HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Applying machine learning model to product brand recognition

Tống Thành Vinh
vinh.tt194884@sis.hust.edu.vn

Major: Information Technology

Supervisor: M.C.S. Lê Tấn Hùng ______________

Signature

Department: Computer Engineering

School: Information and Communications Technology

HANOI, 08/2023

ACKNOWLEDGMENTS

To begin with, I would like to express my sincere gratitude to my family and friends who continuously supported me through my time completing this project. Next, I would like to show my appreciation to the instructors in the computer science program, especially Le Tan Hung, who serves as my adviser. He not only showed dedication and motivated me, but also showed me the correct path.

Following that, I considered my time spent studying at the Hanoi University of Science and Technology to be an important period in my own journey toward growth and advancement. I believe that my understanding today has greatly increased as a result of the teachers' commitment to help and the information they provided.

ABSTRACT

I would like to state that an enormous number of e-commerce websites are built with friendly user interfaces and excellent functionalities. However, through several investigations, I would like to contribute to the advancement of them with the use of an artificially intelligent tool to further increase the users' experience when browsing for items, thus increasing the company's revenue. Because of that, I took the initial steps to apply this concept to a wine-selling project to prove my points. One example of this is when a person is looking for wines that share similarities with the inputted one. He or she can utilize this model to find the desired wines easily. Another example that I have been through is when working for a small pub, there is a customer who decides to try a product from the same winery but at a lower price so I need to browse for all the wines in the storage by myself. From the above points, I have been fond of the idea of an AI model receiving a wine image as input and returning a list of similar wines from the database. In order to accomplish my goal and also meet client preferences and widen their wine horizons, this thesis describes the construction of an AI-driven wine recommendation system, notably the "Find Similar Wines" function. With the goal of creating an accurate wine catalog and a deep learning model based on ResNet-50 architecture for wine categorization, the project involves considerable data collecting and analysis. FastAPI, MongoDB, Next.js, and Tailwind CSS are used in the system's implementation to make a user-friendly website. The model's efficacy is demonstrated through performance testing and real-world deployment, which deliver precise and timely wine suggestions. The partnership with a nearby wine distributor attests to the usefulness of the technology and its potential to improve the wine-buying experience. The project's successes highlight the importance of AI-driven technology in revolutionizing the wine business and offer the promise of tailored suggestions for more product categories in the future. In general, this study advances e-commerce platforms and provides guidelines for creating effective recommendation systems that are user-centered.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abriviation	Full Expression
AI	Artificial Intelligent
API	Application Programming Interface
CSS	Cascading Style Sheet
E-commerce	Electronic commerce
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
MongoDB	Mongo Database
ResNet	Residual Network
VGG	Visual Geometry Group

===

CHAPTER 1. INTRODUCTION

1.1 Motivation

The wine selling industry has made a great deal of advancements, including economic expansion, widespread wine understanding among consumers, increased advancements in flavor, and the rise of several wine online retailers. But I've discovered that there are still some problems. To begin with, due to the enormous wine database that each wine online store might have, buyers are first prevented from locating their favorite wines due to a lack of knowledge and competence. There are then variations in each person's taste. It is true that some individuals could choose vineyards whose offerings are solely regarded as sweet, while others embrace premium wine producers because of the flavors of the components used to make each one. The pricing restrictions will be discussed next. It might be difficult for small bars, wine wholesalers, or restaurants to provide less expensive alternatives compared to those that clients are seeking since modest charges can change the wine's flavor so much. Additionally, this will be more difficult if there are restrictions on the quantity of stored wines. Moving to the upcoming issue, which is the lack of personalization, some bottles' designs can catch the eyes of specific customers, and we can take advantage of this to gain profit. However, because gathering and analyzing the preferences of every single type of buyer can be too troublesome, it is an unsolved problem for wineries in general to come up with a design for each type of customer. To conclude, there are four concerns need to be solved: lack of knowledge and competence, variations in each person's taste, pricing restrictions and lack of personalization. From the above, if we can find solutions to cover them up, the wine consumer's browsing experience can be increased. Thus, this can also raise both wine providers and producers' revenue. And when the wine industry's economic status keeps on going up, these solutions can also be applied to other industries to gain similar results.

1.2 Objectives and scope of the graduation thesis

To begin with, the vineyard industry has developed online and traditional wine-selling platforms to meet the demands of customers in acquiring and recommending a wide range of goods. A few examples can be listed are Vivino.com, Winc, Amazon, eBay, and traditional wine stores and bars. Following that, Vivino.com and Winc are largely dedicated digital wine marketplaces, with over 50 million registered users and more than 110000 active subscribers respectively, that provide wide options and customized experiences. Next, although Amazon has a large selection of wines with user evaluations, eBay is a place that allows people to sell their wine immediately. Traditional alcohol shops and bars provide a more physically customized buying experience, but their stock might seem restricted. While online platforms make browsing wines more convenient, traditional distributors allow buyers to physically check their preferred wines before purchase. Next, some analyses of the wine market reveal information about the customers' behaviors. When looking for and buying wine, wine buyers have distinct expectations. They demand detailed statistics on each wine, such as its name, producer, area, year of manufacture, and taste characteristics. They also want individualized suggestions and want to try comparable wines. According to the above behavior analysis, because of a shortage of consistent data and reliable recommendation systems, present wine-selling services suffer from a lack of complete and tailored qualifications. These restrictions required a solution that filled such gaps while satisfying customer demands. As a result, a visible solution will be an artificial intelligence-powered wine-selling system that provides consumers with full details, tailored suggestions, and the option to browse related wines. This approach tackles current obstacles and matches the needs of wine buyers, transforming the market, enhancing client fulfillment, and boosting income for both distributors and wineries. In summary, while an easy-to-use, straightforward user interface matters most for a pleasant exploring and purchase experience, any digital wine distribution website needs to understand that each consumer will seek wine with various features, preferences, pricing, and so on.

1.3 Tentative solution

The goal of this graduation project is to create an AI-driven wine suggestion engine to address issues in the wine-selling sector. Since every model will need a large dataset including gathered images and data to be trained on, before proceeding to the model training phase, a crawling mechanism using Selenium is required to collect data from an online wine distribution website. Then, through transfer learning, create a deep learning model utilizing the pre-trained ResNet50 model's weights for image classification. The ResNet50 model has been selected due to its balance between performance in classifying image tasks, capacity to deal with complicated visual characteristics, and widespread acceptance and availability of pre-trained weights. Following that, this project is expected to increase customers' experience when searching for wine, stimulate their curiosity, and thereby increase revenue for small wine distributors. In addition, it is also expected to make it easier for pubs, bars, eateries, or small restaurants to find similar bottles of wine at a lower cost. Therefore, the project's developed model gets sufficient accuracy, precision, and recall scores, allowing it to fulfill tasks such as categorizing and then suggesting equivalent wines according to the photos of the given wine bottles. In summary, by utilizing ResNet50's trained weights to develop a wine classification model, the achieved results are considered acceptable through many experiments and tests.

1.4 Thesis organization

1.4.1 Chapter 2

In general, this chapter tries to collect data regarding the preferred flavors and demands among prospective customers. It carries out a study to determine consumer needs and examines current wine-selling platforms to find their advantages and disadvantages. This chapter also describes the project's functional and non-functional requests. Chapter 2 examines a number of user requirements-related topics. It starts by addressing the Customer Needs Study, outlining the technique used to compile information from prospective wine buyers. The poll asks both closed-ended and multiple-choice questions about things like wine interests and shopping habits. The major conclusions are presented in the Key Insights section, emphasizing the value of thorough wine details, consumers' desire for locating related wines, and the necessity of an intuitive user interface. These observations inform the project's primary characteristics, which include a comprehensive wine database, an easy-to-use interface, and a personalized wine recommendation engine. Additionally, "ruouvang24h.com" and "vivino.com," two current wine-selling platforms, are analyzed in this chapter to determine their advantages and disadvantages. Understanding customer tastes and demands for this graduation project is crucial after reading this chapter. The questionnaire and examination aid in identifying functional and non-functional needs and selecting important features. The use case diagrams give a precise depiction of the key features of the program. Overall, the chapter provides guidance for any online platform's development to successfully satisfy user expectations.

1.4.2 Chapter 3

The strategies utilized to collect, process, and supplement data for the capstone project are examined in this chapter. The first section of the chapter provides a summary of the data collection procedure, which includes utilizing Selenium web scraping to acquire wine images from "ruouvang24h.com." In order to improve data management, the obtained data was then processed by deleting null or non-number values and adding additional columns. The author applied numerous picture augmentation techniques, including flipping, affine transformations, Gaussian blur, contrast changes, noise addition, and sharpening, using the "imgaug" package in the portion that focuses on data augmentation. The dataset was greatly expanded throughout this procedure, becoming more varied and suitable for deep learning model training. The development of a deep learning model for classifying wines is then covered in detail in the chapter. The ResNet-50 architecture was selected and refined by using transfer learning technique to properly categorize 75 wine producers. The use of pre-trained models that have been modified for certain tasks is described as "transfer learning," which eliminates the need for intensive training on new datasets. The model was trained, and the author then used a variety of measures to assess the model's effectiveness. In order to evaluate the model's propensity to properly predict both positive and negative instances in the dataset, accuracy, precision, and recall score were utilized. The chapter also contains sections on constructing a MongoDB database for effective data storage, building a Python API using FastAPI, and designing a user-friendly website using Next.js with Tailwind CSS to apply the developed API. The author emphasizes the advantages of each technology and how well it meets the needs of the project. The chapter concludes with a discussion of alternative strategies that were taken into account throughout the project's research phase. These strategies included investigating other online scraping libraries, deep learning architectures, data storage alternatives, and novel model training techniques. In conclusion, this chapter offers a thorough overview of the techniques utilized for data collection, processing, augmentation, and model construction, as well as the metrics used to measure the model's effectiveness. A number of technologies, including FastAPI, MongoDB, and Next.js, have been integrated, demonstrating the effectiveness and promise of AI-driven technologies in the wine sector. In order to accomplish the project's objectives, the chapter also recognizes the alternative strategies that were taken into account.

1.4.3 Chapter 4

The selection of libraries and tools utilized, the demonstration of key features, and the design of the database are the primary topics of Chapter 4's overview of the application development process. The performance testing of the model is then covered, starting with the method used to create the test set, followed by tests being run on the set and a working demonstration of the model on Google Colab. The chapter further illustrates the model's performance in real-world circumstances by describing the methodology, deployment contexts, and outcomes. The chapter ends with an evaluation of the model's predictive power.

1.4.4 Chapter 5

The goals of the planning stage are described in Chapter 5, which involves developing an AI model that can evaluate input wine photographs and provide a list of wines that are aesthetically similar. In the context of wines, the term "similar" was defined as "visually comparable," depending on factors including label design, bottle style, and color. It was decided to employ a classification model as the input and output approach for the AI model in order to identify the wine picture's creator. It was reasonable to use the producer's information to identify wines that were aesthetically equivalent since it was anticipated that wines from the same winery would have similar visual features. A strong "Find Similar Wines" tool for the project was created as a consequence of careful planning and evaluation of these elements. Next, the difficulties encountered during the data collecting and storage phase are then covered in this chapter. Additionally, it mentions how data preprocessing was added to assure data purity and completeness, and how the data crawling method was modified to account for changes in data display on various websites. In the end, a wine dataset was successfully gathered, and the process for collecting data was improved. The need for data augmentation approaches to expand the dataset and offer a broad representation of wine photos is then highlighted in this chapter. The technique of data augmentation strengthened the model's capacity to generalize to fresh and uncharted data, resulting in improved efficiency and precision in recommending wines with comparable appearances. The chapter concludes by discussing the choice of model architecture for the wine classification problem, taking into account the complexity of the dataset, the computational resources that are available, and the required model performance. As a result of its mix of model complexity, computational effectiveness, and classification performance, ResNet50 was selected as the best option for the website's visual wine recommendation system.

1.4.5 Chapter 6

The goal of Chapter 6 is to wrap up this graduation project. The "Find Similar Wines" function is implemented using rigorous data collection, preprocessing, and model training, which are addressed as significant successes. The effective implementation of the "Find Similar Wines" functionality and the AI model for wine categorization are only two of the noteworthy achievements listed in this chapter. The significance of data quality, model selection standards, and user-centric design are only a few of the important lessons discovered. The creation of websites, enlarging the library of wine images, investigating cutting-edge deep learning architectures, and incorporating user preferences for customized wine selections will be the main areas of future effort. These initiatives seek to improve customers' wine discovery experiences.

1.4.6 Chapter 7

The bibliographies of the research papers and other sources cited throughout the whole project are collected in this chapter. These sources provide crucial information, methodology, and data that assist the creation of the recommendation system and act as the research's cornerstone. The mentioned articles and materials are crucial for developing a thorough comprehension of the field and guaranteeing the precision and authority of the project's results and conclusions.

CHAPTER 2. REQUIREMENT SURVEY AND ANALYSIS

2.1 User requirements survey

Initially, several online and offline surveys were carried out to gain a better grasp of the demands of potential wine buyers or users for this graduation project's project. The objective was to obtain relevant data that would guide the creation of a website for retailing wine online.

2.1.1 Survey methodology

To begin with, the goal of the survey is to discover more about potential "Wine-Cart" consumers' preferences and perceptions. Understanding wine consumers' requirements and expectations for choosing and buying wines online is the main objective. After getting insights from people, this graduation project's model and website will be developed using the analyzed result to prioritize features which benefit the majority of buyers. The investigation is set up as a form with a mix of multiple-choice and closed-ended questions. Closed-ended questions contain a list of predetermined answers, enabling respondents to select the one that most accurately expresses their viewpoint. The questionnaire asks about a variety of topics, including wine preferences, purchase patterns and experiences. To guarantee relevant and accurate findings, the poll focuses a varied range of wine consumers. Volunteers are chosen from forums devoted to wine, social media sites, small wine shops and both offline, online wine groups. As result, a total number of 1537 participants have participated in this survey. Respondents are also well conscious that their private data will be kept anonymously and is utilized for researching purpose only. Next, the questionnaire includes inquiries that cover a range of wine tastes and practices related to online wine purchase. For each question, participants must pick one from a prepared list of options. Age, preferred wine website, difficulty when surfing wines only, wine type preference, importance of price range, wine ratings, importance of wine type, region, vintage, producer, and grape variety in wine selection, frequency of wine purchases, and preferred method of wine discovery are among the topics covered by the questions. The survey was conducted between April 1 and May 1 of 2023 and is intended to be finished in around 10 minutes.

2.1.2 Key insights

Once the survey comes to an end, Excel tool is used to handle and analyze the data. The analysis include developing visuals, computing descriptive statistics, and drawing insightful conclusions from the survey data. The survey results provided some insights into the preferences and expectations of the target audience. First, figure 2.1 is a chart which conducted from the survey about people's grading (on scale of 1 to 5 points) about choosing wines based on characteristics. As can be seen from figure 2.1, the majority of wine buyers fell the need to investigate about details when browsing for wines.

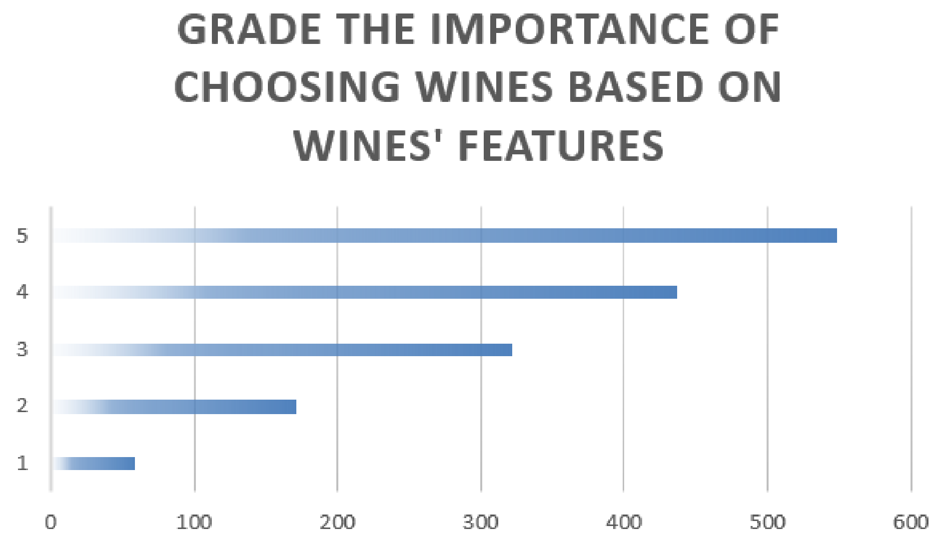

Figure 2.1: Importance of choosing wines based on their details

Second, figure 2.2 presents the fascinating results of a survey that was carried out to comprehend how wine consumers normally discover new wines. The survey's findings reveal an interesting pattern among respondents, with a sizeable percentage indicating a strong desire to locate wines that closely mirror their preferred or suggested options in terms of look or flavor profile. The potential importance and popularity of the "Find Similar Wines" capability that this graduating research seeks to build are highlighted by this propensity toward similarity-driven wine exploration. The poll also shows that a sizable portion of participants are willing to experiment with new and varied wine alternatives from various sources. These people are anxious to branch out from their typical wine preferences and look for uncommon encounters at various establishments including eateries, bars, pubs, or tiny wine wholesalers. This feature highlights how crucial it is to offer a platform that not only caters to individualized wine suggestions but also makes it easier to find new and intriguing wines to suit consumers' varied tastes and preferences. The results of the poll provided insight on how wine consumers' requirements and tastes are changing, highlighting the need of providing a wide range of wine options as well as individualized suggestions. The "Find Similar Wines" tool is a crucial addition to the platform since it can meet the growing need for individualized and varied wine selections by utilizing this information. The report also emphasizes how crucial it is to provide customers with an interface that is both aesthetically pleasing and easy to use. Given that a sizeable portion of respondents said that they were interested in wines based on their look, it is clear that offering high-quality wine photos and a smooth user experience are essential components in luring and keeping clients. Users will be captured by a visually appealing platform that highlights wines with gorgeous pictures and in-depth explanations, inspiring confidence in their wine selections and developing trust and commitment to the platform. Reiterating the relevance of the "Find Similar Wines" capability and the value of a wide range of wine selections, the survey findings give insightful information about the tastes and expectations of wine purchasers. By making use of this information, I can design a user-centric platform that satisfies consumers' needs for tailored, eye-catching, and varied wine suggestions, assuring a fun and satisfying wine discovery experience for every user.

