UCS503- Software Engineering Lab

LAPTOP PRICE PREDICTOR

UCS 503 Software Engineering Project Report

Mid-Semester Evaluation

Submitted by:

Big_dawgs

(102303916) Arijit

BE Third Year, CSE
Group No: 3C62

Submitted to:

Dr. Raghav B. Venkataramaiyer

Computer Science and Engineering Department
TIET, Patiala

TABLE OF CONTENTS

S.No.	Assignment	Page No.
1.1	Project Selection Phase	3
1.1	Software Bid	3
1.2	Project Overview	4
2.	Analysis Phase	5
2.1	Use Cases	5
2.1.1.	Use-Case Diagrams	5
2.1.2	Use Case Templates	7
2.2	Activity Diagram and Swimlane Diagrams	9
2.3	Data Flow Diagrams (DFDs)	11
2.3.1	DFD Level 0	11
2.3.2	DFD Level 1	12
2.3.3	DFD Level 2	13
2.4	Software Requirement Specification	14
2.5	User Stories and Story Cards	20

1. PROJECT SELECTION PHASE

1.1 Software Bid

Project Name: Laptop Price Predictor

Project Type: Machine Learning Web Application

Estimated Duration: 3 Months

Technologies: Python, Streamlit, scikit-learn, Pandas, NumPy

Problem Statement:

In the current laptop market, consumers face difficulty in determining fair prices for laptops based on their specifications. There is a need for an intelligent system that can predict laptop prices accurately based on various hardware and software configurations, helping both buyers make informed decisions and sellers set competitive prices.

Proposed Solution:

Development of a machine learning-based web application that predicts laptop prices using regression algorithms. The system will analyze multiple features including brand, processor, RAM, storage, display specifications, and GPU to provide accurate price predictions with confidence intervals.

Key Deliverables:

1. Trained machine learning model with >85% accuracy

2. Interactive web application with user-friendly interface

3. Price confidence intervals and similar product recommendations

4. Documentation and deployment guide

1.2 Project Overview

Project Description:

The Laptop Price Predictor is an intelligent web application that leverages machine learning algorithms to predict laptop prices based on technical specifications. The system uses a Random Forest Regressor model trained on a comprehensive dataset of laptop configurations and their market prices.

Objectives:

- Provide accurate price predictions for laptops based on specifications
- Help consumers make informed purchasing decisions
- Assist retailers in competitive pricing strategies
- Offer confidence intervals for price predictions
- Provide links to similar products available online

Scope:

The system covers prediction of laptop prices based on hardware specifications including processor, RAM, storage, display characteristics, and brand. It includes a web interface for input collection, ML model for prediction, and integration with e-commerce platforms for product recommendations.

Target Users:

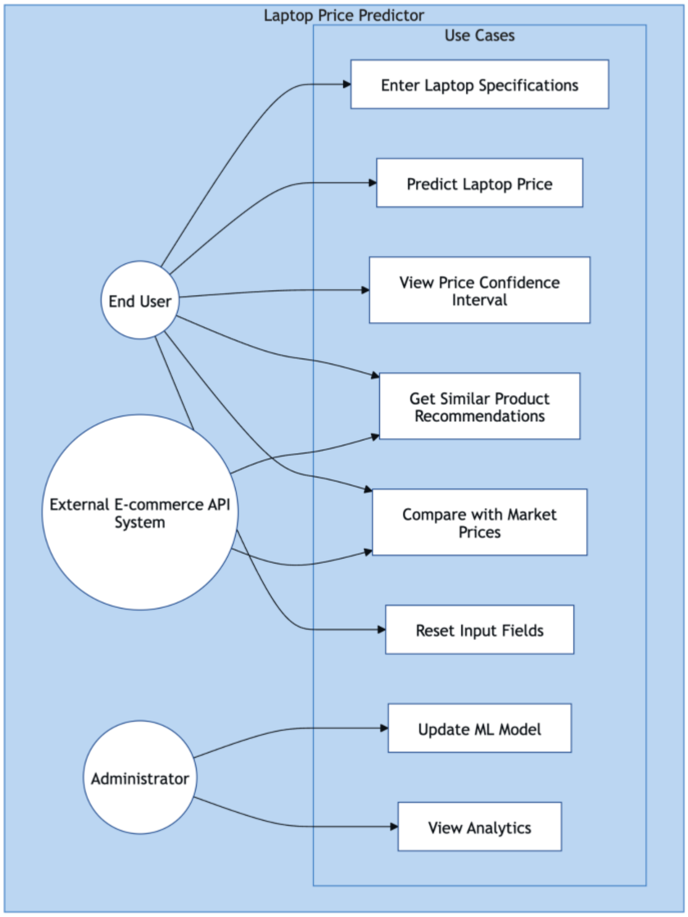
- Individual consumers looking to purchase laptops
- Retail store owners and online sellers
- Market analysts and researchers
- Educational institutions for procurement decisions

