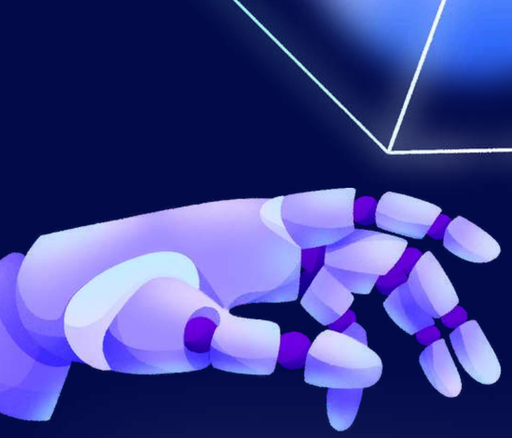
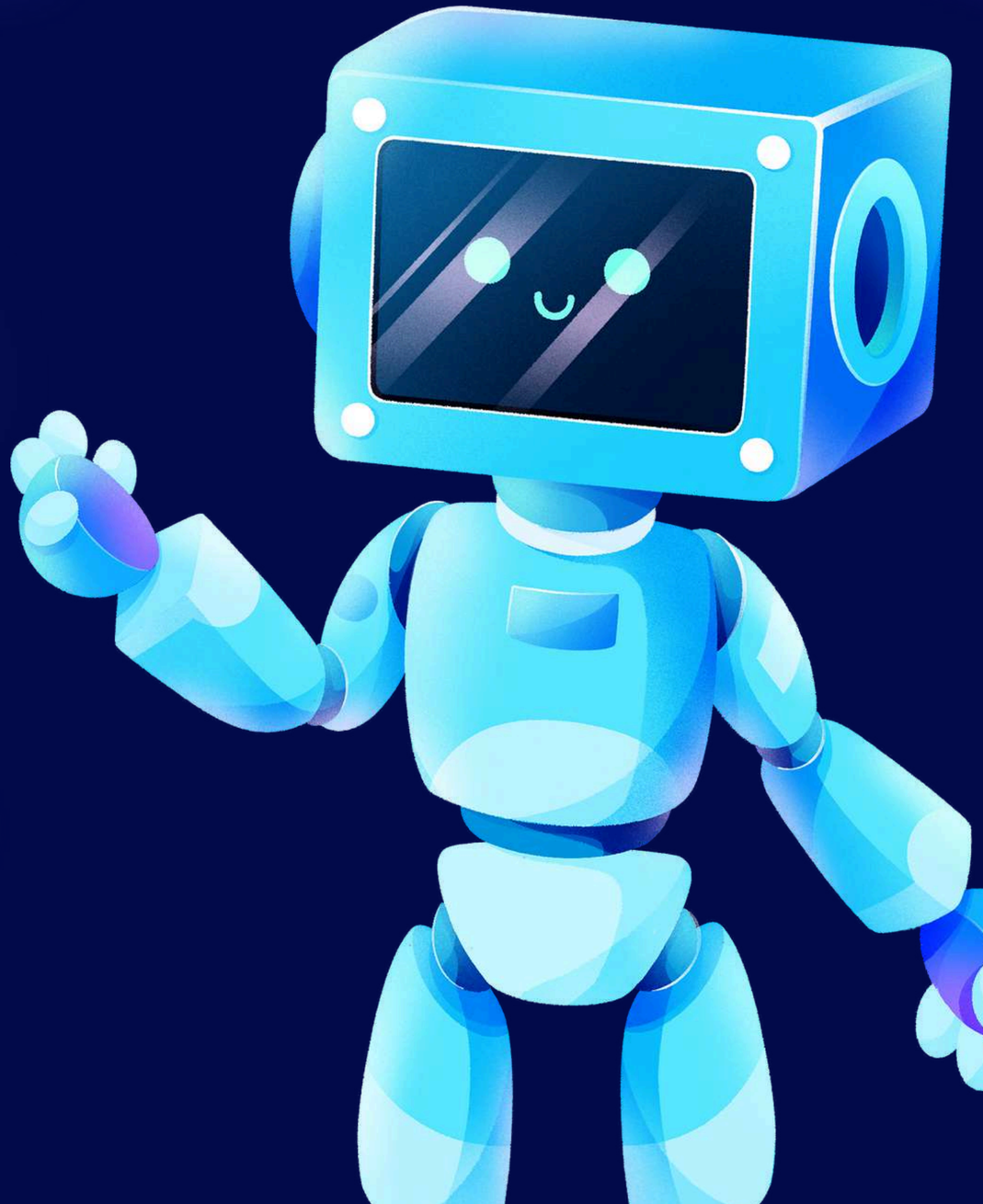
SKY_HAWKS SYSTEM



MARKET BASKET ANALYSIS USING PYTHON

PROJECT CREATED BY: D.AJAY KUMAR(192211089)
J.JOEL ANDREW(192211111)
M. RAM JEYANTH(192211128)
S.HARI HARAN(192211110)



INTRODUCTION

Market Basket Analysis is a powerful data mining technique used to uncover patterns and associations within transactional data. In retail, it helps in understanding the purchase behavior of customers by identifying products that are frequently bought together. Utilizing Python, this analysis becomes more accessible and efficient through libraries such as pandas, NumPy, and mlxtend. Python's robust ecosystem allows for the seamless handling of large datasets, the application of association rule algorithms like Apriori, and the visualization of results. This enables businesses to optimize product placement, improve marketing strategies, and ultimately enhance customer satisfaction and sales performance.

