Project Report

GitHub URL

https://github.com/ankitbahl85/UCDPA_Ankit-Bahl.git

Abstract

To build a model to predict the quality of wine based on the characteristics.

For the purpose of this exercise, there are limited characteristics and dataset to create a model and test the most suitable prediction model.

Introduction

The dataset has records of various qualities of wine with varying characteristics. The intention of this project is to use python programming language and predict the quality of wine based on the characteristics.

Dataset

Source: Kaggle

This dataset contains various types of wines. This describes the amount of various chemicals present in wine and their effect on its quality. This data frame contains the following columns:

Input variables (based on physicochemical tests):
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
12 - Quality (score between 0 and 10)

Implementation Process

Importing data & python classes

- Import the libraries to utilize data frames efficiently
- Import the .csv file of dataset downloaded from Kaggle.

```
data = pd.read_csv ("c:/Users/ADMIN/Project/WineQT.csv")
```

Checking for table structure

```
data.head()
```

5]:

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

Exploratory Data Analysis

- Checking the existing structure of the data to ensure efficient application of operations
- Validate redundancies and remove duplicates columns / values

```
Removing duplicates
```

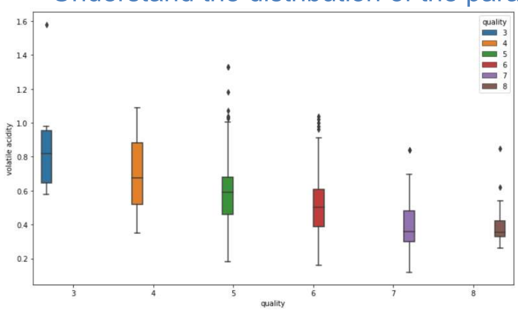
In [13]:

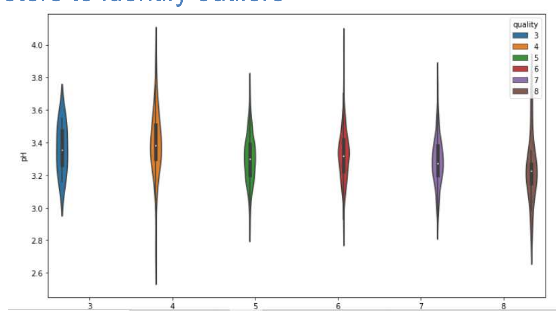
```
duplicate_rows = data[data.duplicated()]
print("# of duplicate rows:", duplicate_rows.shape[0])
```

```
# of duplicate rows: 0
```

Visual representation of the dataset

- Understand the distribution of the parameters to identify outliers





- Remove outliers and smoothen the charts by developing iterative functions

```
def outlier_treating(data_copy,variable):
    data=data_copy.copy()
    def outlier_detector(data_copy):
        outliers=[]
        q1=np.percentile(data_copy,25)
        q3=np.percentile(data_copy,75)
        Range=q3-q1
        lb=q1-(Range*1.5)
        ub=q3+(Range*1.5)
        for i,j in enumerate(data_copy):
            if(j<lb or j>ub):
                outliers.append(i)
        return outliers
    for i in variable:
        outlier_variable=outlier_detector(data[i])
        data.loc[outlier_variable,i]=np.median(data[i])
    return data
```

```
variable=list(data.select_dtypes(include=['float64']).columns)
```

```
data=outlier_treating(data,variable)
```

- Check to see if there is any correlation between the characteristics

- Synthetic oversampling - In case the training data is biased in distribution

```
! pip install imblearn
```

```
Requirement already satisfied: imblearn in c:\users\admin\anaconda3\lib\site-packages (0.0)
Requirement already satisfied: imbalanced-learn in c:\users\admin\anaconda3\lib\site-packages (from imblearn) (0.9.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblearn)
(1.7.1)
Requirement already satisfied: joblib>=1.0.0 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblear
n) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->i
mblearn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblear
n) (1.20.3)
Requirement already satisfied: scikit-learn>=1.1.0 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->im
blearn) (1.1.2)
```

