Naive Bayes and Decision Tree

Group:
Arkaan Muhammad /491604
Jose Otto Ranier/496673

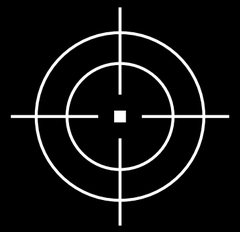
Contents

01/ Problem Statement
02/ Data Visualization
03/ Analysis
04/ Conclusions

01 - Problem Statement

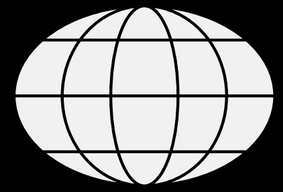
Problem Statement

The analysis demonstrated that both the Decision Tree and Naive Bayes models are effective in predicting the quality of red wine based on physicochemical features. The insights derived from these models can assist winemakers in **assessing and enhancing wine quality**, ultimately benefiting consumers seeking higher quality products.



OBJECTIVE

To **predict the quality** of red wine as either "Good" or "Not Good" and **evaluate** the performance of Decision Tree and Naive Bayes models in assisting winemakers to assess and enhance wine quality.



KEY QUESTIONS

1. How do Decision Tree and Naive Bayes models compare in terms of prediction accuracy?
2. What insights can winemakers gain from these models to improve wine production?
3. Can these models effectively classify wine quality for real-world application?


02 - DATA
VISUALIZATION

Data Visualization

Panda Profiling

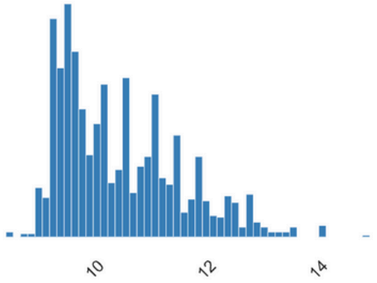
Dataset statistics

Number of variables	12
Number of observations	1599
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	220
Duplicate rows (%)	13.8%
Total size in memory	150.0 KiB
Average record size in memory	96.1 B

alcohol

Real number (ℝ)

Distinct	65	Minimum	8.4
Distinct (%)	4.1%	Maximum	14.9
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	10.422983	Memory size	12.6 KiB

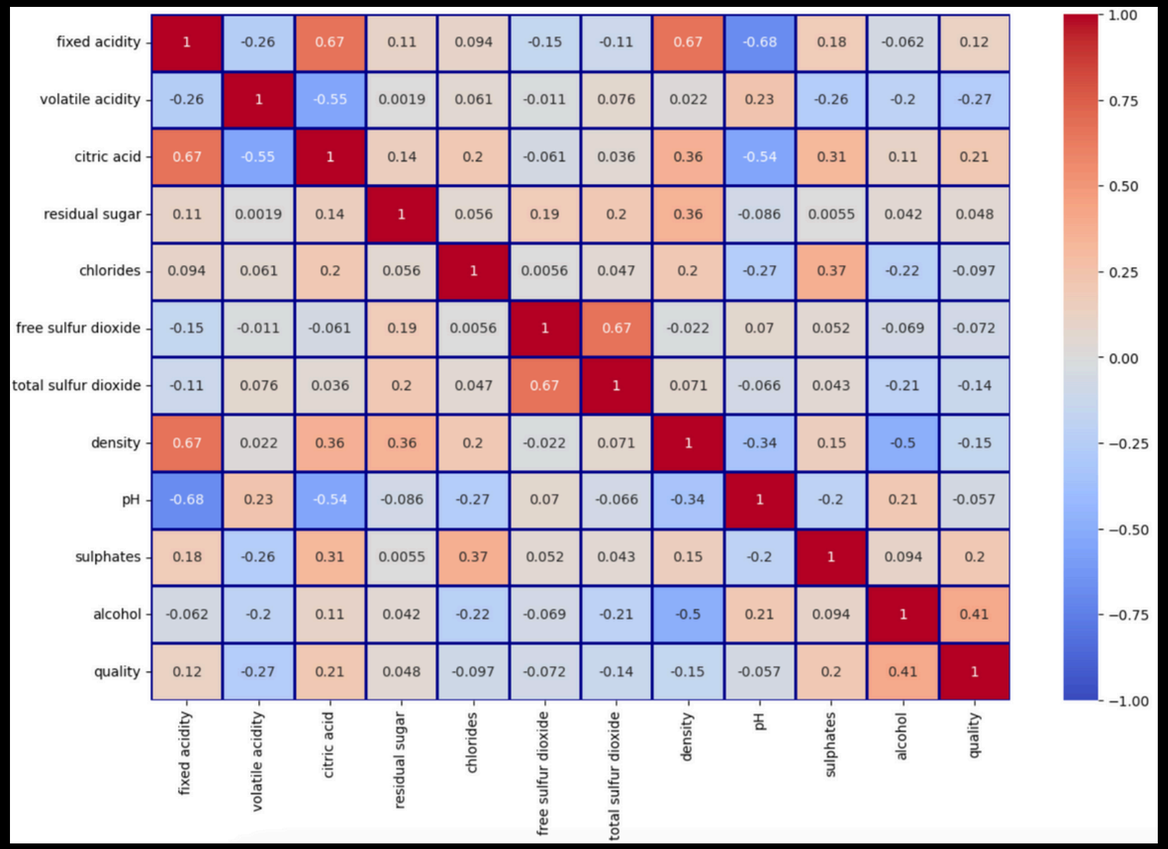


	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	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