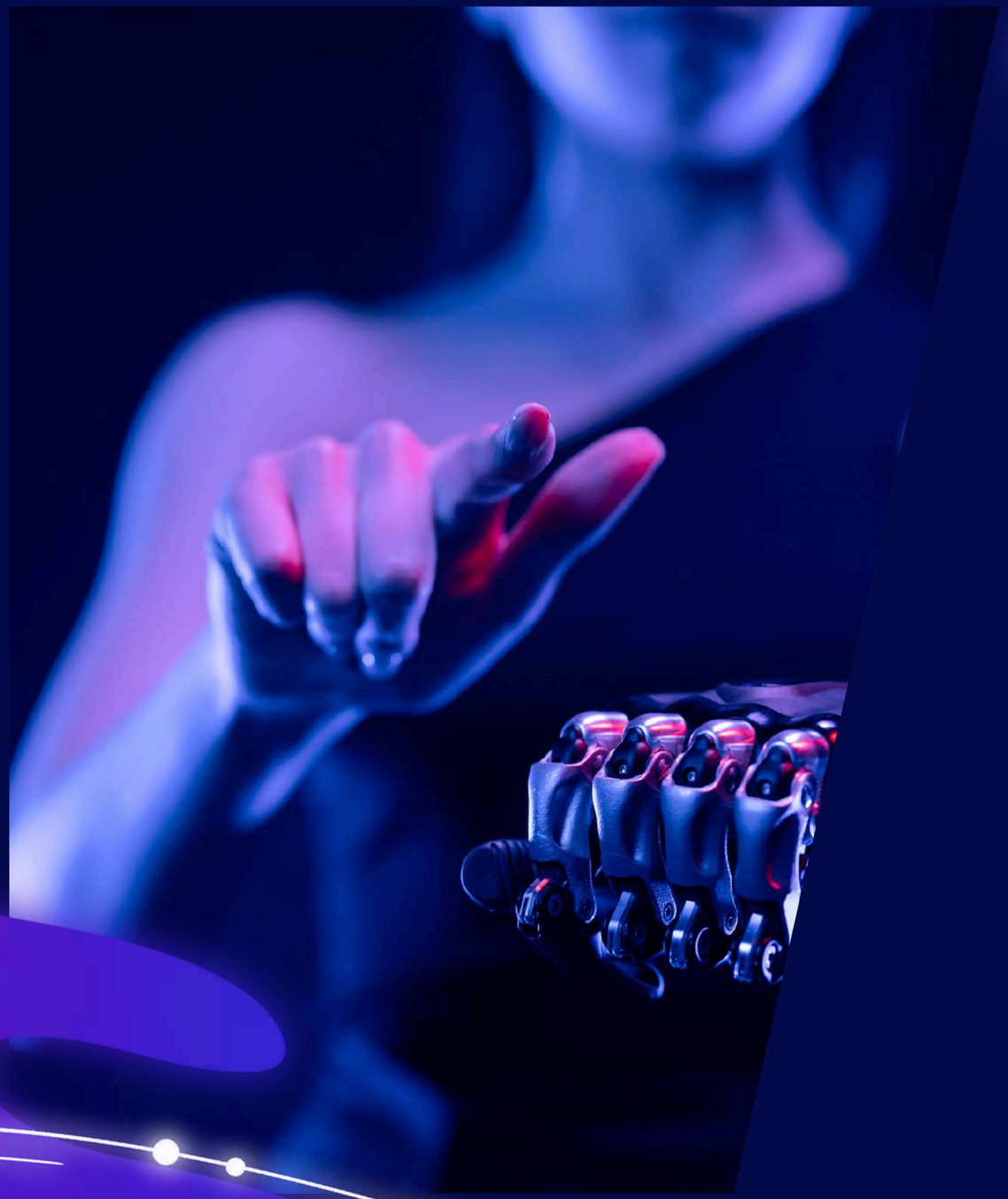
OBJECTIVE

THE ART OF EXPLORATION

The objective of Market Basket Analysis using Python is to identify and analyze patterns of co-occurrence in transactional data to uncover associations between products. By applying data mining techniques and leveraging Python's powerful libraries, the goal is to generate actionable insights that can help businesses optimize inventory management, enhance marketing strategies, and improve customer experience.

