

مبحث اول:

- ۱- وجود نداشتن یک یا چند ویژگی در داده‌ها اختلال است.
 - پر کردن داده‌ها خالی با یک سری ویژگی‌ها که دارای مانده میانی، میان و (مثلاً) میانگین (همان ویژگی می‌کنیم)
 - سطرهایی که داده‌ها miss شده دارند را حذف کنیم.
 - سطرهایی که شامل داده‌ها miss شده است را حذف کنیم (فیلت)
 - پر کردن داده‌ها miss شده با اولین داده قبلی (Forward fill) یا بعدی (Backward fill)
 - train کردن یک مدل مثل کلاس، KNN، ... برای پیش‌بینی مقادیر خالی

۲- نام‌مقابل بودن توزیع داده‌ها در کلاس:

- می‌توان برای مقادیر کردن داده‌ها از scaler ها مثل StandardScaler، MinMaxScaler، Scale، ... استفاده کرد. مثلاً در Scale داده‌ها را نرمال استandar می‌کند. وابسته به توزیع که می‌خواهیم از زیر توزیع‌ها مقادیر می‌توان استفاده کرد

۳- وجود نویز در داده‌ها:

- می‌توان داده‌ها که نویز outlier را از طریق نمودارها مثل boxplot و دیگر روش‌های آماری

مقیاس‌هایی که همیشه فیلتر می‌باشند حذف دارند و حذف می‌کنیم

Scale و دیگرها یا کاهش پیدا

یک سری داده‌ها در دست می‌گیریم که به نویز کمتری می‌باشند

در نقطه نظر داده‌ها نویز در مدل‌ها موانع

۴- ویرگه ها Correlated
 - در نتایج که Correlation بالا به چشم داره را می توان میله ها را کنار حذف کرد و دیگری را نگه داشت
 - بهنگام PCA داده جدید تولید کنیم که مرتبط نباشند

سوال دوم: می توان یک متغیر بدین از عوامل دادن را هم به مدل گریه های افغان کرد

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

 multivariate linear Regression
 مدل داده ها
 متغیرهای وابسته

least square method به صورت
 باید کنیم طوری که بهترین باشد - به این شکل تابع

$$\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_n x_{in}))^2$$

به این شکل از تمام متغیرهای وابسته به پارامترها ، مقادیر پارامترها را

Summation را می گیریم . (نیت به هر کدام از $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ یک بار مشتق می گیریم و برابر با صفر می گذاریم)
 چند دستگاه معادله می شود که باید از آن $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ به دست می آید

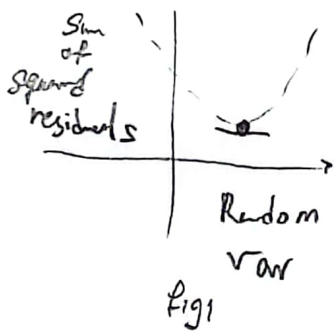
استفاده از gradient descent: یک یاخته داریم بهینه a_0, a_1, a_2, a_3 داریم و در ابتدا مقادیر رندوم به تعریف می کنیم
 می دهیم پس یک loss function که به صورت $\sum (y_i - (a_0 + a_1 x_{i1} + a_2 x_{i2} + a_3 x_{i3}))^2$ تعریف می شود، را بهینه کنیم
 پس از همان این \sum ، معیار مقیاس که محاسبه است مشتق می گیریم. حال مقدارهای مختلف را به پایانه محاسبه در نظر می گیریم
 و مقدار را حساب می کنیم (مقدار پس از مشتق در یک نقطه شده در اول کار، در واقع این همان شیء در یک نقطه می شود که با مقدارهای رندوم می گیریم)
 بعد به کمک یک فریب که به عنوان learning rate تعریف می شود، $step size$ را حساب می کنیم:

$$Step\ Size = Slope \times learning\ rate$$

در پس آن مقدار محاسبه را اینگونه update می کنیم:

$$new\ value\ for\ x_{random\ var} = old\ value\ of\ it - step\ size$$

حالا این مقدار جدید به جای مقدار قبلی در محاسبه قرار می دهیم تا به یک مقدار جدید (new value) در نقطه جدید حساب می کنیم



هر چه $Step\ size$ که چقدر شود به یاخته کردن نقطه منحنی (Fig) نزدیک می شویم

نهایت آنکه جمع ما متوقف می شود که $Step\ size$ خیلی نزدیک به صفر باشد (اگر که slope به صفر
 تبدیل می شود)

روش دیگر استفاده از معادله normal equation است. این روش مستقیماً فریب بهینه را بدون تکرار از روش iterative

مانده Gradient descent به می آورد

$$B = (X^T X)^{-1} X^T y$$

↓
 بردار ضرایب محاسبه
 ↑
 ماتریس ویژگیها
 ↑
 بردار از متغیرها
 ↑
 هدف (فریب شده)

