

Wine Quality Prediction

AIML NIELIT 6 Weeks Course

Abstract

This project predicts wine quality based on various chemical properties, aiming to provide a reliable and efficient method for assessing wine standards

Suramya Singh
suramya.singh03@gmail.com

Contents

Chapter 1

AI Ecosystem

1.1. Introduction	4
1.2. benefits of incorporating AI ecosystems	5

Chapter 2

Python Programming

2.1 Introduction	7
2.2 Why Python?	8
2.3 Python programming with Colab	9

Chapter 3

NumPy

3.1 Introduction	10
------------------------	----

Chapter 4

EDA with Pandas

4.1 Introduction	11
------------------------	----

Chapter 5

Machine Learning

5.1 Introduction	12
------------------------	----

Chapter 6

Wine Quality Prediction

6.1 About Project	14
6.2 Dataset	14
6.3 Pre-Processing Data	15
6.3.1 Import the packages	15
6.3.2 reading the CSV file	15
6.3.3 cleaning the data - NumPy/Pandas	16
6.3.4 Visualize the clean data – matplotlib	16
6.3.5 Machine learning	17
6.4 Front End	
6.4.1 HTML / CSS/ JavaScript	18

6.4.2 Web server Development using Flask	19
6.5 Project walkthrough	20
References / Bibliography	21

List of Figures

Figure 1:AI Ecosystem

Figure 2: AI Tools

Figure 3: Google Colab

Figure 4: Pie chart (student Result column)

List of Program Code

Google Colab

Index.html

Result.html

App.py

Chapter 1

AI Ecosystem

1.1. Introduction

1. Definition and Overview of AI Ecosystem

- Describe what an AI ecosystem is: a network of interconnected AI technologies, tools, and platforms.
- Mention the key components: hardware, software, data, algorithms, and human expertise.
- Highlight the importance of collaboration among different stakeholders (researchers, developers, businesses, and policymakers).

2. Historical Context

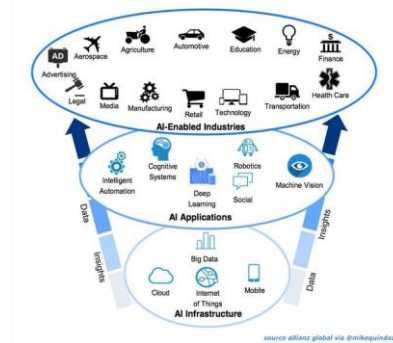
- Provide a brief history of AI development and the evolution towards ecosystem-based approaches.
- Discuss milestones in AI that led to the need for more integrated systems.

3. Current Landscape

- Explain the current state of AI ecosystems globally.
- Mention leading AI ecosystems such as those in Silicon Valley, China, and Europe.
- Discuss major players in the AI ecosystem (tech giants, startups, academic institutions).

4. Purpose and Scope of the Discussion

- Outline the goals of the discussion: to understand what AI ecosystems are and explore their benefits.
- Explain the scope: focusing on the integration, collaboration, and innovation within AI ecosystems.



1.2. Benefits of Incorporating AI Ecosystems

1. Enhanced Collaboration and Innovation

- Describe how AI ecosystems foster collaboration among diverse stakeholders.
- Discuss the role of open-source platforms and shared resources in driving innovation.
- Provide examples of successful AI collaborations resulting in breakthroughs.

2. Accelerated Development and Deployment

- Explain how ecosystems streamline the development and deployment of AI solutions.
- Mention the advantages of having access to shared tools, libraries, and frameworks.
- Highlight the role of cloud computing and AI-as-a-Service models in rapid deployment.

3. Improved Data Access and Utilization

- Discuss the importance of data in AI and how ecosystems enhance data sharing and integration.
- Explain how AI ecosystems improve data quality, availability, and interoperability.
- Provide examples of data collaborations that have led to significant AI advancements.

4. Economic Growth and Competitiveness

- Explain how AI ecosystems contribute to economic growth by creating new markets and job opportunities.
- Discuss how ecosystems help companies stay competitive by leveraging cutting-edge AI technologies.
- Highlight the role of government and policy in supporting AI ecosystem growth.

5. Enhanced Scalability and Flexibility

- Describe how ecosystems allow for scalable AI solutions that can grow with demand.
- Discuss the flexibility provided by modular AI components and microservices.
- Mention the role of AI marketplaces and platforms in offering scalable solutions.

6. Ethical and Responsible AI Development

- Explain how ecosystems promote ethical AI practices through shared standards and guidelines.
- Discuss the role of collaborative efforts in addressing AI ethics, fairness, and transparency.
- Provide examples of initiatives within AI ecosystems focused on responsible AI development.

7. Case Studies and Real-World Examples

- Present case studies of successful AI ecosystems and their impact.
- Discuss specific projects or initiatives that exemplify the benefits of AI ecosystems.
- Highlight lessons learned and best practices from these case studies.