EXISTING SYSTEM

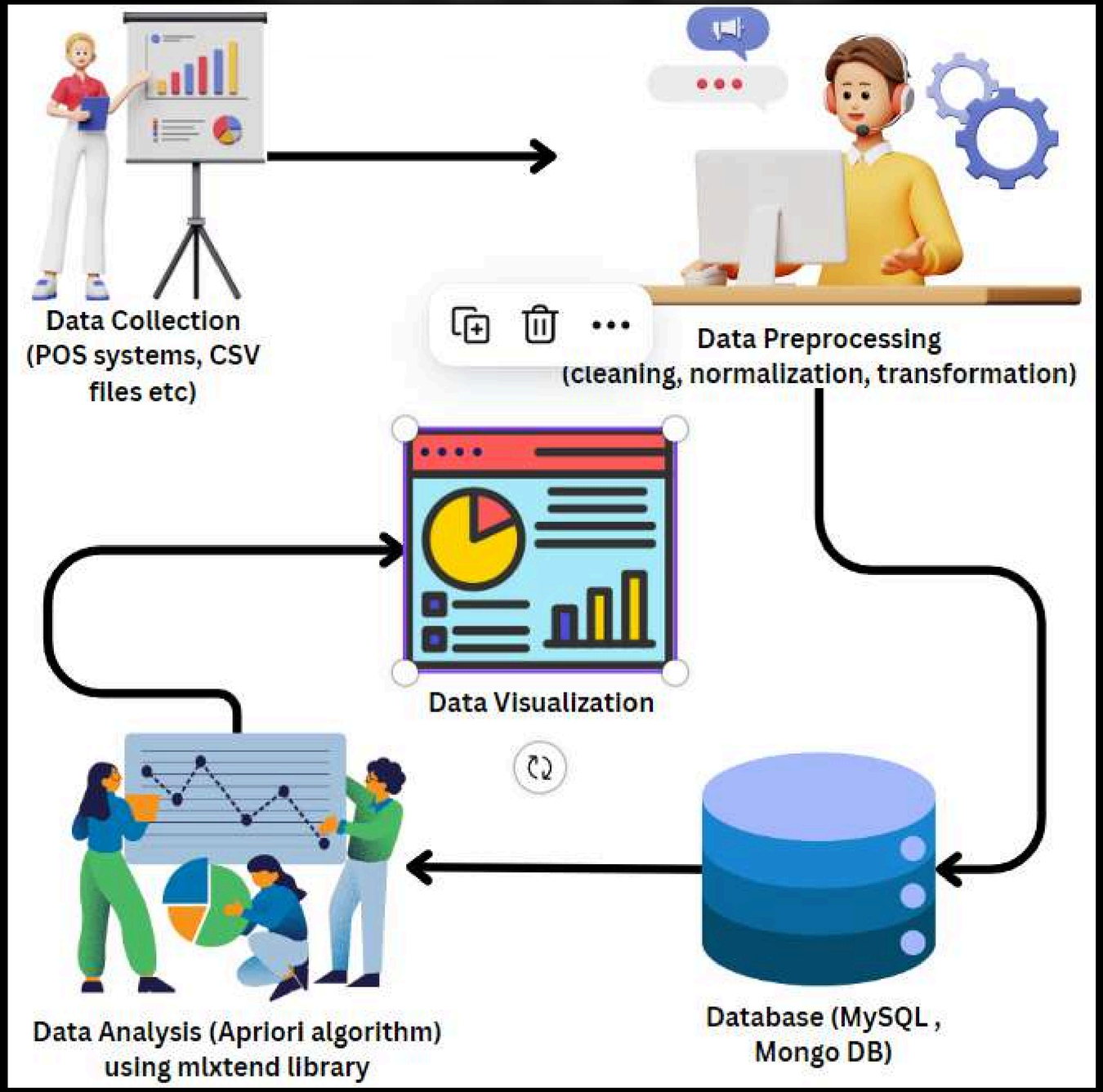
In the existing system, Market Basket Analysis is often performed using traditional data mining tools and manual analysis techniques. These methods can be time-consuming, prone to errors, and limited in handling large volumes of transactional data. Typically, businesses rely on basic statistical methods or specialized software that may not fully integrate with modern data processing workflows. This system lacks the flexibility and scalability needed to quickly adapt to changing market trends

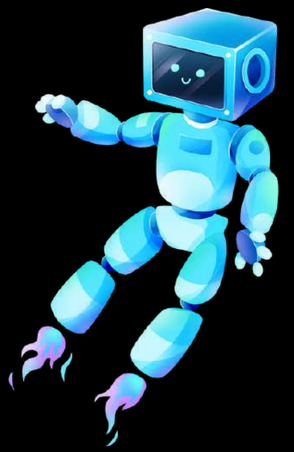


PROPOSED SYSTEM

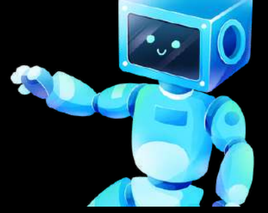
The proposed system utilizes Python's advanced data analytics libraries to enhance Market Basket Analysis. By leveraging tools like pandas, NumPy, and mlxtend, the system can efficiently process large datasets and uncover valuable associations between products. The integration of machine learning algorithms, such as Apriori, automates the discovery of frequent itemsets and actionable insights. This scalable and flexible solution enables businesses to quickly adapt to market trends, optimize product placement, and improve marketing strategies.

