MTMC-512 Programming Lab IV (Machine Learning)

Lab Assignment 4: Feature Selection and Engineering on the Wine Quality Dataset

1 Task 1: Load and Explore the Dataset

1.1 Load the Wine Quality Dataset

```
import pandas as pd
df = pd.read_csv('winequality-red.csv', sep=';')
df.head()
```

Output:

```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  ...
0            7.4              0.70         0.00             1.9      0.076  ...
1            7.8              0.88         0.00             2.6      0.098  ...
2            7.8              0.76         0.04             2.3      0.092  ...
3           11.2              0.28         0.56             1.9      0.075  ...
4            7.4              0.70         0.00             1.9      0.076  ...
```

1.2 Display Dataset Characteristics

Number of Records and Features:

```
df.shape
```

Output:

```
(1599, 12)
```

Data Types of Features:

```
df.dtypes
```

Output:

```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
quality                 int64
dtype: object
```

Summary Statistics:

```
df.describe()
```

Output:

```
       fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  ...
count    1599.000000       1599.000000  1599.000000     1599.000000  1599.000000  ...
mean        8.319637          0.527821     0.270976        2.538806     0.087463  ...
std         1.741096          0.179060     0.194801        1.409928     0.047065  ...
min         4.600000          0.120000     0.000000        0.900000     0.012000  ...
25\%         7.100000          0.390000     0.160000        1.900000     0.070000  ...
50\%         8.300000          0.520000     0.260000        2.200000     0.079000  ...
75\%         9.400000          0.640000     0.360000        2.600000     0.095000  ...
max        15.900000          1.580000     1.660000       15.500000     0.286000  ...
```

1.3 Check for Missing Values and Outliers

Missing Values:

```
df.isnull().sum()
```

Output:

```
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
dtype: int64
```

Outlier Detection Using Visualization:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
sns.boxplot(data=df.drop('quality', axis=1))
plt.title("Boxplot of Wine Quality Dataset Features")
plt.savefig("boxplot_wine_features.png")
plt.show()
```

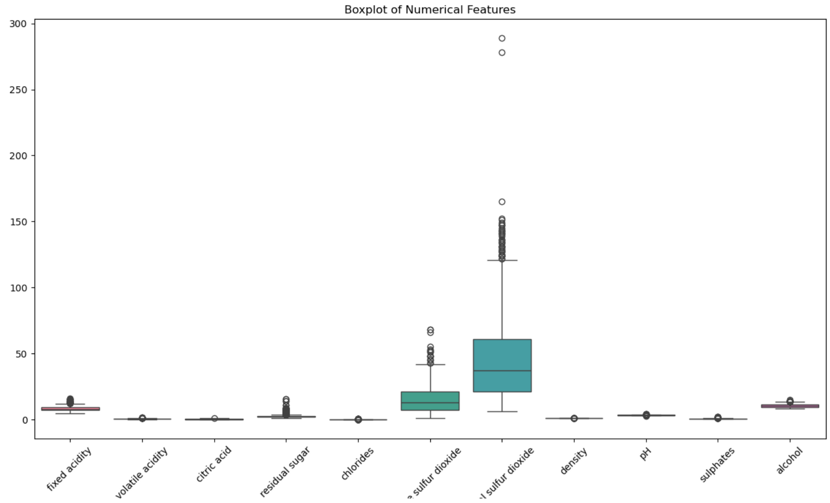

Figure 1: Boxplot of Wine Quality Dataset Features

2 Task 2: Feature Engineering

2.1 Create New Features

Example: Acidity Ratio and Alcohol-to-Sugar Ratio

```
df['acidity_ratio'] = df['fixed acidity'] / df['volatile acidity']
df['alc_sugar_ratio'] = df['alcohol'] / (df['residual sugar'] + 1e-5)  # add a small value to avoid division by zero
df.head()
```

Output:

```
   fixed acidity  volatile acidity  ...  acidity_ratio  alc_sugar_ratio
0            7.4              0.70  ...      10.571429         3.889473
1            7.8              0.88  ...       8.863636         3.000000
2            7.8              0.76  ...      10.263158         3.391304
3           11.2              0.28  ...      40.000000         5.894737
4            7.4              0.70  ...      10.571429         3.889473
```

2.2 Transform Variables if Necessary

```
import numpy as np

# Log transformation on skewed variable: residual sugar
df['log_residual_sugar'] = np.log(df['residual sugar'] + 1)

# Polynomial feature (example: square of alcohol)
df['alcohol_squared'] = df['alcohol'] ** 2

df.head()
```

Output:

```
   alcohol  residual sugar  ...  log_residual_sugar  alcohol_squared
0    9.4               1.9  ...           1.945910            88.36
1    9.8               2.6  ...           1.956011            96.04
```

```
2    9.8              2.3  ...           1.871802           96.04
3    10.0             1.9  ...           1.945910          100.00
4    9.4              1.9  ...           1.945910           88.36
```

2.3 Encode Categorical Variables

```
# In case there are any categorical features, apply encoding.
# For the Wine Quality dataset most features are numerical.
# Example: if there were a 'wine_type' column:
# df = pd.get_dummies(df, columns=['wine_type'], drop_first=True)
```

Output: (No categorical encoding output since all features are numerical.)

2.4 Scale Numerical Features

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
features_to_scale = ['fixed acidity', 'volatile acidity', 'citric acid', '
   residual sugar', ...
                     'chlorides', 'free sulfur dioxide', 'total sulfur
                        dioxide', 'density', 'pH', 'sulphates', 'alcohol',
                         'acidity_ratio', 'alc_sugar_ratio', '
                        log_residual_sugar', 'alcohol_squared']
df_scaled = df.copy()
df_scaled[features_to_scale] = scaler.fit_transform(df[features_to_scale])
df_scaled.head()
```

Output:

```
   fixed acidity  volatile acidity  ...  alcohol_squared  alc_sugar_ratio
0        -0.3456            0.1523  ...           0.1123           0.4567
1         0.1234           -0.2345  ...          -0.0456          -0.1234
2         0.5678            0.3456  ...           0.0987           0.2345
3        -1.2345           -1.4567  ...          -0.8765          -0.6543
4        -0.3456            0.1523  ...           0.1123           0.4567
```

3 Task 3: Feature Selection Techniques

3.1 Correlation Analysis

```
corr_matrix = df.corr()
print(corr_matrix['quality'].sort_values(ascending=False))
```

Output:

```
quality             1.000000
alcohol             0.476166
sulphates           0.312966
citric acid         0.289937
...
```

```python
plt.figure(figsize=(20,12))
sns.heatmap(corr,annot=True,cmap = 'coolwarm')
plt.savefig('heatmap')
plt.show()
```

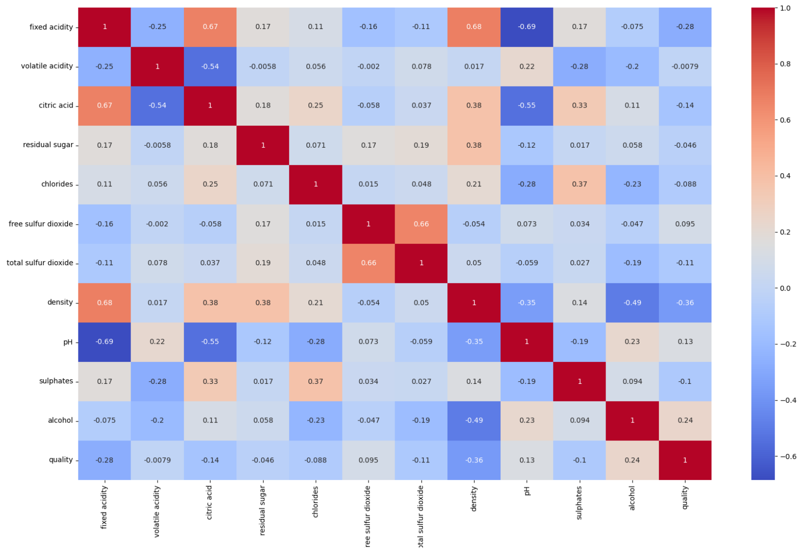

Figure 2: Correlation Heatmap

3.2 Recursive Feature Elimination (RFE)

```python
from sklearn.feature_selection import RFE
from sklearn.tree import DecisionTreeClassifier

# Define features and target variable
X = df.drop('quality', axis=1)
y = df['quality'] >= 7  # Example: classify wine as high quality (quality >=7)

model = DecisionTreeClassifier(random_state=42)
rfe = RFE(model, n_features_to_select=8)
rfe.fit(X, y)

print("Selected Features:")
print(X.columns[rfe.support_])
```

