سوال 1_

بله ارائه درصد اطمینان برای svm رو می توانیم فاصله دیتای تست تا decision boundry در نظر بگیریم اما قابلیت تبدیل مستقیم به احتمال را مانند سایر الگوریتم ها logistic regression ، naïve bayes و یا .. را ندارد پس باید از متدی مانند platt scaling یا logistic calibration استفاده کرد . شیوه این متد به این صورت می باشد که ابتدا دیتا را به دو بخش آموزشی و validation تقسیم می کنیم. سپس بر روی svm score های بدست آمده از دیتای آموزشی مدل لاجستیک رگرشن را اعمال می کنیم. به این شکل لاجستیک رگرشن می تواند احتمال کلاسی را بر حسب svm score پیشبینی کند.

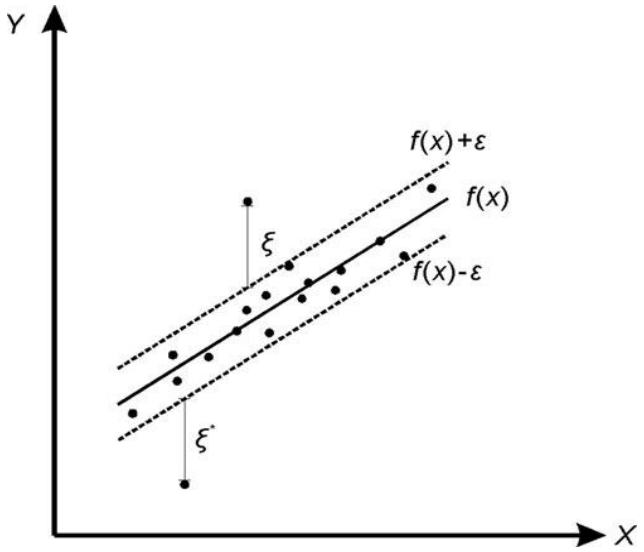
سوال2_

وقتی به اندرفیتینگ در کرنل rbf برخوردیم نیاز است هردو مقدار گاما یا c را افزایش دهیم. C که به عنوان میزان کنترل اجازه به میس کلسیفای شدن تعریف می شود با افزایش مقدار سخت گیری بیشتری در فیت شدن با دیتای آموزشی به خرج می دهد در نتیجه در کیس های اندرفیت از افزایش آن استفاده می شود. با افزایش مقدار گاما هم در واقع میزان K کرنل X ها افزایش می یابد یعنی کرنل تاثیر بیشتری از دیتای آموزشی گرفته پس پیچیده تر شده است.

| Kernel | Kernel function, $k(x, y)$ |
|---|---|
| Linear | $x^\top y$ |
| Poly | $(\gamma x^\top y + c_0)^p$ |
| RBF | $\exp(-\gamma\|x - y\|^2)$ |

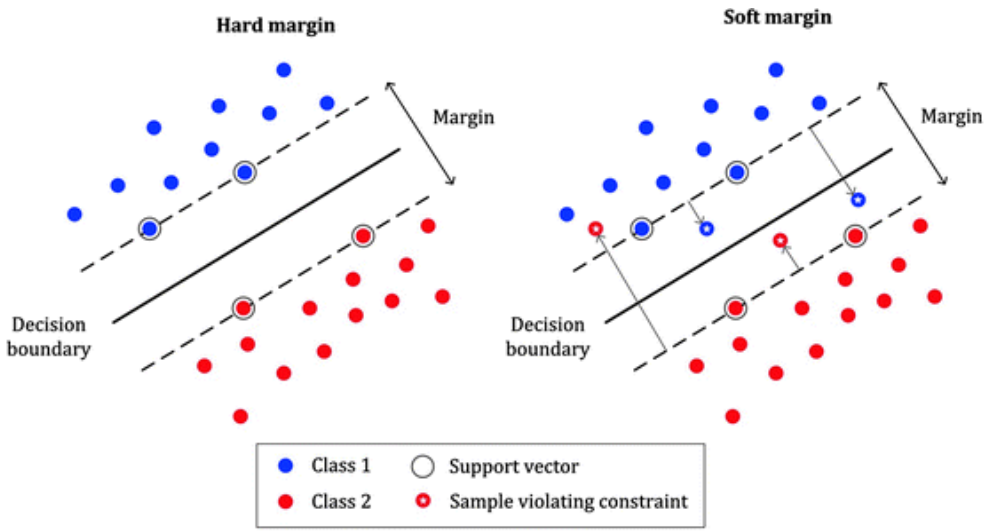
سوال3_

در svr ها که به عبارتی svm برای رگرشن هستند تصور می کنیم که یک دیتای رگرشنی داریم که بجای فیت کردن یک خط صاف به دیتای آن ها مانند لینیر رگرشن یک شکل لوله مانندی دور دیتا فیت می شود که به ناجیه درون tube ناحیه e-insensetive می گویند که از خطی که فرض کنیم فیت ترین خط لینیر برای دیتاست به اندازه اپسیلون فاصله دارد. و هر نقطه ای که در این لوله پیشبینی شود خطای آن را صفر در نظر می گیریم. در واقع به اندازه اپسیلون اجازه میس کلسیفای کردن به دیتا می دهیم.



