IBM Machine Learning Professional Certificate

WINE QUALITY PREDICTION

Supervised Machine Learning: Classification

PRESENTED BY :
Yuvraj Singh Sukhmani

CONTENT

OVERVIEW

The wine industry relies heavily on accurate quality assessment for market competitiveness. Leveraging AI, this project efficiently predicts wine quality on a 1 to 10 scale. The selected model showcases superior predictive abilities, streamlining the evaluation process and minimizing errors. AI's role is pivotal, enabling data analysis for identifying key quality indicators, offering real-time insights, and ensuring consistent production standards. By automating quality control, AI detects subtle variations, maintaining product integrity. This integration not only enhances production processes and resource optimization but also ensures superior wine quality, meeting the dynamic demands of discerning consumers.

IBM Machine Learning Professional Certificate

DATASET DESCRIPTION

It is related to red “vinho verde” wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests.

- **CHARACTERISTICS**
  Multivariate
- **SUBJECT AREA**
  Business
- **ASSOCIATED TASKS**
  Classification, Regression
- **FEATURE TYPE**
  Real
- **INSTANCES**
  4898
- **FEATURES**
  11

	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

FIXED ACIDITY

Most acids involved with wine or fixed or non-volatile

Continuous

VOLATILE ACIDITY

The amount of acetic acid in wine

Continuous

CITRIC ACID

Found in small quantities, can add 'freshness to wine

Continuous

	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

RESIDUAL SUGAR

The amount of sugar remaining after fermentation stops *Continuous*

CHLORIDES

The amount of salt in the wines *Continuous*

FREE SULFUR DIOXIDE

The free form of SO2 that exists in equilibrium *Continuous*

	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

TOTAL SULFUR DIOXIDE

The amount of free and bound forms of SO2; in low concentrations *Continuous*

DENSITY

The density of water is close to that of water depending on the % alcohol and sugar content *Continuous*

PH

It describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic) *Continuous*

	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

SULPHATES

A wine additive which can contribute to sulfur dioxide gas (SO2) levels — ***Continuous***

ALCOHOL

The percent alcohol content of the wine — ***Continuous***

QUALITY

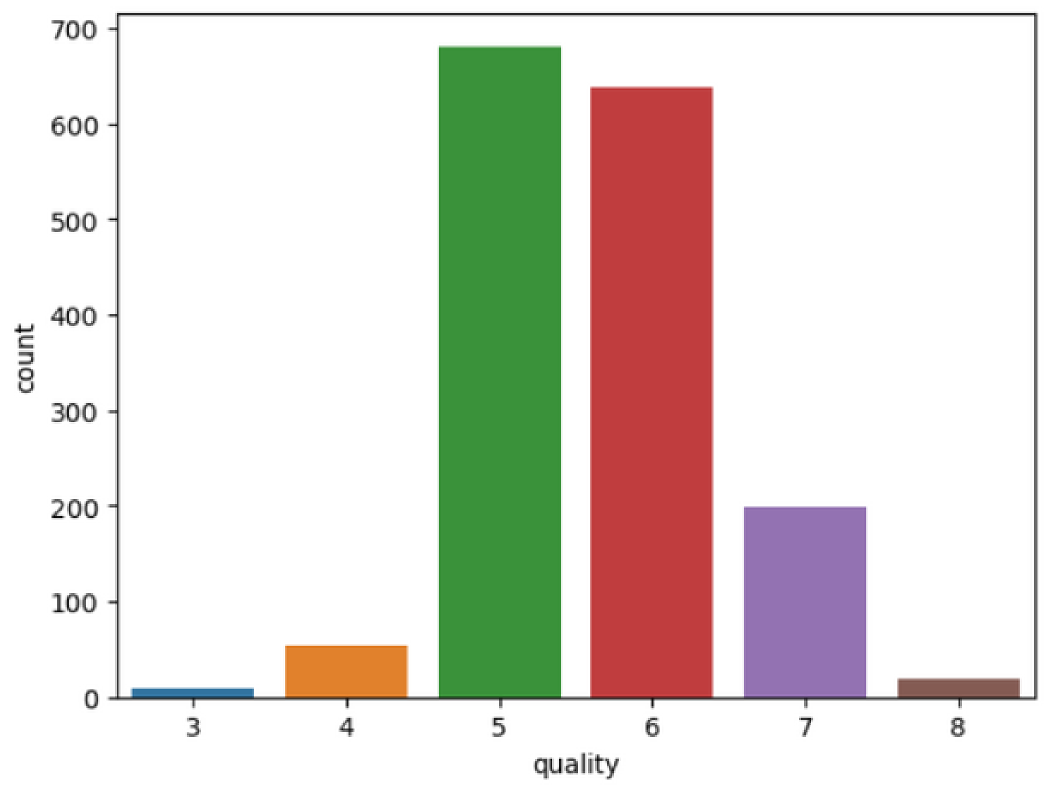
The output variable (based on sensory data, score between 0 and 10) — ***Integer***

