

به نام خدا



دانشگاه صنعتی شاهرود
Shahrood University of Technology

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

عنوان تمرین : رگرسیون خطی

استاد: دکتر مرضیه رحیمی

نویسنده: مریم درویشیان

تاریخ: 1402/01/23

مراحل زیر را به ترتیب انجام دهید:

فاز اول: پیش پردازش دیتاست

الف- دیتاست را با استفاده از کتابخانه pandas بارگیری کنید.

```
import pandas as pd

dataset = pd.read_csv("T:/Assignments of ML/Ex_3/Advertising.csv")
```

✓ 17.3s Python

dataset

✓ 0.4s Python

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

200 rows × 5 columns

ب- ستون اول در دیتاست را حذف کنید

```
df = dataset.loc[:, ~dataset.columns.str.contains('^Unnamed')]
```

✓ 1.3s Python

df

✓ 0.0s Python

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

پ- داده‌های دیتاست را به دو متغیر ورودی ها (ویژگی ها) و خروجی (هدف) تقسیم کنید. ستون Sales ستون هدف ماست.

```
df.shape
✓ 0.0s
Python

(200, 4)

x=df
y=df.Sales
✓ 0.0s
Python
```

ت- سپس با استفاده از تابع `train_test_split` از کتابخانه `sklearn` داده ها را با نسبت 80 / 20 به دادههای آموزش و تست تقسیم کنید.

```
import sklearn.model_selection as ms

xtr,xte,ytr,yte=ms.train_test_split(X,Y,train_size=0.8,random_state=1)
✓ 31.5s
Python
```

ث- از متدهای `feature engineering` نرمالسازی دادهها میباشد.این تکنیک را بر روی داده های ورودی پیاده سازی کنید.

```
from sklearn import preprocessing
import numpy as np

# normalize the data attributes
normalized = preprocessing.normalize(X)
print("Normalized Data = ", normalized)
✓ 0.3s
Python

Output exceeds the size limit. Open the full output data in a text editor
Normalized Data = [[0.94211621 0.15476746 0.28333091 0.09048574]
 [0.59113524 0.52205877 0.59910561 0.13815296]
 [0.20142628 0.53752711 0.81156054 0.10891072]
 [0.89863215 0.24497365 0.34699657 0.10973396]
 [0.94788063 0.05662119 0.30617383 0.06763086]
 [0.09640672 0.54187225 0.83109241 0.07978487]
 [0.80725471 0.46048617 0.32992149 0.16566271]
 [0.97684844 0.15928643 0.09427156 0.10727454]
 [0.84982063 0.20751434 0.09881635 0.47431849]
 [0.99295394 0.01292132 0.10535848 0.05267924]
 [0.92901149 0.08151689 0.34012221 0.12086988]
 [0.99043367 0.11071452 0.01845242 0.08026803]
 [0.30163094 0.44484227 0.83518819 0.11659683]
 [0.98945567 0.0771268 0.0730675 0.09843815]
 [0.95983344 0.15472082 0.21632699 0.08935245]
 [0.93412001 0.22803237 0.25289124 0.10708438]
 [0.49072941 0.26490703 0.82512025 0.09047371]
 [0.96825675 0.13625788 0.19199974 0.08395688]
 [0.91887889 0.27221123 0.24299832 0.15004814]
 [0.97452547 0.15812056 0.12636413 0.09659248]
 [0.96106683 0.12189355 0.23498612 0.0792088 ]
 [0.99354663 0.02134409 0.09835024 0.05231395]
 [0.2443368 0.29431478 0.91811402 0.10365803]]
```

فاز دوم `Linear Regression` :
الف- مدل را بسازید و آموزش دهید.

```
import sklearn.linear_model as lm
LR=lm.LinearRegression()
LR.fit(xtr,ytr)
trAcc=LR.score(xtr,ytr)
teAcc=LR.score(xte,yte)
print('Train accuracy:', trAcc)
print('Test accuracy:', teAcc)
✓ 0.0s
Python

Train accuracy: 1.0
Test accuracy: 1.0
```

ب- مدل را ارزیابی کنید. برای این منظور، از `MSE` و `R2` استفاده کنید. این شاخص ها را توضیح دهید.

```

import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error

def calc_train_error(X_train, y_train, model):
    ypred = LR.predict(X_train)
    mse = mean_squared_error(y_train, ypred)
    rmse = np.sqrt(mse)
    return mse

```

✓ 0.1s

Python

```

MSE=calc_train_error(xtr,ytr,LR)
print (MSE)

```

✓ 0.1s

Python

1.6285355460947363e-29

```

from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

```

```

print("Coefficients: \n", LR.coef_)
ypred = LR.predict(xtr)

