International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135


Journal Website: http://sciencebring.com/index.php/ijasr



Research Article

# NON-PARAMETRIC METHODS. K-NEAREST NEIGHBORS MODEL

**Submission Date:** December 01, 2023, **Accepted Date:** December 05, 2023,
**Published Date:** December 09, 2023
**Crossref doi:** https://doi.org/10.37547/ijasr-03-12-04

**Salimov Jamshid Obid , O'G'Li**
**Assistant Jizzakh Branch Of National University Of Uzbekistan**

**Xudoyqulov Diyorbek Shakar O'G'Li**
**Student Jizzakh Branch Of National University Of Uzbekistan**

**Avazov Asadbek Egamberdi O'G'Li**
**Student Jizzakh Branch Of National University Of Uzbekistan**

## ABSTRACT

One of the machine learning algorithms k-Nearest Neighbors algorithm is widely used for classification tasks in the construction of artificial intelligence programs. In the k-NN algorithm, when determining which class a new object belongs to, the distances from this object to all objects are measured, and if there are more objects belonging to which class among the nearest k selected objects, the new object is considered to belong to this class, which makes it makes it an intuitive and powerful tool for solving complex problems. In this article, a model for determining whether a patient is diagnosed with breast cancer or not is created using the k-NN algorithm. This problem is calculated using binary classification.

## KEYWORDS

Dataset, testset, trainset, hyperparameters, classification, prediction.

## INTRODUCTION

**What is classification?**

- Supervised Machine Learning type

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135

- Classification of unknown elements into categories (classes)
- Classifiers can be of two types.
- Binary classifiers
- Multiclass classifiers

**Generating k-Nearest Neighbors methods using scikit-learn**

Implementation of breast cancer detection algorithm using k nearest neighbors method.

**Description:** Breast cancer is the most common cancer among women in the world. It accounts for 25% of all cancer cases. Breast cancer begins when cells in the breast grow out of control. These cells are usually detected by analyzing tumors that can be seen on X-rays.

```
import pandas as pd

import numpy as np

df=pd.read_csv("/content/Breast_cancer_data.csv")

df
```

| | mean_radius | mean_texture | mean_perimeter | mean_area | mean_smoothness | diagnosis |
|---|---|---|---|---|---|---|
| 0 | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0 |
| 1 | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0 |
| 2 | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0 |
| 3 | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0 |
| 4 | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0 |
| ... | ... | ... | ... | ... | ... | ... |
| 564 | 21.56 | 22.39 | 142.00 | 1479.0 | 0.11100 | 0 |
| 565 | 20.13 | 28.25 | 131.20 | 1261.0 | 0.09780 | 0 |
| 566 | 16.60 | 28.08 | 108.30 | 858.1 | 0.08455 | 0 |
| 567 | 20.60 | 29.33 | 140.10 | 1265.0 | 0.11780 | 0 |
| 568 | 7.76 | 24.54 | 47.92 | 181.0 | 0.05263 | 1 |

569 rows × 6 columns

```
df.shape
```

**Result:**

**(569,32)**

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

**Result:**

```
B  357
M  212
Name: diagnosis, dtype: int64
```

Let's change these values to 0 and 1. M->1, B->0
For this, you can use the **LabelEncoder** in sklearn or the **.replace()** method in pandas. Both ways are shown below.

```
from sklearn.preprocessing import LabelEncoder
```

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135


```
labelencoder = LabelEncoder()

df['diagnosis'] = labelencoder.fit_transform(df['diagnosis'].values)

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

**Result:**

```
0   357
1   212
Name: diagnosis, dtype: int64
```

Correlation (linear relationship) can be seen in the graph below

```
corr_matrix = df.corr().abs()

corr_matrix.style.background_gradient(cmap='coolwarm')
```

| | mean_radius | mean_texture | mean_perimeter | mean_area | mean_smoothness | diagnosis |
|---|---|---|---|---|---|---|
| mean_radius | 1.000000 | 0.323782 | 0.997855 | 0.987357 | 0.170581 | 0.730029 |
| mean_texture | 0.323782 | 1.000000 | 0.329533 | 0.321086 | 0.023389 | 0.415185 |
| mean_perimeter | 0.997855 | 0.329533 | 1.000000 | 0.986507 | 0.207278 | 0.742636 |
| mean_area | 0.987357 | 0.321086 | 0.986507 | 1.000000 | 0.177028 | 0.708984 |
| mean_smoothness | 0.170581 | 0.023389 | 0.207278 | 0.177028 | 1.000000 | 0.358560 |
| diagnosis | 0.730029 | 0.415185 | 0.742636 | 0.708984 | 0.358560 | 1.000000 |

