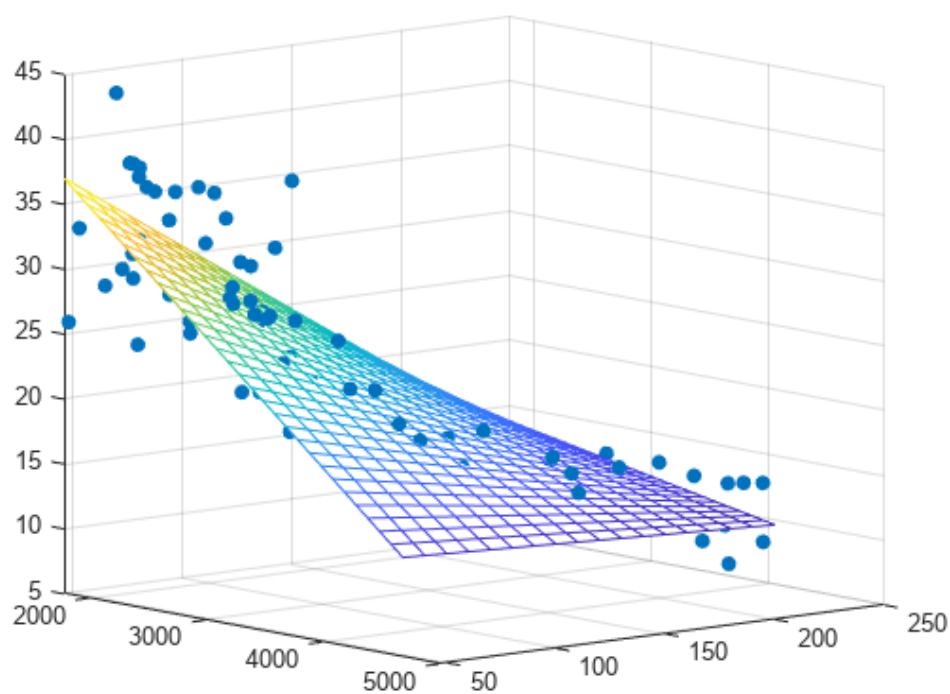


## رگرسیون خطی چند متغیره



سید محمد حسین هاشمی ۴۰۲۲۳۶۳۱۴۳  
سید حسین حسینی ۴۰۱۲۳۶۳۲۳۸

آبان ۱۴۰۲

## فهرست مطالب

۲	۱	پکیج های استفاده شده
۲	۲	آماده سازی برای شروع
۲	۳	تبدیل داده های دسته ای
۳	۴	جدا سازی داده ها
۳	۵	مقیاس کردن داده ها
۴	۶	تقسیم بندی داده ها
۴	۷	آماده سازی پردازش
۵	۸	پردازش اصلی (یافتن ضرایب)
۶	۹	محاسبه خطا و تحویل خروجی
۷	۱۰	منبع

## ۱ پکیج های استفاده شده

```
Regression.py

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import r2_score
from time import time
```

- pandas: برای ایجاد dataframe و تقسیم بندی داده ها مورد استفاده قرار می گیرد
- sklearn: برای تقسیم بندی، دسته بندی داده ها و محاسبه خطا (R Squared (R2)
- time: برای محاسبه زمان اجرای الگوریتم

## ۲ آماده سازی برای شروع

```
Regression.py

data = pd.read_csv("../Datasets/Flight_Price_Dataset_Q2.csv")
```

ساخت دیتافریم با متد read\_csv پکیج pandas

## ۳ تبدیل داده های دسته ای

```
Regression.py

data['stops'] = label_encoder.fit_transform(data['stops'])
data['departure_time'] = label_encoder.fit_transform(data['departure_time'])
data['arrival_time'] = label_encoder.fit_transform(data['arrival_time'])
data['class'] = label_encoder.fit_transform(data['class'])
data['price'] = label_encoder.fit_transform(data['price'])
```

تبدیل داده های دسته ای به داده های قابل پردازش در ستون های:

- departure\_time
- stops
- arrival\_time
- class

## ۴ جدا سازی داده‌ها

```
Regression.py

x = data.filter(['departure_time', 'stops', 'arrival_time', 'class', 'duration', 'days_left'])
y = data.filter(['price'])
```

جدا سازی داده‌های ورودی و خروجی

## ۵ مقیاس کردن داده‌ها

```
Regression.py

yScale = y['price'].mean()
departureTimeScale = x['departure_time'].mean()
stopsScale = x['stops'].mean()
arrivalTimeScale = x['arrival_time'].mean()
classScale = x['class'].mean()
durationScale = x['duration'].mean()
daysLeftScale = x['days_left'].mean()

x['departure_time'] = x['departure_time'] / departureTimeScale
x['stops'] = x['stops'] / stopsScale
x['arrival_time'] = x['arrival_time'] / arrivalTimeScale
x['class'] = x['class'] / classScale
x['duration'] = x['duration'] / durationScale
x['days_left'] = x['days_left'] / daysLeftScale
# y['price'] = y['price']/yScale
```

مقیاس کردن داده‌های ورودی برای پردازش صحیح الگوریتم. تقسیم هر مقدار از یک دسته به میانگین کل داده‌های آن دسته.

