Wine Quality Prediction using Machine Learning

A PROJECT REPORT

Submitted by

Bhanupriya Sharma

in partial fulfillment for the award of the degree

of

BACHELORS IN COMPUTER APPLICATION

IN (AI & MACHINE LEARNING)

THE ICFAI UNIVERCITY

IcfaiTech::IUJ

25 oct 2023

ABSTRACT

The main goal of this project is to predict wine quality whether it is good or bad. For centuries tasting has been done by humans and they have always predicted on the basis of sensory organs. But in recent times the industries are adopting newer technologies and applying them in all kinds of areas. But, still there are many areas in which human expertise is needed like product quality assurance. Nowadays, it becomes an expensive process as the demand of product is growing over the time. Therefore, this project searches different machine learning techniques such as MLP classifier, Decision Tree classifier, Support Vector Machines (SVM) for product quality assurance. These techniques do quality assurance process with the help of available characteristics of product and automate the process by minimizing human interference.

List of Abbreviations

ACRONYM	DEFINITIONS
1. SVM	Support Vector Machines
2. KNN	K-Neighbour Nearest
3. MLP	Multi-Layer Perceptron
4. SGD	Stochastic Gradient Descent

Chapter – 1

1.1 INTRODUCTION

The most defining period of human history will always be remembered as computing moved from mainframes to PCs to cloud and now to artificial intelligence. An important area of artificial intelligence which came in lime light, called as Machine Learning, allows computers to get into some kind of self-learning mode involuntary. With the concepts and ideas from machine learning, we have been able to spread from miscellaneous accurate reduplications to big data iteration that too with at a marvellous speed. This spectacle has been in momentum over the last several years. On the other hand, data mining includes data discovery and sorting it among large data sets vacant to identify the required designs and begin affiliations with the aim of answering teething worries over and done with data analysis. Basically linking, device learning and data mining use the same type of method and set of processes, except the kind of data pre-dealing out and end guess varies. Between these two core expanses to predict and present the truest results potential.

1.2 PROBLEM STATEMENT

Predicting on the test data of Red Wine Quality Dataset and finding the accuracy of the model using Logistic Regression, involving import of dataset, quality check on the data (Data Wrangling), and performing Exploratory Data Analysis (Univariate and Bivariate Analysis) using Histograms, Boxplots and Scatter Plots. Thus, modelling the dataset using various machine learning algorithms.

1.3 OBJECTIVE

- Build a Jupyter notebook in Anaconda, import data, and view numbers loaded obsessed by the notebook.
- Practice Pandas to clean and formulate data.
- Use scikit-learn to create the machine learning exemplary.
- Use Matplotlib to see the model's performance.

Chapter -2

Jupyter notebooks are highly collaborative, and since they can take in executable enigma, they provide the seamless platform for manipulating data and edifice predictive models from it.

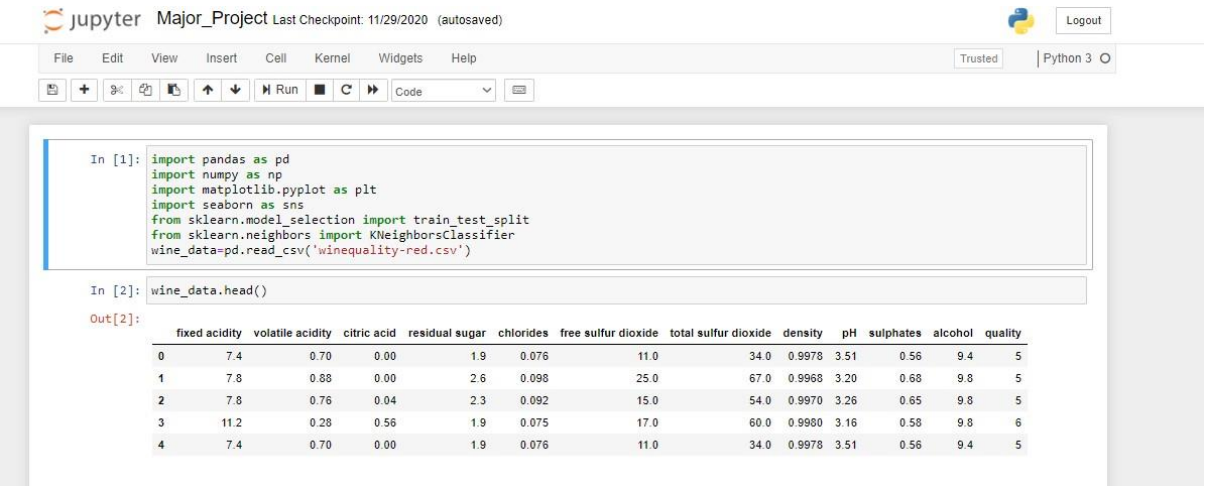
1. Firstly we download the dataset from the Kaggle.