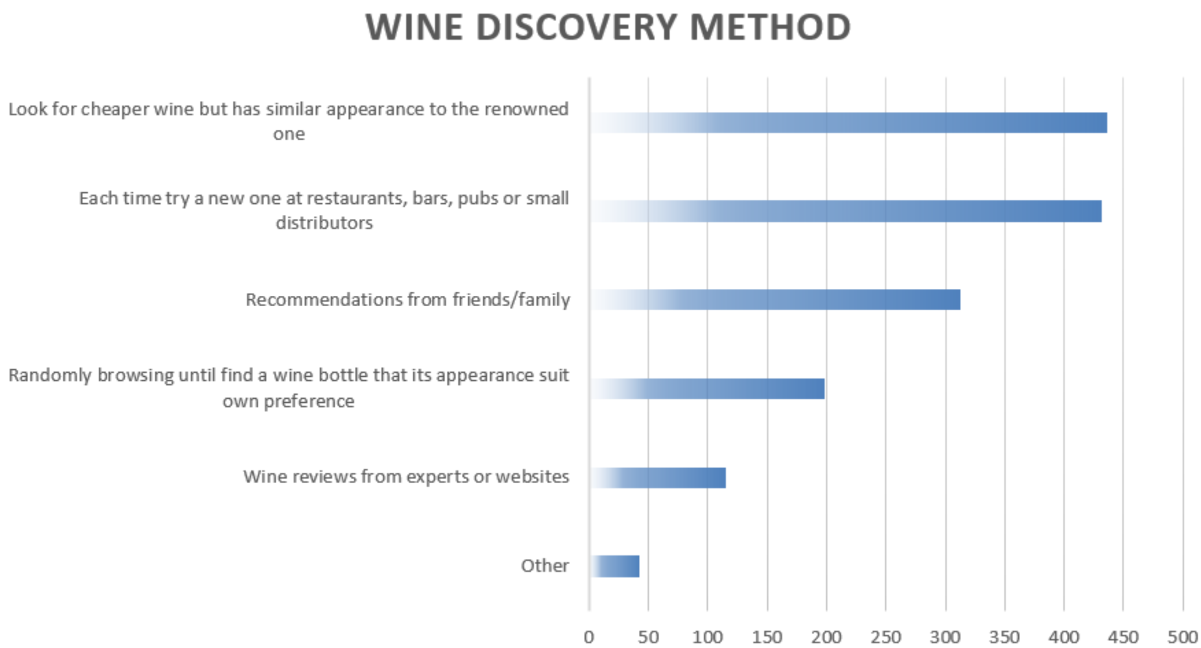

Figure 2.2: Wine discovery method

Thirdly, we explore the essential issue of whether users are more interested in a wine online store website that boasts a user-friendly layout in figure 2.3. The data visualization paints a clear picture, showing that the majority of wine consumers do in fact give this a lot of thought. The majority of participants expressed agreement with this remark, which highlights the significance of a well-designed and user-friendly user interface in luring and keeping clients in the cutthroat online wine industry. The results of this poll clearly imply that a user-friendly design is crucial in determining how users interact with a wine merchant website as a whole. Customers are more likely to be happy with their online purchase experience when they can simply browse the site, explore various wine alternatives, and access pertinent information with ease. Increased consumer loyalty and repeat business as a result of this great experience can help the wine-selling platform succeed and endure. Higher engagement rates can also result from a user-friendly interface. Customers explore different wines and interact with the available features as they spend more time on the site, which opens up potential for the platform to present tailored suggestions and promotions. By making customers feel appreciated and understood, this degree of customisation improves the whole purchasing experience, which eventually leads to increased customer satisfaction and brand loyalty. A website's user interface, on the other hand, might discourage visitors from making purchases or even utilizing the platform further. Customers may search for other platforms with a better user experience if they have a bad experience with the website's navigation, loading speeds, or information. The study results and the industry's increased emphasis on user-centric design make it clear that an online wine retailer's decision to invest in a user-friendly interface is of the utmost significance. Wine retailers can create a welcoming online shopping experience that draws and retains customers, distinguishes them from rivals, and ultimately helps them achieve long-term success in the fast-paced world of e-commerce by continuously improving the user interface, taking into account customer feedback, and staying up to date with the most recent design trends.

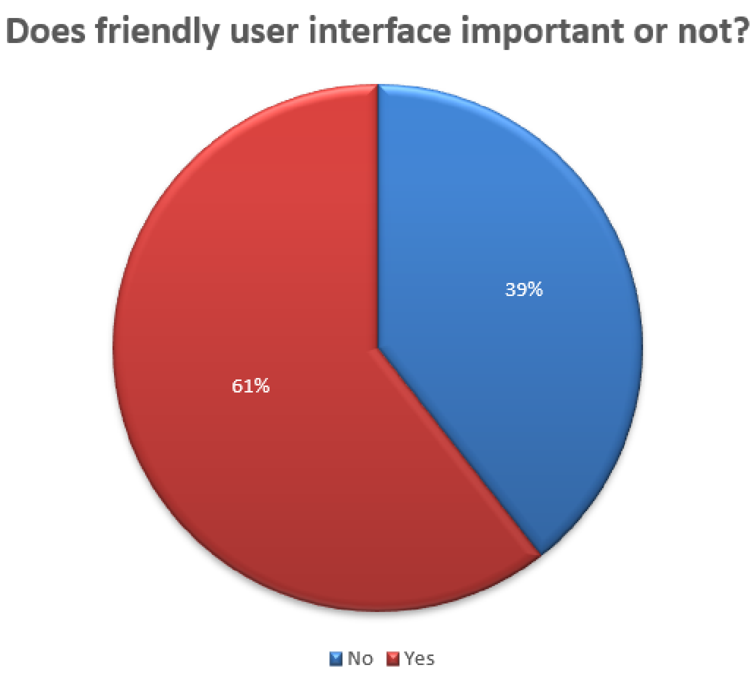

Figure 2.3: Does a friendly user interface matter for wine website?

In conclusion, people tend to decide buying wines based on wines' characteristics, the similarities between them and wine enthusiasts' recommended or preferred one. Not only that, they are more likely to be attracted to an online wine website with friendly UI.

2.1.3 Feature prioritization

Since the key insights have been concluded from **??**, Detailed wine data, user-friendly interface, and customized wine suggestion system were the primary criteria that are prioritized based on their influence on user satisfaction. A detailed wine database is required to start since it is crucial to provide accurate and complete information about each wine, such as the appellation, producer, region, country, vintage, wine type, grape variety, alcohol level, volume, and price. Users place a great value on having access to detailed information so they may make wise purchase selections. The next most important factor is a user-friendly interface since most respondents underlined how important it is to have a platform that is simple to use and improves the entire purchasing experience. Tailored wines suggestions system is the last. It is critical to put in place a system that makes customized wine suggestions based on customer preferences, previous purchase history, and taste traits. This feature has the potential to increase user pleasure, interest and stimulate curiosity.

2.2 Analyze existing wine-selling platforms

To fully grasp the advantages and disadvantages of the current wine-selling platforms, I shall analyze them in this section and try to find significant components that can improve and enrich this graduation research by comparing and assessing the features and capabilities provided by these rivals or comparable platforms. There are several wine selling platforms available today, each with its own special features and services. These platforms' broad assortment of wines from various producers and areas is one of its key advantages since it gives clients a wide range of possibilities to explore. Customers may browse the vast wine library with ease thanks to the user-friendly interfaces and smart search features that are offered by several platforms. However, certain wine-selling platforms can have some restrictions. For instance, some platforms might not have individualized wine recommendation algorithms, leaving users to make choices based only on the most fundamental filters and categories. Users may have a less personalized and engaging user experience as a result, perhaps missing out on possibilities to boost client retention and satisfaction. To enhance the training and effectiveness of other platforms' recommendation systems, I will also look at how other platforms handle the data collection, storage, and data augmentation procedures. To get the most effective and efficient outcomes, I may improve and optimize our own data preparation and model training approaches by studying their tactics.

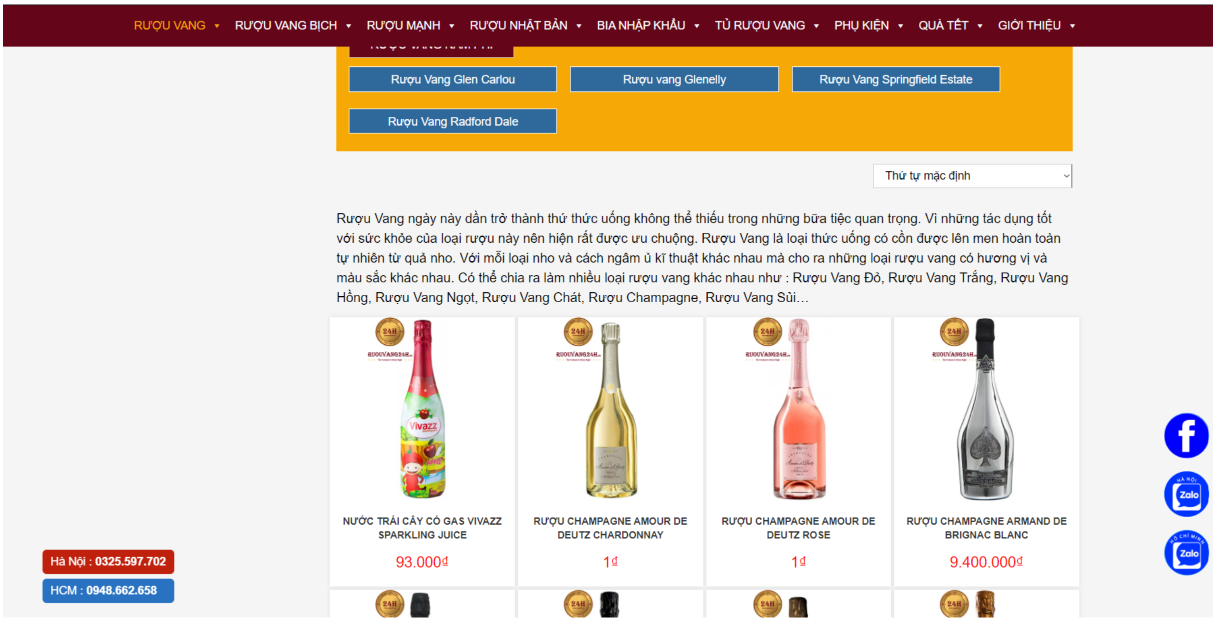

Figure 2.4: ruouvang24h.com

The first website under consideration is "ruouvang24h.com," which was established in 2010 with the goal of providing the Vietnamese market with a wide selection of imported wines from different nations, regions, and producers. This platform's reasonable pricing is one of its key advantages, giving clients access to wines that are less expensive than the typical market price. The platform also features a sizable assortment of wines, providing a wide range of options to suit the interests of various wine connoisseurs. Additionally, "ruouvang24h.com" stands out for its broad selection of imported wines from many nations, regions, and producers, offering clients the chance to experience wines from other places of the world. This wide variety adds to the platform's attractiveness and makes it a desirable option for clients looking for a varied wine purchase experience. Nevertheless, "ruouvang24h.com" has several shortcomings despite its advantages. The user interface is one of the obvious drawbacks and could not be as user-friendly as expected. The platform's navigation might be enhanced to give users a more natural and smooth experience while looking for wines. A disadvantage might also be the lack of a reliable recommendation system for novices. For users, especially those who are unfamiliar with the world of wines, a well-designed recommendation system might make the process of choosing a wine easier. The technology might increase user engagement and encourage client loyalty by offering personalized suggestions based on individual interests. "ruouvang24h.com" has gained notoriety for its price and wide range of wine offers when compared to other platforms. However, improving the platform's user interface and putting in place a suggestions engine could improve user appeal and convenience.

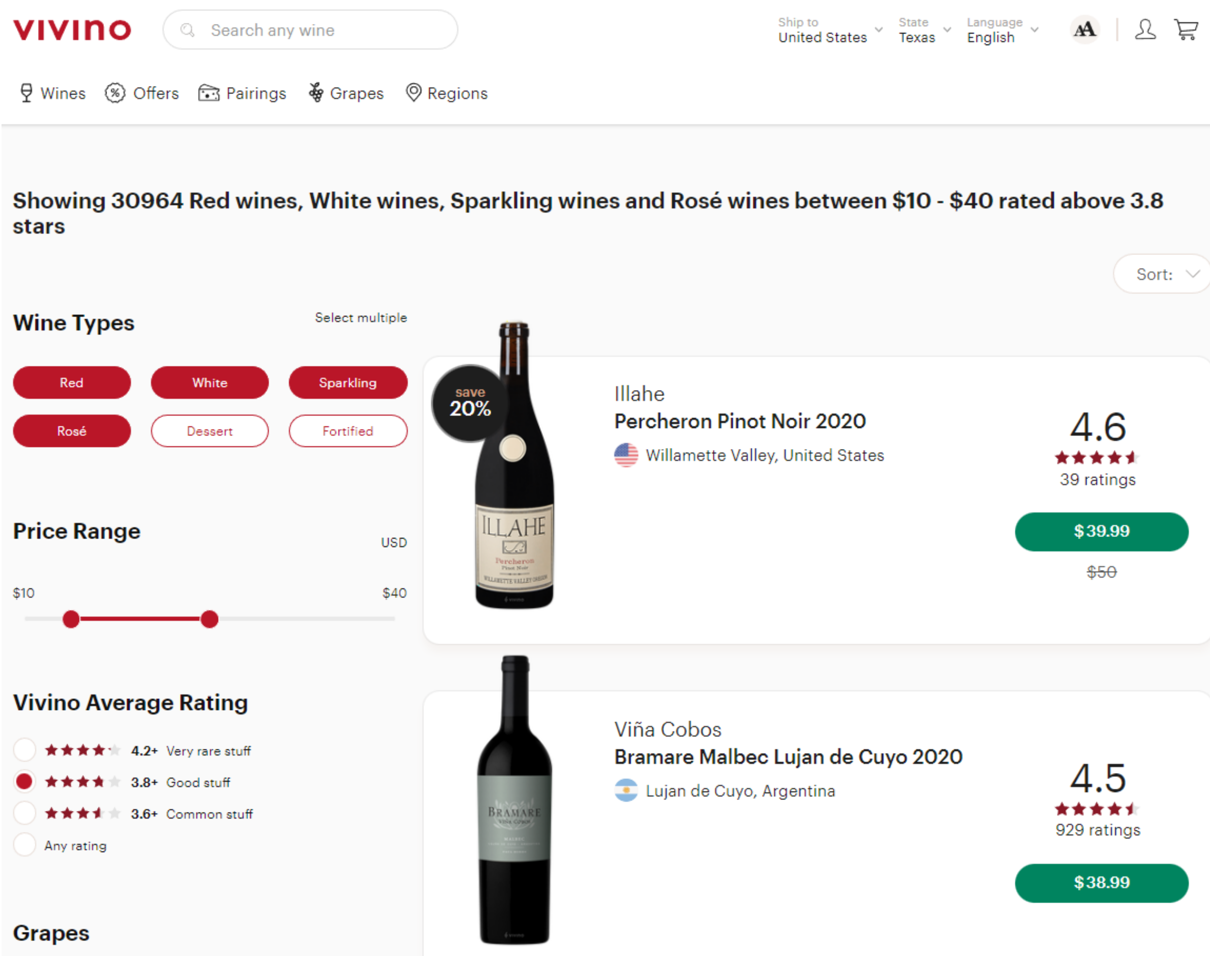

Figure 2.5: vivino.com

Moving on to "vivino.com," which was also introduced in 2010 and has established a reputation as having one of the biggest collections of wine reviews anywhere in the globe. The extensive wine database on "vivino.com" is one of its main benefits since it enables users to research and find new and unusual wines. Users are given the ability to make knowledgeable judgments based on the experiences and suggestions of other wine aficionados thanks to the platform's extensive collection of ratings and reviews that are provided by wine consumers themselves. The lively wine community on "vivino.com" is one distinctive feature that makes it stand out and promotes a sense of community among wine consumers. Users' active participation in the community improves the entire experience as wine lovers offer their knowledge, advice, and personal stories, creating a lively environment for wine exploration. A useful barcode scanner tool on "vivino.com" enables users to quickly obtain reviews and ratings by just scanning a wine bottle's barcode. While purchasing wines in physical locations, consumers may quickly and easily obtain pertinent information and insights, which helps them make better decisions.

Like every platform, "vivino.com" does, however, encounter some difficulties. The precision and dependability of user-generated assessments are one possible problem. The reviews' consistency and quality may vary because they were submitted by people with a range of interests and preferences. When making their wine selections exclusively based on user-generated material, users should use caution. Furthermore, even if "vivino.com" provides excellent services and features, some cutting-edge capabilities could only be available to individuals with a premium membership. For those looking for a more thorough wine-buying experience, the restricted access that free users may have to some premium services may be a worry.

