نام خدا

تمرین سوم درس هوش مصنوعی

سوال دوم

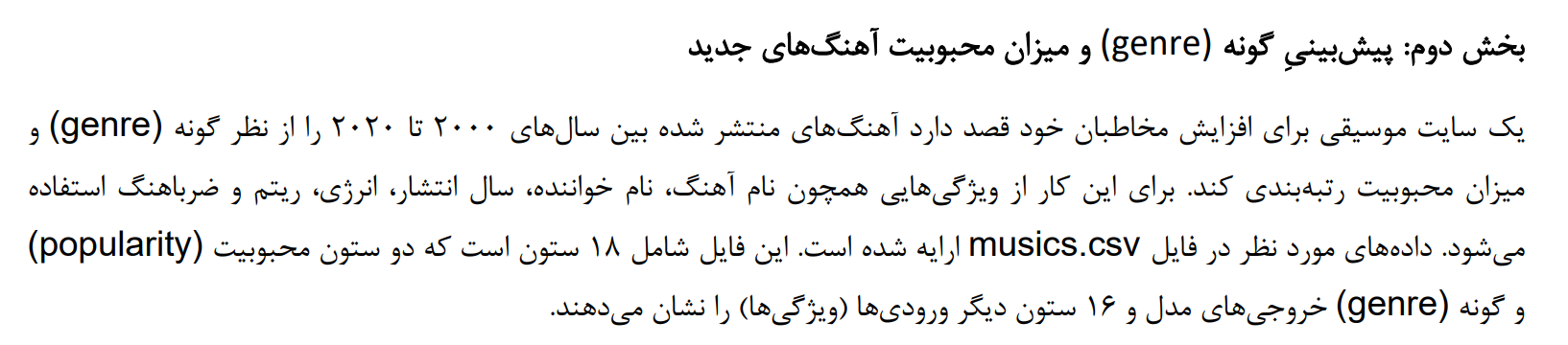
محمد امین سلطانی چم حیدری

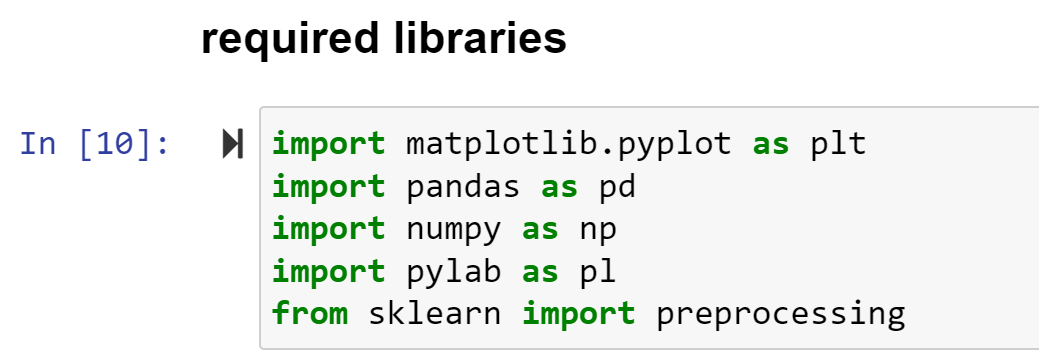
810601081

استاد مربوطه:

دکتر مسعود شریعت پناهی

بهار1402





به منظور تربیت مدل برای این سوال ابتدا کتابخانه های مورد نظر در Jupyter notebook فراخوانی شده اند.

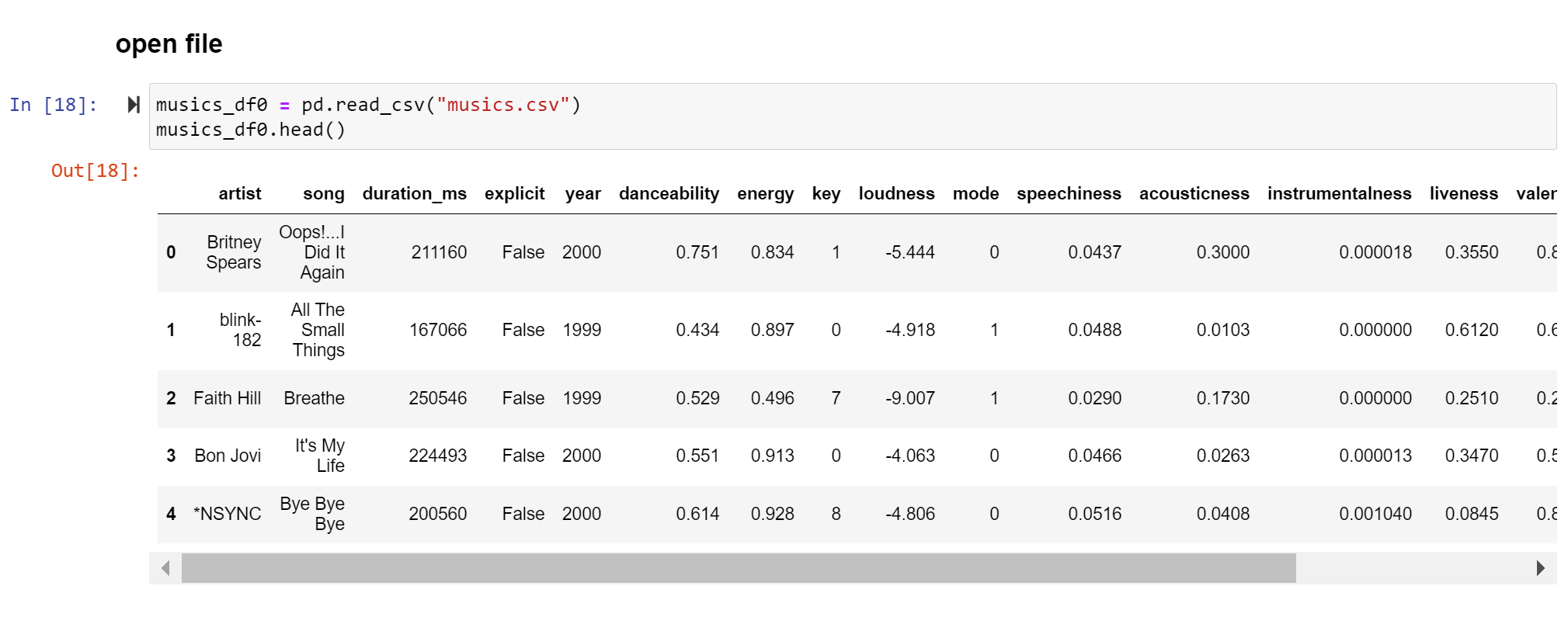
matplotlib 🡨 نمایش data و رسم نمودار

pandas 🡨 کار با داده های ساختار یافته مانند جداول،فایل csv و excel و...

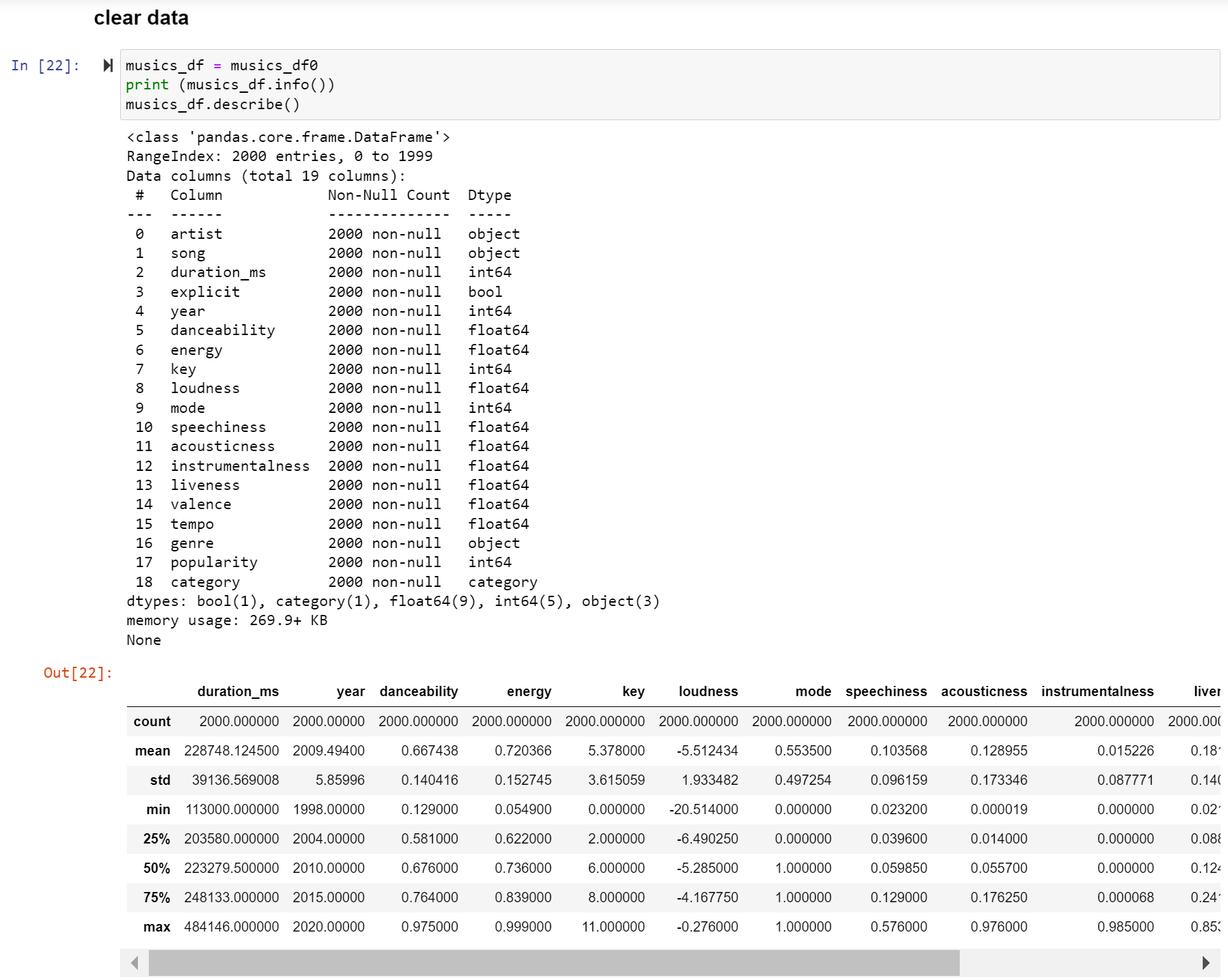
Numpy 🡨 انجام عملیات ریاضی،آماری و عددی

Pylab 🡨 تجسم داده های علمی و عملی.ترکیبی از matplotlib و numpy می باشد

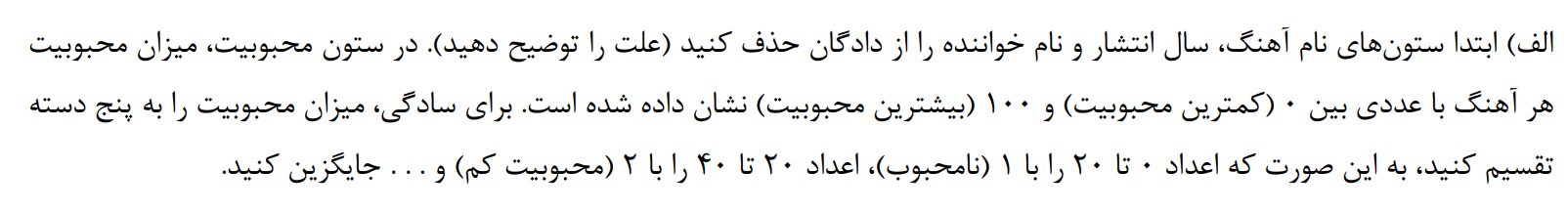
Sickit\_learn 🡨 یادگیری ماشین و تحلیل داده

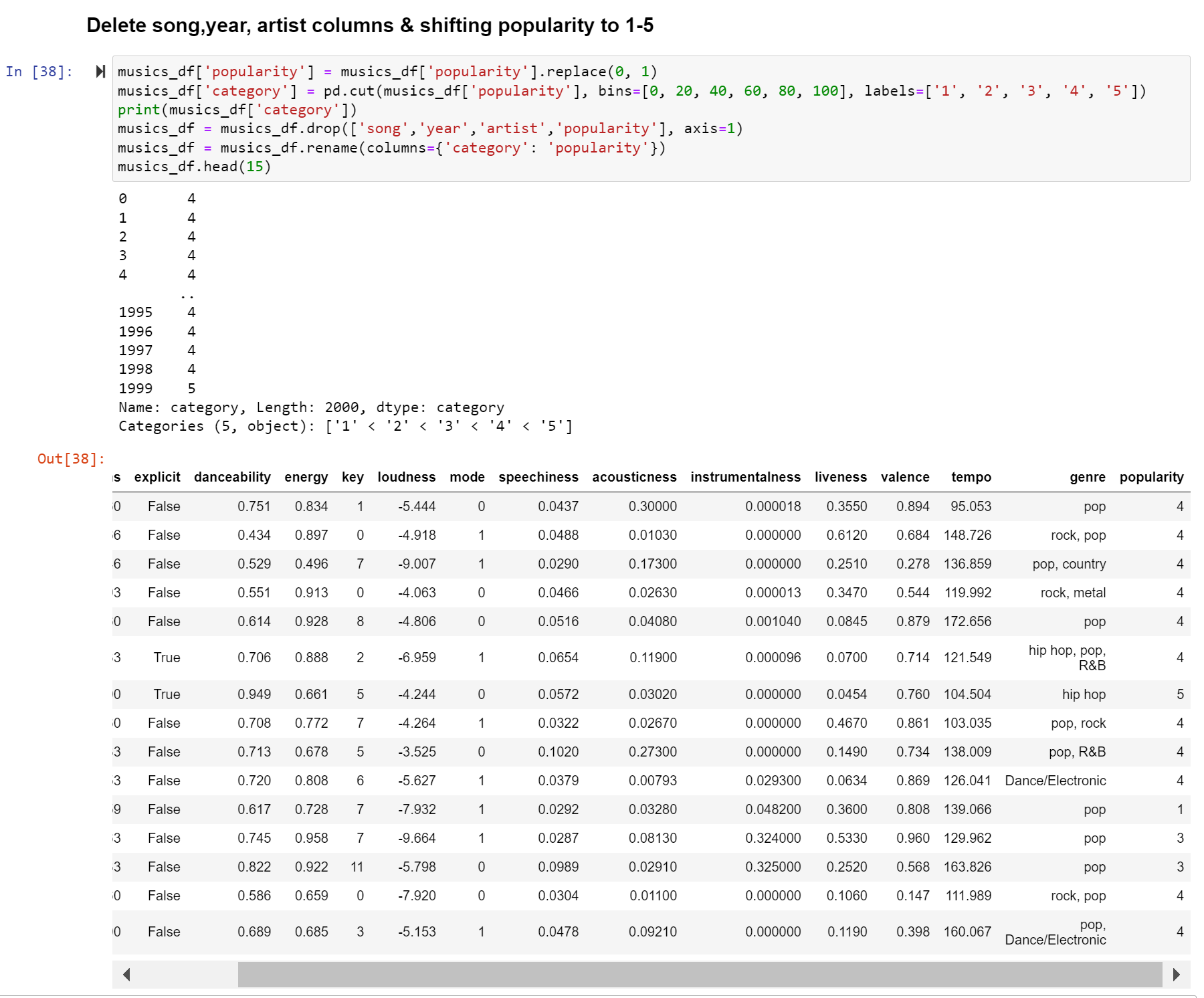


در مرحله بعد داده های مربوط به آهنگ های منتشر شده که شامل ویژگی های ورودی و خروجی ها می باشد به کمک دستور pd از کتابخانه pandas باز شده اند.



