Project Report Format

1. **INTRODUCTION**
    1.1 Project Overview
    1.2 Purpose
2. **LITERATURE SURVEY**
    2.1 Existing problem
    2.2 References
    2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    3.1 Empathy Map Canvas
    3.2 Ideation & Brainstorming
4. **REQUIREMENT ANALYSIS**
    4.1 Functional requirement
    4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    5.1 Data Flow Diagrams & User Stories
    5.2 Solution Architecture
6. **PROJECT PLANNING & SCHEDULING**
    6.1 Technical Architecture
    6.2 Sprint Planning & Estimation
    6.3 Sprint Delivery Schedule
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    7.1 Feature 1
    7.2 Feature 2
    7.3 Database Schema (if Applicable)
8. **PERFORMANCE TESTING**
    8.1 Performace Metrics
9. **RESULTS**
    9.1 Output Screenshots
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    Source Code
    GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

This project's main goal is to use cutting-edge machine learning techniques to completely change the online buying experience. We've created a system that accurately forecasts user behavior on e-commerce sites, with a particular emphasis on differentiating between prospective customers and casual viewers.

1.2 Purpose

This project's main goal is to give businesses useful information about what customers intend to buy when they purchase online. Our goal is to provide businesses with tailored user experiences and focused marketing strategies by determining a visitor's likelihood of making a purchase.

The concept, development, and implementation phases will be covered in detail in the parts that follow, along with the features and techniques that make our project successful. The purpose of this document is to give a thorough summary of our accomplishments and the impact that we have had on the internet commerce industry.

2. LITERATURE SURVEY

2.1 Existing Problem

Online shopping has grown in popularity for a variety of goods, most notably apparel. The dimensions of online shopping qualities and predictors influencing the intention to buy apparel, jewelry, or accessories are the subject of research by Eun Young Kim and Youn-Kyung Kim. The research, which was published in July 2004 in the European Journal of Marketing, aims to clarify how consumers view various aspects of online buying, such as transaction costs, reward schemes, site layout, and interactivity. The results highlight the significance of incentive schemes and transactional features in addition to demographic factors in predicting the propensity to buy particular products online.

Minjeong Kim, Ph.D., and Sharron Lennon, Ph.D., have also looked at the impact of verbal and visual cues on attitudes and buy intentions when it comes to online purchasing. Their 2008 publication of the research questions presumptions.

2.2 References

Predicting online purchase intentions for clothing products

- Authors: Eun Young Kim, Youn-Kyung Kim
- Published: European Journal of Marketing, July 2004
- Key Findings: In order to ascertain the desire to buy apparel, jewelry, or accessories online, the study defines dimensions of online shopping features and predictors, highlighting the significance of transaction/cost, incentive programs, and demographic variables.

The effects of visual and verbal information on attitudes and purchase intentions in internets shopping

- Authors: Minjeong Kim, Ph.D., Sharron Lennon, Ph.D.
- Published: January 11, 2008
- Key Findings: Expectations about the supremacy of visual information in online shopping are challenged by the research. The importance of thorough product descriptions is shown by the fact that, although verbal and visual information both have an impact on consumer sentiments, only verbal information has a substantial impact on purchase intentions.

Towards early purchase intention prediction in online session-based retailing systems

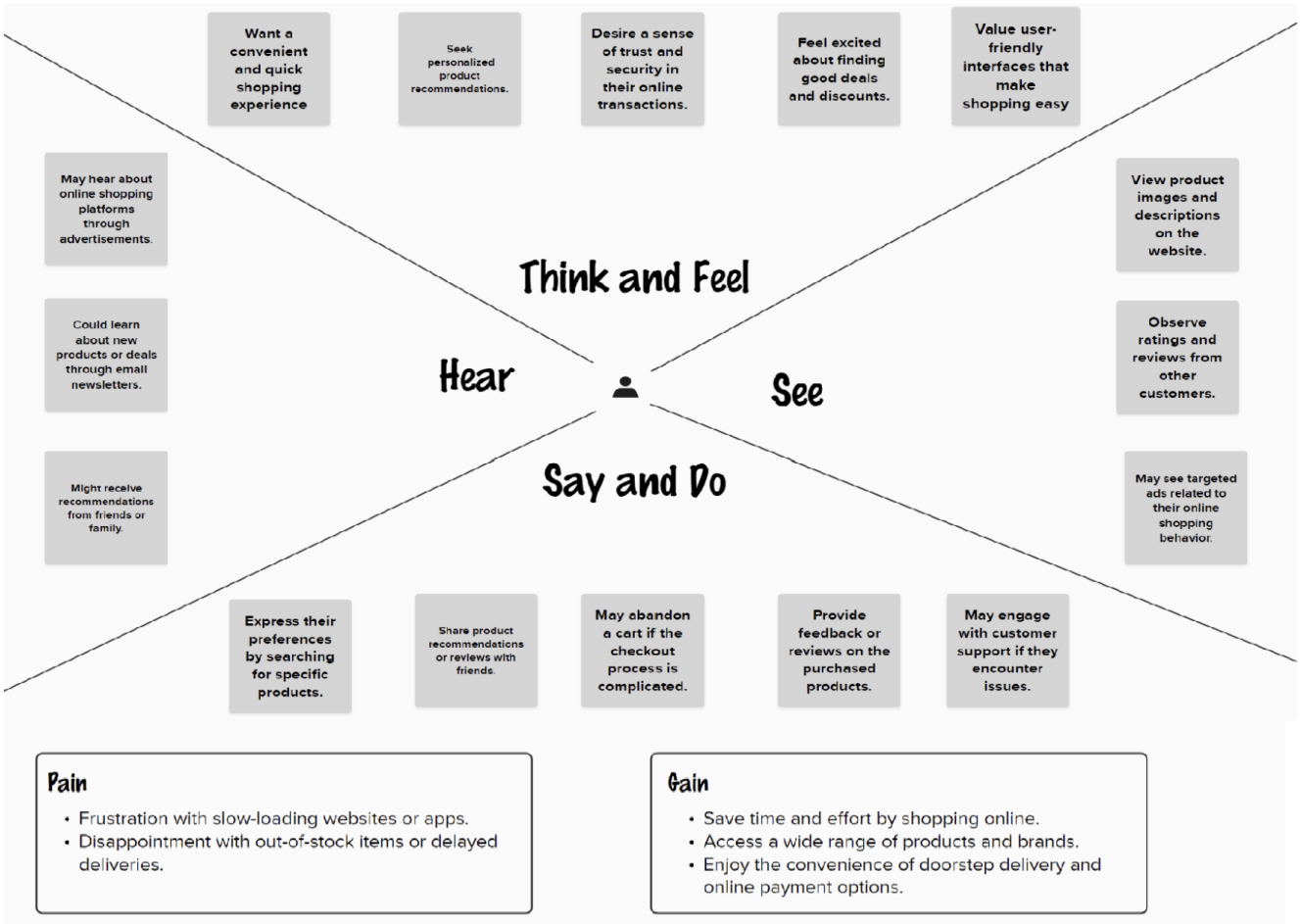
- Authors: Ramazan Esmeli, Mohamed Bader-El-Den, Hassana Abdullahi
- Published: Electronic Markets, December 19, 2020
- Key Findings: In order to forecast purchase intents during an ongoing online session, the study suggests an early buy prediction framework based on machine learning models. The study examines user behavior on e-commerce websites to help with real-time tailored marketing methods.

