



Project Evaluation

Problem Statement:

Predict/Recommend the customer segment for a new wine based on the wine shop dataset by performing dimensionality reduction using LDA

Group Ng:

Chetan Samrit
Darshana Nagpure
Deepak Suryawanshi
Suryadeep Khungar
Yash Dorshetwar



Table of Contents

- Introduction
- What is customer segmentation?
- What is LDA?
- Model Code
- API Overview
- General API Working
- Components used in provided API
- Process Workflow for API creation
- UI Workflow
- Technology Stack
- Frontend Development
- Backend Development
- Database Management



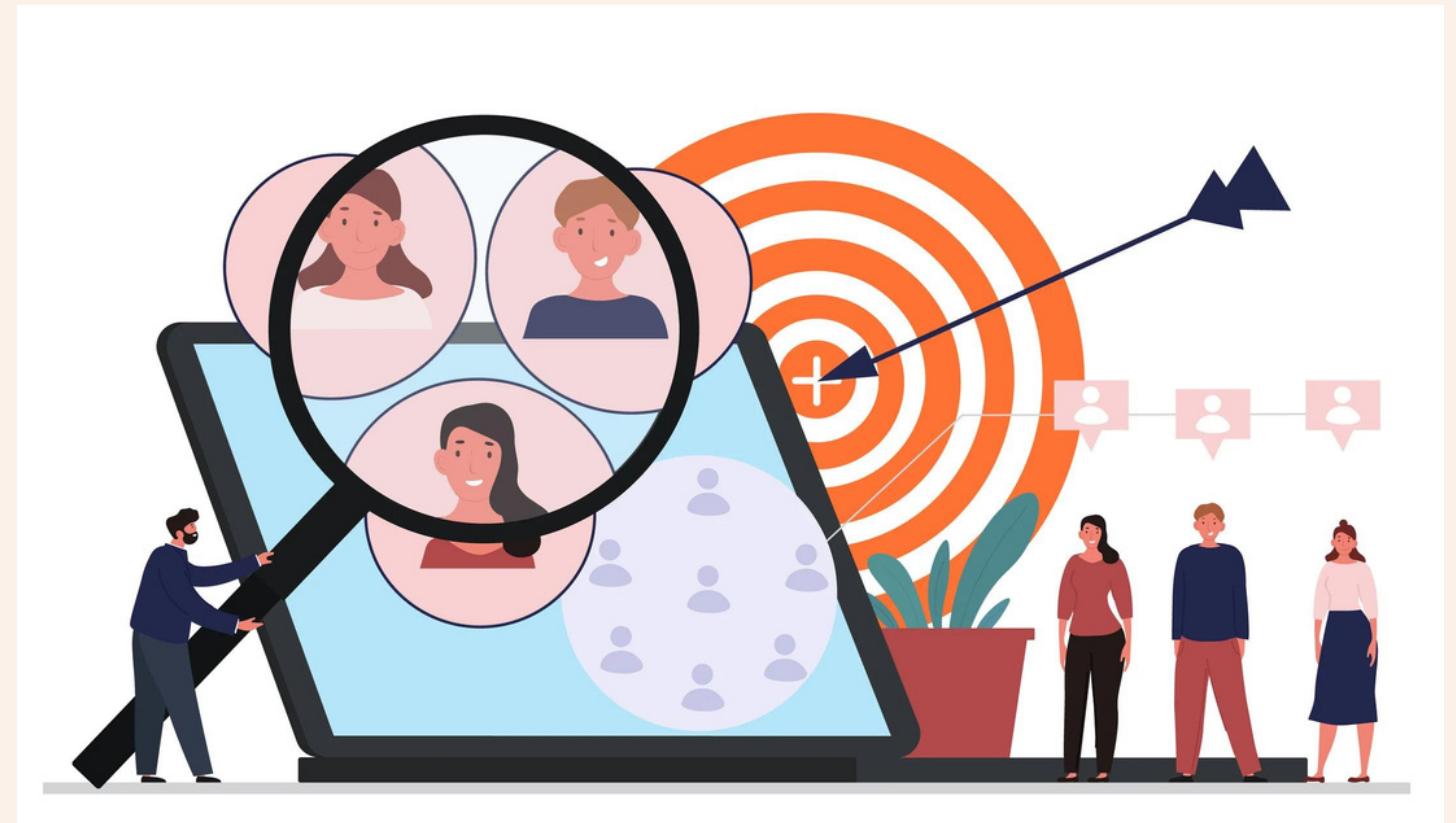
Introduction

- Why is customer segmentation so important in the wine industry?

Well, simply put, it allows wineries to better understand their customers and tailor their products and marketing efforts accordingly.

- What does our model provide?

Our model divides a customer base into groups of individuals that have similar needs according to wine ingredients.



