**RED WINE QUALITY PREDICTION PROJECT**

**Project Description**

The dataset is related to red and white variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

This dataset can be viewed as classification task. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods. Attribute Information

**Input variables (based on physicochemical tests) (Features)**

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol

**Output variable (based on sensory data) (Target)**

1. quality (score between 0 and 10)

What might be an interesting thing to do, is to set an arbitrary cutoff for your dependent variable (wine quality) at e.g. 7 or higher getting classified as 'good/1' and the remainder as 'not good/0'. This allows you to practice with hyper parameter tuning on e.g. decision tree algorithms looking at the ROC curve and the AUC value.

You need to build a classification model.

Inspiration Use machine learning to determine which physiochemical properties make a wine 'good'!

**Dataset Link-**

[**https://github.com/FlipRoboTechnologies/ML-Datasets/blob/main/Red%20Wine/winequality-red.csv**](https://github.com/FlipRoboTechnologies/ML-Datasets/blob/main/Red%20Wine/winequality-red.csv)

1. **Problem Statement :**

**Predicting on the test data of Red Wine Quality Dataset and finding the accuracy of the model using classification model.**

By using dataset link we can check and import data. Click on Raw Data and copy paste link of raw data in our python program (code).

**Import Libraries**

Import libraries are starting phase to start any program in any language. Some important Libraries are:

* import pandas as pd
* import numpy as np
* import warnings
* warnings.filterwarnings('ignore')

1. **Data Analysis**

Step by step we check our csv file dataset.

1. What is shape of dataframe : 1599x12
2. Information of dataframe : By using this we can get actual information of all data like In this dataframe 1599 rows and 12 columns are present, no null values are present, float64 and int64 type data are available only.
3. Where any missing data is present or not, confirm using heatmap as well, if NaN is present then findout them and fill missing data using mode(), mean() or any other method as well. In our dataframe no missing values or NaN are present.
4. Check Unique values present in all columns.
5. Check Description:

* There are No missing Values.
* Data distribustion is good.
* Some Outliers are present, but we confirm using plots.

1. **Exploratory Data Analysis**

It helps data to visualize the patterns, characteristics and relationships between variables.

In this step we can check our data in different format like graphs, pattern.

* In this dataset we first start from pairplot, by using pairplot we can clear the data distribution of all input and output variable.
* Plot Distplot and check data distribution of all variables, by use of distplot we clear that our data is distributed uniformly or not. In our red wine dataset data not distributed uniformly.
* Check Box plot to find out outliers are actually present or not, in this red wine dataset lots of outliers are present.
* Check value\_counts of every column separatly.
* We can separate our feature and target variable
* Use z\_score to remove outliers present in features.
* Check correlation of variables using corr() with heatmap.

1. **Pre-processing Pipeline**

* **Data Cleaning**
* **Data Integration**
* **Data Transformation**
* **Data Reduction**
* Create x and y variable

x = features and y = target variable

y = quality column

x = all column excluding quality column

* Use standard scaler to standardize our x value.
* Use train\_test\_split to train the input data to predict test data.

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size=0.25, random\_state=348)

* Use label encoder to covert quality column data in two parts like 2 – 6.5 – 8, 2-6.5 is Bad Quality and 6.5 - 8 is Good Quality. And then covert Bad to Zero (0) and Good to One (1).
* Now no outliers are present and all features are important, data is clean.
* Check outliers and skewness and remove it.
* Checking all columns which column are more important or correlate with target variable (quality of wine), In this project we considering all columns are important and not dropping any columns.
* Now our Data is totally clean and we can start Model Building step.

1. **Building Machine Learning Models**

* Import classification model like Decision Tree Classifier, Random Forest Classifier, Extra Trees Classifier, Ada Boost Classifier.
* Initialize the model and fit the model.
* Check accuracy of model using

**Accuracy** of model using Decision Tree Classifier is 64%.

* Checking **confusion matrix**.
* Now using **Hyperparameter Tunning** to tunned the parameters.

Using GridSearchCV to tunned the parameter and select best parameter to get best train and test accuracy.

After tunning parameters we get **train accuracy : 66.39%** and **test accuracy : 60.75%**

* Run model one by one after tunning more parameter and check accuracy score and confusion matrix.
* Using **Random Forest Classifier**, we get **Accuracy = 77%**

Using **Extra Trees Classifier**, we get **Accuracy = 76.66%**

Using **Ada Boost Classifier**, we get **Accuracy = 70.62%**

Using **Decision Tree Classifier**, we get **Accuracy = 71.66%**

* Save the model using pickle and load the model.
* Check predicted and actual value of wine quality.

1. **Conclusion**

**By using Extra Trees Classifier, we get 76.66% accuracy to predict quality of red wine. If we tunned more parameters then we will also get more better accuracy to predict Wine Quality.**