2.3 Problem Statement Definition

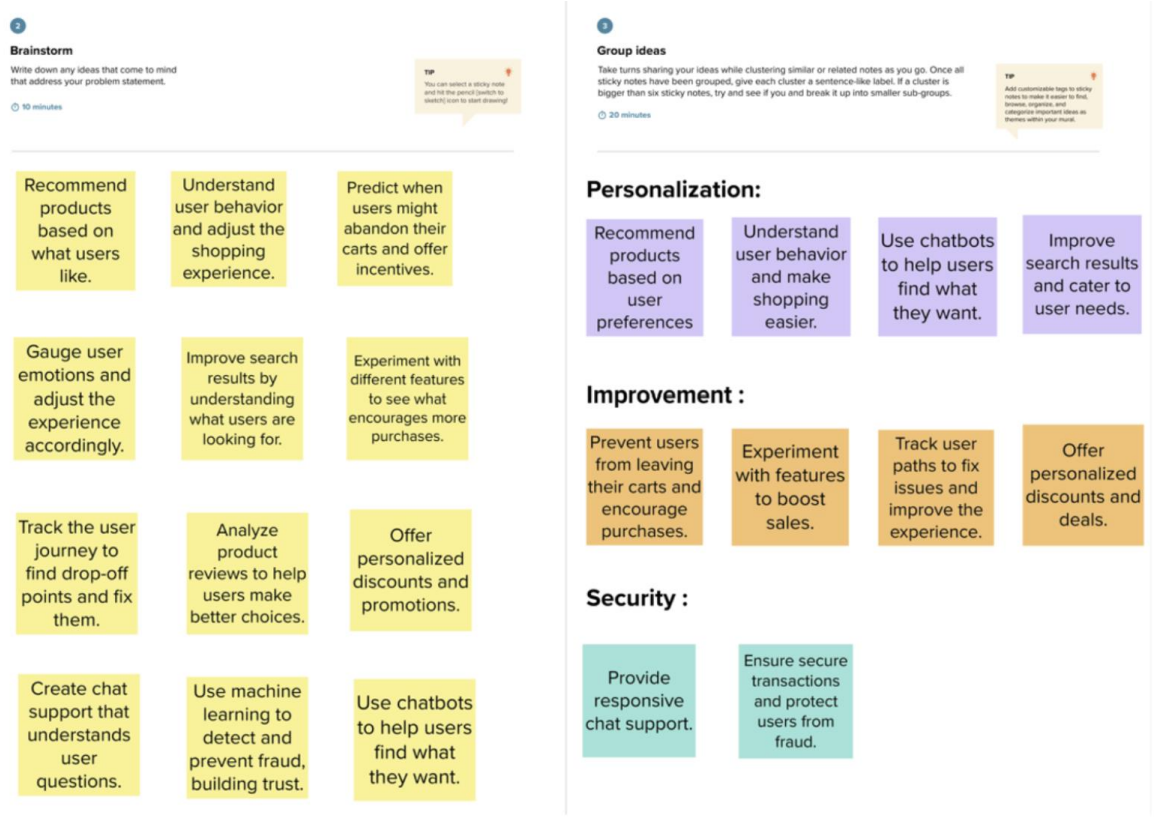
Within the domain of electronic commerce, a basic problem still exists: identifying people who are actually ready to buy ("actual buyers") from those who are just looking. The challenge of classifying users in online buying environments is intricate due to their behavior. It becomes essential to apply cutting-edge machine learning algorithms in order to correctly classify people, providing organizations with a game-changing tool that allows them to improve customer satisfaction, personalize content, and maximize marketing efforts. The main challenge is creating a solid system that can consistently distinguish between users who are genuinely interested in making a purchase and those who are just browsing products, allowing e-commerce businesses to better customize their strategies and products.

