IBM Innovation Center for Education

PHEMESOFT
Software That Matters
TRUSTED DELIVERY PARTNER for IBM ICE

PROJECT SYNOPSIS

FOR

WINE QUALITY PREDICTION SYSTEM

Mohamed Vasal
22BDACC151
BCA (Big Data Analytics, Cloud Computing and Cybersecurity) With IBM
Yenepoya Institute of Arts Science Commerce and Management
Mangalore
Under the guidance of
Mr. Sumit Kumar Shukla
Faculty, Computer Science Department
Yenepoya Institute of Arts Science Commerce and Management
Mangalore

Submitted to:

RABBI ZIDNI 'ILMA
YENEPOYA
(DEEMED TO BE UNIVERSITY)
Recognized under Sec 3(A) of the UGC Act 1956
Accredited by NAAC with 'A' Grade

YENEPOYA INSTITUTE OF ARTS, SCIENCE, COMMERCE AND MANAGEMENT
BALMATTA, MANGALORE
YENEPOYA (DEEMED TO BE UNIVERSITY)

TABLE OF CONTENT

I. TITLE OF THE PROJECT

Wine Quality Prediction System (Qualivino)

II. STATEMENT OF THE PROBLEM

Wine quality assessment is essential for winemakers and enthusiasts, but traditional methods rely on subjective sensory evaluations, which are inconsistent and time-consuming. Existing automated solutions often lack user-friendly interfaces, secure authentication, or features like prediction history and report generation, making them impractical for everyday use. Many platforms are not responsive across devices, limiting accessibility, and fail to prioritize data security, risking user information exposure. The Qualivino Wine Quality Prediction System addresses these challenges by providing a web-based platform that uses machine learning to deliver accurate wine quality predictions (0-10) based on chemical properties (e.g., alcohol, pH). It offers a secure login system, a responsive interface with glassmorphism design, prediction history tracking, and downloadable PDF reports. By combining precision, security, and ease of use, Qualivino empowers users to make informed decisions about wine quality, suitable for winemakers, enthusiasts, and researchers, while ensuring accessibility on computers, tablets, and phones.

III. WHY THIS PARTICULAR TOPIC CHOSEN?

- Practical Relevance: Wine quality assessment is critical for the wine industry, making an automated, accurate tool valuable for practical applications.

- User-Centric Gap: Existing solutions lack intuitive interfaces and comprehensive features, justifying a user-friendly platform like Qualivino.

- Technical Learning: Developing Qualivino enabled exploration of machine learning (Random Forest, XGBoost, LightGBM), web development (Flask, Bootstrap), and security (Bcrypt), enhancing technical skills.

- Industry Alignment: The project aligns with IBM’s focus on data analytics and cybersecurity, integrating ML and secure authentication.

- Enhanced Accessibility: A responsive design ensures the app works seamlessly across devices, meeting modern user expectations.

- Security Focus: Implementing secure login and data handling addresses critical cybersecurity needs, protecting user privacy.

- Engaging Development: Combining ML, web design, and wine-themed aesthetics made the project both challenging and enjoyable, fostering creative problem-solving.

IV. OBJECTIVE AND SCOPE

Objective

- Deliver accurate wine quality predictions (0-10) using machine learning models based on chemical properties.
- Provide a secure user authentication system to protect personal data and ensure personalized access.
- Enable prediction history tracking to allow users to review past results.
- Offer downloadable PDF reports summarizing prediction details for documentation.
- Create a responsive, user-friendly interface compatible with computers, tablets, and phones.

Scope

- Develop a web application with an intuitive interface for wine quality predictions and history management.
- Utilize UCI wine quality datasets for training ML models (Random Forest, XGBoost, LightGBM).
- Implement secure user authentication with login, registration, and logout features.
- Include prediction history storage and PDF report generation using xhtml2pdf.
- Ensure compatibility with modern browsers (Chrome, Firefox, Safari, Edge) and devices.
- Limit scope to core features, excluding advanced analytics (e.g., real-time data feeds) or external APIs

V. METHODOLOGY

- Planning and Design: Defined Qualivino's goals to include accurate predictions, secure authentication, history tracking, and PDF reports, designing a clean, wine-themed interface with glassmorphism for aesthetic appeal and usability across devices.

- Technology Selection: Chose Python for backend logic, Flask for web framework, SQLite for lightweight database storage, and Bootstrap for responsive frontend design. Employed scikit-learn, XGBoost, and LightGBM for ML models, with Bcrypt for secure password hashing and xhtml2pdf for PDF generation.

- Development Process: Built the app in phases, starting with user authentication (login/registration), followed by ML model training on UCI datasets, prediction functionality, history storage in sessions, and PDF report generation.

- Data Integration: Processed UCI wine quality datasets, applying SMOTE for class imbalance and feature engineering (e.g., acid balance) to enhance model accuracy.

- User Interface Creation: Designed an intro page with a video background, a login page with secure forms, and a predictor page with input forms, prediction results, and a history panel, optimized for all devices using Bootstrap.