Chapter 2

Python Programming

2.1 Introduction

1. Overview of Python

- Brief history of Python: Created by Guido van Rossum and first released in 1991.
- General characteristics: High-level, interpreted, and general-purpose programming language.

2. Popularity and Use Cases

- Discuss Python's popularity in various domains such as web development, data science, artificial intelligence, automation, and scientific computing.
- Mention some of the key organizations and projects that utilize Python.

3. Community and Ecosystem

- Highlight the large and active Python community.
- Discuss the extensive ecosystem of libraries and frameworks that support Python development.

2.2. Why Python?

1. Simplicity and Readability

- Explain how Python's syntax is designed to be readable and straightforward, making it easy for beginners to learn and for developers to maintain code.
- Provide examples of Python code demonstrating its simplicity.

2. Versatility and Flexibility

- Discuss the wide range of applications for which Python can be used.
- Mention Python's ability to integrate with other languages and technologies.

3. Extensive Libraries and Frameworks

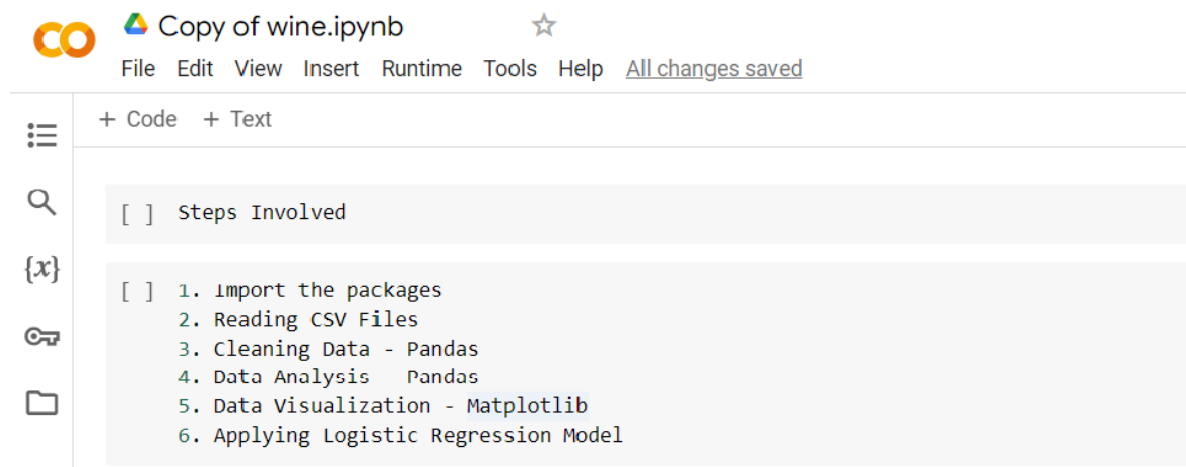
- Highlight key libraries such as NumPy, pandas, matplotlib, and frameworks like Django and Flask.

- Discuss how these libraries and frameworks accelerate development and reduce the need for writing code from scratch.
- 4. **Strong Community Support**
 - Emphasize the role of Python's community in providing support through forums, documentation, and tutorials.
 - Mention popular community platforms such as Stack Overflow, GitHub, and the Python Software Foundation.
- 5. **Industry Adoption**
 - Provide examples of industries and companies that heavily use Python.
 - Discuss job market demand for Python developers.

2.3. Python Programming with Colab

1. **Introduction to Google Colab**
 - Describe what Google Colab is: a free cloud-based Jupyter notebook environment provided by Google.
 - Mention the benefits of using Colab, such as no installation required and access to powerful computing resources.
2. **Getting Started with Colab**
 - Provide a step-by-step guide on how to start using Colab, including how to create a new notebook and basic interface navigation.
 - Explain how to install and import Python libraries in a Colab notebook.
3. **Key Features of Colab**
 - Discuss important features like collaboration (sharing and co-editing notebooks), integration with Google Drive, and access to GPUs and TPUs.
 - Mention how Colab can be used for teaching, learning, and collaborative projects.
4. **Writing and Executing Python Code**
 - Show examples of writing and running Python code in Colab.
 - Explain the process of writing code cells, running cells, and viewing outputs.
5. **Data Science and Machine Learning with Colab**
 - Describe how Colab is particularly useful for data science and machine learning projects.
 - Provide examples of loading datasets, performing data analysis, and training machine learning models within Colab.
6. **Saving and Sharing Work**

- Explain how to save notebooks to Google Drive or download them.
- Discuss options for sharing notebooks with others and controlling access permissions.



Google Colab Link-

<https://colab.research.google.com/drive/1hh0L8O1bub7lCbBUUx9slj72JTUjSR8i?usp=sharing>

Chapter 3

NumPy

3.1 Introduction

NumPy, short for Numerical Python, is a fundamental library for scientific computing in Python. Created by Travis Oliphant in 2005, it provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy serves as the backbone for many other scientific computing libraries in Python, including SciPy, pandas, and scikit-learn, making it an essential tool for data scientists and engineers.

