گزارش تکلیف ۴ درس یادگیری ماشین

کسرا سینایی
شماره دانشجویی ۸۱۰۶۹۶۲۵۴

۱۰ دی ۱۴۰۰

۱ سوال یک

الف

در روش KNN هرچه مقدار K را کوچکتر انتخاب کنیم پیچیدگی مدل و مرز تصمیم افزایش می‌یابد و به همین نسبت جنرالیزیشن مدل کمتر و حساسیت به نویز نیز بیشتر می‌شود. در واقع هر چه K کوچکتر باشد واریانس بیشتر خواهد بود.
با افزایش K به حد بهینه‌ای برای مدل می‌رسیم و افزایش بیشتر آن منجر به خطای بایاس می‌شود.
در تخمین پنجره پارزن هایپرپارامتر مدل طول پنجره در تابع کرنل می‌باشد. این پارامتر (h_n) مانند یک smoothing factor عمل می‌کند. هر چه طول پنجره بزرگتر باشد شکل تابع به دست آمده برای توزیع متغییرها نرم‌تر می‌شود. برای h های بسیار کوچک تخمین نان‌پارامتری پارزن به تابع دلتا دیراک شبیه خواهد شد. به زبان دیگر انتخاب طول بزرگ برای پنجره منجر به خطای بایاس و انتخاب اندازه کوچک برای طول پنجره منجر به خطای واریانس می‌شود.

ب

در روش‌های پارامتریک فرض می‌کنیم داده‌ها از یک توزیع شناخته شده و فرمول‌بندی شده مانند گوسی، پواسون و ... آمده‌اند و برای توصیف آن‌ها به تخمین پارامترهای مدل فرض شده می‌پردازیم.
در روش‌های نان‌پارامتری هیچ فرض اضافه‌ای برای مدل حاکم بر داده‌های یادگیری قرار نمی‌دهیم و با قوانینی مانند نزدیک‌ترین همسایه، استفاده از تابع کرنل و ... توزیعی دلخواه به داده‌های یادگیری فیت می‌کنیم.

ج

چون این روش‌ها نان‌پارامتریک هستند لازم است برای پیش‌بینی نقاط دیده نشده تمام مجموعه یادگیری را همراه با مدل نگه داریم. (در واقع مدل مستقل و به کمک یک تابع و چند پارامتر تعریف نشده که بتوان با استفاده از آن پیش بینی کرد، بلکه مدل با استفاده از داده‌های مجموعه یادگیری در هر نقطه تعریف می‌شود)
با افزایش ابعاد فیچرها برای دست‌یابی به تخمینی دقیق تعداد نقاط مورد نیاز در دیتاست به صورت نمایی افزایش می‌یابد و در مواردی که نتوانیم حجم داده‌ها را افزایش دهیم ممکن است دچار نفرین ابعادی شویم.

د

در روش پنجره پارزن مفهوم حجم وابسته به کرنل استفاده شده است. برخی کرنل‌ها به تمامی نقاط موجود در دیتاست وزن می‌دهند (مانند کرنل گوسی) و برخی کرنل‌ها مانند hyper cube kernel تنها داده‌های موجود در ابر مکعب حول نقطه مورد نظر را برای طبقه بندی یا تخمین احتمال در نظر می‌گیرند.
در روش KNN حجم با یک ابر کره که شعاع آن فاصله دورترین نقطه در بین K همسایه نقطه‌ای که می‌خواهیم احتمال آن را به دست بیاوریم، تعریف می‌شود.

۲ سوال دو

در این سوال می‌خواهیم نشان دهیم که سلول‌های voronoi تشکیل شده با الگوریتم نزدیک‌ترین همسایه محدب هستند. به ازای هر دو نقطه درون یک سلول، خط وصل کننده آن‌ها نیز داخل سلول قرار خواهد گرفت ،فرض کنید x^* در یک سلول با لیبلی مشخص واقع شده باشد و y نیز نقطه‌ای با یک لیبل متفاوت از x^* باشد. در این حالت ابر صفحه‌ای وجود دارد که فضا را طوری تقسیم می‌کند که یک طرف آن نقاط به x^* نزدیک تر هستند و طرف دیگر آن نقاط به y نزدیک ترند. اگر نقاط $x(0)$ و $x(1)$ در سلول متعلق به x^* قرار داشته باشند خطی که این دو نقطه را به هم وصل می‌کند نیز به صورت: $x(\lambda) = (1-\lambda)x(0) + \lambda x(1); 0 \leq \lambda \leq 1$ خواهد بود. این خط نیز به طور کامل درون سلول متعلق به نقطه x^* قرار خواهد گرفت. چون فضای نصف شده توسط ابر صفحه محدب است، تمامی نقاط موجود بر روی خط تعریف شده ($x(\lambda)$) نیز به به x^* نزدیک تر هستند تا y.
این موضوع به ازای تمام نقاط موجود در Voronoi Cell متعلق به نقطه x^* صادق است. همچنین برای سایر سمپل‌ها نیز صادق است (شرطی برای y نداریم) بنابراین Voronoi Cell تشکیل شده با قانون نزدیک ترین همسایه همواره محدب می‌باشند.

۳ سوال سه

الف

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \longrightarrow P_n(x) = \frac{1}{nh_n} \sum_{i=1}^{n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

$$\bar{P}_n(x) = E\{P_n(x)\} = \frac{1}{nh_n} \sum_{i=1}^{n} E\left\{\varphi\left(\frac{x-x_i}{h_n}\right)\right\} = \frac{1}{nh_n} \int_{-\infty}^{\infty} \varphi\left(\frac{x-v}{h_n}\right) P(v)\, dv$$

$$= \frac{1}{h_n} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-v}{h_n}\right)^2\right) \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2}\left(\frac{v-\mu}{\sigma}\right)^2\right) dv$$

$$= \frac{1}{2\pi h_n \sigma} \exp\left(-\frac{1}{2}\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2}\right)\right) \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} v^2 \left(\frac{1}{h_n^2} + \frac{1}{\sigma^2}\right) - 2v\left(\frac{x}{h_n^2} + \frac{\mu}{\sigma^2}\right)\right) dv$$

$$= \frac{1}{2\pi h_n \sigma} \exp\left(-\frac{1}{2}\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2}\right) + \frac{1}{2}\frac{\alpha^2}{\theta^2}\right) \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\left(\frac{v-\alpha}{\theta}\right)\right) dv$$