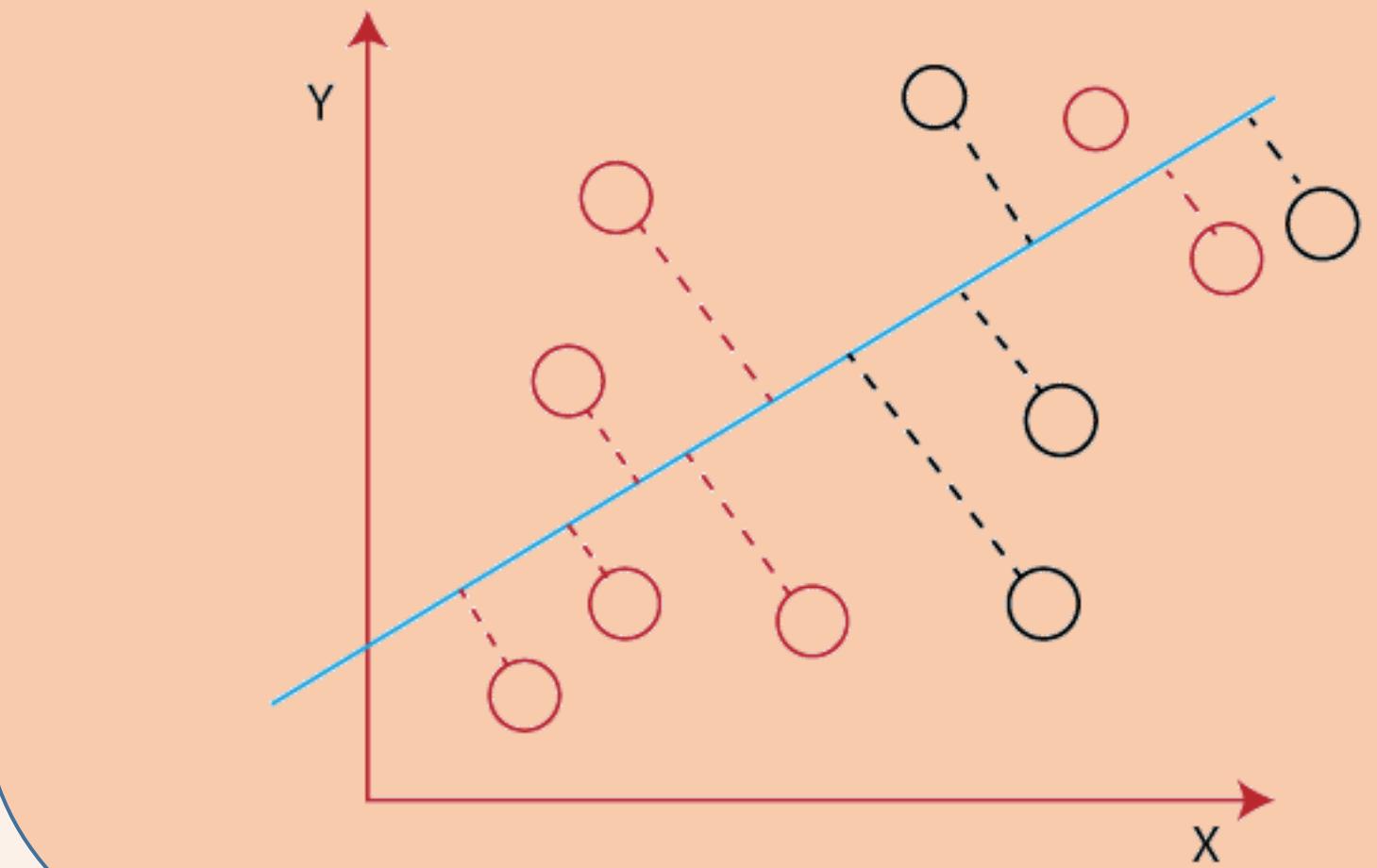
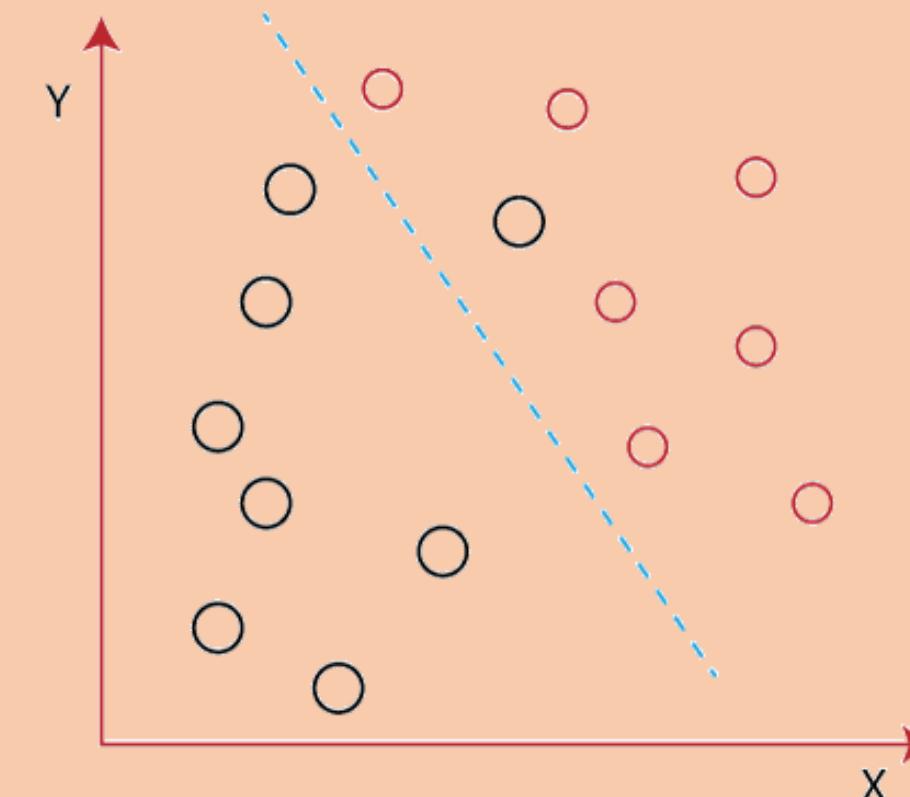
What is customer segmentation?



