بسمه تعالی

پیاده‌سازی رگرسیون خطی به روش پرسپترون

پیش‌بینی قیمت مسکن با دیتاست Boston Housing

چکیده

در این تمرین یک مدل رگرسیون خطی تک‌لایه، مشابه پرسپترون با خروجی پیوسته، بدون استفاده از مدل‌های آماده پیاده‌سازی شد. هدف، پیش‌بینی مقدار میانه قیمت خانه‌ها در دیتاست Boston Housing بود. از میان ۱۳ ویژگی ورودی، دو ویژگی RM و LSTAT به دلیل بیشترین قدر مطلق همبستگی با متغیر هدف MEDV انتخاب شدند. داده‌ها به نسبت ۸۰ درصد آموزش و ۲۰ درصد آزمون تقسیم و ویژگی‌ها و هدف با StandardScaler استاندارد شدند. مدل با گرادیان‌کاهشی دسته‌ای، نرخ یادگیری ۰٫۰۵ و ۱۰۰۰ epoch آموزش دید. روی ۱۰۲ نمونه آزمون، مقادیر MSE برابر ۳۱٫۲۴۳، RMSE برابر ۵٫۵۹۰، MAE برابر ۳٫۸۹۹ و ضریب تعیین R^2 برابر ۰٫۵۷۴ به دست آمد. نتایج نشان می‌دهد مدل ساده با دو ویژگی توانسته است ۵۷٫۴ درصد تغییرات قیمت را توضیح دهد، اما برای ثبت روابط غیرخطی و عوامل مؤثر دیگر محدودیت دارد.

۱. تعریف مسئله و هدف

مسئله از نوع رگرسیون نظارت‌شده است؛ زیرا خروجی MEDV یک مقدار عددی پیوسته است. مدل باید بر اساس اطلاعات هر ناحیه، قیمت میانه خانه را پیش‌بینی کند. برخلاف پرسپترون طبقه‌بندی، در خروجی تابع پله استفاده نمی‌شود و خروجی خطی باقی می‌ماند. رابطه مدل با دو ویژگی انتخاب‌شده به‌صورت زیر است:

$$\hat{y} = w_1 \times RM + w_2 \times LSTAT + b$$

- RM: میانگین تعداد اتاق‌ها در هر واحد مسکونی.
- LSTAT: درصد جمعیت با وضعیت اجتماعی–اقتصادی پایین‌تر.
- MEDV: مقدار میانه قیمت خانه‌ها بر حسب هزار دلار در تعریف تاریخی دیتاست.

هدف آموزشی تمرین، درک مستقیم پارامترهای وزن و بایاس، محاسبه تابع خطا، استخراج گرادیان و به‌روزرسانی دستی پارامترها است. به همین دلیل الگوریتم رگرسیون در قالب کلاس RegressionPerceptron از ابتدا نوشته شده است.

۲. معرفی دیتاست Boston Housing

دیتاست از OpenML با نام boston و نسخه ۱ دریافت شده است. این مجموعه شامل ۵۰۶ رکورد، ۱۳ ویژگی توضیحی و یک متغیر هدف است؛ بنابراین DataFrame نهایی ۵۰۶ سطر و ۱۴ ستون دارد. همه ستون‌ها برای محاسبات رگرسیون به نوع عددی تبدیل شده‌اند.

ویژگی	شرح
CRIM	نرخ جرم سرانه در هر ناحیه
ZN	نسبت زمین‌های مسکونی با قطعات بزرگ
INDUS	نسبت زمین‌های تجاری و صنعتی
CHAS	مجاورت با رودخانه چارلز
NOX	غلظت اکسیدهای نیتروژن
RM	میانگین تعداد اتاق‌ها
AGE	نسبت ساختمان‌های قدیمی
DIS	فاصله وزنی تا مراکز اشتغال
RAD	شاخص دسترسی به بزرگراه
TAX	نرخ مالیات بر دارایی
PTRATIO	نسبت دانش‌آموز به معلم
B	متغیر تاریخی جمعیتی دیتاست
LSTAT	درصد جمعیت با وضعیت اقتصادی–اجتماعی پایین‌تر
MEDV	متغیر هدف: میانه ارزش خانه

