# MARKET SEGMENTATION ANALYSIS

A Mini Project

Submitted to

**JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD**

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By


| | |
|---|---|
| **POLAMPELLI ARCHANA** | **21UK1A0515** |
| **MOHAMMED NASREEN BANU** | **21UK1A0534** |
| **NAGINENI NITHIN RAO** | **21UK1A0528** |
| **BEDADA BHASKAR** | **22UK5A0501** |


Under the guidance of


**Mr. N. SRAVAN KUMAR**

Assistant Professor




**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VAAGDEVI ENGINEERING COLLEGE**

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S)

506005

**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

**VAAGDEVI ENGINEERING COLLEGE (WARANGAL)**

## CERTIFICATE OF COMPLETION

This is to certify that the UG Minor project entitled **"MARKET SEGMENTATION ANALYSIS**" is being submitted by **POLAMPELLI ARCHANA (21UK1A0515), MOHAMMED NASREEN BANU (21UK1A0534), NAGINENI NITHIN RAO (21UK1A0528), BEDADA BHASKAR (22UK5A0501),**in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

**Project Guide**
Mr.N.Sravan kumar
(Assistant Professor)

**HOD**
Dr. R.Naveen kumar
(Professor)

**EXTERNAL**

# ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHAMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Minor Project in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Minor Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Minor Project and for their support in completing the UG Minor Project.

We express heartfelt thanks to the guide, **N. SRAVAN KUMAR,** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this UG Minor Project.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

| | |
|---|---|
| **POLAMPELLI ARCHANA** | **21UK1A0515** |
| **MOHAMMED NASREEN BANU** | **21UK1A0534** |
| **NAGINENI NITHIN RAO** | **21UK1A0528** |
| **BEDADA BHASKAR** | **22UK5A0501** |

# ABSTRACT

Marketing segmentation is widely used for targeting a smaller group of market in consumer product industry, and is useful for decision makers to effectively focus on their consumers' purchasing behaviour based on one marketing mix. In order to have a better understanding of the relationship between marketing segmentation in conventional business field and data science field, marketing segmentation is introduced from the perspectives of business and data science in this paper. In addition, on the basis of the introduction of both methods, conclusion can be drawn that data science methods are used in marketing segmentation as an assistance to present more precise output, based on the theory generated in business practise

Market segmentation is becoming very familiar and essential to every marketer in the process of designing and implementing an effective target-marketing strategy. It is confirmed in the grocery retail industry about the importance of appropriate market segmentation. In this industry, customer purchasing behaviour needs to be acknowledged not only in specific products, but also the interaction among the whole range of products. Therefore, the motivation for this thesis is to discover a segmentation based on this purchasing behaviour among whole range of products, which is called purchasing pattern. The Purchasing pattern is interpreted by purchasing portfolios, which include list of categories that a certain customer purchases and also consumption behaviour on these categories.

As a result, the availability of segmentation is verified from a technical view and the practical significance of segmentation is confirmed from a marketing view. The result from data mining has shown four segments from the analysis of purchasing portfolios. These four segments cover most of the market, and remain over time. The segmentation is assessed from marketing view to be appropriate for practical application.

Furthermore, there are three segments that are selected to be analyzed further. They represent three distinct purchasing behaviours. Three specific purchasing portfolios are built for each segment, which can be used to direct for marketing strategy

## TABLE OF CONTENTS:-

# 1. INTRODUCTION

## 1.1. OVERVIEW

Market segmentation is a process that consists of sectioning the target market into smaller groups that share similar characteristics, such as age, income, personality traits, behaviour, interests, needs, or location.

Knowing your market segmentation will help you target your product, sales, and marketing methods. It can help your product development processes by guiding how you build product offers for various groups, such as males versus women or high-income versus low-income. These segments can be used to optimize products, marketing, advertising, and sales efforts.

Segmentation allows brands to create strategies for different types of consumers, depending on how they perceive the overall value of certain products and services. In this way, they can introduce a more personalized message with the certainty that it will be received successfully.

## 1.2. PURPOSE

The purpose of conducting market segmentation analysis using a McDonald's dataset revolves around several key objectives aimed at enhancing business strategy and customer satisfaction:

1. **Understanding Customer Diversity:** By analyzing demographic, behavioural, and psychographic data, McDonald's aims to gain insights into the diverse preferences and behaviours of its customer base.

2.**Targeted Marketing Strategies:** Segmentation helps McDonald's tailor its marketing efforts more effectively. By identifying distinct customer segments, the company can create targeted promotions, advertisements, and menu offerings that resonate with specific groups.

3.**Enhancing Customer Experience:** By understanding what different customer segments value and prefer, McDonald's can optimize its service offerings. This includes everything from menu choices to the overall dining experience, both in-store and through digital channels.

4. **Optimizing Operational Efficiency:** Segmentation allows McDonald's to allocate resources more efficiently. By focusing on segments with higher profitability or growth potential, the company can prioritize initiatives that are most likely to yield positive returns.

5. **Driving Competitive Advantage:** Effective segmentation enables McDonald's to differentiate itself in a highly competitive market. By catering to the unique needs and preferences of different customer segments, the company can build stronger customer loyalty and attract new customers.

6. **Adapting to Market Changes:** Segmentation analysis provides insights into evolving consumer trends and behaviours. This enables McDonald's to adapt its strategies and offerings in response to changes in the market landscape.

Overall, market segmentation analysis using a McDonald's dataset serves as a strategic tool to better understand and serve its diverse customer base, ultimately driving growth and profitability in the competitive fast-food industry.

# 2. LITERATURE SURVEY

## 2.1 EXISTING PROBLEM

Market segmentation, while a powerful tool for targeting specific customer groups, faces several challenges that can impact its effectiveness and implementation:

**Over-Simplification**: Segmentation models can sometimes oversimplify customer diversity, leading to generalized assumptions that may not accurately represent the complexity of customer behaviour and preferences.

**Static Segmentation**: Market segments are often treated as static groups, whereas consumer behaviour and preferences can evolve rapidly. This can lead to outdated targeting strategies that fail to adapt to changing market dynamics.

**Data Availability and Quality**: Effective segmentation relies heavily on accurate and comprehensive data. Issues such as data privacy concerns, incomplete datasets, and data silos can limit the ability to create meaningful segments.

**Segmentation Criteria**: Choosing the right criteria for segmentation is crucial but challenging. Deciding which variables (demographic, behavioural, psychographic, etc.) to use and how to weigh their importance requires careful consideration and can vary depending on the industry and market.

**Overlap and Cannibalization**: There's often a risk of overlap between segments or cannibalization of sales, where targeting one segment may inadvertently affect sales to another segment.

**Implementation Complexity**: Once segments are identified, implementing targeted marketing strategies can be complex and resource-intensive. This includes creating customized messaging, developing tailored products/services, and managing different distribution channels.

**Measurement and Evaluation**: Assessing the effectiveness of segmentation strategies can be challenging. Metrics for evaluating whether segments are profitable or achieving their intended goals may not always be straightforward or easy to quantify.

**Dynamic Nature of Markets**: Markets are dynamic and constantly changing due to factors such as technological advancements, economic shifts, and changes in consumer behaviour. This makes it difficult to maintain the relevance and effectiveness of segmentation strategies over time

## 2.2 PROPOSED SOLLUTION

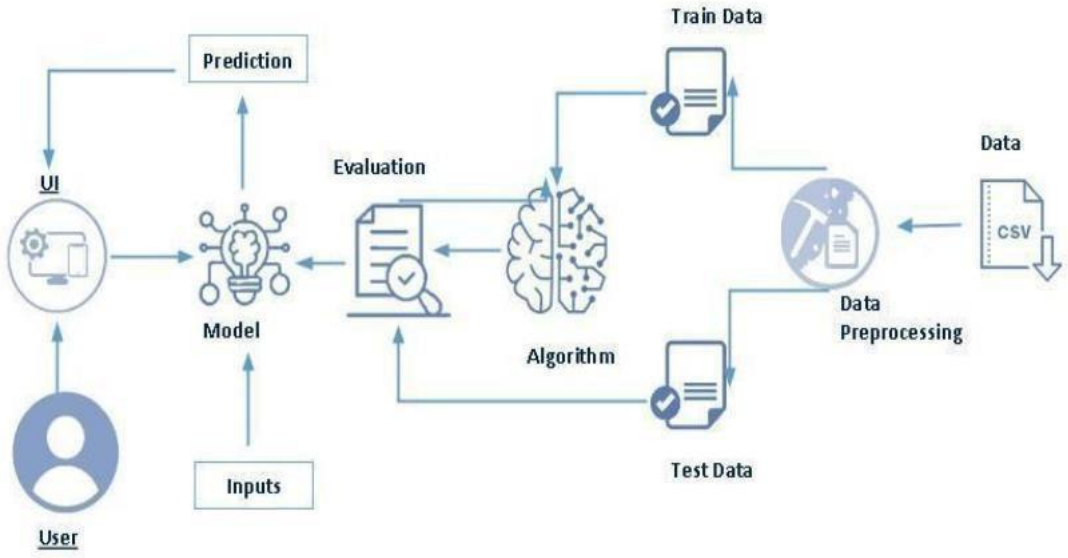
To address the challenges associated with market segmentation effectively, here are some proposed solutions:

1. **Advanced Data Analytics**: Invest in advanced data analytics capabilities to gather, analyze, and interpret large volumes of data. Utilize techniques such as machine learning and predictive analytics to uncover hidden patterns and segment customers based on more nuanced criteria.
2. **Dynamic Segmentation Models**: Move away from static segmentation models towards dynamic ones that can adapt in real-time to changes in consumer behaviour and market conditions. Implement algorithms that continuously update segments based on new data inputs.
3. **Integration of Data Sources**: Break down data silos by integrating data from multiple sources (e.g., CRM systems, social media platforms, transactional data). This integrated approach provides a more holistic view of customer behaviour and preferences.
4. **Personalized Marketing Strategies**: Develop personalized marketing strategies for each segment that resonate with their unique needs and preferences. Use personalized messaging, offers, and content to enhance engagement and conversion rates.
5. **Continuous Monitoring and Evaluation**: Establish metrics and KPIs to monitor the performance of segmentation strategies over time. Regularly evaluate the effectiveness of each segment in achieving business objectives and make adjustments as necessary.
6. **Segment Overlap Management**: Implement strategies to manage overlap between segments and mitigate cannibalization effects. This may involve refining segmentation criteria, adjusting marketing tactics, or developing cross-selling strategies.
7. **Customer Feedback and Engagement**: Incorporate customer feedback and engagement data into segmentation models to ensure they accurately reflect customer sentiments and preferences. Use surveys, reviews, and social media interactions to gather qualitative insights.
8. **Collaboration Across Departments**: Foster collaboration between marketing, sales, customer service, and product development teams to ensure alignment on segmentation strategies. This interdisciplinary approach can lead to more cohesive and effective implementation.

9. **Adoption of Agile Methodologies**: Adopt agile methodologies to iterate quickly on segmentation strategies and respond promptly to market changes. Test new segmentation hypotheses through pilot projects or A/B testing before full-scale implementation.
10. **Ethical Considerations**: Ensure compliance with data privacy regulations and ethical guidelines when collecting and utilizing customer data for segmentation purposes. Build trust with customers by being transparent about data usage and respecting their privacy preferences.

# 3. THEORITICAL ANALYSIS

## 3.1. BLOCK DIAGRAM




## 3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

### 1.Data Collection and Preparation

- **Data Sources**: Gather relevant data sources such as customer demographics, purchase history, website behaviour, etc.
- **Data Cleaning**: Use Python libraries like Pandas for cleaning and preprocessing the data to handle missing values, outliers, and inconsistencies.
- **Data Integration**: Merge and integrate data from various sources into a unified dataset.

### 2. Exploratory Data Analysis (EDA)

- **Descriptive Statistics**: Use Pandas and NumPy for basic statistics (mean, median, mode, etc.) to understand the data.

## 3. Market Segmentation Techniques

- **Clustering Algorithms**: Implement clustering algorithms (e.g., k-means clustering, hierarchical clustering) using libraries such as Scikit-learn or SciPy.
- **Dimensionality Reduction**: Apply techniques like Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimensionality of the data and visualize clusters in lower dimensions.

## 4. Model Training and Evaluation

- **Train Segmentation Models**: Fit clustering models on the prepared dataset to identify segments.
- **Evaluation**: Evaluate cluster quality using metrics such as silhouette score or inertia to determine the optimal number of clusters and assess clustering performance.

## 5. Segmentation Visualization

- **Visualize Clusters**: Use Matplotlib, Seaborn, or Plotly to visualize clusters and segment characteristics.

## 6. Integration with Jupyter Notebook

- **Interactive Analysis**: Leverage Jupyter Notebook's interactivity to iterate on segmentation models, visualize results, and explore segment characteristics dynamically.
- **Document Insights**: Use Markdown cells in Jupyter Notebook to document findings, insights, and recommendations based on segmentation analysis.

## 7. Deployment and Integration

- **Export Models**: Export trained models using joblib or pickle for deployment in production environments.
- **Integration**: Integrate segmentation results and insights with other business applications or platforms using APIs or direct data exports.

## 8. Iterative Development and Collaboration

- **Version Control**: Use Git for version control to track changes and collaborate on the development of segmentation software.

- **Feedback Loop**: Gather feedback from stakeholders and end-users to iteratively improve the software based on their needs and use cases.

# 4. EXPERIMENTAL INVESTIGATION

An experimental investigation into market segmentation can provide insights into its effectiveness and best practices.

