به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

یادگیری ماشین

تمرین چهارم

نام و نام خانوادگی : حسین سیفی

شماره دانشجویی : ۸۱۰۱۰۰۳۸۶

خرداد ۱۴۰۲

سوال ۱

بخش ۱

همانطور که می‌دانیم SVM با به حداقل رساندن مقدار تابع خطا در تلاش است که ابرصفحه‌ای با معادله $y = w^t x + b$ پیدا کند که دارای بیشترین Margin ممکن باشد. با استفاده از پارامترهای به دست آمده از معادله فوق می‌توان جهت‌گیری و مکان صفحه جداساز کلاس‌ها را مشخص کرد. جهت صفحه با استفاده از مقادیر w مشخص می‌شود بدین صورت که به صورت مثال در صورتی که فضای ویژگی‌ها را در دو بعد داشته باشیم، جهت‌گیری جداساز دو کلاس که در این فضا یک خط است با کمک بردار وزن که یک شیب خط است مشخص می‌شود. همچنین مکان صفحه نیز به کمک پارامتر b قابل تشخیص است. در مثال فضای دوبعدی مقدار b همان عرض از مبدا است که مشخص می‌کند صفحه‌ای با جهت معین در کدام مکان و فاصله‌ای از مبدا مختصات قرار بگیرد.

بخش ۲

بله روش SVM توانایی تفکیک کلاس‌های دادگان در صورت وجود نویز را دارد. با استفاده از کرنل‌هایی که ابعاد داده‌ها را به بی‌نهایت بعد یا تعداد ابعاد زیادی افزایش می‌دهد، تاثیر نویز بر روی دادگان کاهش می‌یابد. همچنین با در نظر گرفتن خطا برای هر نمونه که در سمت اشتباهی از Margin قرار می‌گیرد و تلاش بر کاهش میزان خطا می‌توان دادگان نویزی را دسته‌بندی کرد. به طور ساده‌سازی شده تابع خطا مورد استفاده به شکل زیر درمی‌آید:

$$L = \frac{||w||^2}{2} + C$$

که در فرمول فوق، C همان تعداد نمونه‌هایی است که به اشتباه دسته‌بندی شده‌اند. با این روش و روش‌های پیچیده‌تر که خطای دقیق‌تر و مطمئن‌تری را برای نمونه‌هایی با دسته‌بندی اشتباه انتخاب می‌کنند، امکان دسته‌بندی دادگان با وجود نویز و بدون مرز قطعی وجود دارد.

بخش ۳

الگوریتم SVM تلاش می‌کند تا با استفاده از یک Hyperplane(در دو بعد با استفاده از خط، و در سه بعد با استفاده از صفحه) داده‌های کلاس‌های متفاوت را از یکدیگر جدا کند اما اگر نتوان داده‌ها را در فضای ویژگی ابتدایی به کمک Hyperplane تفکیک کرد، به کمک یک تابع کرنل از پیش تعریف شده که دارای ویژگی‌هایی مشخص است، داده‌ها را به تعداد ابعادی بالاتر می‌بریم تا بتوان یک Hyperplane مناسب برای تفکیک داده‌ها را پیدا کرد.

می‌دانیم که مقدار w به شکل زیر محاسبه می‌شود:

$$w = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i)$$

برای دو کرنل داده شده مقدار اندازه w به شکل زیر قابل محاسبه است:

$$||w_1|| = \left|\left|\sum_{i=1}^{n} \alpha_i y_i \left(x_i . x_i^2\right)\right|\right|$$

$$||w_2|| = \left|\left|\sum_{i=1}^{n} \alpha_i y_i \left(2x_i . 2x_i^2\right)\right|\right| = 4\left|\left|\sum_{i=1}^{n} \alpha_i y_i \left(x_i . x_i^2\right)\right|\right|$$

همچنین می‌دانیم که مقدار Margin بر اساس w با عبارت زیر به دست می‌آید:

$$margin = \frac{2}{||w||}$$

بنابراین می‌توان نسبت Margin دو تابع را محاسبه کرد:

$$\frac{margin_2}{margin_1} = \frac{\frac{2}{||w_2||}}{\frac{2}{||w_1||}} = \frac{||w_1||}{||w_2||} = \frac{\left|\left|\sum_{i=1}^{n} \alpha_i y_i \left(x_i . x_i^2\right)\right|\right|}{4\left|\left|\sum_{i=1}^{n} \alpha_i y_i \left(x_i . x_i^2\right)\right|\right|} = \frac{1}{4}$$

بنابر محاسبات فوق، با توجه به مقدار به دست آمده فوق مشخص است که حاشیه(Margin) کرنل اول ۴ برابر بزرگتر از حاشیه کرنل دوم است.

سوال ۲

بخش ۱

می‌توان عبارت موجود در توان عدد e را باز کرد و به عبارت زیر رسید:

$$K(x.y) = \exp\left(-\frac{x^T x}{\sigma^2} + \frac{2x^T y}{\sigma^2} - \frac{y^T y}{\sigma^2}\right)$$
$$= \exp\left(-\frac{x^T x}{\sigma^2}\right)\exp\left(\frac{2x^T y}{\sigma^2}\right)\exp\left(-\frac{y^T y}{\sigma^2}\right)$$

اثبات می‌شود که $x^T y$ یک کرنل معتبر است بنابراین $\frac{2x^T y}{\sigma^2}$ یک کرنل با ضریب $\frac{2}{\sigma^2}$ است و در نهایت $K'(x.y) = \exp\left(\frac{2x^T y}{\sigma^2}\right)$ تابع نمایی یک کرنل است که در نتیجه خود آن نیز یک کرنل است. دو عبارت $f(x) = \exp\left(-\frac{x^T x}{\sigma^2}\right)$ و $f(\mathrm{y}) = \exp\left(-\frac{y^T y}{\sigma^2}\right)$ نیز در یک کرنل ضرب شده‌اند و کرنل فوق به شکل زیر درمی‌آید:

$$K(x.y) = f_x K'(x,y) f_y$$

در بخش‌های بعدی اثبات می‌شود که عبارت فوق نیز یک کرنل معتبر است. در نتیجه عبارت زیر یک کرنل معتبر است:

