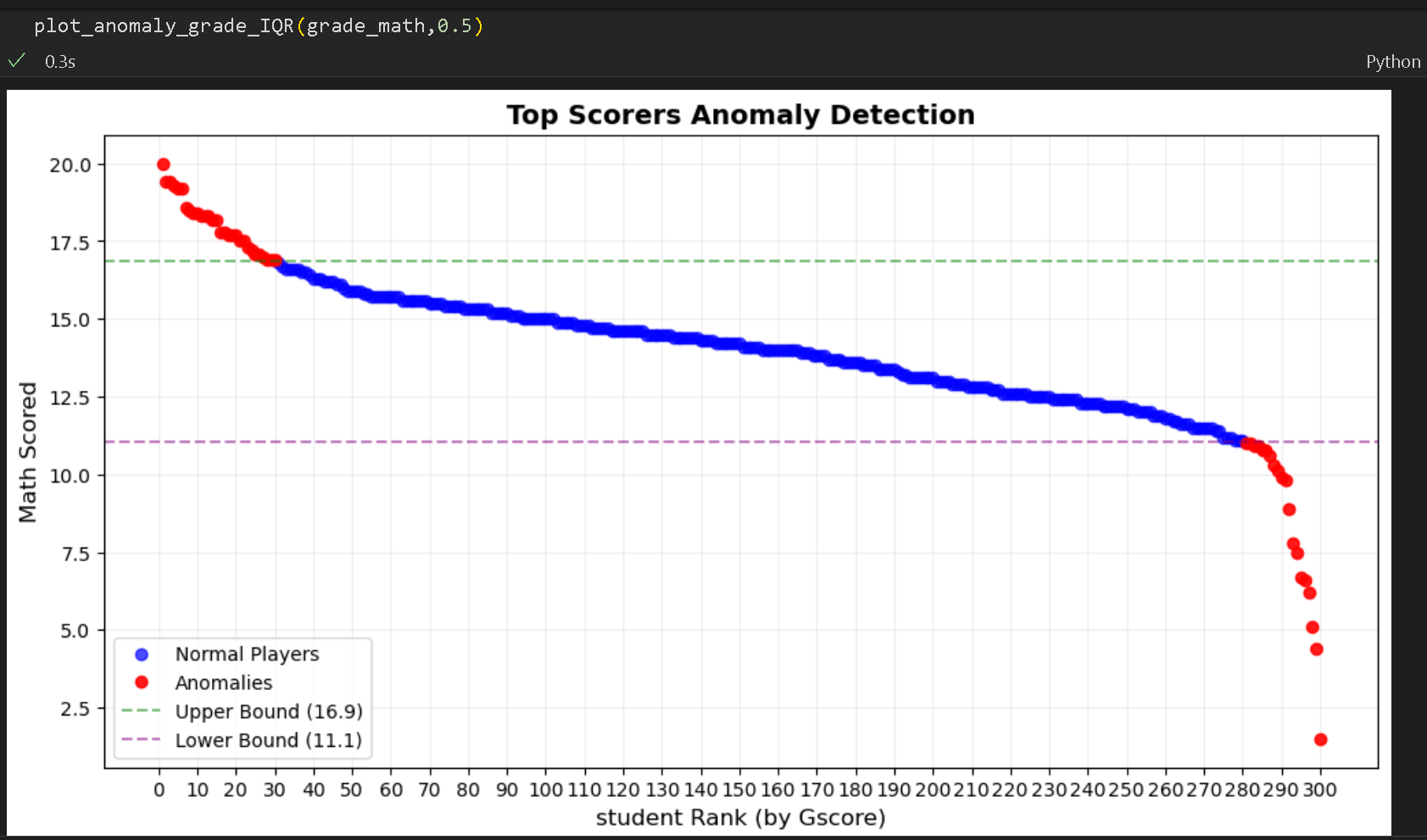
**

*حال تابع را صدا می زنیم وکد و خروجی آن را در زیر مشاهده می کنید*

**

*با ضرایب دیگه مثل 1 و 0.5 هم نمودارش می کشیم و خروجی آن در صفحه بعد هست*

**

**

***مفاهیم* GMM**

*مدل* GMM *یا* Gaussian Mixture Model *(مدل مخلوط گاوسی) یک مدل آماری احتمالی است که برای مدل‌سازی داده‌هایی استفاده می‌شود که فرض می‌شود از ترکیبی از چندین توزیع گاوسی (نرمال) تشکیل شده‌اند. این مدل داده‌ها را به صورت ترکیبی از چند توزیع گاوسی با پارامترهای مختلف (میانگین، واریانس و وزن) توصیف می‌کند.*

**GMM *چیست؟***

GMM *فرض می‌کند که داده‌ها از* K *توزیع گاوسی تشکیل شده‌اند، که هر کدام پارامترهای خاص خود* )*میانگین* μk​، *کوواریانس*  *Σk و* GMM *وزن را دارند. احتمال هر داده به صورت زیر محاسبه می‌شود:*

*​*

* *:* k *تابع چگالی توزیع گاوسی برای مؤلفه*
* : وزن مؤلفه K (جمع وزن برابر با 1 است.(
* : میانگین توزیع گوسی k
* *: ماتریس کوواریانس توزیع گاوسی* K

GMM از الگوریتم **Expectation-Maximization (EM)** برای یادگیری پارامترهای این توزیع‌ها (وزن‌ها، میانگین‌ها و کوواریانس‌ها) استفاده می‌کند.

**کاربردهای GMM**

1. **خوشه‌بندی (Clustering):** GMM برای خوشه‌بندی داده‌ها استفاده می‌شود، مشابه الگوریتم K-Means، اما با این تفاوت که GMM توزیع‌های نرمال را فرض می‌کند و می‌تواند خوشه‌های غیرکروی یا همپوشان را مدل کند.
2. **مدل‌سازی چگالی (Density Estimation):** GMM برای تخمین چگالی داده‌ها در کاربردهایی مثل تحلیل داده‌های چندبعدی استفاده می‌شود.
3. **تشخیص آنومالی (Anomaly Detection):** GMM برای شناسایی داده‌های غیرعادی (آنومالی) به کار می‌رود، که در ادامه توضیح داده می‌شود.
4. **تقسیم‌بندی تصویر (Image Segmentation):** در پردازش تصویر، GMM برای تقسیم‌بندی پیکسل‌ها به کلاس‌های مختلف استفاده می‌شود.
5. **تشخیص گفتار و پردازش صوت:** GMM برای مدل‌سازی ویژگی‌های صوتی در سیستم‌های تشخیص گفتار استفاده می‌شود.
6. **بیوانفورماتیک:** برای تحلیل داده‌های ژنومی یا پروتئینی.

**استفاده از GMM برای تشخیص آنومالی**

تشخیص آنومالی با GMM بر اساس این ایده است که داده‌های نرمال (عادی) با احتمال بالا توسط مدل مخلوط گاوسی توصیف می‌شوند، در حالی که داده‌های غیرعادی (آنومالی‌ها) احتمال کمتری دارند. مراحل کلی عبارتند از:

1. **آموزش مدل GMM:**
   * یک مجموعه داده آموزشی (ترجیحاً بدون آنومالی یا با تعداد کمی آنومالی) استفاده می‌شود.
   * GMM با استفاده از الگوریتم EM روی داده‌ها آموزش داده می‌شود تا پارامترهای توزیع‌های گاوسی) μk​، , Σk
   * تعداد مؤلفه‌های گاوسی (K) معمولاً با روش‌هایی مثل BIC

(Bayesian Information Criterion) یا آزمون و خطا انتخاب می‌شود.

1. **محاسبه احتمال داده‌ها:**

* برای هر داده جدید x، (p(x)) با استفاده از مدل GMM محاسبه می‌شود.
* داده‌هایی که چگالی احتمال آن‌ها کمتر از یک آستانه مشخص باشد، به عنوان آنومالی شناسایی می‌شوند.

1. **تعیین آستانه:**

* آستانه معمولاً با تحلیل داده‌های آموزشی یا با روش‌های آماری (مثل صدک‌ها یا قانون

یا IQR . 1.5) تعیین می‌شود.

* به عنوان مثال، داده‌هایی که چگالی احتمال آن‌ها کمتر از صدک 5٪ باشد، می‌توانند به عنوان آنومالی در نظر گرفته شوند.

1. **شناسایی آنومالی:**

* داده‌هایی که p(x) آن ها کمتر از به عنوان غیرعادی (آنومالی) طبقه‌بندی می‌شوند.
* این روش به‌ویژه در داده‌های چند بعدی که توزیع پیچیده‌ای دارند، مفید است.

**مزایای استفاده از GMM برای تشخیص آنومالی**

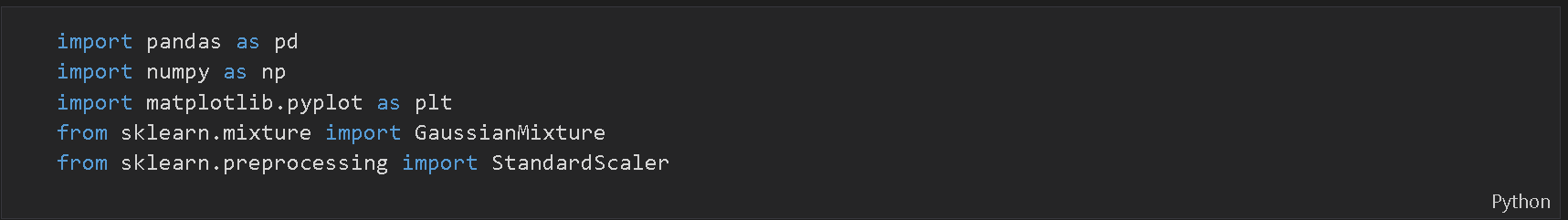
* **انعطاف‌پذیری** GMM :می‌تواند توزیع‌های پیچیده و چندوجهی (multimodal) را مدل کند، برخلاف روش‌های ساده‌تر که فرض توزیع تک‌مؤلفه‌ای دارند.
* **احتمال‌محور** GMM:چگالی احتمال را ارائه می‌دهد که امکان تعیین آستانه‌های دقیق‌تر را فراهم می‌کند.
* **کارایی در داده‌های چندبعدی** GMM :در داده‌های با ابعاد بالا خوب عمل می‌کند.

