۱- چرا در رگرسیون لجستیک از تابع هزینه آنتروپی متقاطع به جای خطای میانگین مربعات استفاده می شود؟

آنتروپی متقاطع به طور کلی تفاوت بین دو توزیع احتمال را محاسبه می کند. به عبارت دیگر آنتروپی متقابل اندازه گیری تفاوت بین دو توزیع احتمال برای یک متغیر تصادفی معین یا مجموعه ای از رویدادها است. در رگرسیون لجستیک از تابع سیگموئید استفاده می کنیم و یک تبدیل غیر خطی برای بدست آوردن احتمالات انجام می دهیم. مربع کردن این تبدیل غیر خطی منجر به non-convexity با مینیموم های محلی می شود. یافتن مینیمم جهانی در چنین مواردی با استفاده از نزول گرادیان امکان پذیر نیست. به همین دلیل، MSE برای رگرسیون لجستیک مناسب نیست. آنتروپی متقاطع به عنوان تابع هزینه برای رگرسیون لجستیک استفاده می شود. در تابع هزینه برای رگرسیون لجستیک، پیش‌بینی‌های قطعا اشتباه به شدت جریمه می‌شوند. پیش بینی های درست هم کمتر پاداش می گیرند. با بهینه سازی این تابع هزینه، همگرایی حاصل می شود.

۲- در رابطه با مسئله طبقه بندی چندکلاسه تحقیق کنید و در رابطه با تکنیک یک در مقابل همه بنویسید

طبقه بندی چند کلاسه یک کار طبقه بندی با بیش از دو کلاس است. هر نمونه فقط می تواند به عنوان یک کلاس برچسب گذاری شود. یک کار طبقه بندی با بیش از دو کلاس؛ به عنوان مثال، مجموعه ای از تصاویر میوه ها را که ممکن است پرتقال، سیب یا گلابی باشند، طبقه بندی کنید. طبقه بندی چند طبقه ای این فرض را ایجاد می کند که هر نمونه به یک و تنها یک برچسب اختصاص داده می شود: یک میوه می تواند یک سیب یا گلابی باشد اما نه هر دو در یک زمان. در طبقه بندی چند کلاسه، باید داده ها را به بیش از دو کلاس یا دسته طبقه بندی کنیم. به عنوان مثال، طبقه بندی یک تصویر داده شده به یکی از دسته بندی های متعدد مانند گربه، سگ، پرنده، و غیره. اکنون، رویکردهای مختلفی برای حل این مشکل وجود دارد، و یکی از این رویکردها، استراتژی یک در مقابل همه است.

یک در مقابل همه (به اختصار OvR) یک روش اکتشافی برای استفاده از الگوریتم های طبقه بندی باینری برای طبقه بندی چند کلاسه است. این شامل تقسیم مجموعه داده چند کلاسه به مسائل طبقه بندی باینری متعدد است. سپس یک طبقه‌بندی‌کننده باینری برای هر مسئله طبقه‌بندی باینری آموزش داده می‌شود و با استفاده از مدلی که مطمئن‌ترین مدل است، پیش‌بینی‌ها انجام می‌شود. در استراتژی one vs rest، ما هر بار یک کلاس را می گیریم و آن را به عنوان کلاس مثبت و کلاس های دیگر را به عنوان کلاس های منفی در نظر می گیریم. ما یک طبقه‌بندی‌کننده باینری را برای هر کلاس به طور جداگانه آموزش می‌دهیم که در آن کلاس مثبت کلاس فعلی مورد بررسی است و کلاس‌های منفی همه کلاس‌های دیگر هستند. در طول استنتاج، برای یک ورودی داده شده، آن را از طریق تمام طبقه‌بندی‌کننده‌های آموزش دیده اجرا می‌کنیم، و هر طبقه‌بندی کننده یک امتیاز احتمالی را خروجی می‌دهد که نشان می‌دهد ورودی به آن کلاس مثبت تعلق دارد یا خیر. سپس کلاس با بیشترین احتمال را به عنوان پیش بینی نهایی خود انتخاب می کنیم. این یک رویکرد محبوب برای طبقه بندی چند کلاسه است زیرا ساده است و می توان آن را با هر طبقه بندی کننده باینری استفاده کرد. با این حال، ممکن است برای مجموعه داده های نامتعادل که برخی از کلاس ها نمونه های بسیار کمتری نسبت به سایرین دارند، به خوبی کار نکند. در این صورت، ممکن است نیاز به استفاده از تکنیک‌های پیشرفته‌تر مانند استراتژی‌های وزنی یک در مقابل استراحت یا یک در مقابل یک، در میان سایر موارد داشته باشیم.