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

بخش ۲

می‌دانیم برای اینکه یک کرنل معتبر باشد، باید Positive semidefinite باشد. بدین معنی است که برای معتبر بودن کرنل K باید عبارت $f'Kf > 0$ هموار صحیح باشد. بنابراین معتبر بودن کرنل ۲ را می‌توان به شکل زیر اثبات کرد:

$$f'Kf = f_x K(x.y) f_y$$
$$= f_x\big(bK_1(x.y) + aK_2(x.y)\big) f_y$$
$$= \{f_x bK_1(x.y) f_y\} + \{f_x aK_2(x.y) f_y\}$$
$$= b\{f'K_1 f\} + a\{f'K_2 f\}$$

با توجه به اینکه کرنل‌های K_1 و K_2 کرنل‌های معتبری هستند، عبارات درون آکلادها همواره مثبت هستند و تنها در صورتی کل عبارت همواره مثبت خواهد بود که متغیرهای a و b دارای مقادیر مثبتی باشند. به عبارتی دیگر:

$$f'Kf > 0\ if\ a > 0\ \&\ b > 0$$

بخش ۳

با استدلالی مشابه با بخش ۲ می‌توان معتبر بودن کرنل ۳ را نیز بررسی کرد:

$$f'Kf = g_x f(x) 1 f(y) g_y$$
$$= h_x 1 h_y$$
$$= h' 1 h$$

تابع h به صورت $h = g_y f(y)$ تعریف شده است و عبارت فوق همواره مثبت است بنابراین K یک کرنل معتبر است.

بخش ۴

این کرنل نیز همانند دیگر بخش‌های این سوال اثبات می‌شود:

$$f'Kf = f_{g(x)}K1\big(g(x).\,g(y)\big)f_{g(y)}$$

این اثبات بیش از این ادامه نمی‌یابد چرا که تابع g تنها یک نگاشت(Transformation) بر روی ابعاد داده‌هاست و تغییری بر روی کرنل ایجاد نمی‌کند. بنابراین عبارت فوق نیز هموار مثبت است.

بخش ۵

مشابه با بخش‌های قبل اثبات می‌شود که:

$$\begin{aligned} f'Kf &= f_x K_1(x.\,y)K_2(x.\,y)f_y \\ &= K_1(x.\,y)\{f_x K_2(x.\,y)f_y\} \\ &= K_1(x.\,y)K_3(x.\,y) \\ &= trace(K_1, K_3') \end{aligned}$$

در عبارت فوق K_3 به صورت $K_3(x.\,y) = f_x K_2(x.\,y) f_y$ تعریف شده است و اثبات می‌شود که کرنل K_3 نیز Positive semidefinite و معتبر است. و در نتیجه ماتریس نتیجه K_1K_3' نیز Positive semidefinite است و دارای Eigenvalueهای نامنفی است بنابراین مجموع Eigenvalueهای ماتریس حاصل(یا همان trace) نیز نامنفی است و بنابراین عبارت فوق همواره مثبت است و این کرنل نیز معتبر است.

سوال ۳

با استفاده از تابع هزینه داده شده می‌توان مسئله مورد نظر برای یک Soft Margin SVM را به شکل یک بهینه‌سازی مقید و به صورت زیر نوشت:

$$L = \min \frac{1}{2}\left||w|\right|^2 + \frac{c}{2}\sum_{i=1}^{N} \varepsilon_i^2$$
$$subject\ to: y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

که با روش Lagrangian می‌توان بهینه‌سازی فوق را به شکل زیر بازنویسی کرد(توجه شود که به ازای هر نمونه، یک قید داریم بنابراین در مجموع N قید داریم):

$$L(w, b, \varepsilon, \lambda) = \frac{1}{2}\left||w|\right|^2 + \frac{c}{2}\sum_{i=1}^{N} \varepsilon_i^2 + \sum_{i=1}^{N} \lambda_i\big(1 - \varepsilon_i - y_i(w^T x_i + b)\big)$$

با مشتق‌گیری از عبارت فوق نسبت به هر یک از پارامتر می‌توان مقادیر آن‌ها را به دست آورد. این عملیات در ادامه انجام شده است:

$$\frac{dL}{dw} = w - \sum_{i=1}^{N} \lambda_i y_i x_i$$
$$\frac{dL}{db} = -\sum_{i=1}^{N} \lambda_i y_i$$

با توجه به مقادیر به دست آمده فوق به نظر می‌آید که با توجه به تابع هزینه داده شده می‌توان پارامترهای مدلی که توانایی تفکیک کلاس‌ها را داشته باشد به دست آورد.

سوال ۴

در این سوال عملکرد طبقه‌بند ماشین بردار پشتیبان بر روی مجموعه دادگان Iris بررسی می‌شود.

بخش ۱

برای این بخش از سوال دو ویژگی اول مجموعه داده Iris که Sepal width و Sepal length نام دارند را انتخاب می‌کنیم و طبقه‌بندی ماشین بردار پشتیبان با کرنل خطی را با استفاده از آن‌ها آموزش می‌دهیم و مقدار معیارهای ارزیابی و ماتریس آشفتگی را بر روی دادگان آموزشی به دست می‌آوریم. ابتدا مجموعه داده به صورت زیر بار می‌شود و ویژگی‌های مورد نظر انتخاب می‌شوند:

```
df = load_iris()

X = df.data[:,:2]

y = df.target
```

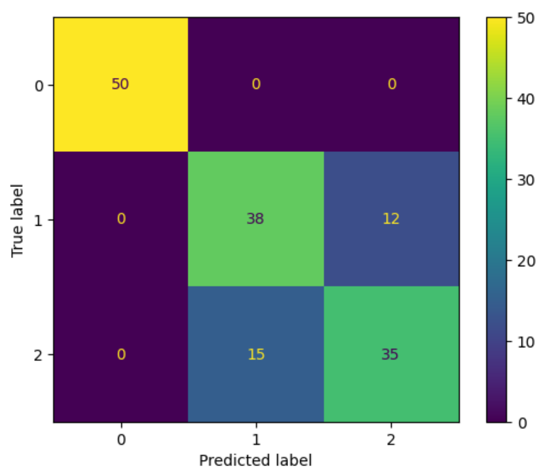
سپس مدل به صورت زیر ایجاد می‌شود و با استفاده از داده‌های آموزشی، آموزش می‌بینند:

```
svmmodel = SVC(kernel='linear').fit(X,y)
```

