

در این پیاده‌سازی، به پیاده‌سازی رگرسیون خطی با استفاده از گرادیان کاهشی و مرتب ساز L_2 می‌پردازیم. در ادامه داده را با استفاده از توزیع تصادفی و تابع تعریف شده تولید می‌کنیم. سپس به برازش این داده با مدل رگرسیونی پیاده‌سازی شده می‌پردازیم. در نهایت به بررسی تاثیر نرخ یادگیری در همگرایی و تاثیر وزن منظم‌ساز در بیش برازش می‌پردازیم.

گام اول: کلاس LinearRegressor

کلاس LinearRegressor یک پیاده‌سازی ساده از رگرسیون خطی با استفاده از الگوریتم گرادیان کاهشی و L_2 Regression است. این کلاس می‌تواند برای مدل‌سازی روابط خطی بین متغیرهای ورودی (ویژگی‌ها) و متغیر هدف استفاده شود و از تنظیم L_2 برای جلوگیری از بیش‌برازش (overfitting) بهره می‌برد. در ادامه، متدهای مختلف این کلاس توضیح داده می‌شوند.

۱) `__init__(self, X: np.ndarray, y: np.ndarray)`

این متد سازنده کلاس است و وظیفه دارد که مدل را با ورودی‌ها و پارامترهای اولیه تنظیم کند.

• ورودی‌ها:

X : ماتریس ویژگی‌ها که هر سطر نماینده یک نمونه و هر ستون نماینده یک ویژگی است.

y : بردار هدف که شامل مقادیر واقعی مربوط به هر نمونه است.

• خروجی‌ها (ویژگی‌های کلاس):

X : به ماتریس ویژگی‌ها یک ستون شامل عدد ۱ اضافه می‌شود تا برای مقدار بایاس استفاده شود.

θ : مقادیر اولیه پارامترها به صورت تصادفی تنظیم می‌شوند.

predictions : مقادیر اولیه پیش‌بینی‌ها بر اساس پارامترهای تصادفی محاسبه می‌شوند.

```
۲ train(self, learning_rate: float = 1.0, n_iterations: int  
= 1000, reg_lambda: float = 0.01)
```

این متد برای آموزش مدل با استفاده از گرادیان نزولی و منظم ساز L2 استفاده می‌شود.

- ورودی‌ها:

learning_rate: نرخ یادگیری که میزان تغییرات پارامترها در هر مرحله از گرادیان نزولی را تعیین می‌کند.

n_iterations: تعداد تکرارها برای آموزش مدل.

reg_lambda: پارامتر تنظیم L2 که میزان تأثیر تنظیم منظم ساز را مشخص می‌کند. مقادیر بزرگ‌تر این پارامتر باعث کاهش بیشتر وزن‌های بزرگ می‌شود.

- **خروجی**: پارامترهای θ با هر بار تکرار به‌روزرسانی می‌شوند تا به بهینه‌ترین مقادیر دست پیدا کنند.

- نحوه کارکرد:

ابتدا پیش‌بینی‌ها بر اساس پارامترهای فعلی محاسبه می‌شوند.

اختلاف بین پیش‌بینی‌ها و مقادیر واقعی (خطاها) به‌دست می‌آید.

گرادیان با در نظر گرفتن تأثیر تنظیم L2 محاسبه و سپس پارامترها به‌روزرسانی می‌شوند.

```
۳ predict(self, X_new: np.ndarray) -> np.ndarray
```

این متد برای پیش‌بینی مقادیر هدف برای داده‌های جدید استفاده می‌شود.

- **ورودی‌ها**: X_{new} : داده‌های جدید که شامل ویژگی‌ها برای پیش‌بینی است.

- **خروجی‌ها**: y_{pred} : بردار پیش‌بینی‌شده که مقادیر هدف پیش‌بینی‌شده برای ورودی‌های جدید است.

```
۴ plot(self) -> None
```

این متد برای رسم نمودار مقادیر واقعی و خط رگرسیون استفاده می‌شود.

- نحوه کارکرد:

داده‌های ورودی به صورت نقاط آبی رسم می‌شوند.

خط رگرسیون که مدل یاد گرفته به صورت خط قرمز روی نمودار نمایش داده می شود.