ARCHITECTURE DESIGN





MATHEMATICAL PROOF (APRIORI ALGORITHM)



mba_1.py - C:\Users\ecute\AppData\Local\Programs\Python\Python312\mba_1.py (3.12.0)

File Edit Format Run Options Window Help

```python
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
import tkinter as tk
from tkinter import ttk, messagebox
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import seaborn as sns
import threading

# Define the file path
FILE_PATH = r'C:\Users\ecute\OneDrive\Desktop\supermarket_data.csv'

# Function to load data and initialize variables
def initialize_data():
    global df, transactions, df_transformed, rules
    try:
        print("Loading data...")
        status_label.config(text="Loading data...")
        # Load the data
        df = pd.read_csv(FILE_PATH, on_bad_lines='skip')
        df = df.dropna(subset=['Description'])
        transactions = df.groupby('Invoice')['Description'].apply(list).values

        # Transform the data
        print("Transforming data...")
        status_label.config(text="Transforming data...")
        te = TransactionEncoder()
        te_ary = te.fit(transactions).transform(transactions)
        df_transformed = pd.DataFrame(te_ary, columns=te.columns_)

        # Perform Market Basket Analysis
        print("Performing market basket analysis...")
        status_label.config(text="Performing market basket analysis...")
        frequent_itemsets = apriori(df_transformed, min_support=0.01, use_colnames=True)
        rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)

        # Display rules
        print("Displaying rules...")
        status_label.config(text="Displaying rules...")
        display_rules()
        status_label.config(text="Data loaded successfully!")
    except Exception as e:
        print(f"An error occurred during initialization: {e}")
        messagebox.showerror("Error", f"An error occurred: {e}")
        status_label.config(text="Error loading data")
```

Definition 1: Suppose $T=\{T_1, T_2, \dots, T_m\},(m\geq 1)$ is a set of transactions, $_{Ti}= \{I_1, _{I2}, \dots, I_n\},(n\geq 1)$ is the set of items, and k-itemset = $\{i_1, i_2, \dots, i_k\},(k\geq 1)$ is also the set of k items, and k-itemset $\subseteq$ I.

Definition 2: Suppose σ (itemset), is the support count of itemset or the frequency of occurrence of an itemset in transactions.

Definition 3: Suppose C_k is the candidate itemset of size k, and L_k is the frequent itemset of size k.

