ADTA 5340

PREDICTING THE WINE QUALITY USING THE CHEMICAL PROPERTIES OF WINE

Group 6: Great Eagles

Maneesha Shaik – 1660929

Pavan Sai Bopparaju – 11654839

Sharathchandra Reddy Yedavelli – 11641205

Sowmya Madala – 11754948

Venugopal Reddy Vennapusa – 11640661

Contents:

Introduction:

We work for Winery wines, a wine making company. Predictive modeling techniques that hire machine analyzing algorithms to predict the notable of wine primarily based on its chemical composition were created due to this studies. These modes are crafted from sizeable statistics gadgets that encompass a extensive variety of wines every of that is surprisingly described with recognize to their chemical compositions and sensory houses.

Abstract:

The quality of wine is affected by its chemical composition. In recent years, the use of machine learning algorithms and chemical analysis techniques to predict wine quality based on its chemical composition has become more popular. This research investigates how far component data such as acidity sugar content levels of alcohol and violet compounds can be employed to estimate wine quality through predictive modeling methods. Taste mouthfeel, and smell are some of the attributes which subjective evaluations from large datasets together with machine learning algorithms that consider the chemical composition identify as being correlated with or indicative of good or bad wines.it is important to note that apart from the chemical make up there are many other factors involved in wine production that affect its overall quality. These may include but not limited to the winemaking process itself, surrounding environmental conditions is one among others.

Team members Roles:

Sharathchandra Reddy Yedavelli: Dataset collection, Building ML algorithms, evaluation of ML models and Final report.

Pavan Sai Bopparaju: Data Preprocessing, dataset collection and presentation.

Venugopal Reddy Vennapusa: Exploratory data analysis and Presentation.

Sowmya Madala: Data Preprocessing and final report

Maneesha Shaik: Exploratory data analysis and presentation.

Executive Summary:

The measurement of wine quality is a complicated and subjective concept that is affected by several variables notably chemical composition. Though wine's chemical composition has been shown to possess an important impact on its sensory characteristics it remains impossible to predict wine quality just in chemical data.

The accurate assessment of the wine's quality according to chemical characteristics involves several important challenges despite advancements in analytical instruments and information analysis processes. The wide range and complexity of chemical components present in wine the subjective nature of sensory evolution and the effect of non-chemical factors like terroir and winemaking techniques overall constitute a few of these challenges.

Additionally, the natural unpredictability observed in wine manufacturing procedures and the complex relationships between chemical composition and sensory qualities are not addressed by present predictive modelling methods.

This is done by using a Support Vector Machine and Random Forest Regression models. First, we have collected the data and then data preprocessing was done by checking for missing values in the dataset, merging two datasets white wine and Red wine into 1 by adding Type feature for other steps, and converting Type feature into categorical feature as '1-red wine' and '0-white wine'. We have done Exploratory data analysis on the data to understand the features more. From EDA we found that we have more red wine instances than white wine.

Modelling will be done to capture the underlying patterns and relationships within the dataset. By modelling data we can make accurate predictions or classifications. The productivity of the step depends on the selection of a right algorithm and the building of a model that properly describes these patterns. Algorithm selection, model training, assessment Matrics, cross validation and model interpretability are the most important factors to consider. Which results in a strong model with exact classification or prediction skills that is required for making decisions based on the data. Next, we build two Machine learning algorithms Random Forest Regressor and Support Vector Machine to predict the quality of wine. Then splitting into training and testing sets for better model performance and feature scaling is done to ensure that we have 0 mean and 1 standard deviation which is a very crucial step in support vector machine (SVM). Trained the Random Forest Regression and SVM model and then fit the regression model in trained data to establish the baseline performance. Visualized predicted vs actual values of the quality of wine using scatter plot.Finally, We evaluated random forest regression and support vector machine models for predicting wine quality. Below were the evaluation metrics used for evaluating the built models.

- Mean squared error (MSE)
- Mean absolute error (MAE)
- R-squared error (R2)
- Cross-validation result (negative MSE) with 5 folds.
- Explained variance score.

In order to forecast the quality of red and white wines we have used support vector machine(SVM) and random forest regression models. Lower values shows greater model accuracy. Metrics including mean squared error(MSE) and mean absolute error(MAE) evaluate the mean variation between actual and predicted values. Higher values shows greater predictive ability. R-squared score and explained variance score explain the percentage of variance in wine quality. Cross validation result model explain about the model performance on unknown data that contributes the verification of their durability. By this, we can find underlying patterns and findings in the dataset aiming to most suitable regression model for wine quality.

Data Generation and Collection:

Dataset is collected from UCI machine learning repository. The dataset contains 6497 instances with two types of wine types and 13 features. Each instance represent different observation of corresponding wine sample.

The Features in dataset were:

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
12. Quality (Target variable)

13. Type

Data Wrangling:

Data cleaning operations are performed to eliminate the irregular values, missing values and replacing null values with appropriate mean or median.

```
[69] # Converting the 'Type' column to binary (1-Red wine,0-white wine)
     wine_quality['Type'] = wine_quality['Type'].apply(lambda x: 1 if x=='Red' else 0)
```

Fig:1

The above code converts the Type column to the binary for the next steps.

```
[71] # Sorting the dataset by 'quality' column in descending order
     wine_quality.sort_values('quality',ascending=False)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Type
2475	6.9	0.360	0.34	4.2	0.018	57.0	119.0	0.98980	3.28	0.36	12.7	9	0
3204	7.1	0.260	0.49	2.2	0.032	31.0	113.0	0.99030	3.37	0.42	12.9	9	0
2426	7.4	0.240	0.36	2.0	0.031	27.0	139.0	0.99055	3.28	0.48	12.5	9	0
2373	9.1	0.270	0.45	10.6	0.035	28.0	124.0	0.99700	3.20	0.46	10.4	9	0
2419	6.6	0.360	0.29	1.6	0.021	24.0	85.0	0.98965	3.41	0.61	12.4	9	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3649	11.8	0.230	0.38	11.1	0.034	15.0	123.0	0.99970	2.93	0.55	9.7	3	0
1374	6.8	0.815	0.00	1.2	0.267	16.0	29.0	0.99471	3.32	0.51	9.8	3	1
4906	9.4	0.240	0.29	8.5	0.037	124.0	208.0	0.99395	2.90	0.38	11.0	3	0
1299	7.6	1.580	0.00	2.1	0.137	5.0	9.0	0.99476	3.50	0.40	10.9	3	1
1469	7.3	0.980	0.05	2.1	0.061	20.0	49.0	0.99705	3.31	0.55	9.7	3	1

6497 rows × 13 columns

Fig:2

The above figure displays the wine_quality dataset in descending order by “Type” Feature. This helps us to identify the properties of best quality wine.

