**2 Wine Quality Prediction Model Project**

**Report**:

**Wine Quality Prediction**

The objective of this project is to predict the quality of red wine using machine learning. The dataset used for this project is the "winequality-red.csv" dataset which consists of 1599 instances and 12 attributes. The attributes include fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol and quality. The quality attribute is the target variable which is the quality rating of the wine ranging from 0 to 10.

**Data Preprocessing**

The first step of any machine learning project is to preprocess the data. In this project, we have loaded the dataset using pandas and created two variables - predictor (X) and target (y). We have dropped the quality column from the predictor variable and created a new target variable where the quality rating is classified as 1 if it is greater than or equal to 7 and 0 otherwise. This is done to convert the problem into a binary classification problem. We have also split the data into training and testing sets using the train\_test\_split function from scikit-learn.

**Model Training**

We have used the Random Forest Classifier algorithm to train the model. The Random Forest Classifier is an ensemble learning method that uses decision trees to make predictions. It works by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. We have trained the model using the training data and calculated the accuracy of the model on the test data.

**Web Application**

We have also created a web application using Streamlit which allows the user to input the features of the wine and get the predicted quality rating. The user inputs the features separated by commas in a text box, and the web application outputs whether the wine is of good or bad quality based on the predicted quality rating.

**Conclusion**

In this project, we have successfully predicted the quality of red wine using machine learning. We have used the Random Forest Classifier algorithm and achieved an accuracy of 70% on the test data. We have also created a web application using Streamlit which allows the user to input the features of the wine and get the predicted quality rating. This project can be extended by using other machine learning algorithms and by using more data

**About Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

import streamlit as st

# Load the wine dataset

wine\_df = pd.read\_csv('winequality-red.csv')

# Create the predictor (X) and target (y) variables

X = wine\_df.drop('quality', axis=1)

y = wine\_df['quality'].apply(lambda yval: 1 if yval >= 7 else 0)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=3)

# Train the Random Forest Classifier model

model = RandomForestClassifier()

model.fit(X\_train, Y\_train)

# accuracy on test data

X\_test\_prediction = model.predict(X\_test)

print(accuracy\_score(X\_test\_prediction, Y\_test))

# web app

st.title("Wine Quality Prediction Model")

input\_text = st.text\_input('Enter all Wine Features (comma-separated)')

if input\_text:

    input\_text\_list = input\_text.split(',')

    # Check if the number of input features matches the model's expected input size

    if len(input\_text\_list) == len(X.columns):

        try:

            features = np.asarray(input\_text\_list, dtype=np.float32)  # Convert input to float

            prediction = model.predict([features])

            if prediction[0] == 1:

                st.write("Good Quality Wine")

            else:

                st.write("Bad Quality Wine")

        except ValueError:

            st.write("Please enter valid numeric values for all wine features.")

    else:

        st.write("Please enter all wine features.")

**About Data:**

The data provided appears to be a tabular dataset with 12 columns and multiple rows, where each row represents a sample of wine and each column represents a different feature of that wine. The features are described below:

**Fixed acidity**: the amount of fixed acids in the wine (g/dm^3)

**Volatile acidity**: the amount of volatile acids in the wine (g/dm^3)

**Citric acid:** the amount of citric acid in the wine (g/dm^3)

**Residual sugar**: the amount of residual sugar in the wine (g/dm^3)

**Chlorides:** the amount of chlorides in the wine (g/dm^3)

**Free sulfur dioxide:** the amount of free sulfur dioxide in the wine (mg/dm^3)

**Total sulfur dioxide:** the amount of total sulfur dioxide in the wine (mg/dm^3)

**Density**: the density of the wine (g/cm^3)

**pH**: the pH level of the wine

**Sulphates**: the amount of sulphates in the wine (g/dm^3)

**Alcohol**: the alcohol content of the wine (% vol)

**Quality**: a rating of the quality of the wine (scored between 0 and 10)

**About Code:**

Here, we are reading the wine dataset from a CSV file using Pandas and storing it in a Pandas DataFrame called wine\_df.

We are creating the predictor variable X by dropping the quality column from the wine\_df DataFrame. We are creating the target variable y by applying a lambda function to the quality column, which sets the value of 1 for quality score of 7 or higher and 0 for quality scores below 7.

We are splitting the dataset into training and testing sets using the train\_test\_split() function from Scikit-learn. The training set contains 80% of the data, and the testing set contains 20% of the data. The random\_state parameter sets the seed value for the random number generator to ensure that we get the same results every time we run the code.

We are initializing a Random Forest Classifier model and fitting it to the training data using the fit() method.

We are using the trained model to predict the wine quality for the testing data and calculating the accuracy of the model on the testing data using the accuracy\_score() function from Scikit-learn. The accuracy score is printed to the console.

Finally, we are using Streamlit to create a simple web app that allows users to input wine features and predict the quality of the wine using the trained model. If the predicted quality score is 1, the app displays "Good quality wine", otherwise it displays "Bad Quality wine".