

Département Mathématiques et Informatiques

Branch :

Computer Science Engineering : Big Data and Cloud Computing

II-BDCC

Wine Quality Prediction Project

Students :

Abderrazzak NOUARI

Abdellatif HASSANI

Professor :

Mr. Soufiane HAMIDA

Année Universitaire : 2023-2024

Sommaire

Sommaire	2
Introduction	3
1. Problem Statement	4
2. Objectifs	4
3. Libraries Employed in the Project	4
4. Tools Utilized in the Project	6
5. Choosing the Model:	7
6. Steps that we followed	8
6.1. Dataset Selection	8
6.2. Data Analysis and Visualization	9
6.3. Data Pre-processing:	11
6.4. Train-Test Split:	12
6.5. Model Training:	12
6.6. Model Evaluation:	13
6.7. Console Predictive Model:	14
6.8. Model Deployment:	14
6.9. Web Application Predictive Model:	15
Conclusion	16

Introduction

This mini-project in machine learning focuses on predicting wine quality, a task that encompasses various stages of data analysis, modeling, and deployment. We navigated through distinct phases, starting from dataset exploration on Kaggle to the creation of a web-based application using Flask for wine quality prediction. This report documents the journey through each step, highlighting methodologies, challenges faced, and outcomes achieved. Through this project, the aim was to gain practical experience in applying machine learning techniques to real-world problems, furthering understanding of data science and software engineering concepts.

1. Problem Statement

The task at hand involves predicting the quality of red wine using the test data from the Red Wine Quality Dataset. This entails importing the dataset, performing data wrangling to ensure data quality, and conducting exploratory data analysis (EDA) through various visualizations such as histograms, boxplots, and scatter plots. Subsequently, the dataset will be modeled using different machine learning algorithms to determine the most effective approach for predicting wine quality.

2. Objectifs

- The primary objective of this project is to develop a Jupyter notebook within the Anaconda environment to import and examine the dataset. Specific goals include:
- Utilizing Pandas to clean and preprocess the data, ensuring its suitability for modeling.
- Employing scikit-learn to create machine learning models capable of accurately predicting wine quality.
- Utilizing Matplotlib to visualize and evaluate the performance of the models, providing insights into their effectiveness and areas for improvement.

By achieving these objectives, we aim to gain valuable insights into the factors influencing wine quality and develop robust predictive models to aid in quality assessment and decision-making processes within the wine industry.

3. Libraries Employed in the Project

3.1. NumPy:



NumPy is a fundamental package for numerical computing in Python.

It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

3.2. Pandas:



Pandas is a powerful library for data manipulation and analysis.

It offers data structures such as DataFrame and Series, which allow for easy handling of structured data.

Pandas simplifies tasks such as data cleaning, transformation, and exploration.

3.3. Matplotlib (plt): matplotlib

Matplotlib is a comprehensive library for creating static, interactive, and animated visualizations in Python.

It provides a MATLAB-like interface for generating plots, histograms, scatter plots, and more.

Matplotlib is highly customizable, allowing users to fine-tune their visualizations to suit specific requirements.

3.4. Seaborn:



Seaborn is built on top of Matplotlib and provides a higher-level interface for creating attractive statistical graphics.

It offers a wide range of built-in themes and color palettes, making it easy to create visually appealing plots.

Seaborn simplifies the process of creating complex visualizations such as heatmaps, pair plots, and violin plots.

3.5. Scikit-learn:



Scikit-learn is a versatile library for machine learning in Python.

It provides a wide range of tools for classification, regression, clustering, dimensionality reduction, and more.

Scikit-learn offers a consistent API, making it easy to experiment with different algorithms and techniques.

It includes modules for data preprocessing, model selection, evaluation metrics, and model deployment.

4. Tools Utilized in the Project

4.1. Anaconda:



Anaconda is a popular open-source distribution of Python and R programming languages for data science and machine learning tasks.

