CC 5.2.1: Generating Example Classification Data

Generating Example Classification Data, Question 1

0/2 points (graded)

Implement the `gen_data()` and `plot_data()` functions from the video:

```
import scipy.stats as ss
import matplotlib.pyplot as plt


def gen_data(n, h, sd1, sd2):
   x1 = ss.norm.rvs(h, sd1, n)
   y1 = ss.norm.rvs(0, sd1, n)
   x2 = ss.norm.rvs(-h, sd2, n)
   y2 = ss.norm.rvs(0, sd2, n)
   return (x1, y1, x2, y2)


def plot_data(x1, y1, x2, y2):
   plt.figure()
   plt.plot(x1, y1, "o", ms=2)
   plt.plot(x2, y2, "o", ms=2)
   plt.xlabel("$X_1$")
   plt.ylabel("$X_2$")
```

Which of the following function calls will produce data that would be **easiest** to classify correctly?

`gen_data(1000, 0, 1, 1)`

`gen_data(1000, 1, 2, 2.5)`

`gen_data(1000, 10, 100, 100)`

`gen_data(1000, 20, .5, .5)`

correct

Which of the following function calls will produce data that would be **hardest** to classify correctly?

```
gen_data(1000, 0, 1, 1)
```

correct

```
gen_data(1000, 1, 2, 2.5)
```

```
gen_data(1000, 10, 100, 100)
```

```
gen_data(1000, 20, .5, .5)
```

CC 5.2.2: Logistic Regression

Logistic Regression, Question 1

1/1 point (graded)

What is one of the problems with using linear regression to predict probabilities?

Linear regression may predict values outside of the interval between 0 and 1.
correct

Our data only have classes and not probabilities.

Linear regression only works if the categories are ordered (e.g., small, medium, and large).

Probabilities are nonlinear.

Logistic Regression, Question 2

1/1 point (graded)

The following code creates a function that converts probability to odds:

```
def prob_to_odds(p):
    if p <= 0 or p >= 1:
        print("Probabilities must be between 0 and 1.")
    return p / (1-p)
```

Assume that there are only two classes and all data points belong to one of these two classes. The probability that a given data point belongs to Class 1 is 0.2.

What are the odds that a given data point belongs to Class 2 as given by the function above?

0.8

1

4
correct

20

CC 5.2.3: Logistic Regression in Code

Logistic Regression in Code, Question 1

2/4 points (graded)

In Video 5.2.3, we performed a series of steps using common (and important) methods on the classifier object `clf()`.

Part 1. If you have data and want to train a model, which method would you use?

`clf.predict_proba()`

`clf.fit()`

correct

`clf.estimate()`

`clf.score()`

`clf.predict()`

correct

Part 2. If you want to compute the accuracy of your model, which method would you use?

`clf.predict_proba()`

`clf.fit()`

`clf.estimate()`

`clf.score()`

correct

`clf.predict()`

Part 3. If you want to estimate the probability of a data point being in each class, which method would you use?

`clf.predict_proba()`

correct

`clf.fit()`

`clf.estimate()`

`clf.score()`

`clf.predict()`

Part 4. If you want to know to which class your model would assign a new data point, which method would you use?

`clf.predict_proba()`

`clf.fit()`

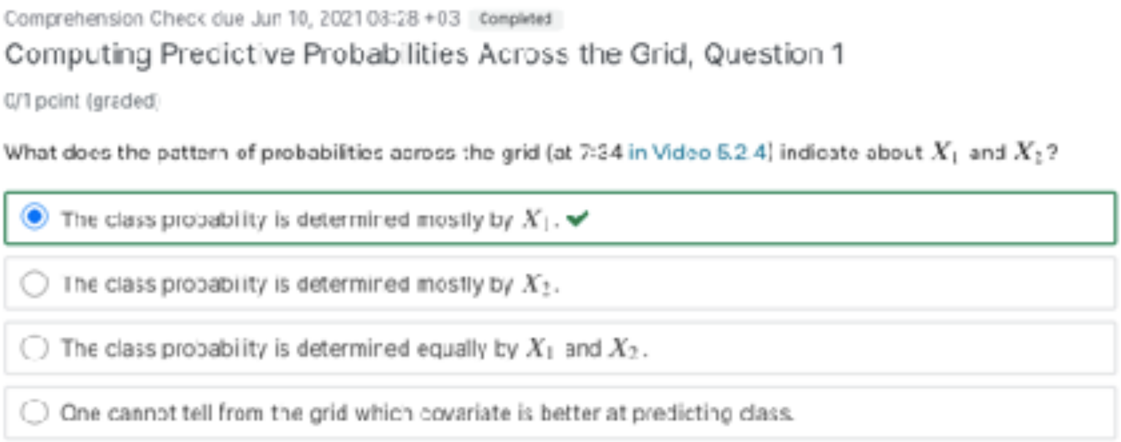
`clf.estimate()`

`clf.score()`

`clf.predict()`

correct

CC 5.2.4: Computing Predictive Probabilities Across the Grid



Comprehension Check due Jun 10, 2021 03:28 +03 Completed

Computing Predictive Probabilities Across the Grid, Question 1

0/1 point (graded)

What does the pattern of probabilities across the grid (at 7:34 in Video 5.2.4) indicate about X_1 and X_2?

● The class probability is determined mostly by X_1. ✓

○ The class probability is determined mostly by X_2.

○ The class probability is determined equally by X_1 and X_2.

○ One cannot tell from the grid which covariate is better at predicting class.

Computing Predictive Probabilities Across the Grid, Question 2

1/1 point (graded)

The sum of the class probabilities:

will only equal 1 when there are two classes.

will always equal 1 for any number of classes.
correct

is usually above 1.

is usually below 1.