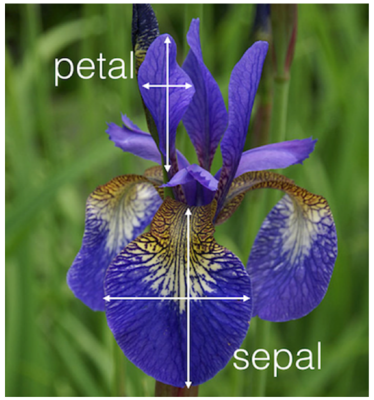
Implementation details

About Dataset



Three types of Iris:

- class 0: 'setosa’
- class 1: 'versicolor’
- class 2: 'virginica’

Four features:

- 'sepal length (cm)’,
- 'sepal width (cm)’,
- 'petal length (cm)’,
- 'petal width (cm)’

Data:

- #150 items
- …with equal portion of classes 50/50/50

Our pipeline

Normalized Dataset

	sepal length	sepal width	petal length	petal width	target
4	0.784810	0.536585	0.652174	0.60	1
35	0.632911	0.780488	0.173913	0.08	0
50	0.632911	0.804878	0.202899	0.08	0
46	0.974684	0.682927	0.971014	0.80	2
37	0.810127	0.707317	0.623188	0.52	1
52	0.721519	0.682927	0.594203	0.52	1
40	0.620253	0.585366	0.478261	0.40	1
16	0.569620	0.560976	0.188406	0.12	0

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
115	6.4	3.2	5.3	2.3	2
95	5.7	3.0	4.2	1.2	1
96	5.7	2.9	4.2	1.3	1
75	6.6	3.0	4.4	1.4	1
149	5.9	3.0	5.1	1.8	2
92	5.8	2.6	4.0	1.2	1
148	6.2	3.4	5.4	2.3	2
114	5.8	2.8	5.1	2.4	2

Original Dataset

Shuffled, Splitted

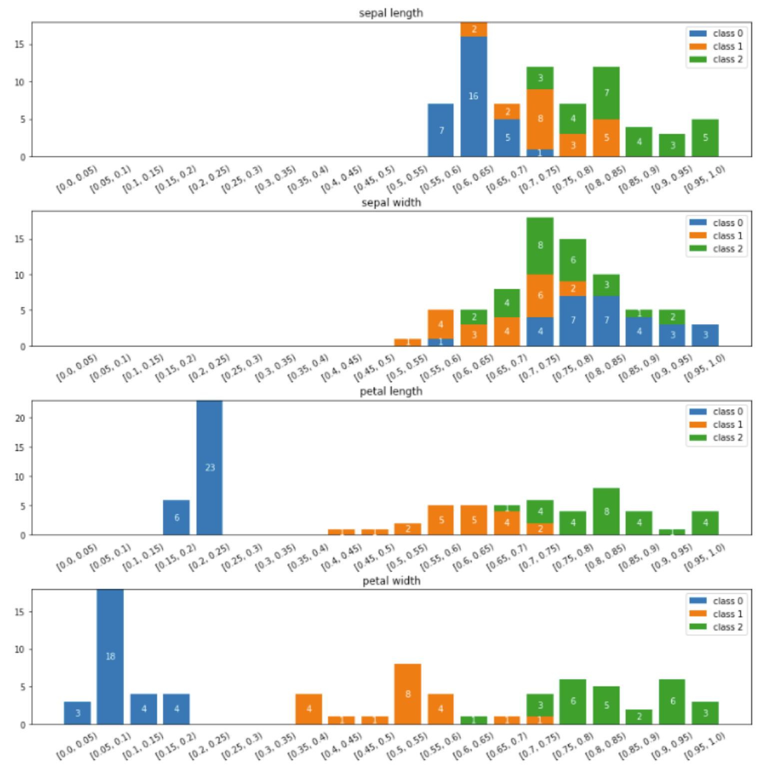
Train

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
9	4.9	3.1	1.5	0.1	0
65	6.7	3.1	4.4	1.4	1
94	5.6	2.7	4.2	1.3	1
66	5.6	3.0	4.5	1.5	1
6	4.6	3.4	1.4	0.3	0

Test

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
83	6.0	2.7	5.1	1.6	1
54	6.5	2.8	4.6	1.5	1

Our pipeline



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
115	6.4	3.2	5.3	2.3	2
95	5.7	3.0	4.2	1.2	1
96	5.7	2.9	4.2	1.3	1
75	6.6	3.0	4.4	1.4	1
149	5.9	3.0	5.1	1.8	2
92	5.8	2.6	4.0	1.2	1
148	6.2	3.4	5.4	2.3	2
114	5.8	2.8	5.1	2.4	2