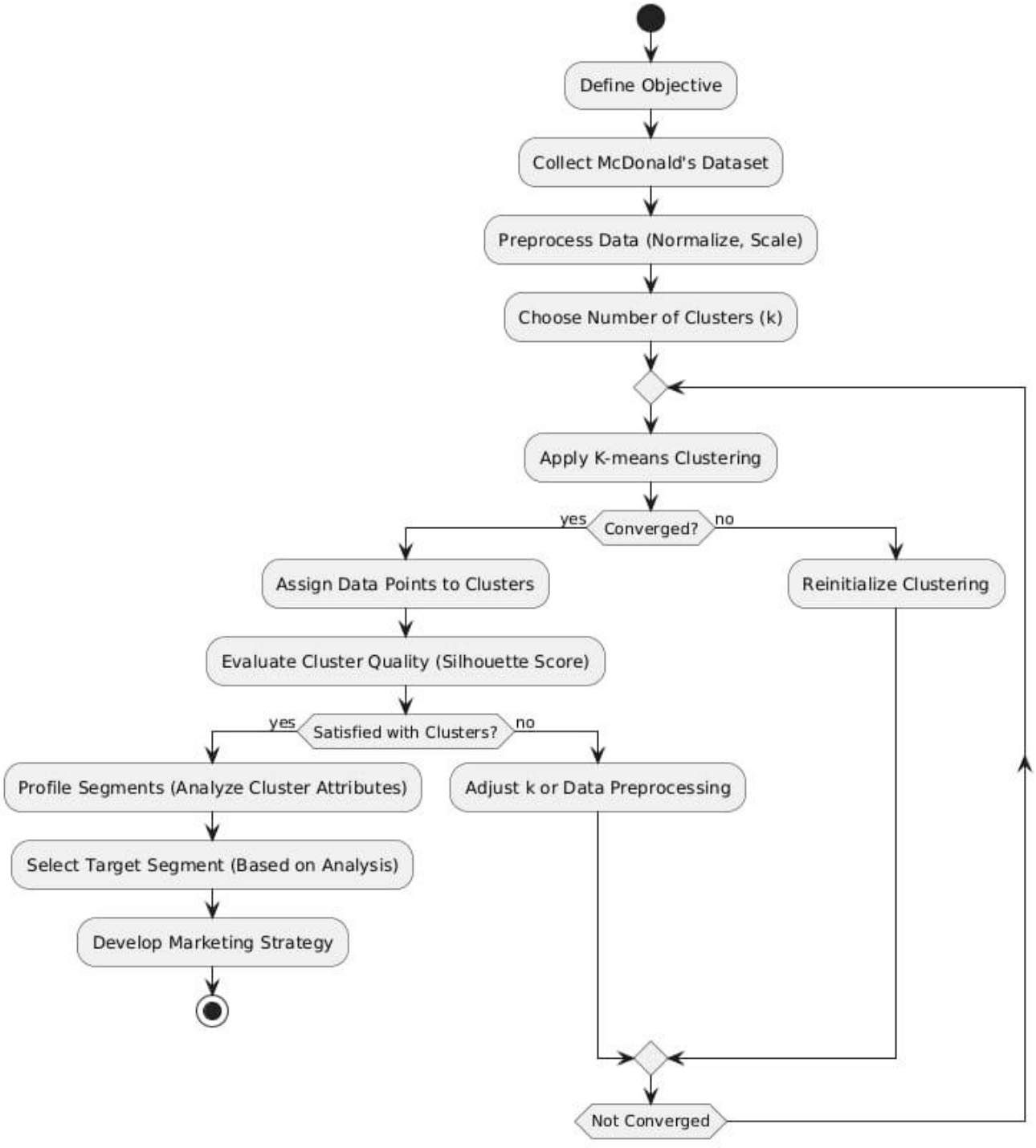
## Steps for Conducting an Experimental Investigation on Market Segmentation:

1. **Define Objectives:**
   - Determine the goals of the segmentation study.
   - Identify key questions to be answered, such as the effectiveness of segmentation strategies, customer responses, and profitability impacts.
2. **Select a Market:**
   - Choose a specific market or industry to investigate.
   - Ensure the market has diverse customer segments to study.
3. **Identify Segmentation Variables:**
   - Demographic: age, gender, income, education.
   - Geographic: location, climate, urban vs. rural.
   - Psychographic: lifestyle, values, personality.
   - Behavioural: purchasing behaviour, brand loyalty, usage rate.
4. **Collect Data:**
   - Use surveys, interviews, and existing market data.
   - Ensure data is comprehensive and representative of the target market.
5. **Analyze Data:**
   - Use statistical methods and clustering techniques to identify distinct segments.
   - Tools like K-means clustering, hierarchical clustering, and factor analysis can be useful.
6. **Develop Profiles:**
   - Create detailed profiles for each segment.
   - Include demographics, preferences, purchasing behaviour, and other relevant characteristics.
7. **Design Experimental Marketing Strategies:**
   - Develop tailored marketing strategies for each segment.
   - Create variations of marketing messages, promotions, and product offerings.
8. **Implement and Test:**
   - Launch marketing campaigns targeting specific segments.
   - Use control groups to compare the effectiveness of segmented vs. non-segmented strategies.
9. **Measure Results:**
   - Track key performance indicators (KPIs) such as sales, conversion rates, customer satisfaction, and ROI.

- Use A/B testing to determine the impact of different strategies.

10. **Analyze and Interpret Findings:**
    - Compare the performance of different segments.
    - Identify which segmentation variables were most effective.
    - Assess the overall impact of segmentation on business objectives.
11. **Refine and Optimize:**
    - Adjust segmentation strategies based on findings.
    - Continuously monitor and refine segments as market conditions and consumer behaviours change.

# 5. FLOWCHART

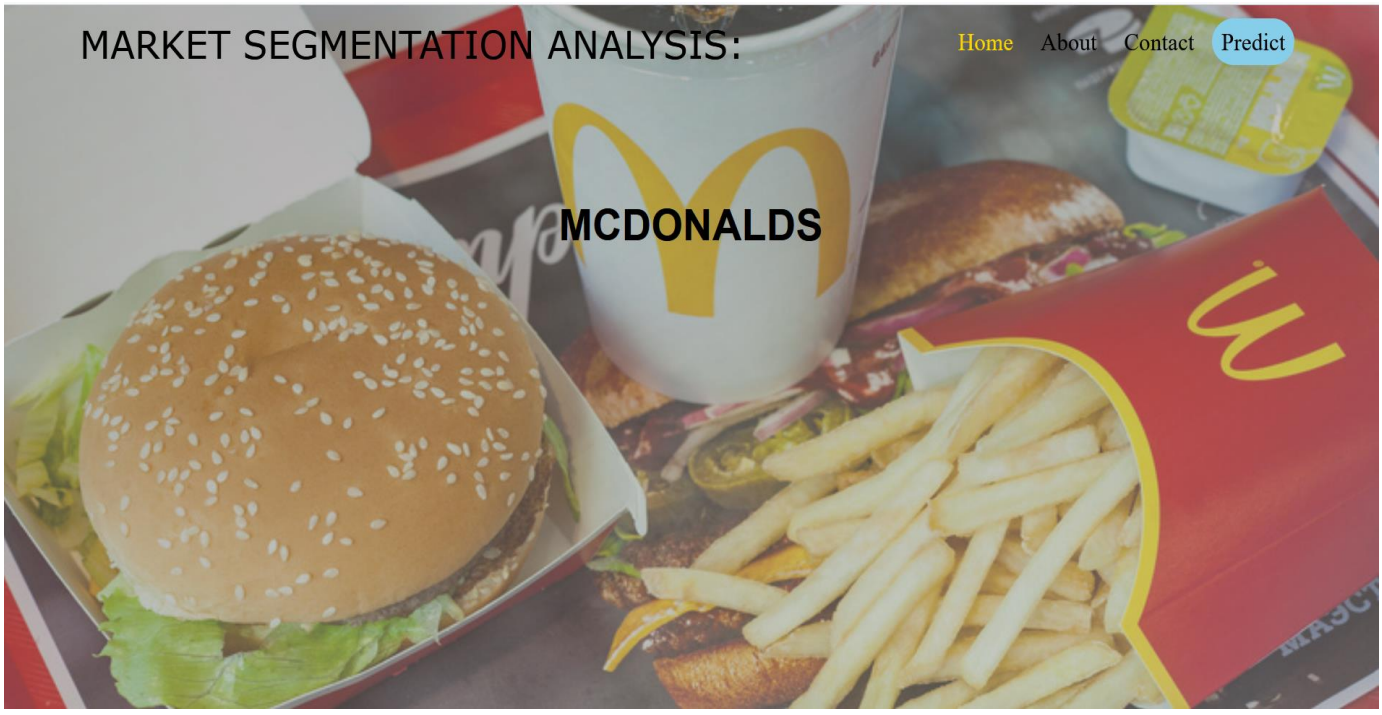

Define Objective
Collect McDonald's Dataset
Preprocess Data (Normalize, Scale)
Choose Number of Clusters (k)
Apply K-means Clustering
Converged?
yes
no
Assign Data Points to Clusters
Reinitialize Clustering
Evaluate Cluster Quality (Silhouette Score)
Satisfied with Clusters?
yes
no
Profile Segments (Analyze Cluster Attributes)
Adjust k or Data Preprocessing
Select Target Segment (Based on Analysis)
Develop Marketing Strategy
Not Converged

# 6. RESULT

## *DEVELOPMENTSERVER*

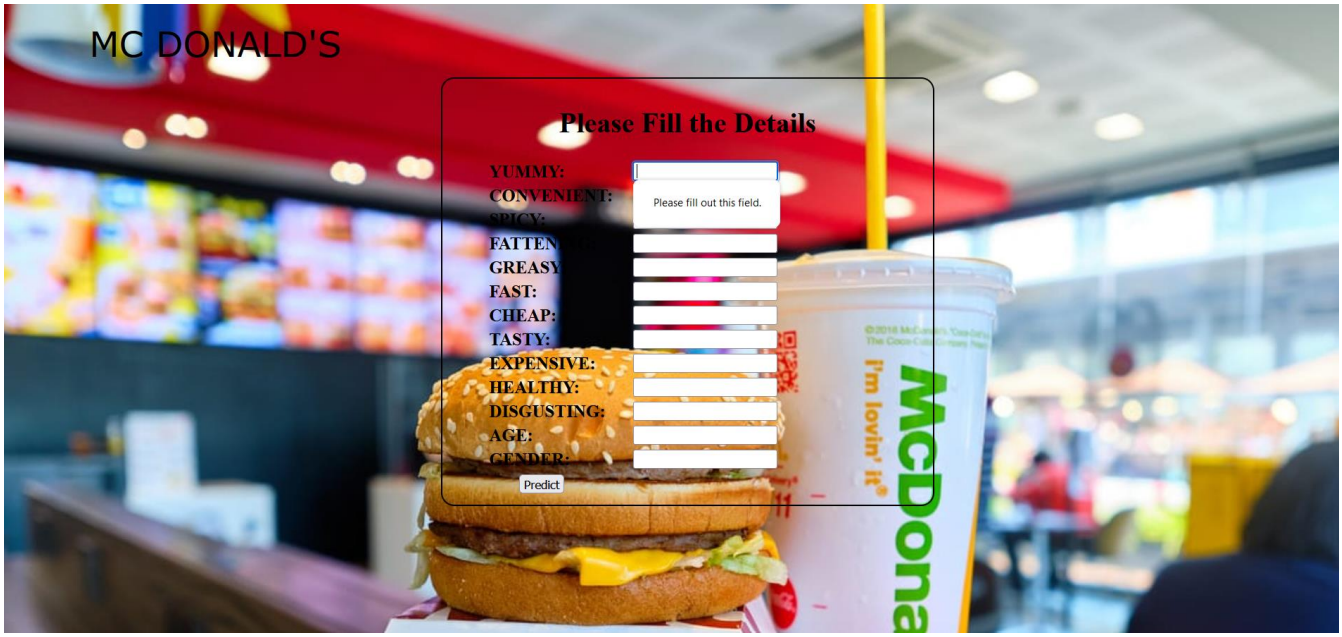
```
In [1]: runfile('C:/Users/nagin/OneDrive/Desktop/market/Flask/app.py', wdir='C:/Users/nagin/OneDrive/Desktop/market/Flask')
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
```