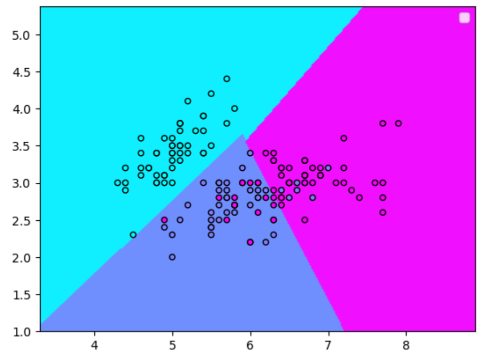
و در نهایت با استفاده از تابع score که تمامی معیارهای ارزیابی مورد نظر را به صورت تجمیع شده محاسبه می‌کند و نمایش می‌دهد، مدل آموزش دیده را ارزیابی می‌کنیم. مقدار معیارهای نام برده به شرح زیر است:

	Precision	Recall	F1-score
Value	0.821	0.82	0.82

و ماتریس آشفتگی به شکل زیر است:



نمودار مرزهای تصمیم بین کلاس‌ها بر اساس دو ویژگی استفاده شده به همراه دادگان آموزشی استفاده شده به شکل زیر می‌باشد:

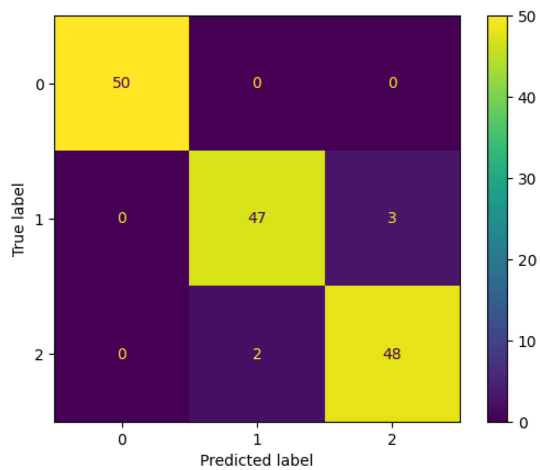


بخش ۲

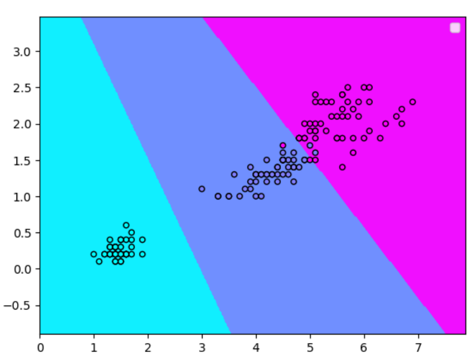
تمامی عملیات بخش ۱ بر روی ویژگی‌های ۳ و ۴ از مجموعه داده Iris با نام‌های Petal length و Petal width انجام می‌گیرد. مقدار معیارهای ارزیابی به شرح زیر است:

	Precision	Recall	F1-score
Value	0.967	0. 967	0. 967

و ماتریس آشفتگی به شکل زیر است:



و مرز تصمیم به شکل زیر رسم شده است:



همانگونه که مشاهده می‌شود مدل SVM با کرنل مشابهی با قسمت قبل، با استفاده از ویژگی‌های جدید عملکرد بهتری نسبت به ویژگی‌های قبلی نشان می‌دهد. این اتفاق نشان می‌دهد که ویژگی‌های Petal length و Petal width تفکیک بیشتری را بر روی برچسب دادگان موجود ایجاد می‌کند و یا به عبارتی دیگر به مقدار بیشتری آنتروپی(بی نظمی) برچسب دادگان را کاهش می‌دهد و Information Gain بیشتری دارد. همچنین با توجه به مرز تصمیم رسم شده، با استفاده از این دو ویژگی کلاس‌ها تفکیک بسیار بیشتری نسبت به دو ویژگی قبل دارند.

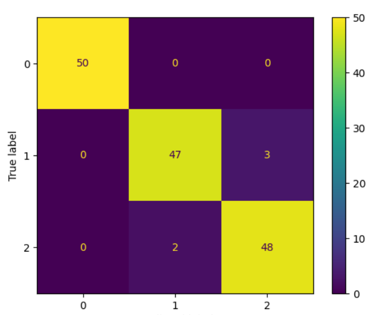
بخش ۳

کرنل خطی یک کرنل ساده است که وقتی داده‌ها به صورت خطی یا حالتی بسیار شبیه به خط قابل تفکیک باشند، یعنی زمانی که می توان از یک خط مستقیم یا یک ابرصفحه برای جداسازی دو کلاس استفاده کرد، به خوبی کار می کند و نسبت به دو کرنل دیگر بسیار کارآمدتر است و دارای بار محاسباتی و هزینه کمتری است. کرنل Poly کرنل پیچیده‌تری است که می تواند روابط غیر خطی بین ویژگی‌ها را درک کند. کرنل چند جمله‌ای داده ها را در فضایی با ابعاد بالاتر ترسیم می‌کند به طوری که مرز تصمیم بتواند یک تابع غیر خطی باشد. درجه چند جمله‌ای پیچیدگی مرز تصمیم را تعیین می‌کند. یک چند جمله‌ای درجه بالا می‌تواند داده‌ها را بهتر تطبیق دهد، اما همچنین می‌تواند منجر به Overfitting شود. کرنل Poly زمانی برای داده‌هایی مفید است که دارای مرز تصمیم Smooth باشند، دارای الگوی خاصی باشند یا شامل داده‌هایی با ترکیبی از روابط خطی و غیر خطی باشند. کرنل RBF یک کرنل پرکاربرد است که برای داده‌های غیرخطی مناسب است. کرنل RBF داده‌ها را در یک فضای بینهایت بعدی نگاشت می‌کند، جایی که مرز تصمیم می‌تواند یک تابع غیر خطی باشد که می تواند هر شکلی داشته باشد. کرنل RBF برای داده‌هایی با ابعاد بالا، نویزی و دارای مرز تصمیم بسیار پیچیده بهترین عملکرد را نسبت به دیگر کرنل‌ها از خود نشان می‌دهد.

