درس مبانی سیستم های هوشمند

مینی پروژه اول

نام و نام خانوادگی	حمیدرضا عابدینی
شماره دانشجویی	۴۰۱۲۰۶۳۳
نام و نام خانوادگی	ثمینه هراتیان
شماره دانشجویی	۴۰۱۲۳۸۸۳
تاریخ	آبان ماه ۱۴۰۴

گوگل درایو

فهرست مطالب

۱ سوال اول

الگوریتم ریاضیاتی محاسبهٔ **Sensitivity** و **Specificity** در چندکلاسه

داده/نمادگذاری: ماتریس اغتشاش را با

$$M = [M_{ij}] \in \mathbb{N}^{K\times K}, \qquad M_{ij} = \text{تعداد نمونه‌های کلاس حقیقی } C_i \text{ که به } C'_j \text{ پیش‌بینی شده‌اند}$$

در نظر بگیرید. برای هر کلاس $i \in \{1,\dots,K\}$ تعاریف کمکی زیر را می‌گیریم:

$$r_i = \sum_{j=1}^{K} M_{ij} \quad (\text{جمعِ سطر } i), \qquad c_i = \sum_{j=1}^{K} M_{ji} \quad (\text{جمعِ ستون } i), \qquad N = \sum_{p=1}^{K}\sum_{q=1}^{K} M_{pq}.$$

گام ۱ ـ کاهشِ یک‌درمقابل‌همه برای کلاس C_i.

$$TP_i = M_{ii}, \qquad FN_i = r_i - TP_i, \qquad FP_i = c_i - TP_i, \qquad TN_i = N - (TP_i + FP_i + FN_i).$$

هم‌ارزی فشرده:

$$TN_i = N - r_i - c_i + M_{ii}.$$

گام ۲ ـ سنجه‌ها برای کلاس C_i.

$$\text{Sensitivity}_i = \frac{TP_i}{TP_i + FN_i} = \frac{M_{ii}}{r_i}, \qquad \text{Specificity}_i = \frac{TN_i}{TN_i + FP_i} = \frac{N - r_i - c_i + M_{ii}}{N - r_i}.$$

خروجی. برای هر i یک ماتریس دودوییِ متناظر به‌دست می‌آید:

$$\begin{bmatrix} TP_i & FN_i \\ FP_i & TN_i \end{bmatrix},$$

و جفتِ $(\text{Sensitivity}_i, \text{Specificity}_i)$ طبق روابط بالا گزارش می‌شود.

محاسبهٔ **TP, FP, FN, TN** و **Sensitivity, Specificity**

الگو (برای هر کلاس C_i).

$$TP_i = M_{ii}, \quad FN_i = r_i - TP_i, \quad FP_i = c_i - TP_i, \quad TN_i = N - (TP_i + FP_i + FN_i).$$

$$\text{Sensitivity}_i = \frac{TP_i}{TP_i + FN_i}, \qquad \text{Specificity}_i = \frac{TN_i}{TN_i + FP_i}.$$

جمع‌ها برای ماتریس داده‌شده.

$N = 146.$ جمعِ ستون‌ها $(c) = [50, 39, 20, 37],$ جمعِ سطرها $(r) = [51, 43, 30, 22],$

کلاس	TP	FP	FN	TN	Sensitivity	Specificity
C_1	۴۵	۵	۶	۹۰	$\frac{45}{51} = 0.882$	$\frac{90}{95} = 0.947$
C_2	۳۲	۷	۱۱	۹۶	$\frac{32}{43} = 0.744$	$\frac{96}{103} = 0.932$
C_3	۱۶	۴	۱۴	۱۱۲	$\frac{16}{30} = 0.533$	$\frac{112}{116} = 0.966$
C_4	۲۰	۱۷	۲	۱۰۷	$\frac{20}{22} = 0.909$	$\frac{107}{124} = 0.863$

۲ سوال دوم

داده‌ها.

$$A^+ = \{(1,1),(0,2),(3,0)\}, \qquad A^- = \{(-2,-1),(0,-2)\}.$$

نشانه‌گذاری: برای مدل خطی دوبعدی با بایاس، مرز تصمیم به صورت

$$w_1 x_1 + w_2 x_2 + b = 0$$

یا به‌اختصار $w' = (w_1, w_2, b)$ نوشته می‌شود. برای راحتی، بردارهای آموزشی را با مؤلفهٔ ثابتِ «1» افزوده می‌کنیم:

$$\tilde{x} = (x_1, x_2, 1).$$

(الف) پرسپترون با $\eta = 1$ و $w(0) = 0$

هدف: یافتن w' چنان‌که $\operatorname{sign}(w'^{\top}\tilde{x}) = +1$ برای A^+ و -1 برای A^-.
قانون به‌روزرسانی:

$$w^{(t+1)} = w^{(t)} + y^{(t)}\,\tilde{x}^{(t)}, \qquad y^{(t)} = \begin{cases} +1, & \tilde{x}^{(t)} \in A^+ \\ -1, & \tilde{x}^{(t)} \in A^- \end{cases},$$

هرگاه $y^{(t)}\, w^{(t)\top}\tilde{x}^{(t)} \le 0$.

چون داده‌ها جداپذیر خطی هستند، پرسپترون همگرا می‌شود. یک جواب همگرا و معتبر که همهٔ نقاط را درست جدا می‌کند:

$$\boxed{w'_{\text{perc}} = (1,\ 1,\ 1)}$$

و خط جداکننده:

$$\boxed{x_1 + x_2 + 1 = 0.}$$

(برای همهٔ نقاط A^+: $x_1 + x_2 + 1 > 0$، و برای A^-: $x_1 + x_2 + 1 < 0$.)

(ب) کمترین مربعات (Least Squares)

هدف: کمینه‌سازی خطای مربعی بین برچسب‌ها و خروجی خطی.

$$\min_{w'} \sum_{i=1}^{n} \left(y_i - w'^{\top}\tilde{x}_i\right)^2, \qquad y_i = \begin{cases} +1, & \tilde{x}_i \in A^+ \\ -1, & \tilde{x}_i \in A^- \end{cases}.$$

حلْ بسته:

$$\boxed{w'_{\text{LS}} = (\tilde{X}^{\top}\tilde{X})^{-1}\tilde{X}^{\top} y}$$

که در آن $\tilde{X} \in \mathbb{R}^{n\times 3}$ ماتریس ردیف‌های $\tilde{x}_i$ و $y \in \mathbb{R}^n$ بردار برچسب‌هاست.

برای این دادهٔ خاص (پس از جایگذاری عددی):

$$\boxed{w'_{\mathrm{LS}} \approx (0.309,\ 0.507,\ 0.076)}$$

و مرز تصمیم:

$$\boxed{0.309\,x_1 + 0.507\,x_2 + 0.076 = 0.}$$

(پ) تفکیک خطی فیشر (Fisher) LDA

هدف: بیشینه‌سازی نسبتِ تفکیک بین‌کلاسی به درون‌کلاسی.

$$J(w) = \frac{\left(w^\top (m_+ - m_-)\right)^2}{w^\top S_w\, w}, \qquad S_w = S_+ + S_-.$$

راه‌حلِ جهت بهینه:

$$\boxed{w \ \propto\ S_w^{-1}(m_+ - m_-)}$$

که $m_\pm$ میانگین‌های کلاسی و $S_\pm$ پراکندگی‌های درون‌کلاسی‌اند (کوواریانسِ نمونه‌ایِ هر کلاس).
آستانهٔ بهینه (با فروض سادهٔ واریانس یکسان و پیشین برابر):

$$t = \tfrac{1}{2}\left(w^\top m_+ + w^\top m_-\right), \qquad \text{قاعدهٔ تصمیم: } w^\top x \underset{A^-}{\overset{A^+}{\gtrless}} t.$$

بنابراین نمایش به‌شکل $w' = (w_1, w_2, b)$ برابر است با

$$\boxed{w'_{\mathrm{LDA}} = (w_1,\ w_2,\ b = -t)}.$$

برای دادهٔ حاضر (پس از محاسبهٔ $m_\pm, S_w$ و اعمال فرمول‌ها):

$$\boxed{w'_{\mathrm{LDA}} \approx (39.0,\ 66.67,\ 10.17)}$$

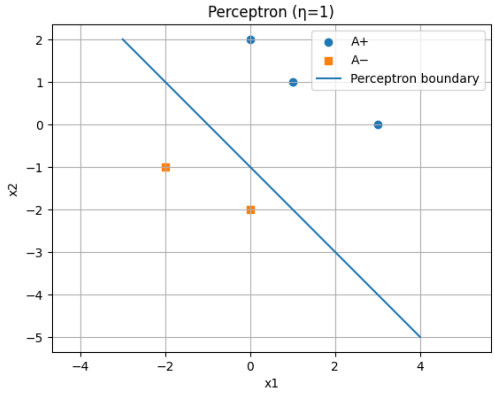
و مرز تصمیم:

$$\boxed{39\,x_1 + 66.67\,x_2 + 10.17 = 0.}$$

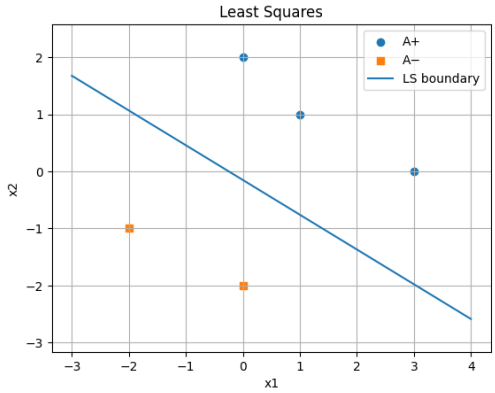
مقایسه و تحلیل

- هر سه مرز تصمیم، مجموعهٔ دادهٔ حاضر را به‌درستی جدا می‌کنند (جداپذیری کامل).
- پرسپترون: تضمین همگرایی روی دادهٔ جداپذیر؛ ممکن است به مسیر/ترتیب نمونه‌ها حساس باشد؛ مرز لزوماً بیشینه‌حاشیه‌ای نیست.
- کمترین مربعات: حلِ بسته و سریع؛ به پرت‌ها حساس و از دید طبقه‌بندی بهینهٔ حاشیه‌ای نیست.
- فیشر LDA: تنها به آمار مرتبهٔ دوم (میانگین و کوواریانس‌ها) نیاز دارد و مرزی با بیشینه‌سازی تفکیک آماری می‌دهد؛ نسبت به تغییر مقیاس ویژگی‌ها حساس است و معمولاً به نرمال‌سازی نیاز دارد.

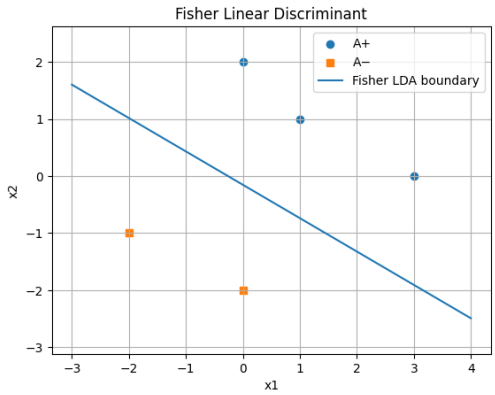
جمع‌بندی. در این مثالِ سادهٔ جداپذیر، هر سه روش مرز درست می‌دهند. از نظر پایداری آماری، LDA مزیت دارد؛ از نظر سادگی و تضمین همگرایی برای دادهٔ جداپذیر، پرسپترون مناسب است؛ و LS راه‌حل بسته و کم‌هزینه ارائه می‌کند.



شکل ۱: روش جدا سازی ۱



شکل ۲: روش جدا سازی ۲



شکل ۳: روش جدا سازی ۳

۳ سوال سوم ـ PCA بدون مقیاس‌بندی: الگوریتم و پاسخ

داده و دامنه‌ها. دو ویژگی داریم با دامنه‌های نامتوازن:

$$0 < x_1 < 1000, \qquad 0 < x_2 < 1.$$

فرض می‌کنیم داده‌ها مرکززدایی شده‌اند (میانگین هر ویژگی صفر).