ANALYSIS

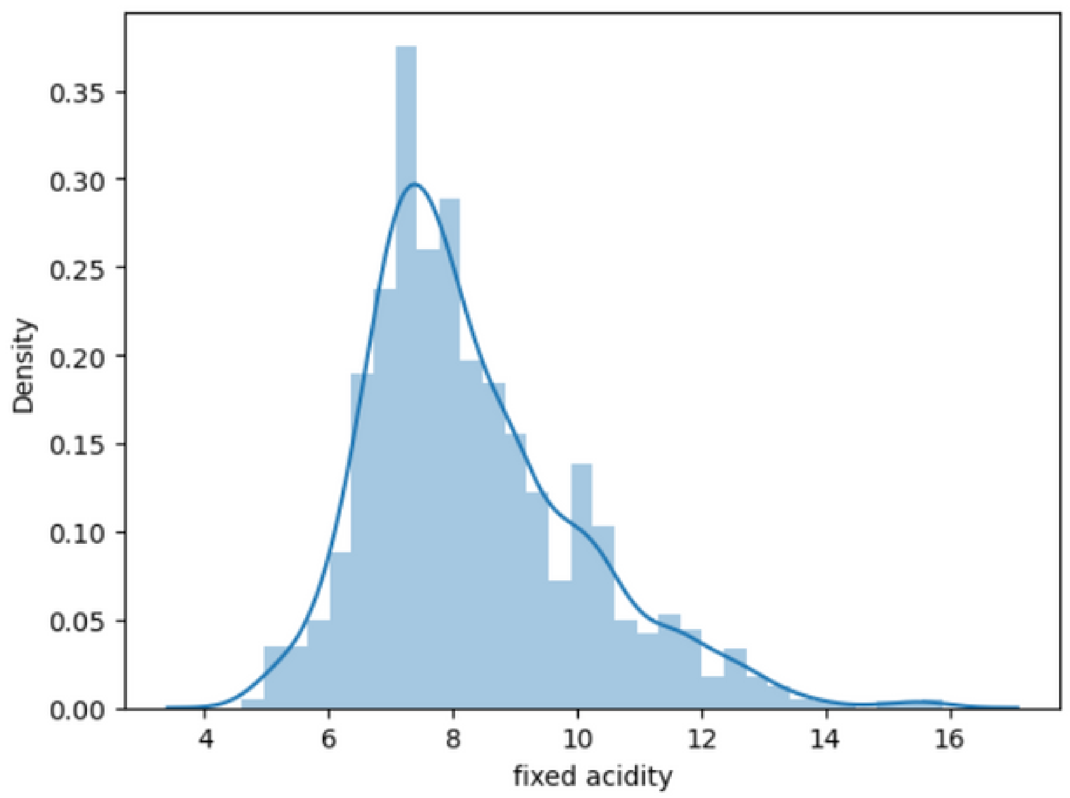
By exploring the correlations between different wine characteristics and the assigned quality ratings, the analysis aims to uncover the most significant contributors to overall wine quality.

```
<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)
```