✧ ***HOMEPAGE***



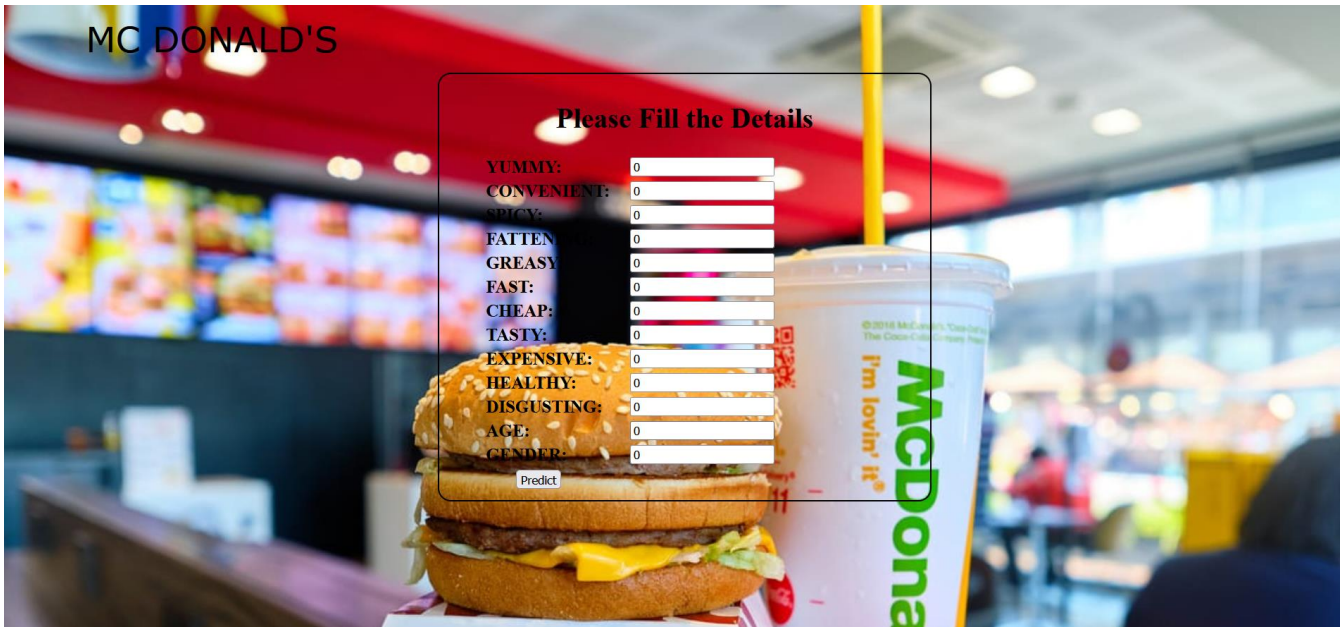
## ***ABOUT PAGE:***


## ***PREDICTIONPAGE***

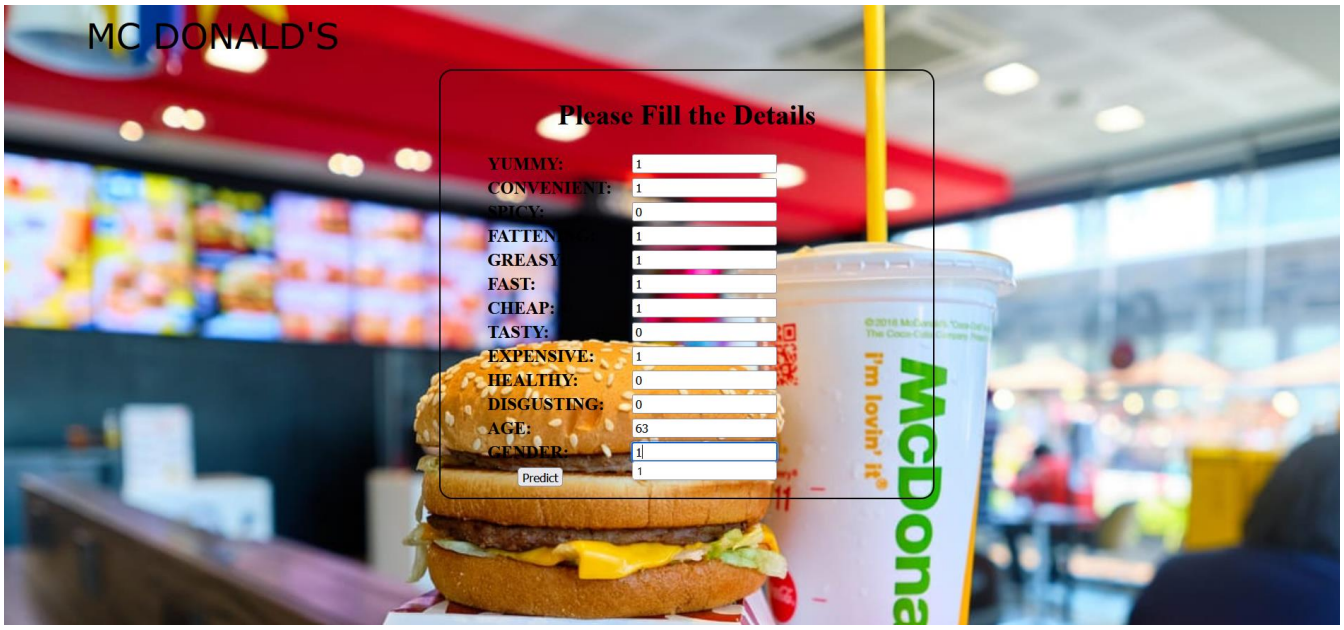


# PREDICTIONS:

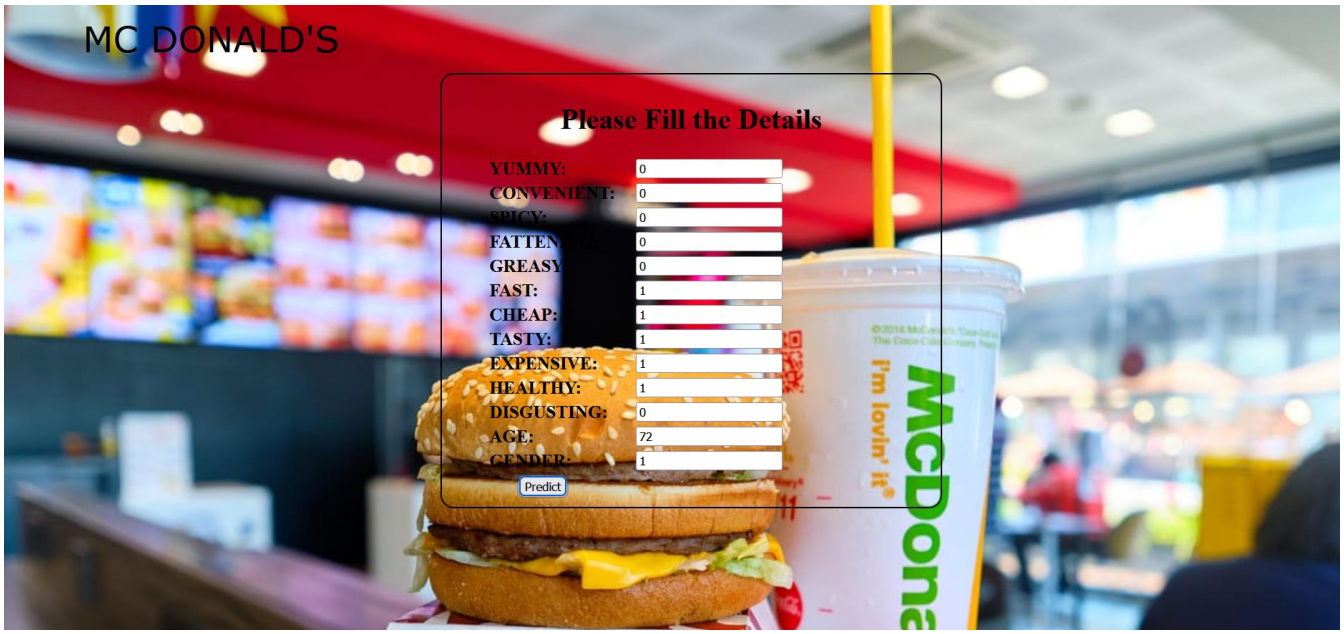
## ❖ PREDICTION-1




## ❖ PREDICTION-2




## ❖ PREDICTION-3




# 7. ADVANTAGES AND DISADVANTAGES

## ADVANTAGES:

- **Improved Customer Understanding:**

  - **Targeted Marketing:** Allows businesses to tailor marketing strategies to specific groups, increasing relevance and effectiveness.
  - **Personalized Experience:** Enhances customer experience by addressing specific needs and preferences.

- **Efficient Resource Allocation:**

  - **Optimized Spending:** Directs marketing efforts and budgets to the most promising segments, improving ROI.
  - **Focus:** Enables businesses to concentrate on high-potential areas, reducing waste and inefficiency.

- **Competitive Advantage:**

  - **Differentiation:** Helps differentiate products and services by appealing to unique customer needs.
  - **Brand Loyalty:** Builds stronger connections with customers, fostering loyalty and repeat business.

- **Product Development:**

  - **Customer Feedback:** Provides insights into customer preferences, guiding product improvements and innovation.
  - **Niche Markets:** Identifies underserved segments, creating opportunities for new product lines.

- **Increased Market Share:**

  - **Segment-specific Strategies:** Tailored approaches can capture a larger share of each targeted segment.
  - **Better Positioning:** Enhances market positioning by aligning products with segment needs.

.

## ✧ DISADVANTAGES:

### Increased Costs:

- **Research and Analysis:** Conducting segmentation studies requires significant investment in time and resources.
- **Customized Marketing:** Developing different marketing strategies for each segment can be costly.

### Complexity in Management:

- **Coordination Challenges:** Managing multiple marketing campaigns and strategies can be complex and resource-intensive.
- **Inconsistent Messaging:** Risk of inconsistent brand messaging across segments if not carefully managed.

### Risk of Over-segmentation:

- **Diminished Returns:** Excessive segmentation can lead to small, unprofitable segments.
- **Resource Dilution:** Spreading resources too thinly across many segments can reduce overall effectiveness.

### Market Changes:

- **Dynamic Preferences:** Consumer preferences andbehaviours can change rapidly, making segments less relevant over time.
- **Data Obsolescence:** Segmentation data can become outdated, requiring continuous updates and adjustments.

### Potential Exclusion:

- **Missed Opportunities:** Focusing too narrowly on certain segments may lead to neglecting other profitable customer groups.
- **Bias and Assumptions:** Incorrect assumptions about segments can lead to ineffective strategies and missed market potential.

# 8. APPLICATIONS

Market segmentation analysis finds diverse applications across various industries and business functions. Here are some key applications:

1. **Marketing Strategy Development**: Segmentation analysis helps businesses tailor their marketing strategies to specific customer segments. It allows them to identify the most effective channels, messages, and promotional tactics for each segment, maximizing the impact of marketing efforts.

2. **Product Development**: Understanding different market segments enables businesses to develop products that meet specific needs and preferences. This can involve creating variations of existing products or entirely new products tailored to different segments' requirements.

3. **Customer Retention and Loyalty Programs**: By segmenting customers based on factors like behaviour, preferences, and demographics, businesses can design personalized retention strategies. This might include targeted loyalty programs, exclusive offers, or personalized communications to strengthen customer relationships.

4. **Pricing Strategy**: Segmentation analysis informs pricing strategies by identifying segments willing to pay premium prices for certain features or benefits. It helps businesses optimize pricing to maximize profitability and competitiveness within each segment.

5. **Market Expansion and Entry**: Segmentation analysis guides decisions about entering new markets or expanding existing ones. It helps businesses identify untapped segments or niche markets that align with their capabilities and offerings.

6. **Advertising and Communication**: Segmentation analysis informs advertising and communication strategies by identifying which messages resonate most with each segment and which media channels are most effective in reaching them. This ensures that marketing communications are relevant and impactful.

7. **Channel Management**: Different market segments may prefer different distribution channels or methods of purchasing. Segmentation analysis helps businesses optimize their channel strategies to ensure products are available where and how customers prefer to buy.

8. **Sales and Territory Management**: Segmentation analysis aids in allocating sales resources effectively. It helps determine which sales techniques are most effective for each segment and how sales territories should be structured based on segment potential.

9. **Customer Service and Support**: Segmentation analysis allows businesses to customize their customer service and support offerings to meet the unique needs of different segments. This might include specialized support teams, tailored service packages, or preferred communication channels.

10. **Risk Management**: Diversifying across different segments can reduce risk by ensuring revenue is not overly dependent on any single segment. If one segment faces challenges, others can provide stability and continuity.

Overall, market segmentation analysis is a versatile tool that enables businesses to better understand their customers, tailor strategies and offerings, and ultimately improve competitiveness and profitability in the marketplace.

# 9. CONCLUSION

In conclusion, this project successfully developed and evaluated a machine learning-based market segmentation. Market segmentation is a vital strategy for businesses and organizations aiming to enhance their marketing effectiveness, customer satisfaction, and overall competitiveness. By dividing a broad market into distinct groups of consumers with common needs and characteristics, businesses can tailor their products, services, and marketing efforts to meet the specific demands of each segment. Future advancements in real-time data analytics and machine learning techniques will further improve the system's effectiveness and applicability in combating evolving segmentation problems.

# 10. FUTURE SCOPE

The future scope of market segmentation analysis is promising, driven by several trends and advancements in technology and consumer behaviour:

1. **Advanced Data Analytics**: With the rise of big data and AI-driven analytics, companies can now gather and analyze vast amounts of customer data in real-time. This enables more precise segmentation based on behavioural patterns, preferences, and interactions across multiple channels.

2. **Personalization**: Consumers increasingly expect personalized experiences and offerings. Market segmentation allows companies to tailor their marketing messages, products, and services to meet the unique needs of different customer segments, enhancing customer satisfaction and loyalty.

3. **Micro-Segmentation**: As technology improves, companies are able to identify smaller, more niche segments within broader markets. This allows for highly targeted marketing strategies that resonate more deeply with specific customer needs and preferences.

4. **Global Markets**: As businesses expand globally, understanding diverse consumer preferences and cultural nuances becomes crucial. Market segmentation helps companies adapt their offerings and marketing strategies to different regions and demographics, maximizing their relevance and effectiveness.

5. **Ethical and Responsible Marketing**: Consumers are increasingly concerned with ethical practices and social responsibility. Market segmentation can help companies align their offerings and messages with consumer values, fostering trust and loyalty.

6. **Integration of Channels**: With the proliferation of digital and social media channels, integrating segmentation across these platforms becomes essential. Companies can use segmentation to deliver consistent and relevant messages across multiple touchpoints, enhancing the overall customer experience.

7. **Predictive Segmentation**: Leveraging predictive analytics, companies can anticipate future trends and behaviours based on past data. This allows for proactive segmentation strategies that stay ahead of market shifts and evolving consumer preferences.

8. **Sustainability and Green Segmentation**: Growing awareness of environmental issues is influencing consumer behaviour. Market segmentation can help identify segments interested in sustainable products and practices, allowing companies to develop offerings that appeal to eco-conscious consumers.

# 12.BIBILOGRAPHY

## 1.Books

**Kotler, P., & Keller, K. L. (2016).** Marketing Management (15th Edition). Pearson Education.

This book covers various aspects of marketing management, including detailed chapters on market segmentation and target marketing.

**Wedel, M., & Kamakura, W. A. (2000).** Market Segmentation: Conceptual and Methodological Foundations. Springer Science & Business Media.

A comprehensive guide to the theory and practice of market segmentation, including advanced techniques and applications.

**Smith, W. R. (1956).** Product Differentiation and Market Segmentation as Alternative Marketing Strategies. Journal of Marketing, 21(1), 3-8.

A seminal paper that laid the foundation for modern market segmentation practices.

**Dibb, S., Simkin, L., Pride, W. M., & Ferrell, O. C. (2016).** Marketing Concepts and Strategies (7th Edition). Cengage Learning EMEA.

This book provides a thorough understanding of marketing concepts, including market segmentation strategies.

## 2.Articles

**Yankelovich, D., & Meer, D. (2006).** Rediscovering Market Segmentation. Harvard Business Review, 84(2), 122-131.

This article discusses the evolution of market segmentation and provides practical insights for modern marketers.

**Wind, Y. (1978).** Issues and Advances in Segmentation Research. Journal of Marketing Research, 15(3), 317-337.

A comprehensive review of the advances and ongoing issues in segmentation research.

**Hoek, J., Gendall, P., & Esslemont, D. (1996).** Market Segmentation: A Search for the Holy Grail? Journal of Marketing Practice: Applied Marketing Science, 2(1), 22-34.

This article examines the challenges and effectiveness of market segmentation in practice.

## 3.Online Resources

**American Marketing Association (AMA):** Market Segmentation

The AMA provides resources and articles on market segmentation, including best practices and case studies.

**Harvard Business Review (HBR):** Market Segmentation

A collection of articles and resources on market segmentation from the Harvard Business Review.

**Investopedia:** Market Segmentation

A detailed explanation of market segmentation concepts and types, with examples.
**Marketing Science Institute (MSI):** Market Segmentation and Target Marketing
Research reports and insights on market segmentation and target marketing from the MSI.
Academic Journals
**Journal of Marketing**
Publishes research on various aspects of marketing, including market segmentation.
**Journal of Marketing Research**
Focuses on the study of marketing research and methodologies, including segmentation techniques.
**Journal of Consumer Research**
Provides insights into consumer behavior, which is critical for effective market segmentation.