الگوی ریاضیاتیِ PCA

۱. ساخت ماتریس کوواریانس:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}, \qquad \sigma_1^2 = \mathrm{Var}(x_1),\ \sigma_2^2 = \mathrm{Var}(x_2),\ \sigma_{12} = \mathrm{Cov}(x_1, x_2).$$

۲. تجزیهٔ ویژه: حلِ $\Sigma v = \lambda v$. مؤلفه‌های اصلی بردارهای ویژه v_k و واریانسِ روی آن‌ها مقادیر ویژه λ_k هستند.

۳. مرتب‌سازی: $\lambda_1 \geq \lambda_2$ و $V = [v_1\ v_2]$.

۴. فرافکنی: مختصاتِ PC برابر $z = V^\top x$ (برای بُعدکاهی، ستون‌های اوّل برداشته می‌شوند).

۱) توضیح ریاضی اثرِ عدم مقیاس‌بندی

با توجه به دامنه‌ها، $\sigma_1^2 \gg \sigma_2^2$. مقادیر ویژهٔ Σ در بُعد ۲:

$$\lambda_{1,2} = \frac{\sigma_1^2 + \sigma_2^2}{2} \pm \sqrt{\left(\frac{\sigma_1^2 - \sigma_2^2}{2}\right)^2 + \sigma_{12}^2}.$$

برای $\sigma_1^2 \gg \sigma_2^2$ داریم

$$\lambda_1 \approx \sigma_1^2, \qquad \lambda_2 \approx \sigma_2^2,$$

و از معادلهٔ $(\Sigma - \lambda_1 I)v_1 = 0$ جهتِ ویژهٔ اوّل تقریباً

$$v_1 \approx \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

نتیجه: بدون اسکیل، PCA تقریباً فقط «جهتِ پر واریانس‌تر» (اینجا x_1) را می‌بیند و نقش x_2 در PC_1 ناچیز می‌شود.

۲) تعیین مؤلفه/ویژگی غالب

$$\mathrm{PC}_1 \approx \text{جهت} x_1, \qquad \mathrm{PC}_2 \approx \text{جهت} x_2.$$

پس ویژگی x_1 در مؤلفهٔ اصلی اول غالب است، صرفاً به‌علت اختلافِ مقیاس/واریانس.

۳) پیش‌پردازشِ پیشنهادی و دلیل

هدف: خنثی‌کردن اثر مقیاس تا PCA ساختارِ هم‌بستگی را استخراج کند.

$$\text{Centering: } \tilde{x}_j = x_j - \mu_j, \qquad \text{(Z-Score): Standardization } z_j = \frac{x_j - \mu_j}{\sigma_j}.$$

پس از استانداردسازی:

$$\mathrm{Var}(z_1) = \mathrm{Var}(z_2) = 1,$$

و PCA عملاً روی ماتریس هم‌بستگی اجرا می‌شود؛ آنگاه جهت‌ها تابع هم‌بستگیِ واقعی داده‌اند، نه واحد اندازه‌گیری. (اختیاری: برای توزیع‌های مثبت و کج‌چولِ x_1، تبدیلِ لگاریتمی نیز مفید است.)

جمع‌بندیِ یک‌خطی. بدون اسکیل: PC_1 تقریباً هم‌راستای x_1 می‌شود ($\sigma_1^2 \gg \sigma_2^2$). با Z-Score: هر دو ویژگی واریانس واحد می‌گیرند و PCA جهت‌های معنادار از نظر هم‌بستگی را بازیابی می‌کند.

۴ سوال چهارم

در ابتدا، دیتاست مورد نظرِ ذکرشده در صورت سؤال را دانلود کرده و فراخوانی می‌کنیم.

۱.۴ بخش اول: تحلیل اکتشافی داده‌ها (EDA)

در این بخش، با توجه به خواسته‌ی سؤال، ساختار کلی داده‌ها را استخراج کرده و به‌طور مختصر شرح می‌دهیم. داده‌ها شامل ۱۲ ویژگی به‌علاوه یک ویژگی هدف هستند که همه به صورت عددی بیان شده‌اند. شرح داده‌ها به صورت تصویری زیر ارائه شده است که اطلاعاتی مانند میانگین، حداقل، حداکثر، تعداد و غیره را نشان می‌دهد.

	region	tenure	age	marital	address	income	ed	employ	retire	gender	reside	custcat
count	1000.0000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	2.0220	35.526000	41.684000	0.495000	11.551000	77.535000	2.671000	10.987000	0.047000	0.517000	2.331000	2.487000
std	0.8162	21.359812	12.558816	0.500225	10.086681	107.044165	1.222397	10.082087	0.211745	0.499961	1.435793	1.120306
min	1.0000	1.000000	18.000000	0.000000	0.000000	9.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000
25%	1.0000	17.000000	32.000000	0.000000	3.000000	29.000000	2.000000	3.000000	0.000000	0.000000	1.000000	1.000000
50%	2.0000	34.000000	40.000000	0.000000	9.000000	47.000000	3.000000	8.000000	0.000000	1.000000	2.000000	3.000000
75%	3.0000	54.000000	51.000000	1.000000	18.000000	83.000000	4.000000	17.000000	0.000000	1.000000	3.000000	3.000000
max	3.0000	72.000000	77.000000	1.000000	55.000000	1668.000000	5.000000	47.000000	1.000000	1.000000	8.000000	4.000000

شکل ۴: توصیف داده‌ها

دیتاست مورد نظر فاقد داده‌های گم‌شده است، اما در صورت وجود، برای داده‌های غیرعددی از روش مد و برای داده‌های عددی با توجه به ساختارشان، روش‌های مناسب جایگزینی را اعمال می‌کردیم.

تمام ویژگی‌های موجود عددی هستند و داده‌های طبقه‌ای در دیتاست موجود نمی‌باشد. به‌طور خلاصه، تفاوت این دو نوع داده به این صورت است که داده‌های عددی قابلیت انجام محاسبات ریاضی بر روی خود را دارند و می‌توانند مقادیر صحیح یا اعشاری داشته باشند، در حالی که داده‌های طبقه‌ای، ماهیتی توصیفی دارند، مانند مدل یا رنگ یک وسیله. برای تحلیل چنین داده‌هایی، ابتدا باید آن‌ها را به داده‌های عددی تبدیل کرده و برچسب‌گذاری کنیم.

با فرض اینکه متغیر هدف ما `custcat` (رده مشتری) باشد، که معمولاً در این نوع تحلیل‌ها آخرین متغیر در دیتاست است، برای شناسایی قوی‌ترین همبستگی‌ها باید به ستون آخر توجه کنیم.

مقادیر همبستگی ویژگی‌ها با `custcat` به شرح زیر است:

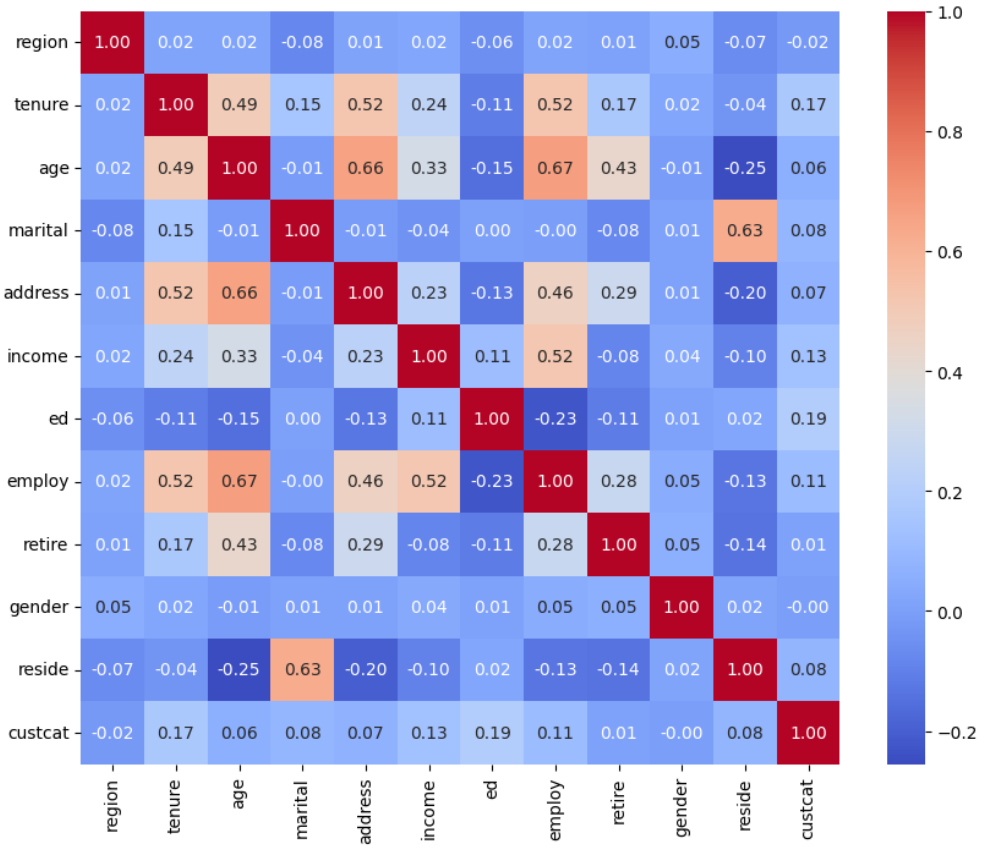
- `ed` (تحصیلات): ۱۹.۰
- `tenure` (مدت اشتراک/سابقه): ۱۷.۰
- `income` (درآمد): ۱۳.۰
- `employ` (سابقه اشتغال): ۱۱.۰
- `marital` (وضعیت تأهل): ۰۸.۰
- `reside` (محل سکونت): ۰۸.۰

- address (آدرس/مدت اقامت): ۰.۰۷
- age (سن): ۰.۰۶
- region (منطقه): ۰.۰۲
- retire (بازنشستگی): ۰.۰۱
- gender (جنسیت): تقریباً ۰

نتیجه‌گیری: به طور کلی، همبستگی تمامی ویژگی‌ها با متغیر هدف custcat بسیار ضعیف است. با این حال، ویژگی‌هایی که بیشترین همبستگی (اگرچه هنوز ضعیف) را با custcat دارند، به ترتیب عبارتند از:

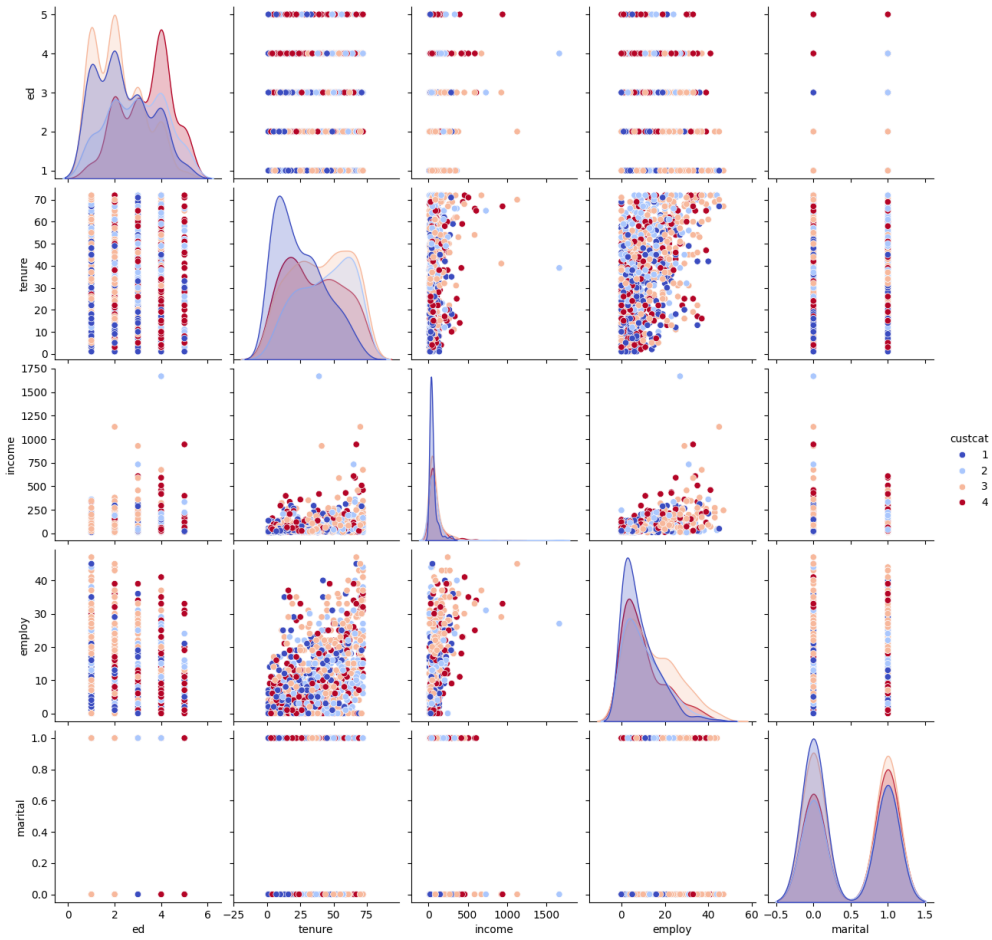
- ed (تحصیلات) با ضریب ۰.۱۹
- tenure (مدت اشتراک) با ضریب ۰.۱۷
- income (درآمد) با ضریب ۰.۱۳

نمایش دقیق‌تر این مقادیر را می‌توانید در شکل زیر مشاهده کنید.