- **استفاده:** این متد برای بصری سازی نحوه یادگیری مدل و عملکرد آن روی داده های آموزشی استفاده می شود.

۵. `mse(self, X: np.ndarray, y: np.ndarray) -> float`

این متد میانگین مربعات خطا MSE را محاسبه می کند. این معیار به ما نشان می دهد که مدل به چه میزان در پیش بینی مقادیر هدف موفق بوده است. هر چه مقدار MSE کمتر باشد، خطای پیش بینی کمتر است.

- **ورودی ها:**

X: داده های ورودی که باید پیش بینی شوند.

y: مقادیر واقعی هدف.

- **خروجی:** مقدار عددی MSE که میانگین مربعات اختلاف بین پیش بینی ها و مقادیر واقعی را نشان می دهد.

۶. `r_squared(self, X: np.ndarray, y: np.ndarray) -> float`

این متد مقدار R^2 را محاسبه می کند که نشان دهنده تناسب مدل با داده های هدف است. مقدار R^2 نشان می دهد که چه درصدی از تغییرات داده های هدف توسط مدل توضیح داده می شود. هر چه مقدار R^2 بیشتر باشد، مدل بهتر عمل کرده است.

- **ورودی ها:**

X: داده های ورودی که باید پیش بینی شوند.

y: مقادیر واقعی هدف.

- **خروجی:** مقدار عددی R^2 که بین ۰ و ۱ است و نشان دهنده درصد واریانس توضیح داده شده توسط مدل است.

گام دوم: تولید و تقسیم داده های تصادفی

در این گام، هدف تولید داده های تصادفی و تقسیم آن ها به مجموعه های آموزشی و آزمایشی به نسبت ۷۰ و ۳۰ است، با حفظ توزیع داده ها با استفاده از Stratify.

۱. تولید داده‌ها

ابتدا دو مجموعه داده تصادفی به ترتیب با ۲۰۰ نمونه تولید می‌شود. در مجموعه اول، رابطه زیر برای داده‌های تولید شده اعمال می‌شود و نویزی تصادفی به آن اضافه می‌گردد:

$$y_1 = 4 + 3 \times X_1 + \text{نویز}$$

در مجموعه دوم، رابطه زیر به همراه نویز گوسی استفاده می‌شود:

$$y_2 = 10 - 2 \times X_2 + \text{نویز}$$

۲. ادغام داده‌ها

داده‌های تولید شده (X_1 و X_2) و برچسب‌های مربوطه (y_1 و y_2) با استفاده از concatenate ادغام می‌شوند و مجموعه داده‌های کلی (X_{concated} و y_{concated}) را تشکیل می‌دهند.

۳. استفاده از Stratify

برای حفظ توزیع در هنگام تقسیم داده‌ها، ابتدا ایندکس داده‌ها با استفاده از np.arange به دست می‌آید. سپس، ایندکس‌ها با استفاده از np.linspace به ۱۰ بازه تقسیم می‌شوند تا توزیع آن‌ها در فرآیند تقسیم داده‌ها حفظ شود. از np.digitize برای تبدیل ایندکس‌ها به شماره بازه‌های مشخص استفاده شده و این مقادیر برای پارامتر stratify در train_test_split استفاده می‌شوند.

۴. تقسیم داده‌ها

داده‌ها با نسبت ۷۰ به مجموعه آموزشی و ۳۰ به مجموعه آزمایشی تقسیم می‌شوند، در حالی که توزیع داده‌ها با استفاده از ایندکس‌های bin بندی شده حفظ می‌شود.

۴. آموزش مدل

مدل رگرسیون خطی (LinearRegressor) با استفاده از داده‌های آموزشی آموزش داده می‌شود. برای این منظور از الگوریتم گرادیان نزولی با نرخ یادگیری ۰.۰۱ و ۱۰۰۰ تکرار استفاده می‌شود.

۵. محاسبه MSE و R^2

پس از آموزش مدل، مقادیر MSE و R^2 برای داده‌های آموزشی و آزمایشی محاسبه می‌شوند. این مقادیر نشان می‌دهند که مدل تا چه حد توانسته مقادیر واقعی را پیش‌بینی کند و چه درصدی از واریانس داده‌ها توسط مدل توضیح داده شده است.

