Naman Thaker

29BCE529

DATA MINING

PRACTICAL 10 NAIVE BAYES

```
data = [
    [["Chinese", "Beijing", "Chinese"], True],
    [["Chinese", "Chinese", "Shanghai"], True],
    [["Chinese", "Macao"], True],
    [["Tokyo", "Japan", "Chinese"], False],
]

test_document = ["Chinese", "Chinese", "Chinese", "Tokyo", "Japan"]



data_unique_words = []
priors_count = 0

for i in data:
    if i[1]:
        priors_count += 1
    for q in i[0]:
        if q not in data_unique_words:
            data_unique_words.append(q)

data_unique_words = sorted(data_unique_words)
print(data_unique_words)

test_document_unique = list(set(test_document))
print(test_document_unique)

p_c = priors_count / len(data)
p_c_bar = (len(data) - priors_count) / len(data)

print(f"P(c): {p_c} and P(c'): {p_c_bar}")
```

```
['Beijing', 'Chinese', 'Japan', 'Macao', 'Shanghai', 'Tokyo']
['Japan', 'Tokyo', 'Chinese']
P(c): 0.75 and P(c'): 0.25
```

```
p_test_c = []
for i in test_document_unique:
    word_count = 0
    total_words = 0
    for j in data:
        if j[1]:
            word_count += j[0].count(i)
            total_words += len(j[0])
```

```
    p_i_c = (word_count + 1) / (total_words + len(data_unique_words))
    print(f"P({i})|c: {p_i_c}")
    p_test_c.append(p_i_c)
```

```
 P(Japan)|c: 0.07142857142857142
 P(Tokyo)|c: 0.07142857142857142
 P(Chinese)|c: 0.42857142857142855
```

```
p_test_c_bar = []
for i in test_document_unique:
    word_count = 0
    total_words = 0
    for j in data:
        if not j[1]:
            word_count += j[0].count(i)
            total_words += len(j[0])

    p_i_c_bar = (word_count + 1) / (total_words + len(data_unique_words))
    print(f"P({i})|c': {p_i_c_bar}")
    p_test_c_bar.append(p_i_c_bar)
```

```
 P(Japan)|c': 0.2222222222222222
 P(Tokyo)|c': 0.2222222222222222
 P(Chinese)|c': 0.2222222222222222
```

```
p_c_test = 1
for i in test_document:
    index = test_document_unique.index(i)
    p_c_test *= p_test_c[index]

p_c_test *= p_c

print(f"P(c|test_document): {p_c_test}")
```

```
 P(c|test_document): 0.00030121377997263036
```

```
p_c_bar_test = 1
for i in test_document:
    index = test_document_unique.index(i)
    p_c_bar_test *= p_test_c_bar[index]

p_c_bar_test *= p_c_bar

print(f"P(c'|test_document): {p_c_bar_test}")
```

```
 P(c'|test_document): 0.00013548070246744226
```