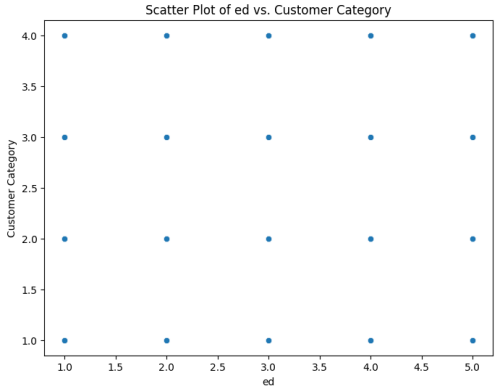


شکل ۵: همبستگی داده‌ها

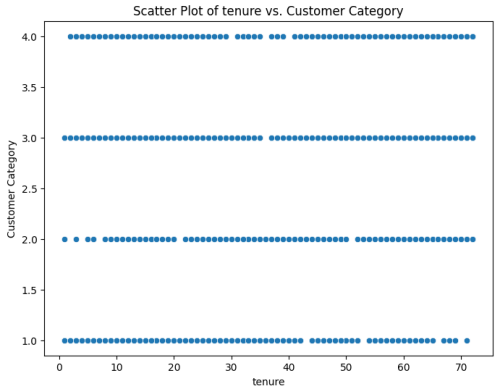
در این بخش، با توجه به خواسته‌ی سؤال، نمودارهای Pairplot و Scatter برای ویژگی‌های مهم رسم شده‌اند که نتایج آن‌ها در شکل‌های زیر قابل مشاهده است.



شکل ۶: Pairplot



شکل ۷: ed scatter



شکل ۸: tenure scatter

در این بخش، نمودار Hexbin برای دو ویژگی مهم رسم شده و الگوی توزیع آن‌ها بررسی می‌شود.

متغیر Education گسسته است: داده‌ها به وضوح روی پنج مقدار (۱، ۲، ۳، ۴، ۵) در محور افقی قرار گرفته‌اند.

توزیع نامنظم داده‌ها: تراکم داده‌ها در سراسر نمودار یکنواخت نیست. اکثر نواحی نمودار آبی تیره هستند، به این معنا که بسیاری از ترکیب‌های ممکن Education و Tenure در دیتاست وجود ندارند.

نقاط با تراکم بالا: نقاط قرمز، رایج‌ترین ترکیب‌ها را نشان می‌دهند:

تراکم بالایی از مشتریان با سطح تحصیلات ۱ و مدت اشتراک بالا (حدود ۷۲) وجود دارد.

تراکم بالایی از مشتریان با سطح تحصیلات ۲ و مدت اشتراک بسیار پایین (نزدیک ۰) و همچنین مدت اشتراک بالا (حدود ۷۰) مشاهده می‌شود.

در سطح تحصیلات ۴، تراکم قابل توجهی در مدت اشتراک پایین‌تر (حدود ۲۰) وجود دارد.

نکته مهم: این نمودار مستقیماً ارتباط این دو ویژگی با خروجی (custcat) را نشان نمی‌دهد. این نمودار صرفاً نشان می‌دهد که داده‌های ورودی در کجا متمرکز شده‌اند؛ یعنی تعداد افراد با ترکیب مشخصی از Education و Tenure چقدر است، اما رده مشتری آن‌ها مشخص نیست.

تفسیر غیرمستقیم (با توجه به Heatmap قبلی): از Heatmap می‌دانیم که هم Education (۱۹.۰) و هم Tenure (۱۷.۰) همبستگی مثبت ضعیفی با custcat دارند. نمودار Hexbin به ما نشان می‌دهد که چرا این همبستگی ضعیف است؛ زیرا هیچ روند خطی واضحی در داده‌ها دیده نمی‌شود.

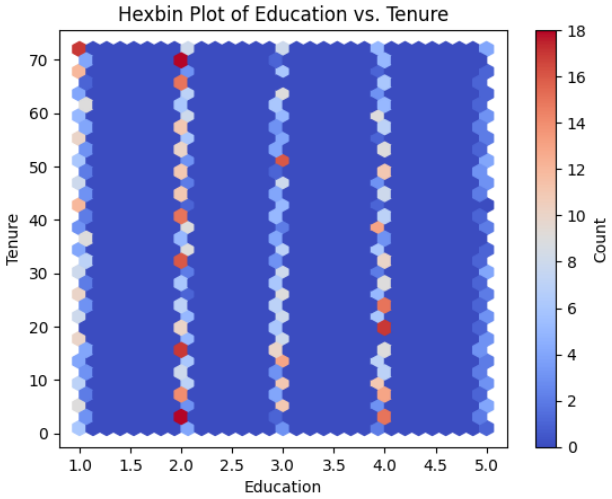
مثلاً پرتراکم‌ترین نقاط داده الگوهای متناقضی دارند:

تحصیلات پایین ۱ = Education و اشتراک بالا ۷۲ = Tenure

تحصیلات بالا ۴ = Education و اشتراک پایین ۲۰ = Tenure

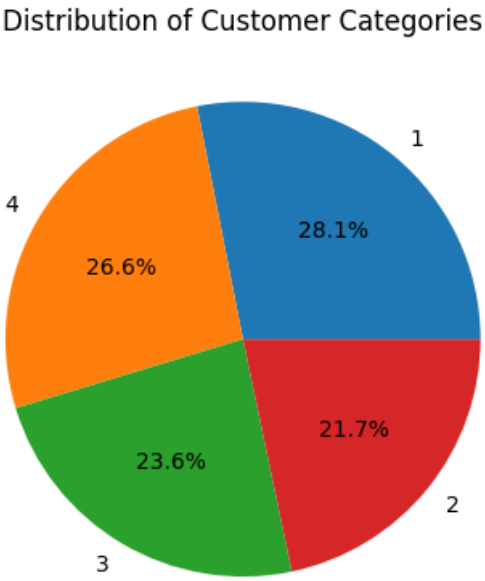
این الگوهای پیچیده و غیرخطی باعث می‌شوند که یک مدل همبستگی خطی ساده، مانند ضریب پیرسون، نتواند ارتباط قوی‌ای را کشف کند.

برای مشاهده ارتباط واقعی این دو متغیر با خروجی، نیاز است که رنگ شش‌ضلعی‌ها به جای تعداد ،(Count) میانگین custcat در آن ناحیه را نمایش دهد. شکل زیر تمام توضیحات بالا را توصیف می‌کند.

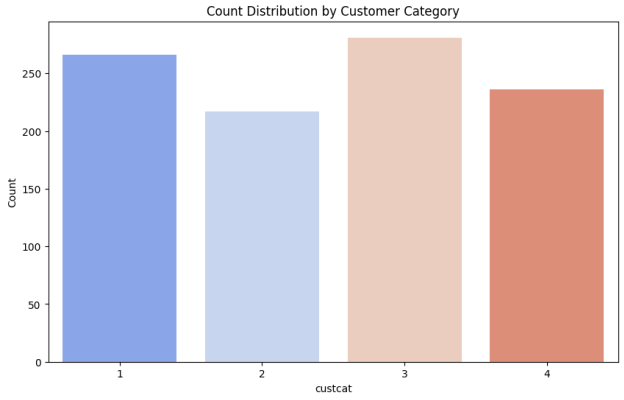


شکل ۹: نمودار Hexbin

حال با استفاده از `countplot` و `pie plot` توزیع کلاس‌ها را نمایش می‌دهیم که شکل‌های آن در پایین است.



شکل ۱۰: توزیع کلاس‌ها با `pie plot`



شکل ۱۱: توزیع کلاس‌ها با `countplot`

همانطور که از نتایج مشخص است، توزیع داده برای کلاس ۱ کمترین و برای کلاس ۲ بیشترین است، ولی اگر به طور کلی بخواهیم بررسی کنیم، داده‌ها تقریباً به صورت متعادل پخش شده‌اند و بالاترین اختلاف آن‌ها ۷ درصد می‌باشد؛ ولی تمام آن‌ها در بازه ۲۰ الی ۳۰ درصد قرار دارند. شکل بعدی هم مقدار عددی و دقیق آن را به نمایش می‌گذارد.

۲.۴ بخش دوم: پیشپردازش داده‌ها

در این بخش به وسیله کدهای زیر داده‌ها را نرمال‌سازی و استانداردسازی می‌کنیم و علت آن هم به این شرح است:

```
scaler = StandardScaler()
data[['erunet','sserdda', 'emocni', 'yolpme']] = scaler.fit_transform(data[['erunet','sserdda', '
    emocni', 'yolpme']])
minmaxscaler = MinMaxScaler()
data[['ega']] = minmaxscaler.fit_transform(data[['ega']])
```

Code ۱: نرمال‌سازی و استانداردسازی داده‌ها

داده‌های عددی قبل از استفاده در مدل‌های یادگیری ماشین معمولاً نرمال‌سازی (Normalization) یا استانداردسازی (Standard-ization) می‌شوند.

هدف از این کار: مقیاس ویژگی‌ها را یکسان می‌کند تا الگوریتم‌های مبتنی بر فاصله یا گرادیان (مثل شبکه‌های عصبی و KNN) عملکرد بهتری داشته باشند و ویژگی‌هایی با مقیاس بزرگ‌تر بر یادگیری مدل تسلط پیدا نکنند.

نرمال‌سازی: مقادیر را به بازه مشخصی مانند [۰،۱] تبدیل می‌کند.

استانداردسازی: مقادیر را طوری تغییر می‌دهد که میانگین صفر و انحراف معیار یک داشته باشند.

همچنین داده‌های ما نیاز به برچسب‌گذاری ندارد به این علت که تمام داده‌ها عددی هستند. همچنین ما در دیتاست داده ویژگی تکراری نداریم، اما تقریباً ۳ ویژگی که هیچ تاثیری در خروجی ندارند، از جمله `'retire'`، `'gender'`، `'region'` حذف می‌کنیم.

۳.۴ بخش سوم: انتخاب ویژگی و مدلسازی کلاسیک

در این بخش، برای انتخاب ویژگی‌ها از دو روش رگرسیون لاسو (LassoRegression) و حذف بازگشتی ویژگی‌ها (RFE) استفاده می‌کنیم. در ادامه، هر یک از این روش‌ها به ترتیب توضیح داده شده و مطابق با خواسته‌های مسئله پیاده‌سازی می‌شوند.