```
df.corrwith(df["diagnosis"]).abs().sort_values(ascending=False)

diagnosis          1.000000
mean_perimeter     0.742636
mean_radius        0.730029
mean_area          0.708984
mean_texture       0.415185
mean_smoothness    0.358560
dtype: float64
```

Data is extracted:

```
X=df.drop("diagnosis", axis=1).values

y=df["diagnosis"]
```

Using the module below, values are standardized because all column values are in different ranges.

```
from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()

X=scaler.fit_transform(X)
```

The train_test_split module below is used to split the DataSet into train_set and test_set.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =train_test_split(X,y,test_size=0.20,
```

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135



random_state=12)

The k-NN model is recalled.

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn=KNeighborsClassifier(n_neighbors=5)
```

X_train and Y_train data are fed to the model to train the model.

```
knn.fit(X_train, y_train)
```

X_test data allocated for testing is passed to find predictive values

Predictive values are found using the .predict() function.

```
y_pridect=knn.predict(X_test)
```

Actual values and predicted values are as follows.

```
print(y_test.to_numpy())
```

```
print(y_pridect)
```

**Result:**

**Original values:**

[0 1 1 1 1 1 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 1 0 1 1

0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0

0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 0 1

1 1 1]

**Predicted values:**

[0 1 1 1 1 0 1 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0 0 1 0 1 1

0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 0 0

0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1

1 1 1]

**Model evaluation.**

Evaluation by the Jaccard index evaluation criterion:

```
from sklearn.metrics import jaccard_score
```

```
jaccard_score(y_test, y_pridect)
```

**Result:**

0.8958333333333334

**Confusion matrix evaluation criteria:**

```
from sklearn.metrics import confusion_matrix
```

```
import seaborn as sns
```

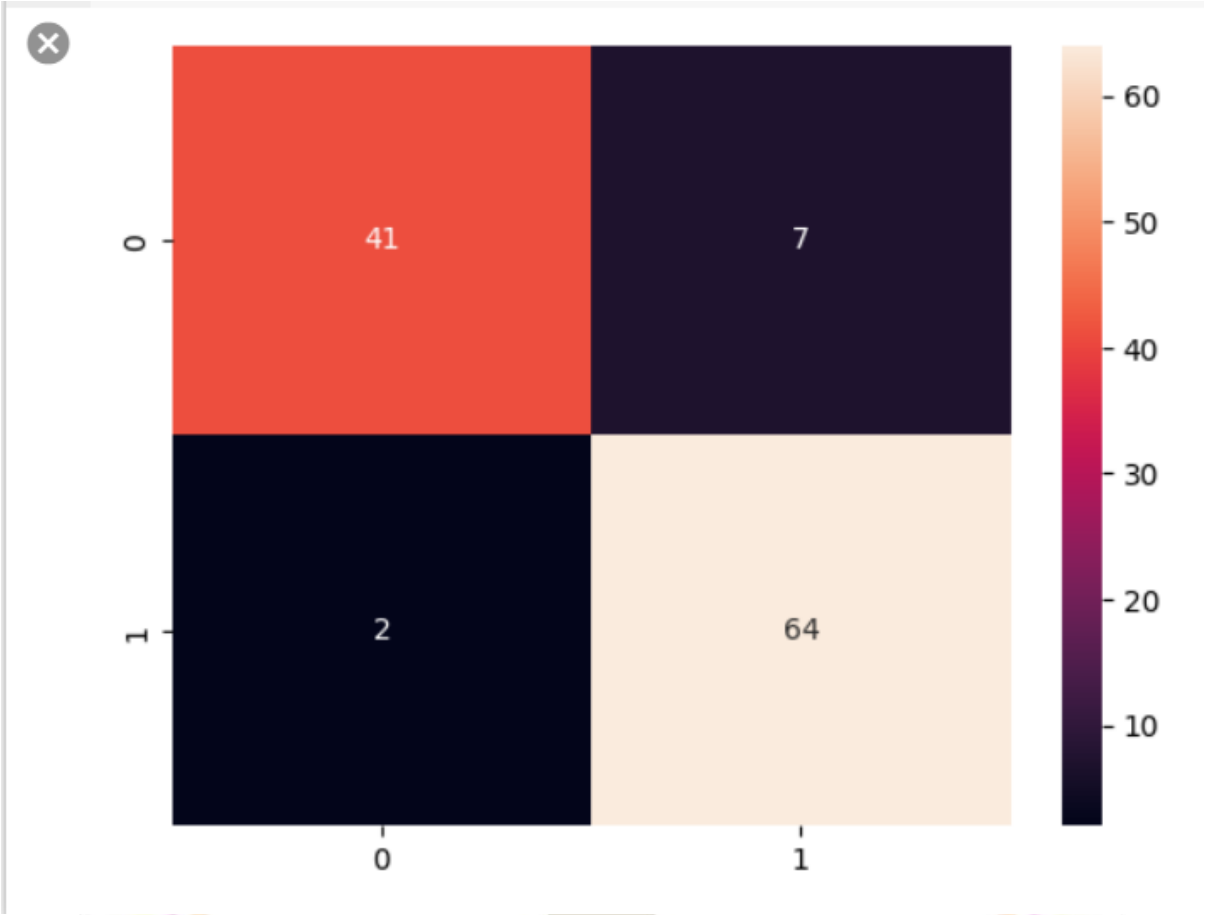
```
import matplotlib.pyplot as plt
```

