



به نام خدا

درس یادگیری ماشین

تمرین سوم

حدیثه مصباح

۸۱۰۱۰۲۲۵۳



پاسخ ۱.

۱/۱) حقیقی تعیین مدل ما برای هر نمونه:

$$y_i = \beta_0 + \varepsilon_i \rightarrow \varepsilon_i = y_i - \beta_0$$

برای تابع هزینه از معیار SSE استفاده می‌کنیم

$$J = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0)^2$$

در این جا چون هدف ما β_0 هست باید نسبت به اون مشتق بگیریم

$$\frac{\partial J}{\partial \beta_0} = 0 \rightarrow -2 \sum_{i=1}^n (y_i - \beta_0) = 0 \rightarrow \sum_{i=1}^n y_i - (n \beta_0) = 0 \rightarrow \beta_0 = \frac{\sum_{i=1}^n y_i}{n} = \bar{y}$$

$$\xrightarrow{\text{حاصلی}} \frac{46+80+26+52+78+75+10+70+64+30}{10} = \boxed{\frac{561}{10}}$$

۱/۲) حقیقی که ما در این تعیین داریم به سبب زیر است

$$y_i = \beta_0 + \alpha_i + \varepsilon_i \rightarrow \varepsilon_i = y_i - \beta_0 - \alpha_i$$

برای محاسبه تابع هزینه از معیار SSE استفاده می‌کنیم

$$J = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \alpha_i)^2$$

مثل سوال قبل مشتق گرفته و برابر صفر قرار می‌دهیم

$$\frac{\partial J}{\partial \beta_1} = 0 \rightarrow -2 \sum_{i=1}^n \alpha_i (y_i - \beta_0 - \alpha_i) = 0$$

$$\sum_{i=1}^n \alpha_i y_i - \sum_{i=1}^n \beta_0 \alpha_i - \sum_{i=1}^n \alpha_i^2 = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = \sum_{i=1}^n \beta_0 \alpha_i + \sum_{i=1}^n \alpha_i^2 \rightarrow \beta_1 = \frac{\sum_{i=1}^n \alpha_i y_i}{\sum_{i=1}^n \alpha_i^2}$$

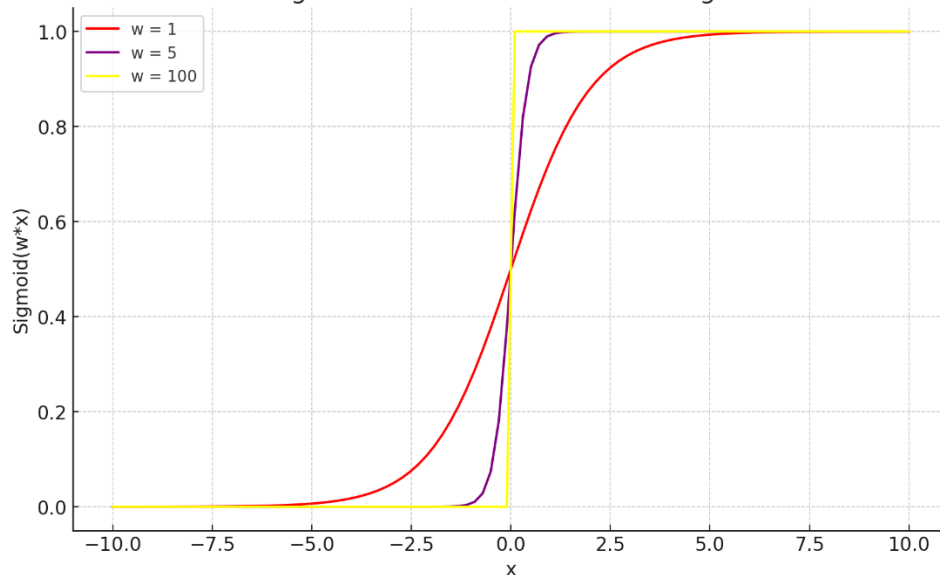
$$\xrightarrow{\text{حاصلی}} \beta_1 = \frac{(16 \times 46) + (27 \times 80) + (11 \times 36) + (2 \times 92) + (3 \times 78) + (26 \times 75) + (5 \times 10) + (24 \times 70) + (21 \times 64) + (1 \times 30)}{16^2 + 27^2 + 11^2 + 2^2 + 3^2 + 26^2 + 5^2 + 24^2 + 21^2 + 1^2} = \frac{12521}{4173} \approx \boxed{3.0004}$$

۱/۳) **فرض** احتمال این که دقیقاً ۲۲ بسوز خنک کم است. با توجه به داده‌های من می‌توانیم که حدودی که از ۲۲ حاصل می‌شود. چون این خطای تعیین می‌دهد که مجموع خطاهای ما کمینه شود حالا ممکن مقداری اشتباه هم کند اما ما محاسبه می‌کنیم که این با توجه به هم داده‌ها تعیین می‌کنیم به سبب این که خطای کمترین خطای دارد.

۱/۴

$$\sigma^2 = MSE = \frac{SSE}{\frac{\partial F}{\partial \beta_0}} = \frac{SSE}{n-1-(k-1)} = \frac{7}{16-1-1} = \frac{7}{14} = \boxed{0.5}$$

Sigmoid Function for Different Weights



با توجه به شکل که در بالا به ازای w های مختلف برای تابع (Sigmoid) رسم کردیم می توان متوجه شد که هر چه w ها بزرگتری انتخاب کنیم منحنی بیش تر و تندتر می شود. ما می دانیم که منحنی هر چه قدر تند تر باشد یعنی مدل ما از انتخاب کلاس در هر اطمینان بالاتری دارد. وقتی وزن های بزرگتری انتخاب می کنیم تغییرات کوچک در ورودی باعث تغییرات زیادی در کلاس احتمال ما در نهایت منجر به تغییر ضربه می شود و همین باعث می شود که $overfit$ واقعیه می کند نشان می دهد که با وزن های بزرگتر ما دچار مشکل می شویم. به عنوان مثال هارفتی وزن کمی مثل ۱ یا ۵. را انتخاب کنیم منطبق با $Sigmoid$ حلال تر است. و این باعث می شود نسبت به تغییرات ورودی کمتر واکنش نشان دهد و هر چه در بزرگتر می شود پیوسته تابع پله می شود. در این حالت مدل برای تعمله هایی که خیلی از مرز تصمیم هم دور نیستن نظر قطعی می ده واجب سون هلاک هیله که قطعی متعلق به کلاس اول هست در صورتی که نزدیک به مرز هست و نباید در مرز بعدی حال کلاس اول هست. در مورد $overfit$ ما هر چه حساسیت بیشتری داشته باشیم امکان این که باعث $overfit$ بشود هست. در این مواقع می گوئیم مدل به حباب یادگیری دیسای آموزش را حفظ کرده است. در نتیجه می توان گفت با w بزرگتر منحنی بیش تر می شود و تغییرات جزئی می تواند سایه بزرگی روی تخصیص کلاس ها بگذارد، باعث $overfit$ می شود زیرا نویز ها و داده هایی که اهمیت ندارند را یاد گرفته است و وقتی دیسای جدید به آن می دهیم برای بعضی منه دچار این بیازس شو و نتایج خوبی به ما نمی دهد.



$$\omega = [\omega_1, \dots, \omega_d]^T \longrightarrow L(\omega) = \log(P(\omega) \prod_{j=1}^n P(y^j | x^j, \omega)) \quad (2.2)$$
$$P(\omega) = \prod_{i=1}^d \frac{1}{\sqrt{2\pi}} e^{-\frac{\omega_i^2}{2}}$$

مسئله تقسیم

MAP

$$\omega^* = \arg \max_{\omega} L(\omega) = \arg \max_{\omega} \left[\sum_{j=1}^n \log(P(y^j | x^j, \omega)) - \sum_i \frac{\omega_i^2}{2} \right]$$

$$\omega_i^{(t+1)} \leftarrow \omega_i^{(t)} + \alpha \frac{\partial L(\omega)}{\partial \omega_i} \longrightarrow \text{در زمان } t$$

$$\frac{\partial L(\omega)}{\partial \omega_i} = \frac{\partial \log P(\omega)}{\partial \omega_i} + \frac{\partial \log \left(\prod_{j=1}^n P(y^j | x^j, \omega) \right)}{\partial \omega_i} \longrightarrow -\omega_i = \frac{\partial \log P(\omega)}{\partial \omega_i}$$

به روز رسانی فرایند ها

$$\left(-\omega_i^t + \sum_j x_i^j (y^j - P(y=1 | x^j, \omega^{(t)})) \right) \alpha + \omega_i^{(t)} \rightarrow \omega_i^{(t+1)}$$



پاسخ 3.

سوال 1

$$\min L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \lambda_i y_i (a_i w + b) + \sum_{i=1}^n \lambda_i \leftarrow \text{لاگرانژ}$$

$$\max g(\lambda) = L_D(\lambda_i) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j x_i^T x_j \leftarrow \text{دوال}$$

$$s.t. \sum_{i=1}^n \lambda_i y_i = 0, \lambda_i \geq 0$$

ما نمونه داریم پس $n=3$ و بزرگترین $g(\lambda)$ به شرح زیر است:

$$g(\lambda) = \lambda_1 \lambda_2 \lambda_3 - \frac{1}{2} [\lambda_1^2 y_1^2 x_1^T x_1 + \lambda_1 \lambda_2 y_1 y_2 x_1^T x_2 + \lambda_1 \lambda_3 y_1 y_3 x_1^T x_3 + \lambda_2 \lambda_1 y_2 y_1 x_2^T x_1 + \lambda_2^2 y_2^2 x_2^T x_2 + \lambda_2 \lambda_3 y_2 y_3 x_2^T x_3 + \lambda_3 \lambda_1 y_3 y_1 x_3^T x_1 + \lambda_3 \lambda_2 y_3 y_2 x_3^T x_2 + \lambda_3^2 y_3^2 x_3^T x_3]$$

$$\xrightarrow{\text{حالا}} g(\lambda) = \lambda_1 + \lambda_2 + \lambda_3 - \frac{1}{2} [5\lambda_1^2 + 5\lambda_2^2 + 5\lambda_3^2 + 6\lambda_1\lambda_2 + 10\lambda_1\lambda_3 + 6\lambda_2\lambda_3]$$

موضوعی مسئله‌ای بهینه سازی با تابع هدف زیر است:

$$w^* = \sum_{i=1}^n \lambda_i y_i a_i, b^* = y_i - \sum_{i=1}^n \lambda_i y_i a_i a_j \quad \text{و} \quad \text{حل معادله} \quad y_i (w^{*T} a_i + b^*) = 1$$

حال باید تا $g(\lambda)$ را برای این مقیودی که در مسئله داریم حداکثر می‌نماییم

$$\max g(\lambda) = \lambda_1 + \lambda_2 + \lambda_3 - \frac{1}{2} [5\lambda_1^2 + 5\lambda_2^2 + 5\lambda_3^2 + 6\lambda_1\lambda_2 + 10\lambda_1\lambda_3 + 6\lambda_2\lambda_3]$$

$$s.t. \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 = 0, \lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0$$

حال برای حل مسئله بهینه سازی را به کینه سازی تبدیل کنیم. در صورت ضرب یک هشتی در $g(\lambda)$ و استفاده از قضیه قسری:

μ_i دنا مساوی λ_i استفاده می‌کنیم

$$\min \frac{1}{2} (5\lambda_1^2 + 5\lambda_2^2 + 5\lambda_3^2 + 6\lambda_1\lambda_2 + 10\lambda_1\lambda_3 + 6\lambda_2\lambda_3) - \lambda_1 - \lambda_2 - \lambda_3 + \mu_1 (\lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3) - \mu_1 \lambda_1 + \mu_2 \lambda_2 - \mu_3 \lambda_3$$

حال مشتق می‌گیریم $\mu_1, \mu_2, \mu_3 \geq 0$

مساوی صفر قرار می‌دهیم، شرط حبابی به دست می‌آید



$$\frac{\partial (g(\lambda_i))}{\partial \lambda_i} = 0 \rightarrow$$

$$\begin{cases} 5\lambda_1 + 3\lambda_2 + 5\lambda_3 - 1 - \mu_1 = 0 \\ 5\lambda_2 + 3\lambda_1 + 3\lambda_3 - 1 - \mu_2 = 0 \\ 5\lambda_3 + 3\lambda_1 + 3\lambda_2 - 1 - \mu_3 = 0 \end{cases}$$

$$y_3 = 1$$

$$y_2 = -1$$

$$y_1 = -1$$

$$\frac{\partial (g(\lambda_i))}{\partial \mu_i} = 0 \rightarrow$$

$$\begin{cases} -\lambda_1 = 0 \\ -\lambda_2 = 0 \\ -\lambda_3 = 0 \end{cases} \rightarrow \begin{cases} -\mu_1 \lambda_1 = 0 \\ -\mu_2 \lambda_2 = 0 \\ -\mu_3 \lambda_3 = 0 \end{cases}$$

برای یافتن پاسخ بهینه باید حالت های مختلف را در نظر گرفت

$$\lambda_1 \neq 0, \mu_2 = \lambda_2 = \mu_3 = \lambda_3 = 0, \mu_1 = 0 \rightarrow \boxed{\lambda_1 = 0}$$

باید حالت زیادی بررسی شود اما چون این تابع convex هست می توانیم تنها یک حالت را بنویسیم و بماند و در اکثر حالات میماند
کنیم که برای هر چند عدد که در بین کلمات است و بررسی می شود بهینه را بررسی کنیم

$$\lambda_2, \lambda_3 \neq 0, \mu_1 = \lambda_1 = \mu_2 = \mu_3 = 0$$

جواب مسئله است

$$\begin{cases} 3\lambda_2 + 5\lambda_3 - 1 = 0 \\ 5\lambda_2 + 3\lambda_3 - 1 = 0 \end{cases} \xrightarrow{\text{حل کردن}} \lambda_2 = \lambda_3 = \frac{1}{8}$$

$$\rightarrow \lambda^* = \begin{bmatrix} 0 \\ \frac{1}{8} \\ \frac{1}{8} \end{bmatrix} \xrightarrow{\text{جایگزینی در معادله}} w^* = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$$

$$y_i (w^{*T} x_i + b^*) = 1 \xrightarrow{\text{جایگزینی در معادله}} b^* = 0$$

$$-0.5 a_2 = 0 \rightarrow a_2 = 0$$

رابطه خط جدا کننده

سوال چهارم

4-1) در حالت کلی کرنل‌ها باعث می‌شوند که SVM بتواند مسائل طبقه‌بندی پیچیده و غیر خطی را حل کند.

