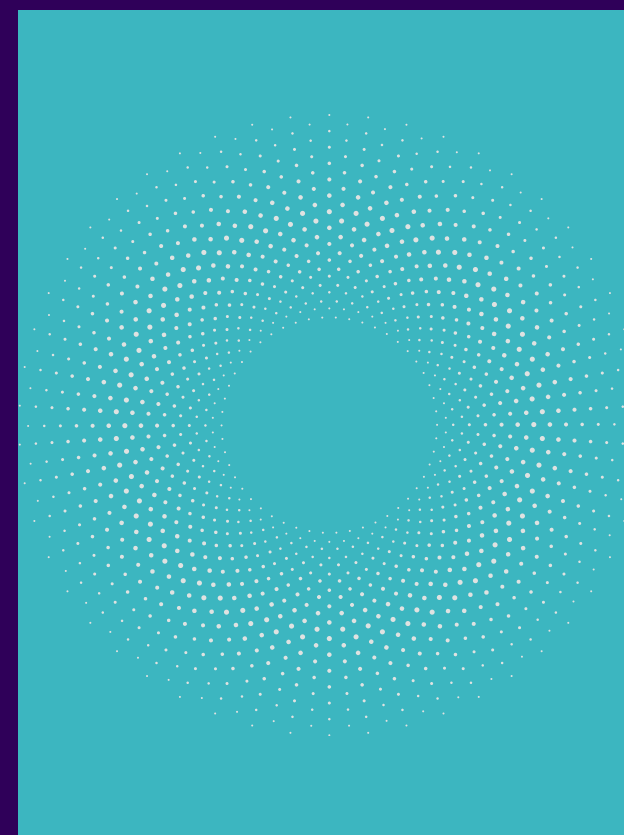


# MARKET & BASKET





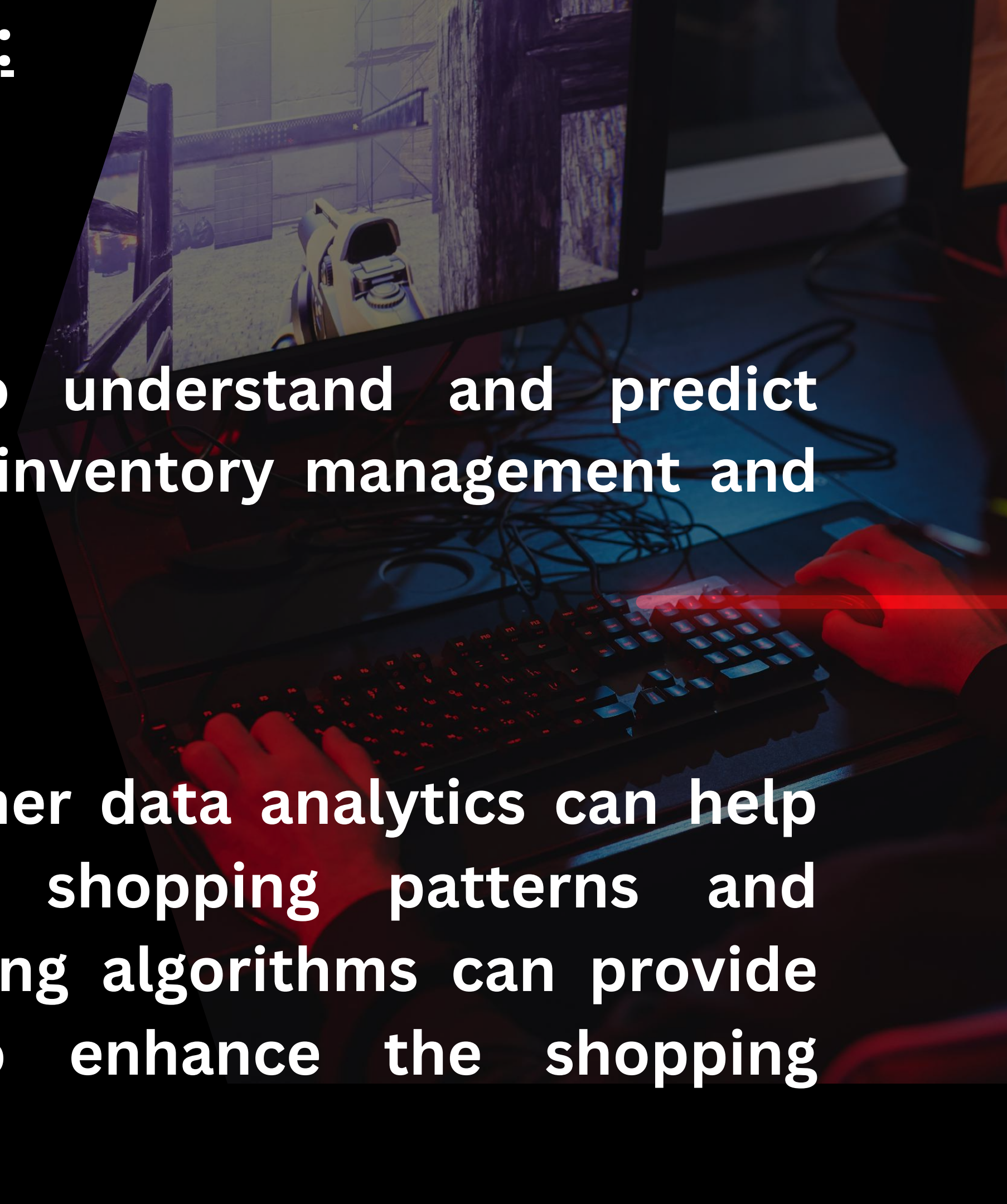
# Shopper Behavior Understanding:

## Problem statement:

- Supermarkets often struggle to understand and predict shopper behavior, which affects inventory management and marketing strategies.

## Solution for problem:

Market basket analysis and customer data analytics can help supermarkets gain insights into shopping patterns and preferences. AI and machine learning algorithms can provide personalized recommendations to enhance the shopping experience





Certainly, here's a well-structured README file template that you can use to explain how to run your code and its dependencies. Be sure to customize it to fit your specific project **code:**

**# Project Name**

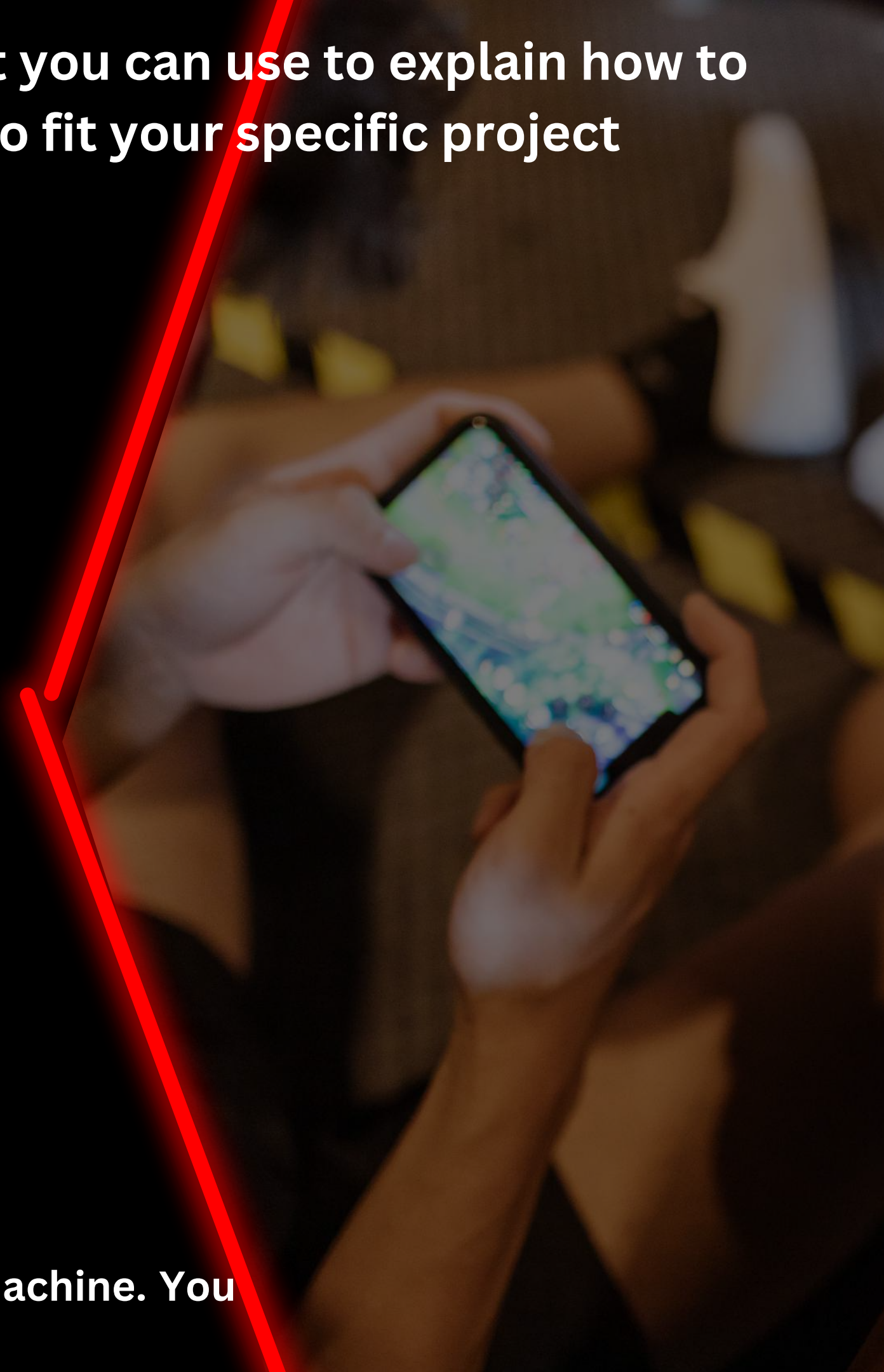
**Brief description of your project.**

**## Table of Contents**

- [Getting Started](#getting-started)
- [Prerequisites](#prerequisites)
- [Installation](#installation)
