

# به نام خداوند بخشنده مهربان



عنوان پژوهش و گزارش:

تمرین سری سوم (۳)

نگارش:

نگار درویشی

اساتید گرامی

جناب آقای دکتر فراهانی ، جناب آقای دکتر خردپیشه

## چکیده و مقدمه

تلفن های همراه بیش از ۴۰ سال است که به خوبی در بازار توسعه یافته اند. همانطور

که به تدریج در حال تکامل بودند

بازار رقابتی ، توسعه دهندگان شروع به اضافه کردن دیگر

فن آوری به توابع اساسی مانند ارسال پیام کوتاه و تماس کردند. بنابراین امروزه ، تلفن

ما بسیار سبک تر و راحت تر

نسبت به گذشته است. مجموعه داده های ما از ویژگی های مختلف تلفن های همراه

تشکیل شده است.

هدف مقایسه روشهای مختلف طبقه بندی مورد استفاده در

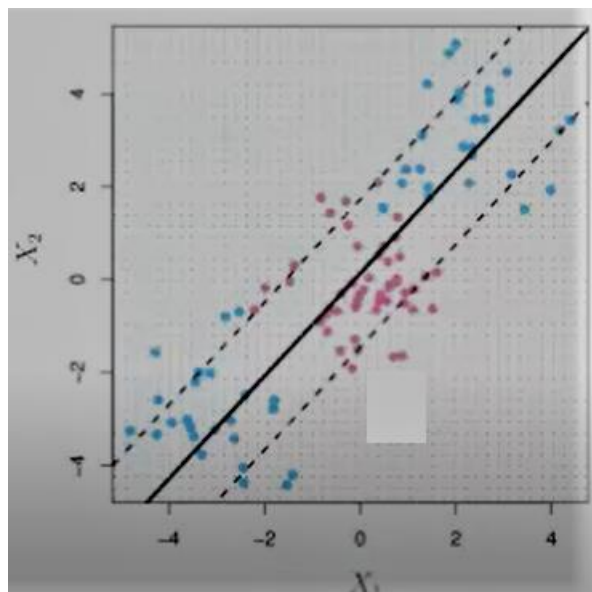
پیش بینی قیمت تلفن همراه و کشف الگوریتم ها

و بسته ها در پایتون است.

**سوال (۱) :** در خصوص کرنل های پرکاربرد روش SVM تحقیق کنید . به صورت کلی چرا ما از ایده کرنل در بحث SVM بهره میبریم . آیا میتوان در خصوص کرنل ها و استفاده ی آنها حکم کلی داد . به طور مثال بگوییم از کرنل RBF در این مواقع خاص استفاده میکنیم .

قبل از پاسخ به این سوال ابتدا به مقدماتی می پردازیم.

زمانی که داده های ما به صورت شکل زیر در یک فضای دو بعدی قرار داشته باشند .  
یعنی دیتاست ما به این صورت شکل زیر باشد :



ما نمیتوانیم داده های خود را به صورت خطی جدا و کلسیفای کنیم . یعنی کلسیفایر

$$f(x) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \text{ کارساز نیست.}$$

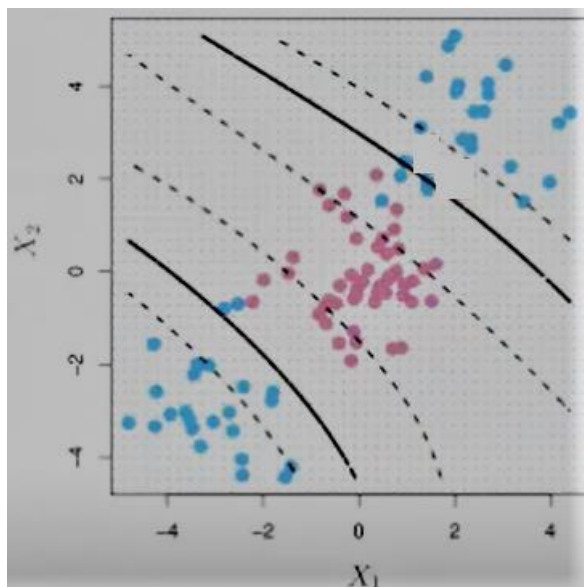
بعنوان راه حل این مشکل Feature Expansion انجام می دهیم . یعنی در قبل که دو فیچر  $X_1, X_2$  داشتیم را با اعمالی مثل به توان دو رساندن و ضرب کردن مانند زیر گسترش می دهیم.

$$(X_1, X_2, X_1^2, X_2^2, X_1X_2)$$

و حالا به آن کلسیفایر قبلی عبارات جدیدی را اضافه میکنیم . مثل اینکه آن کلسیفایر قبلی را به توان دو می رسانیم :

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$

و این باعث می شود که داده های صفحه ی قبل را اینگونه مانند شکل زیر جدا کنیم :



$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 + \beta_6 X_1^3 + \beta_7 X_2^3 + \beta_8 X_1 X_2^2 + \beta_9 X_1^2 X_2 = 0$$

حالا به جای کلسیفایر  $f(x)$  قبلی می نویسیم :

$$f(x) = \beta_0 + \sum_{i=1}^n \alpha_i \langle x, x_i \rangle \quad - n \text{ parameters}$$

یا

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i \langle x, x_i \rangle$$

$S$  is the support set of indices  $i$  such that  $\hat{\alpha}_i > 0$

که  $f(x)$  را به فرم ضرب داخلی نوشتیم. که در آن

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^p x_{ij} x_{i'j} \quad - \text{inner product between vectors}$$