① در مسئله‌های دنیای واقعی داده‌ها به صورت خطی قابل تفکیک نیستند و هیچ خطی یا صفحه‌ای نمی‌تواند این داده‌ها را از هم جدا کند. کرنل‌ها به رفع این مشکل کمک می‌کنند.

کرنل‌ها باعث می‌شوند که SVM بتواند داده‌ها را به ابعاد بالاتری نگاشت کند و به این ترتیب داده‌ها به صورت خطی از هم تفکیک شوند (کتاب‌های نیاز به محاسبه مقدمات در این وقت).

② کرنل‌های مختلف شباهت بین نقاط داده را به روش‌های مختلف اندازه‌گیری می‌کنند.

با توجه به کرنل و انتخاب مناسب با توجه به داده‌های من می‌توان SVM مناسب‌تری از داده‌ها را دارم.

③ کرنل‌ها باعث می‌شوند که از overfit جلوگیری شود با استفاده از تنظیم سازی و به حداقل رساندن ریسک.

با استفاده از پارامترهای کرنل می‌توان پیچیدگی مدل را کنترل کرد.

④ یک مفهوم مهم kernel Trick است که امکان محاسبه ضرب داخلی برداری را در فضای ویژگی با ابعاد بالا و سریع می‌دهد.

در کرنل‌های مختلفی داریم که هر کدام می‌تواند فضای ویژگی را به روش‌های مختلف به فضای ویژگی تبدیل کند و به این ترتیب می‌تواند داده‌ها را به صورت خطی از هم تفکیک کند.

4-2) فرض می‌کنیم که $k_1(x, x') = \langle \phi_1(x), \phi_1(x') \rangle$ و $k_2(x, x') = \langle \phi_2(x), \phi_2(x') \rangle$ باشد.

اگر x, x' بردارهایی باشند بعد از تبدیل به فضای ویژگی ϕ_1 و ϕ_2 خواهیم داشت:

$$K(x, x') = k_1(x, x') k_2(x, x') = \langle \phi_1(x), \phi_1(x') \rangle \langle \phi_2(x), \phi_2(x') \rangle$$

$$= \left(\sum_i \phi_1^i(x) \phi_1^i(x') \right) \left(\sum_j \phi_2^j(x) \phi_2^j(x') \right) = \sum_{i,j=1}^l \left(\phi_1^i(x) \phi_2^j(x) \right) \left(\phi_1^i(x') \phi_2^j(x') \right) = \langle \phi'(x), \phi'(x') \rangle$$

تابع را به صورت زیر تعریف می‌کنیم:

$$\phi'(x) = \begin{bmatrix} \phi_1^1(x) \phi_2^1(x) \\ \phi_1^1(x) \phi_2^2(x) \\ \phi_1^2(x) \phi_2^1(x) \\ \phi_1^2(x) \phi_2^2(x) \\ \vdots \end{bmatrix}$$

ϕ' یک بردار نیست زیرا x بعدی است جایی که ϕ_1 و ϕ_2 متناظر با هر جهت i, j (از 1 تا l) هستند.

$\phi'(x_i) = \phi_1^1(x_i) \phi_2^1(x_i)$ است. در نتیجه $K(x, x')$ یک kernel هست به این معنی که می‌تواند به صورت خطی از هم تفکیک کند.



پاسخ ۵.

(۵.۱)

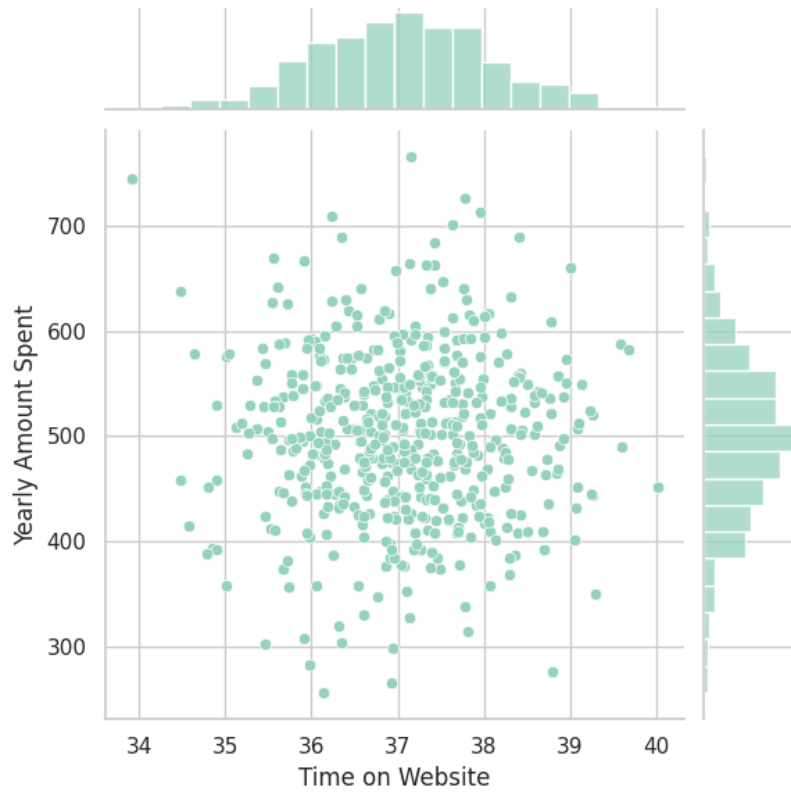
```
General Information:
↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                ---
0   Email                                500 non-null    object
1   Address                             500 non-null    object
2   Avatar                              500 non-null    object
3   Avg. Session Length                 500 non-null    float64
4   Time on App                         500 non-null    float64
5   Time on Website                     500 non-null    float64
6   Length of Membership                 500 non-null    float64
7   Yearly Amount Spent                 500 non-null    float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB

Statistical Summary:
      Avg. Session Length  Time on App  Time on Website  \
count      500.000000      500.000000      500.000000
mean         33.053194        12.052488        37.060445
std           0.992563         0.994216         1.010489
min          29.532429         8.508152        33.913847
25%          32.341822        11.388153        36.349257
50%          33.082008        11.983231        37.069367
75%          33.711985        12.753850        37.716432
max          36.139662        15.126994        40.005182

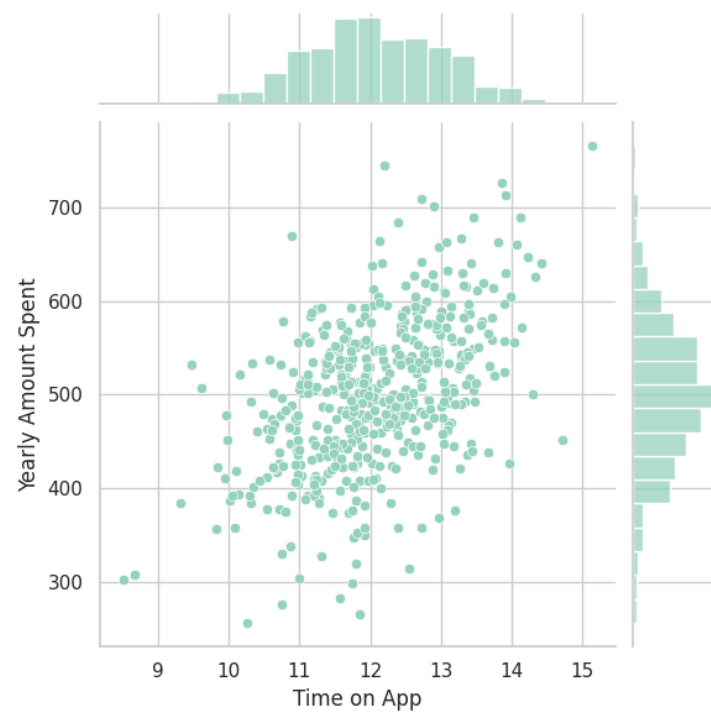
      Length of Membership  Yearly Amount Spent
count      500.000000      500.000000
mean         3.533462        499.314038
std           0.999278         79.314782
min           0.269901        256.670582
25%           2.930450        445.038277
50%           3.533975        498.887875
75%           4.126502        549.313828
max           6.922689        765.518462
```

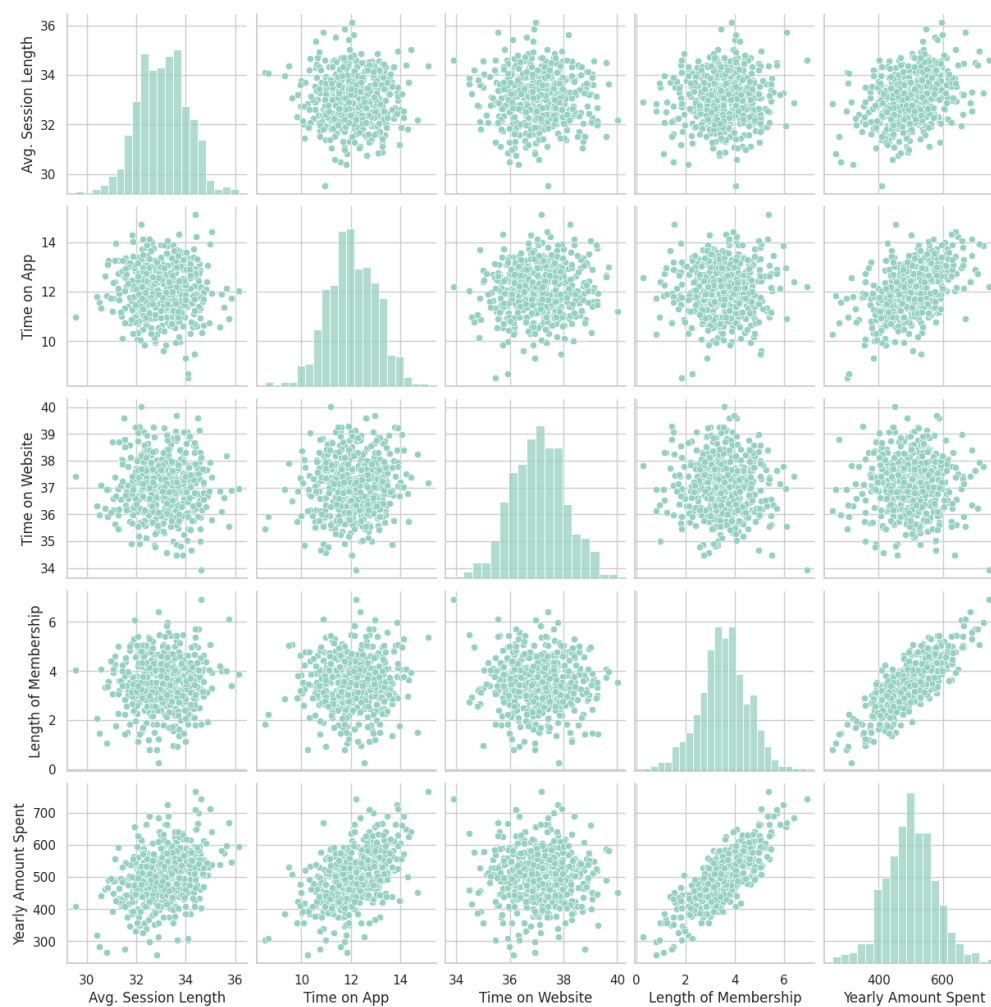
بعد از خواندن اطلاعات با استفاده از توابع داده شده ان ها به شکل بالا نمایش داده میشود.

(۵.۲)



(۵.۳





نمودار برای مجموعه داده ها در بالا نمایش داده شده است. این نمودار روابط بین تمام ویژگی‌های عددی در مجموعه داده‌ها را نشان می‌دهد.