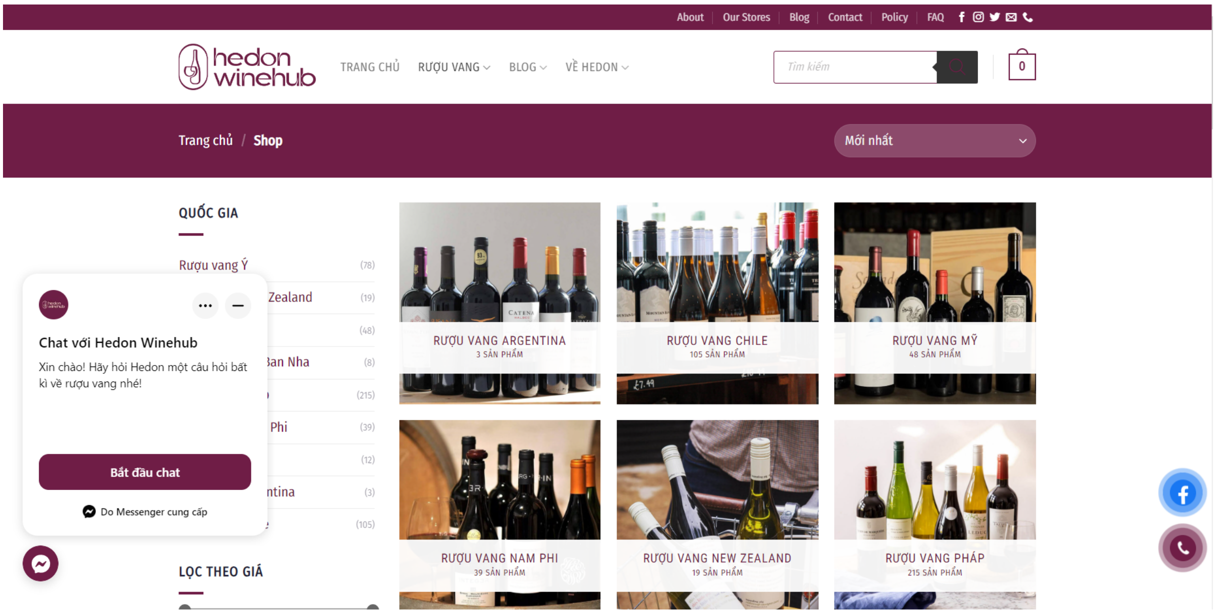

Figure 2.6: hedon.com.vn

The third site up for inspection is "hedon.com.vn," a well-known wine marketplace with a chatbot system as a standout feature. Hedon has been known for its pleasant user experience and support service made possible by the chatbot. The unique capacity of Hedon's chatbot to interact with clients in real-time, making tailored wine suggestions and responding to questions about various wine alternatives, is what sets it apart from other chatbots. The chatbot's cognitive algorithms examine user preferences and previous interactions to give personalized suggestions, simplifying and improving the wine selection experience. The user experience is improved by Hedon's user interface, which is designed to be simple to use and intuitive. Customers may easily explore a wide variety of wines from various nations, regions, and producers in order to select the ideal bottle that meets their preferences. However, the lack of picture input capabilities on Hedon's platform is a drawback. Hedon's platform only accepts text input and wine descriptions, in contrast to our project, which uses AI to identify wines based on photos. Customers who want a more visible and participatory wine discovery experience may find this to be a hindrance. In general, "hedon.com.vn" is a well-known wine-selling website recognized for its cutting-edge chatbot system, which improves user experience and helps consumers locate appropriate wine suggestions. A flawless shopping experience is guaranteed by its user-friendly UI. Although our product shines at this aspect, the absence of picture input capabilities limits the visual investigation of wines. The benefits of our project's image-based recommendation engine may be combined with "hedon.com.vn" website's chatbot functionality to create a cutting-edge platform that provides an interactive and tailored wine-buying experience.

2.3 Consolidation of functional requirements

In accordance with the acquired understanding from sections 2.1 and 2.2, the functional requirements for this graduation project have been summarized. In the beginning, the project's ability to match users' chosen wines with others is one of its main features. When "ruouvang24h.com", "vivino.com" and "hedon.com.vn", three existing wine-selling platforms, were analyzed, it became clear that neither platform had a strong recommendation mechanism for clients to be shown equivalent wines. By providing individualized and related wine recommendations based on the customer's tastes, such a tool might significantly improve the user experience. The ability to investigate and uncover unfamiliar wines that fit their preferences was a key point in the poll of potential consumers. By offering an artificial intelligence-powered "wine finder" tool, users can quickly locate wines that are comparable to a certain choice they enter, broadening their wine tastes and exposing them to alternatives they might not think about them. Additionally, from survey and the analysis of both websites, it is ideal to include the function of showing wines' characteristics such as appellation, country, region, price and description so that not only customers can based on those information when choosing wines but the staffs from restaurants, pubs, bars or small wine distributors can filter a buyer's desired wine easily. In conclusion, the requirements that are summarized from the survey and analysis are a wine recommendation system and a function that can provide wine information such as appellation, country, region, price, and description.

2.4 Non-functional requirements

In order to guarantee the general efficacy and sustainable existence of this graduating project, certain non-functional standards are outlined in this section. Performance-wise, the program must be fast and is capable of managing several many users at once without experiencing any noticeable lag or slowness. In order to provide users with a smooth shopping experience, reliability demands that the system run consistently, without frequent interruptions, and without downtime. Usability is a crucial factor that requires an intuitive and user-friendly interface that accommodates users with different levels of technical competence, promoting simple navigation and effective job completion. The program should be created with clean, modular code for maintainability, facilitating updates, bug repairs, and additions. The whole thing must be scalable in order to handle upcoming expansions and increased demand without jeopardizing performance or dependability. In summary, To ensure project efficacy and sustainability, the program must be fast, reliable, user-friendly, and scalable. Moreover, it should run consistently, maintain clean code, and handle expansions without jeopardizing performance.

2.5 General use case diagram

In this section, a general use case diagram is shown to demonstrate the key features of this graduation project's website.

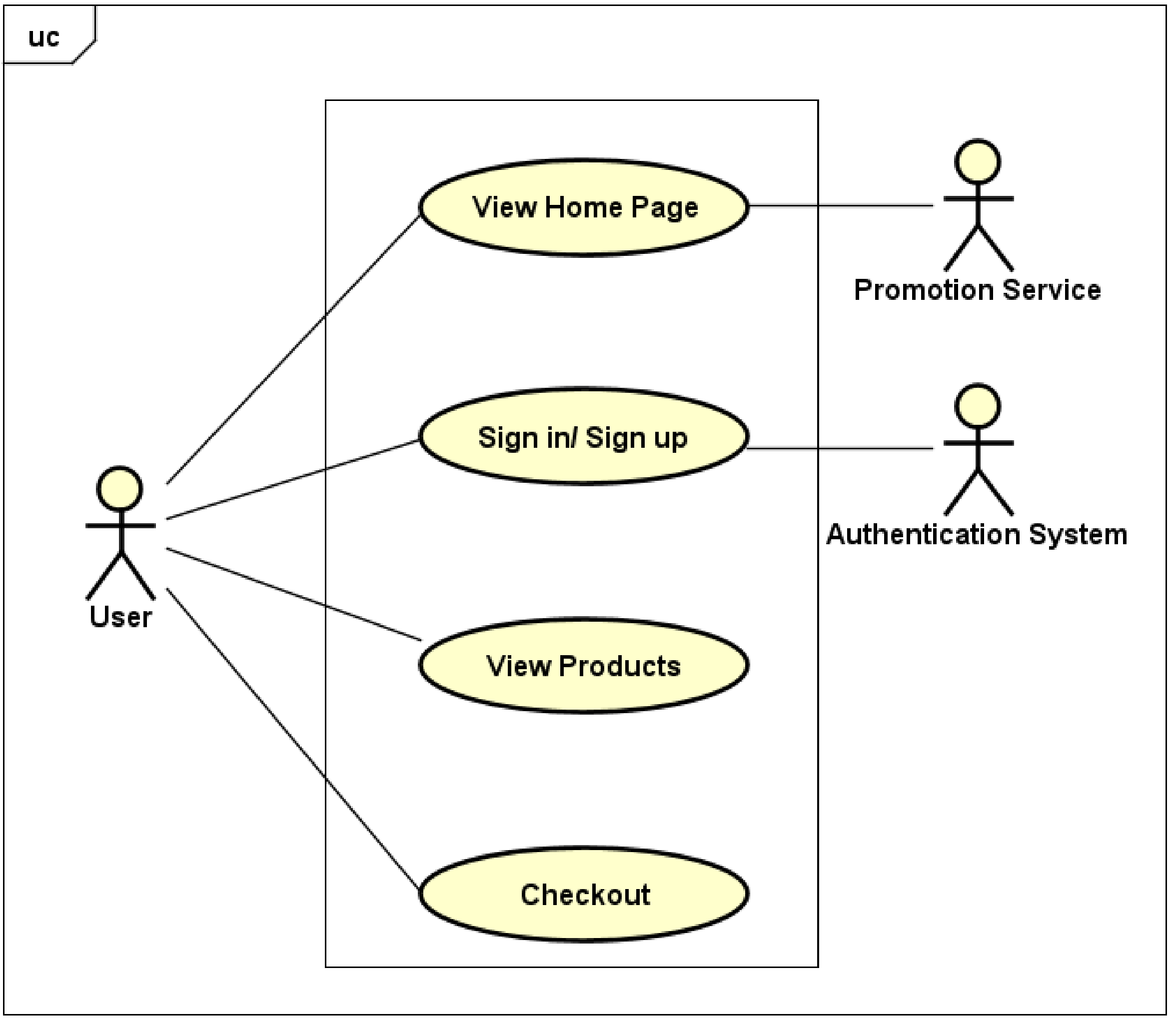

Figure 2.7: General Use Case Diagram

Create a high-level Use Case diagram to represent the major functionalities of this graduation project's application."User," "Promotion System," and "Authentication System" are the three main players in the system. The four basic interactions between the "User" actor and the program are "View Home Page", "Sign in/Sign up", "View Products," and "Checkout." The "view Home Page" function enables the user to go to the application's homepage and browse the most recent discounts and wine. A user can register for an account or log in using the "Sign in/Sign up" function. Users may browse through the vast selection of wines offered on the site by selecting "view Products" after clicking the "Products" button. The "User" may then continue to the payment process to complete the purchase of their chosen wines by selecting the "checkout" step. The "Promotion System" interacts with the "User" by presenting promotional offers, discounts, or special deals depending on the user's preferences and past purchases, as well as on a daily basis, special events, and other factors. The "Authentication System" is in charge of handling user authentication and making sure that the platform is accessible securely. Overall, this application's key operations and features are represented in this general use-case diagram, which also highlights how users may browse, choose, and buy wines while also taking advantage of tailored promotions and a safe authentication procedure.

2.6 Detailed use case diagram

In this section, I'll give an outline of the main interactions and purposes of the application of this master's thesis in this part. The "View Home Page" option enables visitors to browse the website's opening page, which features current specials, newly released wines, and best-sellers. The "Sign in/Sign up" process gives users the opportunity to log in using pre-existing accounts or establish new ones, assuring a rapid and safe authentication process. By providing choices like "Show More," "View Product," "Add To Cart," "Filter Wines," and "Find Similar Wines," the "View Products" use case improves the user experience while supporting easy wine browsing and discovery based on personal preferences. Customers may evaluate the items in their shopping cart, submit personal information, and finish the payment process using the "Checkout" use case, which streamlines the purchase of desired wines. These extensive use cases add to the application's overall efficacy and efficiency, resulting in a seamless and pleasurable experience for all users.

2.6.1 View Home Page

No matter if users have registered or not, they can see the website's main page, which is referred to as the "View Home Page" use case. The homepage offers a wide variety of parts that improve users' surfing experiences and is created to adapt to their interests and preferences. Next, "Daily Deals" features exclusive offers and reductions on particular wines. This promotes client loyalty and engagement by incentivizing users to browse the platform frequently and take advantage of tempting offers. Additionally, users get direct access to the most recent additions to the platform's wine library through the "New Wine Arrivals" feature. Another part is "Best-Sellers", which compiles a list of the wines that customers find to be the most well-liked and highly rated. The website makes use of social proof by displaying these in-demand wines, enticing users to investigate and choose wines that have garnered favorable reviews from other consumers. The "Top Picks" area of the webpage makes use of a recommendation engine which targets mainly on analyzing the top most purchase products. By making this kind of wine recommendations, users are continually exposed to the newest styles and top wine choices thanks to the system's frequent changes to this part, which ups the excitement and novelty factor of their wine research adventure. In summary, the "View Home Page" use case of the platform used for this graduation project provides all users with an interesting and educational experience. The portal improves user happiness and allows easy wine discovery by carefully utilizing dynamic sections like "Daily Deals," "New Wine Arrivals," "Best-Sellers," and "Top Picks." Users may easily browse the platform's most recent wine offerings and suggestions, regardless of whether they are registered members or first-time visitors. This creates the ideal environment for an enjoyable and fruitful wine purchasing experience.

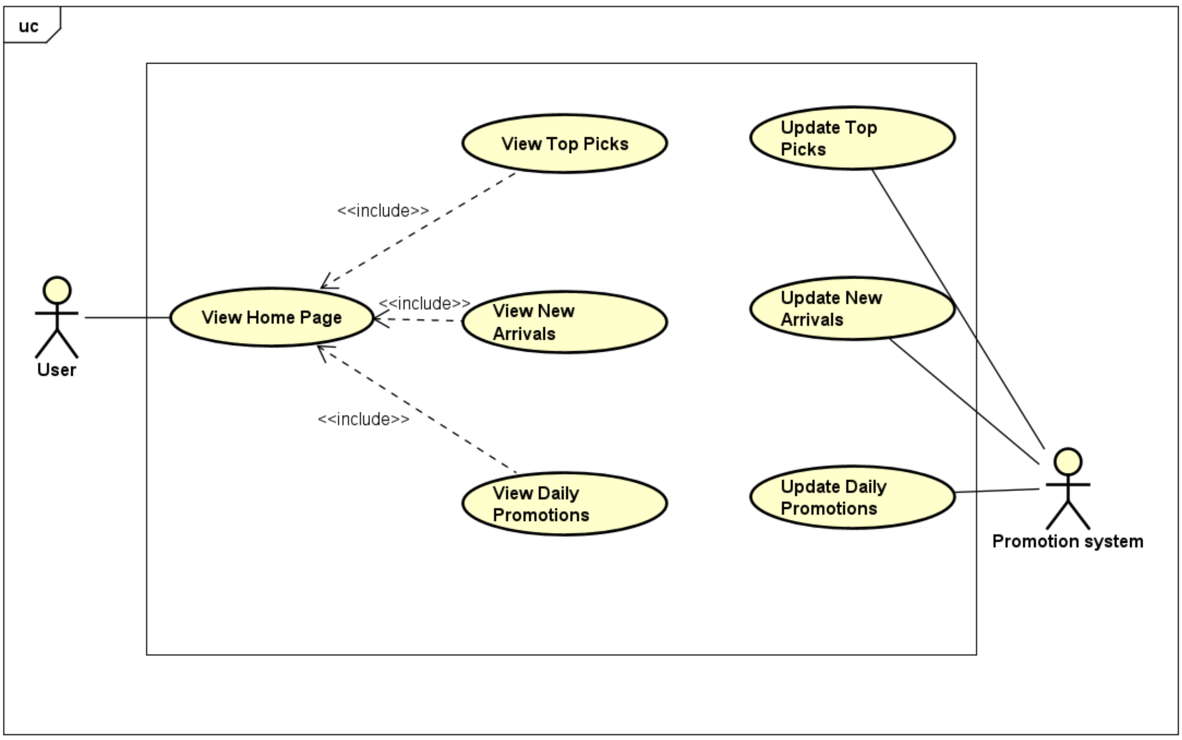

Figure 2.8: Specific Use Case Diagram About "View Home Page" action

2.6.2 Sign in/ Sign up

In the "Detailed Use Case Diagram," the "Sign in/Sign up" use case, users are given the choice of signing in with an existing account or creating a new one. Users are requested to enter their email address and password, which are the two necessary areas for verification, when they access the login page. On the other hand, users must complete the username box on the sign-up page if they decide to create a new account. This distinct division between logging in and signing up makes for a simple and easy user experience. The authentication system carries out a rigorous verification procedure to guarantee the security and accuracy of user data. It verifies that the entered data is in the appropriate format and alerts users to make the required modifications if any errors or missing data are found. This proactive method of data validation improves user satisfaction by preventing potential mistakes and guaranteeing that all necessary fields are filled in correctly. The entire user experience throughout the sign-in and sign-up processes is considerably enhanced by the use of such an authentication system. Users may confidently enter their data because they know that the system is carefully checking and protecting it. Users are more likely to participate more actively on the platform because they feel safe and in the right. Additionally, the authentication mechanism is crucial in protecting user accounts and sensitive data. The danger of unwanted access or data breaches is reduced by requiring adequate verification of email addresses and passwords. Users can trust the platform with their data because of this, which helps to create an overall strong and dependable system. Additionally, the system's capacity to recognize and correct input mistakes in real-time speeds the sign-in and sign-up procedures, lessening user annoyance and the possibility of abandoned registrations. This improved user experience increases user retention and adds to a favorable opinion of the brand. In conclusion, the "Sign in/Sign up" use case's authentication process offers a thorough and safe method of user verification. The technology guarantees a quick and gratifying user experience by authenticating user data in real-time and keeping a distinct line between signing in and signing up. Additionally, the platform's emphasis on data security strengthens user confidence and trust, making it a desirable and reliable option for customers looking for smooth wine browsing and purchase experiences.

