دانشکده مهندسی برق

# تکلیف کامپیوتری درس بازشناسی آماری الگو، سری چهارم

**استاد**

دکتر ابریشمی مقدم

نیمسال اول ۱۴۰۵-۱۴۰۴

مهلت تحویل: ۱۴۰۴/۹/۲۸

**با سلام و آرزوی شادی، موفقیت و سلامتی؛**

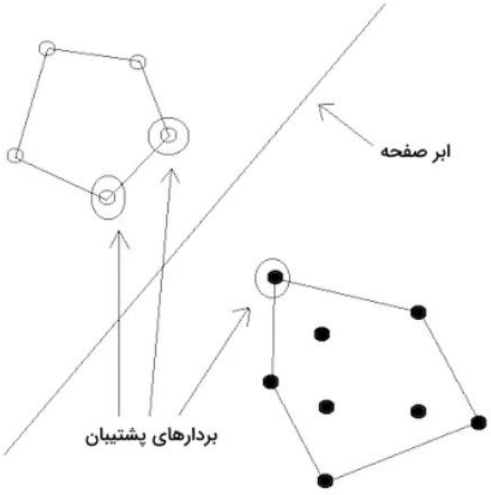
**لطفا در تحویل پاسخ‌های خود موارد زیر را مدنظر داشته باشید:**

- فقط برنامه‌هایی که به زبان ترجیحا پایتون و یا متلب باشند قابل قبول خواهند بود.
- تحویل همزمان گزارش و کدها الزامی است.
- گزارش باید شامل خروجی‌های کدهای نوشته شده باشد، که موارد خواسته شده در سوالات هستند و سایر توضیحات خواسته شده دیگر در متن سوالات نیز پاسخ داده شود (از آوردن کد در گزارش خودداری کنید).
- لطفا کدهای برنامه به صورت ماجولار و همراه با توضیحات کافی باشند؛ طوری که بخش‌های مختلف برنامه کاملا قابل تفکیک بوده و اجرا و ارزیابی هر بخش توسط کاربر به آسانی و بدون نیاز به ورود به جزییات برنامه میسر باشد.
- فایل تحویلی پاسخ شما باید تنها یک فایل زیپ، تحت عنوان "SPR_CHW4_Student ID" محتوی دو پوشه باشد. گزارش خود را در پوشه اول با عنوان "Report" و کدهای خود را ترجیحا به فرمت Jupyter Notebook در پوشه "Codes" قرار دهید.
- با این که همکاری، مشورت، و استفاده از ابزارهای کمکی در حل سوالات پیشنهاد می‌شود، حتما به صورت مستقل به نوشتن کدها و گزارش بپردازید.
- ممکن است از دانشجویی خواسته شود در زمانی که تعیین خواهد شد جزئیات کدش را در جلسه‌ای مجازی توضیح دهد، نتایج را تحلیل کند و حتی تغییراتی در پارامترهای کد اعمال کند. در صورتی که دانشجویی تمارین را تحویل داده باشد اما نتواند کد خود را توضیح دهد و یا تغییراتی روی آن اعمال کند، و یا اینکه کد یا گزارش تحویلی به پاسخ دیگران شباهت غیرمنطقی داشته باشد، نمره تمرین صفر لحاظ شده و نمره‌ای منفی هم لحاظ خواهد شد.
- در صورت وجود هرگونه سوال یا ابهام، مشکل مربوطه را با آی دی تلگرام زیر در میان گذارید:

@omid_Emaa

## 1 - SVM For Classification

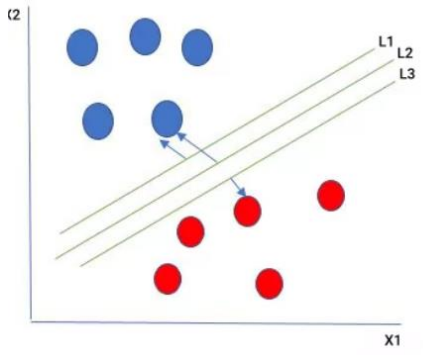
ماشین بردار پشتیبان یا SVM نوعی الگوریتم یادگیری ماشین از نوع نظارت شده است که برای حل مسائل طبقه‌بندی و رگرسیون مورد استفاده قرار می‌گیرد .به‌طور ویژه، الگوریتم SVM در حل مسائل طبقه‌بندی دودویی که داده‌های دیتاست به دو گروه مجزا دسته‌بندی می‌شوند کاربرد دارد .هدف از به‌کارگیری این الگوریتم، پیدا کردن بهترین خط یا به اصطلاح مرز تصمیمی است که بتواند نقاط داده گروه‌ها یا کلاس‌های مختلف را از یک‌دیگر جدا کند. مرزی که هنگام کار با تعداد زیادی از ویژگی‌ها، اَبر صفحه (Hyperplane) نامیده می‌شود. هر چه فاصله یا حاشیه بیشتری میان ابر صفحه و نزدیک‌ترین نقاط داده از هر کلاس وجود داشته باشد، یعنی دسته‌بندی بهتری توسط الگوریتم SVM صورت گرفته است.



