Shri Ramdeobaba College of Engineering and Management
Department of Electronics Engineering

Machine Learning [ECSP303]

Odd Semester –2024 -25

Wine Quality Prediction

Name	Aayush Agrawal , Arihant Jain
Batch/RollNo.	26, 27
Semester	5th Sem
Section	A / A2
Subject	Machine Learning Project
Name of Faculty	Prof. Pravin Dwaramwar

Git-hub link:-

https://github.com/Arihant0001/Machine-Learning-project.git

Abstract

This project will predict the quality of wine using its physicochemical properties by applying machine learning models. The prediction of wine quality helps winemakers control the quality and identify critical chemical factors that affect ratings of wine quality. The project is organized as a machine learning pipeline: data preprocessing, hyperparameter tuning, and finally the evaluation of multiple models. The proposed models are Logistic Regression, Random Forest, and Support Vector Machine with their related accuracy, F1-score, and interpretability at the end of the results with the conclusion of best model based on the features learned about feature importance
.

Introduction

This article introduces the context for predicting wine quality

- **Problem Formulation:** The wine quality has been determined by multiple physicochemical properties of wines. Predicting such quality may help in making production standards better while ensuring that required traits in wines are provided that enhance customer satisfaction.

- **Objective:** Design a machine learning model that would predict the quality of wines based on these chemical features and provide information on which properties have the highest influence on quality.

- **Significance:** This can benefit the wine industry by providing an understanding of the chemical composition that affects the quality rating, thereby making decisions on meeting market demand.

Related Work

The research related to the prediction of wine quality has focused on various machine learning and statistical methods. Some of the most successful approaches include:

- **Logistic Regression and Linear Models:** The model is rather intuitive and simple, it's not easily handled the case of more complex patterns among variables in data

- **Ensemble Learning (Random Forest, Gradient Boosting**): These can fit the data with very complex interactions but are usually very prone to overfitting the case and are much harder to be interpreted.

- **SVM and Neural Networks:** Suitable in the case of higher dimensionality of the input, but these have considerable computing effort. Models being of smaller size but of adequate tuning might make sense in terms of finding that perfect balance between performance and interpretability for the usual business applications within the industries.

Dataset and Features

- **Dataset:** The used dataset is Wine Quality Dataset from the UCI Machine Learning Repository. It is a description of red and white wine samples.

- **Features:** There are several physicochemical features in the data set, including the following:

  - **Fixed Acidity:** It is the tartaric acid level, giving an overall sense of acidity.

  - **Volatile Acidity:** The acetic acid in the wine, if it gets too high, makes a vinegary taste.

  - **Citric Acid:** Freshness and flavor.

  - **Residual Sugar:** The amount of sugar left over after fermentation contributes to sweetness.

- **Chlorides:** The salt content, which adds to taste.

- **Free Sulphur Dioxide and Total Sulphur Dioxide:** Antioxidants which do not allow the oxidation spoilage.

- **Density, PH and Alcohol Content:** taste, mouthfeel, Strength.

- **Target Variable (Quality):** It will be rated in terms of how good the wine tastes: 0-10.

The data was partitioned into training 80%, Testing 20% because model needs to know is how general it will fit.

Methodology

1. **Data Preprocessing**

    - **Handling missing values**: checked the given data for missing and made a list of the lines to delete.

    - **Data Scaling**: Used StandardScaler to standardize the features since models such as SVM and Logistic Regression perform better on normalized data.

    - **Outlier Detection**: Box plots were used for checking outliers, especially within acidity and residual sugar values, which might skew predictions from the model.

2. **Feature Engineering**

    - **Interaction Terms**: Calculated interaction terms such as residual sugar per pH level to see if combined features yield additional predictive power.

- **Dimensionality Reduction**: PCA was used, which mainly was exploratory, applied to reduce dimensionality in data, so that efficiency of computation may be enhanced.

3. **Model Selection and Training**

- **Logistic Regression** was used primarily because of the interpretability and is a weak predictor that fails in capturing the intricate and complex, non-linear relations in data.

- **Random Forest** Feature Importance insights with strong resistance towards overfitting with some hyperparameter tuning.

- **Support Vector Machine(SVM) -**Highly effective on very high dimensional data though costly in hyperparameter tuning.

4. **Cross Validation**

- The used **k-fold cross-validation**, with k=5 for each model, to try to evaluate the performance on different data splits and, thereby, reduce the possibility of overfitting.