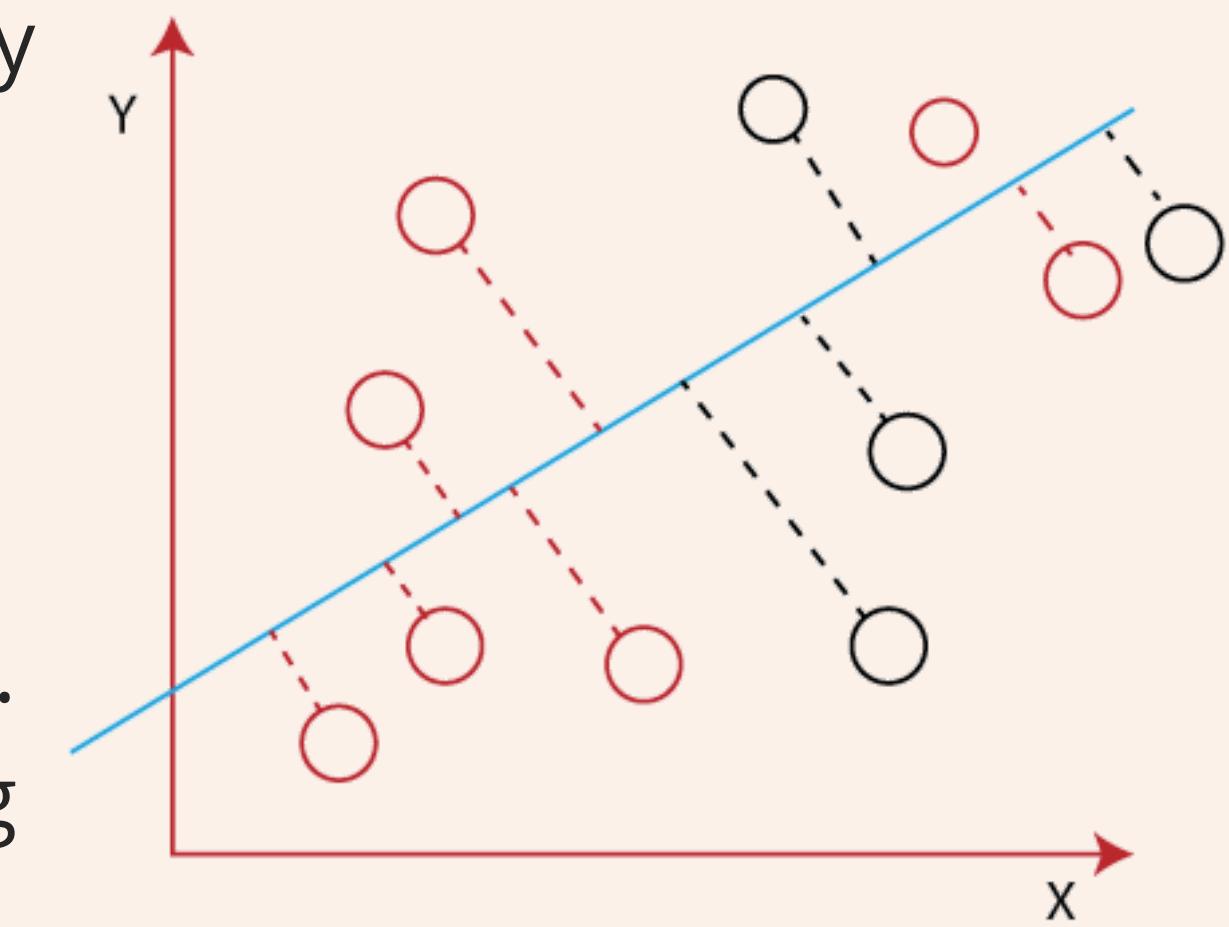
- Customer segmentation is the process of dividing a customer base into groups of individuals that have similar needs or characteristics. By doing this, businesses can tailor their marketing efforts and product offerings to specific segments, resulting in increased customer satisfaction and loyalty.
- In the wine industry, customer segmentation is especially important because wine consumers have diverse preferences and behaviors. For example, some customers may prefer red over white wine, while others may be more interested in organic or sustainably produced wines. By segmenting customers based on these preferences, wineries can develop targeted marketing campaigns and product offerings that meet the needs of each segment.

What is LDA?

- Linear Discriminant Analysis (LDA) is a supervised learning algorithm used for classification tasks in machine learning.
- It is a technique used to find a linear combination of features that best separates the classes in a dataset.
- It provides an informative low-dimensional view on the data, which is both useful for visualization and feature engineering.



- Linear Discriminant analysis is used as a dimensionality reduction technique in machine learning, using which we can easily transform a 2-D and 3-D graph into a 1-dimensional plane.
- LDA enables us to draw a straight line that can completely separate the two classes of the data points. LDA uses an X-Y axis to create a new axis by separating them using a straight line and projecting data onto a new axis.
- To create a new axis, Linear Discriminant Analysis uses the following criteria:
 1. It maximizes the distance between means of two classes.
 2. It minimizes the variance within the individual class.



