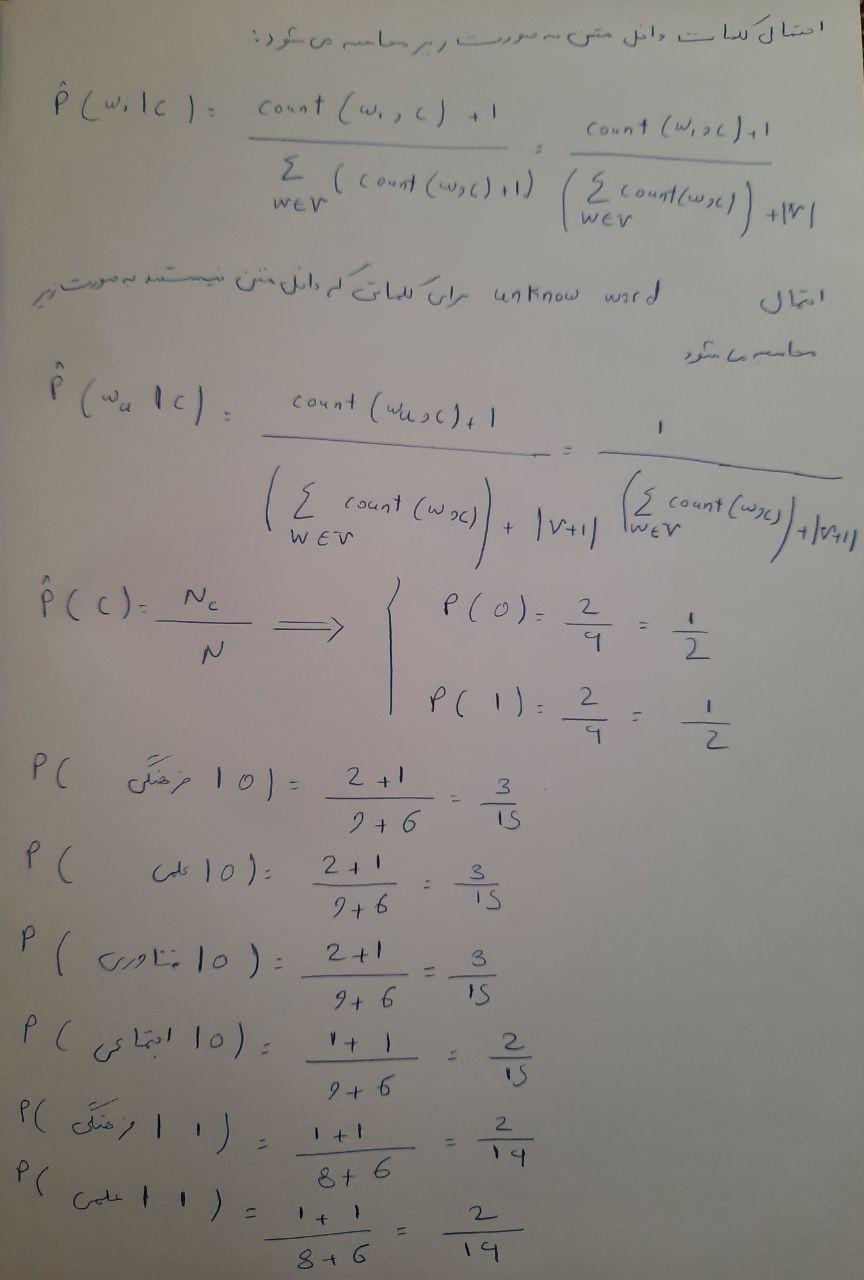
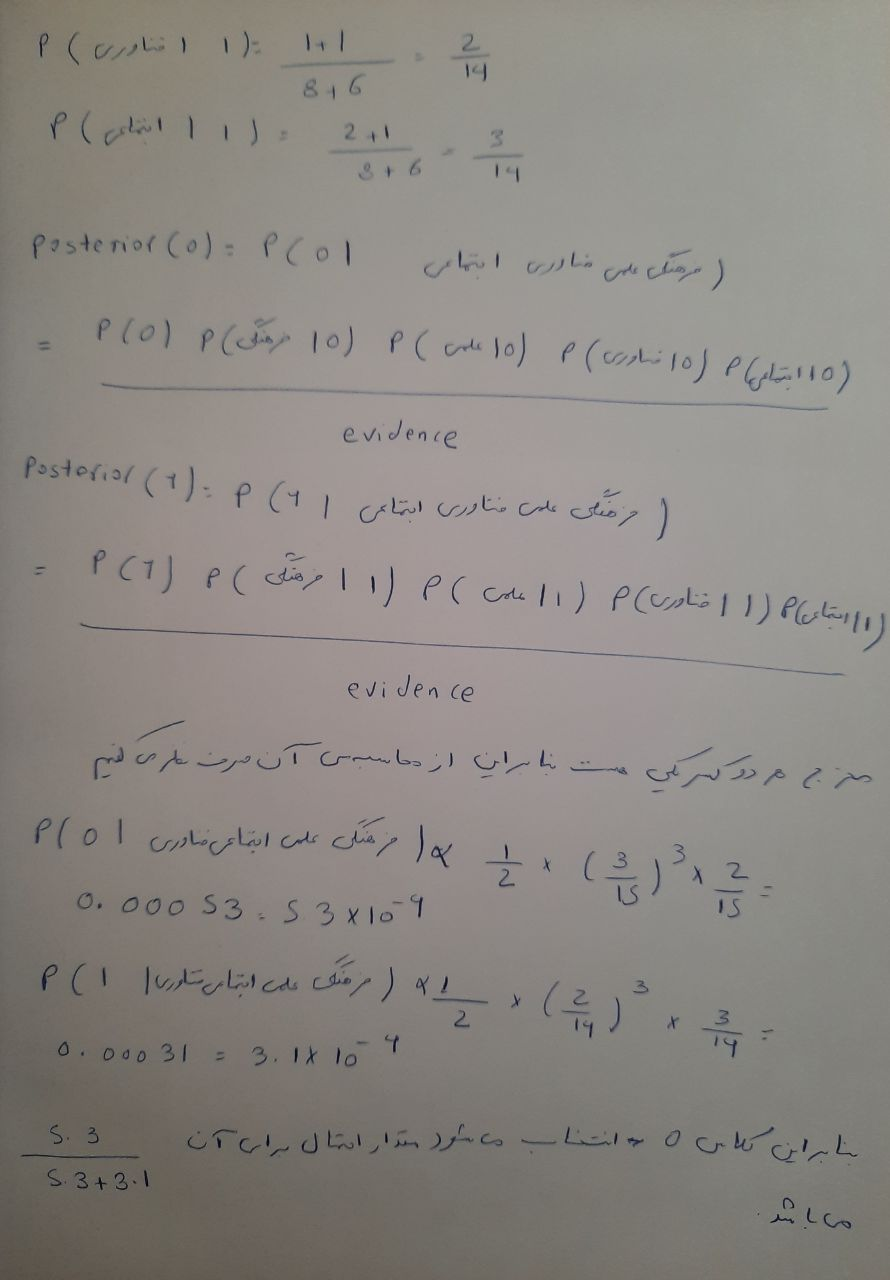
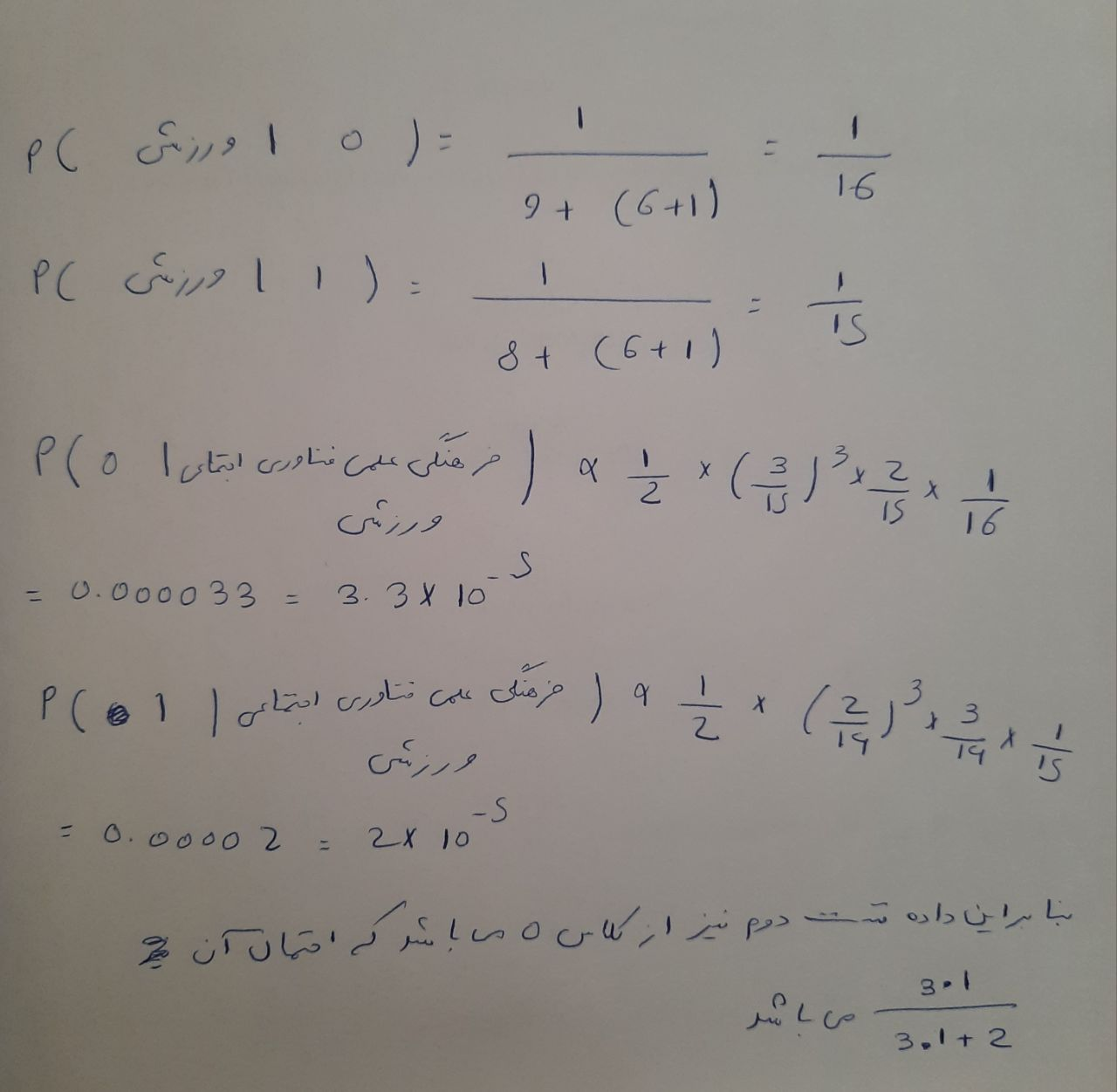
1. از Naïve Bayes برای حل مسئله استفاده می‌کنیم. در صورتی که در متن خبر تست داده جدیدی داشته باشیم می‌توان از روش‌های smoothing مانند Laplace(add-1) smoothing استفاده کرد. در این روش فرض می‌کنیم هر کلمه را یکبار بیشتر از تعداد دفعاتی که واقعا وجود داشته است دیده‌ایم و به شمارش همه‌ی کلمات یک واحد اضافه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

