بسم الله الرحمن الرحیم

پروژه سوم درس مبانی علوم داده

دکتر بهرک و دکتر یعقوب زاده

محمد امین توانایی - ۸۱۰۱۰۱۳۹۶

سید علی تهامی - ۸۱۰۱۰۱۳۹۷

مهدی وجهی - ۸۱۰۱۰۱۵۵۸

فهرست

بخش ۳

در آغاز ماجرا بیایید و ببینیم هر کدام از این feature ها چه چیزی از یک آهنگ را بیان میکند

1. **explicit** : نشاندهنده داشتن محتوای غیر اخلاقی در یک آهنگ است
2. **danceability** : نشاندهنده میزان مناسب بودن یک آهنگ برای رقصیدن است.
3. **energy** : مفهوم انرژی در موسیقی معمولاً در رابطه با قدرت صدا، دامنه صدا و غیره استفاده می‌شود. مقادیر گام بالا فرکانس‌های بالایی دارند و بنابراین حالت‌های انرژی بالا را نشان می‌دهند.
4. **loudness** : مقدار بلندی آهنگ بر حسب دسی بل
5. **speechiness** : هر چه آهنگ شبیه گفتار باشد مقدار آن بیشتر و هر چه به موسیقی نزدیکتر مقدار آن کمتر خواهد بود.
6. **acousticness** : صرفاً از طریق هوای ارتعاشی و وسایل صوتی، به جای تکیه بر ابزارهای الکترونیکی یا مجازی. این نوع موسیقی بر سادگی در اشعار، هارمونی ها و ملودی های خود تأکید دارد و یک تجربه موسیقی اصیل و ارگانیک را ارائه می دهد. یک قهوه‌خانه دنج را تصور کنید که در آن خواننده‌ای یک گیتار آکوستیک را می‌نوازد، صدای او به طور طبیعی بدون هیچ گونه پیشرفت الکترونیکی طنین‌انداز می‌شود - این همان موسیقی آکوستیک است.در زمینه تحلیل موسیقی، آکوستیک بودن ارزشی است که میزان آکوستیک بودن یک آهنگ را توصیف می کند. امتیاز 1.0 نشان می دهد که آهنگ به احتمال زیاد آکوستیک است. این نشان دهنده میزان تکیه آهنگ بر آلات آکوستیک و تولید صدای طبیعی است. بنابراین، وقتی به آهنگی با آکوستیک بالا گوش می‌دهید، در ذات خام و فیلتر نشده موسیقی غوطه‌ور می‌شوید.
7. **Valence** : هر چه آهنگ دارای طرب بیشتر و شادی بیشتری باشد به ۱ نزدیکتر و هر چه دارای اضطراب بیشتر یا حالت افسردگی بیشتر باشد به صفر نزدیکتر است.
8. **تمپو** : سرعت ریتم آهنگ

بقیه ویژگی ها نسبت به کلمه توصیفشان مشخص هستند و نیاز به توضیح ندارند.

Transform

من برای تبدیل داده ها نیاز داشتم تا بعضی از ستون ها را از دیتای دسته بندی به دیتا عددی و بعضی ستون ها را از دیتای عددی به دیتای دسته بندی تبدیل کنم.

برای مثال explicit را که یک دیتای بولین بود به دیتای عددی و

دیتاهایی که یک عدد بین صفر و یک بودند را به ده بازه تبدیل کردم

و برای سال نیر آنها را به دهه های مختلف

و برای tempo نیز آنرا به دسته بندی معروفی که موسیقیدانها استفاده میکنند تقسیم کردم.

Aggregation

در این قسمت داده هایی را که آماده کردم برای تحلیل استفاده کردم

در این قسمت استفاده از توابع sum زیاد کارایی ندارد زیرا توزیع ها یکسان نبوده و تحلیل درست انجام نمیشود برای همین در بیشتر تحلیل ها از تابع count , avg استفاده شده است.

دقت شود هر تحلیلی که در زیر ارائه می شود بیان رابطه همبستگی است نه رابطه علت و معلولی