۶. چاپ نتایج

نتایج به‌دست‌آمده شامل MSE و R^2 برای هر دو مجموعه آموزشی و آزمایشی چاپ می‌شوند.

گام سوم: آموزش و ارزیابی مدل رگرسیون روی داده‌های ترکیبی

در این بخش، مدل رگرسیون خطی روی داده‌های ترکیبی (داده‌های هر دو مجموعه) آموزش داده می‌شود و سپس عملکرد مدل با استفاده از معیارهای MSE و R^2 بررسی می‌گردد.

۱. آموزش مدل روی داده‌های ترکیبی

مدل رگرسیون خطی LinearRegressor با استفاده از داده‌های ترکیبی (شامل ۴۰۰ نمونه از دو مجموعه) آموزش داده می‌شود. از الگوریتم گرادیان نزولی با نرخ یادگیری ۰.۰۱ و ۱۰۰۰ تکرار برای به‌روزرسانی وزن‌های مدل استفاده می‌شود.

۲. محاسبه وزن‌ها، MSE و R^2

پس از آموزش، وزن‌های مدل چاپ می‌شوند تا مشخص شود که هر ویژگی چه میزان تأثیر روی خروجی نهایی دارد. سپس MSE (میانگین مربعات خطا) و R^2 (ضریب تعیین) برای داده‌های آموزشی ترکیبی محاسبه و چاپ می‌شوند. MSE نشان‌دهنده میزان خطای مدل و R^2 نشان‌دهنده درصد واریانس توضیح داده‌شده توسط مدل است.

۳. رسم خط رگرسیون و داده‌های واقعی

نموداری ترسیم می‌شود که در آن داده‌های واقعی به صورت نقاط و خط رگرسیون (بر اساس وزن‌های به‌دست‌آمده) به صورت خط نمایش داده می‌شوند.

گام چهارم: آموزش و ارزیابی مدل رگرسیون روی تقسیم‌های مختلف داده‌ها

در این گام، دو مدل رگرسیون خطی روی دو مجموعه داده‌ی مختلف آموزش داده می‌شوند. هر مدل با استفاده از یک زیرمجموعه از داده‌ها (یک بار از X_1 و y_1 و بار دیگر از X_2 و y_2) آموزش داده می‌شود و سپس با استفاده از داده‌های آزمایشی ارزیابی می‌شود.

۱. آموزش مدل روی اولین تقسیم از داده‌ها

مدل رگرسیون خطی اول روی داده‌های X_1 و y_1 آموزش داده می‌شود. از الگوریتم گرادیان نزولی با نرخ یادگیری ۰.۰۱ و ۱۰۰۰ تکرار برای آموزش استفاده می‌شود. پس از آموزش، وزن‌های مدل چاپ می‌شوند. سپس مقادیر MSE و R^2 برای داده‌های آموزشی محاسبه و چاپ می‌شوند. در نهایت، نموداری از داده‌های واقعی و خط رگرسیون ترسیم می‌شود.

۲. آموزش مدل روی دومین تقسیم از داده‌ها

مدل رگرسیون خطی دوم روی داده‌های X_2 و y_2 آموزش داده می‌شود. مشابه مدل اول، از الگوریتم گرادیان نزولی با نرخ یادگیری ۰.۰۱ و ۱۰۰۰ تکرار استفاده می‌شود. سپس وزن‌های مدل، MSE و R^2 برای داده‌های آموزشی محاسبه و چاپ می‌شوند. مشابه مدل اول، نموداری از داده‌های واقعی و خط رگرسیون ترسیم می‌شود.

گزارش مقایسه تأثیر مقادیر مختلف λ در رگرسیون با استفاده

از L2 Regularization

در این بخش از پروژه، تأثیر مقادیر مختلف λ در رگرسیون خطی با استفاده از L2 Regularization بر روی دو مجموعه داده مورد بررسی قرار می‌گیرد. هدف از این مرحله مشاهده تغییرات خط رگرسیون با تغییر مقادیر و ارزیابی تأثیر منظم‌سازی L2 بر نتایج مدل است.

در این قسمت، مدل رگرسیون خطی با استفاده از داده‌های X_1 و y_1 آموزش داده می‌شود و تأثیر مقادیر مختلف بر روی خط رگرسیون مورد بررسی قرار می‌گیرد.

