Wine Quality Analysis

Classify the wine ingredients with Data Science

Analyzing With Python Packages

By : Ramanan V M

Project Overview

- Objective: Analyze wine quality using physicochemical properties
- Predict wine quality (score 0–10)
- Task Type: Regression or Classification
- Focus on:
    - Feature relevance
    - Class imbalance
    - Outlier detection

Dataset Description

- **Fixed Acidity**
- **Volatile Acidity**
- **Citric Acid**
- **Residual Sugars**
- **Chlorides**
- **Free Sulphur Dioxide**
- **Total Sulphur Dioxide**
- **Density,ph**
- **Sulphates**
- **Alcohol**
- **Quality**

Data Characteristics

- Quality scores are **imbalanced**
- Most wines score between 5–6
- Rare: Scores near 0 or 10 (outliers)
- Visualization idea: bar plot of quality value counts

Feature Selection

- Method: SelectKBest with f_regression
- Top Features (example):
  - Alcohol
  - Sulphates
  - Volatile Acidity
  - Citric Acid
  - Density
- Helps improve model accuracy & interpretability

Model Building

- Model: Random Forest Regressor
- Split: Train/Test (80/20)
- Input: Selected top features

- Output: Predicted quality score

Conclusion

- Regression is effective for quality prediction
- Feature selection improves performance
- Alcohol & sulphates are most influential features
- Future Work:
  - Try classification (Low/Medium/High)
  - Handle class imbalance
  - Apply outlier detection techniques

References / Tools

- Dataset source: UCI ML Repository (Wine Quality)
- Tools: Python, Pandas, Scikit-learn, Seaborn, Matplotlib

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error,
mean_absolute_error, r2_score
from sklearn.feature_selection import SelectKBest,
f_regression
df = pd.read_csv('C:/Users/raman/Downloads/wine.csv')  #
Replace with your CSV path
print("Dataset Info:\n", df.info())
print("\nMissing Values:\n", df.isnull().sum())
print("\nDescriptive Stats:\n", df.describe())
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
X = df.drop('quality', axis=1)
y = df['quality']
```

```
selector = SelectKBest(score_func=f_regression, k=8)
X_new = selector.fit_transform(X, y)
selected_features = X.columns[selector.get_support()]
print("Selected Features:", list(selected_features))
X_selected = df[selected_features]
X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.2, random_state=42)
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("\nModel Evaluation:")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"R2 Score: {r2:.2f}")
importances = model.feature_importances_
plt.figure(figsize=(8, 5))
sns.barplot(x=importances, y=selected_features)
plt.title("Feature Importances")
plt.xlabel("Importance Score")
plt.ylabel("Features")
```

```
plt.tight_layout()
plt.show()
```

Output:

```
Descriptive Stats:
       fixed acidity  volatile acidity  citric acid  residual sugar  ...           pH    sulphates      alcohol      quality
count    1599.000000       1599.000000  1599.000000     1599.000000  ...  1598.000000  1599.000000  1599.000000  1598.000000
mean        8.319637          0.527821     0.270976        2.538806  ...     3.498586     0.658149    10.422983     5.636421
std         1.741096          0.179060     0.194801        1.409928  ...     0.080346     0.169507     1.065668     0.807665
min         4.600000          0.120000     0.000000        0.900000  ...     2.740000     0.330000     8.400000     3.000000
25%         7.100000          0.390000     0.090000        1.900000  ...     3.520000     0.550000     9.500000     5.000000
50%         7.900000          0.520000     0.260000        2.200000  ...     3.520000     0.620000    10.200000     6.000000
75%         9.200000          0.640000     0.420000        2.600000  ...     3.520000     0.730000    11.100000     6.000000
max        15.900000          1.580000     1.000000       15.500000  ...     3.900000     2.000000    14.900000     8.000000