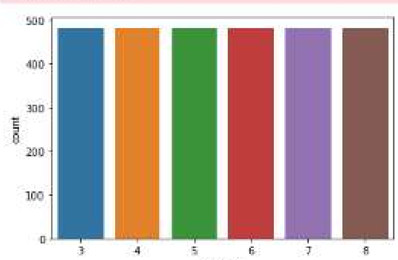
```
from imblearn.over_sampling import SMOTE
```

```
sm = SMOTE(sampling_strategy='auto', random_state=42)
```

```
X,y=sm.fit_resample(X,y)
```

```
sns.countplot(y)
plt.show()
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Machine learning

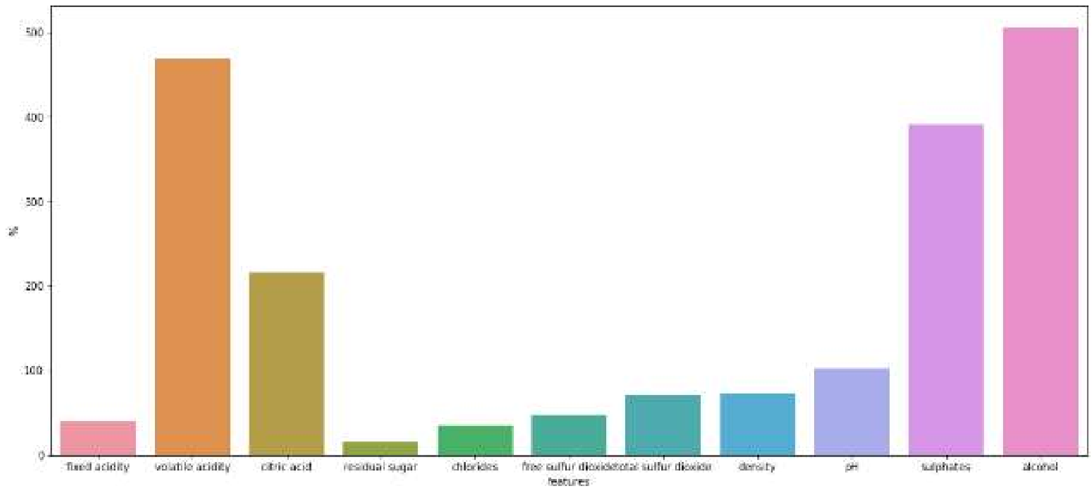
- Identifying the characteristics that are most crucial for prediction. These parameters influence quality the most

```
for i,j in enumerate(X.columns):
    print(f'{j} : {feature_contribution[i]:.2f}%')
plt.figure(figsize=(18,8))

sns.barplot(x=X.columns,y=fs.scores_)

#plt.title("test")
plt.xlabel("features")
plt.ylabel("%")
plt.show()
```

```
fixed acidity : 2.06%
volatile acidity : 23.82%
citric acid : 10.97%
residual sugar : 0.78%
chlorides : 1.80%
free sulfur dioxide : 2.41%
total sulfur dioxide : 3.61%
density : 3.73%
pH : 5.22%
sulphates : 19.89%
alcohol : 25.72%
```



- Utilize different models to predict the quality of wine (Random Forest, SVC, KNN, Decision tree)

```
classifier=DecisionTreeClassifier(criterion = 'entropy', random_state = 1)
classifier.fit(X_training,y_training)
y_predict=classifier.predict(X_testing)
print(f"Model Accuracy : {accuracy_score(y_predict,y_testing)*100:.2f}%")
print(f"Model F1-Score : {f1_score(y_predict,y_testing,average='weighted')*100:.2f}%")
accuracies = cross_val_score(estimator = classifier, X = X_training, y = y_training, cv = 5)
print("Cross Val Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Cross Val Standard Deviation: {:.2f} %".format(accuracies.std()*100))
print(classification_report(y_predict,y_testing,zero_division=1))
model_comparison['Decision Tree']=[accuracy_score(y_predict,y_testing),f1_score(y_predict,y_testing,average='weighted'),(acc
```