از این نمودار، به نظر می‌رسد که ویژگی "مدت عضویت" بیشترین همبستگی مثبت را با "میزان خرج سالانه" دارد. این بدان معناست که هرچه مشتری برای مدت طولانی‌تری عضو باشد، معمولاً به طور سالانه بیشتر خرج می‌کند. ویژگی‌های دیگر مانند "زمان استفاده از اپلیکیشن" و "میانگین طول جلسات" نیز تا حدودی همبستگی مثبتی با خرج سالانه نشان می‌دهند، اما "مدت عضویت" به عنوان بیشترین مورد برجسته است.



```
# Assuming 'Email', 'Address', 'Avatar' are the categorical features to be dropped
features = ecommerce_df.drop(['Email', 'Address', 'Avatar', 'Yearly Amount Spent'], axis=1)
label = ecommerce_df['Yearly Amount Spent']

# Scaling the features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

برای تقسیم داده‌ها به ویژگی‌ها و برچسب‌ها و سپس انجام پیش‌پردازش‌های لازم، این مراحل را انجام می‌دهیم:

۱. تقسیم داده‌ها به ویژگی‌ها و برچسب:

- 'ویژگی‌ها' شامل تمام ستون‌ها به جز 'میزان خرج سالانه' خواهند بود.

- 'برچسب' ستون 'میزان خرج سالانه' خواهد بود که متغیر هدف ما است.

۲. مراحل پیش‌پردازش:

- رسیدگی به داده‌های دسته‌ای: اگر ویژگی‌های دسته‌ای (مانند 'ایمیل'، 'آدرس' و 'آواتار' در این مجموعه داده) وجود داشته باشند، باید تصمیم بگیریم چگونه با آن‌ها برخورد کنیم. برای مدل‌های یادگیری ماشین، ممکن است لازم باشد این‌ها را با استفاده از تکنیک‌هایی مانند کدگذاری یک-به-یک به مقادیر عددی تبدیل کنیم. با این حال، برای این مجموعه داده، این ویژگی‌های دسته‌ای ممکن است برای پیش‌بینی میزان خرج سالانه بسیار مرتبط نباشند، بنابراین می‌توانیم در نظر داشته باشیم که آن‌ها را حذف کنیم.

- مقیاس‌بندی ویژگی‌های عددی: مهم است که ویژگی‌های عددی را مقیاس‌بندی کنیم تا همه آن‌ها به طور مساوی در عملکرد مدل مشارکت کنند. ویژگی‌هایی مانند 'میانگین طول جلسات'، 'زمان استفاده از اپلیکیشن' و غیره، باید مقیاس‌بندی شوند. ما می‌توانیم از StandardScaler یا MinMaxScaler از sklearn.preprocessing استفاده کنیم.

- تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی: معمولاً داده‌ها به یک مجموعه آموزشی و یک مجموعه آزمایشی تقسیم می‌شوند. مجموعه آموزشی برای آموزش مدل و مجموعه آزمایشی برای ارزیابی عملکرد آن استفاده می‌شود.



```
# Splitting the data into training and testing sets with a 70%-30% split
X_train, X_test, y_train, y_test = train_test_split(features_scaled, label, test_size=0.3, random_state=101)

# Checking the shapes of the splits
print("Training Features Shape:", X_train.shape)
print("Testing Features Shape:", X_test.shape)
print("Training Labels Shape:", y_train.shape)
print("Testing Labels Shape:", y_test.shape)
```

```
Training Features Shape: (350, 4)
Testing Features Shape: (150, 4)
Training Labels Shape: (350,)
Testing Labels Shape: (150,)
```

با توجه به کد بالا خواسته مسئله را پیاده سازی میکنیم

(۵.۷)

```
from sklearn.linear_model import LinearRegression

# Create a linear regression model
lr_model = LinearRegression()

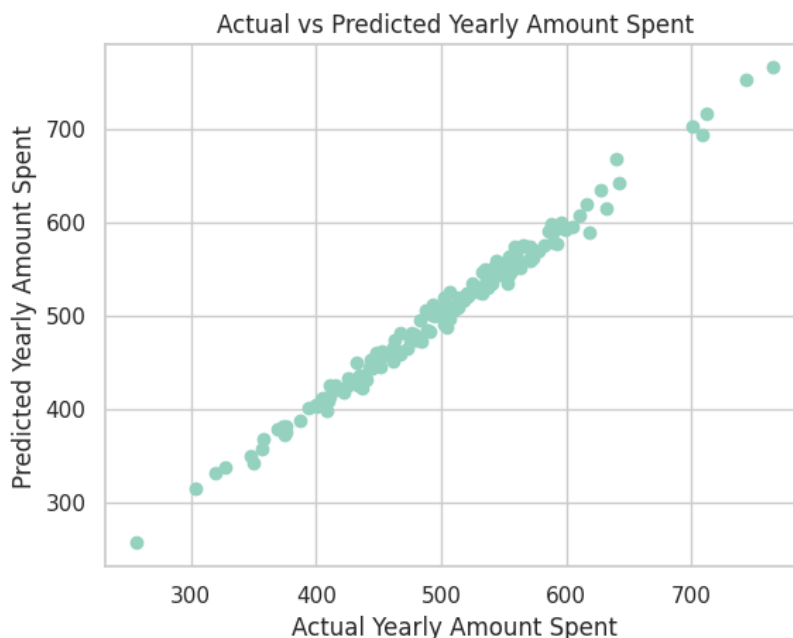
# Train the model on the training data
lr_model.fit(X_train, y_train)
```

با توجه به کد بالا خواسته مسئله را پیاده سازی میکنیم.

(۵.۸)

