Prediction of Water Quality using Data Mining

Contents

Introduction

Machine Learning Tasks

Dataset

Data preprocessing

Binary Classification

- Logistic Regression
- K-Nearest Neighbors

Clustering

- K-means

Conclusion

References

Introduction

Water is the most important natural resource on earth

Therefore, water quality and safety is of great importance

In this project, we use data mining algorithms to predict whether water is safe or not based on concentrations of its elements

Machine Learning Tasks and Algorithms

Binary Classification

Classify water quality into safe or unsafe using:

- Logistic Regression algorithm
- K-Nearest Neighbors (KNN) algorithm

Clustering

Group the water quality into clusters using:

- K-Means clustering algorithm

Dataset

Number of instances: 8000

Number of attributes: 21

Target attribute: is_safe

#	name	Type
1	aluminum	float64
2	ammonia	float64
3	arsenic	float64
4	barium	float64
5	cadmium	float64
6	chloramine	float64
7	chromium	float64
8	copper	float64
9	fluoride	float64
10	bacteria	float64
11	viruses	float64
12	lead	float64
13	nitrates	float64
14	nitrites	float64
15	mercury	float64
16	perchlorate	float64
17	radium	float64
18	selenium	float64
19	silver	float64
20	uranium	float64
21	is_safe	float64

Data preprocessing

Missing-value Treatment

- There are six missing values in the dataset
- We removed the instances with the missing values

Normalization

- We normalized the dataset to improve accuracy of data mining algorithms
- We used the Min-Max normalization

Binary Classification

Logistic Regression algorithm

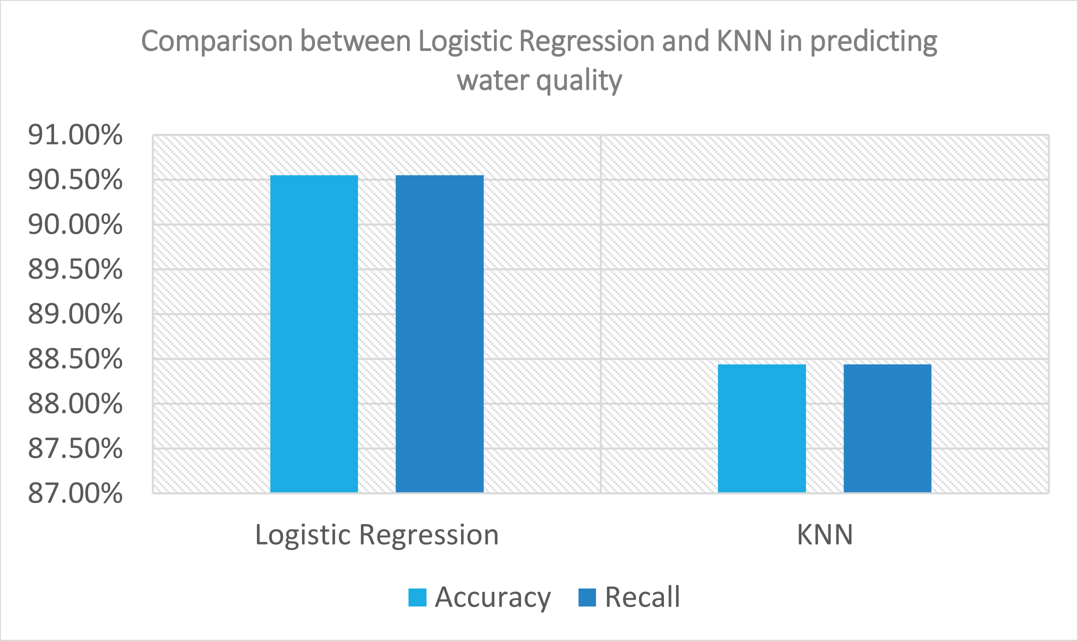
- The Logistic Regression is a supervised linear classification algorithm.
- It can be used to classify objects of binary and multi-class problems.