[8 rows x 12 columns]
```

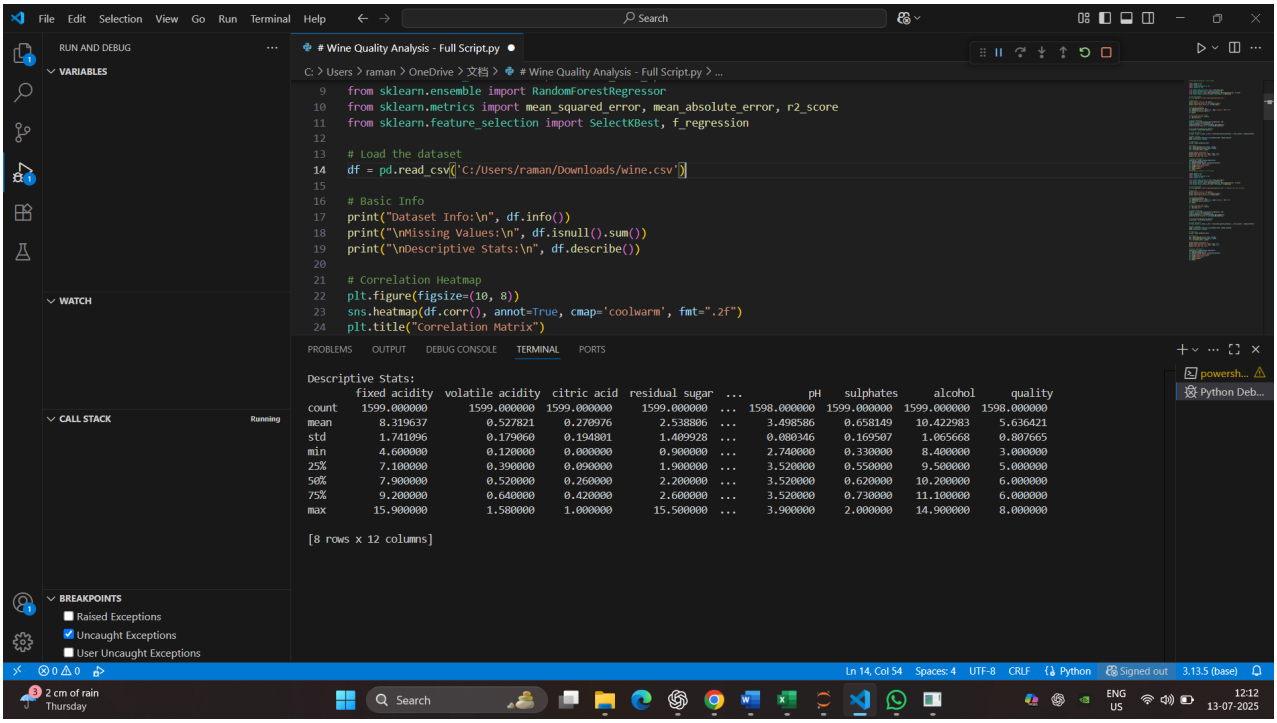
File Edit Selection View Go Run Terminal Help
Search
RUN AND DEBUG
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
Wine Quality Analysis - Full Script.py
C: > Users > raman > OneDrive > 文档 > # Wine Quality Analysis - Full Script.py > ...
9 from sklearn.ensemble import RandomForestRegressor
10 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
11 from sklearn.feature_selection import SelectKBest, f_regression
12
13 # Load the dataset
14 df = pd.read_csv('C:/Users/raman/Downloads/wine.csv')
15
16 # Basic Info
17 print("Dataset Info:\n", df.info())
18 print("\nMissing Values:\n", df.isnull().sum())
19 print("\nDescriptive Stats:\n", df.describe())
20
21 # Correlation Heatmap
22 plt.figure(figsize=(10, 8))
23 sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
24 plt.title("Correlation Matrix")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Descriptive Stats:
fixed acidity volatile acidity citric acid residual sugar ... pH sulphates alcohol quality
count 1599.000000 1599.000000 1599.000000 1599.000000 ... 1598.000000 1599.000000 1599.000000 1598.000000
mean 8.319637 0.527821 0.270976 2.538806 ... 3.498586 0.658149 10.422983 5.636421
std 1.741096 0.179060 0.194801 1.409928 ... 0.080346 0.169507 1.065668 0.807665
min 4.600000 0.120000 0.000000 0.900000 ... 2.740000 0.330000 8.400000 3.000000
25% 7.100000 0.390000 0.090000 1.900000 ... 3.520000 0.550000 9.500000 5.000000
50% 7.900000 0.520000 0.260000 2.200000 ... 3.520000 0.620000 10.200000 6.000000
75% 9.200000 0.640000 0.420000 2.600000 ... 3.520000 0.730000 11.100000 6.000000
max 15.900000 1.580000 1.000000 15.500000 ... 3.900000 2.000000 14.900000 8.000000
[8 rows x 12 columns]
powersh...
Python Deb...
Ln 14, Col 54 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.5 (base)
2 cm of rain
Thursday
Search
ENG
US
12:12
13-07-2025


Figure 1
Correlation Matrix
fixed acidity 1.00 -0.26 0.67 0.11 0.09 -0.15 -0.11 0.67 0.03 0.18 -0.06 0.12
volatile acidity -0.26 1.00 -0.55 0.00 0.06 -0.01 0.08 0.02 -0.05 -0.26 -0.20 -0.39
citric acid 0.67 -0.55 1.00 0.14 0.20 -0.06 0.04 0.36 0.01 0.31 0.11 0.23
residual sugar 0.11 0.00 0.14 1.00 0.06 0.19 0.20 0.36 0.04 0.01 0.04 0.01
chlorides 0.09 0.06 0.20 0.06 1.00 0.01 0.05 0.20 -0.26 0.37 -0.22 -0.13
free sulfur dioxide -0.15 -0.01 -0.06 0.19 0.01 1.00 0.67 -0.02 -0.03 0.05 -0.07 -0.05
total sulfur dioxide -0.11 0.08 0.04 0.20 0.05 0.67 1.00 0.07 -0.16 0.04 -0.21 -0.18
density 0.67 0.02 0.36 0.36 0.20 -0.02 0.07 1.00 -0.04 0.15 -0.50 -0.17
pH 0.03 -0.05 0.01 0.04 -0.26 -0.03 -0.16 -0.04 1.00 -0.26 0.23 0.13
sulphates 0.18 -0.26 0.31 0.01 0.37 0.05 0.04 0.15 -0.26 1.00 0.09 0.25
alcohol -0.06 -0.20 0.11 0.04 -0.22 -0.07 -0.21 -0.50 0.23 0.09 1.00 0.48
quality 0.12 -0.39 0.23 0.01 -0.13 -0.05 -0.18 -0.17 0.13 0.25 0.48 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
density
pH
sulphates
alcohol
quality
2 cm of rain
Thursday
Search
ENG
US
12:11
13-07-2025