```
[72] #Displaying summary statistics of the dataset
     wine_quality.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Type
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000
mean	7.215307	0.339666	0.318633	5.443235	0.056034	30.525319	115.744574	0.994697	3.218501	0.531268	10.491801	5.818378	0.246114
std	1.296434	0.164636	0.145318	4.757804	0.035034	17.749400	56.521855	0.002999	0.160787	0.148806	1.192712	0.873255	0.430779
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.000000	3.000000	0.000000
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000	9.500000	5.000000	0.000000
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000	10.300000	6.000000	0.000000
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000	11.300000	6.000000	0.000000
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000	14.900000	9.000000	1.000000

Fig:3

The above figure displays the descriptive statistics like mean,median, and standard deviation of the wine_quality dataset.

```
# Displaying information about dataset
wine_quality.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Fixed_Acidity         6497 non-null   float64
 1   Volatile_Acidity      6497 non-null   float64
 2   Citric_Acid           6497 non-null   float64
 3   Residual_Sugar        6497 non-null   float64
 4   Chlorides             6497 non-null   float64
 5   Free_Sulfur_Dioxide   6497 non-null   float64
 6   Total_Sulfur_Dioxide  6497 non-null   float64
 7   Density               6497 non-null   float64
 8   pH                    6497 non-null   float64
 9   Sulphates             6497 non-null   float64
 10  Alcohol               6497 non-null   float64
 11  Quality               6497 non-null   int64
 12  Type                  6497 non-null   int64
dtypes: float64(11), int64(2)
memory usage: 660.0 KB
```

Fig 4

The above code .info() displays the information like data type and count about the dataset like

```
[73] # Renaming the columns of the dataset
     wine_quality.columns=['Fixed_Acidity','Volatile_Acidity','Citric_Acid','Residual_Sugar','Chlorides','Free_Sulfur_Dioxide','Total_Sulfur_Dioxide','Density','pH','S
```

Fig:5

The above code helps to rename the features to the correct formats.

```
[76] # Checking for missing values in the dataset
     wine_quality.isnull().sum()

     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
     Type                    0
     dtype: int64
```

```
[77] # Checking for NaN values in the dataset
     wine_quality.isna().sum()

     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
     Type                    0
     dtype: int64
```

Fig:6

The above code checks for the missing values and Not a Number values in the dataset and displays the count of number of missing values and Not a Null values.

```
[79] # Displaying the shape of the dataset
     wine_quality.shape

     (6497, 13)
```

Fig:7

The above shape function displays the shape of the dataset like number of instances and features. wine_quality dataset has 6497 instances and 13 features.

```
wine_quality
```

	Fixed_Acidity	Volatile_Acidity	Citric_Acid	Residual_Sugar	Chlorides	Free_Sulfur_Dioxide	Total_Sulfur_Dioxide	Density	pH	Sulphates	Alcohol	Quality	Ty
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6	
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	
6492	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0.50	11.2	6	
6493	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0.46	9.6	5	
6494	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0.46	9.4	6	
6495	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0.38	12.8	7	
6496	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0.32	11.8	6	

6497 rows × 13 columns

Fig:8

The above figure displays the changes made on the dataset. ‘Type’ feature represents 1 for Red wine and 0 for White wine.

Exploratory Data Analysis:

EDA includes:

- We can find connections between chemical properties and quality of the wine.
- Representing the data visually using scatter plots, histograms, and box plots.
- These plots helps us to find the hidden patterns and trends for further model generation.