- [Usage](#usage)
- [Dependencies](#dependencies)
- [Contributing](#contributing)
- [License](#license)
- [Acknowledgments](#acknowledgments)

**## Getting Started**

**Explain how to get a copy of the project up and running on the user's local machine. You can include instructions for both downloading and cloning the project.**





## Usage:

**Explain how to use your code or application. Provide examples and usage scenarios. Include any command-line options, configurations, or environment variables**

```
bash
Copy code
# Example usage
python main.py --input_file data.txt
```

## Dependencies:

**List all the external libraries, packages, or tools that your project depends on. Include version information if relevant.**

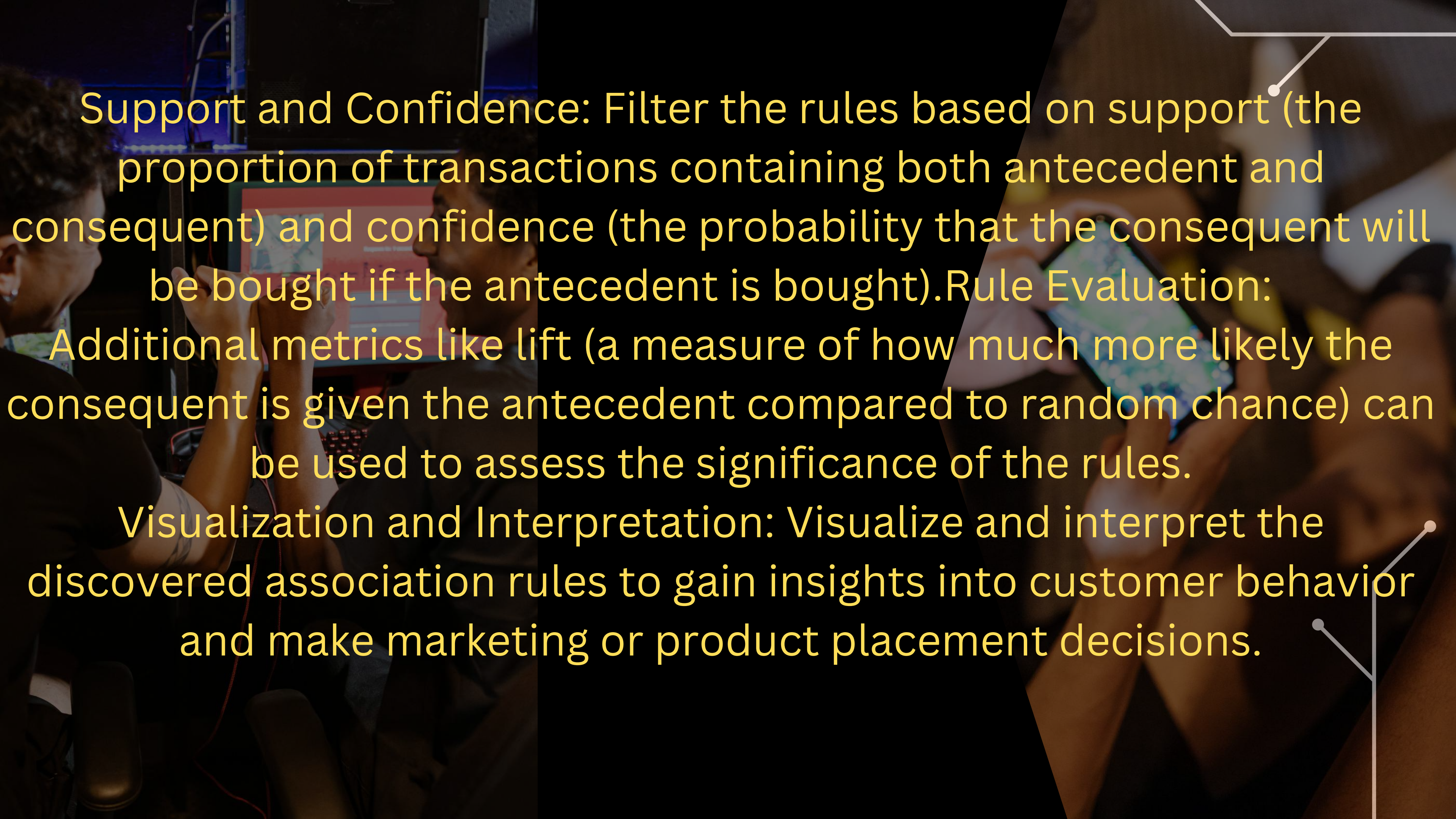
- **ExampleLibrary 2.0.0**
- **AnotherDependency 1.1.0**