```

✓ 0.0s

Pythc

Coefficients:

```
[ 3.26025511e-17 -1.11022302e-16 -8.32667268e-17  1.00000000e+00]
```

```

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import ShuffleSplit, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

s = StandardScaler()
lr = LinearRegression()

estimator = Pipeline([("scaler", s), ("linear_regression", LR)])
# Cross validation R2 score
cv = ShuffleSplit(n_splits=5, test_size=0.30, random_state=42)
linear_reg_r2_score = cross_val_score(estimator, X, Y, cv=cv, scoring='r2').mean().round(4)
print('Cross validation R2 score:', linear_reg_r2_score)

```

✓ 2.9s

Pythc

Cross validation R2 score: 1.0

میانگین مربعات خطا (MSE) نشان دهنده خطای برآوردگر یا مدل پیش بینی ایجاد شده بر اساس مجموعه مشاهدات داده شده در نمونه است. میانگین اختلاف مجذور بین مقادیر پیش‌بینی‌شده و مقادیر واقعی را اندازه‌گیری می‌کند و اختلاف بین پیش‌بینی‌های مدل و مشاهدات واقعی را کمی‌سازی می‌کند. به طور شهودی، MSE برای اندازه‌گیری کیفیت مدل بر اساس پیش‌بینی‌های انجام شده در کل مجموعه داده آموزشی در مقابل مقدار برچسب/خروجی واقعی استفاده می‌شود. به عبارت دیگر، می‌توان از آن برای نشان دادن هزینه‌های مرتبط با پیش‌بینی‌ها یا زیان‌های متحمل شده در پیش‌بینی‌ها استفاده کرد. در سال 1805، ریاضیدان فرانسوی آدرین ماری لژاندر، که برای اولین بار روش مجموع مربعات را برای سنجش کیفیت مدل معرفی کرد، اظهار داشت که مربع کردن خطا قبل از جمع کردن همه خطاها برای یافتن ضرر کل راحت است.

R-Squared، همچنین به عنوان ضریب تعیین شناخته می‌شود، معیار آماری دیگری است که برای ارزیابی عملکرد مدل‌های رگرسیون استفاده می‌شود. نسبت کل تغییرات متغیر وابسته (خروجی) را اندازه‌گیری می‌کند که می‌تواند توسط متغیرهای مستقل (ورودی) در مدل توضیح داده شود. از نظر ریاضی، می‌توان آن را به عنوان نسبت مجموع مربعات رگرسیون (SSR) و مجموع مجذورات کل (SST) نشان داد. رگرسیون مجموع مربعات (SSR) نشان‌دهنده تغییر کل همه مقادیر پیش‌بینی‌شده در خط یا صفحه رگرسیون از مقدار میانگین همه مقادیر متغیرهای پاسخ است. مجموع مجذورات کل (SST) نشان‌دهنده تغییرات کل مقادیر واقعی از مقدار میانگین همه مقادیر متغیرهای پاسخ است.

مقدار R-squared برای اندازه‌گیری خوب بودن خط مناسب یا بهترین برازش استفاده می‌شود. هر چه مقدار R-Squared بیشتر باشد، مدل رگرسیون بهتر است زیرا بیشتر تغییرات مقادیر واقعی از مقدار میانگین توسط مدل رگرسیون توضیح داده می‌شود.

نتیجه می‌گیریم که:

1. MSE نشان دهنده خطای باقیمانده است که چیزی جز مجموع مجذور اختلاف بین مقادیر واقعی و مقادیر پیش بینی شده / تخمینی تقسیم بر تعداد کل رکوردها نیست.
2. R-Squared نشان دهنده کسری از واریانس است که توسط مدل رگرسیون گرفته شده است.
3. نقطه ضعف استفاده از MSE این است که مقدار MSE بر اساس مقیاس بندی یا عدم مقیاس بندی مقادیر متغیر پاسخ متفاوت است. اگر مقیاس بندی شود، MSE کمتر از مقادیر مقیاس نشده خواهد بود.

پ- بررسی کنید که چرا برای ارزیابی از MSE استفاده کردیم و از accuracy برای این تمرین استفاده نمیکنیم.