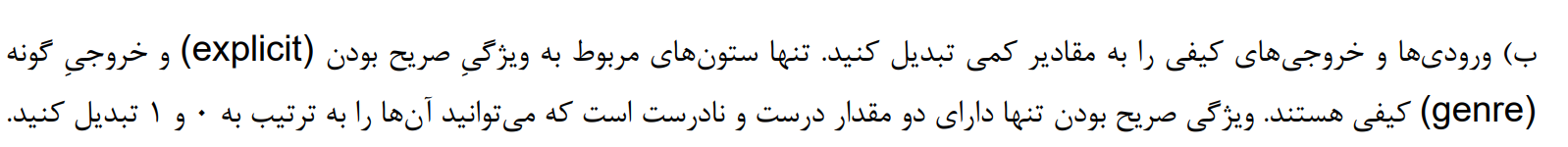
در قسمت بعد توصیفی از داده های ورودی مشاهده می شود. این مرحله به ان علت انجام شده که در صورت وجود هرگونه نقص و عیب در داده های ورودی سطر و ستون مربوط به ان داده ورودی حذف شود که البته در این سوال مشاهده می شود هیچ گونه نقصی در داده های ورودی وجود ندارد.

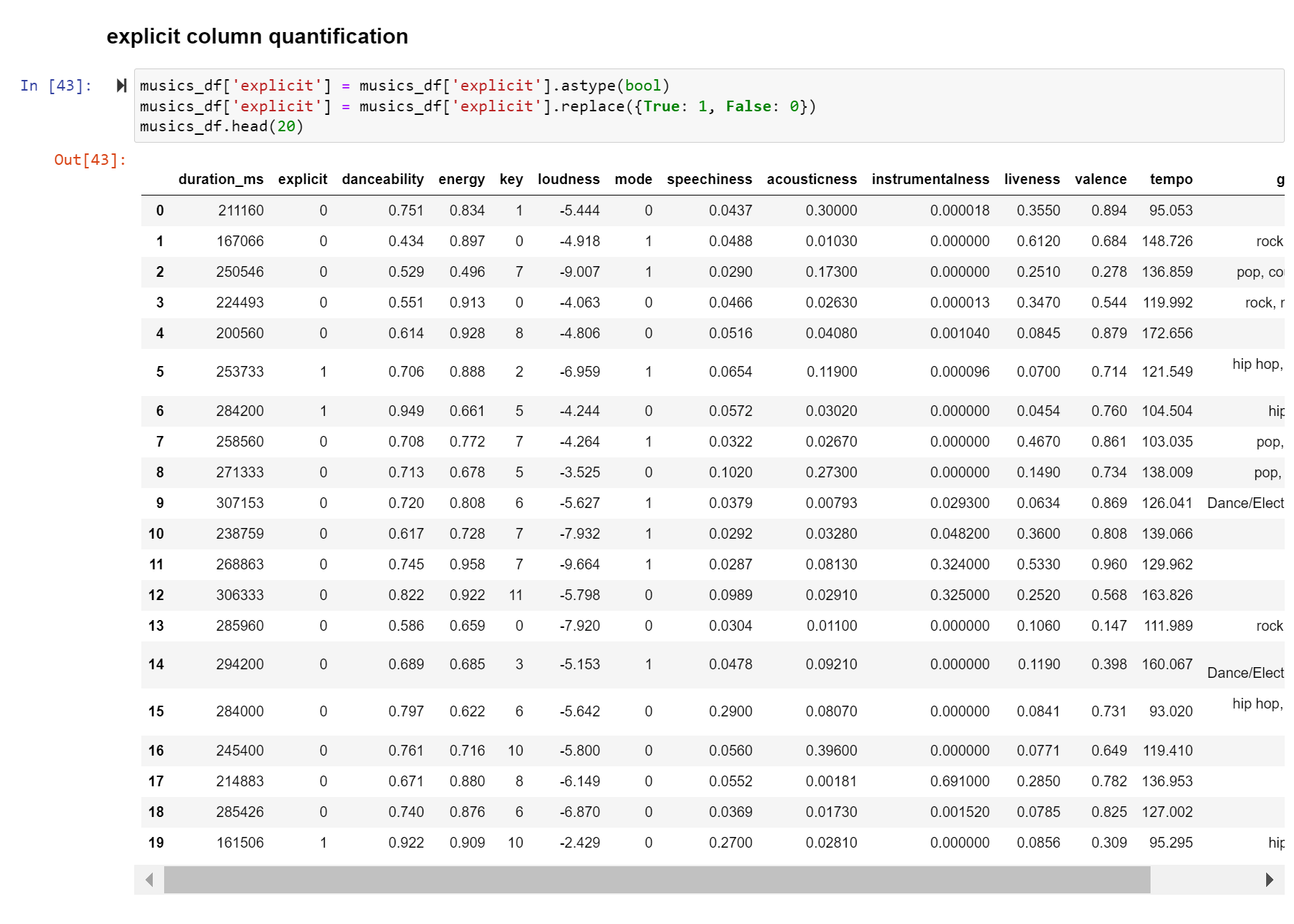




در این قسمت ستون های نام آهنگ، سال انتشار و نام خواننده حذف شده اند. علت این امر این است که برای مثال نام آهنگ در هر سطر مقدار متفاوتی دارد لذا پراکندگی داده های این سطر ها زیاد می باشد.پراکندگی زیاد داده ها و همچنین کیفی بودن این مقادیر موجب می شود این ستون ها در تربیت مدل موجب افزایش خطا شود و به طور کلی کمکی به تربیت بهتر مدل نکنند.

**همچنین در این قسمت ستون popularity که میزان محبوبیت را از 0 تا 100 نشان می داد به مقادیر 1 تا 5 شیفت شده است.**





**در مرحله اول از قست ب برای کمی سازی ستون explicit مطابق شکل بالا مقادیر این ستون به جای True , False با 1 , 0 جایگذاری می شوند.**

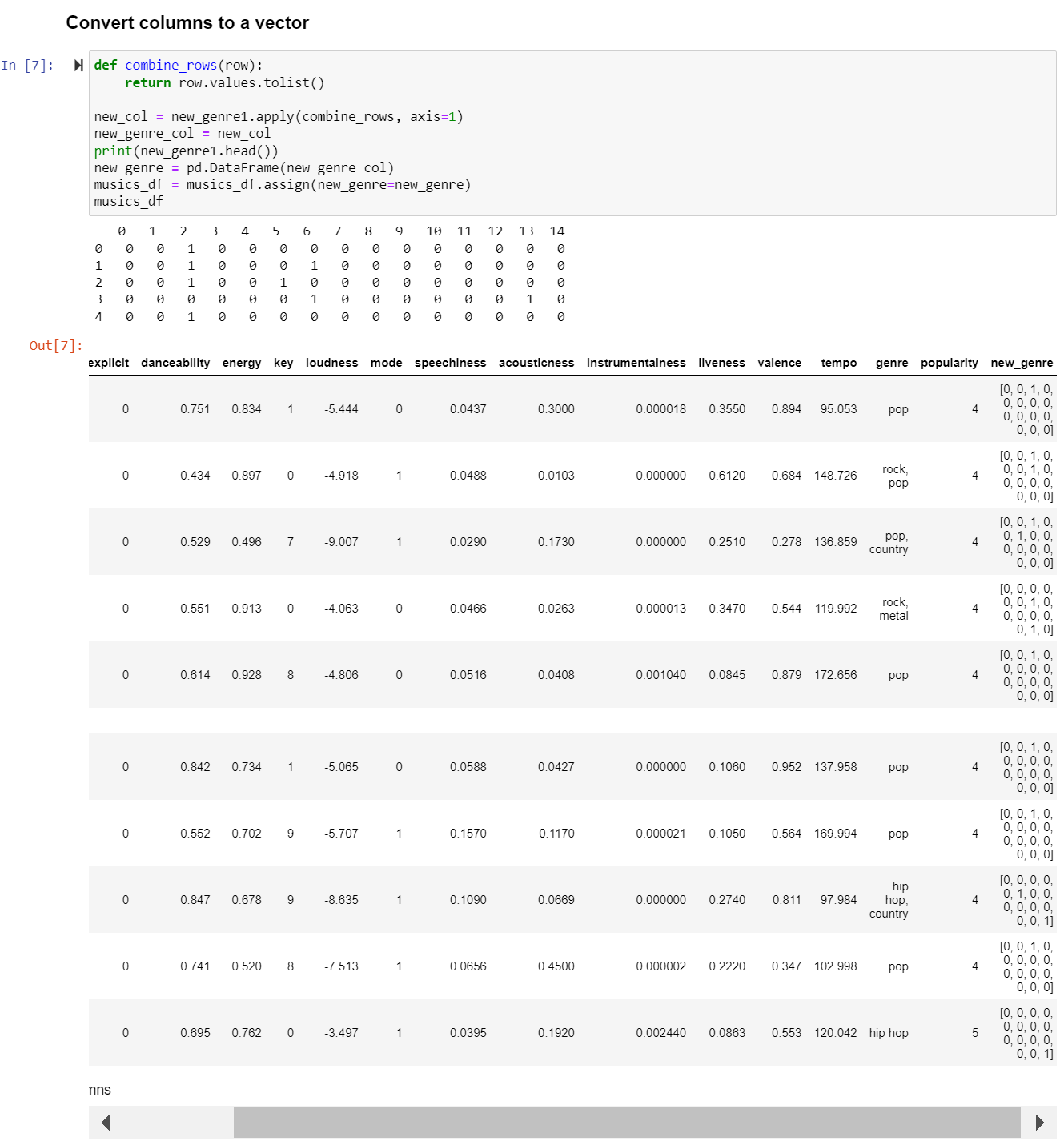


برای کمی سازی ستون genre با توجه به اینکه مقادیر این ستون مانند مرحله قبل از دو نوع تشکیل نشده اند نمی توان از روش قبل برای کمی سازی استفاده کرد.

در این قسمت برای کمی سازی ابتدا مقادیر این ستون که بین آن ها کاما می باشد جدا سازی می شوند. سپس یک فایل جدید تعریف می شود به این صورت که در هر سطر، ستون مربوط به آن مقدار کیفی برابر یک و سایر ستون ها برابر صفر قرار داده می شوند.

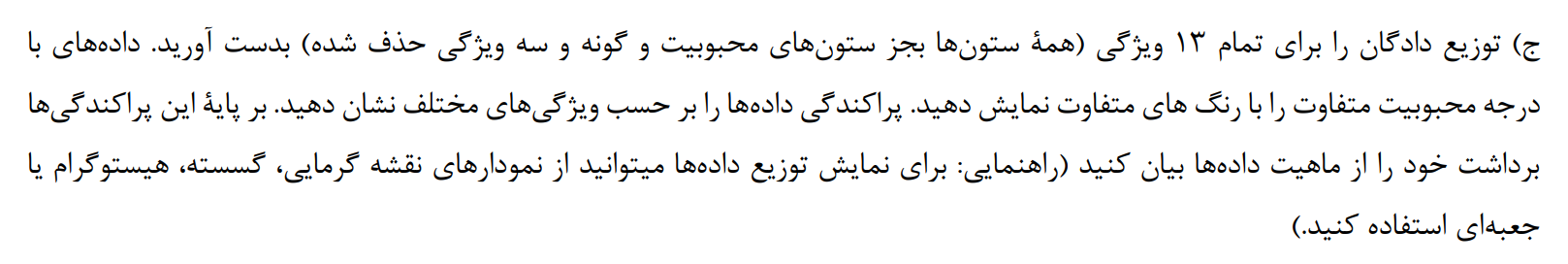
برای این کار می توان از دستور one hot encoding و دستور dummies یا مطابق شکل بالا از یک حلقه for استفاده کرد. از آن جایی که جداسازی داده های هر سطر نیز مد نظر می باشد در اینجا از دو حلقه for برای جداسازی و اعمال one hot encoding استفاده شده است.

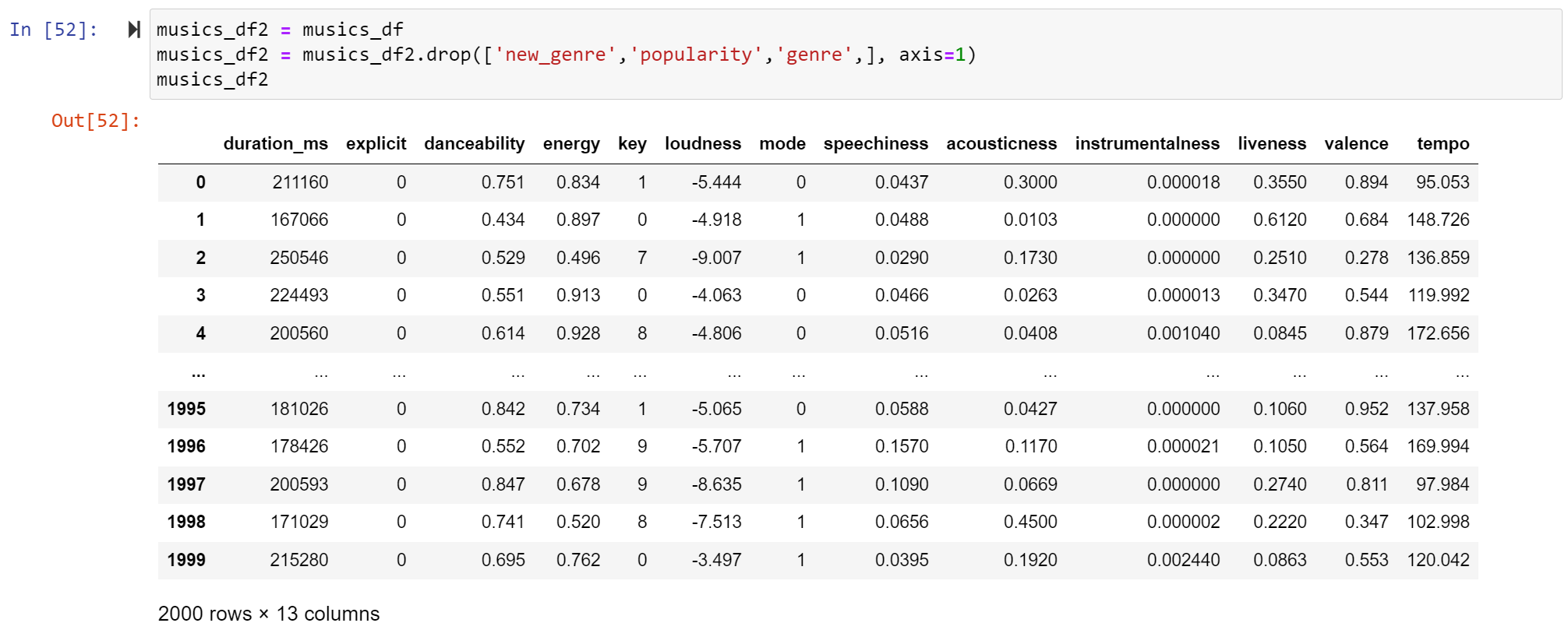
برای مثال در سطر اول چون genre آن pop است، ستون دوم که نشان دهنده pop می باشد برابر یک قرار داده شده و سایر ستون ها صفر هستند. یا در سطر سوم که شامل pop و country است مقادیر ستون دوم و پنجم برابر یک و سایر ستون ها برابر صفر قرار داده می شوند. به این ترتیب در هر سطر ستون مربوط به آن genre برابر یک قرار داده می شود. باتوجه به اینکه در فایل ورودی 15 ژانر متفاوت وجود دارد یک ماتریس 15 ستونه حاصل می شود.



