ABCDEats Inc.

Project of Data Mining

Jan-Louis Schneider, 20240506
Marta Boavida, 20240519
Matilde Miguel, 20240549
Sofia Gomes, 20240848

Fall/Spring Semester 2024-2025

1 Introduction

In an increasingly competitive market, food delivery companies face the challenge of delivering unique and targeted experiences to diverse customer bases.

ABCDEats Inc., a fictional food delivery service, has tasked us with analyzing its customer data to uncover insights that can inform a data-driven marketing strategy. By focusing on data collected over three months from three different cities, this project aims to segment customers based on economic and behavioral perspectives.

This integrated segmentation strategy is expected to empower ABCDE to develop personalized marketing strategies that align with distinct customer segments and maximize engagement and profitability.

This project was divided into the following phases:

- Exploration of the dataset
- Relationship between features
- New features

Throughout the report there are graphs for easier and more intuitive viewing.

2 Exploration of the dataset

After importing the dataset "DM2425_ABCDEats_DATASET.csv" and the necessary libraries, we used several functions already defined in the libraries in order to better understand the data.

To see all the columns:

```
df.columns.values
```

To see the first ten lines:

```
df.head(10)
```

To get information about the index dtype and columns, non-null values and memory usage:

```
df.info()
```

2.1 Incorrect data

After an analysis, we noticed that some variables are incorrect (the variable type is float and should be int), namely:

- **customer_age**
- **first_order**
- **HR_0**

So we converted the types using the function:

```
df["customer_age"]= df["customer_age"].astype('Int64')
df["first_order"] = df["first_order"].astype("Int64")
df["HR_0"] = df["HR_0"].astype("Int64")
```

Another detail about incorrect data is between product_count and vendor_count. There is information about products that were not purchased but there is information about sellers who sold them.

2.2 Duplicates

We looked for duplicate values and found 0.041%, so we removed them using the function:

```
df.drop_duplicates(inplace=True)
```

2.3 Missing values

We check for missing values and found some variables:

- Percentage of missing values in **customer_region** is: 1.387%
- Percentage of missing values in **customer_age** is: 2.281%
- Percentage of missing values in **first_order** is: 0.333%
- Percentage of missing values in **last_promo** is: 52.53%
- Percentage of missing values in **HR_0** is: 3.652%

2.4 Numerical variables

Numerical variables represent measurable quantities and can be analyzed mathematically. The numerical variables that exist in this dataset are:

- **customer_age**: 75% of customers are young, but there are a few older individuals
- **vendor_count**: Most customers ordered from few unique vendors, but there are customers with much higher count
- **product_count**: Most customers don't buy many products
- **is_chain**: relative small amount of orders made in chain restaurants
- **first_order**: On average customers place their first order 28 days after joining the app
- **last_order**: customers are buying more recently than they did earlier
- **CUI** (Overral analysis): 75% of customers spend nothing or a significant small amount but the maximum values of these cuisines are significantly high (specially for American (280) and Asian (896)) cuisines which indicate the presence of potential outliers who frequently order or spend heavily
- **DOW_0 to DOW_6** (Overral analysis): most customers only ordered 1 time in each day of the week
- **HR_0 to HR_23** (Overral analysis): No activity at midnight (HR_0). 75% of customers placed no order in HR_1 to HR_23

2.5 Categorical variables

Categorical variables represent characteristics or qualities that group data into distinct categories or labels. In this dataset, the categorical variables are:

- **customer_region**: Customers are from 8 different regions. Most customers are located in region 8670
- **last_promo**: There are promotions in 3 different categories. Most customers use promotions in the delivery category
- **payment_method**: There are 3 different payment methods used by customers. Most customers use Card as their preferred payment method

After analysing the categorical variables, we realise that isn´t strange data.

3 Relationship between features

In order to explore the relationships between the features, we looked for trends and patterns.

3.1 Numerical variables

Write something

3.2 Categorical variables

Write something

4 New features

Create new features can significantly enhance our analysis by providing additional insights and improving the performance of models.

In order to that, we create some:

1. **Customer Lifetime**: Interval of customer activity, so we have an idea of how many days the customer ordered

```
df['lifetime_days'] = df['last_order'] - df['first_order']
```

2. **Most frequent order day of the week:** Indicates the days of the week on which the customer placed the most orders
   - 6h-12h: Morning (Breakfast)
   - 12h-18h: Afternoon (Lunch)
   - 18h-00h: Evening (Dinner)
   - 00h-6h: Night
3. **Total monetary units spend**: Sum all total expenses

```
cuisine = df.filter(like='CUI_').columns.tolist()
df['total_expenses'] = df[cuisine].sum(axis=1)
```

4. **Average monetary units per product:** Show the average monetary of all products

```
df['avg_per_product'] =df['total_expenses'] / df['product_count'].
    replace(0, pd.NA)
```

5. **Average monetary units per order:** Show the average monetary per order

```
df['avg_per_order'] = df['total_expenses'] / df[dows].sum(axis=1).
    replace(0, pd.NA)
```

6. **Average order size:** Help identifing users who make larger orders

```
df['avg_order_size'] = df['product_count'] / df[dows].sum(axis=1).
    replace(0, pd.NA)
```

7. **Culinary profile:** A proportion of ordered cuisines. A higher number indicates more diversity of types of cuisine you ordered.

```
total_cuisine = len(cuisine)
df['culinary_variety'] = (df[cuisine].gt(0).sum(axis=1) / total_cuisine)
    * 100
```

8. **Loyalty to chain restaurants:** Proportion of orders from restaurant chains. A high value indicates that you prefer to try different restaurant chains. A lower value is only more faithful to certain chains.

```
df['chain_preference'] = df['is_chain'] / df[dows].sum(axis=1).replace
    (0, pd.NA)
```

9. **Loyalty to venders:** Proportion of orders from specific restaurants. A high value indicates that you prefer to try different restaurants. A lower tend to be more loyal to specific restaurants.

```
df['loyalty_to_venders'] = df[dows].sum(axis=1) / df['vendor_count'].
    replace(0, pd.NA)
```

5 Appendix A

Appendixes are for materials, tables, or more explanation material only done by the student.