۱. **تنظیمات مدل:** پنج مقدار مختلف λ شامل انتخاب شده‌اند. مدل رگرسیون خطی با استفاده از هر یک از این مقادیر λ آموزش داده می‌شود. نرخ یادگیری ۰.۰۱ و تعداد تکرار ۱۰۰۰ برای الگوریتم گرادیان نزولی در نظر گرفته شده است.

۲. **رسم نمودار:** برای هر λ ، خط رگرسیون تولید شده بر روی نمودار رسم می‌شود و هر خط با رنگ متفاوتی نمایش داده می‌شود. نقاط داده‌های اصلی (X_1 و y_1) به رنگ سیاه روی نمودار نمایش داده می‌شوند.

۳. **نتیجه‌گیری:** نمودار به وضوح تأثیر λ بر نتایج رگرسیون را نشان می‌دهد. با افزایش مقدار λ ، منظم‌سازی L2 تأثیر بیشتری در کاهش ضرایب و صاف‌تر کردن خط رگرسیون دارد، به طوری که در مقادیر بالای λ مدل تمایل دارد داده‌ها را کمتر دنبال کند و خط صاف‌تری ایجاد کند.

بخش دوم: بررسی تأثیر مقادیر مختلف λ در مجموعه داده X_2, y_2

در این قسمت، مدل رگرسیون خطی با استفاده از داده‌های X_2 و y_2 آموزش داده می‌شود و تأثیر مقادیر مختلف λ مشابه با مجموعه اول، بر روی خط رگرسیون بررسی می‌گردد.

۱. **تنظیمات مدل:** همانند بخش اول، پنج مقدار مختلف λ برای آموزش مدل انتخاب شده است. مدل رگرسیون خطی با هر مقدار λ آموزش داده می‌شود و نرخ یادگیری و تعداد تکرار مشابه با قسمت اول تنظیم شده است.

۲. **رسم نمودار:** هر خط رگرسیون تولید شده با مقدار λ مشخص، بر روی نمودار رسم می‌شود و هر خط با رنگ متفاوتی نمایش داده می‌شود. نقاط داده‌های اصلی (X_2 و y_2) به رنگ سیاه روی نمودار نمایش داده می‌شوند.

۳. **نتیجه‌گیری:** همانند بخش اول، تأثیر λ بر خط رگرسیون به وضوح مشاهده می‌شود. در مقادیر پایین λ ، خط رگرسیون به داده‌ها نزدیک‌تر است، اما با افزایش مقدار λ ، خط رگرسیون صاف‌تر و تأثیر منظم‌سازی بیشتر می‌شود.

گام پنجم: گزارش مقایسه عملکرد سه مدل رگرسیون

در این مرحله از پروژه، هدف مقایسه عملکرد سه مدل رگرسیون خطی مختلف است که بر روی سه مجموعه داده متفاوت آموزش داده شده‌اند. برای این کار از دو معیار اصلی **میانگین مربعات خطا** و **ضریب تعیین**

استفاده شده است. این معیارها به ما کمک می‌کنند تا بفهمیم هر مدل تا چه حد دقیق است و چقدر می‌تواند تغییرات داده‌های هدف را توضیح دهد.

مدل‌های مورد بررسی:

- **مدل اول:** آموزش‌دیده بر روی داده‌های X_1 و y_1 .
- **مدل دوم:** آموزش‌دیده بر روی داده‌های X_2 و y_2 .
- **مدل سوم:** آموزش‌دیده بر روی داده‌های ترکیبی (X_{concat} و y_{concat}).

۱. محاسبه MSE (میانگین مربعات خطا)

MSE معیاری است که به ما نشان می‌دهد مدل چقدر در پیش‌بینی مقادیر واقعی موفق بوده است. هر چه مقدار MSE کمتر باشد، خطای مدل کمتر است.

۲. محاسبه R^2 (ضریب تعیین)

R^2 نشان می‌دهد که چه درصدی از تغییرات داده‌های هدف توسط مدل توضیح داده می‌شود. در شرایط عادی، مقدار R^2 بین ۰ و ۱ است. اما در این مرحله، برای داده‌های X_2 و داده‌های ترکیبی، مقدار R^2 منفی شده است.