2. In the notebook's following cell, enter the following Python code to load **winequality-red.csv**, craft a Pandas DataFrame from it, and ceremony the first five commotions.

```
In [1]: 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
        from sklearn.neighbors import KNeighborsClassifier
        wine_data=pd.read_csv('winequality-red.csv')
```

```
In [2]: wine_data.head()
```

3.Connect the **Run** button to execute the code. Sanction that the output remind us of the output below.



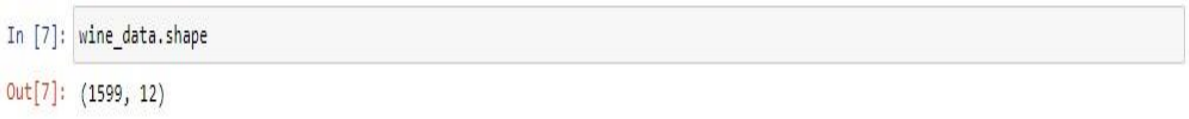
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

The **Data Frame** that we formed contains information of all the contents percentages that are present in red wine and the wine quality as well. It has more than 1000 rows and 12 columns. (The output says "5 rows" because Numbers Frame's head job only returns the first five rows.) Each row embodies the amount of content available in the wine as well as it's quality as well .We'll mine at the data more closely a bit later in this segment.

Chapter-3

A Data frame is a two-dimensional characterized data structure. The columns in a Files Frame can be of changed types, just like columns in a binge sheet or catalogue table. It is the most commonly used object in Pandas. In this exercise, we will observe the Data Frame and the data inside it more thoroughly.

1. One of the first possessions you archetypally want to know about a dataset is how many dins it contains. To get a calculation, type the resulting statement into an bare cell at the end of the notebook and run it:



Column	**Explanation**
Fixed acidity	Percentage of Fixed acidity in wine
Volatile acidity	Percentage of Volatile acidity in wine
Citric acid	Percentage citric acid in wine
Residual sugar	Percentage of residual sugar in wine
chlorides	Percentage of chlorides in wine
Free sulphur dioxide	Percentage of Free sulphur dioxide in wine

Total sulphur dioxide	Percentage of Total sulphur dioxide in wine
Column	**Explanation**
density	Percentage of Density in wine
pH	Percentage of pH in wine
sulphates	Percentage of sulphates in wine
alcohol	Percentage of alcohol in wine
quality	Quality of Wine

2. Yield a flash to survey the 12 columns in the dataset. Here is a ample list of the columns in the dataset.

X

The dataset takes in an even dispersal of quantities of various substances used in making of a particular wine and it's quality. The substances used in the dataset are often commonly measured in making of a particular wine and after the wine has been made it's quality is checked and accordingly scored.

One of the most central aspects of fixing a dataset for practise in apparatus learning is decide on the "feature" columns that are significant to the outcome we are trying to predict while filtering out columns that do not affect the outcome, could bias it in a negative way, or might produce multicollinearity. Another important task is to exclude missing values, either by accordingly scoring them or by filling them with the average value of that column. In this exercise, we will check for missing value rows/columns.

1. One of the first things data scientists typically look for in a dataset is missing values. There's an easy way to check for missing values in

Pandas. To demonstrate, execute the following code in a cell at the end of the notebook:

```
In [3]: wine_data.isna().sum().any()
Out[3]: False
```

2. The next step is to find out where the missing values are. To do so, execute the following code:

```
Python

df.isnull().sum()
```

Confirm that we see the following output listing no count of missing values in each column:

```
In [4]: wine_data.isna().sum()
Out[4]: fixed acidity           0
        volatile acidity        0
        citric acid             0
        residual sugar          0
        chlorides               0
        free sulfur dioxide     0
        total sulfur dioxide    0
        density                 0
        pH                      0
        sulphates               0
        alcohol                 0
        quality                 0
        dtype: int64
```

The dataset is now "clean" in the sense that missing values have been replaced and the list of columns has been narrowed to those most relevant to the model.