```
sns.heatmap(confusion_matrix(y_test, y_pridect), annot=True)
```

```
plt.show()
```

**Result:**

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135




```
confusion_matrix(y_test, y_pridect)
```

**Result:**

```
array([[41,  7],
       [ 2,  64]])
```

**Calculate Precision, Recall, F1 and Accuracy**

```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

precision=precision_score(y_test, y_pridect)

recall=recall_score(y_test, y_pridect)

f1=f1_score(y_test, y_pridect)

accuracy=accuracy_score(y_test, y_pridect)

print(f"Precision: {precision}")

print(f"Recall: {recall}")

print(f"f1: {f1}")

print(f"Accuracy: {accuracy}")
```

**Result:**

Precision: 0.9014084507042254

Recall: 0.9696969696969697

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135

f1: 0.9343065693430657

Accuracy: 0.9210526315789473

We can actually get this information in general view.

```
from sklearn.metrics import classification_report
a=classification_report(y_test, y_pridect)
print(a)
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.85 | 0.90 | 48 |
| 1 | 0.90 | 0.97 | 0.93 | 66 |
| accuracy | | | 0.92 | 114 |
| macro avg | 0.93 | 0.91 | 0.92 | 114 |
| weighted avg | 0.92 | 0.92 | 0.92 | 114 |

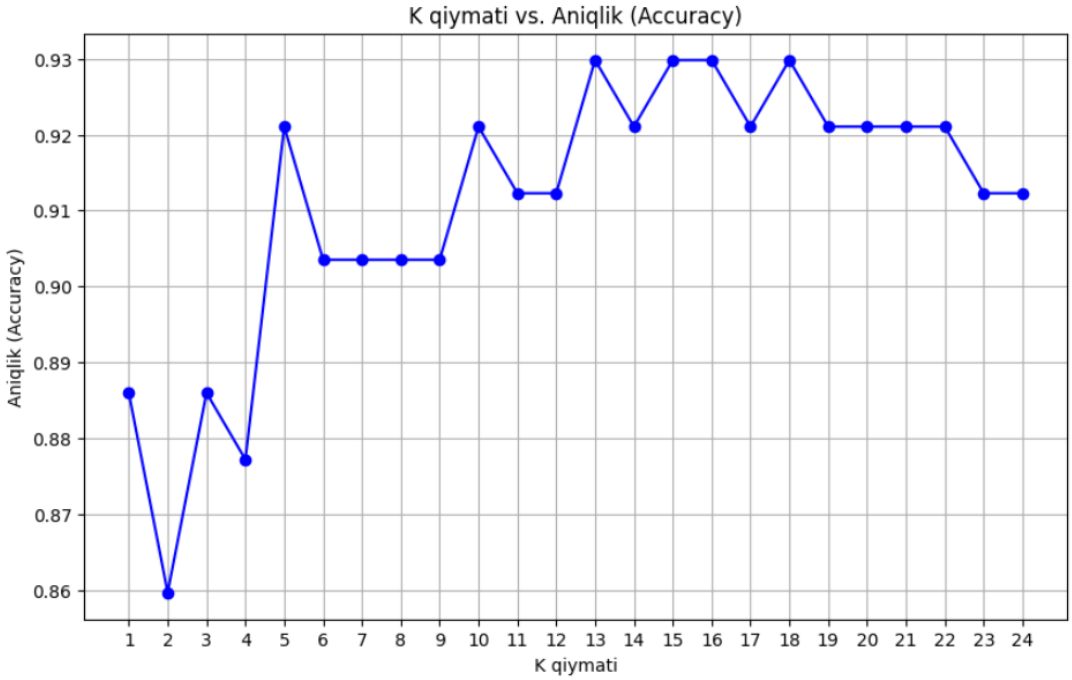
**Selecting the best k by k nearest neighbors method**