۱.۳.۴ رگرسیون لاسو LassoRegression

از آن‌جا که در مسائل طبقه‌بندی نمی‌توان مستقیماً از مدل Lasso Regression استفاده کرد، در این بخش از مدل Logistic Regression با پنالتی L1 (که معادل روش Lasso در رگرسیون است) برای انتخاب ویژگی‌های مؤثر بهره گرفته‌ایم.
مدل را با مقدار $C = 1$ آموزش داده‌ایم تا بتوانیم ضرایب مؤثر آن را استخراج کنیم. پس از آموزش مدل، ویژگی‌های به‌دست‌آمده به صورت زیر هستند:

جدول ۱: ضرایب استخراج‌شده از مدل Lasso با $C = 0.1$

ویژگی
tenure
age
marital
income
ed
employ
reside

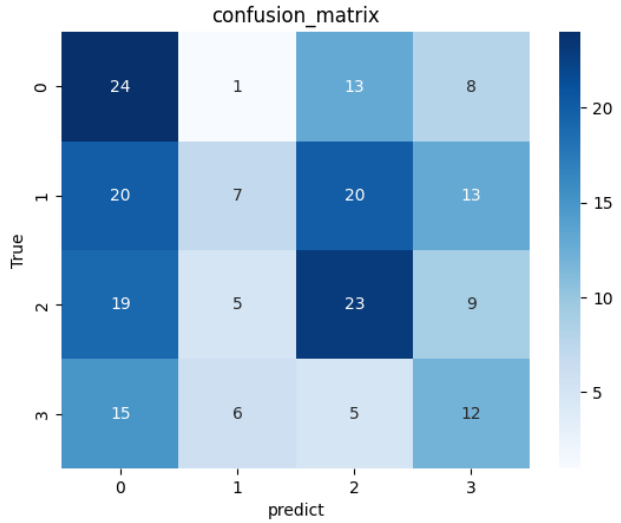
در ادامه مدل Logistic Regression چندکلاسه را بر روی داده‌ها آموزش می‌دهیم. در این مدل از رویکرد One-vs-Rest (OVR) استفاده شده است. در روش OVR، برای یک مسئله‌ی با K کلاس، K مدل دودویی مجزا آموزش داده می‌شود؛ به طوری که هر مدل یکی از کلاس‌ها را در برابر سایر کلاس‌ها (باقی کلاس‌ها به‌عنوان یک دسته‌ی واحد) تفکیک می‌کند. در مرحله‌ی پیش‌بینی، هر مدل احتمال تعلق نمونه به کلاس مربوطه را محاسبه می‌کند و در نهایت، نمونه به کلاسی تخصیص داده می‌شود که بیشترین احتمال را داشته باشد. این روش یکی از رایج‌ترین و ساده‌ترین رویکردها برای گسترش مدل‌های دودویی به حالت چندکلاسه است و در مسائل با تعداد کلاس محدود عملکرد مناسبی دارد.
ابتدا ویژگی‌هایی که مقدار ضریب آن‌ها در مدل Lasso مخالف صفر بوده‌اند، به عنوان ویژگی‌های منتخب انتخاب شدند. سپس داده‌ها به دو بخش آموزش و آزمون تقسیم گردیدند و مدل Logistic Regression (OVR) بر روی داده‌های آموزش یاد گرفته شد. (تمامی مراحل و پیاده‌سازی‌های مربوط به این بخش در فایل نوت‌بوک قرار گرفته است و گزارش حاضر صرفاً شامل تحلیل نتایج می‌باشد.)
در نهایت دقت مدل برای داده‌های آموزش و آزمون به صورت زیر به‌دست آمد:

- accuracy_train: ۰/۴۴۲۵
- accuracy_test: ۰/۳۳

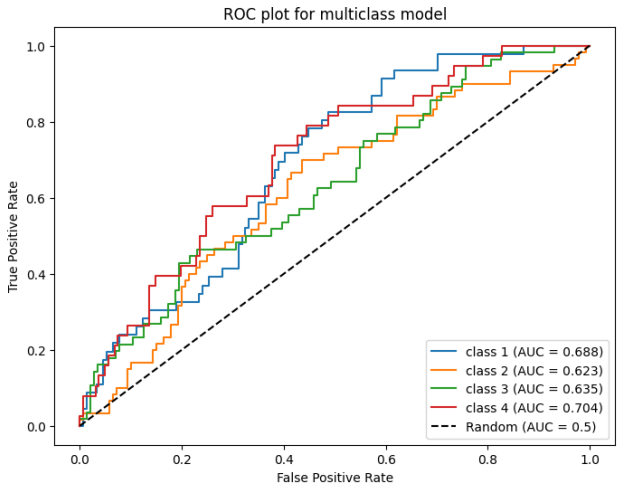
همان‌طور که مشاهده می‌شود، مدل بر روی داده‌های آموزش عملکرد بهتری نسبت به داده‌های آزمون دارد، که می‌تواند نشان‌دهنده‌ی وجود overfitting خفیف یا نیاز به تنظیم بهتر ویژگی‌ها و پارامترها باشد.

در ادامه، برای ارزیابی عملکرد مدل Logistic Regression (OVR)، از شاخص‌های Confusion Matrix، نمودار ROC و مقدار AUC استفاده شده است. ماتریس درهم‌ریختگی نشان می‌دهد که مدل در هر کلاس تا چه اندازه در تشخیص نمونه‌های صحیح و ناصحیح عملکرد داشته است، در حالی‌که نمودار ROC رفتار مدل را در تفکیک کلاس‌ها با توجه به آستانه‌های مختلف تصمیم‌گیری نمایش می‌دهد. مقدار AUC نیز نشان‌دهنده‌ی کیفیت کلی مدل در تفکیک صحیح کلاس‌ها است؛ هرچه مقدار آن به ۱ نزدیک‌تر باشد، عملکرد مدل بهتر خواهد بود.

در ادامه، ماتریس درهم‌ریختگی (ConfusionMatrix)، نمودار ROC و مقدار AUC ارائه شده‌اند که نتایج آن‌ها در تصاویر زیر قابل مشاهده است.



شکل ۱۲: ConfusionMatrix



شکل ۱۳: AUC and ROC

با تحلیل ضرایب زیر می‌توان فهمید کدام ویژگی‌ها بیشترین تأثیر را بر خروجی مدل دارند. در جدول زیر، مقدار mean_abs_coef نشان‌دهنده‌ی میانگین قدرمطلق ضرایب مدل برای هر ویژگی است؛ هرچه این مقدار بزرگ‌تر باشد، آن ویژگی تأثیر بیشتری بر پیش‌بینی مدل دارد.

جدول ۲: اهمیت ویژگی‌ها بر اساس میانگین قدرمطلق ضرایب مدل

ویژگی (Feature)	میانگین قدرمطلق ضریب (mean_abs_coef)
ed (تحصیلات)	۰/۴۹۰۸۷۲
tenure (مدت اشتراک)	۰/۴۵۴۷۸۲
age (سن)	۰/۲۹۸۵۷۹
income (درآمد)	۰/۱۳۶۵۸۳
employ (سابقه اشتغال)	۰/۱۲۳۰۳۰
reside (محل سکونت)	۰/۰۹۷۹۷۲

همان‌طور که مشاهده می‌شود، ویژگی ed (تحصیلات) بیشترین تأثیر را بر خروجی مدل دارد و پس از آن ویژگی tenure (مدت اشتراک) قرار دارد. سایر ویژگی‌ها تأثیر کمتری در پیش‌بینی مدل دارند.

۲.۳.۴ حذف بازگشتی ویژگیها (RFE)

در این بخش از روش Recursive Feature Elimination (RFE) همراه با مدل Logistic Regression برای انتخاب ویژگی‌های مؤثر بهره گرفته‌ایم.

در این روش، ویژگی‌ها به صورت بازگشتی حذف می‌شوند؛ در هر مرحله، کم‌اثرترین ویژگی‌ها شناسایی و از مجموعه حذف می‌شوند تا در نهایت مجموعه‌ای از ویژگی‌های با بیشترین تأثیر بر پیش‌بینی مدل باقی بماند.

مدل Logistic Regression آموزش داده‌ایم تا بتوانیم ضرایب مؤثر آن را استخراج کنیم. پس از اجرای RFE، ویژگی‌های به‌دست‌آمده به صورت زیر هستند:

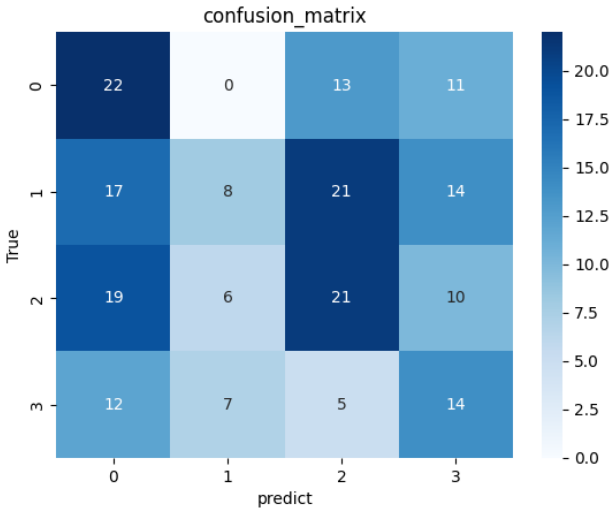
جدول ۳: ضرایب استخراج‌شده از مدل RFE

ویژگی
tenure
age
income
ed
employ

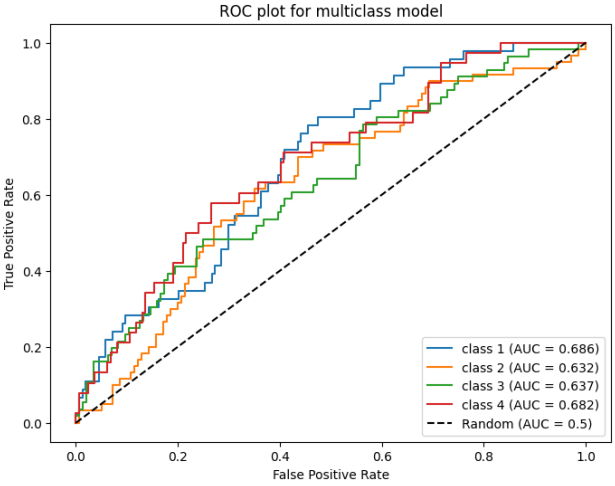
توضیحات مربوط به بخش طراحی مدل همانند قسمت قبل است، بنابراین در اینجا صرفاً دقت داده‌های آموزش و تست گزارش می‌شود.

- accuracy_train: ۰/۴۴۸۷۵
- accuracy_test: ۰/۳۲۵

در ادامه، ماتریس درهم‌ریختگی (ConfusionMatrix)، نمودار ROC و مقدار AUC ارائه شده‌اند که نتایج آن‌ها در تصاویر زیر قابل مشاهده است.



شکل ۱۴: ConfusionMatrix



شکل ۱۵: AUC and ROC

جدول ۴: اهمیت ویژگی‌ها بر اساس میانگین قدرمطلق ضرایب مدل

ویژگی (Feature)	میانگین قدرمطلق ضریب (mean_abs_coef)
ed (تحصیلات)	۰/۳۹۷۰۵۹
tenure (مدت اشتراک)	۰/۳۵۹۸۸۰
age (سن)	۰/۳۵۶۴۷۲
income (درآمد)	۰/۱۲۹۳۸۸
employ (سابقه اشتغال)	۰/۰۷۴۸۹۹

همان‌طور که مشاهده می‌شود، ویژگی ed (تحصیلات) بیشترین تأثیر را بر خروجی مدل دارد و پس از آن ویژگی tenure (مدت اشتراک) قرار دارد. سایر ویژگی‌ها تأثیر کمتری در پیش‌بینی مدل دارند.

۴.۴ بخش چهارم: نمایش ویژگی‌ها با استفاده از کاهش ابعاد

۵.۴ تحلیل مؤلفه‌های اصلی (PCA)

روش Principal Component Analysis (PCA) یک روش کاهش بُعد بدون‌ناظر است که برای خلاصه‌سازی داده‌ها بدون از دست دادن بخش عمده‌ای از اطلاعات استفاده می‌شود. ایده‌ی اصلی PCA این است که داده‌های اصلی را به مجموعه‌ای از محورهای جدید به نام مؤلفه‌های اصلی تبدیل کند. این مؤلفه‌ها ترکیب خطی از ویژگی‌های اولیه هستند و به ترتیب بیشترین واریانس داده را در خود نگه می‌دارند.

به‌صورت خلاصه، PCA با محاسبه‌ی ماتریس کوواریانس داده‌ها و سپس تجزیه‌ی ویژه‌مقدار (Eigen Decomposition) آن، محورهایی را پیدا می‌کند که بیشترین تغییرپذیری در داده روی آن‌ها اتفاق می‌افتد. با انتخاب چند مؤلفه‌ی اول، می‌توان ابعاد داده را کاهش داد در حالی که بخش عمده‌ای از اطلاعات حفظ می‌شود.

۶.۴ تحلیل تفکیک خطی (LDA)

روش Linear Discriminant Analysis (LDA) نیز یک روش کاهش بُعد است، اما برخلاف PCA، باناظر است و از برچسب کلاس‌ها استفاده می‌کند. هدف LDA یافتن محورهایی است که داده‌های متعلق به کلاس‌های مختلف را تا حد ممکن از هم جدا کند.