```
Model Accuracy : 76.38%
Model F1-Score : 76.86%
Cross Val Accuracy: 74.24 %
Cross Val Standard Deviation: 0.65 %
              precision    recall  f1-score   support

           3       0.99      0.94      0.96       102
           4       0.86      0.81      0.83       103
           5       0.57      0.71      0.63        78
           6       0.49      0.52      0.51        93
           7       0.73      0.70      0.72       101
           8       0.94      0.87      0.90       103

    accuracy                           0.76       580
   macro avg       0.76      0.76      0.76       580
weighted avg       0.78      0.76      0.77       580
```

Prediction

- Compare the prediction models for accuracy

```
Model_com_df=pd.DataFrame(model_comparison).T
Model_com_df.columns=['Model Accuracy','Model F1-Score','CV Accuracy','CV std']
Model_com_df=Model_com_df.sort_values(by='Model F1-Score',ascending=False)
Model_com_df.style.format("{:.2%}").background_gradient(cmap='Blues')
```

	Model Accuracy	Model F1-Score	CV Accuracy	CV std
Random Forest	79.83%	80.47%	79.68%	2.20%
KNN	75.17%	77.39%	74.24%	1.16%
Decision Tree	76.38%	76.86%	74.24%	0.65%
Support Vector Classifier	74.31%	75.28%	72.86%	1.46%

Hyper parameter tuning – Yet to perform

Results

We successfully imported the dataset from Kaggle and inspected it in the exploratory data analysis phase. We validated the redundancies and duplications. We then removed the column that seemed redundant and confirmed there were no null values.

In the next phase, we identified the correlation and dependencies of the quality of wine on multiple characteristics. To check the various combinations of characteristics, we plot these as pairs to understand if there is any dependency or relation among the characteristics. We created multiple visuals to understand the dependencies and the impact of each characteristic on the quality of wine.