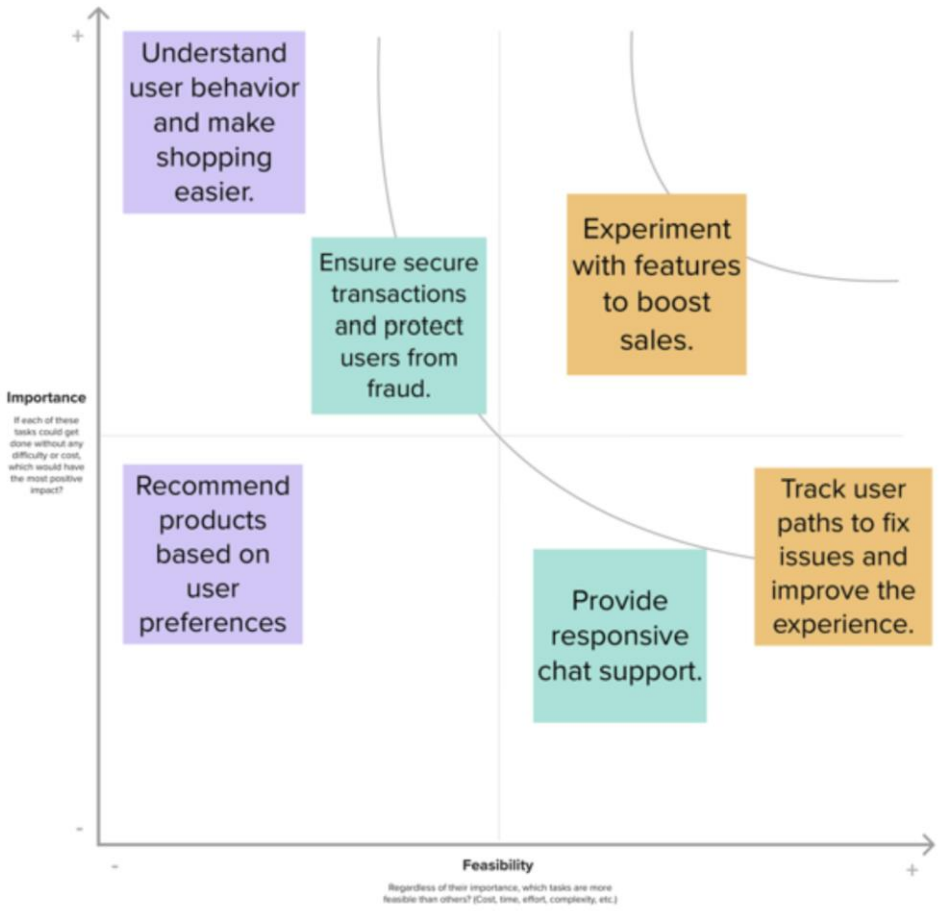
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming





4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

4.1.1 Home Page

- Navigation: Users should be able to navigate to the Home page.
- Menu Options: The Home page should have menu options for easy access to other sections.

4.1.2 About Page

- Navigation: Users should be able to navigate to the About page.
- Information Display: The About page should display information about the project and its objectives.

4.1.3 Predictor Page

- Navigation: Users should be able to navigate to the Predictor page.
- Input Form: The Predictor page should contain a form for users to input various parameters related to online shopping behavior.
- Prediction: Upon form submission, the system should predict the user's intention and display the result.

4.1.4 Submission Page

- Display Result: The Submission page should display the prediction result based on user input.

4.2 Non-Functional Requirements.

4.2.1 Performance

- Response Time: The system should provide predictions within a reasonable time frame, ensuring a smooth user experience.
- Scalability: The application should handle multiple concurrent users without a significant decrease in performance.

4.2.2 User Interface

- Intuitive Design: The user interface should be intuitive, allowing users to navigate through the pages effortlessly.
- Responsive: The application should be responsive and accessible from various devices.

4.2.3 Security

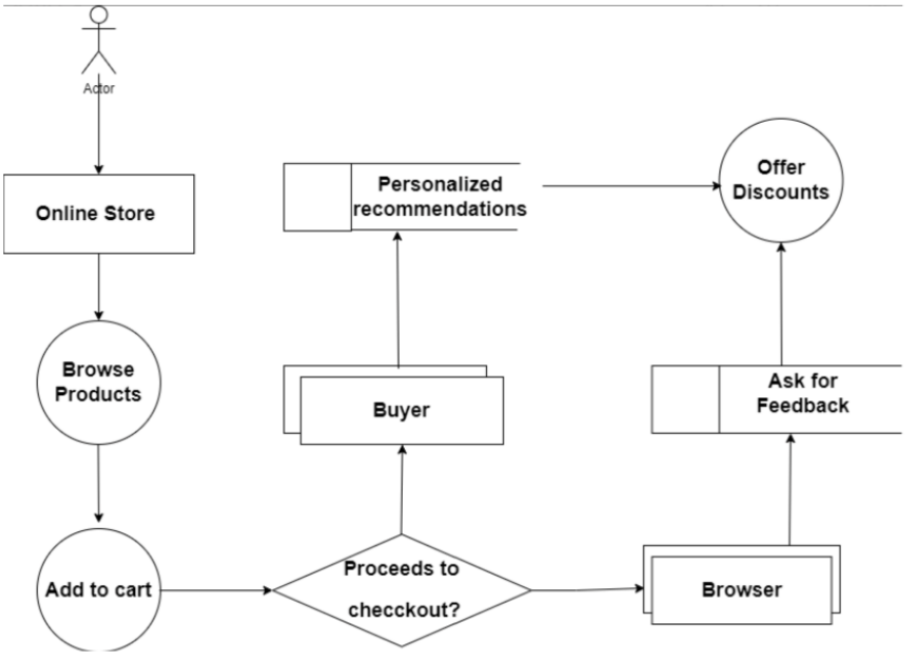
- Data Protection: User inputs and predictions should be handled securely, with measures to protect sensitive information.
- Access Control: Ensure that only authorized users have access to certain features or data.

4.2.4 Data Privacy

- User Privacy: Ensure that user data is handled with strict privacy measures, adhering to relevant data protection regulations.