3. Univariate and Bivariate Analysis is carried out on the dataset to discover patterns. The below figures show univariate analysis using histogram and boxplot for ‘fixed acidity’ and ‘quality’ columns, respectively.

```
In [9]: wine_data.plot(kind='hist',y='fixed acidity')
        plt.show()
```

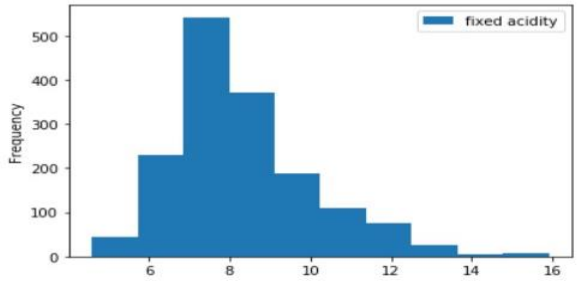


Fig1:-Histogram for one variable

```
In [17]: sns.boxplot(y='quality',data=wine_data)
         plt.show()
```

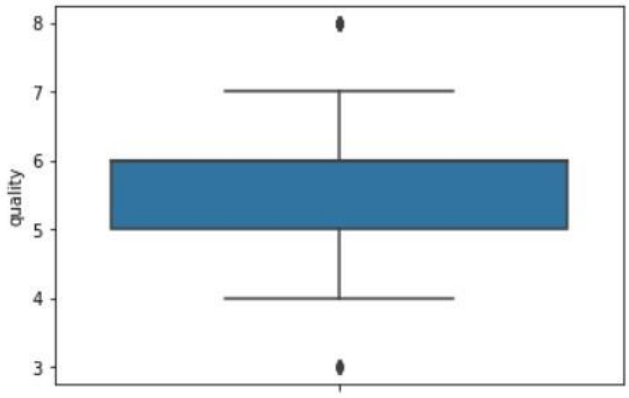


Fig2: -Box plot for one variable

The below figures show bivariate analysis using scatter plot for ‘fixed acidity’ and ‘quality’ columns which reflects uniformly spread data.

```
In [28]: wine_data.plot(kind='scatter',x='fixed acidity',y='quality')
         plt.show()
```

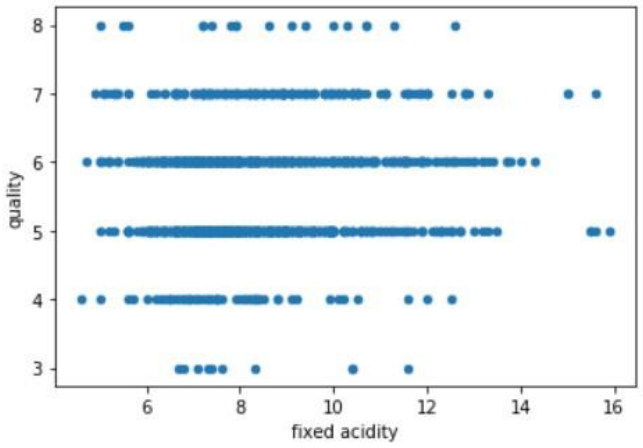


Fig3: -Scatter plot for two variables

XIII

A heat map is extremely powerful way to visualize relationships between variables in high dimensional space. For example, in this case a correlation matrix with heat map colouring is shown below. A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable in the table is correlated with each of the other values in the table. This allows us to see which pairs have the highest correlation.

In [34]:
```
corr=wine_data.corr()
plt.subplots(figsize=(15,10))
sns.heatmap(corr,xticklabels=corr.columns,yticklabels=corr.columns,annot=True)
plt.show()
```