۳- یک دیتاست چند متغیره را درنظر بگیرید، فرض کنید مشاهده کردیم که یکی از ضرایب محاسبه شده در عملیات رگرسیون خطی مقدار خیلی بزرگ منفی نسبت به باقی متغیرها پیدا کرده است کدام یک از گزاره های زیر صحیح است؟

برای اینکه بتون در مورد اول و دوم تصمیم گیری کرد به اطلاعات بیشتری نیاز داریم. همچنین باید دلیل این مقدار را نیز بفهمیم دلایلی مانند همبستگ و.... در نتیجه گرینه سوم صحیح است.

۴- با ارائه دلیل صحیح یا غلط بودن هر یک از گزاره های زیر را ثابت کنید:

- اگر بایاس زیاد است اضافه کردن تعداد داده های آموزش کمک زیادی به کم کردن بایاس نمیکند. غ با افزایش اندازه مجموعه داده آموزشی، می‌توانیم به مدل کمک کنیم تا الگوهای بیشتری را از داده‌ها یاد بگیرد و بگیرد. با داده های بیشتر، مدل می تواند روابط پیچیده تری را بین ویژگی های ورودی و هدف خروجی شناسایی کند که می تواند به کاهش بایاس کمک کند. این امر به ویژه در شرایطی که داده های آموزشی اولیه محدود هستند یا فضای مشکل را نشان نمی دهند بسیار مهم است. با این حال، توجه به این نکته مهم است که افزودن داده های آموزشی بیشتر ممکن است همیشه برای حذف بایاس کافی نباشد. عوامل دیگری مانند پیچیدگی معماری مدل، انتخاب ویژگی و تکنیک‌های منظم‌سازی نیز نقش مهمی در توانایی مدل یادگیری ماشین برای یادگیری الگوهای پیچیده در داده‌ها و پیش‌بینی‌های دقیق دارند.
- کم کردن خطای مدل روی داده های آموزش منجر به کاهش خطای مدل روی داده های تست میشود. غ نه لزوما ممکن است اورفیتینگ رخ دهد و در داده های تست خطای زیادی مشاهده شود.
- افزایش پیچیدگی مدل رگرسیون همواره منجر به کاهش خطای مدل روی داده ی آموزش و افزایش خطای مدل روی داده ی تست می شود. غ در حالت کلی بله ولی همیشه هم خطا در داده های تست زیاد نمیشود. با افزایش پیچیدگی مدل، عملکرد داده های مورد استفاده برای ساخت مدل (داده های آموزشی) بهبود می یابد. با این حال، عملکرد در یک مجموعه مستقل (داده های اعتبارسنجی) تا یک نقطه بهبود می یابد، سپس شروع به بدتر شدن می کند. (overfitting). به طور کلی، افزایش پیچیدگی مدل رگرسیون می تواند منجر به کاهش خطای مدل در داده های آموزشی شود، اما ممکن است همیشه منجر به بهبود داده های تست نشود. این به این دلیل است که با افزایش پیچیدگی مدل، ممکن است داده های آموزشی بیش از حد برازش کند و نتواند به خوبی به داده های جدیدی که قبلاً ندیده است تعمیم دهد. در برخی موارد، افزودن ویژگی‌های بیشتر یا افزایش پیچیدگی مدل نیز ممکن است نویز یا اطلاعات نامربوط را به مدل وارد کند که می‌تواند باعث افزایش بیشتر خطا در داده‌های تست شود. یافتن تعادل مناسب بین پیچیدگی و عملکرد مدل نیاز به آزمایش و ارزیابی دقیق دارد و ممکن است مواردی وجود داشته باشد که مدل‌های ساده‌تر بهتر از مدل‌های پیچیده‌تر عمل کنند. به طور کلی، رابطه بین پیچیدگی و خطا همیشه ساده نیست و می‌تواند به عوامل مختلفی مانند میزان و کیفیت داده‌ها، انتخاب ویژگی‌ها و الگوریتم‌ها و مشکل خاصی که در حال تلاش هستید بستگی داشته باشد.