Key Features:

- **N-dimensional array object (ND array):** Efficient storage and manipulation of large datasets.
- **Broadcasting:** Performing arithmetic operations on arrays of different shapes.
- **Mathematical Functions:** A wide range of mathematical operations on arrays.
- **Integration Tools:** Facilitating integration with C/C++ and Fortran code.
- **Specialized Functions:** Linear algebra, Fourier transform, and random number generation.

Importance and Use Cases: NumPy is crucial for various fields such as data analysis, machine learning, financial modelling, and scientific simulations due to its speed and efficiency. Its ability to perform complex mathematical operations on large datasets with minimal code makes it a preferred choice over traditional Python lists.

Comparison with Other Libraries: Compared to MATLAB and R, NumPy offers better performance through vectorized operations. While pandas is used for data manipulation, it relies on NumPy for its underlying data structures and operations.

Community and Ecosystem: NumPy boasts a vibrant community contributing to its continuous development and support. It is a foundational library in the Python ecosystem, supporting a wide array of scientific and data analysis tools.

Chapter 4

EDA with Pandas

4.1 Introduction

Overview of Pandas – Pandas is a powerful and versatile Python library designed for data manipulation and analysis. Developed by Wes McKinney in 2008, Pandas provides data structures and functions needed to work with structured data seamlessly. It is widely used in data science, statistics, and various fields that require data analysis.

Key Features:

- **Data Frame and Series:** The primary data structures in Pandas. Data Frame represents a 2-dimensional labelled data structure, while Series is a 1-dimensional labelled array.
- **Data Handling:** Efficiently handles missing data, performs data alignment, and provides tools for reshaping and pivoting datasets.
- **Data Wrangling:** Allows for easy merging, joining, and concatenation of datasets.
- **Input/Output Tools:** Supports reading and writing data from various file formats like CSV, Excel, SQL databases, and more.

Importance of EDA: Exploratory Data Analysis (EDA) is a critical step in the data analysis process. It involves examining the main characteristics of the data, often visualizing it, and summarizing its main features. EDA helps in:

- **Understanding Data:** Gaining insights into data structure and patterns.
- **Detecting Anomalies:** Identifying missing values, outliers, and errors.
- **Formulating Hypotheses:** Generating questions and hypotheses about the data.
- **Preparing for Modelling:** Informing feature selection and engineering for machine learning models.

Why Use Pandas for EDA: Pandas is ideal for EDA due to its robust data manipulation capabilities and easy integration with other data analysis and visualization libraries like NumPy, Matplotlib, and Seaborn. Its intuitive syntax and rich functionality make it a preferred tool for data analysts and scientists.

Chapter 5

Machine Learning

5.1 Introduction

Overview of Machine Learning Machine Learning (ML) is a subset of artificial intelligence (AI) that enables computers to learn from and make decisions based on data. Instead of being explicitly programmed to perform a task, ML algorithms use statistical techniques to infer patterns and make predictions. This capability has revolutionized many fields, from healthcare and finance to marketing and autonomous systems.

Key Concepts:

- **Supervised Learning:** Involves training a model on a labelled dataset, where the output is known. Examples include regression and classification tasks.
- **Unsupervised Learning:** Deals with unlabelled data and aims to find underlying patterns or structures, such as clustering and dimensionality reduction.
- **Semi-Supervised Learning:** Combines both labelled and unlabelled data to improve learning accuracy.
- **Reinforcement Learning:** Involves training agents to make a sequence of decisions by rewarding them for good actions and penalizing them for bad ones.

Importance and Applications: Machine learning is pivotal in numerous applications:

- **Healthcare:** Predicting disease outbreaks, personalizing treatment plans, and diagnosing diseases from medical images.
- **Finance:** Fraud detection, algorithmic trading, and credit scoring.
- **Retail:** Customer segmentation, recommendation systems, and inventory management.
- **Autonomous Vehicles:** Real-time decision making, object detection, and path planning.

Machine Learning Workflow:

1. **Data Collection:** Gathering relevant data from various sources.
2. **Data Preprocessing:** Cleaning and transforming data to make it suitable for analysis (handling missing values, encoding categorical variables, etc.).
3. **Feature Engineering:** Creating new features or selecting the most relevant features to improve model performance.
4. **Model Selection:** Choosing the appropriate machine learning algorithm (e.g., linear regression, decision trees, neural networks).
5. **Model Training:** Feeding the algorithm with training data to learn patterns and make predictions.
6. **Model Evaluation:** Assessing the model's performance using metrics such as accuracy, precision, recall, and F1 score.
7. **Model Tuning:** Adjusting hyperparameters to optimize the model's performance.
8. **Model Deployment:** Integrating the model into a production environment to make real-time predictions.