این دیتاست تاریخی دارای محدودیت‌های اجتماعی و اخلاقی شناخته‌شده است و یکی از متغیرهای آن بر مبنای تعریف‌های جمعیتی قدیمی ساخته شده است. در این تمرین فقط RM و LSTAT استفاده شده‌اند؛ بااین‌حال نتایج نباید برای تصمیم‌گیری واقعی درباره افراد یا مناطق به کار روند.

۳. تحلیل همبستگی و انتخاب ویژگی

ضریب همبستگی پیرسون هر ویژگی با MEDV محاسبه و بر اساس قدر مطلق مرتب شد. LSTAT با ضریب منفی ۰٫۷۳۸ و RM با ضریب مثبت ۰٫۶۹۵ قوی‌ترین ارتباط خطی را با قیمت داشتند؛ به همین دلیل برای مدل دوبعدی انتخاب شدند. علامت منفی LSTAT

یعنی افزایش این متغیر معمولاً با کاهش قیمت همراه است و علامت مثبت RM یعنی خانه‌های دارای تعداد اتاق بیشتر معمولاً قیمت بالاتری دارند.

رتبه	ویژگی	همبستگی با MEDV	تفسیر
۱	LSTAT	−۰,۷۳۸	رابطه منفی قوی
۲	RM	+۰,۶۹۵	رابطه مثبت قوی
۳	PTRATIO	−۰,۵۰۸	رابطه منفی متوسط
۴	INDUS	−۰,۴۸۴	رابطه منفی متوسط
۵	TAX	−۰,۴۶۹	رابطه منفی متوسط
۶	NOX	−۰,۴۲۷	رابطه منفی متوسط
۷	CRIM	−۰,۳۸۸	رابطه منفی ضعیف‌تر
۸	RAD	−۰,۳۸۲	رابطه منفی ضعیف‌تر
۹	AGE	−۰,۳۷۷	رابطه منفی ضعیف‌تر
۱۰	ZN	+۰,۳۶۰	رابطه مثبت ضعیف‌تر
۱۱	B	+۰,۳۳۳	رابطه مثبت ضعیف‌تر
۱۲	DIS	+۰,۲۵۰	رابطه مثبت ضعیف
۱۳	CHAS	+۰,۱۷۵	کمترین رابطه خطی

انتخاب بر اساس همبستگی، مدل را ساده و قابل رسم می‌کند؛ اما همبستگی فقط رابطه خطی تک‌متغیره را می‌سنجد و وجود هم‌خطی، اثر ترکیبی یا رابطه غیرخطی را بررسی نمی‌کند.

۴. توضیح کد

سلول ۱ ـ واردکردن کتابخانه‌ها و تنظیم Seed

NumPy برای محاسبات ماتریسی، Pandas برای DataFrame، Matplotlib برای رسم، ابزار OpenML برای دریافت دیتاست، train_test_split برای تقسیم داده، StandardScaler برای استانداردسازی و معیارهای sklearn برای ارزیابی وارد شده‌اند. Seed برابر ۴۲ تکرارپذیری تقسیم داده را حفظ می‌کند.

```
np.random.seed(42)
plt.style.use('seaborn-v0_8-whitegrid')
```

سلول ۲ ـ دریافت و آماده‌سازی دیتاست

دیتاست Boston Housing از OpenML با خروجی DataFrame دریافت می‌شود. ویژگی‌ها کپی و target با نام MEDV به آن افزوده می‌شود. تبدیل pd.to_numeric از عددی‌بودن همه ستون‌ها اطمینان می‌دهد. خروجی شکل (۵۰۶, ۱۴) و پنج ردیف اول را نمایش داده است.

```
boston = fetch_openml(name='boston', version=1, as_frame=True, parser='auto')
df['MEDV'] = pd.to_numeric(boston.target)
```

