شناسایی آماری الگو

تمرین کامپیوتری شماره ۲

با سلام و آرزوی شادی، موفقیت و سلامتی؛

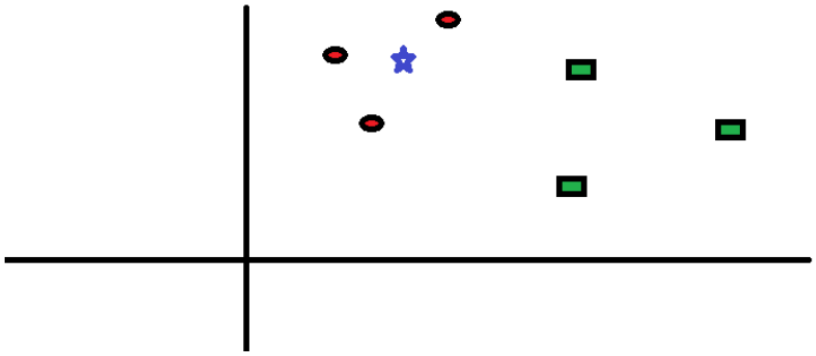
لطفا در تحویل پاسخ‌های خود موارد زیر را در نظر داشته باشید:

- 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 که موارد خواسته شده در سوالات هستند و سایر توضیحات خواسته شده دیگر در متن سوالات باشد (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_CHW2_Student_ID"، محتوی دو پوشه باشد. گزارش خود را در پوشه اول با عنوان "Report" و کدهای خود را در پوشه "Codes" قرار دهید.
- با این که همکاری و مشورت 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 ولی نتواند کد خود را توضیح دهد و یا تغییراتی روی آن اعمال کند، و یا اینکه کد و یا گزارش تحویلی به پاسخ دیگران شباهت غیرمنطقی داشته باشد، نمره تمرین صفر لحاظ شده و نمره‌ای منفی هم لحاظ خواهد شد.
- حتما در صورت وجود هر گونه سوال یا ابهام، مشکل مربوطه را با من در میان بگذارید.

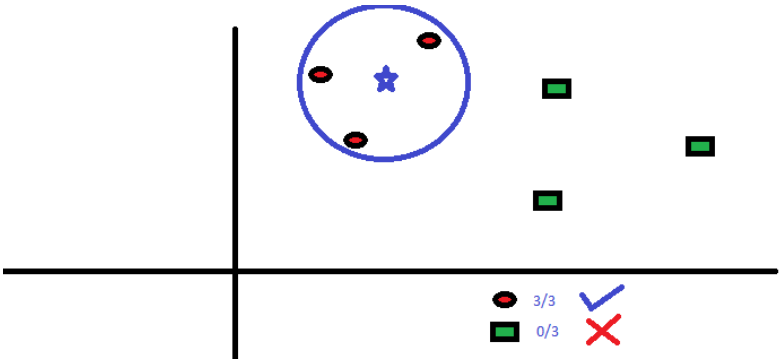
E-Mail: *asariaslani76@yahoo.com*
Telegram ID: *@Sa97amir*

K-Nearest Neighbors (KNN) For Classification

الگوریتم k نزدیک‌ترین همسایه، برای مسائل طبقه‌بندی و رگرسیون قابل استفاده است. اگرچه در اغلب مواقع از آن برای مسائل طبقه‌بندی استفاده میشود. این الگوریتم اغلب به دلیل سهولت تفسیر نتایج و زمان محاسبه پایین مورد استفاده قرار میگیرد. برای درک بهتر شیوه کار این الگوریتم، عملکرد آن با یک مثال ساده مورد بررسی قرار گرفته است درشکل زیر نحوه توزیع دایرههای قرمز(RC) و مربعهای سبز(GS) را مشاهده میکنید.



