Wine Quality Analysis

Introduction

Team Contributions

- Sanjay Reddy Javidi: Project Lead and Data Analysis
- Rakesh Reddy Kangari: Data Sampling and Shaping
- Guna Sai Koniki: Feature selection and Modeling
- Vaishnavi Koya: Statistical Analysis and Modeling
- Unmesha Kupekar: Database design

Project Overview

- Objective: Model wine quality from physicochemical properties
- Methods: Machine learning and statistical analysis

Problem Relevance

Importance of Wine Quality Analysis

- Impact on Consumer Satisfaction
- Influence on Winery Reputation and Sales

Our Approach

- Comprehensive Data Analysis
- Application of Machine Learning Models (regression & classification)

Problem Relevance

Problem We Addressed

- Predicting Wine Quality from Physicochemical Properties

Nature of the Dataset

- Extensive Data on Red and White Wines
- Multiple Physicochemical Parameters

Anticipated Challenges

- Complex Multidimensional Relationships
- Data Quality and Completeness - Biased data

Objectives and Goals

Intermediate Objectives

- Data Exploration and Understanding
- Developing Accurate Predictive Models
- Assessing Model Performance

Ultimate Goal

- Provide Reliable Wine Quality Predictions

Solution Design

Data Exploration and Analysis

- Examination of Physicochemical Features
- Descriptive and Inferential Statistics
- Feature importance analysis

Data Shaping and Sampling

- Standardization and Transformation
- Various Sampling Techniques (Random, Systematic, Stratified, Clustered)

Predictive Model Development

- Gaussian Mixture Models for Synthetic Data Generation
- Linear Regresion model & Classifier Models for Quality Prediction

Data exploration

- Total there are 6497 rows
  - 4898 red & 1599 white wine
- Total there are 13 columns
  - All are Physicochemical features of each wine
  - All are numerical except the colour (red or white)
  - Red and white are mapped to 0 & 1 to make it numerical
- No missing values
- Duplicated rows are 1177
- Quality column ranges from 3 to 9 (with bell shaped distribution)

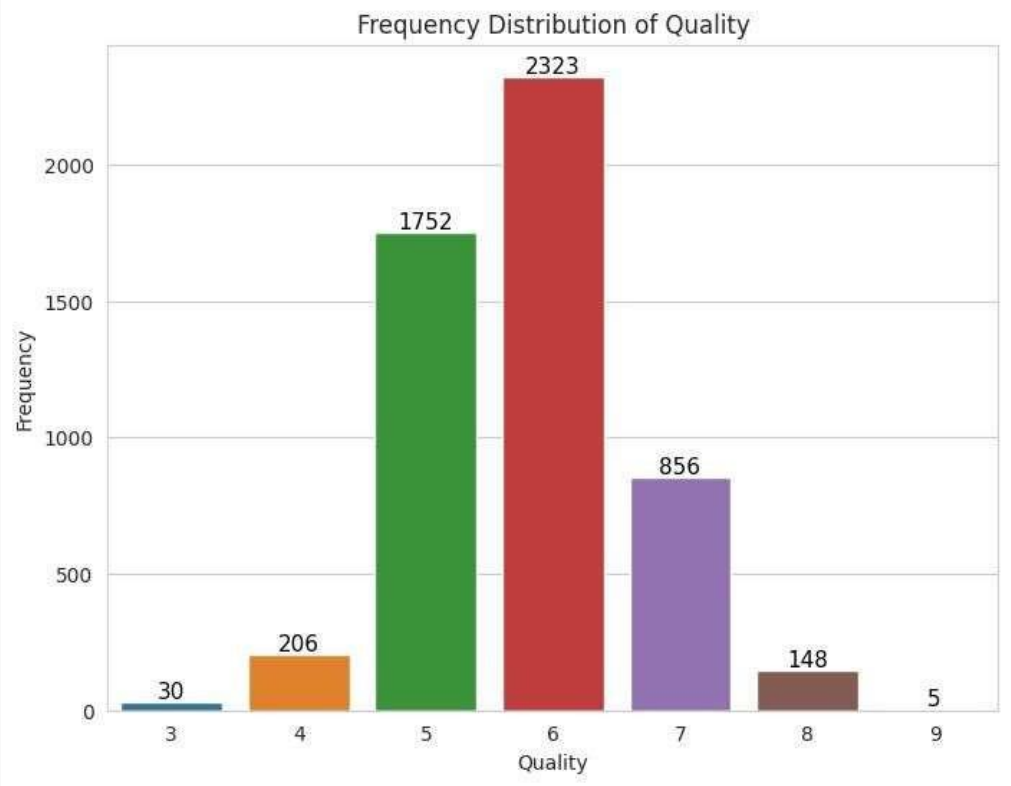
	FA	VA	CA	RS	Cl	FSD	TSD	Den	pH	Sul	Alc	Qual	Col
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	red
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	red
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	red
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	red
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	red