**معایب**

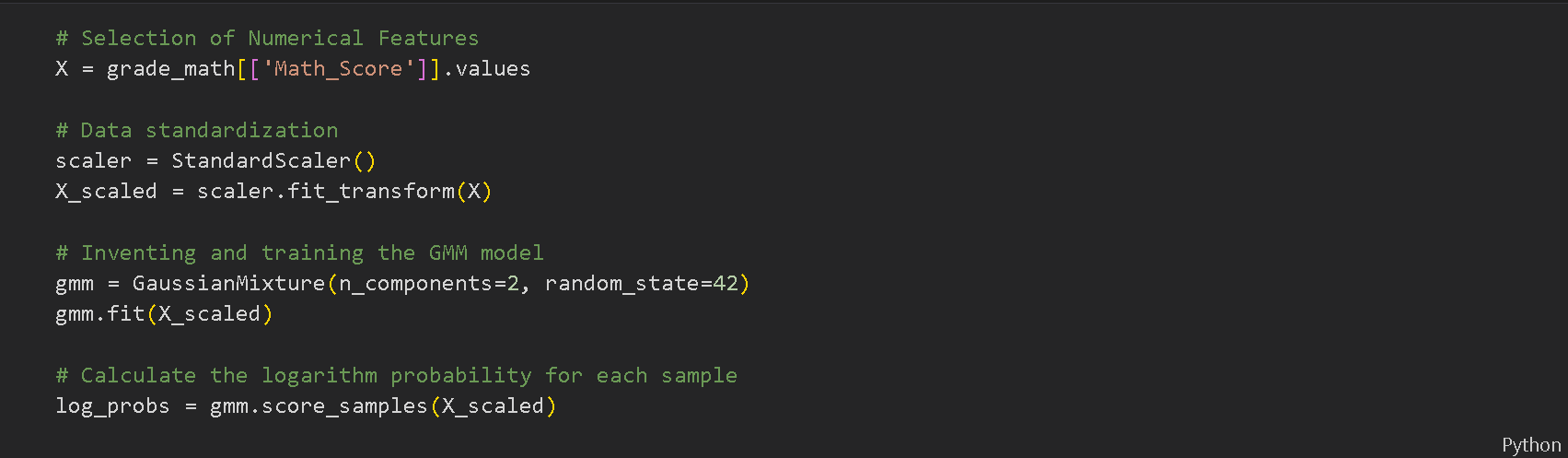
* **حساسیت به تعداد مؤلفه‌ها :** انتخاب تعداد مناسب مؤلفه‌های گاوسی (K) چالش‌برانگیز است.
* **محاسبات سنگین :** آموزش GMM روی داده‌های بزرگ یا ابعاد بالا می‌تواند زمان‌بر باشد.
* **فرض گاوسی بودن :** اگر داده‌ها از توزیع گاوسی تبعیت نکنند، عملکرد مدل کاهش می‌یابد.

حال به بررسی کد آن می پردازیم.

ابتدا برای زدن کد مدل GMM باید library های مدنظرمون را import کنیم. علاوه بر matplotlib , pandas , numpy باید کتابخانه gaussianmixture , standardscalar را هم import کنیم. برای استاندارد کردن داده( یعنی به شکل توزیع نرمال با انحراف معیار 1 و میانگین 0) باید از sklearn.preprocessing که مال کتابخانه standardscalar است باید استفاده کنیم. و برای اجرای مدل GMM باید از sklearn.mixture که مال کتابخانه gaussianmixture است استفاده کنیم. که کد به شرح زیر است



در مراحل ابتدا دیتاست با pandas مثل قبلی می خونیم، بعد از آن وارد اجرای مراحل استفاده از مدل GMM میشویم ابتدا ستون نمرات در متغیری قرار می دهیم بعد از StandardScalerبرای استانداردسازی (نرمال‌سازی) داده‌ها استفاده شده است. این کار مقادیر Math\_Score را به مقیاس استاندارد (میانگین 0 و انحراف معیار 1) تبدیل می‌کند. استانداردسازی برای مدل‌هایی مانند GMM مهم است، زیرا این مدل‌ها به توزیع داده‌ها حساس هستند. در خط بعد یک مدل GMM با 2 خوشه (n\_components=2) ایجاد شده و با داده‌های استانداردشده (X\_scaled) آموزش داده شده است. random\_state=42برای اطمینان از تکرارپذیری نتایج استفاده شده است. در آخرین خط لگاریتم احتمال (log-likelihood) هر نمونه را تحت مدل GMM محاسبه می‌کند. این مقادیر نشان‌دهنده میزان تطابق هر نمونه با مدل هستند. کد آن نیز به شرح زیر است.



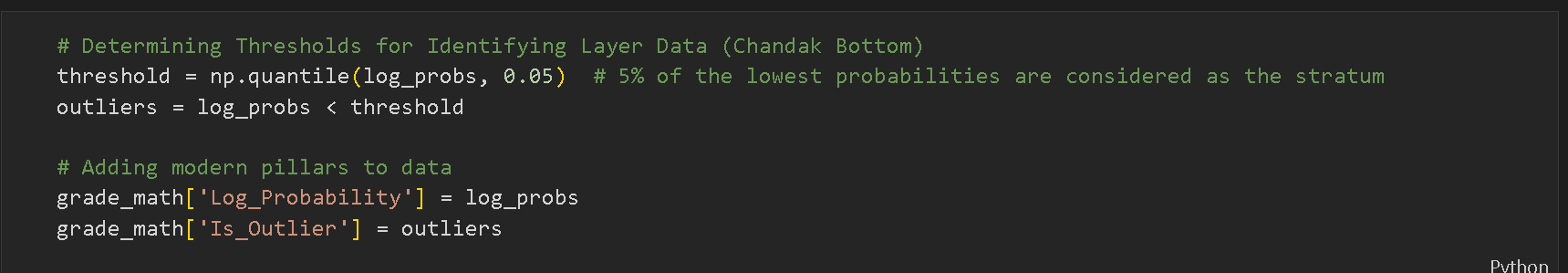
در بخش بعدی می خواهیم آنومالی با لگاریتم احتمال حساب کنیم.

در خط اول در اینجا، آستانه (threshold) با استفاده از تابع np.quantile محاسبه می‌شود. این تابع صدک 5٪ (0.05) از آرایه log\_probs را پیدا می‌کند. به عبارت دیگر، 5٪ از داده‌هایی که کمترین لگاریتم احتمال (log-likelihood) را دارند، به عنوان داده‌های غیرعادی یا متعلق به لایه خاص) در اینجا به عنوان لایه چندک پایین یا Chandak Bottom اشاره شده( در نظر گرفته می‌شوند. این آستانه نشان‌دهنده نقطه‌ای است که داده‌های با احتمال کمتر از آن به عنوان غیرعادی شناسایی می‌شوند.

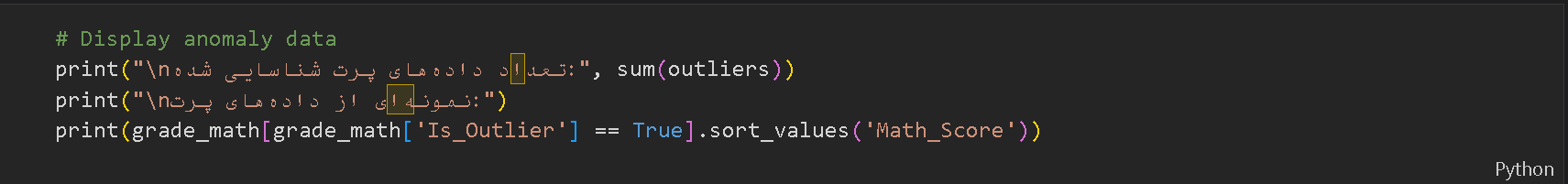
در خط دوم ین خط یک آرایه بولی (True/False) ایجاد می‌کند که مشخص می‌کند کدام نمونه‌ها لگاریتم احتمالی کمتر از آستانه دارند. اگر log\_probs[i] < threshold باشد، مقدار True به آن نمونه اختصاص داده می‌شود، به این معنی که آن نمونه به عنوان غیرعادی (outlier) شناسایی شده است.

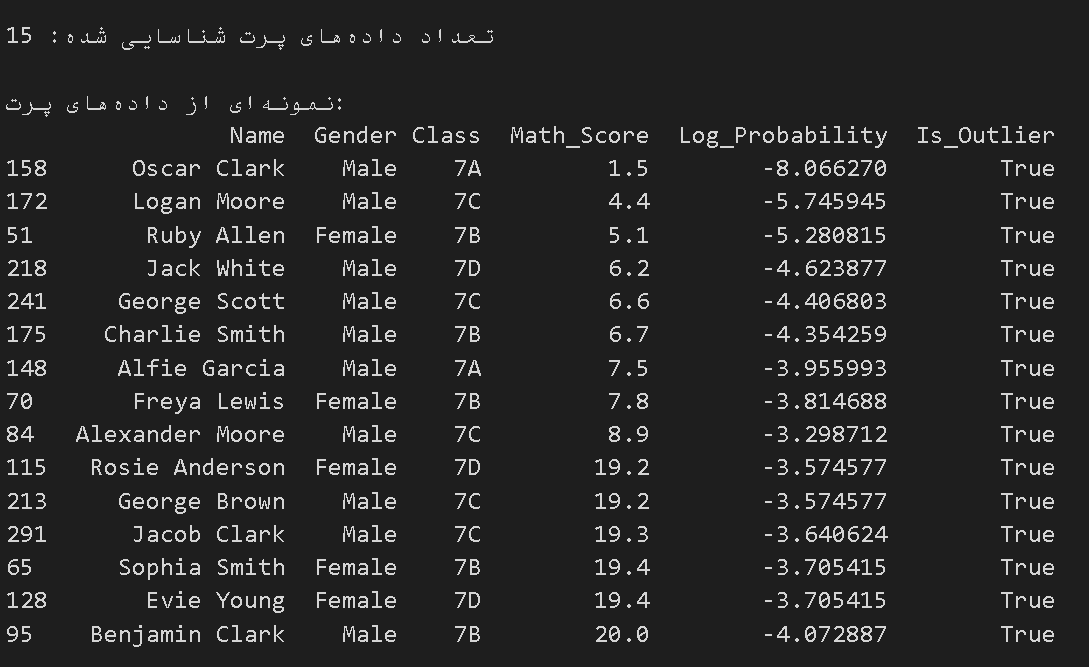
در خط سوم ستون جدیدی به نام Log\_Probabilityبه دیتافریم grade\_math اضافه می‌شود که حاوی مقادیر لگاریتم احتمال برای هر نمونه است. این ستون به شما امکان می‌دهد میزان تطابق هر نمونه با مدل GMM را بررسی کنید.