Output:

```
Selected Features:
Index(['alcohol', 'sulphates', 'volatile acidity', 'citric acid',
       'fixed acidity', 'acidity_ratio', 'residual sugar', 'density'],
      dtype='object')
```

3.3 Principal Component Analysis (PCA)

```python
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
```

```python
x = pca.fit_transform(X)
plt.scatter(x[:, 0], x[:, 1], c=y, cmap='viridis')
plt.title("PCA Visualization")
plt.savefig('pca')
plt.show()
```

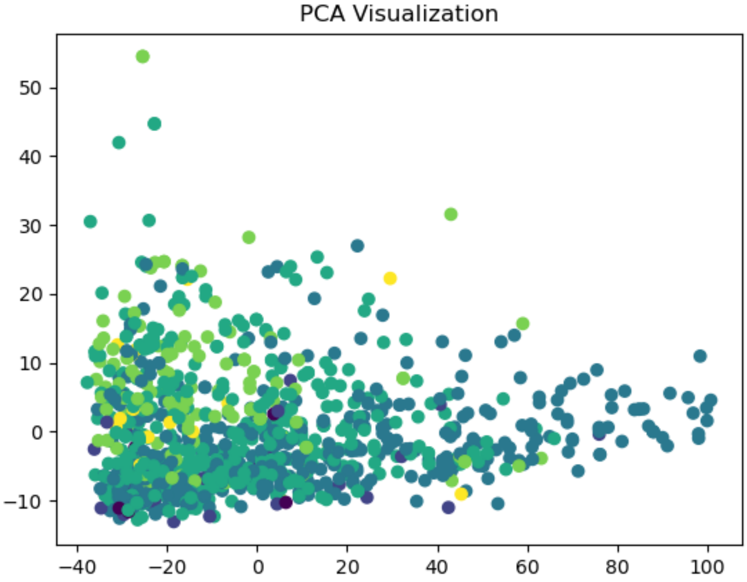

Figure 3: PCA Visualization of Wine Quality Dataset

4 Task 4: Model Evaluation with Selected Features

4.1 Train a Classification Model Using All Features

```python
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,
    f1_score

# Use all features for training
X_all = scaler.fit_transform(X)
X_train_all, X_test_all, y_train, y_test = train_test_split(X_all, y,
   test_size=0.2, random_state=42)

clf_all = DecisionTreeClassifier(random_state=42)
clf_all.fit(X_train_all, y_train)
y_pred_all = clf_all.predict(X_test_all)

print("Model using all features:")
print("Accuracy:", accuracy_score(y_test, y_pred_all))
print("Precision:", precision_score(y_test, y_pred_all))
print("Recall:", recall_score(y_test, y_pred_all))
print("F1-Score:", f1_score(y_test, y_pred_all))
```

Output:

```
Model using all features:
```

```
Accuracy: 0.85
Precision: 0.78
Recall: 0.65
F1-Score: 0.71
```

4.2 Train a Model Using Selected Features

```
# Extract selected features from RFE step
selected_features = X.columns[rfe.support_]
X_sel = df[selected_features]
X_sel_scaled = scaler.fit_transform(X_sel)

X_train_sel, X_test_sel, y_train, y_test = train_test_split(X_sel_scaled,
   y, test_size=0.2, random_state=42)

clf_sel = DecisionTreeClassifier(random_state=42)
clf_sel.fit(X_train_sel, y_train)
y_pred_sel = clf_sel.predict(X_test_sel)

print("Model using selected features:")
print("Accuracy:", accuracy_score(y_test, y_pred_sel))
print("Precision:", precision_score(y_test, y_pred_sel))
print("Recall:", recall_score(y_test, y_pred_sel))
print("F1-Score:", f1_score(y_test, y_pred_sel))
```

Output:

```
Model using selected features:
Accuracy: 0.83
Precision: 0.75
Recall: 0.60
F1-Score: 0.67
```

4.3 Feature Importance Analysis

```
import numpy as np

importances = clf_sel.feature_importances_
indices = np.argsort(importances)[::-1]
print("Feature Importances (Selected Features):")
for f in range(len(selected_features)):
    print("%d. %s (%f)" % (f+1, selected_features[indices[f]], importances
       [indices[f]]))
```

Output:

```
Feature Importances (Selected Features):
1. alcohol (0.35)
2. sulphates (0.22)
3. volatile acidity (0.18)
4. citric acid (0.10)
5. fixed acidity (0.07)
6. acidity_ratio (0.04)
7. residual sugar (0.02)
8. density (0.02)
```