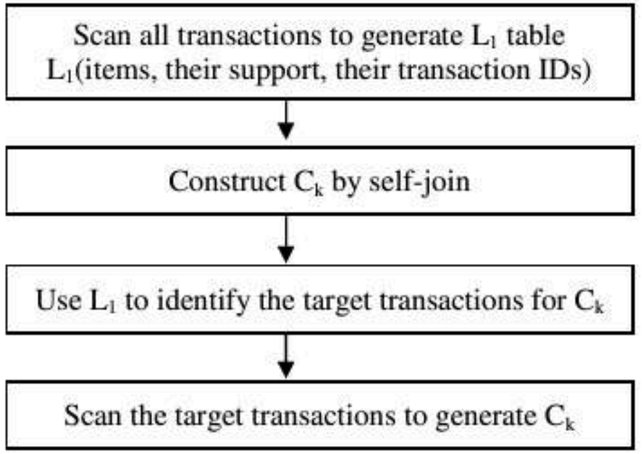
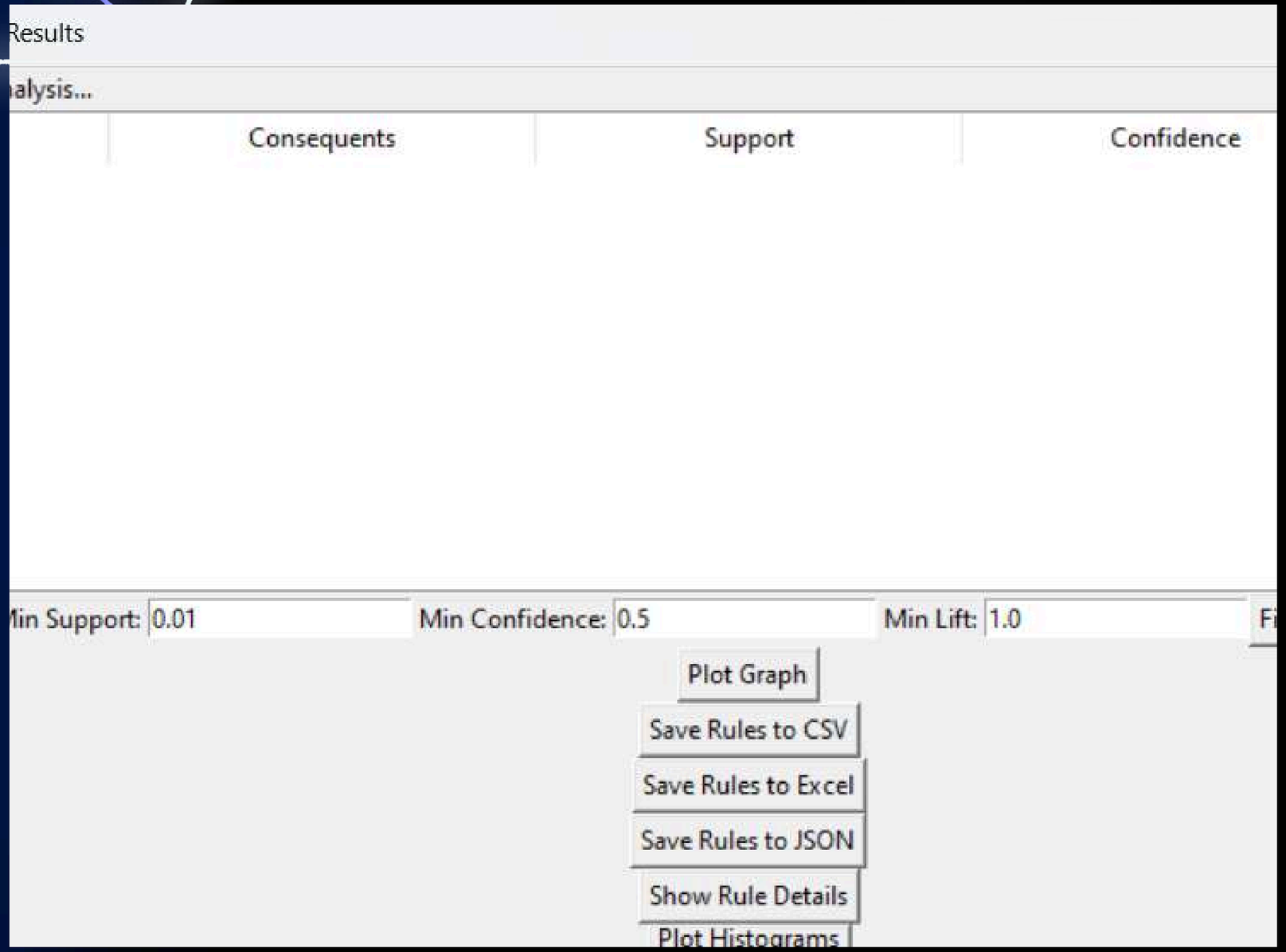


Figure 1: Steps for C_k generation

Confidence of rule $a \rightarrow b$: the fraction of times itemset b is purchased when itemset a is purchased.

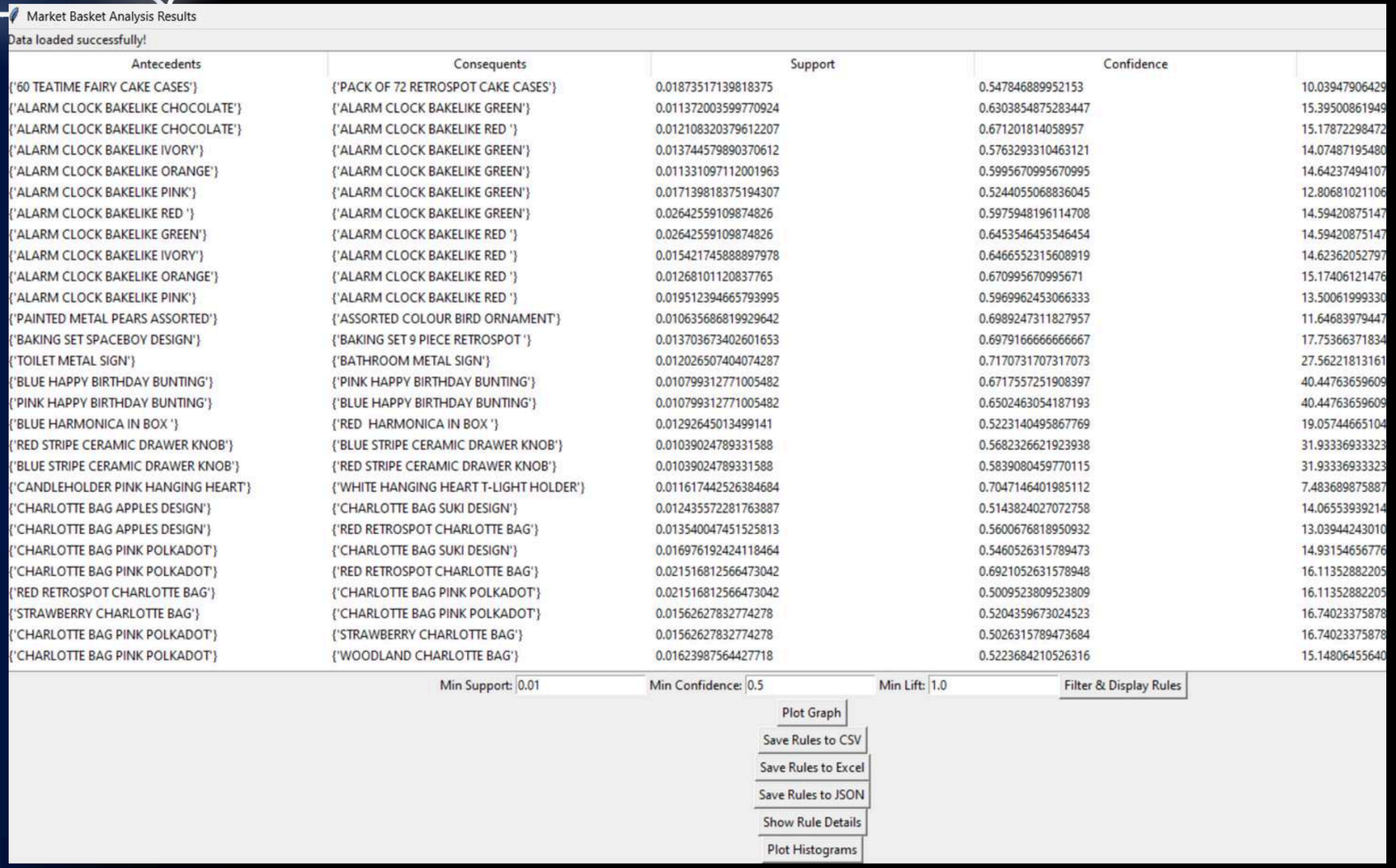
$$\begin{aligned} \text{Conf}(a \rightarrow b) &= \frac{\text{Supp}(a \cup b)}{\text{Supp}(a)} = \frac{\#\text{times } a \text{ and } b \text{ are purchased}}{\#\text{times } a \text{ is purchased}} \\ &= \hat{P}(b|a). \end{aligned}$$