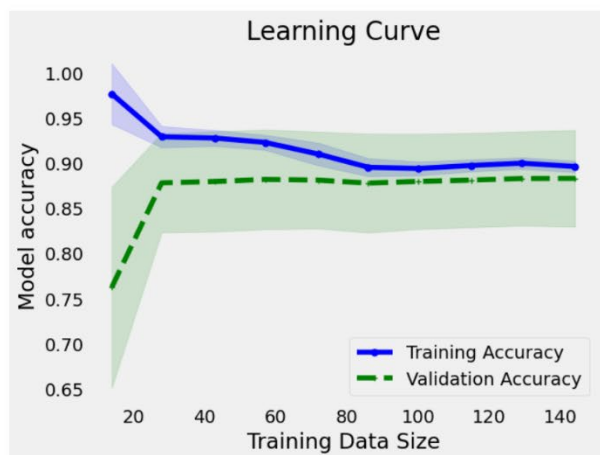
```
import sklearn.linear_model as lm
LR=lm.LinearRegression()
LR.fit(xtr,ytr)
trAcc=LR.score(xtr,ytr)
teAcc=LR.score(xte,yte)
print('Train accuracy:', trAcc)
print('Test accuracy:', teAcc)
```

✓ 0.0s Python

Train accuracy: 1.0
Test accuracy: 1.0

پیش بینی دقیق مقدار متغیر پیوسته در یک مسئله رگرسیونی برای یک مدل ایده آل یا ممکن نیست. یک مدل رگرسیون فقط باقیمانده ها است. اگر از accuracy برای مقادیر پیوسته استفاده کنیم همانند شکل بالا میزان دقت را به درستی برآورد نمی کند. میانگین مربعات خطا (MSE) به ما می گوید که یک خط رگرسیون چقدر به مجموعه ای از نقاط نزدیک است. این کار را با گرفتن فواصل از نقاط تا خط رگرسیون (این فواصل خطاها هستند) و مربع کردن آنها انجام می دهد. علاوه بر آن، برای حذف هر گونه علائم منفی، مربع سازی لازم است.

ت- نمودار روند همگرایی را رسم کنید.



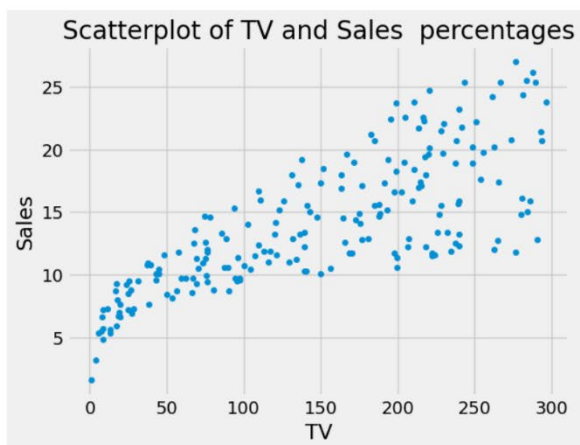
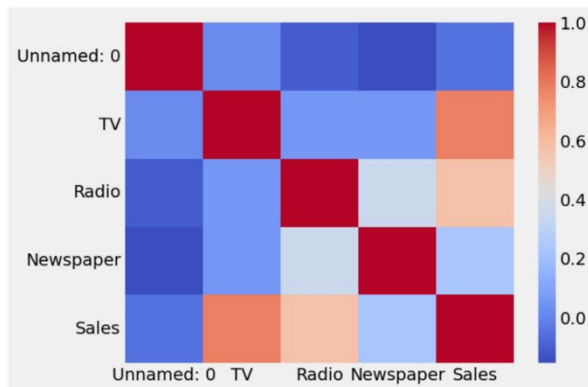
ث- میزان خطای پیش بینی را با توجه به فرمول زیر بدست آورده و در یک دیتافریم شامل سه ستون هدف، مقدار پیشبینی شده و میزان خطا نمایش دهید.

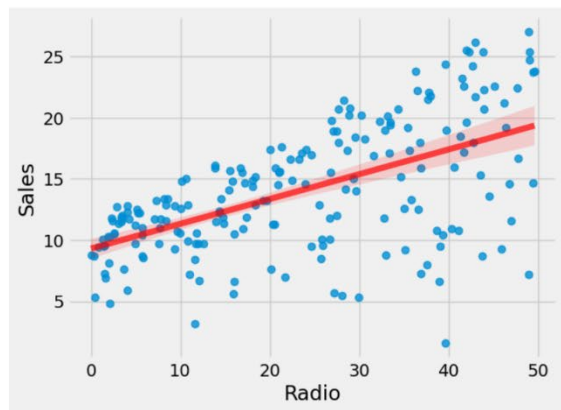
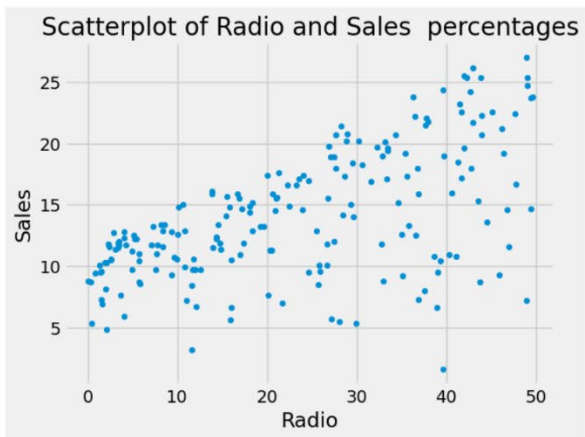
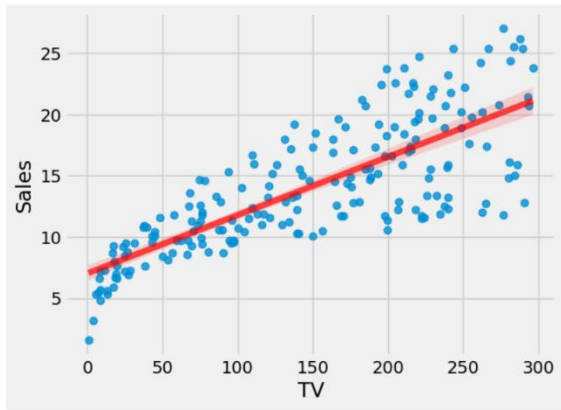
$$error = y_{pred} - y_{test}$$