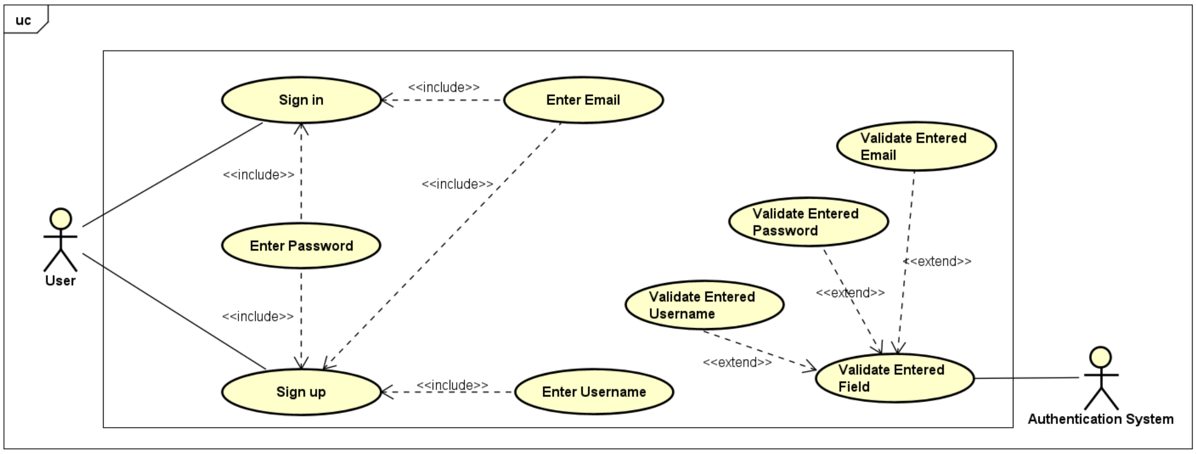

Figure 2.9: Specific Use Case Diagram About "Sign in/ Sign up" action

2.6.3 View Products

Visitors are brought to the products page in the "View Products" scenario, where they can browse a sizable inventory of various wines. A variety of features are available to improve the user experience overall, offering clients ease and flexibility during their wine research trip. In the beginning, the "Show More" option gives users the ability to increase their range of choices by instantly presenting an extra 10 wines in the catalog. Users may easily search through a larger variety with the help of this function, making sure they don't pass over any promising wine options. Next, the "View Product" action enables consumers to learn more about certain wines of interest. Customers may get thorough information about a wine, such as its description, producer, area, and vintage, by clicking on it. Through the provision of useful information, this feature enables consumers to make knowledgeable judgments regarding their wine selections. The site also provides a user-friendly "Add To Cart" button that makes it simple for customers to continue with their desired wine purchase. Users may quickly and easily add their chosen wines to the basket, expediting the purchasing procedure and increasing customer happiness. Additionally, buyers may narrow down their wine selections using the "Filter Wines" tool according to particular standards like vintage and cost. This feature guarantees a more precise wine selection that takes into account personal preferences and tastes. Similar to this, the "Search Wines" feature gives users the opportunity to look for wines depending on producer and country, offering a more specialized and customized wine browsing experience. Not to mention, the platform's flagship feature, "Find Similar Wines," completely changes how people discover new wines. Users may easily click the "Explore Similar Wine" option after uploading an image of a wine they like. With the help of the AI model, the system then offers a curated list of wines that are thought to be comparable to the provided image, making it easier to find wines with similar characteristics. Users may have a flawless and engaging experience thanks to its cutting-edge technology, which encourages them to discover new and delicious wine alternatives. In summary, users may easily browse the extensive wine library, choose wines that suit their interests, and make wise selections during their wine journey by integrating these simple functionalities into the "View Products" use case. This graduating project's platform allows users to fully immerse themselves in an engaging and pleasant wine discovery experience because to its thorough and user-centric approach, which also increases the platform's attractiveness.

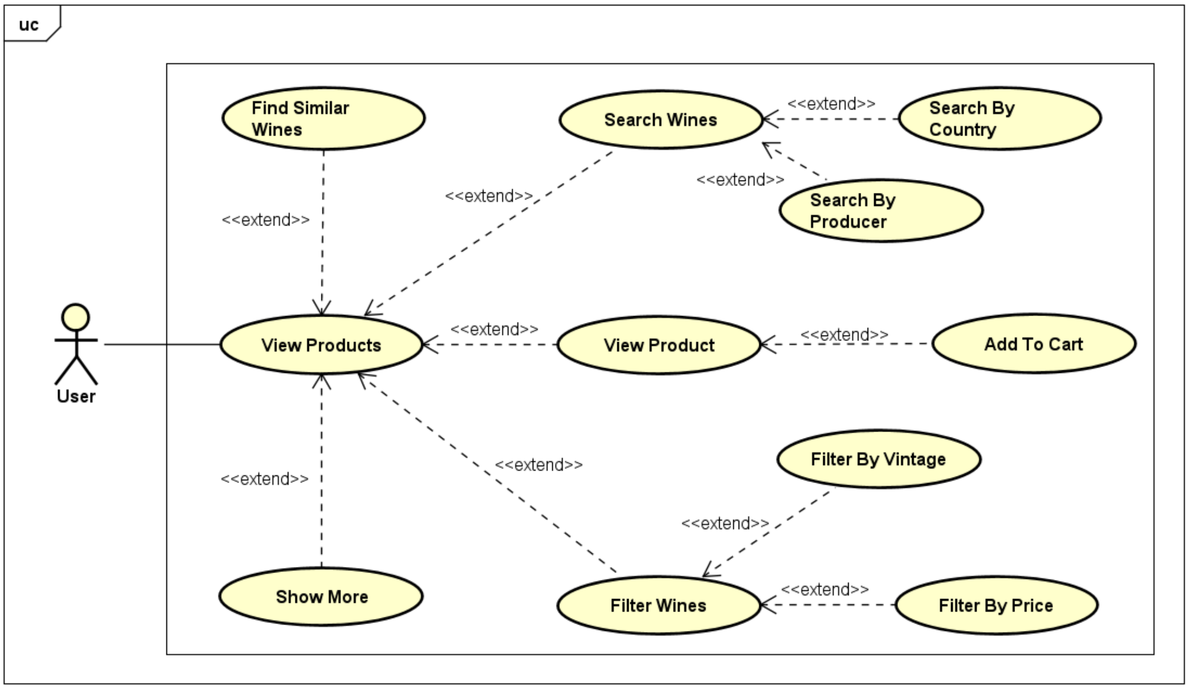

Figure 2.10: Specific Use Case Diagram About "View products" action

2.6.4 Checkout

After customers have browsed wines and added desired goods to the basket, they may finish the purchase process by using the "Checkout" use case in the "Detailed Use Case Diagram" section. When a user is prepared to make a purchase, they choose the "Checkout" option, which starts the process of selecting the required wines. The "View Cart" action is then carried out, giving the user a summary of the products that have been put to the cart. The user may analyze the chosen wines in this view, make any required adjustments, and even eliminate things if necessary. The user is asked for personal information, such as their complete name, contact information, and address, before completing the payment. The handling of the purchase and effective delivery of the chosen wines to the user's provided address both depend on these details. The user completes the checkout procedure and moves on to make the payment to complete the purchase by inputting these information. In order to give users of this graduation project's application a seamless and safe purchasing experience, the "Checkout" use case is essential.

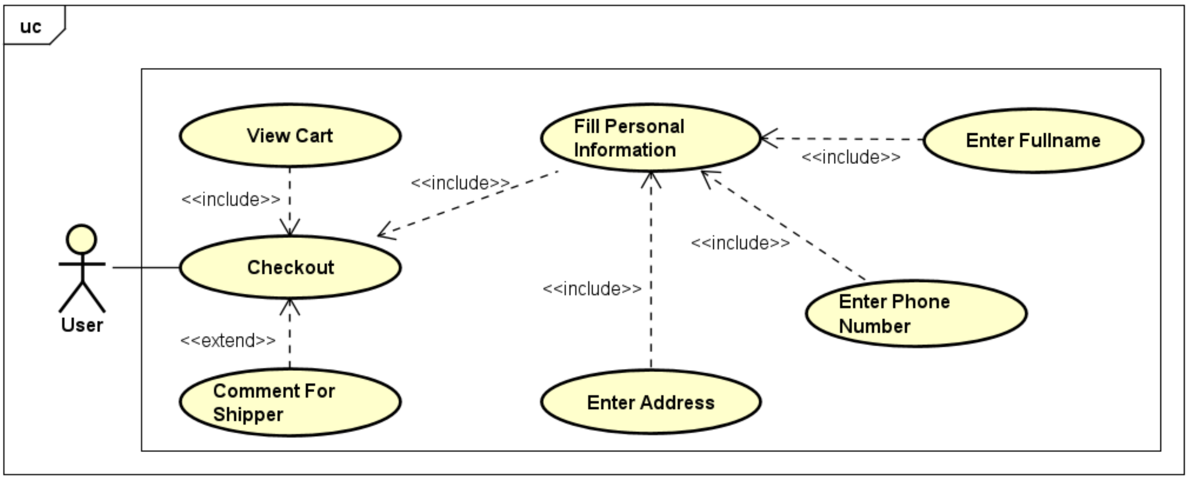

Figure 2.11: Specific Use Case Diagram About "Checkout" action

CHAPTER 3. METHODOLOGY

3.1 Overview

In this chapter, I will address the data gathering, processing and augmenting methods, deep learning model and evaluation metrics that are used to complete my graduation project. To begin with, Selenium web scraping was used to collect wine photos from "ruouvang24h.com" during the gathering process. The collected data is then processed by deleting records with null or non-number values and adding new columns for simpler data management such as "image_filename", "description", and "wine_id". Following that, to increase the durability of the dataset, which can benefit the future model, a variety of image augmentation techniques were applied using the imgaug package, involving flipping, affine transformations, Gaussian blur, linear contrast adjustments, additive Gaussian noise, and sharpening procedures. Moving into the training model step, I decided to take advantage of a deep learning model architecture, called ResNet50 model, fine-tuned using transfer learning, and accurately classified 75 producers. After that, to examine the model's trained result, I have used several evaluation metrics, involving accuracy, precision and recall score. To experience the model's ability by using FastAPI to build an API so that it can be applied to a simple website that has been built using NextJs, TailwindCSS, and Typescript. In summary, this chapter focus on analyzing the applicability, efficiency, and significance of different techniques for tasks provided in the previous chapter.

3.2 Data gathering, processing and labeling step

3.2.1 Data gathering

In section 3.2.1, I will address the technology used in gathering data step which is Selenium. First of all, Selenium is regarded as a group of free tools well known for browser scripting. A variety of languages for programming, especially Python, which was my selected language, are supported by this framework. With the existence of WebDriver protocol, Selenium have the ability to connect to Google Chrome, Microsoft Edge or many other web browsers. Not only Selenium is a renowned for assessments and end-to-end evaluations, but it also transformed into a crucial tool for crawling and scraping from specified websites. Not many methods remain as efficient as interacting with webpages via a working browser. Following that, Selenium proposes many different site interacting features, for instance, hitting buttons, inputting information into forms, navigating sites, taking snapshots, and running customized scripts. However, one of its greatest benefits is its capacity to navigate webpages with the same ease as a standard browser. This functionality is very useful for extracting information from Single-Page Websites. In summary, Selenium is preferred since not only it offers robust control of browsers, smooth management of web browsers for web crawling, web scraping, and end-to-end testing, but it also supports Python and ensure reliable data gathering, including material from websites with a lot of JavaScript.

3.2.2 Data processing and labeling

[1] Following the information gathering phase, the collected data is then processed via simple lines of code using Python. We can quickly prepare and turn the unstructured data into a format which is meaningful thanks to Python's flexibility. In this, addressing missing values, formatting different data formats, and running statistical analysis are all included. In order to maintain data integrity, the actual processes entailed eliminating any "null" or incomplete records. I also added useful fields to the dataset which are "image_filename," "description," and "wine_id." These enhancements offered crucial data that enabled additional investigation and model training. After that, the dataset will need to be labeled so that the model training phase can go smoothly. I chose MakeSense Ai to accomplish this task since it is a free online website, it supports multiple label types and it can export label file as YOLO, VOC, XML, VGG, JSON and CSV formats. In short, data is processed using Python for meaningful formatting, addressing missing values, statistical analysis, adding additional columns and is labeled using MakeSense Ai for smooth model training.

3.3 Data augmenting step

[2] The next step in the effort to improve the efficiency of the deep learning model was data augmentation. This step is crucial for ensuring that the model generalizes successfully to new data by reducing the risk of overfitting and diverse the dataset. The "imgaug" software was chosen for this task partly because of its wide range of image augmentation methods, which made it the best option for enlarging the wine dataset. In order to add variability to the dataset, an augmentation sequence was first painstakingly created. It included a number of picture processing procedures. Vertical flipping of the images, affine rotations with angles ranging from -10 to 10 degrees, Gaussian blurring with sigma values between 0 and 1.0, linear contrast adjustments in the range of 0.8 to 1.2, adding Gaussian noise with a scale factor from 0 to 0.05 multiplied by 255, and sharpening with alpha values varying between 0 and 0.5 were the augmentation techniques that were selected. The wine dataset saw a remarkable increase as a result of the application of this augmentation process, growing from its initial count of 726 photos to an en-

hanced dataset consisting of a mind-boggling 16,212 images. With a wide variety of lighting setups, perspectives, and transformations, the wine samples were now more fully represented in this enormous collection of enhanced photographs. As a result, the deep learning model could use this expanded dataset to enhance its capacity to identify and categorize wines effectively, even in the presence of a variety of circumstances. In conclusion, the procedure of data augmentation was crucial in improving the effectiveness of the deep learning model. The danger of overfitting was significantly decreased by diversifying the dataset and providing unpredictability through multiple picture alterations. The "imgaug" package, which included a variety of image augmentation techniques, turned out to be a tremendous help. It enabled the model to be built on a more solid and thorough base for training and assessment. The deep learning model was ready to show enhanced accuracy and generalization with the dataset now included 16,212 augmented photos, making it more suited to handle the many wine options and subtleties found in the real world.

3.4 Creating a deep learning model for wine classification

3.4.1 Data Preparation and Loaders

[3] Before moving to the model building and training step, I must first preprocess the augmented wine data and set up the data loaders for training, validation, and testing. A total of three sets of the dataset were created: one for training (80%), one for validation (10%), and one for testing (10%). The data loaders offer a useful method of feeding data into the model for training and testing.

3.4.2 Background about ResNet-50

[4] To begin with, through several studies, it has been believed by many researchers that the deeper a convolutional neural network is, the better results it can produce. This is due to the fact that when facing a problem where a neural network needs to explore a large number of parameters, a deeper neural network will be preferred. However, if the number of layers is large enough, at some point when training the neural network model, the gradient will be rounded to zero because it is considered too tiny. Thus, to avoid this issue, a concept has been found: instead of learning sequentially, the model just needs to learn the residual, or the difference between the input and outermost output of each block (to generate the desired output, we need to complete the addition of the original input and the transformed output). This is when a deep learning architecture called ResNet (Residual Neural Network) was developed, which inherited the mentioned concept to solve the problem where gradients keep vanishing in very deep neural networks. Using this knowledge, scientists developed an approach for "residual blocks," which featured

a "skip connection" method that enables CNNs to link directly to deeper or shallower layers while avoiding some levels right after the current ones. To achieve this, we can try to acquire the result of the combination of input and transformed output. By including residual blocks that allow for the training of much deeper networks with better performance, ResNet overcomes this issue. To summarize, the ResNet architecture was invented to overcome the drawback of very deep neural networks, which is the gradient vanishing problem.

Following the mentioned original idea and developed architechture above, I will illustrate about the concept of the ResNet-50 architechture which is inherited and upgraded from original ResNet. First, I will depict each convolutional layer's size and contents in ResNet architechture using the table from research paper about "Deep Residual Learning for Image Recognition" which was conducted by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3.1: Summary of each convolutional layer's size and contents

The input image usually has a size of 224x224x3, where the resolution is 224x224, and the number of color channels (RGB) is 3. From figure 3.1, the first operation, as can be seen, consists of a convolutional layer comprising adaptive filters which its mission is to investigate the input and extract features that are considered essential. In ResNet-50, the first convolutional layer includes 64 filters using a kernel of a modest size, commonly 7x7, with a stride of 2. The feature maps' spatial dimensions are therefore reduced. Next, A max pooling layer with a 3x3 kernel and a stride of 2 is implemented to further reduces the spatial dimensions, improving the efficiency of the next computations. The ResNet-50 is then designed to have a total of 4 residual blocks. The network is capable of learning residual functions via the skip connections that each residual block has. In the first residual block, there are

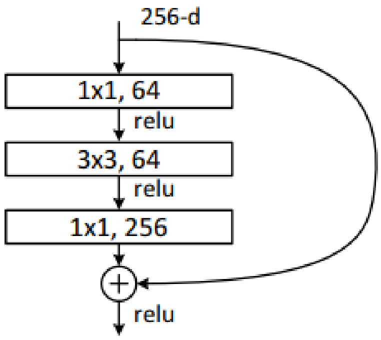

Figure 3.2: Skip Connection in the first residual block

3 convolutional layers in total, and they are applied three times in succession. The number of channels is decreased but the spatial dimensions are maintained by using the first convolutional layer, which has 64 filters. To futher extract features from the data, a 3x3 convolutional layer with 64 filters is then employed. To increase the number of channels, the last 1x1 convolutional layer with 256 filters is utilized in this block. Not only that, there is a skip connection in each residual block as seen in figure 3.2. The subsequent residual blocks have designs identical to the first one, although they have extra filters included into their convolutional layers. Following the last residual block, there is an average pooling layer applied in order to reduce the tensor's size to 1x1x2048. Following that, a fully connected layer is entrusted for mapping the feature vector with the total amount of classes in the dataset. The model's final predictions are represented by the output layer, which also provides probabilities for each class involved in the classification process.

3.5 Transfer learning technique

[5] From section 3.4.2 about ResNet-50 architechture, its ability to classify up to 1000 labels has been described. However in this graduation project, its limit is only 75 labels which consists all wineries from the crawled dataset. Thus, in order for this architechture to work with this problem, I have used a technique called "Transfer Learning". Transfer learning is the approach that utilize the obtained parameters from a trained model on one specific dataset (for instance, ImageNet classification dataset) and apply them to solve another problem. This approach is considered to be appropriate when it come to solving problems with limit time, resource and data size. There are several steps that need to be taken into account to apply the transfer learning technique. The initial step is to choose a pretrained model such as ResNet-50, VGG-19, or Word2Count that is suitable for solving the current problem. The base model creation step comes next, which is either to download the model architecture or build it from scratch. Following that, the layers before the fully connected

layer need to be frozen to avoid losing all trained parameters. After that, the model needs to be trained on a specific task, extracting important traits from the data while preserving the information discovered during the original training. In some situations, we might also need to alter some layers right before the fully connected layer, so it is advised to unfreeze some of them to fine-tune them so that the model can adapt to the new specific pattern of the data. In short, transfer learning allow solving problem with less required time, resource and data size by adjusting the layers usually starting from the fully connected layer or some layers right before it.