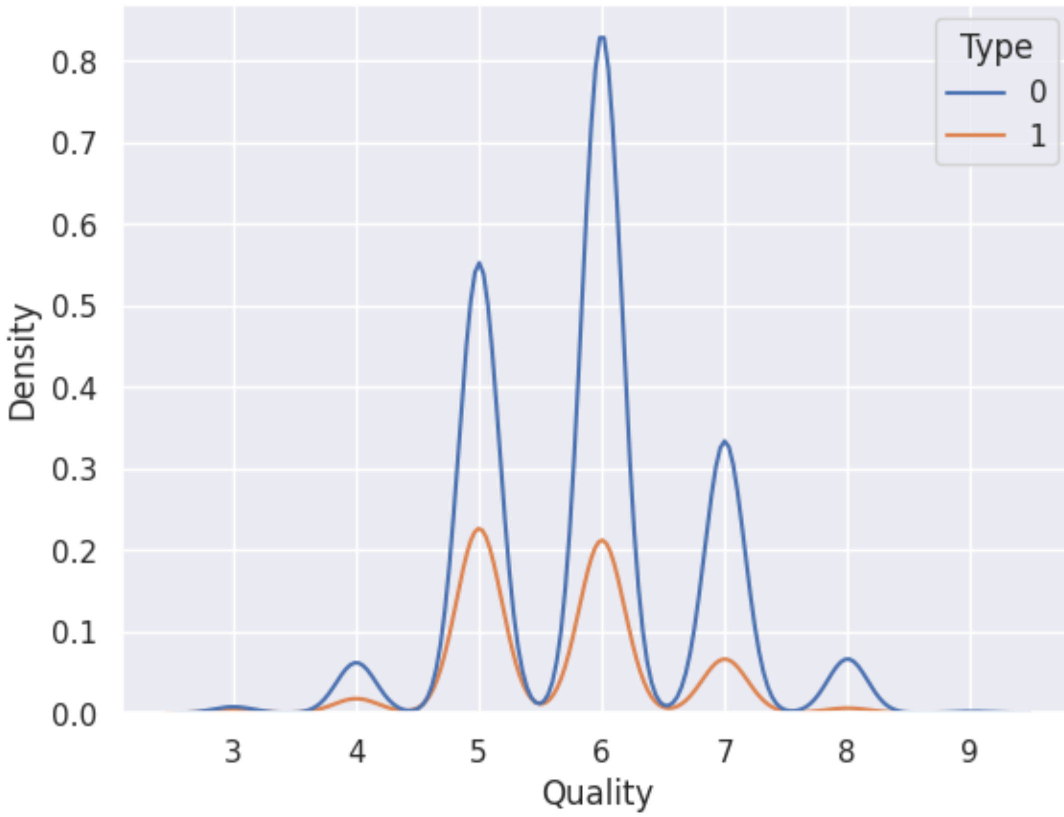

Fig:9

The above figure shows the density pot of distribution of the wine quality for each quality of wine by Type.

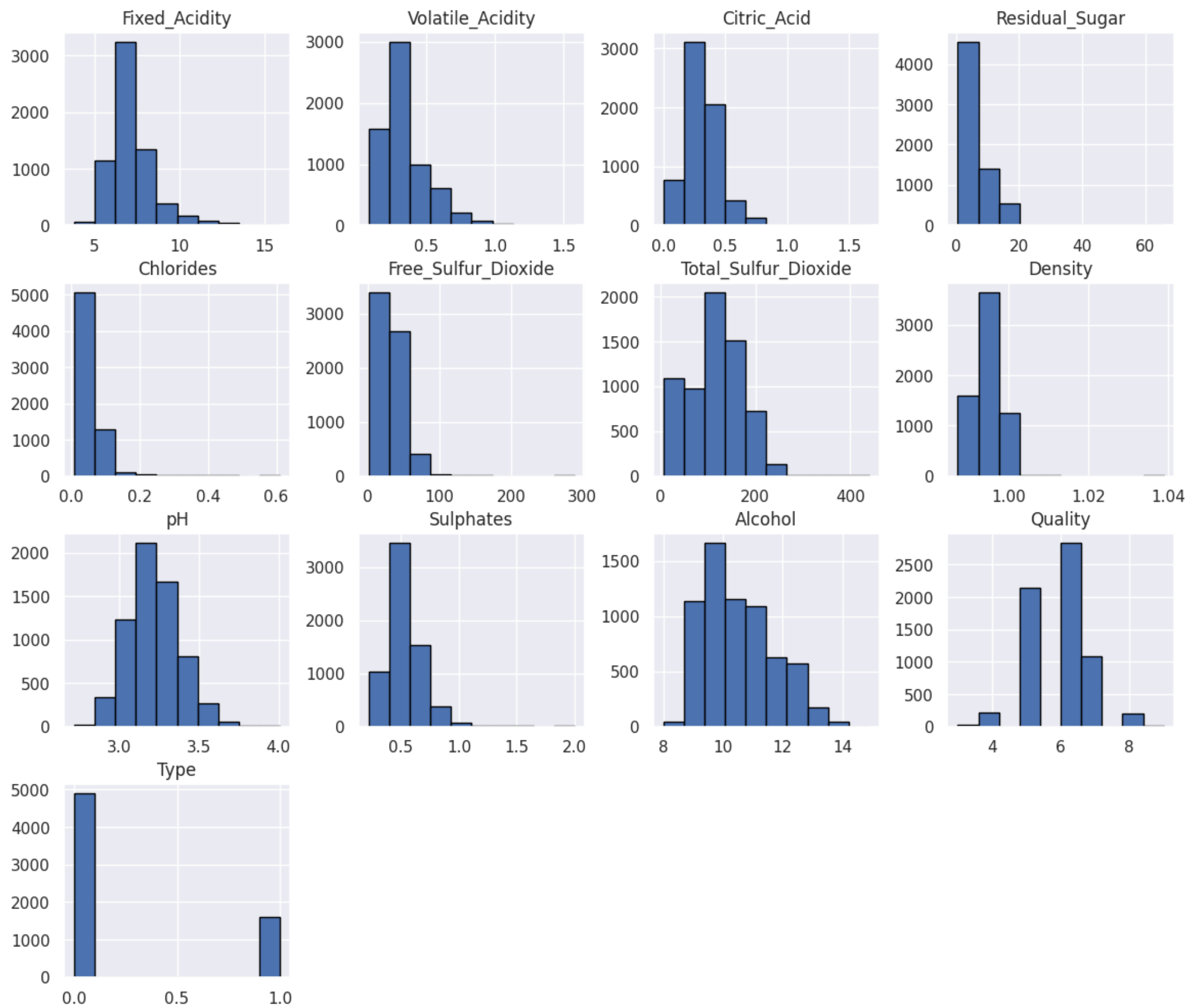

Fig:10

The above Generated histogram visualizes the frequency of each feature in the dataset. This histogram helps us to find the skewness and distribution of each feature in the data.