	Actual	Predicted	Error
108	5.3	3.659418	-1.640582
107	8.7	7.256630	-1.443370
189	6.7	5.993589	-0.706411
14	19.0	18.472067	-0.527933
56	5.5	8.379175	2.879175
..
133	19.6	19.296079	-0.303921
137	20.8	21.036108	0.236108
72	8.8	10.109336	1.309336
140	10.9	9.410774	-1.489226
37	14.7	15.351927	0.651927

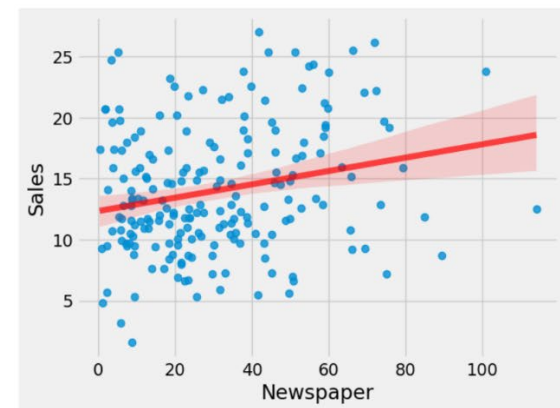
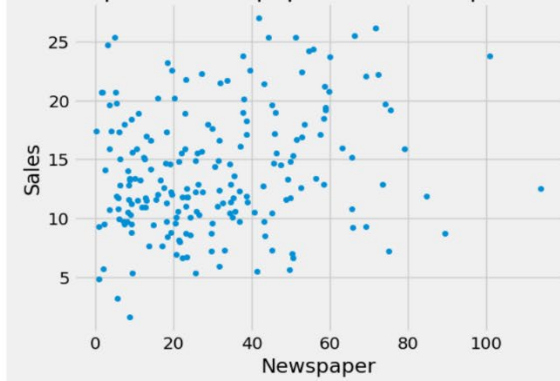
[160 rows x 3 columns]

ج- تحلیل خود را از نتیجه به دست آمده توسط این مدل بنویسید.
*نمره اضافه: رسم نمودار داده ها و fit کردن خط پیشبینی مدل رو آنها.





Scatterplot of Newspaper and Sales percentages



```
print(df.corr())
```

✓ 0.0s

Python

	Unnamed: 0	TV	Radio	Newspaper	Sales
Unnamed: 0	1.000000	0.017715	-0.110680	-0.154944	-0.051616
TV	0.017715	1.000000	0.054809	0.056648	0.782224
Radio	-0.110680	0.054809	1.000000	0.354104	0.576223
Newspaper	-0.154944	0.056648	0.354104	1.000000	0.228299
Sales	-0.051616	0.782224	0.576223	0.228299	1.000000

```
print(df.describe())
```

✓ 0.7s

Python

	Unnamed: 0	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

```
r_sq = model.score(X, y)
print(f"coefficient of determination: {r_sq}")
print(f"intercept: {model.intercept_}")
intercept: 5.52257927519819
print(f"coefficients: {model.coef_}")
```

✓ 0.0s

Python

```
coefficient of determination: 0.8962051825476649
intercept: 2.9246804226395344
coefficients: [-1.36610117e-04  4.68432118e-02  1.78501247e-01  2.52272234e-03]
```



```

• y_pred = model.predict(X)
print(f"predicted response:\n(y_pred)")
✓ 6.8s
Python

