بسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی صنایع

تحلیل داده‌های اسپاتیفای

پروژه درس مبانی داده کاوی

استاد:

دکتر مجید خدمتی

مهرداد مرادی ۹۶۱۰۴۵۶۸

عارف روشن ۹۶۱۱۰۰۲۳

تیر ماه ۱۴۰۰

با تشکر و قدردانی از استاد محترم، جناب آقای دکتر مجید خدمتی که در این درس از معلومات و تدریسشان بهره بردیم و همچنین با تشکر از دستیاران آموزشی محترم که در یادگیری هر چه بهتر این درس ما را یاری نمودند.

چکیده

در این نوشتار سعی شده است که محبوبیت آهنگ‌ها در اسپاتیفای با استفاده از مدل‌های مختلف داده کاوی تخمین و پیش‌بینی شود. یک مجموعه داده متشکل از ۵۸۶۶۷۲ آهنگ و ۱۹ ویژگی داریم که با مطالعه روی آن‌ها و ایجاد تغییرات لازم در آن، مدل را آموزش و ارزیابی کرده و بهترین مدلمان، مدل bagging classifier با پارامتر ۲۵۰ شد که دقت ۸۶.۲٪ داشت.

عبارات کلیدی: داده کاوی، پیش‌بینی، مدل‌سازی، اسپاتیفای، محبوبیت آهنگ

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل اول: مقدمه

مقدمه

اسپاتیفای یک اپلیکیشن شنیدن موسیقی، پادکست و هرگونه محتوای صوتی دیگر می‌باشد که یک سرویس فری‌میوم است، یعنی ویژگی‌های اساسی همراه با تبلیغات، رایگان هستند، در حالی که ویژگی‌های اضافی شامل کیفیت پخش بهتر و دانلودهای موسیقی آفلاین از طریق پرداخت اشتراک ارائه داده می‌شود.

این برنامه با داشتن بیش از ۳۰ میلیون آهنگ و ۱۰۰ میلیون کاربر فعال ماهانه، در قله صنعت شنیدن آنلاین به موسیقی قرار داشته و بدین سبب، مرجعی قابل اعتماد برای تحلیل آهنگ‌ها و میزان محبوبیت آن‌ها بین شنوندگان می‌باشد.

از آنجایی که صنعت موسیقی، یکی از بزرگ‌ترین صنعت‌های سرگرمی به حساب می‌آید و سرمایه‌گذاری روی این صنعت هر روزه در حال افزایش می‌باشد و همچنین به این دلیل که پیروی از مد و جریان روز در این صنعت بسیار کاربرد دارد؛ بنابراین تحلیل عمیق آهنگ‌ها در این مرجع (اسپاتیفای) می‌تواند چشم‌انداز مناسبی به هنرمندان و سرمایه‌گذاران برای فهم و درک بهتر جریان موجود در موسیقی و عوامل موفقیت یک آهنگ در این صنعت بزرگ بدهد.

با توجه به داده‌های موجود آهنگ‌ها در اسپاتیفای و همچنین مطالعات موجود در این صنعت، به صورت کلی برای مثال معیارهای زیر می‌تواند در میزان محبوبیت یک آهنگ تاثیرگذار باشد:

- محبوبیت خواننده یا خوانندگان
- سال تولید و پخش
- میزان انرژی
- میزان آکوستیک
- میزان رقص‌پذیری
- طول آهنگ

هدف اصلی این پروژه، پیش‌بینی محبوبیت آهنگ‌ها توسط مدل‌های داده کاوی می‌باشد و در این راستا با مطالعه، تحلیل، تغییر و پردازش داده‌ها، سعی شده است مناسب‌ترین مدل برای این هدف به دست آورده شده و ارزیابی شود.

مرور ادبیات

به واسطه‌ی ارائه‌ی داده‌های جدید و کامل‌تری که اسپاتیفای در دسامبر ۲۰۲۰ انجام داده است، مطالعات جدید و محدودی در این زمینه در سایت‌های مختلف انجام شده است. این مطالعات را در دو دسته‌ی «سیستم پیشنهاد دهی آهنگ» و «پیش‌بینی محبوبیت آهنگ» می‌توان قرار داد.

مطالعات در زمینه سیستم پیشنهاد دهی آهنگ

یکی از این مدل مطالعات در سایت Kaggle توسط vatsal mavani صورت پذیرفته است که با clustering آهنگ‌ها و ژانرها، گروه‌های شبیه به یکدیگری را ساخته و با این روش برای هر آهنگ می‌تواند پیشنهاداتی شبیه به خود آن، که داخل گروه این آهنگ هستند را بدهد و بدین ترتیب یک سیستم پیشنهاد دهی آهنگ را طراحی نموده است. {۱}

یکی دیگر از مطالعات انجام شده در این زمینه توسط sunku sowmya Sree در سایت medium صورت گرفته است که در آن با استفاده از Neighbourhood Collaborative Filtering و similarity metrics method، نزدیک‌ترین فاصله یک آهنگ را به آهنگی دیگر پیدا کرده و آن را به عنوان پیشنهاد به کاربر ارائه می‌دهد. {۲}

مطالعات در زمینه پیش‌بینی محبوبیت آهنگ

یکی از این مدل مطالعات در سایت Kaggle توسط steven tran صورت پذیرفته است که با استفاده از روش‌های RandomForestClassifier، DecisionTreeClassifier، KNeighborsClassifier، LogisticRegression و XGBClassifier، سعی در مدل‌سازی کرده است و در نهایت مدل ساخته شده توسط مدل‌های RandomForestClassifier و DecisionTreeClassifier با دقت ۹۲٪ و ۸۷.۵٪ بهترین عملکرد را داشته‌اند. {۳}

یکی دیگر از مطالعات انجام شده در این زمینه توسط sunku sowmya Sree در سایت medium صورت گرفته است که در آن با استفاده از Decision TreeRegressor، Decision Tree Regressor with Grid search CV و Random Forest سعی در مدل‌سازی کرده است و در نهایت مدل ساخته شده توسط روش Decision Tree Regressor with Grid search CV با دقت ۷۶.۶٪ بهترین عملکرد را داشته است. {۲}

در {۷}، کلاسیفیکیشن برای پیش بینی این‌که یک آهنگ، بسیار محبوب (hit) است یا خیر انجام شده است که بهترین مدل، random forest با دقت ۸۸ درصد به دست آمده است.

در {۸}، کلاسیفیکیشن برای پیش بینی اینکه یک آهنگ موجود در ۵۰ تای برتر، در دوماه بعد آیا جزو ۵۰ تای برتر خواهد بود یا خیر انجام شده است که بهترین مدل، SVM با هسته‌ی خطی بوده است که دقت ۷۰ درصد را داده است. همچنین در این منبع، به این نتیجه رسیده‌اند که ویژگی‌های موسیقیایی یک آهنگ در این کلاسیفیکیشن تاثیر گذار بوده است.

در {۹}، کلاسیفیکیشن برای پیش بینی اینکه یک آهنگ در ۵۰ تای برتر می‌آید یا خیر انجام شده است. بهترین مدل، SVM با هسته‌ی RBF بوده است که دقت ۸۰ درصدی داشته است. در این مقاله نیز، ویژگی‌های موسیقیایی موثر بوده‌اند.

در {۱۰}، کلاسیفیکیشن برای پیش بینی اینکه یک آهنگ در ژانر خاصی، محبوب می‌شود یا خیر، انجام شده است. بهترین مدل‌ها، random forest و gradient boosting با f1 score برابر با ۸۶ درصد بوده‌اند.

در {۱۱}، کلاسیفیکیشن برای پیش بینی اینکه یک آهنگ، آیا جزو ۱۰۰ تای برتر خواهد شد یا خیر، انجام شده است. بهترین مدل‌ها، شبکه‌ی عصبی با یک لایه‌ی پنهان و رگرسیون لاجیستیک بوده‌اند که دقت ۷۵ درصد داده‌اند.

فصل دوم: تشریح داده‌ها

داده‌های مورد استفاده شامل چندین دیتاست هستند که هر کدام یک کلید اصلی دارد و راه ارتباط دیتاست‌های مختلف از طریق کلیدهای مهمان می‌باشد، شکل کلی این دیتاست‌ها در زیر آمده‌است:

تصویر ۱ - ساختار کلی داده‌ها

تشریح دیتاست tracks

این دیتاست مهم‌ترین منبع ما بوده که در آن آهنگ‌های موجود در اسپاتیفای آورده شده است. حال برای درک بهتر آن به تشریح این دیتاست خواهیم پرداخت.