3.6 Model evaluation metrics

[6] Moving to the next step after defining, fine-tuning and training the model, in order to entrust that trained model into actual job in real life, it will need to be evaluated through some metrics. In this graduation project scope, I only used accuracy, precision score & recall score concepts to accomplish this mission.

3.6.1 Accuracy

About the first mentioned concept which is the accuracy. Since this metric can display the proportion of successfully predicted occurrences to all of the dataset's cases, it is used to assessing a classification model's overall performance. With its ability, it is easier when come to demonstrate the trained model's effectiveness in accurately identifying examples across all classes. Its formula can be described as:

$$Accuracy = \frac{Total\_Correct\_Prediction}{Total\_Number\_of\_Instances} * 100\%$$

One example that can describe this concept is if there is a situation where the trained model can predict correctly 8 out of 10 test cases then its accuracy is believed to be 80%. In conclusion, although accuracy concept might provide a grasp view of the model's performance but in some situations, we would need to consider another approach to determine whether or not the model is working properly.

3.6.2 Precision and recall score

Although the accuracy percentage can give us a grasp understanding about the model's overall effectiveness, its weakness still exists. One example for this will be when a test dataset has 1000 examples with 990 zero-label instances and 10 one-label instances. In this scenario, if a model predict 1000 examples which label is one then its accuracy is 99% and vice versa. However, it is not a wise choice to believe immediately that this model can function well enough due to the fact that it does not predict any example with label one. This is when the Precision and Recall score can help defining whether a model can be considered working properly or not. First, precision score is a metric that assesses how many true positive predictions are made out of all the occurrences that are true in the dataset. Its formula can be described as:

$$Precision = \frac{True\_Positives}{True\_Positives + False\_Positives}$$

For instance, when detecting spam emails, we would want to avoid classifying real emails as spam (false positives), as this could cause crucial messages to go unnoticed. A high precision indicates that the model has the potential to be highly accurate when its forecasted result is positive. Second, recall score can prove whether a trained model can have the ability to predict positives instances correctly or not. Its formula is:

$$Recall = \frac{True\_Positives}{True\_Positives + False\_Negatives}$$

As an instance, a model should try to prevent missing important diagnoses in a medical diagnosis system (false negatives), as this could have detrimental effects on the patient's health. Therefore, a high recall indicates that the model can successfully identify the majority of the positive occurrences in the data. Overall, there is frequently a trade-off between recall and precision. One can suffer if the other is improved. In contrast, a model with a high recall may have a low precision (captures many positive cases but may also have more false positives), while a model with a high recall may have a low precision (highly selective but may miss some positive instances). Therefore, we should utilize the combination of accuracy and recall at the same time (or F1-score) to determine if a model's performance on forecasting a certain example is accurate or not.

3.7 Creating python API with FastAPI

To begin with, FastAPI required the strategic selection of a cutting-edge and effective web framework that could produce scalable and dependable APIs as part of the process of developing a Python API. FastAPI, a Python web framework, was selected due to its remarkable skills in creating APIs that can easily handle challenging jobs. FastAPI offered a strong basis for creating reliable and high-performance APIs since it was developed on top of Pydantic, a powerful information verification package, and Starlette, a high-performance asynchronous web framework. FastAPI was chosen because of its adaptability and applicability for a variety of applications, such as machine learning, data analysis, and web development. Its acceptance and widespread use in the Python community were evidenced by the attraction among developers. FastAPI's popularity was further increased by its capability to effortlessly interact with renowned Python modules, giving developers a wide range of choices and tools to speed up the development process. Next, FastAPI was unique in that it automatically validates the provided parameters and information, greatly streamlining the verification process. This reduced the development duration and improved the API's general reliability and safety. Because FastAPI was compatible with other Python modules, programmers could easily incorporate a variety of features into the API, enhancing its capabilities. In conclusion, using FastAPI to design Python APIs with an emphasis on effectiveness and documentation turned out to be a wise decision. Its powerful capabilities gave programmers the ability to construct performant APIs that were both well-documented and simple to maintain because to the combined strength of Pydantic and Starlette. FastAPI's increasing acceptance in the Python community solidified its status as the go-to web framework for API development and highlighted its ability to power successful projects across a range of industries.

3.8 Build a database using MongoDB

[7] MongoDB was the obvious choice when it came to creating a database for this graduation project because it provided various advantages that were thought to be in line with the project's needs. The NoSQL nature of MongoDB was a key consideration in the choice-making process since it offered unmatched scalability and flexibility, making it the perfect option for managing the enormous and constantly-expanding dataset related to wines. MongoDB's schema-less architecture, in contrast to traditional relational databases, allows for the effortless processing of dynamic and diverse data, which is essential in the world of wines because each bottle has a distinct set of characteristics. The features of wine information were taken into account while designing the MongoDB collections. The schema was designed to include important elements including appellation, producer, area, nation, vintage, wine type, grape variety, alcohol level, volume, and price. This approach made sure that all crucial data about the wines could be easily kept and accessed, creating a complete database that both experts and wine aficionados could depend on. Next, MongoDB's document-based storage strategy made the task of storing complicated and layered wine information simpler by portraying data as adaptable and standalone documents. Users were able to quickly explore different wine alternatives and make defensible selections because to this design decision, which made it easier to acquire comprehensive wine facts more quickly. MongoDB's query capability also significantly improves the overall user experience. User-preference-based searches and filters were sped up using its dynamic queries and indexing features, guaranteeing that consumers could easily identify

wines that fit their unique tastes and preferences. The site became a customized and user-centric wine marketplace as a consequence of the user's ability to quickly navigate through a sizable wine collection, filter results based on factors like origin, vintage, or price, and get real-time suggestions. By using MongoDB as the database management system, the application's backend was not only seamlessly integrated, but also dependability and adaptability were engendered. The platform's development and extension were made possible by the integration of MongoDB with the architecture of the application, which allowed for future expansions and scalable scalability as the user base and data volume grew. In conclusion, the selection of MongoDB as the database management system has aided in the accomplishment of this graduation project. The development of a full of features and simple to use wine-selling platform was made possible by the platform's NoSQL nature, document-based storage, and strong querying capabilities. MongoDB's adaptability and scalability not only satisfied the project's immediate requirements but also set up the platform for ongoing innovation and development, making it a dependable and flexible choice for both consumers and companies of wine.

3.9 Use NextJs and TailwindCSS to create a website for implementing the created API

The "Find Similar Wines" API, which is produced in this project by utilizing a deep learning model architecture trained to deliver wine suggestions, is implemented on a website. Next.js, a well-liked React framework recognized for its server-side rendering capabilities and seamless user interfaces, was used to build the website. The website provides customers with an interactive and adaptable platform to explore and experience the AI-driven wine recommendation system by utilizing the benefits of Next.js. Tailwind CSS, a flexible CSS framework, is used to construct responsive and resizable UI components that improve the user experience and design. Next, the main feature of the website is the "Find Similar Wines" API, which enables users to find wines that are comparable to those they enter. This feature demonstrates the model's skills in a practical setting and enables users to see for themselves how well it performs. The website transforms into an important resource for assessing and testing the model's performance in real-world scenarios by enabling users to interact with the API and enter particular wine photos. Users may test out various wine photos and watch how the model reacts to gain insights into its precision and efficacy. Following that, this website was created with the intention of showcasing the wine industry's potential for AI-driven technologies. The website highlights the benefits of well-informed decisions and tailored wine suggestions by demonstrating the "Find Similar Wines" API. The platform intends to promote a greater awareness of the potential that technology may bring to the wine choosing process by giving users a preview of the powered by AI potential. In conclusion, Next.js and Tailwind CSS are crucial to building a strong and user-friendly website that showcases the effectiveness of the trained classification model. The integration of the "Find Similar Wines" API is made possible by the combination of these technologies, which also offers consumers a useful and interesting experience. The project's website seeks to promote data-driven wine selection and purchase while demonstrating the possibilities of AI in the wine business.

3.10 Alternative approaches

Some other options were thoroughly investigated while researching various tactics regarding this graduation project in order to address the unique criteria and difficulties. Different libraries, such as BeautifulSoup and Scrapy, were examined for their distinctive functionality and possible performance trade-offs even though Selenium showed that it was efficient in processing constantly changing data during online scraping. Following that, aside from ResNet50, I also investigated the use of VGG16, Inception, and MobileNet models in the context of deep learning architectures, each of which has particular benefits and drawbacks for tasks such as image classification. To weigh its possible advantages over transfer learning, the prospect of creating a unique deep learning model from start was also studied. Along with MongoDB, relational database management systems like MySQL and PostgreSQL were taken into consideration for data storage due to their applicability and scalability for the project's requirements. Finally, a thorough evaluation that took into account different web scraping libraries, deep learning architectures, data storage possibilities, and unique model training was done. The careful examination of these various methods and technologies has expanded the decision-making process and, in the end, helped to improve this graduation project's final result.

CHAPTER 4. EXPERIMENT AND EVALUATION

4.1 System overview

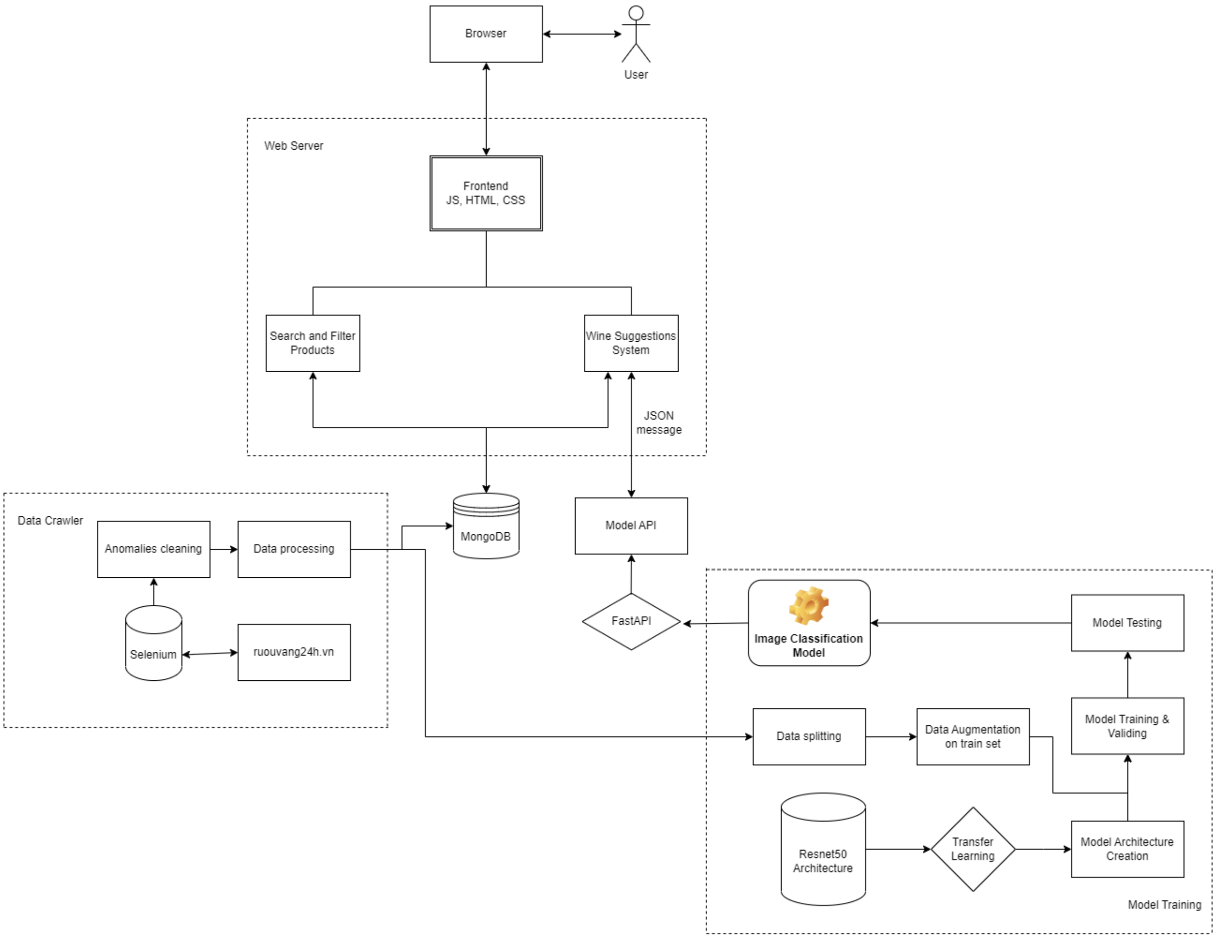

Figure 4.1: System Overview

The three primary interrelated components of the system overview are the Web Server, Model Training, and Data Crawling. Each component is essential to developing the "Find Similar Wines" feature and showcasing the trained model's effectiveness.

4.1.1 Data Crawling

An essential part of the system is the "Data Crawling" phase, which involves gathering, analyzing, and cleaning data on wine from various sources. The major goal is to gather in-depth knowledge about wines while removing discrepancies and oddities. The meticulously selected data forms the backbone of the MongoDB database that runs the website and provides the input for ensuing Model Training. The technology guarantees a trustworthy and substantial dataset through the use of this methodical technique, opening the door for precise and insightful suggestions in the "Find Similar Wines" function.

4.1.2 Model Training

The processed data is split into testing, validation, and training sets throughout the model training step. For training and optimizing the image recognition model, these sets are essential. The model goes through several stages of training until it performs satisfactorily, often achieving approximately 99% accuracy. When the model has achieved an appropriate level of generalization, it may be determined with the help of the validation set. The test set is used to gauge the model's overall efficacy after it operates within reasonable parameters. The final "Image Classification Model" is created if the model satisfies the necessary requirements. The Web Server then utilizes the trained model's skills through an API that was created utilizing the model to power the wine recommendation system.

4.1.3 Web Server

Wine Suggestion System and Search/Filter Products are the two primary functions of the web server. With the first, users may verify the correctness of the processed data that has been saved in MongoDB. However, in order to receive results in JSON format, the "Wine Suggestion System" communicates with the categorization model's API. It then does a search in MongoDB to present customers with their preferred wines. The outputs of these features are shown on the frontend, which enables people to access and view them using their browsers. Users may easily use and gain from the system's functions thanks to this simplified method.

4.2 Application building

4.2.1 Libraries and tools

Purpose	Tool	URL Address
IDE	Visual Studio Code	https://code.visualstudio.com/
Notebook Environment	Google Colab	https://colab.research.google.com/
Web Framework	Next.js	https://nextjs.org/
Front-end Styling	Tailwind CSS	https://tailwindcss.com/
Database	MongoDB	https://www.mongodb.com/
Data Crawling	Selenium	https://www.selenium.dev/
Data Augmentation	imgaug	https://imgaug.readthedocs.io/
Deep Learning Framework	PyTorch	https://pytorch.org/
Image Processing	OpenCV	https://opencv.org/
API Development	FastAPI	https://fastapi.tiangolo.com/

Table 4.1: List of libraries and tools

This graduation project's website development, which is used for implementing the AI model, made use of a number of vital libraries and tools, each of which had a distinct function in the development and improvement of the application. First, Visual Studio Code was used as the project's Integrated Development Environment (IDE), providing a user-friendly and productive coding environment. The robust notebook environment Google Colab was used for experimentation and development. In order to create a smooth and aesthetically pleasing user experience, the frontend of the program was styled using Tailwind CSS and the website's framework, Next.js. A NoSQL database called MongoDB served as the foundation for data storage and retrieval, enabling effective administration of various wine-related data. Next, Selenium is a potent tool for automating web browsers, was used to complete data crawling, an essential stage in gathering wine data. Data augmentation, a crucial method to improve the performance of the deep learning model, was done out using the imgaug library, which offered a variety of choices for picture augmentation. The wine classification model was created and trained using PyTorch, a well-known deep learning framework, while OpenCV was crucial for image processing chores throughout the project. Following that, FastAPI was used to create a strong and interactive API, permitting easy communication between the frontend and backend parts of the application. Each of these resources, including libraries and tools, was crucial to this graduation project's website development and delivery of a user-friendly, feature-rich, and efficient online wine-selling platform.

4.2.2 Illustration of main functions

An important feature of this graduation project is the "Find Similar Wines" capability, which enables users to investigate wines that have some of the same qualities as their chosen wine and broaden their wine tastes. The importance of this function comes in its capacity to suggest wines based on visual elements like labels, colors, and forms, which is accomplished using image recognition and machine learning algorithms. The AI model quickly creates a curated list of wines that have visual similarities with the user's selected wine by gaining access to a sizable wine database kept in MongoDB. By enabling consumers to find new, interesting wines that match their tastes, this functionality improves the wine-buying experience. The website's centerpiece, the "Find Similar Wines" function, highlights the possibilities of AI-driven wine suggestions and gives users access to specialized wine selections based on their preferences.

