Richard Lung

AI Wine Quality Evaluator

Quality wines, quality life.

Problem

Remember your first wine purchase? You took a sip, grimaced, and thought, “What’s all the fuss about?” After trying another bottle and pouring it down the drain, you might have concluded, “Maybe I’m just not a wine person.”

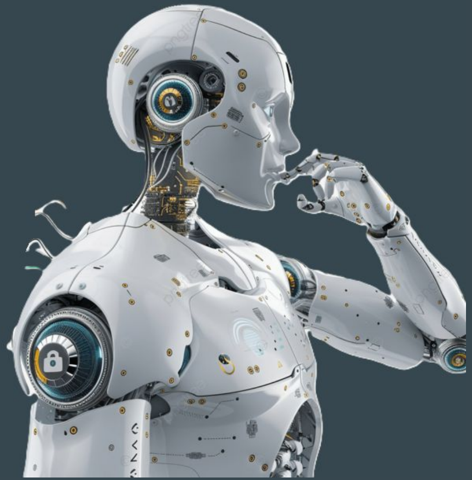
Well, hold on! You could be missing out because you just picked the wrong bottle—or maybe the store was stocking some real duds!

I’m here to help stores choose better wines, ensuring you have a delightful wine experience instead of a headache.

Cheers to discovering the wine you’ll actually love!

Solution

WAIne™ is an AI prediction model trained on 1,000 different chemical test results, specifically designed to forecast a wine's taste and overall experience score based on its chemical composition. By analyzing various compounds found in wine, WAIne can provide a score with 95% accuracy.
This high level of precision is due to the strong correlation between the chemical compound levels and the resulting taste experience. With WAIne, you can trust that the scores are not just numbers—they're a reliable guide to your next great wine discovery!



Database

Wine main chemical composition database, containing the following columns:

Fixed Acidity (e.g: Citric Acid) - Fermented acid residue. Impacts taste, stability and preservation.

Volatile Acidity - Mainly Acetic acid. Gives a vinegar aroma if in high concentrates.

pH - Total level of acidity.

Residual Sugar - Remaining sugars not converted to alcohol. Determines how dry or sweet the wine is, including overall taste and mouthfeel.

Chlorides - Sodium Chloride is basically salt. Small traces of chlorides are found in wine and enhance the flavour.

Total Sulfur Dioxide (e.g: Free Sulfur Dioxide) - Prevents oxidation and bacteria formation. Essentially preserves the wine.

Sulphates - High levels of sulphate ions help balance acidity levels. Characterises the crisp and clean signature taste of wine.

Target Audience

Wineries can utilize WAIne to evaluate their annual wine production, ensuring consistency and quality among their produce.

Enthusiasts, whether seasoned wine lovers or those just starting their journey, can rely on WAIne instead of 'experts' for recommendations. With WAIne, the possibilities for exploring and expanding your wine hobby are limitless!

Stores & Supermarkets can confidently stock high-quality wines, guaranteeing that customers receive only the best, regardless of their choice.

Exploratory Data Analysis - Inspection

Raw Data

```
Number of rows: 1143
Number of columns: 13
```

Number of Missing|NaN|Dups

```
Number of missing values:
 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
Id                      0
dtype: int64
Number of duplicate rows:
(0, 13)
```

Statistical Aggregation

	fixed acidity	volatile acidity	citric acid	residual sugar
mean	8.311111	0.531339	0.268364	2.532152
median	7.900000	0.520000	0.250000	2.200000
std	1.747595	0.179633	0.196686	1.355917
skew	1.044930	0.681547	0.371561	4.361096
kurt	1.384614	1.375531	-0.714686	27.675366

	chlorides	free sulfur dioxide	total sulfur dioxide	density
mean	0.086933	15.615486	45.914698	0.996730
median	0.079000	13.000000	37.000000	0.996680
std	0.047267	10.250486	32.782130	0.001925
skew	6.026360	1.231261	1.665766	0.102395
kurt	47.078324	1.932170	5.098748	0.888123

	pH	sulphates	alcohol	quality
mean	3.311015	0.657708	10.442111	5.657043
median	3.310000	0.620000	10.200000	6.000000
std	0.156664	0.170399	1.082196	0.805824
skew	0.221138	2.497266	0.863313	0.286792
kurt	0.925791	12.017377	0.221179	0.314664

Exploratory Data Analysis - Modification

Dropping 'Id' column

```
# Dropping 'Id' column, due to irrelevance.
df = df.drop( labels: ['Id'], axis=1)
print(df.head(5))
# No need for renaming, all names are valid and important to understand the wine compounds.
```