It comes bundled with a wide range of pre-installed libraries and tools essential for data analysis, including NumPy, Pandas, Matplotlib, and scikit-learn.

Anaconda provides an easy-to-use platform for managing Python environments, packages, and dependencies, making it ideal for developing and deploying machine learning projects.

4.2. Jupyter:



Jupyter Notebook is an interactive computing environment that allows users to create and share documents containing live code, equations, visualizations, and narrative text.

It supports various programming languages, including Python, R, and Julia, making it a versatile tool for data analysis, prototyping, and collaboration.

Jupyter Notebooks facilitate exploratory data analysis, model development, and documentation, enabling users to iteratively explore and analyze datasets.

4.3. Flask:



Flask is a lightweight web application framework for Python, designed to make it easy to build web applications and APIs.

It provides features such as routing, request handling, and template rendering, allowing developers to create web-based interfaces for machine learning models.

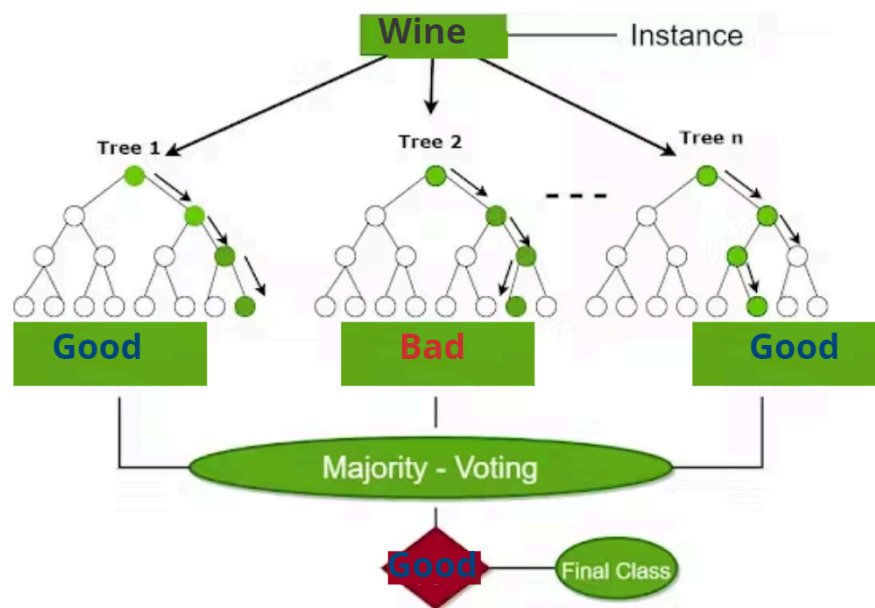
Flask is well-suited for deploying machine learning models as web services, enabling users to interact with models through web browsers or HTTP requests.

5. Choosing the Model:

The Random Forest Classifier was chosen as the primary model for training due to its robustness, versatility, and effectiveness in handling high-dimensional data.

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output, and the greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The algorithm's ability to capture complex relationships between features and the target variable makes it well-suited for the wine quality prediction task.

Random Forest inherently performs feature selection and handles outliers and non-linear relationships, which are common in real-world datasets like the one used in this project.

Moreover, Random Forest provides built-in mechanisms for estimating feature importance, enabling us to identify the key attributes influencing wine quality.

Overall, the Random Forest Classifier offers a powerful and interpretable solution for predicting wine quality, making it the ideal choice for this project.

6. Steps that we followed

6.1. Dataset Selection

The dataset used for this project was sourced from Kaggle, a platform for sharing and discovering datasets.

The dataset consists of various physicochemical properties of red wine samples, such as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality.