در خط چهارم ستون دیگری به نام Is\_Outlierبه دیتافریم اضافه می‌شود که نشان می‌دهد آیا هر نمونه غیرعادی است یا خیر True) برای غیرعادی، False برای عادی( این ستون به تحلیل داده‌ها کمک می‌کند تا نمونه‌های غیرعادی به راحتی شناسایی شوند. در ادامه کد آن را در زیر می بینید



در بخش بعدی داده های پرت را نمایش می دهیم تعداد آن ها و خود آن ها را نمایش می دهیم. کد و خروجی آن را می بینید



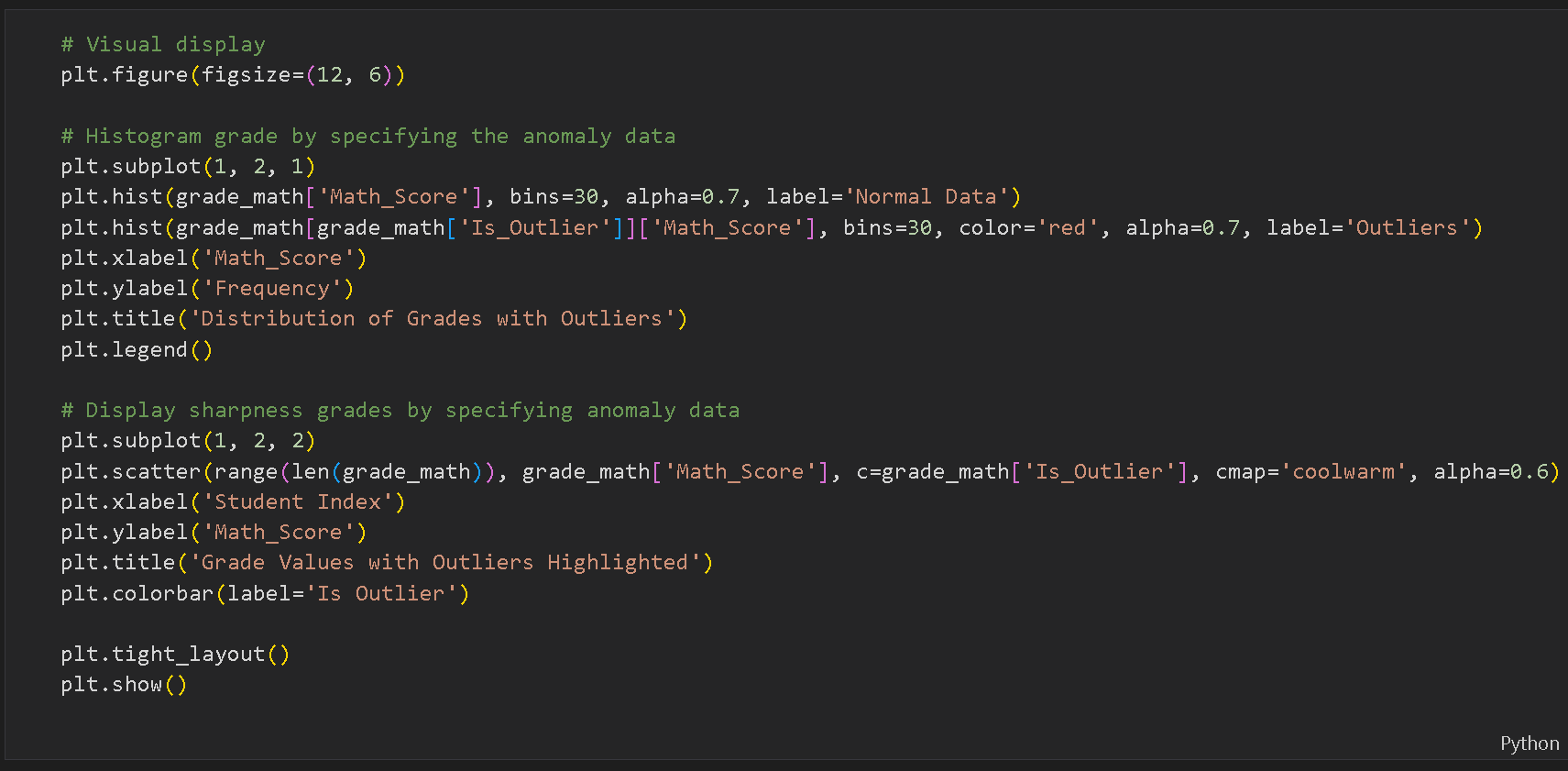


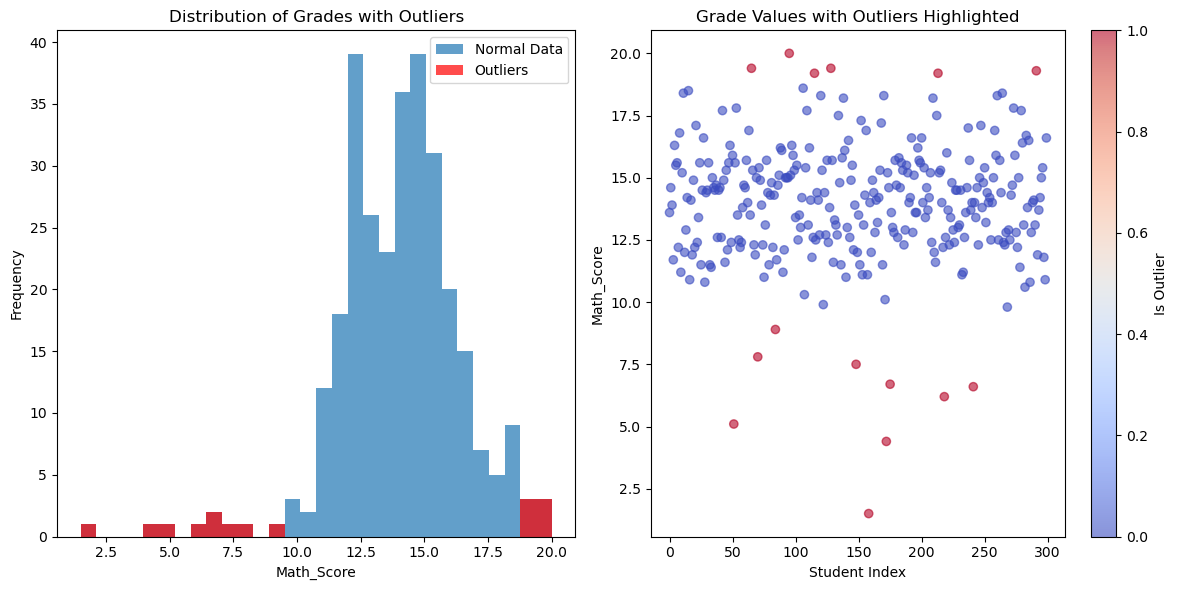
در انتهای می خواهیم این آنومالی ها را در دو نمودار نمایش دهیم. ابتدا اندازه صفحه خروجی نمودار مشخص کرده بعد subplot تعریف کرده دو نمودار یکی هیستوگرام و دیگری scatterplot هست جاشون در خروجی مشخص کرده در قسمت مربوط به کد histogram مشخص داده های نرمال ستونشون آبی و داده های آنومالی ستون هاش قرمز است بعد کشیدن hist لیبیل و تیتر نمودار مشخص می کند.

در بخش مربوط به scatterplot داده های آنومالی را با قرمز و داده های نرمال با آبی نشان می دهد

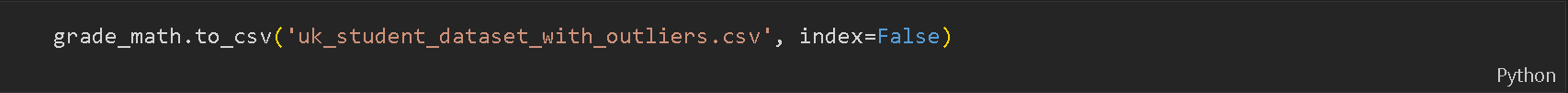
دستور plt.tight\_layout() در Matplotlib برای تنظیم خودکار فاصله‌ها و چیدمان عناصر یک نمودار (مانند زیرنمودارها، برچسب‌ها، عنوان‌ها و غیره) استفاده می‌شود تا از همپوشانی (overlap) جلوگیری کرده و نمودار خواناتر شود. در انتها با دستور plt.show() نمودار نمایش می دهیم.

خروجی و کد این visualization را در صفحه بعد ببینید





در انتهای کدی می زنیم دیتاستی جدید بسازد که این مدل کدام دیتاها را به عنوان آنومالی در نظر گرفته



**مفاهیم Isolation forest**

**Isolation Forest** یک الگوریتم یادگیری ماشین برای شناسایی ناهنجاری‌ها (Anomaly Detection) است که بر اساس مفهوم جداسازی داده‌ها کار می‌کند. این الگوریتم به‌ویژه برای شناسایی داده‌های غیرعادی یا پرت (Outliers) در مجموعه‌های داده‌ای بزرگ و چندبعدی مناسب است.