K-Nearest Neighbors (KNN) algorithm

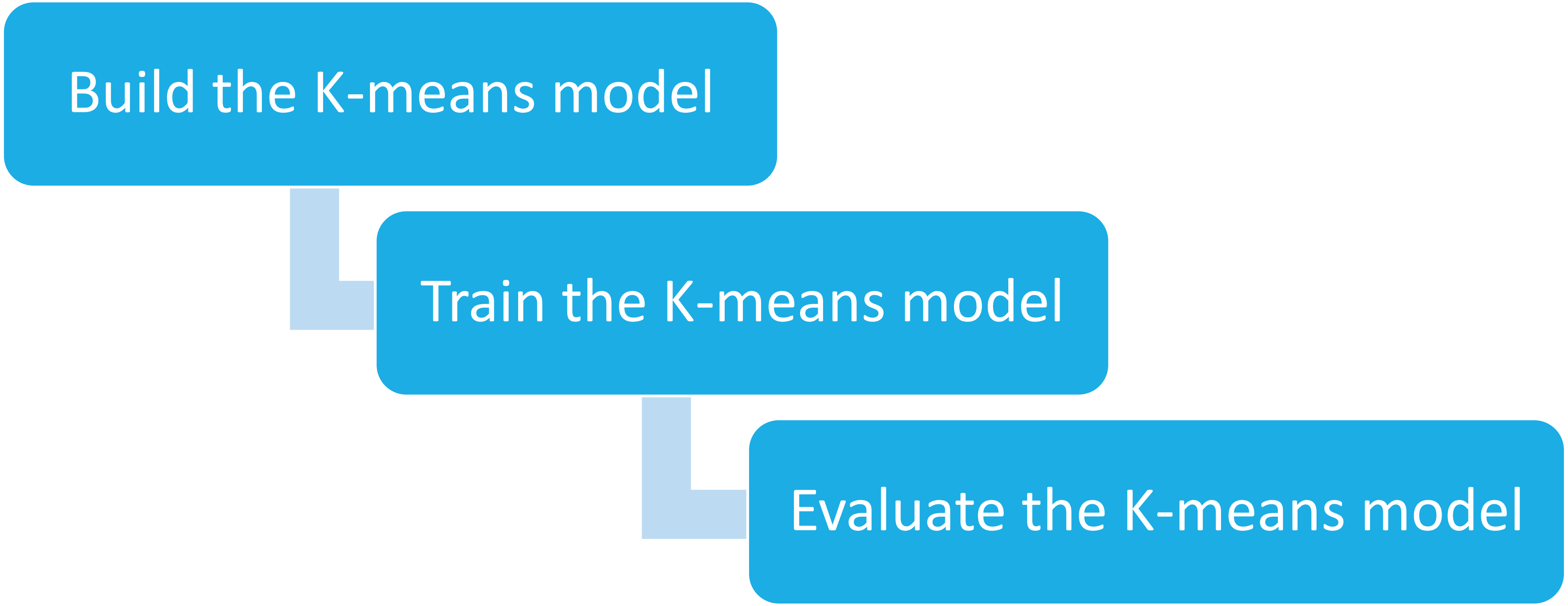
- KNN is a supervised machine learning algorithm.
- It classifies an object based on the distance between the object and the classes in the dataset.

Results

Performance Metric	Accuracy	Recall
Logistic Regression	90.55%	90.55%
KNN	88.44%	88.44%



