

# Calculating Classification Probability

## Calculating Classification Probability with KNN

### Function Signature

```
def classify_probability(self, query_point, expected_val):
```

### Parameters

- **query\_point**: The point for which classification probability is to be determined.
- **expected\_val**: The label value for which the probability is to be calculated.

### Return Value

Returns the probability (in percentage) that the `query_point` belongs to the class specified by `expected_val` based on the stored values in the KNN instance.

### Description

The `classify_probability` function calculates the probability that a given `query_point` belongs to the class specified by `expected_val`. It first retrieves the nearest neighbors of the `query_point` using the

`classify_neighbors` function. It then counts how many of these neighbors have the label `expected_val` and calculates the probability based on this count.

It's important to note that the `store_vals` function must be called prior to using the `classify_probability` function to ensure that the necessary values are stored in the KNN instance.

# Examples

```
from deeprai.models import KNN

# Sample data
x_vals = [[1, 2], [2, 3], [3, 4]]
y_vals = [0, 1, 0]
query_point = [2, 2]

# Create an instance of the classifier
classifier = KNN()

# Store the values in the classifier
classifier.store_vals(x_vals, y_vals, p=3, k=2)

# Calculate the probability that the query_point belongs to class 1
probability = classifier.classify_probability(query_point, 1)
print(f"The probability that the query point belongs to class 1 is {probability}%")
```

---

Revision #1

Created 12 October 2023 05:06:20 by Kieran Carter

Updated 12 October 2023 05:08:30 by Kieran Carter