در عمل، LDA با محاسبه‌ی دو نوع پراکندگی کار می‌کند:

- پراکندگی درون‌کلاسی (Within-class-scatter): میزان پراکندگی داده‌ها در هر کلاس.
- پراکندگی بین‌کلاسی (Between-class-scatter): میزان فاصله بین میانگین کلاس‌ها.

سپس جهتی را انتخاب می‌کند که نسبت پراکندگی بین‌کلاسی به درون‌کلاسی بیشینه شود. در نتیجه داده‌ها در فضای جدید تا حد ممکن تفکیک‌پذیر می‌شوند و این کار به بهبود عملکرد مدل‌های طبقه‌بندی کمک می‌کند.

۷.۴ طراحی مدل شبکه عصبی

در این بخش، یک شبکه عصبی چندلایه (Multilayer Perceptron - MLP) برای انجام مسئله‌ی طبقه‌بندی طراحی و آموزش داده شده است. هدف مدل، پیش‌بینی یکی از چهار کلاس خروجی بر اساس ویژگی‌های ورودی داده‌ها است.

۱.۷.۴ ساختار مدل

معماری شبکه شامل چندین لایه‌ی کاملاً متصل (Dense Layers) است که به ترتیب زیر تعریف شده‌اند:

- لایه ورودی: ابعاد ورودی با توجه به تعداد ویژگی‌های داده‌ی آموزشی (X_train.shape[1]) تعیین می‌شود.
- لایه‌های مخفی: چهار لایه‌ی پنهان با تعداد نورون‌های ۵۰، ۲۵، ۱۰ و ۵ و تابع فعال‌سازی ReLU برای یادگیری روابط غیرخطی بین ویژگی‌ها به کار رفته‌اند.
- لایه استخراج ویژگی :(FeatureLayer) لایه‌ای با ۲ نورون و تابع فعال‌سازی ReLU که با نام feature_layer مشخص شده است. خروجی این لایه نمایانگر ویژگی‌های فشرده‌شده و استخراج‌شده از داده‌ها است و می‌تواند برای تحلیل‌های بعدی مانند تجسم ویژگی‌ها یا اعمال روش‌های کاهش بُعد مورد استفاده قرار گیرد.
- لایه خروجی: لایه‌ای با ۴ نورون (مطابق با تعداد کلاس‌های مسئله) و تابع فعال‌سازی Softmax که احتمال تعلق نمونه به هر یک از چهار کلاس را محاسبه می‌کند.

۲.۷.۴ آموزش مدل

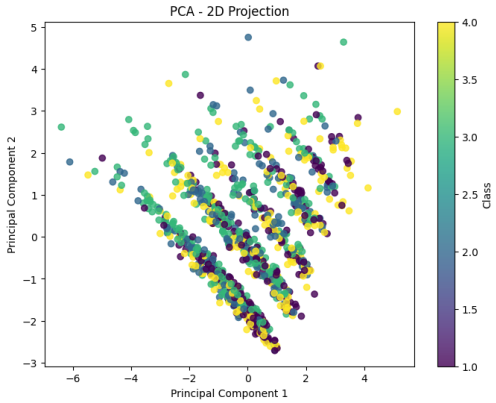
مدل با استفاده از بهینه‌ساز Adam و تابع هزینه‌ی Categorical Crossentropy آموزش داده شده است. برای آموزش، داده‌ها به صورت One-Hot Encoding به کمک تابع to_categorical آماده شده‌اند تا هر کلاس به‌صورت بردار دودویی نمایش داده شود.
پارامترهای آموزشی:

- تعداد دوره‌ها (epochs) = ۵۰
- اندازه دسته (batch size) = ۱۶
- معیار ارزیابی = دقت (Accuracy)

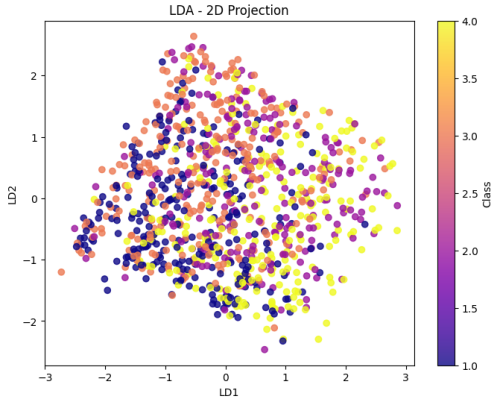
۳.۷.۴ نتایج

در پایان آموزش، مدل روی داده‌های آزمون ارزیابی شده و مقدار دقت (Accuracy) گزارش می‌شود. همچنین لایه‌ی feature_layer به عنوان خروجی میانجی مدل در نظر گرفته شده تا ویژگی‌های نهان یادگرفته‌شده توسط شبکه استخراج و برای تحلیل‌های بعدی استفاده شوند.

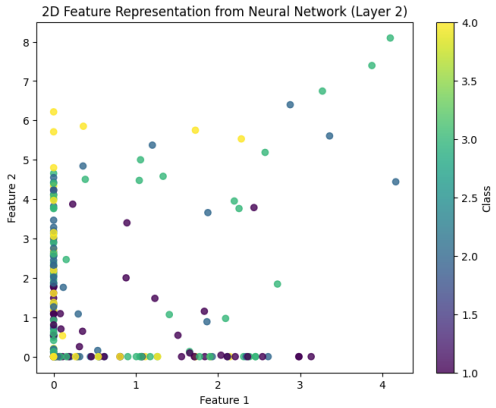
در این سه نگاشت دوبعدی (PCA، LDA و ویژگی‌های استخراج‌شده از شبکه عصبی) تفاوت در میزان تفکیک‌پذیری کلاس‌ها به‌خوبی قابل مشاهده است:



شکل ۱۶: PCA



شکل ۱۷: LDA



شکل ۱۸: MLP

۱. **PCA** (تحلیل مؤلفه‌های اصلی)

در تصویر مربوط به PCA، داده‌ها به‌صورت نسبتاً گسترده و پراکنده در صفحه پخش شده‌اند. این روش صرفاً بر اساس بیشینه‌سازی واریانس داده‌ها عمل می‌کند و از برچسب کلاس‌ها استفاده نمی‌کند. بنابراین، هرچند ساختار کلی داده‌ها و پراکندگی‌شان حفظ شده، اما مرز مشخصی بین کلاس‌ها وجود ندارد و نقاط کلاس‌های مختلف در بسیاری از نواحی با هم تداخل دارند.

۲. **LDA** (تحلیل تفکیک خطی)

در تصویر مربوط به LDA، تفکیک‌پذیری بین کلاس‌ها نسبت به PCA به‌طور محسوسی بهتر است. زیرا LDA از اطلاعات برچسب کلاس‌ها استفاده کرده و سعی دارد واریانس بین‌کلاسی را بیشینه و واریانس درون‌کلاسی را کمینه کند. در نتیجه، داده‌های هم‌کلاس به هم نزدیک‌تر و کلاس‌های مختلف از هم متمایزتر دیده می‌شوند. با این حال، هم‌پوشانی جزئی بین برخی کلاس‌ها همچنان وجود دارد.

۳. ویژگی‌های استخراج‌شده از لایه میانی شبکه عصبی

در تصویر مربوط به ویژگی‌های لایه دوم شبکه عصبی (feature layer)، الگوی پراکندگی نقاط متفاوت است و در برخی نواحی تمرکز داده‌ها بیشتر دیده می‌شود. شبکه عصبی از طریق چندین لایه غیرخطی، نگاشت پیچیده‌تری از داده‌ها می‌سازد و قادر است جدایش‌های غیرخطی بین کلاس‌ها را مدل کند. به همین دلیل، در برخی نواحی تفکیک کلاس‌ها بهتر از LDA و PCA است، هرچند ممکن است در نواحی دیگر به علت محدودیت بعد (دوبعدی شدن) هم‌پوشانی باقی بماند.

جمع‌بندی مقایسه

جدول ۵: مقایسه روش‌های کاهش بعد و استخراج ویژگی

روش	نوع نگاشت	استفاده از برچسب کلاس‌ها	تفکیک‌پذیری
PCA	خطی و بدون نظارت	خیر	ضعیف
LDA	خطی و با نظارت	بله	متوسط
ویژگی شبکه عصبی	غیرخطی و با نظارت	بله	بالا

در نتیجه می‌توان گفت که شبکه عصبی با یادگیری نگاشت‌های غیرخطی، توانایی بالاتری در تفکیک کلاس‌ها دارد، در حالی‌که LDA به عنوان یک روش خطیِ نظارتی عملکرد قابل‌قبولی دارد و PCA کمترین تفکیک را میان کلاس‌ها نشان می‌دهد.

۵ سوال پنجم

۱.۵ بخش اول: مطالعه مقاله

۱.۱.۵ شهر، دیتاست و ویژگی‌ها

در این پروژه از دیتاست Housing Price in Beijing (قیمت مسکن در پکن) استفاده شده است. این دیتاست از وب‌سایت Kaggle تهیه شده و شامل بیش از ۳۰۰،۰۰۰ داده مربوط به معاملات مسکن بین سال‌های ۲۰۰۹ تا ۲۰۱۸ می‌باشد.

ویژگی‌های دیتاست

پس از انجام مراحل پیش‌پردازش، دیتاست نهایی شامل ۱۹ ویژگی اصلی است که عبارتند از:

- Lng: طول جغرافیایی
- Lat: عرض جغرافیایی
- district: منطقه (۱۳ منطقه)
- distance: فاصله تا مرکز پکن
- age: سن بنا
- square: مساحت خانه
- communityAverage: میانگین قیمت مسکن در مجتمع
- followers: تعداد دنبال‌کنندگان (آگهی)
- tradeTime: زمان معامله
- livingRoom: تعداد اتاق خواب (که در دیتاست اولیه به‌اشتباه اتاق نشیمن ترجمه شده بود)
- floorType: نوع طبقه
- floorHeight: ارتفاع طبقه
- buildingType: نوع ساختمان
- renovationCondition: وضعیت بازسازی
- buildingStructure: سازه ساختمان
- ladderRatio: نسبت جمعیت به تعداد آسانسور
- elevator: داشتن یا نداشتن آسانسور
- fiveYearsProperty: آیا ملک پنج‌ساله است یا خیر
- subway: آیا خانه نزدیک مترو است یا خیر

پیش‌پردازش داده‌ها

مراحل پیش‌پردازش و مهندسی ویژگی‌های اعمال‌شده در مقاله به شرح زیر است:

۱. مدیریت داده‌های گمشده: ویژگی‌هایی با بیش از ٪۵۰ داده گمشده (مانند Day on market) حذف شدند. همچنین، ردیف‌هایی که دارای مقادیر گمشده بودند نیز از دیتاست حذف گردیدند.

۲. مهندسی ویژگی‌ها:

- حذف ویژگی‌های مبهم مانند تعداد آشپزخانه، حمام و اتاق پذیرایی؛
- محدود کردن تعداد اتاق خواب (متغیر livingRoom) به بازه ۱ تا ۴؛
- افزودن ویژگی distance (فاصله تا مرکز پکن)؛
- جایگزینی ویژگی constructionTime با age (سن بنا)؛
- تنظیم حداقل مقدار برای price و area؛
- تقسیم ویژگی floor به دو ویژگی floorType و floorHeight.

۳. حذف داده‌های پرت :(Outliers) داده‌های پرت با استفاده از روش Interquartile Range (IQR) شناسایی و حذف شدند.

۴. استانداردسازی و رمزگذاری: ویژگی‌های عددی با روش StandardScaler استانداردسازی و ویژگی‌های طبقه‌ای با روش One-Hot Encoding رمزگذاری شدند. در نتیجه این مرحله، تعداد ویژگی‌ها به ۵۸ افزایش یافت.

۵. تقسیم داده‌ها: دیتاست نهایی با نسبت ۴ به ۱ به دو بخش آموزش (Train) و آزمون (Test) تقسیم گردید.

مدل‌های مورد استفاده

مقاله برای پیش‌بینی قیمت مسکن از چندین مدل یادگیری ماشین استفاده کرده است:

- Random Forest (جنگل تصادفی)
- Extreme Gradient Boosting (XGBoost)
- Light Gradient Boosting Machine (LightGBM)
- Hybrid Regression (رگرسیون ترکیبی متشکل از سه مدل بالا)
- Stacked Generalization (تعمیم انباشته)

۲.۵ بخش دوم: دادگان

در این بخش، مشخصات کلی دیتاست مورد استفاده آورده شده است.

