

به نام خدا

گزارش تمرین چهارم که همان دیتاست تمرین 1 یعنی آگهی های خانه در آلمان می باشد.

محمدسعید حیلری 400422075

در این گزارش اقداماتی که در جریان حل تسکهای مشخص شده برای تمرین 4 انجام داده شده شرح و توضیحات و تحلیلهای مربوط به آنها نیز ارائه شده است.

پیاده سازی مدل رگرسیون:

قبل از پیاده سازی مدل رگرسیون به پیش پردازش داده ها پرداخته شد که در ادامه به چند مورد از آنها اشاره می شود:

1- جایگذاری مقادیر گم شده با مقدار مناسب برای هر ویژگی. به طور مثال مقادیر گم شده ویژگی serviceCharge با مقدار median آن جایگذاری شده است و همچنین ویژگی ها heatingType و telekomUploadSpeed و مقدار هدف totalRent به ترتیب با مقدار mode و median و median جایگذاری شده اند.

2- در ادامه با محاسبه ی چارک اول و سوم داده ها اقدام به محاسبه مرز بالا و پایین برای مقادیر قیمت خانه شد که هدف از این امر حذف داده های نویز می باشد و در صورتی که داده های نویز حذف نشوند مدل دچار خطا شده و رگرسیون به درستی انجام نمی شود.

3- در مرحله ی بعد داده ها، به داده های آموزش و آزمون تقسیم شدند تا برای بررسی صحت مدل از داده های آزمون استفاده شود.

4- در مرحله بعد با استفاده از متد OneHotEncoder مقادیر ویژگی heatingType که غیر عددی هستند به مقادیر عددی تبدیل شدند تا امکان استفاده از آنها در آموزش مدل فراهم شود. در این مرحله همچنین مابقی ویژگی ها نیز با استفاده از StandardScaler نرمال سازی شدند تا مقدار تمام ویژگی ها در بازه یکسانی قرار گیرد و در نتیجه فرآیند آموزش برای مدل آسان تر شود.

5- در نهایت مدل رگرسیون پیاده سازی شد و با استفاده از داده‌های آموزش، آموزش داده شد و با استفاده از داده‌های آزمون صحت آموزش مورد بررسی قرار گرفت که در ادامه نتایج حاصل ارائه می‌شود:

```
: y_pre_test_mse = model_mse.predict(X_test)
: print('linear regression from scrach (MSE) R^2: test', metrics.r2_score(y_test, y_pre_test_mse))

linear regression from scrach (MSE) R^2: test 0.31261126734724354

: mse = mean_squared_error(y_test, y_pre_test_mse)
: mse

: 0.6959165091316912
```

1- در این قسمت به مقایسه مدل رگرسیون پیاده سازی شده و مدل رگرسیون کتابخانه sklearn پرداخته شده است:
نتایج حاصل از آموزش و آزمون مدل رگرسیون sklearn به صورت زیر می‌باشد:

```
mse_lr = mean_squared_error(y_test, y_pred_test)
mse_lr

0.6906342979408073

print('linear reggression score train:', model_lr.score(X_train, y_train))
print('linear reggression R^2: test', metrics.r2_score(y_test, y_pred_test))

linear regression score train: 0.3163613124953878
linear regression R^2: test 0.3178287502039676
```

همان‌طور که از مقایسه دو تصویر قابل برداشت است امتیاز حاصل از مدل sklearn در حدود 0.005 از مدل پیاده سازی شده بیشتر است و همچنین مقدار مجموع مربعات خطا نیز در این مدل در حدود 0.005 کمتر از مدل پیاده سازی شده است.

2- در این قسمت با استفاده از پکیج skleran رگرسیون lasso و ridge پیاده سازی شد که نتایج در ادامه ارائه شده است:

الف) Ridg:

```
print('ridge reggression score train:', model_ridge.score(X_train, y_train))
print('ridge reggression R^2: test', metrics.r2_score(y_test, y_pred_test_ridge))

ridge regression score train: 0.31636591645465806
ridge regression R^2: test 0.3178220151104242
```

ب) Lasso :

```
print('lasso regression score train:', model_lasso.score(X_train, y_train))
print('lasso regression R^2: test', metrics.r2_score(y_test, y_pred_test_lasso))
```

```
lasso regression score train: 0.26767139069296775
lasso regression R^2: test 0.27167470403297456
```

همان‌طور که از تصاویر بالا قابل برداشت است نتایج حاصل از مدل Ridge به مدل LinearRegression بسیار نزدیک است اما مدل Lasso عملکرد مناسبی نداشته است.

4- در این قسمت مدل رگرسیون خطی با استفاده از خطای Absolute Error آموزش داده شده و نتایج آن در ادامه ارائه می‌شود:

```
y_pre_test_ae = model_ae.predict(X_test)
y_pre_test_ae = y_pre_test_ae
print('linear regression from scrach (ae) R^2: test', metrics.r2_score(y_test, y_pre_test_ae))
```

```
linear regression from scrach (ae) R^2: test -251903214.89325467
```

همان‌طور که مشخص می‌باشد نتایج حاصل از استفاده از Absolute Error به عنوان تابع هزینه مناسب نمی‌باشد.

تمامی مراحل به طور کامل در فایل ژوپیتر با توضیحات انجام شده

باتشکر محمدسعید حیدری