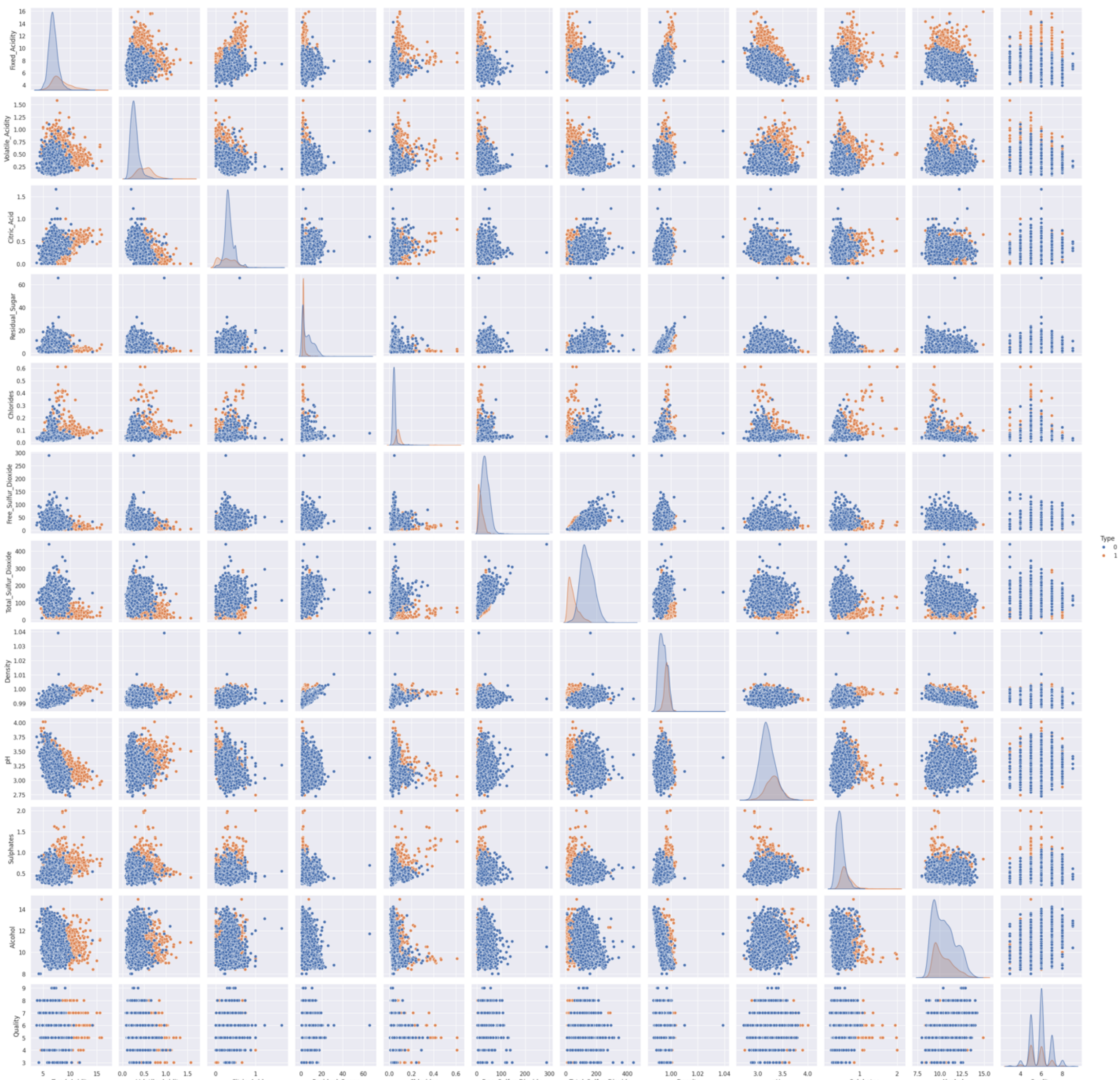

Fig:11

This pair plot demonstrates the diagonal histograms and scatterplots for each pair of numerical variables in the dataset. It helps for quick identification of outliners and trends.

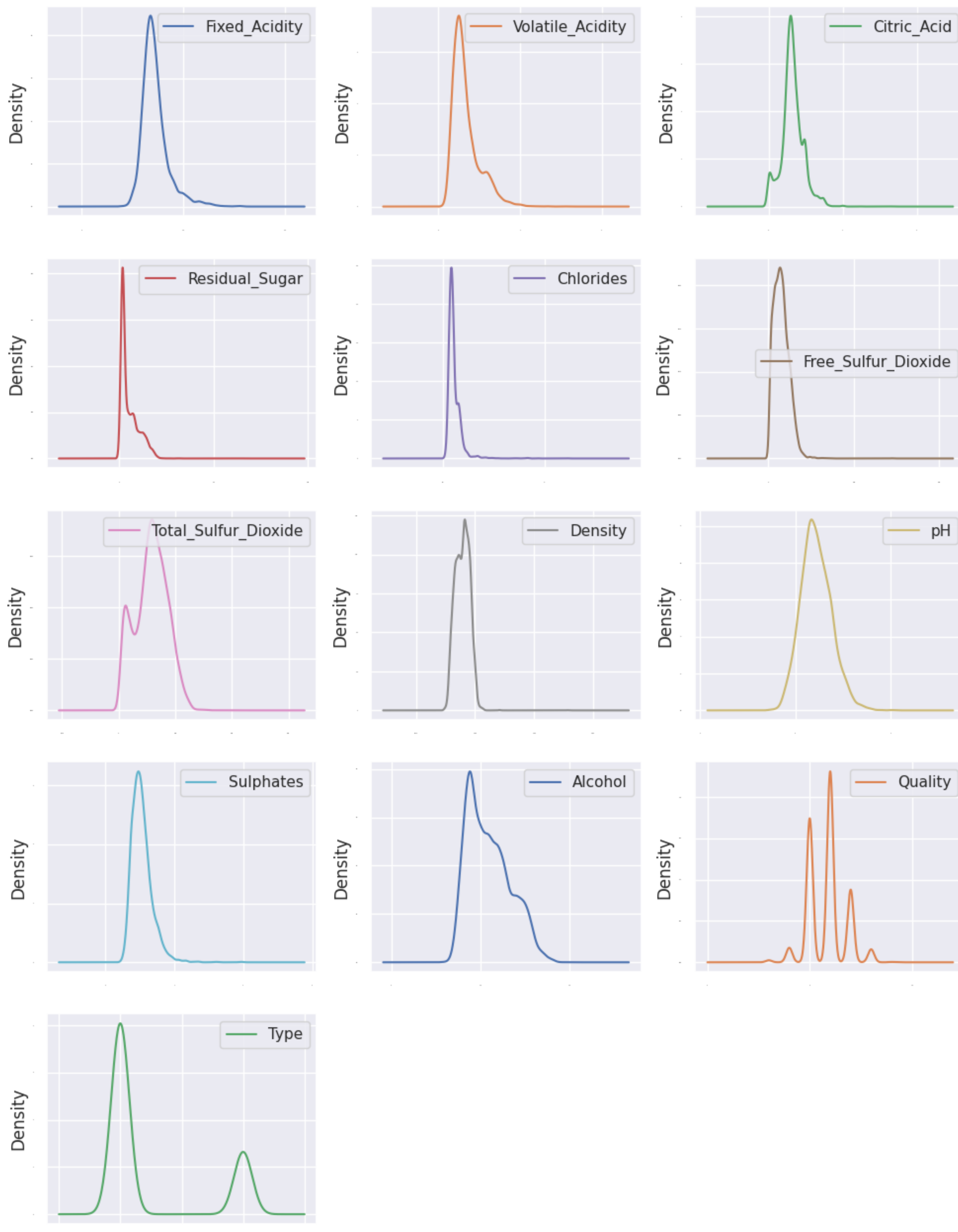
Fixed_Acidity
Volatile_Acidity
Citric_Acid
Residual_Sugar
Chlorides
Free_Sulfur_Dioxide
Total_Sulfur_Dioxide
Density
pH
Sulphates
Alcohol
Quality
Type
Density