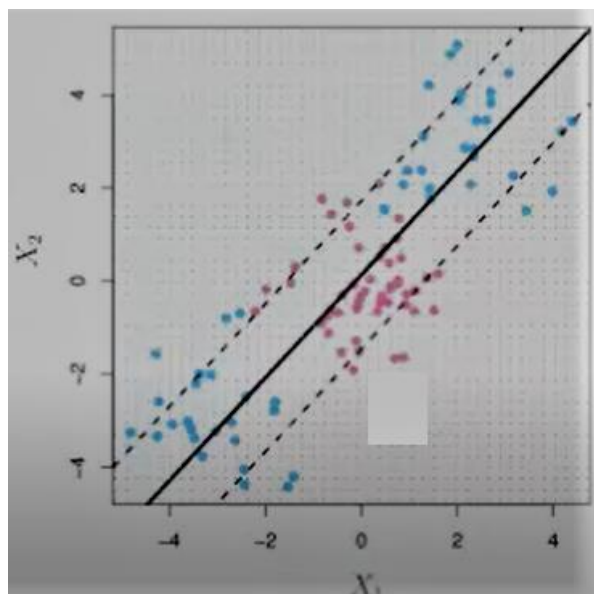
حال می توانیم به جای ضرب داخلی این دو بردار عبارت بالا از کرنل این دو بردار استفاده کنیم:

$$K(x_i, x_{i'}) = \left( 1 + \sum_{j=1}^p x_{ij} x_{i'j} \right)^d$$

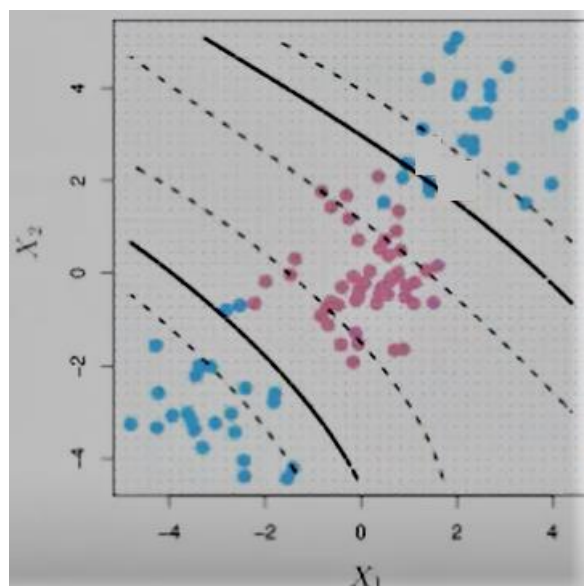
و در نهایت چیزی که حاصل می شود به صورت زیر است:

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i K(x, x_i)$$

که زمانی که شما از این نوع کرنل استفاده کردید، به ازای مثلاً  $d=2,3$  اتفاقی که می افتد این است که برای مثال در شکل زیر



که خطی که به عنوان کلسیفایر داشتیم بعنوان مثال تبدیل به یک رویه می شود و داده ها را می توانیم به راحتی جدا کنیم.

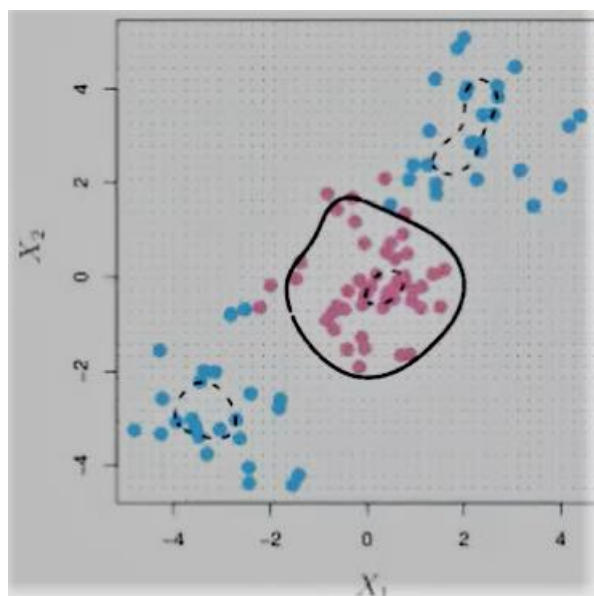


و یک نوع کرنل دیگر می تواند Radial Kernel باشد. و عبارت زیر را به جای ضرب داخلی آن دو بردار جایگزین میکنم :

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2)$$

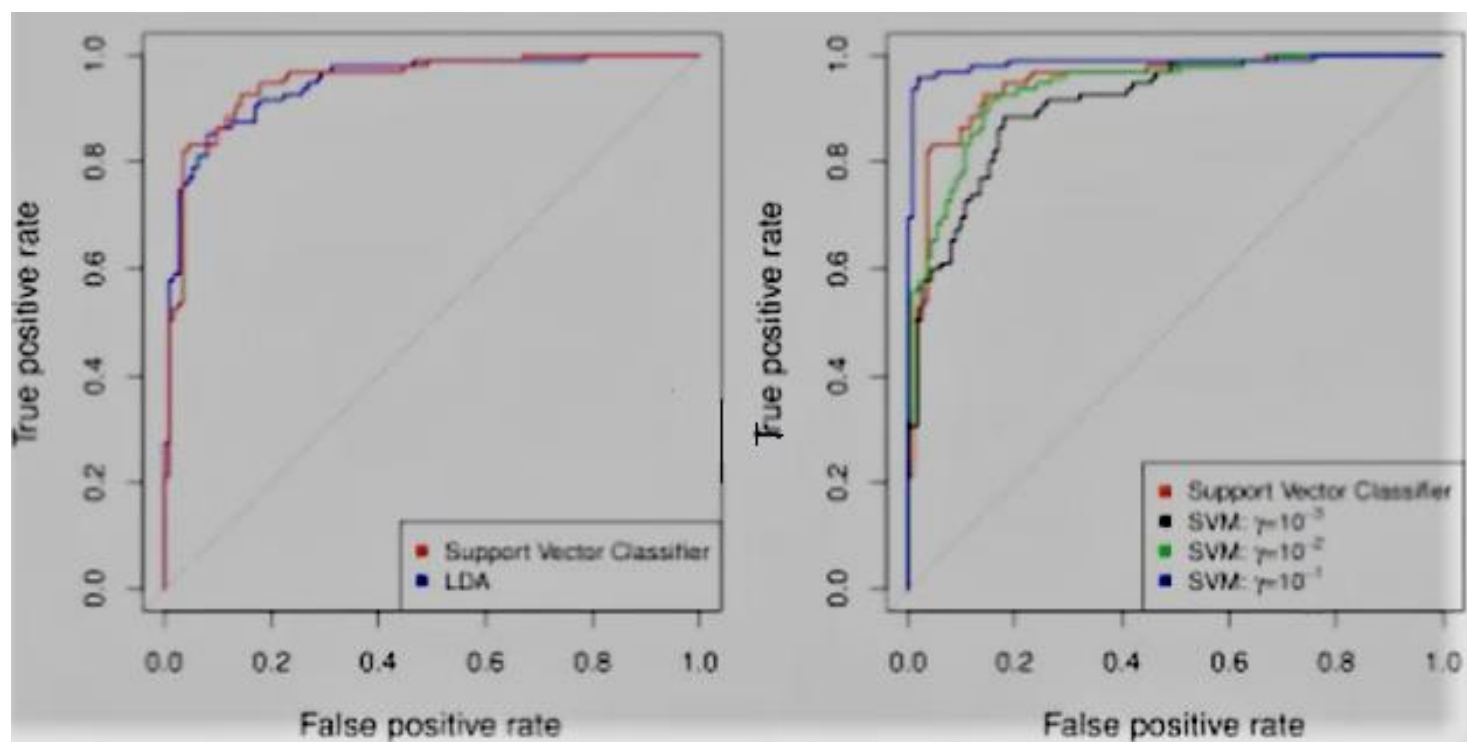
$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i K(x, x_i)$$