Column	Explanation
Fixed acidity	Percentage of Fixed acidity in wine
Volatile acidity	Percentage of Volatile acidity in wine
Citric acid	Percentage citric acid in wine
Residual sugar	Percentage of residual sugar in wine
chlorides	Percentage of chlorides in wine
Free sulphur dioxide	Percentage of Free sulphur dioxide in wine
Total sulphur dioxide	Percentage of Total sulphur dioxide in wine
density	Percentage of Density in wine
pH	Percentage of pH in wine
sulphates	Percentage of sulphates in wine
alcohol	Percentage of alcohol in wine
quality	Quality of Wine


```
In [4]: # first 5 rows of the dataset
wine_dataset.head()
```

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Quality serves as the target variable, with values ranging from 3 to 8, representing the quality rating of the wine.

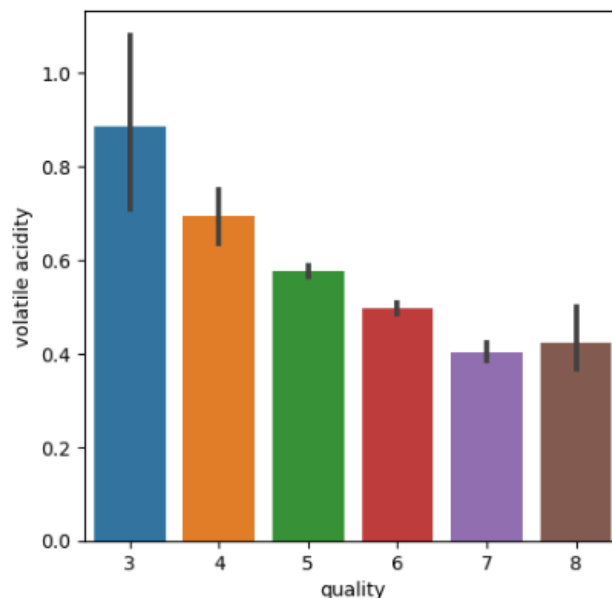
6.2. Data Analysis and Visualization

Data analysis and visualization were conducted using Python libraries such as Pandas, Matplotlib, and Seaborn.

Through histograms, boxplots, and scatter plots, we explored the relationships between different attributes and their impact on wine quality.

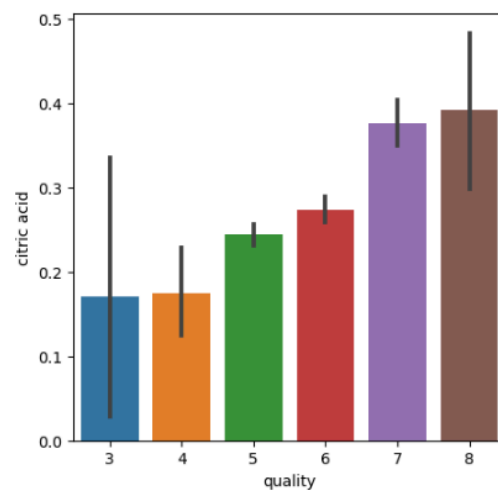
```
In [8]: # volatile acidity vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y='volatile acidity', data = wine_dataset)
```

Out[8]: <Axes: xlabel='quality', ylabel='volatile acidity'>



```
In [9]: # citric acid vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y='citric acid', data = wine_dataset)

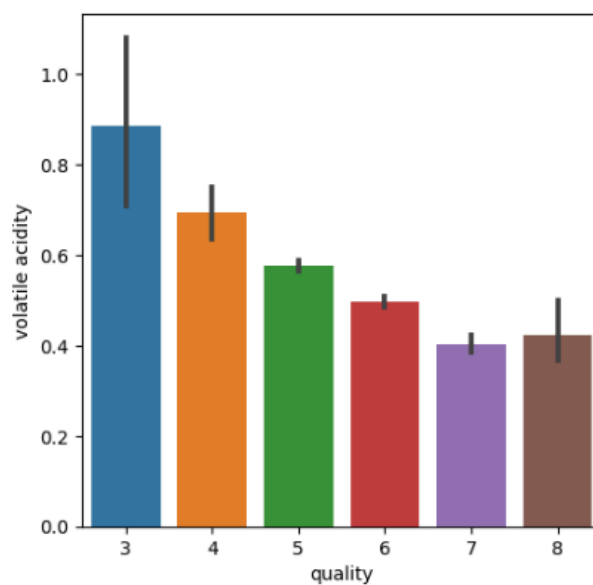
Out[9]: <Axes: xlabel='quality', ylabel='citric acid'>
```