```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

predicted response:

```

[20.62508638 12.13780391 12.09800573 17.54056133 13.46839052 12.24931186
11.53133371 12.08202962 3.70367789 12.800173 7.11587113 17.2743995
10.46941411 8.86475411 18.47206706 20.72361975 12.9190638 23.31331854
9.86907647 14.13631712 18.23156699 15.01189382 6.50316564 16.69847344
8.1348788 15.9101564 14.87675976 17.20665167 19.47046397 9.1866599
21.80138915 11.41220629 7.81676602 18.9323745 7.67136433 17.29038239
23.25304732 15.35192718 9.79282591 20.40969244 16.46512564 17.2697623
21.62099862 14.17654045 8.79101351 15.21638656 8.97731938 21.61028177
16.5069678 8.23296156 12.91762709 9.34332894 20.59771293 19.86571537
20.40382581 21.20347003 8.37917484 12.76590362 21.73993796 18.07559665
5.83344779 22.91633801 16.96129422 13.03156301 16.76970588 7.81017782
8.78776938 12.05465019 18.97236705 20.97554883 17.80123209 10.69076728
10.10933565 10.0725012 17.33495441 11.73198421 4.54017212 13.68174006
8.52774244 9.78029867 11.31467639 14.97142416 10.14619942 14.15039527
20.67491592 15.41320628 11.43608026 15.50478896 11.78572735 16.71779771
10.10140968 4.60282986 19.23837529 21.36248926 10.46917854 16.33515376
12.80728632 15.37662858 24.16140062 16.80351661 14.22200005 23.52921213
17.89292639 14.82769065 20.20435262 17.80117714 6.11958201 7.25663041
3.65941819 19.68896813 15.09295797 21.07295893 13.89456954 16.43156289
15.01300171 12.80724997 12.04642165 6.6675191 15.58311117 6.72932218
14.42747962 7.78928864 13.86851429 14.88156505 19.61089755 9.16384884
10.34405625 6.68722896 21.9812511 7.94952083 10.03017181 15.95559834
8.16052589 19.29607918 11.69039129 13.57963033 11.09016101 21.03610787
9.59484783 19.40735825 9.41077446 18.48847387 19.25592596 8.90904728
...
10.7206428 14.16808849 6.62465124 24.22833697 18.66597294 20.60198096
9.87571941 17.01963484 18.7865211 5.99358935 12.0999278 8.37806357
4.51559105 18.21775997 16.28025099 5.38258371 8.20548894 12.86508713
23.84628648 15.32672627]

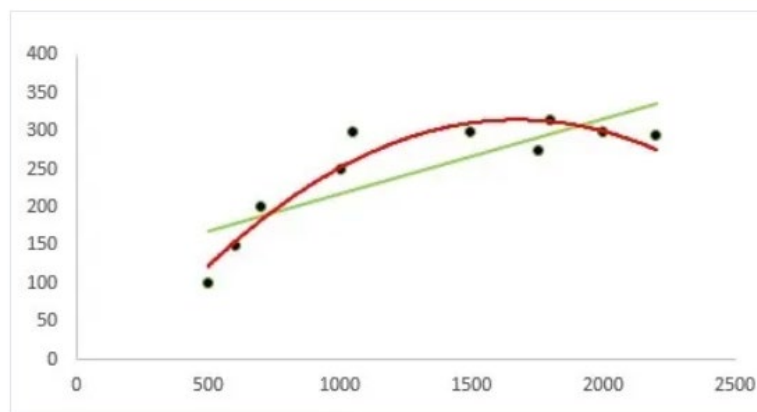
```

فاز سوم Polynomial Regression :

الف - Polynomial Regression چیست؟

رگرسیون چندجمله‌ای سعی دارد با استفاده از تعداد متغیرهای بسیار کمتر از تعداد نقاط، پارامترهای چندجمله‌ای یا رگرسیون را برآورد کند. در نتیجه پیچیدگی مدل کاهش یافته ولی در عوض خطا نیز بوجود خواهد آمد. به این ترتیب یک توازن بین بیش‌برازش (Overfitting) و کم‌برازش (Underfitting) در مدل برقرار خواهد شد.

در آمار، رگرسیون چندجمله‌ای شکلی از تحلیل رگرسیونی است که در آن رابطه بین متغیر مستقل X و متغیر وابسته Y به عنوان چندجمله‌ای درجه n در X مدل می‌شود. رگرسیون چندجمله‌ای با یک رابطه غیرخطی بین مقدار X و میانگین شرطی مربوط به Y که $E(Y | X)$ نشان داده می‌شود، برازش می‌کند. اگرچه رگرسیون چندجمله‌ای یک مدل غیرخطی را به داده‌ها فیت می‌کند، اما به عنوان یک مسئله تخمین آماری خطی است، به این معنا که تابع رگرسیون $E(Y | X)$ در پارامترهای مجهولی که از داده‌ها تخمین زده می‌شوند خطی است. به همین دلیل، رگرسیون چندجمله‌ای یک مورد خاص از رگرسیون خطی چندگانه در نظر گرفته می‌شود.



