Задача 6

Дано

Вы — аналитик данных в розничной сети. Вам предоставили файл sales_data.csv

с историей продаж за 6 месяцев

Цель: выявить категории товаров с нестабильной динамикой и предложить меры по стабилизации продаж.

Файл sales_data.csv, содержащий помесячные данные о продажах. Формат таблицы:

- product_id — идентификационный номер продукта (целое число)
- month — месяц в формате YYYY-MM (строка или дата)
- sales — объём продаж (число, может быть ноль)
- category - категории продуктов (строка)

Требуется

1.Загрузить данные из файла sales_data.csv - файл должен находиться в том же проекте, что и программа.

- Используйте pd.to_datetime(), sort_values()
- Решение можно оформить так: загрузка → преобразование месяца → сортировка.

Решение

```
df = pd.read_csv('sales_data.csv')

df['month'] = pd.to_datetime(df['month'])

df = df.sort_values(['product_id', 'month'])
```

Требуется

2.Для каждой товарной позиции (product_id) определить, изменились ли продажи по сравнению с предыдущим месяцем: Используйте groupby() и diff() для вычисления разницы.

○ Создайте новый столбец sales_changed, который принимает значение:

i. True, если продажи изменились

ii. False, если продажи не изменились

○ Для расчёта используйте diff().ne(0).

Решение

```
# Исходные данные:
# product_id  month       sales
# 1           2024-01     100
# 1           2024-02     100
# 1           2024-03     120
# 2           2024-01      50
# 2           2024-02      60

# После группировки получаем две отдельные группы:
# Группа product_id=1: [100, 100, 120]
# Группа product_id=2: [50, 60]
```

df['sales_change'] = df.groupby('product_id')['sales'].diff().ne(0)

df.groupby('product_id')['sales'] -Группирует данные по идентификаторам товаров и выбирает столбец с продажами.

.diff() -Вычисляет разницу между текущей и предыдущей строкой внутри каждой группы. Для первой строки в группе: NaN (нет предыдущего значения) Для последующих строк: текущее значение - предыдущее значение.

.ne(0) -Проверяет, не равно ли значение нулю (not equal to 0).

Если разница = 0 → продажи не изменились → False

Если разница ≠ 0 → продажи изменились → True

Если разница = NaN (первая строка группы) → True (т.к. NaN ≠ 0)

Требуется

3.Итоговая таблица должна содержать следующие столбцы (в указанном порядке):

- product_id
- month
- sales
- category
- sales_change
- Приведите дату обратно к формату 'YYYY-MM' через .dt.strftime()
- Проверьте, что порядок и формат соответствуют.

Решение

df['month'] = df['month'].dt.strftime('%Y-%m')

Преобразует столбец с датами из формата datetime обратно в строковый формат 'ГГГГ-ММ'

1**.df['month']** - обращение к столбцу 'month' в DataFrame

Предполагается, что ранее этот столбец был преобразован в datetime с помощью pd.to_datetime()

2. .**dt - аксессор для работы с datetime-свойствами**

Позволяет использовать специальные методы для работы с датами

Доступен только для столбцов типа datetime

3. **.strftime('%Y-%m')** - метод форматирования дат

strftime() = "string format time"

%Y - год с веком (4 цифры, например: 2024)

%m - месяц (2 цифры, например: 01, 02, ..., 12)

- -разделитель между годом и месяцем

Решение

result_df = df[['product_id', 'month', 'sales', 'category', 'sales_change']]

Создает новый DataFrame, содержащий только указанные столбцы в заданном порядке

Требуется

4. Вывести первые 10 строк итоговой таблицы.

Решение

```
print("Первые 10 строк итоговой таблицы:")

print(result_df.head(10))
```

Требуется

5. Для каждого сочетания category и month рассчитать медианные sales

Решение

median_sales = df.groupby(['category', 'month'])['sales'].median().reset_index()

['sales'] - Указывает, что мы хотим применять статистические функции только к столбцу sales внутри каждой группы

.median() - Вычисляет медиану продаж для каждой группы.