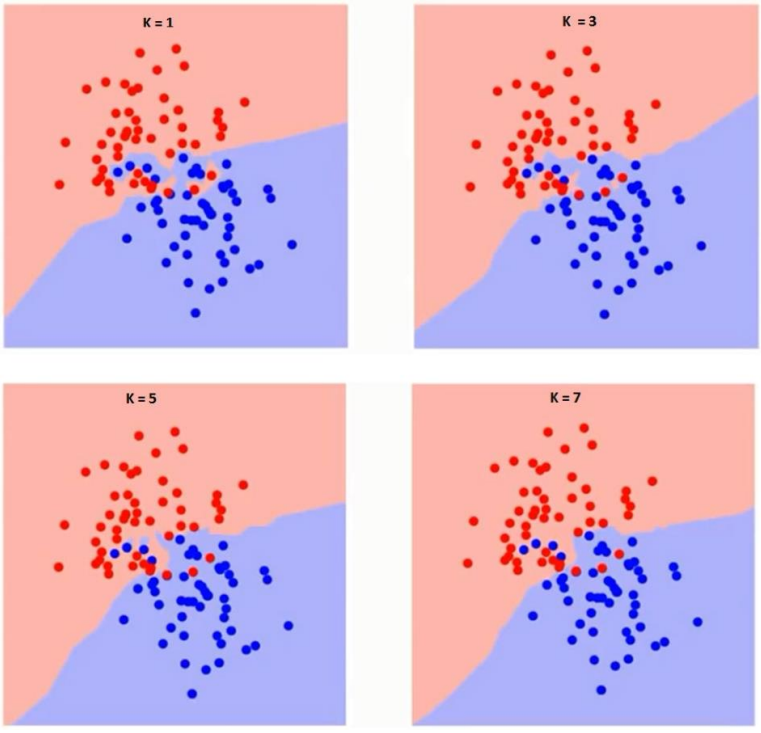
فرض کنید قصد پیدا کردن کلاسی که ستاره آبی (BS) متعلق به آن است را دارید (یعنی دایره های قرمز). دراین مثال درکل دو کلاس دایره های قرمز و مربعهای سبز وجود دارند و ستاره آبی بر اساس منطق کلاسیک میتواند تنها متعلق به یکی از این دو کلاس باشد«K» .در الگوریتم k–نزدیکترین همسایگی، تعداد نزدیک‌ترین همسایه‌هایی است که بر اساس آن‌ها رأی‌گیری درباره وضعیت تعلق یک نمونه داده به کلاسهای موجود انجام می‌شود. فرض کنید k=3 است. در شکل زیریک دایره آبی دور سه تا از نزدیکترین همسایه‌های ستاره آبی ترسیم شده‌است.



سه نقطه نزدیکتر به BS، همه دایره های قرمز هستند. با میزان اطمینان خوبی می توان گفت که ستاره آبی به کلاس دایره‌های قرمز تعلق دارد. در این مثال، انتخاب بسیار آسان است چون هر سه نزدیکترین همسایه ستاره آبی، دایره‌های قرمز هستند و در واقع هر سه رأی متعلق به دایره‌های قرمز است. انتخاب پارامتر k در الگوریتم k- نزدیکترین همسایگی، بسیار حائز اهمیت است. در ادامه عوامل تأثیرگذار بر انتخاب بهترین k مورد بررسی قرار می‌گیرند.

انتخاب پارامتر k:

در ابتدا بهتر است تأثیر انتخاب k در الگوریتم k-نزدیکترین همسایگی بر نتایج خروجی مورد بررسی قرار بگیرد. اگر به مثال پیشین توجه کنید، می بینید که هر شش نمونه (دایره‌های قرمز و مستطیل‌های سبز) ثابت هستند و تنها انتخاب k می تواند مرزهای یک کلاس را دستخوش تغییر کند. در ادامه مرزهای دو کلاس که با انتخاب kهای گوناگون تغییر پیدا می‌کنند را مشاهده می‌کنید