و اینگونه به شکل زیر داده های ما کلسیفای می شود :



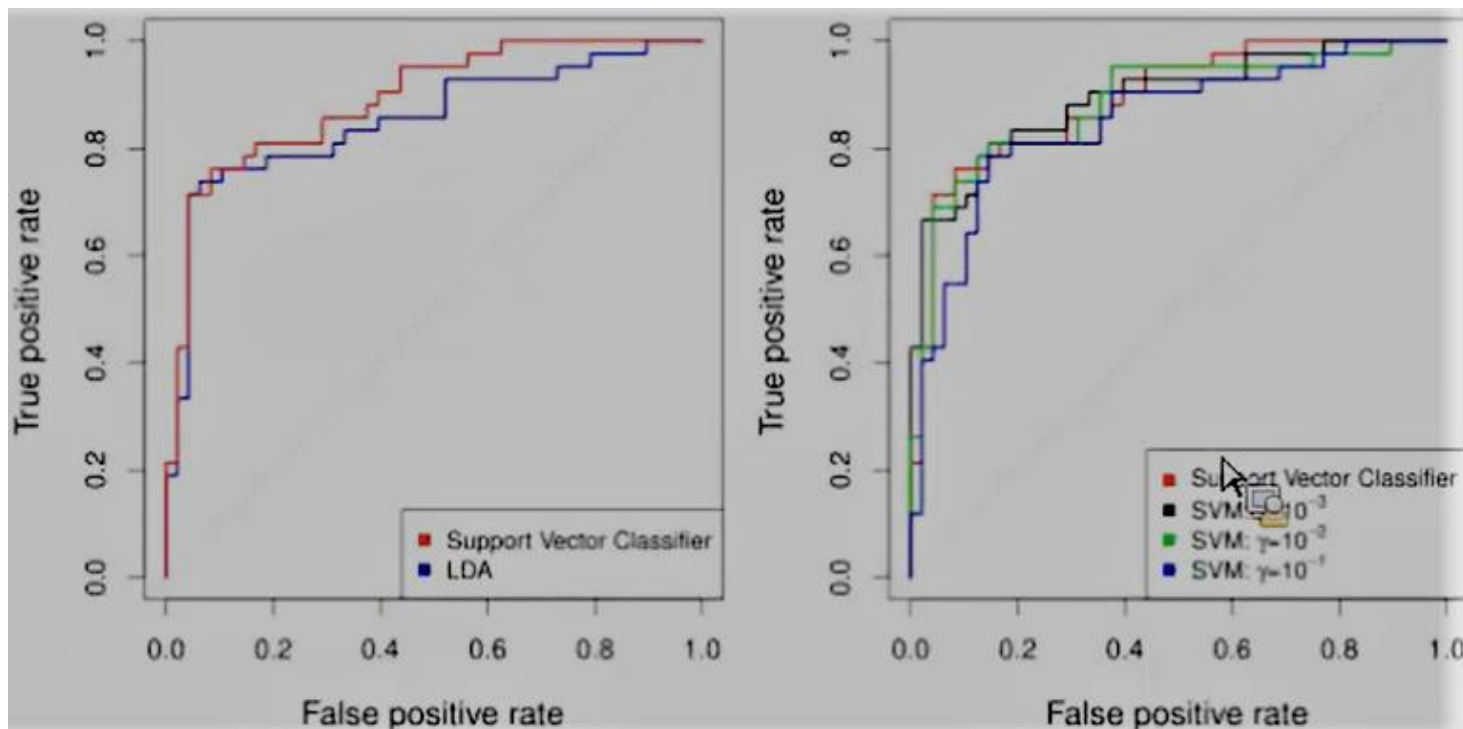
و ممکن است به ازای  $\gamma$  های مختلف سایزهای مختلفی از اینها را داشته باشیم و به شکل های مختلفی بتوانیم این داده ها را جدا کنیم.