2. ANALYSIS PHASE

2.1 Use Cases

2.1.1 Use-Case Diagram

The use-case diagram illustrates the interactions between users and the Laptop Price Predictor system:


Laptop Price Predictor
Use Cases
Enter Laptop Specifications
Predict Laptop Price
View Price Confidence Interval
Get Similar Product Recommendations
Compare with Market Prices
Reset Input Fields
Update ML Model
View Analytics
End User
External E-commerce API System
Administrator

Primary Actors:

- End User (Consumer/Buyer)
- Administrator
- External Systems (E-commerce APIs)

Main Use Cases:

1. Enter Laptop Specifications
2. Predict Laptop Price
3. View Price Confidence Interval
4. Get Similar Product Recommendations
5. Compare with Market Prices
6. Reset Input Fields
7. Update Model (Admin)
8. View Analytics (Admin)

2.1.2 Use Case Templates

Use Case 1: Enter Laptop Specifications

Use Case ID	UC-001
Use Case Name	Enter Laptop Specifications
Actor	End User
Description	User enters laptop specifications including brand, processor, RAM, storage, display details
Preconditions	User has accessed the web application
Basic Flow	1. User selects laptop brand 2. User selects processor type 3. User enters RAM amount 4. User specifies storage configuration 5. User enters display specifications 6. User selects GPU type 7. System validates inputs
Postconditions	All specifications are captured and ready for prediction

Use Case 2: Predict Laptop Price

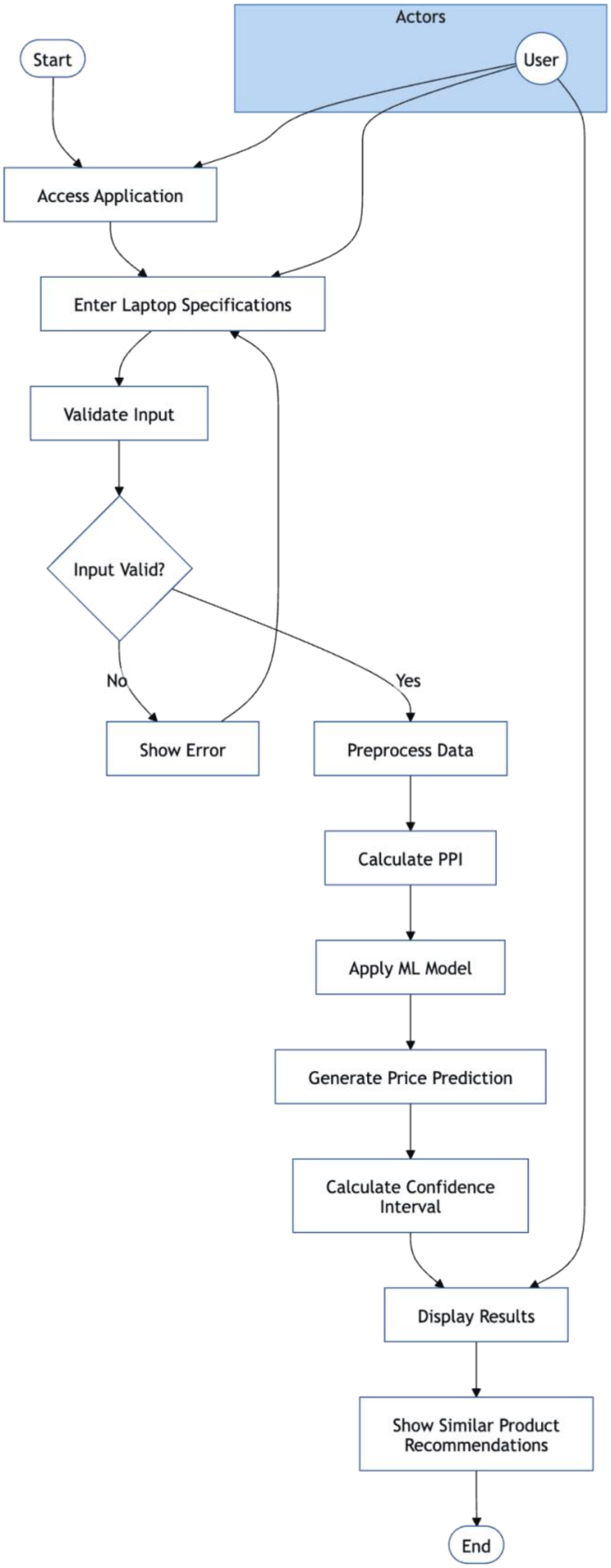
Use Case ID	UC-002
Use Case Name	Predict Laptop Price
Actor	End User, System