5. PROJECT DESIGN

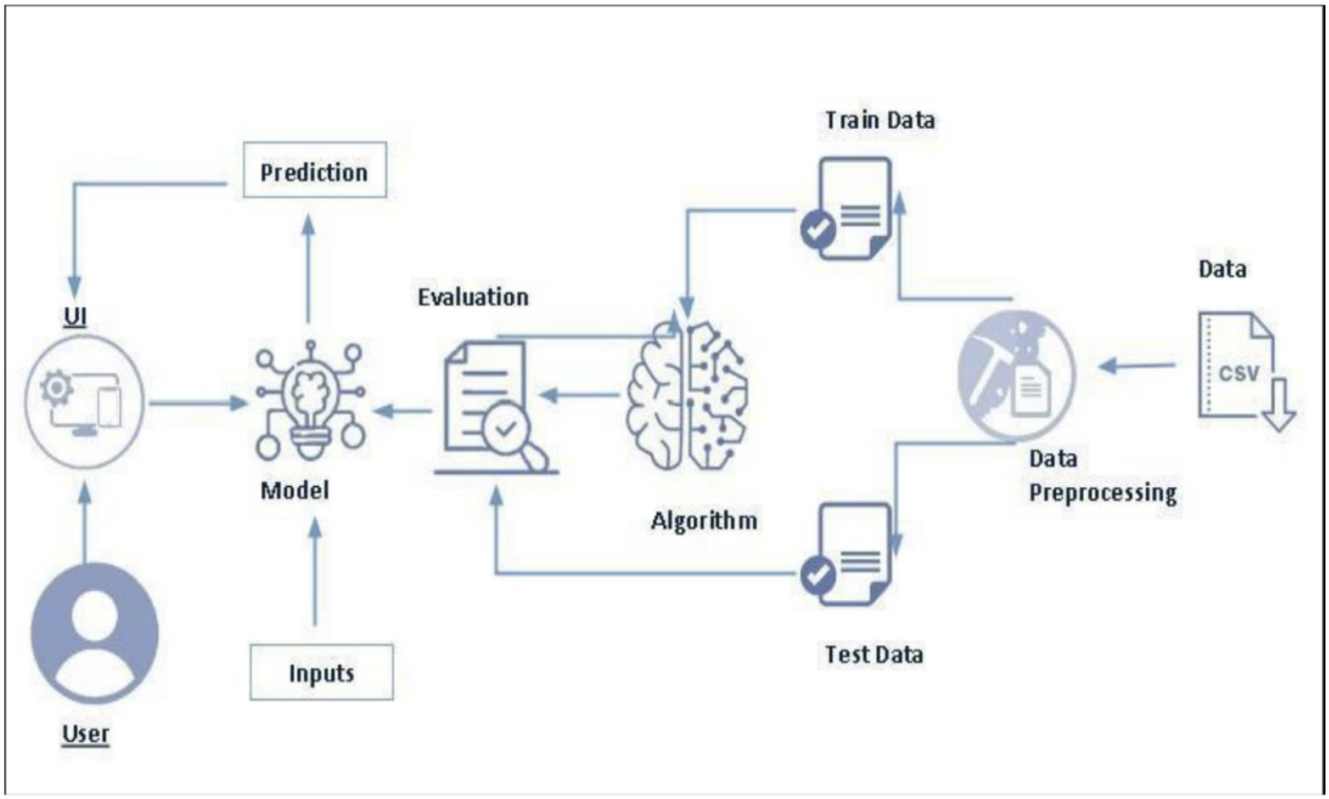
5.1 Data Flow Diagrams & User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Mobile User		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Mobile User		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Mobile User		USN-4	As a user, I can register for the application through Gmail	I can successfully register and access the dashboard using my Gmail account for authentication.	Medium	Sprint-1
Mobile User	Login	USN-5	As a user, I can log into the application by entering email & password	I can successfully log in using a valid email and password combination.	High	Sprint-1
Web User	Registration	USN-6	As a web user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/dashboard.	High	Sprint-1

Customer (Web user)	Login	USN-7	As a web user, I can log into the application by entering my email and password.	I can access my account/dashboard.	High	Sprint-1
Customer Care Executive	Customer Support	USN-8	As a customer care executive, I need access to customer information and order history to assist customers effectively.	I can view customer profiles and order details.	High	Sprint-1
Administrator	User Management	USN - 9	As an administrator, I can view and manage user accounts, including registration details and access privileges.	I can access a user management dashboard.	High	Sprint-1
Administrator	Data Analytics	USN-10	As an administrator, I can access analytics and reports on user behavior, conversion rates, and other key performance indicators.	I can view graphical and tabular analytics reports.	High	Sprint-1
Product Reviewer	Product Reviews	USN-11	As a product reviewer, I can write and submit reviews for products I have purchased.	I can access a product review section.	High	Sprint-1
Browser	Wishlist Management	USN-12	As an online shopper, I can create and manage a wishlist of products I intend to purchase in the future.	I can add products to my wishlist while browsing.	Medium	Sprint-1

5.2 Solution Architecture



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

- Frontend:
  - *HTML:* Used for structuring the web pages and creating their content.
  - *CSS:* Employed for styling and visually enhancing the presentation of HTML elements.
- Backend:

    - *Flask (Python):* Chosen as the web framework for building the backend. Flask is a lightweight and versatile framework that facilitates the development of web applications in Python.
- Machine Learning:
    - *Scikit-learn:* Utilized for implementing machine learning algorithms. Scikit-learn is a robust machine learning library that provides tools for data mining and data analysis.
    - *Pickle for Model Persistence:* Implemented to serialize and deserialize machine learning models. Pickle allows the model to be stored persistently, enabling easy reusability.
    - *NumPy:* Used for numerical operations and handling arrays. NumPy is a fundamental package for scientific computing with Python.
    - *Pandas:* Employed for data manipulation and analysis. Pandas provides data structures and functions needed to manipulate and analyze structured data.
    - *Matplotlib:* Utilized for creating visualizations such as plots and charts. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
    - *Seaborn:* Used for statistical data visualization. Seaborn is built on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Display	USN-1	As a user, I want to...As a user, I want to be able to see the home page (home.html).	2	High
Sprint-1	Information	USN-2	As a user, I want to have an about page (about.html)	2	High
Sprint-1	Navigation	USN-3	As a user, I want to navigate between the home and about pages	2	Medium
Sprint-2	Form	USN-4	As a user, I want to have a page to make predictions (predict.html)	2	High
Sprint-2	Accessible	USN-5	As a user, I want to navigate to the prediction page.	2	High
Sprint-2	Placeholder	USN-6	As a user, I want to experience the beginning of Flask integration	2	High
Sprint-3	Operational input form	USN-7	As a user, I want to fill out a form on the prediction page	3	High
Sprint-3	Python Integration	USN-8	As a user, I want to experience the final Python code for integration.	3	High
Sprint-4	Performance	USN-9	As a user, I want to ensure the application runs smoothly and test it.	2	High