اگر به مثال Heart Data که مربوط به بیماری های قلبی ست توجه کنیم . در شکل زیر سمت چپ linear discreament analysis را با support vector classifier روی داده ی ترین مقایسه کرده است . در شکل سمت راست به ازای  $\gamma$  های مختلف دیاگرام true positive rate و false positive rate که ROC curves بود را در نظر می گیریم. و می بینیم که به ازای svm با این مقادیر گاما که استفاده کردیم و ابتدا این نمودار آبی و سپس svm classifier معمولی را داریم سپس نمودار سبز و بعد آبی.

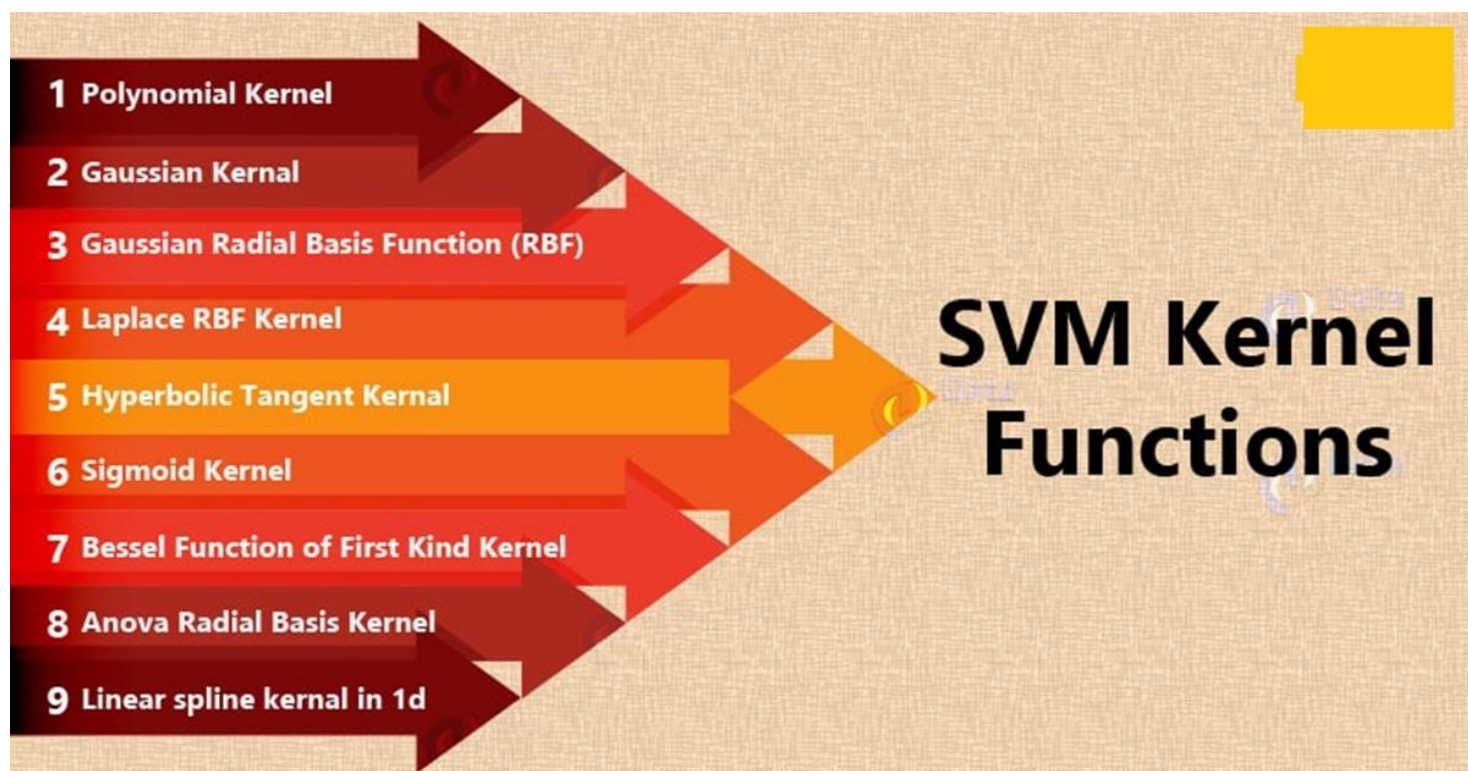




اما در شکل زیر که این عملیات روی داده ی تست انجام شده می بینیم که نمودار آبی که خوب بود در اینجا دیگر خوب نیست.



به طور کلی ما به این دلیل از کرنل استفاده می کنیم که داده هایی که به طور خطی جدایی پذیر نبودند در فضاها ی با بعد بالاتر و کارهایی که به روی آنها انجام می دهیم این داده ها را جدا کنیم.



الگوریتم های SVM از مجموعه ای از توابع ریاضی استفاده می کنند که به عنوان کرنل تعریف می شوند. الگوریتم های مختلف SVM از انواع مختلف توابع کرنل استفاده می کنند. این توابع می توانند انواع مختلفی داشته باشند. به عنوان مثال خطی ، غیرخطی ، چند جمله ای ، تابع پایه شعاعی (RBF) و سیگموئید.

توابع کرنل را برای داده های توالی ، نمودارها ، متن ، تصاویر و همچنین بردارها معرفی کنید. پرکاربردترین نوع عملکرد هسته RBF است. زیرا دارای پاسخ محلی و متناهی در کل محور  $X$  است.

## کرنل چند جمله ای (Polynomial kernel) :

در پردازش تصویر محبوب است. معادله عبارت است از:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$$

که  $d$  درجه چند جمله ای است.