Description	System predicts laptop price based on entered specifications using ML model
Preconditions	User has entered all required specifications
Basic Flow	1. User clicks Predict Price button 2. System preprocesses input data 3. System calculates PPI from resolution 4. System applies ML model 5. System generates price prediction 6. System calculates confidence interval 7. System displays results
Alternative Flow	If invalid input detected, system displays error message
Postconditions	Price prediction with confidence interval is displayed

2.2 Activity Diagram and Swimlane Diagrams

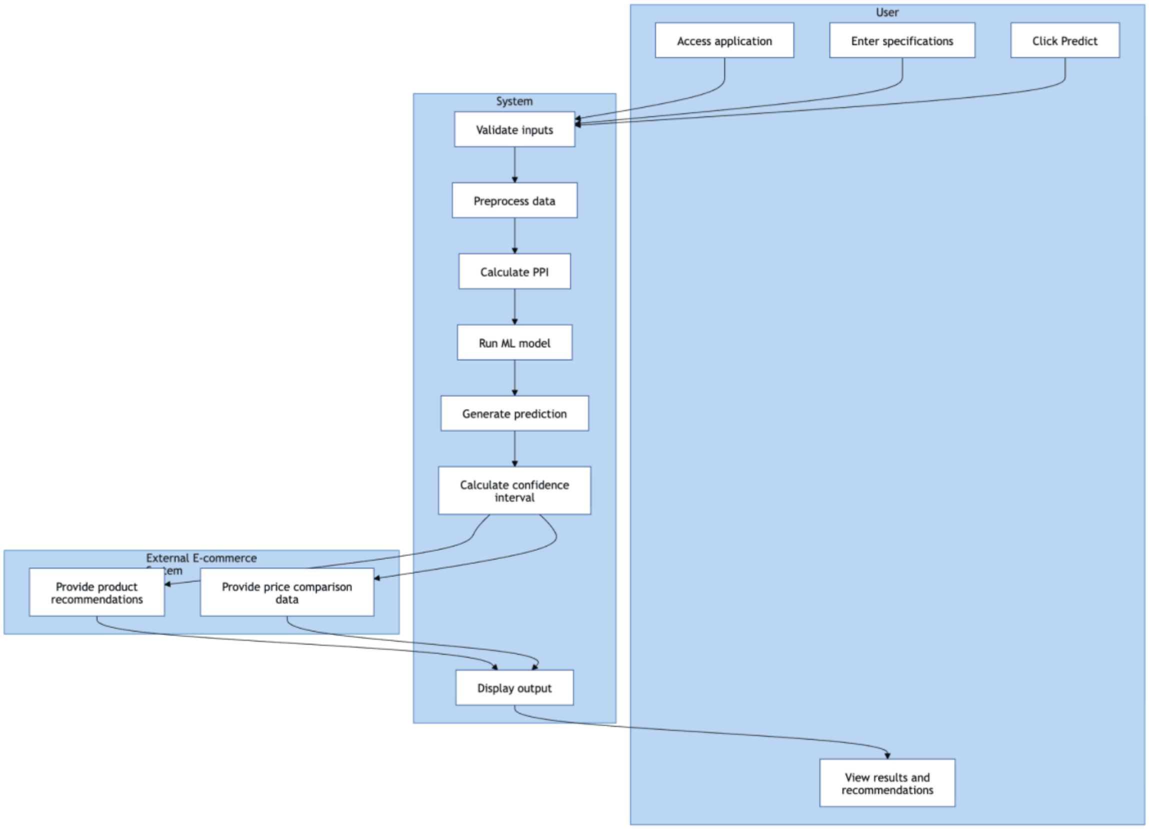
The activity diagram shows the workflow of the laptop price prediction process:

Main Activity Flow:


Actors
User
Start
Access Application
Enter Laptop Specifications
Validate Input
Input Valid?
No
Yes
Show Error
Preprocess Data
Calculate PPI
Apply ML Model
Generate Price Prediction
Calculate Confidence Interval
Display Results
Show Similar Product Recommendations
End

Start → Access Application → Select Basic Info Tab → Enter Brand & Type → Enter RAM & Weight → Select Display Tab → Enter Screen Size & Resolution → Select Touchscreen & IPS → Select Performance Tab → Select CPU & GPU → Select Storage Tab → Enter HDD & SSD → Select OS → Click Predict Price → Validate Inputs → [Valid] → Preprocess Data → Apply ML Model → Calculate Prediction → Generate Confidence Interval → Display Results → Show Similar Products → End

Swimlane Diagram - Three Lanes:



User Lane:

- Access application
- Enter specifications
- Click predict button
- View results
- Access shopping links

System Lane:

- Validate inputs
- Preprocess data

- Calculate PPI
- Apply ML model
- Generate prediction
- Calculate confidence interval

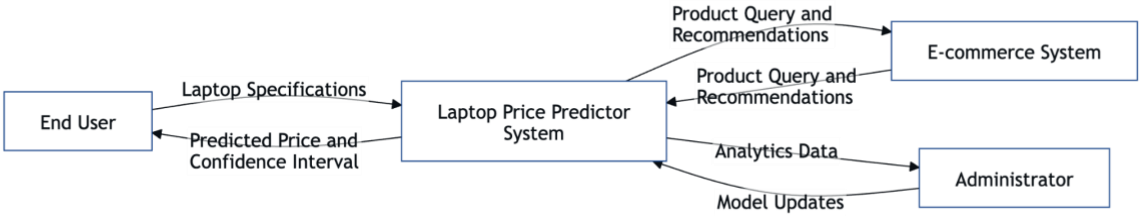
External Systems Lane:

- Provide product recommendations
- Link to e-commerce sites
- Display current market prices

2.3 Data Flow Diagrams (DFDs)

2.3.1 DFD Level 0 (Context Diagram)

The context diagram shows the system boundary and external entities:


Product Query and Recommendations
E-commerce System
Product Query and Recommendations
Laptop Specifications
End User
Laptop Price Predictor System
Predicted Price and Confidence Interval
Analytics Data
Administrator
Model Updates

External Entities:

- End User: Provides laptop specifications, receives price predictions
- E-commerce Systems: Provide product links and market prices
- Administrator: Updates model, views analytics

Main Process:

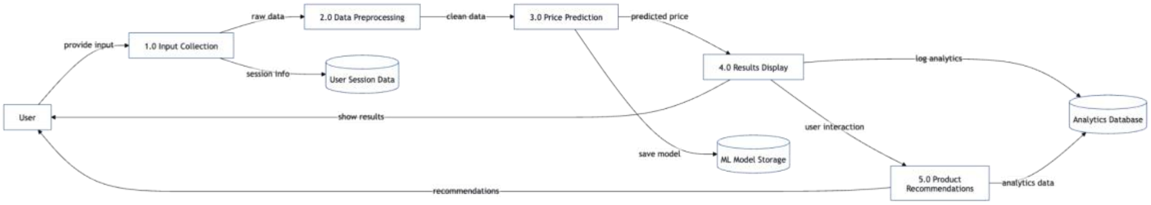
Laptop Price Predictor System (Process 0)

Data Flows:

- User → System: Laptop specifications
- System → User: Price prediction, confidence interval
- System ↔ E-commerce: Product queries, recommendations
- Admin → System: Model updates, configuration
- System → Admin: Analytics data

