Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

======== RESTART: C:/Morphy/RedWineQuality/RedWineQuality\_BuildModels.py =======

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1599 entries, 0 to 1598

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 fixed acidity 1599 non-null float64

1 volatile acidity 1599 non-null float64

2 citric acid 1599 non-null float64

3 residual sugar 1599 non-null float64

4 chlorides 1599 non-null float64

5 free sulfur dioxide 1599 non-null float64

6 total sulfur dioxide 1599 non-null float64

7 density 1599 non-null float64

8 pH 1599 non-null float64

9 sulphates 1599 non-null float64

10 alcohol 1599 non-null float64

11 quality 1599 non-null int64

dtypes: float64(11), int64(1)

memory usage: 150.0 KB

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model Name | Model Type | Model Parameters | Accuracy | Cross-Validation Score | Accuracy on test data | Time |
| LR | Logistic Regression | Trained on whole data set | 60.913% | 59.659% | 0.675 | 150.12520170211792 |
| LR2 | Logistic Regression | With Train Test Split | 59.903% | 59.003% | 0.65 | 149.21641659736633 |
| KNN | K-Nearest Neighbors | Trained on whole data set | 100.000% | 64.795% | 0.99375 | 0.4687507152557373 |
| KNN2 | K-Nearest Neighbors | with Train Test Split | 100.000% | 63.515% | 0.7 | 0.3906261920928955 |
| DT | Decision Tree | Trained on whole data set | 79.800% | 61.288% | 0.85 | 0.2812478542327881 |
| DT2 | Decision Tree | with Train Test Split | 100.000% | 61.364% | 0.625 | 0.2812464237213135 |
| ~~RF~~ | ~~Random Forest~~ | ~~Trained on whole data set~~ | ~~69.168%~~ | ~~61.229%~~ | ~~0.76875~~ | ~~35.81570839881897~~ |
| RF | Random Forest | Trained on whole data set | 68.543% | 61.411% | 0.76875 | 3.84700345993042 |
| ~~RF~~ | ~~Random Forest~~ | ~~Using Train Test Split~~ | ~~69.771%~~ | ~~60.807%~~ | ~~0.70625~~ | ~~34.546037435531616~~ |
| RF | Random Forest | Using Train Test Split | 69.840% | 60.391% | 0.69375 | 3.6850035190582275 |
| NB | Naive Bayes | Trained on whole data set | 56.473% | 54.787% | 0.60625 | 0.14999794960021973 |
| NB2 | Naive Bayes | using Train Test Split | 55.525% | 53.162% | 0.58125 | 0.16000008583068848 |
| SVM | Supporting vector machine | Trained on whole data set | 51.032% | 50.406% | 0.60625 | 4.126319169998169 |
| SVM2 | Supporting vector machine | Using Train Test split | 49.896% | 49.544% | 0.60625 | 3.4702250957489014 |
| SV | Soft Voting | Trained on whole data set ; LR+NNN+DT; | 94.371% | 65.226% | 0.93125 | 160.2510666847229 |
| SV2 | Soft Voting | LR+NNN+DT; using Train Test Split | 93.398% | 64.070% | 0.71875 | 151.6797263622284 |
| HV | Hard Voting | Trained on whole data set | 86.304% | 65.040% | 0.875 | 152.99926829338074 |
| HV2 | Hard Voting | Using Train Test Split | 86.101% | 63.870% | 0.675 | 151.2754521369934 |
| Stack1 | Stacking | using KNN as Meta; Estimators = lr2, knn2, dtree2, hard\_vote2 | 95.761% | 66.784% | 0.69375 | 1596.1465604305267 |
| Stack2 | Stacking | using Logistic Regression as Meta; Estimators = knn, dtree, soft\_vote | 88.555% | 63.785% | 0.85 | 3168.730623483658 |
| Stack3 | Stacking | using Logistic Regression as Meta; estimators = knn2, dtree, soft\_vote2),hard\_vote2 | 95.066% | 65.884% | 0.6875 | 1725.27055478096 |