به نام خدا

گزارش پروژه 1

اعضا گروه

امیر سرتیپی

Contents

[‌فایل‌های برنامه 3](#_Toc70981130)

[sql\_manager.py 3](#_Toc70981131)

[excel\_manager.py 3](#_Toc70981132)

[q2.py 3](#_Toc70981133)

[q1.py 4](#_Toc70981134)

[q3.py 4](#_Toc70981135)

[نتایج بدست آمده 4](#_Toc70981136)

# ‌فایل‌های برنامه

## sql\_manager.py

به منظور افزایش سرعت در مراحل بعدی تصمیم بر تبدیل فایل Excel به یک فایل SQL گرفته شد.

دراین فایل متدهای

* 1. excel\_to\_SQL

این متد وظیفه تبدیل فایل excel را به فایل SQL دارد.

که خوب دیتاهای outlier مانند مقدار فیلد Description که مقادیری همچون Manual ,Discount, Damaged و یا خالی دارند را به table های پایگاه داده اضافه نمی‌کند.

* 1. create\_database

وظیفه ساختن دیتابیس با ستون‌هایی که در فایل اکسل Project1-groceries هست را برعهده دارد.

## excel\_manager.py

برای مدیریت فایل‌های اکسل ساخته شده است.

1. delet\_excel

برای حذف یک‌ فایل اکسل به کار می‌رود.

1. get\_create\_excel

برای ساخت یک فایل اکسل استفاده می‌شود.

1. create\_sheet

برای ایجاد شیت‌های متفاوت در یک فایل اکسل استفاده می‌شود.

1. add\_row

برای ایجاد سطر در فایل اکسل موجود استفاده می‌شود.

1. add\_rows

برای افزودن چندین سطر به یک‌ فایل اکسل موجود استفاده می‌شود.

1. read\_rows

برای خواندن سطرهای یک فایل اکسل موجود استفاده می‌شود.

## q2.py

پیاده سازی الگوریتم Apriori که از طریق بدست آوردن مجموعه کاندید و سپس بدست آوردن مجموع بزرگ از طریق مقدار کمینه باعث پیدا کردن آیتم‌های پرتکرار می‌شود.این کلاس برای برآورده کردن خواسته سوال 2 نوشته شده است

1. Apriori:تابع اصلی که در ابتدا L[1] تولید میشود و C[k] ها را به بررسی می‌کند و L[k] را تولید می‌کند.
2. apriori\_gen : مسئولیت تولید L[k] را بر عهده دارد.
3. has\_infrequent\_subset: مسئولیت بررسی پر تکرار بودن زیر مجموعه‌ها را بر عهده دارد.

خروجی این فایل یک فایل اکسل است که هر شیت آن شامل اسم با min\_sup مربوطه و در آن زمان و large item ها ذخیره شده است.

## q1.py

1. find\_transactions

این متد خواسته سوال 1 را برآورده می‌کند و در انتها خروجی را در یک فایل اکسل در پوشه Out به اسم می‌نویسد.

1. find\_items\_count

این متد خواسته سوال 2 را برآورده می‌کند و از اقلام داده­های موجود در مجموعه داده را به همراه تعداد تکرار آن به دست می‌آورد و در قالیب عکس‌هایی به صورت نمودار که در فایل Out قرار دارد آن‌را نمایش می‌دهد.

خروجی این فایل برنامه موارد زیر است:

* q1\_item\_frequency\_10: k پرفروش ترین اقلام بر روی نمودار (sort شده)
* ferequency.xlsx : تعداد فروش اقلام در فرمت اکسل (sort شده)

## q3.py

برای استخراج قوانین به کار می‎رود.

1. make\_subset

برای ساختن زیر مجموعه چپ و راست قوانین به کار می‌رود و در possiable\_rulesلیست‌هایی را اضافه می‌کند که شامل دو عضو است که با این کار سمت راست و چپ قوانین ساخته می‌شود.

1. Write\_strongs

این تابع برای ساختن خروجی اکسل از قوانین استفاده می‌شود که شامل 5 ستون است.

* ستون 1: سمت چپ قانون‌ها
* ستون 2: سمت راست قانون‌ها
* ستون 3: مقدار lift قوانین‌
* ستون 4: مقدار confidence قانون
* ستون 5: خود قانون

1. که همه‌ی موارد ذکر شده از تابع generate\_strongsبه‌دست آورده شده است.
2. read\_frequent

برای بدست اوردن frequent item set ها استفاده می‌شود. به این صورت که از فایل اکسلی که در frequent ها با min\_sup مشخص و جود دارند، سطرهای آن را استخراج کرده و در صورتی که ستونی از آن None بود آن را اضافه نمی‌کند.

1. make\_rule

در این تابع به تولید قوانین و محاسبه مقدار Confidence و Lift می‌پردازیم

1. prepare\_data

براساس make\_subset خروجی این تابع اطلاعات مورد نیاز برای محاسبه قوانین را به دست می‌آوریم.

## نتایج بدست آمده

برای بدست آوردن قواعد بهتر و قوی تر لازم است تا مقدار confidence را بالا ببریم و هرچه این مقدار بیشتر شود قوانین کمتری برای ما تولید خواهد شد. قوانین با confidence های مختلف در فایل excel قابل مشاهده می‌باشد.

با توجه به رخداد فروش متوجه می‌شویم که لبنیات، سبزیجات از اقلام اصلی سبد خرید بوده است چون به مقدار بسیار زیادی از آن‌ها خریداری شده است. قواعدی بدست آمد که برای مثال کسی که همبرگر می‌خرد، ژامبون نیز خریداری می‌کند.

.