Fig:12

The above figure illustrates the density plot with five rows and 3 columns produced for each feature in the dataset. The density distribution of each feature can be seen as subplots. Density plot reduces the noise and smooth out the distribution of values.

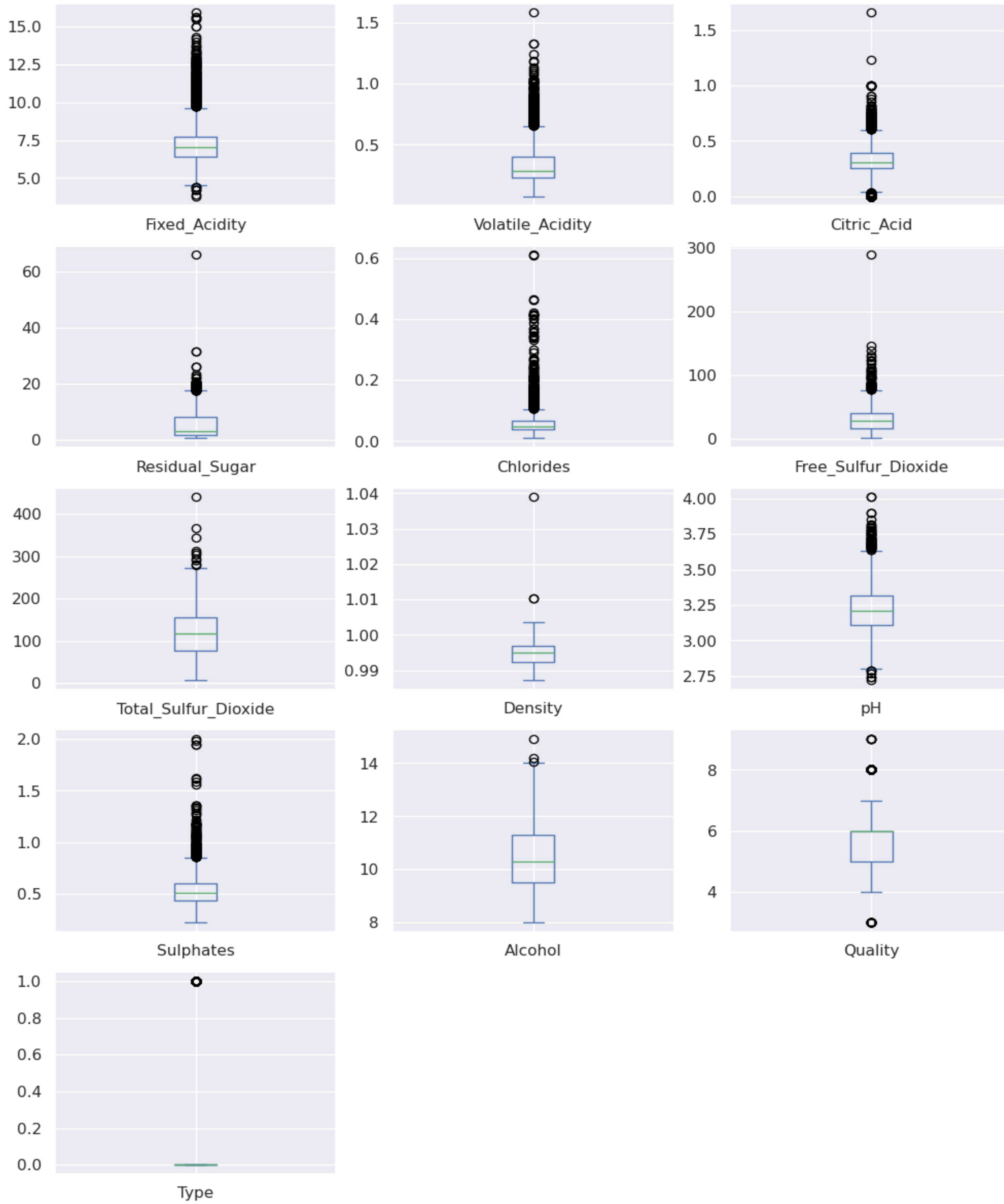

Fig: 13

The above figure shows the box plot for each feature which helps to find the outliners and distribution of values in each feature.