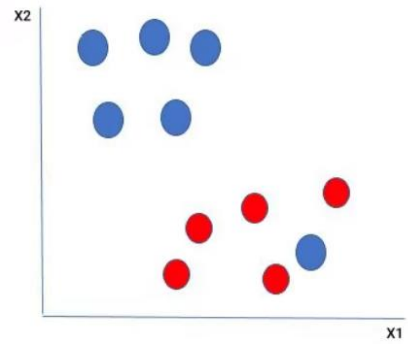

همچنین الگوریتم SVM در یادگیری ماشین و هنگام تحلیل داده‌های پیچیده، زمانی که نتوان داده‌ها را با یک خط راست ساده از یک‌دیگر جدا کرد نیز مفید واقع می‌شود. به این گروه از الگوریتم ماشین بردار پشتیبان، SVM غیر خطی گفته می‌شود که از نوعی تکنیک ریاضیاتی برای نگاشت داده‌ها به فضایی با ابعاد بالاتر و پیدا کردن مرز تصمیم کمک می‌گیرد.

الگوریتم SVM را می‌توان به دو نوع خطی و غیر خطی تقسیم کرد : نوع خطی الگوریتم SVM از یک مرز تصمیم خطی برای جدا کردن نقاط داده کلاس‌های مختلف از یک‌دیگر استفاده می‌کند. این الگوریتم زمانی کاربرد دارد که داده‌ها به‌طور خطی قابل جداسازی باشند. به بیان ساده‌تر، باید بتوان با تنها یک خط راست در فضای ۲ یا چند بعدی، داده‌ها را در کلاس‌های مجزایی طبقه‌بندی کرد. ابر صفحه‌ای که فاصله میان کلاس‌ها را به حداکثر برساند همان مرز تصمیم است اما در نوع غیر خطی زمانی از نوع غیر خطی الگوریتم SVM استفاده می‌شود که نتوانیم داده‌ها را با یک خط راست به دو کلاس مختلف تقسیم کنیم. در SVM غیر خطی، ابتدا داده‌های ورودی از طریق توابعی با عنوان توابع کِرنل (Kernel Functions) به فضایی با ابعاد بالاتر نگاشت شده و سپس به‌طور خطی از

هم جدا می‌شوند .برای ایجاد مرز تصمیم غیر خطی در این فضای ویژگی تغییر یافته، از یک SVM خطی استفاده می‌شود.



