

1- از Naïve Bayes برای حل مسئله استفاده می‌کنیم. در صورتی که در متن خبر تست داده جدیدی داشته باشیم می‌توان از روش‌های smoothing مانند Laplace(add-1) smoothing استفاده کرد. در این روش فرض می‌کنیم هر کلمه را یکبار بیشتر از تعداد دفعاتی که واقعا وجود داشته است دیده‌ایم و به شمارش همه‌ی کلمات یک واحد اضافه می‌کنیم یک کلمه هم به عنوان unknown word به مجموعه لغات (vocabulary) اضافه می‌کنیم و برای محاسبه احتمال کلماتی که در متن نیستند از احتمال آن استفاده می‌کنیم.

محاسبات در صفحه بعد:

احتمال کلمات داخل متن به صورت زیر محاسب می شود:

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} = \frac{\text{count}(w_i, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V|}$$

احتمال unknown word برای کلماتی که داخل متن نیستند به صورت زیر محاسب می شود

$$\hat{P}(w_u | c) = \frac{\text{count}(w_u, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V| + 1} = \frac{1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V| + 1}$$

$$\hat{P}(c) = \frac{N_c}{N} \Rightarrow \begin{cases} P(0) = \frac{2}{9} = \frac{1}{2} \\ P(1) = \frac{2}{9} = \frac{1}{2} \end{cases}$$

$$P(0 | \text{مرضی}) = \frac{2+1}{9+6} = \frac{3}{15}$$

$$P(0 | \text{علی}) = \frac{2+1}{9+6} = \frac{3}{15}$$

$$P(0 | \text{مناور}) = \frac{2+1}{9+6} = \frac{3}{15}$$

$$P(0 | \text{ایمانی}) = \frac{1+1}{9+6} = \frac{2}{15}$$

$$P(1 | \text{مرضی}) = \frac{1+1}{8+6} = \frac{2}{14}$$

$$P(1 | \text{علی}) = \frac{1+1}{8+6} = \frac{2}{14}$$

$$P(1 | 1) = \frac{1+1}{8+6} = \frac{2}{14}$$

$$P(1 | 1) = \frac{2+1}{3+6} = \frac{3}{9}$$

$$\text{Posterior}(0) = P(0 | \text{فرهنگ علی ضارری ابتیام})$$

$$= \frac{P(0) P(10 | \text{فرهنگ}) P(10 | \text{علی}) P(10 | \text{ضارری}) P(10 | \text{ابتیام})}{\text{evidence}}$$

$$\text{Posterior}(1) = P(1 | \text{فرهنگ علی ضارری ابتیام})$$

$$= \frac{P(1) P(1 | \text{فرهنگ}) P(1 | \text{علی}) P(1 | \text{ضارری}) P(1 | \text{ابتیام})}{\text{evidence}}$$

مخرج دو کسر یکی هست بنابراین از محاسبه آن صرف نظر کنیم

$$P(0 | \text{فرهنگ علی ابتیام ضارری}) \propto \frac{1}{2} \times \left(\frac{3}{15}\right)^3 \times \frac{2}{15} = 0.00053 = 5.3 \times 10^{-4}$$

$$P(1 | \text{فرهنگ علی ابتیام ضارری}) \propto \frac{1}{2} \times \left(\frac{2}{14}\right)^3 \times \frac{3}{14} = 0.00031 = 3.1 \times 10^{-4}$$

$$\frac{5.3}{5.3+3.1} \quad \text{بنابراین کسری 0 انتخاب می شود مقدار احتمال برای آن}$$

مقدار

$$P(0 \text{ ورزشی} | 0) = \frac{1}{9 + (6+1)} = \frac{1}{16}$$

$$P(1 \text{ ورزشی} | 1) = \frac{1}{8 + (6+1)} = \frac{1}{15}$$

$$P(0 \text{ ورزشی} | \text{فرهنگی علمی متاورس احتمالی} | 0) \propto \frac{1}{2} \times \left(\frac{3}{15}\right)^3 \times \frac{2}{15} \times \frac{1}{16}$$