**ایده اصلی Isolation Forest**

* این الگوریتم فرض می‌کند که داده‌های ناهنجار (پرت) به دلیل ویژگی‌های خاص خود، راحت‌تر از داده‌های عادی جدا می‌شوند.
* از ساختار درختی (مشابه درخت تصمیم‌گیری) به نام **درخت‌های ایزولاسیون** (Isolation Trees) استفاده می‌کند.
* داده‌های پرت معمولاً در مراحل اولیه تقسیم‌بندی درخت جدا می‌شوند، زیرا تعداد کمتری از آن‌ها وجود دارد و ویژگی‌هایشان با اکثریت داده‌ها متفاوت است.

**چگونه کار می‌کند؟**

1. **ساخت درخت‌های ایزولاسیون**:
   * به‌صورت تصادفی یک ویژگی (Feature) و یک مقدار آستانه (Threshold) برای تقسیم داده‌ها انتخاب می‌شود.
   * داده‌ها به‌صورت بازگشتی تقسیم می‌شوند تا هر داده در یک گره جداگانه قرار گیرد.
   * این فرآیند چندین بار تکرار می‌شود تا مجموعه‌ای از درخت‌ها (Forest) ساخته شود.
2. **امتیازدهی به ناهنجاری‌ها**:
   * داده‌ای که زودتر در درخت جدا می‌شود (یعنی طول مسیر آن در درخت کوتاه‌تر است)، احتمالاً یک ناهنجاری است.
   * امتیاز ناهنجاری (Anomaly Score) بر اساس میانگین طول مسیر در درخت‌ها محاسبه می‌شود. امتیاز نزدیک به ۱ نشان‌دهنده ناهنجاری و نزدیک به ۰ نشان‌دهنده داده عادی است.

**مزایا**

* **سرعت بالا**: به‌ویژه برای داده‌های بزرگ، چون نیازی به محاسبات پیچیده ندارد.
* **مقیاس‌پذیری**: برای داده‌های چندبعدی و حجم بالا مناسب است.
* **نیازی به فرض توزیع داده**: برخلاف برخی روش‌ها، نیازی به فرض نرمال بودن داده‌ها ندارد.
* **مصرف حافظه کم**: به دلیل ساختار درختی ساده.

**معایب**

* **عملکرد ضعیف در داده‌های خیلی پیچیده**: اگر ناهنجاری‌ها به داده‌های عادی بسیار نزدیک باشند، ممکن است تشخیص سخت شود.
* **حساسیت به پارامترها**: انتخاب تعداد درخت‌ها یا نمونه‌های تصادفی می‌تواند روی دقت تأثیر بگذارد.

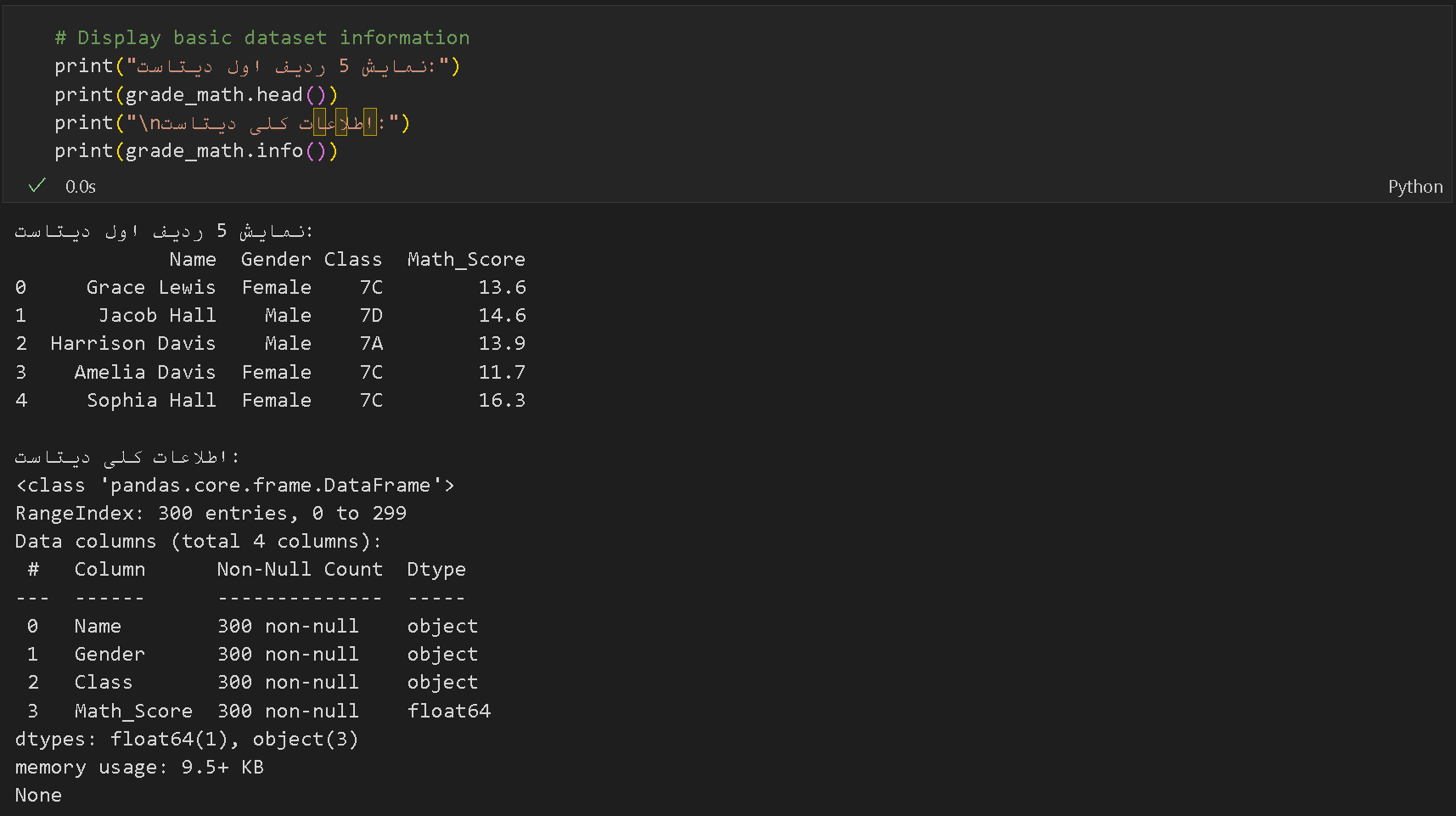
**کاربردها**

1. **تشخیص تقلب (Fraud Detection)**: شناسایی تراکنش‌های غیرعادی در سیستم‌های بانکی یا مالی.
2. **امنیت سایبری**: شناسایی حملات یا نفوذهای غیرعادی در شبکه.
3. **تشخیص خطا در سیستم‌های صنعتی**: شناسایی خرابی‌های تجهیزات یا خطاهای تولید.
4. **پزشکی**: شناسایی داده‌های غیرعادی در داده‌های پزشکی، مانند نتایج آزمایش غیرعادی.
5. **تحلیل داده‌های بازاریابی**: شناسایی رفتارهای غیرعادی مشتریان.

حال به بررسی کد زده شده آن می پردازیم.

ابتدا library های مدنظرم برای اجرای مدل isolation forest را import کنیم. مثل قبلی هاnumpy , pandas , matplotlib را import می کنیم. علاوه بر این ها seaborn برای کشیدن نمودار استفاده کرده و از sklearn.ensemble برای import کردن isolationforest استفاده میشه و از sklearn.preprocessing برای import کردن labelencoder استفاده می شود. بعد از آن دیتاست را با pandas می خونیم.

در ابتدا می آیم اطلاعات کلی دیتاست را پرینت می کنیم کد و خروجی آن به شرح زیر است.



این کد بخشی از فرآیند استفاده از الگوریتم Isolation Forest برای شناسایی ناهنجاری‌ها (Anomalies) در یک مجموعه داده است. در اینجا، داده‌های کیفی به عددی تبدیل شده و سپس مدل Isolation Forest برای تشخیص داده‌های غیرعادی استفاده می‌شود. حالا خط به خط توضیح می‌دهم:

ابتدا سه خط اول کد بررسی کرده

: **LabelEncoder**از کتابخانه scikit-learn برای تبدیل داده‌های کیفی (مثل رشته‌های متنی) به مقادیر عددی استفاده می‌شود. این کار برای آماده‌سازی داده‌ها برای مدل‌های یادگیری ماشین ضروری است، چون مدل‌ها فقط با داده‌های عددی کار می‌کنند.

: **Gender\_encoded** ستون جدیدی به نام Gender\_encoded ایجاد می‌شود که در آن مقادیر ستون Math\_Score (نمره ریاضی) به مقادیر عددی تبدیل می‌شوند. **نکته**: به نظر می‌رسد اینجا یک اشتباه وجود دارد، چون معمولاً LabelEncoder برای تبدیل متغیرهای کیفی (مثل جنسیت: "مرد"/"زن") استفاده می‌شود، نه برای ستون عددی مثل Math\_Score. احتمالاً منظور این بوده که ستون Gender (جنسیت) به عددی تبدیل شود.

:**Class\_encoded**ستون )Class مثلاً کلاس‌های A، B، (Cبه مقادیر عددی تبدیل می‌شود. مثلاً اگر Classشامل مقادیر "A"، "B"، "C" باشد، ممکن است به 0، 1، 2 تبدیل شوند.

به بخش دوم کد می پردازیم.

: **features** لیستی از ستون‌هایی که می‌خواهیم در مدل استفاده کنیم. اینجا سه ویژگی انتخاب شده: Gender\_encoded (جنسیت کدگذاری‌شده)، Class\_encoded (کلاس کدگذاری‌شده) و )Math\_Scoreنمره ریاضی.(

: **X** یک زیرمجموعه از داده‌ها (grade\_math) شامل فقط ستون‌های انتخاب‌شده (features) ایجاد می‌شود. این داده‌ها به‌عنوان ورودی مدل استفاده خواهند شد.

به بخش بعدی کد که مربوط به آموزش مدل می پردازیم.