MODULE 01



In this Market Basket Analysis project, the GUI output is generated using Python's Tkinter library, providing a user-friendly interface to interact with the analysis results. The main window displays the association rules derived from the dataset, showing columns for antecedents, consequents, support, confidence, and lift. Users can filter these rules by specifying minimum values for support, confidence, and lift in the provided input fields.

MODULE 02



Antecedents	Consequents	Support	Confidence	
{'60 TEATIME FAIRY CAKE CASES'}	{'PACK OF 72 RETROSPOT CAKE CASES'}	0.01873517139818375	0.547846889952153	10.039479064291
{'ALARM CLOCK BAKELIKE CHOCOLATE'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.011372003599770924	0.6303854875283447	15.395008619491
{'ALARM CLOCK BAKELIKE CHOCOLATE'}	{'ALARM CLOCK BAKELIKE RED '}	0.012108320379612207	0.671201814058957	15.178722984721
{'ALARM CLOCK BAKELIKE IVORY'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.013744579890370612	0.5763293310463121	14.074871954801
{'ALARM CLOCK BAKELIKE ORANGE'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.011331097112001963	0.5995670995670995	14.642374941071
{'ALARM CLOCK BAKELIKE PINK'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.017139818375194307	0.5244055068836045	12.806810211061
{'ALARM CLOCK BAKELIKE RED '}	{'ALARM CLOCK BAKELIKE GREEN'}	0.02642559109874826	0.5975948196114708	14.594208751471
{'ALARM CLOCK BAKELIKE GREEN'}	{'ALARM CLOCK BAKELIKE RED '}	0.02642559109874826	0.6453546453546454	14.594208751471
{'ALARM CLOCK BAKELIKE IVORY'}	{'ALARM CLOCK BAKELIKE RED '}	0.015421745888897978	0.6466552315608919	14.623620527971
{'ALARM CLOCK BAKELIKE ORANGE'}	{'ALARM CLOCK BAKELIKE RED '}	0.01268101120837765	0.670995670995671	15.174061214761
{'ALARM CLOCK BAKELIKE PINK'}	{'ALARM CLOCK BAKELIKE RED '}	0.019512394665793995	0.5969962453066333	13.500619993301
{'PAINTED METAL PEARS ASSORTED'}	{'ASSORTED COLOUR BIRD ORNAMENT'}	0.010635686819929642	0.6989247311827957	11.646839794471
{'BAKING SET SPACEBOY DESIGN'}	{'BAKING SET 9 PIECE RETROSPOT '}	0.013703673402601653	0.6979166666666667	17.753663718341
{'TOILET METAL SIGN'}	{'BATHROOM METAL SIGN'}	0.012026507404074287	0.7170731707317073	27.562218131611
{'BLUE HAPPY BIRTHDAY BUNTING'}	{'PINK HAPPY BIRTHDAY BUNTING'}	0.010799312771005482	0.6717557251908397	40.447636596091
{'PINK HAPPY BIRTHDAY BUNTING'}	{'BLUE HAPPY BIRTHDAY BUNTING'}	0.010799312771005482	0.6502463054187193	40.447636596091
{'BLUE HARMONICA IN BOX '}	{'RED HARMONICA IN BOX '}	0.01292645013499141	0.5223140495867769	19.057446651041
{'RED STRIPE CERAMIC DRAWER KNOB'}	{'BLUE STRIPE CERAMIC DRAWER KNOB'}	0.01039024789331588	0.5682326621923938	31.933369333231
{'BLUE STRIPE CERAMIC DRAWER KNOB'}	{'RED STRIPE CERAMIC DRAWER KNOB'}	0.01039024789331588	0.5839080459770115	31.933369333231
{'CANDLEHOLDER PINK HANGING HEART'}	{'WHITE HANGING HEART T-LIGHT HOLDER'}	0.011617442526384684	0.7047146401985112	7.4836898758871
{'CHARLOTTE BAG APPLES DESIGN'}	{'CHARLOTTE BAG SUKI DESIGN'}	0.012435572281763887	0.5143824027072758	14.065539392141
{'CHARLOTTE BAG APPLES DESIGN'}	{'RED RETROSPOT CHARLOTTE BAG'}	0.013540074515258 13	0.5600676818950932	13.039442430101
{'CHARLOTTE BAG PINK POLKADOT'}	{'CHARLOTTE BAG SUKI DESIGN'}	0.016976192424118464	0.5460526315789473	14.931546567761
{'CHARLOTTE BAG PINK POLKADOT'}	{'RED RETROSPOT CHARLOTTE BAG'}	0.021516812566473042	0.6921052631578948	16.113528822051
{'RED RETROSPOT CHARLOTTE BAG'}	{'CHARLOTTE BAG PINK POLKADOT'}	0.021516812566473042	0.5009523809523809	16.113528822051
{'STRAWBERRY CHARLOTTE BAG'}	{'CHARLOTTE BAG PINK POLKADOT'}	0.01562627832774278	0.5204359673024523	16.740233758781
{'CHARLOTTE BAG PINK POLKADOT'}	{'STRAWBERRY CHARLOTTE BAG'}	0.01562627832774278	0.5026315789473684	16.740233758781
{'CHARLOTTE BAG PINK POLKADOT'}	{'WOODLAND CHARLOTTE BAG'}	0.01623987564427718	0.5223684210526316	15.148064556401