تحلیل بر اساس دهه تولید آهنگ:

decade	count	explicit_rate	danceability_rate	loudness_rate	acousticness_rate	valence_rate
1910	58	0.0	0.5808793103448276	-14.754137931034485	0.7028051724137931	0.6722448275862069
1920	52	0.0	0.5069423076923076	-16.892826923076928	0.8109284615384618	0.4808923076923073
1930	461	0.0	0.6124967462039039	-13.351028199566167	0.9500889370932765	0.627997613882863
1940	453	0.0	0.5720066225165568	-13.777507726269317	0.8476748335540841	0.5724106181015447
1950	653	0.0	0.41908728943338447	-14.964433384379793	0.8948390505359883	0.40191163859111795
1960	3159	0.0	0.4265659069325732	-15.60912060778727	0.8343791959480841	0.4246127572016468
1970	8784	0.0015938069216757742	0.4332575705828775	-14.831245901639285	0.7013520055783236	0.4562189219034621
1980	17183	4.6557644183204327E-4	0.48137746610021637	-13.500431356573372	0.5292524871663888	0.5128659547226905
1990	28595	0.011680363699947544	0.48064590662703166	-14.313751320160883	0.4729876649809412	0.4771593250568252
2000	153049	0.030526171356885703	0.4635383873792124	-15.110748459643775	0.5558407619547243	0.42681747440362866
2010	423753	0.03630652762340326	0.49172432879531597	-11.722861374432677	0.4660945789864887	0.4454481272345003
2020	498089	0.08180064205393012	0.49653124863226716	-10.87806376370498	0.40285504698876146	0.40972252153330374
2030	69726	0.3082207497920431	0.5589047629291765	-9.674015116312216	0.3120187908007013	0.40812349955540367

همانطور که دیده میشود با گذر زمان میزان بلندی آهنگ و داشتن محتوای غیر اخلاقی و استفاده کمتر از سازهای آکوستیک و استفاده از ساز های الکترونیک افزایش و میزان شادی و طرب آهنگ ها کاهش پیدا کرده است.

explicit_stat	count	danceability_rate	energy_rate	loudness_rate	speechiness_rate	acousticness_rate	instrumentalness_rate
0	1121367	0.4830483240099053	0.4965064366940072	-12.117940979179641	0.07377358206547045	0.46580304277687556	0.2993331429687508
1	82648	0.6288146561320302	0.6863260764398469	-7.612601139773397	0.22831835737102102	0.18829414126052443	0.05931239955727969

همانطور که دیده میشود آهنگ هایی که محتوای غیر اخلاقی دارند دارای توانایی رقص و انرژی بیشتر هستند و بلندی صدا در آنها بیشتر است همینطور در این محتواها موسیقی کمتری استفاده می شود و اکثرا به زبان گفتار نزدیکتر است و همینطور بیشتر از سازهای الکتریکی در آنها استفاده می شود.

liveness_label	count	energy_rate	loudness_rate	speechiness_rate	instrumentalness_rate
0-0.1	334846	0.47248503496860994	-12.753904209696293	0.07167325367482556	0.32054288386636326
0.1-0.2	502041	0.4532486739274314	-12.547020014301529	0.07290533263219814	0.3111585501831036
0.2-0.3	133221	0.5846215666268859	-10.194569054428445	0.09011841226232988	0.2226870813823613
0.3-0.4	115152	0.64715324052471	-9.365629420244524	0.1035335469640104	0.21076962729670184
0.4-0.5	29951	0.6489240573603482	-9.662788454475702	0.13072165203165143	0.2083070309555608
0.5-0.6	18991	0.6257604850508103	-10.169823705965971	0.13807168658838384	0.2118642719451323
0.6-0.7	24302	0.5847604744136286	-11.040682207225673	0.14064916879269249	0.2254916109888074
0.7-0.8	17288	0.5954675768047191	-10.856019666820929	0.15672966797778765	0.18163583476978273
0.8-0.9	11026	0.6690745819698883	-9.943725013604187	0.17312614728822817	0.16626583553147095
0.9-1	17197	0.7049577881956105	-9.498000581496816	0.14506812234692154	0.13749985103041185

دقت شود که اکثر آهنگ ها زنده بودن کمی دارند و زنده بودن با میزان انرژی و میزان بلندی صدا و میزان گفتاری بودن رابطه همبستگی مستقیم و با میزان بی کلام بودن آهنگ رابطه عکس دارند.