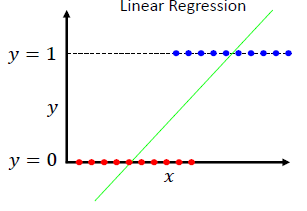
محاسبات در صفحه بعد:



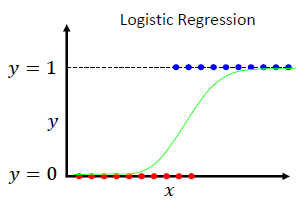




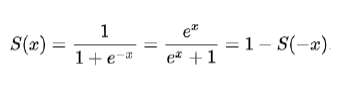
1. Logistic Regression اسم آن رگرسیون هست اما در عمل classification هست. برای مثال می‌خواهیم ببینیم یک ایمیل spam هست یا خیر، نقد یک فیلم مثبت هست یا منفی و ... . برای چنین کاری نمی‌توانیم مثلا از linear regression استفاده کنیم .:



همانطور که در شکل فوق می‌بینید Linear regression مقدار y برای یک سری از xهای کوچک منفی می‌شود و برای یک سری از xهای بزرگ از یک بیشتر می‌شود بنابراین باید چاره‌ای اندیشید. یکی از راه‌حل‌ها 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 xهای کوچک مقدار صفر و برای xهای 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.