به کمک حل کردن معادله بالا، میتوان ضرایب را بدست آورد

True \ Predicted	N	P
	Y ₀₀	Y ₀₁
N	TN	FP
P	FN	TP

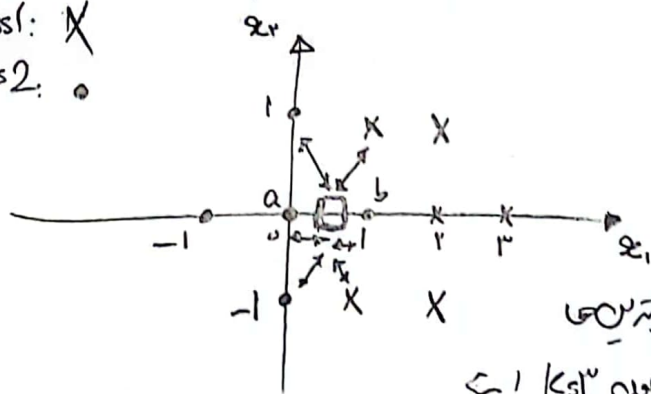
$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FN + FP} = \frac{500}{550} \approx 0.909$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{100}{550} \approx 0.1818$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{100}{450} \approx 0.2222$$

$$F1\text{-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \times 0.1818 \times 0.2222}{0.1818 + 0.2222} \approx 0.2$$

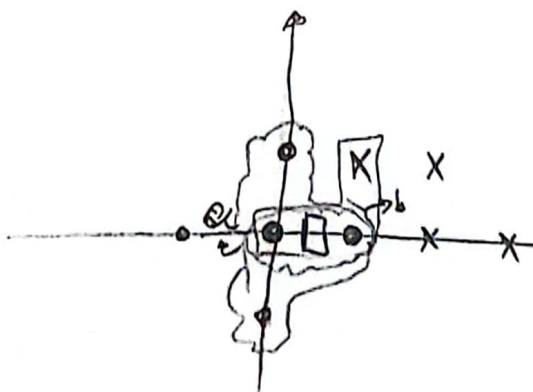
class: X
class 2: o



(0.5, 0.5): داده است

① فاصله اقلیدسی: اگر به همای ها داده است نگاه کنیم، به دو دسته می شود
هستند که از کلاس (0) هستند. این نقطه سوم هم نیاز داریم چون $k=3$ است
اما مستقل از نقطه سوم که چند حالت می تواند داشته باشد (در شکل نشان داده ایم)

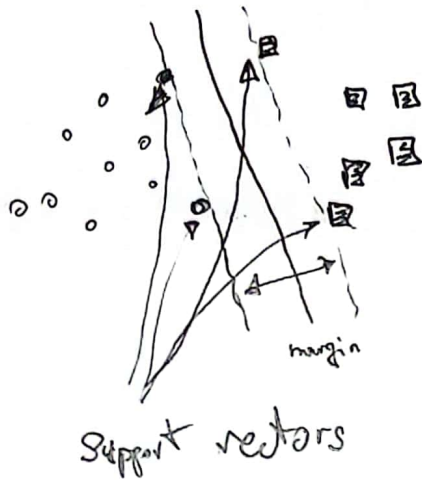
چون دنیا از کلاس (0) است. دسته کلاس (1) می باشد. داده است از نوع class 1 می باشد (0) می شود.



② فاصله منصفی

در این روشها هم اگر از فاصله منصفی استفاده کنیم، می بینیم که دو حالت است.
طوری که اگر این دو نقطه سوم متفاوت می توانند باشند که باز هم با توجه به این
طوری که از کلاس (0) می شوند و 2/3 هستند، نقطه دومی هم می تواند باشد و نمی تواند
در کلاس دنیا است ما اینهم class 2 می باشد (0) می شود.

Support Vector Machine (SVM) : 1) نقطه‌ای که روی یک خط اول صفحه جدا کننده (hyper plane) قرار دارد و دو موقعیت
 hyper plane از آن می‌گذرد. به کمک این نقطه می‌توان margin آن Classifier را مشخص کرد. $Support\ vector$ ها معیار تقسیم موقعیت hyper plane می‌شوند. این نقاط در نهایت
 در مسافتی معین از هم جدا می‌شوند.



2) این الگوریتم برای دیتاست‌ها مناسب نیست و جواب عملی نمی‌دهد و به همین دلیل training SVM به سختی به اندازه
 تعداد داده بستگی دارد.

3) زمانی که دیتاست ما داده‌ها $noise$ و $outlier$ زیادی داشته باشد
 می‌تواند منجر به نتایج هر چه اشتباه‌تر شود. از تعداد نمونه‌ها داده کم‌تر می‌گیرد و به همین دلیل SVM یک مدل ضعیف دارد.
 4) زمانی که $kernel$ مناسب هم انتخاب نشود عملکرد SVM ضعیف می‌شود. مثلاً وقتی داده‌ها به صورت غیر خطی
 قابل واریاسیون باشند اما $kernel$ ما $linear$ انتخاب کنیم.

5) کرنل‌ها تابعی هستند که برای انتقال داده‌ها به فضای با ابعاد بیشتر استفاده می‌شوند. روشی است که به کمک آن
 داده‌ها (که به صورت خطی قابل جدا سازی نیستند) به یک فضای ویژگی با ابعاد بالاتر استفاده می‌شوند که ممکن
 است به صورت خطی از هم جدا شوند.