5. **Hyperparameter Tuning**

- **Random Forest**: n_estimators, max_depth, min_samples_per_leaf

- **SVM**: The parameters C and gamma for RBF kernel are tuned by GridSearchCV in order to optimize the model's performance.

Experimentation and Evaluation Metrics

- **Accuracy :** This measure is the ratio of the total correct predictions out of all predictions. Its usefulness is seen when class distributions are balanced.

- **Precision and Recall:** Precision calculates the accuracy of positive predictions (minimizing false positives), while Recall measures the ability of the model to identify all relevant cases (minimizing false negatives).

- **F1-Score:** F1-score gives a balanced relation between Precision and Recall. Especially helpful for imbalanced data; a high F1-score means good performance on both Precision and Recall.

- **Confusion Matrix:** It will indicate the number of accurate and wrong classifications made for each class. It will indicate where the model is weakest.

Results Explanation of Every Model

Logistic Regression

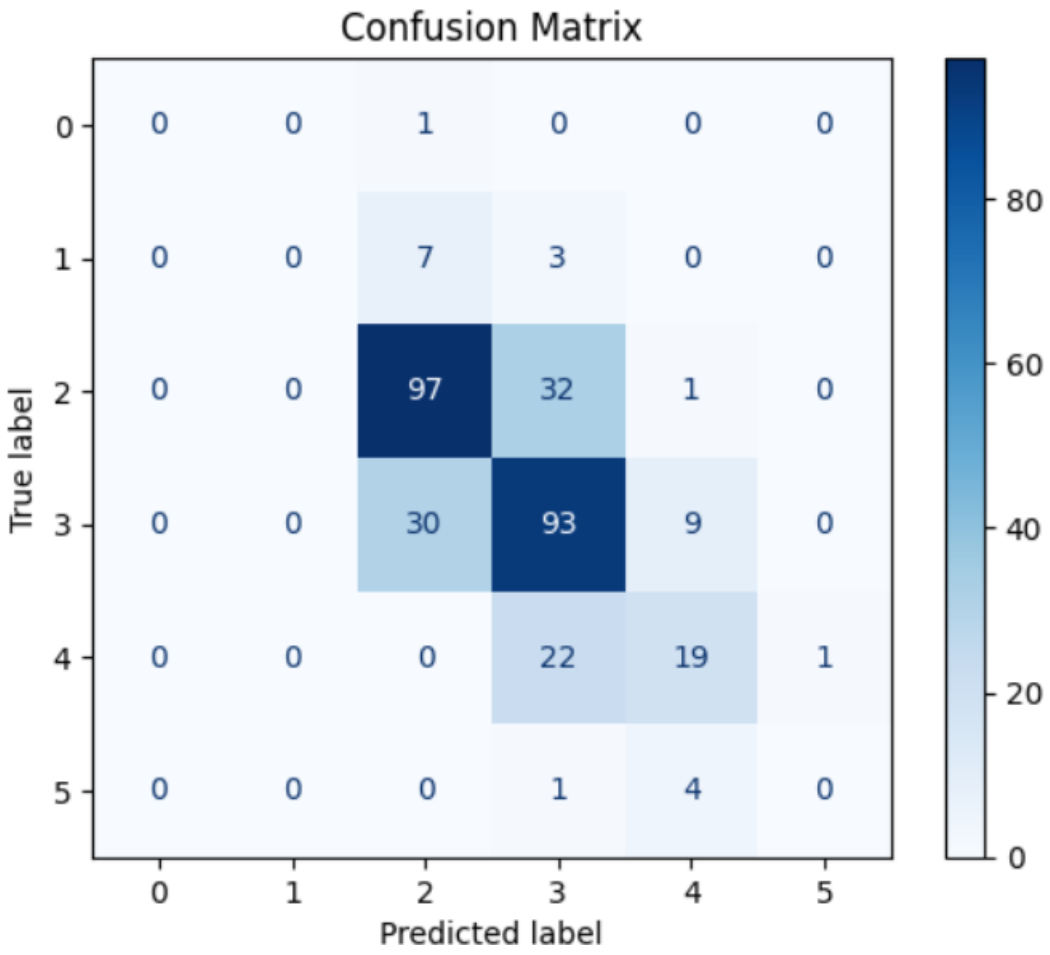
- Test Accuracy: 57.5%
- Recall: 0.65
- Precision: 0.65
- F1 Score: 0.64

Summary: Logistic Regression obtained an accuracy of 57.5% on the test set but doesn't aim to model non-linear relations between variables. It can be really useful if interpretative insights into feature influences on wine quality are required, however with lower predictability.