6.3 Sprint Delivery Schedule

Sprint 1: Start Date - 10/11/2023, End Date - 11/11/2023
Sprint 2: Start Date - 12/11/2023, End Date - 14/11/2023
Sprint 3: Start Date - 15/11/2023, End Date - 18/11/2023
Sprint 4: Start Date - 19/11/2023, End Date - 20/11/2023

7. CODING & SOLUTIONING

Feature 1 : Home page
Feature 2 : About page
Feature 3 : Predict form
Feature 4 : Submit page

Code snippets : (Python Flask)

```python
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/about.html")
def about():
    return render_template("about.html")

@app.route("/predict.html")
def predictpage():
    return render_template("predict.html")
```

```python
@app.route( rule: "/predict", methods=["POST"])
def predict():
    # Your existing code here
    Administrative = request.form["administrative"]
    Administrative_Duration = request.form["adminDuration"]
    Informational = request.form["informational"]
    Informational_Duration = request.form["infoDuration"]
    ProductRelated = request.form["productRelated"]
    ProductRelated_Duration = request.form["productDuration"]
    BounceRates = request.form["bounceRates"]
    ExitRates = request.form["exitRates"]
    PageValues = request.form["pageValues"]
    SpecialDay = request.form["specialDay"]
    Month = request.form["month"]
    OperatingSystems = request.form["os"]
    Browser = request.form["browser"]
    Region = request.form["region"]
    TrafficType = request.form["trafficType"]
    VisitorType = request.form["visitorType"]
    Weekend = request.form["weekend"]

    total = [[int(Administrative), float(Administrative_Duration), int(Informational), float(Informational_Duration),
              int(ProductRelated), float(ProductRelated_Duration), float(BounceRates), float(ExitRates),
              float(PageValues), float(SpecialDay), int(Month), int(OperatingSystems), int(Browser), int(Region),
              int(TrafficType), int(VisitorType), int(Weekend)]]

    print(total)

    # Assuming your 'model' is already defined and loaded
    prediction = model.predict(total)
    print(prediction)
```

```python
# Assuming your 'model' is already defined and loaded
prediction = model.predict(total)
print(prediction)

if prediction == 0:
    text = "The visitor is not interested in buying products."
else:
    text = "The visitor is interested in buying products."

return render_template( template_name_or_list: "submit.html", result=text)
```

```python
@app.route("/submit.html")
def submit():
    return render_template("submit.html")
```

8. PERFORMANCE TESTING

8.1 Performance Metrics

```
RandomForestClassifier :
Confusion matrix
[[3005  110]
 [ 257  327]]
Classification report
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      3115
           1       0.75      0.56      0.64       584

    accuracy                           0.90      3699
   macro avg       0.83      0.76      0.79      3699
weighted avg       0.89      0.90      0.89      3699
```

Confusion Matrix:

- True Positive (TP): Instances correctly predicted as positive.
- True Negative (TN): Instances correctly predicted as negative.
- False Positive (FP): Instances incorrectly predicted as positive.
- False Negative (FN): Instances incorrectly predicted as negative.

Classification Report:

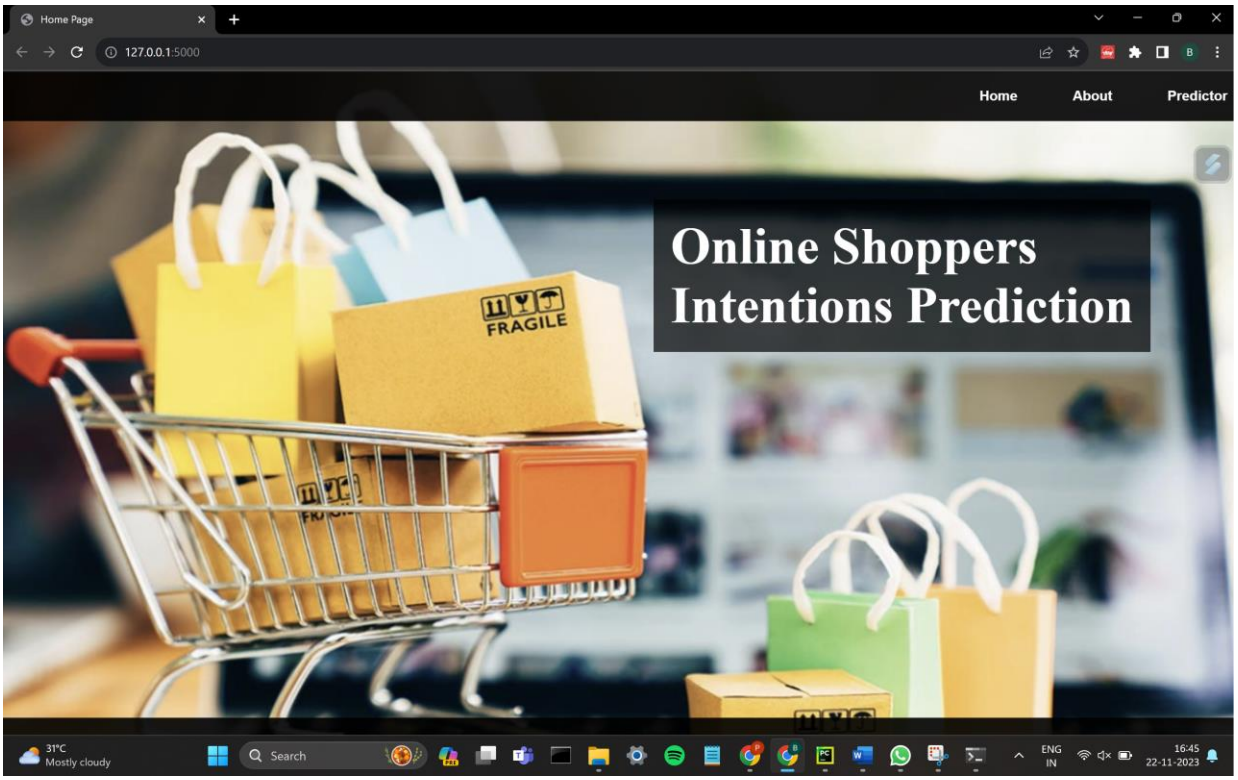
- Precision: Accuracy of positive predictions.
- Recall (Sensitivity): Ability to capture all positive instances.
- F1-Score: Balance between precision and recall.
- Support:Number of instances of the class in the dataset.