	FA	VA	CA	RS	Cl	FSD	TSD	Den	pH	Sul	Alc	Qual	Col
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000
mean	7.215307	0.339666	0.318633	5.443235	0.056034	30.525319	115.744574	0.994697	3.218501	0.531268	10.491801	5.818378	0.753886
std	1.296434	0.164636	0.145318	4.757804	0.035034	17.749400	56.521855	0.002999	0.160787	0.148806	1.192712	0.873255	0.430779
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.000000	3.000000	0.000000
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000	9.500000	5.000000	1.000000
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000	10.300000	6.000000	1.000000
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000	11.300000	6.000000	1.000000
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000	14.900000	9.000000	1.000000

Data snapshot and the summary statistics

	Data Type	Missing Values	Unique Values	Duplicate Rows
FA	float64	0	106	1177
VA	float64	0	187	1177
CA	float64	0	89	1177
RS	float64	0	316	1177
Cl	float64	0	214	1177
FSD	float64	0	135	1177
TSD	float64	0	276	1177
Den	float64	0	998	1177
pH	float64	0	108	1177
Sul	float64	0	111	1177
Alc	float64	0	111	1177
Qual	int64	0	7	1177
Col	int64	0	2	1177

Missing, unique and duplicate values


Frequency Distribution of Quality
Frequency
Quality
0
500
1000
1500
2000
3
4
5
6
7
8
9
30
206
1752
2323
856
148
5

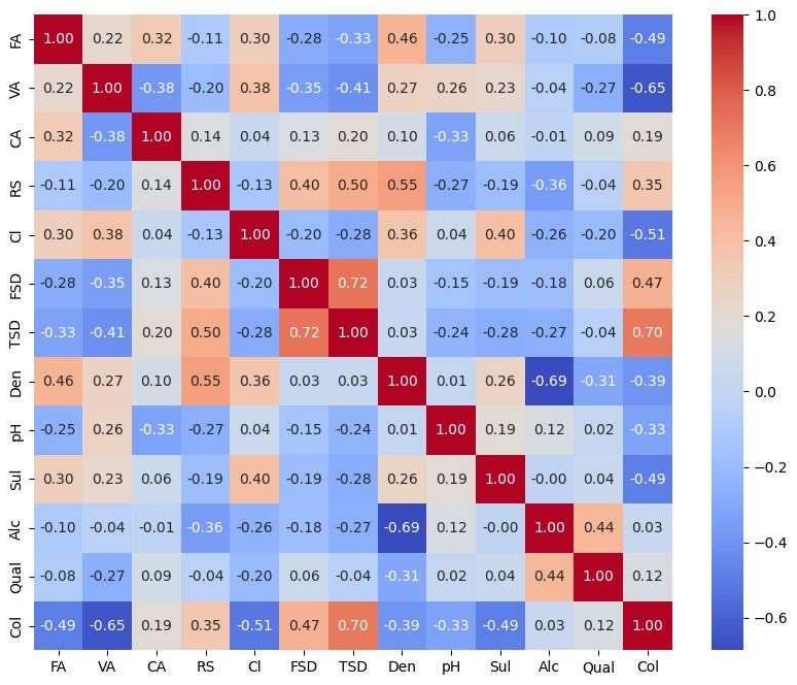
Synthetic Data generation (using GMM)

- Data Standardization: Scale wine_data to mean of 0, standard deviation of 1.
- Gaussian Mixture Model (GMM): Fit GMM with 10 components to the standardized data.
- Synthetic Data Generation: Create 2000 synthetic data samples using GMM.
- Inverse Transformation: Convert synthetic data back to original data scale.
- Data Cleaning: Round and convert ‘Qual’ column to integer; adjust within original range.

