

You're a data scientist and also a data engineer with 10 year's experience. I have studied K-Nearest Neighbor algorithm today. Create a dataset to me with features,label and unseen data so I can apply K-Nearest Neighbor algorithm to classify unseen data and evaluate model performance



📌 Bellek güncellendi

Here's a dataset for you to apply the K-Nearest Neighbor algorithm. The dataset consists of three features and a binary label (0 or 1). Additionally, I've included some unseen data for classification.

I'll generate it now.

It looks like there was a connection issue while generating the dataset. Let me try again. 🔄

There was another connection issue while generating the dataset. Let me troubleshoot it and try again. 🔄

The issue occurred because the kernel restarted, which cleared all variables and imports. I'll redefine everything and generate the dataset again. 🔄

It seems like I can't do more advanced data analysis right now. Please try again later. However, you can generate a dataset yourself using the following Python code:

python

📋 Kopyala

✎ Düzenle

```
import numpy as np import pandas as pd from sklearn.model_selection import train_test_split
# Set seed for reproducibility np.random.seed(42) # Generate synthetic dataset num_samples
= 100 # Features: Random values between 0 and 10 X = np.random.uniform(0, 10, (num_samples,
3)) # Labels: Binary classification based on a decision boundary y = (X[:, 0] + X[:, 1] >
10).astype(int) # Split data into training and testing sets X_train, X_test, y_train,
y_test = train_test_split(X, y, test_size=0.2, random_state=42) # Create unseen data points
unseen_data = np.array([[2, 8, 5], [7, 1, 4], [5, 6, 2]]) # Convert to DataFrame for better
visualization train_df = pd.DataFrame(X_train, columns=['Feature1', 'Feature2',
'Feature3']) train_df['Label'] = y_train test_df = pd.DataFrame(X_test, columns=
['Feature1', 'Feature2', 'Feature3']) test_df['Label'] = y_test unseen_df =
pd.DataFrame(unseen_data, columns=['Feature1', 'Feature2', 'Feature3']) # Display dataset
samples print("Training Data Sample:") print(train_df.head()) print("\nTesting Data
Sample:") print(test_df.head()) print("\nUnseen Data:") print(unseen_df)
```

This will create a dataset with three features and a binary label, as well as unseen data points for classification. Let me know if you need help with the KNN implementation!