سلول ۳ ـ محاسبه همبستگی و انتخاب ورودی

ماتریس همبستگی عددی ساخته و ستون MEDV استخراج می‌شود. پس از حذف همبستگی هدف با خودش، ویژگی‌ها بر اساس قدر مطلق مرتب می‌شوند. RM و LSTAT به‌عنوان دو ویژگی مدل انتخاب و y به بردار ستونی تبدیل می‌شود تا عملیات ماتریسی وزن‌ها صحیح باشد.

```
features = ['RM', 'LSTAT']
X = df[features].to_numpy(dtype=float)
y = df['MEDV'].to_numpy(dtype=float).reshape(-1, 1)
```

سلول ۴ ـ تقسیم داده و استانداردسازی

با test_size برابر ۰٫۲۰، تعداد ۴۰۴ نمونه برای آموزش و ۱۰۲ نمونه برای آزمون اختصاص می‌یابد. دو StandardScaler جدا برای X و y فقط روی train برازش می‌شوند؛ سپس test با همان پارامترهای train تبدیل می‌شود. این روش از نشت اطلاعات مجموعه آزمون جلوگیری می‌کند.

```
X_train, X_test, y_train, y_test = train_test_split(... test_size=0.20,
random_state=42)
X_train_s = x_scaler.fit_transform(X_train)
X_test_s = x_scaler.transform(X_test)
```

سلول ۵ ـ تعریف RegressionPerceptron

کلاس مدل شامل نرخ یادگیری، تعداد epoch، تاریخچه loss، متد fit و متد predict است. وزن‌ها با صفر و بایاس با صفر آغاز می‌شوند. در هر epoch پیش‌بینی خطی، خطا و MSE محاسبه می‌شود. سپس مشتق MSE نسبت به وزن و بایاس به دست می‌آید و پارامترها در جهت مخالف گرادیان حرکت می‌کنند.

```
y_pred = X @ self.weights + self.bias
error = y_pred - y
dw = (2 / n_samples) * X.T @ error
db = (2 / n_samples) * np.sum(error)
```

سلول ۶ ـ آموزش و ارزیابی

مدل با نرخ یادگیری ۰٫۰۵ و ۱۰۰۰ epoch روی داده استاندارد آموزش می‌بیند. پیش‌بینی آزمون از مقیاس استاندارد به مقیاس اصلی قیمت بازگردانده می‌شود. چهار معیار محاسبه شده و وزن‌های استاندارد گزارش می‌شوند. وزن RM مثبت و وزن LSTAT منفی است که با تحلیل همبستگی سازگار است.

```
model = RegressionPerceptron(learning_rate=0.05, epochs=1000)
model.fit(X_train_s, y_train_s)
y_pred = y_scaler.inverse_transform(model.predict(X_test_s))
```

سلول ۷ ـ رسم روند خطای آموزش

مقادیر MSE ذخیره‌شده در loss_history در برابر epoch رسم می‌شوند. نمودار نشان می‌دهد گرادیان‌کاهشی با نرخ یادگیری انتخاب‌شده پایدار بوده و loss به مقدار همگرا می‌رسد. صاف‌شدن منحنی در epochهای بعدی بیانگر کم‌شدن اندازه گرادیان و تثبیت پارامترها است.

```
plt.plot(model.loss_history)
plt.xlabel('Epoch')
plt.ylabel('MSE (standardized scale)')
```

سلول ۸ ـ صفحه رگرسیون سه‌بعدی

برای هر دو ویژگی RM و LSTAT شبکه‌ای ۶۰×۶۰ ساخته می‌شود. شبکه با scaler آموزش استاندارد، توسط مدل پیش‌بینی و سپس به مقیاس اصلی MEDV بازگردانده می‌شود. نقاط آبی داده‌های واقعی test و سطح نارنجی صفحه پیش‌بینی مدل است. فاصله عمودی هر نقطه تا سطح نمایانگر خطای باقیمانده آن نمونه است.

```
RM, LSTAT = np.meshgrid(rm_grid, lstat_grid)
Y_PRED = y_scaler.inverse_transform(model.predict(x_scaler.transform(grid)))
```

