**GENERAL INFORMATION OF THE PROJECT;**

**Project Name: Wine Quality**

**Key Words:** quality (target), fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol

**Subject of the Project:**Machine Learning analyses that on wine quality dataset

**Details of the Project:**

Step 1: Set up your environment.

* Python 2.7+ or Python 3
* NumPy
* Pandas
* Scikit-Learn (a.k.a. sklearn)

## Step 2: Import libraries and modules.

import numpy as np

import pandas as pd

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

from sklearn import preprocessing

from sklearn.ensemble import RandomForestRegressor

from sklearn.pipeline import make\_pipeline

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.externals import joblib

## Step 3: Load red wine data.

dataset\_url = 'http://mlr.cs.umass.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv'

data = pd.read\_csv(dataset\_url, sep=';')

## Step 4: Split data into training and test sets.

y = data.quality

X = data.drop('quality', axis=1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

                                                    test\_size=0.2,

                                                    random\_state=123,

                                                    stratify=y)

## Step 5: Declare data preprocessing steps.

pipeline = make\_pipeline(preprocessing.StandardScaler(),

                         RandomForestRegressor(n\_estimators=100))

## Step 6: Declare hyperparameters to tune.

pipeline = make\_pipeline(preprocessing.StandardScaler(),

                         RandomForestRegressor(n\_estimators=100))

## Step 7: Tune model using a cross-validation pipeline.

clf = GridSearchCV(pipeline, hyperparameters, cv=10)

clf.fit(X\_train, y\_train)

## Step 8: Refit on the entire training set.

## Step 9: Evaluate model pipeline on test data.

pred = clf.predict(X\_test)

print r2\_score(y\_test, pred)

print mean\_squared\_error(y\_test, pred)

## Step 10: Save model for future use.

## joblib.dump(clf, 'rf\_regressor.pkl')

**Targeted Objectives and Results:** Used R2 scoring and mean squared error methods and got 0.45 and 0.35 respectively.

**Potential of Commercialization:**

Wine producers can make more their profit and sales in wine market if they use these values. And can reduce the cost very effectively. Presenting wines to customers according to their drinking habits and their statistical taste can be used to offer a new wine and probably cause with good feedbacks. Of course, this will make more customer to the wine producer.

**Direction of Innovative:** This project can be used in to find the place of brand new wine in the market. Also, using the values can be produce a new wine that will best fit to customer’s desire.

**Target group:** Gourmets and entrepreneurs whose will be attend to wine market.

**References:**

[1] Python Machine Learning Tutorial, Scikit-Learn: Wine Snob Edition. (2019, January 25). Retrieved from https://elitedatascience.com/python-machine-learning-tutorial-scikit-learn.

[2] Wine Dataset. (n.d.). Retrieved December 12, 2019, from http://mlr.cs.umass.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv.

[3] https://github.com/kayamuhammedd/CMPE462