



به نام خدا



دانشگاه تهران
دانشکده مهندسی مکانیک
هوش مصنوعی

پروژه 1

نام و نام خانوادگی	محمد مشرقی
شماره دانشجویی	
تاریخ ارسال گزارش	

به این دلیل که تمام موارد مورد بررسی در یک رنج باشند. با این کار در محاسبات خطا به ثبات و بهبود می رسیم و تغییرات ناگهانی نداریم.

حال چند روش برای این هست که در این تمرین روش شماره یک استفاده شده (مقیاس بندی خطی)

تکنیک عادی سازی	فرمول	چه موقع باید استفاده کرد
مقیاس بندی خطی	$x' = (x - x_{min}) / (x_{max} - x_{min})$	هنگامی که ویژگی کم و بیش به طور یکنواخت در یک محدوده ثابت توزیع شده است.
بریدن	اگر $x > \max$, $x' = \max$ اگر $x < \min$, $x' = \min$	زمانی که این ویژگی حاوی برخی نقاط پرت شدید باشد.
مقیاس گذاری ورود به سیستم	$x' = \log(x)$	هنگامی که ویژگی با قانون قدرت مطابقت دارد.
امتیاز Z	$x' = (x - \mu) / \sigma$	زمانی که توزیع ویژگی حاوی مقادیر پرت شدید نباشد.

2-

ابتدا داده ها را نرمالایز کرده و سپس آموزش می دهیم.

```
train_test_split(X_normalized, y, test_size = 0.2, random_state = 42)
```

خروجی قسمت مدل خطی :

```
from sklearn.metrics import r2_score
average=r2_score(y_test,y_pred)
y1_r2= r2_score(y_test[:,0],y_pred[:,0])
y2_r2 =r2_score(y_test[:,1],y_pred[:,1])
print(average,y1_r2,y2_r2)
```

```
0.9030258683831562 0.913844125587917 0.8922076111783953
```

```
[ ] from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_pred)
```

```
9.483989633088544
```

حال با درجه ۳ حل می کنیم:

```

▶ from sklearn.metrics import r2_score
  average_poly=r2_score(y_test,y_pred_poly)
  y1_poly_r2= r2_score(y_test[:,0],y_pred_poly[:,0])
  y2_poly_r2 =r2_score(y_test[:,1],y_pred_poly[:,1])
  print(average_poly ,y1_poly_r2,y2_poly_r2)

☞ 0.9843622255963262 0.9976416071226276 0.97108284407002

[ ] from sklearn.metrics import mean_squared_error
    mean_squared_error(y_test,y_pred_poly)

1.4626023689377026

```

حال می رویم و چک می کنیم اگر بدون نرمالایز حل می کردیم چه می شد:

بدون نرمالایز خطی:

```
[ ] from sklearn.metrics import r2_score
    average_3=r2_score(y_test_2,y_pred_2)
    y1_r2_3= r2_score(y_test_2[:,0],y_pred_2[:,0])
    y2_r2_3 =r2_score(y_test_2[:,1],y_pred_2[:,1])
    print(average_3,y1_r2_3,y2_r2_3)

0.9028963100022006 0.9120062665636756 0.8937863534407257

[ ] from sklearn.metrics import mean_squared_error
    mean_squared_error(y_test_2,y_pred_2)

9.506630922228643
```

با توجه به نتایج می فهمیم که زیاد فرقی با با نرمالایز کردن ندارد اما در کل خطا کمی بیشتر شده .

بدون نرمالایز درجه ۳ :

```
from sklearn.metrics import r2_score
average_poly_wo_norm=r2_score(y_test,y_pred_poly_wo_norm)
y1_poly_wo_norm_r2= r2_score(y_test[:,0],y_pred_poly_wo_norm[:,0])
y2_poly_wo_norm_r2 =r2_score(y_test[:,1],y_pred_poly_wo_norm[:,1])
print(average_poly_wo_norm ,y1_poly_wo_norm_r2,y2_poly_wo_norm_r2)

-530585991.81569517 -1042224611.383392 -18947372.247998517

from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_pred_poly_wo_norm)

55194596806.20246
```

با توجه به نتایج می فهمیم که با وقتی درجه رو بالا می بریم خطا ها به شدت بالا می رود و بدون نرمالایز کردن اصلا نمی توان مسئله را حل کرد.