One of the most important parameters in the k-NN algorithm is the value of k. That is, the number of neighbors is important in determining whether a new object belongs to a certain class.

It is possible to find the best value of k by calculating the values of f1_score or accuracy_score determined using the evaluation criterion seen above, by calculating at which value of k of the model the model achieves the best efficiency.

```
f11=[]
for k in range(1,25):
 knn=KNeighborsClassifier(n_neighbors=k) # k-ni qiymati
 knn.fit(X_train, y_train)
 y_pridect1=knn.predict(X_test)
 f11.append(accuracy_score(y_test,y_pridect1))
plt.figure(figsize=(10,6))
plt.plot(range(1,25),f11,marker='o‘, linestyle='-', color='b')
plt.xticks(range(1,25))
plt.title('K qiymati vs. Aniqlik (Accuracy)')
plt.xlabel('K qiymati')
plt.ylabel('Aniqlik (Accuracy)')
plt.grid()
plt.show()
```

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135




As can be seen from the graph, the model achieves the highest accuracy at values k={13,19}.

## REFERENCES

1. Amrullayevich K. A., Obid o'g'li S. J. ELEKTRON TALIM MUHITIDA TALABALARDA AXBOROT BILAN ISHLASH KOMPETENTLIKNI SHAKLLANTIRISH //International Journal of Contemporary Scientific and Technical Research. – 2022. – С. 641-645.
2. Obid o'g A. S. J. et al. Numpy Library Capabilities. Vectorized Calculation In Numpy Va Type Of Information //Eurasian Research Bulletin. – 2022. – Т. 15. – С. 132-137.
3. Javlon X. et al. Классификатор движения рук с использованием биомиметического распознавания образов с помощью сверточных нейронных сетей с методом динамического порога для извлечения движения с использованием датчиков EF //Journal of new century innovations. – 2022. – Т. 19. – №. 6. – С. 352-357.
4. Фитратович В. и др. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МНОГОФАЗНОЙ ФИЛЬТРАЦИИ В НЕФТЕГАЗОВОМ ПЛАСТЕ ПРИ ЕГО ЗАВОДНЕНИИ //INTERNATIONAL CONFERENCES ON LEARNING AND TEACHING. – 2022. – Т. 1. – №. 4. – С. 520-525.
5. Jamshid S. ENTROPY EVALUATION CRITERION IN DECISION TREE ALGORITHM EVALUATION //International Journal of Contemporary Scientific and Technical Research. – 2023. – С. 236-239.
6. Салимов Ж., Абулаева А. Классификации дерева в машинном обучении и гиперпараметрах //Информатика и

International Journal of Advance Scientific Research
(ISSN – 2750-1396)
VOLUME 03 ISSUE 12 Pages: 18-25
SJIF IMPACT FACTOR (2021: 5.478) (2022: 5.636) (2023: 6.741)
OCLC – 1368736135

инженерные технологии. – 2023. – Т. 1. – №. 1. – С. 71-79.

7. Obid o'g'li S. J., Nodir o'g'li X. A., Jasurjonovich B. J. SUPERVISED LEARNING REGRESSION ALGORITHM SIMPLE LINEAR REGRESSION //Academia Science Repository. – 2023. – Т. 4. – №. 04. – С. 69-76.