Exercise 6

1/1 point (graded)

The dataset remains stored as `data` . Because most wines in the dataset are classified as low quality, one very simple classification rule is to predict that all wines are of low quality. In this exercise, we determine the accuracy of this simple rule.
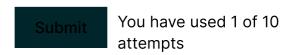
Instructions

Use `accuracy()` to calculate how many wines in the dataset are of low quality. Do this by using 0 as the first argument, and `data["high_quality"]` as the second argument.

Print your result.

What proportion of wines in the dataset are of low quality?

| 36.69385870401724 | ✔ |

36.69385870401724

| Submit | You have used 1 of 10 attempts |

✔ Correct (1/1 point)

Exercise 7

1/1 point (graded)

In Exercise 7, we will use the kNN classifier from `scikit-learn` to predict the quality of wines in our dataset.
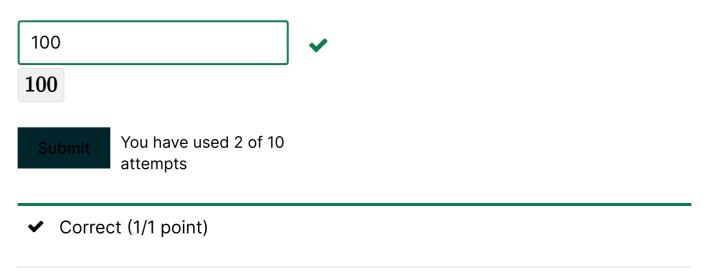
Instructions

- Use `knn.predict(numeric_data)` to predict which wines are high and low quality and store the result as `library_predictions`.

- Use `accuracy` to find the accuracy of your predictions, using `library_predictions` as the first argument and `data["high_quality"]` as the second argument.

- Print your answer. Is this prediction better than the simple classifier in Exercise 6?

Here's the sample code to get you started:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 5)
knn.fit(numeric_data, data['high_quality'])
# Enter your code here!
```

What is the accuracy of the KNN classifier? Please round your answer to the nearest integer.

```
100
```
✔

100

Submit   You have used 2 of 10 attempts
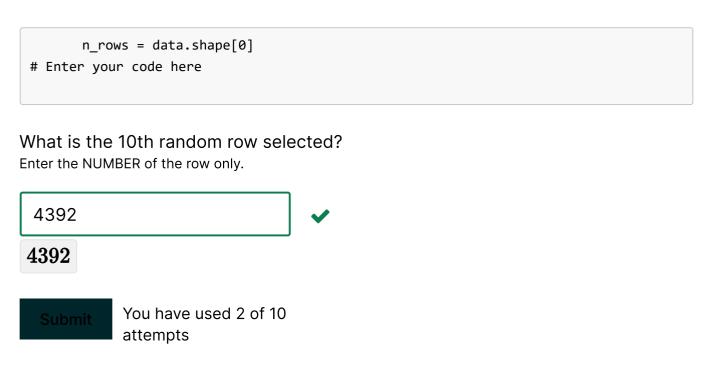
✔ Correct (1/1 point)

Exercise 8

1/1 point (graded)
Unlike the `scikit-learn` function, our homemade kNN classifier does not take any shortcuts in calculating which neighbors are closest to each observation, so it is likely too slow to carry out on the whole dataset. In this exercise, we will select a subset of our data to use in our homemade kNN classifier.

Instructions

Fix the random generator using `random.seed(123)`, and select 10 rows from the dataset using `random.sample(range(n_rows), 10)`. Store this selection as `selection`.

Use this sample code to get started:

```
    n_rows = data.shape[0]
# Enter your code here
```

What is the 10th random row selected?

Enter the NUMBER of the row only.

4392 ✔

4392

Submit   You have used 2 of 10 attempts

✔ Correct (1/1 point)