## Web Resources:

**Scikit-learn Documentation**: https://scikit-learn.org/stable/documentation.html
-Comprehensive documentation on the Scikit-learn library, which includes implementations of many machine learning algorithms used in the project.
**XGBoost Documentation**: https://xgboost.readthedocs.io/
-Detailed documentation for the XGBoost library, used for model training in the project.
**Flask Documentation**: https://flask.palletsprojects.com/
-Documentation for the Flask web framework, which was used to develop the user interface for the fraud detection system.
**Kaggle**: market segmentation analysis
**Dataset**: https://homepage.boku.ac.at/leisch/MSA/datasets/mcdonalds.csv
-Source of the dataset used for clustering the data

# 12. APPENDIX

**Model building:**

1)Dataset

2)Jupyter Notebook and VS code Application Building

1. HTML file (Home file, Predict file, Result file )
1. CSS file (Style file, Predict-style file)
2. Models in pickle format

## SOURCE CODE:

## HOME.HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>market segmentation</title>
    <link rel="stylesheet" href="{{url_for('static',filename='assets/css/index1.css')}}">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <style type="text/css">
    label{
    width:200px;
    display:inline-block;
    }
    </style>
  </head>
  <body>
     <div id="lg">
      <div id="ns">
        <header class="name">MARKET SEGMENTATION ANALYSIS:</header>
        <nav id="nav">
          <ul>
            <li><a href="#" class="active">Home</a></li>
            <li><a href="#about">About</a></li>
            <li><a href="#">Contact</a></li>
            <li><a href="#predict" id="pr">Predict</a></li>
          </ul>
        </nav>
      </div>
        <div id="tq">
        <h1 id="title">MCDONALDS</h1>
        </div>
      </div>
      <!--about-->
      <div id="about">
       <nav id="nava">
          <ul>
```

```
            <li><a href="#lg" class="active">Home</a></li>
            <li><a href="#predict" id="pr">Predict</a></li>
          </ul>
         </nav>
        <div id="ac">
         <h3><aside id="au">Aboutus:</aside></h3>
         <main id="cont">
          <b> <i><p>Market segmentation involves dividing the market into distinct groups of customers who
have similar needs, wants, or behaviors.</p>.
            <br><br>This segmentation allows businesses to tailor their marketing strategies, product
offerings, and services to better meet the specific needs of each segment.
            <br><br>By employing these segmentation strategies, McDonald's effectively targets different
customer segments with products and promotions that meet their specific needs and preferences. This
approach not only enhances customer satisfaction but also boosts sales and strengthens brand loyalty.
           </i></b></p>
         </main>
        </div>
       </div>
}
```

## PREDICT.HTML

```
<div id="predict">
        <header id="namep">MC DONALD'S</header>
        <div id="formbox">
          <h1><b>Please Fill the Details</b></h1>
          <form action="/output" method="POST">
          <table>
          <tr><td>
            <b><label for="yummy">YUMMY:</label></td><td>
            <input type="binaryInput" id="yummy" name="yummy" pattern1="[01]" required>
            <br></td> </tr> <tr><td>
            <b><label for="convenient">CONVENIENT:</label></td><td>
            <input type="binaryInput" id="convenient" name="convenient" pattern1="[01]" required>
            <br></td> </tr> <tr><td>
            <b><label for="spicy">SPICY:</label></td><td>
            <input type="binaryInput" id="spicy" name="spicy" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="fattening">FATTENING:</label></td><td>
            <input type="binaryInput" id="fattening" name="fattening" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="greasy">GREASY:</label></td><td>
            <input type="binaryInput" id="greasy" name="greasy" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="fast">FAST:</label></td><td>
            <input type="binaryInput" id="fast" name="fast" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="cheap">CHEAP:</label></td><td>
            <input type="binaryInput" id="cheap" name="cheap" pattern1="[01]" required>
```

```
            <br></td></tr><tr><td>
            <b><label for="tasty">TASTY:</label></td><td>
            <input type="binaryInput" id="tasty" name="tasty" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="expensive">EXPENSIVE:</label></td><td>
            <input type="binaryInput" id="expensive" name="expensive" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="healthy">HEALTHY:</label></td><td>
            <input type="binaryInput" id="healthy" name="healthy" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="disgusting">DISGUSTING:</label></td><td>
            <input type="binaryInput" id="disgusting" name="disgusting" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="Age">AGE:</label></td><td>
            <input type="binaryInput" id="Age" name="Age" pattern1="[01]" required>
            <br></td></tr><tr><td>
            <b><label for="Gender">GENDER:</label></td><td>
            <input type="binaryInput" id="Gender" name="Gender" pattern1="[01]" required>
            <br></td></tr><tr><tr><td>
            <center><button id="button" type="submit"  style="background-color:sky-blue"
calss="btn">Predict</button></center>
            </td></tr></tr>
            </b>
            </table>
          </form>
        </div>
      </div>

  </body>
</html>
```

**RESULT.HTML**

```
<!DOCTYPE html>
<html>
  <head>
    <title>market segmentation analysis</title>
    <link rel="stylesheet" href="{{url_for('static',filename='assets/css/output.css')}}">
  </head>
  <body>
    <center><header id="name" class="output">{{predict}}</header></center>
  </body>
</html>
```

## APP.PY

```
from flask import Flask,render_template,url_request
from flask import request,jsonify
import numpy as np
import pandas as pd
import pickle
import joblib
from sklearn.preprocessing import StandardScaler

model=joblib.load(open('model.pkl','rb'))
scaler=joblib.load(open('scaler3.pkl','rb'))
app=Flask(__name__)

@app.route('/')
def home():
    return render_template('index1.html')

@app.route('/output', methods=['GET', 'POST'])
def predict():
    yummy =int(request.form['yummy'])
    convenient=int(request.form['convenient'])
    spicy=int(request.form['spicy'])
    fattening=int(request.form['fattening'])
    greasy=int(request.form['greasy'])
    fast=int(request.form['fast'])
    cheap =int(request.form['cheap'])
    tasty =int(request.form['tasty'])
    expensive =int(request.form['expensive'])
    healthy =int(request.form['healthy'])
    disgusting=int(request.form['disgusting'])
    Age=int(request.form['Age'])
    Gender=int(request.form['Gender'])
    total=[[yummy, convenient, spicy, fattening, greasy, fast, cheap,
          tasty, expensive, healthy, disgusting, Age, Gender]]
    prediction=model.predict(scaler.transform(total))
    prediction =int(prediction[0])
    if prediction==0:
        return render_template('output.html',predict="Predicts Customer belong to cluster 0")
    if prediction==1:
        return render_template('output.html',predict="Predicts Customer belong to cluster 1")
    if prediction==2:
        return render_template('output.html',predict="Predicts Customer belong to cluster 2")
    else:
        return render_template('output.html',predict="Predicts Customer belong to cluster 3")
if __name__ == "__main__":
    app.run(debug=True)
```

# CODE SNIPPETS

## importing necessary libraries

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import seaborn as sns
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
from collections import Counter
```

## importing dataset (MCDONALD'S)

```
df=pd.read_csv(r"C:\Users\nagin\Downloads\mcdonalds.csv")
```

```
df.head()
```

Python

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Like | Age | VisitFrequency | Gender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | No | Yes | No | Yes | No | Yes | Yes | No | Yes | No | No | -3 | 61 | Every three months | Female |
| 1 | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | Yes | No | No | +2 | 51 | Every three months | Female |
| 2 | No | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes | Yes | No | +1 | 62 | Every three months | Female |
| 3 | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No | Yes | +4 | 69 | Once a week | Female |
| 4 | No | Yes | No | Yes | Yes | Yes | Yes | No | No | Yes | No | +2 | 49 | Once a month | Male |

+ Code + Markdown

```
df.tail()
```

Python

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Like | Age | VisitFrequency | Gender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1448 | No | Yes | No | Yes | Yes | No | No | No | Yes | No | Yes | I hate it!-5 | 47 | Once a year | Male |
| 1449 | Yes | Yes | No | Yes | No | No | Yes | Yes | No | Yes | No | +2 | 36 | Once a week | Female |
| 1450 | Yes | Yes | No | Yes | No | Yes | No | Yes | Yes | No | No | +3 | 52 | Once a month | Female |
| 1451 | Yes | Yes | No | No | No | Yes | Yes | Yes | No | Yes | No | +4 | 41 | Every three months | Male |
| 1452 | No | Yes | No | Yes | Yes | No | No | No | Yes | No | Yes | -3 | 30 | Every three months | Male |

shape of dataset

```
df.shape
```

[5]

```
(1453, 15)
```

## checking for null values

```
df.info()
```

[6]

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1453 entries, 0 to 1452
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   yummy           1453 non-null   object
 1   convenient      1453 non-null   object
 2   spicy           1453 non-null   object
 3   fattening       1453 non-null   object
 4   greasy          1453 non-null   object
 5   fast            1453 non-null   object
 6   cheap           1453 non-null   object
 7   tasty           1453 non-null   object
 8   expensive       1453 non-null   object
 9   healthy         1453 non-null   object
 10  disgusting      1453 non-null   object
 11  Like            1453 non-null   object
 12  Age             1453 non-null   int64 
 13  VisitFrequency  1453 non-null   object
 14  Gender          1453 non-null   object
dtypes: int64(1), object(14)
memory usage: 170.4+ KB
```

```
df.isnull().sum()
```

[7]

```
yummy           0
convenient      0
spicy           0
fattening       0
greasy          0
fast            0
cheap           0
tasty           0
expensive       0
healthy         0
disgusting      0
Like            0
Age             0
VisitFrequency  0
Gender          0
dtype: int64
```

Number of values

```
df['Age'].value_counts()
```

✓ 0.0s

```
Age
55    53
60    38
37    37
59    36
57    36
52    36
58    35
36    35
49    34
62    34
50    34
32    33
44    32
56    32
64    32
53    31
26    31
24    30
35    30
51    30
47    30
42    30
23    30
39    29
...
69    14
68    13
19    10
71     1
Name: count, dtype: int64
```

*Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...*

```
df['Like'].value_counts()
```

✓ 0.0s

```
Like
+3    229
+2    187
0     169
+4    160
+1    152
-5    152
+5    143
-3     73
-4     71
-2     59
-1     58
Name: count, dtype: int64
```

```
df['VisitFrequency'].value_counts()
```

```
VisitFrequency
Once a month            439
Every three months      342
Once a year             252
Once a week             235
Never                   131
More than once a week    54
Name: count, dtype: int64
```