Interesting insights were discovered, including correlations between certain attributes and wine quality, which guided feature selection and model training.

```
In [8]: # volatile acidity vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y='volatile acidity', data = wine_dataset)

Out[8]: <Axes: xlabel='quality', ylabel='volatile acidity'>
```



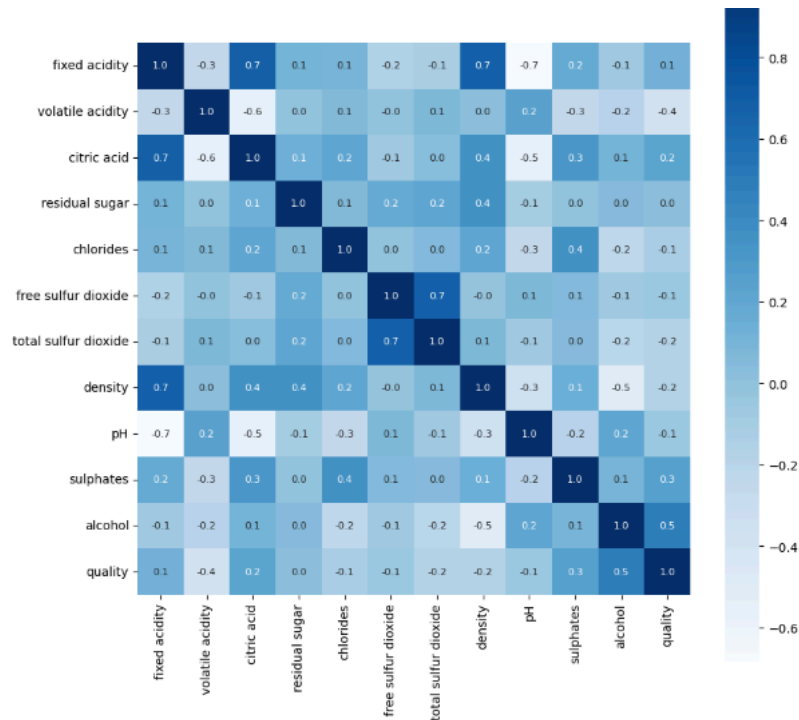
Correlation

1. Positive Correlation
2. Negative Correlation

```
In [10]: correlation = wine_dataset.corr()
```

```
In [11]: # constructing a heatmap to understand the correlation between the columns
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt = '.1f', annot = True, annot_kws={'size':8}, cmap = 'Blues')
```

Out[11]: <Axes: >



6.3. Data Pre-processing:

Data pre-processing involved handling missing values, if any, and ensuring data consistency and cleanliness.

```
In [5]: # checking for missing values
wine_dataset.isnull().sum()
```

```
Out[5]: fixed acidity      0
volatile acidity    0
citric acid         0
residual sugar      0
chlorides           0
free sulfur dioxide 0
total sulfur dioxide 0
density             0
pH                  0
sulphates           0
alcohol             0
quality             0
dtype: int64
```

6.4. Train-Test Split:

We split The dataset into training and testing sets using an 80:20 ratio.