در قدم اول با استفاده از tracks.info متوجه خواهیم شد که دیتاست ما دارای ۵۸۶۶۷۲ داده (آهنگ) و ۲۰ ستون ویژگی می‌باشد. ۵ عدد از این ویژگی‌ها به صورت object، ۶ عدد از آن‌ها به صورت int64 و ۹ عدد دیگر به صورت float64 هستند. این تنوع به ما نشان می‌دهد که در مراحل آینده باید به یکپارچه سازی نوع این ویژگی‌ها بپردازیم. همچنین تنها ۷۱ اسم آهنگ در این دیتاست مفقود است که مقدار خوبی است.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 586672 entries, 0 to 586671
Data columns (total 20 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                586672 non-null  object
 1   name              586601 non-null  object
 2   popularity        586672 non-null  int64
 3   duration_ms       586672 non-null  int64
 4   explicit          586672 non-null  int64
 5   artists           586672 non-null  object
 6   id_artists        586672 non-null  object
 7   release_date      586672 non-null  object
 8   danceability      586672 non-null  float64
 9   energy            586672 non-null  float64
 10  key               586672 non-null  int64
 11  loudness          586672 non-null  float64
 12  mode              586672 non-null  int64
 13  speechiness       586672 non-null  float64
 14  acousticness      586672 non-null  float64
 15  instrumentalness  586672 non-null  float64
 16  liveness          586672 non-null  float64
 17  valence           586672 non-null  float64
 18  tempo             586672 non-null  float64
 19  time_signature    586672 non-null  int64
dtypes: float64(9), int64(6), object(5)
```

تصویر ۲ - اجرای tracks.info

در قدم بعدی با استفاده از tracks.describe متوجه خواهیم شد که این داده‌ها به صورت کمی در چه شرایطی قرار دارند. در این تحلیل متوجه خواهیم شد که scale مقادیر ویژگی‌ها با هم تفاوت شایانی داشته و نیازمند استاندارد کردن آن‌ها وجود دارد. برای مثال popularity از ۰ تا ۱۰۰ مقدار گرفته ولی energy از ۰ تا ۱ مقدار دارد.

	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
count	586672.000000	5.866720e+05	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000	586672.000000
mean	27.570053	2.300512e+05	0.044086	0.563594	0.542036	5.221603	-10.206067	0.658797	0.104864	0.449863	0.113451	0.213935	0.552292	118.464857	3.873382
std	18.370642	1.265261e+05	0.205286	0.166103	0.251923	3.519423	5.089328	0.474114	0.179893	0.348837	0.266868	0.184326	0.257671	29.764108	0.473162
min	0.000000	3.344000e+03	0.000000	0.000000	0.000000	0.000000	-60.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	13.000000	1.750930e+05	0.000000	0.453000	0.343000	2.000000	-12.891000	0.000000	0.034000	0.096900	0.000000	0.098300	0.346000	95.600000	4.000000
50%	27.000000	2.148930e+05	0.000000	0.577000	0.549000	5.000000	-9.243000	1.000000	0.044300	0.422000	0.000024	0.139000	0.564000	117.384000	4.000000
75%	41.000000	2.638670e+05	0.000000	0.686000	0.748000	8.000000	-6.482000	1.000000	0.076300	0.785000	0.009550	0.278000	0.769000	136.321000	4.000000
max	100.000000	5.621218e+06	1.000000	0.991000	1.000000	11.000000	5.376000	1.000000	0.971000	0.996000	1.000000	1.000000	1.000000	246.381000	5.000000

تصویر ۳ - اجرای tracks.describe

توضیح ویژگی‌های دیتاست tracks

این دیتاست از ۴ مدل ویژگی تشکیل شده است:

۱. اصلی:

۱٫۱. Id: این ویژگی به صورت منحصر به فرد برای هر آهنگ توسط اسپاتیفای ساخته شده است.

۲. عددی:

۲٫۱. Acousticness: میزان آکوستیک بودن یک آهنگ که در بازه ۰ تا ۱ نمره داده شده است.

۲٫۲. Danceability: میزان رقص پذیر بودن یک آهنگ که در بازه ۰ تا ۱ نمره داده شده است.

۲٫۳. Energy: میزان انرژی یک آهنگ که در بازه ۰ تا ۱ نمره داده شده است.

۲٫۴. Duration_ms: مدت زمان یک آهنگ که معمولا در بازه ۲۰۰ تا ۳۰۰ هزار میلی ثانیه قرار دارد.

۲/۵. Instrumentalness: میزان بهره‌گیری از ساز در آهنگ که در بازه ۰ تا ۱ نمره داده شده است.

۲/۶. Valence: میزان انرژی مثبت هر آهنگ که در بازه ۰ تا ۱ نمره داده شده است.

۲/۷. Popularity: میزان محبوبیت یک آهنگ که در بازه ۰ تا ۱۰۰ نمره داده شده است.

۲/۸. Tempo: ریتم هر آهنگ که معمولا در بازه ۵۰ تا ۱۵۰ قرار دارد.

۲/۹. Liveness: میزان طبیعی و بدون دستگاه (آمپلی فایر و...) بودن یک آهنگ که در بازه ۰ تا ۱ نمره داده شده است.

۲/۱۰. Loudness: میزان بلندی هر آهنگ که معمولا در بازه ۶۰- تا ۰ قرار دارد.

۲/۱۱. Speechiness: میزان حضور شعر و خواننده در آهنگ که در بازه ۰ تا ۱ قرار دارد.

۳. صفر و یکی:

۳/۱. Mode: ویژگی تخصصی موسیقیایی که ۰ به معنای ماینور (minor) و ۱ به معنای ماژور (major) می‌باشد.

۳/۲. Explicit: به معنای استفاده از محتوای صریح و نامناسب برای برخی افراد که ۰ به معنای عدم استفاده و ۱ به معنای استفاده از آن در آهنگ می‌باشد.

۴. گسسته (categorical):

۴/۱. Key: ویژگی تخصصی موسیقیایی مربوط به اکتاو هر آهنگ که در بازه ۰ تا ۱۱ قرار دارد.

۴/۲. Timesignature: ویژگی تخصصی موسیقیایی مربوط به نت هر آهنگ که در بازه ۰ تا ۵ قرار دارد.

۴/۳. Artists: نام خواننده یا خوانندگان یک آهنگ.

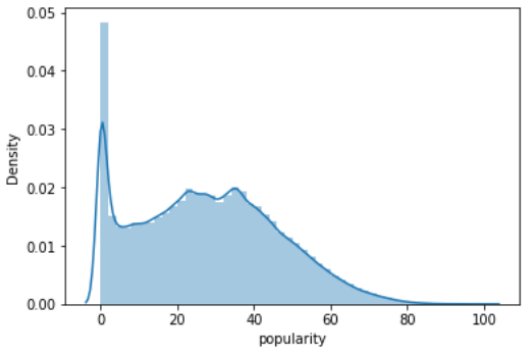
۴/۴. Id_artists: id منحصر به فرد هر خواننده یک آهنگ.

۴/۵. Release_date: تاریخ انتشار هر آهنگ که به صورت yy/mm/dd قرار دارد.

۴/۶. Name: نام آهنگ.

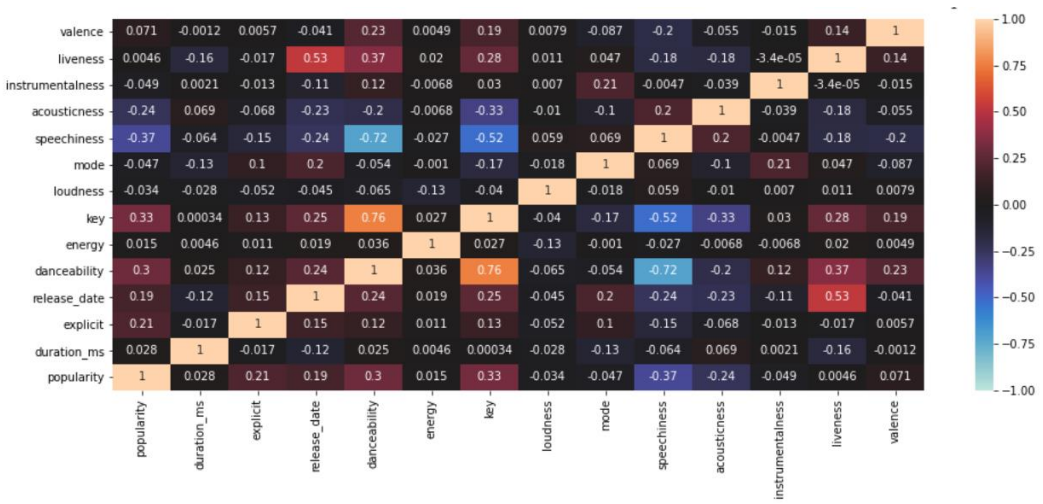
مصورسازی دیتاست tracks

در این قسمت نموداری از توزیع popularity خواهیم دید که به نظر می‌رسد از دو قسمت نرمال تشکیل شده است. همچنین همانطور که انتظار می‌رود و از نمودار زیر نیز قابل مشاهده و دریافت است، اکثر آهنگ‌ها محبوبیت پایینی داشته و کم گوش داده می‌شوند (اکثرا اصلا گوش داده نمی‌شوند). این تحلیل به ما نشان می‌دهد که این داده‌ها از توزیع نرمال پیروی نمی‌کنند.



تصویر ۴ - توزیع popularity در دیتاست tracks

از هیت مپ روابط زیر به نکاتی مانند رابطه عکس انرژی آهنگ با آکوستیک بودن آن و همچنین رابطه مثبت انرژی آهنگ با بلندی آن، می‌توان پی برد. البته چون در این قسمت محبوبیت خوانندگان و سال پخش آهنگ نیامده بنابراین اساسی‌ترین رابطه‌ها با محبوبیت آهنگ وجود ندارند که در قسمت پیش‌پردازش این مشکل حل شده و روابط نشان داده خواهند شد.



تصویر ۵ - هیت مپ روابط در دیتاست tracks

تشریح دیتاست data by year

در قدم اول با استفاده از dby.info متوجه خواهیم شد که دیتاست ما دارای ۱۰۰ داده (سال) و ۱۴ ستون ویژگی می‌باشد. ۳ عدد از آن‌ها به صورت int64 و ۱۱ عدد دیگر به صورت float64 هستند. همچنین هیچ داده مفقودی در این دیتاست وجود ندارد که بسیار خوب است.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   mode              100 non-null    int64
 1   year              100 non-null    int64
 2   acousticness      100 non-null    float64
 3   danceability      100 non-null    float64
 4   duration_ms       100 non-null    float64
 5   energy            100 non-null    float64
 6   instrumentalness  100 non-null    float64
 7   liveness          100 non-null    float64
 8   loudness          100 non-null    float64
 9   speechiness       100 non-null    float64
 10  tempo             100 non-null    float64
 11  valence           100 non-null    float64
 12  popularity        100 non-null    float64
 13  key               100 non-null    int64
dtypes: float64(11), int64(3)
```