```
df['Gender'].value_counts()
```

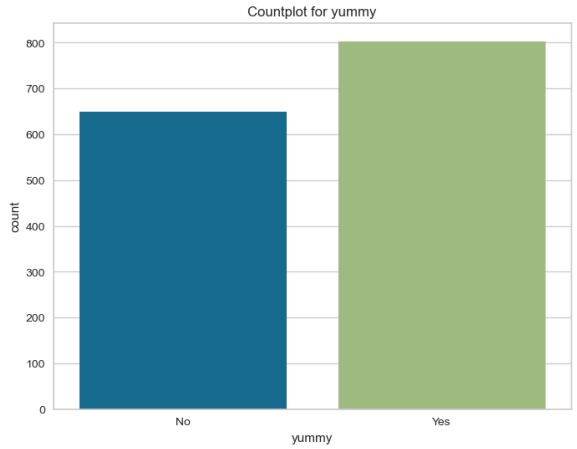
```
Gender
Female    788
Male      665
Name: count, dtype: int64
```

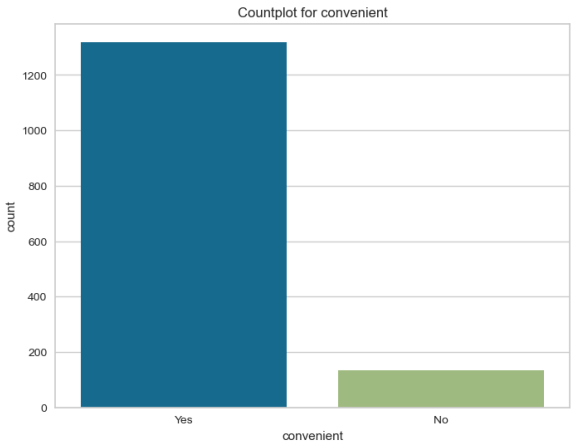
# univariate analysis

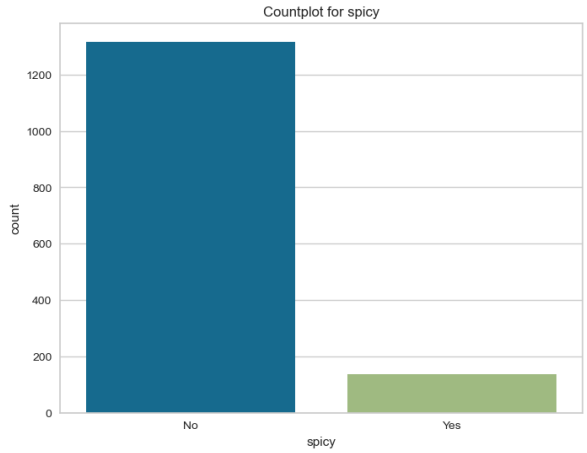
# countplot

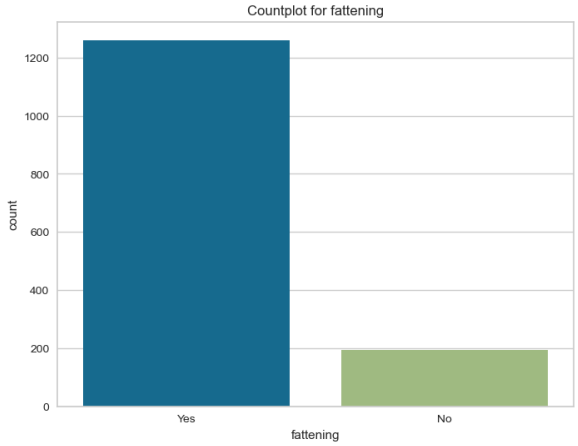
```
for column in df.columns:
    plt.figure(figsize=(8, 6))
    sns.countplot(x=df[column])
    plt.title(f'Countplot for {column}')
    plt.xlabel(column)
    plt.show()
```

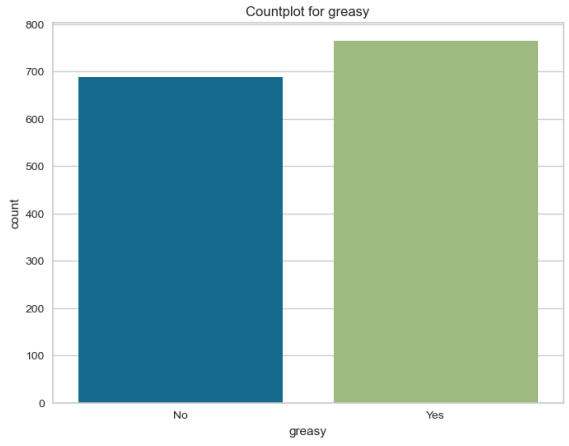
✓ 42s

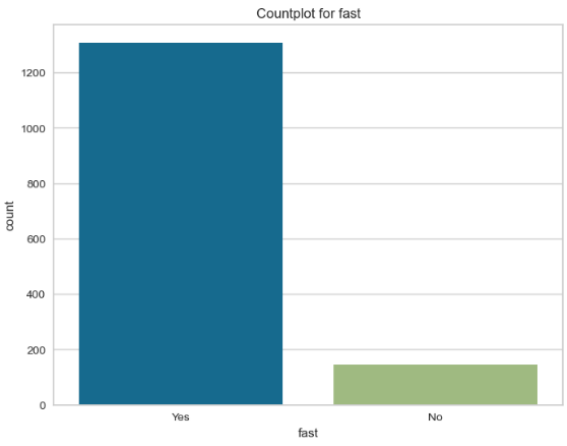


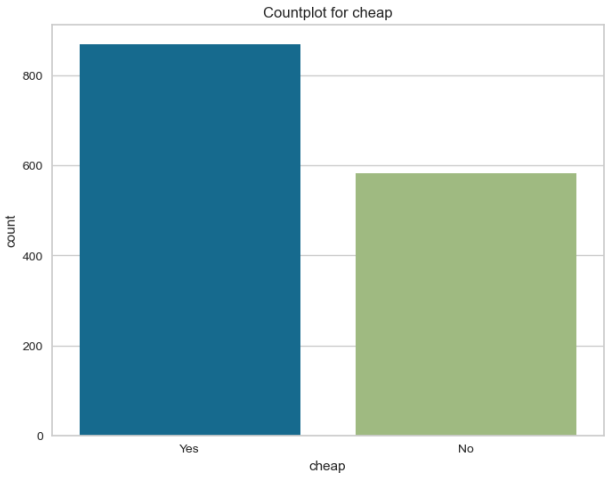



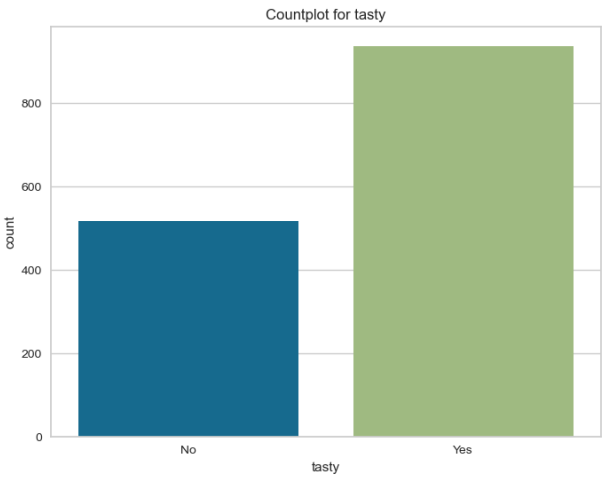



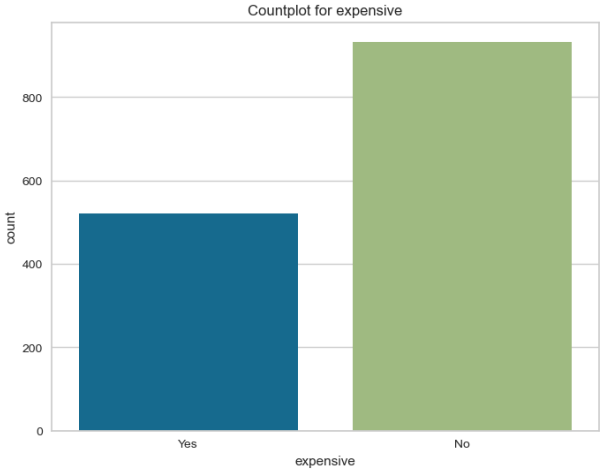



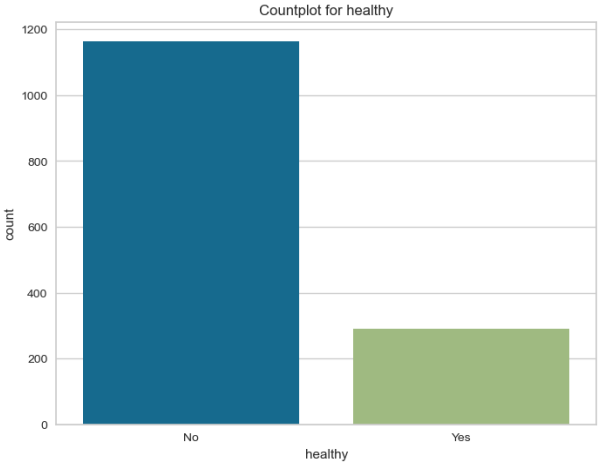



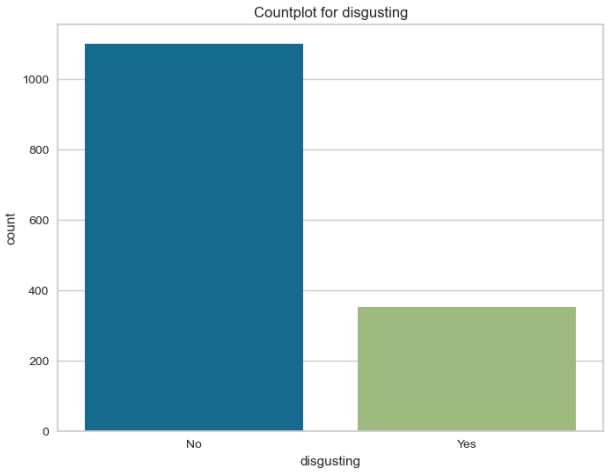



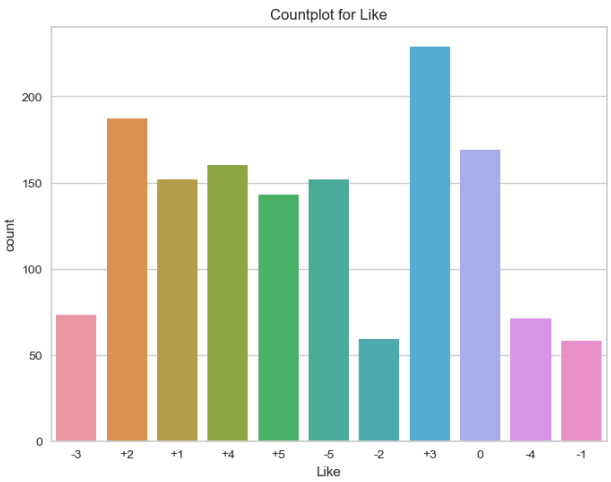

Countplot for cheap
count
cheap
Yes
No
0
200
400
600
800

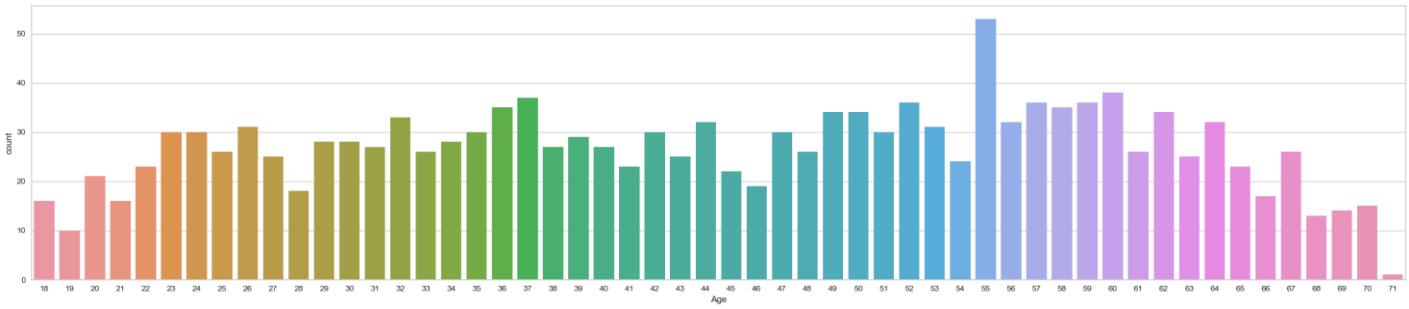

Countplot for tasty
count
tasty
No
Yes
0
200
400
600
800

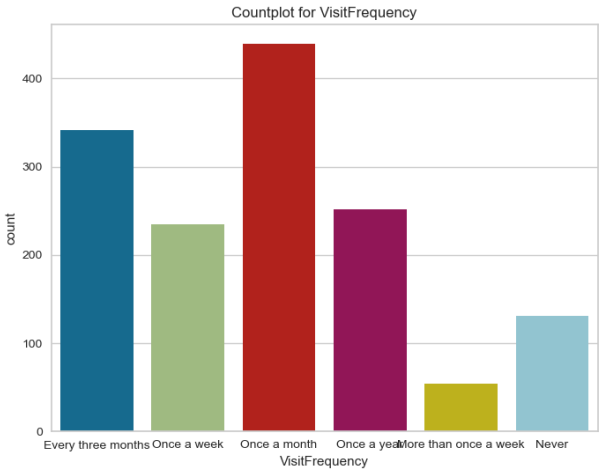

Countplot for expensive
count
expensive
Yes
No
0
200
400
600
800

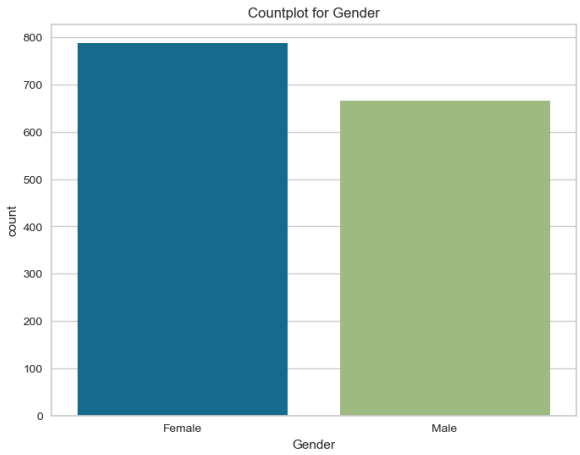

Countplot for healthy
count
healthy
No
Yes
0
200
400
600
800
1000
1200


Countplot for disgusting
count
disgusting
No
Yes
0
200
400
600
800
1000


Countplot for Like
count
Like
-3
+2
+1
+4
+5
-5
-2
+3
0
-4
-1
0
50
100
150
200



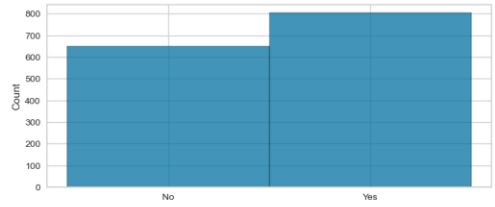



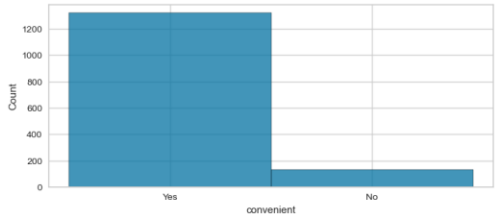



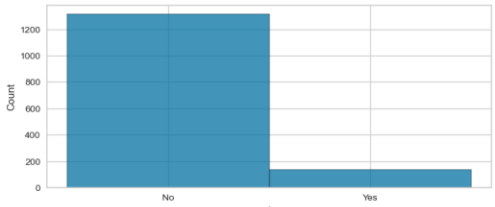

histplot

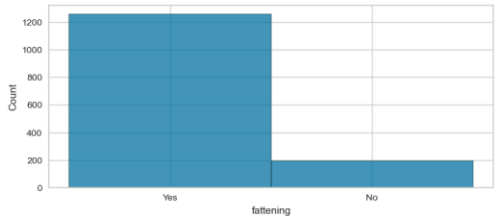
```
plt.figure(figsize=(15,50))
for i, col in enumerate(df.columns):
    plt.subplot(len(df.columns), 2, i+1)
    sns.histplot(df[col])
    plt.tight_layout()
```

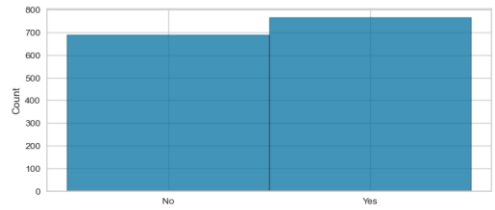
146] ✓ 2.8s

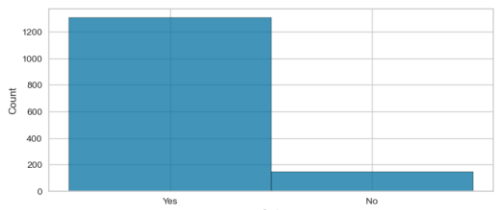

Count
No
Yes
yummy

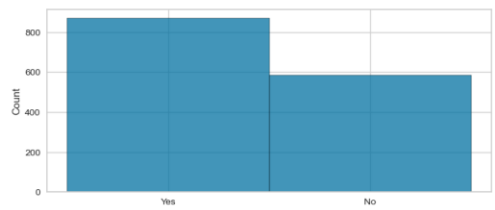

Count
Yes
No
convenient

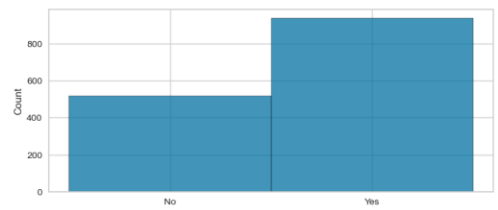

Count
No
Yes
spicy

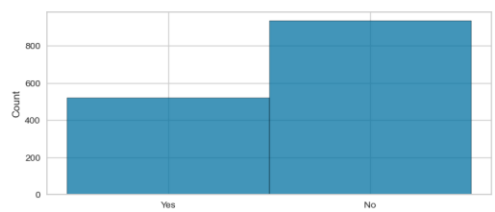

Count
Yes
No
fattening

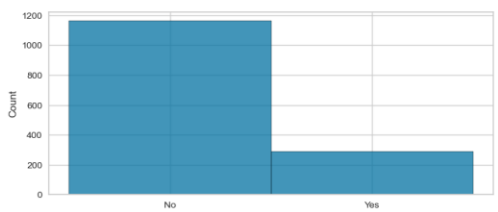

Count
No
Yes
greasy

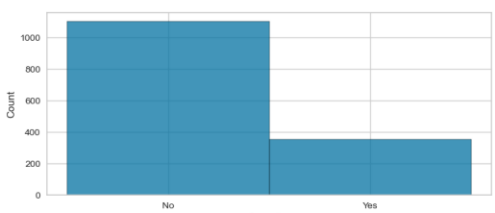

Count
Yes
No
fast

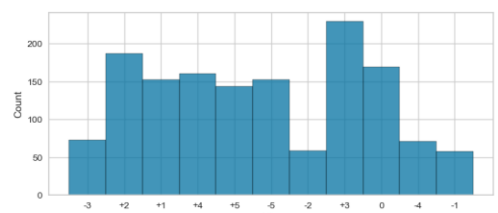

Count
Yes
No
cheap

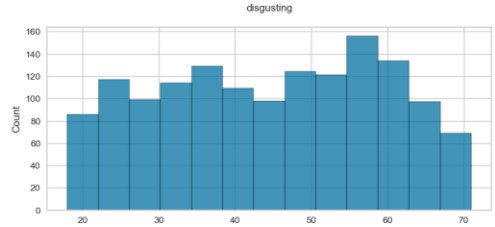

Count
No
Yes
tasty

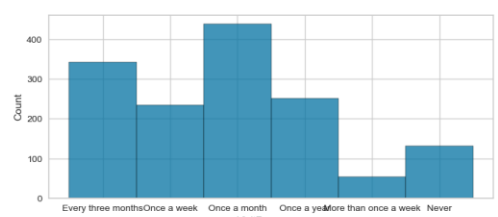

Count
Yes
No
expensive

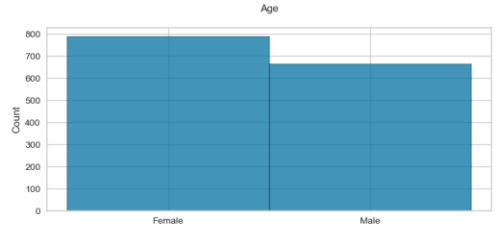

Count
No
Yes
healthy


Count
No
Yes
disgusting


Count
-3
+2
+1
+4
+5
-5
-2
+3
0
-4
-1
Like


Count
20
30
40
50
60
70
Age

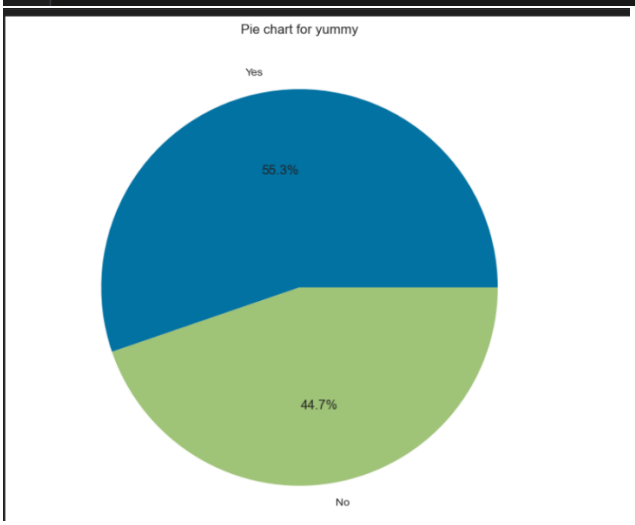

Count
Every three months
Once a week
Once a month
Once a year
More than once a week
Never
VisitFrequency

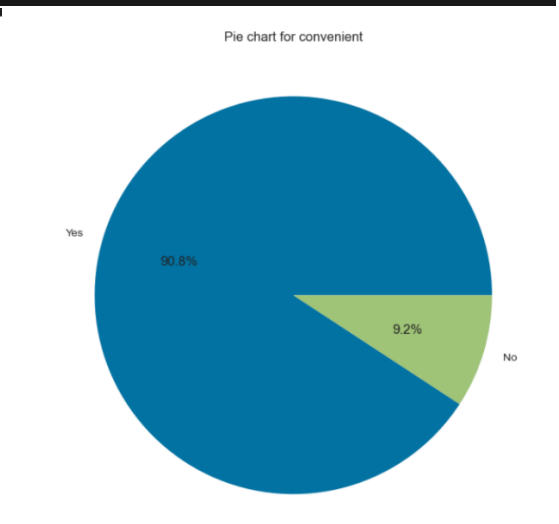

Count
Female
Male
Gender

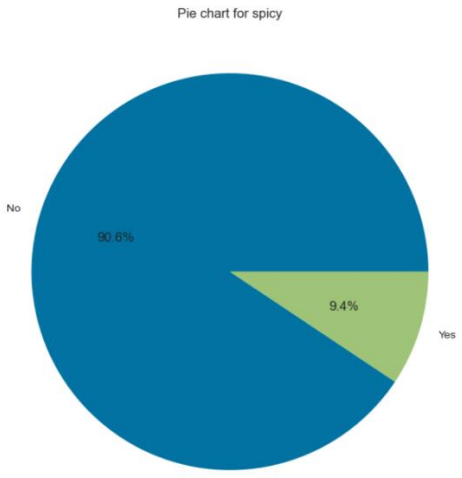
pie chart

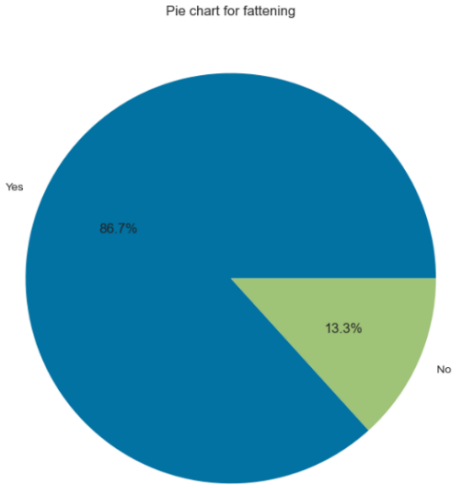
```
plt.figure(figsize=(100,100))
for i, col in enumerate(df.columns):
    plt.subplot(len(df.columns), 1, i+1)
    plt.title(f'Pie chart for {col}')
    x = df[col].value_counts()
    plt.pie(x.values, labels=x.index, autopct='%1.1f%%')
plt.tight_layout()
plt.show()
```

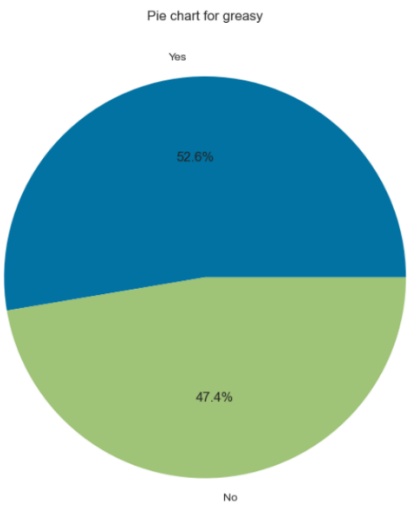
[191] ✓ 1.9s

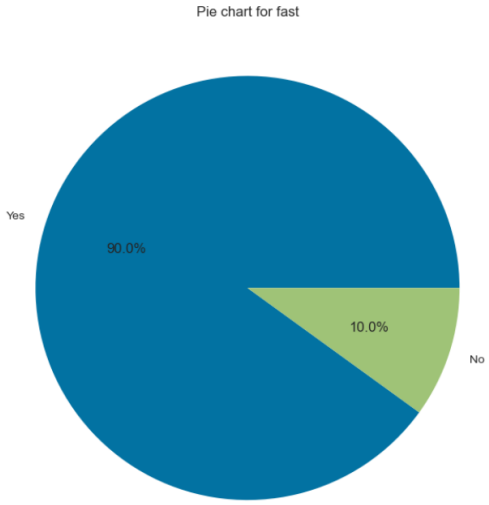


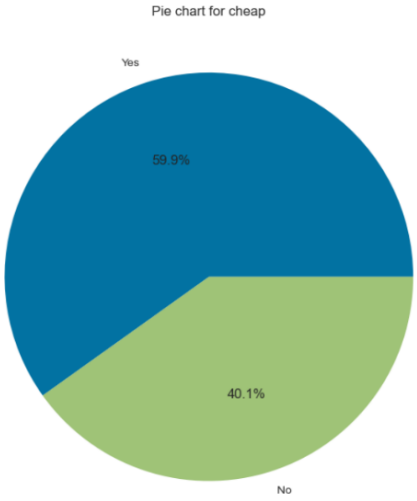



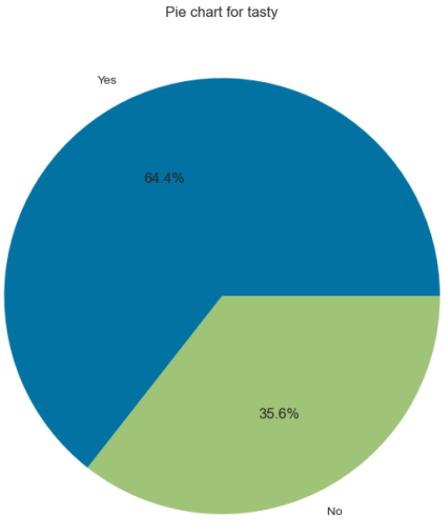



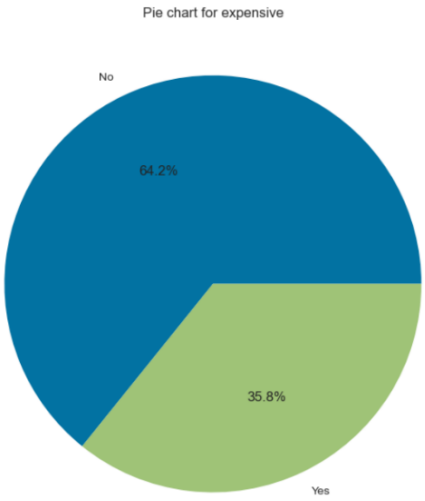



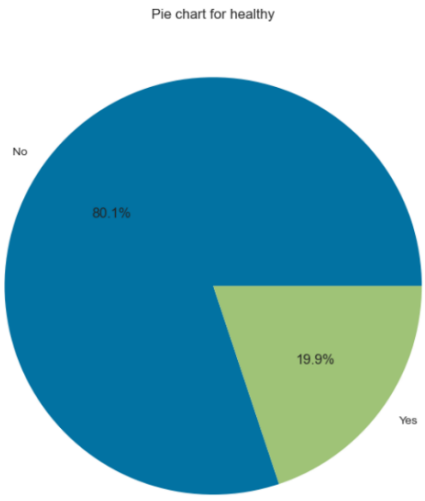

Pie chart for greasy
Yes
52.6%
47.4%
No


Pie chart for fast
Yes
90.0%
10.0%
No

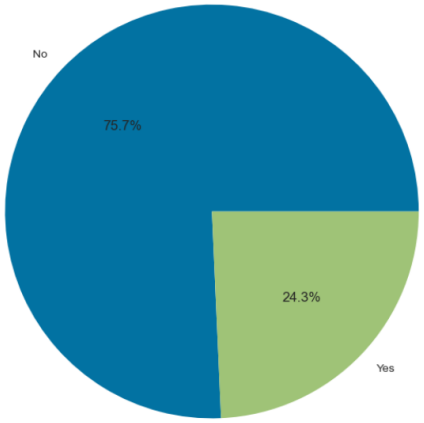

Pie chart for cheap
Yes
59.9%
40.1%
No


Pie chart for tasty
Yes
64.4%
35.6%
No

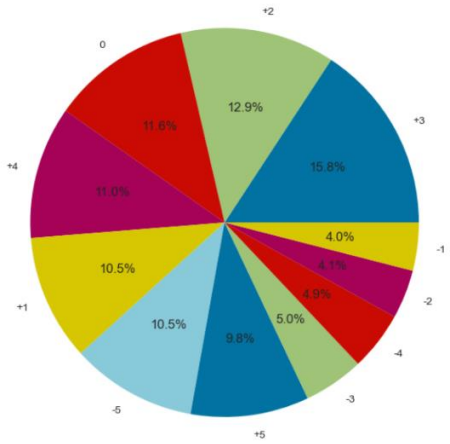

Pie chart for expensive
No
64.2%
35.8%
Yes


Pie chart for healthy
No
80.1%
19.9%
Yes

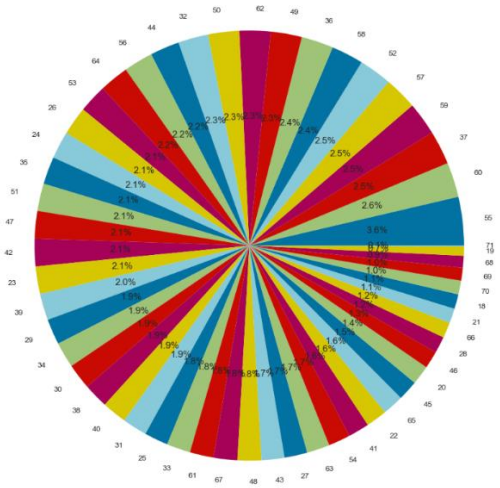
Pie chart for disgusting



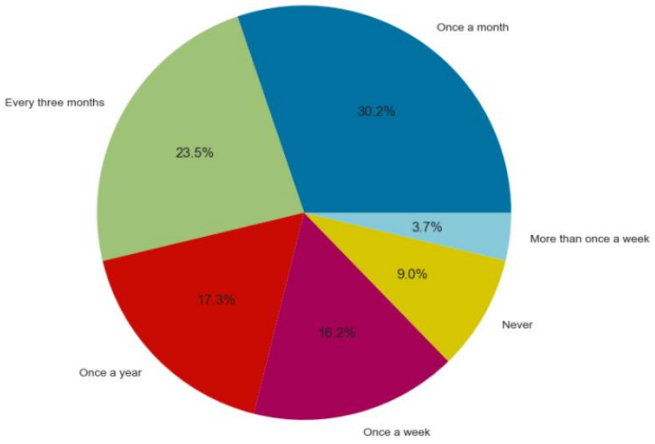

Pie chart for Like



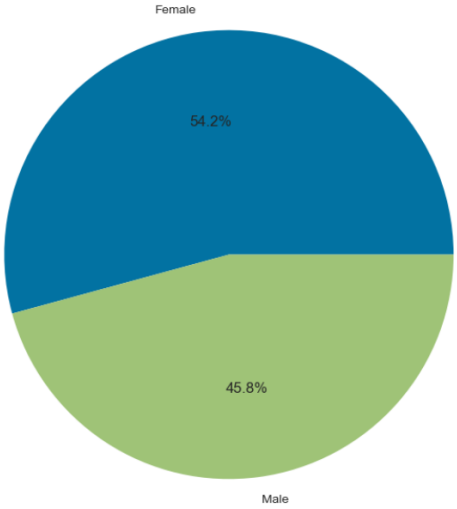

Pie chart for Age



Pie chart for VisitFrequency




Pie chart for Gender

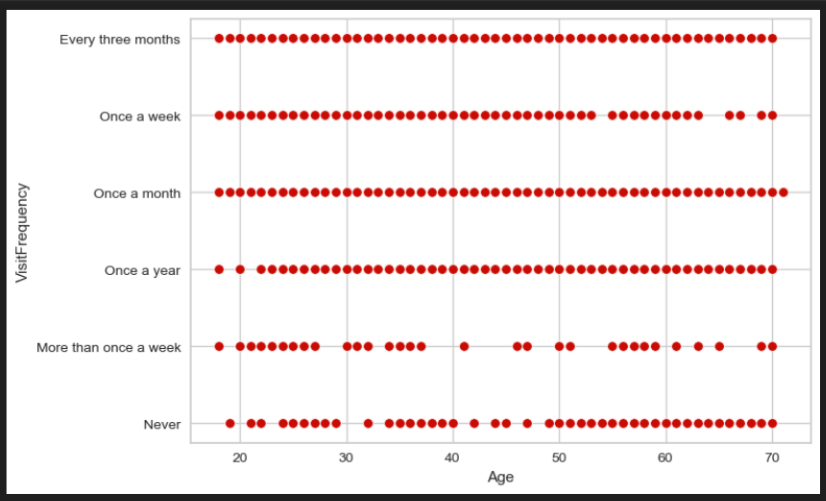


# bivariate analysis

## scatter plots

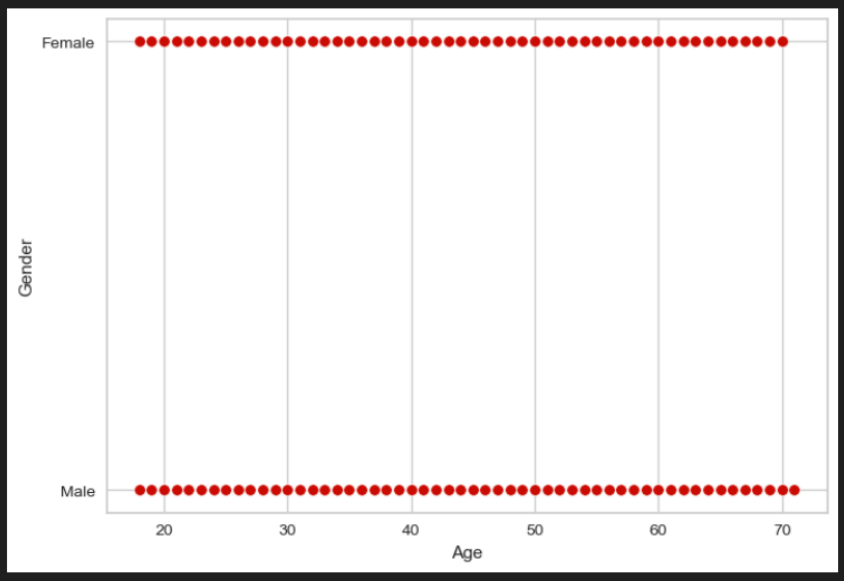
```
for i,col in enumerate(df.columns):
    sns.scatterplot(x='Age',y='VisitFrequency',data=df)
```

0.4s