```
# Predicting the final purchase amount for the test data
y_pred = lr_model.predict(X_test)
```

(۵.۹)



- همبستگی: اغلب نقاط نزدیک به یک خط مستقیم قرار دارند، که نشان دهنده همبستگی قوی بین مقادیر واقعی و پیش‌بینی شده است.

- دقت پیش‌بینی: هرچه نقاط به خط $y = x$ (که معمولاً با یک خط مورب نشان داده می‌شود و نقاط در آن واقعی و پیش‌بینی شده برابر هستند) نزدیک‌تر باشند، دقت پیش‌بینی بالاتر است. در این نمودار، بیشتر نقاط نزدیک به این خط قرار دارند، که نشان دهنده دقت بالا در پیش‌بینی است.

- پراکندگی: اندکی پراکندگی در نقاط وجود دارد، که نشان دهنده خطاهای پیش‌بینی است.

- نقاط خارج از الگو: اگر نقاطی به طور قابل توجهی از خط اصلی فاصله داشته باشند، می‌توانند نشان دهنده نویز، داده‌های پرت یا مسائلی در داده‌ها باشند که باید بیشتر بررسی شوند.

به طور کلی، نمودار نشان می‌دهد که مدل رگرسیون خطی توانسته است به طور موثری میزان خرج سالانه مشتریان را پیش‌بینی کند، اگرچه برخی از نقاط خارج از الگو وجود دارند که می‌توانند فرصت‌هایی برای بهبود مدل فراهم کنند



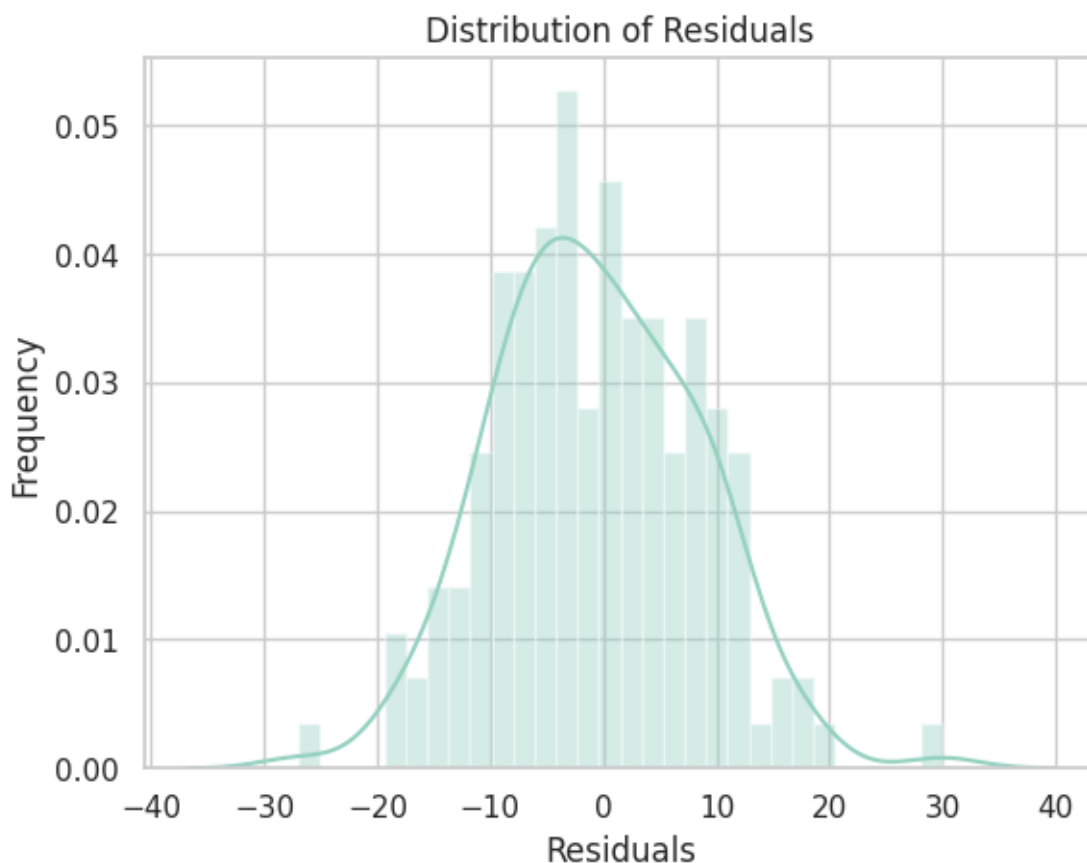
Mean Squared Error: 79.81305165097451
Root Mean Squared Error: 8.933815066978637
Mean Absolute Error: 7.228148653430832

خطای مربعات میانگین (MSE): نمایانگر میانگین مربعات خطاها است، یعنی میانگین تفاوت مربعات بین مقادیر تخمین زده شده و مقدار واقعی.

ریشه خطای مربعات میانگین (RMSE): این مقدار ریشه دوم MSE است. این معیار مفید است زیرا خطای اندازه‌گیری را به همان واحد متغیر هدف باز می‌گرداند.

خطای مطلق میانگین (MAE): نمایانگر میانگین تفاوت‌های مطلق بین مقادیر پیش‌بینی شده و مقادیر مشاهده شده است.

(۵.۱۱)



توزیع نرمال می‌تواند نشان دهد که مدل قادر به درک الگوهای موجود در داده‌ها بوده و توانسته است این الگوها را به درستی پیش‌بینی کند. توزیع نرمال نشان‌دهنده این است که خطاها (یعنی اختلاف بین مقادیر واقعی و



پیش‌بینی‌شده) به طور تصادفی اتفاق می‌افتند و هیچ الگوی منظم یا سیستماتیکی در خطاها وجود ندارد. این موضوع مهم است زیرا اگر خطاها الگومند باشند، این می‌تواند نشان‌دهنده این باشد که مدل ما از در نظر گرفتن برخی جنبه‌های مهم داده‌ها ناتوان بوده است، مانند وجود یک متغیر مستقل مهم که در مدل گنجانده نشده است. همچنین، توزیع نرمال باقی‌مانده‌ها می‌تواند به ما اطمینان دهد که هم‌پراکندگی وجود دارد، یعنی واریانس خطاها در سراسر محدوده مقادیر متغیرهای مستقل ثابت است. این موضوع به این معنا است که مدل ما به یک اندازه برای تمام نقاط داده‌ای دقیق است و اینکه دقت پیش‌بینی‌های مدل تحت تأثیر اندازه یا مقیاس متغیرهای مستقل قرار نمی‌گیرد.

در صورتی که توزیع باقی‌مانده‌ها (اختلاف‌های بین مقادیر واقعی و پیش‌بینی‌شده) از نرمالیت خارج باشد، این مسئله می‌تواند شاخص‌هایی را در مورد مشکلات احتمالی در مدل‌سازی یا داده‌ها فراهم آورد که به شرح زیر است:

- عدم خطی بودن داده‌ها: اگر رابطه بین متغیرهای مستقل و وابسته خطی نباشد و مدل رگرسیون خطی استفاده شود، باقی‌مانده‌ها ممکن است الگوهای منظمی نشان دهند که این نشان‌دهنده عدم مناسبت مدل خطی با داده‌ها است. در این حالت، استفاده از مدل‌های غیرخطی یا تبدیل متغیرها ممکن است لازم باشد تا رابطه‌ای مناسب‌تر بین متغیرهای مستقل و وابسته ایجاد شود.

- نقاط پرت: نقاطی که به شدت از الگوی کلی داده‌ها فاصله دارند می‌توانند باقی‌مانده‌های بزرگی ایجاد کنند و موجب ایجاد توزیعی با دم‌های سنگین در باقی‌مانده‌ها شوند. نقاط پرت می‌توانند نتیجه خطاهای اندازه‌گیری یا ویژگی‌های غیرمعمول در داده‌ها باشند که باید شناسایی و تصحیح یا حذف شوند.

- متغیرهای گم‌شده یا نادرست مدل‌سازی شده: اگر متغیرهای مهمی که تأثیر قابل توجهی بر متغیر وابسته دارند در مدل گنجانده نشوند یا به درستی مدل‌سازی نشوند، مدل نمی‌تواند تغییرات در متغیر وابسته را به درستی توضیح دهد. این موضوع می‌تواند منجر به توزیع باقی‌مانده‌هایی شود که نشان‌دهنده سیستماتیک بودن خطاها به جای تصادفی بودن آنها است.

در نتیجه، اگر توزیع باقی‌مانده‌ها نشان‌دهنده نرمالیت نباشد، این می‌تواند به عنوان یک سیگنال برای بررسی بیشتر و ارزیابی مجدد مدل و داده‌ها عمل کند. بررسی‌های بیشتر می‌تواند شامل تجزیه و تحلیل گرافیکی باقی‌مانده‌ها، تست‌های آماری برای نرمالیت، و اضافه کردن یا حذف کردن متغیرها از مدل باشد.



	Coefficient
Avg. Session Length	25.762527
Time on App	38.328552
Time on Website	0.192210
Length of Membership	61.173557

نرمال سازی داده ها برای جدولی که ضرایب مدل رگرسیونی را نشان می دهد اهمیت دارد زیرا:

۱. مقایسه پذیری ضرایب: زمانی که ویژگی ها در مقیاس های مختلفی هستند، ضرایب بزرگتر لزوماً به معنای اهمیت بیشتر نیستند. نرمال سازی این امکان را فراهم می کند که ضرایب را به صورت مستقیم با یکدیگر مقایسه کنیم و درک بهتری از اهمیت نسبی هر ویژگی در مدل داشته باشیم.

۲. تأثیر ویژگی ها بر مدل: بدون نرمال سازی، ویژگی هایی با مقیاس بزرگتر می توانند تأثیر نامتناسبی روی مدل داشته باشند، که می تواند منجر به تصمیم گیری های نادرست شود. برای مثال، اگر "زمان استفاده از وبسایت" دارای مقیاسی بسیار بزرگتر از "مدت زمان عضویت" باشد، ضریب کمتر "زمان استفاده از وبسایت" ممکن است نشان دهنده کم اهمیت بودن این ویژگی نباشد.

۳. بهینه سازی مدل: برخی الگوریتم های یادگیری ماشین برای کار کردن بهینه، به داده هایی با مقیاس مشابه نیاز دارند. نرمال سازی می تواند به همگرایی سریع تر الگوریتم های بهینه سازی کمک کند، که این مستقیماً بر سرعت و دقت یادگیری مدل تأثیر می گذارد.

۴. جلوگیری از بیش برازش: وقتی ویژگی ها بر اساس مقیاس اصلی شان متفاوت هستند، مدل ممکن است بیش از حد به ویژگی های با مقیاس بزرگتر وابسته شود، که این می تواند منجر به بیش برازش شود و عمومیت مدل را کاهش دهد.

به همین دلایل، نرمال سازی داده ها قبل از اعمال مدل رگرسیونی یک اقدام استاندارد و مهم است تا اطمینان حاصل شود که تمام ویژگی ها به طور عادلانه و منصفانه در تحلیل نهایی و تفسیر نتایج شرکت دارند.

(۵.۱۳)

ضریب مثبت: نشان می دهد که افزایش در این ویژگی با افزایش 'میزان خرج سالانه' همراه است. یک ضریب مثبت بزرگتر نشان دهنده یک رابطه قوی تر است.



ضریب منفی: نشان می‌دهد که افزایش در این ویژگی با کاهش 'میزان خرج سالانه' همراه است. یک ضریب منفی بزرگ‌تر نشان‌دهنده یک رابطه معکوس قوی‌تر است.

ویژگی‌هایی با بزرگترین ضرایب مثبت مناطقی هستند که سرمایه‌گذاری در آن‌ها می‌تواند بیشترین افزایش را در خرج سالانه مشتریان به دنبال داشته باشد.

با توجه به ضرایب به دست آمده از مدل رگرسیون خطی، به نظر می‌رسد که "زمان استفاده از وبسایت" تأثیر اندکی بر "میزان خرج سالانه" دارد، در حالی که "زمان استفاده از اپلیکیشن" ارتباط قوی‌تری نشان می‌دهد. از این رو، درآمد حاصل از اپلیکیشن به نظر مهم‌تر می‌رسد. با این حال، به جای اینکه کاملاً وبسایت را کنار بگذاریم، شرکت می‌تواند با بهبود وبسایت تلاش کند تا درآمد خود را از این کانال افزایش دهد. سپس، تجزیه و تحلیل را دوباره انجام دهد. با این حال، مهم‌ترین متغیری که بر میزان خرج تأثیر می‌گذارد، "طول مدت عضویت" است.

در واقع، این ضرایب به ما می‌گویند که مشتریانی که برای مدت طولانی‌تری عضو هستند، تمایل دارند سالانه مبلغ بیشتری خرج کنند. این امر می‌تواند نشان‌دهنده وفاداری و رضایت مشتری باشد که طی زمان به دست آمده است. بنابراین، سرمایه‌گذاری بر روی افزایش طول عضویت، مانند برنامه‌های وفاداری و بهبود تجربه کاربری، می‌تواند بازدهی بالایی داشته باشد.