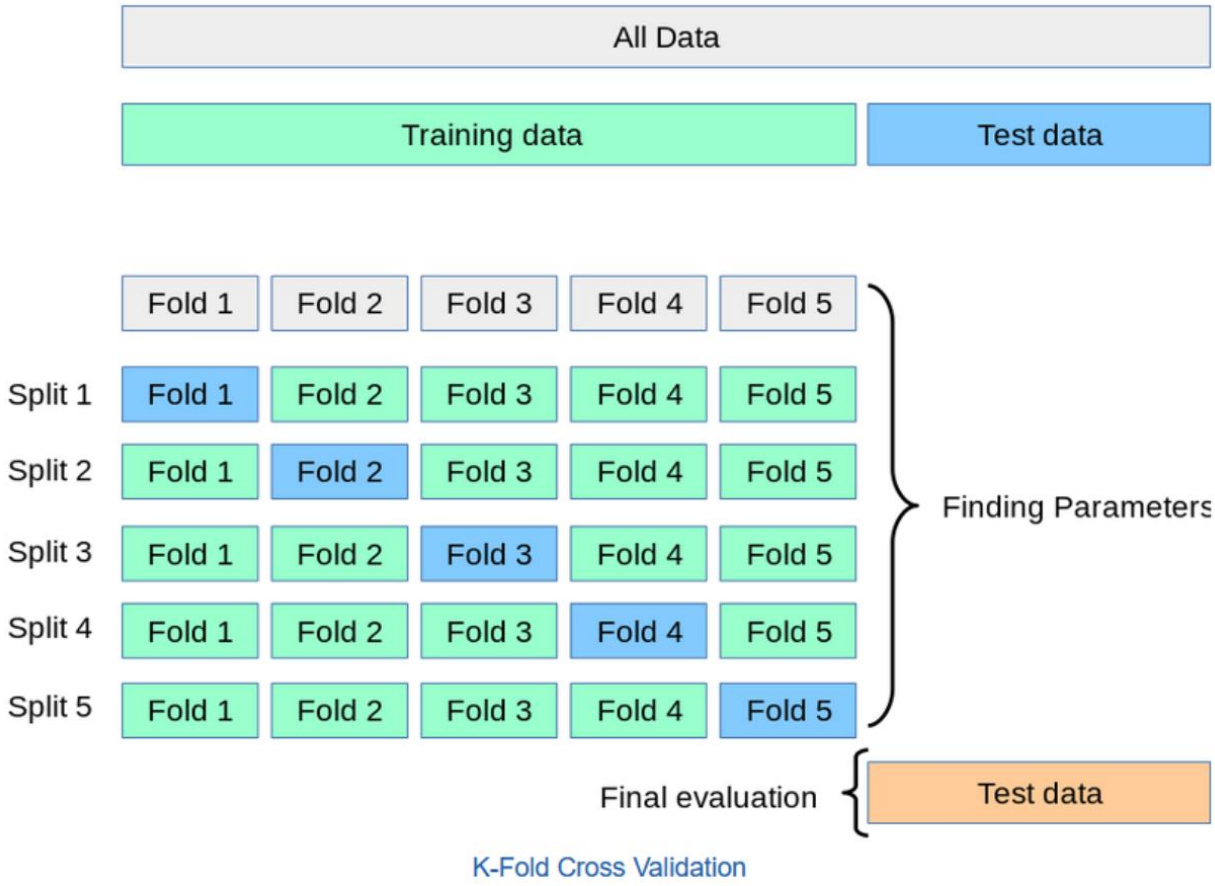


.

همان طور که از تصاویر مشهود است، با افزایش مقدار k مرزهای کلاس‌ها نرم‌تر می شود. در این تمرین قصد داریم با استفاده از یک روش، مقدار بهینه k را پیدا کنیم. واضح است که به ازای مقدار بهینه k بیشترین صحت را به دست می آوریم. احتمالا راه حل پیشنهادی شما، تست دسته‌بند ساخته شده به ازای kهای 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 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 (دقت داشته باشید که kدر اینجا با K در دسته‌بند بررسی شده بدون ارتباط است) بپردازیم.



K-Fold Cross Validation

همانطور که در تصویر بالا مشخص است، درk-fold cross validation، ابتدا مجموعه داده به دو بخش مجموعه داده تست و آموزش تقسیم میشود. سپس، مجموعه داده‌های آموزش به k زیرنمونه یا «لا» (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 می کند. این فرآیند منجر به محاسبه kخطا میگردد که میانگین‌گیری روی آنها صورت می گیرد. نهایتا، مقدار بهینه k 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 ،k با اعمال روی مجموعه داده تست مشخص می گردد.