تغییر متغیر

$$\theta^2 = \frac{1}{1/h_n^2 + 1/\sigma^2} = \frac{(h_n \sigma)^2}{h_n^2 + \sigma^2} \quad , \quad \alpha = \theta^2 \left(\frac{x}{h_n^2} + \frac{\mu}{\sigma^2}\right)$$

$$\Longrightarrow \bar{P}_n(x) = \frac{\sqrt{2\pi}\,\theta}{2\pi h_n \sigma} \exp\left(-\frac{1}{2}\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2}\right) + \frac{1}{2}\frac{\alpha^2}{\theta^2}\right)$$

$$= \frac{1}{\sqrt{2\pi}\, h_n \sigma} \frac{h_n \sigma}{\sqrt{h_n^2 + \sigma^2}} \exp\left(-\frac{1}{2} \underbrace{\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2} - \frac{\alpha^2}{\theta^2}\right)}_{A}\right)$$

$$A = \frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2} - \frac{\theta^4}{\theta^2}\left(\frac{x}{h_n^2} + \frac{\mu}{\sigma^2}\right)^2 = \frac{(x-\mu)^2}{h_n^2 + \sigma^2} \Longrightarrow \bar{P}_n(x) = \frac{1}{\sqrt{2\pi}\sqrt{h_n^2+\sigma^2}} \exp\left(\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right) \checkmark$$

ب

$$P(x) - \bar{P}(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) - \frac{1}{\sqrt{2\pi}\sqrt{h_n^2+\sigma^2}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right)$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right)\left[1 - \frac{\sigma}{\sqrt{h_n^2+\sigma^2}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2+\sigma^2} + \frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)\right]$$

$$= P(x)\left[1 - \frac{1}{\sqrt{1+(h_n/\sigma)^2}} \exp\left(-\frac{(x-\mu)^2}{2}\left(\frac{1}{h_n^2+\sigma^2} - \frac{1}{\sigma^2}\right)\right)\right]$$

$$= P(x)\left[1 - \frac{1}{\sqrt{1+(\frac{h_n}{\sigma})^2}} \exp\left(\frac{1}{2}\frac{h_n^2(x-\mu)^2}{h_n^2+\sigma^2}\right)\right]$$

if $h_n \to 0$: $\frac{1}{\sqrt{1+(\frac{h_n}{\sigma})^2}} \simeq 1 - \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2$, $\exp\left(\frac{h_n^2}{2\sigma^2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right) = 1 + \frac{h_n^2}{2\sigma^2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}$

$$\Longrightarrow P(x) - \bar{P}(x) = P(x)\left[1 - \left(1 - \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\right)\left(1 + \frac{h_n^2}{2\sigma^2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right)\right.$$

$$= P(x)\left[1 - 1 + \frac{1}{2}\frac{h_n^2}{\sigma^2} - \frac{h_n^2}{2\sigma^2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right] = \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\left[1 - \frac{(x-\mu)^2}{\sigma^2}\right] P(x) \checkmark$$

ج

$$Var\{P_n(x)\} = Var\left\{\frac{1}{nh_n}\sum^{n}\varphi\left(\frac{x-x_i}{h_n}\right)\right\} = \frac{1}{n^2h_n^2}\sum_{i=1}^{n} Var\left\{\varphi\left(\frac{x-x_i}{h_n}\right)\right\}$$

$$= \frac{1}{nh_n^2} Var\left\{\varphi\left(\frac{x-v}{h_n}\right)\right\} = \frac{1}{nh_n^2}\left(\underbrace{E\left\{\varphi^2\left(\frac{x-v}{h_n}\right)\right\}}_{A} - \underbrace{E\left\{\varphi\left(\frac{x-v}{h_n}\right)\right\}^2}_{B}\right)$$

$$A = \int \varphi^2\left(\frac{x-v}{h_n}\right) P(v)\,dv = \int_{-\infty}^{+\infty} \frac{1}{2\pi} exp\left[-\frac{(x-v)^2}{h_n^2}\right] exp\left[-\frac{1}{2}\left(\frac{v-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\,\sigma}\,dv$$

$$= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{x-v}{h_n/\sqrt{2}}\right)^2\right] exp\left[-\frac{1}{2}\left(\frac{v-\mu}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma}\,dv$$

$$= \frac{h_n/\sqrt{2}}{2\pi\sqrt{\frac{h_n^2}{2}+\sigma^2}}\, exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2/2+\sigma^2}\right) \qquad if\ h_n\to 0 : \sqrt{\frac{h_n^2}{2}+\sigma^2} \simeq \sigma$$

$$\longrightarrow \frac{1}{nh_n^2} E\left\{\varphi^2\left(\frac{x-v}{h_n^2}\right)\right\} = \frac{1}{nh_n}\ \frac{1}{2\pi}\, exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) = \frac{1}{2nh_n\sqrt{\pi}}\, P(x)$$

$$B = h_n^2\, \frac{1}{\sqrt{2\pi}\sqrt{h_n^2+\sigma^2}}\, exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right) \simeq \frac{1}{\sqrt{2\pi}\,\sigma} exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \simeq 0$$

$$\Longrightarrow Var\{P_n(x)\} = \frac{P(x)}{2nh_n\sqrt{\pi}} \quad \checkmark$$

۴ سوال چهار

$\alpha_k > 0 \longrightarrow D(a,b) \geqslant 0$ non-negative ✓

if $a = b \rightarrow$ all components of Σ become $0 \longrightarrow D(a,b) = 0$ reflexivity ✓