## کرنل گوسی (Gaussian kernel) :

این یک کرنلی است که برای اهداف عمومی استفاده می شود. زمانی که هیچ دانش قبلی در مورد داده ها وجود ندارد. معادله عبارت است از:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

## عملکرد پایه شعاعی گوس (RBF) (Gaussian radial basis function) :

این یک کرنلی است که برای اهداف عمومی استفاده می شود. زمانی که هیچ دانش قبلی در مورد داده ها وجود ندارد. معادله عبارت است از:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

برای :

$$\gamma > 0$$

گاهی اوقات با استفاده از پارامتر:

$$\gamma = 1/2\sigma^2$$

**کرنل RBF لاپلاس (Laplace RBF kernel) :**

این کرنل عمومی است. استفاده می شود زمانی که هیچ دانش قبلی در مورد داده ها وجود ندارد. معادله عبارت است از:

$$k(x, y) = \exp \left( -\frac{\|x - y\|}{\sigma} \right)$$

**کرنل مماس هذلولی (Hyperbolic tangent kernel) :**

می توانیم از آن در شبکه های عصبی استفاده کنیم. معادله عبارت است از:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$$

، برای برخی (نه هر)  $k > 0$  و  $c < 0$ .

## کرنل سیگموئید (Sigmoid kernel) :

می توانیم از آن به عنوان پروکسی شبکه های عصبی استفاده کنیم. معادله عبارت است از :

$$k(x, y) = \tanh(\alpha x^T y + c)$$

## عملکرد Bessel از نوع کرنل اول ( Bessel function of the first kind ) : (Kernel)

ما می توانیم از آن برای حذف مقطع عرضی در توابع ریاضی استفاده کنیم. معادله عبارت است از:

$$k(x, y) = \frac{J_{v+1}(\sigma \|x - y\|)}{\|x - y\|^{-n(v+1)}}$$

که J اولین نوع تابع بسل است.

## کرنل پایه شعاعی ANOVA (ANOVA radial basis kernel) :

ما می توانیم از آن در مشکلات رگرسیون استفاده کنیم. معادله عبارت است از:

$$k(x, y) = \sum_{k=1}^n \exp(-\sigma (x^k - y^k)^2)^d$$

## کرل خطی خطی بصورت یک بعدی (Linear splines kernel in one-dimension):

هنگام کار با بردارهای بزرگ داده پراکنده بسیار مفید است. این اغلب در طبقه بندی متن استفاده می شود. کرل splines همچنین در مشکلات رگرسیون عملکرد خوبی دارد. معادله عبارت است از:

$$k(x, y) = 1 + xy + xy \min(x, y) - \frac{x + y}{2} \min(x, y)^2 + \frac{1}{3} \min(x, y)^3$$

**سوال (۲) : قبلا با دیتاست کلاس بندی قیمت موبایل در کگل کار کرده ایم . بر روی دیتاست ، روش SVM را اجرا کنید . ( استفاده از پکیج ها همانند sklearn مجاز است . )**

ابتدا مقدماتی را معرفی میکنیم. مفهوم های سوال دو و سه را وابسته و ممتد بیان کردیم. روش های طبقه بندی و ارزیابی :

- K-Nearest Neighbors
- Random Forest
- Support Vector Machine
- Neural Network

ما از ۶۰٪ داده به عنوان مجموعه آموزش استفاده می کنیم , بقیه ۴۰٪ به عنوان مجموعه تست. معیارهای مقایسه هر روش طبقه بندی این است : دقت تست

ماشین بردار پشتیبان (SUPPORT VECTOR MACHINE) :

الگوریتم ماشین بردار پشتیبانی در بسته "scikit-learn" تعبیه شده است. در مجموعه داده ۴ کلاس در محدوده قیمت وجود دارد اما svm برای مشکلات طبقه بندی باینری طراحی شده است. در این بسته مسئله طبقه بندی چند طبقه را می توان به چند مسئله طبقه بندی باینری تقسیم کرد و هر یک (OVO) یک مدل طبقه بندی باینری را آموزش داد. از این رو این برنامه اولیه محدب را برای هر جفت کلاس و برای ۶ طبقه بندی (4 x 3/2) برای کلاس i و j حل می کند:

$$\min \frac{1}{2} (w)^T w + C \sum_{i=1}^n (\zeta_i)$$

$$\text{subject to } ((w)^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ ((w)^T \phi(x_i) + b) \leq -1 + \zeta_i, \zeta_i \geq 0$$

where  $x_i \in R^p$ ,  $\phi(x_i)$  maps  $x_i$  into a higher dimensional space and  $y_i = \{-1, 1\}^n$ . And  $C > 0$  is the regularization parameter. Here, let  $C = 1$ . The kernel function of radial SVM is:

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) = \phi(x_i)^T \phi(x_j)$$

که  $\gamma = \frac{1}{20} = 0.05$  در اینجا از آنجا که ۲۰ ویژگی وجود دارد. سپس برنامه

دوگانه مربوطه را حل کنید تا خروجی های بهینه  $w$  و  $b$  را بدست آورید.



و با این تابع تصمیم گیری:

$$\text{sgn}(\omega^T \phi(x_i) + b)$$

برچسب های پیش بینی شده جمع آوری و ثبت می شوند. برچسب های پیش بینی شده نهایی با اکثریت آرا پیش بینی ها از ۶ برنامه دوگانه تعیین می شود.

تاکنون الگوریتم اصلی بسته که از آن استفاده می کند ، نزول مختصات (coordinate descent) است.

توسط پایتون امتیاز دقت این روش: ۹۴,۸٪ است.

SVM در SKLEARN (SVM IN SKLEARN) :