Original Dataset

Shuffled, Splitted

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
9	4.9	3.1	1.5	0.1	0
65	6.7	3.1	4.4	1.4	1
94	5.6	2.7	4.2	1.3	1
66	5.6	3.0	4.5	1.5	1
6	4.6	3.4	1.4	0.3	0

Train

Normalized Dataset

	sepal length	sepal width	petal length	petal width	target
4	0.784810	0.536585	0.652174	0.60	1
35	0.632911	0.780488	0.173913	0.08	0
50	0.632911	0.804878	0.202899	0.08	0
46	0.974684	0.682927	0.971014	0.80	2
37	0.810127	0.707317	0.623188	0.52	1
52	0.721519	0.682927	0.594203	0.52	1
40	0.620253	0.585366	0.478261	0.40	1
16	0.569620	0.560976	0.188406	0.12	0

Problem statement

```python
print('params: ', iris_bc.X_test[0],'class: ', iris_bc.y_test[0])
```

```
params:  [5.8 2.8 5.1 2.4] class:  2
```

```python
iris_bc.normalize(iris_bc.X_test[0])
```

```
array([0.73417722, 0.68292683, 0.73913043, 0.96      ])
```

```python
iris_bc.predict(iris_bc.X_test[0], verbose = True)
```

```
[(0.7, 0.75), (0.65, 0.7), (0.7, 0.75), (0.95, 1.0)]
```

so, the problem statement:

What’s the probability, that **X_test[0]** belongs to **class N**, if:

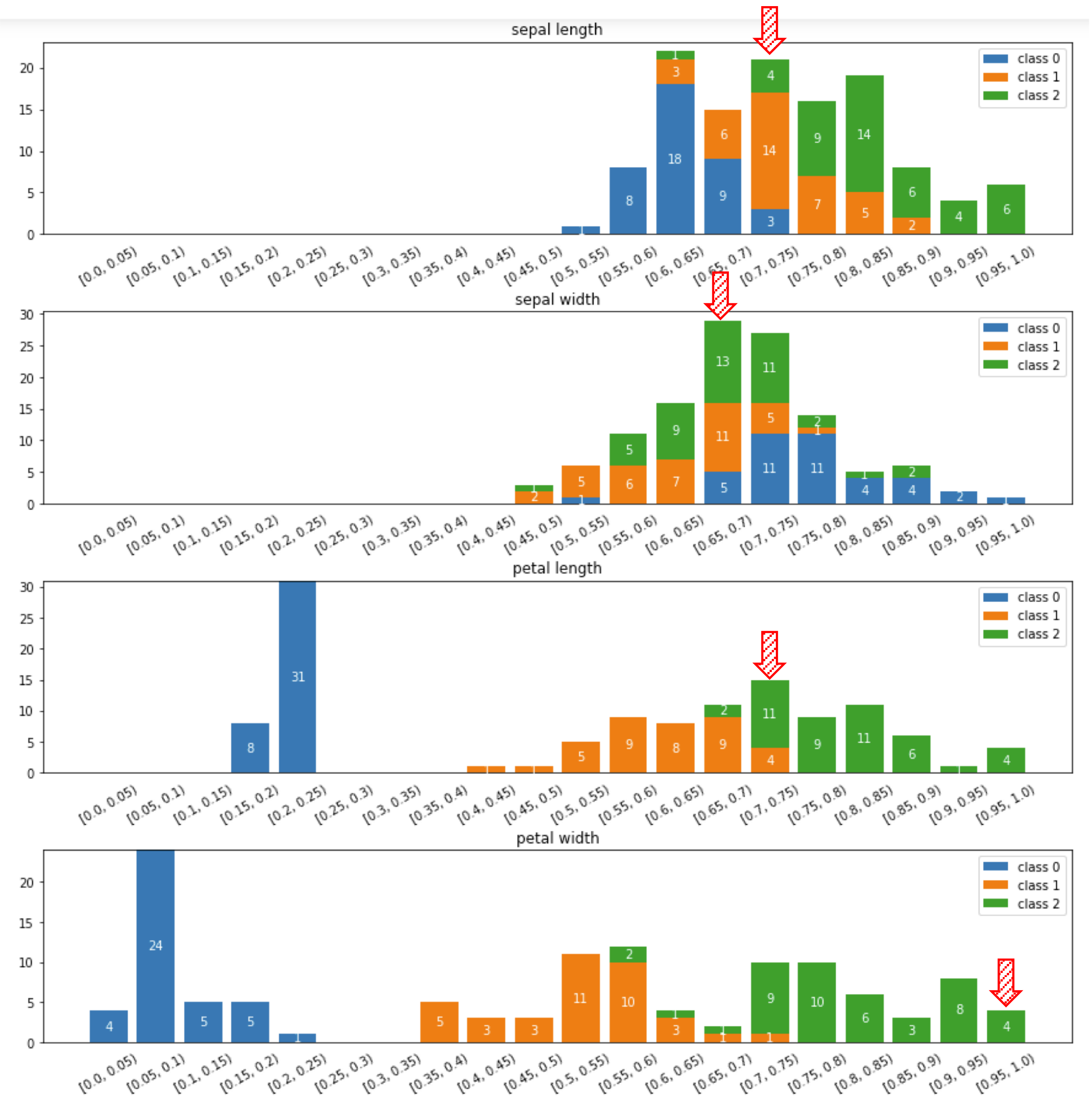
- **F1** is here **(0.7, 0.75)** and
- **F2** is here **(0.65, 0.7)** and
- **F3** is here **(0.7, 0.75)** and
- **F4** is here **(0.95, 1.0)**