$$= 0.000033 = 3.3 \times 10^{-5}$$

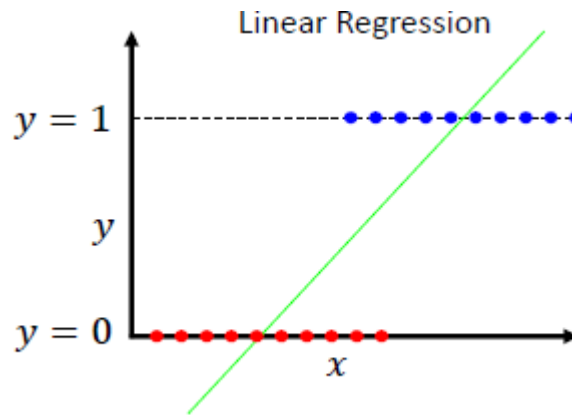
$$P(1 \text{ ورزشی} | \text{فرهنگی علمی متاورس احتمالی} | 1) \propto \frac{1}{2} \times \left(\frac{2}{14}\right)^3 \times \frac{3}{14} \times \frac{1}{15}$$

$$= 0.00002 = 2 \times 10^{-5}$$

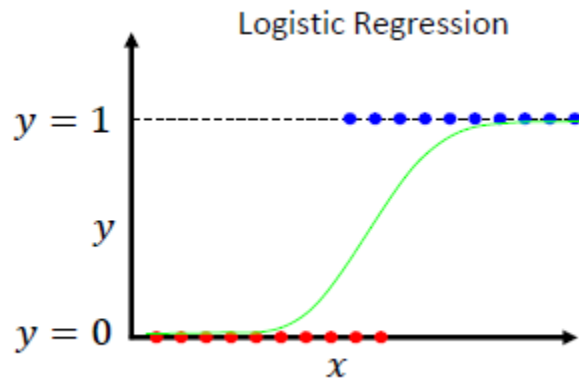
با این داده تست دوم نیز از کلاس 0 صابا سر که احتمال آن 0.33

$$\frac{3.3}{3.3 + 2}$$

2- Logistic Regression اسم آن رگرسیون هست اما در عمل classification هست. برای مثال می‌خواهیم ببینیم یک ایمیل spam هست یا خیر، نقد یک فیلم مثبت هست یا منفی و ... برای چنین کاری نمی‌توانیم مثلاً از linear regression استفاده کنیم.:



همانطور که در شکل فوق می‌بینید Linear regression مقدار  $y$  برای یک سری از  $x$ های کوچک منفی می‌شود و برای یک سری از  $x$ های بزرگ از یک بیشتر می‌شود بنابراین باید چاره‌ای اندیشید. یکی از راه‌ها اعمال یک تابع sigmoid بر روی خروجی آن می‌باشد. که خروجی به صورت زیر خواهد شد:



همانطور که دیده می‌شود با استفاده از تابع sigmoid،  $y$  برای  $x$ های کوچک مقدار صفر و برای  $x$ های بزرگ مقدار 1 دارد و در میانه مقدار  $y$  به صورت احتمال مطرح می‌شود. تابع sigmoid برای این تک ویژگی  $x$  به صورت زیر هست.

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x)$$

در توضیحات بالا  $x$  به صورت تک ویژگی مطرح می‌باشد اما در حالت کلی  $x$  می‌تواند یک بردار ویژگی متشکل از چند ویژگی باشد. برای محاسبه احتمال هر کلاس یعنی احتمال پسین (posterior) که به صورت  $p(y|x)$  تعریف می‌شود از فرمول زیر استفاده می‌کنیم:

$$p(y|x, w) = \frac{1}{1 + e^{-(w \cdot x)}}$$

یکی از مزایای Logistic Regression این است که به راحتی می‌توان مقادیر  $w$  را بدست آورد و از طریق معادلات ریاضی می‌توان بهترین مقادیر را بدست آورد.

اگرچه شکل توابع Logistic Regression و شبکه‌های عصبی (Artificial Neural Network) متفاوت می‌باشد اما درواقع Logistic Regression همان شبکه‌های عصبی می‌باشد که لایه مخفی (hidden layer) ندارد و activation function آن تابع sigmoid می‌باشد. می‌توان گفت شبکه‌های عصبی به نوعی توسعه یافته Logistic Regression هستند. در هر دو روش پارامترهای مدل به وسیله‌ی برآورد حداکثر احتمال (maximum likelihood estimation) آپدیت می‌شود. برای بهینه کردن وزن‌ها یا همان پارامترها می‌توان از روش‌هایی مثل gradient decent استفاده کرد.

تابع هزینه Logistic Regression به صورت محدب هست بنابراین تضمین می‌شود که مینیمم سراسری را پیدا کنیم اما هنگامی که تابع فعالساز Logistic (sigmoid) را در یک شبکه عصبی چند لایه قرار می‌دهیم این خاصیت محدب بودن را از دست می‌دهیم و در شبکه عصبی چند مینیمم محلی خواهیم داشت.