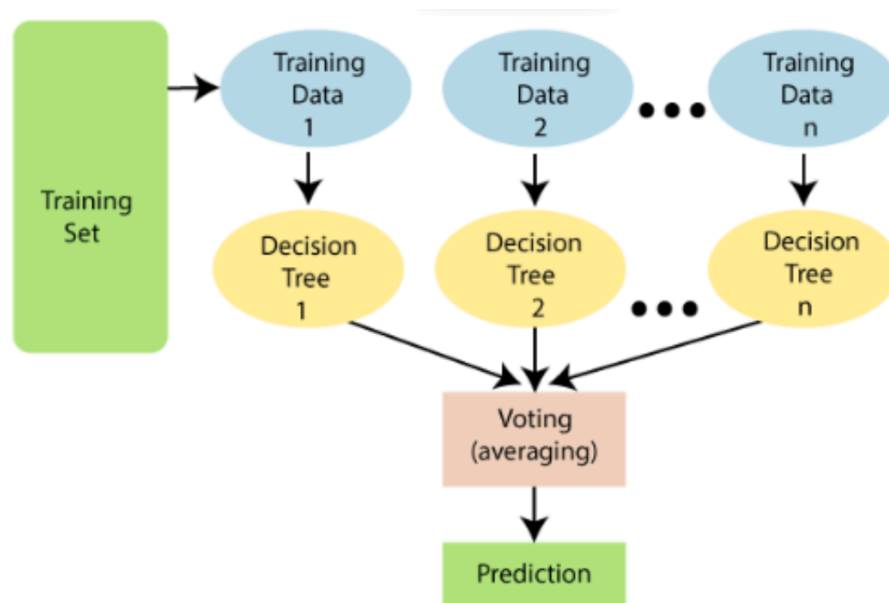
The purpose of this split is to train the model on a portion of the data and evaluate its performance on unseen data

```
In [17]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)

In [18]: print(Y.shape, Y_train.shape, Y_test.shape)
(1599,) (1279,) (320,)
```

6.5. Model Training:

The Random Forest Classifier algorithm was chosen for its ability to handle high-dimensional data and capture complex relationships between features and the target variable.



Hyperparameters were tuned to optimize model performance, including the number of trees in the forest and the maximum depth of each tree.

```
In [18]: model = RandomForestClassifier()

In [19]: model.fit(X_train, Y_train)

Out[19]: RandomForestClassifier
RandomForestClassifier()
```

6.6. Model Evaluation:

Model evaluation was performed using metrics such as accuracy, precision, recall, and F1-score.

The model achieved a high accuracy score on the test data, indicating its effectiveness in predicting wine quality.

```
In [20]: # accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [21]: print('Accuracy : ', test_data_accuracy)

Accuracy : 0.93125
```

6.7. Console Predictive Model:

We developed a console application to enable users to input wine attributes and receive predictions on the quality of the wine.

Users interact with the application by entering values for each attribute, and the trained model provides a prediction of the wine quality.

```
In [22]: input_data = (7.5,0.5,0.36,6.1,0.071,17.0,102.0,0.9978,3.35,0.8,10.5)

# changing the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the data as we are predicting the label for only one instance
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_resaped)
print(prediction)

if (prediction[0]==1):
    print('Good Quality Wine')
else:
    print('Bad Quality Wine')

[0]
Bad Quality Wine
```

6.8. Model Deployment:

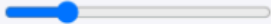
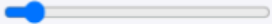
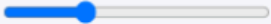







We exported The trained model using the joblib library, allowing for seamless integration into other applications or platforms.

Future enhancements could involve deploying the model as a web-based application, making it accessible to a wider audience.

6.9. Web Application Predictive Model:

Wine Quality Prediction

Predicted wine quality: Good

fixed acidity:  7.4	volatile acidity:  0.25	citric acid:  0.29
residual sugar:  2.2	chlorides:  0.054	free sulfur dioxide:  19
total sulfur dioxide:  49	density:  0.99666	pH:  3.4
sulphates:  0.76	alcohol:  10.9	

Predict

Conclusion

In conclusion, our project has provided valuable insights into the application of machine learning in real-world scenarios. Through diligent exploration and experimentation, we successfully developed and deployed a predictive model capable of evaluating wine quality. This project allowed us to gain hands-on experience with various Python libraries, some of which were new to us, further expanding our technical toolkit. Looking ahead, we remain eager to apply these newfound skills and insights to future projects, contributing to ongoing innovation and advancement in the field of data science and software engineering.