Or…

$$\boldsymbol{P}\ (C = 0 \mid x1 = \mathrm{F1}, x2 = \mathrm{F2}, x3 = \mathrm{F3}, x4 = \mathrm{F4}) = ?$$

$$\boldsymbol{P}\ (C = 1 \mid x1 = \mathrm{F1}, x2 = \mathrm{F2}, x3 = \mathrm{F3}, x4 = \mathrm{F4}) = ?$$

$$\boldsymbol{P}\ (C = 2 \mid x1 = \mathrm{F1}, x2 = \mathrm{F2}, x3 = \mathrm{F3}, x4 = \mathrm{F4}) = ?$$

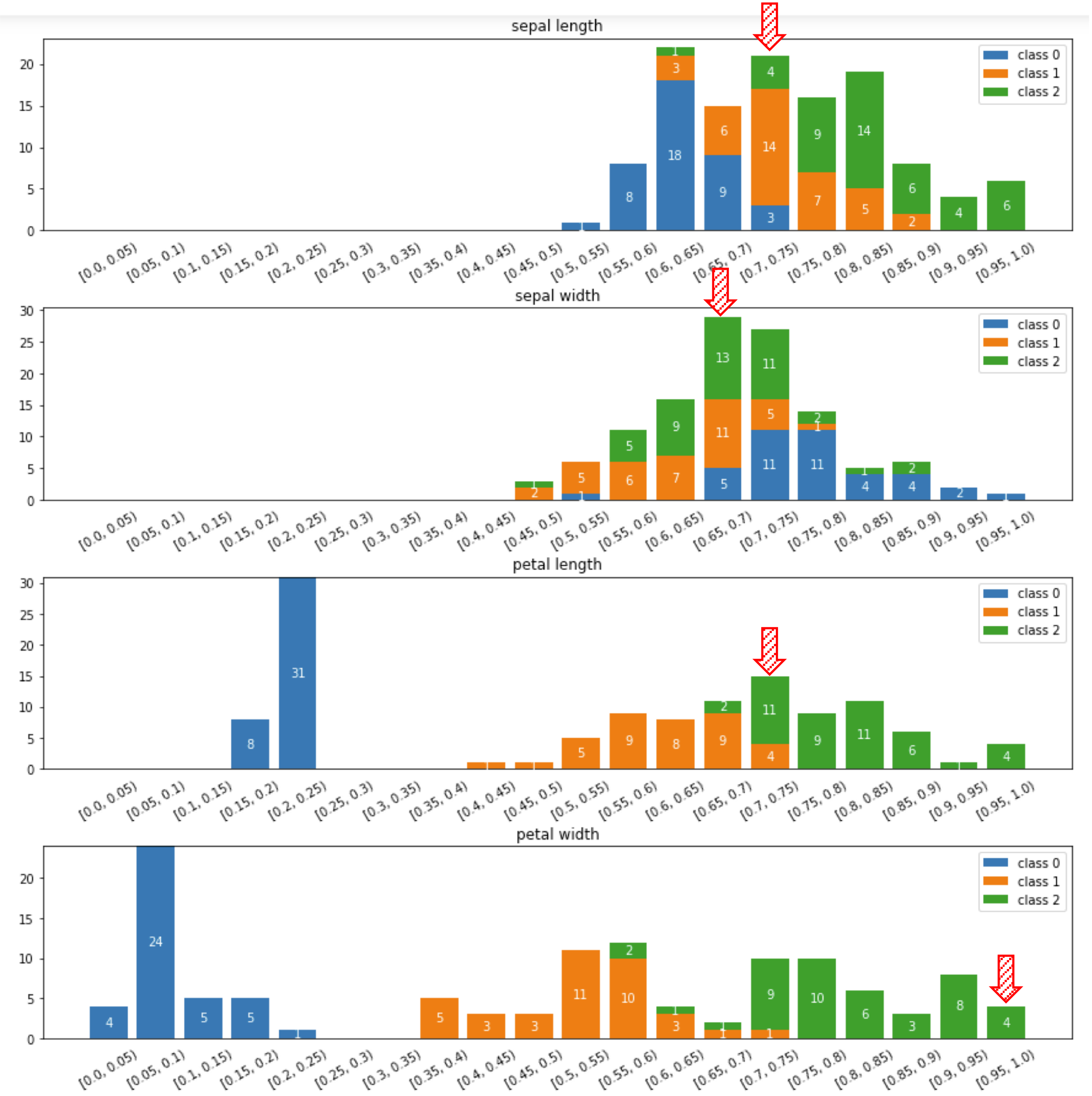


Bayes Theorem

$$\boldsymbol{P}\,(C = 0 \mid x1 = \mathrm{F1}, x2 = \mathrm{F2}, x3 = \mathrm{F3}, x4 = \mathrm{F4}) = ?$$

