

## به نام خدا

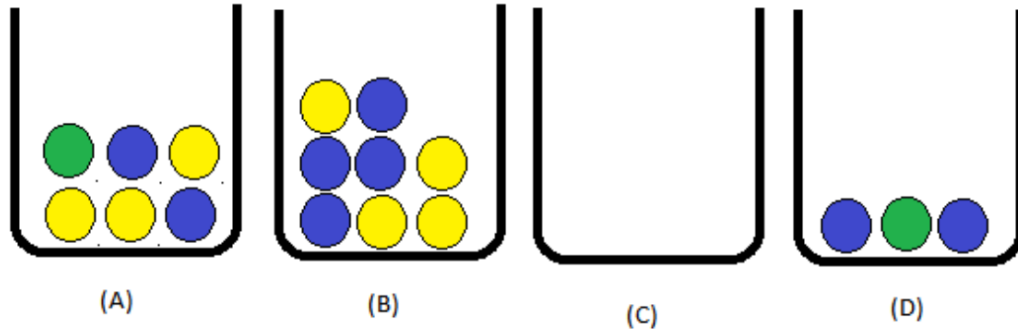
تمرین اول درس یادگیری عمیق

دکتر محمدی

مجتبی نافذ 96431335

۱. چهار جعبه A، B، C و D و سه رنگ توپ سبز و آبی و زرد داریم. فرض کنید به صورت تصادفی یک جعبه را انتخاب و از درون آن یک توپ به صورت تصادفی برمی داریم. احتمال انتخاب جعبه های A و B و C و D به ترتیب برابر ۰.۳، ۰.۴، ۰.۲، ۰.۱ است. احتمال های زیر را حساب کنید.

1.  $P(\text{Box} = B \mid \text{Ball} = \text{Green})$
2.  $P(\text{Box} = A \mid \text{Ball} = \text{Green})$
3.  $P(\text{Box} = B \mid \text{Ball} = \text{Yellow})$
4.  $P(\text{Box} = D \mid \text{Ball} = \text{Blue})$



$$1- P(\text{Box} = B \mid \text{Ball} = \text{Green}) = (P(\text{Ball} = \text{Green} \mid \text{Box} = B) * P(\text{Box} = B)) / P(\text{Ball} = \text{Green}) = 0$$

Equal to zero

$$2- P(\text{Box} = A \mid \text{Ball} = \text{Green}) = (P(\text{Ball} = \text{Green} \mid \text{Box} = A) * P(\text{Box} = A)) / P(\text{Ball} = \text{Green})$$

$$= \frac{0.3}{0.083333333} = 0.600000004$$

$$P(\text{Ball} = \text{Green}) = P(\text{Ball} = \text{Green} \mid \text{Box} = A) * P(\text{Box} = A) + 0 + 0 + P(\text{Ball} = \text{Green} \mid \text{Box} = D) * P(\text{Box} = D)$$

$$= 0.3 * 0.3 + 0 + 0 + 0.1 * 0.1 = 0.083333333$$

$$3- P(\text{Box} = B \mid \text{Ball} = \text{Yellow}) = (P(\text{Ball} = \text{Yellow} \mid \text{Box} = B) * P(\text{Box} = B)) / P(\text{Ball} = \text{Yellow})$$

$$= \frac{4/8 * 0.4}{0.35} = 0.571428571$$

$$P(\text{Ball} = \text{Yellow}) = P(\text{Ball} = \text{Yellow} \mid \text{Box} = A) * P(\text{Box} = A) + P(\text{Ball} = \text{Yellow} \mid \text{Box} = B) * P(\text{Box} = B) + 0 + 0$$

$$= \frac{3}{6} * 0.3 + 0.4 * 4/8 = 0.35$$

$$4- 3- P(\text{Box} = D \mid \text{Ball} = \text{Blue}) = (P(\text{Ball} = \text{Blue} \mid \text{Box} = D) * P(\text{Box} = D)) / P(\text{Ball} = \text{Blue})$$