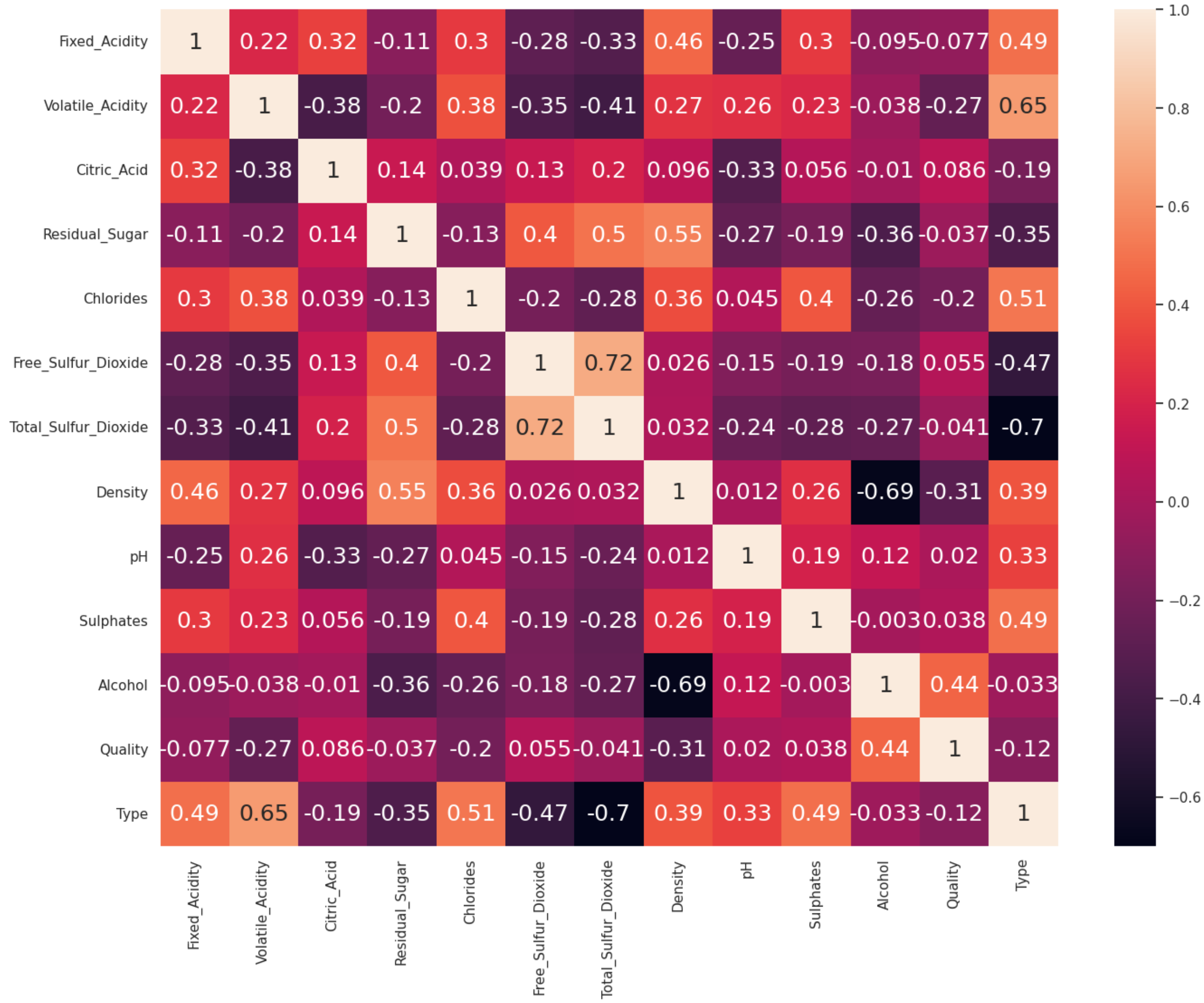

Fig:14

The above figure illustrates the heatmap of the dataset’s correlation matrix. It helps us to find the relation between two features.

Modeling Data:

In this section we build models to train in and identify the hidden patterns and relationships within the dataset. Our main goal is to build models which will accurately identify the patterns.

Random Forest Regression:

Random Forest runs by constructing decision trees based on the training sets and for creating a single model, multiple decision trees are combined and average prediction in the regression. It is a choice for wide range of applications across the domains.

Support Vector Machine:

SVM's are supervised learning models used for classification and regression problems. It is operated by determining the best hyperplane in the feature space. Kernel selection is crucial for SVM based on what we are using. Feature scaling is also crucial to ensure that all values have mean as 0 and standard deviation as 1.

```
[ ] # creating input and output dataframes for Red wine
    Rw_X=Red_wine.drop('Quality',axis=1)
    Rw_y=Red_wine['Quality']
    # Creating training and testing sets
    Rw_X_Train,Rw_X_Test,Rw_y_train,Rw_y_test=train_test_split(Rw_X,Rw_y,test_size=0.2,random_state=42)
```

```
[ ] # creating a dataframe with input features
    Ww_X=White_wine.drop('Quality',axis=1)
    # creating a dataframe with output featues
    Ww_y=White_wine['Quality']
    # Creating training and testing sets
    Ww_X_Train,Ww_X_Test,Ww_y_train,Ww_y_test=train_test_split(Ww_X,Ww_y,test_size=0.2,random_state=42)
```

Fig: 15

Creating input and output dataframes for Red_wine and White_wine and also creating training and testing sets for both Red and White wines.

1. **Red_wine Random forest regression model**

```
[138] #Initializing the Ramdom forest Model
      Rw_Rf_Model=RandomForestRegressor(n_estimators=100,max_features='sqrt',bootstrap=True,random_state=42)
      # Fit the random forest model to the 'Red_wine' training set
      Rw_Rf_Model.fit(Rw_X_Train,Rw_y_train)
```

```
▾                 RandomForestRegressor
RandomForestRegressor(max_features='sqrt', random_state=42)
```

Fig:16

Initialization Random forest regression model and fitting the model to the Red_wine training set.

```
# predict the quality of wine on the testing sets
Rw_y_Pred=Rw_Rf_Model.predict(Rw_X_Test)
```

Fig:17

Using the Random forest regression trained model on the testing set for predicting the quality of wine.