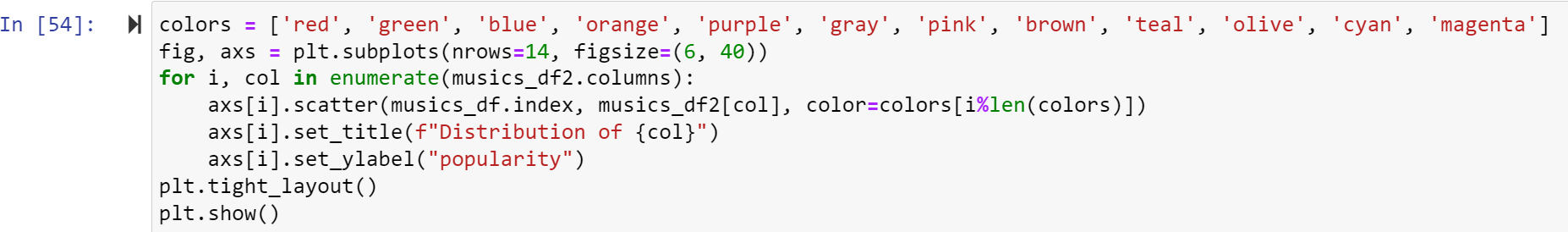
پس از اینکه یک ماتریس 15 ستونه برای نمایش genre هر سطر بدست آمد لا توجه به خواسته ی سوال مقادیر هر سطر به یک بردار 1\*15 تبدیل می شوند و در یک ستون جدید با نام new genre ریخته می شوند.

**هر بردار از ستون new genre داده های ستون genre را با مقادیر 0 , 1 نشان می دهد.**

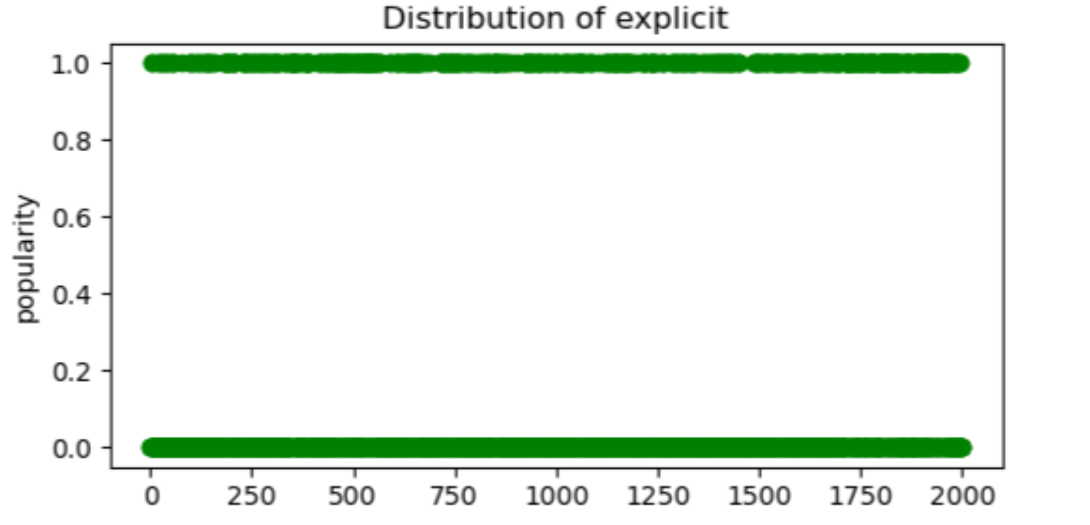
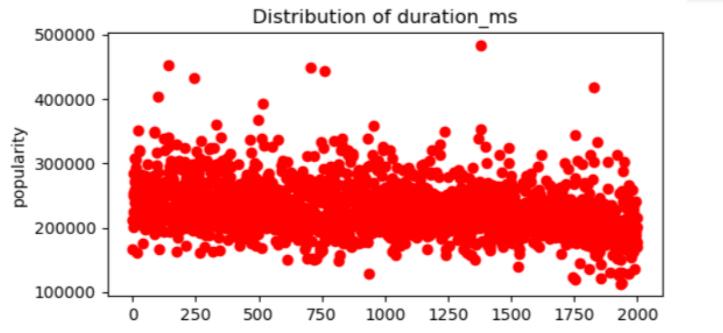


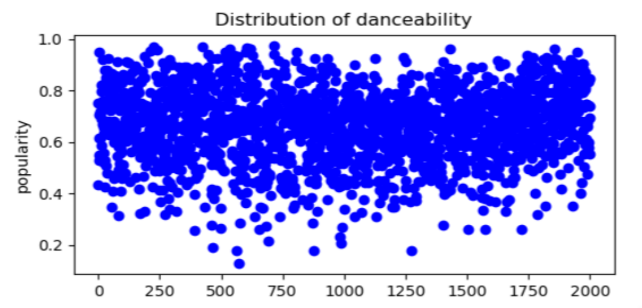


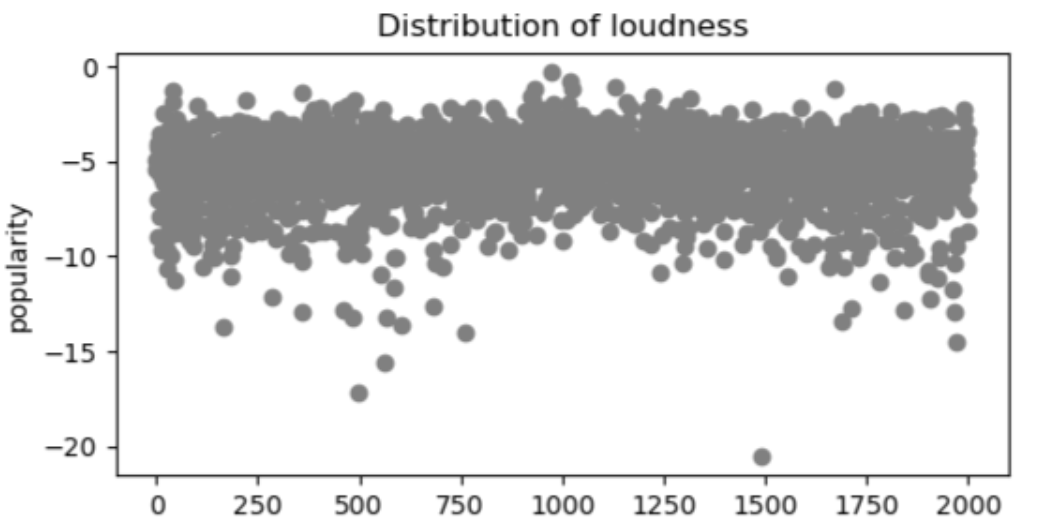
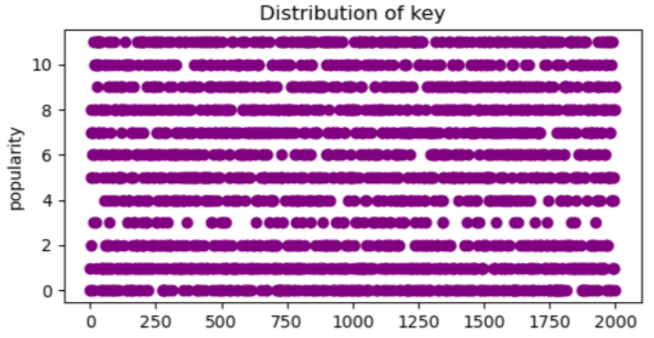
برای نمایش پراکندگی داده ها بر حسب ویژگی ابتدا ستون های خواسته شده حذف می شوند یک data frame جدید شامل ویژگی هایی که قرار است توزیع دادگان آن ها محاسبه شود ساخته می شود.

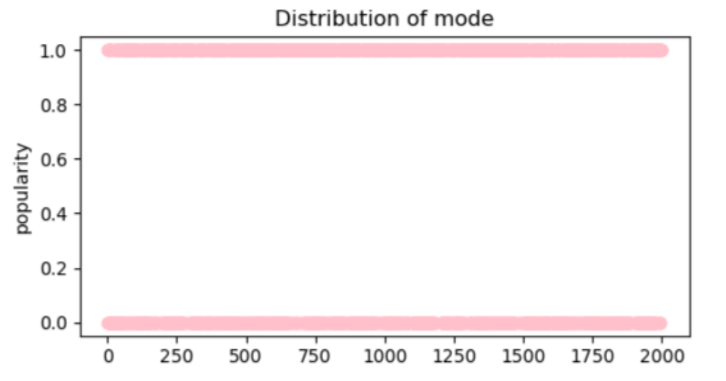


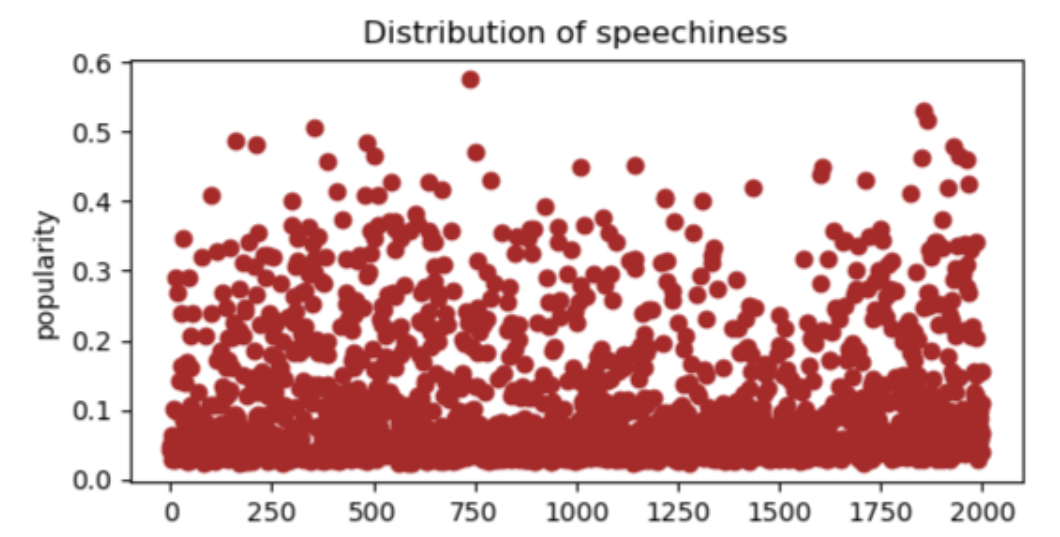
برای نمایش توزیع دادگان از دستور plt.suplots از کتابخانه matplotlib استفاده می شود و به کمک یک حلقه for برای هر ویژگی نمودار نقطه ای توزیع داده ها بر حسب محبوبیت بدست می آید که در ادامه این نمودارها برای هرکدام از ویژگی ها نمایش داده شده اند.

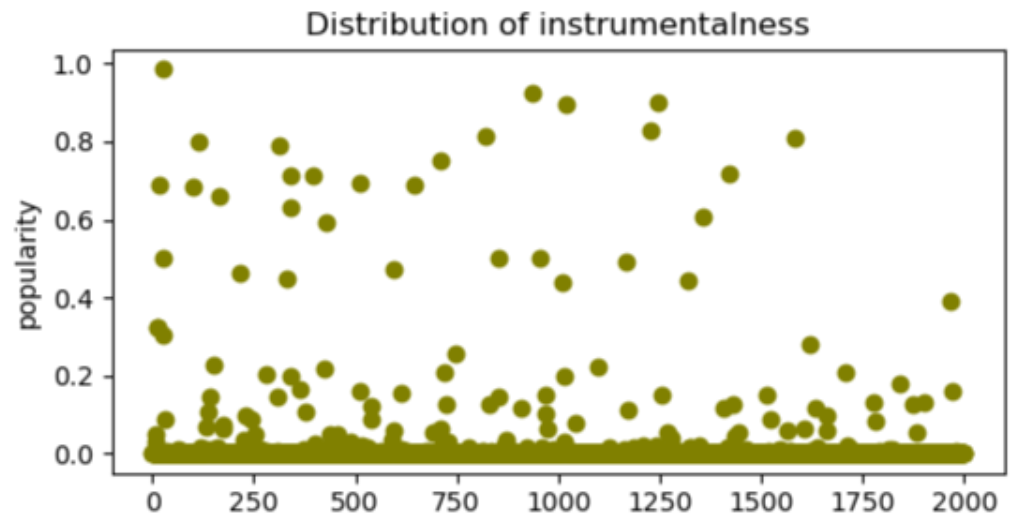
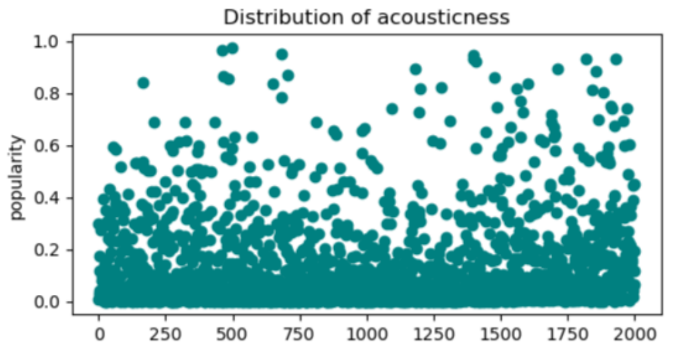