Why Python for Machine Learning: Python is the preferred language for machine learning due to its simplicity, readability, and extensive ecosystem of libraries. Key libraries include:

- **NumPy:** For numerical computations.
- **pandas:** For data manipulation and analysis.
- **scikit-learn:** For traditional machine learning algorithms.
- **TensorFlow and PyTorch:** For deep learning and neural networks.
- **Matplotlib and Seaborn:** For data visualization.

Chapter 6

Wine Quality Prediction

6.1 About the Project

The Wine Quality Prediction project aims to utilize machine learning techniques to predict the quality of wine based on various chemical properties. By leveraging logistic regression, this project provides a reliable and efficient method for assessing wine quality, which can be invaluable for vintners, sommeliers, and wine enthusiasts. The project involves data preprocessing, exploratory data analysis, model training, and evaluation to achieve accurate predictions.

6.2 Dataset

The dataset used for this project consists of chemical properties of red and white wines, which include features like acidity, sugar content, pH levels, and alcohol content. Each wine sample is also assigned a quality rating ranging from 0 to 10. The dataset is publicly available from the UCI Machine Learning Repository and is widely used for benchmarking machine learning models.

Key Features:

- Fixed acidity
- Volatile acidity
- Citric acid
- Residual sugar
- Chlorides
- Free sulfur dioxide
- Total sulfur dioxide
- Density
- pH
- Sulphates
- Alcohol
- Quality (target variable)

6.3 Pre-Processing Data

6.3.1 Import the Packages

To begin the data preprocessing, necessary Python packages need to be imported. This includes NumPy for numerical operations, pandas for data manipulation, and Matplotlib for data visualization.

Importing Packages

```
[ ] import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
```

6.3.2 Reading the CSV File

The dataset is typically provided in a CSV format. The following code reads the CSV file into a pandas Data Frame

```
import pandas as pd
path = '/content/drive/MyDrive/wineQT.csv'
data = pd.read_csv(path)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	0
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5	1
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5	2
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6	3
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	4

6.3.3 Cleaning the Data- NumPy/Pandas

Data cleaning is essential to ensure the dataset is free of errors and missing values, and that it is properly formatted for analysis.

```
[ ] data.head()
```

Show hidden output

```
[ ] data.describe()
```

Show hidden output

```
data.isnull().sum()
```

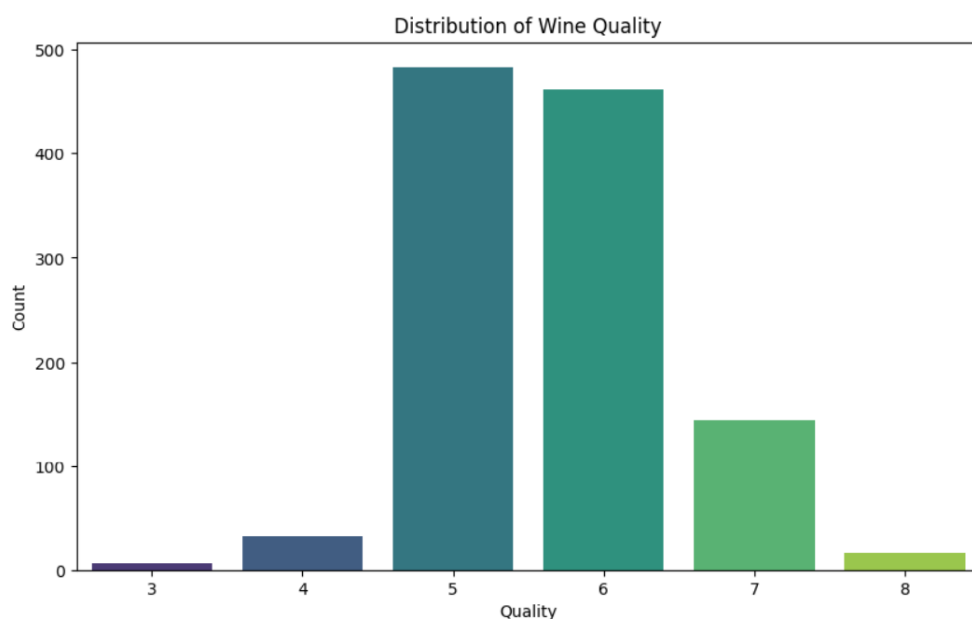
Show hidden output

6.3.4 Visualize the Clean Data- Matplotlib

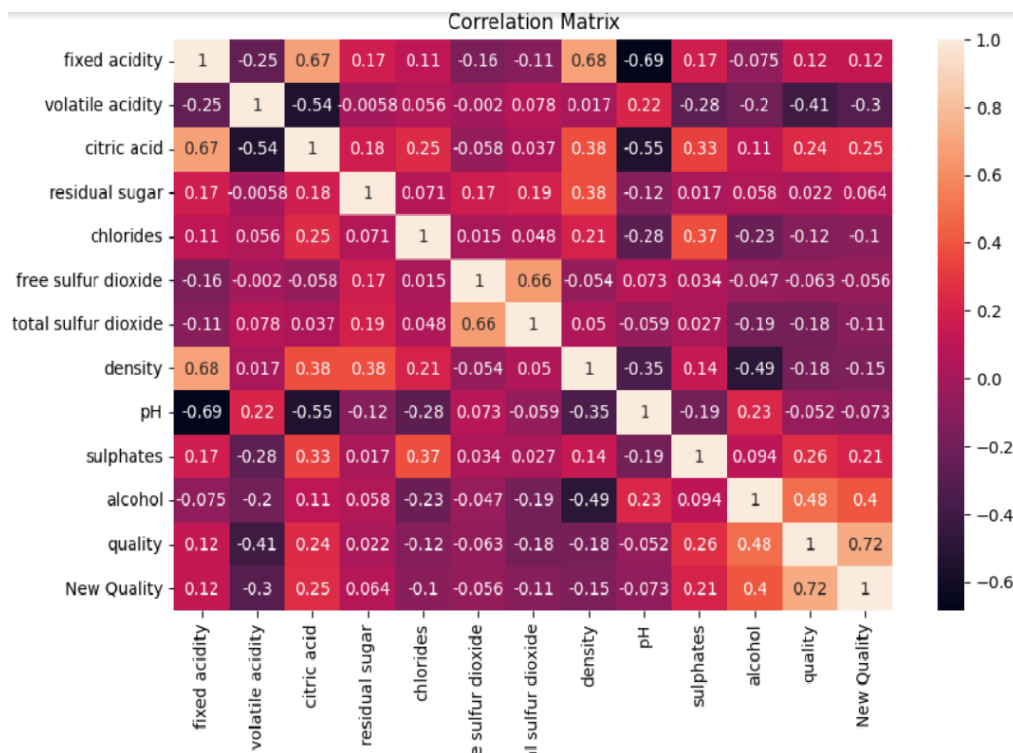
Visualizing the data helps in understanding its distribution and identifying any patterns or anomalies.

✓
1s