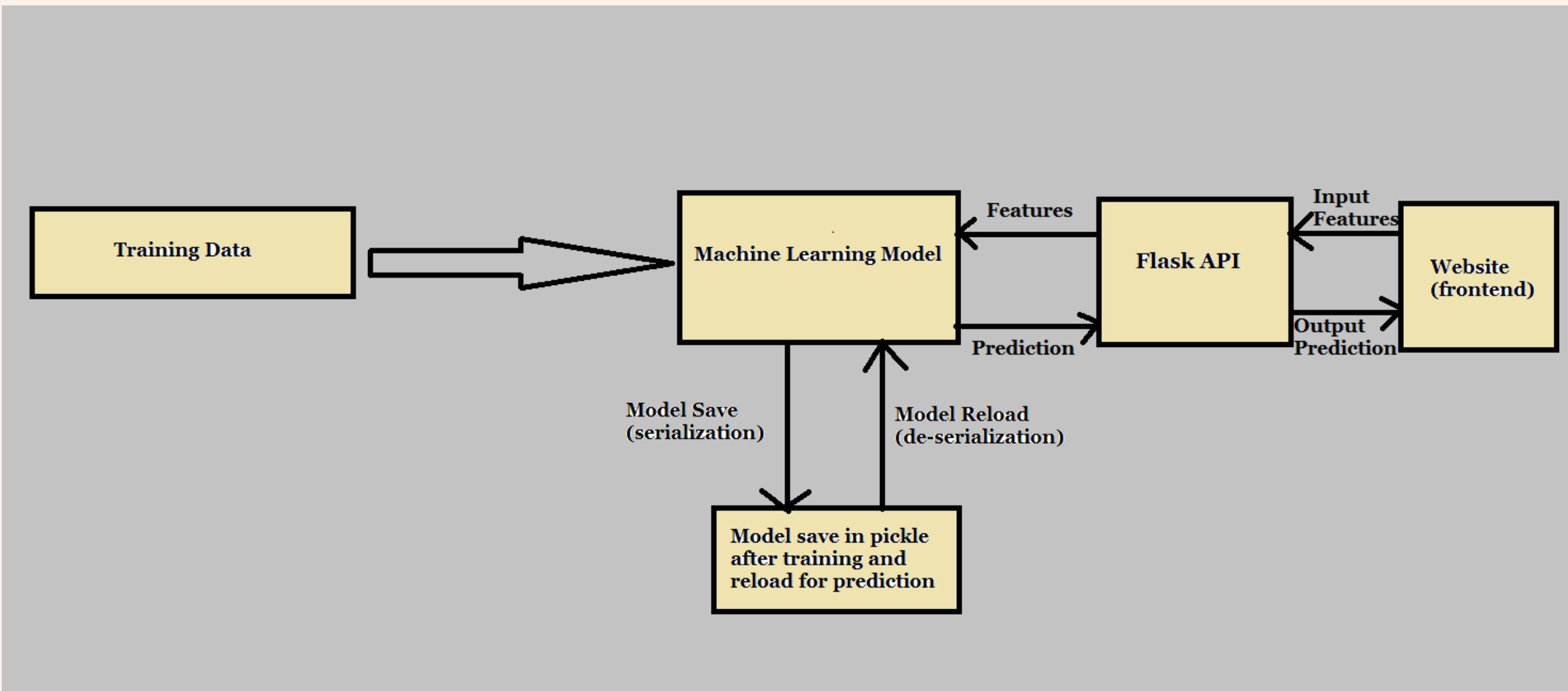
Model

1. Importing the necessary libraries
 2. Importing the dataset
 3. Correlation Matrix
 4. Correlation for Customer Segment column
 5. Percentage of customer groups in the dataset
 6. Pie chart visualization
 7. Scatter plot
 8. Splitting the data into training set and test set
 9. Feature Scaling
 10. Applying LDA
 11. Logistic Regression
 12. Predictions and Evaluations
 13. Visualizing the Training and Test set results
 14. Model Saving
- Overall, the code performs exploratory data analysis, applies LDA for dimensionality reduction, trains a logistic regression classifier, evaluates the model, and saves the trained model for future use.