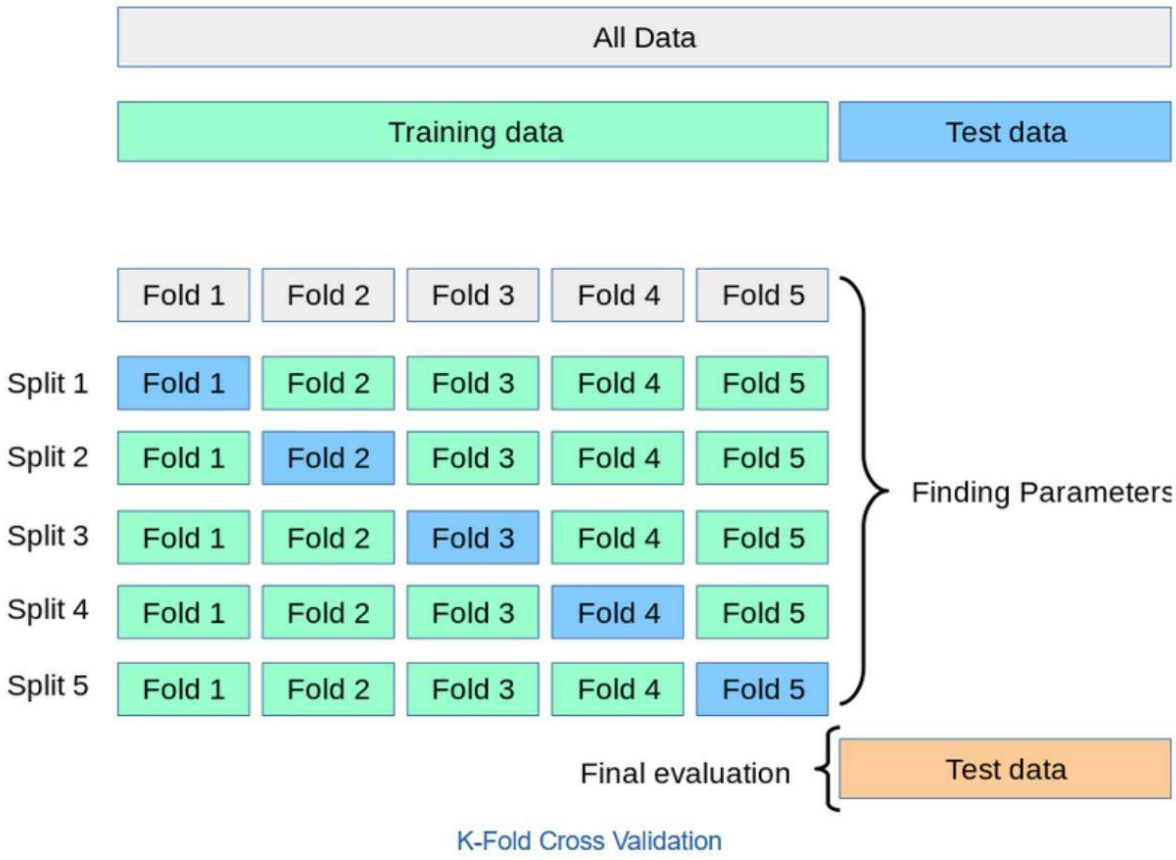

چند ابر صفحه جداکننده داده‌ها را در تصویر بالا مشاهده می‌کنیم. در الگوریتم ماشین بردار پشتیبان و برای تعیین مرز جداسازی، تنها نزدیک‌ترین نقاط داده یا همان بردارهای پشتیبان به ابر صفحه اهمیت دارند و ابر صفحه‌ای را انتخاب می‌کنیم که بیشترین جدایی یا حاشیه تمایز را میان دو کلاس ایجاد کند. بنابراین، ابر صفحه‌ای را برمی‌گزینیم که فاصله آن تا نزدیک‌ترین نقاط داده از هر طرف بیشینه باشد. در صورت وجود، چنین ابر صفحه‌ای را حاشیه سخت (Hard Margin) می‌نامیم. با این توصیف، در تصویر بالا ابر صفحه L۲ انتخاب می‌شود.




در اینجا یک دایره آبی در گروه قرمزها قرار گرفته است. در ابتدا شاید دسته‌بندی این دو گروه دشوار به‌نظر برسد، اما الگوریتم SVM در یادگیری ماشین، دایره آبی را نمونه ای پَرت از گروه آبی در نظر گرفته و با نادیده گرفتن آن، بهترین ابر صفحه، یعنی مرز تصمیمی که بیشترین فاصله را با هر دو کلاس داشته باشد انتخاب می‌کند .به بیان ساده‌تر، می‌گوییم الگوریتم SVM نسبت به نمونه‌های پرت مقاوم است. در این شرایط نیز مانند زمانی که دیتاست فاقد نمونه پرت است، الگوریتم SVM مرز تصمیم با بیشترین حاشیه را پیدا کرده و هر بار که نمونه‌ای از این حاشیه رد شود، مقدار جریمه‌ای به ابر صفحه اضافه می‌کند. در نتیجه به چنین ابر صفحه‌ای حاشیه نرم (Soft Margin) گفته می‌شود. اما گاهی نمی‌توان داده‌ها را به شکل خطی از هم جدا کرد. راه‌حل الگوریتم SVM برای حل چنین مسئله‌ای، ساخت یک متغیر جدید با استفاده از یک کرنل است.