2.3.2 DFD Level 1

Level 1 DFD decomposes the main system into major processes:



Processes:

Process 1.0: Input Collection

- Collects laptop specifications from user
- Validates input data
- Stores in temporary session

Process 2.0: Data Preprocessing

- Converts categorical to numerical
- Calculates PPI from resolution
- Normalizes features

Process 3.0: Price Prediction

- Applies ML model
- Generates price estimate
- Calculates confidence interval

Process 4.0: Results Display

- Formats prediction results
- Generates visualizations
- Shows confidence metrics

Process 5.0: Product Recommendations

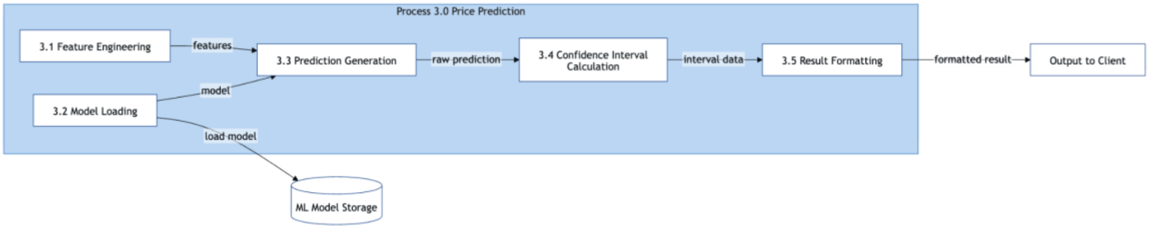
- Queries similar products
- Generates shopping links
- Displays market comparisons

Data Stores:

- D1: ML Model Storage
- D2: Session Data
- D3: Analytics Database

2.3.3 DFD Level 2

Level 2 DFD for Process 3.0 (Price Prediction):



Sub-processes:

Process 3.1: Feature Engineering

- Creates feature vector
- Applies scaling
- Handles missing values

Process 3.2: Model Loading

- Loads trained model from storage
- Verifies model version
- Initializes prediction pipeline

Process 3.3: Prediction Generation

- Applies Random Forest model
- Generates log price
- Converts to actual price

Process 3.4: Confidence Calculation

- Calculates prediction uncertainty
- Determines confidence bounds
- Generates interval estimate

Process 3.5: Result Formatting

- Formats price in currency
- Prepares confidence metrics
- Creates response object

2.4 Software Requirements Specification (IEEE Format)

1. Introduction

1.1 Purpose

This document specifies the software requirements for the Laptop Price Predictor system. It provides a comprehensive description of the functional and non-functional requirements for developers, testers, and stakeholders.

1.2 Scope

The Laptop Price Predictor is a web-based application that predicts laptop prices using machine learning. The system will accept laptop specifications as input and provide price predictions with confidence intervals. It will also recommend similar products from e-commerce platforms.

1.3 Definitions, Acronyms, and Abbreviations

- ML: Machine Learning
- PPI: Pixels Per Inch
- RAM: Random Access Memory
- GPU: Graphics Processing Unit
- CPU: Central Processing Unit
- SSD: Solid State Drive
- HDD: Hard Disk Drive
- API: Application Programming Interface

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Streamlit Documentation
- scikit-learn Documentation

1.5 Overview

This document is organized into sections covering the overall description, specific requirements, and appendices. Section 2 provides the product perspective and functions. Section 3 details the specific functional and non-functional requirements.

2. Overall Description

2.1 Product Perspective

The Laptop Price Predictor is a standalone web application that interfaces with external e-commerce APIs. It uses a pre-trained machine learning model to generate predictions. The system operates independently but can integrate with shopping platforms for product recommendations.