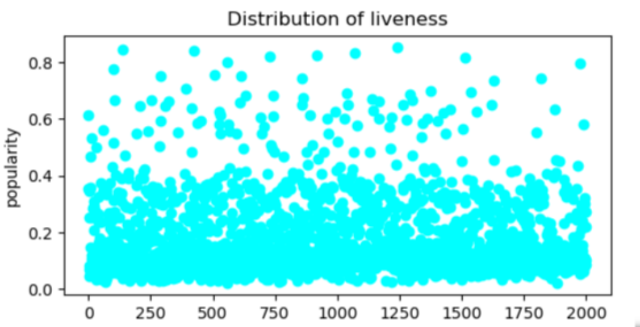


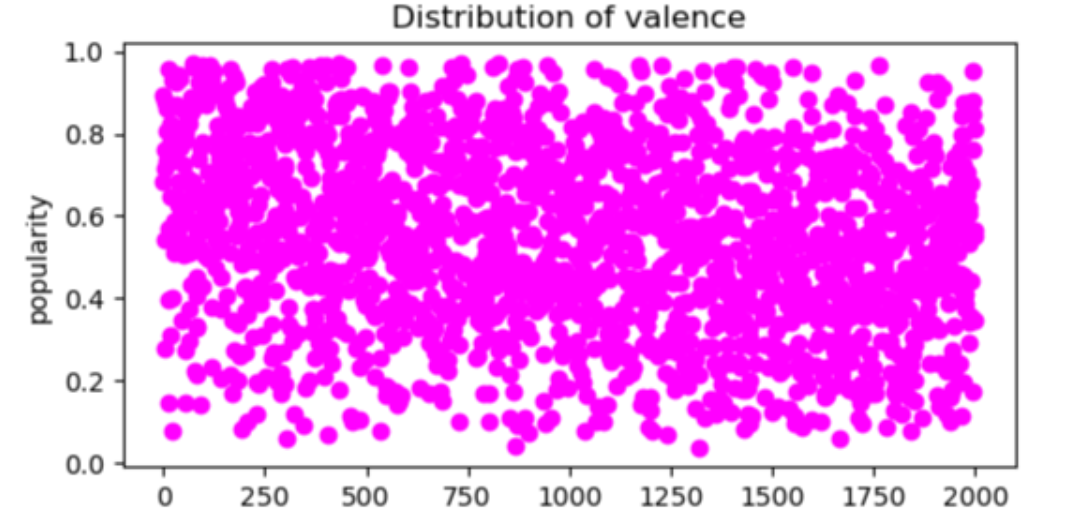


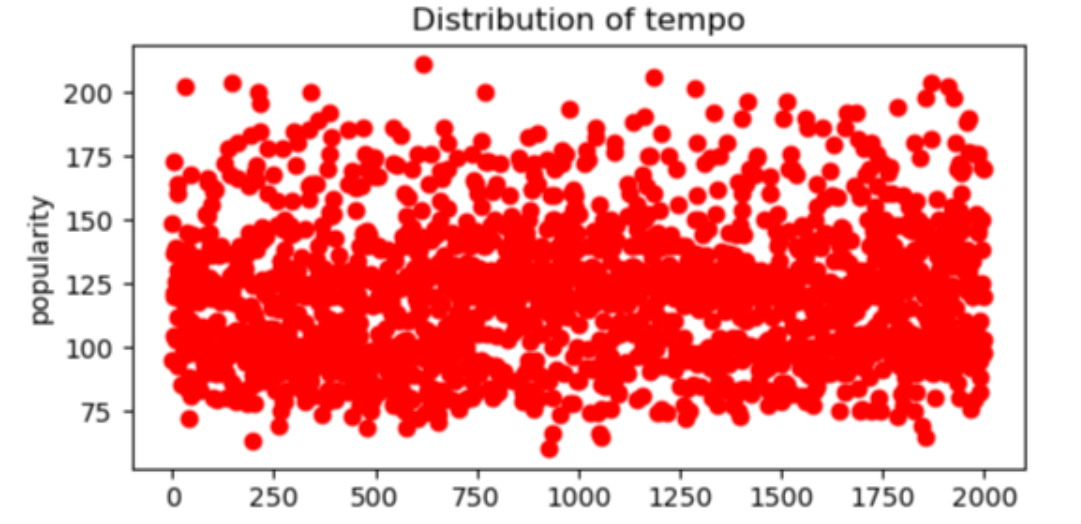






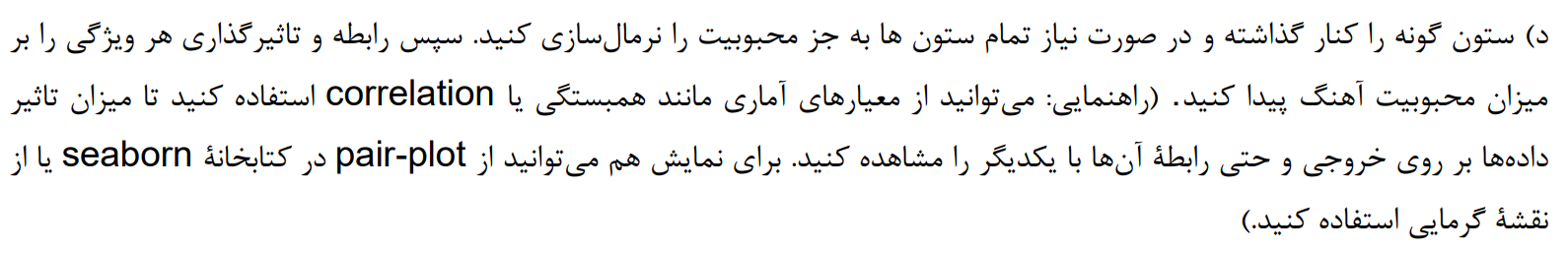


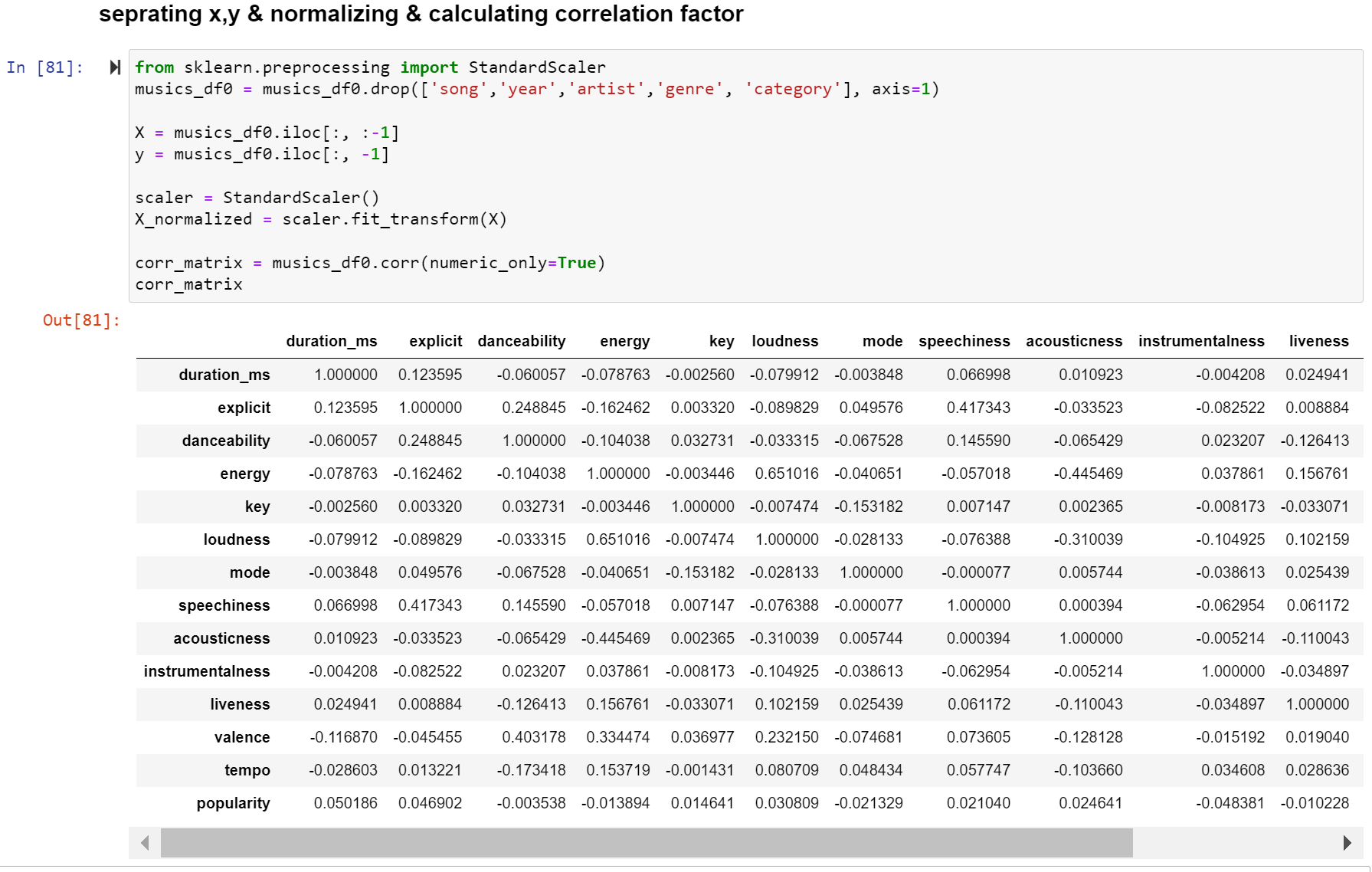




برای داده‌های دو بعدی، پراکندگی نشان‌دهنده‌ی این است که داده‌ها به چه شکلی درون فضا پخش شده‌اند و چقدر از نواحی فضا خالی هستند. در واقع، پراکندگی داده‌ها نشان‌دهنده‌ی شدت و تنوع توزیع داده‌ها در مجموعه داده‌ها است. همچنین، پراکندگی داده‌ها می‌تواند در مدل‌هایی مانند KNN و SVM، در تنظیم پارامترهای مانند تعداد همسایگان در KNN و پارامتر C در SVM مؤثر باشد.

در توزیع داده های این سوال برای مثال ویژگی های tempo و valence با پراکندگی خوب در کل فضا و عدم تمرکز در ناحیه مشخصی موجب تربیت بهینه مدل می شوند درصورتی که در ویژگی هایی مانند instrumentalness یا loudness اکثریت داده ها در یک ناحیه تمرکز دارد در حالی که تعداد کمی از داده ها در نواحی دیگر پخش شده اند و این عامل موجب می شود داده های این ویژگی ها نتوانند تمامی شرایط موجود در داده ها را پوشش دهند و در نهایت دقت کاهش یابد.



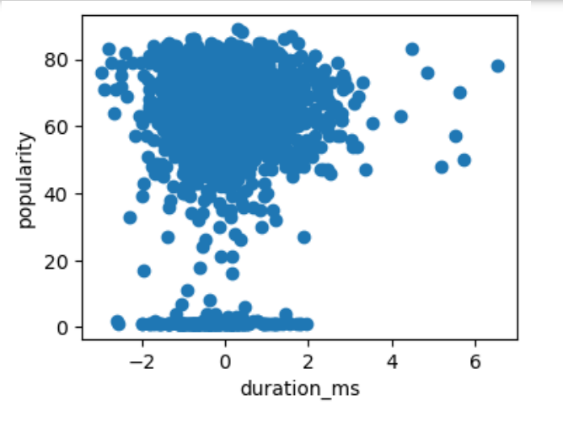


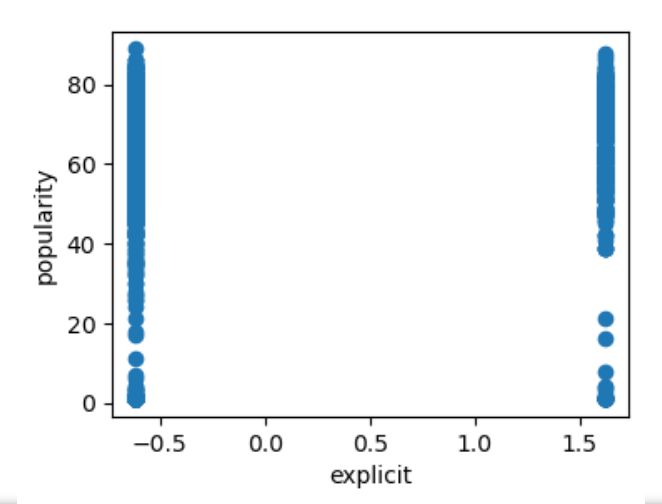
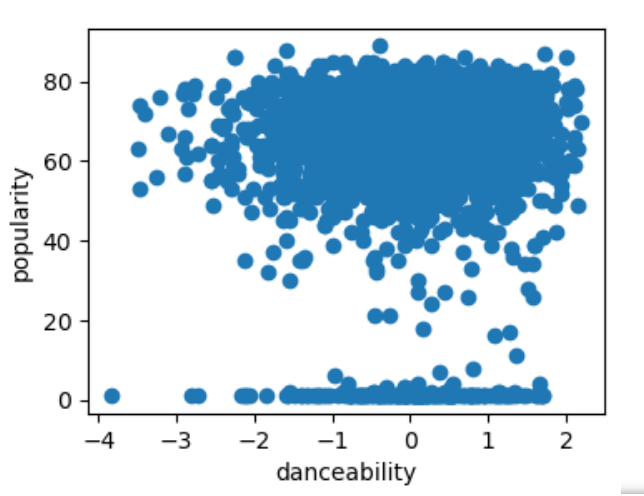
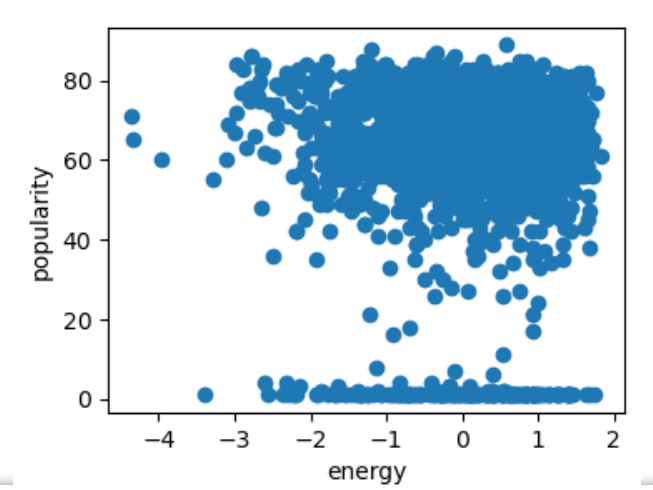
**داده های بالا ضریب correlation را برای ویژگی ها برحسب سک دیگر و محبوبیت نشان می دهد.**