Linear Modeling

- Normalize the Data to scale features (using standard scaler).
- Feature Selection based on the correlation and mutual information (MI) scores
  - Selected - 'Alc', 'Den', 'Cl', 'CA', 'VA', 'TSD', 'FSD'
- Use a linear regression model from sklearn and train it with the selected features.
- Evaluate the model's performance using appropriate metrics (MAE, R2)
  - R-squared: 0.27
  - Mean Absolute Error (MAE): 0.57
- Visualizations to communicate the validity and performance of the model.

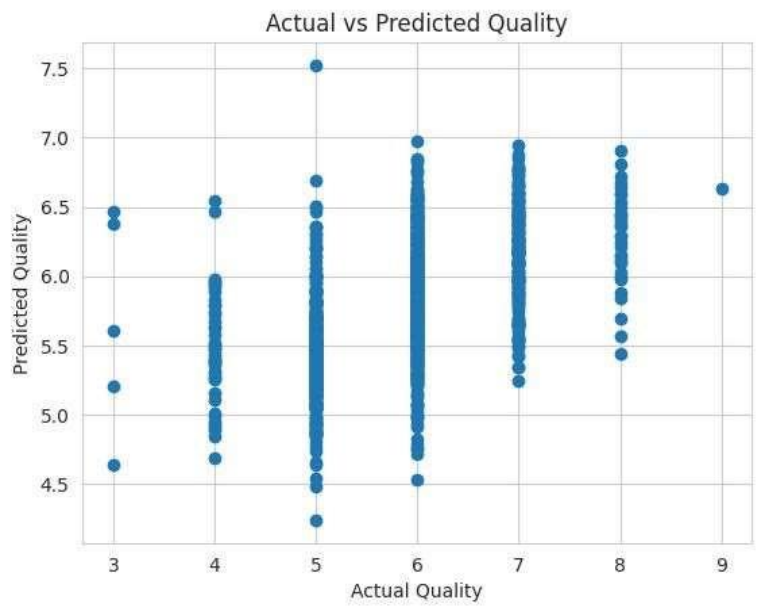
Feature Selection using correlation matrix & mutual information

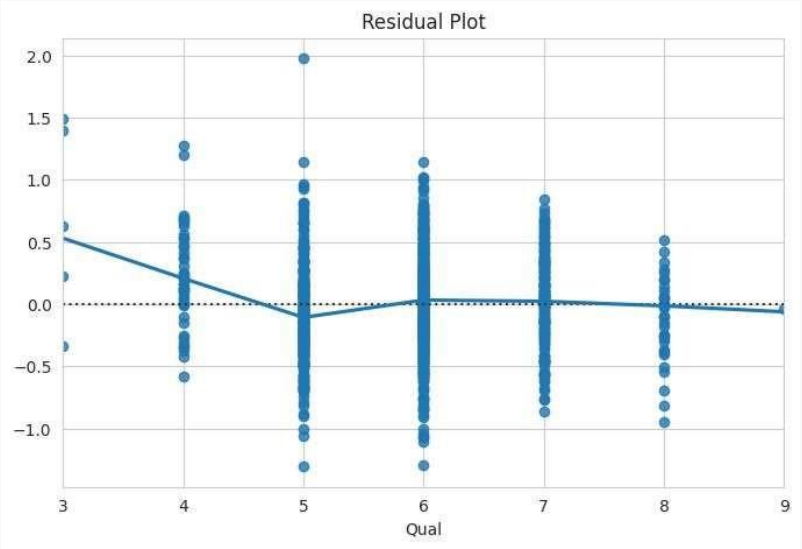



```python
# Separate the features and the target variable 'quality'
X = df.drop('Qual', axis=1)
y = df['Qual']

# Calculate mutual information scores
mi_scores = mutual_info_regression(X, y)

# Convert to a Series for easy handling and naming the index
mi_scores = pd.Series(mi_scores, name="MI Scores", index=X.columns)

# Sort the features by mutual information score
mi_scores = mi_scores.sort_values(ascending=False)

# Display the top 5 features
print(mi_scores.head(11))
```

```
Alc     0.156776
Den     0.085962
Cl      0.053036
TSD     0.045359
VA      0.042290
CA      0.039258
FSD     0.029174
RS      0.026253
Sul     0.009659
FA      0.000000
pH      0.000000
```

Visual Validation of Linear Model





Classification Modeling

- Binarize wine quality into 'high' (1) and 'low' (0) categories.
- Used median value (6) as a metric and classified to high or low category
- Standardized features for classification.

