Hyperparameter Tuning of KNN Classifier

Saurav Agrawal · Follow

3 min read · Jun 4, 2023

15 1

Photo By TruShotz On Pexels

K-Nearest Neighbor Classifier is a machine learning algorithm used for classification and regression. It works by finding the K nearest points in the training dataset and uses their class to predict the class or value of a new data point.

Need for finding the correct ‘k’ parameter—

> *Larger k= less complex model = can lead to underfitting*
>
> *Smaller k= more complex model = can lead to overfitting*

In the following section we try to fit the KNN model with the correct K.

1. Loading several libraries that will be used in the analysis —

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

2. Importing the dataset and checking for null values —

```
df = pd.read_csv("/kaggle/input/pima-indians-diabetes-database/diabetes.csv")
print(df.head())
print(df.shape)
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
(768, 9)
```

Dataset Head and Shape

```
print(df.isna().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

0 Null values in the dataset

3. Splitting the dataset —

```
X = df.drop("Outcome", axis=1).values
y = df["Outcome"].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(614, 8)
(154, 8)
(614,)
(154,)
```

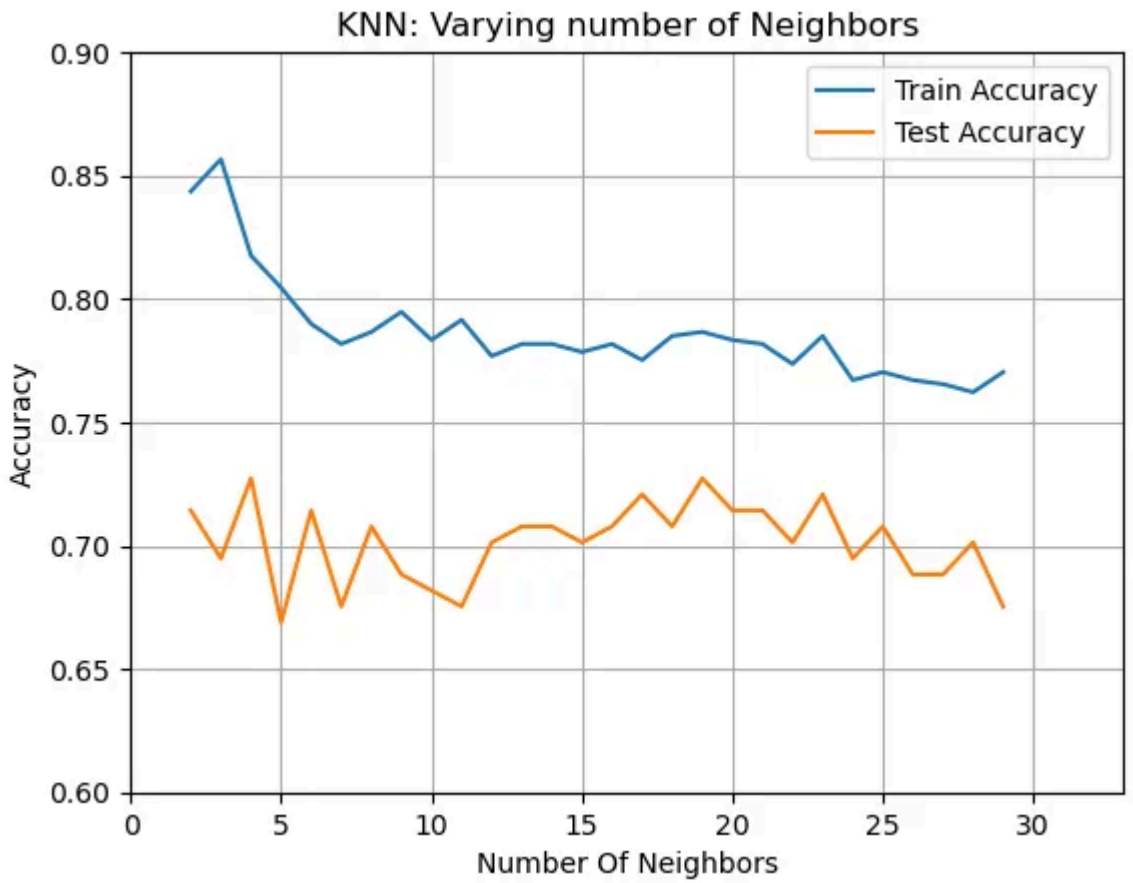
Train and Test Dataset Shape

4. Hyperparameter Tuning the K parameter —