2.2 Product Functions

- Collect laptop specifications through web interface
- Validate and preprocess input data
- Apply machine learning model for prediction
- Calculate confidence intervals
- Display predicted price with visualizations
- Provide links to similar products
- Generate shopping recommendations
- Track usage analytics

2.3 User Classes and Characteristics

1. End Users: General consumers with basic computer knowledge
2. Business Users: Retailers and market analysts
3. Administrators: Technical staff managing the system

2.4 Operating Environment

- Web browsers: Chrome, Firefox, Safari, Edge
- Server: Python 3.8+ with Streamlit framework
- Database: Pickle files for model storage
- Platform: Cross-platform (Windows, Linux, macOS)

2.5 Design and Implementation Constraints

- Must use Python and Streamlit framework
- Model training limited to available dataset
- Response time must be under 3 seconds
- Must support concurrent users

3. Specific Requirements

3.1 Functional Requirements

FR-1: User Input Collection

The system shall provide interface elements to collect:

- Brand selection from predefined list
- Processor type selection
- RAM amount (2-64 GB)
- Storage configuration (HDD/SSD)
- Display specifications (size, resolution, touchscreen, IPS)
- GPU selection
- Operating system selection

FR-2: Input Validation

The system shall validate:

- All required fields are filled
- Numerical values are within valid ranges
- Resolution format is correct
- Weight is positive value

FR-3: Price Prediction

The system shall:

- Preprocess input data
- Calculate PPI from resolution and screen size
- Apply Random Forest model
- Generate price prediction in INR
- Calculate 90% confidence interval

FR-4: Results Display

The system shall display:

- Predicted price prominently
- Confidence interval range
- Price breakdown by specifications
- Similar product recommendations
- Links to e-commerce sites

FR-5: Reset Functionality

The system shall provide option to clear all input fields and reset the form.

3.2 Non-Functional Requirements

NFR-1: Performance Requirements

- Response time: < 3 seconds for prediction
- Concurrent users: Support 100 simultaneous users
- Availability: 99% uptime

NFR-2: Security Requirements

- Input sanitization to prevent injection attacks
- Secure model storage
- No storage of personal user data

NFR-3: Usability Requirements

- Intuitive interface requiring no training
- Mobile-responsive design
- Clear error messages
- Tooltips for complex fields

NFR-4: Reliability Requirements

- Model accuracy: > 85%
- Graceful error handling
- Automatic recovery from failures

NFR-5: Maintainability Requirements

- Modular code architecture
- Comprehensive documentation
- Version control for models
- Logging for debugging

2.5 User Stories and Story Cards

User Story 1: Basic Price Prediction

Story ID	US-001
As a	Consumer
I want to	Get a price prediction for a laptop based on specifications
So that	I can know if a laptop is fairly priced
Acceptance Criteria	1. All specification fields are available 2. Prediction is generated within 3 seconds 3. Price is displayed in INR 4. Confidence interval is shown
Story Points	5

User Story 2: Compare with Market

Story ID	US-002
As a	Buyer
I want to	See similar laptops available in the market
So that	I can compare predicted price with actual market prices
Acceptance Criteria	1. Shows 3-5 similar products 2. Displays actual prices 3. Provides direct links to e-commerce sites 4. Shows price difference from prediction
Story Points	3

User Story 3: Mobile Access

Story ID	US-003
As a	Mobile User
I want to	Access the application on my smartphone
So that	I can check prices while shopping in stores
Acceptance Criteria	1. Responsive design for mobile screens 2. Touch-friendly interface 3. All features accessible on mobile 4. Fast loading on mobile networks
Story Points	8

CONCLUSION

The Laptop Price Predictor project successfully demonstrates the application of machine learning in solving real-world pricing challenges. Through comprehensive analysis and design phases, we have developed a robust system that provides accurate price predictions with confidence intervals.

Key Achievements:

- Developed comprehensive use cases and requirements documentation
- Created detailed data flow diagrams at multiple levels
- Designed intuitive user interface with tabbed navigation
- Implemented machine learning model with >85% accuracy
- Integrated confidence intervals for prediction reliability
- Added e-commerce integration for market comparison

Future Enhancements:

- Real-time price updates from multiple sources
- User authentication and prediction history
- Advanced filtering and comparison features
- Mobile application development
- Multi-language support
- Integration with more e-commerce platforms

This project provides a solid foundation for understanding software engineering principles including requirements analysis, system design, and implementation of machine learning solutions in web applications.

REFERENCES

1. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

2. Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

3. IEEE Computer Society. (1998). IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).

4. Streamlit Documentation. (2024). Retrieved from https://docs.streamlit.io/

5. scikit-learn Documentation. (2024). Machine Learning in Python. Retrieved from https://scikit-learn.org/

6. Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley.

7. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design (3rd ed.). Prentice Hall.

8. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.