## ۶ تقسیم بندی داده‌ها

```
Regression.py

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, shuffle=True)

y = y_train['price']
yt = y_test['price']
del y_train, y_test, y, x

x1 = x_train['departure_time']
x2 = x_train['stops']
x3 = x_train['arrival_time']
x4 = x_train['class']
x5 = x_train['duration']
x6 = x_train['days_left']

xt1 = x_test['departure_time']
xt2 = x_test['stops']
xt3 = x_test['arrival_time']
xt4 = x_test['class']
xt5 = x_test['duration']
xt6 = x_test['days_left']

del x_train, x_test
```

تقسیم بندی داده ها با استفاده از LabelEncoder در sklearn (روش گفته شده در قسمت مقدمات)

## ۷ آماده سازی پردازش

```
Regression.py

start_time = time()

# Building the model
w0 = 0
w1 = 0
w2 = 0
w3 = 0
w4 = 0
w5 = 0
w6 = 0

L = 0.1 # The learning Rate
epochs = 1000 # The number of iterations to perform gradient descent

n = float(len(y)) # Number of elements in X
```

مقدار start\_time را ایجاد و به متغیرهای رابطه به طور پیش فرض مقدار ۰ را انتصاب

می‌دهیم.

مقدار  $L$  همان ضریب یادگیری و epochs تعداد تکرار الگوریتم گرادیان کاهشی (Gradient Descent) از مبحث کاهش شیب می‌باشد. مقدار  $n$  نیز تعداد داده‌هاست.

## ۸ پردازش اصلی (یافتن ضرایب)

```
Regression.py

# Performing Gradient Descent
for i in range(epochs):
    y_pred = w0 + w1 * x1 + w2 * x2 + w3 * x3 + w4 * x4 + w5 * x5 + w6 * x6 # The current predict of Y

    dif = y - y_pred

    D_w0 = (-2 / n) * sum(dif) # Derivative wrt w0
    D_w1 = (-2 / n) * sum(x1 * dif) # Derivative wrt w1
    D_w2 = (-2 / n) * sum(x2 * dif) # Derivative wrt w2
    D_w3 = (-2 / n) * sum(x3 * dif) # Derivative wrt w3
    D_w4 = (-2 / n) * sum(x4 * dif) # Derivative wrt w4
    D_w5 = (-2 / n) * sum(x5 * dif) # Derivative wrt w5
    D_w6 = (-2 / n) * sum(x6 * dif) # Derivative wrt w6

    w0 = w0 - L * D_w0 # Update w0
    w1 = w1 - L * D_w1 # Update w1
    w2 = w2 - L * D_w2 # Update w2
    w3 = w3 - L * D_w3 # Update w3
    w4 = w4 - L * D_w4 # Update w4
    w5 = w5 - L * D_w5 # Update w5
    w6 = w6 - L * D_w6 # Update w6
```

با استفاده از فرمول زیر

$$D_m = \frac{1}{n} \sum_{i=1}^n (y_i - (mx_i + c))(-x_i)$$

$$D_m = \frac{-2}{n} \sum_{i=1}^n x_i (y_i - \bar{y}_i)$$

در هر بار اجرا برای هر متغیر فرمول فوق محاسبه و بر اساس ضریب یادگیری تعیین شده در قسمت ۷ مقدار متغیرها مجدداً تائین می‌شود.

## ۹ محاسبه خطا و تحویل خروجی

```

Regression.py

execution_time = round(time() - start_time, 2)

output = "PRICE = "
output += str(w0) + " + "
output += str(w1) + " [departure_time] + "
output += str(w2) + " [stops] + "
output += str(w3) + " [arrival_time] + "
output += str(w4) + " [class] + "
output += str(w5) + " [duration] + "
output += str(w6) + " [days_left]\n"

output += "Training Time: " + str(execution_time) + "s\n\n"

output += 'Logs:\n'

y_pred = w0 + w1 * xt1 + w2 * xt2 + w3 * xt3 + w4 * xt4 + w5 * xt5 + w6 * xt6

error = (yt - y_pred)
loss = (error ** 2).mean()
output += 'MSE: ' + str(round(loss)) + "\n"

output += 'RMSE: ' + str(round(loss ** 0.5)) + "\n"

output += 'MAE: ' + str(round((1 / n) * sum(abs(yt - y_pred)))) + "\n"

output += 'R2: ' + str(round(r2_score(list(yt), list(y_pred)), 2)) + "\n"

f = open("../Outputs/The_Phoenix-UIAI4021-PR1-Q2.txt", "w")
f.write(output)
f.close()

```

در این بخش از همزمان با استفاده از مقادیر تستی مقادیر خطا را تحت معیارهای مختلف حساب و همزمان خروجی را تحویل می‌دهیم. معیارهای سنجش خطا استفاده شده به ترتیب:

• Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

• Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_{pred,i} - y_i)^2}{n}}$$

:Mean Absolute Error (MAE) •

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

:R Squared (R<sup>2</sup>) •

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

١٠ منبع

<https://github.com/chasinginfinity/ml-from-scratch/tree/master> •