## 1 - SVM خطی (Linear SVM) :

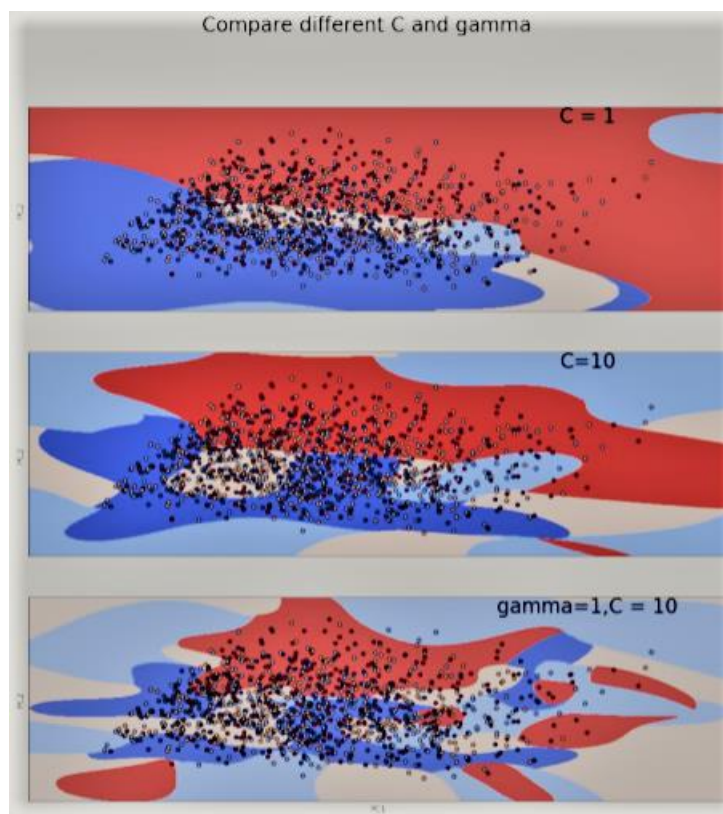
Kernel SVM یک روش عالی با دقت بالاتر از است SVM خطی اما گاهی اوقات ، به عنوان مثال ، هنگامی که تعداد نمونه ها کوچک است ، نیازی به استفاده از ترفند هسته نیست. در عوض ، SVM خطی مناسب تر خواهد بود. در sklearn بسته ، الگوریتمی نیز برای حل این بهینه سازی وجود دارد برنامه با همان برنامه اولیه Kernel SVM.

SVM خطی از آنجا که آموزش می دهد فقط نیاز به حل ۴ برنامه از این دست دارد

مجموعه داده با OVR ، نه OvO.

## 2 - پارامتر C ، گاما (Parameter C, gamma) :

C و گاما دو پارامتر SVM در sklearn هستند (فقط C اگر خطی باشد). تغییر C و گاما باعث می شود نتیجه دقیق طبقه بندی C. بزرگتر ، صاف تر است و معکوس تر عملکرد تصمیم خواهد بود. و بیشتر گاما ، بهتر است که شکل بردارهای قطار را بدست آورد. بنابراین ، افزایش C و گاما منجر به نصب بیش از حد می شود.



هنگامیکه از پکیج sklearn برای svm استفاده کردم. از کرنل RBF استفاده شده است.. Score ای در محدوده ی عدد 0/95 بدست آمد. یعنی svm ما روی داده های آموزشی به درستی فیت شده و روی داده های تست هم با دقت 90 درصد و تاحدودی عالی عمل می کند و برچسب های داده های تست را به خوبی شناسایی می کند.

```
[141  3  0  0]
[  7 138  0  0]
[  0  7 143  8]
[  0  0  3 150]]
```

**سوال (۳) : برای سوال ۲ حداقل ۵ حالت مختلف از قبیل کرنل ها و**

**پارامترها را بررسی کنید و نتایج آن را گزارش دهید .**

ابتدا قبل از پاسخ به سوال:

کرنل ها و پارامترها در امتداد و ارتباط با سوال (۲):

## K-نزدیکترین همسایگان (K-NEAREST NEIGHBORS) :

الگوریتم k - نزدیکترین همسایگان برای طبقه بندی در بسته Sklearn

فاصله اقلیدسی بین را اندازه می گیرد . فاصله اقلیدسی به صورت زیر

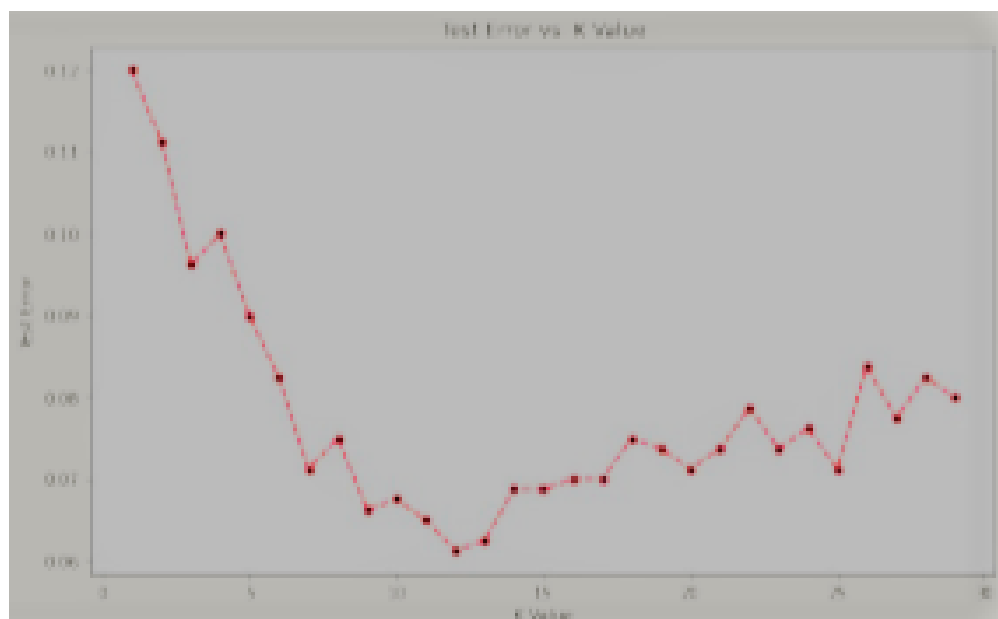
$$distance(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

محاسبه می شود:

که در آن  $\mathbf{x}$  و  $\mathbf{y}$  نقاط داده در  $n$ -space هستند. مقدار  $k$  مطلوب با

ترسیم  $k$  مختلف انتخاب شد مقادیر در برابر خطای آزمون آنها. در این

پروژه دقت برابر است با ۹۳/۹ درصد با  $k = 12$ .