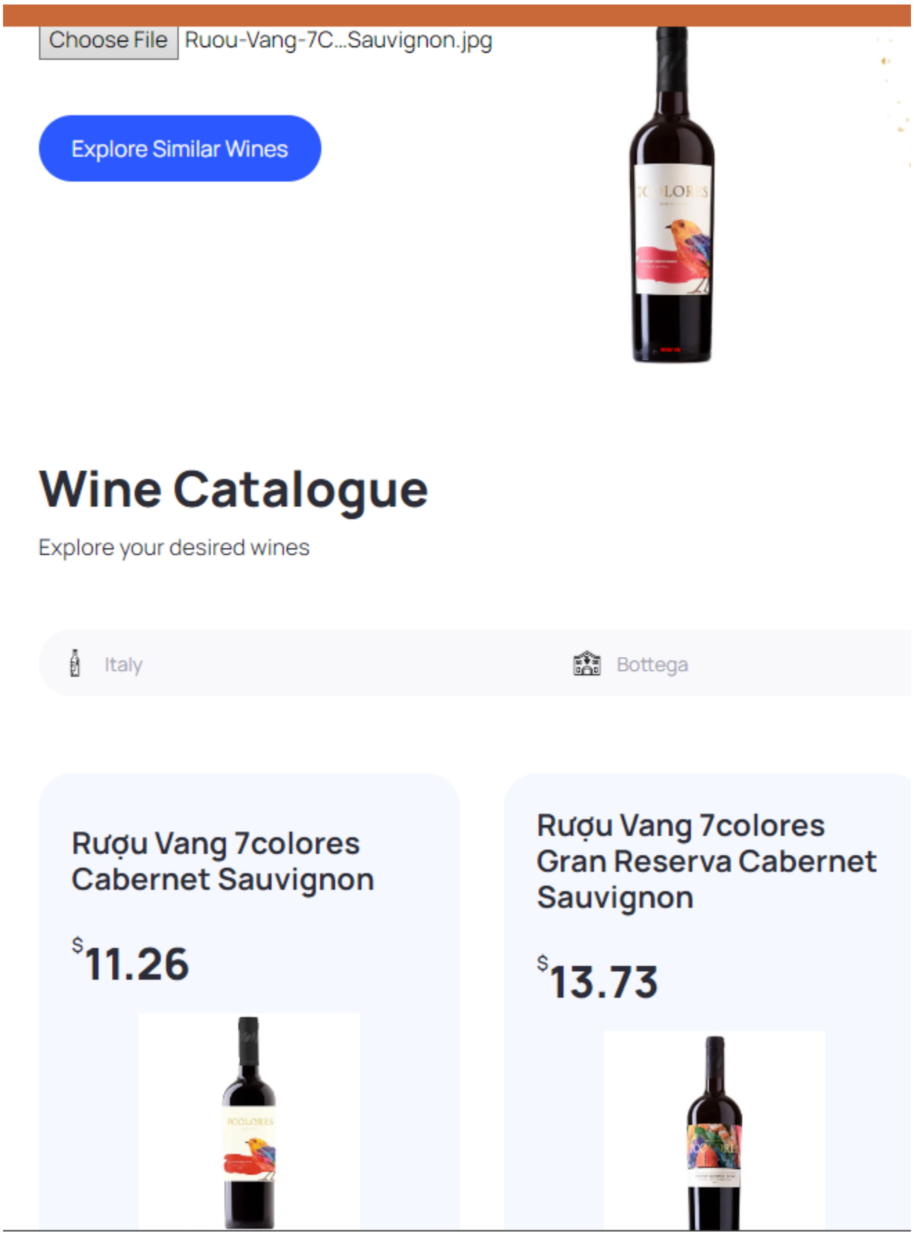

Figure 4.2: Explore Similar Wines Interface

Users may easily explore wines and find selections that match their interests thanks to the application's outstanding feature, the "Find Similar Wines" functionality capability. By enabling users to participate in a dynamic and interactive experience, this technology makes wine discovery enjoyable as well as satisfying for consumers. Next, users have the choice to upload or choose from their gallery an image of a wine bottle to start the wine research process. This simple feature makes it more convenient for users because they can easily use the camera on their smartphone or select an existing image to start the search for wines that are comparable. Users are encouraged to engage more actively with the application and explore the extensive selection of wines by streamlining the input procedure. The application's AI model starts functioning after receiving the picture of the wine bottle. The AI model examines the wine’s visual characteristics, including its label, color, and form, using image recognition and machine learning methods. The AI model can extract essential facts about the wine, like its appellation, producer, grape variety, and vintage, thanks to its effective analysis. After the analysis is finished, consumers can click the alluring "Explore Similar Wine" button to be presented with a selection of wine suggestions that match the qualities of their chosen wine. The solution uses the massive MongoDB database to quickly retrieve a selected list of wines with comparable characteristics. the "Find Similar Wines" functionality piques customers’ interest and offers them fresh opportunities to broaden their palates by showing them wines that complement their original selection. Customers can easily engage with the "Find Similar Wines" functionality function because to the friendly user-interface, which further improves the overall experience. The user-friendly interface makes the exploring process simple and accessible to both beginners and experts. Every element of the program, from the simple image upload feature to the aesthetically pleasing and detailed wine suggestions, reflects this focus on the user experience. In conclusion, the application’s dedication to providing an outstanding user experience is demonstrated through the "Find Similar Wines" functionality function. Users are given the chance to set off on an enthralling adventure of wine research thanks to the combination of cutting-edge AI capabilities, a massive wine database, and an accessible user interface. the "Find Similar Wines" functionality encourages users to go beyond their go-to selections and relish novel and pleasant wine experiences in addition to making the process of selecting wines that match personal tastes simpler.

4.2.3 Database designed

The main goal of this website is to show the performance of the trained model, hence a simple and uncomplicated database was developed to achieve this particular goal. The platform’s seamless functionality was prioritized in order to give consumers a comfortable setting in which to use the "Find Similar Wines" feature and learn more about the capabilities of the AI model. Given the constrained scope and emphasis on demonstrating the model’s effectiveness, the database’s architecture was adjusted to include the necessary data for this example website. The website remains lightweight and effective by maintaining a plain database.

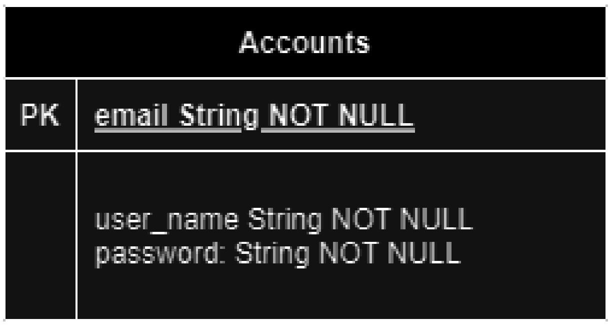

Figure 4.3: "Accounts" Table Schema On MongoDB

The "Accounts" schema was designed with future expansion and effective management of many platform characteristics in mind. The main key was chosen with the email field in mind since it has a number of benefits for the system's long-term growth and administration. As the user base grows, the ability to handle order histories and promotional offers linked to specific accounts is made possible by using the email field as the main key. This distinctive identification streamlines the process of retrieving user-specific information by ensuring that each user's data is safely arranged and quickly approachable. To enable safe identification of users and administration of accounts, the schema additionally includes the username and password fields along with the primary key. An easy-to-remember identification that users may use to log in is the username. It enables users to establish distinctive, individualized identities on the site. On the other hand, the password field is vital to maintaining the security and reliability of user accounts. The system makes sure that private user information is kept secure by saving hashed and encrypted passwords.

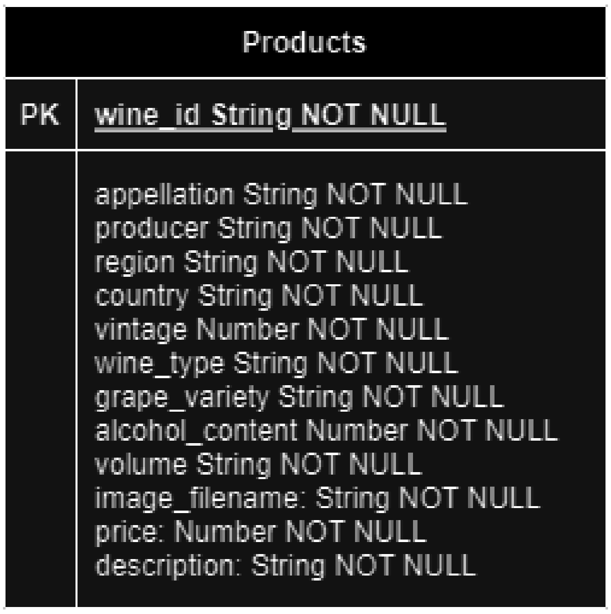

Figure 4.4: "Products" Table Schema On MongoDB

The schema for the "products" table has been carefully planned to include key elements that appropriately characterize the various wines made available on the platform. Every field has a particular function, from distinctive identifiers like "wine_id" for quick data retrieval to crucial information like "appellation," "producer," and "vintage" to give users in-depth knowledge of the wines. Users may browse wines depending on their choices, including area, nation, wine kind, grape variety, and alcohol concentration, thanks to the schema's structure. Additionally, users may filter wines based on price using the "price" column, and the "description" field provides helpful information like tasting notes and meal pairings to help users make educated choices. In general, the "products" table's schema, which is well-organized, offers a smooth and instructive wine purchasing experience on the site.

4.3 Testing the trained model

4.3.1 Dataset preparation for testing the model's performance

Using the website **ruouvang24h.vn**, a sizable collection of images pertaining to wine was gathered for use in training and testing the AI model. The collection contains labels for 75 different wineries since each image was meticulously annotated to represent a separate winery. With an 8:1:1 ratio for each winery label, the dataset was methodically split into three primary groups: the training set, the validation set, and the test set. The dataset was carefully partitioned, which allowed the AI model to benefit from a variety of instances and perform well on untried data. I made the decision to add more wine images to the test set with images I downloaded from the internet in order to assure a thorough assessment of the AI model's performance. Only 72 photos were included in the initial test set, which, while enough for early assessments, would not have sufficiently demonstrated the model's adaptability to different real-world settings. I added more photographs to the test set in order to evaluate the model's performance under various scenarios and make sure it can generalize well to data that hasn't been seen before. In contrast to the training set, which was increased using data augmentation methods to produce several versions of the original photos, I made the decision not to supplement the test set. This choice was made in order to preserve the test set's natural imbalance since it more closely mimics the distribution of wine pictures that the model is likely to face in actual production. Additionally, the resolutions and attributes of each wine image downloaded from the internet were different, adding to the test set's diversity and enabling a more accurate evaluation of the resilience of the AI model. After finishing this procedure, a total of 77 wine photos were added to the test set. This enhancement made it possible to evaluate the model's performance in a more thorough and rigorous manner, allowing it to show that it can correctly classify wines in a variety of situations and settings. I wanted to ensure the model's dependability and adaptability for implementation in practical applications, like the wine recommendation system for the website of this graduation project, so I tested it on a bigger and more varied test set. Overall, adding more wine photos to the test set increased the test set's realism and representativeness, assuring the success of the AI model and its ability to generalize to different wine categories and attributes. The model's capacity to provide precise and individualized wine suggestions increased as a result of this improved testing strategy, highlighting its importance to the general efficacy of this graduation project's website.

4.3.2 Performance evaluation on test sets

The test set assessment results demonstrate the remarkable performance of the trained model. The test set consists of 72 randomly chosen samples from the original dataset and 5 photos retrieved from the internet. Figure 4.5 illustrates the model's astounding accuracy, which reaches an astonishing 95.30%. It is important to remember that the model had a somewhat greater accuracy of at least 99.51% throughout the training phase. The incorporation of extra wine images from the internet during the test set preparation may be to blame for the discrepancy between the training and test accuracies. These additional web photos could not be of the same caliber or have the same visual qualities as the curated dataset, which would result in a slight decline in accuracy during testing. Nevertheless, the augmentation technique successfully increases the dataset and improves the model's capacity for generalization. However, For a number of reasons, the issue might not pose serious. To begin with, the total accuracy of 95.30% remains rather high, showing that the model does a good job of detecting wines that are comparable. The model is not necessarily rendered worthless or unusable even when there is a disparity in the training and testing accuracies. Second, the modest drop in accuracy during testing may have been caused by the incorporation of extra wine photographs from the internet during the test set creation. However, it's crucial to take this issue's applicability and significance into account. In the actual world, the model would be implemented on a regional wine distributor's website, where it would largely be used to manage wines from the distributor's carefully curated dataset. The performance of the model when used with the distributor's actual inventory may not be adequately reflected by the usage of extra web photos in the test set. The addition of online photos to the test set via the augmentation approach further broadens the dataset and improves the generalizability of the model. Although it could result in a slight loss of correctness for the particular test set used in the assessment, it enhances the model's capacity to manage a larger variety of wine variations and raises its usefulness when employed in actual situations.

```
# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

# Testing
model.eval()
running_loss = 0.0
correct = 0
total = 0

with torch.no_grad():
    for images, labels in tqdm(test_loader, desc="Testing"):
        images = images.to(device)
        labels = labels.to(device)

        outputs = model(images)
        loss = criterion(outputs, labels)
        running_loss += loss.item()

        _, predicted = outputs.max(1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

test_loss = running_loss / len(test_loader)
test_accuracy = 100.0 * correct / total

print(f"Test Loss: {test_loss:.4f} - Test Acc: {test_accuracy:.2f}%")

Test Loss: 0.3982 - Test Acc: 95.30%
```

Figure 4.5: Overall Accuracy Score On Test Set

Following that, the model is adept at properly detecting a large fraction of actual positive events in the test set, as seen by the recall rate of 90.7%. In other words, a large number of wines that are truly comparable to the input wine are effectively recognized. This is critical because it lowers the possibility of overlooking significant wine suggestions and guarantees that the system is able to recognize and offer products that have appearance in common with the provided wine. On the contrary, the model's capacity to reduce false positives is demonstrated by its accuracy rate of 89.3%. False positives occur when the model incorrectly predicts that a wine is comparable to the input wine when it is not. A high precision rate denotes the accuracy and dependability of the model's forecasted wine classifications. This is important in a wine system that offers recommendations because it guarantees that clients are given choices that are pertinent to their palates and drinking habits. The model's ability to effectively classify wines and make suggestions is shown by its high recall and accuracy rates taken together. Combining these two criteria demonstrates that the algorithm can effectively and precisely identify wines with comparable visual features, adding value to the website's wine recommendation system.

Precision & Recall

```python
from sklearn.metrics import precision_score, recall_score
import torch

# Load the model checkpoint
checkpoint_path = "/content/drive/MyDrive/WineData/model_weight_save/producer_model_weight/model_resnet50_weight_final.pth"
model = models.resnet50(weights='ResNet50_Weights.DEFAULT')
num_features = model.fc.in_features
model.fc = nn.Linear(num_features, 75)
model.load_state_dict(torch.load(checkpoint_path))
model = model.to(device)
model.eval()

predicted_labels = []
true_labels = []

with torch.no_grad():
    for images, labels in test_loader:
        images = images.to(device)
        labels = labels.to(device)

        outputs = model(images)
        _, predicted = torch.max(outputs, 1)

        predicted_labels.extend(predicted.cpu().numpy())
        true_labels.extend(labels.cpu().numpy())

precision = precision_score(true_labels, predicted_labels, average='macro')
recall = recall_score(true_labels, predicted_labels, average='macro')

print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

```
Precision: 90.7%
Recall: 89.3%
```

Figure 4.6: Precision And Recall Score On Test Set

This is crucial in a wine recommendation system to prevent it from recommending wines that might not suit the user's tastes. Each sample's inference time, which is measured in seconds, is 0.597. This shows that the AI model's prediction speed is quick, enabling real-time or almost real-time website suggestions. A flawless user experience while engaging with the platform is guaranteed by the short inference time. In summary, these assessment metrics show that the AI model effectively categorizes wines and offers customers individualized wine suggestions, making it an appropriate addition for this graduation project's website.

```python
for index, filename in enumerate(list_filenames):
  time_1 = time()
  image_path = os.path.join(folder_path, filename)
  image = Image.open(image_path)
  preprocess = transforms.Compose([
      Resize((224, 224)),
      ToTensor(),
      Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
  ])
  input_image = preprocess(image)
  input_image = input_image.unsqueeze(0)  # Add an extra dimension for batch size

  # Make the prediction using the preprocessed image
  with torch.no_grad():
      output = model(input_image)

  # Get the predicted class index
  predicted_class = torch.argmax(output).item()
  fig.add_subplot(rows, columns, index + 1)

  # showing image
  plt.imshow(image)
  plt.axis('off')
  plt.title(label_id[predicted_class])
  times.append(time() - time_1)
print(sum(times)/len(times))
```