Accuracy: Ratio of correct predictions to total predictions.

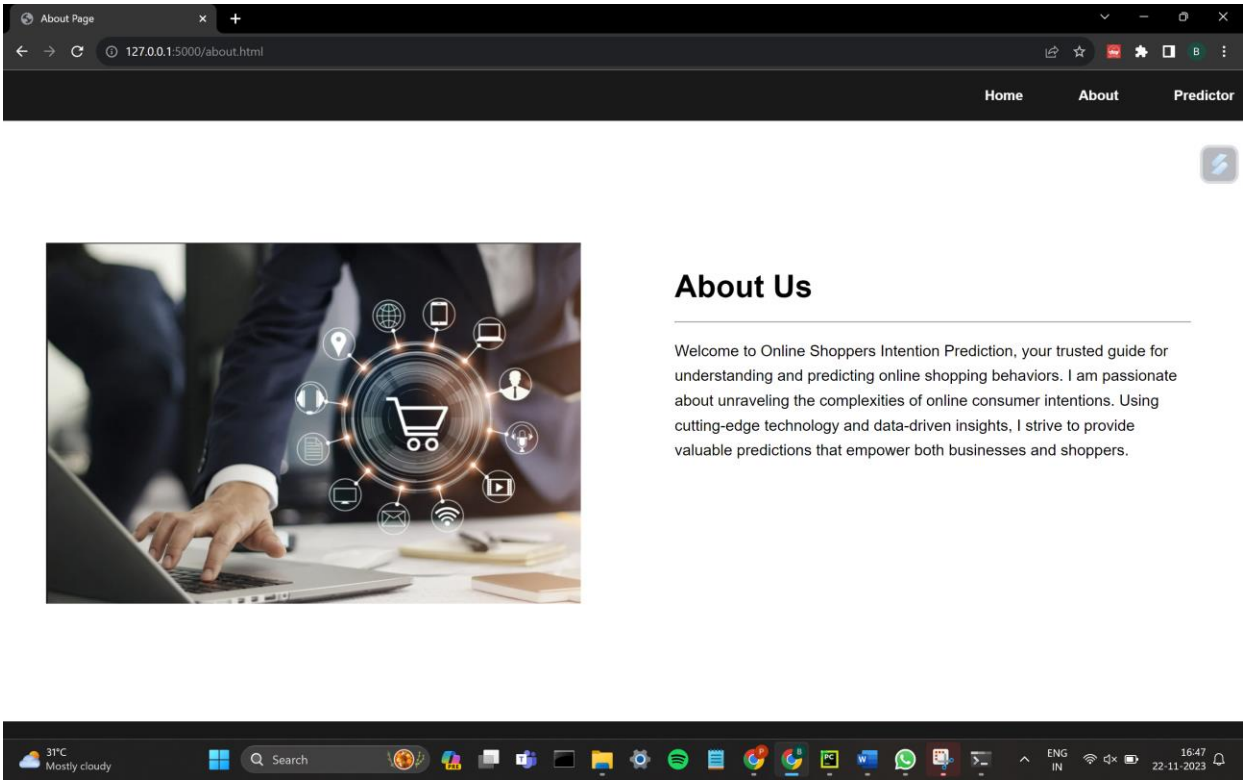
9. RESULTS