در شکل بالا تفاوت رگرسیون خطی با رنگ سبز و رگرسیون درجه 2 با خط قرمز نمایش داده شده است.

ب- دلیل استفاده از Polynomial Regression چیست؟

در رگرسیون خطی بعد از بررسی نمودار متوجه میشویم که خط مستقیم قادر به ثبت درست الگوهای داده نیست. این نمونه ای از عدم تناسب است. مدل خود را می سازیم و متوجه می شویم که رگرسیون خطی عملکرد بسیار بدی دارد، ما بین مقدار واقعی و بهترین خط مناسب، که مدل ایجاد کرده، اختلاف می بینیم. به نظر می رسد که مقدار واقعی دارای یک نوع منحنی در نمودار است و خط ما در پیش بینی و تقسیم بندی داده ها ناتوان بوده است. در این گونه موارد رگرسیون چند جمله ای برای مدل سازی استفاده می شود، بهترین خط مناسب را که از الگوی منحنی داده ها پیروی می کند، پیش بینی میکند و به این ترتیب یک توازن بین بیش برآزش و کم برآزش در مدل برقرار خواهد شد.

رگرسیون چند جمله ای روشی پارامتری برای برآزش منحنی روی داده ها می باشد که در آن ارتباط بین متغیر هدف و متغیرهای مستقل به صورت چند جمله ای برآورد می شود. در بسیاری از مواقع در مورد نحوه ارتباط بین متغیرها اطلاع زیادی در دست نیست. در این صورت بهتر است به جای مفروض داشتن یک الگوی پارامتری خاص مانند چند جمله ای برای داده ها از روشی استفاده شود که داده ها ماهیت روند خود را بهتر نشان دهند. برای نمونه های با حجم بزرگ رگرسیون چند جمله ای روشی آسان تر و سریع تر است.

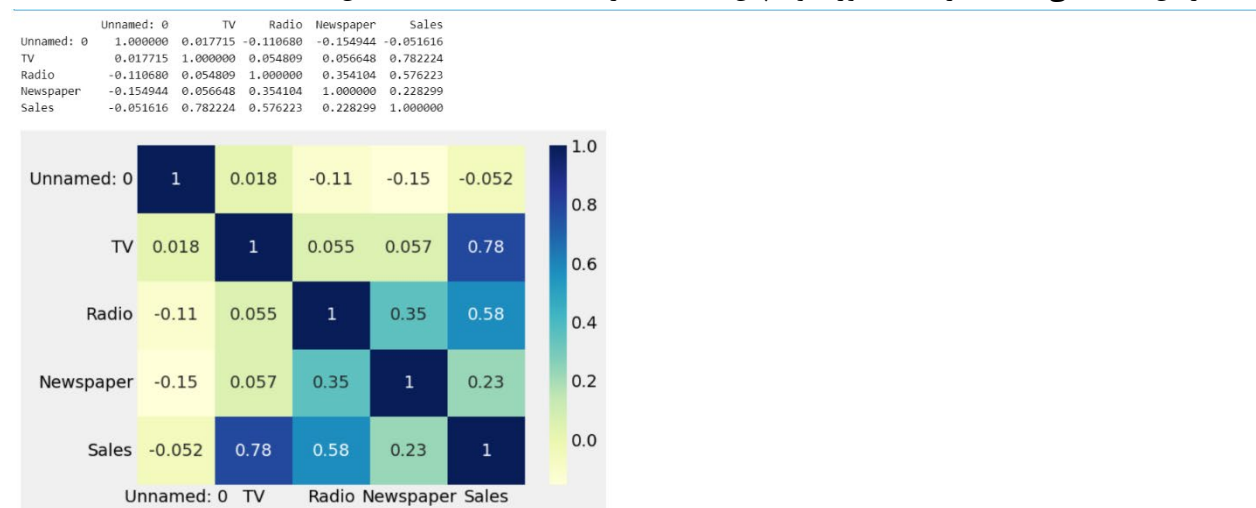
پ- مجدداً از داده های اولیه دیتاست استفاده میکنیم.