```
plt.figure(figsize=(10, 6))  
sns.countplot(x='quality', data=data, palette='viridis')  
plt.title('Distribution of Wine Quality')  
plt.xlabel('Quality')  
plt.ylabel('Count')  
plt.show()
```



```
correlation = data.corr()  
plt.figure(figsize=(10, 6))  
sns.heatmap(correlation, annot=True)  
plt.title("Correlation Matrix")  
plt.show()
```

6.3.5 Machine Learning

Data Preparation: Separate the features and the target variable, and split the data into training and testing sets.

```
X=data.drop(['quality','New Quality'],axis=1)
Y=data['New Quality']
```

```
[ ] X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=7)
```

Model Training: Train a logistic regression model using the training data.

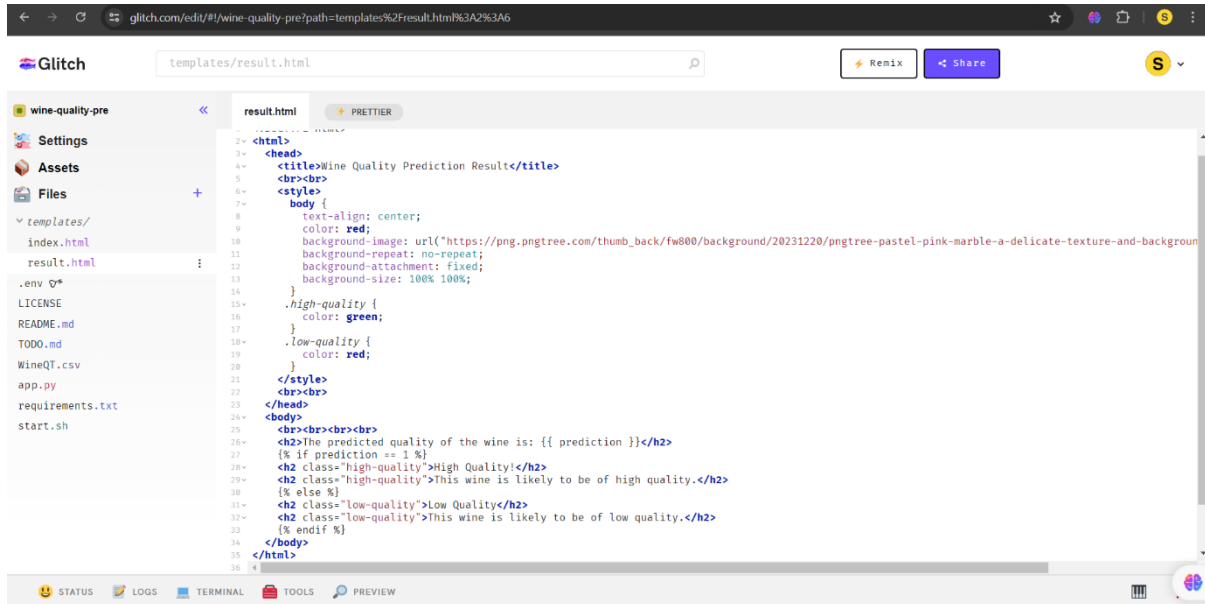
```
model=LogisticRegression()
model.fit(X_train,Y_train)
y_pred=model.predict(X_test)

model_res.loc[len(model_res)]=['Logistic Regression',accuracy_score(Y_test,y_pred)]
model_res
```

6.4 Front End

6.4.1 HTML/CSS/JAVASCRIPT

This is result.html and index.html page are made using html , CSS, JavaScript.