$$\boldsymbol{P}\,(C = 1 \mid x1 = \mathrm{F1}, x2 = \mathrm{F2}, x3 = \mathrm{F3}, x4 = \mathrm{F4}) = ?$$

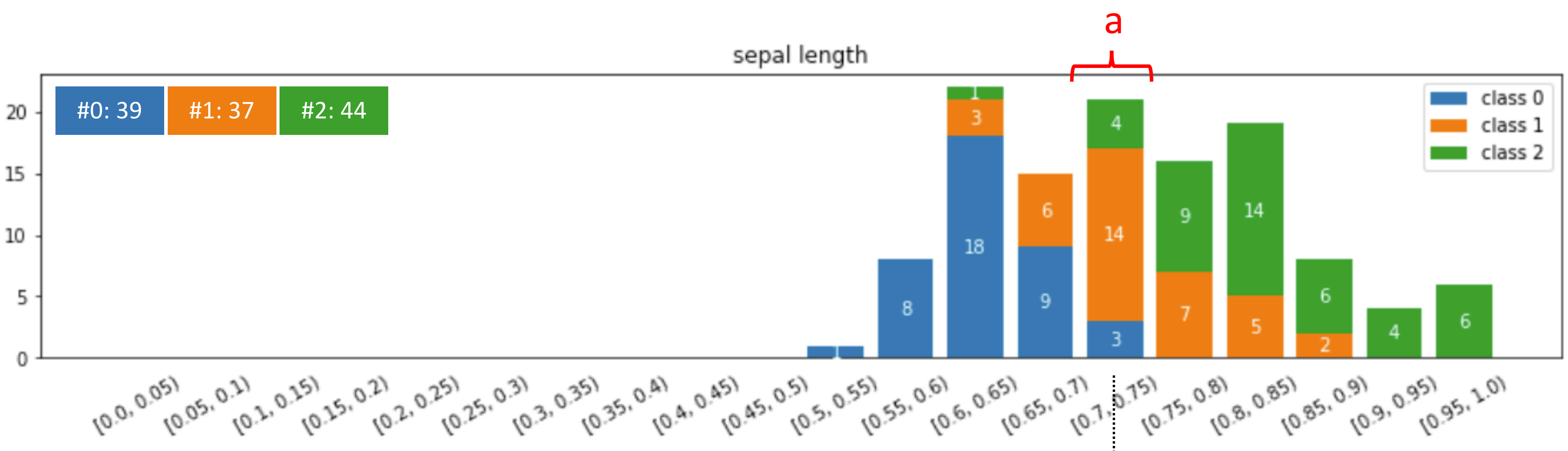
$$\boldsymbol{P}\,(C = 2 \mid x1 = \mathrm{F1}, x2 = \mathrm{F2}, x3 = \mathrm{F3}, x4 = \mathrm{F4}) = ?$$



let’s check it for more simple case for only feature:

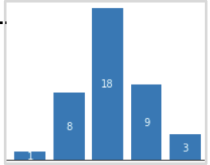
$$\boldsymbol{P}\,(C = 0 \mid x1 = \mathrm{F1}) = \ ?$$

Bayes Theorem



let’s check it for more simple case for only feature:

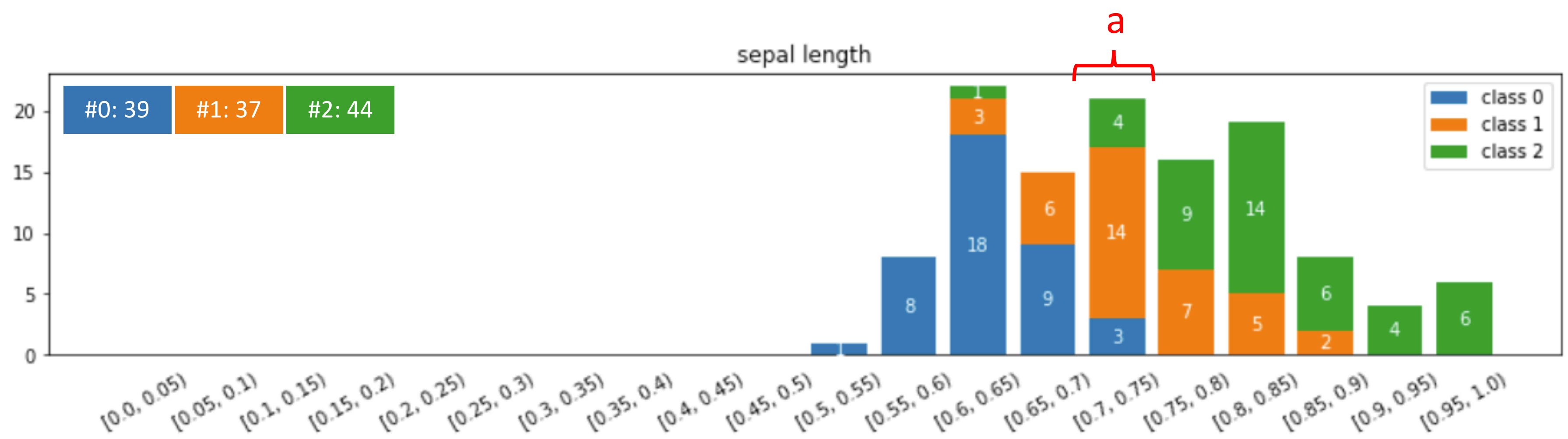
$$P(x1 = \mathbf{a} | C = 0) = \frac{N(x1 = \mathbf{a} | C = 0)}{N(C = 0)}$$



$$P(C = 0) = \frac{N(C = 0)}{N}$$

$$P\,(C = 0 \mid x1 = \mathbf{a}) = \frac{P(x1 = \mathbf{a} | C = 0) * P(C = 0)}{\sum_{i=1}^{\#Classes} P(x1 = \mathbf{a} | C = i) P(C = i)}$$

Bayes Theorem



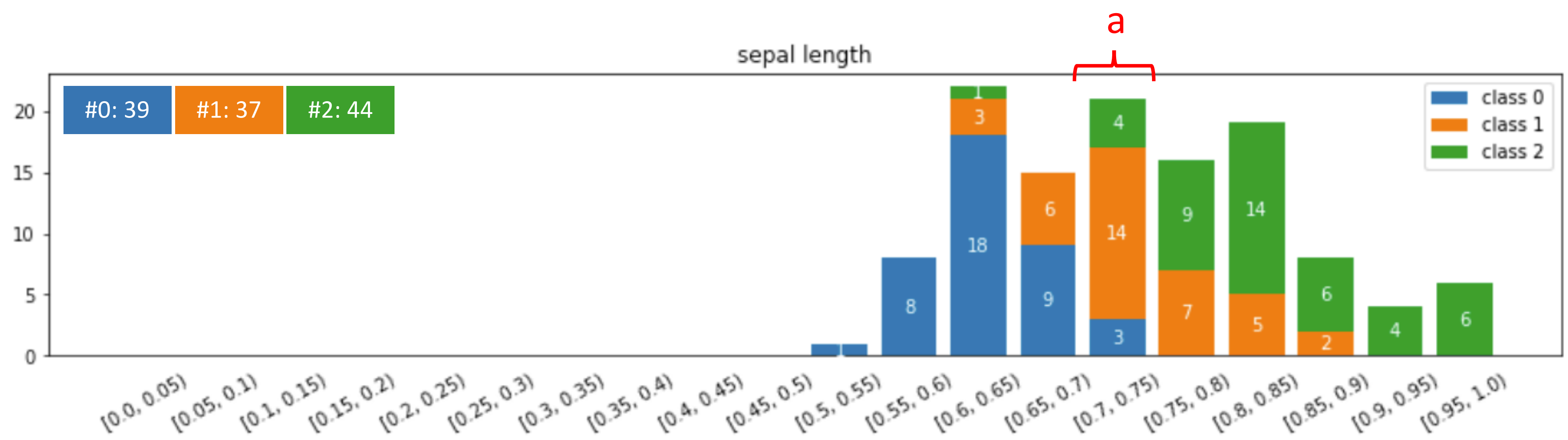
let's check it for more simple case for only feature:

$$P(x1 = \boldsymbol{a} | C = 0) = \frac{N(x1 = \boldsymbol{a} | C = 0)}{N(C = 0)} = \frac{3}{39}$$

$$P(C = 0) = \frac{N(C = 0)}{N} = \frac{39}{120}$$

$$P\ (C = 0 \mid x1 = \boldsymbol{a}) = \frac{P(x1 = \boldsymbol{a} | C = 0) * P(C = 0)}{\sum_{i=1}^{\#Classes} P(x1 = \boldsymbol{a} | C = i) P(C = i)}$$

Bayes Theorem



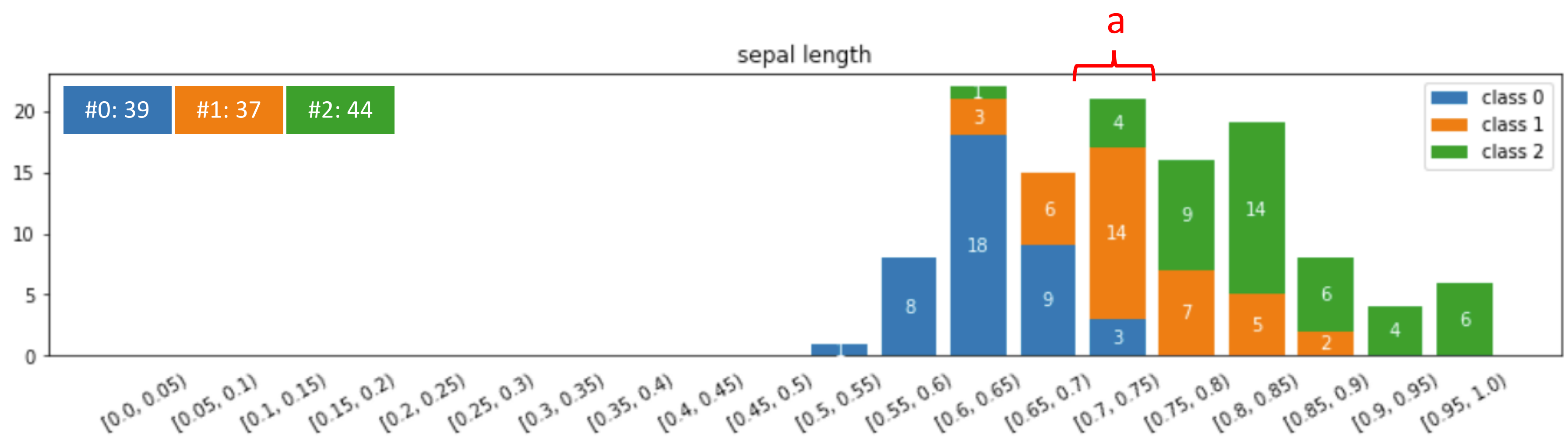
let's check it for more simple case for only feature:

$$P(x1 = \mathbf{a} | C = 0) = \frac{N(x1 = \mathbf{a} | C = 0)}{N(C = 0)} = \frac{3}{39}$$

$$P(C = 0) = \frac{N(C = 0)}{N} = \frac{39}{120}$$

$$P\,(C = 0 \mid x1 = \mathbf{a}) = \frac{P(x1 = \mathbf{a}|C = 0) * P(C = 0)}{\sum_{i=1}^{\#Classes} P(x1 = \mathbf{a}|C = i)P(C = i)} = \frac{\frac{3}{39} * \frac{39}{120}}{\frac{3}{39} * \frac{39}{120} + \frac{14}{37} * \frac{37}{120} + \frac{4}{44} * \frac{44}{120}} = \frac{3}{3 + 14 + 4} = 0{,}14$$

Bayes Theorem



Back to complex case:

```
       Class 0     Class 1     Class 2
F1  [[0.14285714 0.66666667 0.19047619]
F2   [0.001      0.4375     0.5625    ]
F3   [0.001      0.26666667 0.73333333]
F4   [0.001      0.001      1.        ]]
```

```
iris_bc.normalize(iris_bc.X_test[0])
```

```
array([0.73417722, 0.68292683, 0.73913043, 0.96      ])
```

