Machine Learning Assignment Report

1. Problem Description

This machine learning project focuses on classifying wine quality based on various physicochemical features. We aim to predict whether a wine sample is of high quality (score >= 7) using two supervised learning algorithms: Logistic Regression and Random Forest.

2. Dataset Description

The dataset combines two publicly available datasets of red and white wine qualities from the UCI Machine Learning Repository. Each entry includes 11 physicochemical attributes (e.g., acidity, pH, alcohol) and a quality score from 0 to 10. For classification purposes, a new binary label 'quality_label' was created where 1 represents high-quality wines (quality >= 7) and 0 represents lower quality.

3. Methodology

1. The datasets were loaded, labeled (red = 0, white = 1), and combined.
2. The target variable 'quality_label' was defined.
3. Features were scaled using StandardScaler.
4. Data was split into training and testing sets using stratified sampling.
5. Two models were trained:
   - Logistic Regression (with balanced class weights)
   - Random Forest Classifier (also with balanced class weights)
6. Evaluation metrics included accuracy, classification report, and ROC AUC score.

4. Results and Discussion

Logistic Regression and Random Forest models were both evaluated on the test set. The Random Forest model showed higher classification performance overall. ROC AUC scores and classification reports indicated better precision and recall for the Random Forest model, which is more capable of handling nonlinear relationships and interactions between features.

5. Conclusion and Future Work

The Random Forest model outperformed Logistic Regression in predicting high-quality wine. Future improvements could involve:

- Hyperparameter tuning
- Cross-validation
- Ensemble techniques
- Incorporating additional external features such as price or origin

6. Appendix: Source Code

```
# Import libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, roc_auc_score

# Load data
red = pd.read_csv('winequality-red.csv', sep=';')
white = pd.read_csv('winequality-white.csv', sep=';')
```

Machine Learning Assignment Report

```
# Add 'type' column and combine

red['type'] = 0

white['type'] = 1

wine = pd.concat([red, white], ignore_index=True)


# Create binary target (quality >=7)

wine['quality_label'] = (wine['quality'] >= 7).astype(int)


# Split features (X) and target (y)

X = wine.drop(['quality', 'quality_label'], axis=1)

y = wine['quality_label']


# Train-test split with stratification

X_train, X_test, y_train, y_test = train_test_split(

    X, y, test_size=0.2, random_state=42, stratify=y

)


# Scale features

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)


# Logistic Regression

lr = LogisticRegression(class_weight='balanced', random_state=42)
```

```
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)
print("Logistic Regression Results:")
print(classification_report(y_test, y_pred_lr))
print("ROC AUC:", roc_auc_score(y_test, lr.predict_proba(X_test_scaled)[:, 1]))

# Random Forest
rf = RandomForestClassifier(class_weight='balanced', random_state=42)
rf.fit(X_train_scaled, y_train)
y_pred_rf = rf.predict(X_test_scaled)
print("\nRandom Forest Results:")
print(classification_report(y_test, y_pred_rf))
print("ROC AUC:", roc_auc_score(y_test, rf.predict_proba(X_test_scaled)[:, 1]))
```