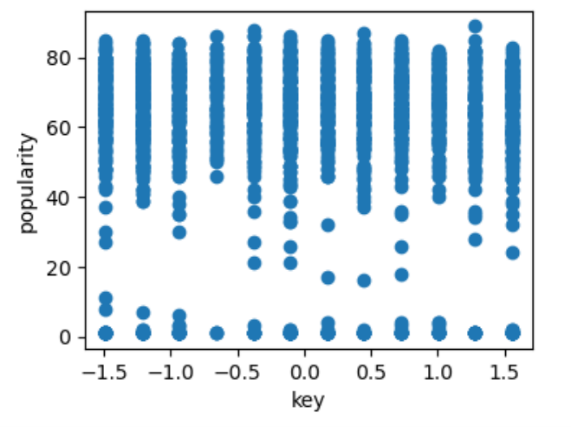
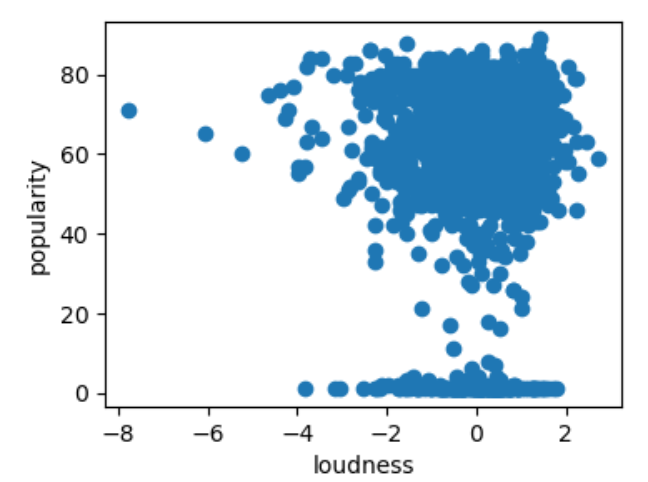
مطابق دستور بالا همه ی ویژگی های ورودی که شامل 13 ستون می باشد درون متغیر x و ستون popularity درون متغیر y ریخته می شوند. سپس همه ی ستون های ورودی با استفاده از دستور StandardScaler از کتابخانه ی sklearn ، normalizing می شوند و با استفاده از دستور corr ضریب همبستگی برای همه داده ها محاسبه می شود.

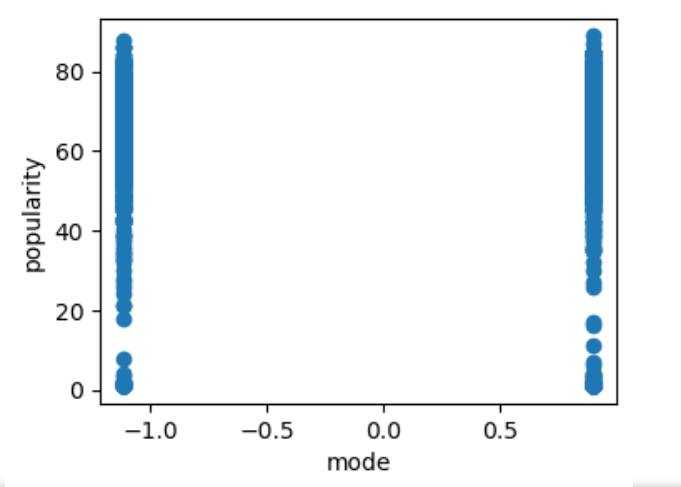
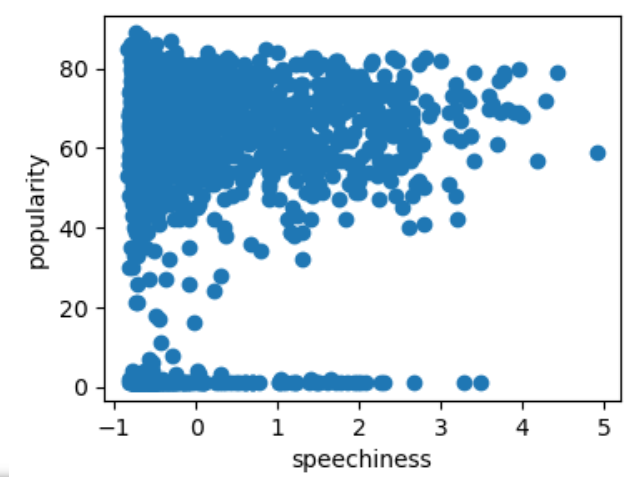
در جدول بالا مشاهده می شود که رابطه و تاثیر گذاری هر ویژگی بر روی میزان محبوبیت و همچنین تاثیر آن بر روی سایر ویژگی ها با استفاده از ضریب correlation نشان داده شده است. ستون آخر نمایانگر تاثیر گذاری هر ویژگی بر روی میزان محبوبیت می باشد.

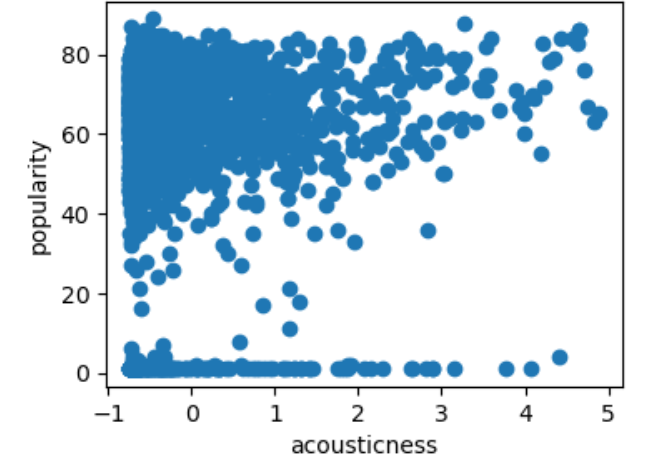
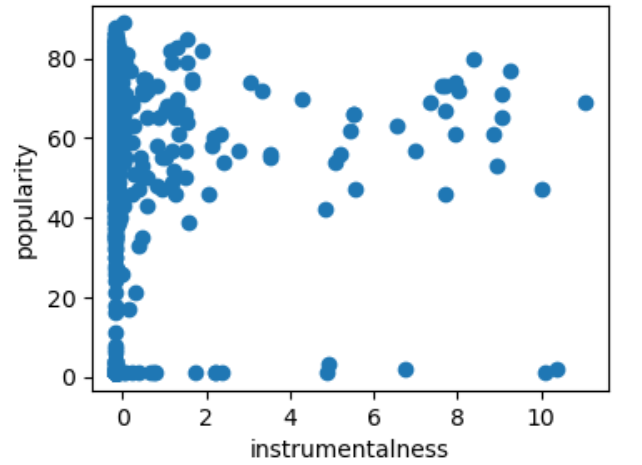
ضریب همبستگی (Correlation coefficient) نشان دهنده میزان رابطه خطی بین دو متغیر است. این ضریب بین دو متغیر به صورت عددی در بازه ی -1 تا 1 تعریف می‌شود. اگر ضریب همبستگی برابر با 1 باشد، به این معناست که دو متغیر کاملاً مرتبط و همبستگی مثبت دارند، یعنی هر چه ارزش یکی افزایش یابد، ارزش دیگری نیز افزایش پیدا می‌کند. اگر ضریب همبستگی برابر با -1 باشد، به این معناست که دو متغیر کاملاً معکوس و همبستگی منفی دارند، یعنی هر چه ارزش یکی افت کاهش یابد، ارزش دیگری افزایش پیدا می‌کند. و در صورتی که ضریب همبستگی برابر با صفر باشد، نشان دهنده عدم وجود همبستگی بین دو متغیر است، یعنی تغییرات یکی از متغیرها با تغییرات دیگری همراه نیست.

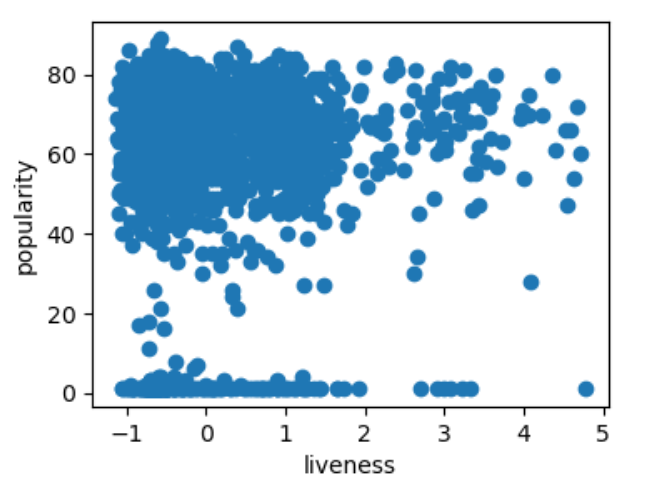
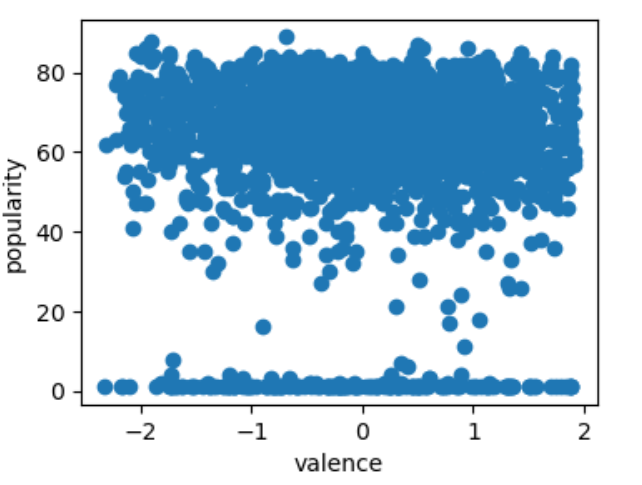
**در ادامه نمودار تاثیر گذاری هر ویژگی بر روی میزان محبوبیت رسم شده است.**

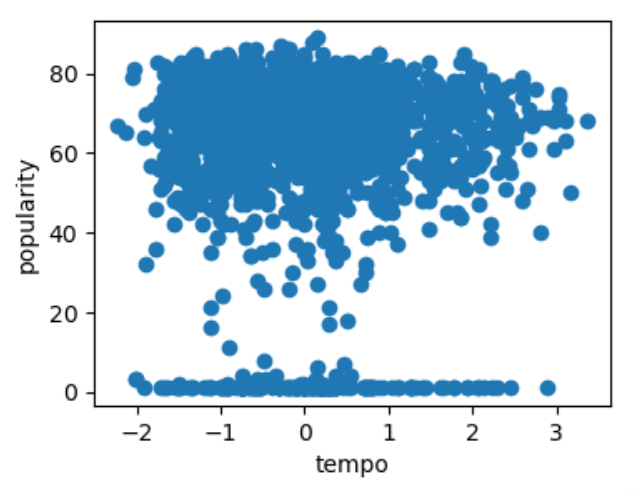
  
 

