

Water Quality

Data Mining

Project Documentation

Building of Classifier Models

To Predict the Water Potability & Calculate the Error Percentage

- **TA. Nourhan Bahnasy**

Name	Section	University ID
أحمد ناصر أحمد حسن	1	20191701016
يوسف عصام فؤاد محمد	9	20191701269
مريم عبدالهادي محمد عبدالغفار	9	20191701195
أمانى جمال رسلان حسن	2	20191701033
جنى هانى أحمد صادق	3	20191701059
عبدالرحمن يسري إبراهيم البابلى	5	20191701124

Used Models

2	Naive-Bayes Classifier
3	KNN (ver1) Classifier
4	KNN (ver2) Classifier
5	Random-Forest Classifier
6	Gradient-Boosting Classifier
7	SGD Classifier
8	Stratified-kFold Classifier
9	ID3 Classifier
10	Logistic Classifier
11	SVM (ver1) Classifier
12	SVM (ver2) Classifier
13	SVM (ver3) Classifier
14	XG-Boost Classifier

*For the first dataset “waterQuality1”, the best model that
succeeded to get the highest testing accuracy is:*

Random-Forest Classifier

1) Naive-Bayes Classifier

a) Data Preprocessing

- i) Sort values
- ii) Drop duplicates
- iii) Fill data missing with mean to improve accuracy
- iv) Shuffling
- v) Stratify
- vi) Label Encoder
- vii) Standard Scaler

b) Testing Accuracy

- i) 64.62093862815884 %

c) Data Separation

- i) 80% training : 20% testing

d) Result & Output

- i) Screen Shot

```
[[305 166]
 [ 30  53]]
0.6462093862815884
0.35379061371841153
[0.57948718 0.66494845 0.61340206 0.59793814 0.59793814 0.63402062
 0.62371134 0.59278351 0.60824742 0.63402062]
```

2) KNN (ver1) Classifier

a) Data Preprocessing

- i) Drop NULL values
- ii) Drop the duplicates
- iii) Normalization
- iv) Selection of the important features
- v) Stratified Sampling

b) Testing Accuracy

- i) 62.03473945409429 %

c) Data Separation

- i) 80% training : 20% testing

d) Result & Output

- i) Screen Shot

```
the result of the knn is [0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1
0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0
1 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0
0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1]

the prediction score is : 62.03473945409429
the mean Square error 0.37965260545905705

Process finished with exit code 0
```

3) KNN (ver2) Classifier

a) Data Preprocessing

- i) Dropping NULL values
- ii) Dropping Duplicates
- iii) Extract the features
- iv) Normalization
- v) Shuffling

b) Testing Accuracy

- i) 71.21588089330024 %

c) Data Separation

- i) 80% training : 20% testing

d) Result & Output

- i) Screen Shot

```
[[206  47]
 [ 69  81]]
Accuracy = 0.7121588089330024
Error = 0.28784119106699757

Process finished with exit code 0
```

4) Random-Forest Classifier

a) Data Preprocessing

- i) Sort values**
- ii) Drop Duplicates**
- iii) Fill data missing with mean to improve accuracy**
- iv) Shuffle**
- v) Stratify**
- vi) Label Encoder**
- vii) Standard Scaler**

b) Testing Accuracy

- i) 73.10469314079422 %**

c) Data Separation

- i) 80% training : 20% testing**

d) Result & Output

- i) Screen Shot**

```
[[317 131]
 [ 18  88]]
0.7310469314079422
0.26895306859205775
```

5) *Gradient-Boosting Classifier*

a) Data Preprocessing

- i) Removing duplicates
- ii) replace outliers by nulls
- iii) replace nulls by mean value

b) Testing Accuracy

- i) max: 65%

c) Data Separation

- i) 80% training 20% testing

d) Result & Output

- i) Screen Shot

```
Learning rate: 0.1
Training Accuracy: 0.626
Validation Accuracy: 63.736 %

Learning rate: 0.2
Training Accuracy: 0.634
Validation Accuracy: 63.599 %

Learning rate: 0.3
Training Accuracy: 0.648
Validation Accuracy: 63.324 %

Learning rate: 0.4
Training Accuracy: 0.657
Validation Accuracy: 64.698 %

Learning rate: 0.5
Training Accuracy: 0.664
Validation Accuracy: 65.110 %

Learning rate: 0.6
Training Accuracy: 0.673
Validation Accuracy: 62.500 %

Learning rate: 0.7
Training Accuracy: 0.673
Validation Accuracy: 64.148 %

Learning rate: 0.8
Training Accuracy: 0.672
Validation Accuracy: 63.324 %
```