In this Market Basket Analysis project, the GUI output generated using Python's Tkinter library provides an interactive and user-friendly interface for loading and analyzing transaction data. Upon successfully loading the dataset, the GUI displays the dataset in a structured format, showcasing various transaction details such as Invoice, StockCode, Description, Quantity, InvoiceDate, Price, Customer ID, and Country.

MODULE 03

nalysis Results

sfully!

Antecedents	Consequents	Support	Confidence
' CAKE CASES'}	{'PACK OF 72 RETROSPOT CAKE CASES'}	0.01873517139818375	0.547846889952153
AKELIKE CHOCOLATE'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.011372003599770924	0.6303854875283447
AKELIKE CHOCOLATE'}	{'ALARM CLOCK BAKELIKE RED '}	0.012108320379612207	0.671201814058957
AKELIKE IVORY'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.013744579890370612	0.5763293310463121
AKELIKE ORANGE'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.011331097112001963	0.5995670995670995
AKELIKE PINK'}	{'ALARM CLOCK BAKELIKE GREEN'}	0.017139818375194307	0.5244055068836045
AKELIKE RED '}	{'ALARM CLOCK BAKELIKE GREEN'}	0.02642559109874826	0.5975948196114708
AKELIKE GREEN'}	{'ALARM CLOCK BAKELIKE RED '}	0.02642559109874826	0.6453546453546454
AKELIKE IVORY'}	{'ALARM CLOCK BAKELIKE RED '}	0.015421745888897978	0.6466552315608919
AKELIKE ORANGE'}	{'ALARM CLOCK BAKELIKE RED '}	0.01268101120837765	0.670995670995671