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)

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

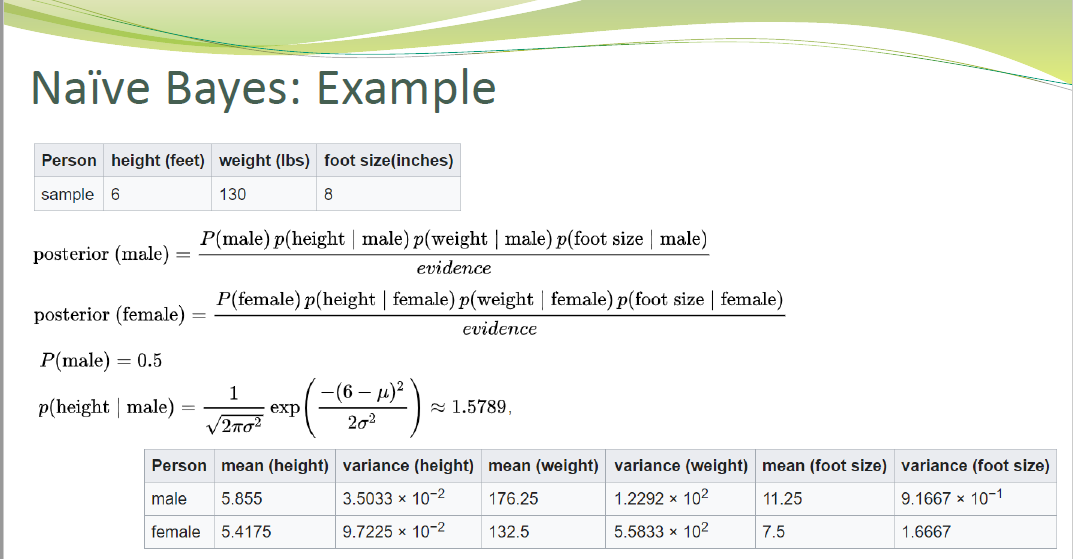
اگرچه شکل توابع Logistic Regression و شبکه‌های عصبی (Artificial Neural Network) متفاوت می‌باشد اما درواقع Logistic Regression همان شبکه‌های عصبی ‌می‌باشد که لایه مخفی (hidden layer) ندارد و activation function آن تابع sigmoid می‌باشد. می توان گفت شبکه های عصبی به نوعی توسعه یافته Logistic Regression 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 نسبتا 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.

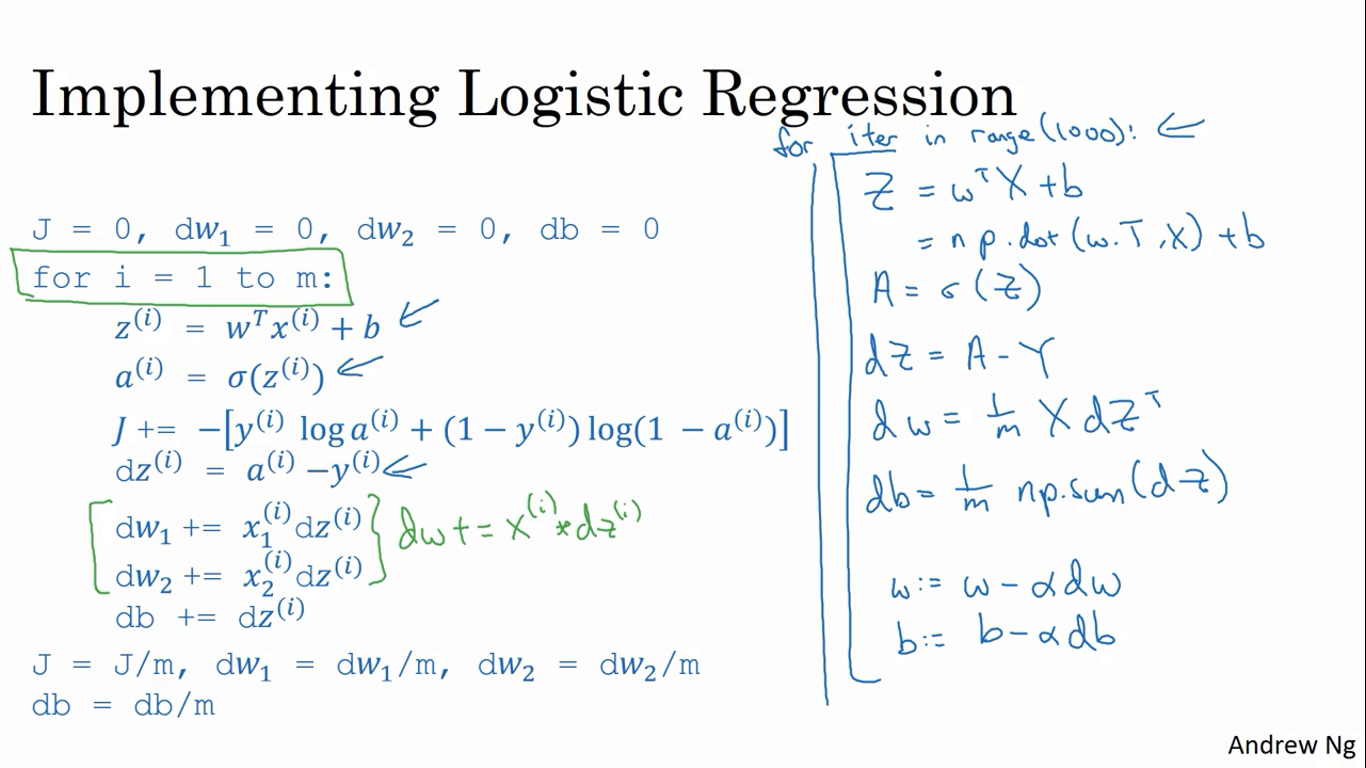
Logistic Regression ریشه در آمار دارد و در مسائل بیولوژی و پزشکی کاربرد داشته است و شبکه عصبی ریشه در computer science دارد.