```
2 <!-- HTML -->
3 <html>
4 <head>
5 <title>Wine Quality Prediction Result</title>
6 <br><br>
7 <style>
8   body {
9     text-align: center;
10    color: red;
11    background-image: url("https://png.pngtree.com/thumb_back/fw800/background/20231220/pngtree-pastel-pink-marble-a-delicate-texture-and-backgroun
12    background-repeat: no-repeat;
13    background-attachment: fixed;
14    background-size: 100% 100%;
15  }
16  .high-quality {
17    color: green;
18  }
19  .low-quality {
20    color: red;
21  }
22 </style>
23 <br><br>
24 </head>
25 <body>
26 <br><br><br><br>
27 <h2>The predicted quality of the wine is: {{ prediction }}</h2>
28 {% if prediction == 1 %}
29 <h2 class="high-quality">High Quality!</h2>
30 <h2 class="high-quality">This wine is likely to be of high quality.</h2>
31 {% else %}
32 <h2 class="low-quality">Low Quality</h2>
33 <h2 class="low-quality">This wine is likely to be of low quality.</h2>
34 {% endif %}
35 </body>
36 </html>
```

This is result.html

6.4.2 Web server Development using Flask

wine-quality-pre

Settings

Assets

Files

templates/

index.html

result.html

.env

LICENSE

README.md

TODO.md

WineQT.csv

app.py

requirements.txt

start.sh

```
1 from flask import Flask, render_template, request
2 import pandas as pd
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.model_selection import train_test_split
5 from sklearn.metrics import accuracy_score, confusion_matrix
6
7 app = Flask(__name__)
8
9 # Load data
10 data = pd.read_csv('WineQT.csv')
11 data.drop('Id',axis=1,inplace=True)
12 data['New Quality'] = data['quality'].apply(lambda x: 1 if x >= 7 else 0)
13 X = data.drop(['quality', 'New Quality'], axis=1)
14 Y = data['New Quality']
15
16 # Split data
17 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=7)
18
19 # Define and train model
20 model = LogisticRegression()
21 model.fit(X_train, Y_train)
22
23 @app.route('/')
24 def home():
25     return render_template('index.html')
26
27 @app.route('/predict', methods=['POST'])
28 def predict():
29     if request.form:
30         fixed_acidity = float(request.form['fixed_acidity'])
31         volatile_acidity = float(request.form['volatile_acidity'])
32         citric_acid = float(request.form['citric_acid'])
33         residual_sugar = float(request.form['residual_sugar'])
34         chlorides = float(request.form['chlorides'])
35         free_sulfur_dioxide = float(request.form['free_sulfur_dioxide'])
36         total_sulfur_dioxide = float(request.form['total_sulfur_dioxide'])
37         density = float(request.form['density'])
38         pH = float(request.form['pH'])
39         sulphates = float(request.form['sulphates'])
40         alcohol = float(request.form['alcohol'])
41
42         input_data = pd.DataFrame({
43             'fixed_acidity': [fixed_acidity],
44             'volatile_acidity': [volatile_acidity],
45             'citric_acid': [citric_acid],
46             'residual_sugar': [residual_sugar],
47             'chlorides': [chlorides],
48             'free_sulfur_dioxide': [free_sulfur_dioxide],
49             'total_sulfur_dioxide': [total_sulfur_dioxide],
50             'density': [density],
51             'pH': [pH],
52             'sulphates': [sulphates],
53             'alcohol': [alcohol]
54         })
55         prediction = model.predict(input_data)
56         return render_template('result.html', prediction=prediction[0])
57     else:
58         return 'Error: No form data provided'
59
60 if __name__ == '__main__':
61     app.run()
```

wine-quality-pre

Settings

Assets

Files

templates/

index.html

result.html

.env

LICENSE

README.md

TODO.md

WineQT.csv

app.py

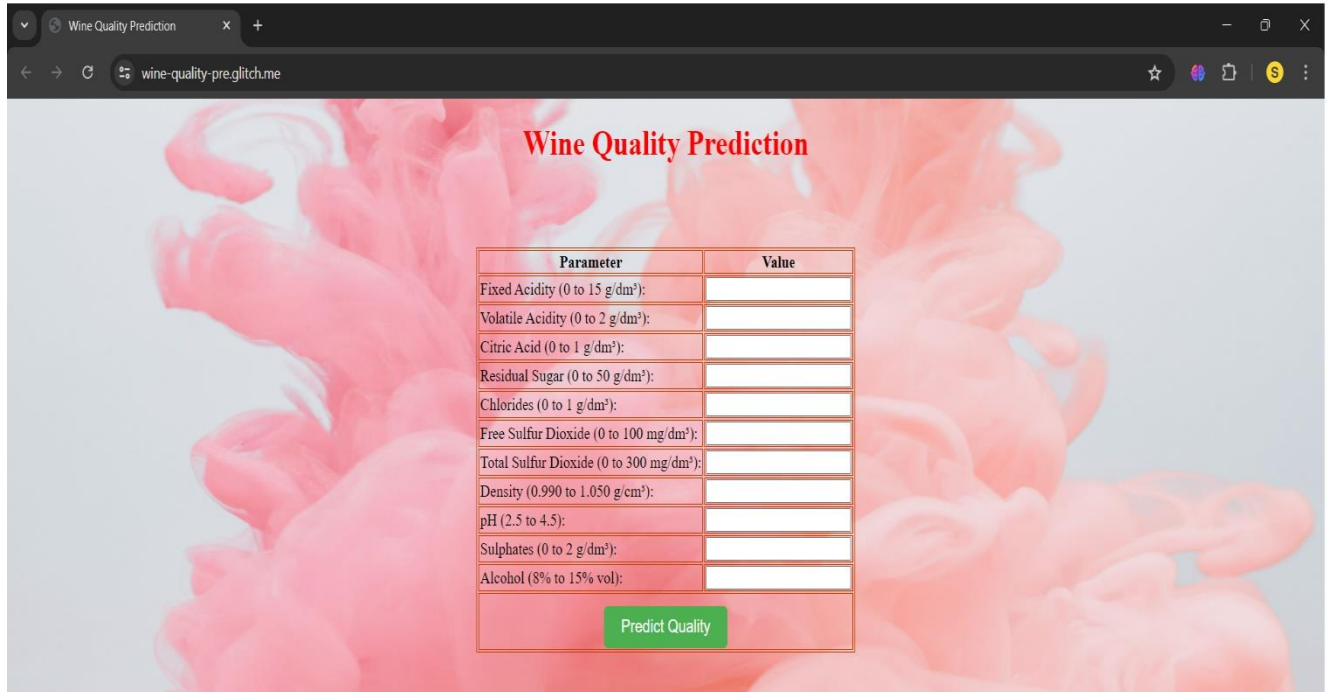
requirements.txt

start.sh