2. **Red_wine suport vector machine regression model**

```
[140] # Trainning and testing Support Vector Regression model for Red_wine
      Rw_svr_Model=SVR(kernel='linear')
      # Fit the svr model to training set
      Rw_svr_Model.fit(Rw_X_Train,Rw_y_train)
```

```
▾        SVR
SVR(kernel='linear')
```

Fig:18

Initializing a SVM regressor model and fitting the model on the Red_wine training set.

```
# Predict the quality for the testing set
Rw_y_Pred_svr=Rw_svr_Model.predict(Rw_X_Test)
```

Fig:19

Using the Red_wine SVM regression trained model on the testing set for predicting the quality of wine.

3. White_wine Random forest regression Model

```
[142] #Initializing the Ramdom forest Model with random state as 42 and number of estimators as 100
      Ww_Rf_Model=RandomForestRegressor(n_estimators=100,max_features='sqrt',bootstrap=True,random_state=42)
      # Fit the random forest model to the training set
      Ww_Rf_Model.fit(Ww_X_Train,Ww_y_train)

▾ RandomForestRegressor
RandomForestRegressor(max_features='sqrt', random_state=42)
```

Fig: 20

Initializing the Random forest regression model and fitting the model on the White_wine training set with number of estimators as 100 and random state as 42.

```
[143] # predict the quality of wine on the testing sets
      Ww_y_Pred=Ww_Rf_Model.predict(Ww_X_Test)
```

Fig:21

Using the White_wine Random forest regression trained model on the testing set for predicting the quality.

4. White_wine support vector machine regression model

```
[144] #Initializing the Support Vector Regressor Model
      Ww_svr_Model=SVR(kernel='linear')
      # Fit the svr model to the training set
      Ww_svr_Model.fit(Ww_X_Train,Ww_y_train)
```

```
▾        SVR
SVR(kernel='linear')
```

Fig:22

Initializing the SVM regression model and fitting the model on the White_wine training sets.

```
[145] # Predict the quality of the wine for the testing set
      Ww_y_Pred_svr=Ww_svr_Model.predict(Ww_X_Test)
```

Fig:23

Using the White_wine SVM regression trained model on the testing set of White_wine for predicting the quality of wine.

Interpreting Data:

```
# Plot feature importance
plt.figure(figsize=(10, 6))
plt.barh(feature_importance_df['Feature'], feature_importance_df['Importance'])
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance')
plt.show()
```

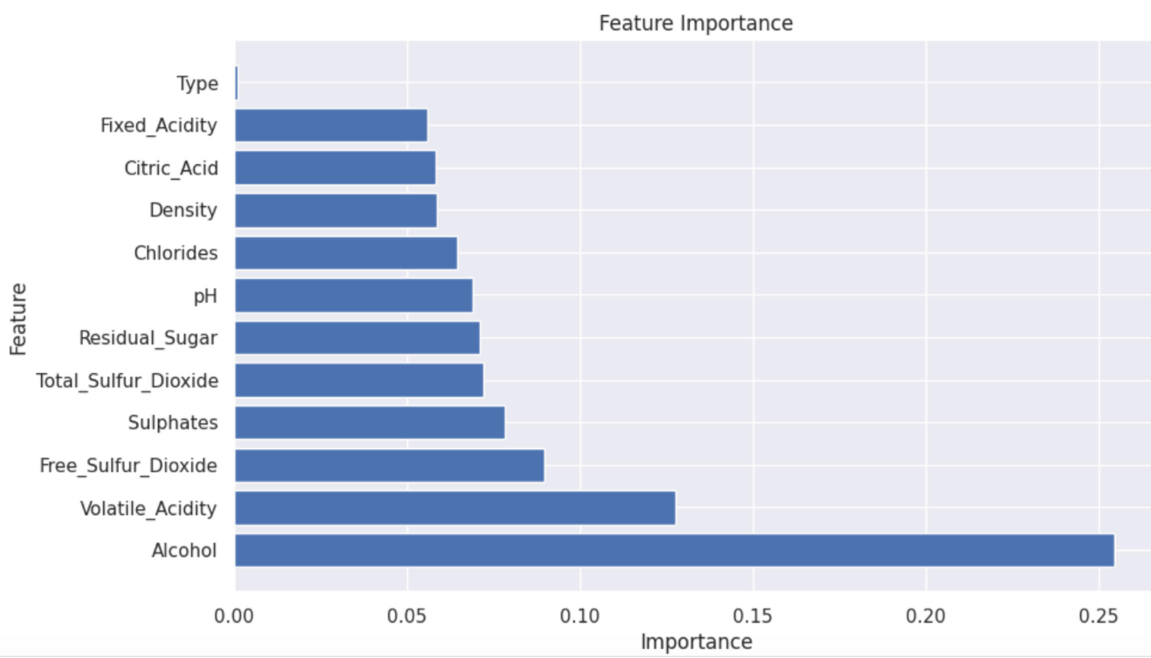

Fig:24

The feature importance of random forest regressor is to predict the most helpful feature for predicting the target variable. Here we used "plt.barh" to create a horizontal bar chart to visualize the feature importance. From the above visualization, Alcohol is having the highest score and it has more influence on the model predictions.