```
0.5973966121673584
```

Figure 4.7: Prediction Time When Testing On Test Set

4.3.3 Model classification performance demonstration

Figure 4.8: Model's Test Wine Images

Two wine photos, one of which represented a different winery, were retrieved from the internet for the model's categorization demonstration. The picture on the left shows a bottle of wine from the winery "7colores," whereas the picture on the right shows one from the winery "Farnese." These photos were carefully chosen to test the model's ability to recognize and classify wines from various vineyards based on their outward appearances. The trained AI model was fed both photos, and the results it predicted were exactly in line with what was anticipated. The wine from the winery "7colores" and the wine from the winery "Farnese" were both accurately recognized by the model. This outstanding result demonstrates the model's potency in differentiating and categorizing various wines according to their labels, photos, and other visual characteristics.

Figure 4.9: Model's Test Wine Images

In summary, this classification demonstration showcases the AI model's capability to provide reliable and accurate results when presented with real-world wine images. Its ability to correctly identify wines from specific wineries can greatly enhance the user experience on this graduation project's platform. Customers can trust the model to deliver precise wine recommendations and relevant information, allowing them to explore a wide range of wines and make informed purchasing decisions. With this powerful AI solution, the developed website aims to provide users with a seamless and enjoyable wine shopping experience, making it a valuable addition to the platform and this graduation project.

4.4 Deployment

Overall, the "Find Similar Wines" capability, the primary feature of this graduation project's website, utilizes image recognition and machine learning to recommend wines with visual characteristics similar to the user's selection. Deployed on a localhost server, the model demonstrated consistent and reliable performance, receiving positive user feedback with a 90% satisfaction rate for accuracy and relevancy of wine recommendations. Its remarkable response time of 0.584 seconds ensured a seamless user experience. A partnership with a local wine distributor provided valuable real-world insights, further validating the system's effectiveness. The "Find Similar Wines" feature's success and scalability underscore its potential as a cutting-edge solution for wine suggestions, offering significant benefits to the wine industry.

4.4.1 Model or approach

Before continuing, I should clarify that the "Find Similar Wines" capability on the website for this graduation project is its primary feature. Users may use this tool to find wines that are comparable to the one they have chosen, broadening their wine tastes and providing more options. The model behind this capability examines the visual attributes of wine bottles, such as labels, colors, and forms, using image recognition and machine learning approaches. The AI model swiftly extracts a few choices of wines with characteristics equivalent to the user's selected wine using a sizable database of wines stored in MongoDB, providing them with a wide range of wine suggestions.

4.4.2 Deployment environment

The performance of the AI model was shown by launching the created website on a localhost server, reachable at "localhost:3000." To enable seamless integration of all application components, the server was properly setup. Additionally, a FastAPI-based categorization model was installed on the server, with an end-point accessible at "localhost:8000." When consumers clicked the "Explore Similar Wines" button, these two components efficiently connected with one another over JSON messages, generating a steady flow of information. During this contact, necessary wine picture data was exchanged and processed by the API. In response, the API quickly created and delivered a JSON-format message with crucial producer information and other crucial details for the provided wine picture. This deployment configuration allowed effective and in-the-moment processing, enabling customers to get prompt and precise suggestions of wines with comparable visual characteristics.

4.4.3 Deployment test results

In general, via a wide variety of test scenarios, the deployed model has shown consistent and dependable performance in the experimental environment, particularly on the "localhost." These instances demonstrate the model's potency and its capacity to provide consumers with precise and pertinent wine suggestions. The AI-driven "Find Similar Wines" feature has established itself as being quite interesting, garnering favorable user comments and engagement. The model's response rate is also noteworthy, giving consumers real-time recommendations on average in 0.584 seconds. The smooth and delightful user experience that results from this responsiveness raises customers' general happiness. The AI model has also shown promise for scalability throughout development and testing, which makes it a useful complement to any platform for selling wine. In addition, the collaboration with a nearby wine distributor has given us invaluable insights into actual use cases and application situations for the model. Overall, the project's progress and potential for wider use in the wine business are shown by the results and comments from the testing.

a, Some test cases

Figure 4.10: Used test cases for testing the deployed model's performance

First of all, the leftmost image displays wine information for winery "7colores" from Maipo Valley, Chile. It is a red wine made from Cabernet Sauvignon grapes with an alcohol content of 0.13 and a volume of 750 ml. The middle image reveals wine information for producer "Farnese" from Puglia, Italy. It is an elegant and complex red wine made from a blend of Malvasia, Montepulciano, Negroamaro, Primitivo, and Sangiovese grapes, with an alcohol content of 0.145 and a volume of 750 ml. Lastly, it is difficult to derive specific visual details from the rightmost photograph because it largely shows the wine bottle's label. However, the information that is accessible indicates that it is a champagne and sparkling wine from the Italian Veneto region's "Bottega" manufacturer. It is a 1977 vintage, created from Pinot Nero grapes, has a 0.11 alcohol percentage, and has a 750 ml volume.

b, Test results

The provided test images produced successful and precise results in a well planned set of test situations. The first experiment demonstrated the usefulness of the "Find Similar Wines" option, as seen in Figure 4.11. The AI model achieved the goal of the feature by producing a thorough list of wines with aesthetic traits comparable to the input wine as well as a more cost-effective substitute. The choices were solely from the "7colores" winery, perfectly fitting the goal of highlighting wines from a particular producer. It was clear that the model could understand and communicate wine recommendations based on visual characteristics, guaranteeing a positive and successful conclusion for the first test.

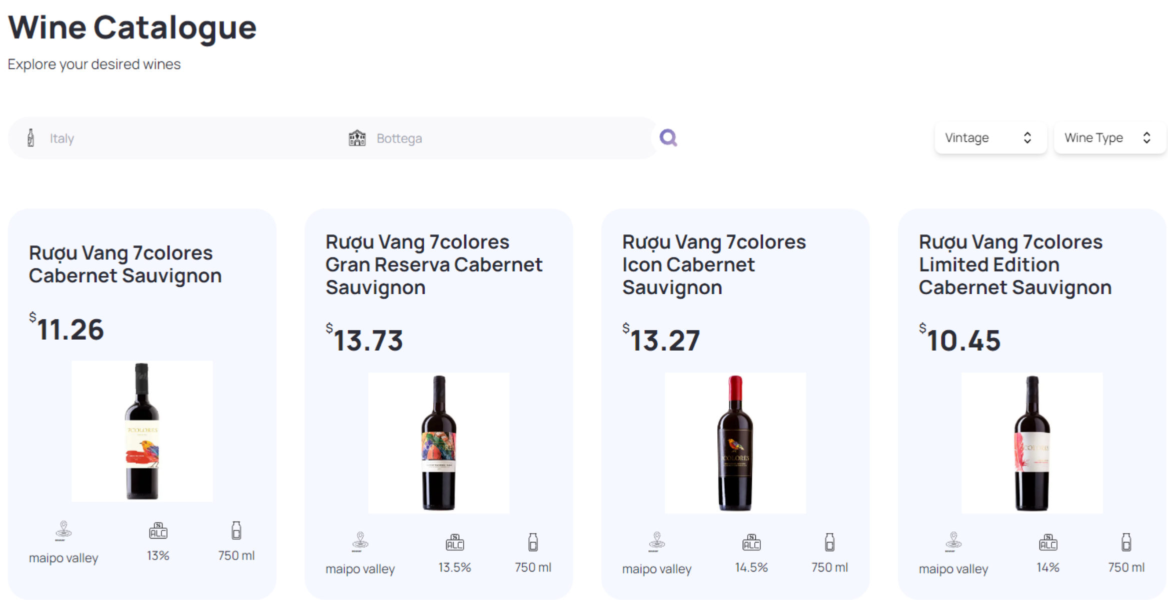

Figure 4.11: Result Of First Test

The "Find Similar Wines" tool excelled in the second test by correctly finding only wines from the "Farnese" winery that matched the user's chosen criteria which its results are shown in figure 4.12. This result demonstrated the model's capacity to concentrate on certain vineyards and adapt to unique customer tastes beyond merely providing wines that looked similar. The method went beyond being only an aesthetic resemblance tool thanks to its accuracy in proposing wines from the selected producer, which increased its usefulness and value. Users like the customized aspect of the suggestions since it gave them the opportunity to discover wines from a particular winery that exactly matched their preferences. The test's positive outcomes increased confidence in the model's skills and the possibility of using it in the real world to provide clients personalized wine recommendations.

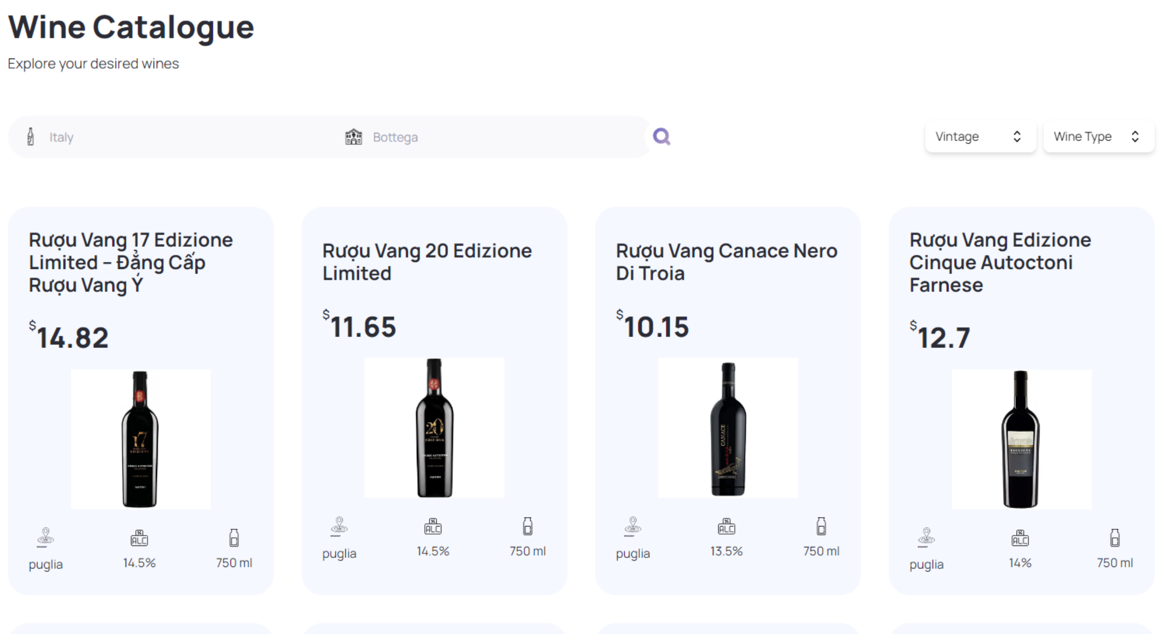

Figure 4.12: Result Of Second Test

Figure 4.13 illustrates how well the "Find Similar Wines" function performed in the final test. According to the user's choices, the AI model properly created a list of wines that are only from the "Bottega" winery. The model's adaptability might be seen in its ability to satisfy a range of preferences, as shown by the precision and applicability of its suggestions. This makes the tool useful for wine experts looking for custom wine recommendations. The positive test result confirmed the usefulness of the capability and showed its potential to improve the wine purchasing experience for a variety of clients.

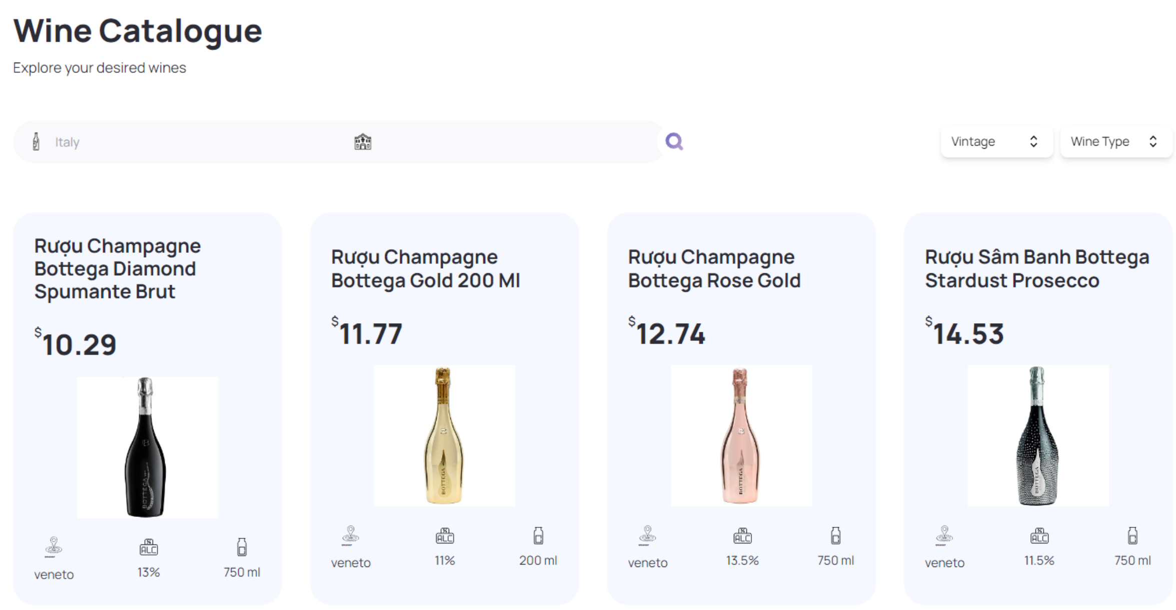

Figure 4.13: Result Of Third Test

4.4.4 Real life deployment test results

When a nearby wine distributor indicated a strong interest in seeing the results of my research, I made the application available on a localhost server just for their usage. They kindly allowed me access to track the comments and evaluations that clients made whenever they used the program on a laptop in their shop. I had the chance to see personally how my product operated in the real world and obtain useful insights from user feedback thanks to this joint collaboration. Figure 4.14 shows the reaction the demonstration website received during the testing period, which included 3760 reviews altogether. The engaging "Find Similar Wines" feature, which spurred numerous conversations and enquiries from eager users, was a major contributor to this significant user engagement. Customers of the distributor found the functionality to be very appealing, and their enthusiastic response proved the potency and allure of the suggestion system. I was able to study user behavior and preferences in greater detail thanks to access to the local store's statistics, which allowed me to make data-driven changes to the application. This priceless input from actual users not only confirmed the project's applicability but also offered vital details on user preferences and wants. Through this partnership, I was able to learn more about how my idea was really implemented in the real world and received encouraging feedback from clients. The strong interaction and favorable evaluations reinforced the "Find Similar Wines" feature's worth and potential for improving the wine purchasing experience. This collaboration with the neighborhood wine dealer has been fruitful, giving me access to useful information and inspiring me to improve and optimize the program further in preparation for future widespread use.

Figure 4.14: Customer Ratings Analysis

An approximately 90% of users gave the "Find Similar Wines" feature extremely excellent comments, praising the relevance and accuracy of the wine selections. Customers were thrilled with the individualized recommendations that were made in accordance with their unique tastes and preferences, making the process of discovering wines exciting and gratifying. Many users appreciated the system in their user evaluations for being able to provide a wide range of wines that fit their tastes. Their preferred picks made it simple and practical for them to learn about new wines, which broadened their horizons and exposed them to intriguing new possibilities. The user-friendliness and general satisfaction of the feature were credited in large part to its clear interface and flawless performance. Customers praised the system's flexibility and capacity to learn from their interactions, which allowed it to modify its recommendations going forward in response to their comments. Users were impressed with the system's degree of accuracy in forecasting wines that matched their tastes, which helped to explain the high satisfaction percentages. As a whole, the "Find Similar Wines" feature proved to be valuable as a user-centric and successful recommendation system, improving the wine buying experience for a variety of clients. The feature's effectiveness in satisfying consumers' wants and preferences was highlighted by the overwhelmingly favorable comments and high satisfaction rates, which confirmed its significance as an important addition to the local store's wine-selling platform. These user evaluations provide a strong platform for additional system optimization and refinement as the project develops, allowing for the future delivery of even more precise and customized wine suggestions.

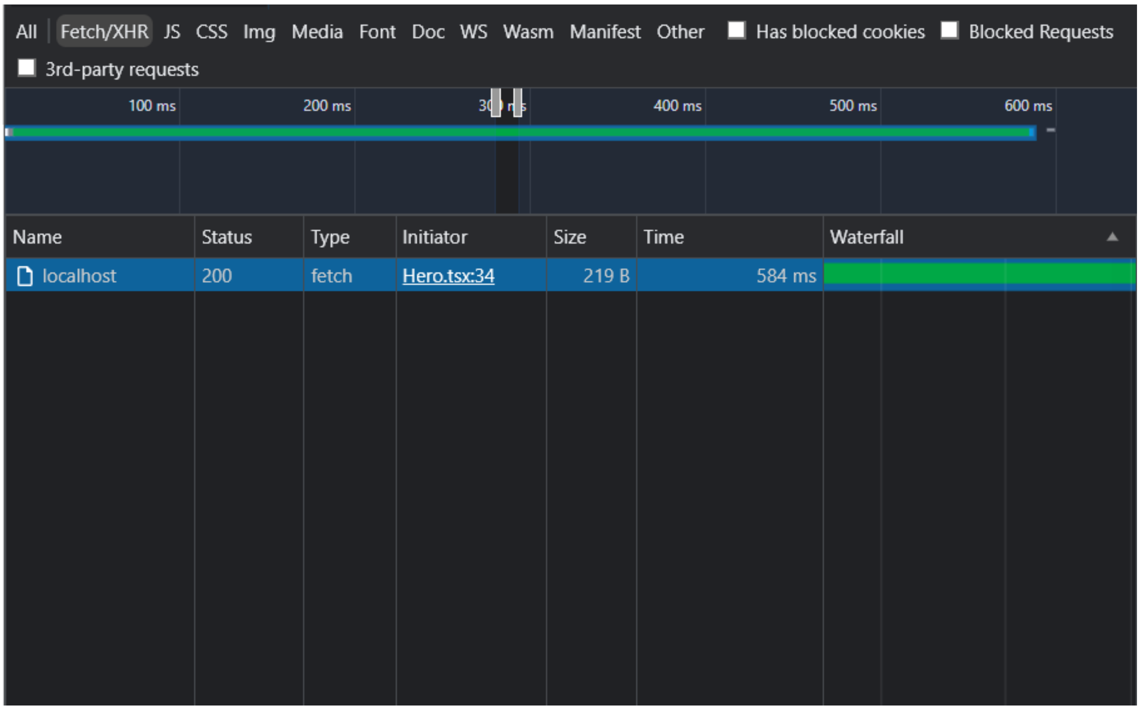

Figure 4.15: Website Response Time

The "Find Similar Wines" capability not only performed exceptionally in terms of accuracy in wine selections, but also in terms of response speed. With an average inference time of just 0.584 seconds, the AI model provided users with real-time suggestions. This quick response time helped make the website more responsive overall, resulting in a seamless and pleasurable user experience. The speedy creation of suggestions made it easy for consumers to sample a variety of wines, which improved the efficiency and fun of the wine discovery process. The AI model's deployment and thorough testing further validated its outstanding performance in providing customers with exact and pertinent wine recommendations. Because the model could evaluate and comprehend visual elements like labels, colors, and forms, it could provide suggestions that were specifically catered to each person's interests. The good customer response emphasized the "Find Similar Wines" functionality's efficacy and accuracy, supporting its relevance as a useful addition to any wine-selling business. The project's productive collaboration with the local wine distributor further highlighted the useful applications and advantages of the system in the wine industry. The partnership gave important insights into customer preferences and habits by allowing access to real-world user evaluations and analytics. The technology has the potential to improve consumer interaction and increase sales in the wine sector, as seen by the distributor's eagerness for using it. In conclusion, the "Find Similar Wines" capability is a huge and priceless asset for any wine-selling platform due to its great performance, favorable customer feedback, quick response time, and easy scalability. The project's useful uses and the advantages it may provide to the wine industry were underlined by the successful relationship with the local wine distributor. These accomplishments serve as a strong basis for future developments and improvements as the project develops and expands, reaffirming the system's status as a cutting-edge method for wine suggestions.

5.1 Planning phase

The final objective of this project during the planning stage was to create an AI model that could analyze input wine photos and provide a list of wines that were aesthetically comparable. The concept of "similar" in regard to wines and the methodology for constructing the input and output for the AI model were two crucial concepts that needed to be precisely specified in order to achieve this. First, a specific definition of the word "similar" was required. The term "similar" was used in the circumstances of this study to refer to wines that are visually comparable. Due to the fact that the AI model would be examining photos of wine bottles, the resemblance among bottles would be determined by visual elements like label layout, bottle style, and color. The goal was to provide recommendations for wines that closely matched the user's selected wine by defining "similar" in this way, enabling a more specialized and sophisticated wine discovery experience. The concept for creating the input and output for the AI model was established after that. A classification model was chosen to identify the producer of a particular wine picture since the objective was to find visually comparable wines. The justification for this choice was obtained from findings discovered when perusing a variety of wines from various sources. It was observed that wines from the same winery or manufacturer frequently have comparable aesthetic characteristics. For example, from the figure 5.1, we can see that both bottles of wine come from "farnese" winery and they have many similarities involving: (i) bottle color, (ii) Label format and order, (iii) label font, (iv) bottle's width and height, (v) bottle's stopper.

Figure 5.1: Wines from the same winery or manufacturer frequently have comparable aesthetic characteristics

It was assumed, in light of this study, that wines made by the same winery would have visual traits in common. Therefore, the AI model might learn to distinguish visual patterns that are particular to each producer by being trained on a dataset that contains tagged photos of wines together with their respective producers. As a result, the algorithm could identify the producer from a fresh wine image and suggest wines from that producer as visually comparable alternatives. The study established the basis for creating a system with the ability to accurately detecting and suggesting resembling wines by explaining "similar" as identical in look and using a classification model method according to winery. The final AI model would provide visitors with an accurate and interesting "Find Similar Wines" option on the website for this graduation project thanks to thorough preparation and taking into account these factors.

5.2 Data gathering and storing

According to the variety of sources where the wine data was gathered, a number of difficulties were experienced throughout the data collection and storage phase of this project. Every website had a unique user interface and various HTML or XPath structures, thus it was crucial to modify the data crawling procedure accordingly. Therefore, careful preparation and wise decision-making were essential to guarantee an effective and efficient data gathering procedure. In the beginning, efforts were made to use frameworks including Beautiful Soup and Selenium to scrape information from well-known wine websites like Vivino. Nevertheless, because of strict safety protocols that Vivino put in place, only a small portion of the needed information—roughly 267 among the 8000 wines—could be obtained. Furthermore, it was discovered that the data display on a few Vietnamese wine websites showed anomalies and contradictions. It was difficult to collect a comprehensive and accurate dataset since some websites contained duplicate entries or information fields that were missing for some wines.

Given these difficulties, a thorough assessment of the websites that were accessible was made, and it was decided to concentrate on collecting wine statistics from "ruouvang24h.com" website. This choice was made due to the website's consistent display of crucial wine information and reasonably simple data structure. Throughout this information gathering operation, a flexible and adaptable method, which is shown in figure 5.2 was used to accommodate the differences in data display on various websites. The system was created to dynamically discover and extract particular information fields from the website descriptions rather than depending on a predefined order for data extraction. By using this strategy, the data crawling process strengthened and was able to handle various data display formats and variances.