**:IsolationForest** مدل Isolation Forest از کتابخانه scikit-learn ایجاد می‌شود.

:**contamination=0.05**این پارامتر مشخص می‌کند که انتظار داریم 5٪ از داده‌ها ناهنجار باشند. این مقدار به مدل کمک می‌کند تا حساسیت خود را برای شناسایی ناهنجاری‌ها تنظیم کند.

:**random\_state=42**برای اطمینان از تکرارپذیری نتایج، یک مقدار ثابت برای تولید اعداد تصادفی تنظیم شده است.

:**model.fit(X)** مدل روی داده‌های X آموزش داده می‌شود. این مرحله شامل ساخت درخت‌های ایزولاسیون برای شناسایی ناهنجاری‌ها است.

بخش بعدی کد مربوط به محاسبه امتیاز یا ناهنجاری و پیش بینی است.

:**anomaly\_score**ستون جدیدی به داده‌ها اضافه می‌شود که شامل **امتیاز ناهنجاری** برای هر داده است. این امتیاز از تابع decision\_functionبه دست می‌آید. امتیازهای نزدیک به صفر یا منفی نشان‌دهنده احتمال بیشتر ناهنجاری هستند.

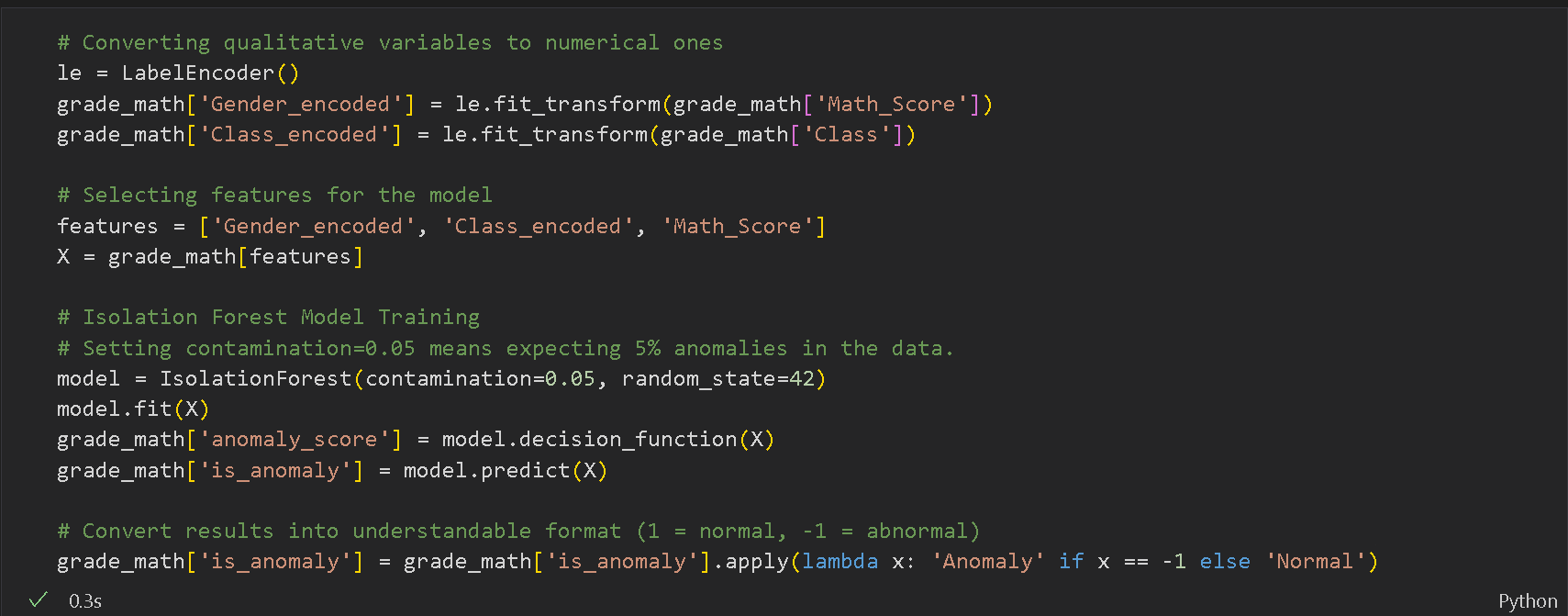
:**is\_anomaly**ستون دیگری اضافه می‌شود که نتیجه پیش‌بینی مدل را نشان می‌دهد. مدل برای هر داده یک برچسب تولید می‌کند:

* :1 داده عادی است.
* :-1داده ناهنجار است.

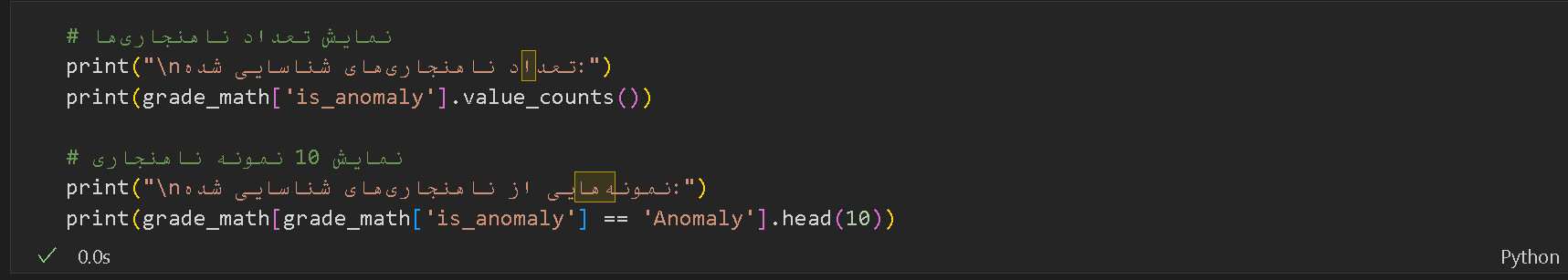
بخش آخرکد مربوط به تبدیل نتایج به فرمت قابل فهم است

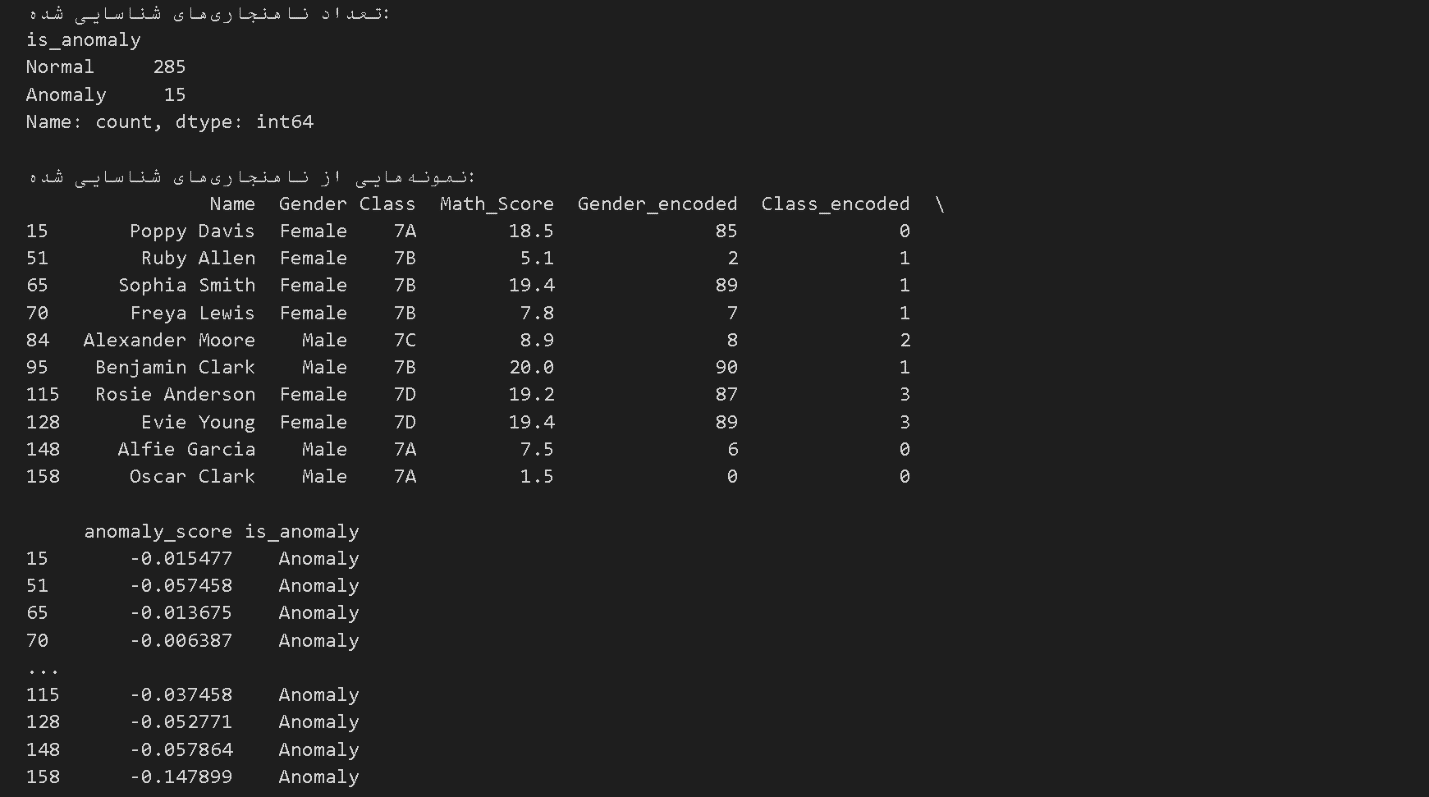
* این خط مقادیر ستون is\_anomaly را از عددی (1 یا -1) به رشته‌های قابل‌فهم (Anomaly یا Normal) تبدیل می‌کند.
* **:lambda x: 'Anomaly' if x == -1 else 'Normal'** یک تابع ساده که اگر مقدار -1 باشد، برچسب Anomaly و اگر 1 باشد، برچسب Normal را اختصاص می‌دهد.