$(a_k - b_k)^2 = (b_k - a_k)^2 \longrightarrow D(a,b) = D(b,a)$ Symmetry ✓

triangle inequality : $D(a,b) + D(b,c) \geqslant D(a,c)$

if $A = \mathrm{diag}(\alpha_1, \dots, \alpha_d) \longrightarrow D(a,b) = \|A(a-b)\|$

if $x = a - c$, $y = c - b$

Proove : $\|Ax\| + \|Ay\| \geqslant \|A(x+y)\|$

if $Ax = x'$, $Ay = y' \rightarrow \|x'\| + \|y'\| \geqslant \|x' + y'\|$

$\xrightarrow{\text{توان } 2} (\|x'\| + \|y'\|)^2 = \|x'\|^2 + \|y'\|^2 + 2\|x'\|\,\|y'\| \geqslant \|x'\|^2 + \|y'\|^2 + 2\sum^{d} x'_i y'_i = \|x' + y'\|^2$

it is correct if $\rightarrow \|x'\|\,\|y'\| \geqslant \sum_{i=1}^{d} x'_i y'_i$ ✓ based on Cauchy-Schwartz is always true

۵ سوال پنج

$P(\omega_1) = P(\omega_2) = \frac{1}{2}$ $\quad P(\omega_1|x) = \begin{cases} 1 & \|x\| \leqslant 1 \\ 0 & ow \end{cases}$ $\quad P(\omega_2|x) = \begin{cases} 1 & \|x-a\| \leqslant 1 \\ 0 & ow \end{cases}$

الف

$P_n(e) = P\{\omega_1 \text{ is true } \&\& \ \omega_2 \text{ is most frequent}\}$

$+P\{\omega_2 \text{ is true } \&\& \ \omega_1 \text{ is most frequent}\}$

$P_n(e) = 2P\{\omega_1 \text{ is true } \&\& \ \omega_2 \text{ is most frequent}\}$

$\longrightarrow P_n(e) = 2P\{\#\omega_1 < \frac{k-1}{2}\} = 2\left(\frac{1}{2}\sum_{j=0}^{\frac{k-1}{2}} P\{\#\omega_1 = j \ \&\& \ \#\omega_2 = n-j\}\right)$

$= \sum_{j=0}^{\frac{k-1}{2}} \binom{n}{j} \frac{1}{2^j}\frac{1}{2^{n-j}} = \left(\frac{1}{2^n}\right)\sum_{j=0}^{\frac{k-1}{2}} \binom{n}{j}$

ب

$$P_n(e;1) = \frac{1}{2^n} < \frac{1}{2^n}\underbrace{\sum^{\frac{k-1}{2}} \binom{n}{j}}_{\text{if } k \geqslant 2 \text{ is greater than } 1} \longrightarrow \text{for } k \geqslant 2 \ ; P_n(e;1) < P_n(e;k)$$

ج

$B(n, P) \rightarrow$ Binomial Distribution

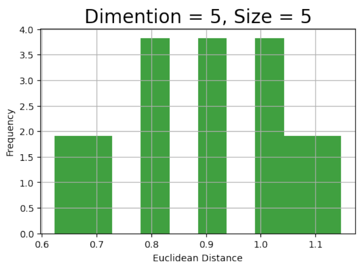
$P_n(e) = \frac{1}{2^n}\sum_{j=0}^{\frac{k-1}{2}} \binom{n}{j} = P\{B(n, \frac{1}{2}) \leqslant \frac{k-1}{2}\} = P\{Y_1 + \dots + Y_n \leqslant \frac{k-1}{2}\}$

$P\{Y_i = 1\} = P\{Y_i = 0\} = \frac{1}{2}$; based on convergence criteria of KNN: $k < \frac{a}{\sqrt{n}}$

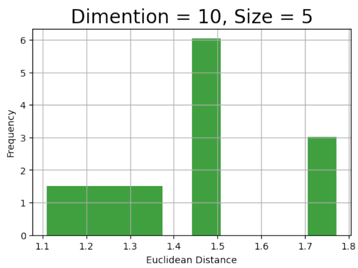
$$\Rightarrow P_n(e) \leqslant P\{Y_1 + \dots + Y_n \leqslant \frac{\frac{a}{\sqrt{n}} - 1}{2}\} \underset{n \to \infty}{=} P\{Y_1 + \dots + Y_n \leqslant 0\} = 0$$

۶ سوال شش (شبیه سازی)