سوال4_

به طور کلی وقتی داده ها به طور خطی قابل جداسازی باشند از hard margin استفاده می شود و برای داده هایی که به طور غیرخطی جدا می شوند soft margin اما در حالتی که از hard margin استفاده کرده ایم ولی مدل دچار اورفیت شده است یعنی به بهای درست دسته بندی کردن تمامی داده ها کارایی کلی مدل پایین آمده و وابسته به دیتای آموزشی شده است از soft margin بجای آن استفاده می کنیم. یعنی به بهای میس کلسیفای شدن برخی داده ها اورفیت کل مدل را کاهش می دهیم. در واقع سافت مارجین معنای انعطاف پذیری آن نسبت به میس کلسیفای را می رساند و هارد مارجین عدم انعطاف برای میس کلسیفای.




سوال5_

اندیس جینی در واقع اندازه گیری میزان ناخالصی می باشد. در درخت های تصمیم گیری در واقع هدف از تبدیل یک نود به دو نود دیگر کاهش این میزان است که معمولا اتفاق می افتد و در نود های فرزند اندیس جینی کاهش می باید اما همیشه این اتفاق نمی افتد فرضا اگر درخت در جایی بنا به فیچر نسبتا نامرتبطی تقسیم شده باشد یا به هر علت دیگری الگوریتم درست کار نکند یا در یک جا ناخالصی بیشتر ناشی از تقسیمی نهایتا منجر به تقسیم بهینه تری در نود های بعدی شود.

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

سوال 6_

خیر در حالت کلی اسکیل کردن برای درخت های تصمیم گیری لازم نیستند و تاثیر چندانی نمیگذارند تنها در حالت های خاصی که اسکیل فیچر ها تفاوت زیادی داشته باشد ممکن است اولویت تاثیر فیچر ها به علت این اسکیل نبودن زیاد دچار خطا شود که این لزوما به معنای اندرفیتینگ نیست.

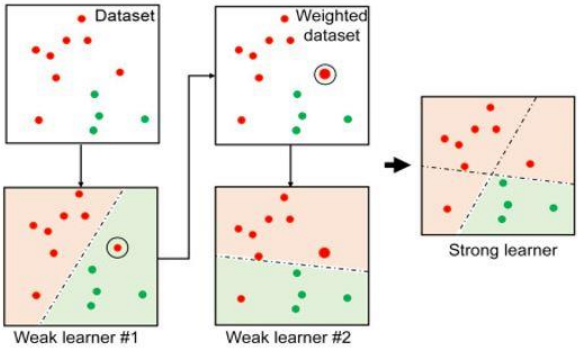
سوال7_

ایده کلی این است که مقیاسی برای تعیین میزان اهمیت فیچر ها داشته باشیم و با قرار دادن یک سطح آستانه فیچرهایی که به میزان مطلوب تری تاثیرگذارند به نحوی برگزیده شوند. به این منظور شیوه های مختلفی وجود دارد : 1- feuture importance که به هر فیچر نمره ای مبنی بر میزان اهمیت بر تارگت میدهد وفیچرهایی با بیشترین نمره انتخاب می شوند.
2-recursive feature elimination(RFB) که عملکرد آن به شکل معکوس مبنی بر حذف ضعیف ترین فیچرها تا رسیدن به تعداد دلخواه فیچر می باشد.