کد این چند بخشی که توضیح دادم به شرح زیر است.



قسمت بعد نتایج این مدل را منتشر می کنیم این نتایج شامل تعداد آنومالی ها و بخشی از آنومالی ها را نمایش داده کد و خروجی آن را در صفحه بعد خواهید دید.





در مرحله آخر مربوط به کشیدن نمودار هاست. می خواهیم آنومالی ها را با نمودار نمایش دهیم.

ابتدا با plt.figure اندازه نمودار تنظیم می کنیم. بعدscatterplot مربوط به کتابخانه seaborn را می کشیم

**: sns.scatterplot** از کتابخانه **Seaborn** برای رسم نمودار پراکندگی استفاده می‌کند.

* **Data = grade\_math**: داده‌ها از دیتا فریم grade\_mathخوانده می‌شوند.
* **: x='Math\_Score'**محور افقی (X) نشان‌دهنده نمره ریاضی دانش‌آموزان است.
* **: y='Class'**محور عمودی (Y) نشان‌دهنده کلاس دانش‌آموزان (مثل A، B، C) است.
* **: hue='is\_anomaly'**رنگ نقاط بر اساس ستون is\_anomaly تنظیم می‌شود. نقاطی که ناهنجار (Anomaly) هستند، رنگ متفاوتی با نقاط عادی (Normal) دارند.
* **: palette = {'Normal': 'blue', 'Anomaly': 'red'}**رنگ نقاط مشخص می‌شود: نقاط عادی به رنگ آبی و نقاط ناهنجار به رنگ قرمز.
* **: style = 'is\_anomaly'**شکل نقاط (مثل دایره، مربع) بر اساس ستون is\_anomaly تغییر می‌کند تا تفاوت بین نقاط عادی و ناهنجار واضح‌تر باشد.
* **: s=100**اندازه نقاط را به ۱۰۰ تنظیم می‌کند تا نقاط به‌راحتی دیده شوند.

در قسمت بعدی تیتر نمودار و لیبل محور x, y و اندازه این تیتر و لیبل را تنظیم کرده بعد با plt.legend راهنمای نمودار می سازیم و در نهایت توضیحات سه خط آخر کد قسمت اول به شرح زیر است

**: plt.grid(True)** خطوط شبکه‌ای به نمودار اضافه می‌کند تا خواندن مقادیر راحت‌تر باشد.

: **plt.tight\_layout()**فاصله‌بندی بین اجزای نمودار (مثل عنوان و محورها) را به‌صورت خودکار تنظیم می‌کند تا همه‌چیز مرتب باشد.

: **plt.show()**نمودار را نمایش می‌دهد.

اما بخش رسم boxplot هم یه سری قسمت های مشابه مثلplt.figure یا plt.show یا plt.title یا plt.xlabel یا plt.ylabel را داره و تنها بخش متفاوت مربوط ساخت boxplot با seaborn دقیق تر توضیح می دهم.

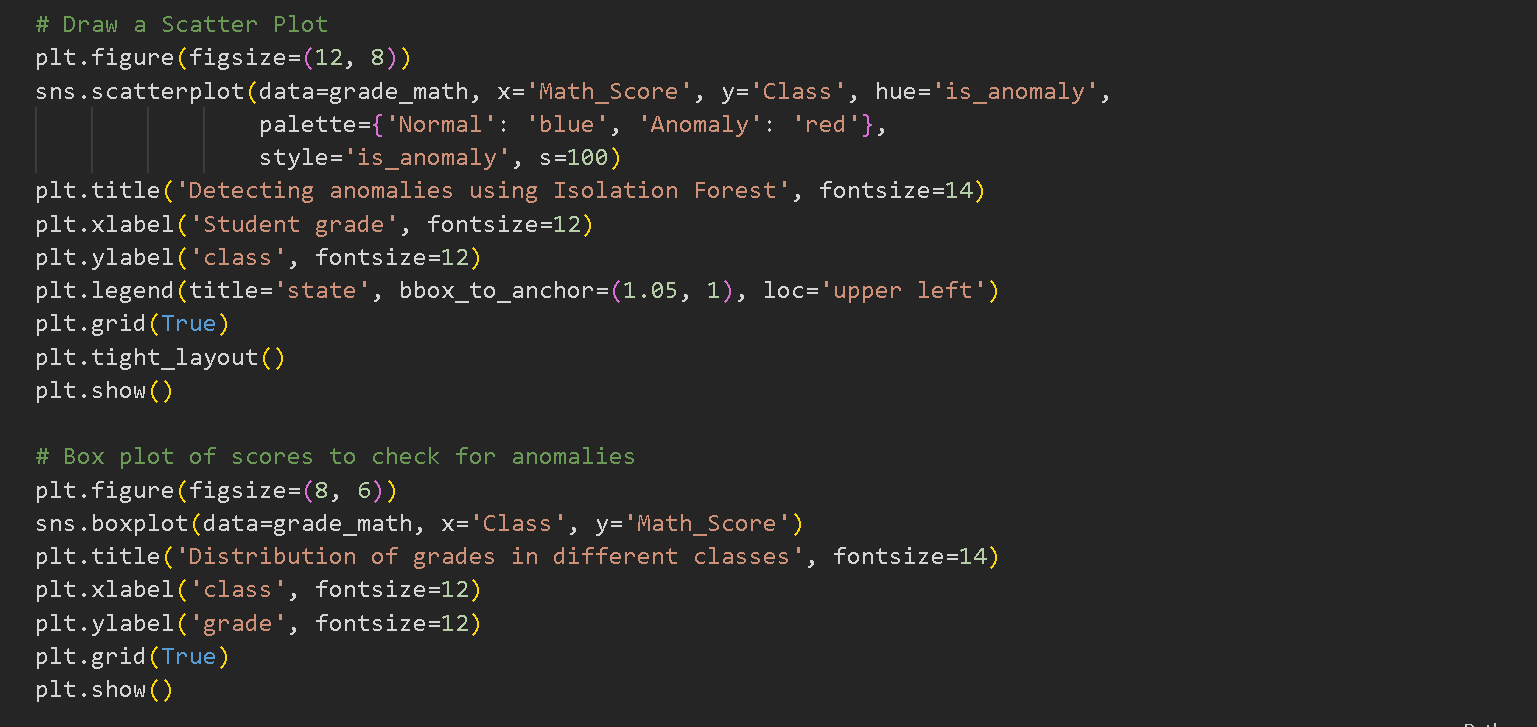
**sns.boxplot**: از **Seaborn** برای رسم نمودار جعبه‌ای استفاده می‌کند.

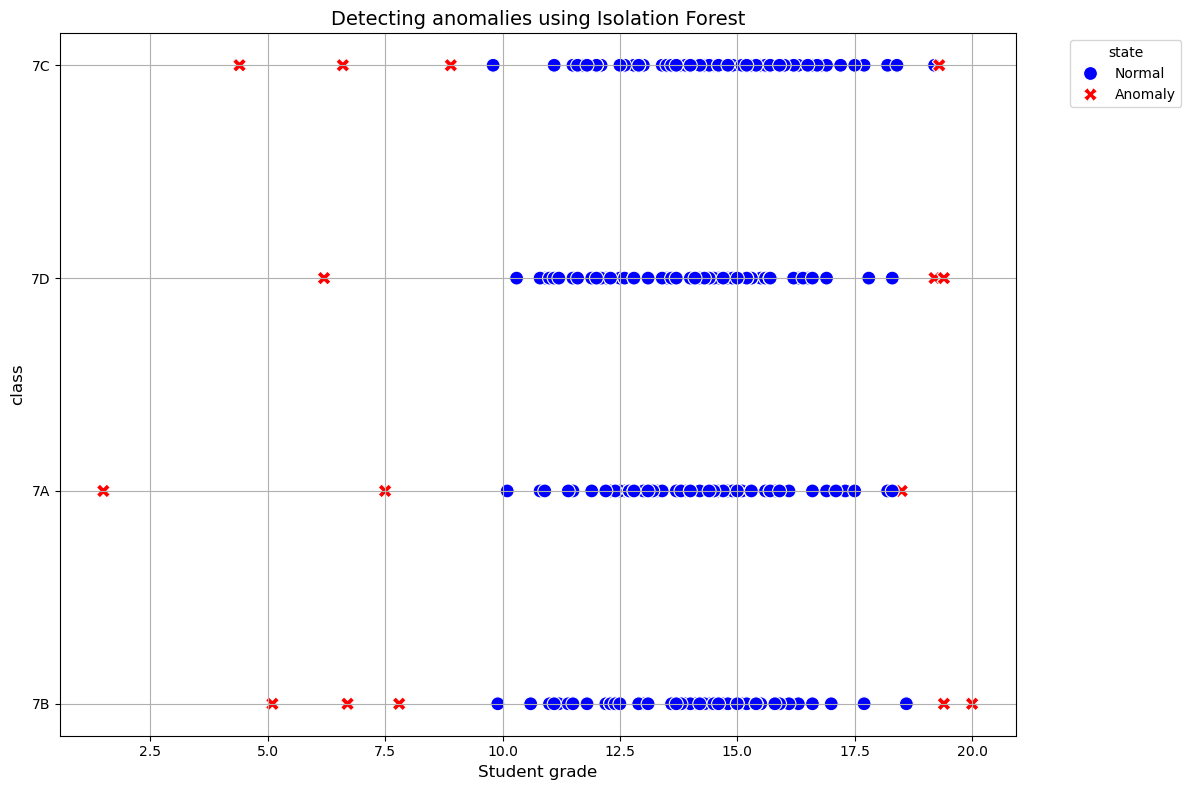
* **: data=grade\_math**داده‌ها از دیتافریم grade\_math گرفته می‌شوند.
* **: x='Class'**محور افقی (X) کلاس‌های مختلف مثل A)، B، (C را نشان می‌دهد.
* **: y='Math\_Score'** محور عمودی (Y) نمرات ریاضی را نشان می‌دهد.