As a part of prediction, created a training and a testing dataset. In order to avoid any biased data, we performed synthetic smoothening of the sample. Created a model based on characteristics that impact the most, to predict the quality of wine.

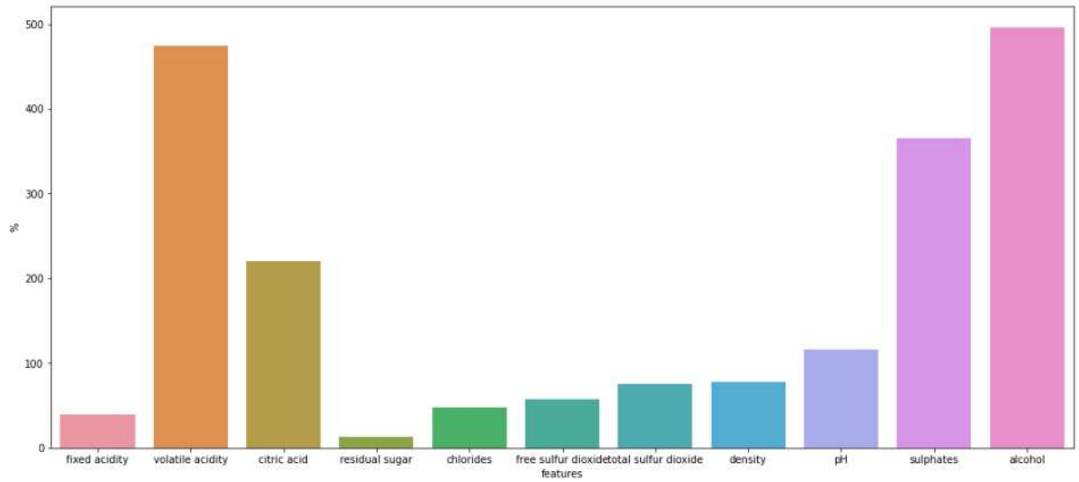
However, our model has a gap of about 20%. Part of this can be explained by the fact that these are not the only criteria to predict the quality of wine. Some of the additional characteristics can be:

- Others relevant physicals features like visual appearance (opacity) and flow (viscosity)
- characteristics like variety of grapes / region
- Amount of tannins that add bitterness to a wine

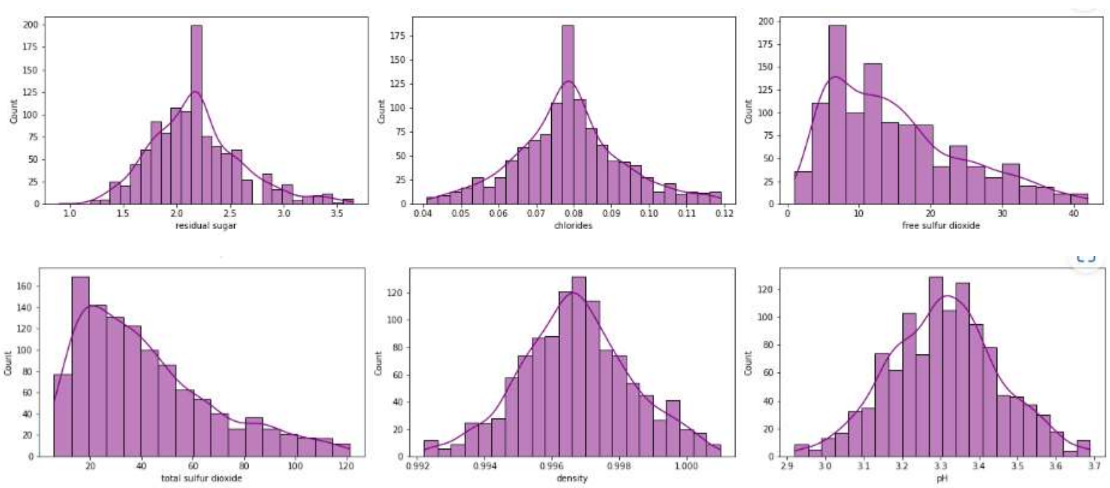
Insights

- Quality of wine depends upon volatile acidity, citric acid, chlorides, total sulfur dioxide, density, pH, sulphate & alcohol the most

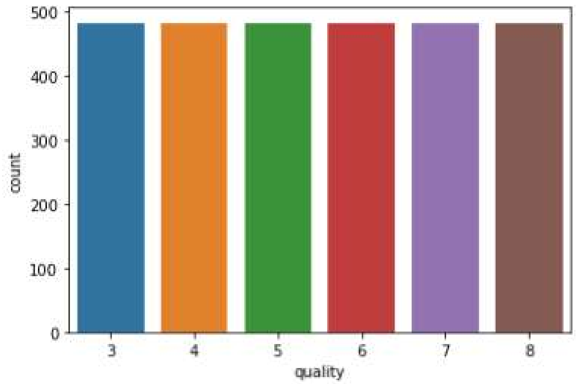
```
fixed acidity : 2.01%
volatile acidity : 23.89%
citric acid : 11.12%
residual sugar : 0.68%
chlorides : 2.42%
free sulfur dioxide : 2.91%
total sulfur dioxide : 3.82%
density : 3.92%
pH : 5.86%
sulphates : 18.38%
alcohol : 25.00%
```



- There is a negative correlation between volatile acidity, chlorides and quality of wine
- The outliers in the characteristics were restricting the prediction scores. Once we are able to limit (or remove the outliers), we get a better prediction



- Synthetic sampling is required to get a homogeneous sample for prediction. The prediction % increased by 10% by smoothening the sample



- The best prediction is using the Random forrest method with an accuracy of 80%

	Model Accuracy	Model F1-Score	CV Accuracy	CV std
Random Forest	80.34%	80.86%	79.72%	0.92%
KNN	74.14%	75.85%	74.85%	0.79%
Decision Tree	74.31%	74.39%	74.55%	1.15%
Support Vector Classifier	72.59%	73.41%	73.86%	1.56%

- Further accuracy can be attained with additional data or by hyper parameter tuning