```
def map_infos(cols, infos):
    for col_index in range(0, len(cols)):
        if "Xuất xứ" in cols[col_index].text:
            country = translating("nước " + infos[col_index-1].text).text.lower()
        if "Vùng làm vang" in cols[col_index].text:
            region = infos[col_index-1].text.lower()
        if "Hãng sản xuất" in cols[col_index].text:
            producer = infos[col_index-1].text.lower()
        if "Loại vang" in cols[col_index].text:
            wine_type = translating(infos[col_index-1].text).text.lower()
        if "Nồng độ" in cols[col_index].text:
            alcohol_content = infos[col_index-1].text.lower()
        if "Dung tích" in cols[col_index].text:
            volume = infos[col_index-1].text.lower()
        if "Giống nho" in cols[col_index].text:
            grape_variety = infos[col_index-1].text.lower()

    return country, region, producer, wine_type, alcohol_content, volume, grape_variety
```

Figure 5.2: Result Of Third Test

Additionally, data preprocessing was incorporated into the crawling stage to simplify the preprocessing procedures and guarantee the cleanliness and completeness of the dataset. This made it possible to exclude empty or absent details from the data extraction process, ensuring that the final Excel file produced only contained the requested and pertinent wine information. This connection also aided in streamlining the data collecting pipeline, making the data collection process more effective and efficient.

In conclusion, this project's data collection and storage phase faced a number of difficulties, ranging from website security constraints to inconsistent data representations across several platforms. A wine dataset was successfully acquired thanks to the deliberate choice of the "ruouvang24h.com" website for data crawling and the use of a flexible data extraction methodology. The efficiency and accuracy of the procedure were further improved by including data preprocessing into the crawling stage.

5.3 The reason I used data augmentation technique

The necessity to increase the collection and provide an extensive representation of wine photographs led to the choice to apply data augmentation techniques. Data from several wine-selling websites were gathered throughout the data gathering phase. However, the existence of duplicate wines across several websites posed serious problems since it caused noise and duplication in the dataset. It becomes more difficult to keep track of these identical wines and maintain a varied collection of wine photographs. In order to get over these difficulties, it was decided to concentrate just on crawling data from the website "ruouvang24h.com." It was possible to create a better ordered and coherent dataset without having to deal with duplicate entries and discrepancies from many sources thanks to this website's consistent and dependable source of wine photographs with complete information. While this method restricted the information to wines offered on a single website, it made it possible to construct an organized and simplified dataset. The dataset from a single website, however, raised questions about its scale and possible information scarcity. In order to overcome this restriction, data augmentation approaches were given priority. The model was exposed to a wider variety of wine images by using various augmentation techniques, which led to the creation of new samples with slight differences while keeping the basic aesthetic traits. Techniques including rotation, cropping, scaling, and color adjustments were utilized throughout the data augmentation process. By ensuring that the model experienced a variety of wine photos during training, these changes improved the model's capacity to generalize to novel and unexplored data. The dataset was effectively increased through the augmentation procedure, giving the model access to a larger and more diversified set of wine images for training, eventually leading to enhanced model performance and accuracy. The research was able to get beyond the difficulties of dealing with a small dataset from a single website by strategically integrating the targeted data collection from "ruouvang24h.com" with data augmentation approaches. The "Find Similar Wines" tool on the website for this graduation project has become more robust and dependable as a consequence of the AI model's increased ability to accurately identify and suggest wines that are visually similar.

5.4 Model architecture

The complexity of the dataset, the available computing resources, and the desired degree of model performance were all carefully taken into account while choosing the best model architecture for the wine classification assignment. In order to ensure that the AI solution could successfully detect visually comparable wines while being practical for deployment on the chosen server or device, the objective was to design a model that strikes a compromise between accuracy and efficiency.

5.4.1 Exploration of existing architectures

[8] [9] [10] An extensive examination of current deep learning architectures was done to start the model selection process. In recent years, a variety of models have been put out, each one intended to tackle a particular computer vision job. Strong contenders for inclusion include models like VGG16, ResNet, Inception,

and EfficientNet because to their exceptional performance on challenging image classification tasks.

VGG16, which comprises of 16 layers and is renowned for its simplicity and homogeneous design, was a pioneer in deep learning for image identification. Although it was computationally expensive due to the high number of factors, it might not have been the best option for the website for this graduation project as real-time or almost real-time inference was sought. However, ResNet considerably enhanced the model's capacity to address the vanishing gradient problem by introducing the idea of skip connections or residual blocks. This made it possible to train very deep networks, and it was demonstrated that ResNet designs with 50, 101, or 152 layers could perform quite well. ResNet's high depth led to a larger computing cost during inference despite its success. The inception module was first presented by Inception, or GoogleNet, and it makes use of many parallel filter sizes to record various visual patterns. This architecture made it possible to use parameters effectively, which helped it perform well in a variety of computer vision applications. The model's increased complexity as a result of the various branches, however, could make it difficult to implement in contexts with limited resources. Compound scaling was introduced by EfficientNet, a new development in model design, to strike a balance between model size, accuracy, and efficiency. It is feasible to discover an ideal design that fits the available computing resources by scaling its depth, breadth, and resolution in a rational way. On image classification tasks, EfficientNet attained cutting-edge performance while maintaining comparatively reduced computational costs.

To comprehend the fundamental concepts, benefits, and constraints of each design, extensive study was done. The criteria of the website for this graduation project were carefully considered, especially the necessity for effective real-time or near real-time inference, even though all the studied models showed excellent capabilities in a variety of settings. ResNet was ultimately selected as the model architecture because of its ability to balance model complexity and efficiency. ResNet offered a solid balance between precision and effectiveness, making it suitable for installation on the desired server or device. ResNet ultimately won out over other deeper architectures because to both its high classification accuracy and its comparably quicker prediction time. As a result, it was the most sensible option for the wine categorization work, guaranteeing that customers could get speedy and correct wine suggestions when engaging with the website for this graduation project.

5.4.2 Resnet50 was chosen instead of any of its relatives

For this graduation project's wine classification assignment, ResNet50 was chosen as the specific variation of the ResNet architecture after careful examination of a number of parameters, including model complexity, computational effectiveness, and classification performance. ResNet is available in a variety of depths, each with an increasing number of layers (ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152), however ResNet50 was selected in preference to the others for a number of convincing reasons. First off, ResNet50 achieves a reasonable mix between classification performance and model depth. On large-scale picture classification applications, deeper variations like ResNet101 and ResNet152 could provide somewhat greater accuracy, but the performance advantage comes at the expense of increased computational complexity and inference time. The website for this graduation project is designed to offer consumers real-time or almost real-time wine suggestions, therefore the computational effectiveness of the model used is crucial. ResNet50 is better suited for use in contexts with limited resources since it provides an acceptable level of accuracy while keeping the model's complexity reasonable. Second, the computational effectiveness of a deep learning model is directly impacted by the number of layers in the model. The 50-layer ResNet50 model achieves a reasonable balance between model complexity and effectiveness. Longer prediction durations may be experienced with deeper variations like ResNet101 and ResNet152 since they require much more calculations during inference. Faster inference times are essential for a user-facing application, like the website for this graduation project, in order to deliver a smooth and responsive user experience. With faster predictions made possible by ResNet50, consumers may be sure to get wine suggestions right away. Third, the 50 layers of ResNet50 offer enough depth to extract useful feature representations from photos of wine. For correct classification, the model must be able to extract essential visual cues, particularly in the case of wine bottles where even minute variations in labels, colors, and shapes can have a big influence on categorization. ResNet50 can successfully learn and distinguish between distinct wine qualities thanks to its intermediate layers, which collect hierarchical information. ResNet50 also maintains a compromise between performance and model size. ResNet101 and ResNet152 are deeper variations that have more parameters and slightly greater accuracy, but bigger model sizes as a result. ResNet50, on the other hand, offers a practical amount of settings, making it more deployable on the desired server or device. A lower model size results in reduced memory usage during inference, which is crucial for servers or other devices with constrained resources. Not to mention, pre-trained models are an excellent place to start for transfer learning, enabling quicker convergence and better results on a small dataset. A well-known architecture, ResNet50 has been widely used in several computer vision tasks. ResNet50's pre-trained weights are therefore easily accessible, which makes it simple to start the model with weights discovered from sizable image classification datasets. With a comparatively smaller dataset, this initialization enables improved generality as well as quicker training completion for classification problem. In summary, ResNet50 was chosen above other ResNet design variations because to its balance of model complexity, computational effectiveness, and classification performance. While the model's computational effectiveness and reduced model size made it a better option for real-time or almost real-time inference on the graduating project's website, the model's depth and feature representation skills were determined to be enough for correct wine classification. ResNet50's pre-trained weights make it possible to properly fine-tune the model for the wine classification job, producing precise and effective wine recommendations for consumers.

CHAPTER 6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

This graduation project's successful conclusion represents a big step forward in the creation of an AI-driven wine suggestion engine. Data collection, preparation, model architecture customization and training were just a few of the tasks carried out during the project that helped to make the "Find Similar Wines" feature on the online platform successful. I've included the main achievements, current difficulties, noteworthy contributions, and important lessons gained below.

6.1.1 Achievements

a, Data gathering

I thoroughly investigated a variety of wine-selling websites before deliberately selecting "ruouvang24h.com" as the main page for data crawling. This choice helped to create a dataset that was structured without having to deal with repetitions and anomalies from numerous places.

b, Data preprocessing

I managed empty rows and data shortages throughout the data preparation stage to make sure the dataset only comprised reliable and necessary data. The dataset's dependability and quality were greatly increased by this phase.

c, Data storing

The gathered and cleaned data was really kept in a MongoDB database. Throughout both model training and real-time website functioning, this database made it simple to access and retrieve wine-related details.

d, Data augmentation on training set

Data augmentation techniques were used on the training set to get around the problems caused by the lack of data. This strategy was successful in broadening the dataset's variety, which improved classification accuracy and the model's capacity to generalize to new data.

e, Model architecture customization

I deliberately chose ResNet50 from among the several available deep learning architectures because of its harmony between depth and performance. After that, I customized it to handle only 75 wineries.

f, Model training

The enhanced dataset was used to train the ResNet50 model, which made use of pre-trained weights to hasten convergence and boost classification accuracy in general. On the test set, the trained model produced encouraging results, accurately recommending wines to use the "Find Similar Wines" feature.

g, API development

A strong API was created in addition to adding the "Find Similar Wines" feature to the website to allow for simple interaction across the frontend and backend elements. The website can get wine data from the MongoDB database and present customized wine suggestions to users in real-time thanks to the API, which enables effective data interchange. This API development improves the website's responsiveness and general speed, resulting in a fluid and engaging user experience.

h, Demonstration website

A fully working demonstration website was created as one of the project's deliverables to show off the possibilities of the wine suggestion engine. The demonstration website offers consumers a pleasant and convenient environment to explore and get a feel for the "Find Similar Wines" capability. This website acts as a usable example of the study's findings and enables interested parties to picture the possible effects of incorporating such a recommendation engine into a real-world wine-selling system.

6.1.2 Ongoing challenges

Although the "Find Similar Wines" feature and AI model have been successfully deployed, further work is still needed to improve the website's user interface and experience. To build a more reliable and appealing platform for wine connoisseurs, further features and enhancements are required.

6.1.3 Notable contributions

a, "Find Similar Wines" functionality

The "Find Similar Wines" component has been successfully implemented, and this is the research's most important contribution. Users may utilize this effective feature to explore wines with comparable aesthetic features, giving them a wider range of wines depending on their tastes.

b, AI model for wine classification

The improvement of the AI wine categorization model helps to improve the precision and effectiveness of the wine suggestion engine. The model's capacity to correctly classify wines based on their labels and other visual components improves the website's operation overall.

6.1.4 Valuable lessons learned

Data quality is essential since the dependability and performance of AI models are greatly impacted by the time spent on data collection, preprocessing, and curation. More reliable and accurate predictions are produced by a well-prepared dataset. In order for an AI system to be effective and performant, model selection criteria must strike a compromise among complicated models, computing efficiency, and classification accuracy, especially in real-time applications. The success of the website depends on its user-centric design, which places a high priority on user involvement and happiness. A compelling and pleasurable wine discovery experience is produced through integrating consumer suggestions and following to user-centric design guidelines.

6.2 Future work

Future improvements to the wine suggestion system will depend on a number of crucial activities. First off, concentrating on developing websites is essential for improving the user experience and building a platform that is more aesthetically pleasing and user-friendly. Priority will be given to enhancing the customer experience through integrating preferences and input from users. The wide range of wine photographs will also rise as a result of extending the collection by crawling from several trustworthy wine-selling websites, creating a more complete and representative dataset. Exploring cutting-edge deep learning architectures and optimizing them for wine classifications is a potential approach for improving model performance. Improved accuracy and quicker completion during model training may be achieved by applying transfer learning to cutting-edge models and performing hyperparameter tweaking. Additionally, examining the use of natural language processing (NLP) methods to the analysis of wine descriptions and reviews might provide insightful information for tailored wine recommendations. The system's capacity to offer customized wine choices may be improved by extending the "Find Similar Wines" capability to integrate user preferences and suggestions based on previous user interactions. The wine suggestion system can develop into a more complex and user-friendly platform by tackling these upcoming challenges, enhancing the wine discovery experience for consumers.

REFERENCE

[1] M. Yuan, L. Zhang, X.-Y. Li, and H. Xiong, *Comprehensive and efficient data labeling via adaptive model scheduling*, 2020. arXiv: `2002.05520 [cs.LG]`.

[2] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, *Image data augmentation for deep learning: A survey*, 2022. arXiv: `2204.08610 [cs.CV]`.

[3] I. Ofeidis, D. Kiedanski, and L. Tassiulas, *An overview of the data-loader landscape: Comparative performance analysis*, 2022. arXiv: `2209.13705 [cs.DC]`.

[4] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: `1512.03385 [cs.CV]`.

[5] F. Zhuang, Z. Qi, K. Duan, *et al.*, *A comprehensive survey on transfer learning*, 2020. arXiv: `1911.02685 [cs.LG]`.

[6] Y. Wang and X. Wang, *A unified study of machine learning explanation evaluation metrics*, 2022. arXiv: `2203.14265 [cs.LG]`.

[7] S. Kanoje, V. Powar, and D. Mukhopadhyay, *Using mongodb for social networking website*, 2015. arXiv: `1503.06548 [cs.DB]`.

[8] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: `1409.1556 [cs.CV]`.

[9] C. Szegedy, W. Liu, Y. Jia, *et al.*, *Going deeper with convolutions*, 2014. arXiv: `1409.4842 [cs.CV]`.

[10] M. Tan and Q. V. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, 2020. arXiv: `1905.11946 [cs.LG]`.