9.1 Output Screenshots

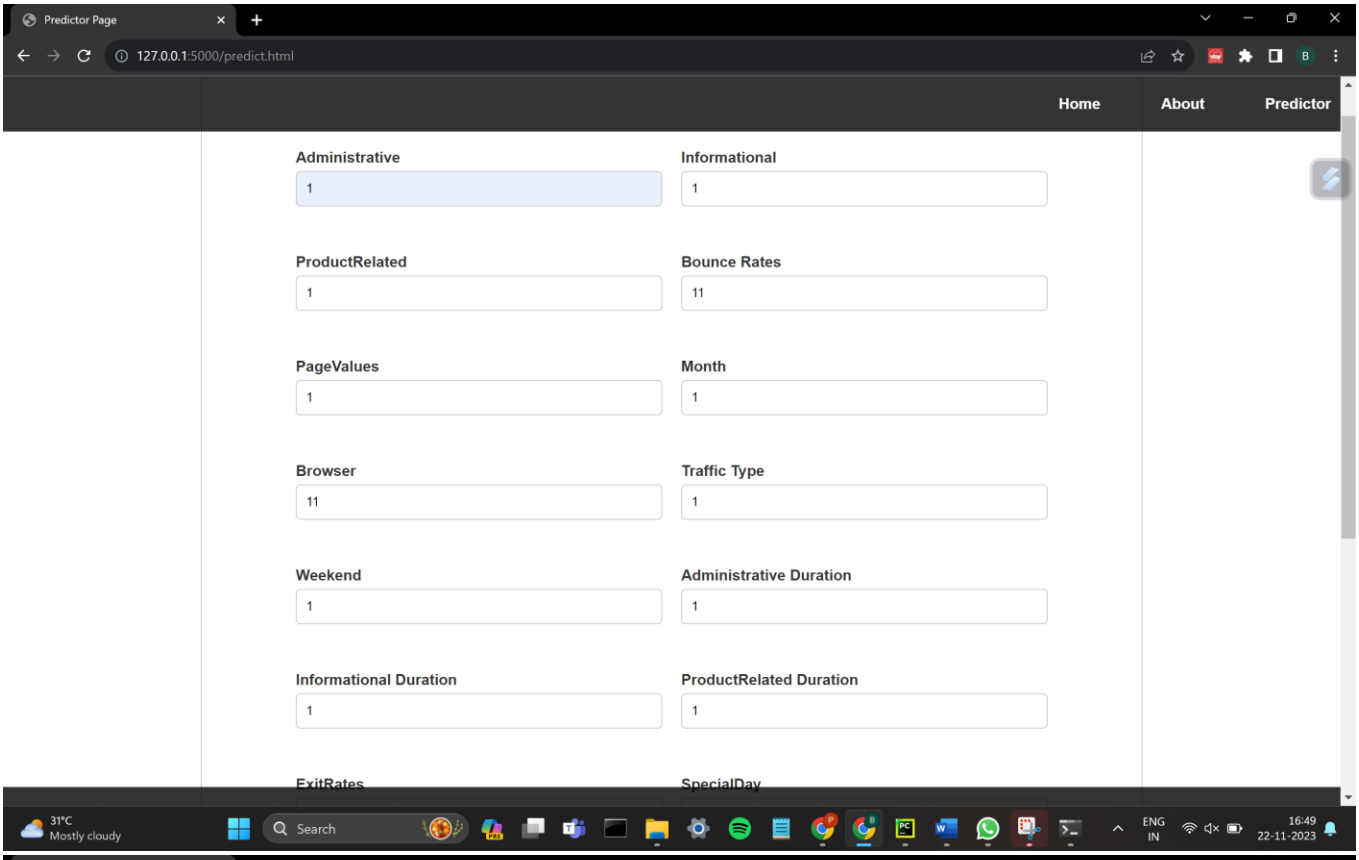
Home Page :

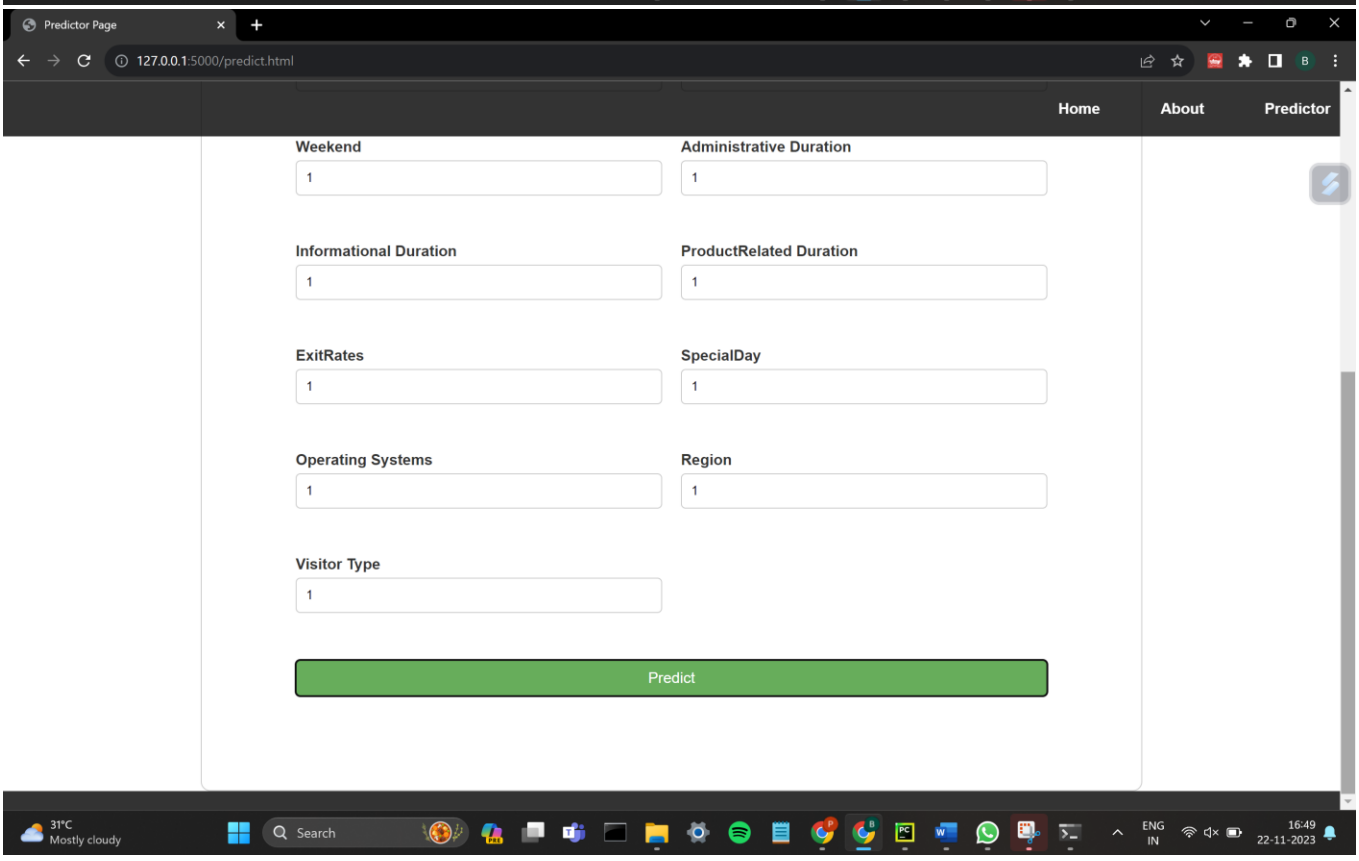


About Page :