Model and evaluation metrics used

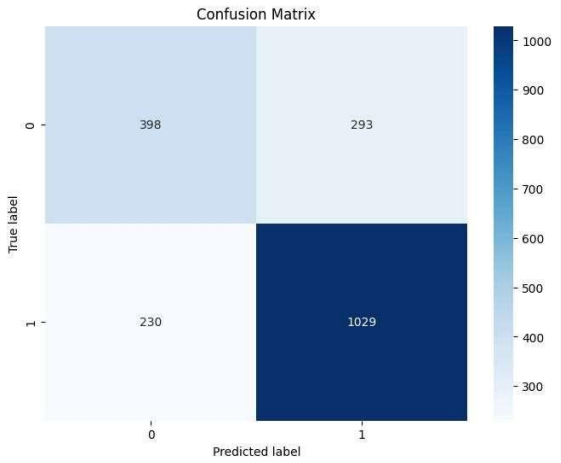
- Logistic Regression (Probabilistic)
- k-Nearest Neighbors (Euclidean)
- k-Nearest Neighbors (Cosine Similarity)
- Confusion matrix and Classification report

Creation of classifiers

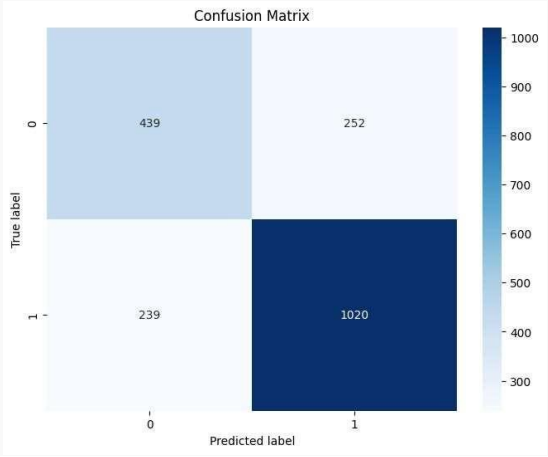
- For first two classifiers we used sklearn library
  - LogisticRegression for probabilistic classifier
  - KNeighborsClassifier for distance based classifier
- For Cosine Similarity-Based classifier
  - CosineKNNClassifier class is initialized with a specified number of neighbors (k) = 5
  - This class uses the NearestNeighbors implementation from sklearn.neighbors, which is configured to use the cosine metric for calculating similarity.
  - The predict method calculates the k-nearest neighbors based on the cosine similarity metric
  - It uses majority voting among the target values of its nearest neighbors to make a prediction.

Metrics of probabilistic, distance, similarity-based classifiers

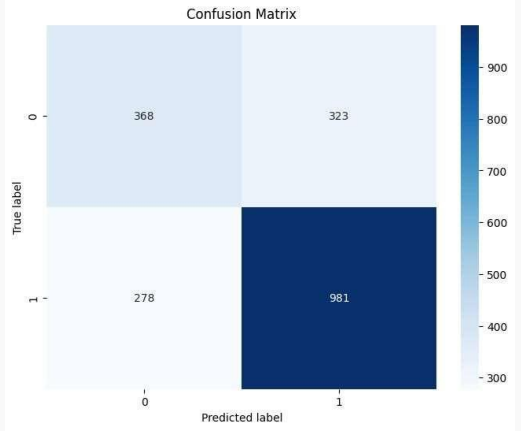
- **Accuracy**: 73%
- **Precision (0/1)**: 0.63/0.78
- **Recall (0/1)**: 0.58/0.82
- **F1-Score (0/1)**: 0.60/0.80



- **Accuracy**: 75%
- **Precision (0/1)**: 0.65/0.80
- **Recall (0/1)**: 0.64/0.81
- **F1-Score (0/1)**: 0.64/0.81



- **Accuracy**: 69%
- **Precision (0/1)**: 0.57/0.75
- **Recall (0/1)**: 0.53/0.78
- **F1-Score (0/1)**: 0.55/0.77



Conclusion

Sdf