Что такое медиана: Значение, которое делит упорядоченный на

Более устойчива к выбросам, чем среднее арифметическое.

.reset_index() - Преобразует результат группировки обратно

в обычный DataFrame.

```
# Исходные данные:
# category   month    sales
# Электроника 2024-01  100
# Электроника 2024-01  150
# Электроника 2024-01  200
# Электроника 2024-02  120
# Одежда      2024-01   80
# Одежда      2024-01   90
# Одежда      2024-02  100

# После groupby создаются группы:
# Группа 1: Электроника + 2024-01 → [100, 150, 200]
# Группа 2: Электроника + 2024-02 → [120]
# Группа 3: Одежда + 2024-01 → [80, 90]
# Группа 4: Одежда + 2024-02 → [100]
```

Требуется

6. Построить линейный график, где:

- Ось X — month
- Ось Y — медианные sales
- Каждая линия — отдельная категория

Решение

categories = median_sales['category'].unique() -Извлекает уникальные значения категорий из DataFrame.

for category in categories:

category_data = median_sales[median_sales['category'] == category]

plt.plot(category_data['month'], category_data['sales'], marker='o', label=category)

Запускает цикл по всем уникальным категориям.

Фильтрует DataFrame, оставляя только данные для текущей категории.

median_sales['category'] == category - создает булев массив (True/False)

median_sales[булев_массив] - выбирает только строки, где условие True

Строим линейный график для текущей категории.

Параметры:

- category_data['month'] - значения по оси X (месяцы)
- category_data['sales'] - значения по оси Y (медианные продажи)
- marker='o' - добавляет кружочки в точках данных
- label=category - задает метку для легенды

Требуется

7. Используйте .median(), .reset_index(), matplotlib.pyplot (plot(), xticks(), legend() и т. д.)

```
plt.xlabel('Месяц')

plt.ylabel('Медианные продажи')

plt.title('Динамика медианных продаж по категориям')

plt.xticks(rotation=45)

plt.legend()

plt.grid(True, linestyle='--', alpha=0.7)

plt.tight_layout()
```

Требуется

8. Сохранить изображение как sales_lineplot.png

Запишите свое решение в файл: task_2.py.

```
plt.savefig('sales_lineplot.png')

plt.show()

print("\nГрафик сохранен как 'sales_lineplot.png'")
```

Задача 2

Вы — аналитик клиентского опыта в крупном интернет-магазине. Вам предоставили

файл reviews_data.csv с историей отзывов клиентов за последние 12 месяцев.

Цель: оценить изменение клиентской удовлетворенности по продуктам и выявить группы с ухудшением восприятия.

Файл: reviews_data.csv, формат таблицы:

product_id Идентификатор продукта (целое число)

review_date Дата отзыва (в формате YYYY-MM-DD)

rating Оценка (целое число от 1 до 5)

sentiment Категория тональности отзыва (positive/negative/neutral)

department Отдел, к которому относится товар (строка)

Требуется:

1. Загрузить данные из reviews_data.csv — файл должен быть в том же проекте, что и программа.

2. Преобразовать даты в формат месяца (YYYY-MM) и сгруппировать данные по:

○ product_id

○ review_month (новый столбец, извлечённый из review_date)

○ Используйте pd.to_datetime(), .dt.to_period('M') или .dt.strftime('%Y-%m')

3. Для каждой группы рассчитать среднюю оценку (avg_rating) за месяц.

○ Используйте groupby(), agg() с mean

Требуется

4. Сформировать итоговую таблицу со столбцами (в указанном порядке):

○ product_id

○ review_month

○ avg_rating

○ department

5. Отсортировать по review_month по возрастанию. Вывести первые 10 строк.

6. Построить гистограмму распределения клиентских оценок — rating

i. Ось X — rating

ii. Ось Y — количество отзывов, поставивших соответствующую оценку

○ Сохранить как rating_gist.png

○ Используйте matplotlib.pyplot (plot(), xticks(), legend(), hist() и т. д.)

○ Добавить подписи осей и заголовок графика.