شبه کد- k نزدیکترین همسایگی:

پیاده سازی k نزدیک‌ترین همسایگی با استفاده از شبه کد زیر امکان‌پذیر است:

- بارگذاری داده ها
- انتخاب اولیه مقدار k
- فاصله داده های تست از هر سطر مجموعه داده آموزش محاسبه شود. مرسوم ترین سنجه شباهت، فاصله اقلیدسی است. دیگر سنجه های قابل استفاده عبارت اند از فاصله چبیشف، کسینوس و. ...
- فاصله‌های محاسبه شده بر اساس مقدار فاصله به صورت صعودی مرتب شود
- سطر بالایی از آرایه مرتب شده انتخاب شود
- کلاسهای دارای بیشترین تکرار در این سطرها دریافت شود
- مقدار کلاس پیشبینیشده بازگردانده شود

در این تمرین قصد داریم دسته‌بند k-NN را بر روی مجموعه داده MNIST پیاده سازی کنیم.

بارگذاری مجموعه داده:

۱- مجموعه داده MNIST را مطابق زیر بارگذاری کنید. هر سطر از این مجموعه داده بیانگر اطلاعات مربوط به یک تصویر خاکستری عددی دست‌نویس است که به صورت یک بردار ذخیره شده‌است.

راهنمایی: اگر از پایتون استفاده میکنید، به منظور بارگذاری این مجموعه داده پیشنهاد میشود 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
mnist = fetch_openml('mnist_784')
mnist_data = mnist.data[:10000]
mnist_target = mnist.target[:10000]
```

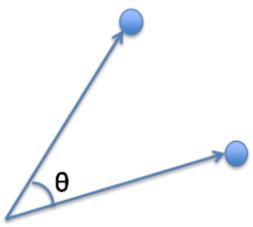
پیاده سازی دسته‌بند k-NN:

۲- یک تابع با نام " metric " بسازید که ورودی آن دو نقطه و خروجی آن معیار (فاصله یا شباهت) محاسبه شده باشد.

```
def metric(p, q):
    # YOUR CODE
    return score
```

راهنمایی: معیارهای شباهت، معیارهایی مانند معیارهای فاصله هستند که میزان دور و یا نزدیک بودن دو موجودیت را مشخص می کنند. بدیهی است که معیار شباهت با معیارهای فاصله رابطه عکس دارند و به عبارتی هر چه میزان شباهت بیشتر باشد می توان نتیجه گرفت فاصله‌ی دو شی کمتر است. شباهت کسینوسی یک معیار شباهت است که پایه آن محاسبه‌ی مقدار کسینوس زاویه‌ی بین دو بردار است. در صورت انطباق دو بردار(در این معیار نشانه شباهت کامل است) که زاویه ی بین دو بردار صفر می باشد مقدار آن برابر 1 خواهد شد و در کمترین میزان شباهت دو بردار یعنی اگر زاویه بین دو بردار 180درجه باشد نتیجه این معیار 1- خواهد شد.

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



3. یک تابع با نام" predict" بسازید که ورودی آن، k، نقطه x که می خواهیم برچسب آن را انتخاب کنیم و مجموعه داده آموزش باشد و خروجی آن، برچسب حدس زده شده باشد. با انجام این مرحله، شما یک کلاس بند k-NN ساخته اید.