Outlier values detection and drop

```
Before any changes:
Number of rows: 1143, Number of columns: 13

After any changes:
Number of rows: 1041, Number of columns: 12
```

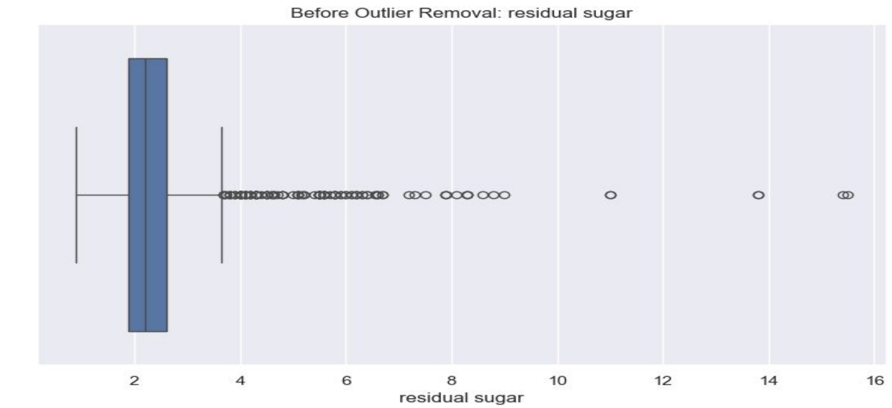
Because there are no NaN|Missing|NULL values, no further modification needed.

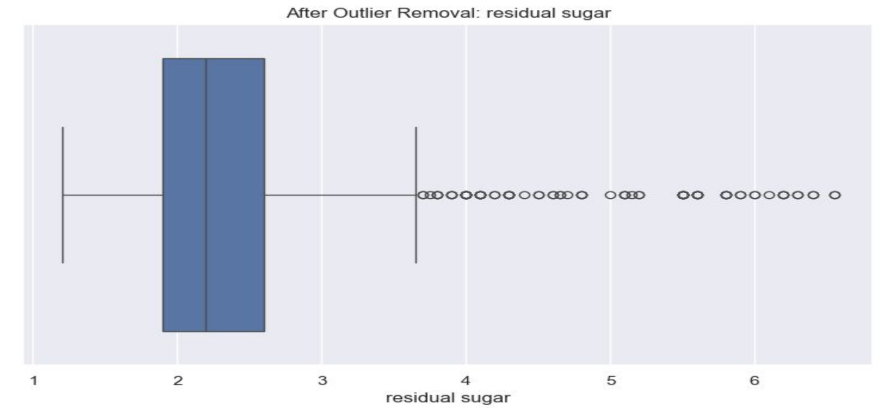
Exploratory Data Analysis - Outlier Box Plots