03 - Analysis

03/ Analysis

Data Analysis



Correlation heatmap that represents the relationships between different variables

THIS HEATMAP HELPS IDENTIFY WHICH FEATURES ARE STRONGLY CORRELATED, INFORMING FEATURE SELECTION FOR PREDICTING WINE QUALITY.

1. **ALCOHOL AND QUALITY:** THERE IS A POSITIVE CORRELATION (0.41) BETWEEN ALCOHOL AND QUALITY, SUGGESTING THAT AS ALCOHOL LEVELS INCREASE, WINE QUALITY TENDS TO INCREASE AS WELL.

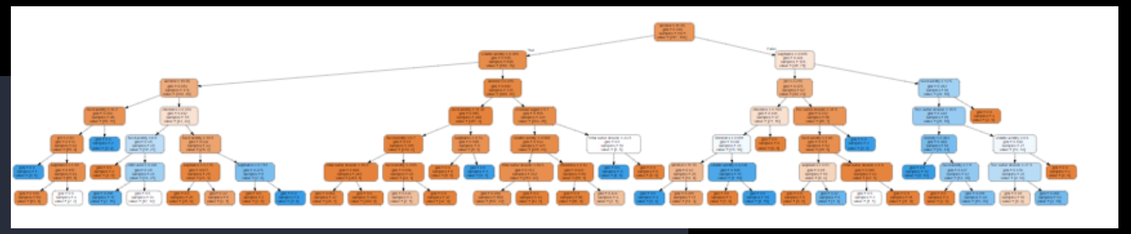
2. **CITRIC ACID AND QUALITY:** A MODERATE POSITIVE CORRELATION (0.21) BETWEEN CITRIC ACID AND QUALITY, INDICATING A RELATIONSHIP, THOUGH IT'S NOT AS STRONG AS ALCOHOL.

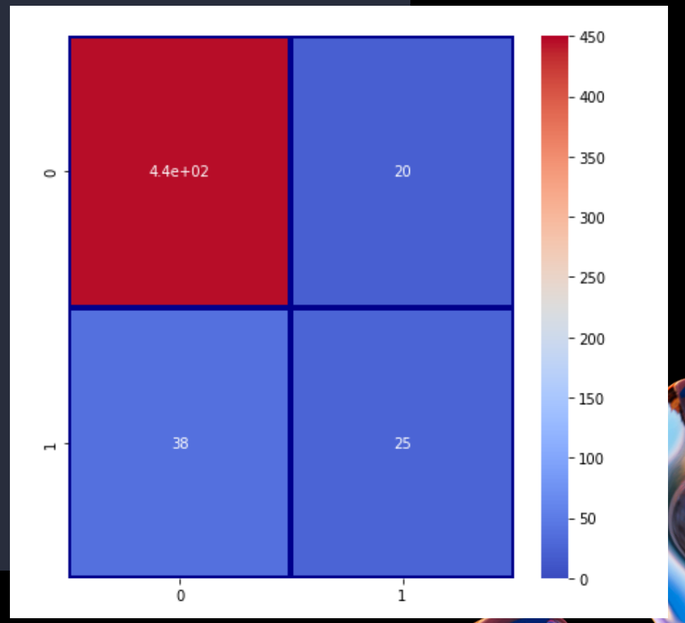
- **1 (DARK RED):** PERFECT POSITIVE CORRELATION – AS ONE VARIABLE INCREASES, THE OTHER ALSO INCREASES.
- **-1 (DARK BLUE):** PERFECT NEGATIVE CORRELATION – AS ONE VARIABLE INCREASES, THE OTHER DECREASES.
- **0 (WHITE OR LIGHT SHADES)**: NO CORRELATION – NO LINEAR RELATIONSHIP BETWEEN THE VARIABLES.