Clustering using K-Means Algorithm

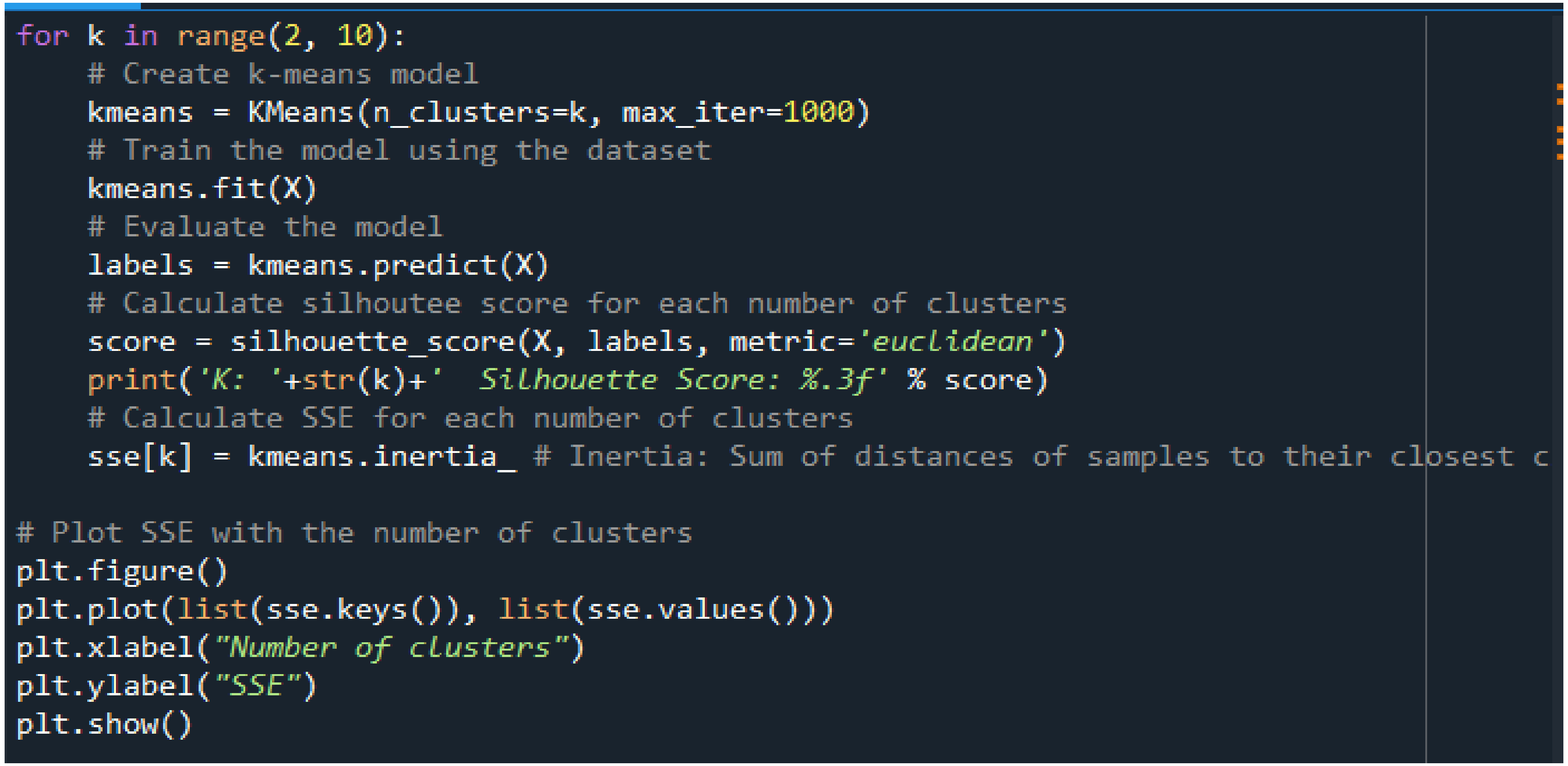


Build the K-means model

Train the K-means model

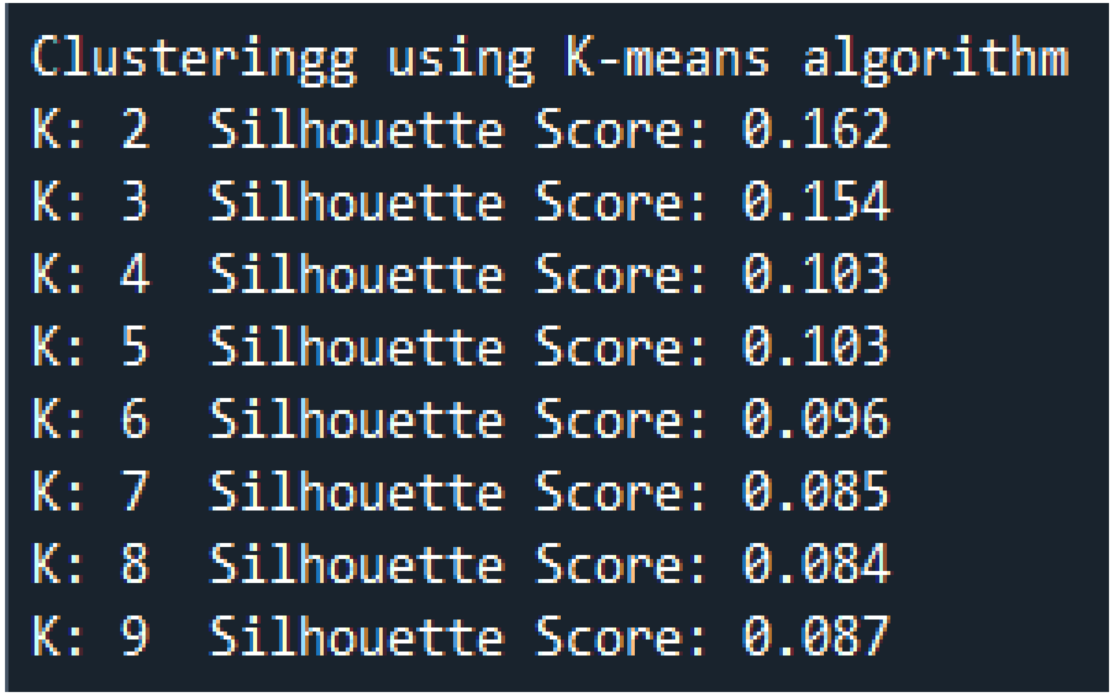
Evaluate the K-means model

Implementation of K-Means Algorithm



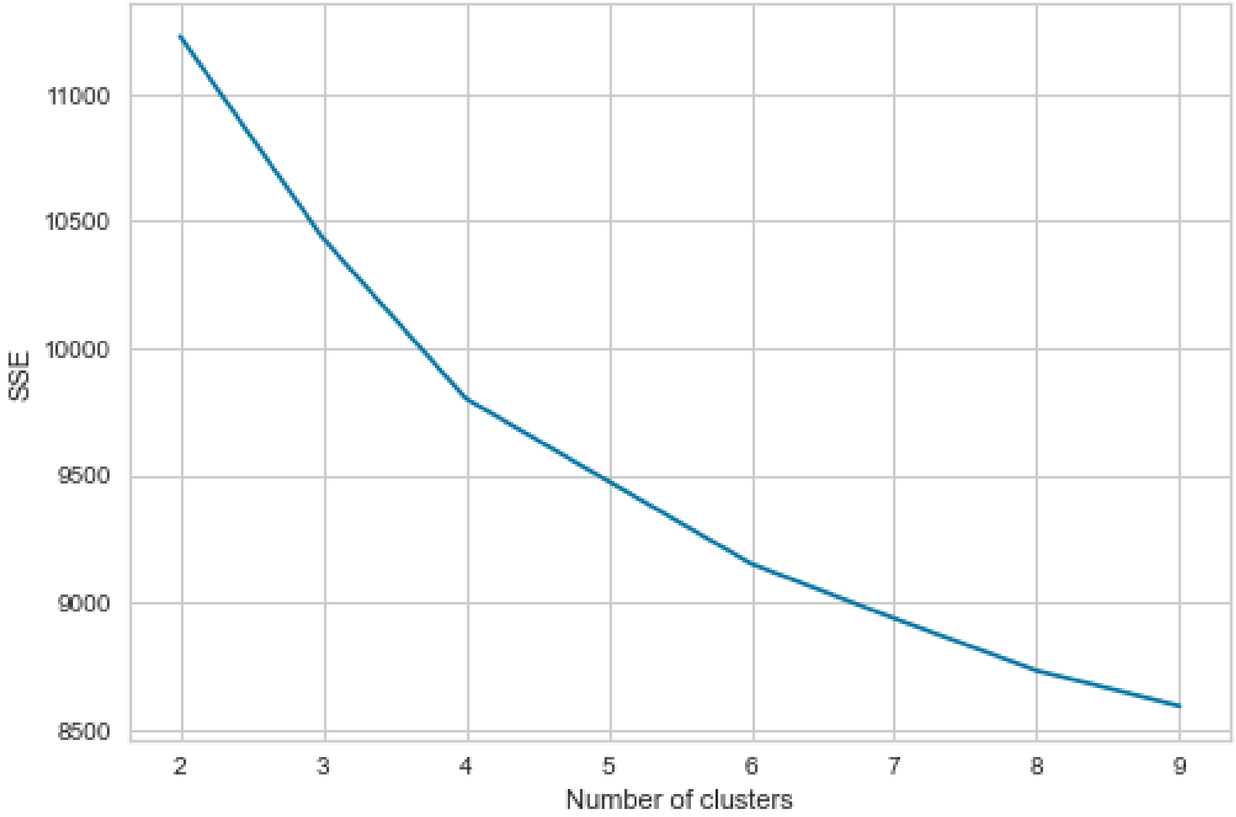
```python
for k in range(2, 10):
    # Create k-means model
    kmeans = KMeans(n_clusters=k, max_iter=1000)
    # Train the model using the dataset
    kmeans.fit(X)
    # Evaluate the model
    labels = kmeans.predict(X)
    # Calculate silhoutee score for each number of clusters
    score = silhouette_score(X, labels, metric='euclidean')
    print('K: '+str(k)+'  Silhouette Score: %.3f' % score)
    # Calculate SSE for each number of clusters
    sse[k] = kmeans.inertia_ # Inertia: Sum of distances of samples to their closest c