با این حال در عمل backpropagation برای شبکه‌های عصبی با یک یا دو لایه نسبتاً خوب کار می‌کند اما حتی اگر مدل در مینیمم محلی گیر کند باز هم این مدل قدرتمند هست. بنابراین توصیه می‌شود برای یک مسئله classification ابتدا از مدل‌های ساده مانند Logistic Regression استفاده کنیم در برخی موارد این مدل مسئله را به اندازه کافی خوب حل می‌کند اما اگر از کارایی حاصل شده راضی نیستید و به اندازه کافی داده برای آموزش در اختیار دارید می‌توانید از شبکه‌های عصبی که از لحاظ محاسباتی گرانتر هستند ولی مزیت یادگیری توابع پیچیده و غیرخطی را دارد استفاده کنید.

Logistic Regression ریشه در آمار دارد و در مسائل بیولوژی و پزشکی کاربرد داشته است و شبکه عصبی ریشه در computer science دارد.

3- برای پیاده سازی Naïve Bayes از اسلاید استاد استفاده شده است:

## Naïve Bayes: Example

Person	height (feet)	weight (lbs)	foot size(inches)
sample	6	130	8

$$\text{posterior (male)} = \frac{P(\text{male}) p(\text{height} | \text{male}) p(\text{weight} | \text{male}) p(\text{foot size} | \text{male})}{\text{evidence}}$$

$$\text{posterior (female)} = \frac{P(\text{female}) p(\text{height} | \text{female}) p(\text{weight} | \text{female}) p(\text{foot size} | \text{female})}{\text{evidence}}$$

$$P(\text{male}) = 0.5$$

$$p(\text{height} | \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(6 - \mu)^2}{2\sigma^2}\right) \approx 1.5789,$$

Person	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	$3.5033 \times 10^{-2}$	176.25	$1.2292 \times 10^2$	11.25	$9.1667 \times 10^{-1}$
female	5.4175	$9.7225 \times 10^{-2}$	132.5	$5.5833 \times 10^2$	7.5	1.6667



با توجه به مثال فوق احتمال **prior** هر کلاس محاسبه شده است. احتمال هر ویژگی به شرط کلاس نیز با توجه به فرمول گاوسین محاسبه شده یعنی با محاسبه واریانس و میانگین برای هر ویژگی از هر کلاس. شایان ذکر است که در دیتاست مربوطه داده‌ها به صورت **categorical** بودند و با حروفی مثل **e,p,n,b,c, ...** نمایش داده شده بودند که برای محاسبه واریانس و میانگین همه را تغییر داده و با اعداد **0/2.6** و **1/2.6** و **2/2.6** جایگزین شده‌اند که تقسیم بر **2.6** باعث افزایش دقت شده است. همچنین چون یکی از ویژگیها ناقص بود از احتساب آن صرف نظر کرده‌ایم و اگر واریانس برای هر ویژگی **0** شده باشد چون وجود واریانس در مخرج باعث تقسیم بر **0** می‌شد از احتساب آن ویژگی نیز صرف نظر کرده‌ایم. **70** درصد از داده به **train** و **30** درصد باقی مانده به **test** اختصاص داده شده است. دقت مدل معمولاً حدود **92** درصد می‌باشد.

برای پیاده سازی **Logistic Regression** از شکل زیر که مربوط به اسلاید درس **deep learning** آقای **Andrew ng** استاد دانشگاه استنفورد می‌باشد استفاده شده است.

## Implementing Logistic Regression

```

J = 0, dw1 = 0, dw2 = 0, db = 0
for i = 1 to m:
    z(i) = wTx(i) + b
    a(i) = σ(z(i))
    J += -[y(i) log a(i) + (1 - y(i)) log(1 - a(i))]
    dz(i) = a(i) - y(i)
    { dw1 += x1(i) dz(i)
      dw2 += x2(i) dz(i)
      db += dz(i) }
J = J/m, dw1 = dw1/m, dw2 = dw2/m
db = db/m

```

for iter in range(1000):

```

z = wTX + b
= np.dot(w.T, X) + b
A = σ(z)
dz = A - Y
dw = 1/m * X * dzT
db = 1/m * np.sum(dz)

w := w - α dw
b := b - α db

```

Andrew Ng

شکل فوق نشان‌دهنده تمام نکات پیاده‌سازی می‌باشد. ابتدا بردار وزن شامل **w** و **b** را به صورت تصادفی مقدار دهی کردیم. بعد حاصل **w.x+b** محاسبه شده است و روی آن تابع فاعسازی **sigmoid** را اعمال کردیم و بعد به تعداد **2000** بار عمل **optimization** را انجام شده است. مقدار **learning rate** را **0.1** کردیم. برای این قسمت هم **70** درصد از داده به **train** و **30** درصد به **test** اختصاص داده شده است. دقت این مدل حدود **94** درصد می‌باشد.