$$\frac{2/3 * 0.1}{0.366666} = 0.181818512$$

$$P(\text{Ball} = \text{Blue}) = P(\text{Ball} = \text{Blue} \mid \text{Box} = A) * P(\text{Box} = A) + P(\text{Ball} = \text{Blue} \mid \text{Box} = B) * P(\text{Box} = B) + 0 + P(\text{Ball} = \text{Blue} \mid \text{Box} = D) * P(\text{Box} = D)$$

$$= \frac{2}{6} * 0.3 + 0.4 * 4/8 + 0 + 2/3 * 0.1 = 0.36666$$

۳. الف) در این سوال از شما می‌خواهیم که الگوریتم دسته‌بندی Naïve Bayes را تنها با استفاده توابع پایه (مانند میانگین و واریانس) پیاده‌سازی کنید و الگوریتم خود را بر روی مجموعه داده banknote که در اختیار شما قرار می‌گیرد ارزیابی کنید. این مجموعه داده دارای ۴ ویژگی از داده‌های مربوط به ۲ کلاس است. دقت شود که خودتان باید پیاده‌سازی را به طور کامل انجام دهید (در کد پیوست نحوه داندلود مجموعه داده و تقسیم داده‌ها به دو بخش آموزش و ارزیابی نوشته شده است).

ب) نتیجه استفاده از کدهای آماده مربوط به دو الگوریتم [Logistic Regression](#) و [Gaussian Naïve Bayes](#) برای حل این مسئله را گزارش کنید و با نتایج قسمت قبل مقایسه کنید.

[http://archive.ics.uci.edu/ml/machine-learning-databases/00267/data\\_banknote\\_authentication.txt](http://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt)

```
class NaiveBayes:

    def calculate_req(self, x_train, y):
        samples_number, features_number = x_train.shape
        self.Lables = np.unique(y)
        self._mean = np.zeros((len(self.Lables), features_number))
        self._var = np.zeros((len(self.Lables), features_number))
        self._priors = np.zeros(len(self.Lables))

        for idx, label in enumerate(self.Lables):
            x_label = x_train[y==label]
            self._mean[idx, :] = x_label.mean(axis=0)
            self._var[idx, :] = x_label.var(axis=0)
            self._priors[idx] = x_label.shape[0] / float(samples_number)

    def predict(self, x_test):
        y_pred=[]
        for x in x_test:
            posteriors = []

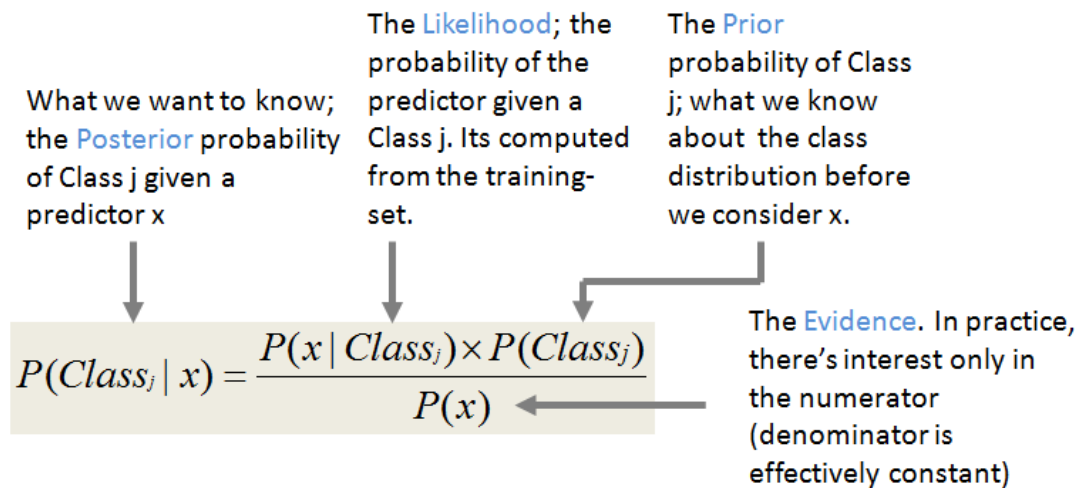
            for idx, c in enumerate(self.Lables):
                prior = np.log(self._priors[idx])

                mean = self._mean[idx]
                var = self._var[idx]
                num = np.exp(-(x-mean)**2 / (2 * var))
                denom = np.sqrt(2 * np.pi * var)

                posterior = np.sum(np.log( num / denom ))
                posteriors.append(prior + posterior)
            y_pred.append(self.Lables[np.argmax(posteriors)])
        return np.array(y_pred)

NaiveBayes = NaiveBayes()
NaiveBayes.calculate_req(x_train, y_train)
y_pred = NaiveBayes.predict(x_test)
```