```
for i,col in enumerate(df.columns):
    sns.scatterplot(x='Age',y='Gender',data=df)
```

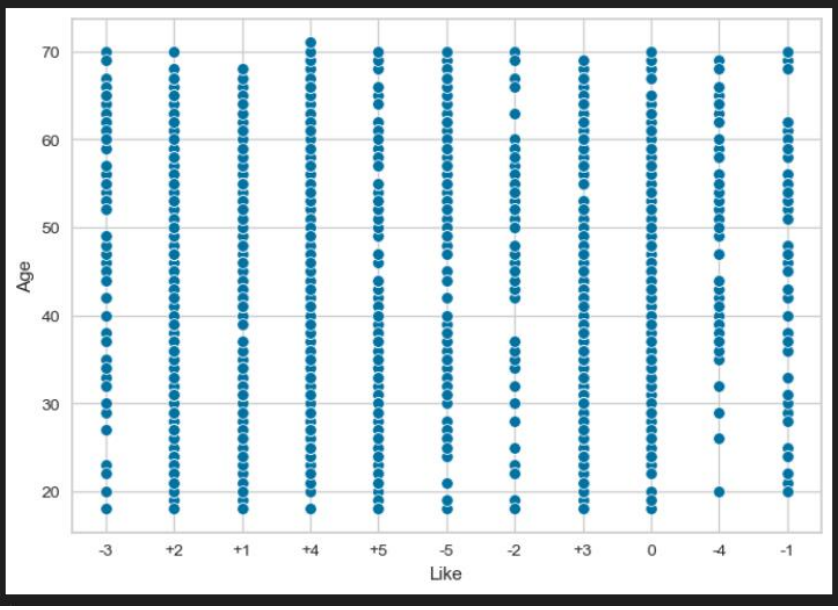
0.7s