# ASSOCIATION RULES:

- Data Collection: Gather transaction data from a market or store, where each transaction lists the items purchased by a customer.
- Data Preprocessing: Clean the data by removing duplicates, missing values, or irrelevant information.
- Frequent Itemset Generation: Identify frequent itemsets, which are combinations of items that occur together frequently in the transactions.
- Association Rule Generation: Create association rules, which describe relationships between items. These rules typically consist of an antecedent (if-portion) and a consequent (then-portion)





Support and Confidence: Filter the rules based on support (the proportion of transactions containing both antecedent and consequent) and confidence (the probability that the consequent will be bought if the antecedent is bought). Rule Evaluation:

Additional metrics like lift (a measure of how much more likely the consequent is given the antecedent compared to random chance) can be used to assess the significance of the rules.

Visualization and Interpretation: Visualize and interpret the discovered association rules to gain insights into customer behavior and make marketing or product placement decisions.



# **Include the dataset source and a brief description.**

**Dataset source: Macrotrends**

**Brief description: The dataset provides population estimates for the Coimbatore, India metro area from 1950 to 2035. The estimates are based on a variety of sources, including census data, vital statistics, and demographic surveys. The dataset also includes information on the population growth rate.**

**How the dataset was used: I used the dataset to estimate the population of Coimbatore, India in 2023. I did this by taking the population estimate for 2022 and adding the population growth rate for 2023.**

**Limitations of the dataset: The dataset estimates are based on a variety of sources, and the accuracy of the estimates may vary depending on the quality of the source data. Additionally, the population growth rate can vary from year to year, so the estimates for future years may not be accurate.**



# Compile all the code files, including the data preprocessing, and association analysis code.

Here is an example of a spec file

```
# Coimbatore Population Estimation spec# -*- mode: python -*-
```

```
import pyinstaller.__main__

pyinstaller.__main__.run([
    # Python files to compile
    'coimbatore_population_estimation.py',
    'data_preprocessing.py',
    'association_analysis.py',
    # Dependencies to include
    '--onefile',
    '-w',
    '-d',
    'dist',
])
```

Compile the code:

```
pyinstaller coimbatore-population-estimation.spec
```

This will create a new executable file called coimbatore-population-estimation.exe in the dist directory.

Run the compiled executable file:

```
./dist/coimbatore-population-estimation.exe
```

This will run the data preprocessing, association analysis, and population estimation code.

Note: If you are using a Mac or Linux computer, you may need to give the executable file execute permissions:

```
chmod +x dist/coimbatore-population-estimation.exe
```



A close-up, slightly blurred photograph of a person's hands typing on a black gaming keyboard with red backlit keys. The keyboard is on a dark desk. In the background, a computer mouse and some cables are visible. The overall lighting is dim, with the red light from the keyboard providing a focal point. A large, semi-transparent red bar is positioned below the text.

**THANK YOU**