recover the basic example of Nearest neigbor

Yes. The basic **K-Nearest Neighbors (KNN)** example we worked on was the **student pass/fail** example.

Basic KNN example

Suppose we want to predict whether a student will **Pass (1)** or **Fail (0)** based on:

- **Hours studied**
- **Assignments completed**

Hours	Assignments	Result
1	2	Fail (0)
2	1	Fail (0)
2	3	Fail (0)
3	2	Fail (0)
7	8	Pass (1)
8	7	Pass (1)
9	8	Pass (1)
8	9	Pass (1)

Now we have a **new student**:

$$X = (6, 7)$$

We want to predict whether this student will pass.

Step 1: Calculate distances

Using Euclidean distance:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

For example, distance from `(6,7)` to `(7,8)`:

$$d = \sqrt{(6-7)^2 + (7-8)^2}$$

$$d = \sqrt{1+1} = \sqrt{2} \approx 1.41$$

The closest points are:

1. `(7,8)` → **Pass**
2. `(8,7)` → **Pass**
3. `(8,9)` → **Pass**

Step 2: Choose K

Let:

$$K = 3$$

So we look at the **3 nearest neighbors**.

They are:

```
(7,8) → Pass
(8,7) → Pass
(8,9) → Pass
```

Step 3: Majority vote

```
Pass: 3
Fail:  0
```

Therefore:

$$\boxed{\text{Prediction} = \text{Pass}}$$

The key idea

KNN is basically:

New point → Calculate distances → Find K closest points → Majority vote

For this example:

$$(6,7) \rightarrow \boxed{\text{Pass}}$$

If you're teaching this in class, this is a very good **first KNN example** because students can calculate everything by hand before seeing the Python implementation.

Fuentes

and the example of seeting the price of your house with K-NN

Yes — the **house-price KNN example** is a better example for explaining that KNN can be used for **regression**, not only classification.

KNN Regression: Predicting a House Price

Suppose we have the following houses:

House	Size (sq ft)	Bedrooms	Price
A	1,000	2	$200,000
B	1,200	2	$220,000
C	1,500	3	$280,000
D	1,600	3	$300,000
E	2,000	4	$360,000

We want to estimate the price of a new house:

$$X = (1,550 \text{ sq ft}, 3 \text{ bedrooms})$$

1. Find the nearest houses

Using the house characteristics, the closest houses are likely:

- House C: 1,500 sq ft, 3 bedrooms → **$280,000**
- House D: 1,600 sq ft, 3 bedrooms → **$300,000**
- House E: 2,000 sq ft, 4 bedrooms → **$360,000**

Suppose we choose:

$$K = 3$$

2. Average their prices

Unlike classification, where we use a **majority vote**, KNN regression takes the **average** of the neighbors' values.

$$\text{Predicted Price} = \frac{280,000 + 300,000 + 360,000}{3}$$

$$= \frac{940,000}{3}$$

$$\boxed{\$313,333}$$

So KNN predicts that the new house is worth approximately:

$$\boxed{\$313,333}$$

The important difference

KNN Classification

> Look at the nearest neighbors → **majority vote**

Example:

```
Neighbor 1 → Pass
Neighbor 2 → Pass
Neighbor 3 → Fail

Prediction → Pass
```

KNN Regression

> Look at the nearest neighbors → **average their values**

```
Neighbor 1 → $280,000
Neighbor 2 → $300,000
Neighbor 3 → $360,000

Prediction → $313,333
```

Simple way to explain it to students

> **"Tell me what the houses most similar to my house sold for, and I will use those houses to estimate my house's price."**

That's the intuition behind **K-Nearest Neighbors regression**.

generate the code in python

Here is a very basic Python example using **KNN regression** to predict the price of a house based on its **size and number of bedrooms**:

Python

Ejecutar

```python
from sklearn.neighbors import KNeighborsRegressor

# Training data
# [size in sq ft, number of bedrooms]
X = [
    [1000, 2],
    [1200, 2],
    [1500, 3],
    [1600, 3],
    [2000, 4]
]

# House prices
y = [
    200000,
    220000,
    280000,
    300000,
    360000
]

# Create the KNN regression model
knn = KNeighborsRegressor(n_neighbors=3)

# Train the model
knn.fit(X, y)

# New house: 1550 sq ft, 3 bedrooms
new_house = [[1550, 3]]

# Predict the price
prediction = knn.predict(new_house)
```

```
print("Predicted house price: $", prediction[0])
```

Expected result

Approximately:

```
Predicted house price: $ 313333.33
```

What is happening?

The important line is:

Python Ejecutar

```
knn = KNeighborsRegressor(n_neighbors=3)
```

This tells Python:

> **Use the 3 houses most similar to the new house.**

Then:

Python Ejecutar

```
prediction = knn.predict([[1550, 3]])
```

asks:

> **What is the predicted price of a 1,550 sq-ft house with 3 bedrooms?**

KNN finds the **3 nearest houses** and averages their prices.

$$\frac{280000 + 300000 + 360000}{3} = 313333.33$$

So:

$$\boxed{\text{Predicted price} \approx \$313,333}$$

One important teaching point: in a real KNN model, the features should usually be **standardized** because square footage and number of bedrooms are on very different scales. This simple example intentionally leaves that out so students can see the basic idea first.

provide the code for the classification example too

Absolutely. Here is the same **simple KNN classification example** using the student **Pass/Fail** problem.

```
from sklearn.neighbors import KNeighborsClassifier

# Training data
# [hours studied, assignments completed]
X = [
    [1, 2],
    [2, 1],
    [2, 3],
    [3, 2],
    [7, 8],
    [8, 7],
    [9, 8],
    [8, 9]
]

# Results
# 0 = Fail, 1 = Pass
y = [
    0,
    0,
    0,
    0,
    1,
    1,
    1,
    1
]

# Create the KNN classification model
knn = KNeighborsClassifier(n_neighbors=3)

# Train the model
knn.fit(X, y)
```

```python
# New student
# 6 hours studied, 7 assignments completed
new_student = [[6, 7]]

# Predict
prediction = knn.predict(new_student)

# Display result
if prediction[0] == 1:
    print("Prediction: Pass")
else:
    print("Prediction: Fail")
```

Expected output

```
Prediction: Pass
```

What KNN is doing

We have a new student:

$$(6, 7)$$

KNN looks for the **3 closest students**:

```
(7,8) → Pass
(8,7) → Pass
(8,9) → Pass
```

The majority is:

```
Pass = 3
Fail = 0
```

Therefore:

$$\boxed{\text{Prediction} = \text{Pass}}$$

Classification vs. Regression

The two examples are useful together:

Problem	KNN model	What do we do with neighbors?
Student → Pass/Fail	`KNeighborsClassifier`	**Majority vote**
House → Price	`KNeighborsRegressor`	**Average values**

The key conceptual difference for students is:

Classification → vote

Regression → average