از سوی دیگر، حتی اگر وبسایت تأثیر کمتری دارد، هنوز هم می‌تواند به عنوان یک ابزار مهم برای جذب مشتری و کمک به توسعه تجربه برند عمل کند. بهبود در وبسایت می‌تواند شامل بهینه‌سازی طراحی برای افزایش تعامل کاربر، سرعت بارگذاری صفحه و ایجاد محتوای جذاب‌تر باشد. این تغییرات می‌تواند تأثیر وبسایت را بر میزان خرج سالانه افزایش دهد و ممکن است باعث شود که مشتریان بیشتری از طریق وبسایت خرید کنند.

در نهایت، با انجام بهبودهایی در هر دو جنبه—اپلیکیشن و وبسایت—و همچنین با تمرکز بر روی افزایش طول مدت عضویت، شرکت می‌تواند تلاش کند تا درآمد خود را به طور چشمگیری افزایش دهد. از این طریق، می‌توان پس از اجرای این تغییرات و جمع‌آوری داده‌های جدید، تجزیه و تحلیل‌ها را مجدداً انجام داد تا اثربخشی استراتژی‌های اتخاذ شده را ارزیابی کرد.

پاسخ ۶.

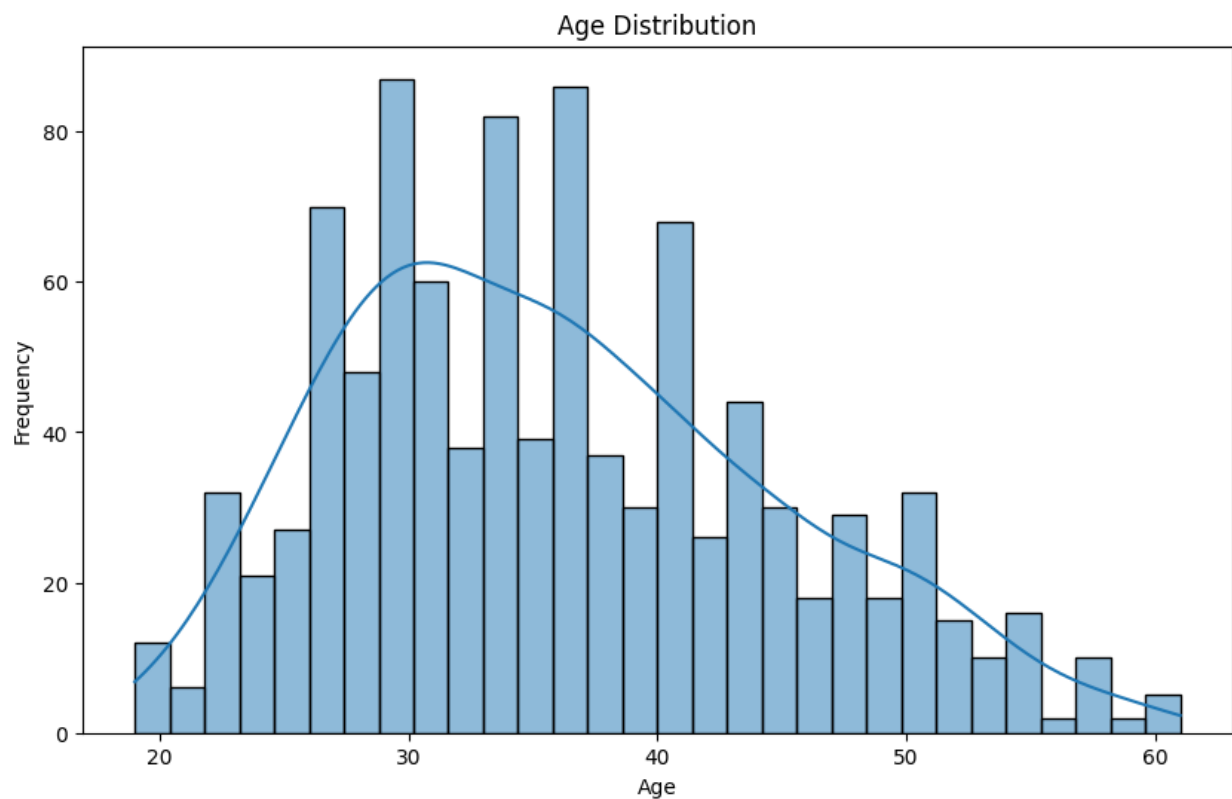
(۶.۱)

نتایج به شرح زیر است :



	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	65.000200	36.009000	55000.000080	180.000100	0.481000	0.500000
std	15.853615	8.785562	13414.634022	43.902339	0.499889	0.500250
min	32.600000	19.000000	13996.500000	104.780000	0.000000	0.000000
25%	51.360000	29.000000	47031.802500	138.830000	0.000000	0.000000
50%	68.215000	35.000000	57012.300000	183.130000	0.000000	0.500000
75%	78.547500	42.000000	65470.635000	218.792500	1.000000	1.000000
max	91.430000	61.000000	79484.800000	269.960000	1.000000	1.000000

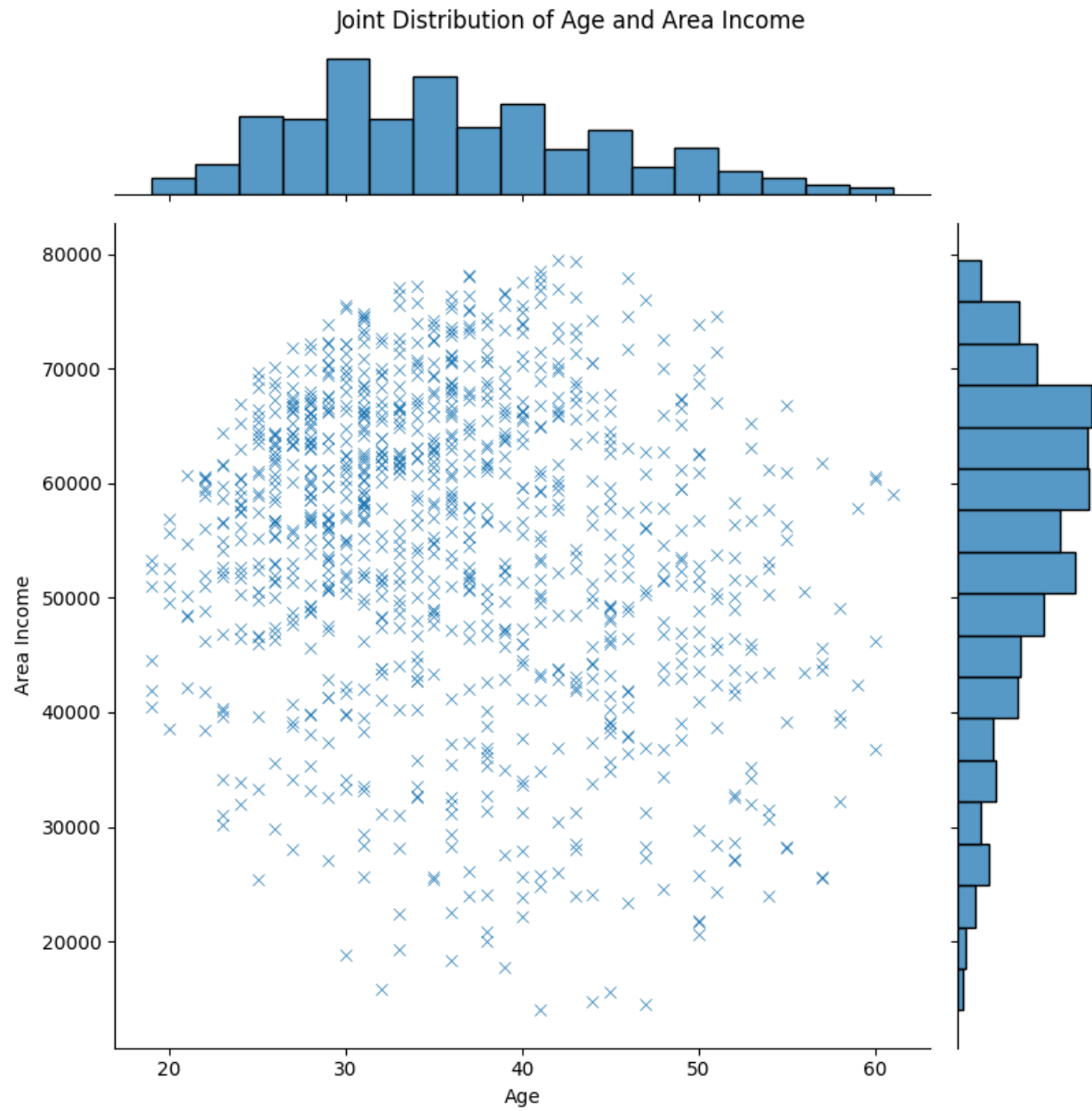
(۶.۲)



نمودار توزیع آماری برحسب سن به شرح بالا است

(۶.۳)

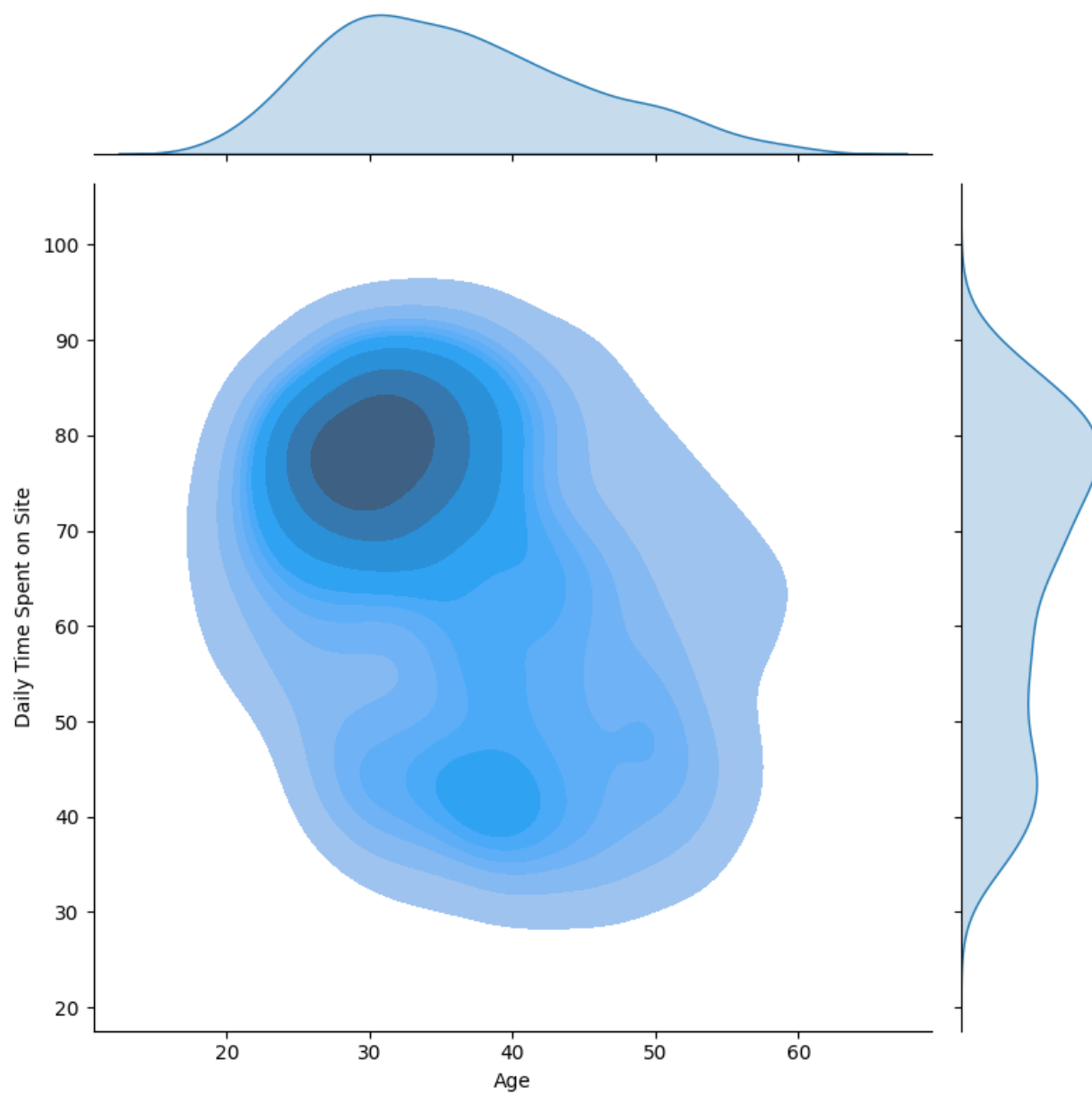
جوینت پلات درآمد برحسب سن به شرح زیر میباشد

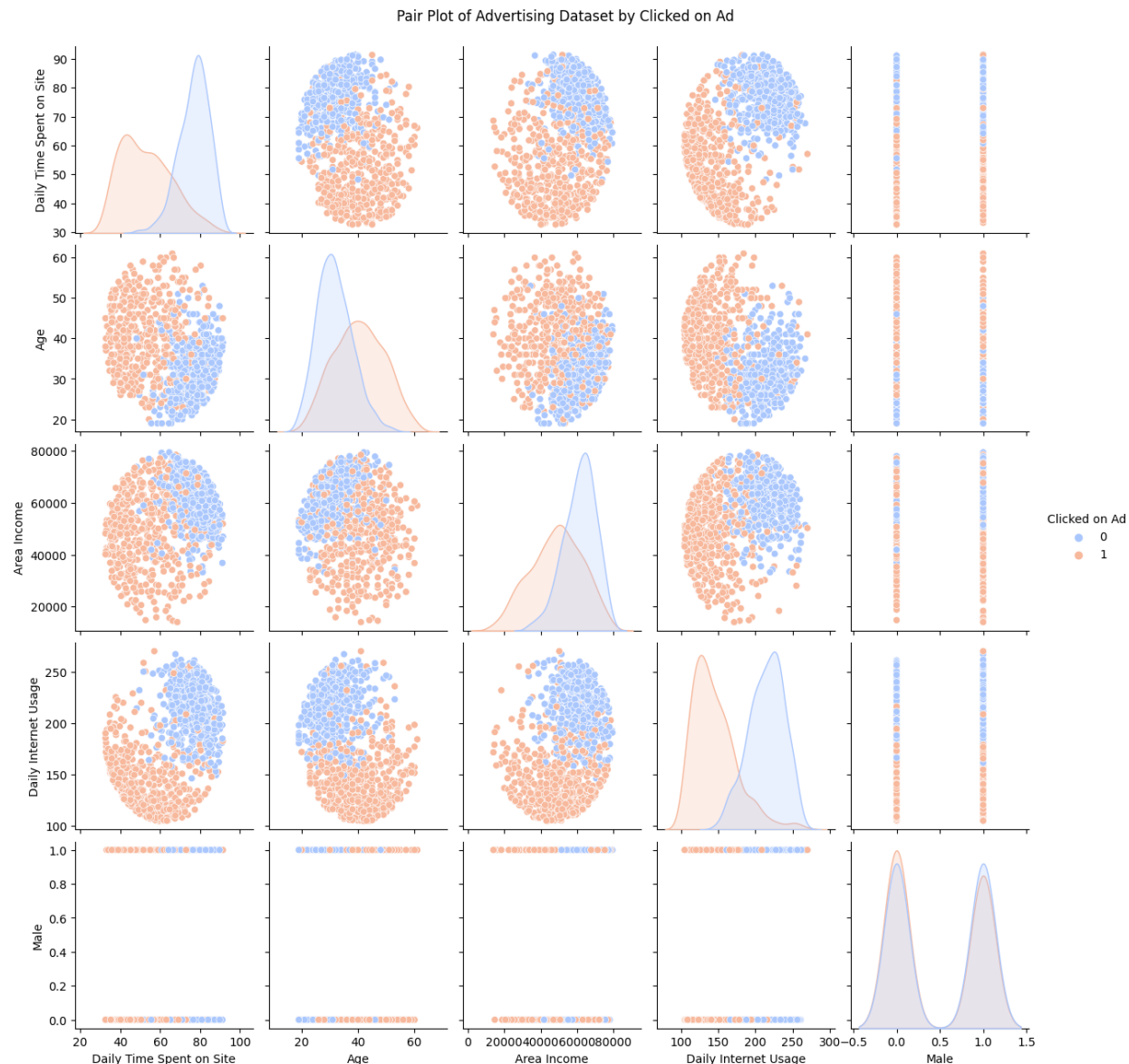


۶.۴ KDE زمان گزاشته شده بر روی سایت بر حسب سن به شرح زیر میباشد



KDE Plot of Daily Time Spent on Site by Age





(۶.۶)

۱. نمودار Pair Plot:

- این نمودار به ما نشان می‌دهد که چگونه متغیرهای مختلف با یکدیگر و با متغیر هدف، که در اینجا کلیک بر روی تبلیغات است، ارتباط دارند.

- وجود دو رنگ نشان‌دهنده دو دسته‌بندی بر اساس کلیک یا عدم کلیک بر روی تبلیغات است.

- می‌توان دید که تفاوت‌های واضحی بین دو دسته در زمینه‌هایی مانند "زمان روزانه صرف شده در سایت"، "سن"، "درآمد منطقه" و "میزان استفاده روزانه از اینترنت" وجود دارد.



۲. نمودار KDE (Kernel Density Estimate) برای زمان صرف شده روزانه در سایت بر اساس سن:

- این نمودار می‌تواند نشان دهد که گروه‌های سنی مختلف چگونه زمان خود را در سایت می‌گذرانند.

- مثلاً ممکن است نشان دهد که افراد جوان‌تر زمان بیشتری را در سایت صرف می‌کنند، که می‌تواند برای تبلیغات هدفمند مفید باشد.

۳. نمودار توزیع مشترک برای سن و درآمد منطقه:

- این نمودار پراکندگی افراد را بر اساس سن و درآمد نشان می‌دهد و می‌تواند برای شناسایی الگوهای خاص در میان داده‌ها مفید باشد.

- برای مثال، ممکن است مشخص شود که افراد با درآمد بالاتر یا گروه‌های سنی خاصی بیشتر تمایل به کلیک بر تبلیغات دارند.

۴. نمودار توزیع سن:

- این نمودار توزیع فرکانس سنی را نشان می‌دهد و می‌تواند در شناسایی دموگرافیک اصلی کاربران سایت کمک کند.

- همچنین می‌تواند برای تعیین استراتژی‌های بازاریابی و تبلیغاتی مبتنی بر سن مورد استفاده قرار گیرد.

(۶.۷)

برای آماده‌سازی مجموعه داده‌های تبلیغاتی برای یک مدل یادگیری ماشین، نیاز به انجام چندین مرحله پیش‌پردازش داده داریم.

۱. برخورد با مقادیر گمشده: برای هرگونه مقادیر گمشده در مجموعه داده‌ها باید جستجو کرد. اگر مقادیر گمشده‌ای یافت شدند، می‌توانید یا آن‌ها را با مقدار مناسبی پر کنید (مانند میانگین یا میانه برای داده‌های عددی، یا مُد برای داده‌های دسته‌بندی شده) یا سطرها/ستون‌های دارای مقادیر گمشده را حذف کنید.

۲. کدگذاری ویژگی‌ها: اگر مجموعه داده‌ها شامل متغیرهای دسته‌بندی باشد، باید آن‌ها را به فرمت عددی تبدیل کنید. این کار می‌تواند با روش‌هایی مانند کدگذاری یک‌به‌یک یا کدگذاری برچسب انجام شود.



۳. مقیاس‌بندی ویژگی‌ها: ویژگی‌های عددی باید مقیاس‌بندی شوند تا اطمینان حاصل شود که تمام ویژگی‌ها به یک اندازه در عملکرد مدل مشارکت می‌کنند. روش‌های رایج عبارتند از نرمال‌سازی (مقیاس‌بندی مقادیر بین ۰ تا ۱) و استانداردسازی (مقیاس‌بندی مقادیر به گونه‌ای که میانگین ۰ و انحراف معیار ۱ داشته باشند).

۴. انتخاب ویژگی: ویژگی‌هایی که بیشترین ارتباط را با کاری که قصد پیش‌بینی آن را داریم، شناسایی و انتخاب کنید. این می‌تواند بر اساس دانش حوزه، تحلیل همبستگی یا با استفاده از تکنیک‌های انتخاب ویژگی باشد.

۵. تقسیم داده‌ها به ویژگی‌ها (X) و برچسب‌ها (y): مجموعه داده‌ها را به 'ویژگی‌ها' (متغیرهای مستقل استفاده شده برای پیش‌بینی) و 'برچسب‌ها' (متغیر وابسته‌ای که می‌خواهید پیش‌بینی کنید) تقسیم کنیم. در این مورد، برچسب می‌تواند ستون 'کلیک بر روی تبلیغ' باشد.

۶. تقسیم آموزش-آزمون: در نهایت، مجموعه داده‌ها را به یک مجموعه آموزشی و یک مجموعه آزمون تقسیم کنید. یک نسبت تقسیم رایج ۸۰٪ برای آموزش و ۲۰٪ برای آزمون است. این امر به مدل اجازه می‌دهد تا روی یک قسمت از مجموعه داده‌ها آموزش داده شود و روی یک قسمت دیده نشده برای ارزیابی عملکرد آزمایش شود. ۶.۸ ساختن یک مدل رگرسیون لجستیک از ابتدا شامل پیاده‌سازی دستی الگوریتم رگرسیون لجستیک است، بدون تکیه بر کتابخانه‌های آماده مانند scikit-learn. در اینجا یک بررسی اجمالی از مراحل درگیر شده است:

۱. تابع سیگموئید: مدل رگرسیون لجستیک از یک تابع سیگموئید برای پیش‌بینی احتمال اینکه یک ورودی معین به یک کلاس خاص تعلق داشته باشد استفاده می‌کند.

۲. تابع هزینه: تابع هزینه‌ای که در رگرسیون لجستیک استفاده می‌شود، انتروپی متقابل دودویی یا لاگ لاس است، که عملکرد یک مدل دسته‌بندی را اندازه‌گیری می‌کند.

۳. گرادیان کاهشی: این یک الگوریتم بهینه‌سازی است که برای کمینه کردن تابع هزینه با تنظیم تکراری وزن‌ها استفاده می‌شود.

۴. آموزش مدل: مدل با تنظیم وزن‌های خود بر اساس الگوریتم گرادیان کاهشی یاد می‌گیرد.

۵. پیش‌بینی‌ها: پس از آموزش، مدل با استفاده از وزن‌های آموخته شده برای ویژگی‌ها، پیش‌بینی‌ها را انجام می‌دهد.

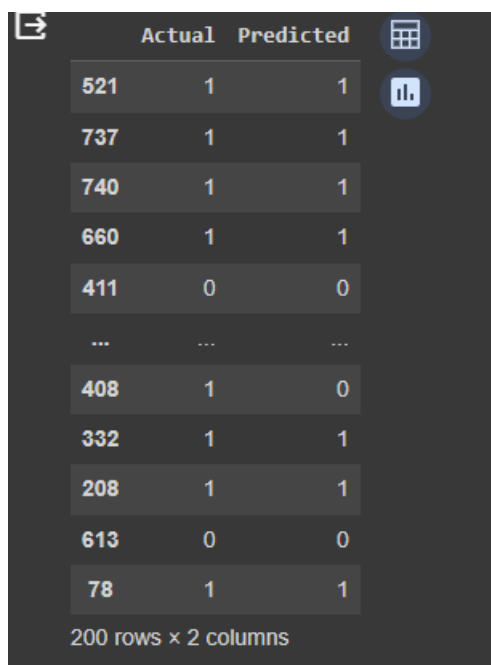


ما با تعریف تابع سیگموئید و تابع هزینه شروع می‌کنیم. سپس، الگوریتم گرادیان کاهشی را پیاده‌سازی کرده و از آن برای آموزش مدل روی داده‌های آموزشی استفاده می‌کنیم.

مراحل رگرسیون لجستیک به شرح زیر است :

۱. مقدمه‌سازی: مدل با نرخ یادگیری ۰.۰۱ شروع شد و تنظیم شد تا ۱۰۰۰ بار برای فرآیند آموزش تکرار شود.
 ۲. تابع سیگموئید: برای محاسبه احتمال نتیجه دودویی استفاده شد.
 ۳. تابع هزینه: برای محاسبه هزینه در حین آموزش، از انتروپی متقابل دودویی استفاده شد.
 ۴. گرادیان کاهشی: این الگوریتم بهینه‌سازی برای به‌روزرسانی وزن‌ها و بایاس در هر تکرار، جهت کمینه کردن تابع هزینه به کار رفت.
 ۵. آموزش: مدل بر روی مجموعه داده‌های آموزشی داده شده آموزش دید.
 ۶. پیش‌بینی: سپس از مدل آموزش دیده برای انجام پیش‌بینی‌ها روی مجموعه آزمون استفاده شد.
- پیش‌بینی‌ها مقادیر دودویی هستند (۱ یا ۰) که نشان می‌دهند آیا کاربر پیش‌بینی شده است که روی تبلیغ کلیک کند یا نه. ده پیش‌بینی اول از مجموعه آزمون به عنوان نمونه نمایش داده شده‌اند.
- (۶.۹)

با استفاده از کد زده شده به نتایج زیر می‌رسیم



	Actual	Predicted
521	1	1
737	1	1
740	1	1
660	1	1
411	0	0
...
408	1	0
332	1	1
208	1	1
613	0	0
78	1	1

200 rows x 2 columns



(۶.۱۰)