valence_label	count	explicit_rate	danceability_rate	energy_rate	loudness_rate	acousticness_rate	instrumentalness_rate	tempo_rate	duration_rate
0-0.1	163934	0.024637964058706552	0.2815761580880098	0.2798768438377655	-18.470007478619337	0.6765733968206166	0.56181748011364	104.4551918393987	5.225780775698337
0.1-0.2	149806	0.04574583127511582	0.398996239135946	0.388160953907722	-14.120375418875065	0.5639718372563944	0.3809823172881542	113.75408188590403	4.565502217868445
0.2-0.3	137781	0.062171126643005926	0.453772450120114	0.459611275321716	-12.16742428201279	0.48340072075156343	0.29072206529768063	117.0864510926746	4.352063452870838
0.3-0.4	145720	0.07701070546253089	0.4928443158111438	0.5137174734902568	-11.05675316360162	0.4260103412957764	0.24577523417519248	119.00724139445661	4.1132534495378925
0.4-0.5	127382	0.09014617449875179	0.525849447331649	0.5594950308638598	-10.273475616649314	0.37896258890282286	0.21330554941011595	119.98224136848214	3.95575542855344
0.5-0.6	127398	0.10004081696729933	0.5551953115433561	0.5917272864236477	-9.765047661658734	0.3524658894302080 7	0.1932915702722933	120.90117871552252	3.853551739823233
0.6-0.7	114094	0.10010167055235157	0.5828019518993085	0.6171510440180921	-9.395797973600654	0.33535476117586666	0.17894189663803153	121.46912650972192	3.742439055953814
0.7-0.8	97899	0.09064443967762695	0.6079800590404404	0.6358944042268075	-9.15031890009102	0.3328156235333345	0.1686980632192363	122.52310298368961	3.632224508081418
0.8-0.9	80996	0.06697861622796188	0.6312733999209961	0.653007256585509	-8.99458236209204	0.3380822203096426	0.17042783009481757	123.72432513951269	3.4985143042042965
0.9-1	59005	0.03423438691636302	0.6527294110668562	0.6620206272959915	-9.117471553258309	0.3798463296857853	0.22011361717922043	125.9826494364876	3.2717760034460603

همانطور که دیده میشود تعداد کمتری آهنگ با طرب بیشتر داریم
همانطور که انتظار می رود توانایی رقص در آهنهای با طرب بیشتر بیشتر است.تمپو(سرعت) موزیک افزایش یافته و انرژی آن بیشتر میشود.اکثرا آهنگهای غمگین زمان بیشتری دارند و در آهنگهای شاد از سازهای الکتریکی بیشتر استفاده میشود.

هادوپ و اسپارک

زمانی که ما نیاز به مدیریت حجم بسیار زیادی اطلاعات داریم و کارمان در مقیاس بزرگ است ساختمان داده های معمولی جوابگوی ما نیستند. برای پاسخ به این نیاز هادوپ و اسپارک ساخته شد. این دو با توضیح پردازش ها آنها را محاسبه می کنند و به این صورت بار پردازشی را تقسیم میکنند همچنین محاسبات را تا حد امکان به تاخیر می اندازند تا لازم نباشد بارها روی حافظه چیزی بنویسند یا موارد مشابه را به صورت مجزا و بار پردازشی بیشتر حساب کنند. یکی از مزایای اسپارک سازگاری بالای آن است این برنامه برای بسیاری از زبان های برنامه نویسی در دسترس است و همچنین با داده های مختلفی مثل sql یا csv و... می تواند کار بکند و همچنین یادگیری آن نیز آسان است. به عنوان مثال ما می خواهم داده های خود را فیلتر کنیم یکبار فقط داده هایی که ستون دوم آنها از ۵۰ بیشتر است را انتخاب کنیم و سپس فقط آنهایی که در ستون ۳ مقدار ۰ دارند را نگه دارد اگر این کار را با این ابزار انجام دهیم برنامه منتظر می مانند و سپس در هنگام نمایش یا ذخیره سازی همه فیلتر ها را با هم اعمال می کند و با یک پیمایش این کار را انجام می دهد که باعث افزایش سرعت پردازش می شود همچنین چون در پایگاه داده های بزرگ که روی دیسک نوشته می شود انجام یکباره عملیات تاثیر به سزایی روی عملکرد برنامه می گذارد.

Parquet فایل

این فایل برای ذخیره ی پایگاه داده هایی برای هدوپ هستند و بر خلاف فایل های csv json خوانا نیستند. همانطور که بالاتر از ویژگی های هدوپ گفتیم انعطاف و توزیع پذیری است بنابراین این فایل ها قابلیت

تقسیم شدن در چند دیسک را دارند. این فرمت داده ها را به صورت ستونی ذخیره می کند و این یعنی داده های ستون ها کنار یکدیگر نگهداری می شوند. این موضوع به ما این امکان را می دهد که ساختار تو در تو تعریف کنیم و همچنین برای خواندن محتوای یک ستون در یک پایگاه داده به سرعت عمل می کند همین موضع می تواند خود را در سرعت بالا فیلتر کردن عناصر نشان دهد و این موضوع برای پرس و جو بسیار خوب است

سوال ۴

ممکن است در زمانی سیستم از کار بیفتد و دیتاهایی که روی آنها پردازش شده است از بین برود و بعد از دوباره روی کار آمدن سیستم پردازش دوباره اطلاعات زمان گیر باشد برای همین باید پس از تغییر روی اطلاعات لازم است هر زمان یک بار آنها را روی سیستم ذخیره کنیم تا وقت زیادی برای پردازش دوباره اطلاعات از دست نرود.