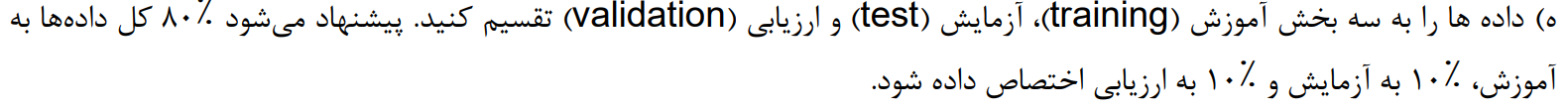
 

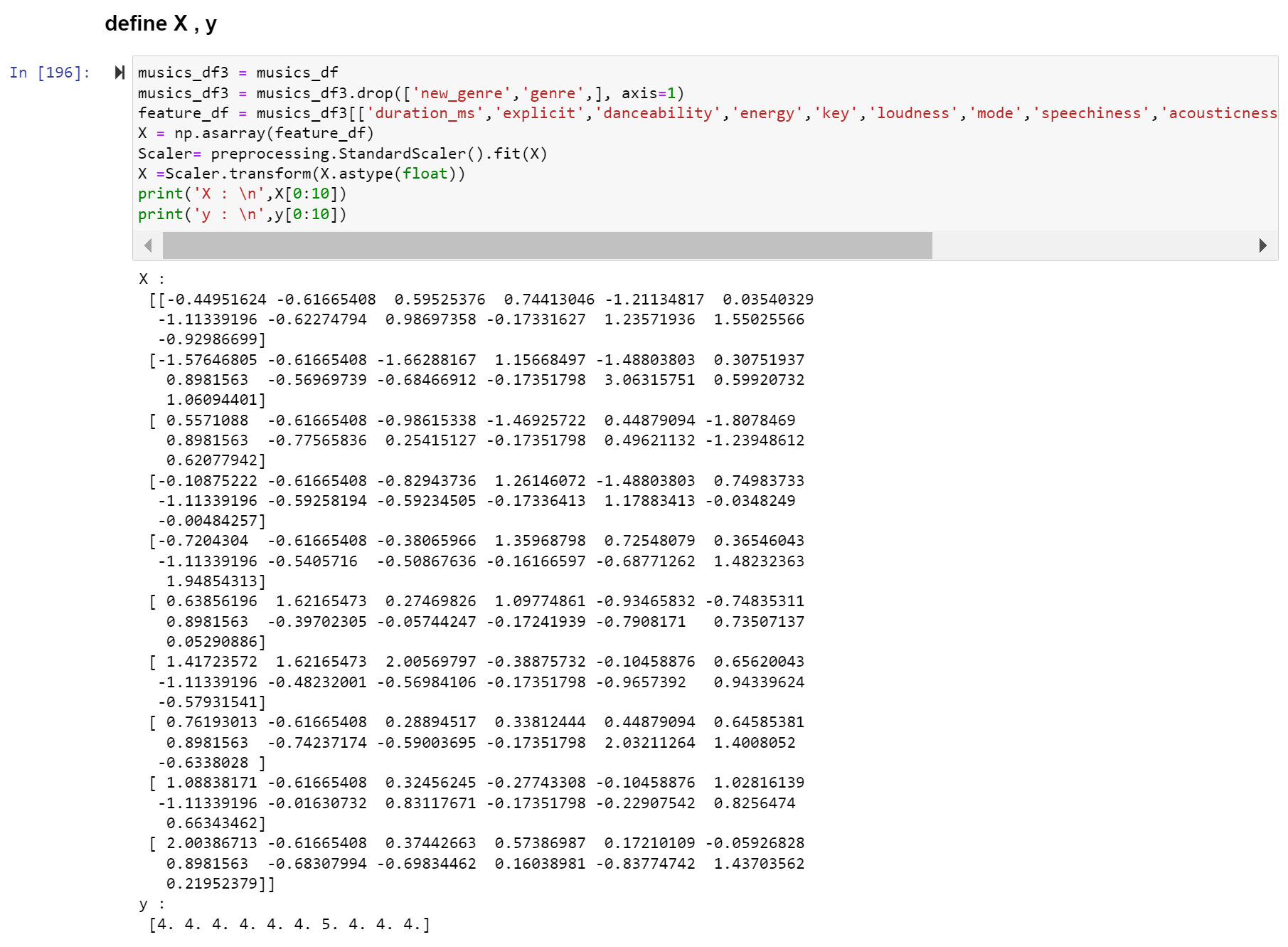
 

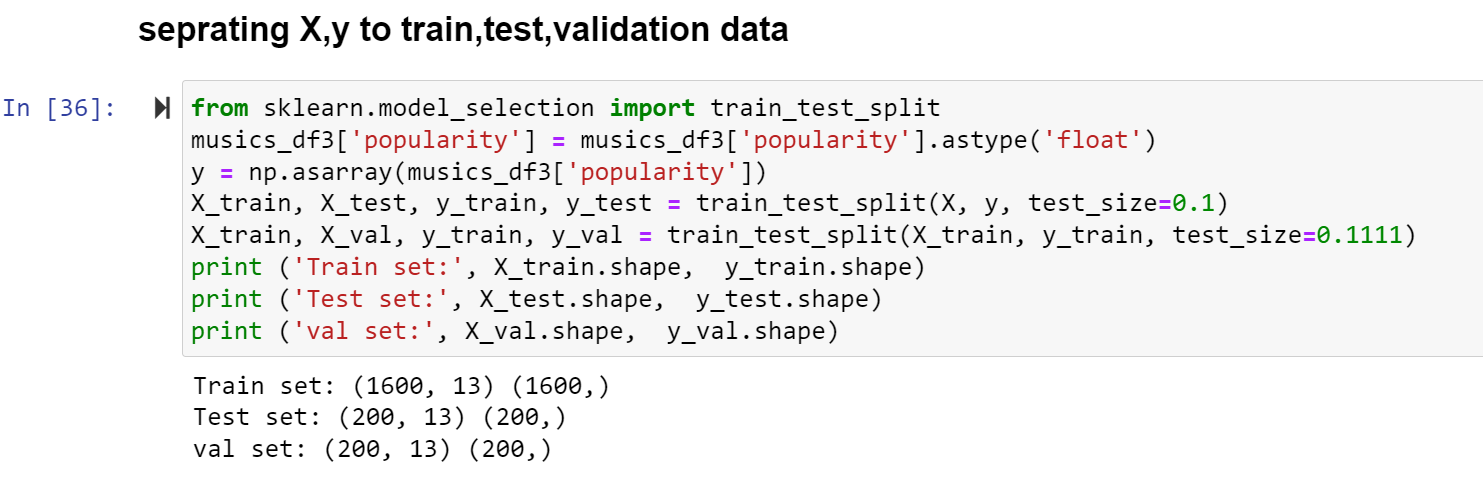
 



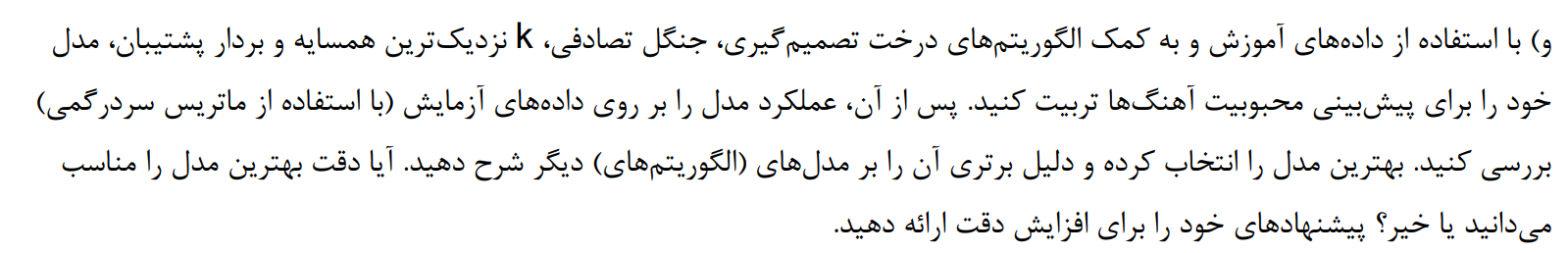




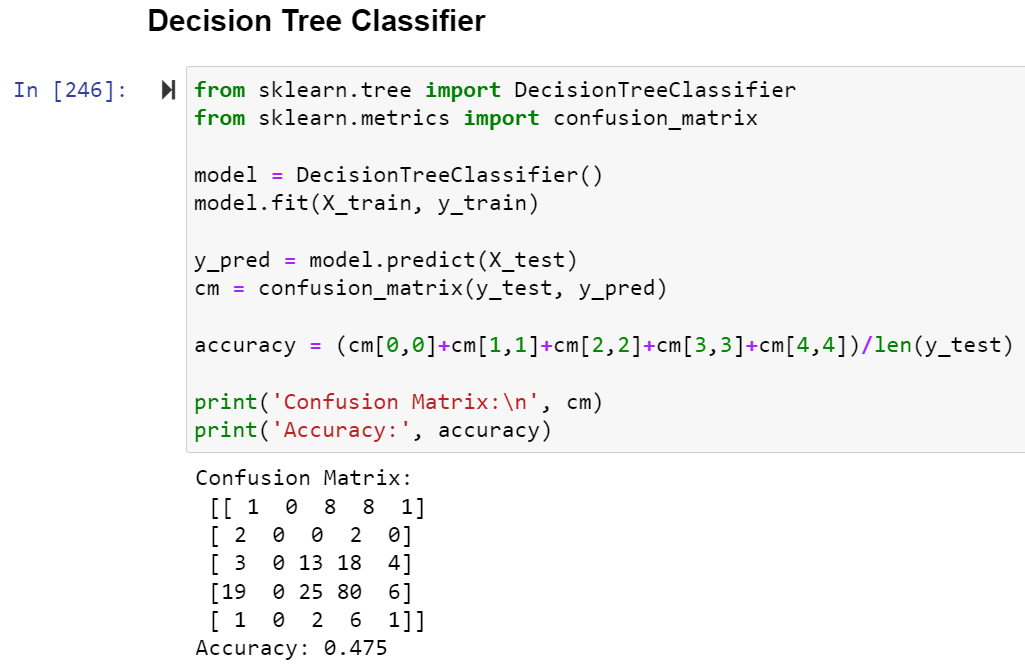
ابتدا مطابق شکل بالا ویژگی های ورودی در ماتریس 13 ستونه X و داده های خروجی در ماتریس تک ستونه y ریخته می شوند.



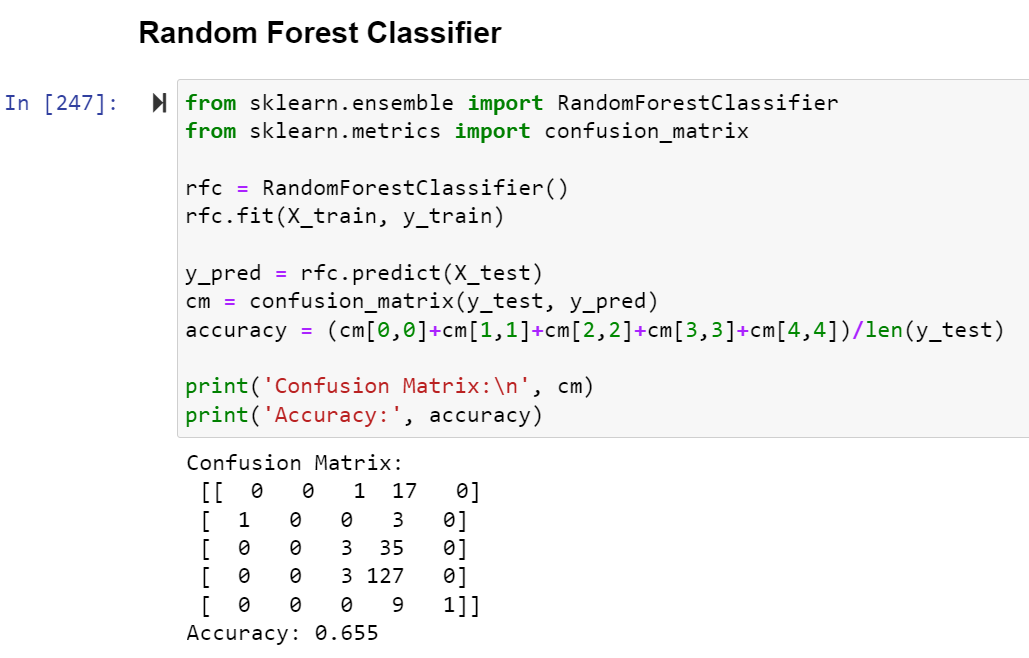
مطابق خواسته سوال داده های ورودی با دستور train\_test\_split از کتابخانه sklearn به سه بخش training , test , validation تقسیم می شوند که 80 درصد داده ها در قسمت train و 10 درصد در بخش validation و 10 درصد در بخش test ریخته می شود.



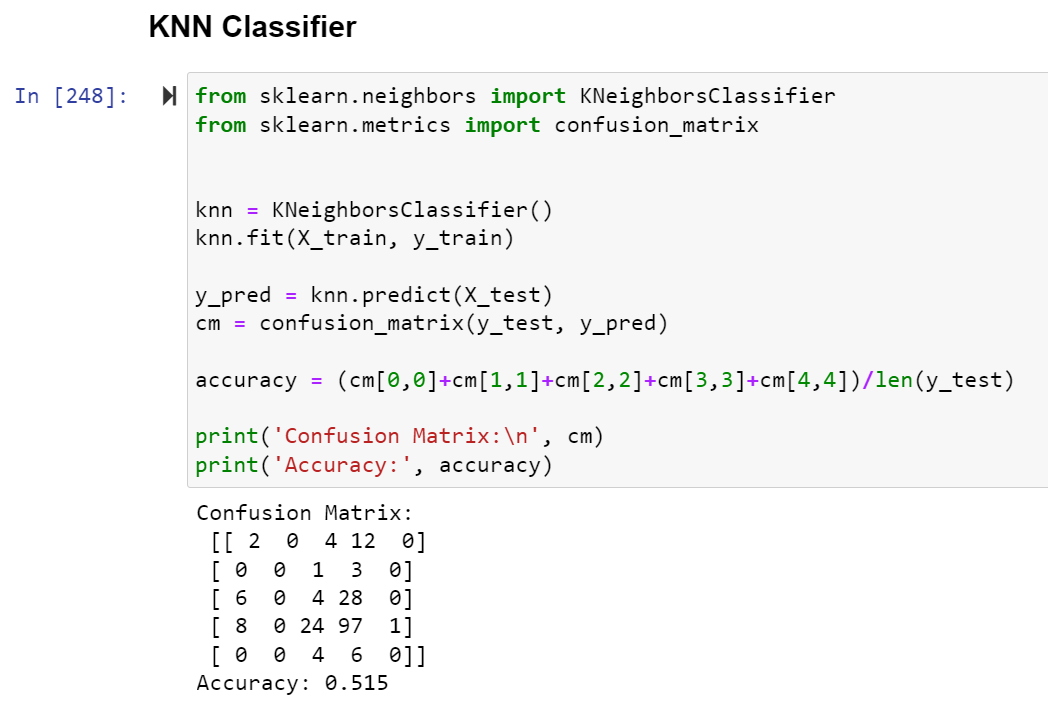
**مدل درخت تصمیم گیری:**



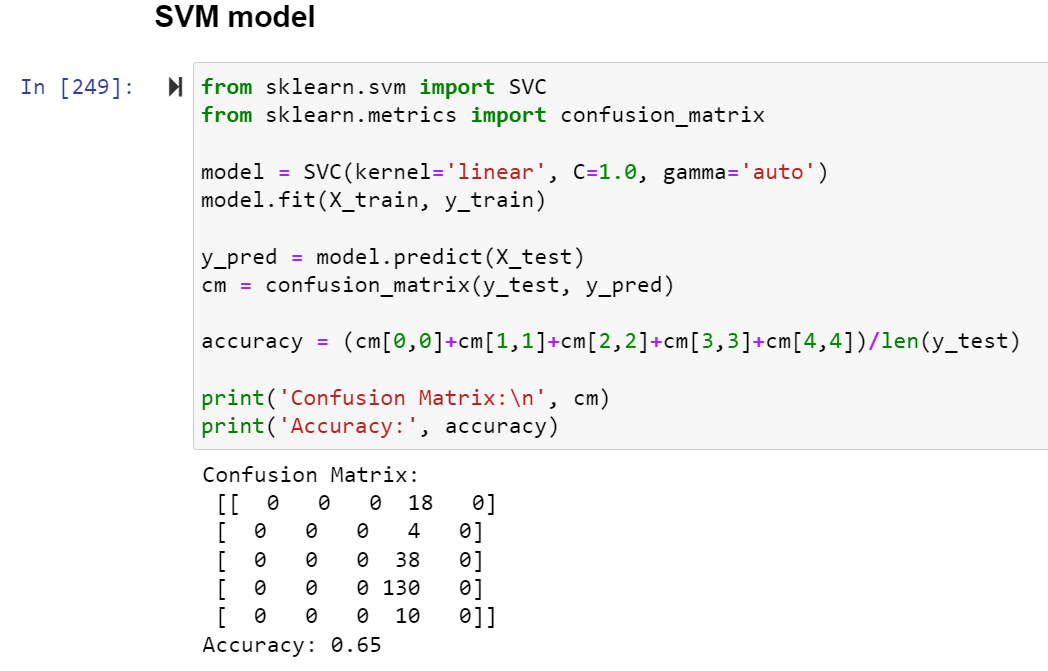
**مدل جنگل تصادفی:**



**مدل KNN :**



**مدل SVM :**



باتوجه به اینکه تقسیم داده ها به گروه های تست، تمرین و ارزیابی به صورت تصادفی بوده است، در هر مرحله اجرای کد، مقادیر ماتریس سردرگمی و دقت های حاصل شده برای هر مدل متفاوت خواهد بود. لذا اعداد ذکر شده درمدل های بالا ثابت نمی باشند و به ازای هربار اجرای کد تغییر می کنند. بنابراین امکان دارد در هر مرحله یکی از مدل ها بیشترین دقت را حاصل کند. ولی به طور کلی و با +50 بار اجرای کد، مدل های Random Forest Classifier , SVM بیشترین دقت را حاصل می کنند.