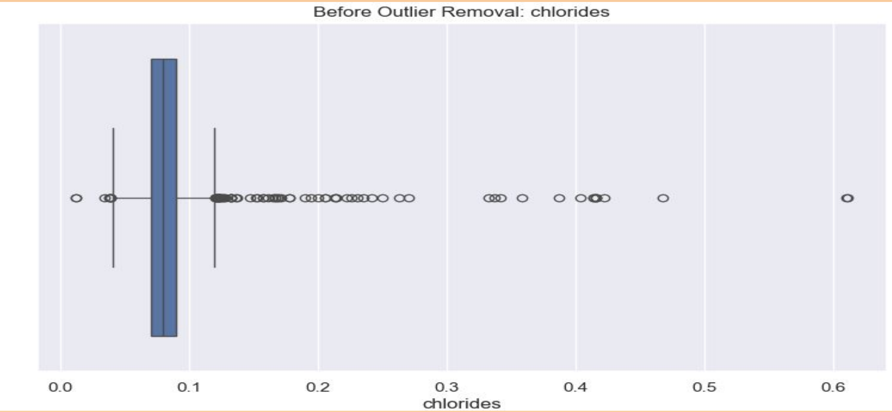
Richard Lung

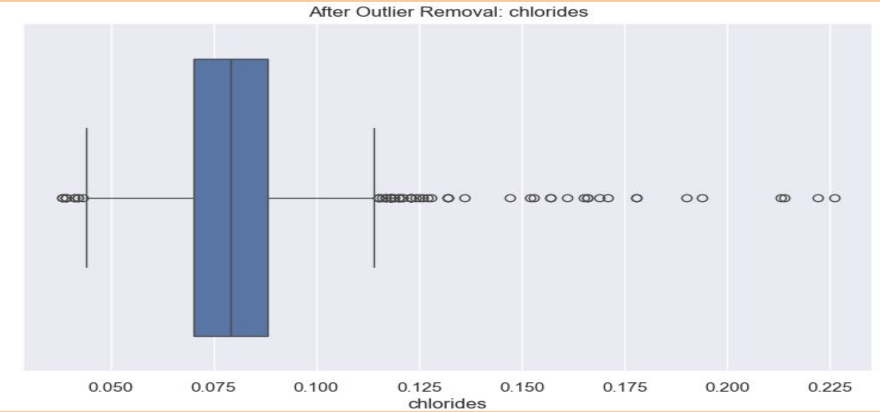
Before

After







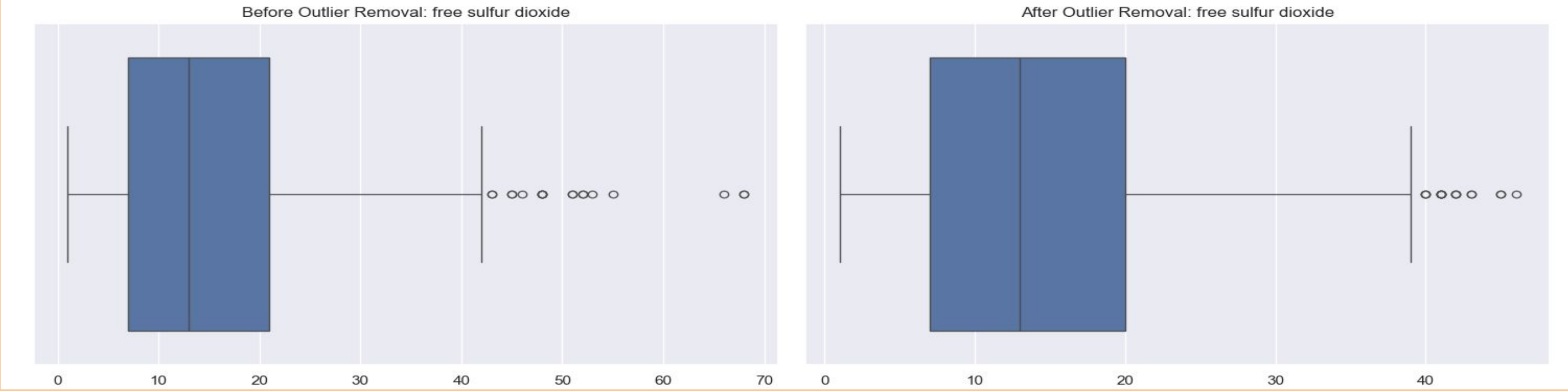


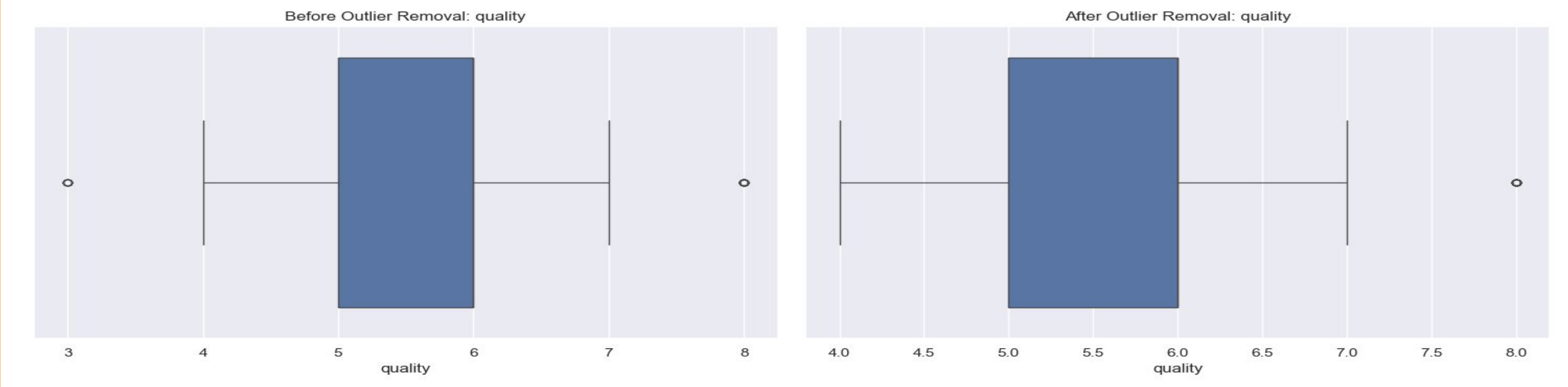
Exploratory Data Analysis - Outlier Box Plots