03/ Analysis

Decision Tree

```
criterion = ["gini", "entropy"]
ModelTree = DecisionTreeClassifier()
max_depth = np.array([2, 3, 4, 5, 6, 7, 8, 9])
min_samples_split = np.array([2, 3, 4, 5, 6, 7, 8, 9])
param_grid = {"criterion": criterion, "max_depth": max_depth, "min_samples_split": min_samples_split}
GridTree = GridSearchCV(estimator = ModelTree, param_grid = param_grid, cv = 5, n_jobs = 3)
GridTree.fit(x_train, y_train)
print(GridTree.best_estimator_.criterion)
print(GridTree.best_estimator_.max_depth)
print(GridTree.best_estimator_.min_samples_split)

ModelTree = DecisionTreeClassifier(max_depth = 6, min_samples_split=6)
ModelTree.fit(x_train, y_train)

Data_dot = export_graphviz(ModelTree, out_file = None, feature_names = list(x_train.columns), filled = True,
                rounded = True, special_characters = True, leaves_parallel = False, rotate = False)
graphviz.Source(Data_dot)

y_predTree = ModelTree.predict(x_test)

MatrixTree = confusion_matrix(y_test, y_predTree)

plt.figure(figsize = (7, 7))
sbn.heatmap(MatrixTree, annot = True, vmin = 0, vmax = 450, cmap = "coolwarm",  linewidth = 2.1, linecolor = "darkblue")
plt.show()

print(f"Accuracy: {accuracy_score(y_test, y_predTree)}")
print(f"Recall: {recall_score(y_test, y_predTree)}")
print(f"Precision: {precision_score(y_test, y_predTree)}")
print(f"f1_score: {f1_score(y_test, y_predTree)}")
```



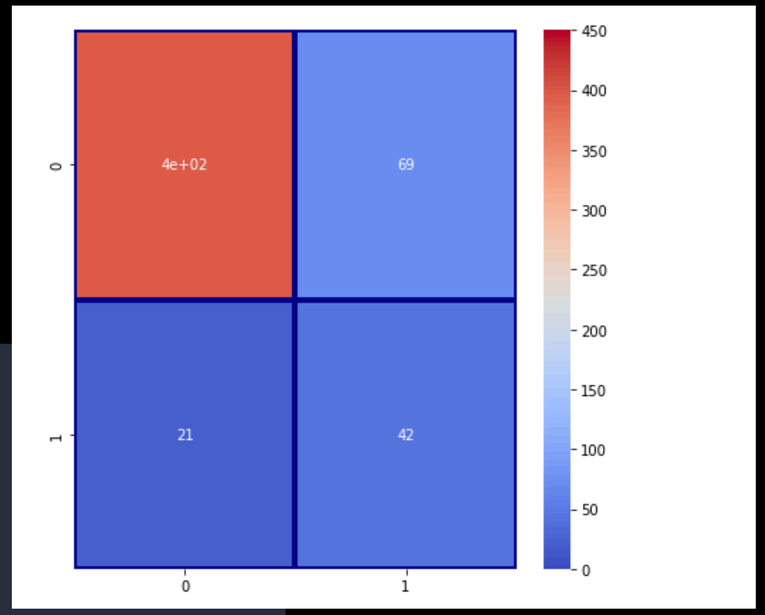


Accuracy: 0.8901515151515151
Recall: 0.3968253968253968
Precision: 0.5555555555555556
f1_score: 0.46296296296296297

03/ Analysis

Naive Bayes

```
ModelNB = GaussianNB()
ModelNB.fit(x_train, y_train)

y_predNB = ModelNB.predict(x_test)

MatrixNB = confusion_matrix(y_test, y_predNB)

plt.figure(figsize = (7, 7))
sbn.heatmap(MatrixNB, annot = True, vmin = 0, vmax = 450, cmap = "coolwarm",  linewidth = 2.1, linecolor = "darkblue")
plt.show()

print(f"Accuracy: {accuracy_score(y_test, y_predNB)}")
print(f"Recall: {recall_score(y_test, y_predNB)}")
print(f"Precision: {precision_score(y_test, y_predNB)}")
print(f"f1_score: {f1_score(y_test, y_predNB)}")
```