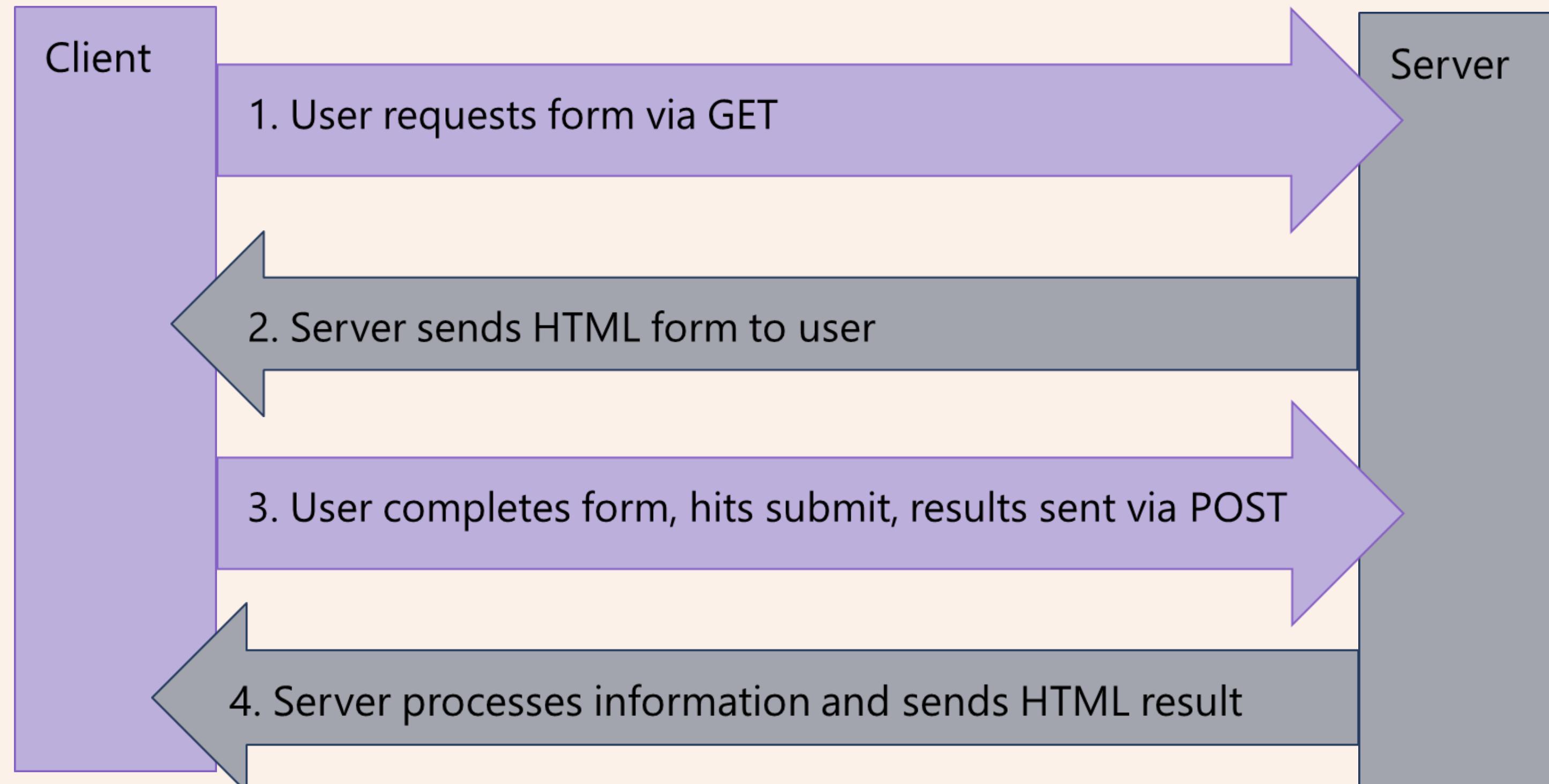
API Overview

Routes can be accessed in many ways, through what are known as methods or verbs

The most common verbs are GET and POST



General API Working



```
@app.route('/', methods=['GET'])
def index():
    # send form to the user
    return render_template('index.html')
```

```
@app.route('/', methods=['POST'])
def index_post():
    # process information, send results
    return render_template('results.html', data=some_data)
```

Components Used in provided API

- **Flask**
- **Request**
- **Jsonify**
- **Sci-kit Learn**
- **Numpy**
- **Pickle**

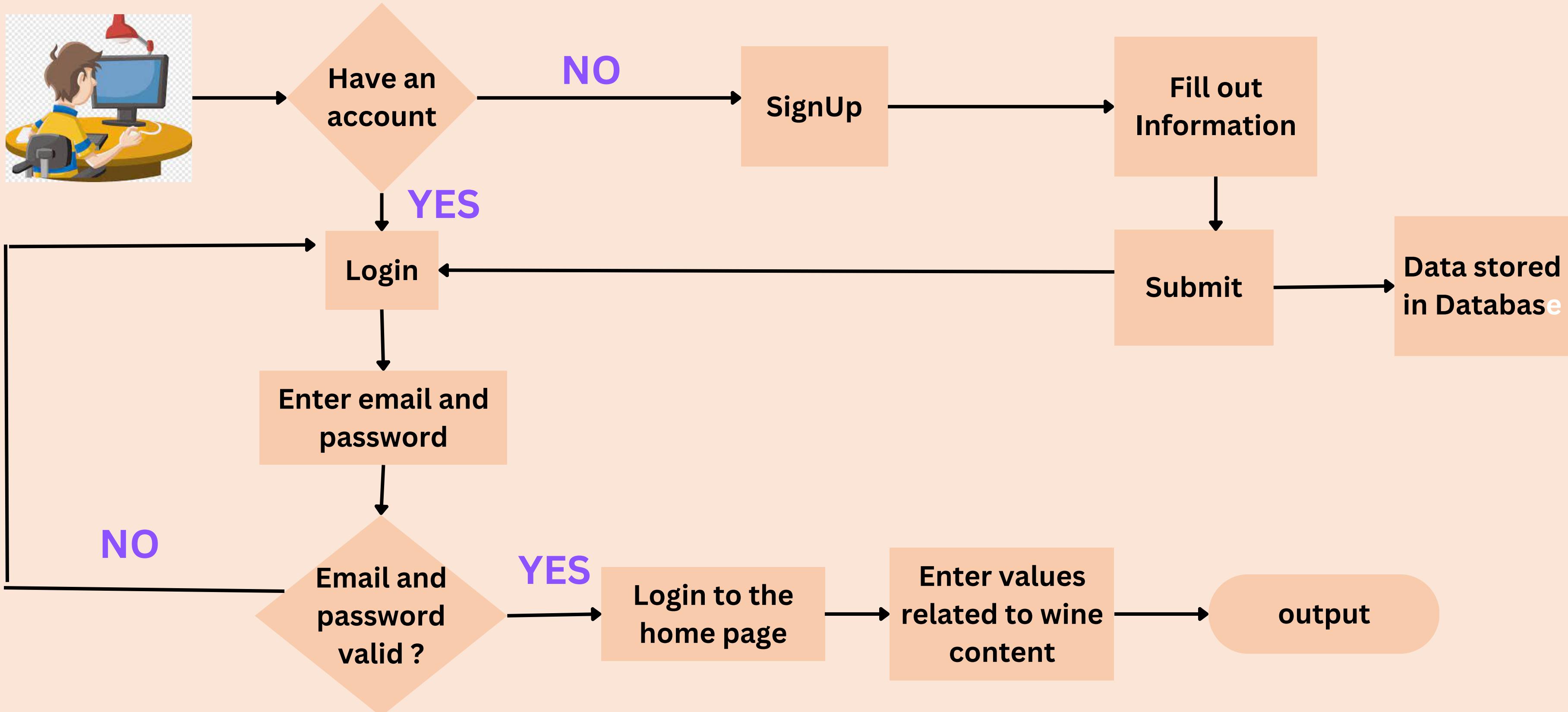
Process Workflow for API creation

- HTML Testing followed by React Testing
- Import necessary modules and libraries like Flask, numpy, pickle, etc.
- Create Flask app
- Load model files
- Define customer segments
- Define prediction endpoint(return render_template for html testing and /predict for react js testing)
- Extract data from request
- Extract input features from data
- Reshape input features array to match the expectations of the model using numpy

Process Workflow for API creation

- Apply scaling to the input features
- Apply LDA transformation
- Make the predictions using the model
- Return result as a list(for html) / as json output being passed (for react js)
- Run the flask app with debugging enabled