Richard Lung

Before

After





Richard Lung

Exploratory Data Analysis - Outlier Aggregation

Before

	fixed acidity	volatile acidity	citric acid	residual sugar
mean	8.311111	0.531339	0.268364	2.532152
median	7.900000	0.520000	0.250000	2.200000
std	1.747595	0.179633	0.196686	1.355917
skew	1.044930	0.681547	0.371561	4.361096
kurt	1.384614	1.375531	-0.714686	27.675366

	chlorides	free sulfur dioxide	total sulfur dioxide	density
mean	0.086933	15.615486	45.914698	0.996730
median	0.079000	13.000000	37.000000	0.996680
std	0.047267	10.250486	32.782130	0.001925
skew	6.026360	1.231261	1.665766	0.102395
kurt	47.078324	1.932170	5.098748	0.888123

	pH	sulphates	alcohol	quality
mean	3.311015	0.657708	10.442111	5.657043
median	3.310000	0.620000	10.200000	6.000000
std	0.156664	0.170399	1.082196	0.805824
skew	0.221138	2.497266	0.863313	0.286792
kurt	0.925791	12.017377	0.221179	0.314664

After

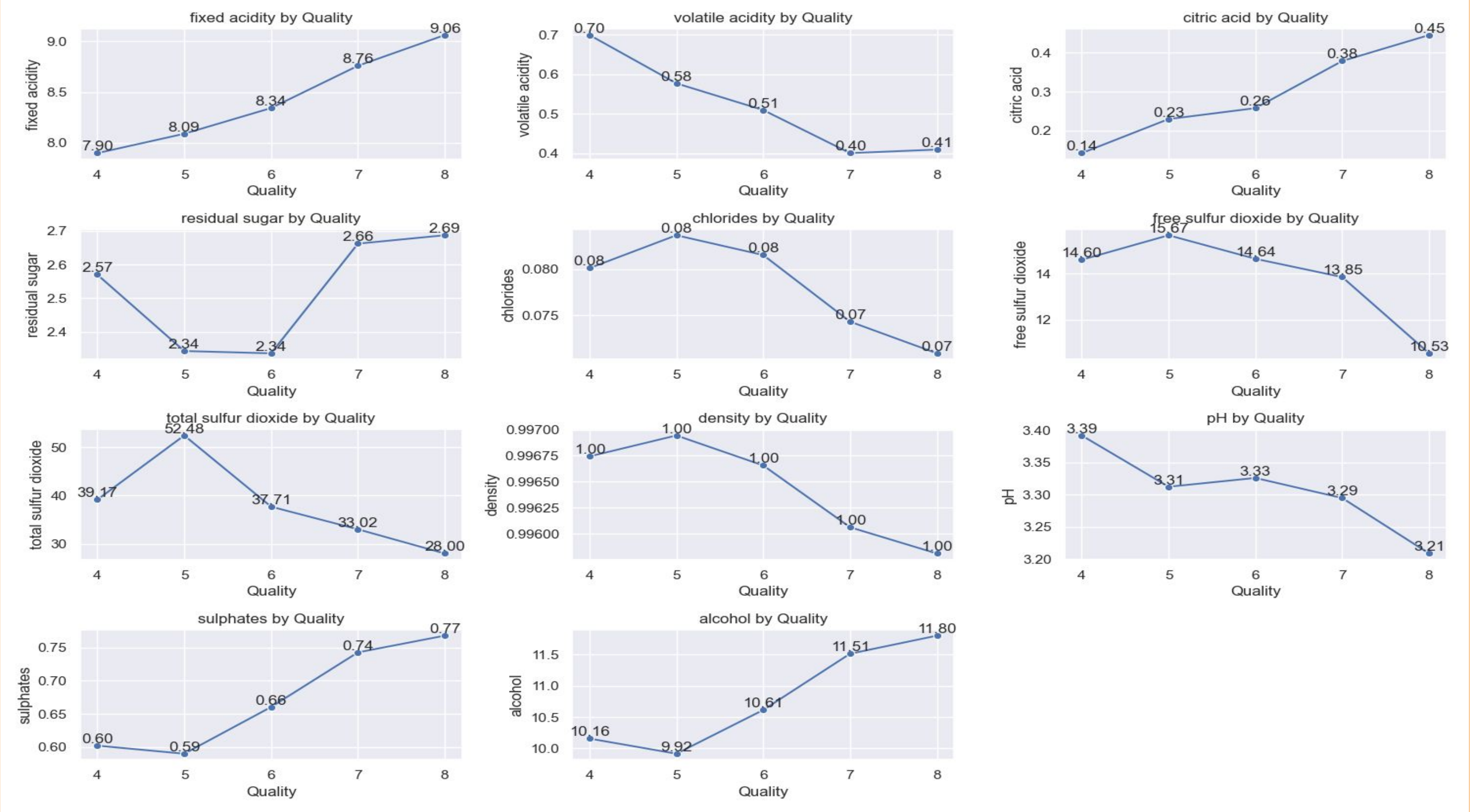
	fixed acidity	volatile acidity	citric acid	residual sugar
mean	8.285879	0.528132	0.260298	2.392651
median	7.900000	0.520000	0.240000	2.200000
std	1.649721	0.170495	0.191656	0.870089
skew	0.878420	0.395274	0.359343	2.404664
kurt	0.428010	-0.027459	-0.833551	6.747981