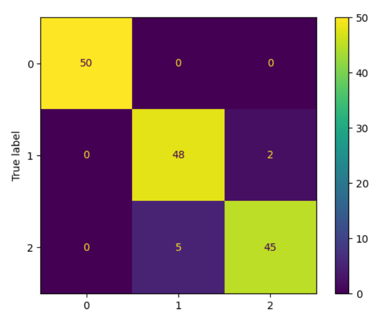
مدل‌هایی با سه کرنل فوق بر روی داده‌های بخش ۲ با همان توضیحات قبلی آموزش داده شدند و نتایج معیارهای ارزیابی برای هر کدام از آن‌ها به شرح زیر است:

Kernel	Precision	Recall	F1-score
Linear	0.967	0. 967	0. 967
RBF	0.954	0.953	0.953
Poly	0.967	0.967	0.967

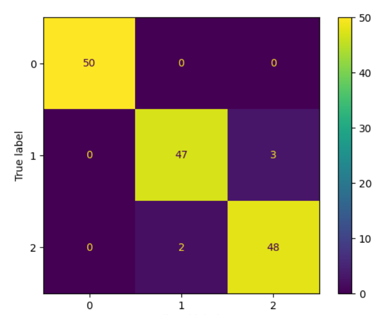
و ماتریس آشفتگی برای هر کدام از کرنل‌ها به شکل زیر است:



Linear kernel

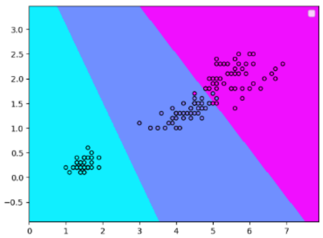


RBF kernel

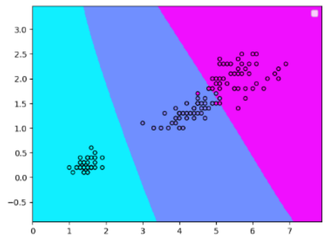


Poly Kernel

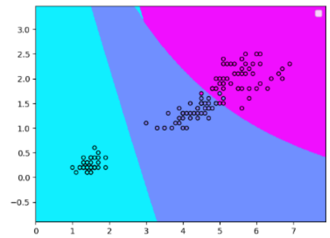
و مرزهای تصمیم به شکل زیر رسم شده‌اند:



Linear kernel



RBF kernel



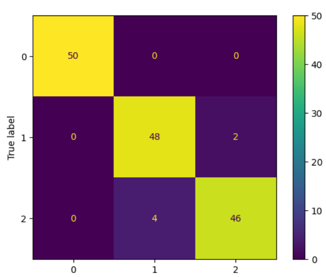
Poly Kernel

همانگونه که مشخص است، با استفاده از این دو ویژگی دادگان آموزشی دارای تفکیک بسیار بالایی هستند و با یک مرز تصمیم خطی دادگان به راحتی از یکدیگر جدا می‌شوند و استفاده از هر کرنل پیچیده‌تر می‌تواند باعث Overfit شدن و افزایش هزینه مدل آموزش دیده شود. همچنین مشاهده می‌شود که دادگان کلاسی که با رنگ آبی روشن مشخص شده است فاصله بسیار زیادی با داده‌های سایر کلاس‌ها دارند و تمامی طبقه‌بندها به خوبی داده‌های این کلاس را از بقیه کلاس‌ها جدا می‌کنند.

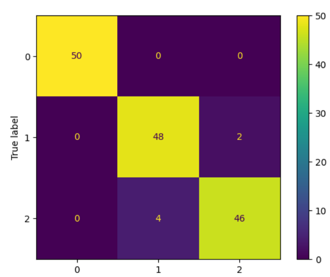
بخش ۴

در ماشین‌های بردار پشتیبان، هایپرپارامتر Regularization اغلب با C نشان داده می‌شود و با کنترل جریمه برای نمونه‌های اشتباه طبقه‌بندی شده، tradeoff بین دستیابی به یک خطای آموزشی کم و یک خطای تست پایین را کنترل می‌کند. به طور خاص، C مقدار وزن داده شده به عبارت Regulator را در بهینه‌سازی مقید به دست آوردن مرزهای تصمیم تعیین می کند. این هایپرپارامتر با سه مقدار ۰.۲۵، ۰.۵ و ۰.۷۵ به همراه کرنل RBF بررسی شده است. لازم به ذکر است که برای این بخش سوال از هر ۴ ویژگی مجموعه داده Iris استفاده شده است. نتایج و ماتریس آشفتگی آن‌ها به شرح زیر است:

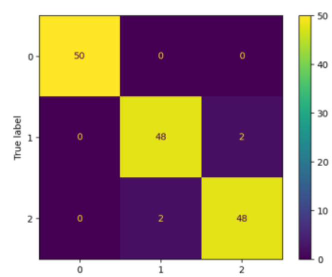
C	Precision	Recall	F1-score
0.25	0.96	0. 96	0. 96
0.5	0.96	0.96	0.96
0.75	0.973	0.973	0.973



C=0.25



C=0.5



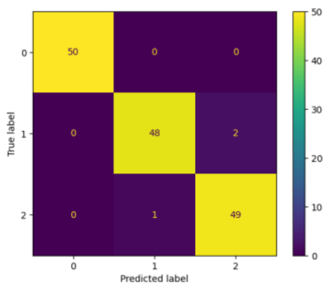
C=0.75

هر چه مقدار C کوچکتر باشد، SVM سعی می‌کند حتی اگر منجر به طبقه‌بندی اشتباه بیشتری در مجموعه آموزشی شود حاشیه را بزرگتر کند. از طرف دیگر، وقتی C بزرگ است، SVM سعی می‌کند تمام نقاط مجموعه آموزشی را به درستی طبقه‌بندی کند و ممکن است توانایی Generalization خود را تا حدی از دست بدهد. با توجه به توضیحات فوق می‌تواند دلیل این را متوجه شد که چرا در آزمایش انجام شده که مدل‌ها با استفاده از داده‌های آموزش ارزیابی شده‌اند، با کاهش مقدار C خطا افزایش یافته است و تعداد نمونه‌هایی که برچسب اشتباه دریافت

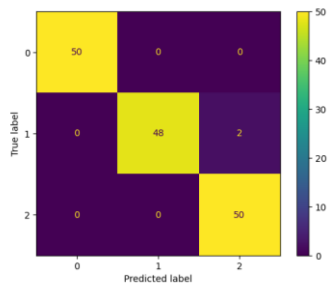
کرده‌اند افزایش یافته است. اگر بخشی از مجموعه داده را به عنوان مجموعه تست در نظر بگیریم احتمالا با کاهش مقدار C شاهد بهبود عملکرد مدل بر روی این دادگان خواهیم بود.