دقت هر دو مدل برای داده **train** چاپ شده که با توجه به دقت **test** هر دو مدل هیچکدام **overfit** نشده‌اند.

4- برای بازی tic tac toe دو کد با دو قانون مختلف طراحی شده است که به شرح زیر هستند.

در کد اول که کمی ساده تر هست ابتدا چک می شود که حریف برده یا نه، سپس ماشین چک می کند آیا با یک حرکت می تواند ببرد و آن حرکت را انجام می دهد. اما اگر حرکتی نبود چک می کند با آیا حریف با یک حرکت می تواند ببرد و اگر حرکتی بود جلوی آن را می گیرد. قانون بعدی به این صورت هست که حالت بردی پیدا می شود که خودش یک خانه را پر کرده باشد ولی حریف هیچ حرکتی در آن انجام نداده باشد تا حالتی ایجاد کنیم که دو خانه برای برد ماشین آماده شوند. در قاعده بعدی حالتی را چک می کند که نه خودش و نه حریف هیچ حرکتی نکرده باشند تا یک خانه برای برد آماده شود. در نهایت اگر هیچ یک از حالات فوق جوابگو نبود از حرکات موجود یکی را به صورت تصادفی انتخاب می کند.

در کد دوم که کمی پیچیده تر هست چند حرکت اول با حالت قبلی مشترک هست یعنی هست ابتدا چک می شود که حریف برده یا نه، سپس ماشین چک می کند آیا با یک حرکت می تواند ببرد و آن حرکت را انجام می دهد. اما اگر حرکتی نبود چک می کند با آیا حریف با یک حرکت می تواند ببرد و اگر حرکتی بود جلوی آن را می گیرد. سپس چک می کند اگر خانه 5 خالی بود آن را انتخاب می کند. حالت بعدی چک می کند اگر حریف یک خانه در گوشه را انتخاب کرده و دوخانه گوشه مجاور را انتخاب نکرده، خانه مخالف آن را انتخاب می کند اگر حالت های قبل برقرار نبود و حریف دو خانه مجاور از گوشه ها را انتخاب کرده بود هر خانه ای به غیر از گوشه انتخاب می شود اگر حالت های قبلی برقرار نبودند یک خانه از گوشه انتخاب می شود و اگر خانه ای هم در گوشه نبود اولین خانه در دسترس انتخاب می شود.

دلیل این قاعده این هست که از خانه های غیر گوشه فقط دو راه برای بردن وجود دارد یعنی 2،5،6 و 4،5،6 که با تصاحب خانه ی 5 هر دوی این راه حل ها برای حریف بلاک می شوند. همچنین تصاحب این خانه در برد خودمان اهمیت بسزایی دارد. خانه های گوشه در 6 حالت باقی مانده نیز حضور دارند که نشان از اهمیت آنهاست بنابراین سعی می کنیم اگر حریف دوتا از آنها را انتخاب کرده که مجاور نیستند و در دو نقطه ی پایانی خط های قطری هستند و باعث ایجاد دو راه بردن بازی برای حریف می شوند را بلاک کنیم.

قوانین گفته شده برای کد دوم به صورت زیر هستند

### The Rules

```
machineRule1: IF Machine or Player can Win [ IF anyone needs 1 place to Win ] -> The Machine Fill That Place.
machineRule2: IF the Center is open -> Fill the center.
machineRule3: IF Player Fill a Corner -> Fill the opposite corner.
machineRule4: IF Player has two adjacent corners filled, fill in a space other than a corner.
machineRule5: IF none of the previous rules were used, fill in any a corner.
machineRule6: IF none of the previous rules were used, fill the first available place.
```

که از لینک زیر گرفته شده است:

<https://github.com/lrmendes/TicTacToe-Never-Lose-AI>



توجه داشته باشید که کد نوشته شده به هیچ وجه کپی کد این لینک نیست و صرفاً ruleها از این لینک گرفته شده‌اند.

رفرنسها:

سوال 1:

اسلاید text classification: Naïve Bayes از اسلایدهای NLP ژورافسکی برای دانشگاه استنفورد

اسلایدهای استاد

سوال 2:

اسلایدهای استاد

[Logistic regression and artificial neural network classification models: a methodology review](#)

[What is the relation between Logistic Regression and Neural Networks and when to use which?](#)