۵. مبانی ریاضی مدل

۵ـ۱. مدل خطی

برای n نمونه و دو ویژگی، پیش‌بینی همه نمونه‌ها به‌صورت ماتریسی محاسبه می‌شود. X ماتریس n×2، w بردار ۲×۱ و b اسکالر است:

$$\hat{y} = Xw + b$$

۵ـ۲. تابع هزینه

تابع هزینه میانگین مربع اختلاف مقدار واقعی و پیش‌بینی است. توان دوم خطاهای بزرگ‌تر را شدیدتر جریمه می‌کند و تابعی مشتق‌پذیر برای گرادیان‌کاهشی می‌سازد:

$$MSE = (1/n)\ \Sigma(\hat{y}_i - y_i)^2$$

۵ـ۳. گرادیان و به‌روزرسانی

مشتق MSE نسبت به وزن و بایاس محاسبه و با نرخ یادگیری η از پارامترها کم می‌شود:

$$dw = (2/n)\ X^T(\hat{y} - y)$$
$$db = (2/n)\ \Sigma(\hat{y} - y)$$
$$w \leftarrow w - \eta dw$$
$$b \leftarrow b - \eta db$$

نرخ یادگیری ۰٫۰۵ اندازه هر گام را تعیین می‌کند. نرخ بسیار بزرگ می‌تواند موجب نوسان یا واگرایی و نرخ بسیار کوچک موجب آموزش کند شود. نمودار loss نشان می‌دهد مقدار انتخاب‌شده در این اجرا مناسب بوده است.

۶. نتایج واقعی مدل

معیار یا پارامتر	مقدار
تعداد نمونه آموزش	۴۰۴
تعداد نمونه آزمون	۱۰۲
MSE	۳۱٫۲۴۳
RMSE	۵٫۵۹۰
MAE	۳٫۸۹۹
R^2	۰٫۵۷۴
وزن استاندارد RM	+۰٫۴۱۵۵
وزن استاندارد LSTAT	−۰٫۴۸۱۹
بایاس استاندارد	تقریباً صفر ($۱٫۹ \times ۱۰^{-۱۵}$)

۶ـ۱. تفسیر خطاها

MAE برابر ۳٫۸۹۹ یعنی قدر مطلق خطای پیش‌بینی، به‌طور میانگین حدود ۳٫۹ واحد MEDV است. با توجه به تعریف تاریخی هدف، این مقدار تقریباً ۳٫۹ هزار دلار در مقیاس دیتاست است. RMSE برابر ۵٫۵۹۰ از MAE بزرگ‌تر است؛ این فاصله نشان می‌دهد