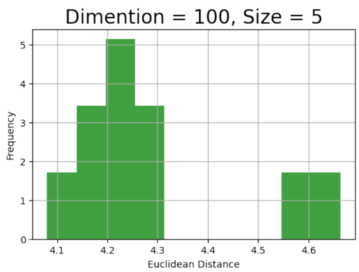
الف



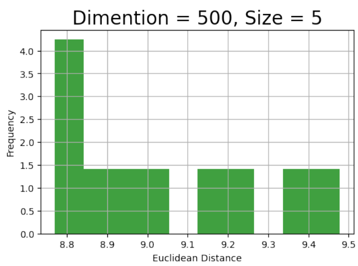
(آ) ۵ بعدی



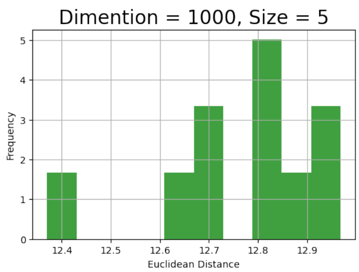
(ب) ۱۰ بعدی



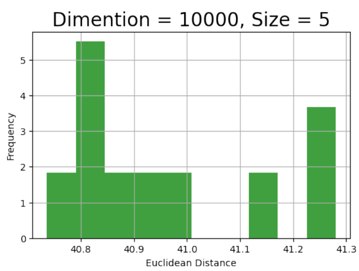
(ج) ۱۰۰ بعدی



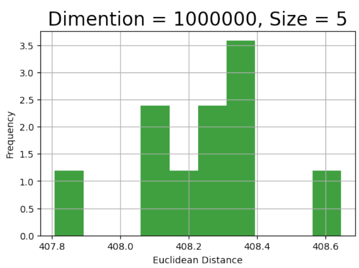
(د) ۵۰۰ بعدی



(ه) ۱۰۰۰ بعدی



(و) ۱۰۰۰۰ بعدی

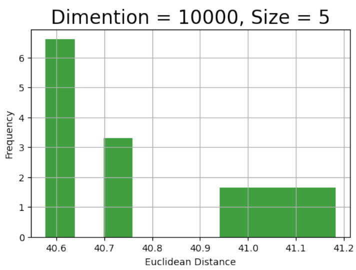


(ز) ۱۰۰۰۰۰ بعدی

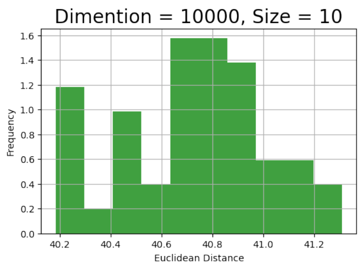
شکل ۱: خواسته‌های قسمت الف

در این سوال با استفاده از تابع رندوم کتابخانه numpy و تابع hist از کتابخانه matplotlib هیستوگرام فاصله نقاط را به دست می‌آوریم. برا این کار تابع do task پیاده سازی شده است که تعداد و بعد نقاط مورد نظر را در ورودی گرفته و نمودارها را رسم می‌کند.

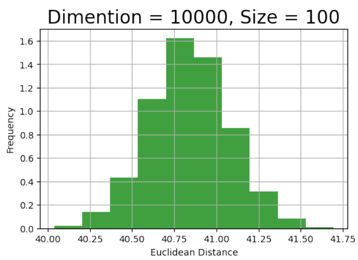
ب



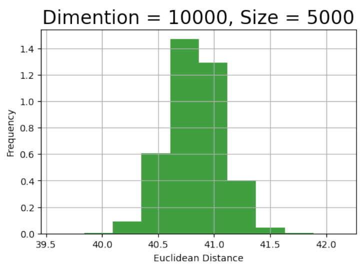
(آ) ۵ نقطه



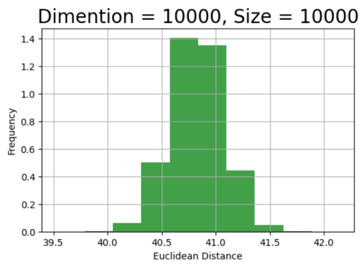
(ب) ۱۰ نقطه



(ج) ۱۰۰ نقطه



(د) ۵۰۰۰ نقطه



(ه) ۱۰۰۰۰ نقطه

شکل ۲: خواسته‌های قسمت ب

ج

اولین مسیله‌ای که به چشم می‌آید افزایش مدت زمان محاسبه فاصله نقاط از یکدیگر می‌باشد. همانطور که می‌دانیم پیچیدگی محاسباتی الگوریتم KNN برابر با $O(mn)$ می‌باشد. (m تعداد فیچر و n تعداد داده‌های یادگیری می‌باشد.) یکی از مشکلات اساسی الگوریتم KNN این است که در ابعاد بالا برای دقیق شدن تخمین باید تعداد داده‌ها را نیز افزایش

دهیم که این موجب افزایش زمان طبقه بندی می‌شود. این مشکل همان نفرین ابعاد (curse of dimentionality) است.

۷ سوال هفت (شبیه سازی)

در این سوال با استفاده از الگوریتم KNN تصاویر اعداد دست نویس فارسی را طبقه بندی می‌کنیم.

الف

برای لود کردن دیتا ست از روی فایل‌های cdb از اسکریپت پایتون معرفی شده در لینک قرار داده شده در صورت سوال استفاده می‌کنیم. یکی از توابع موجود در این اسکریپت read hoda datset می‌باشد. آرگومان‌های ورودی که آدرس فایل دیتاست و سایز تصاویر می‌باشد را مطابق توضیحات پیج گیت‌هاب به تابع می‌دهیم. آرگومان reshape را نیز True قرار می‌دهیم و آرگومان one hot را False، در این صورت فیچرها که پیکسل عکس‌های 32×32 هستند فلت شده و لیبل‌ها نیز یک عدد بین ۰ و ۹ خواهند بود.
در شکل ۳ سه نمونه از تصاویر متعلق به هر کلاس نمایش داده شده‌اند. (رنگ ها غیر واقی می‌باشند، تمام تصاویر یک کاناله و gray scale هستند.)

ب

در این قسمت الگوریتم KNN را به صورت دستی پیاده سازی می‌کنیم. به این منظور یک کلاس به نام KNN با سه متد زیر پیاده سازی شده‌است:

- constructor: این تابع تنها هایپر پارامتر مدل که همان k یا تعداد همسایه‌ها می‌باشد را به همراه دیتاست و فیچرها می‌گیرد و در متغیرهای کلاس ذخیره می‌کند. تعداد سمپل‌ها و ابعاد فیچر نیز از دیگر متغیرهای این کلاس هستند که از روی ورودی‌های این تابع به دست می‌آیند. لازم به ذکر است که ورودی‌های مربوط به سمپل‌ها و لیبل‌ها باید آرایه numpy باشند.

- dist: این تابع بع صورت برداری فاصله اقلیدسی دو بردار که آرگومان‌ها ورودی آن هستند را حساب می‌کند.