در ماشین‌های بردار پشتیبان هایپرپارامتر Gamma شکل مرز تصمیم را تعیین می‌کند و به طور خاص، این هایپرپارامتر تأثیر هر مثال آموزشی را بر روی مرز تصمیم کنترل می کند. برای این هایپرپارامتر مقادیر ۰.۱، ۰.۲ و ۰.۳ بررسی شده‌اند و نتایج و ماتریس آشفتگی آن‌ها به شرح زیر است:

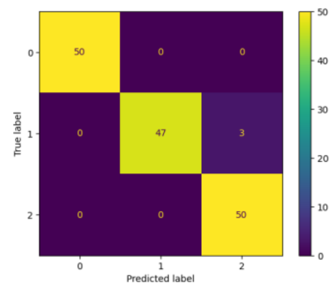
Gamma	Precision	Recall	F1-score
0.1	0.98	0.98	0.98
0.2	0. 987	0. 987	0.987
0.3	0.981	0.98	0.98



Gamma=0.1



Gamma=0.2



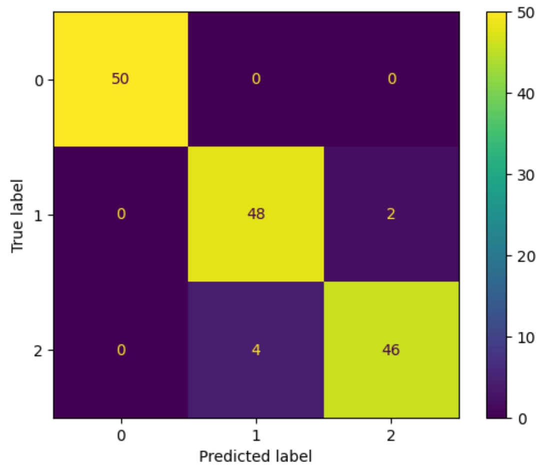
Gamma=0.3

انتخاب مقدار کوچک برای گاما به این معنی است که مرز تصمیم هموارتر و تعمیم‌یافته‌تر خواهد بود و تأثیر هر مثال آموزشی گسترده‌تر خواهد بود. این می تواند منجر به مدلی شود که احتمال Overfit شدن کمتری دارد که بررسی آن با ارزیابی مدل بر روی دادگان تست امکان پذیر است، اما ممکن است با ساده‌سازی مدل منجر به به وجود آمدن خطای Bias بالاتر و خطای واریانس کمتری شود.

بخش ۵

در این بخش همانند بخش ۳ از ویژگی‌های سوم و چهارم مجموعه داده Iris استفاده شده است. می‌توان با استفاده از کتابخانه sklearn و به کمک GridsearchCV بهترین کرنل و مقادیر برای هایپرپارامترهای Gamma و C را به دست آورد. سه کرنل Poly، RBF و Linear و مقادیر ۰.۰۰۱، ۰.۰۱، ۰.۱، ۱، ۱۰ و ۱۰۰ برای هایپرپارامترهای C و Gamma آزمایش شدند و مقادیر معیارهای ارزیابی برای بهترین مدل ایجاد شده به همراه ماتریس آشفتگی به شرح زیر است:

	Precision	Recall	F1-score
Value	0.96	0.96	0.96



همچنین بهترین مدل پیدا شده به کمک Gridsearch دارای پارامترها و کرنل زیر است:

Kernel	C	Gamma
Linear	0.001	0.001

همانگونه که مشاهده می‌شود کرنل خطی به همراهی مقادیر مشخص برای هایپرپارامترهای C و Gamma بهترین عملکرد را از خود نشان داده است. این اتفاق قابل توجهی است که کرنلی که با مقادیر پیش‌فرض برای هایپرپارامترها دارای عملکردی مشابه با کرنل Poly بین سه کرنل موجود بود، با تنظیم مقادیر برای هایپرپارامترها بهترین عملکرد را از خود نشان می‌دهد. همچنین می‌توان مشاهده کرد که این طبقه‌بند نیز مانند تمامی طبقه‌بندهای آموزش دیده در بخش‌های قبل، کلاس ۰ را با بهترین دقت تشخیص می‌دهد و در تشخیص دو کلاس دیگر اندک خطایی دارد.

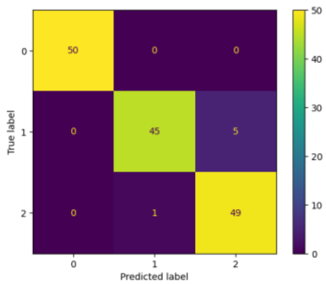
بخش ۶

در این بخش نیز از تمامی ویژگی‌های مجموعه داده Iris استفاده شده است. به کمک کتابخانه Sklearn و کدهای زیر طبقه‌بند One-vs-All و One-vs-One پیاده شده‌اند:

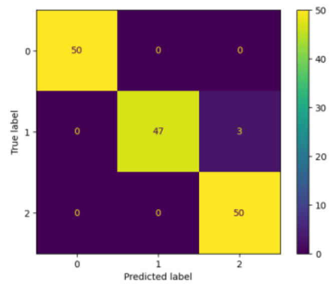
```
model = OneVsOneClassifier(SVC(kernel=k)).fit(X,y)

model = OneVsRestClassifier(SVC(kernel=k)).fit(X,y)
```

مقادیر معیارهای ارزیابی برای رویکرد One-vs-All و کرنل‌های مختلف به شرح زیر است. همچنین برای هر یک ماتریس آشفتگی نیز ضمیمه شده است:

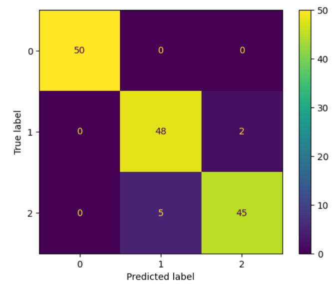
Kernel	Precision	Recall	F1-score
Linear	0.962	0.96	0.96
Poly	0. 981	0. 98	0.98
RBF	0.954	0.953	0.953