Random Forest

- Test Accuracy: 67.8%
- Recall: 0.65
- Precision: 0.65
- F1 Score: 0.64

Confusion Matrix:



Summary: RF outperformed Logistic Regression at 67.8% cross-validation accuracy before hyperparameter tuning. After hyperparameter tuning, with n_estimators=200 and max_depth=20, the model was as good as 69.2% cross-validation accuracy but test accuracy remained around 65.3%. Important predictors found are alcohol content, volatile acidity, and density from the feature importance plot. This model was chosen as the best as it is achieving a good balance between accuracy and interpretation.

Support Vector Machine (SVM)

- Test Accuracy: 60.3%

Summary: SVM achieved an accuracy of 60.3% on the test set and performed with a lot of precision but high computational power. It was good on high-dimensional data and well-balanced between classes, but did not perform better than Random Forest.

Final Model Selection: Random Forest

The best model that developed was the Random Forest, exhibiting higher accuracy (67.8%) and interpretability. The Logistic Regression revealed the linear dependencies but could not capture the non-linear ones. The SVM was very computation-intensive although accurate. It was realized that alcohol, volatile acidity, and density were among the most influential variables for wine quality in the Random Forest model-variables that can thus be a focus of further analysis and practicable application in winemaking.

Discussion

Feature Importance: Alcohol content, volatile acidity, and density were the most significant predictors of quality. High alcohol content and balanced acidity levels positively influenced quality ratings, while high density was generally associated with lower quality.

Model Interpretability: Random Forest allowed for feature importance visualization, making it useful for understanding which chemical factors influence wine quality most.

Trade-offs: Logistic Regression was more interpretable but less accurate, while SVM gave accuracy at a higher computational cost.

Conclusion

The best model was the **Random Forest** for wine quality prediction, having good performance and interpretability. It achieved around 67% accuracy. Random Forest has been shown to be much better than Logistic Regression and Support Vector Machine with respect to predictive accuracy and robustness. Feature importance analysis shows that alcohol content, volatile acidity, and density are the most important factors in the ratings of wine quality. This means that these attributes are the most important factors in determining consumer preferences and overall quality perception.

This project highlights the importance of machine learning in the wine industry, where quality prediction can be very challenging due to the subtle interplay of various chemical components. The model identifies the critical factors using the feature importance measure, thus giving actionable insight to winemakers. Changes in the critical attributes-acidity levels or alcohol concentration-will enhance the quality and fulfill demand for specific consumers. Eventually, the model can be utilized as a quality control tool that enables producers to predict the quality output based on chemical composition and support good decisions in production.

Future Work

Although the current model is very robust for wine quality prediction, several directions for further research and model improvement could potentially yield more accurate and useful results:

- **Ensemble Models**: The combination of multiple models through ensemble techniques such as stacking or blending could improve the overall accuracy and robustness. It may be that Random Forest, Gradient Boosting, and Logistic Regression combined through ensemble models have a strength in each one of the algorithms, allowing them to capture a much broader pattern in the data. It will help eliminate model bias and improve prediction for wines of mid-range quality, where misclassifications occurred.

- **Deep Learning Techniques:** Neural networks tend to learn complex, non-linear patterns, therefore deep learning models like MLPs or CNNs could be a possible direction to improve when applied on larger and more diverse wine datasets, capturing what the simpler model cannot at greater computational and data costs. This training on an expanded data set that spans wine type, region, and vintage will probably unfold some subtle hidden patterns related to the physicochemical properties of wines and quality.

- **Future work on this topic can include accumulating and including further chemical variables**, like aromatics compounds, polyphenols or mineral content. Such determinants are thought to interact with flavor profiling and will be perceived. This broadened feature can better assist in fine-tuning the model and add further sensitivity to wine quality determinants. The other direction by including sensory assessment scores made by expert panellists can introduce another dimension-the subjective quality and further direct the model in a similar direction as it will lead to consumer perceptions.

- **Model Interpretability and Explainability:** Further work may include SHAP (SHapley Additive exPlanations) values or LIME (Local Interpretable Model-agnostic Explanations) to add more interpretability. It provides insights at both feature and instance levels, depicting how each chemical component leads to specific predictions. Explainability would thus enable wine producers to better understand the model's output and its predictions, especially when winemakers adjust their productions according to model outputs.