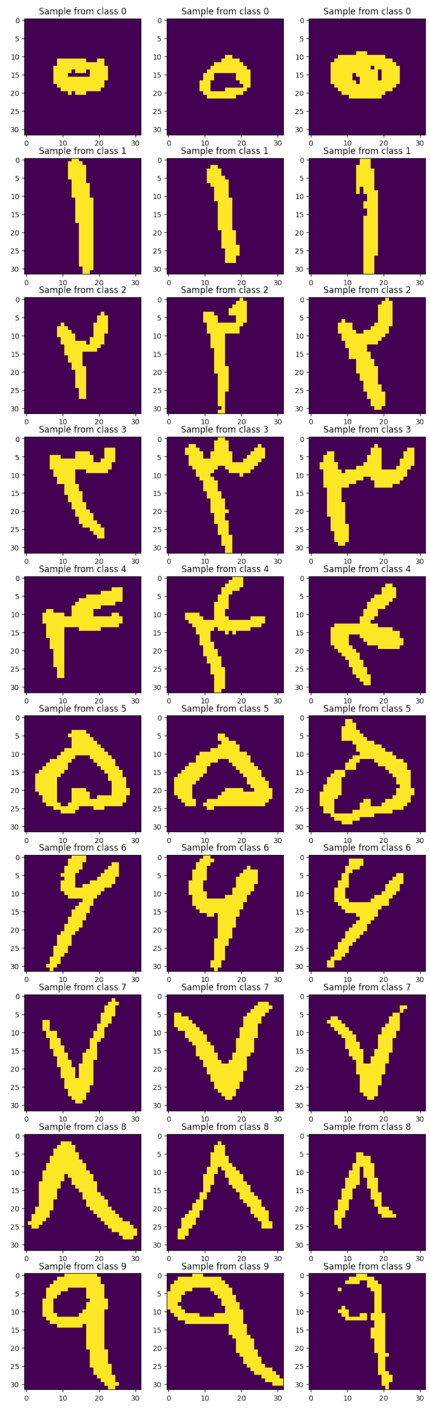
- predict: در این تابع برای هر سطر از متغییر ورودی (x) تخمین نان‌پارامتریک مدل را به دست می‌آوریم. ابتدا با استفاده از متد dist و تابع argsort کتابخانه numpy همسایه‌های نقاط را به دست می‌آوریم. در نهایت با استفاده از تابع unique و argmax پر تکرارترین لیبل در بین kهمسایه نزدیک نقطه مورد بررسی را به دست آورده و آن نقطه را در همان طبقه قرار می‌دهیم.

سرعت محاسبه تابع predict در مقایسه با توابع آماده کتابخانه scikit learn به مقدار قابل توجهی کمتر است. لیبل کردن ۲۰۰۰۰ دیتا از مجموعه تست حدودا ۳۰ دقیقه زمان برد. نتایج طبقه بندی در جدول ۱ و شکل ۴ آورده شده است.

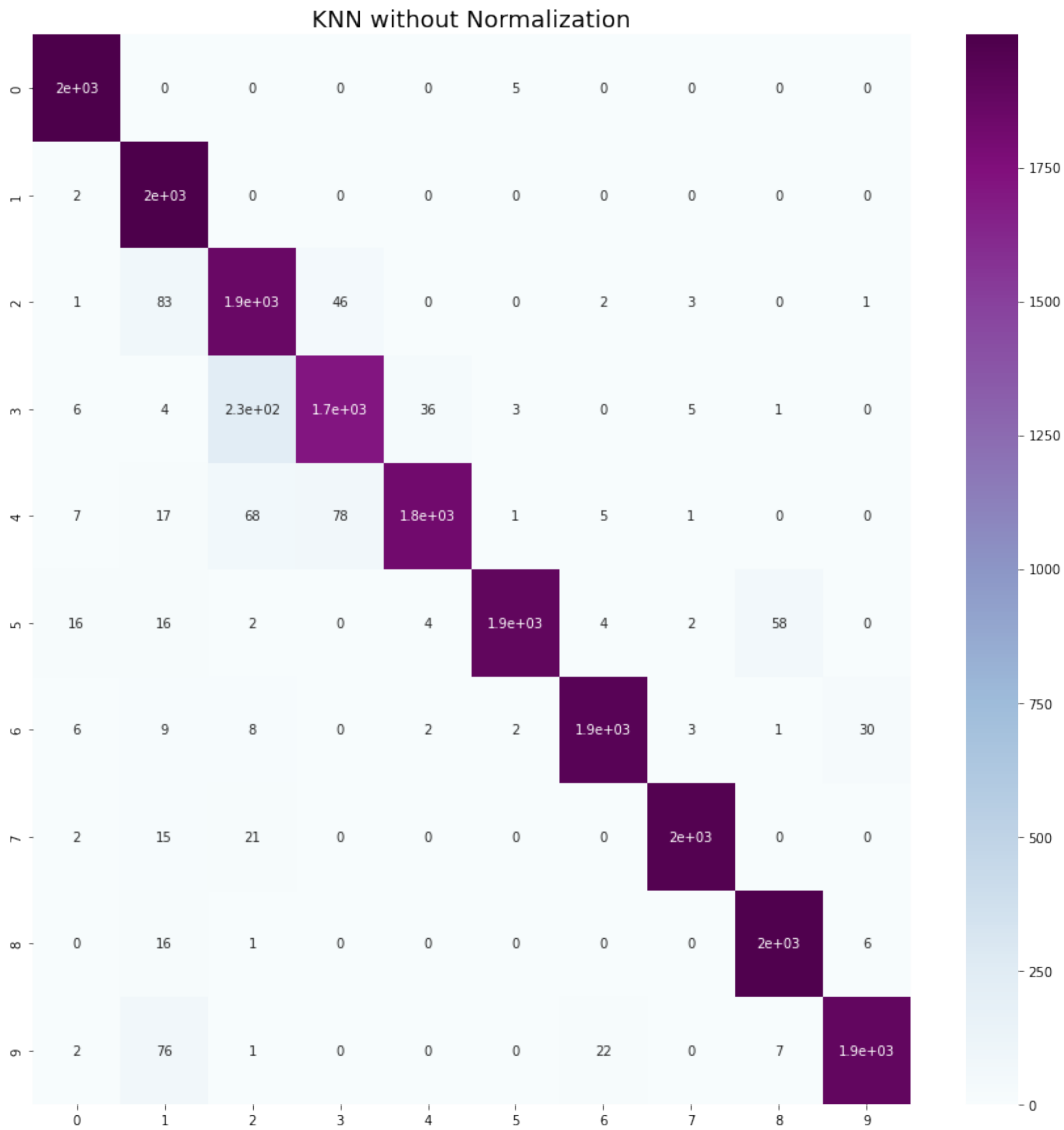
label	0	1	2	3	4	5	6	7	8	9
F1 Score	98.83	94.37	88.82	89.28	94.33	97.10	97.63	98.69	97.77	96.30

جدول ۱: نتایج طبقه بندی KNN با داده‌های نرمالایز نشده

همانطور که از ماتریس آشفتگی و f1 score نیز مشخص است، دقت تمایز بین برخی کلاس‌ها پایین تر است. برای نمونه اعداد ۲ و ۳ فارسی شباهت زیادی به یک دیگر دارند و در مجموعه تست تعدادی از نمونه‌های این دو کلاس اشتباه طبقه بندی شده‌اند. کلاس ۹ و ۱ نیز تعداد اشتباه به نسبت بالایی دارند.