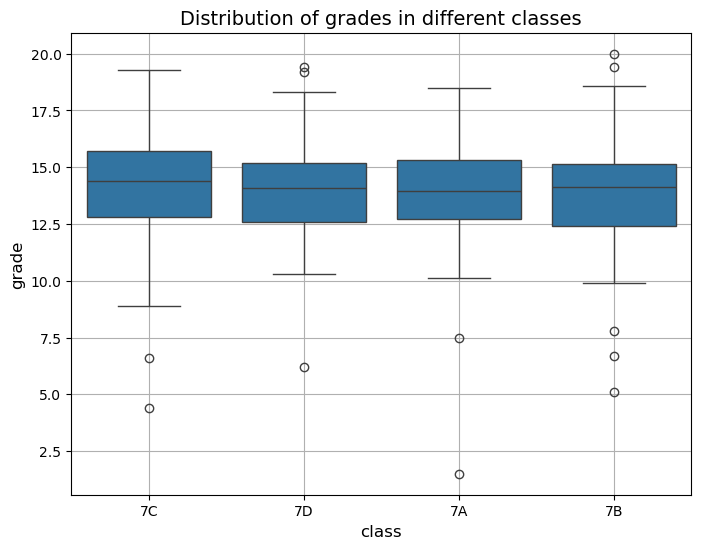
**نمودار جعبه‌ای**: توزیع نمرات را برای هر کلاس نشان می‌دهد و شامل:

* جعبه : نشان‌دهنده محدوده بین چارک اول (Q1) و چارک سوم (Q3).
* خط وسط جعبه : میانه (Median) نمرات.
* خطوط بیرون جعبه (Whiskers) : محدوده داده‌های غیرپرت.
* نقاط خارج از خطوط : داده‌های پرت (Outliers) که ممکن است با ناهنجاری‌های شناسایی‌شده توسط Isolation Forest مرتبط باشند.
* plt.title: عنوان نمودار را به «توزیع نمرات در کلاس‌های مختلف» با فونت ۱۴ تنظیم می‌کند.
* plt.xlabel: برچسب محور افقی را به «کلاس» با فونت ۱۲ تنظیم می‌کند.
* plt.ylabel: برچسب محور عمودی را به «نمره» با فونت ۱۲ تنظیم می‌کند.
* plt.grid(True): خطوط شبکه‌ای به نمودار اضافه می‌کند.
* plt.show: نمودار جعبه‌ای را نمایش می‌دهد.

کد و نتایج آن را در زیر مشاهده می کنید:



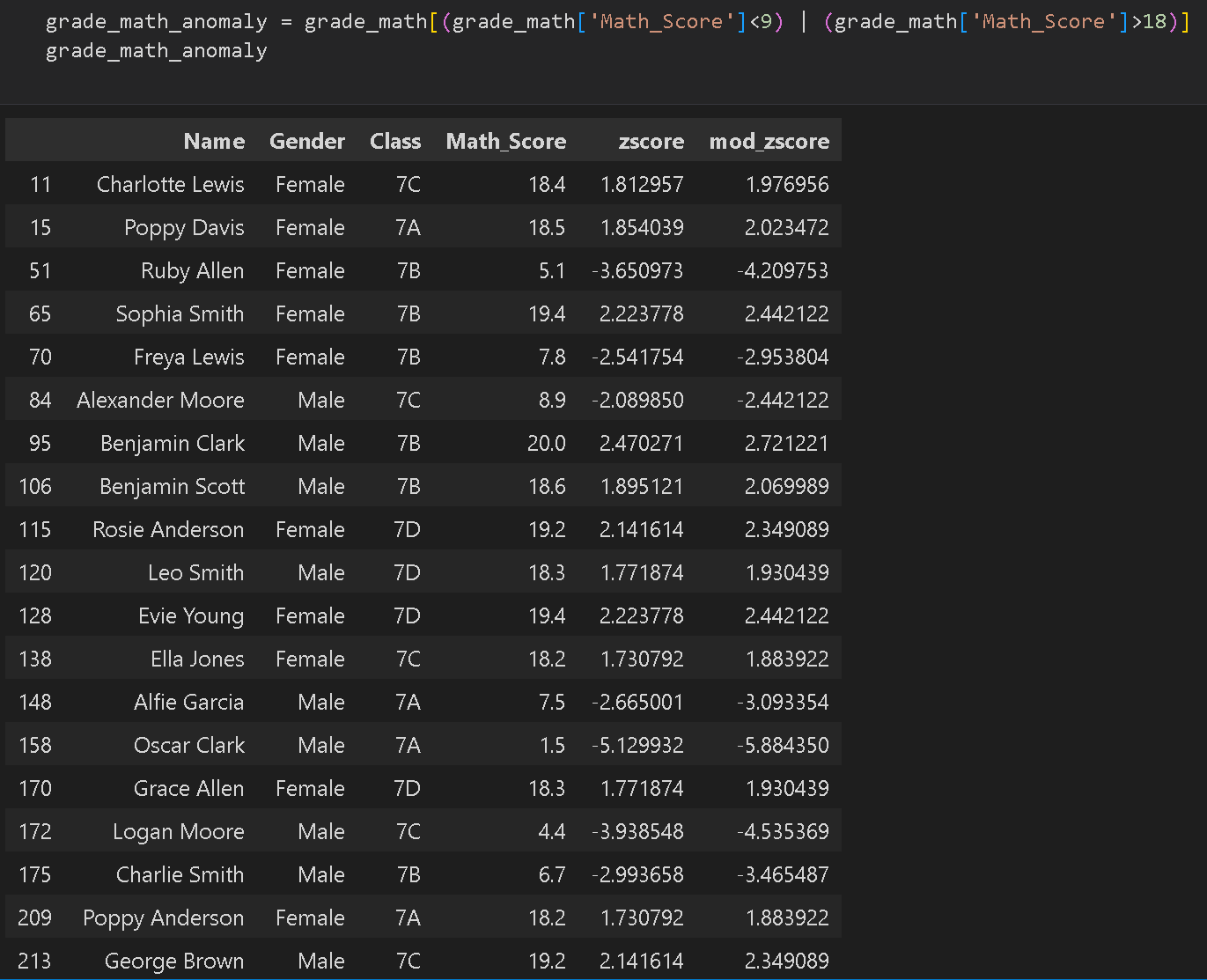




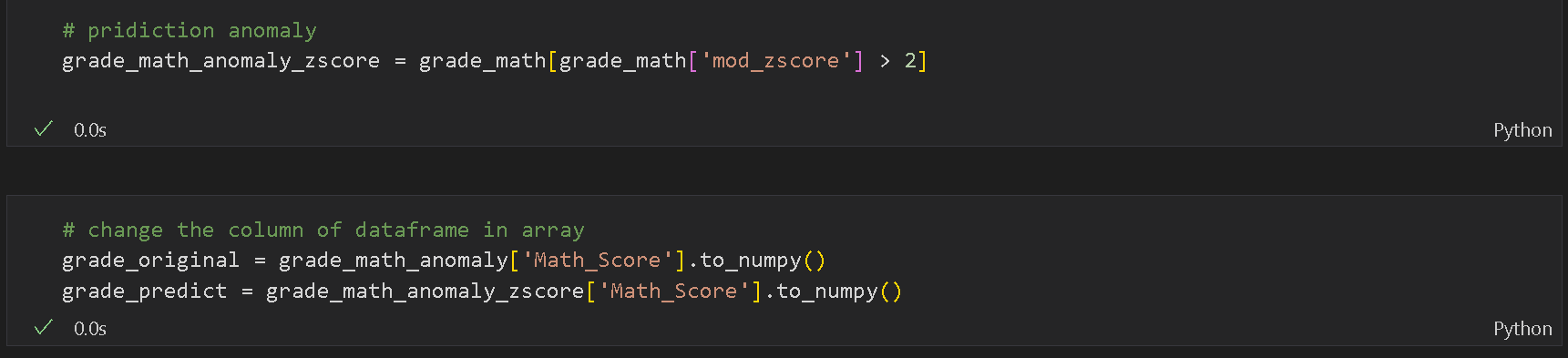
**نتایج و نتیجه گیری**

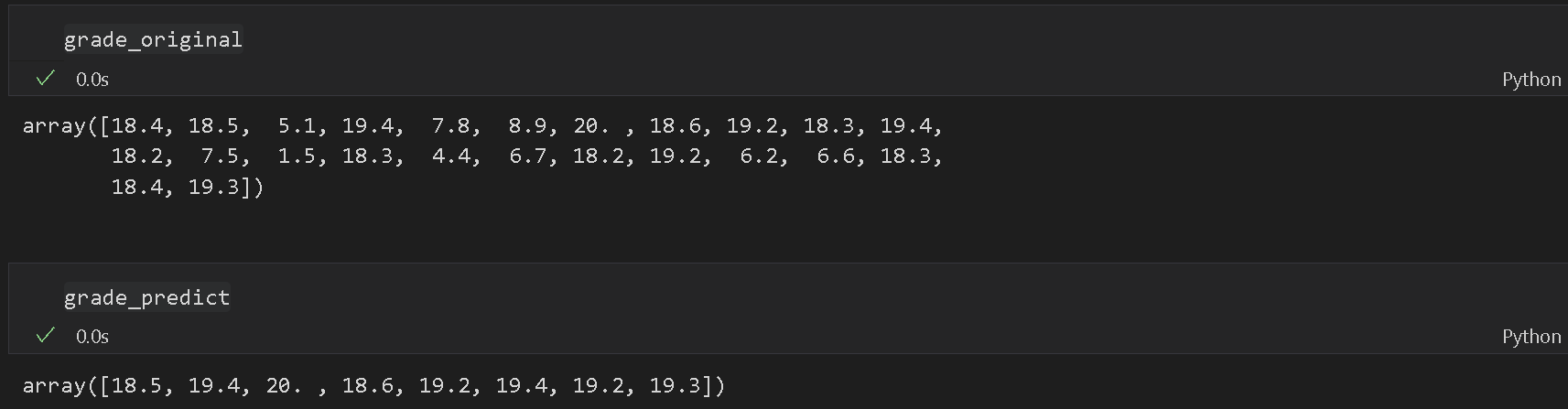
در این قسمت من آمدم درصد پیش بینی آنومالی موجود در دیتاست توسط هرکدام از مدل ها محاسبه کرده و بررسی کردیم کدام یک بهتر در پیش بینی آنومالی عمل کردند.