تصویر ۶ - اجرای dby.info

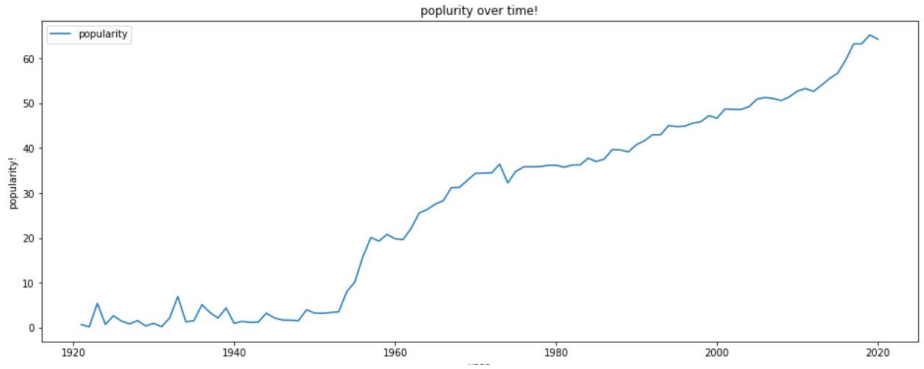
در قدم بعدی با استفاده از dby.describe متوجه خواهیم شد که این داده‌ها به صورت کمی در چه شرایطی قرار دارند. در این تحلیل متوجه خواهیم شد که scale مقادیر ویژگی‌ها، مانند دیتاست tracks، با هم تفاوت شایانی داشته و نیازمند scale کردن آن‌ها وجود دارد.

	mode	year	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence	popularity	key
count	100.0	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.0000
mean	1.0	1970.500000	0.556317	0.536783	227296.752234	0.452705	0.193582	0.208224	-11.969054	0.105861	116.015674	0.532120	27.376065	3.7900
std	0.0	29.011492	0.275358	0.052356	25630.048065	0.161738	0.122488	0.017903	3.105610	0.082128	5.669645	0.057809	20.703197	3.5627
min	1.0	1921.000000	0.219931	0.414445	156881.657475	0.207948	0.016376	0.168450	-19.275282	0.049098	100.884521	0.379327	0.140845	0.0000
25%	1.0	1945.750000	0.289516	0.500800	210889.193536	0.280733	0.103323	0.197509	-14.189232	0.064244	111.718626	0.497174	3.298200	0.0000
50%	1.0	1970.500000	0.459190	0.540976	235520.850833	0.495997	0.127644	0.206074	-11.773061	0.085763	117.455548	0.541503	33.619250	2.0000
75%	1.0	1995.250000	0.856711	0.570948	247702.738058	0.598008	0.276707	0.218493	-9.950542	0.104438	120.606644	0.570080	44.943375	7.0000
max	1.0	2020.000000	0.962607	0.692904	267677.823086	0.681778	0.581701	0.264335	-6.595067	0.490001	124.283129	0.663725	65.256542	10.0000

تصویر ۷ - اجرای dby.describe

مصورسازی دیتاست data by year

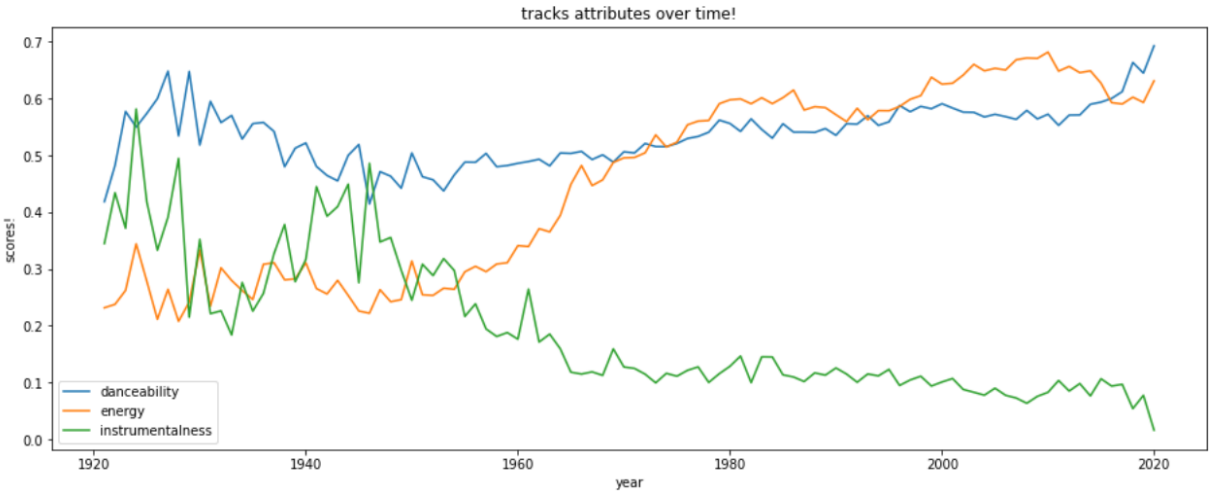
نمودار محبوبیت آهنگ‌ها در طول زمان، اطلاعات خاصی به ما نمی‌دهد چرا که می‌دانیم به طور طبیعی اقبال مردم به یک آهنگ قدیمی کمتر است، از این رو این پیش‌فرض با رسم نمودار و مشاهده تایید می‌شود.



تصویر ۸ - نمودار محبوبیت آهنگ‌ها در طول زمان

نمودار سه ویژگی آهنگ‌ها (danceability، instrumentalness و energy) در طول زمان سه نکته مهم را به ما نشان می‌دهد:

- روند صعودی "رقص پذیری" آهنگ‌ها که در سال ۲۰۲۰ در اوج آن هستیم.
- روند نزولی "استفاده از ساز" در آهنگ ها که در سال ۲۰۲۰ در قعر آن هستیم که علت آن نیز می‌تواند الکترونیکی شدن سازها و آهنگ‌ها باشد.
- روند حدودا صعودی "انرژی" آهنگ‌ها که از ۲۰۰۰ تا ۲۰۱۰ به خاطر روی بورس بودن موسیقی تکنو و دیسکو در اوج بوده است.