تعداد نمونه‌ها و ویژگی‌ها

این دادگان شامل ۵۴۵ نمونه (ردیف داده) و ۱۲ ویژگی می‌باشد.

نوع داده هر ویژگی

در جدول زیر نوع داده‌ی هر ویژگی نمایش داده شده است:

جدول ۶: نوع داده‌های هر ویژگی

ردیف	ویژگی	نوع داده
۱	area	int۶۴
۲	bedrooms	int۶۴
۳	bathrooms	int۶۴
۴	stories	int۶۴
۵	mainroad	object
۶	guestroom	object
۷	basement	object
۸	hotwaterheating	object
۹	airconditioning	object
۱۰	parking	int۶۴
۱۱	prefarea	object
۱۲	furnishingstatus	object

تعداد مقادیر منحصربه‌فرد هر ویژگی

تعداد مقادیر یکتای هر ویژگی در جدول زیر آورده شده است:

جدول ۷: تعداد مقادیر منحصربه‌فرد در هر ویژگی

ویژگی	تعداد مقادیر منحصربه‌فرد
area	۲۸۴
bedrooms	۶
bathrooms	۴
stories	۴
mainroad	۲
guestroom	۲
basement	۲
hotwaterheating	۲
airconditioning	۲
parking	۴
prefarea	۲
furnishingstatus	۳

۳.۵ بخش سوم: تحلیل اکتشافی داده‌ها (EDA)

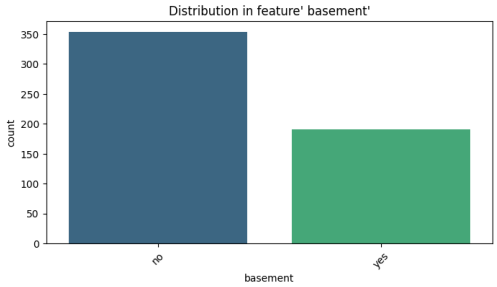
تحلیل اکتشافی داده‌ها (Exploratory Data Analysis-EDA) به فرایندی گفته می‌شود که در آن پژوهشگر با بررسی آماری و تصویری داده‌ها، به درک اولیه‌ای از ساختار، ویژگی‌ها و الگوهای موجود در آن‌ها دست پیدا می‌کند. هدف اصلی EDA کشف روابط پنهان، تشخیص داده‌های پرت، بررسی توزیع متغیرها و شناسایی نواقص داده (مانند مقادیر گمشده) است.

در واقع، تحلیل اکتشافی داده‌ها مرحله‌ای مقدماتی و بسیار حیاتی در فرایند یادگیری ماشین و تحلیل داده محسوب می‌شود؛ زیرا درک درست از داده‌ها پیش‌نیاز انتخاب مدل مناسب، پیش‌پردازش مؤثر و تفسیر نتایج است. این مرحله معمولاً با استفاده از آمار توصیفی (میانگین، میانه، واریانس و ...) و نمودارهایی مانند هیستوگرام، جعبه‌ای، پراکندگی و ماتریس همبستگی انجام می‌شود.

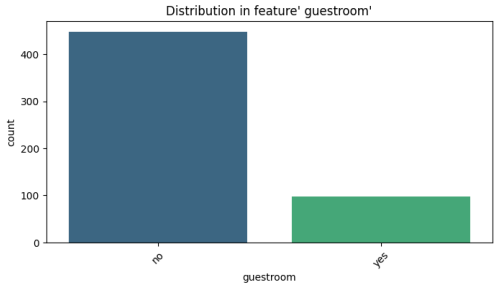
به طور خلاصه، EDA کمک می‌کند تا:

- ماهیت داده‌ها و ویژگی‌های اصلی آن‌ها بهتر درک شود؛
- داده‌های غیرعادی یا اشتباه شناسایی و حذف شوند؛
- روابط بین ویژگی‌ها و متغیر هدف کشف گردد؛
- و مسیر مناسب برای مدل‌سازی بعدی انتخاب شود.

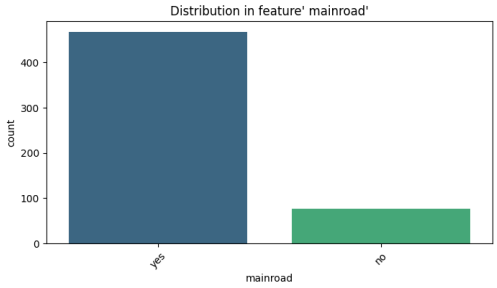
این مجموعه‌داده شامل ۵ ویژگی عددی و ۷ ویژگی دسته‌ای است. در این بخش، می‌توان به تعداد دلخواه، ویژگی‌های دسته‌ای را با استفاده از تابع sns.countplot نمایش داد.



شکل ۱۹: Basement feature in Distribution

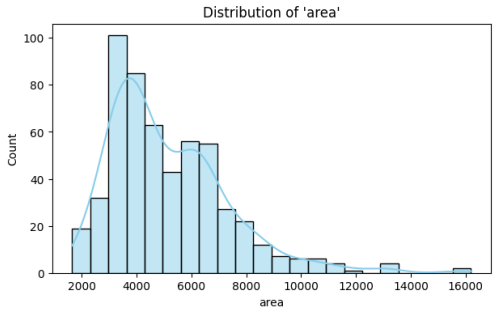


شکل ۲۰: Guestroom feature in Distribution

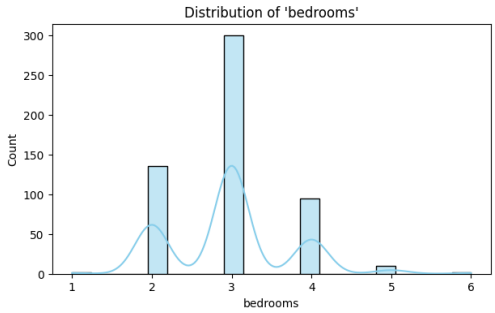


شکل ۲۱: Mainroad feature in Distribution

در این بخش، می‌توان به تعداد دلخواه، ویژگی‌های عددی را با استفاده از sns.distplot نمایش داد.



شکل ۲۲: area of Distribution



شکل ۲۳: bedrooms of Distribution

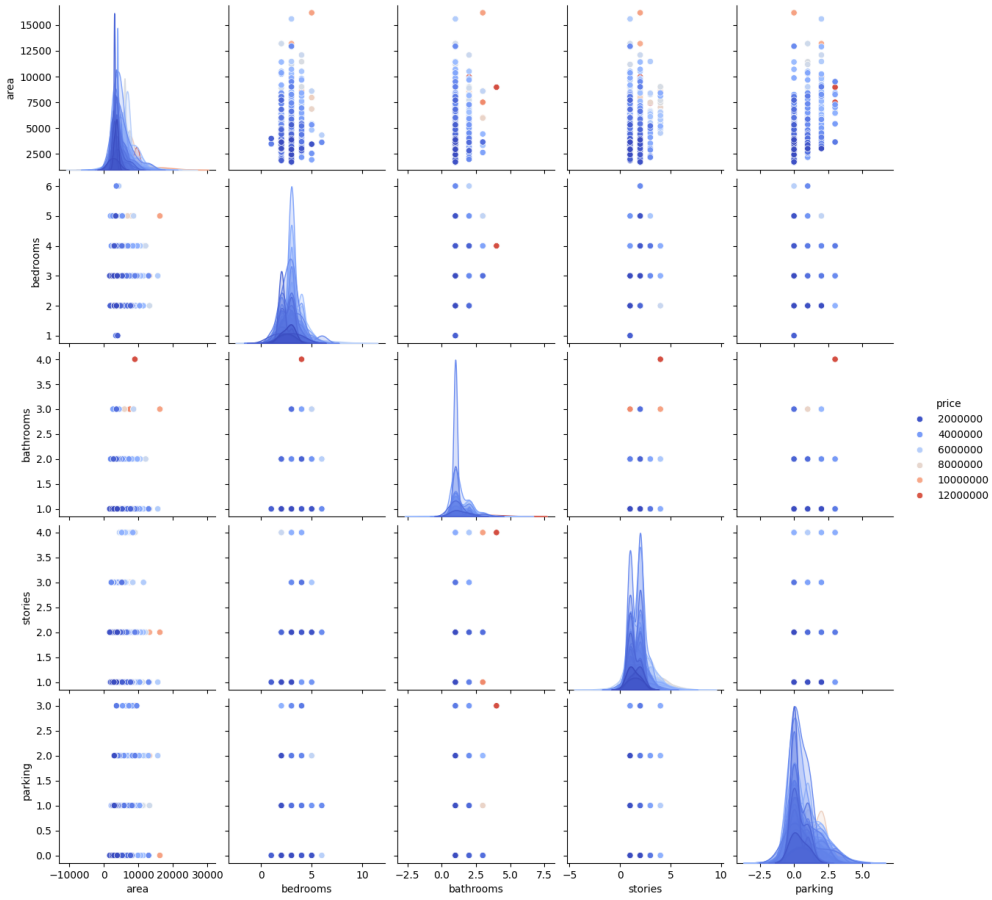
توزیع **area** (مساحت):

- شکل: توزیع دارای چولگی به راست (Right-Skewed) است.
- داده پرت: بله، مقادیر بسیار بزرگ (مانند ۱۲۰۰۰، ۱۴۰۰۰ و ۱۶۰۰۰) که در انتهای "دم" راست نمودار قرار دارند، داده پرت محسوب می‌شوند، زیرا از توده اصلی داده‌ها (متمرکز بین ۲۰۰۰ تا ۸۰۰۰) بسیار دور هستند.

توزیع **bedrooms** (اتاق خواب):

- شکل: توزیع گسسته (Discrete) است.
- داده پرت: بله، مقادیر ۱، ۵ و ۶ داده پرت هستند، زیرا فراوانی آن‌ها در مقایسه با مقادیر رایج (۲، ۳ و ۴) بسیار ناچیز و نزدیک به صفر است.

در نهایت، با استفاده از دستور `sns.pairplot`، روابط بین ویژگی‌های مختلف نمایش داده می‌شود.



شکل ۲۴: pairplot

۴.۵ بخش چهارم: پیشپردازش

داده‌های تکراری با استفاده از دستورات پایتون حذف شده‌اند و دیتاست مورد نظر فاقد داده‌ی گمشده است، بنابراین نیازی به رفع داده‌های ناقص وجود ندارد.

یک نوع کدگذاری مناسب، کدگذاری وجود یا عدم وجود داده است؛ در این روش، برای ۶ ویژگی، وجود داده با ۱ و عدم وجود آن با ۰ نمایش داده می‌شود. نوع دیگر کدگذاری، سطح‌بندی است؛ به‌عنوان مثال، ویژگی‌ای که دارای مقادیر کم، متوسط و زیاد است، از ۰ تا ۲ کدگذاری می‌شود تا نشان‌دهنده افزایش سطح آن ویژگی باشد.

برای شناسایی داده‌های پرت از سه روش رایج استفاده می‌شود:

روش IQR **(Interquartile Range)**

در این روش، فاصله بین چارک سوم و چارک اول محاسبه می‌شود:

$$\text{IQR} = Q_3 - Q_1$$

داده‌هایی که خارج از محدوده زیر قرار داشته باشند، به‌عنوان داده‌های پرت شناخته می‌شوند:

$$[Q_1 - 1.5 \cdot \text{IQR},\ Q_3 + 1.5 \cdot \text{IQR}]$$

مزیت: ساده و رایج و مناسب برای بیشتر داده‌ها.

روش Z-Score

Z-score هر داده نشان‌دهنده فاصله آن از میانگین به صورت انحراف معیار است:

$$Z = \frac{X - \mu}{\sigma}$$

معمولاً داده‌هایی با $|Z| > 3$ به‌عنوان پرت در نظر گرفته می‌شوند.

مزیت: مناسب برای داده‌های تقریباً نرمال.

روش بصری (**Boxplot / Scatterplot**)

با رسم نمودار Boxplot یا Scatterplot، داده‌های پرت قابل مشاهده و حذف هستند.

مزیت: دیداری و سریع برای شناسایی پرت‌های شدید.

