# MLND Capstone Project – Predict Wine Class

## Domain Background

Wine tasting is the sensory examination and evaluation of wine. While the practice of wine tasting is as ancient as its production, a more formalized methodology has slowly become established from the 14th century onwards. Modern, professional wine tasters (such as sommeliers or buyers for retailers) use a constantly evolving specialized terminology which is used to describe the range of perceived flavours, aromas and general characteristics of a wine. More informal, recreational tasting may use similar terminology, usually involving a much less analytical process for a more general, personal appreciation. (Source: Wikipedia)

One of the goal of AI/ML algorithms is to reduce dependency on human intelligence and develop scientific methods for decision making. Here we will be exploring various algorithm to develop model to predict wine class scientifically.

## Problem Statement

In this project we are going to analyse the data, which is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The aim is to develop a model to predict class of wine based on 13 parameters which is result of laboratory test of the wine. As per UCI website the original dataset had 30 attributes but available dataset has 30 attributes.
It is a typical classification problem where we have to predict class based on the given attributes/features, I will explore various classifiers at predicting a certain wine type from available UCI's wine dataset.

## Datasets and Inputs

The initial data set had around 30 variables, but on UCI's website only 13-dimensional version is available. (All attributes are continuous)

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Colour intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

In a classification context, this is a well posed problem with "well behaved" class structures. Number of Instances: class-1(59), class-2(71), class-3(48), Total 178 records in csv format.

https://archive.ics.uci.edu/ml/machine-learning-databases/wine/

## Solution Statement

In the given problem I would try to use Linear Classification model (Logistics Regression) and non-Linear classification model (KNN, SVM, Decision Tree, Naïve Bayes). First I will do the initial check using default parameters for the models and check which algorithm is performing best. I would try to check accuracy before and after doing standardscaler transformation of the data and check which is giving better accuracy. After selecting best algorithms, I would tune it's parameters using GridSearch to get the best result.

Finally I would use Ensembles algorithms (Gradient Boosting, Random Forest, Ada Boost, Extra Trees) to check if accuracy can be improved further, after doing initial check with default parameters I will use tune the parameters with Grid search and get the combination of the parameters for best output.

After selecting best model using above steps I would select the best model and report evaluation matrix for same.

## Benchmark Model

I will use following result on Kaggle as benchmark:

https://www.kaggle.com/brynja/testing-classifiers-for-wine-variety-prediction

Accuracy: 0.98592

I will try tune my algorithms and try few ensembles to improve accuracy and get better result than above mention model. However if I talk in term of independent project I would keep LogisticRegression as benchmark model and would compare other model with it for benchmarking the result.

## Evaluation Metrics

I will use accuracy and scikit-learn Classification Report as evaluation matrix which reports precision, recall and f1-score.

Accuracy measures how often the classifier makes the correct prediction. It's the ratio of the number of correct predictions to the total number of predictions (the number of test data points).

Precision =

[True Positives/(True Positives + False Positives)]

Recall(sensitivity) =

[True Positives/(True Positives + False Negatives)]

F1-Score =

Precision and Recall can be combined to get the F1 score, which is weighted average(harmonic mean) of the precision and recall scores. This score can range from 0 to 1, with 1 being the best possible F1 score(we take the harmonic mean as we are dealing with ratios).

## Project Design

I will use following steps to come up with prediction model:

- ✓ Prepare Problem
  - Load all the required python libraries
  - Load the csv dataset from UCI's website
- ✓ Summarize Data
  - Use Descriptive statistics to get insight of the available data
  - Data visualizations with histogram to check characteristic of the data
- ✓ Prepare Data
  - Data Cleaning may not be required as it contains well behaved data
  - Feature Selection - I will be using all the 13 attributes to develop the predictive model
  - Data Transforms - I will try StandardScaler transformation of the data and check the algorithms accuracy before and after the transformation.
- ✓ Evaluate Algorithms
  - Split-out the dataset into training and validation set
  - Defining test options using scikit-learn such as cross-validation and the evaluation metric to use.
  - Selecting few Linear (Logistic Regression) and Non-Linear Algorithms (KNN, SVM, Decision Tree, Naïve Bayes) and do the initial check with default parameters.
  - The algorithms all use default tuning parameters. We will display the mean and standard deviation of accuracy for each algorithm as we calculate it and collect the results for use later.
  - Once we get the best fit algorithm for our problem we will try to tune the parameters for best result.
- ✓ Improve Accuracy
  - Tuning of the selected Algorithm to achieve highest accuracy
  - Trying Ensembles (Gradient Boosting, Random Forest, Ada Boost, Extra Trees) to improve accuracy
- ✓ Finalize Model
  - Predictions on validation dataset using the best algorithm chosen from above steps
  - Create standalone model on entire training dataset
  - Save model for later use.