به نام خداوند جان و خرد

پروژه شماره ۱ (فصل ۵) - قوانین همبستگی

نام درس: داده کاوی

استاد درس: دکتر رضا رمضانی

حل تمرین‌ها: نوراله کریم‌تبار (saeidkarimtabar@)

مهلت تحویل: ۲۶ آذر ۱۴۰۰

سامانه تحویل: lms.ui.ac.ir

توجه:

- لطفا تمرین‌ها را قبل از ددلاین‌ها تحویل دهید. تاخیر ارسال تمرین بعد از پایان یافتن مهلت تحویل، به ازای هر روز تاخیر ۱۰ درصد کاهش نمره از نمره‌ی تمرین را دارد.
- در صورت دیده شدن تقلب و کپی، نمره‌ی هر دو طرف صفر در نظر گرفته خواهد شد.

پیاده‌سازی الگوریتم Apriori: از الگوریتم Apriori به منظور کشف الگوهای مکرر از یک دیتاست تراکنشی استفاده می‌شود. در این پروژه، دیتاست مربوط به فروشگاه آنلاین (به نام groceries)در اختیار شما قرار گرفته است که شامل ۹۵۰۰ تراکنش (رکورد) از خریدهای مشتریان می‌باشد. هدف از این پروژه پیاده‌سازی الگوریتم Apriori بر روی این دیتاست است.

۱) در گام اول این پروژه، شما باید یک تحلیل داده اکتشافی (EDA[1]) از دیتاست داده شده نمایش دهید. EDA یعنی فهمیدن و درک مجموعه داده‌ها به وسیله جمع‌بندی ویژگی‌های اصلی، که غالباً آنها را به صورت بصری ترسیم می‌کند. برای اینکار طبق تحلیل خود می‌توانید نمودارهایی (شامل Histogram، Box Plot، Scatter plot و موارد دیگر) که نشان دهنده اطلاعاتی درباره دیتاست است را رسم کنید. همچنین می‌توانید مواردی چون تعیین بیشترین و کمترین تعداد آیتم در تراکنش‌ها، میانگین و میانه تعداد آیتم در تراکنش‌ها، تعیین آیتمی که بیشترین و کمترین تعداد فروش را داشته، بدست آوردن مقدار میانگین و میانه تعداد فروش مربوط به آیتم و هر خروجی که به درک بهتر دیتاست کمک می‌کند را در خروجی نمایش دهید.

[1] Exploratory Data Analysis

۲) در دومین گام این پروژه شما باید الگوریتم Apriori کلاسیک (نسخه پایه) را پیاده‌سازی کنید. دقت کنید که الگوریتم Apriori از مرتبه حافظه بالایی برخوردار است به همین دلیل بهتر است برنامه را در محیط Google Colab بنویسید و اجرا کنید. برای آشنایی با محیط Google Colab می‌توانید از لینک زیر استفاده کنید:

آموزش جامع استفاده از محیط Google Colab

https://coderlife.ir/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%AC%D8%A7%D9%85%D8%B9-%D8%A7%D8%B3%D8%AA%D9%81%D8%A7%D8%AF%D9%87-%D8%A7%D8%B2-%D9%85%D8%AD%DB%8C%D8%B7-google-colab-wzmozmwnsvip

برای نوشتن الگوریتم، از الگوی زیر استفاده کنید.

```
class Arules:
  def __init__(self):
    ...

  def get_frequent_item_sets(self,
                transactions,
                min_support,
                min_confidence):
     ...

  def get_arules(self,min_support=None,
          min_confidence=None,
          min_lift=None,
          sort_by='lift'):
    # sort_by: lift , confidence, support
    ...
```

متد get_frequent_item_sets باید بتواند دیتاست داده شده را گرفته و با اجرای الگوریتم Apriori پایه، لیستی از مجموعه‌های پرتکرار را پیدا کند. متد get_arules نیز نتایج بدست آمده در متد قبلی را به قانون تبدیل کرده و بر اساس support , confidence و lift این قوانین را مرتب‌سازی می‌کند. خروجی متد get_frequent_item_sets و get_arules جواب گام ۳ و ۴ شما خواهد بود.

۳) در گام سوم شما باید با اجرای الگوریتم Apriori پیاده‌سازی شده روی این دیتاست، با min_support = 0.005، ۱۰ مجموعه‌ی پرتکرار را نمایش دهید.

۴) از مجموعه الگوهای بدست آمده در مرحله سوم، قوانین مورد نیاز را با min_confidence= 0.2 تولید کنید و معیار lift را برای آنها محاسبه کرده و به صورت نزولی بر اساس این معیار مرتب کنید.

۵) مرحله سوم و چهارم برای ارزیابی درستی پیاده‌سازی شما در نظر گرفته شده بود. اما در گام پنجم شما به عنوان یک تحلیلگر با تغییر مقدار min support و min confidence و محاسبه دوباره مقادیر support , confidence , lift در قوانین بدست آمده هرگونه تحلیل جدید و ابتکاری را بر روی این تراکنش‌ها انجام دهید.

۶- (اختیاری دارای نمره اضافی)- یک الگوریتم Apriori بهبود یافته در یکی از مقالات مربوط به یک مجله/کنفرانس معتبر را پیاده سازی کنید و مراحل ۳ ، ۴ و ۵ را روی این الگوریتم نیز اعمال کنید. نام مقاله و جزئیات آن را تشریح کنید.

نکات مهم

- ترجیحا برنامه خود را با زبان پایتون بنویسید
- سعی کنید نتایج خود را تحلیل دقیق کنید. با تغییر مقادیر support و conf ممکن است به نتایج متفاوتی برسید.
- لطفا در گزارش نهایی خود نتایج تحلیل خود را بنویسید. توضیح کد و پیاده‌سازی به صورت شفاهی خواهد بود.