روش استفاده‌شده در این پروژه:

ما از روش IQR استفاده کردیم. به این صورت که ابتدا چارک اول و سوم هر ویژگی عددی را محاسبه کرده، فاصله بین آن‌ها (IQR) را به دست آوردیم و داده‌هایی که خارج از محدوده

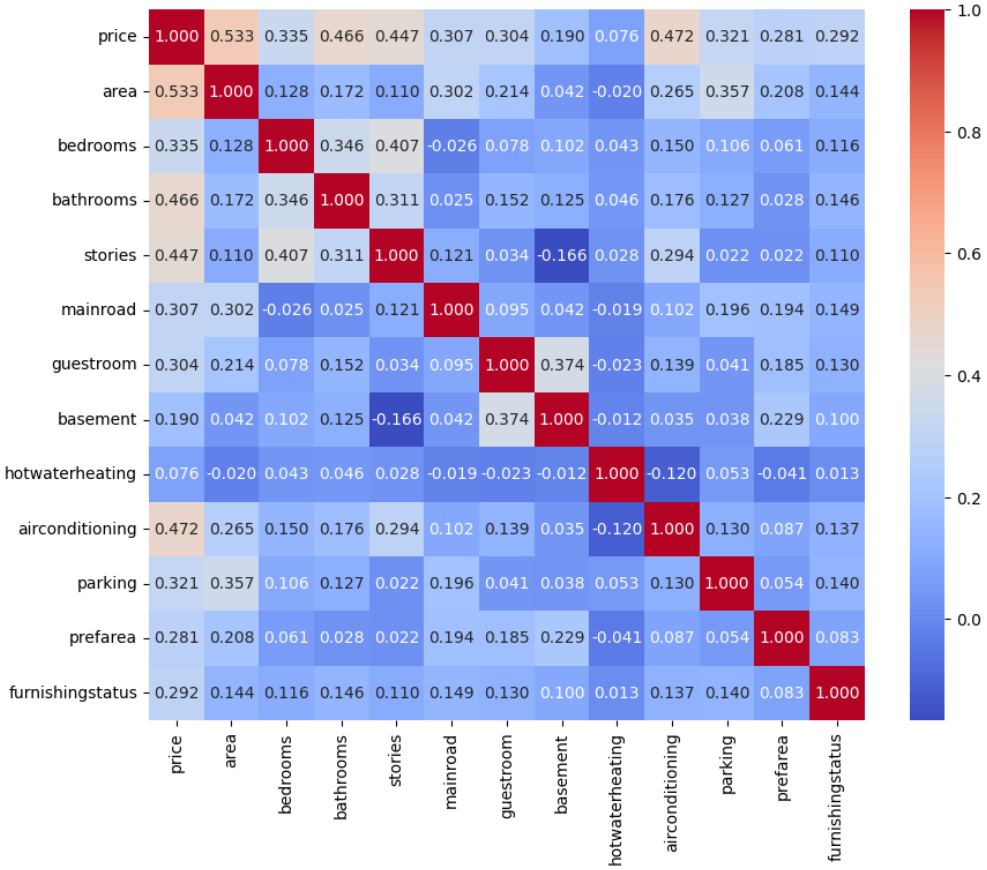
$$[Q_1 - 1.5 \cdot \text{IQR},\ Q_3 + 1.5 \cdot \text{IQR}]$$

قرار داشتند، به‌عنوان داده‌های پرت حذف شدند. این کار باعث شد توزیع داده‌ها متعادل‌تر شود و اثر داده‌های پرت بر تحلیل‌ها کاهش یابد.

در نهایت، داده‌ها به دو بخش آموزش و تست تقسیم شدند و همچنین نرمال‌سازی روی آن‌ها انجام شد تا مقادیر ویژگی‌ها در بازه‌ی ۰ تا ۱ قرار گیرند.

۵.۵ بخش پنجم: انتخاب ویژگی

در این بخش، ابتدا ماتریس همبستگی ویژگی‌ها رسم می‌شود تا مشخص شود کدام ویژگی‌ها بیشترین ارتباط را با یکدیگر دارند.



شکل ۲۵: ماتریس همبستگی

۱. ویژگی‌های کلیدی که بیشترین تأثیر را بر **price** دارند:

- همبستگی مثبت قوی: هرچه رنگ مربع‌ها در ستون/ردیف price به قرمز تیره نزدیک‌تر باشد، آن ویژگی بیشتر با قیمت مرتبط است.
- area (مساحت): با ضریب ۰/۵۳۳ بیشترین همبستگی مثبت را با قیمت دارد. این یعنی بزرگ بودن مساحت خانه، مهم‌ترین عامل در بالا بردن قیمت است.
- airconditioning (سیستم تهویه): با ضریب ۰/۴۷۲ دومین عامل قوی است و نشان می‌دهد وجود تهویه مطبوع، تأثیر زیادی در افزایش قیمت دارد.
- bathrooms (تعداد حمام): با ضریب ۰/۴۶۶ یک پیش‌بینی‌کننده قوی برای قیمت بالاتر است.
- stories (تعداد طبقات): با ضریب ۰/۴۴۷ همبستگی بالایی دارد.

ویژگی‌های با کمترین تأثیر (تقریباً بی‌اثر):

- hotwaterheating (گرمایش آب گرم): با ضریب ۰/۰۷۶ و رنگ خنثی (آبی بسیار روشن)، تقریباً هیچ ارتباط خطی با قیمت خانه ندارد.
- bedrooms (تعداد اتاق خواب): با ضریب ۰/۳۳۵ همبستگی متوسطی دارد، که نشان می‌دهد مساحت (area) و تعداد حمام (bathrooms) مهم‌تر از تعداد اتاق خواب در تعیین قیمت هستند.

۲. همبستگی بین خود ویژگی‌ها:

- Multicollinearity (همبستگی داخلی): نقاطی که در گوشه بالا سمت چپ (زیر قطر اصلی) قرمز تیره هستند، نشان می‌دهند که آن دو ویژگی ممکن است اطلاعات تکراری ارائه دهند.
- area و :price این دو با همبستگی ۰/۵۳۳ بیشترین ارتباط را دارند.
- bathrooms (حمام) و area (مساحت): با ضریب ۰/۴۸۲ با هم همبستگی دارند، که منطقی است (خانه‌های بزرگتر، حمام‌های بیشتری دارند).
- bathrooms (حمام) و bedrooms (اتاق خواب): با ضریب ۰/۳۴۶ با هم مرتبط هستند.

۳. ارتباطات منفی:

- این نمودار ارتباطات منفی کمتری را در متغیرهای اصلی با قیمت نشان می‌دهد.
- basement (زیرزمین) و stories (تعداد طبقات): دارای همبستگی منفی ضعیف (-۰/۱۶۶) هستند.
- mainroad (جاده اصلی) و hotwaterheating (گرمایش آب گرم): دارای همبستگی منفی ضعیف (-۰/۱۹۰) هستند.

نتیجه‌گیری: به طور کلی، این نمودار نشان می‌دهد که مساحت، تهویه مطبوع، تعداد حمام و تعداد طبقات مهم‌ترین متغیرهایی هستند که باید در مدل‌سازی قیمت مسکن به آن‌ها توجه کرد.

۶.۵ بخش ششم: آموزش مدل

در این بخش، با توجه به نیاز به تست ویژگی‌ها در مدل‌ها، برای انتخاب بهترین ویژگی‌ها از طریق PCA تابعی نوشته شد که تمام مدل‌ها را با تعداد ویژگی‌های انتخاب‌شده توسط PCA از ۲ تا ۱۲ بررسی می‌کند. در نهایت، این تابع دقت‌های مربوط به هر حالت را محاسبه و ارائه می‌دهد تا بتوانیم هر یک را تحلیل کنیم.

۱.۶.۵ Multiple Linear Regression

برای این مدل، دقت‌های داده‌های آموزش و تست به شرح زیر است:

جدول ۸: نتایج R^2 برای Multiple Linear Regression با PCA

R^2 Test	R^2 Train	Components PCA
۰.۴۶۲۱	۰.۵۰۸۴	۲
۰.۴۶۹۹	۰.۵۱۷۳	۳
۰.۴۶۹۲	۰.۵۲۲۸	۴
۰.۴۷۰۳	۰.۵۲۹۵	۵
۰.۴۷۰۶	۰.۵۲۹۸	۶
۰.۵۱۴۳	۰.۶۰۷۲	۷
۰.۵۶۵۹	۰.۶۳۷۱	۸
۰.۵۸۱۰	۰.۶۵۲۶	۹
۰.۵۷۴۶	۰.۶۵۸۵	۱۰
۰.۵۹۷۵	۰.۶۷۵۵	۱۱
۰.۵۹۷۵	۰.۶۷۵۸	۱۲

همانطور که نتایج و همبستگی بین ویژگی‌ها مشاهده کردیم، هرچه تعداد ویژگی‌ها بیشتر باشد دقت ما بالاتر می‌رود، برای مدل‌هایی که در ادامه هم نشان می‌دهیم این اصل صدق می‌کند پس نتایج را فقط نمایش می‌دهیم.

۲.۶.۵ Ridge Regression

جدول ۹: نتایج R^2 برای Ridge Regression با PCA

R^2 Test	R^2 Train	Components PCA
۰.۴۶۲۳	۰.۵۰۸۴	۲
۰.۴۷۰۰	۰.۵۱۷۲	۳
۰.۴۶۹۴	۰.۵۲۲۸	۴
۰.۴۷۰۵	۰.۵۲۹۵	۵
۰.۴۷۰۸	۰.۵۲۹۸	۶
۰.۵۱۴۸	۰.۶۰۷۱	۷
۰.۵۶۵۳	۰.۶۳۷۰	۸
۰.۵۸۰۱	۰.۶۵۲۴	۹
۰.۵۷۵۱	۰.۶۵۸۳	۱۰
۰.۵۹۶۸	۰.۶۷۵۲	۱۱
۰.۵۹۶۸	۰.۶۷۵۴	۱۲

۳.۶.۵ Lasso Regression

جدول ۱۰: نتایج R^2 برای Lasso Regression با PCA

R^2 Test	R^2 Train	Components PCA
۰.۴۶۲۱	۰.۵۰۸۴	۲
۰.۴۶۹۹	۰.۵۱۷۳	۳
۰.۴۶۹۲	۰.۵۲۲۸	۴
۰.۴۷۰۳	۰.۵۲۹۵	۵
۰.۴۷۰۶	۰.۵۲۹۸	۶
۰.۵۱۴۳	۰.۶۰۷۲	۷
۰.۵۶۵۹	۰.۶۳۷۱	۸
۰.۵۸۱۰	۰.۶۵۲۶	۹
۰.۵۷۴۶	۰.۶۵۸۵	۱۰
۰.۵۹۷۵	۰.۶۷۵۵	۱۱
۰.۵۹۷۵	۰.۶۷۵۸	۱۲

۴.۶.۵ Polynomial Regression

این مدل چون خودش انتخاب ویژگی باید می‌کرد به همین دلیل دقت آن نسبت به سایر کمی متفاوت خواهد بود.

جدول ۱۱: نتایج R^2 برای Polynomial Regression

R^2 Test	R^2 Train
۰.۲۷۶۵۱۲۳۰۲۱۸۴۲۳۹۶	۰.۷۶۶۱۳۴۳۰۱۰۰۹۷۹۶۶

۵.۶.۵ Multi−Layer Perceptron

جدول ۱۲: نتایج R^2 برای Multi-Layer Perceptron با PCA

R^2 Test	R^2 Train	Components PCA
۰.۴۵۹۱	۰.۴۹۹۵	۲
۰.۴۷۵۱	۰.۵۴۱۵	۳
۰.۴۶۹۴	۰.۵۴۴۲	۴
۰.۴۸۵۹	۰.۵۵۳۴	۵
۰.۵۰۷۳	۰.۵۷۳۷	۶
۰.۵۳۵۲	۰.۶۴۲۲	۷
۰.۵۵۰۹	۰.۶۶۰۹	۸
۰.۵۷۳۰	۰.۶۸۵۹	۹
۰.۵۶۳۵	۰.۷۱۲۰	۱۰
۰.۵۸۱۵	۰.۶۹۳۸	۱۱
۰.۵۸۸۲	۰.۷۰۰۷	۱۲

۶.۶.۵ RegressionElastic−Net

جدول ۱۳: نتایج R^2 برای Elastic−Net Regression با PCA

R^2 Test	R^2 Train	Components PCA
۰.۴۶۲۲	۰.۵۰۸۴	۲
۰.۴۶۹۹	۰.۵۱۷۳	۳
۰.۴۶۹۳	۰.۵۲۲۸	۴
۰.۴۷۰۴	۰.۵۲۹۵	۵
۰.۴۷۰۷	۰.۵۲۹۸	۶
۰.۵۱۴۵	۰.۶۰۷۲	۷
۰.۵۶۵۶	۰.۶۳۷۱	۸
۰.۵۸۰۷	۰.۶۵۲۵	۹
۰.۵۷۴۹	۰.۶۵۸۵	۱۰
۰.۵۹۷۳	۰.۶۷۵۵	۱۱
۰.۵۹۷۳	۰.۶۷۵۷	۱۲

تعریف Elastic-Net Regression یک روش رگرسیون خطی منظم‌سازی‌شده است که ترکیبی از Lasso و Ridge است. این روش به‌ویژه زمانی مفید است که ویژگی‌ها با هم همبستگی بالایی دارند و انتخاب ویژگی‌ها تنها با Lasso ناپایدار می‌شود. ترکیب L1 و L2 باعث می‌شود مدل هم ویژگی‌های مهم را انتخاب کند و هم از بزرگ شدن ضرایب جلوگیری کند.