تصویر ۹ - ویژگی آهنگ‌ها در طول زمان

تشریح دیتاست artists

در قدم اول با استفاده از artists.info متوجه خواهیم شد که دیتاست ما دارای ۱۱۰۴۳۴۹ داده (خواننده) و ۵ ستون ویژگی می‌باشد. ۳ عدد از آن‌ها به صورت object، ۱ عدد از آن‌ها به صورت int64 و ۱ عدد دیگر به صورت float64 هستند. همچنین تنها ۱۳ داده در ستون follower مفقود در این دیتاست وجود دارد که بسیار خوب است.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1104349 entries, 0 to 1104348
Data columns (total 5 columns):
 #   Column      Non-Null Count    Dtype  
---  ------      --------------    -----  
 0   id          1104349 non-null  object 
 1   followers   1104336 non-null  float64
 2   genres      1104349 non-null  object 
 3   name        1104349 non-null  object 
 4   popularity  1104349 non-null  int64  
dtypes: float64(1), int64(1), object(3)
```

تصویر ۱۰ - اجرای artists.info

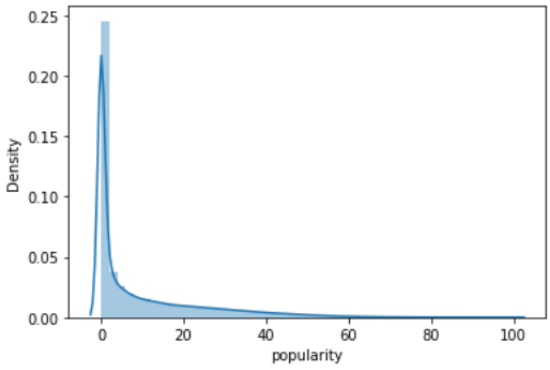
در قدم بعدی با استفاده از artists.describe متوجه خواهیم شد که این داده‌ها به صورت کمی در چه شرایطی قرار دارند.

	followers	popularity
count	1.104336e+06	1.104349e+06
mean	1.074304e+04	9.083884e+00
std	2.609554e+05	1.376310e+01
min	0.000000e+00	0.000000e+00
25%	1.000000e+01	0.000000e+00
50%	5.900000e+01	2.000000e+00
75%	4.510000e+02	1.400000e+01
max	7.890023e+07	1.000000e+02

تصویر ۱۱ - اجرای artists.describe

مصورسازی دیتاست artists

نمودار توزیع محبوبیت خوانندگان به ما نشان می‌دهد که اکثر خوانندگان دارای محبوبیت ۰ و زیر ۴۰ هستند.



تصویر ۱۲ - توزیع محبوبیت خوانندگان

اطلاعات جالب

- محبوب‌ترین آهنگ اسپاتیفای: peaches از justin Bieber

	id	name	popularity	duration_ms	explicit	artists
93802	4iJyoBOLtHqaGxP12qzhQI	Peaches (feat. Daniel Caesar & Giveon)	100	198082	1	['Justin Bieber', 'Daniel Caesar', 'Giveon']

تصویر ۱۳ - محبوب‌ترین آهنگ اسپاتیفای

- رقص‌پذیرترین آهنگ اسپاتیفای: Puisorul cafeniu از Malina Olinescu

	id	name	popularity	duration_ms	explicit	artists	id_artists	release_date	danceability
418558	4tq7Q9bTkLqzXNUi8PVmk2	Puisorul cafeniu	15	84707	0	['Malina Olinescu']	['6KZH1ER38F5smKpXKmLRwb']	2002-01-01	0.991

تصویر ۱۴ - رقص‌پذیرترین آهنگ اسپاتیفای

- یافتن آهنگ‌های ابی در اسپاتیفای و محبوب‌ترین آهنگ ابی: قبله

	id	name	popularity
233902	0pZqMBwwQfJ93djJBFAijs	Golvajeh	31
260141	3TIP50FTm69aaLaQGuQkpV	Pichak	31
300235	5PwZD0bbcwM9xCwIGA2muS	Gheseh Eshgh	38
300281	05DB7X5RlVGBzNff2MquhC	Nazi Naz Kon	36
300294	7c9vvLrzTV5KhhaOcS5ztP	Pooste Shir	34
300956	75wR9u92iSZtRuLFQSkgjF	Ghebleh	39
301209	4Fyb2OMEx1xBiSrO7FsGcO	Atreh Toh	37
364417	04UfIl2w4nrqX34USAzkGw	Shabzadeh	30
364838	4537oC0JfzmGXi3r3TyCCK	Madad Rangi	30
365756	3UOWPDqltRvOviBKQepc9c	Hezaro Yek Shab	32
447011	66iiwXdtywywuHi8lJsZx8	Khallj	28
538942	6GO5HdKsYmqUCgAnHkqPcU	Kee Ashkato Pak Mikoneh	34
564360	0nN0uILFZJNOUbvde9atUi	Koohe Yakh	12

تصویر ۱۵ - آهنگ‌های ابی در اسپاتیفای

- یافتن آهنگ‌های گوگوش در اسپاتیفای و محبوب‌ترین آهنگ گوگوش: پل

	id	name	popularity
235050	5wogkX0KqHJBbXwEDroPOQ	Pol	38
365101	4svcJnKqBZNExCkXC9hM8t	Mano Tou	33
365131	6RnSOAdvpR4rzLU70BgmFd	Kavir	31
366693	6VbjG4GCKlofrkUuTeHaWE	Gharibe Ashena	36
538647	1VaLqJKV7l4holKlgHWknd	Dou Mahi	29
538764	6jzwYUwyXpJVd1XUQNGyrc	Man Aamadeh-Am	32
538948	3m7MznkIFJv9xV3gF0iKAT	Do Panjereh	34

تصویر ۱۶ - آهنگ‌های گوگوش در اسپاتیفای

فصل سوم: پیش‌پردازش داده‌ها

اصلاح مقادیر مفقود

در دیتاست tracks مجموعا، ۷۱ داده در ستون name مقدار مفقود دارند، از آنجا که این ستون در ساخت مدل، حذف خواهد شد و به کار نخواهد آمد، بنابراین نیازی به نگرانی نیست.

در دیتاست artists مجموعا ۱۳ follower مفقود می‌باشد که این مقادیر را ۰ در نظر می‌گیریم.

در دیتاست data by year اصلا داده مفقودی نداریم که عالی است.