- **Data Augmentation and Collection:** The more the elements of the set can be filled with more data coming from several vineyards, vintages, and geolocation areas, this will increase the generalizability of the model such that it will be less susceptible and more resilient to wine variety.

In addition, other methods such as synthetic data or data augmentation can be used to augment the set on class balance and resilience across varieties.

Final Thoughts

There is much hope for the application of machine learning in wine quality prediction. The improvement of predictability models and their potential scope will allow winemakers to achieve better uniformity of products, accommodate market trends, and learn new ways to improve quality. Further model complexity improvements and depth of data in the future could make those tools very powerful decision-aiding tools, and bridge the gap between classic wine production techniques and advanced data science.

Contributions

Arihant Jain

He began by explaining the core problem of the project: predicting wine quality based on chemical features. He did background research on the role of quality prediction in the wine industry, exploring methods like regression, ensemble techniques, and the specific importance of physicochemical features. His insights provided a foundation for understanding the significance of attributes like acidity and alcohol content, shaping the project's approach.

He had to prepare data; in the data, he loaded and cleaned with initial exploration to ensure no missing values existed and that he applied standardization and normalized features so organizing and scaling might be done on those to create a healthy ground for the models with which consistent performance and comparison might be made.

He led the exploratory data analysis by using visualizations to analyze the distribution of quality ratings and correlations between chemical properties. He plotted the distribution of wine quality ratings, assessed the relationship between key features such as alcohol, acidity, and density, and determined potential patterns. This exploratory work informed feature selection and helped focus on variables that significantly impact quality predictions.

He is responsible for the Logistic Regression and SVM models to implement. He fine-tuned the Logistic Regression by implementing regularization on it, whereas for SVM,

he was trying out different kinds of kernels in order to identify an optimal one. It sets up a baseline where he compared his work against other models to demonstrate the relative performance of these simple algorithms on this data as opposed to ensemble methods.

He also helped organize the final presentation in a way that translated complex technical findings into a simpler format. He drafted all the key sections of the report, including methodology, EDA, and the documentation of the Logistic Regression and SVM models. He ensured that the main conclusions and findings of the project were communicated clearly.

He maintained a GitHub repository and documented everything from pre-processing the data to checking the performance of the final model. This ensured his structure was accessible for anyone to refer to later when using his code. To do so, he put up a guide on the pipeline he had set to help other people understand as well as replicate the workflows of his project.

Aayush Agrawal

He brought a research-based approach to advanced techniques for quality prediction work, in which different machine learning strategies were explored, and their relative strengths and limitations weighed. He obtained findings contributing to a comprehensive model selection process that recognized ensemble methods such as Random Forest as relevant choices to be used to capture complex relationships in the data.

He created interaction features and applied PCA in feature engineering to reduce the dimensionality of the dataset. His work identified the potential feature interactions that might help in reducing prediction error, thereby maintaining a balance between complexity and computational efficiency.

For model implementation, his main focus was on implementing the Random Forest model to later choose as the winning model. He used Hyperparameter tuning with Gridsearchcv, where he took great care in fine tuning by varying n_estimators max_depth and other parameters accordingly in order to get utmost accuracy for the model implementation, which improved the final outcome and made it eventually his winner for predicting quality wines.

While interpreting the results, he interpreted feature importance results as important predictors to be critical, and it included alcohol content, volatile acidity, and density as

crucial. He studied the confusion matrix, finding misclassification trends, which should be known for certain specific areas so as to be improved in future enhancements of the model.

He paid more attention to the visualization of results clarifying model performance so that it can be easily interpreted. He has identified through a confusion matrix some areas where the model has poorly performed: mainly in mid-quality wines which often get confused with another. The analysis emphasized more refined feature engineering and more data for better improvement in these aspects.

In addition to his technical contributions, he was always in touch with Pratham on project timelines and progress. His organizational skills and feedback helped in a supporting collaborative environment which essentially ensured that the project came about successfully.

References/Bibliography

1. Datasets: UCI Machine Learning Repository. (n.d.). Wine Quality Dataset.

2. Citation: Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Wine Quality [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T.

3. Online Resources :

    - Scikit-learn Documentation. (n.d.). Machine Learning in Python. Retrieved from Scikit-learn Documentation.

    - Matplotlib Documentation. (n.d.). Matplotlib: Visualization with Python. Retrieved from Matplotlib Documentation.