	chlorides	free sulfur dioxide	total sulfur dioxide	density
mean	0.081318	14.914505	43.267051	0.996693
median	0.079000	13.000000	35.000000	0.996600
std	0.020785	9.232228	29.039955	0.001735
skew	2.242040	0.897179	1.187371	0.107999
kurt	10.477571	0.173154	0.989091	0.209664

	pH	sulphates	alcohol	quality
mean	3.316513	0.640768	10.435142	5.674352
median	3.320000	0.620000	10.200000	6.000000
std	0.143251	0.127751	1.028969	0.784835
skew	0.125458	0.926642	0.751747	0.463899
kurt	0.106212	0.941466	-0.225064	-0.013900

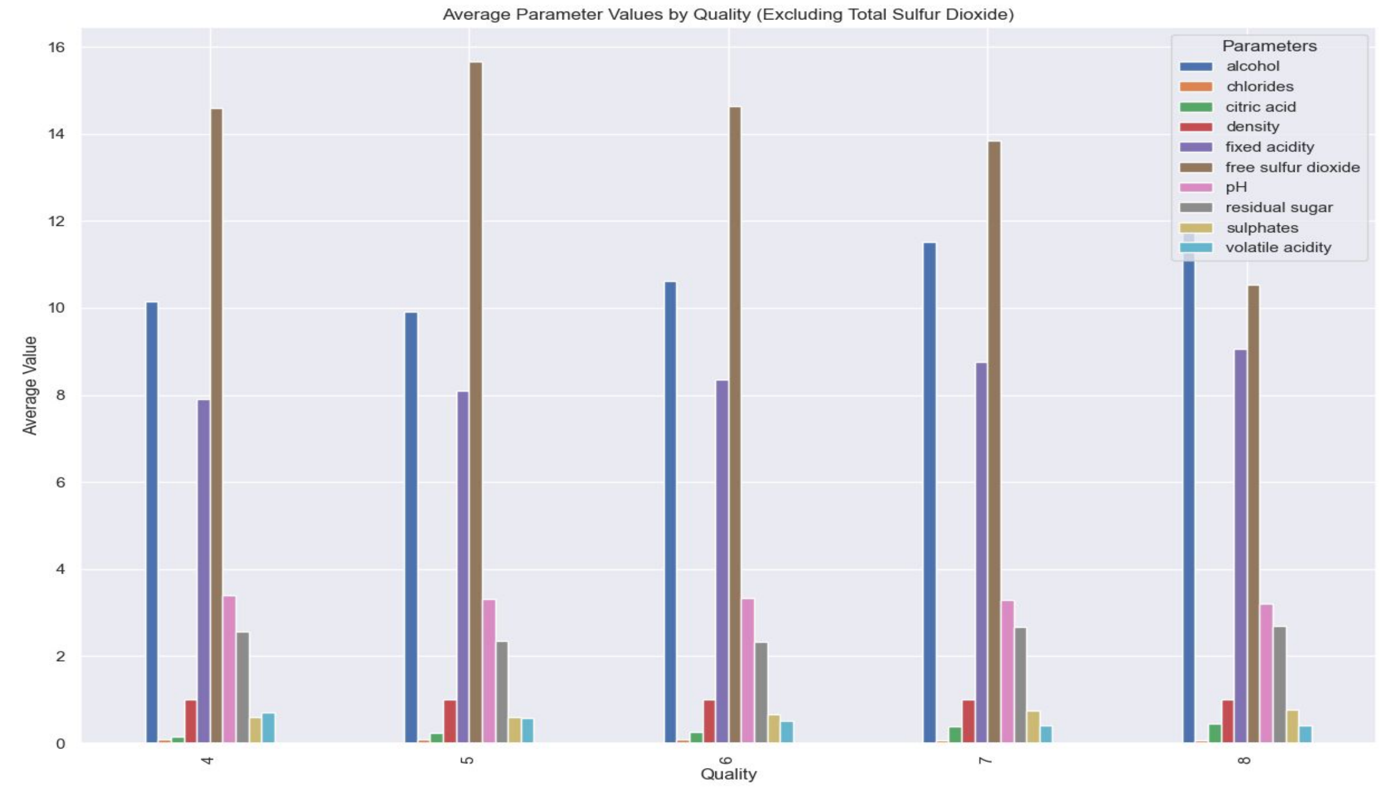
Exploratory Data Analysis - Visualization