سوال 8_

در adaboost فرمول وزن ها به این شکل می باشد:

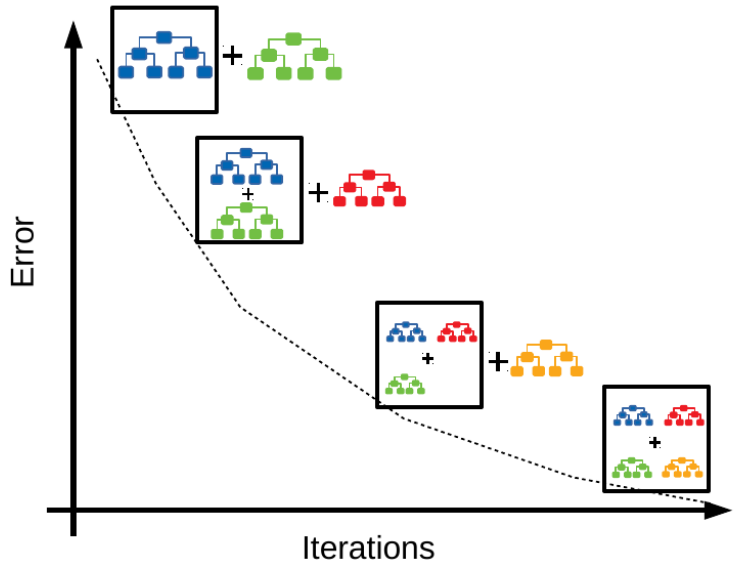
ıing rate*log(1-e/e), where e is the error




b) کمتر - شیوه گرادینت بوستینگ در رگرشن به این شکل می باشد که ابتدا میانگین تارگت هارا به عنوان مقیاس پیشبینی شده برای تک تک داده ها در نظر می گیریم و سپس درخت های کوچکی که معمولا 8 تا 32 برگ دارند ( تعداد برگ ها وابسته به سایز داده) را می سازیم. که هر درخت ناشی از جبران خطای درخت قبلی می باشد. به این شکل که در هر ایتریشن میزان خطای پیشبینی شده و مقدار واقعی را بدست می آوریم و سپس درختی را به منظور پیشبینی میزان خطا می سازیم. و خطاهایی که در یک برگ قرار دارند را میانگین می گیریم و مقدار آن برگ قرار می دهیم. و میزان قبلی پیشبینی شده به ازای هر داده را به علاوه یک learning rate که مقداری بین صفر یا یک است ، ضرب در میزان خطای پیشبینی در برگ مرتبط می کنیم. این learning rate به منظور جلوگیری از اورفیت قرار داده شده تا بجای اینکه در یک گام میزان پیشبینی را به تارگت درست

نزدیک کنیم ذره ذره طی ساخت درخت های بیشتری به مقدار مطلوب نزدیک برسیم. هرچه مقدار این هایپرپارامتر کمتر باشد

جلوگیری از اورف




سوال 9_

به صورت کلی،روش های *ensembling* به دو دسته تقسیم بندی میشوند، *homogeneous ensembling* و *heterogeneous ensembling*. روش اول به این گونست که از ترکیب چندین مدل از یک نوع برای انسمبل کردن استفاده میشود. برای مثال، استفاده از چندین مدل SVM.

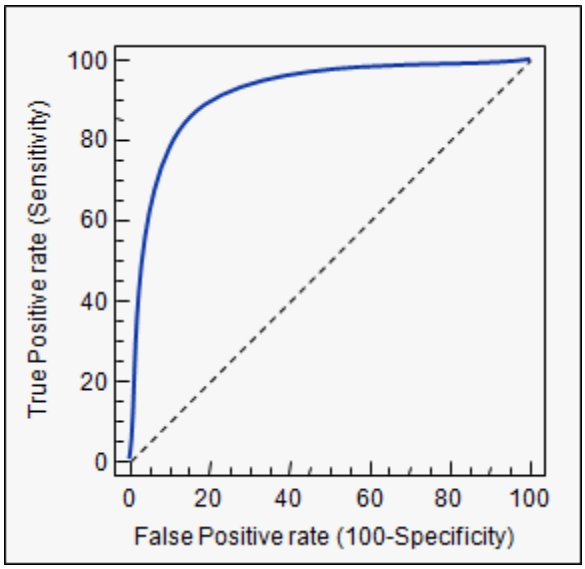
مدل ها به صورت جداگانه train میشوند و تجمیع شده پیشبینی های همه آنها برای نتیجه نهایی استفاده میشود. این روش از این جهت کارآمد است که میتواند از تنوع مدل های هم خانواده برای درک بهتر جنبه های مختلف دیتا و در نتیجه بهبود بخشی به پرفورمنس نهایی بهره مند شود.

و اما روش دوم، از مدل های چندگانه از خانواده های گوناگون برای انسمبل کردن استفاده میکند. به عنوان مثال میتوان به ترکیب دو مدل neural network و decision tree اشاره کرد. هر کدام از مدل ها نقاط ضعف و قوت خود را دارند و میتوانند در فرآیند training مکمل هم باشند. این روش از این جهت کارآمد است که با ترکیب مدل های مختلف، میتواند به میزان robustness و تنوع بیشتر در پیش بینی ها منجر شود.