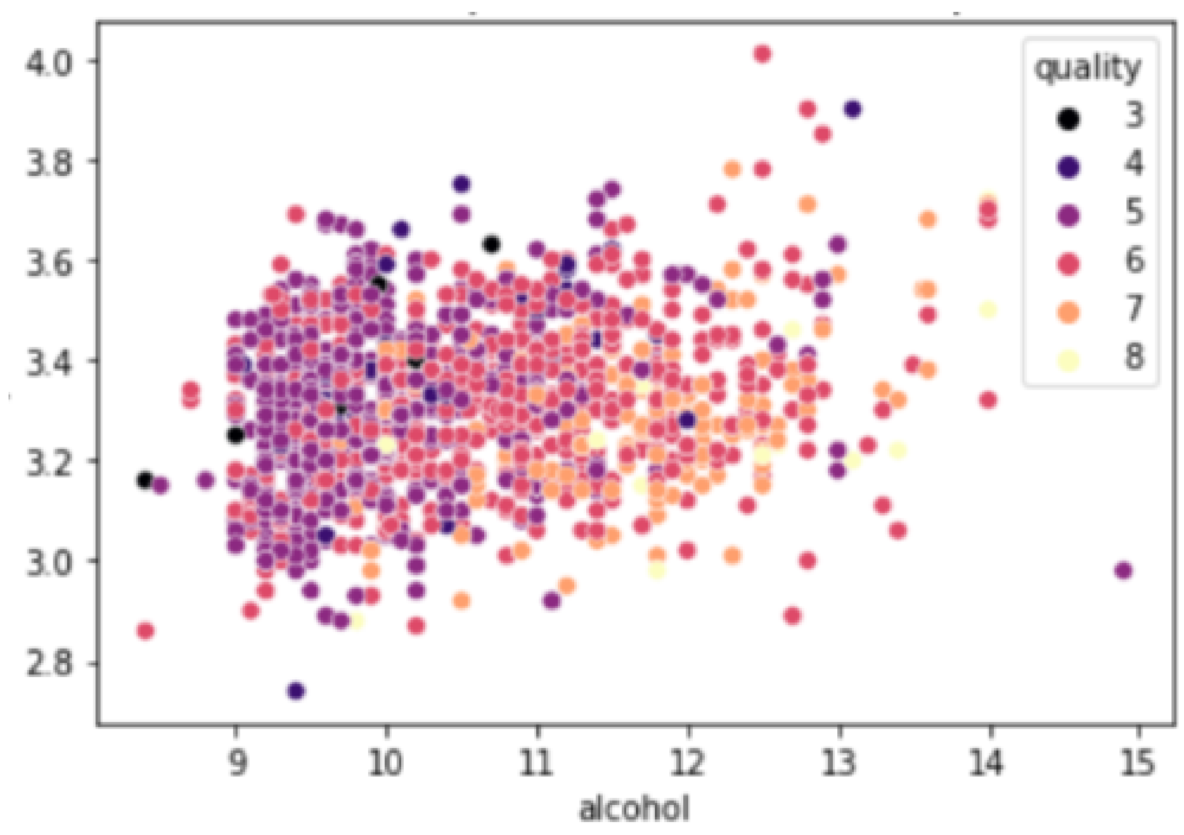
There are 0 null values



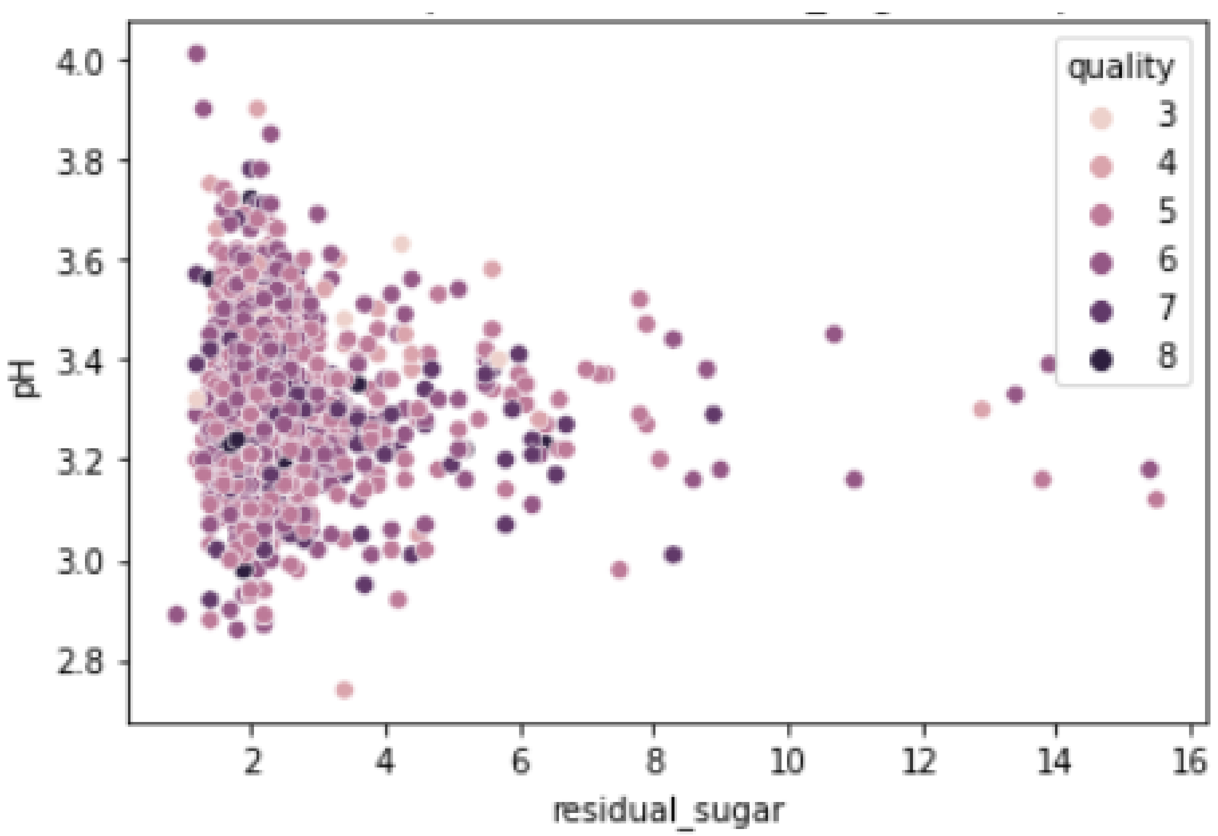
Most of the 'quality' variable are 5 and 6