Richard Lung

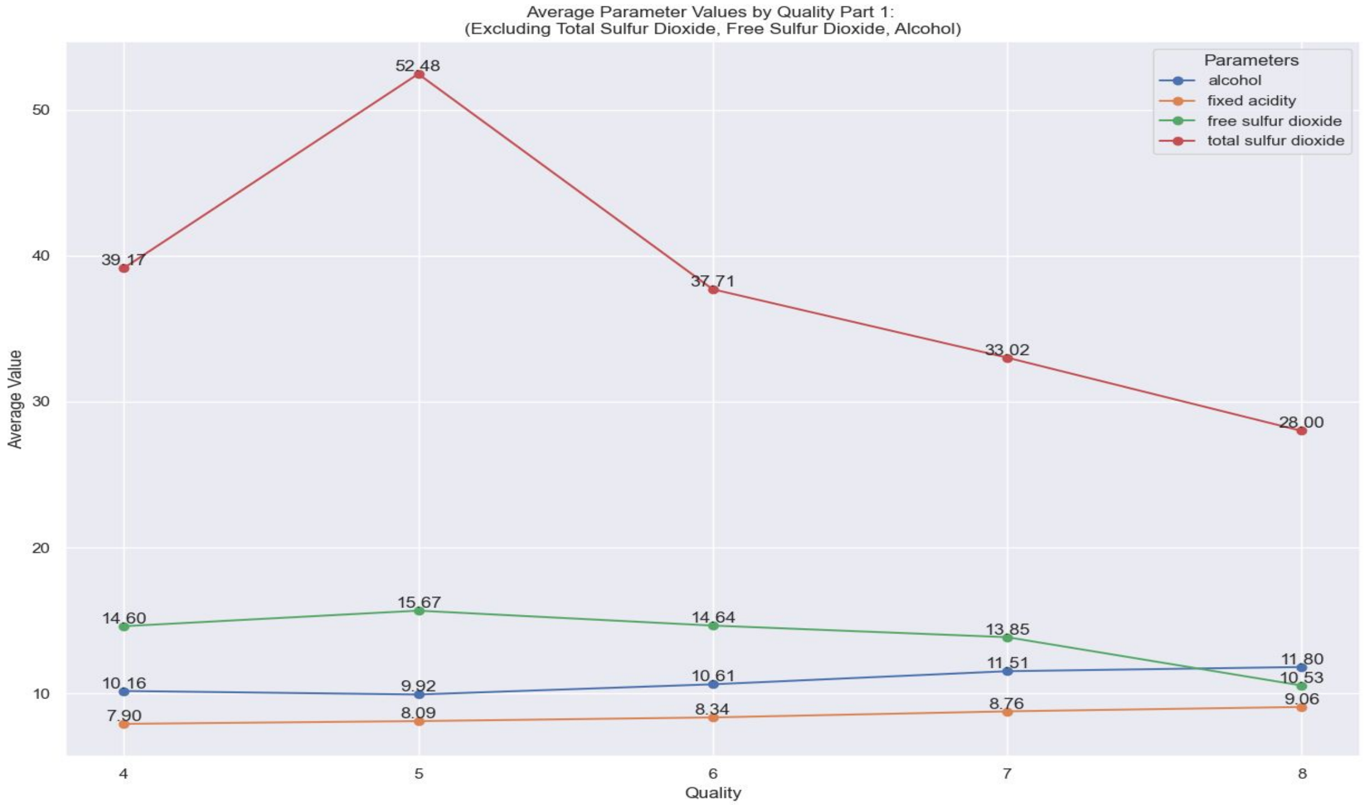


Exploratory Data Analysis - Visualization

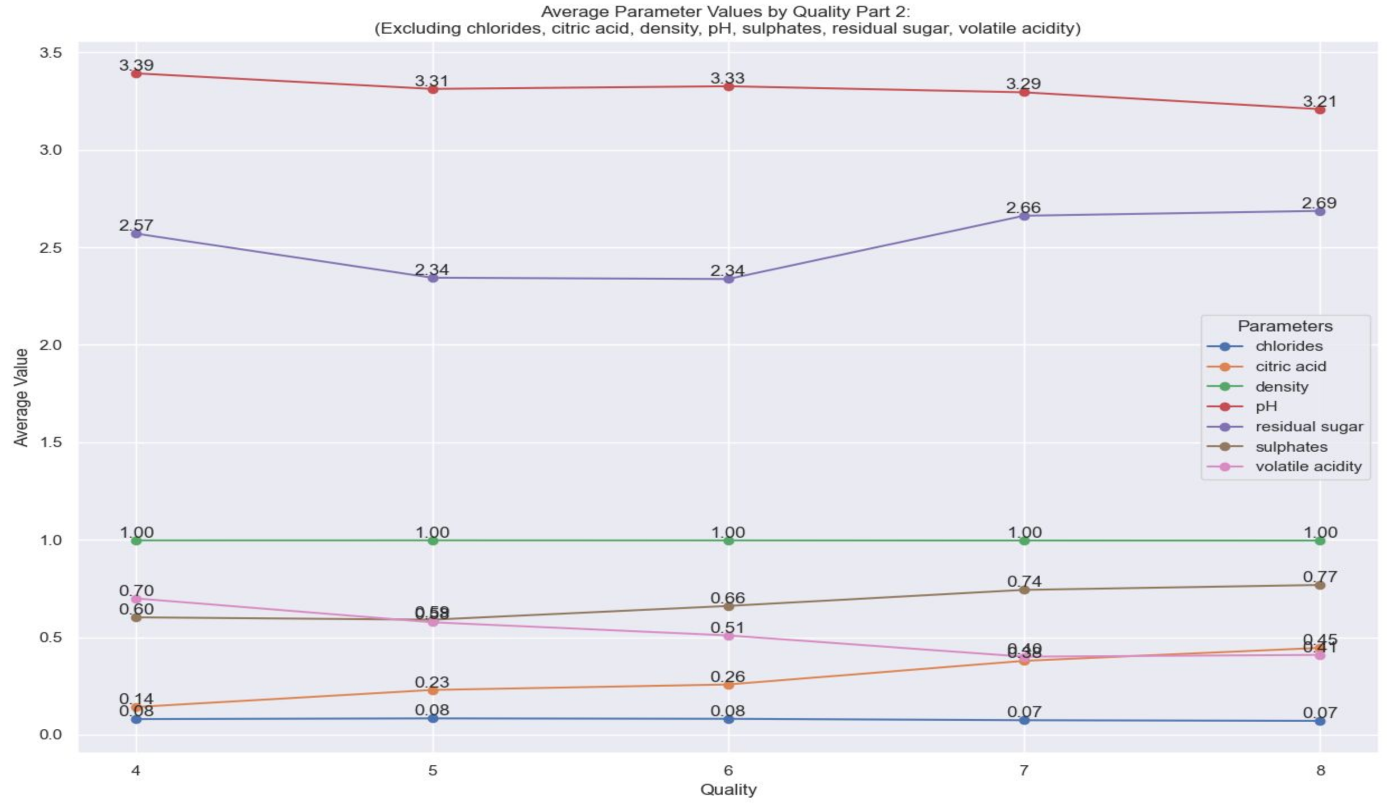
Richard Lung



Exploratory Data Analysis - Visualization



Exploratory Data Analysis - Visualization



Richard Lung

Exploratory Data Analysis - Numerical Analysis

Param \ Values	Origin Mean	Mod Mean	Desired Mean	DM-MM	Trend
Fixed Acidity	8.31	8.29	9.06	0.75	↗
Volatile Acidity	0.53	0.53	0.41	-0.12	↘
Citric Acid	0.27	0.26	0.45	0.19	↗
Residual Sugar	2.53	2.39	2.69	0.3	↗
Chlorides	0.09	0.08	0.07	-0.01	↘
Free Sulfur	15.62	14.91	10.53	-4.38	↘