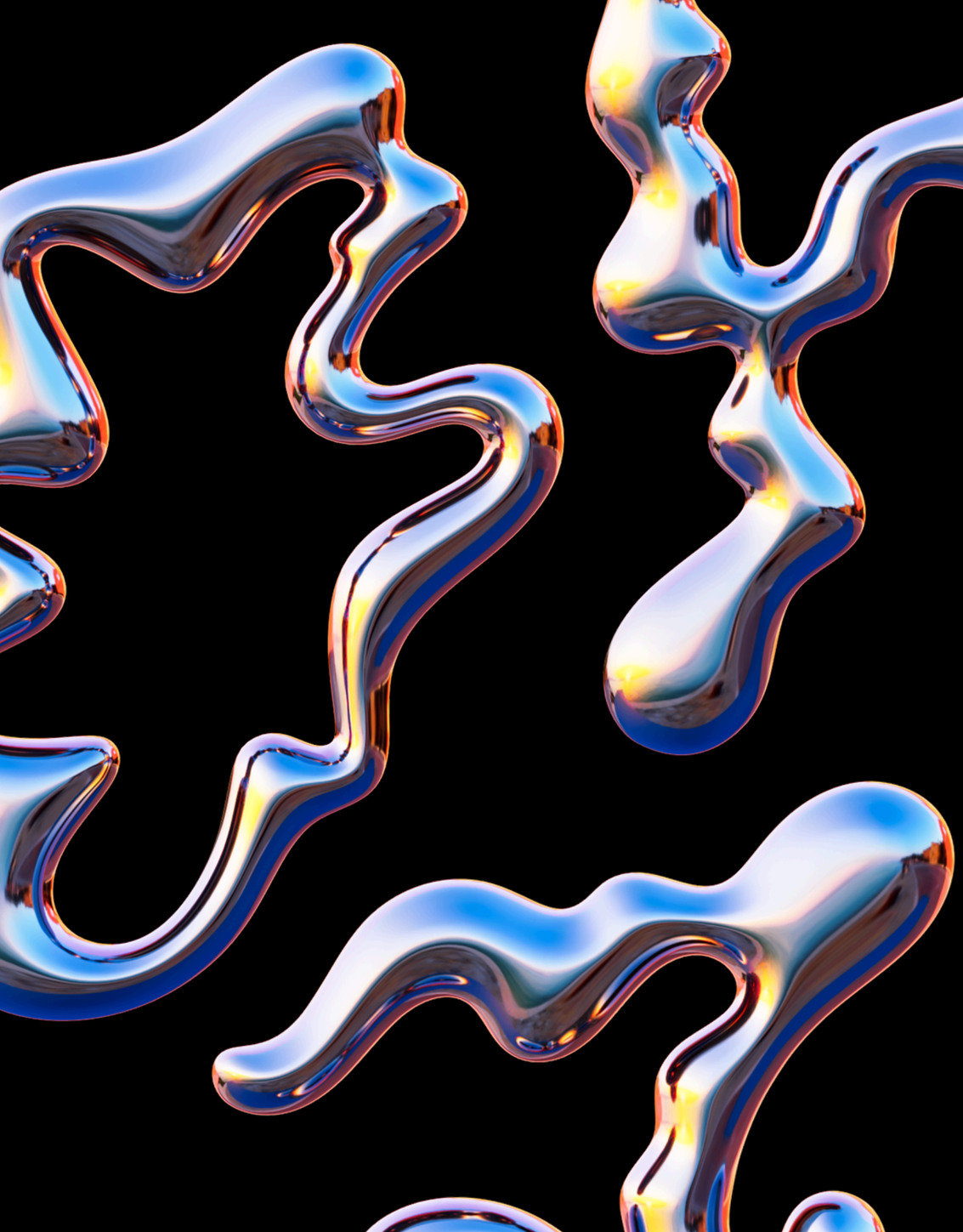
Accuracy: 0.8295454545454546
Recall: 0.6666666666666666
Precision: 0.3783783783783784
f1_score: 0.48275862068965514

03/ Analysis Performance Overview

Model	Accuracy	Recall	Precision	f1- Score
Decision Tree	0.8901515151515151	0.3968253968253968	0.5555555555555556	0.4629629629629629 7
Naive Bayes	0.8295454545454546	0.6666666666666666	0.3783783783783784	0.4827586206896551 4

- **ACCURACY**: THE DECISION TREE PERFORMS BETTER IN TERMS OF ACCURACY (~89% VS. 83%).
- **RECALL**: NAIVE BAYES PERFORMS BETTER WITH A RECALL OF ~67%, INDICATING IT CAPTURES MORE TRUE POSITIVES COMPARED TO THE DECISION TREE (~39%).
- **PRECISION**: THE DECISION TREE HAS BETTER PRECISION (~55%) COMPARED TO NAIVE BAYES (~38%).
- **F1-SCORE**: THE NAIVE BAYES MODEL HAS A SLIGHTLY BETTER F1-SCORE (~48%) COMPARED TO THE DECISION TREE (~46%), SUGGESTING A BETTER BALANCE BETWEEN PRECISION AND RECALL.

04 -Conclusions



04/ Conclusions

In conclusion, the accuracy of accuracy of the Decision Tree model **outperformed** Naive Bayes with an accuracy of 89.2%, but had lower recall (39.7%) compared to Naive Bayes (66.7%). This suggests that Naive Bayes is **better at identifying** "Good" wines but with **lower precision**. We can use these models to improve wine production by focusing on the most influential features, and both models show potential for real-world classification, though trade-offs between accuracy and recall .

Thanks

Link to code and dataset:

https://github.com/arkaan-m/Final-Project-Machine-Learning-Courses.git