همچنین در این تمرین تأثیر انتخاب کرنل و C در الگوریتم SVM بر نتایج خروجی مورد بررسی قرار می‌گیرد. در این تمرین قصد داریم با استفاده از یک روش، مقدار بهینه C را پیدا کنیم. واضح است که به ازای مقدار بهینه بیشترین صحت را به دست می آوریم. احتمالا راه حل پیشنهادی شما، تست دسته‌بند ساخته شده به ازای مقدارهای متفاوت می باشد. اما این کار اشتباه است! زیرا در این صورت، از مجموعه داده تست به عنوان مجموعه داده آموزش استفاده شده است و از آنجایی که ما در حال تنظیم کردن مدل روی داده تست هستیم، خطای دسته‌بند کمتر از مقدار واقعی آن گزارش می شود. پس در چنین حالتی مدل ما دیگر قادر به تعمیم پیدا کردن و دسته‌بندی مشاهدات جدید نخواهد بود و فرآیندی به نام بیش برازش (overfitting) رخ می دهد. پس به یاد داشته باشید که در مرحله بهینه‌سازی برای تمام مدلهای یادگیری ماشینی، مجموعه داده تست را به طورکامل کنار بگذارید و پس از انتخاب پارامترهای بهینه، دسته‌بند را روی این مجموعه داده ارزیابی کنید.

چگونه این کار صورت میگیرد؟ در ابتدا داده را به دو بخش شامل داده تست و داده آموزش تقسیم کنید. در مرحله آموزش مدل، یک بخش از مجموعه داده آموزش را کنار بگذارید. این مجموعه داده، مجموعه داده ارزیابی (validation set) نام دارد. راههای زیادی به منظور ارزیابی وجود دارد که در این تمرین قصد داریم به معروفترین آنها یعنی k-fold cross validation بپردازیم.



همانطور که در تصویر بالا مشخص است، در k-fold cross validation، ابتدا مجموعه داده به دو بخش مجموعه داده تست و آموزش تقسیم میشود. سپس، مجموعه داده‌های آموزش به k زیرنمونه ( Fold) با حجم یکسان تفکیک می شوند. در هر مرحله از فرایند، تعداد k-1 از این لایه ها را به عنوان مجموعه داده آموزشی و یکی را به عنوان مجموعه داده اعتبارسنجی در نظر گرفته می شود. میزان خطا (یا صحت یا...) روی مجموعه داده اعتبارسنجی محاسبه می گردد و این فرآیند k بار تکرار میشود و هر بار یکی از این k فولد، نقش مجموعه داده اعتبار سنجی را ایفا میکند. این فرآیند منجر به محاسبه k خطا میگردد که میانگین گیری روی آنها صورت میگیرد. نهایتا، مقدار بهینه که به ازای آن بهترین صحت روی مجموعه دادههای اعتبارسنجی به دست آمده است انتخاب می شود. نتیجه و عملکرد نهایی کلاس بند به ازای مقدار بهینه، با اعمال روی مجموعه داده تست مشخص می گردد. در صورت علاقه برای توضیحات بیشتر میتوانید به این لینک مراجعه کنید.

در این تمرین قصد داریم دسته‌بند SVM را بر روی مجموعه داده fashion-MNIST پیاده‌سازی کنیم.

## بارگذاری مجموعه داده:

۱- مجموعه داده را بارگذاری کنید. اگر از پایتون استفاده میکنید، به منظور بارگذاری این مجموعه داده میتوانید از Scikit-learn و چند خط کد زیر کمک بگیرید (میتوانید جهت افزایش سرعت عملکرد برنامه، تنها از ۱۰۰۰۰ تصویر اولیه استفاده کنید).

```
from sklearn.datasets import fetch_openml
fashion_mnist = fetch_openml('Fashion-MNIST')
fmnist_data = fashion_mnist.data[:10000]
fmnist_target = fashion_mnist.target[:10000]
```

## پیاده سازی دسته‌بند SVM:

۲- یک تابع با نام train_svm بسازید که ورودی آن مجموعه‌داده‌ی آموزش و پارامترهای مدل (مانند C و نوع کرنل) و خروجی آن، مدل آموزش‌دیده‌ی SVM باشد.

۳- یک تابع با نام predict بسازید که ورودی آن نقطه‌ی x یا مجموعه‌ای از نقاط X و مدل آموزش‌دیده‌ی SVM باشد و خروجی آن، برچسب حدس‌زده‌شده باشد.

## تست مدل با استفاده از مقدار از پیش تعیین شده:

۴- برای هر کلاس، داده آموزش و تست را به طوری جداسازی کنید که ۲۰٪ از داده ها در مجموعه تست و بقیه در مجموعه آموزش قرار گیرند.

۵- Accuracy را برای حالت (kernel=rbf و C = ۱) محاسبه کنید.

## بهینه سازی مقدار:

۶- با استفاده از 10-fold cross validation، یک نمودار از صحت به ازای مقادیر مختلف پارامتر C (و برای هر کرنل) ترسیم کنید.