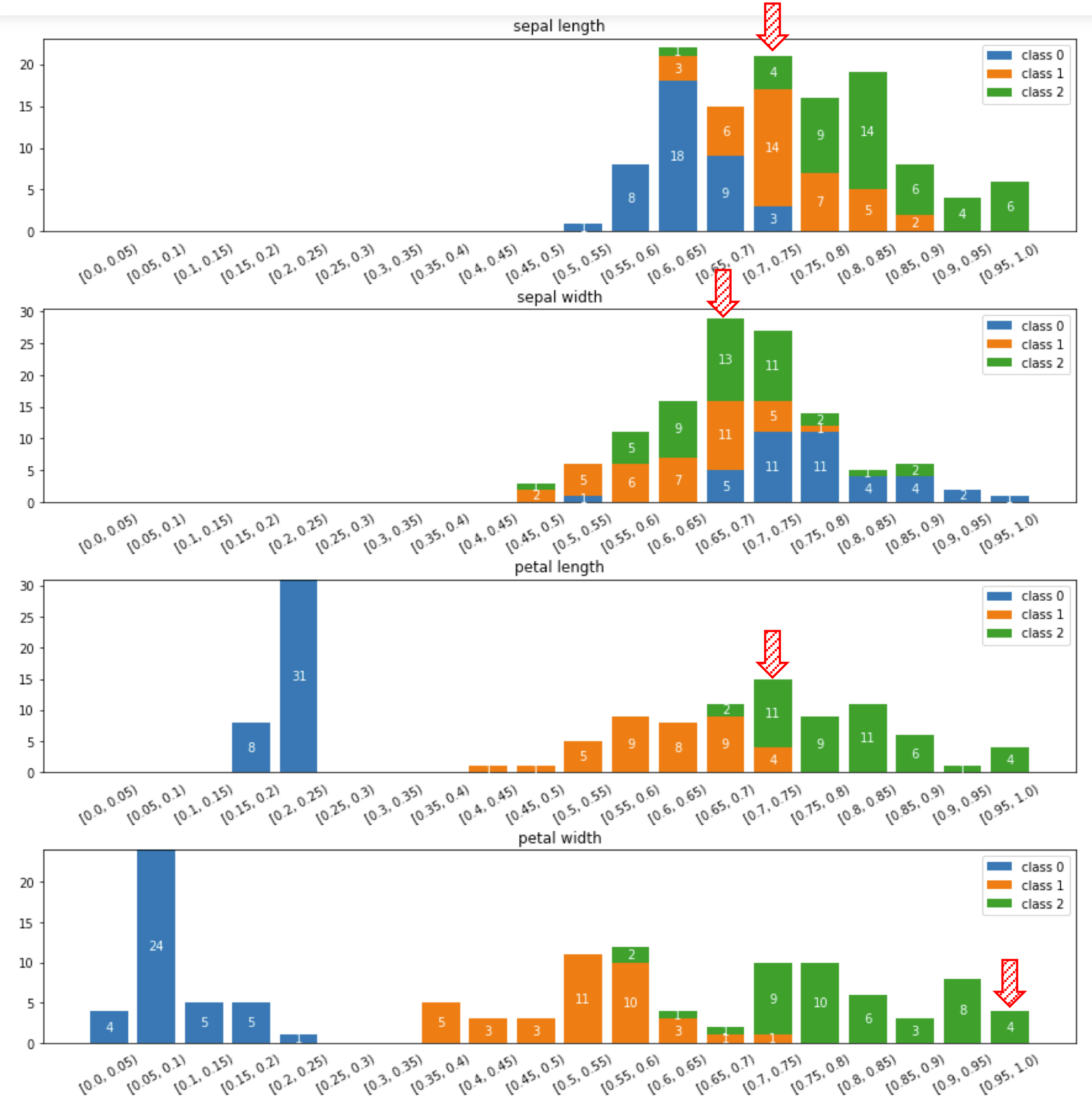
```
iris_bc.predict(iris_bc.X_test[0], verbose = True)
```

```
[(0.7, 0.75), (0.65, 0.7), (0.7, 0.75), (0.95, 1.0)]
```

	Class 0	Class 1	Class 2
F1	[[0.14285714	0.66666667	0.19047619]
F2	[0.001	0.4375	0.5625]
F3	[0.001	0.26666667	0.73333333]
F4	[0.001	0.001	1.]]

```
Predicted class: 2, Original class: 2

[1.42857143e-10, 7.77777778e-05, 7.85714286e-02]
```



```
iris_bc.normalize(iris_bc.X_test[0])
```

```
array([0.73417722, 0.68292683, 0.73913043, 0.96      ])
```