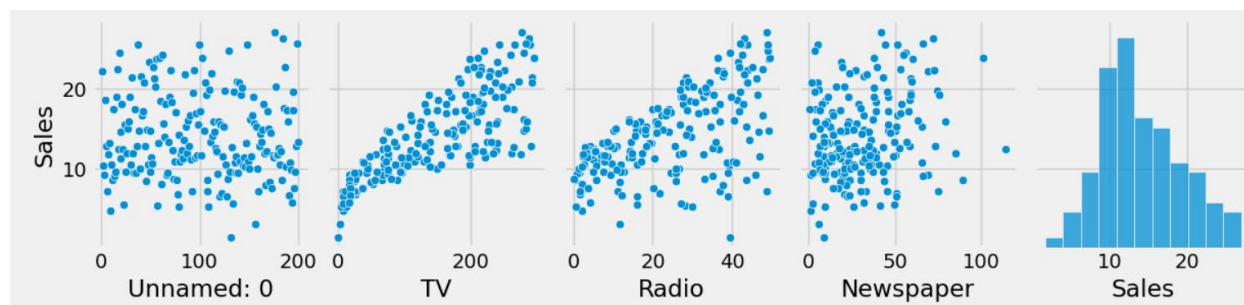
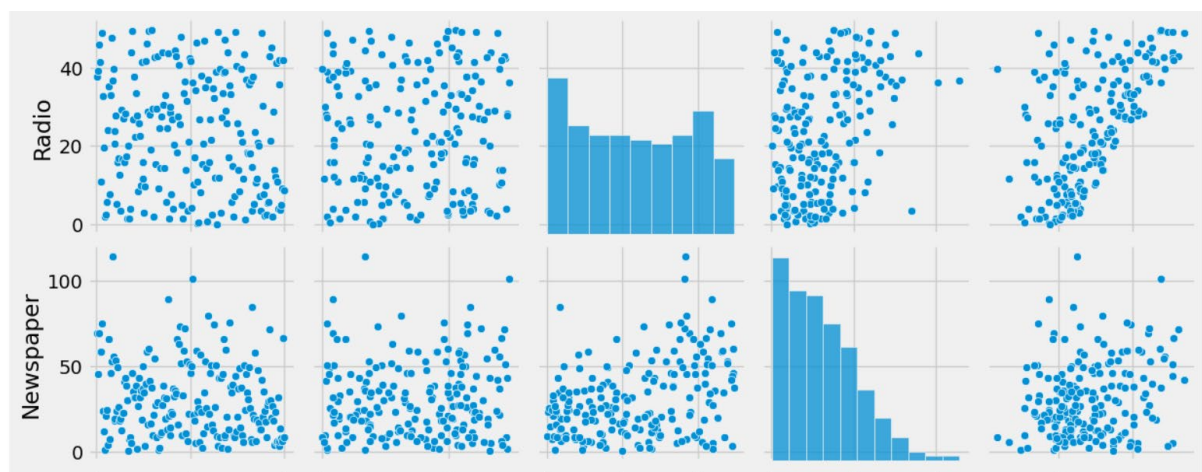
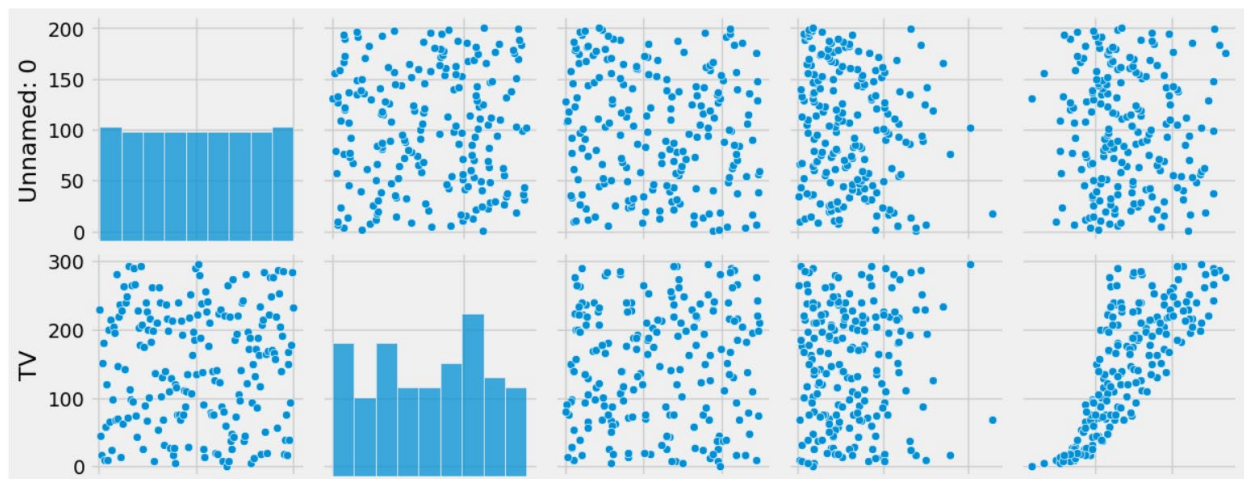
```
import pandas as pd

dataset = pd.read_csv("../Assignments of ML/Ex_3/Advertising.csv")
```

✓ 0.6s Python

ت- ماتریس همبستگی دیتاست را بدست آورده و سپس با استفاده از کتابخانه seaborn نمایش دهید.