شکل ۳: چند نمونه از تصاویر موجود در دیتاست



شکل ۴: Confusion Matrix طبقه بند KNN با داده‌های نرمالایز نشده

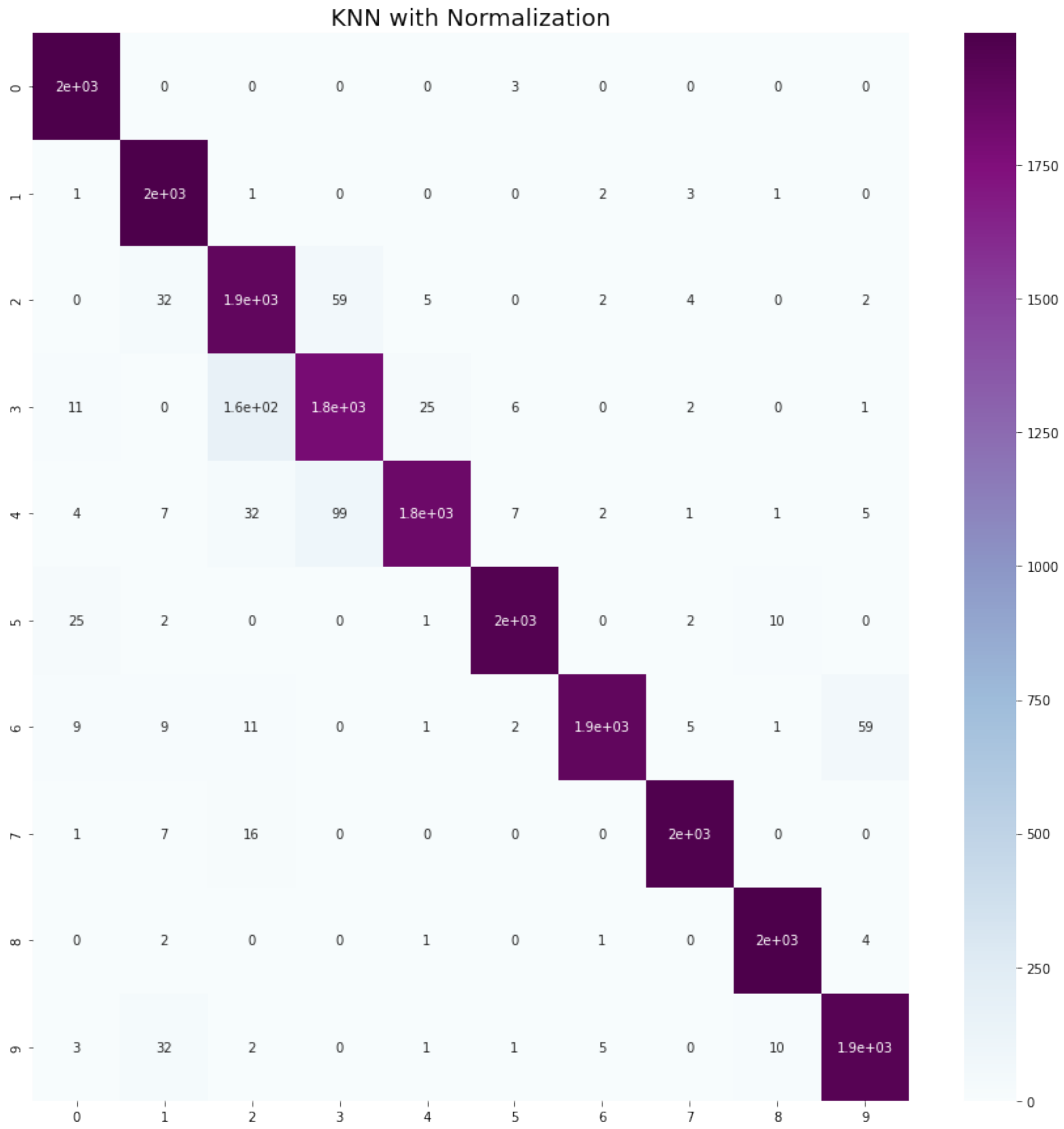
ج

در این قسمت بردار فیچر را با استفاده از رابطه زیر نرمال می‌کنیم:

$$\mathbf{D} = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \cdots, \overrightarrow{x_n}\}$$

$$\overrightarrow{x_i} := \frac{\overrightarrow{x_i}}{\sum_{j=1}^{d} x_i^j}$$

با استفاده از رابطه فوق دو بردار X train و X test نرمال شده و در آرایه‌ی جدیدی ذخیره می‌کنیم و مجدداً عکس‌ها را طبقه‌بندی می‌کنیم. نتایج طبقه بندی با داده‌های نرمال شده بر روی مجموعه تست در شکل ۵ نشان داده شده است.



شکل ۵: Confusion Matrix طبقه بند KNN با داده‌های نرمالایز شده

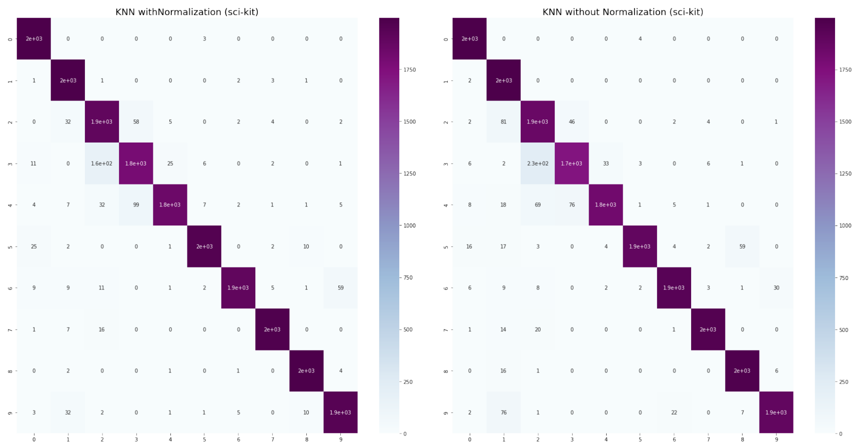
از ماتریس آشفتگی و جدول ۲ می‌توان نتیه گرفت که در مقایسه با نتایج قسمت قبل بهبود اندکی در عملکرد طبقه بند به چشم می‌خورد.
در تمامی قسمت‌های این تمرین از تعداد همسایه $K = 5$ در طبقه بند استفاده شده است.