## جنگل تصادفی (RANDOM FOREST) :

جنگل تصادفی بر اساس درخت تصمیم و پیش بینی ساخته شده است

پیش بینی متوسط طبقه بندی کننده های منفرد هستند.

برای طبقه بندی جنگل تصادفی در بسته Sklearn ، هر کدام

درخت در گروه از یک نمونه بوت استرپینگ از ساخته شده است

مجموعه آموزش

ما دو پارامتر اصلی ، تعداد برآوردگرها را انتخاب می کنیم

و تعداد حداکثر ویژگی ها ، با دقت آزمون. از آنجا که

حداکثر دقت زمانی حاصل می شود که  $n\_estimators = 100$  و

$max\_features = 14$  ، نتیجه می گیریم که این دو پارامتر است

بهینه سازی مدل

در این پروژه دقت ۹۲,۱٪ است.

## شبکه عصبی (Neural Network) :

ما برای پیش بینی شبکه عصبی پی در پی ایجاد می کنیم . قیمت ما از سه لایه نرون با تراکم ۱۶ ، ۱۲ ، و به ترتیب ۴. ما از Relu به عنوان تابع فعال سازی برای دو مورد اول استفاده می کنیم

لایه ها و softmax به عنوان عملکرد خروجی Relu. تابع قسمت غیر منفی متغیر را حفظ می کند. تابع softmax بردار مقادیر را گرفته و آن را به بردار مقادیر تبدیل می کند که جمع آنها ۱ است. ما از آدم به عنوان بهینه ساز استفاده می کنیم. آدم محبوب است روشی که با نزول شیب تصادفی استفاده می شود. برخلاف نزول شیب منظم ، شیب تصادفی نزول فقط از یک دسته تصادفی یا تصادفی استفاده می کند نمونه های آموزشی برای به روزرسانی پارامترها. بنابراین ، تصادفی شیب نزولی زمانی مناسب است که مجموعه داده بسیار مناسب باشد بزرگ بهینه ساز Adam فقط تصادفی نیست شیب در هر تکرار ، اولین لحظه را نیز به روز می کند (میانگین) و دومین لحظه خام (بدون مرکز) واریانس) بر این اساس.

الگوریتم بهینه ساز Adam به عنوان نشان داده شده است در زیر:

```

Require:  $\alpha$ : Stepsize
Require:  $\beta_1, \beta_2 \in [0, 1)$ : Exponential decay rates for the moment estimates
Require:  $f(\theta)$ : Stochastic objective function with parameters  $\theta$ 
Require:  $\theta_0$ : Initial parameter vector
 $m_0 \leftarrow 0$  (Initialize 1st moment vector)
 $v_0 \leftarrow 0$  (Initialize 2nd moment vector)
 $t \leftarrow 0$  (Initialize timestep)
while  $\theta_t$  not converged do
   $t \leftarrow t + 1$ 
   $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$  (Get gradients w.r.t. stochastic objective at timestep  $t$ )
   $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update biased first moment estimate)
   $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update biased second raw moment estimate)
   $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Compute bias-corrected first moment estimate)
   $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Compute bias-corrected second raw moment estimate)
   $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameters)
end while
return  $\theta_t$  (Resulting parameters)

```

ما همچنین از تکنیک ترک تحصیل ( dropout technique ) برای جلوگیری از

پتانسیل استفاده می کنیم. نصب بیش از حد Dropout به طور تصادفی برخی

واحدها را نادیده می گیرد و آنها در هر دو مسیر جلو و عقب گنجانده نخواهد شد.

دقت مدل شبکه عصبی در مجموعه تست ۹۳,۵٪ است.

## SVM در SKLEARN (SVM IN SKLEARN) :

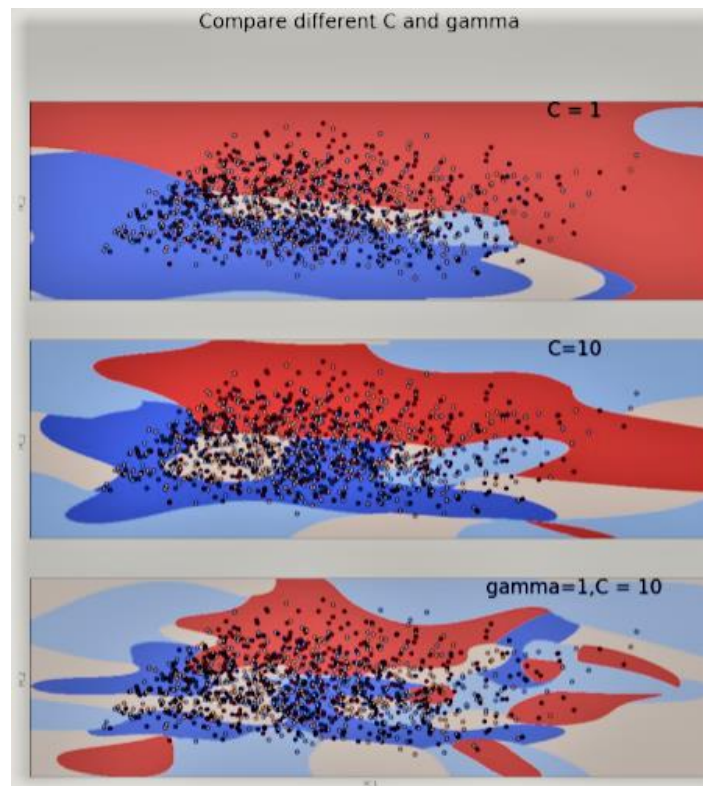
### 1- SVM خطی (Linear SVM) :

Kernel SVM یک روش عالی با دقت بالاتر از است SVM خطی اما گاهی اوقات ، به عنوان مثال ، هنگامی که تعداد نمونه ها کوچک است ، نیازی به استفاده از ترفند هسته نیست. در عوض ، SVM خطی مناسب تر خواهد بود. در sklearn بسته ، الگوریتمی نیز برای حل این بهینه سازی وجود دارد برنامه با همان برنامه اولیه Kernel SVM، SVM خطی از آنجا که آموزش می دهد فقط نیاز به حل ۴ برنامه از این دست دارد مجموعه داده با OvR ، نه OvO.