ث- با استفاده و تحلیل ماتریس همبستگی ویژگی‌هایی براساس ویژگی‌های دیگر تکنیک‌های `combine` و `interactions` تولید کرده و یا ویژگی‌ای که تاثیر بسزایی ندارد را حذف کنید. انجام این بخش نیازمند این است که به نظر خودتون مطمئن باشید. انجام این مرحله کاملاً بصورت شخصی و براساس استدلال فردی می‌باشد. هدف از این قسمت آشنایی شما با تحلیل و آنالیز داده‌های موجود برای `feature engineering` می‌باشد. توجه داشته باشید که تحلیل خود از چرایی تولید ویژگی‌ها رو باید شرح دهید.

با توجه به ماتریس `Correlation` و مقادیر آن که برای بقیه سطرها منفی و برای `TV` مثبت است که نشان‌دهنده تاثیر معکوس آن می‌باشد از آنجا که دیتاست باید شامل تمام نمونه‌های ممکن باشد تا مدل بخوبی آموزش ببیند و کمترین میزان خطا را داشته باشیم. همین‌طور این موضوع که افراد جامعه اکثراً از رسانه‌های تلویزیون و رادیو استفاده می‌کنند میتوان ستون روزنامه را حذف کرد که داده‌های آن شباهت زیادی با رادیو دارد. به این ترتیب فراوانی انواع مختلف دیتا را خواهیم داشت که کمترین شباهت را به دسته‌های مجاور خود دارند و در نتیجه اشتراک داده کمتر و خطای کمتری را به همراه خواهد داشت.

علاوه برآن میتوان داده‌های رادیو و روزنامه را که شباهت زیادی به هم دارند را ترکیب کنیم و یک ستون جدید در دیتاست ایجاد کنیم.

بقیه قسمت‌ها همانند تمرین‌های قبلی که انجام دادم اسان بود ولی بخاطر کمبود وقت بقیه شو انجام ندادم. البته آقایان حل تمرین هم به سوال‌هایم جوابی ندادند که باعث شد مطالب وقت گیر زیادی را بخوانم و فرصت برای انجام بقیه تمرین نداشته باشم.

ج- پس از ایجاد ویژگی‌های جدید، مجدداً با همان نسبت قبلی داده‌ها را به داده‌های آموزش و تست تقسیم نمایید.

چ- مجدداً تکنیک نرمالسازی را بر روی داده‌ها اعمال کنید (`fit_transform`, `transform`).

ح- مدل را براساس داده‌هایی جدید آموزش دهید.

خ- با توجه به فرمول میزان خطای پیش‌بینی همانند مرحله قبل، یک دیتافریم شامل سه ستون هدف، مقدار پیش‌بینی شده و میزان خطا را تشکیل داده و نمایش دهید.

د- تحلیل خود را از نتیجه به‌دست آمده توسط این مدل بنویسید. تفاوت `Polynomial Regression` و `Regression` را بنویسید. همچنین توضیح دهید که آیا این مدل از مدل قبلی بهتر است یا خیر؟ چرا؟
- *نمره اضافه: رسم نمودار داده‌ها و `fit` کردن خط پیش‌بینی مدل رو آن‌ها.

فاز چهارم `SGDRegressor`:

الف- مدل `SGDRegressor` را توضیح دهید.

ب- داده‌ها را مجدداً به داده‌های آموزشی و تست تقسیم کنید.

پ- مدل رگرسیون خطی را با استفاده از آن بسازید.

* پارامترهای `SGDRegressor` را بصورت زیر تنظیم کنید (آرگومان‌های آنرا شرح دهید):

(`max_iter=100`, `verbose=10`, `learning_rate='adaptive'`)

(estimator=..., param_grid=tuned_parameters, cv=5)
ت- مدل GridSearchCV را تفسیر کنید. پارامترهای بالا را توضیح دهید. در میان پارامترهای بالا پارامتری به نام param_grid قرار دارد که میبایست در یک لاین جداگانه تنظیم شود که برای رگرسیون خطی شایعترین نوع تنظیم به صورت زیر میباشد.
tuned_parameters = {'alpha':np.arange(-1,2,0.1)}
پارامتر بالا را توضیح داده و دلیل افزودن آن بیان کنید ؟
ث- مدل را آموزش داده و ارزیابی کنید.
ج- با توجه به فرمول میزان خطای پیش بینی همانند مرحله قبل، یک دیتافریم شامل سه ستون هدف، مقدار پیشبینی شده و میزان خطا را تشکیل داده و نمایش دهید.
چ- تحلیل خود را بیان کرده و با مقایسه با نتایج قبلی بهترین مدل را برای این مسئله مشخص نمایید.
موفق باشید