```
sns.scatterplot(x='Like',y='Age',data=df)
```

✓ 0.1s

```
<Axes: xlabel='Like', ylabel='Age'>
```




## descriptive analysis

```
df.describe(include='all')
```

✓ 0.1s

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Like | Age | VisitFrequency | Gender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453 | 1453.000000 | 1453 | 1453 |
| unique | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 11 | NaN | 6 | 2 |
| top | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No | No | +3 | NaN | Once a month | Female |
| freq | 803 | 1319 | 1317 | 1260 | 765 | 1308 | 870 | 936 | 933 | 1164 | 1100 | 229 | NaN | 439 | 788 |
| mean | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 44.604955 | NaN | NaN |
| std | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 14.221178 | NaN | NaN |
| min | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 18.000000 | NaN | NaN |
| 25% | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 33.000000 | NaN | NaN |
| 50% | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 45.000000 | NaN | NaN |
| 75% | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 57.000000 | NaN | NaN |
| max | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 71.000000 | NaN | NaN |

## changing categorical values to float values

```
label_encoder = LabelEncoder()
df['yummy'] = label_encoder.fit_transform(df['yummy'])
df['convenient'] = label_encoder.fit_transform(df['convenient'])
df['spicy'] = label_encoder.fit_transform(df['spicy'])
df['fattening'] = label_encoder.fit_transform(df['fattening'])
df['greasy'] = label_encoder.fit_transform(df['greasy'])
df['fast'] = label_encoder.fit_transform(df['spicy'])
df['cheap'] = label_encoder.fit_transform(df['cheap'])
df['tasty'] = label_encoder.fit_transform(df['tasty'])
df['expensive'] = label_encoder.fit_transform(df['expensive'])
df['healthy'] = label_encoder.fit_transform(df['healthy'])
df['disgusting'] = label_encoder.fit_transform(df['disgusting'])
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['VisitFrequency'] = label_encoder.fit_transform(df['VisitFrequency'])
df['Like'] = label_encoder.fit_transform(df['Like'])
```

[30] ✓ 0.0s

```
df
```

✓ 0.0s

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Like | Age | VisitFrequency | Gender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 7 | 61.0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 51.0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 62.0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 3 | 69.0 | 4 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 49.0 | 3 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1448 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 47.0 | 5 | 1 |
| 1449 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 36.0 | 4 | 0 |
| 1450 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 52.0 | 3 | 0 |
| 1451 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 3 | 41.0 | 0 | 1 |
| 1452 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 7 | 30.0 | 0 | 1 |

1453 rows × 15 columns

```
df.describe(include='all')
```

✓ 0.0s

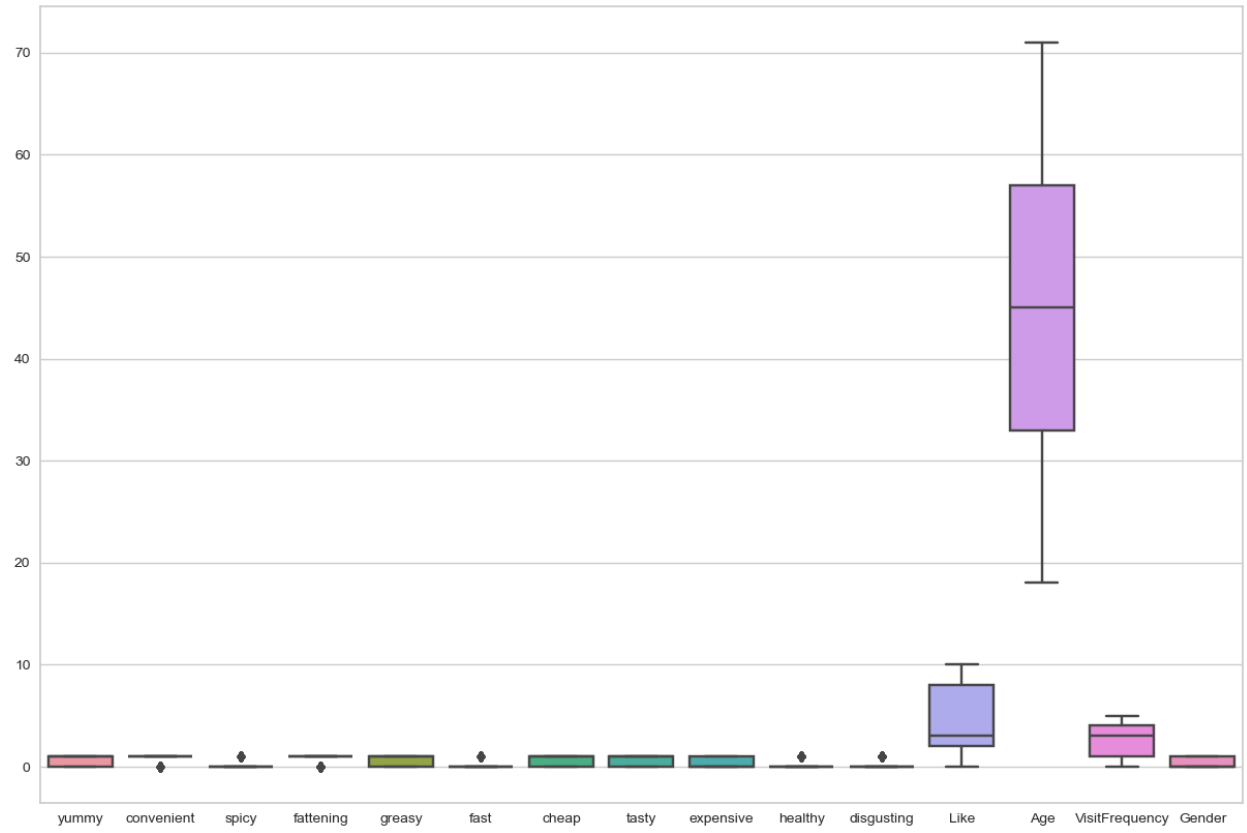
| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Like | Age | VisitFrequency | Gender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 1453.000000 | 1453.0 | 1453.0 | 1453.0 | 1453.000000 | 1453.0 | 1453.000000 | 1453.000000 | 1453.000000 | 1453.0 | 1453.0 | 1453.000000 | 1453.000000 | 1453.000000 | 1453.000000 |
| mean | 0.552650 | 0.0 | 0.0 | 0.0 | 0.526497 | 0.0 | 0.598761 | 0.644184 | 0.357880 | 0.0 | 0.0 | 4.458362 | 44.604955 | 2.637990 | 0.457674 |
| std | 0.497391 | 0.0 | 0.0 | 0.0 | 0.499469 | 0.0 | 0.490318 | 0.478925 | 0.479542 | 0.0 | 0.0 | 3.407245 | 14.221178 | 1.756057 | 0.498377 |
| min | 0.000000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.000000 | 18.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.0 | 2.000000 | 33.000000 | 1.000000 | 0.000000 |
| 50% | 1.000000 | 0.0 | 0.0 | 0.0 | 1.000000 | 0.0 | 1.000000 | 1.000000 | 0.000000 | 0.0 | 0.0 | 3.000000 | 45.000000 | 3.000000 | 0.000000 |
| 75% | 1.000000 | 0.0 | 0.0 | 0.0 | 1.000000 | 0.0 | 1.000000 | 1.000000 | 1.000000 | 0.0 | 0.0 | 8.000000 | 57.000000 | 4.000000 | 1.000000 |
| max | 1.000000 | 0.0 | 0.0 | 0.0 | 1.000000 | 0.0 | 1.000000 | 1.000000 | 1.000000 | 0.0 | 0.0 | 10.000000 | 71.000000 | 5.000000 | 1.000000 |

boxplot

```
plt.figure(figsize=(15,10))
sns.boxplot(df)
```

✓ 0.3s

```
<Axes: >
```

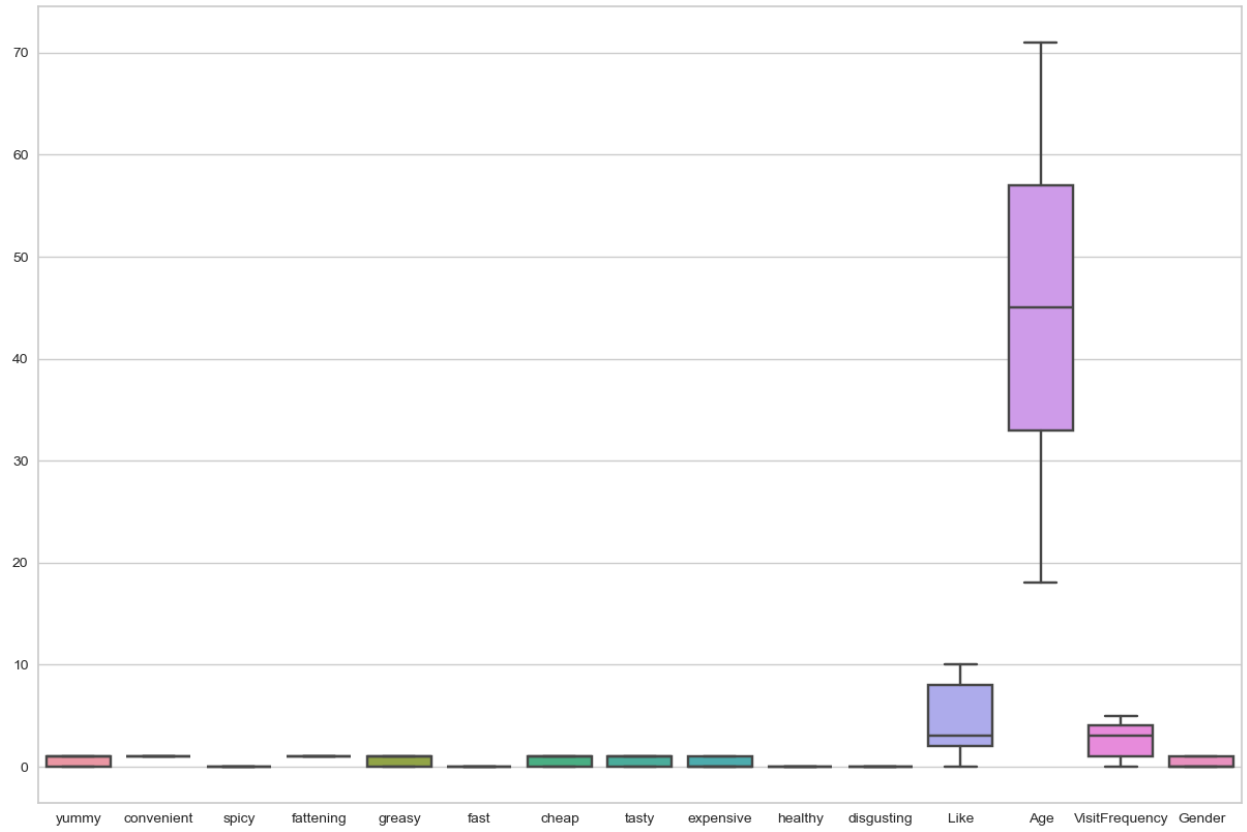



FOUND OUTLIERS IN THE DATASET.

handling outliers

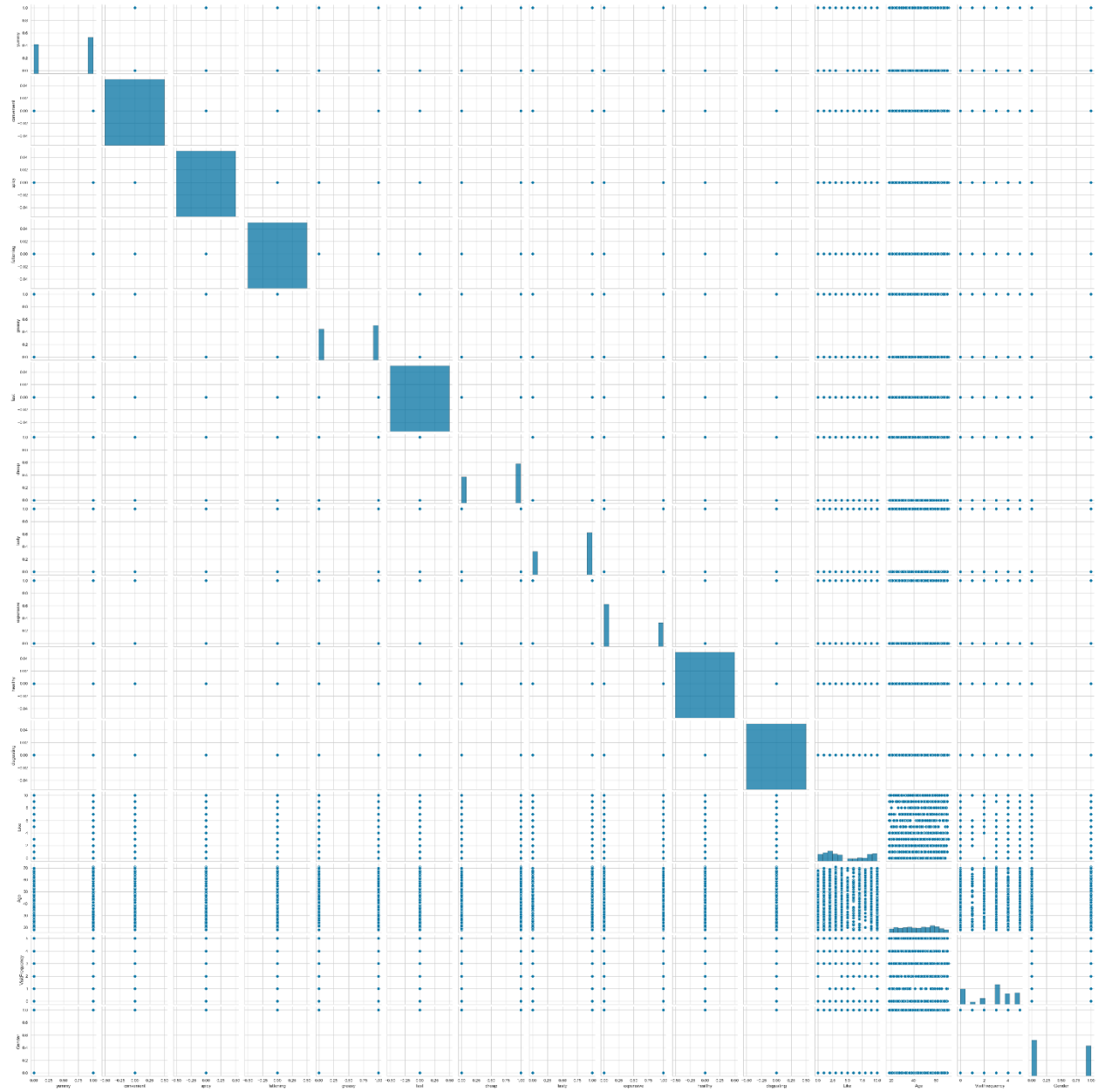
```
for column in df.columns:
    if pd.api.types.is_numeric_dtype(df[column]):
        quant = df[column].quantile(q=[0.75, 0.25])
        Q3 = quant.loc[0.75]
        Q1 = quant.loc[0.25]
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df[column] = np.where(df[column] < lower_bound, lower_bound,df[column])
        df[column] = np.where(df[column] > upper_bound, upper_bound,df[column])
```