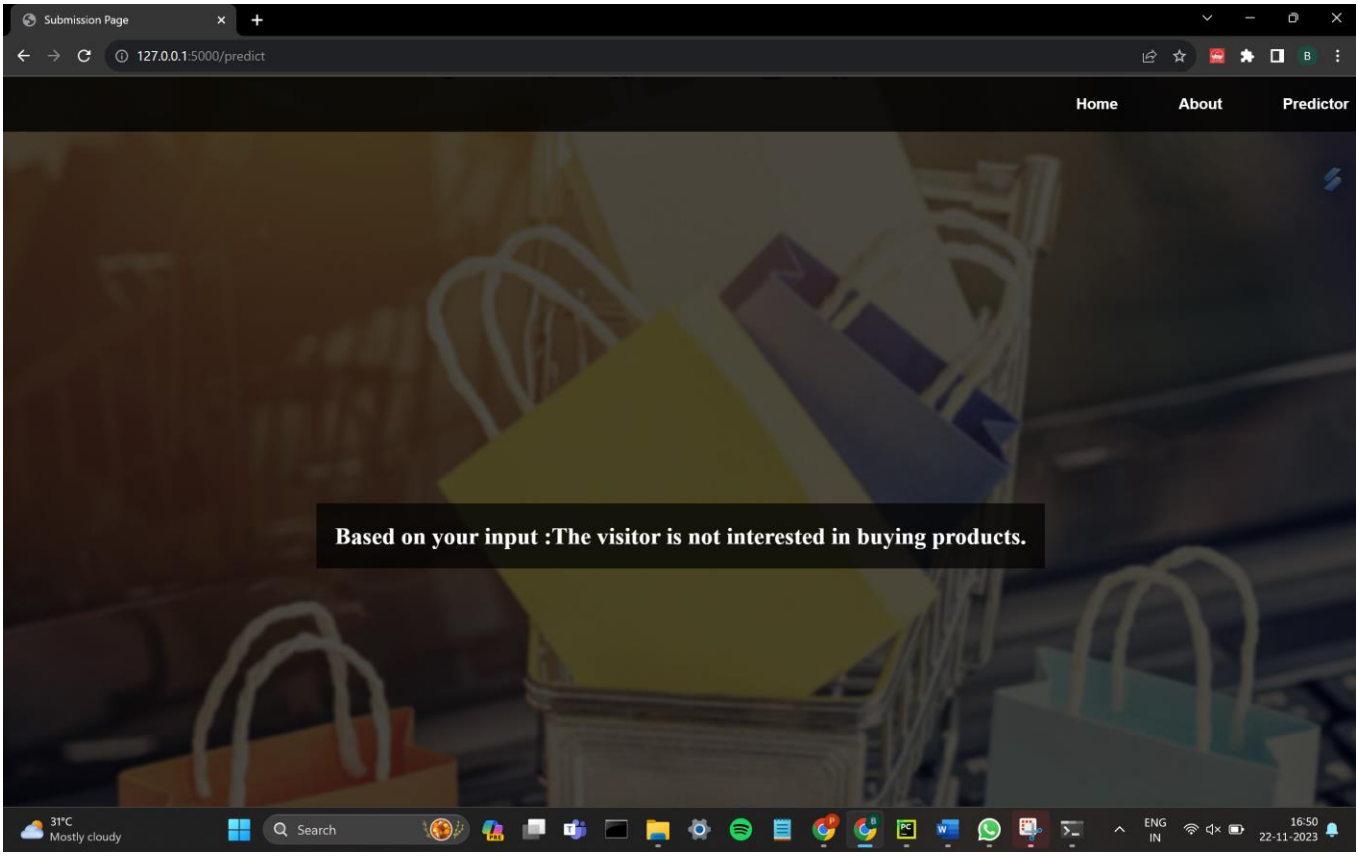


Predictor page :





Submit page :



10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

1. User-Friendly Interface
2. Real-time Prediction
3. Data-Driven Insights
4. Cross-Platform Compatibility

10.2 Disadvantages

1. Dependency on Input Quality
2. Limited Feature Set
3. Sensitivity to Model Changes

11. CONCLUSION

In conclusion, our software combines a robust backend (Flask in Python), an intuitive user interface (HTML, CSS), and astute predictions (Scikit-learn with Pickle for model storage). This combination produces a useful tool for predicting the likelihood that an online visitor will make a purchase.Our model generates accurate forecasts by taking into account multiple criteria such as visitor categories, browser data, and user actions. Data processing is now more efficient thanks to Python tools like NumPy, Pandas, Matplotlib, and Seaborn.Metrics such as the confusion matrix and accuracy checks ensure the dependability of the model throughout testing. When making selections about online shopping, users can rely on the predictions. In the future, we want to improve the model.

12. FUTURE SCOPE

The Online Shoppers' Intentions Prediction project has a lot of promising future applications. A primary goal is to broaden the model's functionality to take into account more facets of online shopper behavior. Important objectives include integrating user feedback for ongoing enhancement and introducing real-time predictions for rapid insights. Future plans include investigating multi-channel support, enhancing security, and creating a mobile app to increase accessibility. The project will advance through internationalization initiatives and cooperation with e-commerce platforms. These upcoming actions are intended to guarantee that the project stays relevant and keeps expanding in the ever-changing realm of internet commerce.

13. APPENDIX

Github Source Code :

https://github.com/smartinternz02/SI-GuidedProject-613083-1698816205/tree/main

Project Demo Link :

https://drive.google.com/file/d/1wJZR2d6NeK-TIDWbP9URfhtpnCQN4Bru/view?usp=sharing