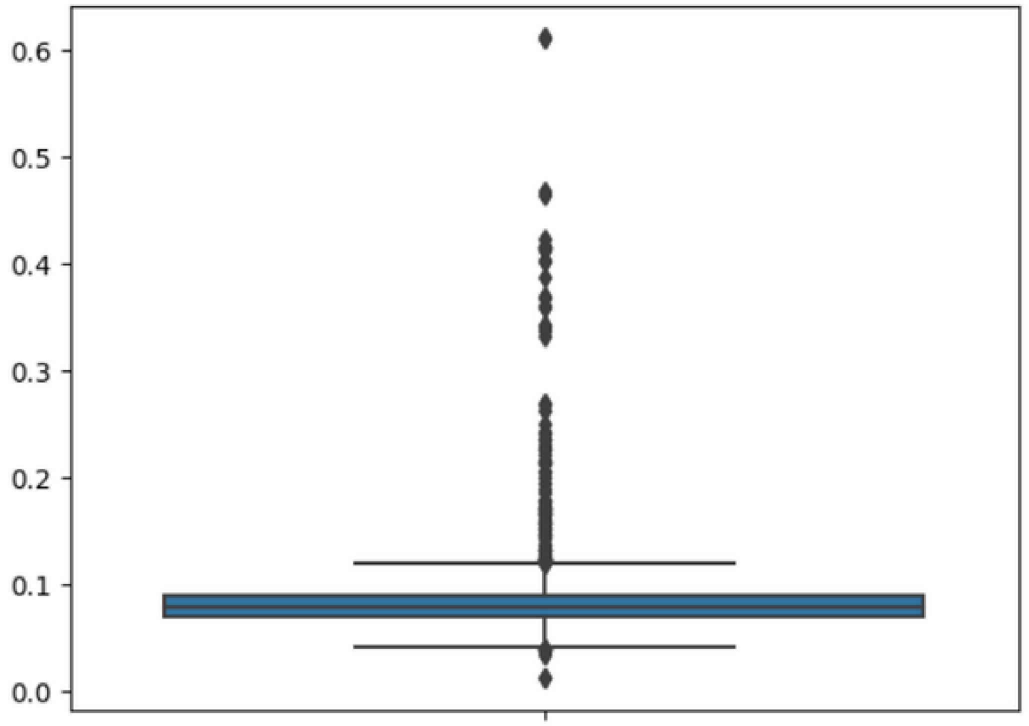
The values of the variable 'fixed_acidity' are relatively normally distributed (but a bit left skewed)