```
train_score = {}
test_score = {}
n_neighbors = np.arange(2, 30, 1)
for neighbor in n_neighbors:
    knn = KNeighborsClassifier(n_neighbors=neighbor)
    knn.fit(X_train, y_train)
    train_score[neighbor]=knn.score(X_train, y_train)
    test_score[neighbor]=knn.score(X_test, y_test)
```

5. Plotting the Train Score and Test Score —


```
plt.plot(n_neighbors, test_score.values(), label="Test Accuracy")
plt.xlabel("Number Of Neighbors")
plt.ylabel("Accuracy")
plt.title("KNN: Varying number of Neighbors")
plt.legend()
plt.xlim(0, 33)
plt.ylim(0.60, 0.90)
plt.grid()
plt.show()
```



Plotting the Train and Test Accuracy on different number of Neighbors.

6. Finding the best param ‘k’ —

```
for key, value in test_score.items():
    if value==max(test_score.values()):
        print(key)
```

```
4
19
```

‘K’ Neighbors with Best Test score

As we can see, n_neighbors shows best accuracy score for 4 neighbors and 19 neighbors. **But the K-NearestNeighbor Classifier will perform differently based on how data is split. This requires Hyperparameter Tuning using GridSearchCV. We can use RandomizedSearchCV too.**

7. Using GridSearchCV to determine the best_params_ for n_neighbors —

```
kf=KFold(n_splits=5,shuffle=True,random_state=42)
parameter={'n_neighbors': np.arange(2, 30, 1)}
knn=KNeighborsClassifier()
knn_cv=GridSearchCV(knn, param_grid=parameter, cv=kf, verbose=1)
knn_cv.fit(X_train, y_train)
print(knn_cv.best_params_)
```

```
Fitting 5 folds for each of 28 candidates, totalling 140 fits
{'n_neighbors': 23}
```

Finding best n_neighbors parameter

8. Using the best parameter from Step 7. to train the model and predict —

```
knn=KNeighborsClassifier(n_neighbors=23)
knn.fit(X_train, y_train)
y_pred=knn.predict(X_test)
accuracy_score=accuracy_score(y_test, y_pred)*100
print("Accuracy for testing dataset after tuning : {:.2f}%".format(accuracy_scor
```