```
27 @app.route('/predict', methods=['POST'])
28 def predict():
29     if request.form:
30         fixed_acidity = float(request.form['fixed_acidity'])
31         volatile_acidity = float(request.form['volatile_acidity'])
32         citric_acid = float(request.form['citric_acid'])
33         residual_sugar = float(request.form['residual_sugar'])
34         chlorides = float(request.form['chlorides'])
35         free_sulfur_dioxide = float(request.form['free_sulfur_dioxide'])
36         total_sulfur_dioxide = float(request.form['total_sulfur_dioxide'])
37         density = float(request.form['density'])
38         pH = float(request.form['pH'])
39         sulphates = float(request.form['sulphates'])
40         alcohol = float(request.form['alcohol'])
41
42         input_data = pd.DataFrame({
43             'fixed_acidity': [fixed_acidity],
44             'volatile_acidity': [volatile_acidity],
45             'citric_acid': [citric_acid],
46             'residual_sugar': [residual_sugar],
47             'chlorides': [chlorides],
48             'free_sulfur_dioxide': [free_sulfur_dioxide],
49             'total_sulfur_dioxide': [total_sulfur_dioxide],
50             'density': [density],
51             'pH': [pH],
52             'sulphates': [sulphates],
53             'alcohol': [alcohol]
54         })
55         prediction = model.predict(input_data)
56         return render_template('result.html', prediction=prediction[0])
57     else:
58         return 'Error: No form data provided'
59
60 if __name__ == '__main__':
61     app.run()
```

6.5 Project Walkthrough

Front page

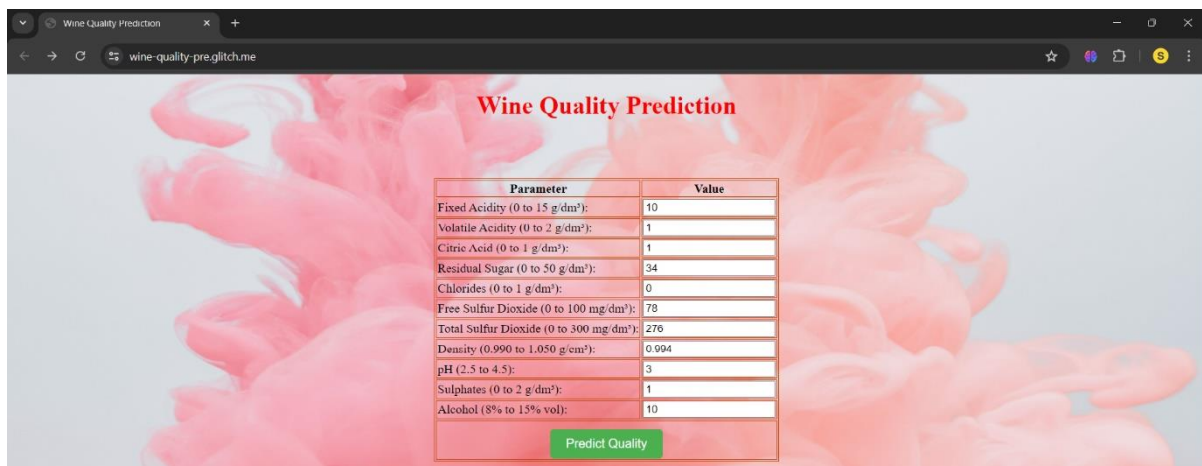


Parameter	Value
Fixed Acidity (0 to 15 g/dm ³):	
Volatile Acidity (0 to 2 g/dm ³):	
Citric Acid (0 to 1 g/dm ³):	
Residual Sugar (0 to 50 g/dm ³):	
Chlorides (0 to 1 g/dm ³):	
Free Sulfur Dioxide (0 to 100 mg/dm ³):	
Total Sulfur Dioxide (0 to 300 mg/dm ³):	
Density (0.990 to 1.050 g/cm ³):	
pH (2.5 to 4.5):	
Sulphates (0 to 2 g/dm ³):	
Alcohol (8% to 15% vol):	

Predict Quality

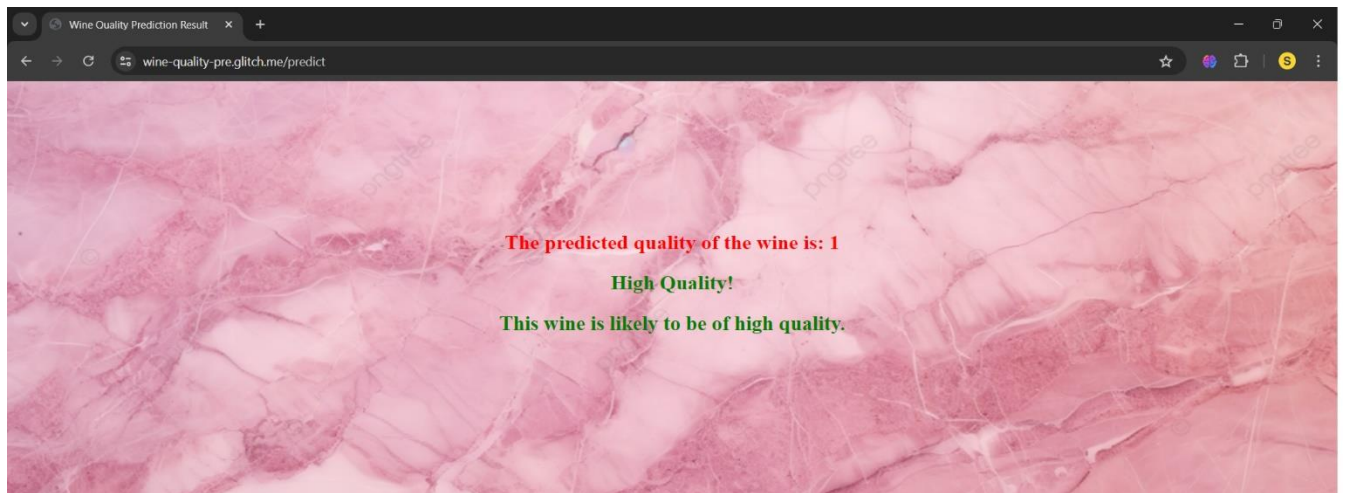
Result Page

When the requirements are filled it will provide result whether the wine according to given parameters are of good quality or bad quality.



Parameter	Value
Fixed Acidity (0 to 15 g/dm ³):	10
Volatile Acidity (0 to 2 g/dm ³):	1
Citric Acid (0 to 1 g/dm ³):	1
Residual Sugar (0 to 50 g/dm ³):	34
Chlorides (0 to 1 g/dm ³):	0
Free Sulfur Dioxide (0 to 100 mg/dm ³):	78
Total Sulfur Dioxide (0 to 300 mg/dm ³):	276
Density (0.990 to 1.050 g/cm ³):	0.994
pH (2.5 to 4.5):	3
Sulphates (0 to 2 g/dm ³):	1
Alcohol (8% to 15% vol):	10

Predict Quality



References and Bibliography

Kaggle dataset – <https://www.kaggle.com/datasets/yasserh/wine-quality-dataset>

Kaggle code- <https://www.kaggle.com/code/aditishelke19/classification-wine>

<https://www.kaggle.com/code/yukunaka1/pj1-wine-quality-prediction>

w3 schools - <https://www.w3schools.com/>