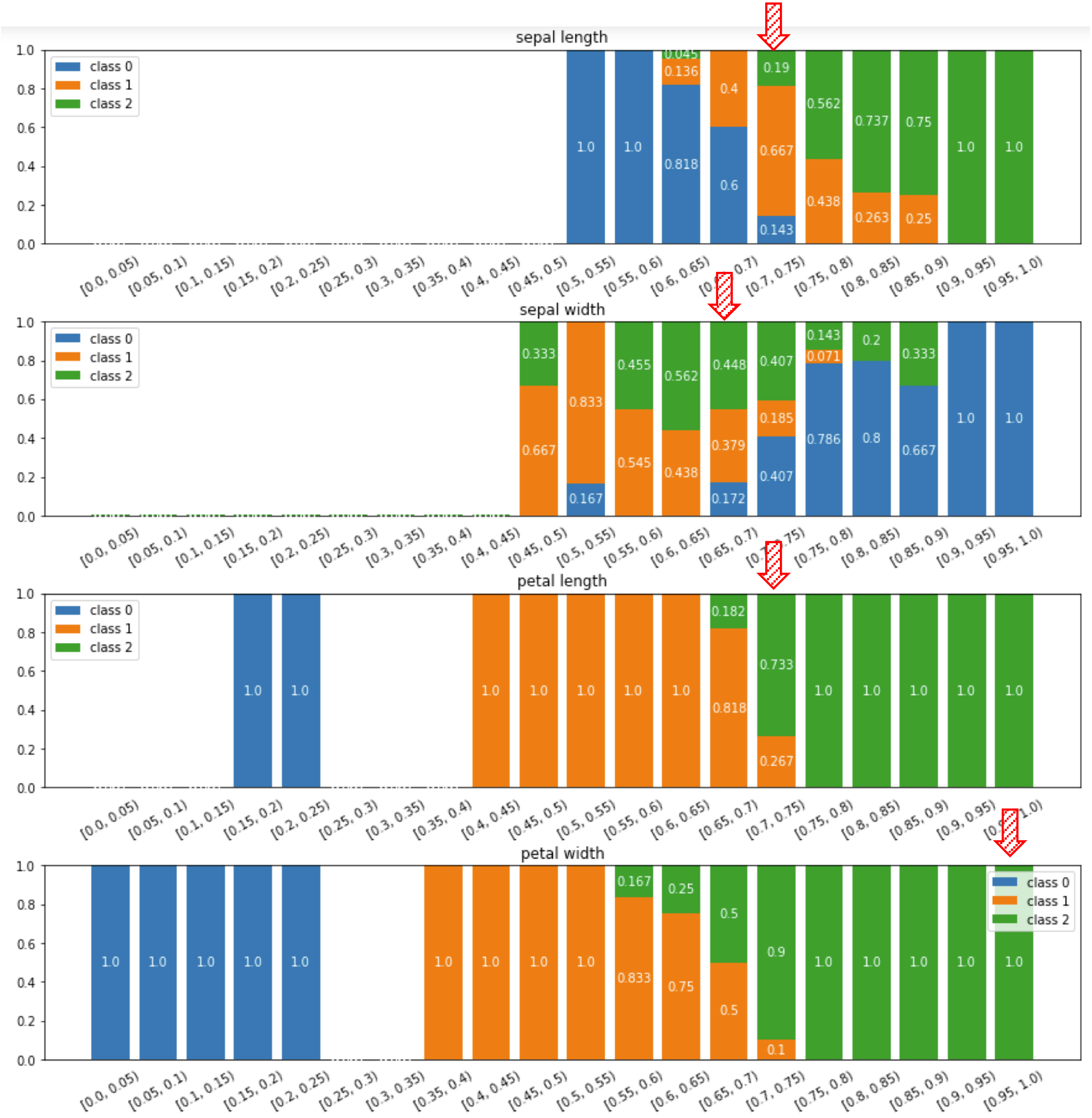
```
iris_bc.predict(iris_bc.X_test[0], verbose = True)
```

```
[(0.7, 0.75), (0.65, 0.7), (0.7, 0.75), (0.95, 1.0)]
```

	Class 0	Class 1	Class 2
F1	[[0.14285714	0.66666667	0.19047619]
F2	[0.001	0.4375	0.5625]
F3	[0.001	0.26666667	0.73333333]
F4	[0.001	0.001	1.]]

```
Predicted class: 2, Original class: 2
```

```
[1.42857143e-10, 7.77777778e-05, 7.85714286e-02]
```



Let’s go to the code…