این کار در اسپارک به صورت زیر انجام میشود :

```
// new context
val ssc = new StreamingContext(...)

//....

//set checkpoint directory
ssc.checkpoint(checkpointDirectory)
// Start the context
ssc.start()
ssc.awaitTermination()
```

یک استریم ایجاد میشود و اگر checkpointDirectory وجود نداشته باشد ایجاد شده و اگر زمانی در اجرای برنامه با شکست مواجه شویم برای بازیابی اطلاعات آنها را از checkpointDirectory بازیابی میکنیم

پانداس در مقابل اسپارک

همانطور که گفته شد اسپارک برای داده های در حجم زیاد پاسخگو است ولی پانداس نهایتا داده هایی با حجم کمتر از ۱۰ گیگ را می تواند پردازش کند اگر سیستم های ما توزیع شده باشند یا داده های ما توزیع شده باشند طبیعی است که اسپارک عملکرد بهتری دارد.
همچنین برای کار با داده های بزرگ به علت مواردی که بالاتر گفتیم و محاسبات موازی و کش کردن داده ها اسپارک عملکرد بهتر و سریع تری دارد.
از نظر مصرف حافظه نیز اسپارک به دلیل پردازش تنبل و استفاده از دیسک حافظه رم کمتری اشغال می کند.
اگر بخش سهولت استفاده پانداس بهتر عمل می کند زیرا آسان تر و آشنا تر و همچنین برای کاوش و نمایش داده ها به خوبی عمل می کند.
در موضوع سرعت همانطور که گفتیم برای داده های بزرگ اسپارک سریع تر است اما برای داده هایی که حجم زیادی ندارد و در رم جا می شوند پانداس چون پیچیدگی های اسپارک را ندارد بهتر عمل می کند.
سازگاری پانداس با سایر کتابخانه ها بیشتر و راحت تر است و می شود از آن همراه numpy, matplotlib و ... استفاده کرد اما برای اسپارک این موضوع کمی پیچیده تر و سخت تر است.
افراد بیشتری از پانداس استفاده می کنند بنابراین جامعه گسترده تری دارد ایرادات راحت تر پیدا می شوند اسناد کتابخانه بهتر و روان تر هستند.
همچنین اسپارک برای کار با داده هایی که در حال بروزرسانی هستند مثل داده های روی یک سرور مناسب است و می توان از آن استفاده کرد.
به علت محاسبات تنبل در اسپارک اعمال اعمال پیچیده محاسباتی در اسپارک بهتر انجام می شود.
در آخر با توجه به نیاز و حجم داده ها و بستری که داده های روی آن هستند و با توجه به موارد بالا یکی از آنها را انتخاب کرد.

فیلتر کردن با سرعت بیشتر

برای ذخیره‌سازی داده‌ها به گونه‌ای که بتوان بر اساس ستون‌های خاصی مانند تاریخ، سریع‌تر از فیلتر کردن معمولی جستجو کرد، می‌توان از روش‌های زیر استفاده کرد:

- **پارتیشن‌بندی**: داده‌ها را بر اساس مقادیر ستون مورد نظر (مانند تاریخ) پارتیشن‌بندی کنیم تا هنگام جستجو، فقط پارتیشن‌های مرتبط بررسی شوند.
- **اندیس‌گذاری**: ایجاد اندیس برای ستون‌هایی که اغلب جستجو می‌شوند تا عملیات جستجو سریع‌تر انجام شود.
- **فرمت‌های فشرده**: استفاده از فرمت‌های فایل فشرده مانند Parquet که جستجو و دسترسی به داده‌ها را بهینه می‌کند.

این روش‌ها به بهبود عملکرد و کاهش زمان لازم برای فیلتر کردن و جستجو در مجموعه داده‌های بزرگ کمک می‌کنند.

منابع

- آشنایی با آپاچی اسپارک (Spark) و پایتون — راهنمای مقدماتی
- مقدمه‌ای بر Apache Spark - ویرگول
- هادوپ (Hadoop) چیست؟ – مفاهیم و تعاریف – فرادرس - مجله
- ارزیابی تنبل وارانه (Lazy Evaluation) در زبان های برنامه نویسی تابع گرا
- فرمت های فایل در هدوپ: Avro، Parquet و ORC - ویرگول
- PySpark vs Pandas: Performance, Memory Consumption and Use Cases - Code Conquest