ابتدا نتایج z\_score بررسی کرده ابتدا آنومالی واقعی را بدست آورده کد آن را در زیر مشاهده می کنید

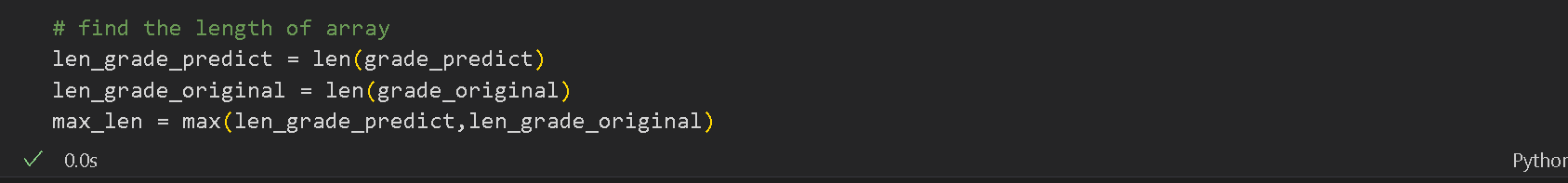


پس آن کل کاری ما انجام می دهیم برای گرفتن درصد پیش بینی را توضیح می دهد. ابتدا ستون Math\_Score در آنومالی های و اصلی و آنومالی پیش بینی شده را به آرایه با استفاده از Numpy و متد to\_numpy تبدیل کرده کد آن به شرح زیر است:

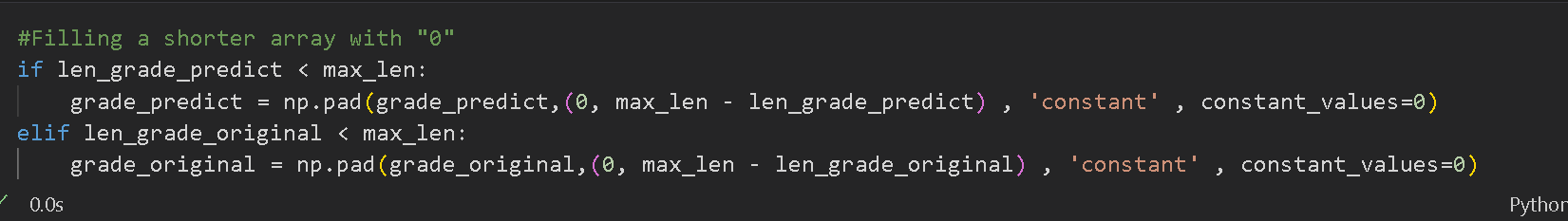




سپس طول آرایه ها را پیدا کرده و آن آرایه ای که طولش بیشتر بود به max\_len در نظر میگیریم. و کد آن به شرح زیر است:

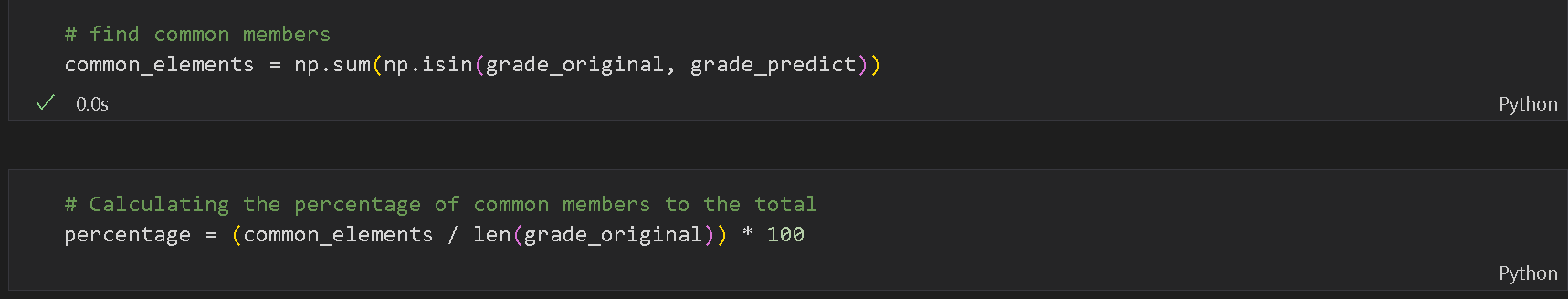


در مرجله بعد آن آرایه طولش کمتر را با تعدادی صفر پر کرده تا از نظر تعداد باهم برابر شوند این کار با if و elif انجام داده میگیم اگه طول آرایه آنومالی پیشبینی شده از max\_len کمتر بود به آرایه آنومالی پیش بینی شده به اندازه اختلاف max\_len و طول آرایه آنومالی پیشبینی شده مقدار 0 را اضافه کرد این اضافه کردن را با استفاده از Numpy و متد pad انجام میشود. متد pad دارای 4 ورودی است یکی آرایه که قرار است مقادیر به آن اضافه شود و دومی تعداد اعضایی که اضافه می شود و سومی نوع مقادیری که اضافه می شود در این حالت چون مقادیر ما ثابت از constant استفاده کرده و در نهایت مقدار ثابتی که اضافه می کنیم را مشخص می کنیم. کد آن به شرح زیر است:

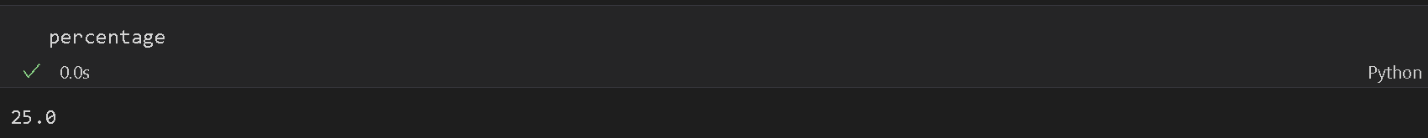


سرانجام برای **پیدا کردن اعضای مشترک باید** با استفاده از np.isin بررسی می‌کنیم که کدام عناصر x در y وجود دارند و تعداد آن‌ها را با np.sum محاسبه می‌کنیم.

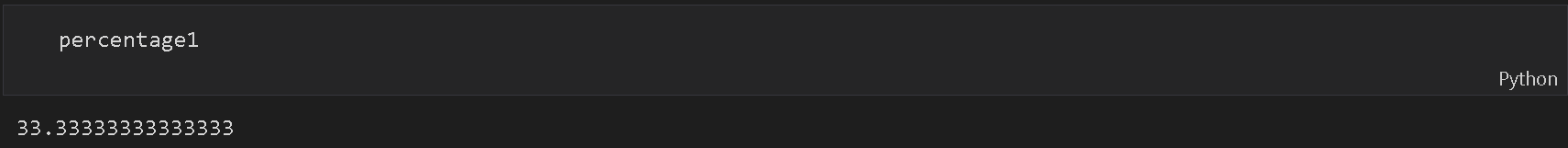
برای **محاسبه درصد** تعداد اعضای مشترک تقسیم بر طول آرایه y شده و در 100 ضرب می‌شود تا درصد به دست آید. کد آن به شرح زیر است



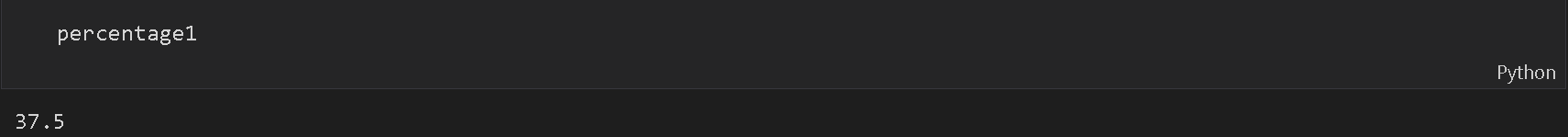
در نهایت درصد پیش بینی درست برای z\_score را نمایش می دهیم که در زیر آن را مشاهده می کنید.



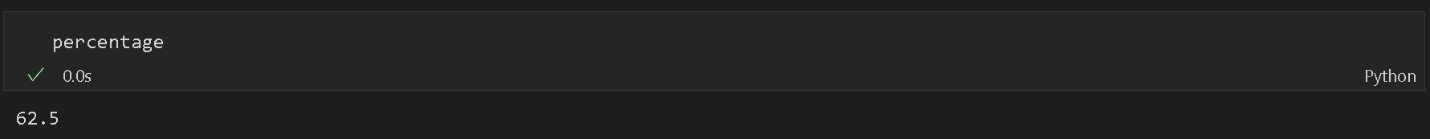
این درصد برای z\_score اصلاح شده به شرح زیر است:



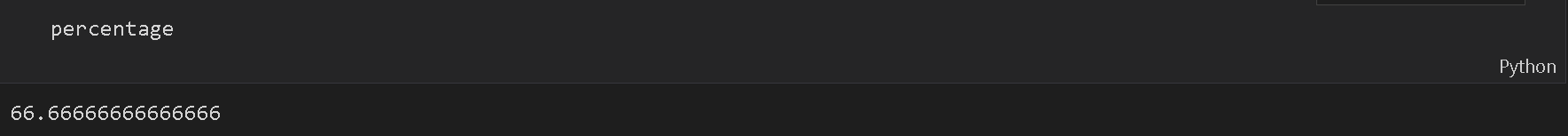
این درصد برای مدل IQR به شرح زیر است:



این درصد برای مدل gmm به شرح زیر است:



این درصد برای مدل isolationforest به شرح زیر است:



با توجه به نتایج می توان نتیجه گرفت استفاده از مدل isolation forest می توناد بهترین گزینه برای پیدا کردن آنومالی در دیتاست شبیه به دیتاست باشد.