**مقایسه مدل های تربیت شده :**

|  |  |
| --- | --- |
| **Accuracy (%)** | **Model** |
| **47.5** | **Decision Tree Classifier** |
| **65.5** | **Random Forest Classifier** |
| **51.5** | **KNN** |
| **65** | **SVM** |

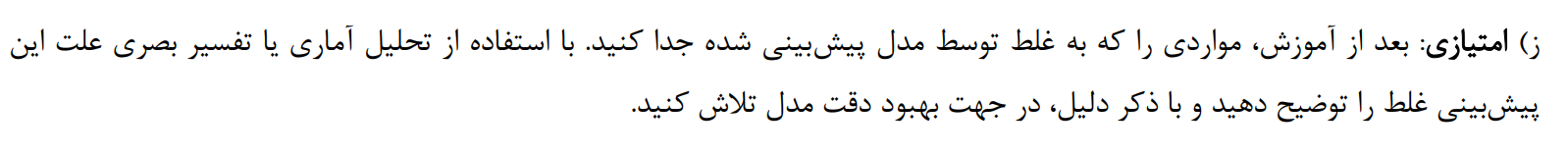
در مدل های تربیت شده معمولا بیشترین دقت مربوط به مدل های SVM و Random Forest Classifier می باشد. ولی با توجه به مقایسه ماتریس های سردرگمی نتیجه گیری می شود که مدل SVM همه ی داده ها را در بازه ی محبوبیت 4 یعنی محبوبیت 60-80 پیشبینی کرده است و صرفا در این سوال باتوجه به تراکم خروجی ها در این بازه مقدار دقت آن بیشتر از سایر مدل ها شده است. به عبارت دیگر باتوجه به اینکه هرداده ورودی به آن وارد شود خروجی ثابتی را برای آن پیش بینی می کند لذا مدل کارآمدی نیست.

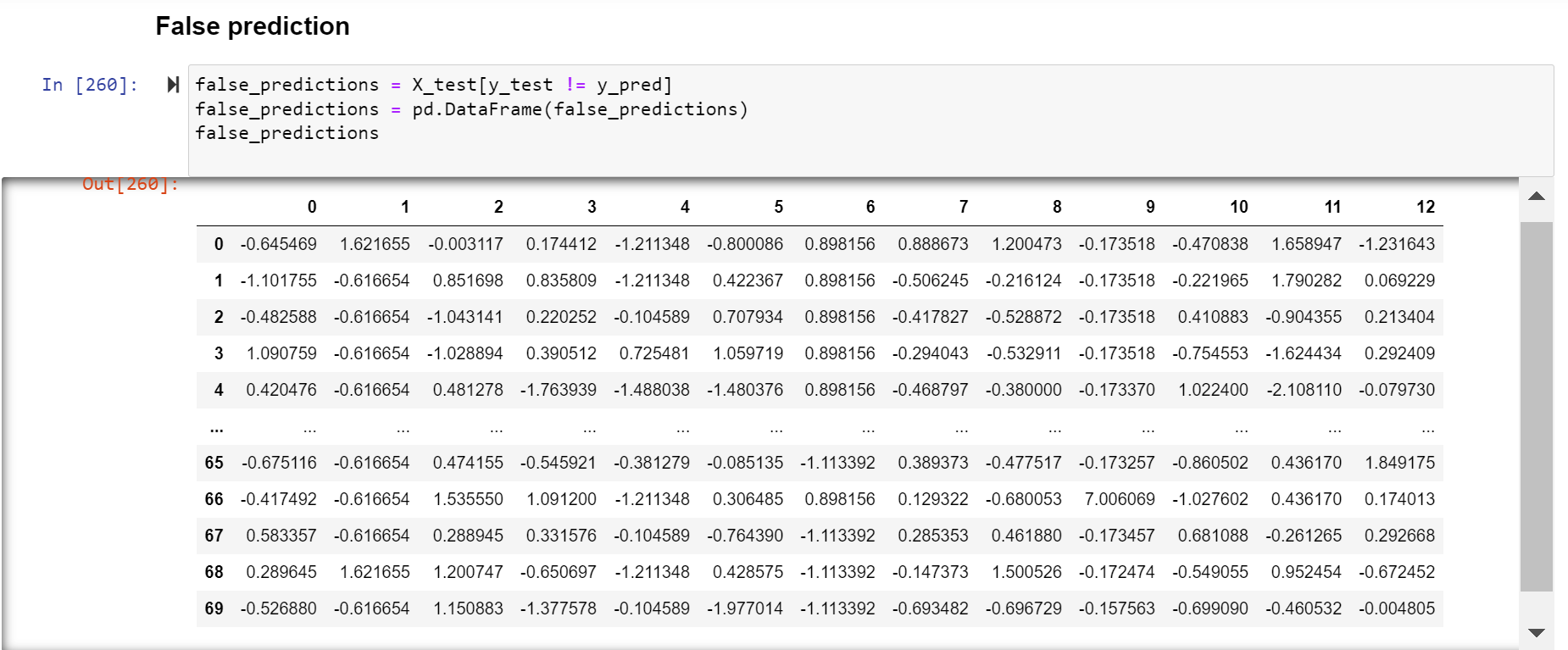
**بنابراین به نظر می رسد برای این مسئله مدل Random Forest Classifier بهترین مدل می باشد که دقت آن به 65.5 نیز می رسد.**

برای افزایش دقت مدل DTS (Decision Tree Classifier)، می‌توان از رویکردهای زیر استفاده کرد:

1. تنظیم پارامترهای مدل مانند max \_ depth یا min \_ sample \_ leaf
2. استفاده از روش‌های ensemble که در این روش‌، چندین مدل DTS با پارامترهای مختلف آموزش داده می‌شوند و سپس نتایج آن‌ها ترکیبه دسته‌بندی نهایی ترکیب می‌شوند.
3. استفاده از روش‌های feature selection که این روش‌ها باعث کاهش تعداد فیچرها و افزایش دقت مدل DTS می‌شوند.
4. اصلاح overfitting با روش‌هایی مانند pruning و regularization

**اما مهمترین عامل کاهش دقت این است که در این مسئله برای پیشبینی خروجی یک دسته بندی چندگانه وجود دارد درصورتی که مدل های ارائه شده برای دسته بندی دوگانه مناسب می باشند. لذا باتوجه به اینکه خروجی شامل دسته بندی چندگانه می باشد، برای افزایش دقت مدل، باید از شبکه های عصبی استفاده شود.**



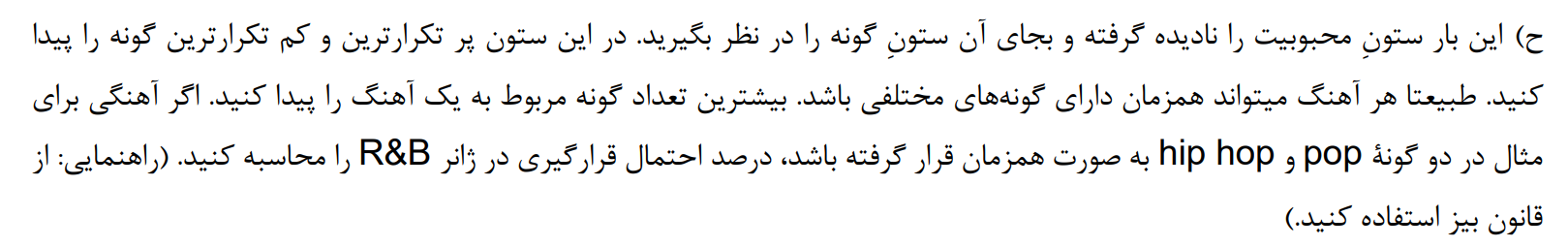


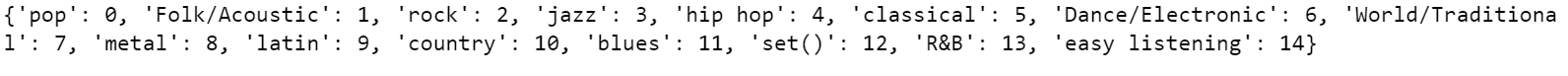
علت داده هایی که به غلط پیش بینی شده اند می تواند ناشی از داده‌های آموزشی کم، داده‌های آموزشی ناهمگن، پارامترهای نامناسب مدل،انتخاب مدل نامناسب و ... باشد.

در این مسئله علت اصلی پیش بینی غلط و پایین بودن دقت مدل چند دسته ای بودن خروجی محبوبیت می باشد در حالی که از مدل های دسته بندی دوگانه برای پیشبینی خروجی استفاده شده است. لذا می‌توان از مدل‌های پیچیده‌تری مانند شبکه‌های عصبی یا شبکه‌های عصبی عمیق برای تربیت این مدل استفاده کرد.

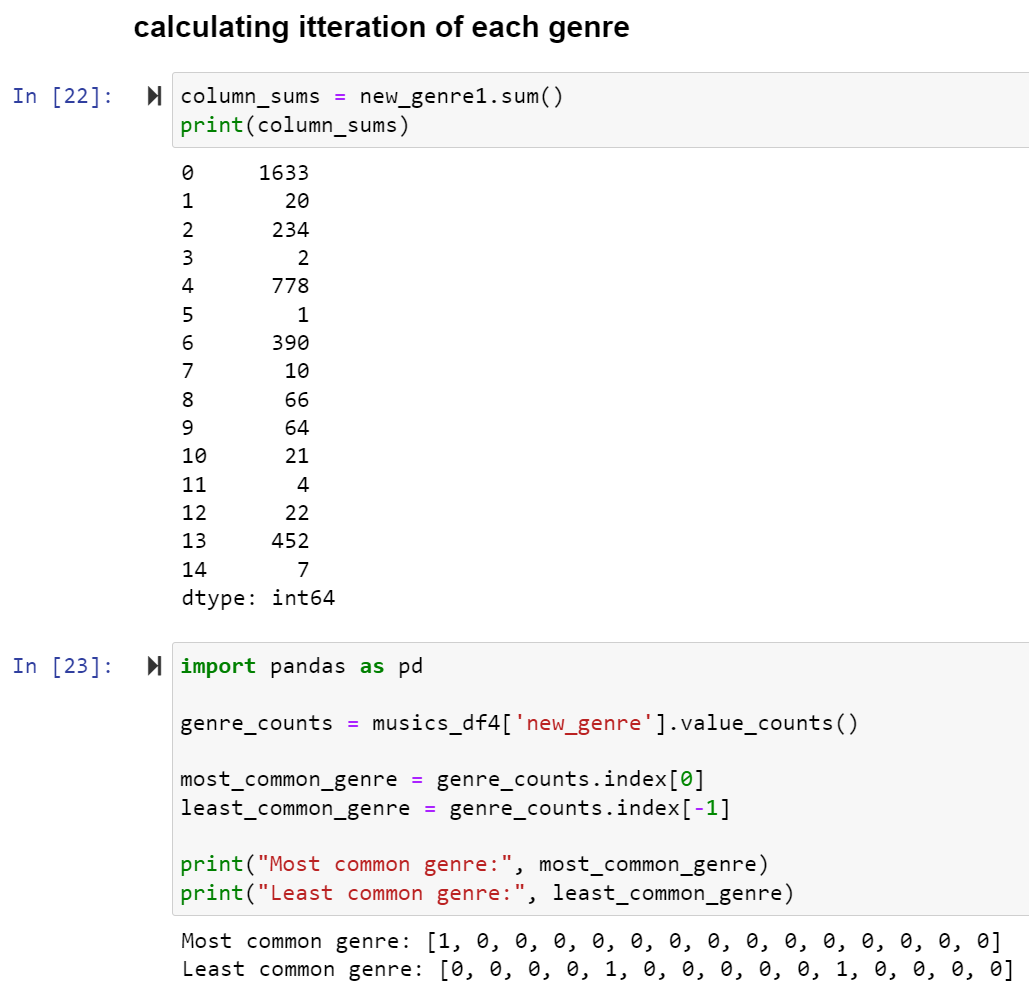
همچنین اگر به داده های سوال دقت شود بیشترین خروجی را بازه یا گروه 4 که شامل محبوبیت 60-80 می باشد تشکیل داده اند در صورتی که ممکن است داده هایی که برای تست به صورت تصادفی جدا می شوند خروجی در این بازه قرار نداشته باشند و این موضوع باعث کاهش دقت می شود.

برای حل این مشکل می توان از روش **k-fold cross validation** برای انتخاب داده های تست، تمرین و ارزیابی استفاده کرد که این موضوع دقت را به میزان قابل توجهی افزایش خواهد داد.





ماتریس new\_ genre که تعریف شد هر درایه آن شامل مقادیر صفر و یک بود. مثلا اگر . مثلا اگر genre یک نمونه jazz می بود درایه شماره 3 از این بردار یک و بقیه درایه ها صفر بودند.



برای اینکه پر تکرار ترین و کم تکرار ترین گونه پیدا شوند از ماتریس new\_ genre1 که همان بردار های

new\_ genre پیش از تبدیل به فرم برداری می باشد استفاده می شود. از آنجایی که در این ماتریس ستون ها شامل genre های متفاوت می باشند و در هر نمونه درایه ستون genre آن برابر یک می باشد لذا اگر مقادیر هر ستون با یکدیگر جمع شوند در پایان ستونی که بیشترین مقدار یک را نشان می داد حاصل جمع آن از همه بیشتر است و آن ستون نشان دهنده پر تکرار ترین genre می باشد. به عبارت دیگر مشاهده می شود که کدام تکرار عدد یک در بین ستون ها یا همان ژانر ها بیشتر از بقیه بوده است.

طبیعتا برای پیدا کردن کم تکرار ترین genre نیز ستونی که جمع مقادیر آن از همه کمتر باشد یعنی genre مربوط به آن کمتر از سایر تکرار شده است.

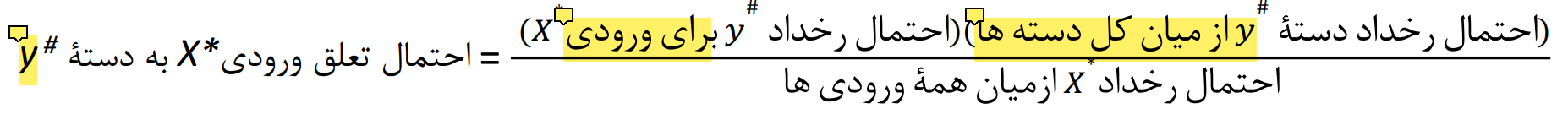
همانطور که مشاهده می شود پرتکرار ترین ستون مربوط به ستون شماره 0 از ماتریس new\_ genre1 می باشد که مطابق اختصاص اعداد به هر genre این ستون معرف pop است. **یعنی ژانر یعنی ژانر pop به صورت تکی یا هنگامی که همراه با یک یا چند ژانر دیگر آمده است پرتکرار ترین ژانر می باشد. همانطور که مشاهده می شود این ژانر 1633 بار به صورت تکی یا مشترک مورد استفاده قرار گرفته است.** همچنین کم تکرار ترین ستون مربوط به ستون 5 است که مطابق اختصاص اعداد این ستون معرف classical می باشد **یعنی ژانر classical کم تکرار ترین ژانر می باشد**. این ژانر تنها یک بار مورد استفاده قرار گرفته است و لذا کم تکرار ترین ژانر می باشد.