اینکه کدام یک از این روش ها بهتر است به نوع مسئله و dataset تحت بحث بستگی دارد، اما به طور کلی روش اول زمانی کارآمدتر است که مدل ها bias های یکسان و noise source های متفاوتی دارند، اما روش دوم زمانی کارآمد تر است که مدل ها دارای bias و noise source های متفاوتی هستند.

سوال10_

ROC Curve نموداریست که به ارتباط بین true positive rate و false positive rate اشاره دارد. این در حالتی می باشد که مدل کلاسیفیکیشنی داشته باشیم که دارای دو کلاس قابل پیشبینی yes و no باشد که سپس بتوانیم TPR را به عنوان میزان positive یا yes هایی که درست پیشبینی شده اند و FPR را به عنوان positive یا yes هایی که درست پیشبینی نشده اند تعریف کنیم.




حالت بهینه که مطلوب بودن مدل را نشان میدهد

کم بودن FPR و زیاد بودن TPR می باشد. پس بهترین حالت

در قسمت خم چپ بالا می باشد.

اگر گوشه سمت چپ بالا زیاد از حد به خط وسط نزدیک باشد

یعنی مدل randomness بالاتری دارد و محتمل تر است که

دچار اورفیت باشد. و هرچقدر دورتر از خط باشد محتمل تر است

دچار underfit باشد.

در این حال AUC (area under curve) به معنای مساحت

زیر نمودار است . و به عنوان تخمینی برای کارایی مدل استفاده می شود. مقدار بیشتر آن به معنای بهتر بودن است.

سوال11_

در مدل های binary classification، میزان threshold مشخص کننده بازه کلاسبندی شدن نمونه هاست. به عنوان مثال، به طور معمول این عدد در این تسک 0.5 است، اینگونه است که مدل برای هر نمونه یک score یا یک احتمال محاسبه میکند که اگر این مقدار از 0.5 بیشتر باشد در کلاس 1 و اگر از 0.5 کمتر باشد در کلاس 0 کلاسبندی میشود، اما اگر data ما imbalance باشند مقدار پیشفرض threshold میتواند باعث کاهش دقت مدل شود. تاثیر این پارامتر بر روی رفتار مدل اینگونه است که با تخصیص یک مقدار خاص، میزان False Positive و False Negative مدل به ازای تغییر این پارامتر، تغییر میکند. همیشه یک trade off میان این دو مقدار وجود دارد، که با تغییر این پارامتر میتوان این trade off را کنترل کرد. به این گونه که با کاهش مقدار threshold، نمونه های بیشتری میتوانند در کلاس 1 قرار بگیرند، که این باعث

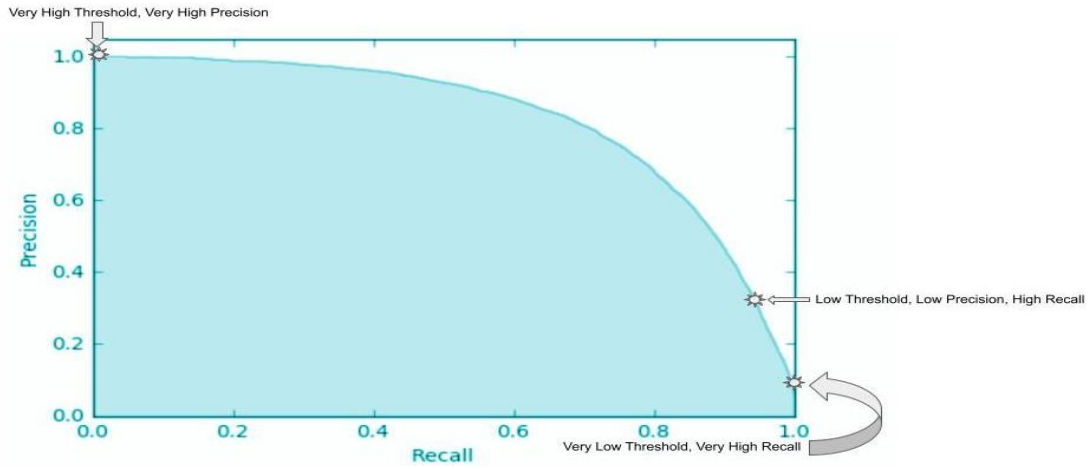
افزایش نرخ FP و کاهش نرخ FN میشود. اما به طور معکوس، افزایش threshold باعث کاهش نرخ FP و افزایش نرخ FN میشود.