# Plot SSE with the number of clusters
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel("Number of clusters")
plt.ylabel("SSE")
plt.show()
```

Results



```
Clusteringg using K-means algorithm
K: 2  Silhouette Score: 0.162
K: 3  Silhouette Score: 0.154
K: 4  Silhouette Score: 0.103
K: 5  Silhouette Score: 0.103
K: 6  Silhouette Score: 0.096
K: 7  Silhouette Score: 0.085
K: 8  Silhouette Score: 0.084
K: 9  Silhouette Score: 0.087
```

Results


11000
10500
10000
9500
9000
8500
SSE
2
3
4
5
6
7
8
9
Number of clusters

Conclusion

In this project, we used Logistic Regression and KNN to predict water quality using measurements of mineral elements.

We also used K-means clustering algorithm to group the dataset into different number of clusters.

The dataset was cleaned before data mining tasks.

Results showed that data mining classification algorithms can successfully predict the quality of water with a classification accuracy greater than 90%.

Results of clustering showed that the optimal number of clusters were two clusters.

References

1. Water quality, Dataset for water quality classification, https://www.kaggle.com/datasets/mssmartypants/water-quality
2. Python, https://www.python.org/
3. Anaconda, https://www.anaconda.com/
4. Spyder, https://www.spyder-ide.org/
5. Pandas, https://pandas.pydata.org/
6. MinMaxScaler, https://scikit learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
7. Train test split, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
8. Logistic Regression, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
9. Accuracy score, precision score, recall score, f1_score, https://scikit-learn.org/stable/modules/model_evaluation.html