label	0	1	2	3	4	5	6	7	8	9
F1 Score (Raw)	98.83	94.37	88.82	89.28	94.33	97.10	97.63	98.69	97.77	96.30
F1 Score (Normal)	98.59	97.57	91.97	90.67	95.04	98.51	97.21	98.97	99.22	96.88

جدول ۲: مقایسه عملکرد طبقه‌بند KNN با داده‌های پردازش نشده و داده‌های نرمال شده

د

در این قسمت با استفاده از کلاس آماده KNeighborsClassifier از کتابخانه Scikit Learn طبقه بندی را برای داده‌های پردازش نشده و پردازش شده انجام می‌دهیم. عملکر طبقه بند با متریک F1 Score و همچنین ماتریس‌های آشفتگی در شکل ۶ و جدول ۳ نشان داده‌شده‌اند.
برای ساخته نمونه از این کلاس تنها کافی است مقدار n neighbours را در کانستراکتور تنظیم کنیم و سپس با استفاده از تابع fit داده‌های یادگیری و لیبل آن‌ها را به آبجکت تشکیل شده بدهیم. در نهایت با استفاده از دستور predict داده‌های تست را طبقه بندی کنیم.



(ب) فیچرهای نرمال شده　　(آ) فیچرهای خام

شکل ۶: طبقه بندی با کلاس کتابخانه scikit learn

label	0	1	2	3	4	5	6	7	8	9
F1 Score (Raw)	98.83	94.44	88.84	89.50	94.37	97.05	97.60	98.69	97.75	96.30
F1 Score (Normal)	98.59	97.57	91.99	90.70	95.04	98.51	97.21	98.97	99.22	96.88

جدول ۳: نتایج طبق بندی با استفاده از کتابخانه scikit learn

۸ سوال هشت (شبیه سازی)

در این تمرین به پیاده سازی روش نان‌پارامتری پنجره پارزن می‌پردازیم. داده‌های موجود در فایل اکسل را با دستورات کتابخانه pandas باز کرده و ستون duration را در یک آرایه numpy ذخیره می‌کنیم.
برای پیاده سازی طبقه بند یک کلاس به نام parzen پیاده سازی شده است. این کلاس چهار متد برای اجرای الگوریتم طبقه بندی و پیش بینی احتمال نقاط جدید دارد.

- constructor: برای ایجاد یک نمونه از این کلاس لازم است داده‌های یادگیری، نوع کرنل و اندازه پنجره برای فذاخوانی این تابع ارسال کرد. تعداد داده‌های یادگیری باید حتماً آرایه numpy باشند.

- gaussian kernel: این تابع با استفاده از رابطه $\phi(\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi})^d h_n^d} \exp\left[-\frac{1}{2}\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h_n}\right)^2\right]$ کرنل گوسی را برای بردار ورودی حساب می‌کند.

- hypercube kernel: برای تست عملکرد طبقه بند قبل از یادگیری وتست بر روی داده‌های اصلی با استفاده از کرنل مکعبی تابع prob تست شده است. برای استفاده از این کرنل در کانستراکتور نمونه ساخته شده از این تابع باید آرگومان آخر را "cube" بگذاریم.

- prob: این تابع با استفاده از کرنل مشخص شده در کانستراکتور و رابطه $p_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}\phi\left[\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h_n}\right]$ احتمال مشاهده هر سطر از بردار ورودی را با توجه به داده‌های دریافت شده از تابع کانستراکتور به دست می‌آورد.

- plot dist: این تابع با استفاده از دستورات displot و line plot از کتاب خانه seaborn هیستوگرام داده‌های یادگیری را به همراه تابع توزیع احتمال تخمین زده شده را رسم می‌کند.

الف

با استفاده از کلاس پیاده سازی شده با کرنل گوسی و $h = 10$ توزیع تابع را برای داده‌های تد به دست آمده و نتیجه در شکل ۷ نشان داده شده است.

ب

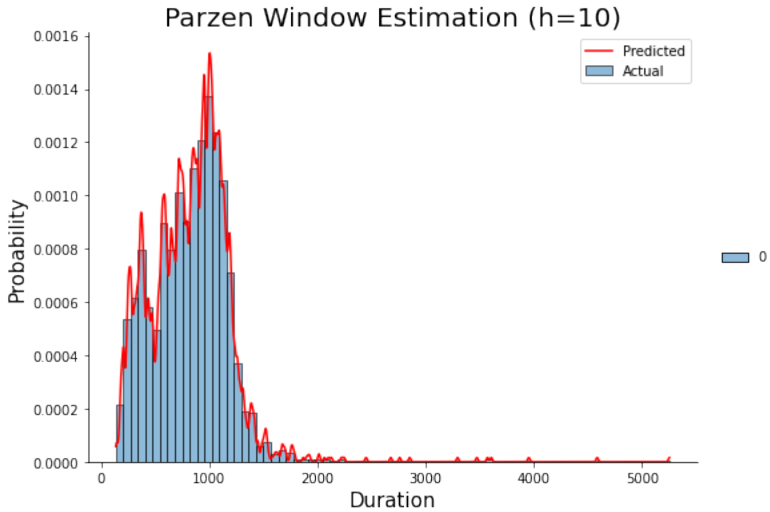
مانند بخش الف این تمرین با تغییر هایپرپارامتر طبقه بند (همان طول پنجره) یادگیری و تخمین توزیع را انجام می‌دهیم نتایج در شکل ۸ نشان داده شده‌اند. اینکه با افزایش h_n تابع توزیع پیش بینی شده نرم‌تر می‌شود کاملاً مشخص است.