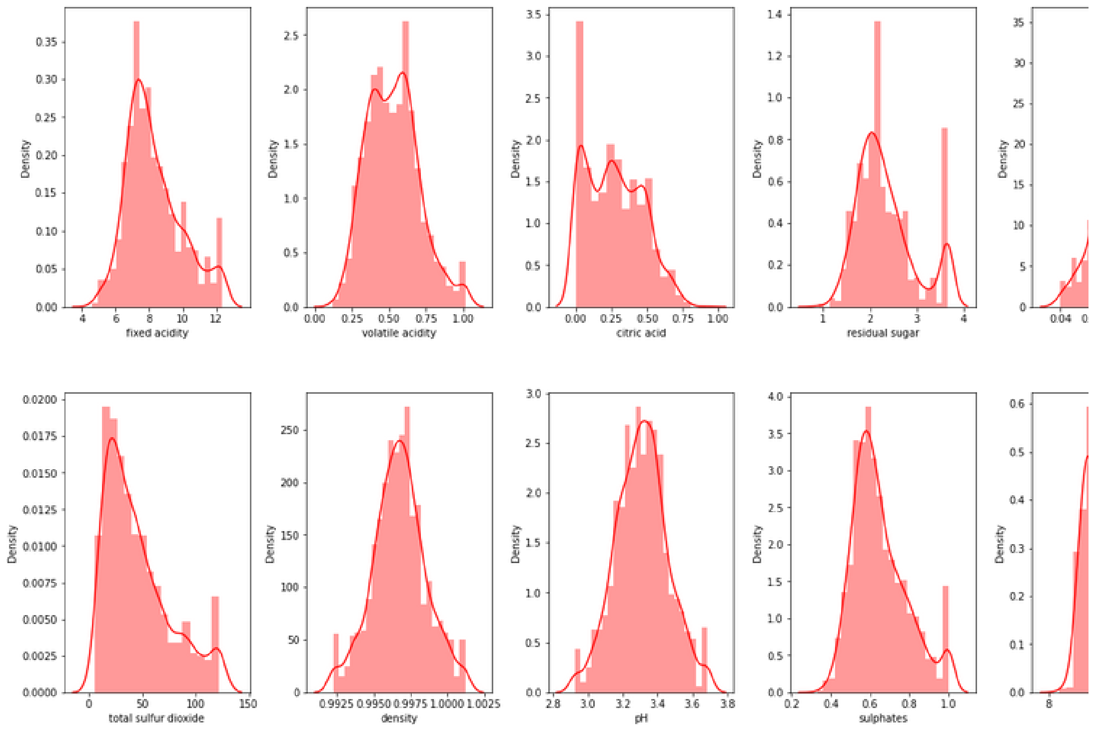
There is not correlation between 'alcohol' and 'pH' variables



There is no correlation between 'residual_sugar' and 'pH' variables



Where majority of ‘chlorides’ values lie near 0.1, we find some outliners too



Few of them are normally distributed where other are rightly. The range of each feature is also not huge.

IBM Machine Learning Professional Certificate

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality
fixed_acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.12
volatile_acidity	-0.26	1	-0.55	0.0019	0.061	-0.011	0.076	0.022	0.23	-0.26	-0.2	-0.39
citric_acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.23
residual_sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.014
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.13
free_sulfur_dioxide	-0.15	-0.011	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.051
total_sulfur_dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.066	0.043	-0.21	-0.19
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.17
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.066	-0.34	1	-0.2	0.21	-0.058
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.25
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.48
quality	0.12	-0.39	0.23	0.014	-0.13	-0.051	-0.19	-0.17	-0.058	0.25	0.48	1

- There is relatively high (0.67, positive) correlation between 'free sulfur dioxide' and 'total_sulfur_dioxide' variables.
- There is relatively high (-0.68, negative) correlation between "pH" and "fixed_acidity" variables.
- And there is about 0.5 correlation between some of other variables.