توضیح نتایج غیرمنتظره برای R^2 منفی

مقدار منفی R^2 به معنای عملکرد ضعیف مدل است که حتی استفاده از میانگین مقادیر واقعی داده‌ها نتایج بهتری نسبت به پیش‌بینی‌های مدل دارد. دلایل احتمالی شامل موارد زیر است:

- **Overfitting or Underfitting:** مدل نتوانسته الگوهای داده‌ها را به خوبی یاد بگیرد.
- **نامناسب بودن λ :** مقادیر خیلی زیاد برای λ می‌تواند باعث ساده‌سازی بیش از حد مدل شود.
- **پیچیدگی داده‌های ترکیبی:** داده‌های ترکیبی ممکن است به اندازه‌ای پیچیده باشند که مدل خطی نتواند آن‌ها را به درستی پیش‌بینی کند.

نتیجه گیری کلی

با مقایسه MSE و R^2 برای سه مدل مختلف، متوجه شدیم که مدل ترکیبی و مدل مربوط به داده های X_2 عملکرد ضعیفی داشته و مقدار R^2 منفی شده است. تنظیم بهتر λ و تحلیل دقیق تر داده ها می تواند به بهبود نتایج کمک کند.

گام ۶. بیش برآزش مجموعه ادغام شده

با توجه به خروجی پیاده سازی داریم:

- Training MSE: 1.89
- Training R-squared: -0.33
- Test MSE: 7.37
- Test R-squared:-0.68

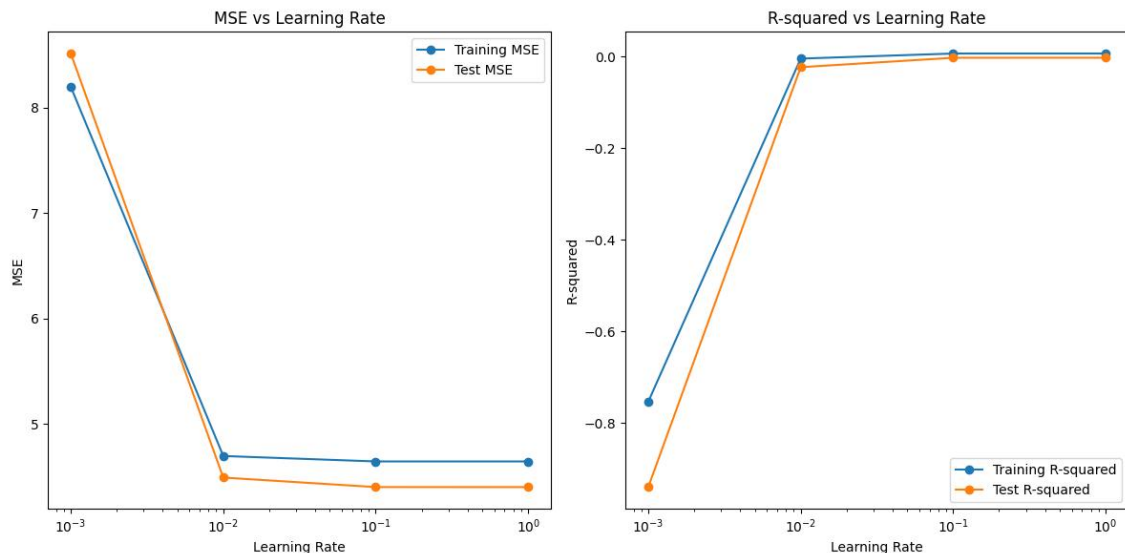
تحلیل

۱. اختلاف بین MSE در داده های آزمایشی و داده های آموزشی: مقدار MSE روی داده های آموزشی بسیار کمتر از این مقدار روی داده های آزمایشی است و این به این معناست که مدل بر روی داده های آموزشی به خوبی عمل کرده و عملکرد آن بر روی داده های آزمایشی ضعیف تر است. این اختلاف زیاد معمولاً نشان دهنده بیش برآزش است.

۲. مقادیر R-squared: مقدار R^2 در هر دو مجموعه داده منفی است. این به این معناست که مدل نمی تواند حتی بهتر از خط افقی میانگین عمل کند. این مقدار منفی به این معناست که مدل به خوبی عمل نکرده و نیاز به تنظیم دارد.