```
tracks.isna().sum(axis=0)
```

```
id                0
name             71
popularity        0
duration_ms       0
explicit          0
artists           0
id_artists        0
release_date      0
danceability      0
energy            0
key               0
loudness          0
mode              0
speechiness       0
acousticness      0
instrumentalness  0
liveness          0
valence           0
tempo             0
time_signature    0
dtype: int64
```

تصویر ۱۷ - داده‌های مفقود دیتاست tracks

تبدیل تاریخ به سال

در دیتاست tracks ویژگی تاریخ انتشار آهنگ (release_date) را داریم که به صورت تاریخ (yy-mm-dd) بوده و عدد صحیح نمی‌باشد و با توجه به عدد بودن دیگر مقادیر ویژگی‌ها، باعث ایجاد خلل در روند مدل‌سازی خواهد شد. بعضی از تاریخ انتشارها، فقط دارای سال هستند، بنابراین برای درست کردن این مشکل، ویژگی release_date را با برداشتن فقط سال انتشار تبدیل به year می‌کنیم و ستون release_date را که دیگر به آن نیاز نداریم، حذف می‌کنیم.

شناسایی و حذف داده‌های تکراری

برای پیدا کردن داده‌های تکراری در دیتاست tracks باید به این نکته توجه کنیم که نحوه جمع‌آوری داده‌ها به صورتی بوده است که برای هر آهنگ یک id در نظر گرفته شده است. بنابراین حتی در صورتی که یک آهنگ چندین بار تکرار شده باشد نیز در هر تکرار یک id منحصر به فرد دارد. با این شرایط ما باید پس از حذف id آهنگ به پیدا و حذف کردن این داده‌های تکراری بپردازیم که مجموعا ۱۶۲۱ مقدار می‌باشد و در فصل بعدی و پس از حذف این ستون‌ها این کار را انجام دادیم.

استاندارد سازی داده‌ها

یکسان نبودن برد متغیرهای مختلف مساله‌ای است که در کار برخی مدل‌های پیش بینی و کلاسیفیکیشن خلل ایجاد می‌کند. بنابراین همه‌ی متغیرهای عددی دیتاست tracks را به روش z-score استاندارد می‌کنیم. البته ستون clf_pop که در آینده، در کلاسیفیکیشن استفاده خواهد شد را استاندارد نمی‌کنیم تا ۴ مقدار ۱، ۲، ۳ و ۴ آن، حفظ شود.

پس از استاندارد سازی، می‌بینیم که ستون year_mode، به دلیل انحراف معیار صفر، همگی مقدار مفقود گرفتند، بنابراین این ستون را حذف می‌کنیم.

البته این مرحله را پس از اضافه کردن یک سری ویژگی جدید به دیتاست (که در فصل چهارم توضیح دادیم) انجام دادیم.

حذف نمودن داده‌های پرت

یکی از روش‌های شناسایی داده‌های پرت، استفاده از معیار ریاضیاتی z_score می‌باشد. در این روش، در همه‌ی ستون‌ها به جز clf_pop، داده‌هایی که z-score بالاتر از ۳ یا کمتر از -۳ داشتند را حذف می‌کنیم. {۴} که در نهایت، به ۴۶۱۷۹۱ سطر، داده خواهیم رسید و با استفاده از این دیتاست کاملا تمیز شده، به مدل سازی در حیطه‌ی کلاسیفیکیشن و پیش بینی می‌پردازیم.

البته این مرحله را نیز پس از اضافه کردن یک سری ویژگی جدید به دیتاست (که در فصل چهارم توضیح دادیم) انجام دادیم.

فصل چهارم: شناسایی و استخراج داده‌های مورد نظر جهت داده‌کاوی

استخراج میانگین محبوبیت و دنبال‌کننده خوانندگان هر آهنگ

از آنجایی که با توجه به ویژگی‌های موجود در دیتاست، نبود محبوبیت خواننده هر آهنگ یکی از مشکلات اصلی دیتاست tracks بود و ما می‌دانیم که قطعا مهم‌ترین ویژگی برای محبوبیت هر آهنگ، خواننده آن آهنگ است پس بر آن شدیم تا این مشکل را با قرار دادن میانگین محبوبیت خواننده هر آهنگ حل کنیم.

در داده هر آهنگ، خواننده یا خوانندگان آن مشخص هستند و این خوانندگان در دیتاست artists با یک id و میزان محبوبیت و دنبال کننده نیز مشخص شده‌اند. بنابراین در این مرحله با پیدا کردن خواننده و یا خوانندگان هر آهنگ از دیتاست tracks و سپس پیدا کردن محبوبیت و تعداد دنبال کننده خواننده و یا میانگین آن برای خوانندگان آن آهنگ در دیتاست artists و سپس اضافه کردن دو ویژگی جدید با نام‌های artists_mean_popularity و artists_mean_followers، بر این مشکل فائق آمدیم.

البته از آنجا که در دیتاست tracks آهنگ‌ها تا سال ۲۰۲۱ آمده‌اند و در data by year و data by artists تا سال ۲۰۲۰، آهنگ‌هایی که خوانندگانشان در سال ۲۰۲۱ برای اولین بار آهنگ، منتشر کرده‌اند، این دو فیچر جدید را ندارد که با بررسی، فهمیدیم تعداد این آهنگ‌ها، ۶۰ عدد می‌باشد که آن‌ها را از دیتاست حذف می‌کنیم.

اضافه کردن ۱۳ ویژگی مربوط به سال انتشار آهنگ

همانطور که در قسمت تشریح داده‌ها نیز توضیح داده شد، ارتباط مثبتی بین سال انتشار آهنگ با محبوبیت آن وجود دارد، بدین صورت که هر چه یک آهنگ جدیدتر باشد به احتمال زیادی محبوب‌تر نیز می‌باشد. بنابراین ما به این نکته پی بردیم که هر آهنگ را با سال انتشار خود نیز می‌توانیم توضیح دهیم.

برای این کار با پیدا کردن سال انتشار هر آهنگ در دیتاست tracks و سپس پیدا کردن میانگین ویژگی‌های آهنگ‌ها در آن سال مشخص از دیتاست data by year و اضافه کردن این ۱۳ ویژگی به هر آهنگ به صورت year_feauters، بر این مشکل نیز فائق آمدیم.

البته در این مرحله نیز، به علت نبود سال ۲۰۲۱ در دیتاست data by year، حدود ۶۰۰۰ آهنگ از دیتاست ما حذف شدند.

اضافه نمودن مقادیر دسته بندی

برای بخش کلاسیفیکیشن، در این قسمت فیچر clf-pop را به صورت زیر اضافه می‌کنیم:

برای این کار، محبوبیت را به ۴ دسته تقسیم نمودیم، آن‌هایی که از ۰ تا ۲۵ محبوب هستند، ۱، از ۲۶ تا ۵۰، ۲، از ۵۱ تا ۷۵، ۳ و از ۷۶ تا ۱۰۰ عدد ۴ را اختصاص دادیم.

حذف ویژگی‌های زائد

پس از اضافه نمودن فیچرهای مورد نیاز، در ساخت مدل‌های دسته بندی و پیش بینی به ستون‌های name، Id، artists و Id_artists نیازی نداریم که آن‌ها را حذف می‌کنیم تا در نهایت به ۳۰ فیچر برای پیش بینی، یک ستون محبوبیت و یک ستون clf_pop که برای کلاسیفیکیشن استفاده خواهد شد برسیم.

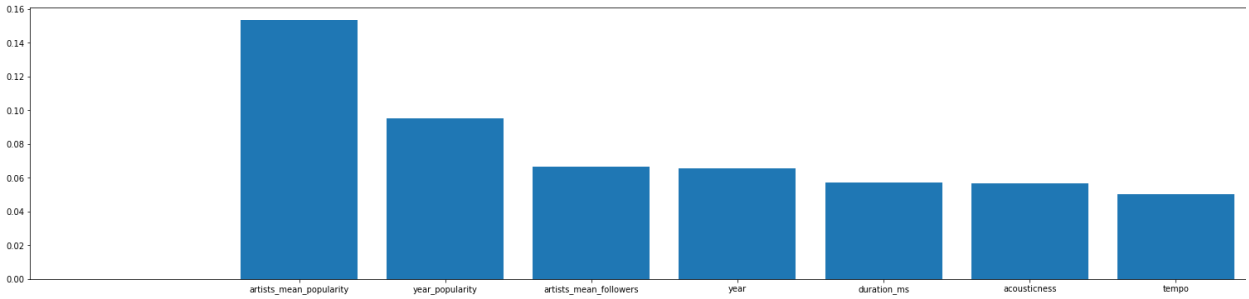
ارزش هر ویژگی (coef)

حال، برای اینکه ویژگی‌های مهم را شناسایی کنیم، از معیار همبستگی هر ستون با ستون محبوبیت، استفاده می‌کنیم که در مدل‌های پیش بینی خطی معیار خوبی برای اهمیت ویژگی‌هاست، که در زیر مشاهده می‌کنیم:

```
popularity                1.000000e+00
year_popularity           5.406300e-01
year                      5.301389e-01
artists_mean_popularity   5.210972e-01
year_loudness             5.174802e-01
year_energy               4.909721e-01
year_danceability         4.600600e-01
year_tempo                4.194020e-01
artists_mean_followers    3.013995e-01
loudness                  2.948479e-01
energy                    2.507709e-01
year_speechiness          1.770361e-01
danceability              1.667925e-01
duration_ms               1.005981e-01
time_signature            7.422275e-02
tempo                     5.521036e-02
year_duration_ms          5.224478e-02
year_key                  4.191245e-02
key                       1.676464e-02
speechiness               6.580582e-03
valence                   4.595697e-04
explicit                  1.436978e-13
mode                     -1.586573e-02
liveness                 -6.202065e-02
year_valence             -1.687950e-01
instrumentalness         -1.807765e-01
acousticness             -3.103064e-01
year_liveness            -3.618694e-01
year_instrumentalness    -4.852811e-01
year_acousticness        -4.866636e-01
Name: popularity, dtype: float64
```

تصویر ۱۸ - همبستگی ویژگی‌ها

برای سنجش اهمیت ویژگی‌ها در مدل‌های کلاسیفیکیشن می‌توانیم یک مدل bagging classifier بسازیم و اهمیت ویژگی‌ها را در این بسنجیم. این اهمیت در مدل‌های کلاسیفیکیشن کاربرد دارد که در پایین ۷ تا از مهم‌ترین ویژگی‌ها را مشاهده می‌کنیم:



تصویر ۱۹ - اهمیت ویژگی‌ها توسط مدل bagging classifier

البته به دلیل تعداد کم ویژگی‌ها و از طرفی برای از دست ندادن اطلاعات مفید، همه ویژگی‌ها را حفظ می‌کنیم.

فصل پنجم: شناسایی و ارزیابی الگوهای پنهان در داده‌ها

ساخت مدل

برای ساخت مدل، دو روش کلی در پیش داشتیم که در ادبیات نیز با هر دو روش مدل‌هایی توسعه یافته است. روش اول، پیش بینی محبوبیت است و روش دوم، کلاسیفای کردن محبوبیت. به نظر می‌رسید که از آنجا که محبوبیت آهنگ، ممکن است تابع بسیاری از شرایطی باشد در دیتاست وجود ندارد، روش‌های کلاسیفیکیشن نتیجه‌ی بهتری بدهد، برای مثال، محبوبیت بعضی از آهنگ‌ها، به شرایط اجتماعی و فرهنگی زمان خود بستگی دارد، یا به کمپین‌های تبلیغاتی بزرگ یا حتی، موضوعات روز سیاسی.

همچنین محبوبیت آهنگ، بسیار به ذائقه‌ی مردم بستگی دارد که در گذر زمان، بر اساس شرایط مختلفی تغییر می‌کند، از این رو شاید پیش بینی دقیق محبوبیت آهنگ، مانند پیش بینی دقیق نتیجه‌ی فوتبال آنقدر نتایج خوبی نداشته باشد.

به همین دلیل، ابتدا مدل‌های مختلف رگرسیونی و غیر رگرسیونی را برای پیش بینی محبوبیت آهنگ به کار بردیم تا ببینیم نتایج این کار به چه صورت است و اگر، دقت مدل‌ها به نسبت پایین است به سراغ مدل‌های کلاسیفیکیشن برویم.

پیش بینی محبوبیت

در دیتاست مورد بررسی، محبوبیت هر آهنگ، عددی طبیعی از ۰ تا ۱۰۰ می‌باشد، برای پیش بینی مدل‌های زیر را به کار بردیم:

- Lasso
- PassiveAggressiveRegressor
- Ridge
- KneighborsRegressor
- LassoLars
- BayesianRidge
- RidgeCV
- MLPRegressor
- ElasticNet
- GradientBoostingRegressor
- AdaBoostRegressor
- RandomForestRegressor(max_depth=10)
- RandomForestRegressor(max_depth=30)
- LinearRegression

در این قسمت در ابتدای انجام کار توسط روش پیش بینی متوجه کاستی‌های این روش برای اجرای رو دیتاست خود شدیم. سلیقه‌ای بودن موسیقی و محبوبیت هر آهنگ، تاثیرپذیری از موارد بیرونی مانند معروفیت موردی یک آهنگ به دلایل اجتماعی و نه لزوما موسیقیایی و بسیاری موارد تاثیرگذار بیرونی و غیر موسیقیایی دیگر، می‌تواند دلیلی بر ضعف این روش برای

دیتاست ما باشد. بنابراین طبق این پیش فرض و نتایجی که دیدیم، متوجه شدیم که این روش برای کار ما مناسب نبوده و به اندازه روش کلاسیفیکیشن به تنظیم پارامترهای آن نپرداختیم.

معیار امتیازدهی و ارزیابی مدل‌ها، MAE و R2 بود، همچنین از روش hold_out استفاده نمودیم و برای اینکه مسئله‌ی over_fitting رخ ندهد، برای هر مدل، ۱۰۰ بار این روش را اجرا کردیم و امتیازها را میانگین گرفتیم که در جدول زیر، نتایج این کار ارائه داده شده است:

مدل	میانگین MAE	میانگین R2
Lasso	۰.۵۲	۰.۵۱
PassiveAggressiveRegressor	۰.۷۲	۰.۰۹
Ridge	۰.۵۲	۰.۵۱
KNeighborsRegressor (n_neighbors=20)	۰.۴۹	۰.۵۵
KNeighborsRegressor (n_neighbors=7)	۰.۵۰	۰.۵۲
LassoLars	۰.۵۲	۰.۵۱
BayesianRidge	۰.۵۲	۰.۵۱
RidgeCV	۰.۵۲	۰.۵۱
MLPRegressor	۰.۴۸	۰.۵۸
ElasticNet	۰.۵۲	۰.۵۱
GradientBoostingRegressor	۰.۴۸	۰.۵۶
AdaBoostRegressor	۰.۵۹	۰.۴۲
RandomForestRegressor(max_depth=10)	۰.۴۶	۰.۵۹
RandomForestRegressor(max_depth=30)	۰.۴۳	۰.۶۴
LinearRegression	۰.۵۲	۰.۵۱

جدول۱ - عملکرد مدل‌های مختلف پیش بینی

بهترین مدل، randomforrestregressor با max_depth=30 می‌باشد که امتیاز r2 برابر با ۶۴٪ می‌دهد.

در این بخش، به دلیل اینکه در کل، دقت مدل‌ها بالا نبود، از این رو دیگر به تنظیم پارامترها نپرداختیم و صرفا می‌خواستیم ببینیم که آیا پیش بینی گزینه‌ی خوبی است یا کلاسیفیکیشن. بنابراین به این نتیجه رسیدیم که اساسا پیش بینی عددی، برای محبوبیت آهنگ، دقت بالایی ندارد، به همین دلیل به سراغ مدل‌های کلاسیفیکیشن رفتیم.