Param \ Values	Origin Mean	Mod Mean	Desired Mean	DM-MM	Trend
Total Sulfur	45.91	43.27	28.00	-15.27	↘
Density	0.9967	0.9966	0.9957	-0.0009	↘
pH	3.31	3.32	3.21	-0.01	↘
Sulphates	0.66	0.64	0.77	0.13	↗
Alcohol	10.44	10.44	11.80	1.36	↗
Quality	5.66	5.67	8.00	3.33	↗

Origin Mean → The original mean of all params
Mod Mean → Removal of outliers and further mods ('clean data')
Desired Mean → Wine quality = 8.0, all params must reach values
DM-MM → Desired Mean - Mod Mean
**All values are in grams/Liter (Alcohol → ABV(%))

WAIne™ Implementation

Training and Building

```
# Use StandardScaler to scale the input features to have a mean of 0 and std of 1
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)  # Fit the scaler and transform the features

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( *arrays: X_scaled, y, test_size=0.15)

# Build the Neural Network Model
model = tf.keras.models.Sequential()

# Input layer and hidden layers
model.add(tf.keras.layers.Dense(256, input_dim=X_train.shape[1], activation='relu'))  # First hidden layer
model.add(tf.keras.layers.Dense(128, activation='relu'))  # Second hidden layer
model.add(tf.keras.layers.Dense(64, activation='relu'))  # Third hidden layer

# Output layer
model.add(tf.keras.layers.Dense(1, activation='linear'))  # Linear activation for regression output
```

Scaling for better training

85% train, 15% test

TensorFlow's Keras model for neural network

```
# Compile and train model
model.compile(optimizer='adam', loss='mean_absolute_error', metrics=['mape'])  # Use MSE for loss and MAPE for evaluation
model.fit(X_train, y_train, epochs=300, batch_size=32, validation_data=(X_test, y_test), verbose=1)

# Step 9: Evaluate the Model on the Test Data
loss, mape = model.evaluate(X_test, y_test)

# Print the final results
print(f"Test MAE (Mean Absolute Error): {loss}")  # MSE is calculated as part of the evaluation
print(f"Test MAPE (Mean Absolute Percentage Error): {mape}")
```

Mean Absolute Error &
Mean Absolute Percentage Error
For loss and metrics

Results

```
Epoch 300/300
28/28 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.1077 - mape: 1.9443 - val_loss: 0.4672 - val_mape: 8.4277
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 571us/step - loss: 0.4493 - mape: 7.9398
Test MAE (Mean Absolute Error): 0.46719735860824585
Test MAPE (Mean Absolute Percentage Error): 8.427702903747559
```

MAE → 0.4 - 0.6

MAPE → 8% - 11%

```
#======= Showing some examples of predictions =======#
#       [fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur, total sulfur, density, pH, sulphates, alcohol]
wine1 = [9.0          , 0.660           , 0.17       , 3.0           , 0.077    , 5.0       , 13.0        , 0.99760, 3.29, 0.55    , 10.40] #quality = 5
wine2 = [11.6         , 0.230           , 0.57       , 1.80          , 0.074    , 3.0       , 8.0         , 0.99810, 3.14, 0.70    , 9.90]  #quality = 6
wine3 = [10.5         , 0.510           , 0.64       , 2.40          , 0.107    , 6.0       , 15.0        , 0.99730, 3.09, 0.66    , 11.80] #quality = 7
wine4 = [7.4          , 0.360           , 0.30       , 1.8           , 0.074    , 17.0      , 24.0        , 0.99419, 3.24, 0.70    , 11.40] #quality = 8
```

```
Predicted Wine 1 Quality: 5.034354209899902, Actual quality value = 5
Predicted Wine 2 Quality: 6.14310884475708, Actual quality value = 6
Predicted Wine 3 Quality: 7.059370040893555, Actual quality value = 7
Predicted Wine 4 Quality: 8.156603813171387, Actual quality value = 8
```

Conclusions

1. Bigger Dataset:
- Lower MAE and MAPE values.
- More accurate results = Trustable AI predictions.
- Anticipation of data visualization and more concise trends.

2. Product Standard Normalization:
- Standardizing quality score for wines = Comparison.
- Competition for better quality across wineries and market = Improvement.
- Quality and product consistency reassurance for winemakers = Trade signature.

Questions?

Thank you for listening!