

Project Report: Practical Machine Learning and Data Exploration
Project 2-2

Student: Eman Alhabashi

Project 2

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, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.metrics import classification_report, accuracy_score

df = pd.read_csv("WineQT.csv")
if 'Id' in df.columns:
    df = df.drop('Id', axis=1)
df.fillna(df.mean(), inplace=True)
X = df.drop('quality', axis=1)
y = df['quality']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("🕒 Processing... Finding the best model settings (this takes ~15 seconds)...")
rf_params = {'n_estimators': [50, 100], 'max_depth': [None, 10]}
rf_grid = GridSearchCV(RandomForestClassifier(random_state=42), rf_params, cv=3)
```

```
rf_grid.fit(X_train, y_train)

best_model = rf_grid.best_estimator_


#Chi-Square

chi_selector = SelectKBest(score_func=chi2, k=5)

chi_selector.fit(X_train, y_train)

chi_feat = X.columns[chi_selector.get_support()]


# Random Forest Importance

rf_imp_model = RandomForestClassifier(n_estimators=100, random_state=42).fit(X_train,
y_train)

indices = np.argsort(rf_imp_model.feature_importances_)[-5:]

rf_feat = X.columns[indices]


# Evaluate

y_pred = best_model.predict(X_test)

print("\n" + "="*40)

print("    PROJECT RESULTS")

print("="*40)

print(f"Final Model Accuracy: {accuracy_score(y_test, y_pred):.2%}")

print(f"Best Parameters: {rf_grid.best_params_}")

print(f"\nTop 5 Features (Chi-Square): {list(chi_feat)}")

print(f"Top 5 Features (RF Importance): {list(rf_feat)}")

print("\nFinal Classification Report:")

print(classification_report(y_test, y_pred))
```

```
plt.figure(figsize=(10, 6))  
sns.barplot(x=rf_imp_model.feature_importances_, y=X.columns, palette='magma')  
plt.title("Feature Importance - Embedded Method Results")  
plt.xlabel("Importance Score")  
plt.ylabel("Wine Features")  
plt.show()
```