راهنمایی:

- ۲۰٪ از داده‌ها را به‌عنوان داده‌ی تست کنار بگذارید و در این مرحله سراغ آن‌ها نروید.
- ۸۰٪ باقی‌مانده‌ی داده‌ها را به ۱۰ بخش مساوی تقسیم کنید.

هر بار یکی از این بخش‌ها را به‌عنوان داده‌ی اعتبارسنجی (Validation) و ۹ بخش دیگر را به‌عنوان داده‌ی آموزش در نظر بگیرید. فرض کنید می‌خواهیم بین مجموعه‌ی مقادیر زیر برای پارامترها، مقدار بهینه را پیدا کنیم :

```
C_values = [0.01, 0.1, 1, 10, 100]
kernels  = ['linear', 'rbf']
```

برای هر مقدار C و کرنل یک مدل روی داده‌های آموزش آن فولدها آموزش دهید. دقت مدل را روی داده‌ی اعتبارسنجی (آن یک بخش) به دست آورید. این کار را ۹ بار دیگر تکرار کنید تا هر بار یکی از ۱۰ بخش نقش داده‌ی اعتبارسنجی را بازی کند. در نهایت، میانگین صحت‌های به‌دست‌آمده از ۱۰ مرحله را به‌عنوان صحت به ازای آن مقدار C و کرنل در نظر بگیرید. تمام مراحل ذکرشده را برای C های مختلف انجام دهید و منحنی میانگین صحت بر حسب C را ترسیم نمایید. برای هر کرنل یک منحنی جداگانه رسم کنید.

۷- مقدار و کرنل بهینه کدام است؟ Accuracy را به ازای مقدار بهینه محاسبه کنید.

راهنمایی: دقت داشته باشید که صحت یک کلاسبند، نتیجه اعمال آن روی مجموعه داده تست میباشد نه مجموعه داده اعتبار سنجی

۸- مراحل قبل را با استفاده از کتابخانه sklearn پیاده‌سازی کنید و نتیجه نهایی را با نتیجه خود مقایسه کنید.

از این به بعد میتوانید از توابع آماده استفاده کنید.

۹- (امتیازی+ ۵٪) جهت کاهش حافظه مورد نیاز و افزایش سرعت الگوریتم ،به ازای مقدار بهینه یافت شده، تعداد دادههای آموزش را با حذف نمونه‌های دارای ابهام کاهش دهید و تعداد نمونه‌های حذف شده را گزارش دهید. (دسته‌بندی را برای تمام داده‌های آموزش انجام دهید و نمونه‌هایی که به درستی توسط داده‌های آموزش دسته-بندی نمیشوند را از مجموعه داده آموزش حذف کنید.)

۱۰- (امتیازی+ ۵٪) دقت دسته‌بندی مجموعه داده تست را، در حالت حذف نمونه‌های دارای ابهام از مجموعه داده آموزش، محاسبه کنید.

۱۱- (امتیازی+ ۵٪) زمان لازم جهت دسته‌بندی مجموعه داده تست را در حالت اولیه و حالت حذف داده‌های دارای ابهام گزارش داده و با یک دیگر مقایسه کنید. (میتوانید از توابع موجود در کتابخانه time استفاده کنید.) راهنمایی: میتوانید، برای مثال ،۵۰ بار هر کدام از حالتهای ذکر شده را انجام داده و زمان محاسبه را اندازه‌گیری کرده و میانگین آنها را با یکدیگر مقایسه کنید.

۱۲- (امتیازی+ ۵٪) با محاسبه p-value نشان دهید که آیا زمان لازم جهت دستهبندی مجموعه تست در دو حالت ذکر شده تفاوتی معنادار دارند یا خیر.

---

## 2: Fisher's Linear Discriminant Analysis

۱۳- از تحلیل تفکیکی خطی فیشر (Fisher's Linear Discriminant Analysis) برای دسته‌بندی گونه-های موجود در مجموعه داده‌ی Seeds استفاده کنید. این مجموعه داده در تمرینهای پیشین در اختیار شما قرار داده شده‌است.

این الگوریتم را به طور کامل از ابتدا پیاده‌سازی کرده و صحت طبقه‌بندی را گزارش کنید.

۱۴- صحت طبقه‌بندی را با تابع موجود در کتابخانه scikit-learn نیز بدست آورده و عملکرد الگوریتم قسمت قبل را با آن مقایسه کنید.

"موفق باشید"