روش دسته بندی Naive Bayes یک روش مبتنی بر احتمال برای دسته بندی است. که به صورت مفهومی سعی دارد بر پایه ی شباهت داده ی تست به داده های آموزشی هر کلاس در نهایت کلاسی را که بیشترین شباهت را دارد انتخاب کند. ابتدا درون تابع `calcualte_req` برای feature های هر کلاس ما واریانس و میانگین داده ها را محاسبه میکنم و برای هر کلاس احتمال رخداد کلاس را محاسبه می کنیم. سپس درون تابع `predict` برای تک تک تست ها کلاسی را که احتمال رخداد آن بیشینه است را محاسبه می کنیم. برای محاسبه احتمال کلاس به شرط فیچر ها :



Applying the **independence** assumption

$$P(x | Class_j) = P(x_1 | Class_j) \times P(x_2 | Class_j) \times \dots \times P(x_k | Class_j)$$

Substituting the independence assumption, we derive the Posterior probability of Class  $j$  given a new instance  $x'$  as...

$$P(Class_j | x') = P(x'_1 | Class_j) \times P(x'_2 | Class_j) \times \dots \times P(x'_k | Class_j) \times P(Class_j)$$

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp \left( -\frac{(x_i - \mu_y)^2}{2\sigma_y^2} \right)$$

دقت کنید که ما در محاسبه  $P(x | \text{Class})$  ضرب با استفاده از لگاریتم به جمع تبدیل کردیم تا در محاسبه با اعداد کوچک مواجه شویم و خطای محاسباتی کمتر شود و دقت کار بالاتر باشد.

قسمت ب:

برای مقایسه همین شکل ها و دقت ها کافی به نظر می رسد.

	precision	recall	f1-score	support
0.0	0.88	0.81	0.84	745
1.0	0.78	0.87	0.82	600
accuracy			0.83	1345
macro avg	0.83	0.84	0.83	1345
weighted avg	0.84	0.83	0.83	1345

#### My Naive Bayes

	precision	recall	f1-score	support
0.0	0.88	0.81	0.84	745
1.0	0.78	0.87	0.82	600
accuracy			0.83	1345
macro avg	0.83	0.84	0.83	1345
weighted avg	0.84	0.83	0.83	1345

#### Gaussian Naive Bayes

	precision	recall	f1-score	support
0.0	1.00	0.94	0.97	745
1.0	0.94	1.00	0.97	600
accuracy			0.97	1345
macro avg	0.97	0.97	0.97	1345
weighted avg	0.97	0.97	0.97	1345

#### Logistic Regression

همانطور که دیدیم Naive Bayes ای که پیاده کردم هیچ تفاوتی با Gaussian Naive Bayes نداشت. ولی logical regression دقت به مراتب بهتری به ما داد.

کد ها و این تست ها همگی درون فایل [HW1.ipynb](#) قابل مشاهده است.