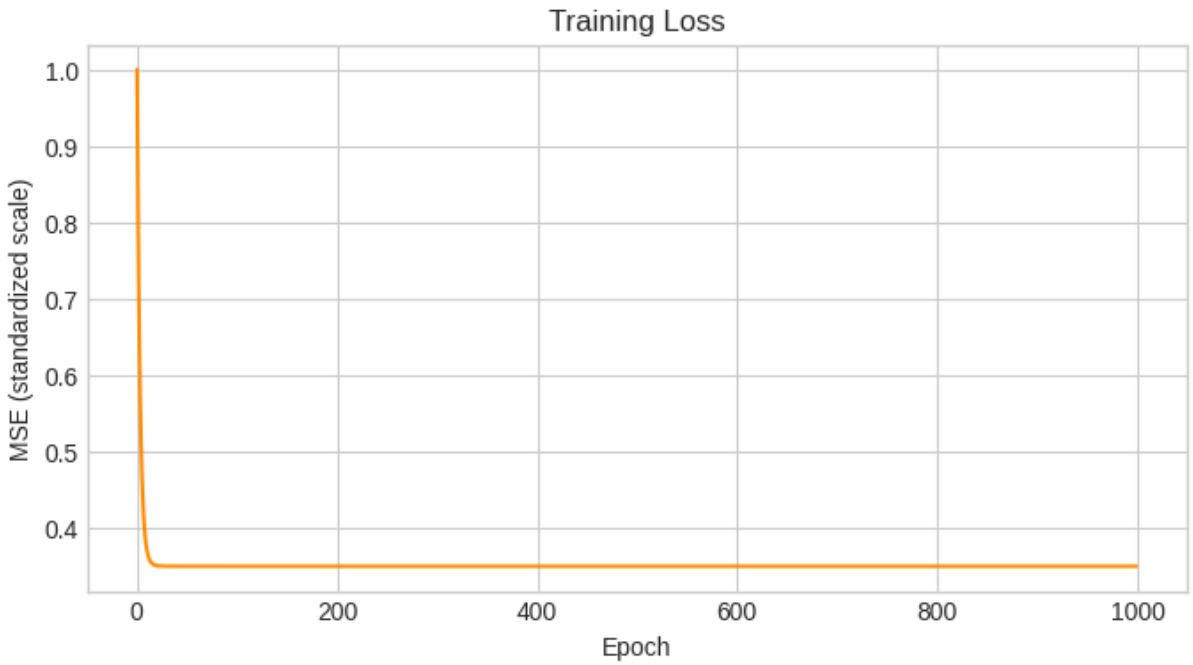
چند نمونه با خطاهای بزرگ‌تر وجود دارند، زیرا RMSE به خطاهای بزرگ حساس‌تر است. MSE برابر ۳۱٫۲۴۳ نیز مربع واحد هدف را دارد و بیشتر برای بهینه‌سازی و مقایسه مدل استفاده می‌شود.

۶-۲. تفسیر R^2

ضریب تعیین ۰٫۵۷۴ نشان می‌دهد مدل خطی دوویژگی حدود ۵۷٫۴ درصد واریانس MEDV در مجموعه آزمون را توضیح می‌دهد. این نتیجه برای یک مدل بسیار ساده قابل قبول است، اما حدود ۴۲٫۶ درصد تغییرات همچنان توضیح داده نشده‌اند. عوامل حذف‌شده، روابط غیرخطی، نقاط پرت و محدودیت فرض صفحه خطی از دلایل این بخش باقیمانده هستند.

