

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

```
df=pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv',sep=';')
df
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

```
X=df.drop('quality',axis=1)
Y=df['quality']

x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=42)

dt=DecisionTreeClassifier(random_state=42)

dt.fit(x_train,y_train)
```

▼

DecisionTreeClassifier

DecisionTreeClassifier(random\_state=42)

```
y_pred=dt.predict(x_test)

print('Decision Tree Classifier:')
print('Classification Report:\n',classification_report(y_test,y_pred))
```

Decision Tree Classifier:					
Classification Report:					
	precision	recall	f1-score	support	
3	0.00	0.00	0.00	1	
4	0.00	0.00	0.00	10	
5	0.63	0.68	0.65	130	
6	0.55	0.53	0.54	132	
7	0.51	0.50	0.51	42	
8	0.00	0.00	0.00	5	
accuracy			0.56	320	
macro avg	0.28	0.28	0.28	320	
weighted avg	0.55	0.56	0.55	320	

```
Bagging=BaggingClassifier(base_estimator=dt,n_estimators=100,random_state=42)

Bagging.fit(x_train,y_train)
```

```

/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `t
warnings.warn(

```

```

> BaggingClassifier
> base_estimator: DecisionTreeClassifier
  > DecisionTreeClassifier

```

```
y_pred=Bagging.predict(x_test)
```

```

print('Bagging Classifier:')
print('Classification Report:\n',classification_report(y_test,y_pred))

```

```

Bagging Classifier:
Classification Report:

```

	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	10
5	0.72	0.78	0.75	130
6	0.63	0.68	0.66	132
7	0.59	0.48	0.53	42
8	0.00	0.00	0.00	5
accuracy			0.66	320
macro avg	0.32	0.32	0.32	320
weighted avg	0.63	0.66	0.64	320

```

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-c
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-c
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-c
_warn_prf(average, modifier, msg_start, len(result))

```

```
Pasting=BaggingClassifier(base_estimator=dt,n_estimators=100,bootstrap=False,random_state=True)
```

```
Pasting.fit(x_train,y_train)
```

```

/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `base_estimator` was renamed to `esti
warnings.warn(

```

```

> BaggingClassifier
> base_estimator: DecisionTreeClassifier
  > DecisionTreeClassifier

```

```
y_pred=Pasting.predict(x_test)
```

```

print('Pasting Classifier :')
print('Classification Report :\n',classification_report(y_test,y_pred))

```

```

Pasting Classifier :
Classification Report :

```

	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	10
5	0.64	0.67	0.65	130
6	0.57	0.55	0.56	132
7	0.53	0.55	0.54	42
8	0.00	0.00	0.00	5
accuracy			0.57	320
macro avg	0.29	0.29	0.29	320
weighted avg	0.56	0.57	0.57	320

```
Bagging=BaggingClassifier(base_estimator=dt,n_estimators=100,random_state=42,oob_score=True)
```

```
Bagging.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_base.py:166: FutureWarning: `base_estimator` was renamed to `estimators` in 1.0.0. Please update your code to use this new parameter name.
warnings.warn(
```

```

> BaggingClassifier
> base_estimator: DecisionTreeClassifier
  > DecisionTreeClassifier

```

```
y_pred=Bagging.predict(x_test)
```

```
print('Out Of Bag Score : ',Bagging.oob_score_)
```

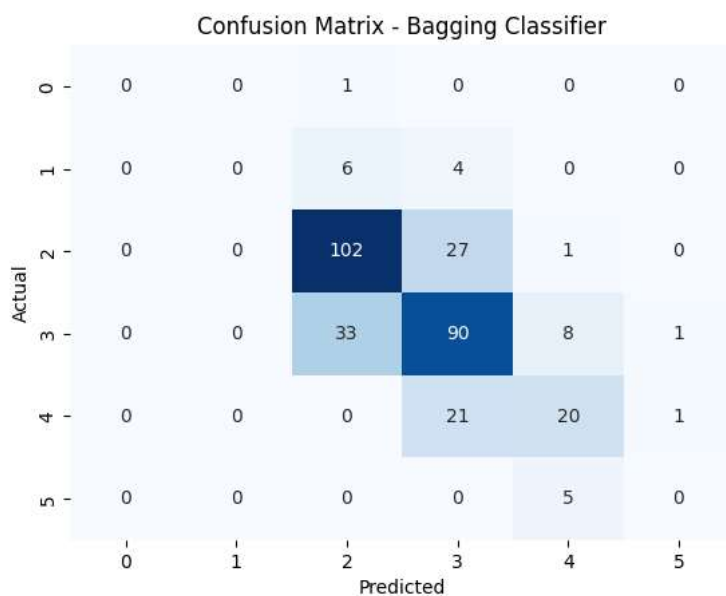
```
Out Of Bag Score : 0.6942924159499609
```

```
con=confusion_matrix(y_test,y_pred)
```

```

sns.heatmap(con,annot=True,fmt='d',cmap=plt.cm.Blues,cbar=False)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Bagging Classifier')
plt.show()

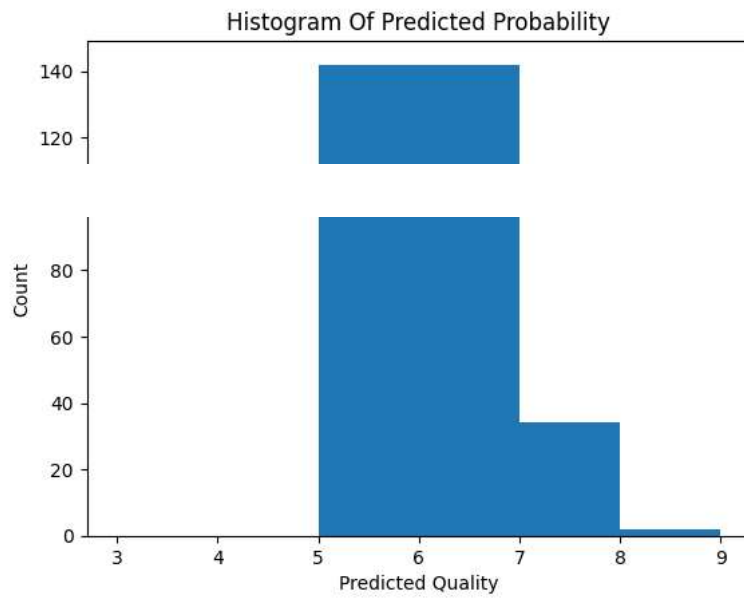
```



```

plt.hist(y_pred,bins=range(3,10))
plt.xlabel('Predicted Quality')
plt.ylabel('Count')
plt.title('Histogram Of Predicted Probability')
plt.show()

```



[Colab paid products](#) - [Cancel contracts here](#)