UI WORKFLOW



TECHNOLOGY STACK

Frontend

- React
- react router dom
- axios

Backend

- Express
- cors
- Node js

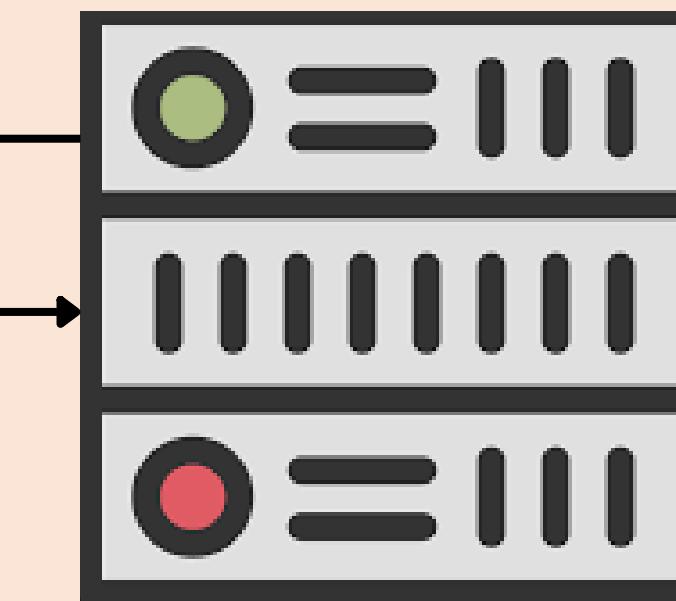
Database

- MySql

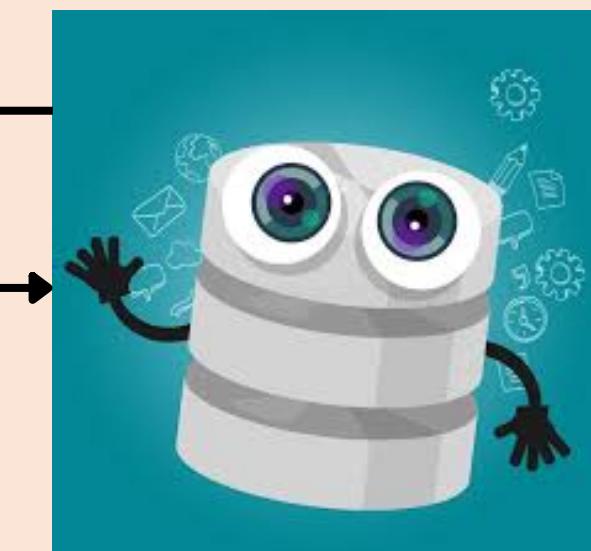
REACT



EXPRESS



MYSQL



Client Side /Frontend

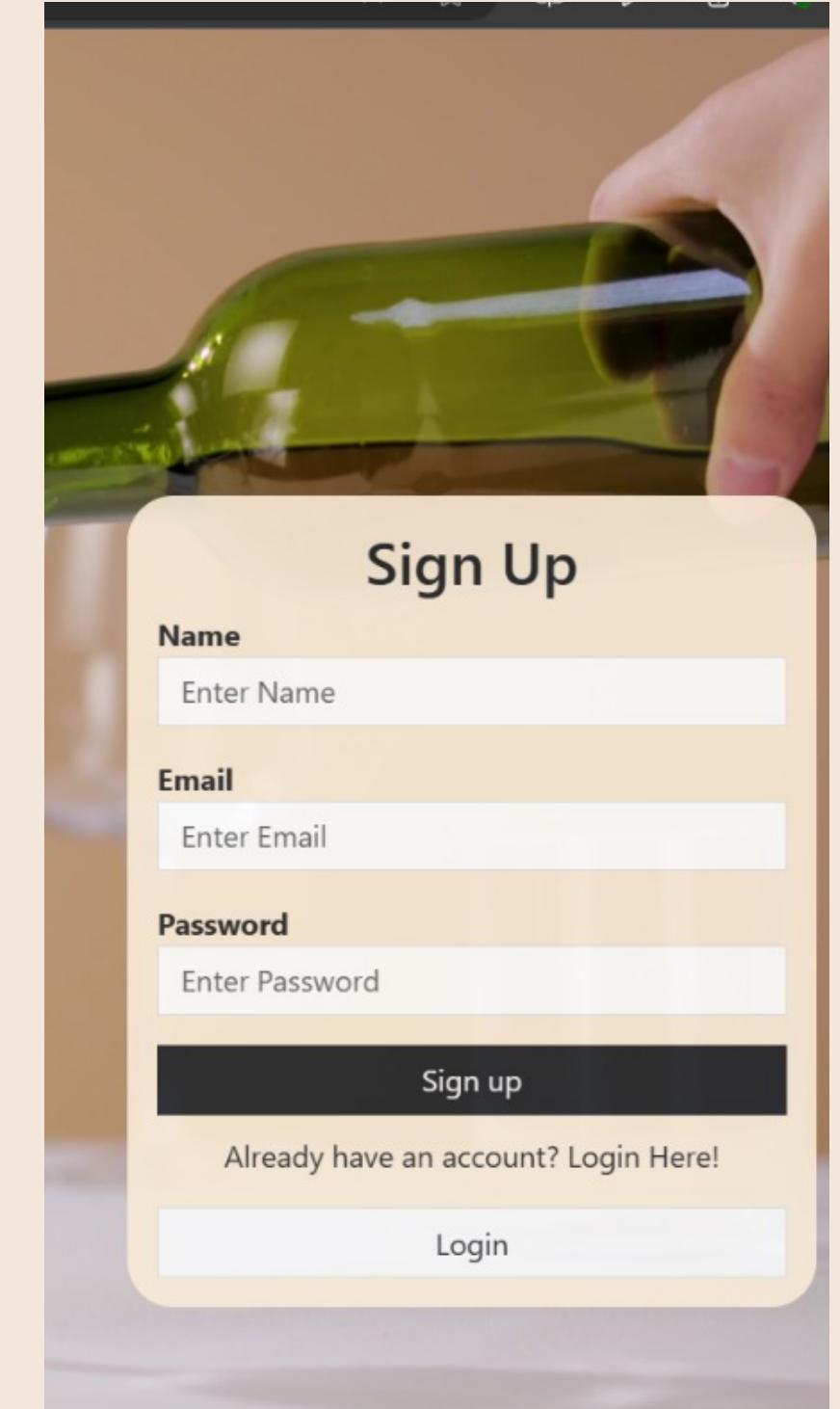
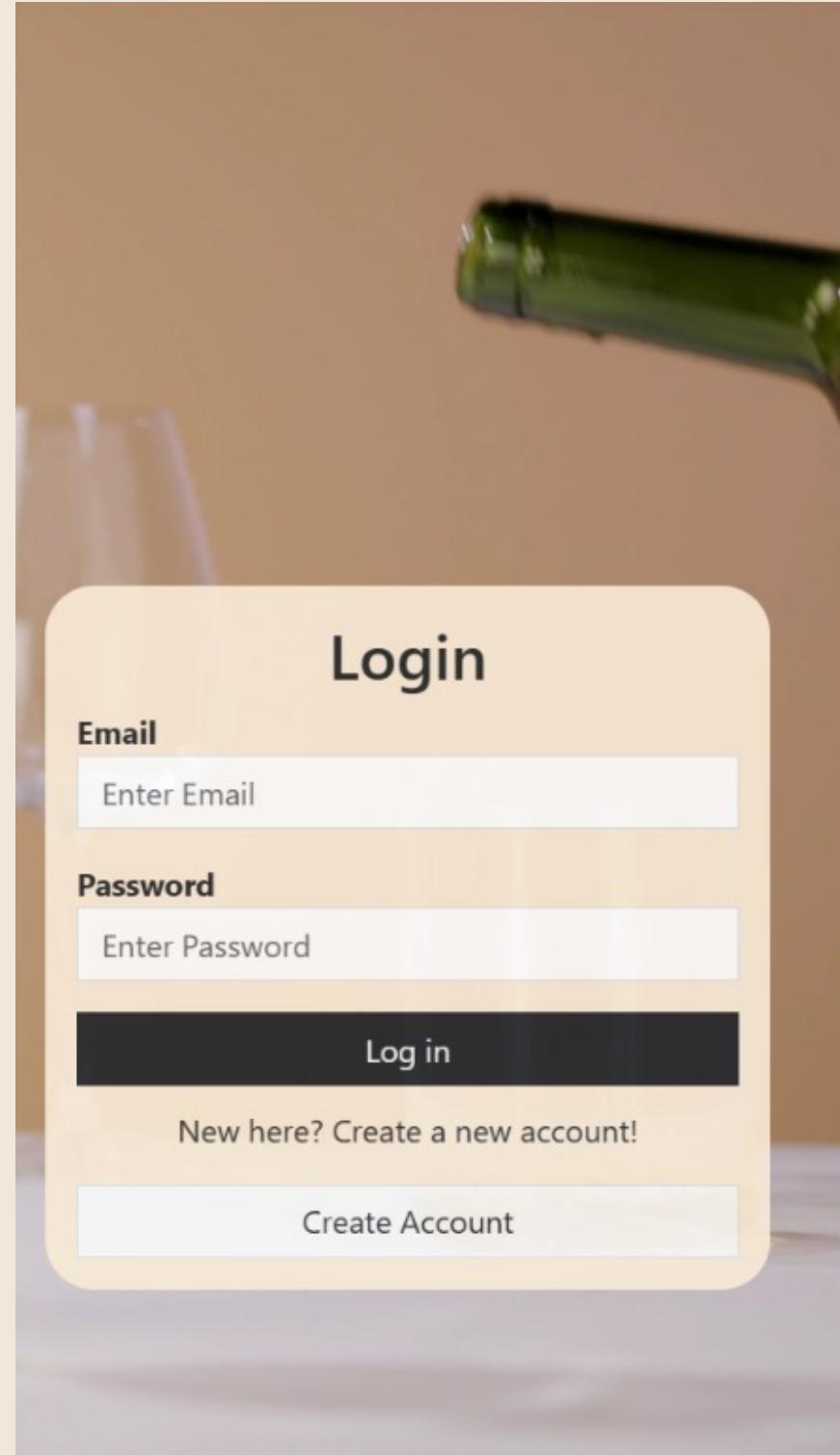
Server side / Backend

Database

FRONTEND DEVELOPMENT

Login/Signup

- React Component
- State Management
- Validation Function
- Input Handling
- Form submission
- Storing Data (Signup)
- Sending Request
- Handling Response
- Redirecting
- Rendering
- CSS Styling



HOME PAGE

- Home Component
- State Management
- Input Handling
- Prediction Request
- Server Request
- Displaying Prediction
- User Interface
- Api integration