Kernel=Linear



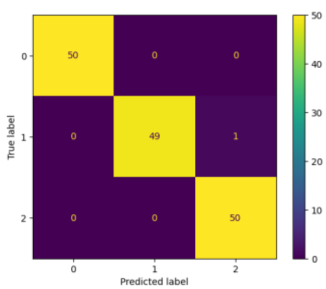
Kernel=Poly



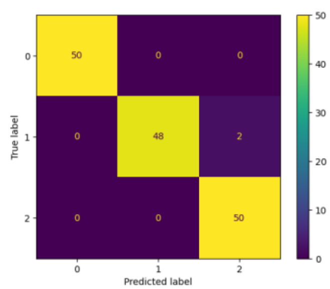
Kernel=RBF

همانگونه که مشخص است بهترین عملکرد را مدلی با کرنل Poly از خود نشان داده است. در ادامه مقادیر معیارهای ارزیابی و ماتریس آشفتگی برای رویکرد One-vs-One و کرنل‌های مختلف نمایش درآمده است:

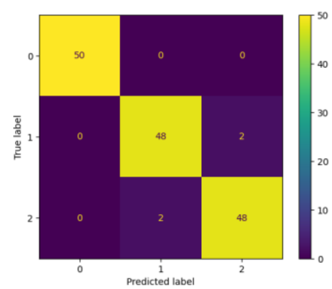
Kernel	Precision	Recall	F1-score
Linear	0.993	0.993	0.993
Poly	0.987	0.987	0.987
RBF	0.973	0.973	0.973



Kernel=Linear



Kernel=Poly



Kernel=RBF

در این رویکرد بهترین عملکرد از آن کرنل Linear است و با اختلاف از سایر کرنل‌ها بهتر عمل کرده است. لازم به ذکر است که مقدار پیش‌فرض هایپرپارامتر d برابر با ۳ و مطابق با خواسته سوال است و نیاز به تغییر دستی آن وجود ندارد.

سوال ۵

بخش ۱

می‌توان نشان داد که:

$$
\left|\left|\varphi(x_i)-\varphi(x_j)\right|\right|^2 =< \varphi(x_i), \varphi(x_i) > + < \varphi(x_j), \varphi(x_j) > -2 < \varphi(x_i), \varphi(x_j) >
$$
$$
= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)
$$

می‌دانیم که $K(x_i, x_i)$ برابر با یک است. با جایگذای مقدار یک به جای $K(x_i, x_i)$ و کرنل RBF داده شده به جای $K(x_i, x_j)$ نتیجه به صورت زیر خواهد بود:

$$
\left|\left|\varphi(x_i)-\varphi(x_j)\right|\right|^2 = 1 + 1 - 2\exp\left(-\frac{1}{2}||x-y||\right) = 2 - 2\exp\left(-\frac{1}{2}||x-y||\right)
$$

مشخص است با توجه به همواره نامنفی بودن عبارت نمایی در نتیجه حاصل شده، در صورتی که این عبارت برابر با صفر باشد، نتیجه برابر با ۲ و در غیر اینصورت نتیجه کوچکتر از ۲ خواهد بود. بنابراین:

$$
\left|\left|\varphi(x_i)-\varphi(x_j)\right|\right|^2 \leq 2
$$

بخش ۲

با انجام ضرب داخلی $x^T y$ و جایگذاری در عبارت تبدیل، این عبارت به شکل زیر درمی‌آید:

$$
K(x,y) = (x_1y_1 + x_2y_2 + \dots + x_ny_n + 1)^2 = \left(\sum_{i=1}^{n} x_iy_i + 1\right)^2
$$

سپس این عبارت را می‌توان با استفاده از اتحاد مربع n+1 جمله‌ای یا همان بسط نیوتن، به شکل زیر گسترش داد:

$$
K(x,y) = ((x_1y_1)^2 + (x_2y_2)^2 + \dots + (x_ny_n)^2) + 2 * (x_1y_1x_2y_2 + x_1y_1x_3y_3 + \dots + x_{n-1}y_{n-1}x_ny_n
$$
$$
+ x_1y_1 + x_2y_2 + \dots + x_ny_n) + 1
$$

همانطور که مشخص است می‌توان نشان داد که پرانتز اول شامل n عبارت است و پرانتز دوم شامل $\frac{n(n+1)}{2}$ عبارت که به دلیل حذف عبارات تکراری و مشابه عمل تقسیم بر دو انجام گرفته است. و در نهایت عدد یک نیز وجود دارد. با جمع کردن تعداد تمامی عبارات فوق به نتیجه زیر می‌رسیم:

$$
\frac{n(n+1)}{2} + n + 1 = \frac{n^2 + n + 2n + 2}{2} = \frac{n^2 + 3n + 2}{2}
$$

با تجزیه عبارت موجود در صورت کسر، نشان داده می‌شود که تعداد ابعاد ایجاد شده با تعداد مطرح شده در سوال برابر است:

$$
\frac{1}{2}(n+1)(n+2)
$$

بخش ۳

در شرایط داده شده به حداقل ۲ SV[1] برای پیدا کردن طبقه‌بند مناسب نیاز است و این تعداد می‌تواند بسیار بیشتر باشد تا جایی که تمامی N نمونه به عنوان SV در دو سمت طبقه‌بند قرار بگیرند. با اضافه کردن یک نمونه به مجموعه دادگان داده شده، در صورتی که فاصله نمونه جدید تا Hyperplane طبقه‌بند برابر با فاصله SVهای قبلی تا طبقه‌بند باشد، تعداد SVها یکی افزایش می‌یابد، در صورتی که این فاصله بیشتر از SVهای قبلی باشد، تعداد SVها ثابت می‌ماند و داده جدید بر روی طبقه‌بند تاثیرگذار نخواهد بود و در صورتی که فاصله نمونه جدید تا Hyperplane کمتر از سایر SVها با Hyperplane باشد، نیاز به تغییر Hyperplane طبقه‌بند وجود دارد چرا که یک طبقه‌بند SVM با فاصله SVهای نامتقارن معتبر نخواهد بود. این تغییر با توجه به شرایط سایر نمونه‌ها می‌تواند شامل اجرای الگوریتم از ابتدا باشد تا طبقه‌بند بر روی دادگان جدید Fit شود یا می‌تواند با دور شدن طبقه‌بند از نمونه جدید تا زمانی که فاصله تا SVها متقارن شود، خاتمه یابد.