ج

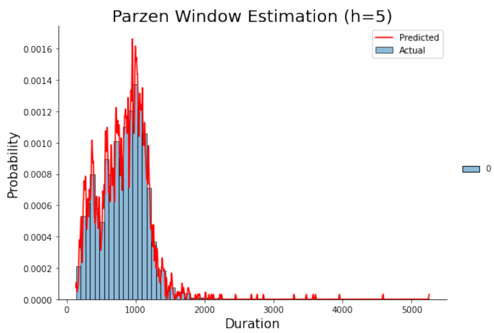
در این قسمت داده‌ها را به طور تدریجی به تخمین‌زن می‌دهیم تا روند تغییر و همگرا شدن به توزیع اصلی را ببینیم. این نمودار در شکل ۹ آورده شده است.

د

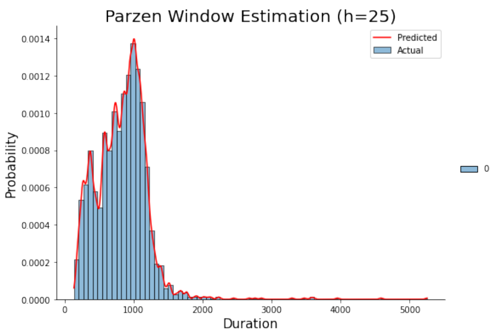
در این قسمت با استفاده از KernelDensity از کتابخانه scikit learn نتایج قسمت الف را صحت سنجی می‌کنیم.
یک آبجکت KernelDensity ساخته و طول پنجره را در آرگومان ورودی (band width) تنظیم می‌کنیم. سپس داده‌های یادگیری را با فراخوانی تابع fit به مدل می‌دهیم. در نهایت می‌توان با تابع score samples احتمال داده‌های تست را حساب کرد. این تابع لگاریتم طبیعی احتمال را در خروجی باز می‌گرداند، پس لازم است آن‌ها را با تابع exp از کتابخانه numpy به احتمال تبدیل کرد.



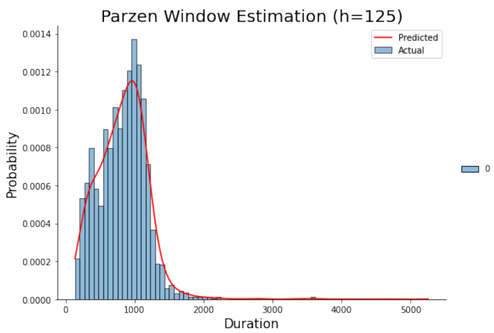
شکل ۷: توزیع تخمین زده شده با $k = 10$



(أ) $h_n = 5$

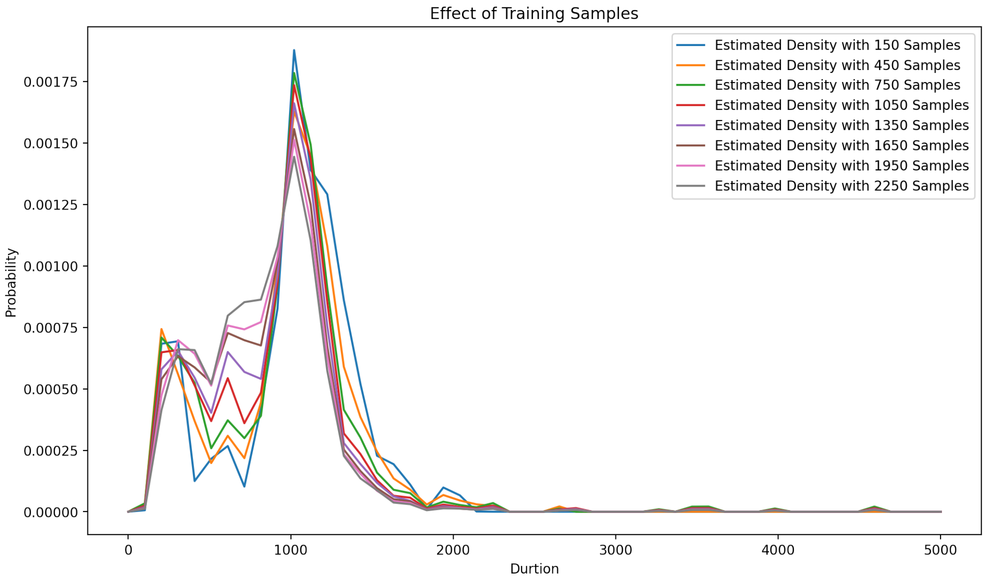


(ب) $h_n = 25$

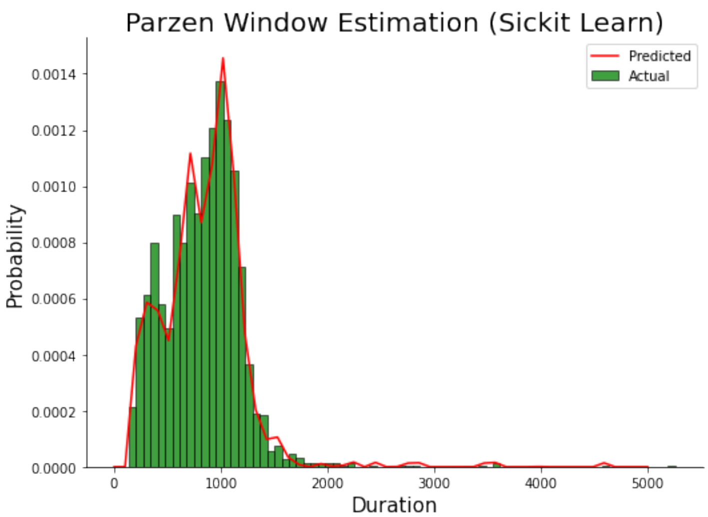


(ج) $h_n = 125$

شکل ۸: توزیع‌های تخمین زده شده با طول پنجره متفاوت



شکل ۹: روند همگرایی تخمین نان‌پارامتری پنجره پارزن



شکل ۱۰: تخمین نان‌پارامتری پارزن با استفاده از کتابخانه scikit learn