FEATURE ENGINEERING

```
bins = (2, 6.5, 8)
group_names = ['bad', 'good']
df['quality'] = pd.cut(df['quality'], bins = bins, labels = group_names)
```

```
label_quality = LabelEncoder()
```

```
df['quality'] = label_quality.fit_transform(df['quality'])
```

```
df['quality'].value_counts()
```

```
0    1382
1     217
Name: quality, dtype: int64
```

CLASSIFIER MODELS

```
from xgboost import XGBClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

```
lr = LogisticRegression()
knn5 = KNeighborsClassifier(n_neighbors = 5)
xgb = XGBClassifier()
rfc = RandomForestClassifier
SVM = SVC(kernel='rbf', random_state=0, gamma=.10, C=1.0)
sc = StandardScaler()
svc = SVC()
```

IBM Machine Learning Professional Certificate

CLASSIFIER MODELS

LOGISTIC REGRESSION

```
estimator = Pipeline([("sc",sc),
                      ("regression", lr)])

params = {'regression__penalty':['l1', 'l2'],
              'regression__C': np.geomspace(4, 20, 30)}

prm = {'regression__penalty': ['l1','l2'], 'regression__C': [0.001,0.01,0.1,1,10,100,1000]}

grid = GridSearchCV(estimator, param_grid =prm, cv=kf,verbose=0)
```

```
accuracy_score(y_test, y_predict)
```

```
0.86875
```

CLASSIFIER MODELS

KNN

```
knn_model = KNeighborsClassifier(n_neighbors = knn_cv_model.best_params_["n_neighbors"],
                                 leaf_size = knn_cv_model.best_params_["leaf_size"],
                                 weights = knn_cv_model.best_params_["weights"])

knn_model.fit(x_train, y_train)
```

```
accuracy_score(y_test, y_predict)
```

```
0.8875
```

IBM Machine Learning Professional Certificate

CLASSIFIER MODELS

SUPPORT VECTOR CLASSIFIER

```
param = {
    'svm__C': [0.1,0.8,0.9,1,1.1,1.2,1.3,1.4],
    'svm__kernel':['linear', 'rbf'],
    'svm__gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(est1, param_grid=param, scoring='accuracy', cv=10)
```

```
accuracy_score(y_test, y_predict)
```

```
0.878125
```

CLASSIFIER MODELS

RANDOM FOREST CLASSIFIER

```
rfc = RandomForestClassifier(n_estimators = 200)
```

```
#Now lets try to do some evaluation for random forest model using cross validation.
rfc_eval = cross_val_score(estimator = rfc, X = X_train, y = y_train, cv = 10)
rfc_eval.mean()
```

```
0.91166338582677164
```

BETTER MODEL

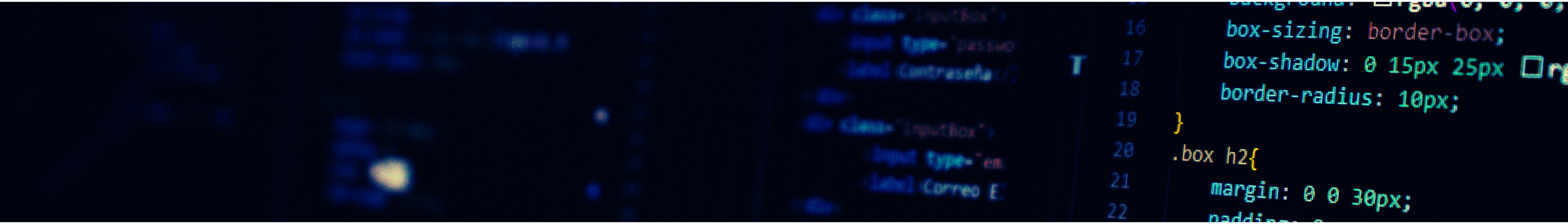
Based on the statistics shown above we can clearly state that the recommended model is **Random Forest Classifier with Cross Validation** and this is due to the extraordinary performance demonstrated in the previous section

CLASSIFIER MODELS	ACCURACY
Logistic Regression	**86.8%**
KNN	**88.7%**
Support Vector Classifier	**87.8%**
Random Forest Classifier	**91.1%**

KEY FINDINGS & INSIGHTS

It is certain that further analysis and more models could be applied to this data set and maybe we can have better results. However, some of the suggestions I see for our next steps include having a GridSearchCV to try to eliminate over-fitting while further enhancing our models to choose the best hyperparameters for each of them. We can also use these models and save them using the pickle library for later use in more sophisticated models or act as a “teacher model” for data distillation

THANK YOU



LINKEDIN :

@sukhmani1303

EMAIL :

uv1303@gmail.com

WEBSITE :

linktr.ee/sukhmani1303