```
plt.figure(figsize=(15,10))
sns.boxplot(df)
```
✓ 0.2s



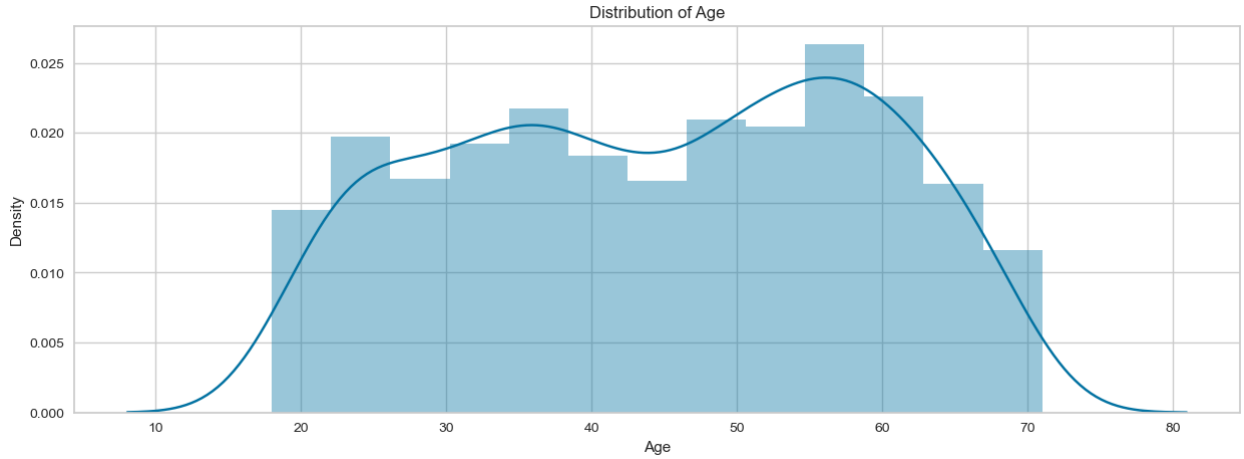

# multivariate analysis

```
[33]
sns.pairplot(df)
```
✓ 43.7s



distplot

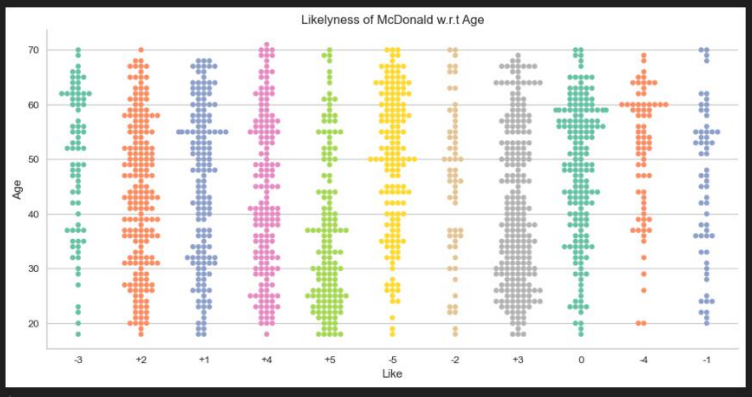
```
plt.figure(figsize=(15,5))
plt.title("Distribution of Age")
sns.distplot(df['Age'])
```




pyschographic segmentation

```
sns.catplot(x="Like", y="Age",data=df,
            orient="v", height=5, aspect=2, palette="Set2",kind="swarm")
plt.title('Likelyness of McDonald w.r.t Age')
plt.show()
```

```
C:\Users\nagin\AppData\Local\Temp\ipykernel_28300\702863475.py:1: FutureWarning: Passing `palette` without assigning `hue` is deprecated.
  sns.catplot(x="Like", y="Age",data=df,
C:\Users\nagin\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\nagin\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



## Droping unnecessary columns

```
df = df.drop(columns=['Like', 'VisitFrequency'])
```
✓ 0.0s

```
df
```
✓ 0.0s

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Age | Gender |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 61.0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 51.0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 62.0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 69.0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 49.0 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1448 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 47.0 | 1 |
| 1449 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 36.0 | 0 |
| 1450 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 52.0 | 0 |
| 1451 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 41.0 | 1 |
| 1452 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 30.0 | 1 |

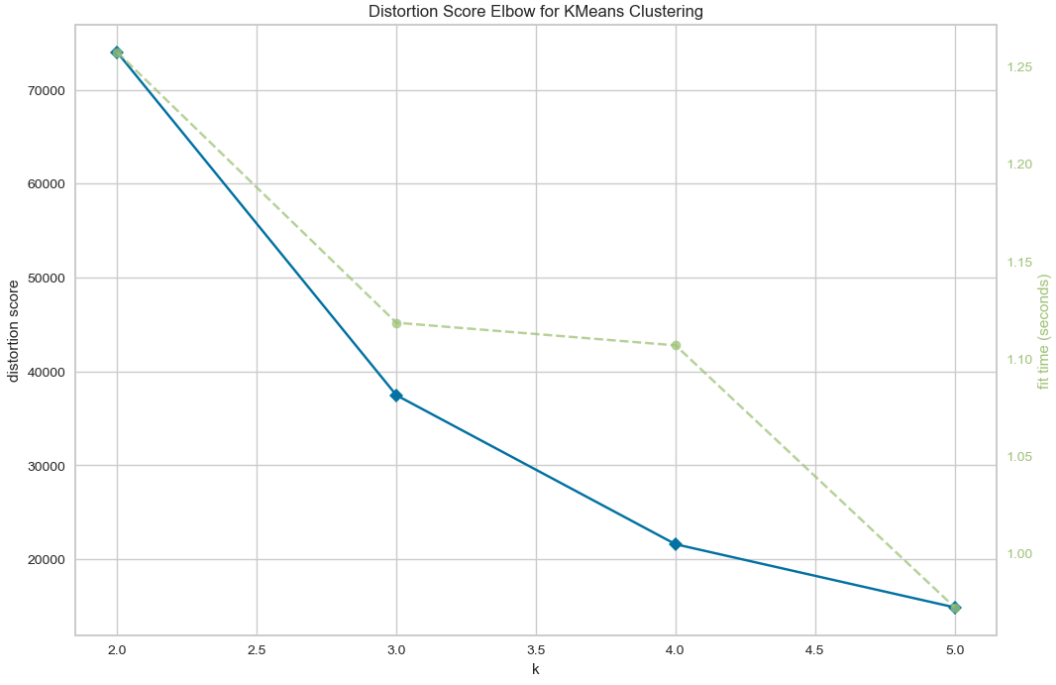
1453 rows × 13 columns

# MODEL BUILDING

model building

kmeans clustering

```
fig = plt.figure(figsize=(12,8))
df.head()
X = df[["yummy", "convenient","spicy","fattening","greasy","fast","cheap","tasty","expensive","healthy","disgusting","Age","Gender"]]
model = KMeans()
visualizer = KElbowVisualizer(model, k=(2,6))
visualizer.fit(X)
visualizer.poof()
```
✓ 4.9s



```
kmeans=KMeans(n_clusters=4)
kmeans.fit(X)
```

```
C:\Users\nagin\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the wa
  warnings.warn(
C:\Users\nagin\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid
  warnings.warn(
```

```
KMeans(n_clusters=4)
```

```
y_kmeans= kmeans.fit_predict(X)

df['cluster'] = pd.DataFrame(y_kmeans)

df.head()
```

```
C:\Users\nagin\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\nagin\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by se
  warnings.warn(
```

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Like | Age | VisitFrequency | Gender | cluster |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 7.0 | 61.0 | 0.0 | 0.0 | 3 |
| 1 | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 51.0 | 0.0 | 0.0 | 0 |
| 2 | 0.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 62.0 | 0.0 | 0.0 | 3 |
| 3 | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 3.0 | 69.0 | 4.0 | 0.0 | 3 |
| 4 | 0.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 49.0 | 3.0 | 1.0 | 0 |

```
x = df.values
x
```

```
array([[0., 1., 0., ..., 0., 0., 3.],
       [1., 1., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 3.],
       ...,
       [1., 1., 0., ..., 3., 0., 0.],
       [1., 1., 0., ..., 0., 1., 1.],
       [0., 1., 0., ..., 0., 1., 2.]])
```

# cluster size

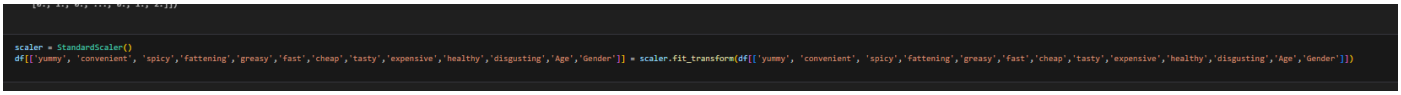
```
Counter(kmeans.labels_)
```

```
Counter({0: 407, 1: 382, 3: 335, 2: 329})
```

```
inertia = kmeans.inertia_
print(f'Inertia: {inertia}')
```

```
Inertia: 21630.404747587723
```

STANDARD SCALAR



```
df
```

| | yummy | convenient | spicy | fattening | greasy | fast | cheap | tasty | expensive | healthy | disgusting | Age | Gender | cluster |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -1.111479 | 0.0 | 0.0 | 0.0 | -1.054476 | 0.0 | 0.818605 | -1.345528 | 1.339489 | 0.0 | 0.0 | 1.153258 | -0.918645 | 3 |
| 1 | 0.899702 | 0.0 | 0.0 | 0.0 | 0.948339 | 0.0 | 0.818605 | 0.743203 | 1.339489 | 0.0 | 0.0 | 0.449839 | -0.918645 | 0 |
| 2 | -1.111479 | 0.0 | 0.0 | 0.0 | 0.948339 | 0.0 | -1.221590 | 0.743203 | 1.339489 | 0.0 | 0.0 | 1.223600 | -0.918645 | 3 |
| 3 | 0.899702 | 0.0 | 0.0 | 0.0 | 0.948339 | 0.0 | 0.818605 | 0.743203 | -0.746553 | 0.0 | 0.0 | 1.715993 | -0.918645 | 3 |
| 4 | -1.111479 | 0.0 | 0.0 | 0.0 | 0.948339 | 0.0 | 0.818605 | -1.345528 | -0.746553 | 0.0 | 0.0 | 0.309156 | 1.088560 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1448 | -1.111479 | 0.0 | 0.0 | 0.0 | 0.948339 | 0.0 | -1.221590 | -1.345528 | 1.339489 | 0.0 | 0.0 | 0.168472 | 1.088560 | 0 |
| 1449 | 0.899702 | 0.0 | 0.0 | 0.0 | -1.054476 | 0.0 | 0.818605 | 0.743203 | -0.746553 | 0.0 | 0.0 | -0.605289 | -0.918645 | 1 |
| 1450 | 0.899702 | 0.0 | 0.0 | 0.0 | -1.054476 | 0.0 | -1.221590 | 0.743203 | 1.339489 | 0.0 | 0.0 | 0.520181 | -0.918645 | 0 |
| 1451 | 0.899702 | 0.0 | 0.0 | 0.0 | -1.054476 | 0.0 | 0.818605 | 0.743203 | -0.746553 | 0.0 | 0.0 | -0.253579 | 1.088560 | 1 |
| 1452 | -1.111479 | 0.0 | 0.0 | 0.0 | 0.948339 | 0.0 | -1.221590 | -1.345528 | 1.339489 | 0.0 | 0.0 | -1.027340 | 1.088560 | 2 |

1453 rows × 14 columns

IMPORTING JOBLIB AND FLASK

```
import joblib
import pickle
```

```
joblib.dump(scaler, 'scaler3.pkl')
```

```
['scaler3.pkl']
```

```
pickle.dump(kmeans,open("model.pkl","wb"))
```