1. برای پیاده سازی Naïve Bayes از اسلاید استاد استفاده شده است:



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 0/2.6و 1/2.6 و 2/2.6 و ... 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 کرده ایم. 70 درصد از داده به train و 30 درصد باقی مانده به test اختصاص داده شده است. دقت مدل معمولا 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.



شکل فوق نشان‌دهنده تمام نکات پیاده‌سازی می‌باشد. ابتدا بردار وزن شامل wها و b را به صورت تصادفی مقدار دهی کردیم. بعد حاصل w.x+b محاسبه شده است و روی آن تابع فعاسازی 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.

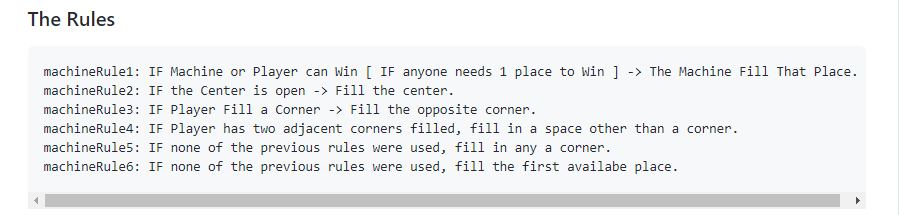
1. برای بازی tic tac toe دو کد با دو قانون مختلف طراحی شده است که به شرح زیر هستند.

در کد اول که کمی ساده تر هست ابتدا چک می‌شود که حریف برده یا نه، سپس ماشین چک می‌کند آیا با یک حرکت می‌تواند ببرد و آن حرکت را انجام می‌دهد. اما اگر حرکتی نبود چک می‌کند با آیا حریف با یک حرکت می‌تواند ببرد و اگر حرکتی بود جلوی آن را می‌گیرد. قانون بعدی به این صورت هست که حالت بردی پیدا می‌شود 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 می‌کند 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 می‌کند.

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

دلیل این قاعده این هست که از خانه‌های غیر گوشه فقط دو راه برای بردن وجود دارد یعنی 2،5،8 و 4،5،6 که با تصاحب خانه‌ی 5 هر دوی این راه‌حل‌ها برای حریف بلاک می‌شوند. همچنین تصاحب این خانه در برد خودمان اهمیت بسزایی دارد. خانه‌های گوشه در 6 حالت باقی‌مانده نیز حضور دارند که نشان از اهمیت آنهاست بنابراین سعی می‌کنیم اگر حریف دوتا از آنها را انتخاب کرده که مجاور نیستند و در دو نقطه‌ی پایانی خطهای قطری هستند و باعث ایجاد دو راه بردن بازی برای حریف می‌شوند را بلاک کنیم.

قوانین گفته شده برای کد دوم به صورت زیر هستند



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 صرفا ruleها از این لینک گرفته شده‌اند.

**رفرنسها:**

سوال 1:

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

اسلایدهای استاد

سوال2:

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?