- Testing Approach: Tested prediction accuracy (targeting ±1 score unit), login security, history functionality, PDF generation, and UI responsiveness, ensuring fast load times (~2 seconds) and secure data handling.

- Deployment Preparation: Configured the app for local execution, providing setup instructions for Flask, SQLite, and ML model dependencies.

VI. PROCESS DESCRIPTION

- **User Authentication**: Users register with a username and password, stored securely in SQLite with Bcrypt hashing. A login page creates a session for personalized access, with logout functionality ending the session to prevent unauthorized use.

- **Wine Quality Prediction**: Users input chemical properties (e.g., fixed acidity, alcohol, pH) via a form on the predictor page. A pre-trained ML model (Random Forest, XGBoost, or LightGBM) processes inputs, enhanced by feature engineering, to predict a quality score (0-10), displayed with a color-coded result (e.g., green for ≥7).

- **Prediction Classification**: The system classifies predictions into categories (e.g., Poor: <5, Average: 5-6, Excellent: ≥7) using a Random Forest Classifier, providing users with an interpretable quality label alongside the score.

- **Prediction History Tracking**: Predictions are stored in the user's session, including inputs, scores, and timestamps, accessible via a toggleable history panel on the predictor page, with options to download individual predictions as PDFs.

- **PDF Report Generation**: Users can generate PDF reports from the history panel, formatted with xhtml2pdf to include prediction details (score, inputs, timestamp) in a professional, wine-themed layout.

- **Responsive Interface**: The app's UI, built with Bootstrap and custom CSS, features a wine-themed design (#800026, #f8d7da) and glassmorphism, ensuring seamless navigation across intro, login, and predictor pages on computers, tablets, and phones.

VII. RESOURCES AND LIMITATIONS

Resources Required

Resources

- **Hardware**: Developed on a standard laptop with a 1.6 GHz processor, 4GB RAM, 500MB storage, and a 5 Mbps internet connection for dataset downloads and library installations.

- **Software Tools**: Used Python for backend and ML, Flask for web routing, SQLite for database, Bootstrap for responsive UI, and JavaScript for dynamic features (e.g., history panel toggle).

- **Datasets and Libraries**: Leveraged UCI wine quality datasets (red, white) for ML training, with scikit-learn, XGBoost, LightGBM for modeling, Bcrypt for security, and xhtml2pdf for reports.

- **Development Environment**: Employed Visual Studio Code for coding, Jupyter Notebook for ML experimentation, and Git for version control to manage project iterations.

- **Limitations**

- **Testing Scope**: Time constraints limited testing to core features (predictions, authentication, history, PDFs), with less focus on edge cases (e.g., extreme input values).

- **Prediction Accuracy**: ML models achieved ±1 score unit accuracy in 85% of tests, but performance depends on input quality and dataset representativeness.

- **Database Scalability**: SQLite’s concurrency limitations restrict handling >100 simultaneous users, suggesting PostgreSQL for future scaling.

- **Offline Functionality**: Requires internet for initial setup and dataset access, limiting offline use.

- **Feature Scope**: Excludes real-time data feeds or advanced analytics due to project timeline and complexity constraints..

VIII. **TESTING TECHNOLOGIES USED**

- Python Testing Tools: Used pytest to test ML models (Random Forest, XGBoost, LightGBM) for prediction accuracy and Random Forest Classifier for quality classification, ensuring reliable outputs.
- Browser Developer Tools: Employed Chrome DevTools to verify UI responsiveness across devices (desktop, tablet, mobile) and browsers (Chrome, Firefox, Safari, Edge).
- Database Testing: Tested SQLite queries with sqlite3 CLI to confirm secure storage and retrieval of user data and prediction history.
- Performance Monitoring: Measured page load times (~2 seconds) and prediction speeds (~1.8 seconds) using Python’s time module, ensuring efficient operation.
- Security Testing: Conducted manual checks for secure login (Bcrypt hashing, session timeouts), input validation (preventing XSS/SQL injection), and HTTPS usage for data transfer**.**

IX. CONCLUSION

- The Qualivino Wine Quality Prediction System successfully delivers a user-friendly, secure web application that accurately predicts wine quality using machine learning, achieving ±1 score unit accuracy in 85% of test cases. It integrates Random Forest, XGBoost, and LightGBM models trained on UCI datasets, enhanced by SMOTE and feature engineering, to provide reliable predictions for winemakers and enthusiasts. The Flask-based app features a responsive, wine-themed UI with glassmorphism, secure authentication, prediction history tracking, and PDF report generation, accessible across devices. Comprehensive testing confirmed fast response times, robust security, and seamless functionality, despite limited testing scope and SQLite scalability constraints. Qualivino demonstrates effective use of ML, web development, and cybersecurity, fulfilling BCA project requirements and offering practical value, with potential for future enhancements like mobile apps and advanced analytics.