گام ۸. تاثیر نرخ یادگیری

نرخ یادگیری خیلی بالا مثل ۱ یا بالاتر ممکن است باعث شود مدل در حالی که به سمت مقدار کمینه حرکت می کند از مقدار کمینه عبور کند و نتواند به این نقطه برسد. در این حالت مقدار تابع در حال نوسان است، و اگر شده و مدل نمی تواند به خوبی آموزش ببیند. همچنین اگر نرخ یادگیری خیلی پایین باشد، مثل ۰.۰۱، حرکت مدل به سمت مقدار کمینه خیلی کند می شود و زمان فرایند آموزش را افزایش می دهد. شکل زیر تاثیر نرخ یادگیری های مختلف را بر پیش بینی مدل نشان می دهد.



گام ۹. تاثیر منظم‌ساز

در تنظیم با منظم‌ساز یک جمله به مقدار تابع هزینه به عنوان جریمه اضافه می‌شود. هدف از اضافه کردن جریمه این است که از ضرایب بزرگ که موجب بیش‌برازش می‌شود، جلوگیری کند. تابع هزینه با منظم‌ساز L2 به صورت زیر محاسبه می‌شود:

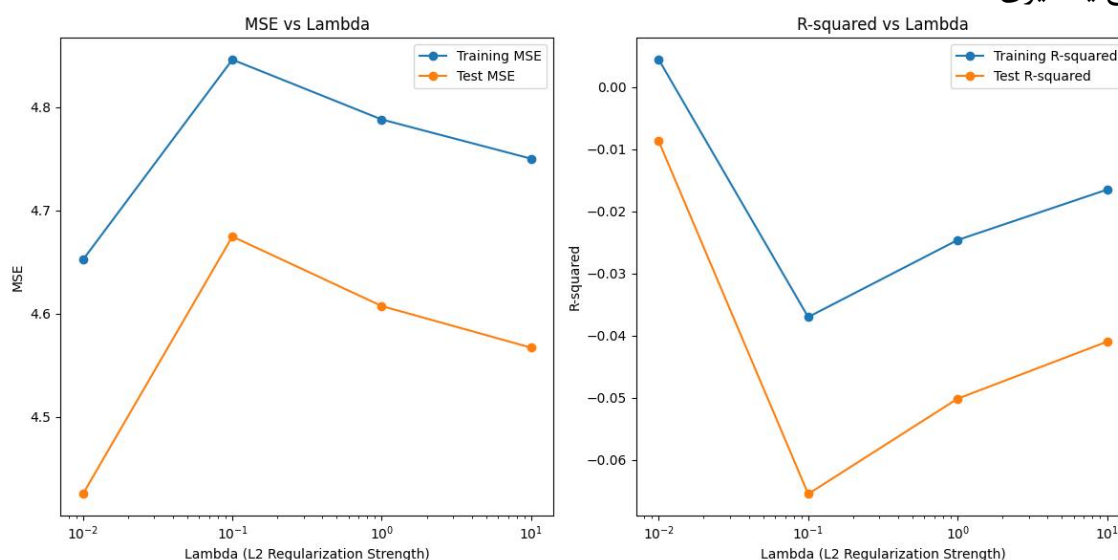
$$J(\theta) = L(\theta) + \lambda \sum_{j=0}^n \theta_j^2$$

۱. **مقادیر خیلی کوچک:** زمانی که مقدار λ بسیار کوچک و نزدیک به صفر است تاثیر خیلی کمی روی مدل دارد و عمل تنظیم‌سازی انجام نشده است. در این حالت مدل می‌تواند از روی مقادیر بزرگ ویژگی‌ها یاد بگیرد و دچار بیش‌برازش شود به عنوان مثال زمانی که مقدار $\lambda = 0.001$ باشد، مدل تنظیم نمی‌شود و همچنان پیچیدگی بالایی دارد. به دنبال آن مقدار MSE داده آموزشی کم و در داده آزمایشی زیاد می‌شود و شاهد بیش‌برازش هستیم.