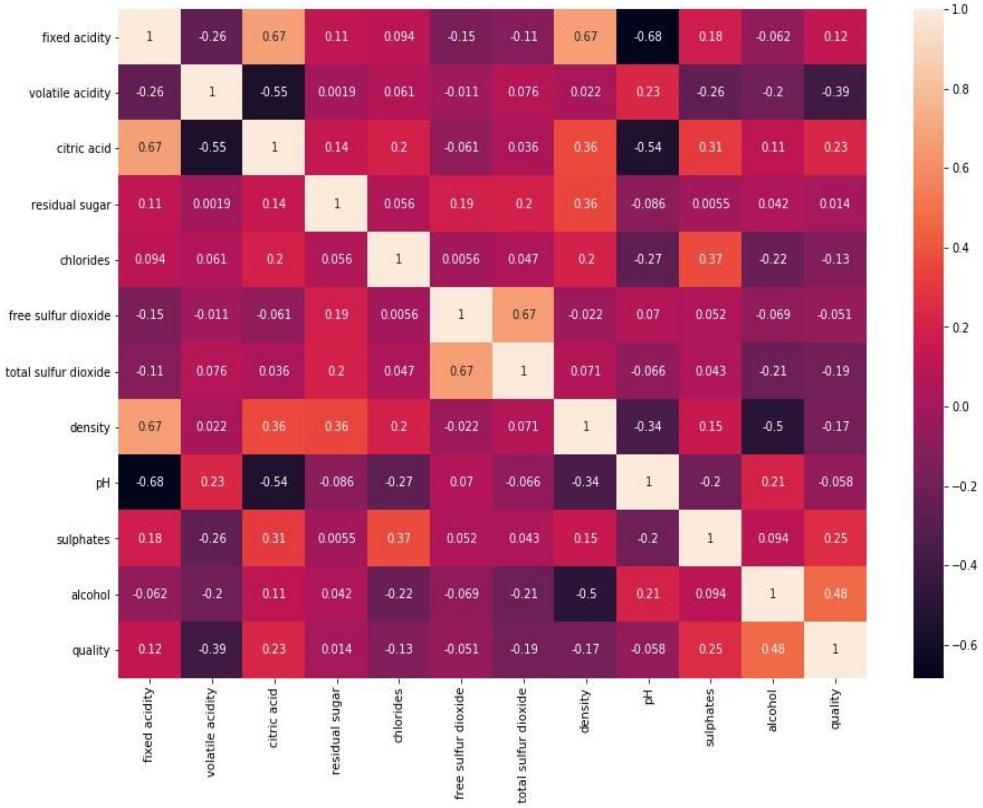


Fig4: - Heat Map

Chapter-4

To fashion a machine learning model, we want two datasets: one for training and one for testing. In practice, we often have only one dataset, so we split it into two. In this exercise, we will perform an 70-30 split on the dataframe we prepared in the previous lab such that it can used to train a machine learning model. We will also isolate the dataframe obsessed by feature columns and label supports. The former has the columns used as input to the wine(for example, the fixed acidity, alcohol, sulphates, etc.), while the latter contains the stake that the model will try to predict — in this case, the quality column, which indicates whether a flight will arrive on time.

1. In a new cell at the end of the notebook, enter and execute the following statements:

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
```

The first statement imports scikit-learn's train_test_split helper function. The second line uses the function to split the DataFrame into a physical activity set having 70% of the unusual data, and a test set enclosing the left over 30%.

2. Now use this command to show the numeral of rows and columns in the DataFrame comprehending the feature columns used for tough:

```
Python

test_x.shape
```

How do the two outputs differ, and why?

There are various types of machine learning copies. One of the most public is the KNN model, stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

One of the doles of using scikit-learn is that we don't have to build these reproductions — or implement the algorithms that they use — by hand. Scikitlearn includes a variety of classes for instigating collective machine learning models. One of them is Decision Tree classifier, which organizes a series of test questions and conditions in a tree structure.

1. Execute the following code in a new cell to create a Decision tree classifier object and train it by calling the fit method.

```
In [247]: clf=DecisionTreeClassifier()
          clf.fit(X,Y)

Out[247]: DecisionTreeClassifier()
```

2. Now call the predict method to test the model using the values in Test, followed by the score method to determine the mean accuracy of the model:

```
In [268]: Y_pred=clf.predict(X_train)

In [269]: from sklearn.metrics import accuracy_score
          print(accuracy_score(Y_train,Y_pred))

          0.9991063449508489
```

The accuracy is 99%, which seems good on the surface.

In the real world, a trained data scientist would look for ways to make the model even more accurate. Among other things, they would try different algorithms and take steps to *tune* the chosen algorithm to find the optimum

combination of parameters. Another likely step would be to expand the dataset to millions of rows rather than a few. But for our purposes, the model is fine as-is.

CONCLUSION

Machine Learning is a technique of training machines to perform the activities a human brain can do, albeit bit faster and better than an average human-being. Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an ecommerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them. Machine Learning algorithms are good at handling data that are multidimensional and multi-variety, and they can do this in dynamic or uncertain environments.

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated. ML needs enough time to let the algorithms learn and develop enough to fulfil their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer

power for you. Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

Machine Learning can be incredibly powerful when used in the right ways and in the right places