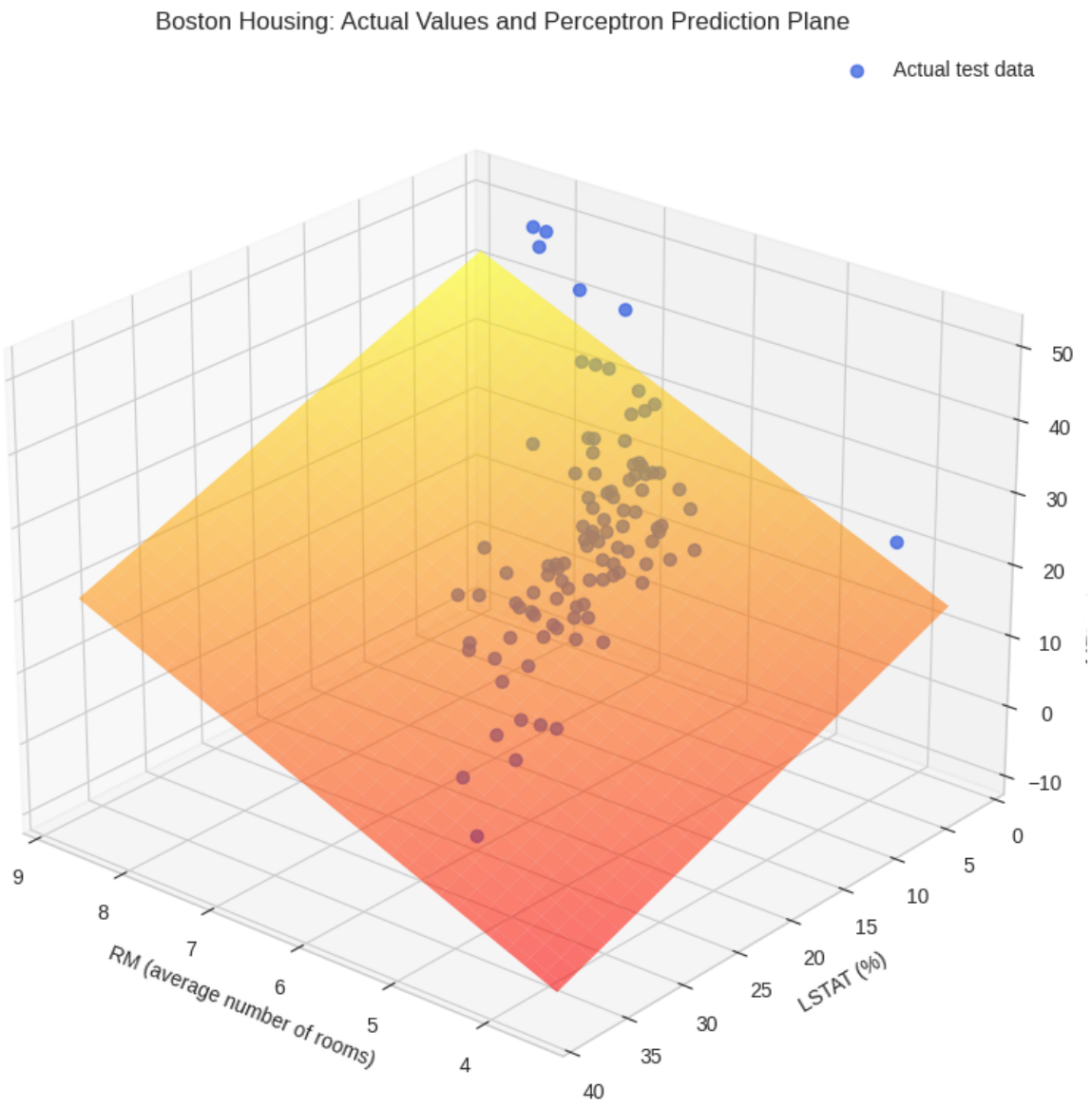
۶-۳. تفسیر وزن‌ها

در مقیاس استاندارد، وزن RM برابر +۰٫۴۱۵۵ است؛ بنابراین با ثابت‌بودن LSTAT، افزایش RM پیش‌بینی قیمت را بالا می‌برد. وزن LSTAT برابر −۰٫۴۸۱۹ است؛ بنابراین با ثابت‌بودن RM، افزایش LSTAT پیش‌بینی قیمت را کاهش می‌دهد. قدر مطلق وزن LSTAT کمی بزرگ‌تر است و در فضای استاندارد اثر خطی قوی‌تری نشان می‌دهد. بایاس تقریباً صفر نتیجه طبیعی مرکزدهی X و y توسط StandardScaler است.



شکل ۱ ـ روند کاهش MSE استانداردشده طی ۱۰۰۰ epoch آموزش

منحنی loss با شیب زیاد در ابتدای آموزش کاهش می‌یابد و سپس به‌سرعت به ناحیه پایدار می‌رسد. نبود نوسان شدید یا افزایش خطا نشان می‌دهد آموزش واگرا نشده است. ادامه‌دادن آموزش پس از همگرایی تغییر محسوسی ایجاد نمی‌کند و می‌توان با معیار توقف زودهنگام تعداد epochها را کاهش داد.



شکل ۲ ـ نقاط واقعی آزمون و صفحه پیش‌بینی پرسپترون رگرسیونی

شیب صعودی سطح در جهت RM و شیب نزولی آن در جهت LSTAT با علامت وزن‌ها مطابقت دارد. قرارنگرفتن همه نقاط روی صفحه نشان می‌دهد رابطه واقعی کاملاً خطی نیست. نقاط دور از صفحه سهم بیشتری در MSE و RMSE دارند.