[1] Support Vector

سوال ۶

احتمال اینکه $\frac{N+1}{2}$ طبقه‌بند از N طبقه‌بند به کلاس صحیح رای بدهند به شکل زیر محاسبه می‌شود که با توجه به راهنمایی مطرح شده در صورت سوال به شکل یک توزیع برنولی می‌توان به آن نگاه کرد که احتمال صحیح بودن تشخیص کلاس با دقت هر مدل مساوی است و برابر با ۰.۵۱ می‌توان در نظر گرفت:

$$P\left(c = \frac{N+1}{2}\right) = \binom{N}{\frac{N+1}{2}} (0.51)^{\frac{N+1}{2}} (1-0.51)^{N-\frac{N+1}{2}}$$

و با توجه به اینکه تعداد رای‌ها به کلاس صحیح باید حداقل برابر با $\frac{N+1}{2}$ باشد بنابراین احتمال کل تشخیص صحیح کلاس دادگان که برابر با دقت طبقه‌بند است به شکل زیر محاسبه می‌شود:

$$P\left(c \geq \frac{N+1}{2}\right) = \sum_{c=\frac{N+1}{2}}^{N} \binom{N}{c} (0.51)^{c} (1-0.51)^{N-c}$$

بخش ۱

با جایگذاری N=5 در فرمول به دست آمده فوق می‌توان دقت این طبقه‌بند را به دست آورد:

$$\begin{aligned}
&P(c \geq 3) = \sum_{c=3}^{5} \binom{5}{c} (0.51)^{c} (0.49)^{5-c} \\
&= \binom{5}{3} (0.51)^{3} (0.49)^{2} + \binom{5}{4} (0.51)^{4} (0.49)^{1} + \binom{5}{5} (0.51)^{5} (0.49)^{0} \\
&= 0.318495051 + 0.1657474245 + 0.0345025251 \\
&= 0.5187450006
\end{aligned}$$

همانطور که مشاهده می‌شود عملکرد ۵ مدل به صورت جزیی از عملکرد مدل‌ها به صورت مستقل بهتر شده است.

بخش ۲

همانند بخش قبل با جایگذاری N=9 در فرمول به دست آمده می‌توان دقت این طبقه‌بند را به دست آورد:

$$\begin{aligned}
&P(c \geq 5) = \sum_{c=5}^{9} \binom{9}{c} (0.51)^{c} (0.49)^{5-c} \\
&= \binom{9}{5} (0.51)^{5} (0.49)^{4} + \binom{9}{6} (0.51)^{6} (0.49)^{3} + \binom{9}{7} (0.51)^{7} (0.49)^{2} + \binom{9}{8} (0.51)^{8} (0.49)^{1} \\
&\qquad + \binom{9}{9} (0.51)^{9} (0.49)^{0} \\
&= 0.25061424091 + 0.17389559573 + 0.07756858935 + 0.02018366355 + 0.00233416517 \\
&= 0.52459625471
\end{aligned}$$

بخش ۳

به صورت تئوری هنگامی که N به سمت بی‌نهایت میل کند، عبارت حاوی ترکیبیات $\binom{N}{C}$ نیز به سمت بی‌نهایت میل می‌کند اما $(0.51)^{c}(0.49)^{5-c}$ به سمت صفر میل می‌کند که حاصلضرب این دو عبارت نشان می‌دهد که دقت کل مدل به سمت ۱ میل می‌کند. اما

در عمل ممکن است گاهی پیش آید که بیش از نیمی از این مدل‌های ضعیف کلاس غلط را پیش‌بینی کنند و در نتیجه نمونه برچسب غلط را دریافت کند بنابراین به نظر نمی‌رسد که در واقعیت بتوان با چنین مدل‌هایی به دقتی برابر با ۱ رسید.

بخش ۴

در صورتی که دقت طبقه‌بندها برابر با ۰.۵ باشد، می‌توان فرمول را به شکل زیر بازنویسی کرد:

$$P\left(c \geq \frac{N+1}{2}\right) = \sum_{c=\frac{N+1}{2}}^{N} \binom{N}{c} (0.5)^N$$

و بنابراین با جایگذاری N=5 به دقت زیر می‌توان رسید:

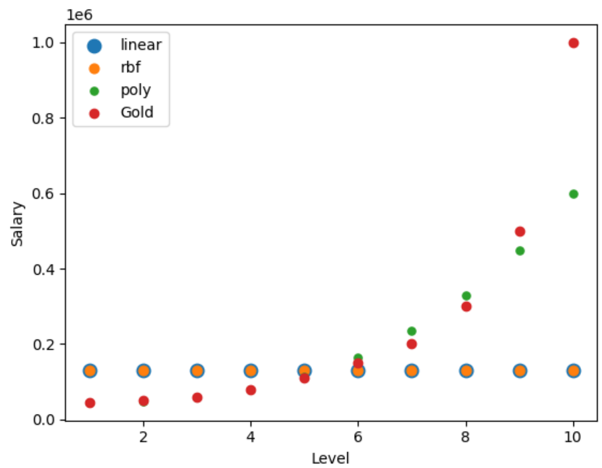
$$\begin{aligned} &P(c \geq 3) = \sum_{c=3}^{5} \binom{5}{c} (0.5)^5 \\ &= (0.5)^5 * (10 + 5 + 1) \\ &= 0.5 \end{aligned}$$

نتیجه می‌گیریم که در صورت استفاده همزمان از تعداد بسیار زیادی از مدل‌هایی که توان تمایز بین کلاس‌ها را ندارند، و تقریبا به شکل تصادفی برچسب نمونه‌ها را انتخاب می‌کند، عملکرد بهبود نمی‌یابد و نتیجه برابر با استفاده از یک مدل است با این تفاوت که در صورت استفاده از تعداد زیادی از این مدل‌ها، وقت و هزینه بیشتری نسبت به استفاده از یک مدل را صرف می‌کنیم.

سوال ۷

بخش ۱

در این سوال به اعمال الگوریتم SVR بر روی مجموعه دادگان داده شده می‌پردازیم. ستون موقعیت شغلی که در هر ستون دارای مقداری یکتاست، با مقادیری مشابه با ستون Level جایگزین می‌شود. سپس با کرنل‌های خطی، چندجمله‌ای و RBF عمل رگرسیون انجام می‌گیرد. پس از آموزش مدل، مقدار متناظر با هر یک از داده‌های آموزشی پیش‌بینی شده است و تمامی مقادیر پیش‌بینی شده به همراه مقدار واقعی نسبت به ستون Level در نمودار زیر به نمایش درآمده است:



همانگونه که مشاهده می‌شود، مدلی با کرنل چندجمله‌ای و مقدار auto برای هایپرپارامتر Gamma بهترین عملکرد و نزدیک‌ترین پیش‌بینی به مقادیر واقعی را از خود نشان می‌دهد در حالی که کرنل‌های خطی و RBF عملکردی مشابه با یکدیگر و ضعیف را دارند.