```
Confusion Matrix:
[[87  2]
 [12 99]]

Classification Report:
              precision    recall  f1-score   support

     0       0.88        0.98        0.93         89
     1       0.98        0.89        0.93        111

 accuracy          0.93
 macro avg         0.93
 weighted avg      0.94

Accuracy Score: 0.93
```

۱. ماتریس اشفستگی:

- این ماتریس نشان می‌دهد که مدل چگونه بر روی داده‌های آزمونی عمل کرده است.
- در سمت چپ بالای ماتریس، ۸۷ نمونه به درستی به عنوان کلاس ۰ پیش‌بینی شده‌اند (True Negative).
- در سمت راست پایین ماتریس، ۹۹ نمونه به درستی به عنوان کلاس ۱ پیش‌بینی شده‌اند (True Positive).
- در سمت راست بالای ماتریس، ۲ نمونه به اشتباه به عنوان کلاس ۱ پیش‌بینی شده‌اند (False Positive).
- در سمت چپ پایین ماتریس، ۱۲ نمونه به اشتباه به عنوان کلاس ۰ پیش‌بینی شده‌اند (False Negative).

۲. گزارش دسته‌بندی:

- دقت (Precision): نشان‌دهنده توانایی مدل در این است که تعداد پیش‌بینی‌های درست از یک کلاس را از تعداد کل پیش‌بینی‌هایی که برای آن کلاس انجام شده است، محاسبه کند.
- بازیابی (Recall): نشان‌دهنده توانایی مدل در پیدا کردن تمام نمونه‌های مرتبط با یک کلاس است.
- نمره FI (FI-Score): میانگین هارمونیک دقت و بازیابی است که تعادل بین دقت و بازیابی را نشان می‌دهد.
- پشتیبانی (Support): تعداد نمونه‌های واقعی برای هر کلاس در مجموعه داده‌های آزمون.

۳. نمره دقت کلی (Accuracy Score):

- دقت کلی مدل ۰.۹۳ است، که نشان‌دهنده عملکرد خوب مدل در دسته‌بندی است.



۱. بارگذاری داده‌ها: ابتدا با بارگذاری مجموعه داده‌های `spamSMS` شروع می‌کنیم.

۲. تحلیل توزیع برچسب‌ها: توزیع برچسب‌ها در مجموعه داده‌ها را تحلیل می‌کنیم.

```
ham      4825  
spam      747  
Name: v1, dtype: int64
```

۳. استخراج ویژگی با استفاده از CountVectorizer: از `CountVectorizer` برای تبدیل متون پیامک‌ها به فرمت عددی که الگوریتم‌های یادگیری ماشین می‌توانند پردازش کنند، استفاده می‌کنیم.

۴. تقسیم داده‌ها به داده‌های آموزشی و تست: مجموعه داده‌ها را با نسبت ۷۰ به ۳۰ به داده‌های آموزشی و تست تقسیم می‌کنیم.

۵. آموزش مدل با جستجوی شبکه‌ای و جستجوی تصادفی:

- جستجوی شبکه‌ای (Grid Search): روشی سیستماتیک برای تنظیم هایپرپارامترها، که هر ترکیب ممکن از مقادیر هایپرپارامتر داده شده را آزمایش می‌کند.

- جستجوی تصادفی (Random Search): روشی تصادفی برای تنظیم هایپرپارامترها که تعداد مشخصی از ترکیب‌های هایپرپارامتر را از محدوده‌های مشخص شده نمونه‌برداری می‌کند.