۷. نقاط قوت پیاده‌سازی

- پیاده‌سازی مدل و گرادیان‌کاهشی از ابتدا و بدون LinearRegression آماده.
- تفکیک صحیح train و test با random_state ثابت.
- برازش scaler فقط روی train و جلوگیری از data leakage.
- استانداردسازی جداگانه ویژگی‌ها و هدف برای آموزش پایدار.
- ارزیابی با چهار معیار مکمل و تفسیرپذیر.
- ثبت تاریخچه loss و نمایش هندسه مدل با صفحه سه‌بعدی.
- تطابق علامت وزن‌ها با همبستگی‌های مشاهده‌شده در داده.

۸. محدودیت‌ها

- مدل فقط از دو ویژگی از ۱۳ ویژگی موجود استفاده می‌کند و بخشی از اطلاعات مفید حذف شده است.
- فرض رابطه خطی قادر به مدل‌کردن خمیدگی و تعامل پیچیده RM و LSTAT نیست.

- فقط یک تقسیم train/test انجام شده و عدم قطعیت معیارها با cross-validation سنجیده نشده است.
- مدل نسبت به نقاط پرت حساس است، زیرا MSE خطاهای بزرگ را با توان دوم جریمه می‌کند.
- تحلیل residual، آزمون نرمال‌بودن خطا و بررسی ناهمسانی واریانس انجام نشده است.
- دیتاست Boston Housing تاریخی، کوچک و برای کاربرد واقعی امروزی یا تصمیم‌گیری اجتماعی مناسب نیست.

۹. پیشنهادهای بهبود و آزمایش‌های بعدی

- آموزش مدل با همه ویژگی‌ها و مقایسه آن با مدل دوویژگی.
- استفاده از Polynomial Features برای ثبت روابط غیرخطی و تعامل RM×LSTAT.
- مقایسه با Ridge و Lasso برای کنترل پیچیدگی و انتخاب ویژگی.
- استفاده از شبکه عصبی چندلایه با لایه‌های ReLU و بررسی بهبود R^2.
- اجرای k-fold cross-validation و گزارش میانگین و انحراف معیار معیارها.
- رسم residual در برابر predicted و شناسایی نقاط پرت یا الگوی سیستماتیک خطا.
- افزودن Early Stopping بر اساس تغییر loss برای کاهش محاسبات غیرضروری.
- استفاده از دیتاست جدیدتر و فاقد محدودیت‌های اخلاقی برای کاربرد واقعی.

۱۰. نتیجه‌گیری

در این تمرین یک پرسپترون رگرسیونی با دو ورودی RM و LSTAT و یک خروجی خطی با موفقیت پیاده‌سازی شد. گرادیان‌کاهشی دسته‌ای پس از استانداردسازی داده پایدار بود و وزن‌های آموخته‌شده با روابط آماری داده سازگار بودند. مدل روی مجموعه آزمون به MAE برابر ۳٫۸۹۹، RMSE برابر ۵٫۵۹۰ و R^2 برابر ۰٫۵۷۴ رسید. این نتیجه نشان می‌دهد دو ویژگی منتخب بخش مهمی از تغییرات قیمت را توضیح می‌دهند، ولی برای دقت بالاتر باید ویژگی‌های بیشتر، مدل غیرخطی، اعتبارسنجی چندبخشی و تحلیل دقیق باقیمانده‌ها به کار گرفته شود. مهم‌ترین دستاورد تمرین، درک عملی رابطه میان مدل خطی، تابع MSE، گرادیان پارامترها، استانداردسازی و تفسیر معیارهای رگرسیون است.

```
)14, 506: (ابعاد دیتاست
404: تعداد نمونه‌های آموزش
102: تعداد نمونه‌های آزمون
MSE  = 31.243
RMSE = 5.590
MAE  = 3.899
R²   = 0.574
Weights = [0.41546965, -0.48191544]
Bias ≈ 0
```