

## Machine Learning Assignment 3

Arya Sagar  
202200673  
RN-08

---

### Problem Statement

Consider remaining features of the wine data and prepare a prediction model for predicting quality of wine.

Python code:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

file_path = 'C:/Users/Maithili/Downloads/winedata.csv' # Update the path if
needed
df = pd.read_csv(file_path)

# Convert quality into binary classification (Good: 1, Bad: 0)
df['quality'] = df['quality'].apply(lambda q: 1 if q >= 6 else 0)

# Split data into features (X) and target (y)
X = df.drop(columns=['quality'])
y = df['quality']

# Split into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```

# Predict on test data
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print("\nConfusion Matrix:\n", conf_matrix)
print("\nClassification Report:\n", class_report)

# Feature Importance
feature_importance = pd.DataFrame({'Feature': X.columns, 'Importance':
model.feature_importances_})
feature_importance = feature_importance.sort_values(by='Importance',
ascending=False)

```

Accuracy: 0.80

Confusion Matrix:

```
[[121  28]
 [ 35 136]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.78	0.81	0.79	149
1	0.83	0.80	0.81	171
accuracy			0.80	320
macro avg	0.80	0.80	0.80	320
weighted avg	0.80	0.80	0.80	320

```
# Plot feature importance
plt.figure(figsize=(10, 5))
sns.barplot(x=feature_importance['Importance'],
            y=feature_importance['Feature'])
plt.title("Feature Importance in Wine Quality Prediction")
plt.show()
```