بخش ۲

در این سوال نیز به آموزش یک مدل SVR با استفاده از دادگان آموزشی داده شده می‌پردازیم. در ابتدا به انجام پیش‌پردازش بر روی دادگان موجود نیاز داریم. این پیش‌پردازش شامل موارد زیر است:

۱. تشخیص ستون‌هایی با مقادیر غیرعددی
۲. تبدیل ستون‌هایی با مقادیر غیرعددی به مقادیر عددی
۳. حذف مقادیر Null و Nan
۴. نرمال سازی مقادیر به کمک StandardScaler

با کمک قطعه کدی که در ادامه مشخص شده است نام ویژگی‌هایی که فرمت غیرعددی دارند در یک لیست ذخیره می‌شوند:

```
for c in train_df.columns:
  if isinstance(train_df.iloc[0][c], str):
      strcols.append(c)
```

سپس با استفاده از کتابخانه Pandas این ستون‌ها را به فرمتی قابل درک برای SVR تبدیل می‌کنیم. کد مربوط به این کار در ادامه قابل مشاهده است:

```
for c in strcols:
  train_df[c] = train_df[c].astype('category').cat.codes
  test_df[c] = test_df[c].astype('category').cat.codes
```

سپس با استفاده از قطعه‌کدی که در ادامه مشخص است نمونه‌هایی که دارای مقادیر گمشده هستند را از مجموعه دادگان حذف می‌کنیم:

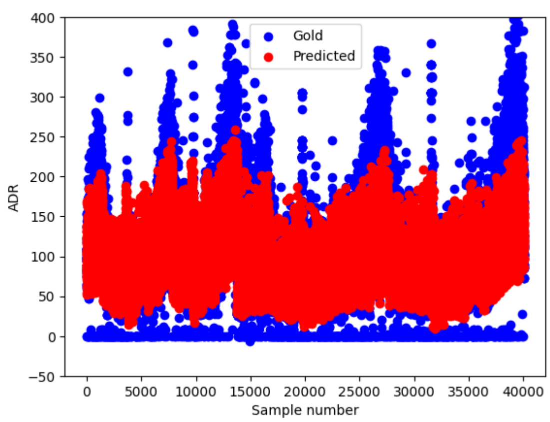
```
train_df.dropna(inplace=True)
test_df.dropna(inplace=True)
```

سپس نرمال‌سازی داده‌ها به کمک پایپلاین کردن یک مبدل[۲] در کنار یک مدل SVR با کرنل چندجمله‌ای انجام می‌دهیم تا به دلیل جدا بودن مجموعه تست از آموزش، از نرمال‌سازی هر دو مجموعه با پارامترهایی یکسان اطمینان حاصل کنیم. کد نرمال‌ساز StandardScaler به همراه مدل SVR و آموزش مدل به شکل زیر است:

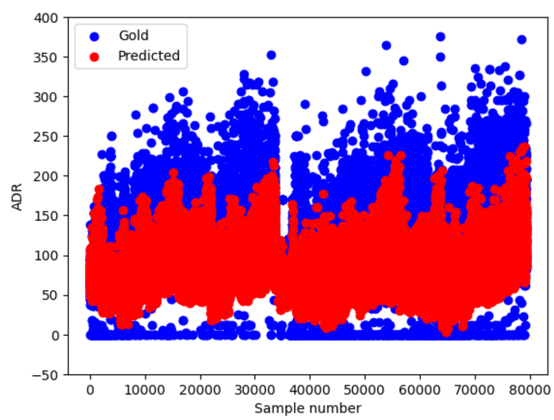
```
model = make_pipeline(StandardScaler(), SVR(kernel='rbf'))
model.fit(X_train, y_train)
```

مدل استفاده شده دارای کرنل RBF و مقادیر پیش‌فرض برای هایپرپارامترها است و به دلیل بزرگ بودن دیتاست مورد نظر، یافتن مقادیر مناسب برای هایپرپارامترها به روش GridSearch نیازمند زمان بسیار زیادی بود و انجام نگرفت بنابراین محتمل است که با استفاده از مقادیر دیگری برای هایپرپارامترها شاهد عملکرد بهتری از مدل باشیم. مقدار Score محاسبه شده توسط تابع مربوطه برای این مدل که Coefficient of determination نام دارد، برابر با ۰.۶۰۷ به دست آمده است.

پس از آموزش مدل به پیش‌بینی مقادیر متناظر با هر یک از نمونه‌های مجموعه دادگان آموزش و تست می‌پردازیم. مقدار پیش‌بینی شده برای دادگان آموزش و تست به همراه مقادیر واقعی بر روی یک نمودار به نمایش درآمده است که به شکل زیر است:



Train



Test

همانطور که مشخص است مدل داده‌های آموزش را با دقت بهتری پیش‌بینی کرده است و می‌توان گفت که مدل تا حدی نویزهای دادگان را نیز یاد گرفته است و مقدار میانگین مطلق خطا[۳] با مقدار ۲۵.۹۳ نیز گواه این گفته است، در صورتی که تخمین مدل بر روی دادگان تست به دقت

---

[2] Transformer

[3] Mean Absolute Error (MAE)

به نسبت ضعیف‌تر است و حتی الگوهایی از دادگان آموزش را می‌توان در مقدار پیش‌بینی شده برای دادگان تست مشاهده کرد. مقدار میانگین مطلق خطا بر روی دادگان تست مقدار ۳۶.۳۲ را نشان می‌دهد که با توجه به بازه مقادیر مجاز برچسب این دادگان که از ۰ تا ۵۴۰۰ است، به نظر نمی‌آید مقداری بسیار ضعیف باشد.

	Train	**Test**
MAE	25.93	36.32
MSE	1480.5	2413.6
RMSE	38.47	49.12
R2	0.61	-0.27

با توجه به اینکه بهترین مقدار برای معیار R2 برابر با ۱ است، به نظر می‌رسد عملکرد مدل بر روی دادگان آموزش عملکرد بدی ندارد اما مقدار این معیار بر روی دادگان تست فاصله بسیار زیادی با دادگان آموزش دارد و عملکرد بسیار ضعیف‌تری دارد.