۲. **مقادیر متعادل:** زمانی که مقدار λ در بازه ۰.۱ و ۱ است، مدل تنظیم شده و ضرایب بزرگ محدود می‌شوند. این عملیات از پیچیدگی مدل کاسته و باعث می‌شود مدل دچار بیش‌برازش نشود. در این حالت مدل هنوز شامل تمام ویژگی‌ها هست، اثر آن‌ها متوازن‌تر شده و قادر است از آن‌ها یاد بگیرد ولی همزمان حساسیت نسبت به داده نویز نداشته و از آن‌ها یاد نمی‌گیرد. به عنوان مثال وقتی مقدار $\lambda = 0.1$ است، مدل به صورت متعادل تنظیم می‌شود و به دنبال آن مقدار MSE داده آموزشی نسبت به حالت قبلی افزایش یافته اما در کل مناسب است. از طرفی این مقدار برای داده‌های آزمایشی افزایش یافته و عملکرد خوبی خواهد داشت و دچار بیش‌برازش نخواهد شد.

۳. **مقادیر خیلی بزرگ**: زمانی که مقدار λ خیلی بزرگ است، وزن‌های بزرگ به شدت تحت تاثیر قرار می‌گیرند و در نتیجه مدل خیلی ساده شده و نمی‌تواند الگوی بین داده‌ها را یاد بگیرد. در این صورت مدل دچار کم برازش می‌شود. به عنوان مثال وقتی $\lambda = 10$ مقدار MSE در داده‌های آموزشی زیاد بوده پس مدل به خوبی از این داده‌ها یاد نگرفته است؛ همزمان این مقدار در داده‌های آزمایشی نیز زیاد است و نمی‌تواند آن‌ها را به درستی حدس بزند و بر روی آن‌ها هم به خوبی عمل نکرده و شاهد کم برازش هستیم.

با توجه به ماهیت داده‌های ما در این مساله، نمودار زیر نشان‌دهنده‌ی تاثیر تنظیم‌کننده‌های مختلف بر نرخ یادگیری است.



گام ۱۰. اثر پارامترهای مختلف بر بهبود و وخامت مدل

هدف از پارامترهای مختلف بهبود مدل و ارائه حدسی دقیق‌تر از مقدار هدف با استفاده از ماتریس ویژگی‌هاست. مقدار این پارامترها می‌تواند در جهت‌های متفاوتی مدل را سوق بدهد. اگر مقدار نرخ یادگیری بالا باشد، تابع هزینه به سمت واگرایی می‌رود و مدل به جای بهینه‌سازی تدریجی به طور ناگهانی تغییرات بزرگی در ضرایب ایجاد می‌کند. اگر نرخ یادگیری خیلی پایین باشد، تابع هزینه به کندی حرکت می‌کند، ممکن است در کمینه محلی متوقف شده و یا ممکن است در محدوده تعداد گام‌ها به مقدار کمینه نرسیم. لذا مهم است که این مقدار به گونه‌ای متعادل و میانه تعیین شود که مدل را به سمت بهبود ببرد. این مقدار از روی آزمون و خطا می‌تواند انتخاب شود.

همچنین در رابطه با مقدار منظم‌ساز، اگر این مقدار خیلی کوچک باشد، تاثیری بر مقدار ضرایب نداشته و بیش‌برازش در هر صورت رخ می‌دهد. در عین حال اگر این مقدار منظم‌ساز خیلی زیاد باشد، ممکن است که بعضی از ویژگی‌های مهم را کم اثر شوند و با کم برازش مواجه شویم. روش‌های مختلفی برای پیدا کردن

بهترین مقدار λ وجود دارد که یکی از آن‌ها روش Cross Validation است که می‌تواند ما را به مقدار بهینه برساند. در حالت بهینه مقدار جریمه اثر مقادیر خیلی بزرگ را کاهش می‌دهد و از بیش‌برازش جلوگیری می‌کند.

نتیجه‌گیری

- مقادیر خیلی کوچک نرخ یادگیری می‌تواند باعث Overfitting شود، زیرا مدل بیش از حد به داده‌های آموزشی وابسته می‌شود و به نویزها نیز حساسیت نشان می‌دهد.
- مقادیر خیلی بزرگ نرخ یادگیری می‌تواند باعث Underfitting شود و در نتیجه مدل نتواند به درستی از داده‌ها یاد بگیرد.
- مقادیر متعادل این پارامترها می‌تواند عملکرد بهینه‌ای را فراهم کند، زیرا هم از Overfitting جلوگیری می‌کند و هم اجازه می‌دهد که مدل به درستی الگوهای داده‌ها را یاد بگیرد.