```
Accuracy for testing dataset after tuning : 72.08%
```

Output of KNN model after hyperparameter tuning

In this article, we tried to find the best n_neighbor parameter by plotting the test accuracy score based on one specific subset of dataset. Then we took into account that the model performance changes based on how we split the dataset into train and test dataset. We then used the cross validation to test the model on the entire dataset using GridSearchCV.

I appreciate you and the time you took out of your day to read this!

Linkedin: https://www.linkedin.com/in/saurav-agrawal-137500214/

StackOverFlow: https://stackoverflow.com/users/11842006/saurav-agrawal

Email: agrawalsam1997@gmail.com

Machine Learning | Knn | Knn Algorithm | Gridsearchcv | Crossvalidation

Written by Saurav Agrawal

21 Followers · 18 Following

3x AWS Certified. Data Engineering, Machine Learning, Stocks and Finance. Buy me a coffee at https://www.buymeacoffee.com/SauravAgrawal

Follow

Responses (1)

What are your thoughts?

Respond

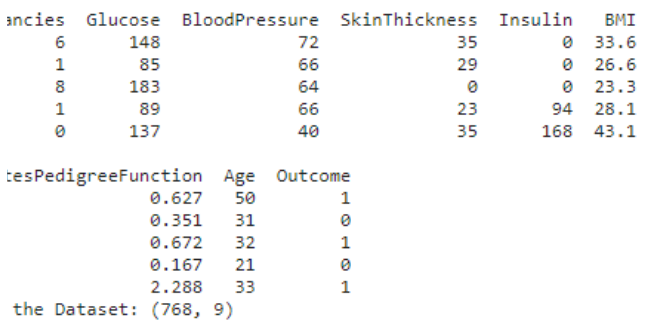
V Valanju
10 months ago

very nicely explained !!

Learnt a lot from it

Reply

More from Saurav Agrawal



Saurav Agrawal

Feature Selection Using Lasso Regression

Lasso Regression is a regularized linear regression that includes a L1 penalty. Lasso...

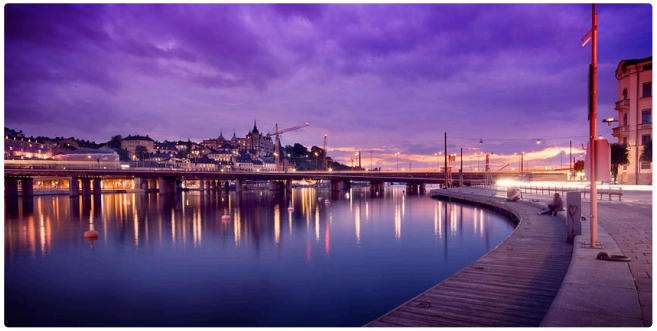
Jun 5, 2023 28 2

Saurav Agrawal

Multiclass Classification: OneVsRest and OneVsOne...

Disclaimer: Multiclass classification is supported with every classifier in scikit-lear...

Jun 22, 2023 15 1



Saurav Agrawal

Recursive SQL CTE for Hierarchical Data

In AWS Tip by Saurav Agrawal

S3 Control Create Job Invalid Request Error

Suppose you have a relational database having parent-child relationship or tree like...

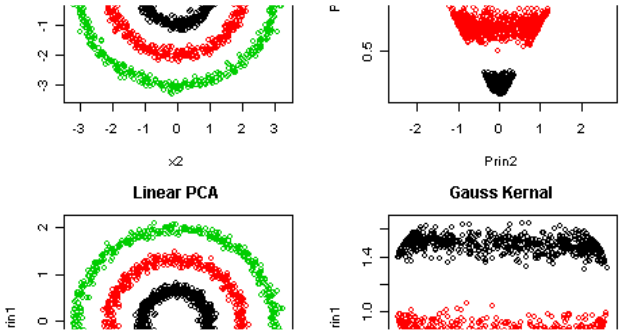
Jul 16, 2024 3

Very often when trying to create S3 Batch Operations job from boto3 we receive the...

Dec 21, 2022 40 1

See all from Saurav Agrawal

Recommended from Medium



Chanaka Prasanna

Converting Categorical Data into Numerical Form in Machine...

Categorical data refers to variables that represent categories or labels rather than...

Aug 10, 2024 52 2

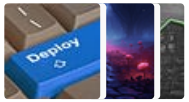
Avicsebooks

Part17: Unsupervised Machine Learning: Kernel Principal...

Kernel PCA: An In-Depth Explanation

Jul 7, 2024

Lists



Predictive Modeling w/ Python

20 stories · 1751 saves

Practical Guides to Machine Learning

10 stories · 2125 saves

Natural Language Processing

1882 stories · 1517 saves

The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 534 saves

7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29

Rohollah

Mastering Feature Selection: Key Applications and Differences—...

Applications of Chi-Square in Feature Selection.How to use Chie-square for featur...

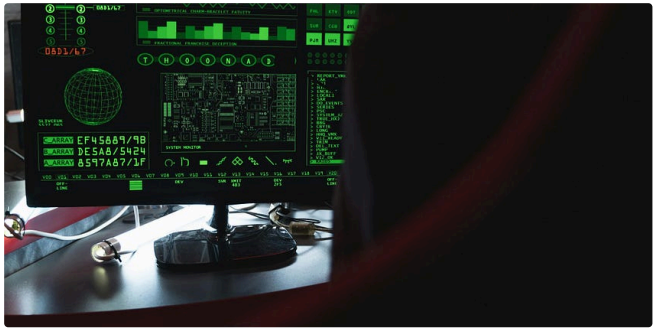
Sep 6, 2024 76

Srishti Sawla

Scikit-Learn Pipelines

In Machine Learning, ensuring an optimized, repeatable, and easy-to-maintain workflow i...

Aug 20, 2024



Ali Raza

Exploring the Weighted K-Nearest Neighbor (K-NN) Algorithm

The Weighted K-Nearest Neighbor (K-NN) algorithm is a refinement of the classic K-N...

19h ago

In Towards Data Science by Dr. Theophano Mitsa

A Guide to 21 Feature Importance Methods and Packages in Machin...

From the OmniXAI, Shapash, and Dalex interpretability packages to the Boruta, Reli...

Dec 19, 2023 422 4