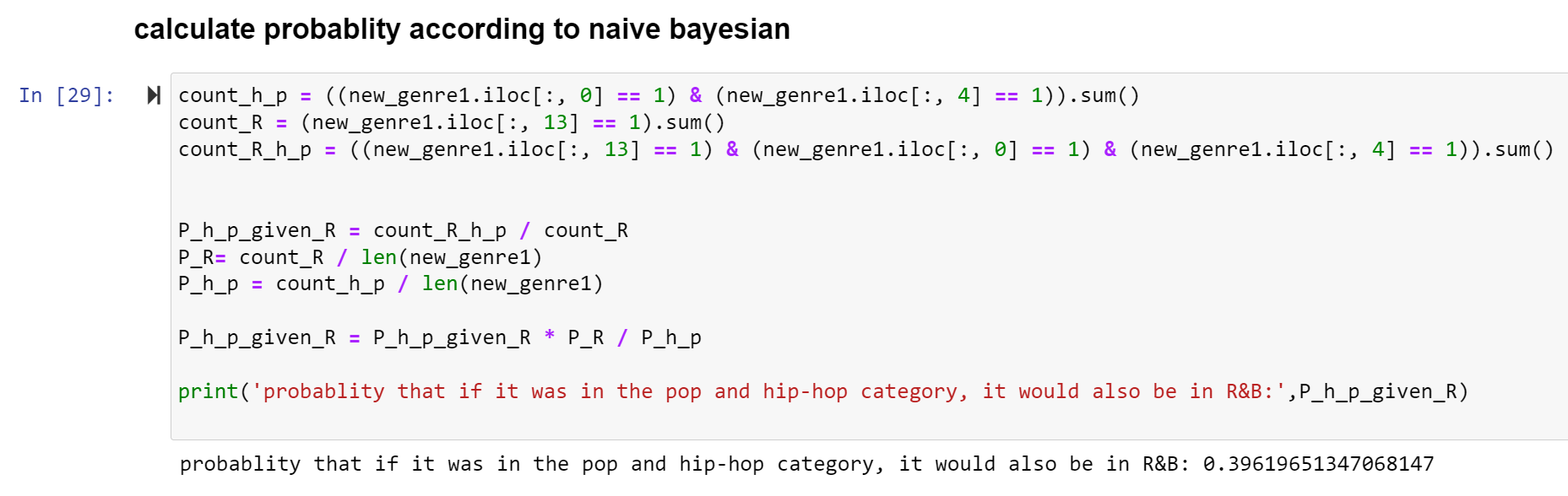


همچنین بیشترین گونه مربوط به یک آهنگ مربوط به زمانی است که فقط ژانر آن pop باشد و کمترین گونه مربوط به یک آهنگ مربوط به زمانی است که ژانر آن تلفیقی از hip-hop و country باشد که خب منطقی به نظر می رسد.

قسمت بعدی این سوال باید با استفاده از قانون بیز در حالتی که یک موزیک در ژانر hip-hop , pop قرار دارد احتمال قرار داشتن آن در ژانر R&B محاسبه شود.

قانون بیز:





برای استفاده از قانون بیز احتمالات مطابق زیر تعریف می شوند:

احتمال تعلق X\* به دسته y# :

احتمال اینکه ورودی از نوع pop , hip hop از نوع R&B نیز باشد.) P\_h\_p\_given\_R (

احتمال رخداد رخداد y# برای ورودی X\* :

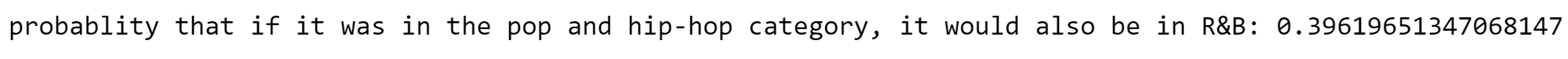
احتمال اینکه یک ورودی شامل R&B شامل pop , hip hop نیز باشد ) P\_h\_p\_given\_R (

احتمال رخداد y# از میان کل دسته ها:

احتمال اینکه در میان کل دسته ها ورودی از نوع R&B باشد. P\_R) (

احتمال رخداد X\* از میان همه ورودی ها:

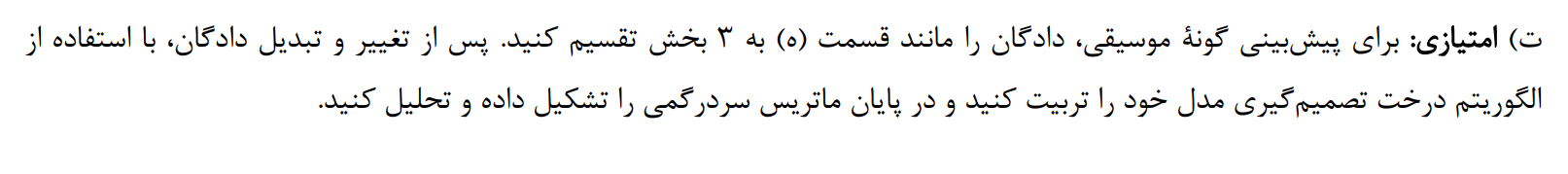
احتمال اینکه ورودی از نوع Pop و hip hop باشد. (P\_h\_p)



بنابر این احتمال تعلق X\* (R&B) به دسته y# (pop, hip hop) مطابق محاسبات برابر با 0.3962 می باشد که مطابق قانون بیز احتمالات کم تر از 0.5 صفر در نظر گرفته می شوند یعنی برای ورودی جدید طبق قانون بیز اگر ورودی شامل pop , hip hop باشد R&B نخواهد بود.

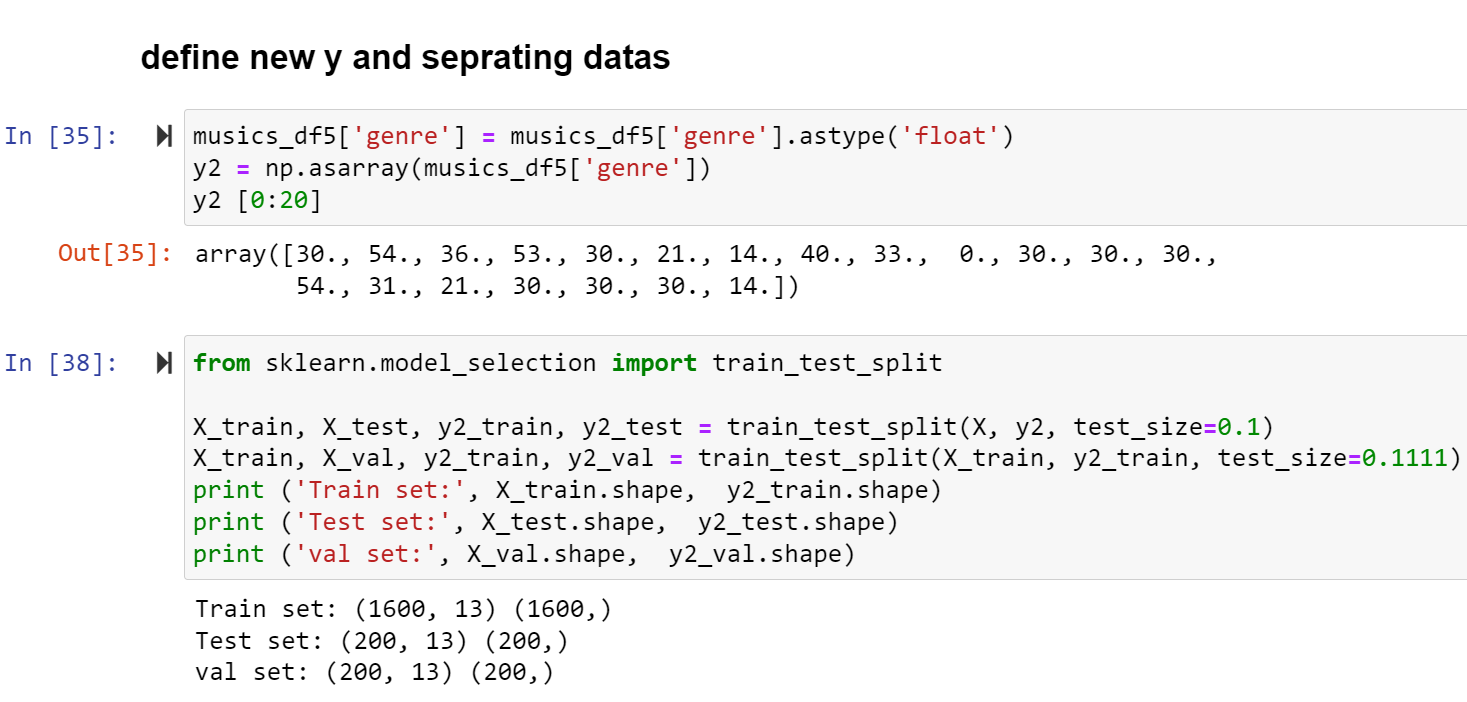
باتوجه به اینکه این احتمالات با استفاده از ستون new\_ genre1 و اعداداختصاص یافته به هر ژانر انجام شده است و اینکه در هر بار اجرای کد اعداد اختصاص یافته به ژانر های pop, hip hop , R&B متفاوت می باشد باید در سه خط اول برای محاسبه احتمالات آن ها جایگذاری شود. مثلا در این جا چون pop ستون شماره 0 ماتریس new\_ genre است برای محاسبه احتمال آن در خط اول شماره ستون آن یعنی 0 جایگذاری شده است ولی در صورت اجرای مجدد کد این عدد امکان دارد تغییر کند و محاسبات دچار خطا شود.

( احتمال اشتباه یا به صورت nan نشان داده می شود)

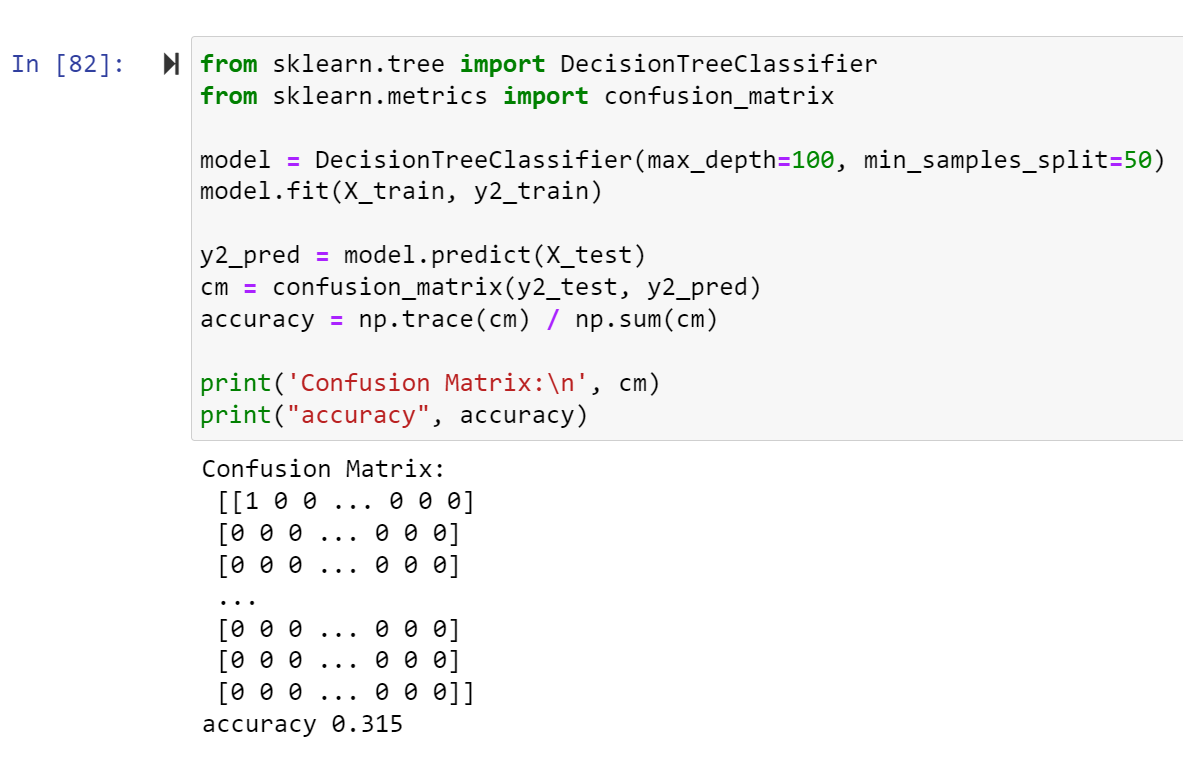




با توجه به اینکه پس از اعمال one hot encoding درایه های ستون genre به صورت بردار تعریف شد و امکان تعریف خروجی به صورت بردار 15 درایه ای وجود ندارد و همچنین مدل را دچار مشکل میکند برای تربیت مدل با استفاده از خروجی genre در این قسمت به هرکدام از داده های خروجی با استفاده از دستور LabelEncoder یک label زده می شود. اگرچه در این روش به دلیل اینکه خروجی ها زیاد هستند هم خطا زیاد خواهد بود ولی از حالتی که خروجی به عنوان برداری از صفر و یک ها باشد بهتر است.



داده ها به سه بخش train, test , validation تقسیم می شوند.



همانگونه که پیش بینی می شد در این مدل باتوجه به اینکه به خروجی ها label زده شد و تعداد حالت های خروجی زیاد شد دقت مدل به اندازه قابل توجهی کاهش یافته است. در تربیت مدل با خروجی محبوبیت در قسمت های قبل به علت تربیت دسته بند دوگانه برای پیشبینی خروجی آن در حالی که خروجی دسته بندی چندگانه می باشد ، مدل تربیت شده دارای خطا بود. در این قسمت باتوجه به اینکه دسته بندی های خروجی افزایش یافته است طبیعتا پیشبینی می شود دقت به میزان قابل توجهی کاهش یابد.

همچنین ماتریس سردرگمی با توجه به این که 25 حالت برای خروجی label زده شده است و به عبارتی 25 خروجی متفاوت تعریف شده است یک ماتریس 25\*25 خواهد بود. برای ماتریس سردرگمی نیز خطا زیاد می باشد و عناصر قطر اصلی آن که نشان دهنده پیشبینی دقیق می باشند از حالت قبل کمتر می باشد که علت آن استفاده از مدل های تربیت دسته بندی دوگانه می باشد در حالی که دسته بندی خروجی اینجا شامل 25 دسته است.

باتوجه به پیچیدگی و چندگانه بودن داده ها و ابعاد زیاد ماتریس سردرگمی برای تربیت این مدل سایر روش ها از جمله شبکه های عصبی روش مناسب تری می باشند.