فرمول‌ها هدف تابع از دست دادن (Loss Function) برای Elastic-Net به صورت زیر است:

$$\mathcal{L}(\beta) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \beta\right)^2 + \lambda\left(\alpha\|\beta\|_1 + \frac{1-\alpha}{2}\|\beta\|_2^2\right)$$

که در آن:

- n تعداد نمونه‌ها است.
- $\mathbf{x}_i$ بردار ویژگی‌های نمونه iام است.
- y_i مقدار هدف نمونه iام است.
- β بردار ضرایب رگرسیون است.
- λ پارامتر منظم‌سازی که شدت جریمه را تعیین می‌کند.
- α نسبت ترکیب بین L1 و L2 است:
  - Lasso $\Rightarrow \alpha = 1$
  - Ridge $\Rightarrow \alpha = 0$

مزایا

- همزمان ویژگی‌های مهم را انتخاب می‌کند و از overfitting جلوگیری می‌کند.
- در داده‌های دارای همبستگی بالا بین ویژگی‌ها پایدارتر از Lasso عمل می‌کند.

نتیجه‌گیری مدل‌ها و تحلیل نتایج

با توجه به نتایج ارائه شده برای مدل‌های مختلف رگرسیونی و شبکه عصبی، می‌توان نکات زیر را استخراج کرد:

- RegressionLinearMultiple: با افزایش تعداد مؤلفه‌های PCA از ۲ تا ۱۲، مقدار R^2 Train از ۰.۵۰۸۴ به ۰.۶۷۵۴ و R^2 Test از ۰.۴۶۲۳ به ۰.۵۹۶۸ افزایش یافته است. این نشان می‌دهد که استفاده از تعداد مؤلفه‌های بیشتر باعث بهبود عملکرد مدل در داده‌های آموزش و تست می‌شود.
- RegressionRidge،Lasso: روند مشابهی مشاهده می‌شود. با افزایش مؤلفه‌ها، R^2 Train و R^2 Test افزایش یافته و تفاوت بین مدل‌های خطی ساده و با پنالتی (Ridge/Lasso) اندک است، که نشان‌دهنده همگرایی مدل‌ها و پایدار بودن نتایج است.
- RegressionPolynomial: R^2 Train بسیار بالا (۰.۷۶۶۱۳۴۳۰۱۰۰۹۷۹۶۶) و R^2 Test پایین (۰.۲۷۶۵۱۲۳۰۲۱۸۴۲۳۹۶) است. این نشانه overfitting شدید مدل پلی‌نومیال است؛ مدل روی داده‌های آموزش عملکرد عالی دارد اما روی داده‌های تست ضعیف عمل می‌کند.
- PerceptronMulti-Layer: با افزایش تعداد مؤلفه‌های PCA، R^2 Train از ۰.۴۹۹۵ به ۰.۷۰۰۷ و R^2 Test از ۰.۴۵۹۱ به ۰.۵۸۸۲ افزایش یافته است. این روند نشان می‌دهد که شبکه عصبی چندلایه توانایی یادگیری روابط پیچیده بین ویژگی‌ها را دارد و عملکرد قابل قبولی در پیش‌بینی قیمت‌ها ارائه می‌دهد.
- RegressionElastic-Net: عملکرد مشابه Ridge و Lasso دارد. R^2 Train و R^2 Test با افزایش تعداد مؤلفه‌ها بهبود یافته و نتایج با سایر مدل‌های خطی منطبق است.
- جمع‌بندی کلی:
  - افزایش تعداد مؤلفه‌های PCA باعث بهبود عملکرد اکثر مدل‌ها می‌شود، اما مدل پلی‌نومیال به دلیل بیش‌برازش عملکرد ضعیفی روی داده‌های تست دارد.
  - مدل‌های خطی با پنالتی (Ridge،Lasso،Elastic-Net) عملکردی مشابه با مدل خطی ساده دارند و پایدار هستند.
  - شبکه عصبی چندلایه (Perceptron Multi-Layer) به دلیل توانایی مدل‌سازی روابط غیرخطی، بهترین تعادل بین داده‌های آموزش و تست را نشان می‌دهد.
  - برای انتخاب مدل نهایی، توجه به R^2 Test مهم است و در این میان شبکه عصبی چندلایه با PCA مناسب، بهترین گزینه است.

۷.۵ بخش هفتم : استفاده از MLP تحت انتخاب کننده ویژگی

حال همانند مرحله قبل، انتخاب ویژگی با استفاده از MLP انجام می‌شود. ویژگی‌ها از لایه‌ی یکی‌مانده به آخر استخراج شده و مدل‌ها مجدداً آموزش داده می‌شوند تا نتایج به‌دست آمده تحلیل شوند.

تحلیل ویژگی‌های استخراج شده از MLP

در این بخش، لایه آخر شبکه عصبی چندلایه (MLP) را برابر با ۳۰ نورون تنظیم کردیم تا بتوانیم ۳۰ ویژگی استخراج شده از داده‌ها داشته باشیم. این افزایش تعداد ویژگی‌ها به ما امکان می‌دهد تا عملکرد مدل‌های رگرسیونی مختلف را با استفاده از این ویژگی‌های جدید بررسی و با نتایج قبلی مقایسه کنیم. جدول زیر نتایج R^2 Train و R^2 Test مدل‌ها را با ویژگی‌های استخراج شده از MLP نشان می‌دهد:

جدول ۱۴: ویژگی‌های استخراج شده از MLP و مقایسه با مدل‌های قبلی

مدل	R^2 Train	R^2 Test
Multiple Linear Regression	۰.۶۷۵۸	۰.۵۹۷۵
Ridge Regression	۰.۶۶۷۲	۰.۵۹۱۲
Lasso Regression	۰.۶۶۴۴	۰.۵۸۵۹
Multi-Layer Perceptron	۰.۶۶۴۴	۰.۵۸۵۹
Elastic-Net Regression	۰.۶۶۴۴	۰.۵۸۵۹

نتیجه‌گیری کلی

- افزایش تعداد ویژگی‌ها با استفاده از لایه آخر MLP باعث شد که مدل‌ها اطلاعات بیشتری برای یادگیری روابط پیچیده داشته باشند.
- با مقایسه این نتایج با نتایج قبلی که از PCA استفاده شده بود، مشاهده می‌کنیم که:
  - R^2 Train تقریباً مشابه مدل‌های PCA بالا است، که نشان می‌دهد ویژگی‌های استخراج شده از MLP توانایی مدل‌سازی خوبی دارند.
  - R^2 Test نیز نسبتاً مشابه یا کمی کمتر از بهترین نتایج PCA است، بنابراین ویژگی‌های استخراج شده از MLP عملکرد قابل قبولی ارائه می‌دهند.
- این مقایسه نشان می‌دهد که استفاده از شبکه عصبی برای استخراج ویژگی‌های غیرخطی می‌تواند جایگزین مناسبی برای کاهش ابعاد با PCA باشد، به‌ویژه وقتی تعداد ویژگی‌های جدید افزایش یابد.

۶ امتیازی: RFE،VIF

۱. VIF(VarianceInflatioFactor)

توضیح مفهومی: VIF معیاری است برای سنجش همخطی چندگانه (Multicollinearity) بین ویژگی‌ها در یک مدل رگرسیونی. همخطی زمانی رخ می‌دهد که برخی ویژگی‌ها با هم همبستگی بالایی دارند و اطلاعات مشابهی ارائه می‌دهند.
وجود همخطی زیاد می‌تواند اثرات زیر داشته باشد:

- ضرایب رگرسیون ناپایدار شوند.

- تفسیر مدل دشوار شود.
- دقت پیش‌بینی کاهش یابد.

فرمول محاسبه: برای هر ویژگی X_i، VIF به صورت زیر تعریف می‌شود:

$$\text{VIF}_i = \frac{1}{1 - R_i^2}$$

که R_i^2 ضریب تعیین (R-squared) حاصل از رگرسیون X_i روی سایر ویژگی‌هاست.

تفسیر: اگر $\text{VIF} \geq 5$ (یا ۱۰)، نشان‌دهنده همخطی زیاد است و آن ویژگی ممکن است از مدل حذف شود تا پایداری و دقت پیش‌بینی افزایش یابد.

نکته: در عمل، مقدار آستانه VIF بسته به داده‌ها و مدل انتخاب می‌شود، اما مقادیر بالاتر از ۵ اغلب هشداردهنده هستند.

۲. RFE(RecursiveFeatureElimination)

توضیح مفهومی: RFE یک روش انتخاب ویژگی (Feature Selection) است که ویژگی‌های مهم برای پیش‌بینی را شناسایی می‌کند. این روش به جای بررسی ویژگی‌ها به صورت جداگانه، تأثیر هر ویژگی همراه با سایر ویژگی‌ها را نیز در نظر می‌گیرد.

روش کار:

۱. انتخاب یک مدل پایه، مانند LinearRegression، Ridge یا RandomForest.

۲. آموزش مدل روی داده‌ها و محاسبه اهمیت هر ویژگی.

۳. حذف کم‌اهمیت‌ترین ویژگی.

۴. تکرار مراحل ۱ تا ۳ تا زمانی که تعداد ویژگی‌های مورد نظر باقی بماند.

مزیت :RFE

- در نظر گرفتن تأثیر ویژگی‌ها همراه با سایر ویژگی‌ها.
- کمک به شناسایی ویژگی‌های مهم و کاهش ابعاد داده‌ها بدون از دست دادن اطلاعات کلیدی.

نکته: انتخاب مدل پایه و تعداد ویژگی‌های نهایی در RFE می‌تواند به طور قابل توجهی بر عملکرد مدل نهایی اثر بگذارد و باید با دقت انجام شود.

در نهایت، نتایج هر دو روش به شرح زیر است:

۱.۰.۶ VIF

جدول VIF ویژگی‌ها

در این بخش، مقدار VIF برای هر ویژگی محاسبه شده است تا میزان همخطی چندگانه بررسی شود. همانطور که مشاهده می‌کنید، برخی ویژگی‌ها مانند area، bedrooms و mainroad دارای VIF بالای ۵ هستند که نشان‌دهنده همخطی نسبتاً زیاد با سایر ویژگی‌ها است و ممکن است باعث ناپایداری ضرایب رگرسیون شوند.

جدول ۱۵: مقدار VIF برای ویژگی‌ها

VIF	Feature
۵/۷۶۵۳	area
۶/۲۹۹۹	bedrooms
۱/۶۹۰۷	bathrooms
۲/۷۵۲۷	stories
۶/۲۶۰۲	mainroad
۱/۵۸۲۳	guestroom
۲/۰۶۹۹	basement
۱/۰۷۷۰	hotwaterheating
۱/۷۴۰۱	airconditioning
۱/۹۴۳۲	parking
۱/۴۲۸۹	prefarea
۲/۵۹۰۴	furnishingstatus

نکته: ویژگی‌هایی با VIF بالای ۵ (area, bedrooms, mainroad) ممکن است همخطی زیادی داشته باشند و برای افزایش پایداری مدل، حذف یا ترکیب آن‌ها با سایر ویژگی‌ها پیشنهاد می‌شود.

۲.۰.۶ RFE

جدول ۱۶: ویژگی‌های انتخاب‌شده برای تحلیل نهایی

ویژگی‌ها
area
bathrooms
stories
hotwaterheating
airconditioning