با توجه به روابط محاسبه precision و recall:

Precision = TruePositives / (TruePositives + FalsePositives)

Recall = TruePositives / (TruePositives + FalseNegatives)

کاهش threshold باعث افزایش FP و کاهش FN در نتیجه، باعث کاهش precision و افزایش recall میشود. افزایش threshold باعث کاهش FP و افزایش FN در نتیجه، باعث افزایش precision و کاهش recall میشود. نمودار پایین این مطالب را تصدیق میکند:

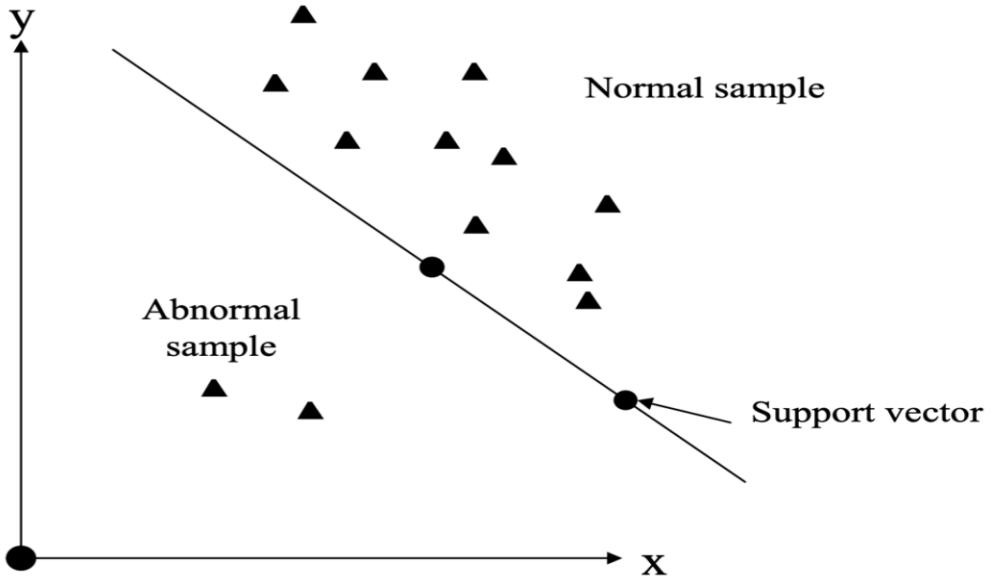



سوال12_

OvO و OvA دو الگوریتم برای دسته بندی مولتی کلاس ها می باشند. عملکرد OvO به این شکل است که اگر دارای N کلاس باشیم $N(N-1)/2$ دسته ترکیبات هر دو کلاس را ایجاد می کند. برای دسته بندی یک داده به کلاسی که تعداد دفعاتی که به هر دسته در آزمایش بین دو دسته ها منتسب شده بیشترین است ، نسبتش می دهیم. در OvA اگر N دسته داشته باشیم N بار هر دسته را مقابل تمام دسته ها دیگر قرار می دهیم یعنی تمام دسته های دیگر را یک دسته فرض می کنیم. و دسته بندی داده جدید را مشخص می کنیم. OvO در خالت کلی عملکرد بهتری دارد زیرا در OvA ممکن است به مشکلات ناشی از imbalance بودن دیتا بربخوریم. اما هزینه محاسباتی آن هم بیشتر می باشد.

سوال 14_

با در نظر گرفتن مسئله به عنوان یک مسئله binary classification، میتوان data را به دو کلاس normal و anomaly تقسیم بندی کرد و هدف ایجاد یک decision boundary بین این کلاس هاست. اما به طور معمول در data های روزمره، میزان نمونه های anomaly بسیار کمتر از نمونه های normal است که این باعث imbalance شدن کلاس ها میشود که SVM نسبت به imbalance بودن بسیار حساس است و ممکن است نتیجه خوبی حاصل نشود، این چالشی است که در این تسک با آن روبرو هستیم. اما یکی از راه های حل این چالش، استفاده از یک حالت خاص SVM تحت عنوان one-class SVM است. SVM در حالت کلی، با استفاده از یک hyperplane مرز میان کلاس ها را مشخص میکند(در این تسک کلاس 0 و کلاس 1). اما در حالت one-class SVM با استفاده از یک hyperplane میتواند dataset را از مبدا مختصات جدا کند به نحوی که حدالامکان به data points نزدیک تر باشد. به نمودار زیر توجه کنید:




تکنیک های دیگری هم برای بهبود این روش وجود دارد. به عنوان مثال میتوان از RBF kernel برای فیت کردن یک non-linear boundary استفاده کرد.