Min Support: 0.01 Min Confidence: 0.5 Min Lift: 1.0 Filter & Display Rules

Plot Graph

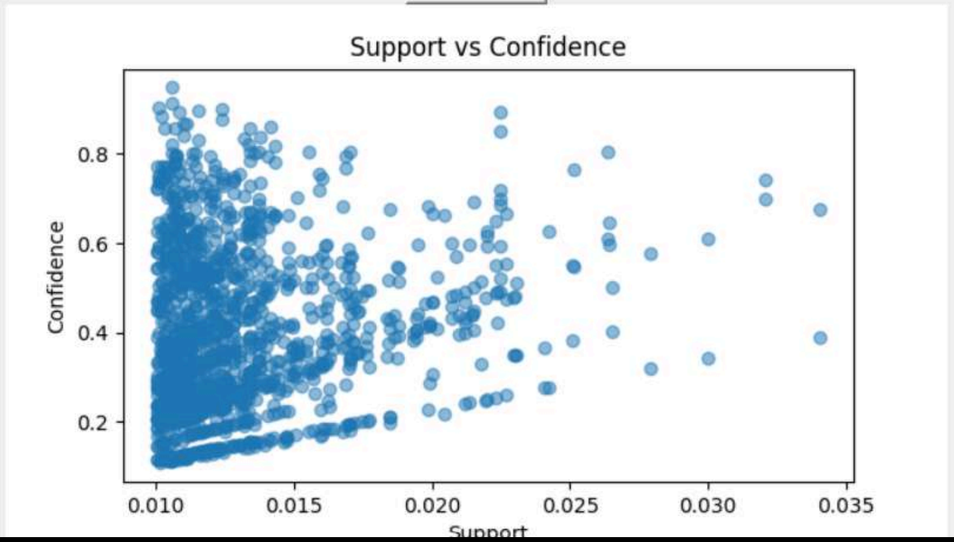
Save Rules to CSV

Save Rules to Excel

Save Rules to JSON

Show Rule Details

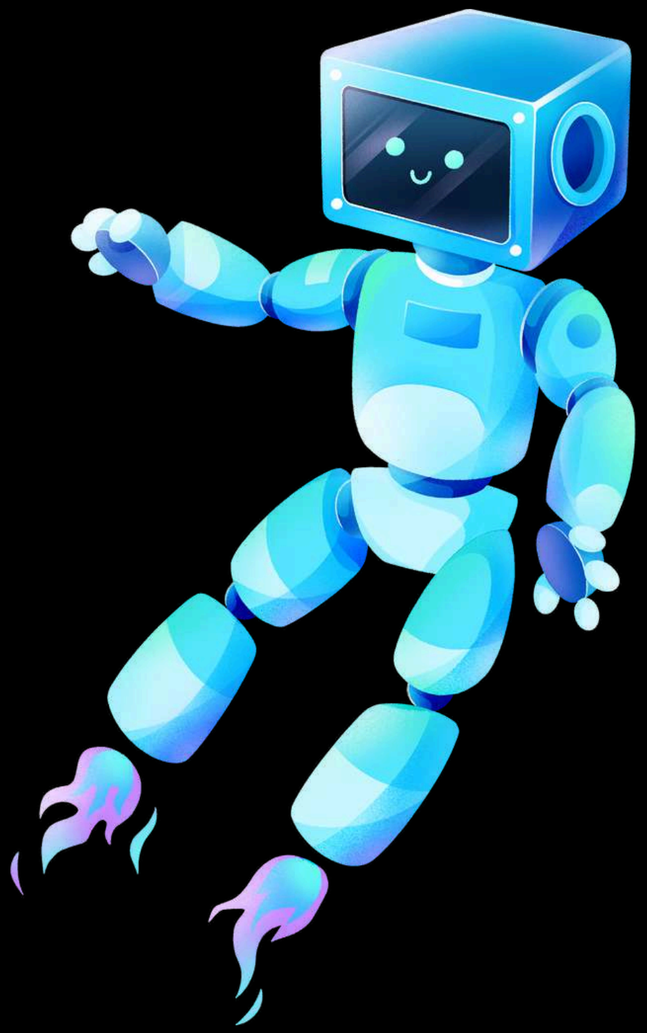
Plot Histograms



In this Market Basket Analysis project, the Support vs. Confidence graph is a crucial visualization that helps in understanding the strength and reliability of the association rules generated from the transaction data. The graph plots support values on the x-axis and confidence values on the y-axis for each rule. Each point represents an association rule, with its position indicating how frequently the rule appears in the dataset (support) and the reliability of the rule (confidence).

CONCLUSION

In conclusion, this Market Basket Analysis project effectively demonstrates the power of association rule mining in uncovering meaningful patterns within transaction data. By utilizing Python's robust data analysis libraries and providing a user-friendly GUI, the project allows users to easily load datasets, generate insightful association rules, and visualize these rules through graphs and tables. The interactive interface and advanced filtering capabilities enable a deeper understanding of customer purchasing behavior, aiding businesses in making data-driven decisions to optimize product placement, marketing strategies, and inventory management.



REFERENCES

[1] Halim, Octavia, and Alianto. 2019. Designing Facility Layout of an Amusement Arcade using Market Basket Analysis. Procedia Computer Science, Vol 161, Page 623–629. ([1] Hasan, KM Azharul, Amit Mondal, and Amit Saha. "A context free grammar and its predictive parser for bangla grammar recognition." In 2010 13th International Conference on Computer and Information Technology (ICCIT), pp. 87-91. IEEE, 2010. http://dx.doi.org/10.1007/978-981-16-0586-4_26

[2] Braune, Beatrix, Stephan Diehl, Andreas Kerren, and Reinhard Wilhelm. "Animation of the generation and computation of finite automata for learning software." In International Workshop on Implementing Automata, pp. 39-47. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. https://doi.org/10.1007/3-540-45526-4_4

THANK YOU!

TEAM

D.AJAY KUMAR

J.JOEL ANDREW

M.RAM JEYANTH

S.HARI HARAN