```
def predict(x, k, train_set):
   # YOUR CODE
   return label
```

راهنمایی: در این تابع، ابتدا با استفاده از تابع metric، میزان شباهت (یا فاصله) نقطه x با تمام نقاط موجود در مجموعه آموزش را محاسبه کنید. حال باید همسایگان را پیدا کنید. به این منظور میزان شباهت‌های (فاصله‌های) محاسبه شده را به ترتیب نزولی (صعودی) مرتب کنید. سپس با بررسی برچسب k نقطه اول که بیشترین شباهت

(کمترین فاصله) را دارند، برچسب نقطه x را حدس بزنید. برای تابع معیار می توانید از فاصله اقلیدسی یا شباهت کسینوسی استفاده کنید.

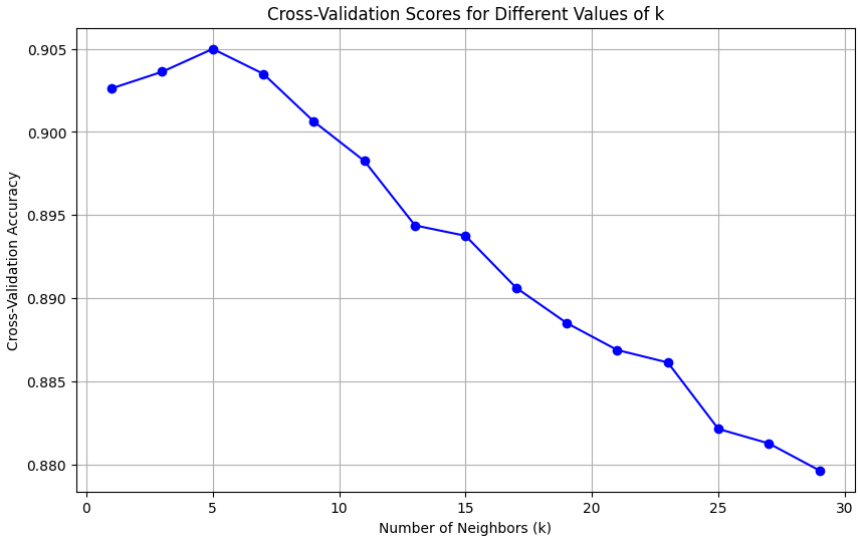
تست مدل با استفاده از مقدار k از پیش تعیین شده:

۴- برای هر کلاس، داده آموزش و تست را به طوری جداسازی کنید که 20٪ از داده ها در مجموعه تست و بقیه در مجموعه آموزش قرار گیرند.

۵- Accuracy را برای دسته‌بند ساخته شده با k=10 محاسبه کنید

بهینه سازی مقدار :

۶- با استفاده از 10-fold cross validation، مشابه شکل زیر، یک نمودار از صحت به ازای kهای مختلف ترسیم کنید



راهنمایی: 20٪ از داده‌ها را به عنوان داده‌ی تست کنار بگذارید و در این مرحله سراغشان نروید. 80٪ باقی داده‌ها را به 10 بخش مساوی تقسیم کنید. هر بار یکی از این بخش‌ها را به عنوان داده اعتبارسنجی و بقیه بخش‌ها را به عنوان داده آموزش در نظر بگیرید. فرض کنید میخواهیم بین مجموعه kهای مقابل، k بهینه را پیدا کنیم.

نتیجه اعمال kNN را به ازای k_i روی مجموعه داده اعتبارسنجی به دست آورید. این کار را ۹بار دیگر تکرار کنید، به طوری که هر بار یکی از این بخش‌ها، داده اعبارسنجی و بقیه بخشها داده آموزش باشند. در نهایت، میانگین صحت‌های به دست آمده از هر مرحله را به عنوان صحت به ازای k_i در نظر داشته باشید. تمام مراحل ذکر شده را برای kهای مختلف انجام دهید و منحنی میانگین صحت بر حسب k را ترسیم نمایید.

۷- مقدار بهینه k چند است؟ Accuracy را به ازای مقدار بهینه k محاسبه کنید.

راهنمایی: دقت داشته باشید که صحت یک کلاسبند، نتیجه اعمال آن روی مجموعه داده تست میباشد (نه مجموعه داده اعتبار سنجی. از این رو دقت طبقه‌بندی را برای 20٪ نمونه موجود در مجموعه داده تست، با در نظر گرفتن 80٪ باقی به عنوان مجموعه داده آموزش و k بهینه به دست آمده، به دست آورید).

8- مراحل قبل را (kfold و kNN) 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 (kfold و kNN) استفاده کنید.

9- جهت کاهش حافظه مورد نیاز و افزایش سرعت الگوریتم، به ازای k 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 (Leave one out method) و نمونه‌هایی که به درستی توسط سایر داده‌های آموزش دسته‌بندی نمی‌شوند را از مجموعه داده آموزش حذف کنید.)

10- دقت دسته‌بندی مجموعه داده تست را، در حالت حذف نمونه‌های دارای ابهام از مجموعه داده آموزش، محاسبه کنید.

11- 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، 50 بار هر کدام از حالت‌های ذکر شده را انجام داده و زمان محاسبه را اندازه‌گیری کرده و میانگین آنها را با یکدیگر مقایسه کنید.

12- (امتیازی +5٪) با محاسبه p-value نشان دهید که آیا زمان لازم جهت دسته‌بندی مجموعه تست در دو حالت ذکر شده تفاوتی معنادار دارند یا خیر.

"موفق باشید"