```
[CV 3/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.974 total time= 0.3s
[CV 4/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.981 total time= 0.3s
[CV 5/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.981 total time= 0.3s
[CV 1/5] END .....C=10, gamma=0.01, kernel=rbf;, score=0.990 total time= 0.5s
[CV 2/5] END .....C=10, gamma=0.01, kernel=rbf;, score=0.985 total time= 0.5s
[CV 3/5] END .....C=10, gamma=0.01, kernel=rbf;, score=0.977 total time= 0.5s
[CV 4/5] END .....C=10, gamma=0.01, kernel=rbf;, score=0.982 total time= 0.5s
[CV 5/5] END .....C=10, gamma=0.01, kernel=rbf;, score=0.978 total time= 0.4s
[CV 1/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.988 total time= 0.3s
[CV 2/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.985 total time= 0.3s
[CV 3/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.974 total time= 0.3s
[CV 4/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.981 total time= 0.3s
[CV 5/5] END ...C=10, gamma=0.01, kernel=linear;, score=0.981 total time= 0.3s
[CV 1/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.986 total time= 0.5s
[CV 2/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.976 total time= 0.4s
[CV 3/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.967 total time= 0.4s
[CV 4/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.973 total time= 0.4s
[CV 5/5] END .....C=10, gamma=0.001, kernel=rbf;, score=0.976 total time= 0.5s
[CV 1/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.988 total time= 0.3s
[CV 2/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.985 total time= 0.3s
[CV 3/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.974 total time= 0.3s
[CV 4/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.981 total time= 0.3s
[CV 5/5] END ..C=10, gamma=0.001, kernel=linear;, score=0.981 total time= 0.3s
[CV 1/5] END .....C=100, gamma=1, kernel=rbf;, score=0.890 total time= 2.3s
[CV 2/5] END .....C=100, gamma=1, kernel=rbf;, score=0.881 total time= 3.0s
[CV 3/5] END .....C=100, gamma=1, kernel=rbf;, score=0.882 total time= 3.4s
[CV 4/5] END .....C=100, gamma=1, kernel=rbf;, score=0.888 total time= 1.9s
[CV 5/5] END .....C=100, gamma=1, kernel=rbf;, score=0.894 total time= 1.9s
[CV 1/5] END .....C=100, gamma=1, kernel=linear;, score=0.988 total time= 0.5s
[CV 2/5] END .....C=100, gamma=1, kernel=linear;, score=0.985 total time= 0.5s
[CV 3/5] END .....C=100, gamma=1, kernel=linear;, score=0.974 total time= 0.4s
[CV 4/5] END .....C=100, gamma=1, kernel=linear;, score=0.981 total time= 0.4s
[CV 5/5] END .....C=100, gamma=1, kernel=linear;, score=0.981 total time= 0.3s
[CV 1/5] END .....C=100, gamma=0.1, kernel=rbf;, score=0.973 total time= 1.1s
```

نمونه ای از اجرای مسأله بالا به صورت عکس بالا است.

۶. مقایسه جستجوی شبکه‌ای و جستجوی تصادفی:

جستجوی شبکه‌ای (Grid Search)

مزایا:

۱. جامعیت: جستجوی شبکه‌ای جامع و دقیق است، زیرا از تمام ترکیب‌های ممکن هایپرپارامترها در شبکه جستجو می‌کند. این امر شانس یافتن بهترین ترکیب را افزایش می‌دهد.

۲. سادگی: از نظر مفهومی ساده است و به آسانی پیاده‌سازی می‌شود.

۳. قابلیت تکرارپذیری: به دلیل طبیعت تعیین‌کننده‌اش، جستجوی شبکه‌ای هر بار برای همان مجموعه داده و هایپرپارامترها نتایج یکسانی تولید می‌کند، که این امر تکرارپذیری را تضمین می‌کند.

معایب:



۱. شدت محاسباتی: می‌تواند بسیار وقت‌گیر باشد، به خصوص با تعداد زیادی از هایپرپارامترها و محدوده‌های گسترده مقادیر، زیرا هر ترکیب در شبکه را ارزیابی می‌کند.

۲. عدم قابلیت مقیاس‌پذیری: با اضافه شدن هر هایپرپارامتر، زمان مورد نیاز به طور نمایی افزایش می‌یابد، که این امر آن را برای مشکلات با تعداد زیادی از هایپرپارامترها کمتر عملی می‌کند.

۳. محدود به شبکه تعریف شده: جستجوی شبکه‌ای فقط هایپرپارامترهایی را که از قبل در شبکه تعریف شده‌اند ارزیابی می‌کند. اگر مقادیر بهینه بین نقاط شبکه قرار داشته باشند، ممکن است از دست بروند.

جستجوی تصادفی (Random Search)

مزایا:

۱. کارآمدی: جستجوی تصادفی می‌تواند برای تعداد زیادی از هایپرپارامترها یا زمانی که مقادیر بهینه در یک محدوده باریک قرار دارند، کارآمدتر باشد، زیرا تعداد مشخصی از ترکیب‌ها را از فضای هایپرپارامتر نمونه‌برداری می‌کند.

۲. قابلیت مقیاس‌پذیری: معمولاً سریع‌تر و قابل مقیاس‌پذیری برای فضاهای بعدی بالاتر از جستجوی شبکه‌ای است.

۳. پتانسیل یافتن مقادیر بهینه: امکان یافتن مقادیر بهینه خارج از یک شبکه ثابت وجود دارد، زیرا از توزیع پیوسته نمونه‌برداری می‌کند.

معایب:

۱. تصادفی بودن: تصادفی بودن به این معناست که ممکن است نقاط مهمی در فضای هایپرپارامتر از دست بروند. این روش جامع نیست و ممکن است ترکیب بهینه را پیدا نکند.

۲. قابلیت تکرارپذیری: مگر اینکه دانه تصادفی ثابت شود، هر بار که اجرا شود نتایج متفاوتی تولید می‌کند، که می‌تواند برای تکرارپذیری مشکل‌ساز باشد.

۳. تنظیم دقیق: ممکن است نیاز به تنظیم دقیق‌تر یا تعداد بیشتری از تکرارها برای رسیدن به سطوح دقت جستجوی شبکه‌ای داشته باشد.



- جستجوی شبکه‌ای برای مسائل با تعداد نسبتاً کمی از هایپرپارامترها و زمانی که منابع محاسباتی مسئله اصلی نیستند، مناسب است.

- جستجوی تصادفی برای فضاهای هایپرپارامتری با بعد بالا یا زمانی که جستجوی سریع و کم هزینه‌تر محاسباتی نیاز است، مناسب‌تر است.

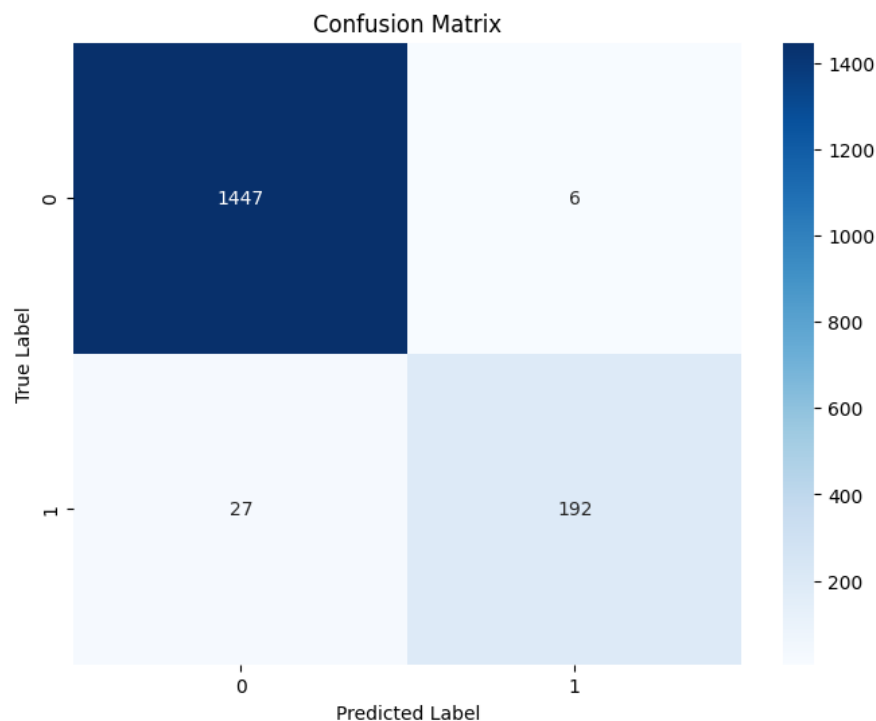
در عمل، می‌توان از ترکیب هر دو استفاده کرد: جستجوی تصادفی برای محدود کردن محدوده هایپرپارامترها و سپس جستجوی شبکه‌ای برای تنظیم دقیق در آن محدوده.

۷. آموزش و ارزیابی مدل‌ها: با استفاده از هر دو هسته خطی و RBF یک دسته‌بند SVM، مدل‌ها را آموزش می‌دهیم و مقادیر مختلفی برای 'C' و 'gamma' را آزمایش می‌کنیم.

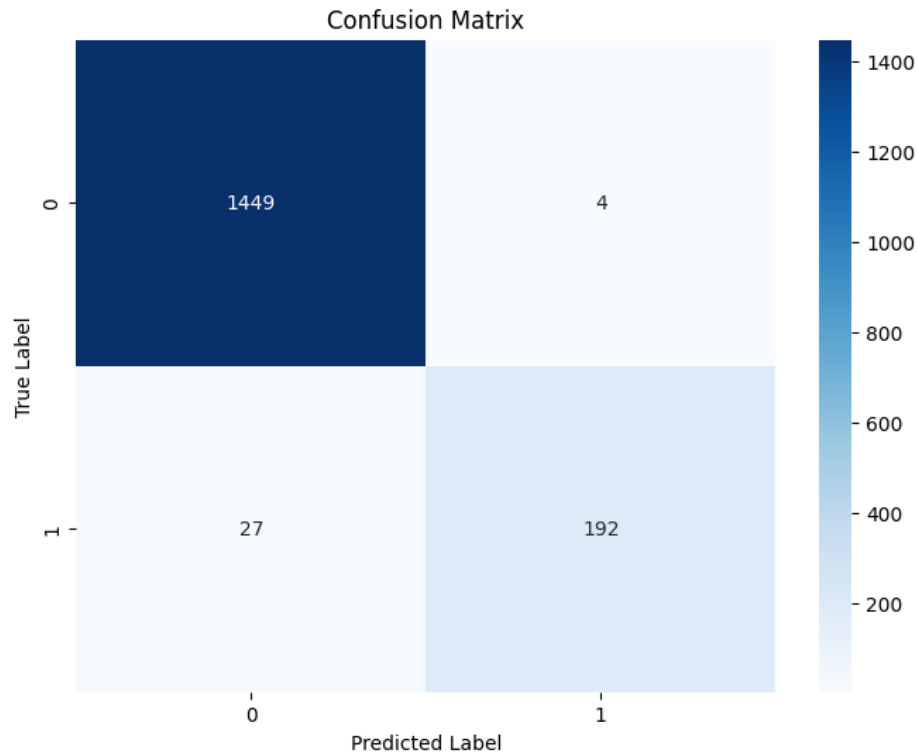
۸. تحلیل نتایج:

- ارزیابی مدل‌ها با استفاده از ماتریس اشتباه و گزارش دقت بر روی داده‌های تست.

Grid Search Model Evaluation:



Random Search Model Evaluation:



در هر دو ماتریس اشفستگی ، اعداد به شرح زیر هستند:

- عدد بالا و سمت چپ (True Negatives): پیش‌بینی‌های صحیح که نمونه‌ای را در کلاس مثبت (مثلاً اسپم نبودن) قرار نداده‌اند.
- عدد پایین و سمت راست (True Positives): پیش‌بینی‌های صحیح که نمونه‌ای را در کلاس مثبت (مثلاً اسپم بودن) قرار داده‌اند.
- عدد بالا و سمت راست (False Positives): پیش‌بینی‌های نادرست که نمونه‌ای را به اشتباه در کلاس مثبت قرار داده‌اند.
- عدد پایین و سمت چپ (False Negatives): پیش‌بینی‌های نادرست که نمونه‌ای را به اشتباه در کلاس منفی قرار داده‌اند.



- True Positives (TP): هر دو مدل به درستی ۱۹۲ بار کلاس مثبت را پیش‌بینی کرده‌اند.

- False Negatives (FN): هر دو مدل تعداد یکسانی از پیش‌بینی‌های نادرست منفی (۲۷) دارند، به این معنا که ۲۷ مورد که مثبت بوده‌اند اشتباهاً به عنوان منفی پیش‌بینی شده‌اند.

- True Negatives (TN): مدل جستجوی تصادفی کمی بیشتر از مدل جستجوی شبکه‌ای True Negatives دارد (۱۴۴۹ در مقابل ۱۴۴۷)، به این معنا که به درستی تعداد بیشتری از نمونه‌های کلاس منفی را پیش‌بینی کرده است.

- False Positives (FP): مدل جستجوی تصادفی تعداد کمتری False Positives نسبت به مدل جستجوی شبکه‌ای دارد (۴ در مقابل ۶)، که نشان می‌دهد اشتباهات کمتری در پیش‌بینی نادرست کلاس منفی به عنوان مثبت داشته است.

از نظر عملکرد کلی، به نظر می‌رسد هر دو مدل به خوبی عمل کرده‌اند، با این حال مدل جستجوی تصادفی به دلیل داشتن تعداد کمتری False Positives کمی برتری دارد. با این حال، تفاوت بسیار ناچیز است و سایر عوامل مانند زمان جستجو، پیچیدگی مدل، و قابلیت تفسیر ممکن است در انتخاب بین استفاده از جستجوی تصادفی یا جستجوی شبکه‌ای برای تنظیم هایپرپارامترها تأثیر بگذارند.

پاسخ ۸.

این مجموعه داده که برای پیش‌بینی قیمت خانه استفاده می‌شود، دارای ۱۴۶۰ داده با ۸۱ ویژگی مختلف می‌باشد. هر کدام از این ویژگی‌ها به شرح زیر می‌باشند:

- MSSubClass: کلاس ساختمان
- MSZoning: طبقه‌بندی کلی منطقه
- LotFrontage: پایانه‌های خطی خیابان متصل به ملک
- LotArea: اندازه زمین به فوت مربع
- Street: نوع دسترسی به جاده
- Alley: نوع دسترسی به کوچه
- LotShape: شکل عمومی ملک



- LandContour: صافی زمین
- Utilities: نوع امکانات موجود
- LotConfig: پیکربندی زمین
- LandSlope: شیب زمین
- Neighborhood: موقعیت‌های فیزیکی در محدوده شهر آمل
- ConditionI: نزدیکی به جاده اصلی یا راه‌آهن
- Condition2: نزدیکی به جاده اصلی یا راه‌آهن (در صورت وجود دومین)
- BldgType: نوع مسکن
- HouseStyle: سبک مسکن
- OverallQual: کیفیت کلی مواد و تمامیت
- OverallCond: امتیاز شرایط کلی
- YearBuilt: تاریخ ساخت اصلی
- YearRemodAdd: تاریخ بازسازی
- RoofStyle: نوع سقف
- RoofMatl: مواد سقف
- Exterior1st: پوشش بیرونی خانه
- Exterior2nd: پوشش بیرونی خانه (در صورت استفاده از بیش از یک ماده)
- MasVnrType: نوع سنگ مصنوعی
- MasVnrArea: مساحت سنگ مصنوعی به فوت مربع
- ExterQual: کیفیت مواد بیرونی
- ExterCond: شرایط حال حاضر مواد بیرونی
- Foundation: نوع بنیاد



- BsmQual: ارتفاع زیرزمین
- BsmCond: شرایط عمومی زیرزمین
- BsmExposure: دیوارهای زیرزمین با دسترسی گذرای یا باغ
- BsmFinType1: کیفیت منطقه پایانی زیرزمین نهایی
- BsmFinSF1: فوت مربع پایان دادن به نوع ۱
- BsmFinType2: کیفیت منطقه دوم پایان داده شده (در صورت وجود)
- BsmFinSF2: فوت مربع پایان داده شده نوع ۲
- BsmUnfSF: فوت مربع ناتمام زیرزمین
- TotalBsmSF: مساحت کل فوت مربع زیرزمین
- Heating: نوع گرمایش
- HeatingQC: کیفیت و شرایط گرمایش
- CentralAir: تهویه مرکزی
- Electrical: سیستم برق
- stFlrSF1: فوت مربع طبقه اول
- ndFlrSF2: فوت مربع طبقه دوم
- LowQualFinSF: فوت مربع پایین کیفیت تمام شده (تمام طبقات)
- GrLivArea: مساحت زندگی بالای زمین (طبقه اصلی)
- BsmFullBath: حمام‌های کامل زیرزمین
- BsmHalfBath: حمام‌های نیمه زیرزمین
- FullBath: حمام‌های کامل بالای سطح
- HalfBath: حمام‌های نیمه بالای سطح
- Bedroom: تعداد اتاق‌های خواب بالای سطح زیرزمین



- Kitchen: تعداد آشپزخانه‌ها
- KitchenQual: کیفیت آشپزخانه
- TotRmsAbvGrd: تعداد اتاق‌های بالای سطح (به جز حمام‌ها)
- Functional: امتیاز کارایی خانه
- Fireplaces: تعداد شومینه‌ها
- FireplaceQu: کیفیت شومینه
- GarageType: محل گاراژ
- GarageYrBlt: سال ساخت گاراژ
- GarageFinish: تمامیت داخلی گاراژ
- GarageCars: اندازه گاراژ به ظرفیت ماشین
- GarageArea: اندازه گاراژ به فوت مربع
- GarageQual: کیفیت گاراژ
- GarageCond: شرایط گاراژ
- PavedDrive: مسیرروبه‌ای سنگفرش
- WoodDeckSF: مساحت دک خرده ای در فوت مربع
- OpenPorchSF: مساحت تراس باز در فوت مربع
- EnclosedPorch: مساحت تراس بسته در فوت مربع
- SsnPorch۳: مساحت تراس سه فصل در فوت مربع
- ScreenPorch: مساحت تراس صفحه نمایش در فوت مربع
- PoolArea: مساحت استخر در فوت مربع
- PoolQC: کیفیت استخر
- Fence: کیفیت حصار



- MiscFeature: ویژگی متفرقه که در دسته‌های دیگر تحت پوشش قرار نگرفته است
- MiscVal: \$ارزش ویژگی‌های متفرقه
- MoSold: ماه فروش
- YrSold: سال فروش
- SaleType: نوع فروش
- SaleCondition: شرایط فروش

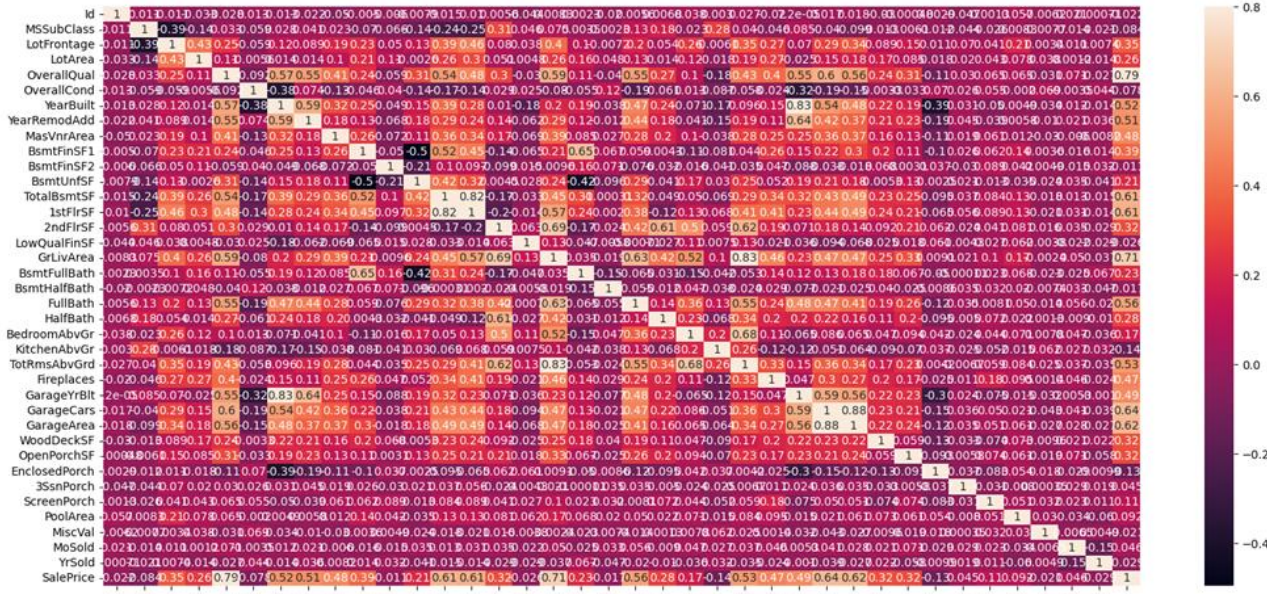
برای درست کردن یک مدل ابتدا نیاز است ویژگی‌ها بررسی شوند و مقادیر و ویژگی‌های اشتباه و بی ارزش حذف یا جایگزین شود، ابتدا مجموعه داده را بررسی می‌کنیم (برای کوتاه تر شدن گزارش فقط بخشی از این مجموعه داده نمایش داده می‌شود):

```
X_train.info()