کلاسیفیکیشن

مدل‌های مورد استفاده در این بخش، مدل‌های زیر بودند:

- Decision Tree
- Random Forest
- Bagging
- KNN
- Naïve Bayesian

در این بخش، برای هر مدل تنظیم پارامتر انجام خواهیم داد، همچنین برای هر پارامتر خاص، ۱۰۰ بار مدل را اجرا می‌کنیم و در نهایت، معیارهای ارزیابی به دست آمده را میانگین گرفته و ارائه می‌دهیم.

برای هر مدل نیز، تنظیم پارامترها را انجام دادیم که در زیر به تشریح هر مدل را آورده‌ایم؛ همچنین، معیار ارزیابی مدل‌ها در این بخش، log_loss و f1_score می‌باشد که average آن به دلیل نامتعادل بودن دیتاست، از نوع micro قرار می‌دهیم که همان معیار accuracy می‌شود، در ارزیابی مدل‌ها، معیار f1_score برایمان اولویت دارد. {۵}

Decision tree classifier

برای رسیدن به بهترین مدل ممکن، در این بخش آرگومان max_depth یا حداکثر عمق درخت را از ۵ تا ۳۰، ۵ تا ۵ تا تغییر داده تا به بهترین مدل برسیم که نتایج آن را در زیر می‌بینید:

حداکثر عمق درخت	F1_score	Log_loss
۵	۰.۶۹	۰.۷۰
۱۰	۰.۷۱	۰.۷۱
۱۵	۰.۷۰	۲.۰۵
۲۰	۰.۶۷	۶.۰۵
۲۵	۰.۶۴	۱۰.۰۲
۳۰	۰.۶۴	۱۱.۸۵

جدول۲ - نتایج مدل‌های مختلف decision tree

بهترین مدل در این بخش، حداکثر عمق ۱۰ را دارد.

Random Forest classifier

در این مدل، تعداد درخت‌هایی که قبل از کلاسیفای کردن می‌سازیم، پارامتر n_estimators می‌باشد، به طور کلی مقادیر بالاتر این پارامتر، عملکرد بهتری را به ما می‌دهد ولی سرعت مدل را پایین می‌آورد. {۶}

از این رو ما این پارامتر را ۵۰، ۱۰۰، ۱۵۰ و ۲۵۰ قرار می‌دهیم، همچنین max_features را برابر با auto می‌گذاریم که به معنای رادیکال تعداد فیچرهاست . در نهایت نتایج را به دست می‌آوریم:

N_estimators	F1_score	Log_loss
۵۰	۰.۸۲۴	۰.۴۷۶
۱۰۰	۰.۸۲۶	۰.۴۴۷
۱۵۰	۰.۸۲۷	۰.۴۳۹
۲۵۰	۰.۸۲۶	۰.۴۳۶

جدول۳ - نتایج مدل‌های مختلف random forest

همانطور که دیده شد، بهترین نتیجه با n-estimators ۱۵۰ بوده و f1_score، ۸۲.۷٪ می‌دهد که پیشرفت قابل ملاحظه‌ای نسبت به مدل decision tree می‌باشد.

Bagging classifier

مشابه با random forest، این الگوریتم هم پارامتر n_estimators را دارد، مقادیر این پارامتر را ۵۰، ۱۰۰، ۱۵۰ و ۲۰۰ می‌گذاریم:

N_estimators	F1_score	Log_loss
۵۰	۰.۸۵۹	۰.۴۳۱
۱۰۰	۰.۸۶۱	۰۳۸۷
۱۵۰	۰.۸۶۲	۰.۳۷۹
۲۵۰	۰.۸۶۲	۰.۳۶۳

جدول۴ - نتایج مدل‌های مختلف bagging

همانطور که مشاهده می‌شود، بهترین مدل امتیاز f1_score برابر ۸۶.۲٪ گرفت که پیشرفت خوبی نسبت به ۸۲.۷٪ مدل random forest می‌باشد.

Gaussian Naive Bayes classifier

اجرای این مدل، f1_score برابر با ۶۰.۴٪ و log_loss برابر با ۰.۹۰۱ داد.

K Neighbors Classifier

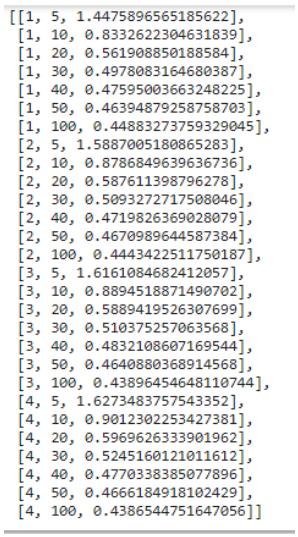
در این روش، پارامتر n_neighbors که تعداد نزدیک‌ترین همسایه‌هایی را که برای کلاسیفیکیشن انتخاب می‌شود، نشان می‌دهد از ۵ تا ۱۰۰ تغییر دادیم: ۵، ۱۰، ۲۰، ۳۰، ۴۰، ۵۰، ۱۰۰.

دو پارامتر دیگر در تنظیم پارامترها می‌تواند تغییر کند:

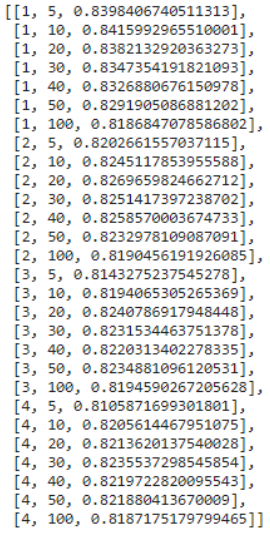
پارامتر weights: این آرگومان، می‌توان مقدار uniform یا distance را بگیرد. Uniform به این معناست که در همسایه‌های نزدیک داده‌ای که می‌خواهیم ببینیم به کدام کلاس یا دسته تعلق دارد، در رای گیری داده‌های همسایه‌ی آن، آن‌هایی که نزدیک هستند با آن‌هایی که دور هستند، ارزش رای برابر دارند اما در distance، نزدیک‌ترها، رای مهم‌تری دارند.

پارامترp: این پارامتر، در فرمول محاسبه‌ی فاصله کاربرد دارد، مقدار ۱ آن، فاصله‌ی منهتن و مقدار ۲ آن، فاصله‌ی اقلیدسی می‌باشد. این مقدار را از ۱ تا ۴ تغییر می‌دهیم.

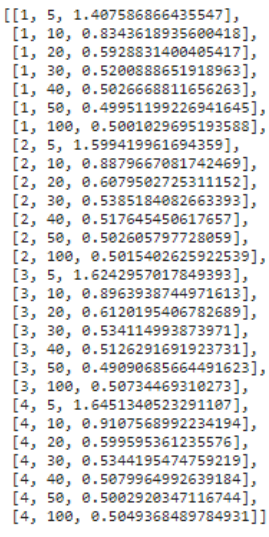
در نهایت، نتایج زیر در قالب یک لیست برای هر ترکیب ممکن ارائه شده است، عدد سمت چپ، p، عدد وسط نیز n_neighbors و عدد سمت راست، امتیاز مورد محاسبه است. بنابر تصاویر زیر بهترین مدل KNN، به ازای p=1، wights=distance و تعداد همسایه‌ی ۱۰ می‌باشد که f1_score برابر با ۸۴.۲٪ دارد.



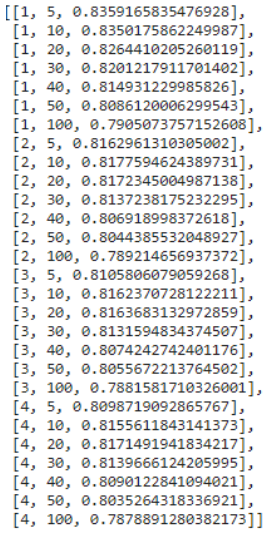
تصویر ۲۰ - معیار log_loss به ازای weights=distance در KNN classifier



تصویر ۲۱ - معیار f1_score به ازای weights=distance در KNN classifier



تصویر ۲۲ - معیار log_loss به ازای weights=uniform در KNN classifier



تصویر ۲۳ - معیار f1_score به ازای weights=uniform در KNN classifier

فصل ششم: ارائه نهایی الگوها و دانش کسب شده

با توجه به الگوهای شناسایی شده در فصل قبل و معیار ارزیابی آن‌ها که برای پیش‌بینی R2 و MAE و برای کلاسیفیکیشن معیارهای f1 score (همان accuracy) و log loss بود، به این نتیجه رسیدیم که مطابق انتظارمان مدل‌های پیش بینی عملکرد خوبی نداشته و بهترین مدل‌های ما از کلاسیفیکیشن می‌باشد.

به همین منظور مدل‌هایی با دقت بالای ۸۰٪ را به عنوان مدل‌های منتخب در نظر گرفتیم که بهترین آن‌ها bagging classifier با دقت ۸۶.۲٪ بود. در جدول زیر مدل‌های منتخب و بهترین دقت هر کدام را مشاهده می‌کنید:

مدل	Parameters	F1_score	Log_loss
Bagging classifier	۲۵۰	۰.۸۶۲	۰.۳۶۳
K Neighbors Classifier	۱، ۱۰، distance	۰.۸۴۲	۰.۸۳۳
Random Forest classifier	۱۵۰	۰.۸۲۷	۰.۴۳۹

جدول ۵ - مدل‌های منتخب

با توجه به بهترین مدلمان که bagging classifier می‌باشد، اهمیت ویژگی‌های مختلف را بر حسب این مدل به دست آوردیم که در زیر مشاهده می‌کنید:

```
[('artists_mean_popularity', 0.15318866992265728),
 ('year_popularity', 0.08441646733708845),
 ('year', 0.07144771708122573),
 ('artists_mean_followers', 0.0659138426524861),
 ('duration_ms', 0.05785270100188596),
 ('acousticness', 0.05643029173337085),
 ('tempo', 0.05013445805188514),
 ('speechiness', 0.050101480061893834),
 ('loudness', 0.049701759200340874),
 ('liveness', 0.049396780384019214),
 ('danceability', 0.04917504877687294),
 ('valence', 0.04756058835341494),
 ('energy', 0.04508978603297316),
 ('instrumentalness', 0.042526837913919466),
 ('key', 0.023502208940179556),
 ('year_duration_ms', 0.01664569772086888),
 ('year_acousticness', 0.01237697809606221),
 ('year_energy', 0.011762614324775019),
 ('year_danceability', 0.010452515580700006),
 ('year_valence', 0.007883560895751568),
 ('year_instrumentalness', 0.007458049962176127),
 ('year_liveness', 0.007331761026409467),
 ('year_speechiness', 0.007203658148426847),
 ('year_loudness', 0.005974564523376572),
 ('year_tempo', 0.005904150256001206),
 ('mode', 0.00482410587697561),
 ('time_signature', 0.0037055749858749044),
 ('year_key', 0.002038131158388115),
 ('explicit', 0.0)]
```

تصویر ۲۴ - اهمیت ویژگی‌ها بر اساس مدل برتر

از تصویر روبه‌رو نکات جالبی می‌توان دریافت کرد:

۱. تاثیرگذارترین ویژگی هر آهنگ برای محبوبیت آن، همانطور که خودمان نیز حدس زده بودیم و به همین دلیل این ویژگی را به دیتاست اضافه نمودیم، محبوبیت خود خواننده یا خوانندگان آن می‌باشد.

۲. یکی دیگر از عوامل تاثیرگذار بر محبوبیت آهنگ، همانطور که خودمان نیز حدس زده بودیم و به همین دلیل این ویژگی را به دیتاست اضافه نمودیم، محبوبیت متوسط آهنگ‌ها در سال انتشار آهنگ می‌باشد.

۳. یکی دیگر از عوامل تاثیرگذار بر محبوبیت آهنگ، همانطور که خودمان نیز حدس زده بودیم و به همین دلیل این ویژگی را به دیتاست اضافه نمودیم، سال انتشار آهنگ می‌باشد. بدینگونه که هر چه به سال‌های اخیر نزدیک می‌شویم، محبوبیت آهنگ‌ها نیز افزایش می‌یابد.

۴. متوسط میزان دنبال کننده خواننده آهنگ نیز، همانند محبوبیت آن خواننده و در همان راستا بر محبوبیت آهنگ تاثیرگذار است.

۵. طول آهنگ نیز طبق این تصویر تاثیر به‌سزایی بر محبوبیت آهنگ دارد.

۶. نکته جالب دیگر، تاثیر نداشتن محتوای نامناسب برای بعضی افراد (expilicit) بر محبوبیت آهنگ می‌باشد. این بدان معناست که استفاده از الفاظ نامناسب در یک آهنگ تاثیری بر محبوبیت آن نداشته و انگار هر چقدر افراد بخاطر این محتوا، آهنگ را گوش ندهند به همان مقدار این آهنگ توسط افراد دیگری شنیده می‌شود.

به طور کلی ویژگی‌هایی که در این پروژه به بررسی آن پرداختیم را می‌توان به ۳ دسته تقسیم نمود و از آن درس‌هایی گرفت:

- ویژگی‌های خواننده یا خوانندگان آهنگ: این دسته از ویژگی‌ها بیشترین تاثیر را بر محبوبیت آهنگ دارند. در این پروژه، ویژگی خواننده همان محبوبیت او بود، اما می‌توان این ویژگی‌های خواننده را به عوامل دیگری مانند زیبایی صدا، قدرت خوانندگی، قدرت رسانه‌ای، زیبایی چهره و هر ویژگی دیگری که مختص به آن خواننده است نیز تعمیم داد. همچنین این ویژگی‌های جدید و شخصی می‌تواند راهی برای بهبود مدل‌های ارائه شده باشد.
- ویژگی‌های مربوط به سال انتشار آهنگ: این دسته از ویژگی‌ها، تاثیر متوسط رو به پایینی بر محبوبیت آهنگ دارد. ترندهای موسیقیایی هر سال، زیر مجموعه‌ی این ویژگی‌ها می‌باشند. به نظر می‌رسد این ترندها، اطلاعات زیادی راجع به محبوبیت آهنگ‌ها می‌دهند؛ به بیان دیگر، ترند هر سال، تاثیر بر محبوبیت دارد. بنابراین می‌توانیم نتیجه بگیریم که برای کلاسیفای کردن محبوبیت یک آهنگ، هر چه داده‌های نزدیک‌تری از نظر زمانی داشته

باشیم، مدل قوی‌تری خواهیم داشت؛ پس، برای کلاسیفای کردن آهنگ‌های ۲ یا ۳ سال دیگر، استفاده از مدل ارائه شده تا ۲۰۲۰، منطقی نخواهد بود و خطای زیادی خواهیم داشت.

- ویژگی‌های موسیقیایی: این دسته از ویژگی‌ها، تاثیر متوسطی بر محبوبیت دارند که نشان می‌دهد، علاوه بر ترندهای سالانه در صنعت موسیقی و قدرت شخصی هنرمندان، هنوز هم یکی از عوامل مهم در محبوبیت آهنگ، ویژگی‌های موسیقیایی است که بعضی از آن‌ها اهمیت بیشتر و بعضی، اهمیت کمتری دارند. بنابراین می‌توان نتیجه گرفت، با کیفیت بودن یک آهنگ، در محبوبیت آن، جدا از موارد تبلیغاتی، می‌تواند موثر باشد.

فصل هفتم: نتیجه‌گیری و پیشنهادات

نتیجه‌گیری

از آنجایی که صنعت موسیقی، یکی از بزرگ‌ترین صنعت‌های سرگرمی به حساب می‌آید و سرمایه‌گذاری روی این صنعت هر روزه در حال افزایش می‌باشد و همچنین به این دلیل که پیروی از مد و جریان روز در این صنعت بسیار کاربرد دارد؛ بنابراین تحلیل عمیق آهنگ‌ها در اسپاتیفای، می‌تواند چشم‌انداز مناسبی به هنرمندان و سرمایه‌گذاران این عرصه برای فهم و درک بهتر جریان موجود در موسیقی و عوامل موفقیت یک آهنگ در این صنعت بزرگ بدهد.

در این نوشتار نیز سعی شده است که محبوبیت آهنگ‌ها در اسپاتیفای با استفاده از مدل‌های مختلف داده کاوی تخمین و پیش‌بینی شود.

همانطور که در ادبیات موضوع، بررسی شد، اکثر مدل‌های اعمال شده از جنس کلاسیفیکیشن بودند که دقت خوبی را می‌دادند. این موضوع قابل انتظار است، چرا که مدل‌های ذهنی انسان‌ها نیز تا حدودی بر این مبنا کار می‌کند، ما می‌توانیم به راحتی بنابر یک سری متغیرهای ذهنی، بگوییم یک آهنگ خاص، محبوب خواهد بود اما دقیقا نمی‌توانیم مقدار محبوبیت آن را بیان کنیم. همچنین، مدل‌های بر مبنای درخت تصمیم، معمولا نتایج خوبی دادند که نشان می‌دهد، این مدل‌ها، در کنار سادگی نسبی منطقشان، بسیار قدرتمند هستند.

در ابتدا، با ویژگی‌های موسیقیایی آهنگ‌ها، رگرسیون خطی زدیم که R2، ٪۲۱ شد. با اضافه کردن سال، این امتیاز را به ٪۲۷ رساندیم و در نهایت، با اضافه کردن ویژگی‌های موسیقیایی سال و ویژگی‌های خوانندگان، این امتیاز را به ٪۶۴ رساندیم. بنابراین ویژگی‌های مربوط به خوانندگان و سال، ویژگی‌های مهمی بودند. ۳ ویژگی مهم در کلاسیفیکیشن، محبوبیت خوانندگان، محبوبیت سال انتشار و سال انتشار بود که بر اهمیت ویژگی‌های مربوط به سال و خوانندگان تاکید می‌کند.

دو عامل که می‌تواند تاثیرات منفی بر عملکرد مدل‌ها داشته باشد، داده‌های پرت و استاندارد نبودن داده‌ها می‌باشد که با استاندارد سازی به روش z-score و حذف داده‌های پرت، این آثار منفی را خنثی نمودیم.

نتایج مدل‌ها با معیارهای مختلف، می‌تواند متفاوت باشد. برای مثال معیاری با f1-score بالاتر، log loss بالاتری داشته باشد، از این رو برای مقایسه‌ی معیارهای مختلف، همیشه باید بیش از یک معیار را مد نظر قرار دهیم. البته در این مسئله، اولویت را با معیار f1-score قرار دادیم.

همانطور که در این پروژه دیدیم، داده کاوی می‌تواند در تصمیم گیری‌های صنعت موسیقی، بسیار مفید باشد و باید منتظر هر چه عمومی‌تر شدن این علم در این صنعت گسترده باشیم.

پیشنهادات

- برای بهبود مدل‌ها، اضافه نمودن ویژگی‌های دیگر خوانندگان، مانند ویژگی‌های مربوط به صدا، چهره و ... می‌تواند موجب بهبود مدل شود.
- با کوچک کردن دیتاست از نظر زمانی، به ترندهای زمانی به‌روزتری می‌رسیم که می‌تواند موجب بهبود مدل شود، برای مثال به جای استفاده از داده‌های آهنگ‌های صد سال گذشته، آهنگ‌های مربوط به ۲۰ سال گذشته را در نظر بگیریم و از آن، برای پیش بینی محبوبیت آهنگ‌های آینده استفاده کنیم. البته هر چه جلوتر می‌رویم این ترندها کوتاه‌تر شده و سریع‌تر عوض می‌شوند.
- در قسمت کلاسیفای کردن متغیر هدف، می‌توان دو کلاس و سه کلاس را آزمود و دقت مدل‌ها را ارزیابی کرد.
- نقطه‌ی شکست متغیر هدف برای تبدیل به متغیر گسسته برای کلاسیفیکیشن، به جای ۲۵، ۵۰ و ۷۵، می‌تواند نقاط دیگری باشد، به ویژه اینکه داده‌ها، بالانس نیستند.
- برای بهبود مدل‌های پیش بینی، می‌توان دیتاست را به چند دسته تقسیم نمود و با انجام تبدیل‌هایی روی هر دسته، متغیر هدف یا محبوبیت را به توزیع نرمال نزدیک نمود تا دقت مدل‌های پیش بینی رگرسیونی بهبود یابد.
- اضافه کردن ژانر هر آهنگ به دیتاست، می‌تواند موجب بهبود مدل در کلاسیفیکیشن و پیش بینی شود.

منابع و مراجع

۱. Music Recommendation System using Spotify Dataset | Kaggle
۲. Spotify — Song Prediction and Recommendation System | by sunku sowmya Sree | The Startup | Medium
۳. Spotify Song Popularity Prediction | Kaggle
۴. Ways to Detect and Remove the Outliers | by Natasha Sharma | Towards Data Science
۵. Choosing the Right Metric for Evaluating Machine Learning Models — Part 2 | by Alvira Swalin | USF-Data Science | Medium
۶. Random Forest Parameter Tuning | Tuning Random Forest (analyticsvidhya.com)
۷. Middlebrook, K. (2019, Sep 18). Song Hit Prediction: Predicting Billboard Hits Using Spotify Data. ArXiv.Org. https://arxiv.org/abs/1908.08609v2
۸. Araujo, C., Cristo, M., & Giusti, R. (2019). Will I Remain Popular? A Study Case on Spotify. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, (pp. 599-610). Porto Alegre: SBC. Doi:10.5753/eniac.2019.9318
۹. Loggi, C. A Model for Predicting Music Popularity on Spotify. url: https://program.ismir2020.net/static/lbd/ISMIR2020-LBD-433-abstract.pdf
۱۰. Adeagbo, A. (2020, August 10). Predicting Afrobeats Hit Songs Using Spotify Data. ArXiv.Org. https://arxiv.org/abs/2007.03137
۱۱. Georgieva, E. Suta, M. Burton, N. HITPREDICT: PREDICTING HIT SONGS USING SPOTIFY DATA. STANFORD COMPUTER SCIENCE 229: MACHINE LEARNING. url: http://cs229.stanford.edu/proj2018/report/16.pdf