6) SGD Classifier

a) Data Preprocessing

- i) Removing duplicates**
- ii) Replace outliers by nulls**
- iii) Replace nulls by mean value**

b) Testing Accuracy

- i) 62.8%**

c) Data Separation

- i) 80% training 20% testing**

d) Result & Output

- i) Screen Shot**



```
Accuracy: 62.80487804878049 %
```


7) Stratified-kFold Classifier

a) Data Preprocessing

- i) Removing duplicates**
- ii) replace outliers by nulls**
- iii) replace nulls by mean value**

b) Testing Accuracy

- i) 60.9%**

c) Data Separation

- i) 80% training 20% testing**

d) Result & Output

- i) Screen Shot**

```
Maximum Accuracy That can be obtained from this model is: 61.16207951070336 %  
  
Minimum Accuracy: 60.85626911314985 %  
  
Overall Accuracy: 60.98903557842918 %  
  
Standard Deviation is: 0.0010339986558718815
```

8) ID3 Classifier

a) Data Preprocessing

- i) Dropping NULL values.**
- ii) Normalization.**
- iii) Shuffling data.**
- iv) Random Sampling.**

b) Testing Accuracy

- i) 61.36645962732919 %**

c) Data Separation

- i) 60% training : 40% testing**

d) Result & Output

- i) Screen Shot**

```
Test Accuracy: 61.36645962732919 %  
Mean Square Error: 0.38633540372670805  
|  
Process finished with exit code 0
```

9) Logistic Classifier

a) Data Preprocessing

- i) Removing rows that has NULL values.
- ii) Normalization.
- iii) Shuffling data.
- iv) Random Sampling

b) Testing Accuracy

- i) 60.12422360248447 %

c) Data Separation

- i) 60% training : 40% testing

d) Result & Output

- i) Screen Shot

```
Test Accuracy: 60.12422360248447 %
Mean Square Error: 0.3987577639751553
|

Report:
      precision    recall  f1-score   support

     0       0.61      0.97      0.75       492
     1       0.28      0.02      0.03       313

 accuracy          0.60       805
 macro avg          0.44      0.49      0.39       805
weighted avg          0.48      0.60      0.47       805
```

10) SVM (*ver1*) Classifier

a) Data Preprocessing

- i) Scaling feature data using MINMAXScaler
- ii) Handling NULL values with Median in the same column.
- iii) Using a Simple Imputer.
- iv) Shuffling data.
- v) Random Sampling.

b) Testing Accuracy

- i) 61.35531135531136 %

c) Data Separation

- i) 50% training : 50% testing

d) Result & Output

- i) Screen Shot

```
train accuracy : 60.62271062271062 %  
test accuracy : 61.35531135531136 %  
Mean Square Error : 0.38644688644688646
```

11) SVM (ver2) Classifier

a) Data Preprocessing

- i) Remove null values
- ii) Dropping for PH and Sulfate features
- iii) Remove Duplicates

b) Testing Accuracy

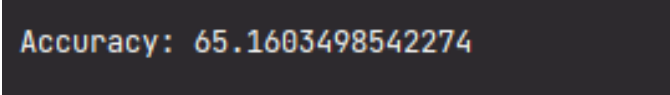
- i) 65.1603498542274 %

c) Data Separation

- i) 80% training : 20% testing

d) Result & Output

- i) Screen Shot



```
Accuracy: 65.1603498542274
```

12) SVM (*ver3*) Classifier

a) Data Preprocessing

- i) Used the Second(new) Dataset
- ii) Convert objects data types into float & integers
- iii) Normalization

b) Testing Accuracy

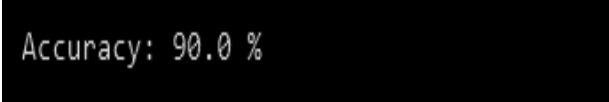
- i) 90.0 %

c) Data Separation

- i) 80% training : 20% testing

d) Result & Output

- i) Screen Shot



Accuracy: 90.0 %

13) *XG-Boost Classifier*

a) Data Preprocessing

- i) Used the Second(new) Dataset**
- ii) Normalization**
- iii) Remove duplicates**
- iv) Replace null values by mean**

b) Testing Accuracy

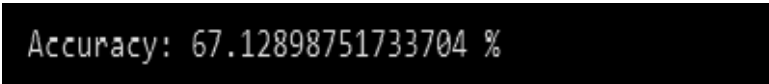
- i) 67.12898751733704 %**

c) Data Separation

- i) 80% training : 20% testing**

d) Result & Output

- i) Screen Shot**



Accuracy: 67.12898751733704 %