Evaluation Metrics:

```python
# Display the evaluation metrics for Random Forest Regressor and Support Vector Regressor Models for both Red and White wine.
print("Evaluation metrics for Red wine")
print("Random Forest Regression Model :")
print("MSE :",mean_squared_error(Rw_y_test,Rw_y_Pred))
print("MAE :",mean_absolute_error(Rw_y_test,Rw_y_Pred))
print("R2 score",r2_score(Rw_y_test,Rw_y_Pred))
Rw_Rf_cv_scores=cross_val_score(Rw_Rf_Model,Rw_X,Rw_y,cv=5,scoring='neg_mean_squared_error')
# display cross-validation scores
print("Cross-Validation Result (negative Mean Squared Error) :",-np.mean(Rw_Rf_cv_scores))
Rw_Rf_explained_variance=explained_variance_score(Rw_y_test,Rw_y_Pred)
print("Explained Variance Score :",Rw_Rf_explained_variance)

print("\nSupport Vector Regression Model :")
print("MSE :",mean_squared_error(Rw_y_test,Rw_y_Pred_svr))
print("MAE :",mean_absolute_error(Rw_y_test,Rw_y_Pred_svr))
print("R2 score",r2_score(Rw_y_test,Rw_y_Pred_svr))
Rw_svr_cv_scores=cross_val_score(Rw_svr_Model,Rw_X,Rw_y,cv=5,scoring='neg_mean_squared_error')
# display cross-validation scores
print("Cross-Validation Result (negative Mean Squared Error) :",-np.mean(Rw_svr_cv_scores))
Rw_svr_explained_variance=explained_variance_score(Rw_y_test,Rw_y_Pred_svr)
print("Explained Variance Score :",Rw_svr_explained_variance)
print("\nEvaluation metrics for White wine")
print("Random forest regression Model :")
print("MSE :",mean_squared_error(Ww_y_test,Ww_y_Pred))
print("MAE :",mean_absolute_error(Ww_y_test,Ww_y_Pred))
print("R2 score",r2_score(Ww_y_test,Ww_y_Pred))
Ww_Rf_cv_scores=cross_val_score(Ww_Rf_Model,Ww_X,Ww_y,cv=5,scoring='neg_mean_squared_error')
# display cross-validation scores
print("Cross-Validation Result (negative Mean Squared Error) :",-np.mean(Ww_Rf_cv_scores))
Ww_Rf_explained_variance=explained_variance_score(Ww_y_test,Ww_y_Pred)
print("Explained Variance Score :",Ww_Rf_explained_variance)

print("\nSupport Vector Regression Model :")
print("MSE :",mean_squared_error(Ww_y_test,Ww_y_Pred_svr))
print("MAE :",mean_absolute_error(Ww_y_test,Ww_y_Pred_svr))
print("R2 score",r2_score(Ww_y_test,Ww_y_Pred_svr))
Ww_svr_cv_scores=cross_val_score(Ww_svr_Model,Ww_X,Ww_y,cv=5,scoring='neg_mean_squared_error')
# display cross-validation scores
print("Cross-Validation Result (negative Mean Squared Error) :",-np.mean(Ww_svr_cv_scores))
Ww_svr_explained_variance=explained_variance_score(Ww_y_test,Ww_y_Pred_svr)
print("Explained Variance Score :",Ww_svr_explained_variance)
```

```
Evaluation metrics for Red wine
Random Forest Regression Model :
MSE : 0.29574312500000005
MAE : 0.4145
R2 score 0.5474514562381386
Cross-Validation Result (negative Mean Squared Error) : 0.4232687409874608
Explained Variance Score : 0.5475277813475994

Support Vector Regression Model :
MSE : 0.3966236876138224
MAE : 0.5006600426889813
R2 score 0.3930831959285792
Cross-Validation Result (negative Mean Squared Error) : 0.4379233282364595
Explained Variance Score : 0.400900979647679

Evaluation metrics for White wine
Random forest regression Model :
MSE : 0.33400867346938773
MAE : 0.4137448979591837
R2 score 0.5687278772364763
Cross-Validation Result (negative Mean Squared Error) : 0.5248287800963082
Explained Variance Score : 0.5689283118939936

Support Vector Regression Model :
MSE : 0.5733812879463094
MAE : 0.5836522702096193
R2 score 0.2596498688584149
Cross-Validation Result (negative Mean Squared Error) : 0.5881480362825262
Explained Variance Score : 0.26299097891857237
```

Fig: 25

The Above figure shows the evaluation metrics for all 4 models that we have built.

Comparision:

Red wine: The Random Forest regression model has lower Mean squared error (MSE) and lower Mean absolute error(MAE) and Having Higher R2 score andexplained variance score. The random Forest Regression Model performs better than the support vector machine. So Random Forest Regression Model is best suited for predicting the quality of Red wine.

White wine: The Random Forest Regression Models has the lower MSE and MAE values and higher r2 score than the support vector machine Regression model. So, Random Forest Regression Model is best suited for White_wine also.

From the above both Red_wine and White_wine evaluation metrics, Random Forest Regression model performs well.

```
# Predict the 'Quality' for certain values using the Random Forest Regressor Model for Red wines
print("\nPredicted quality for Red wine using Random forest regression model :")
Rw_Rf_Model.predict([[8.9,0.16,0.49,8.2,0.218,57.0,119.0,0.9898,3.28,0.36,12.7]])
```

```
Predicted quality for Red wine using Random forest regression model :
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestRegressor was fitted with feature r
  warnings.warn(
array([6.32])
```

```
# Predict the 'Quality' for certain values using the Random Forest Regressor Model for White wines
print("\nPredicted quality for White wine using Random forest regression model :")
Ww_Rf_Model.predict([[8.9,0.16,0.49,8.2,0.218,57.0,119.0,0.9898,3.28,0.36,12.7]])
```

```
Predicted quality for White wine using Random forest regression model :
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestRegressor was fitted with feature r
  warnings.warn(
array([6.72])
```

Fig:26

The above figure shows the predicted quality of the wine using Random Forest Regression models of both Red_wine and white_wine. Using this model we can produce the better quality wines.

Conclusion:

In this study, we implemented Random Forest Regression model and Support Vector Machine model which are trained on different chemical compositions of wine to predict the quality of wine. We had done exploratory data analysis and fount out that alcohol feature is influencing more on the algorithm and the dataset contains less instances of white wine than red wine. Random Forest model shows greater accuracy and ability to understand the hidden patterns and trends that present in the data.. The models demonstrate the potential of ML methods to guide the winery wine to allow stakeholders to make choices on the best to improve the quality of wine.

References:

UCI Machine Learning Repository. (n.d.-b). https://archive.ics.uci.edu/dataset/186/wine+quality

Dahal, K. R., Dahal, J. N., Banjade, H., & Gaire, S. (2021, January 1). *Prediction of Wine Quality Using Machine Learning Algorithms*. Open Journal of Statistics. https://doi.org/10.4236/ojs.2021.112015

ZyBooks. (n.d.-b). https://learn.zybooks.com/zybook/UNTADTA5340FantasiaSpring2024?selectedPanel=my-class-student