انواع $kernel$ ها مختلفی وجود دارد که به صورتی اشاره می‌کنیم:

- ① Linear
- ② Polynomial
- ③ Radial Basis Function (RBF)

(ع) در $hard$ SVM classifier هدف یافتن $hyperplane$ است طبقه بتواند داده‌ها را به دو دسته کامل به دو دسته طبقه‌بندی کند.

انتخاب به دو کلاس (صفت) تقسیم کنند با این حال زمانی که داده‌ها به صورت خطه قابل تقسیم نباشند یا هموار داده‌ها به سخت باشند، این امکان وجود ندارد. در این حالت‌ها این روش دریافتن یک $hyperplane$ که بتواند داده‌ها را کاملاً از هم جدا کند ممکن است فواید محدود و مشکلی به همراه داشته باشد.

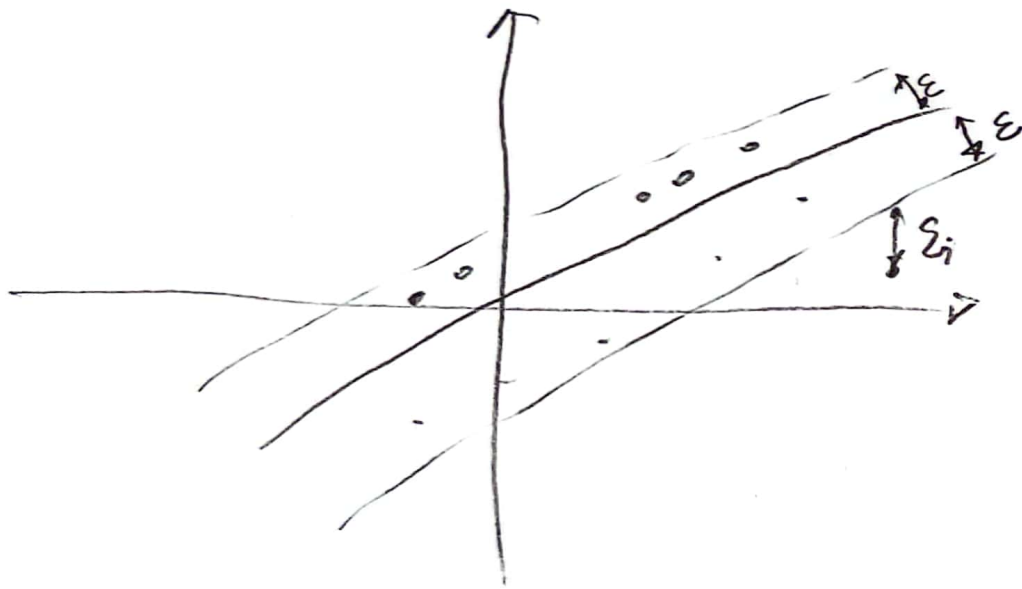
در $Soft$ SVM classifier، $slack$ متغیرها که اجازه می‌دهند به بعضی از نقاط داده‌ها در سمت اشتباه (misclassification) قرار گیرند، امکان طبقه‌بندی اشتباه را فراهم می‌کند. مثلاً بهینه‌سازی در این روش به صورت زیر است:

$$\frac{1}{2} \|w\|^2 + C \sum \xi_i \quad ; \quad y_i (w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0$$

ξ_i ها متغیرهای $slack$ هستند (اجازه خطای می‌دهند) و C یک $hyperparameter$ است که $trade-off$ بین کم کردن $\|w\|^2$ و به حداقل رساندن طبقه‌بندی اشتباه را کنترل می‌کند. مقدار بیشتر C موجب حاصله باریک طبقه‌بندی اشتباه کمتر. در حالی که مقدار کمتری C منجر به حاصله وسیع‌تر اما طبقه‌بندی اشتباه را به حداقل می‌رساند. این روش به نقاط دیتا به دو دسته اشتباه حاصله می‌دهد اشتباه این صفت قدر دارند، بهینه‌سازی در نقطه می‌گیرد. نتیجه به دو اجزای می‌شود که این نقاط در حاصله می‌شود اشتباه. این صفت قدر را کمینه اما در مجموع هدف بهینه‌سازی در این روش به این صفت‌های حاصله می‌شود که به دو دسته می‌رساند در حالی که بهینه‌سازی نقاط داده طبقه‌بندی اشتباه را به حداقل می‌رساند.

① هدف در SVM هدف پیدا کردن تابع f_{opt} (به بدترین و بدترین) به سبب پیدا کردن y (متغیر هدف) با کمترین regression

خط و مساحت margin است. در این الگوریتم به ما اجازه می‌دهد تا به هر اندازه که می‌خواهیم خط را تغییر دهیم و باید در نظر بگیریم خط و مساحت margin که تکرار گیرند (tolerance) است و می‌تواند.



هدف پیدا کردن
نقطه