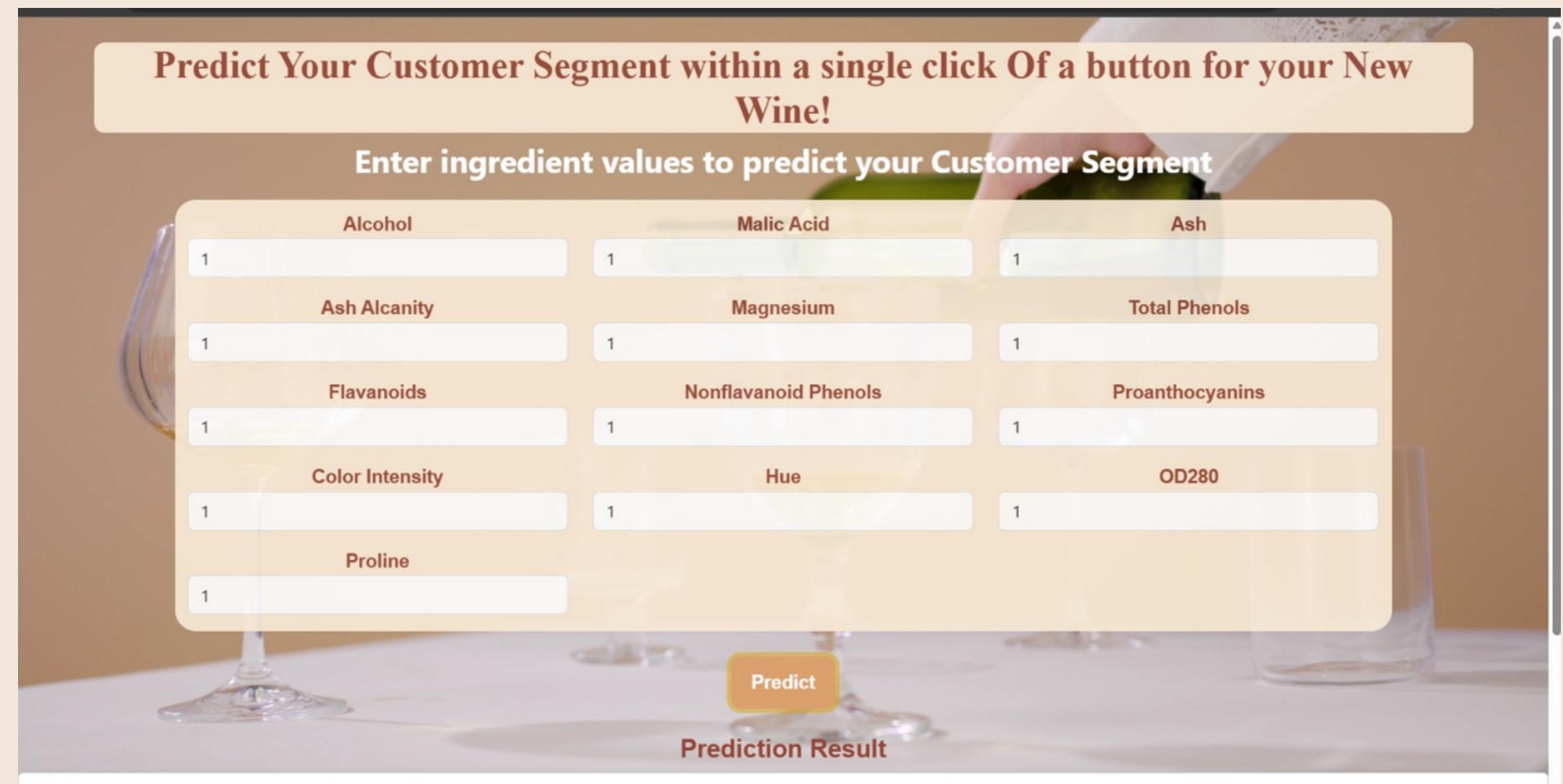
Predict Your Customer Segment within a single click Of a button for your New Wine!

Enter ingredient values to predict your Customer Segment

Alcohol	Malic Acid	Ash
1	1	1
Ash Alcanity	Magnesium	Total Phenols
1	1	1
Flavanoids	Nonflavanoid Phenols	Proanthocyanins
1	1	1
Color Intensity	Hue	OD280
1	1	1
Proline		
1		

Predict

Prediction Result



BACKEND DEVELOPMENT

- Importing Dependencies
- Server Setup
- Mysql Connection
- Signup Endpoint
- Login Endpoint
- Query Execution
- Response Handling
- Server start

The image shows two windows side-by-side. The top window is 'phpMyAdmin' showing a database table named 'logins'. The table has columns 'name', 'email', and 'password'. It contains several rows of data. The bottom window is the 'XAMPP Control Panel v3.3.0'. It lists services: Apache (PID 25364, Port 80, 443), MySQL (PID 17952, Port 3306), FileZilla, Mercury, and Tomcat. Each service has buttons for Stop, Admin, Config, Logs, Netstat, Shell, Explorer, Services, Help, and Quit. Below the table is a log window displaying initialization messages for the control panel.

Service	Module	PID(s)	Port(s)	Actions
	Apache	25364 4020	80, 443	Stop Admin Config Logs Netstat Shell
	MySQL	17952	3306	Stop Admin Config Logs Explorer
	FileZilla			Start Admin Config Logs Services
	Mercury			Start Admin Config Logs Help
	Tomcat			Start Admin Config Logs Quit

```
03:46:04 PM [main] Initializing Control Panel
03:46:04 PM [main] Windows Version: Home 64-bit
03:46:04 PM [main] XAMPP Version: 8.2.4
03:46:04 PM [main] Control Panel Version: 3.3.0 [ Compiled: Apr 6th 2021 ]
03:46:04 PM [main] You are not running with administrator rights! This will work for
most application stuff but whenever you do something with services
there will be a security dialogue or things will break! So think
about running this application with administrator rights!
03:46:04 PM [main] XAMPP Installation Directory: "c:\xampp\"
03:46:04 PM [main] Checking for prerequisites
03:46:04 PM [main] All prerequisites found
03:46:04 PM [main] Initializing Modules
03:46:04 PM [Apache] XAMPP Apache is already running on port 80
```

DATABASE MANAGMENT

- Install XAMPP
- Start XAMPP and launch PHP My Admin
- Create a database
- create a table
- Write SQL query

Ideal Result

- Crisp and Intense!

Your ideal customer segment should be 1

- A taste that's elegant and sophisticated!

Your ideal customer segment should be 2

- Subtle and Smooth!

Your ideal customer segment should be 3

THANK YOU