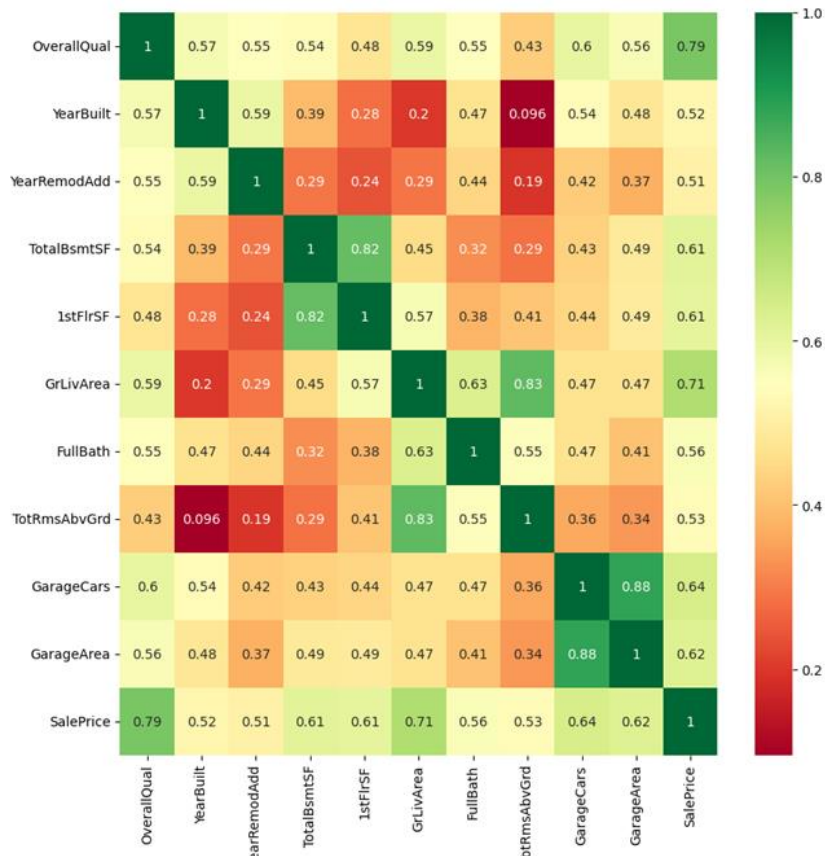
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
#   Column              Non-Null Count  Dtype  
---  -
0    Id                  1460 non-null  int64  
1    MSSubClass          1460 non-null  int64  
2    MSZoning            1460 non-null  object  
3    LotFrontage        1201 non-null  float64 
4    LotArea            1460 non-null  int64  
5    Street             1460 non-null  object  
6    Alley              91 non-null    object  
7    LotShape            1460 non-null  object  
8    LandContour        1460 non-null  object  
9    Utilities          1460 non-null  object  
10   LotConfig           1460 non-null  object  
11   LandSlope           1460 non-null  object  
12   Neighborhood        1460 non-null  object  
13   Condition1          1460 non-null  object
```

مشخص است تعدادی از ویژگی‌ها مقادیر گم شده یا نادرست دارند برای مثال ویژگی Ally فقط برای ۹۱ خانه موجود است پس بهتر است این ویژگی به طور کامل حذف شود. علاوه بر این، هرکدام از ویژگی‌های گفته شده می‌توانند در قیمت خانه تاثیر داشته باشند یا بی تاثیر باشند، برای مثال ویژگی ماه فروش احتمالاً تاثیری در قیمت خانه نخواهد داشت.

برای درک بهتر این موضوع ابتدا کورولیشن بین این ویژگی‌ها و قیمت خانه را بدست می‌آوریم:

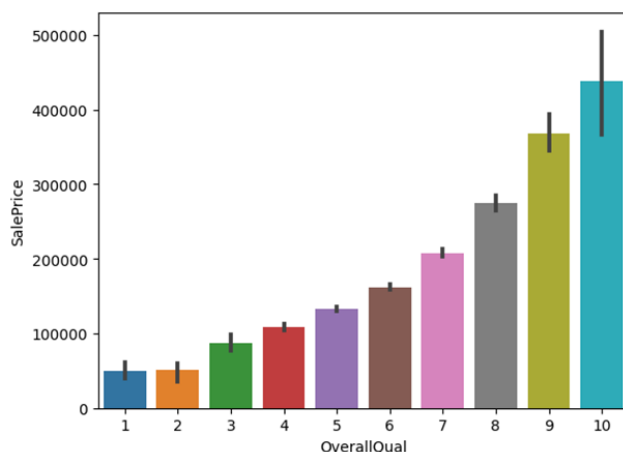


حال فقط ویژگی‌هایی که با قیمت خانه بیشترین کورلیشن را دارند بررسی می‌کنیم:





با استفاده از این ماتریس مشخص است که ویژگی OverallQual بیشترین کورولیشن را با قیمت خانه دارد، با رسم یک باریکات نیز افزایش قیمت خانه با افزایش مقدار Overallquality مشخص است:



حال با وجود یک درک از ویژگی‌ها و داده‌ها به تمیز کردن دیتاست می‌پردازیم، مقادیر گم شده و نا معلوم به شرح زیر می‌باشد:

```
train_nas = X_train.isnull().sum()
train_nas = train_nas[train_nas>0]
train_nas.sort_values(ascending=False)
```

PoolQC	1453
MiscFeature	1406
Alley	1369
Fence	1179
FireplaceQu	690
LotFrontage	259
GarageType	81
GarageYrBlt	81
GarageFinish	81
GarageQual	81
GarageCond	81
BsmtExposure	38
BsmtFinType2	38
BsmtFinType1	37
BsmtCond	37
BsmtQual	37
MasVnrArea	8
MasVnrType	8
Electrical	1
dtype: int64	

در این دیتاست ۱۴۶۱ داده وجود دارد، و مقادیر null در ویژگی‌هایی مانند PoolQC به شدت بالا می‌باشد که نیاز به جایگزینی این مقادیر حس می‌شود.



حال به ترتیب برای هر کدام این کار را انجام می‌دهیم، برای ویژگی‌هایی که تعداد زیادی missing value دارند، کلاً آن ویژگی را حذف می‌کنیم که شامل ویژگی‌های زیر می‌باشد. همچنین ویژگی id را نیز به دلیل بی‌اهمیت بودن، حذف می‌کنیم.

```
['FireplaceQu', 'id', 'Fence', 'Alley', 'MiscFeature', 'PoolQC']
```

البته مشخص است که برای مثال ویژگی PoolQC، برای خانه‌ها احتمالاً به خاطر نبود استخر در خانه مقداری ندارد که البته این ویژگی با ویژگی Poolarea کورولیشن خواهد داشت و به طور کلی بهتر است آن را حذف کنیم.

حال برای ویژگی‌های زیر مقدار که مقداری هستند median را جایگزین می‌کنیم:

```
['GarageYrBlt', 'MasVnrArea', 'LotFrontage']
```

مشخص است که آماره‌ی دیگری مانند mean نیز می‌توانستیم برای این ویژگی‌ها در نظر بگیریم، اما با توجه به وجود outlierهایی برای متراژ این مقدار منطقی‌تر به نظر می‌رسید.

برای ویژگی‌های categorical زیر نیز مقدار None را برای مقادیری که نمی‌دانیم جایگزین می‌کنیم:

```
['GarageQual', 'GarageFinish', 'GarageType', 'GarageCond'] = categorical_cols
```

```
['BsmtCond', 'BsmtQual', 'BsmtExposure', 'BsmtFinType2']
```

```
['Electrical', 'MasVnrType', 'BsmtFinType1']
```

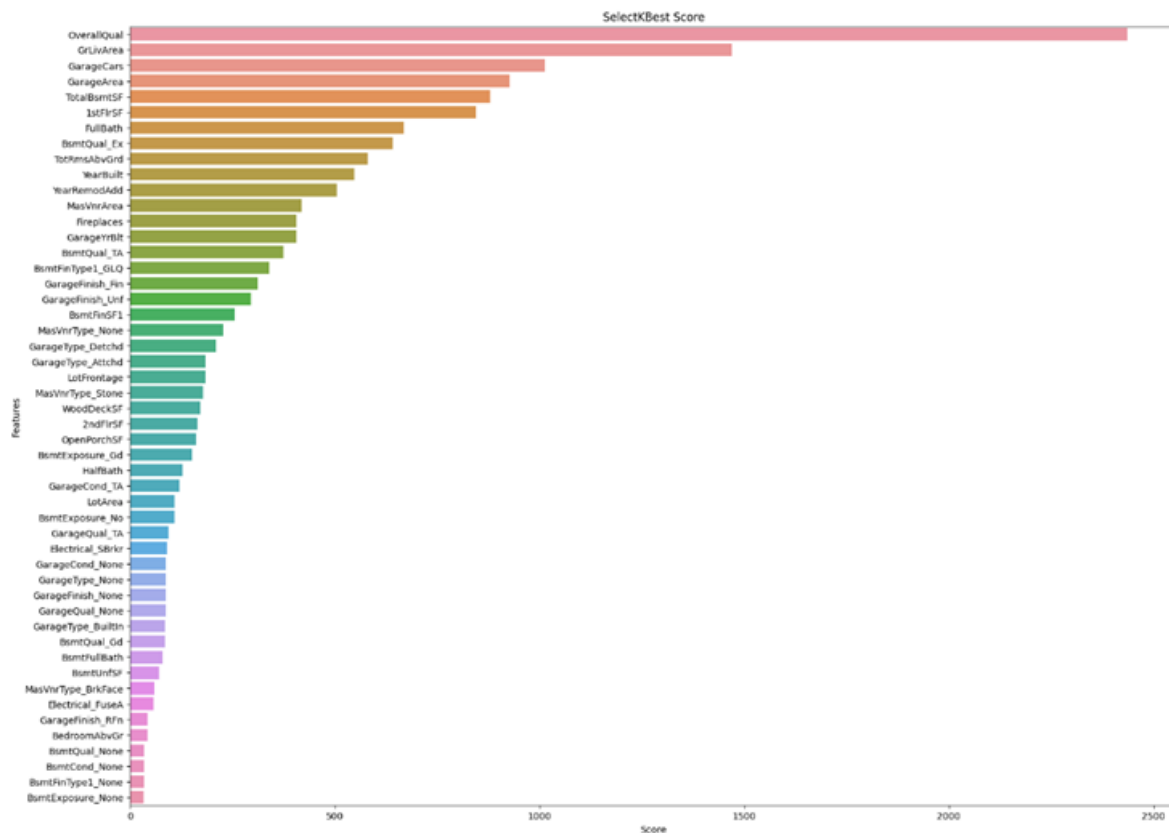
البته می‌توانستیم از آماره‌ای مانند md نیز برای جایگزینی استفاده کنیم ولی مقدار None را انتخاب کردیم.

حال بعد از جایگزینی مقادیر گم شده، باید داده‌ها را برای آموزش آماده کنیم، برای اینکار ابتدا ویژگی‌های categorical را پیدا کرده و آن‌ها با استفاده از one hot encoding به مقادیر عددی تبدیل می‌کنیم، که این شامل ویژگی‌های زیر می‌باشد:



MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functional', 'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond', 'PavedDrive', 'SaleType', 'SaleCondition'

حال برای تمام ویژگی‌ها ۵۰ بهترین ویژگی را با استفاده از SelectKBest پیدا می‌کنیم، نتایج این کار به صورت زیر می‌باشد:



همانند کورلیشن، همچنان ویژگی OverallQual بیشترین تاثیر را در قیمت خانه داشت.

سپس از بین این ویژگی‌ها ده ویژگی برتر که شامل ویژگی‌های زیر می‌باشد را انتخاب می‌کنیم:

'OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea'



'TotalBsmtSF', '1stFlrSF', 'FullBath', 'BsmtQual_Ex'

'TotRmsAbvGrd', 'YearBuilt'

در نهایت ویژگی‌ها به همراه قیمت خانه را نورمالیز می‌کنیم (البته نورمالیزشن را با استفاده از داده‌های آموزش fit می‌کنیم و برای داده‌های تست و ارزیابی از مدل fit شده برای transform استفاده می‌کنیم). تا همه اعدادی بین صفر تا یک داشته باشند و سپس مدل‌های رگرشنی برای آن آموزش می‌دهیم:

SVR:

از مدل SVR با کرنل RBF برای آموزش استفاده می‌کنیم و نتایج آن برای داده‌های تست، آموزش و ارزیابی به شکل زیر می‌باشد.

Validation Mean Squared Error: 0.0014905139603966268

Test Mean Squared Error: 0.004196300717293674

Train Mean Squared Error: 0.0024190359016381066

R-squared score (Training): 0.8148204994525989

R-squared score (Validation): 0.8541622780720175

R-squared score (Test): 0.6293969037580531

رگرشن خطی:

این مدل نیز با داده‌های آموزش می‌دهیم و عملکرد آن به صورت زیر می‌باشد:

Validation Mean Squared Error: 0.0014905139603966268

Test Mean Squared Error: 0.004196300717293674

Train Mean Squared Error: 0.0024190359016381066

R-squared score (Training): 0.8148204994525989

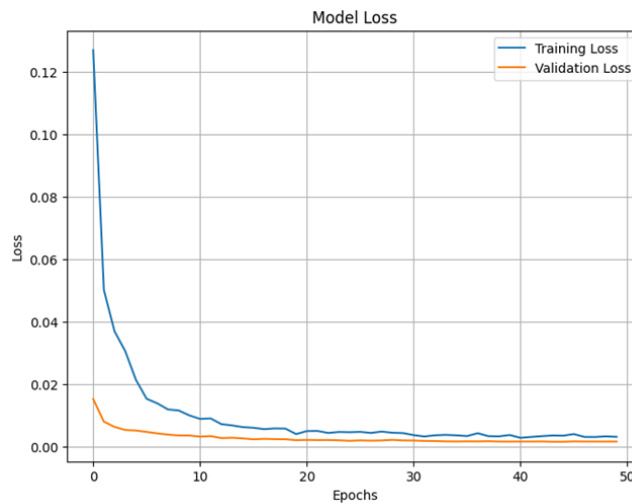
R-squared score (Validation): 0.8541622780720175

R-squared score (Test): 0.6293969037580531

شبکه عصبی:



این مدل را برای ۵۰ تا دوره آموزش می‌دهیم و عملکرد این مدل برای داده‌های آموزش و ارزیابی به شکل زیر می‌باشد:



همچنین مقادیر عملکرد این مدل برای داده‌های تست، آموزش و ارزیابی به شکل زیر می‌باشد:

Test Loss: 0.0025297985412180424

Training Loss: 0.00231863628141582

Training Loss: 0.0015784620773047209