## 2- پارامتر C ، گاما (Parameter C, gamma) :

C و گاما دو پارامتر SVM در sklearn هستند (فقط C اگر خطی باشد). تغییر C و گاما باعث می شود نتیجه دقیق طبقه بندی C. بزرگتر، صاف تر است و معکوس تر عملکرد تصمیم خواهد بود. و بیشتر گاما، بهتر است که شکل بردارهای قطار را بدست آورد. بنابراین، افزایش C و گاما منجر به نصب بیش از حد می شود.



نتیجه :

Method	Accuracy
K-Nearest Neighbors	93.9%
Random Forest	92.1%
SVM	94.8%
Neural Network	93.5%

در انتها به ازای گام‌های بزرگ ، **overfitting** رخ می دهد.

Gamma=scale	C=1	C=0.01	C=100
rbf	0.953	0.581	0.968
linear	0.966	0.968	0.971
poly	0.955	0.763	0.978
sigmoid	0.185	0.24	0.17

Gamma=0.000005	C=1
rbf	0.946
linear	0.966
poly	0.966
sigmoid	0.24

**سوال (۴) :** برای سوال ۲ سعی کنید بحث **margin** و **margin soft** را بررسی کنید و نتایج آن را گزارش دهید.

ابریارامتر  $C$  برای ما زمانی که مدل ما سیمپلی را به اشتباه کلسیفای کرده است یک جریمه ای به مدل می دهد. زمانی که مقدار  $C$  را خیلی کم در نظر گرفتیم یعنی یعنی حاشیه ی ما خیلی نرم می شود و نسبت به خطاهای مدل حساسیت کمتری هم داریم پس خطاها بیشتر می شوند و **overfitting** نیز رخ می دهد. ولی اگر زمانی  $C$  را مقدار خیلی بزرگتری در نظر بگیریم حاشیه سخت می شود و به تبع خطاها خیلی کمتر می شوند و ممکن است مرزی نتوانیم فیت کنیم همانطور که تغییرات  $C$  را می بینید.

**سوال ( ۵ ) : مهندسی ویژگی یکی از بخش های مهم در فرایندهای**

**علم داده میباشد . بر روی دیتاست موارد زیر را اجرا کنید .**

**(آ): بر روی فیچر power battery از روش binning استفاده کنید .**

**(حداقل سه اندازه مختلف برای بین ها در نظر بگیرید و حتی سائز بین**

**ها را نامساوی در نظر بگیرید . )**

با استفاده از bin بندی داده ها به سه قسمت و دسته ی مختلف و متفاوت تقسیم شدند  
در pandas با تابع qcut.

```
1 # 5-a
2
3 data_bin = data.copy()
4 data_bin['battery_power'] = pd.qcut(data_bin['battery_power'], 3, labels = ['small', 'medium', 'large'])
5 data_bin.head()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	nm_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	t
0	small	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	
1	medium	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	
2	small	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	
3	small	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	
4	large	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	

5 rows x 21 columns

**(ب) بر فیچرهای کتگوریکال در دیتاست encoding hot one را اعمال کنید . چرا ما باید به صورت کلی از این کدگذاری بهره ببریم.**

One hot encoding را روی داده ها و فیچرها اعمال کردیم. ما زمانی از تکنیک one hot encoding استفاده می کنیم که وقتی داده های ما categorical هستند و چند حالت ما این حالت ها را با استفاده از این تکنیک به عدد تبدیل میکنیم تا این اعداد را به مدل خود بدهیم. معمولاً برای تبدیل داده های categorical به اعداد از این تکنیک one hot encoding استفاده می کنیم.

(ج) بررسی کنید آیا استفاده از تبدیل هایی از قبیل  $\log$  transform و یا تبدیل نمایی در اینجا کاربرد دارد . به صورت کلی چرا از این دست تبدیلات بهره میبریم . ( در این بخش شما مجاز هستید اگر تبدیل دیگری را مناسب میدانید اعمال کنید این بخش نمره امتیازی برای شما خواهد داشت . حتما دلیل استفاده از تبدیل استفاده شده را بیان کنید.)

وقتی که دیتاها و سمپل ها فاصله ی زیادی از هم دارند و به اصطلاح داده های پرت داریم بعضی از روش ها مثل svm حساس به نوع پراکندگی داده ها هستند و با استفاده از تبدیل  $\log$  یا  $\minmax$  داده ها مقدارشان در یک  $range$  خاصی قرار می گیرد و به هم نزدیک می شوند.

(د) یک فیچر جدید به نام مساحت یا حجم گوشی بسازید.

فیچر مساحت ساختم و تاثیر منفی دارد. می توانیم از متغیر طول و عرض ، قطر بدست آوریم.

**سوال (۶) : برای هریک از حالت های سوال ۵ یک مدل SVM بسازید و بررسی کنید یکبار هم هر ۵ حالت را باهم اعمال کنید و مدل SVM روی آنها اجرا کنید . حاصل این مدل ها را گزارش کنید.**

- روی آن سه دسته ای که داشتیم تاثیری نداشت و خروجی الگوریتم تغییر نکرد و score , 0/95 بود و وقتی فیچر power battery را حذف کردیم score , 0/79 شد و کم شد و برای دو دسته ی دیگر هم تغییر نبود.

- وقتی one hot encoding را به کار بردیم. Score افزایش یافت و 0/958 شد.

- و ساخت فیچر مساحت این score را خیلی کم کرد. و وقتی هم که هر ۵ حالت را با هم اعمال کردیم و svm را روی آن ها اجرا کردیم score , 0/97 شد.

