چکیده:

هدف این پیاده سازی پیشبینی محبوبیت اخبار به صورت یادگیری ماشین نظارت شده با استفاده از مدل های رگرشن معین می باشد. مراحل انجام این کار شامل بررسی داده و تمیز کردن آن ، مطرح کردن تست های آماری و بررسی آن ها و سپس تعریف مدل های رگرشن و بررسی خطای آن ها به روش های متفاوت بر روی دیتای تست می باشد تا بهترین مدل با کمترین خطا برای پیشبینی انتخاب شود.

مقدمه :

در این پروسه با هدف امتحان lasso و ridge regression به عنوان مدل و یافتن بهترین حالت با بهره گیری از اسکیل کردن داده(min-max and standatd normalization) ،اعمال فیچر سلکشن ها (sequential forward and backward feature selection) و فیچر های چندجمله ای. مقیاس تخمین ارور پیشبینی در این پژوهش r2_score می باشد و در عموم حالت ها mse هم محاسبه شده است. در ابتدای کار پیش از اعمال مدل یادگیری ماشین ، برای تمیز کردن داده ها ابتدا اخباری که خالی بودند را پاک کردیم و سپس چند فیچر که به طور واضح تری فقط مدل پیچیده تری را نتیجه می دادند حذف کردیم و سپس تست های آماری که حدس ما بودند را مطرح کردیم و آن ها را به طور نموداری بررسی کردیم .

روش ها:

در ابتدا داده را در دیتافریم df لود می کنیم ، در بخش EDA نیاز است که بر دیتای خود احاطه داشته باشیم و آن را بفهمیم . به این منظور تعداد فیچر ها و سایز داده را با دستور df.shape به دسست می آوریم . مشاهده می کنیم که دیتای ما شامل 61 فیچر و 39644 داده می باشد. این تعداد فیچر ها در ابتدا شمی از این که مدل های رگرشنی که در ادامه قرار است پیاده کنیم احتمالا دچار اندرفیت شوند ، می دهد.

تمام ستون های دیتافریم یا تمام فیچر های دیتای ما به شرح زیر می باشد:

```
Index(['url', ' timedelta', ' n_tokens_title', ' n_tokens_content',
       ' n_unique_tokens', ' n_non_stop_words', ' n_non_stop_unique_tokens',
       ' num_hrefs', ' num_self_hrefs', ' num_imgs', ' num_videos',
       ' average_token_length', ' num_keywords', ' data_channel_is_lifestyle',
       ' data_channel_is_entertainment', ' data_channel_is_bus',
       ' data_channel_is_socmed', ' data_channel_is_tech',
       ' data_channel_is_world', ' kw_min_min', ' kw_max_min', ' kw_avg_min',
       ' kw_min_max', ' kw_max_max', ' kw_avg_max', ' kw_min_avg',
       ' kw_max_avg', ' kw_avg_avg', ' self_reference_min_shares',
       ' self_reference_max_shares', ' self_reference_avg_sharess',
       ' weekday_is_monday', ' weekday_is_tuesday', ' weekday_is_wednesday',
       ' weekday_is_thursday', ' weekday_is_friday', ' weekday_is_saturday',
       ' weekday_is_sunday', ' is_weekend', ' LDA_00', ' LDA_01', ' LDA_02',
       ' LDA_03', ' LDA_04', ' global_subjectivity',
       ' global_sentiment_polarity', ' global_rate_positive_words',
       ' global_rate_negative_words', ' rate_positive_words',
       ' rate_negative_words', ' avg_positive_polarity',
       ' min_positive_polarity', ' max_positive_polarity',
       ' avg_negative_polarity', ' min_negative_polarity',
       ' max_negative_polarity', ' title_subjectivity',
       ' title_sentiment_polarity', ' abs_title_subjectivity',
       ' abs_title_sentiment_polarity', ' shares'],
      dtype='object')
```

این ستون ها نشان می دهد برای ترین مدل ماشین لرنینگ target همان فیچر shares می باشد که نشان دهنده محبوبیت است و باقی فیچر های مدل هستند.

بر اساس نوع این ستون ها تست های آماری را که حدس ما از ارتباط میان آن ها می باشد مطرح می کنیم:

1- اخباری که محبوبیت بیشتری در آخرهفته ها دارند در چنل شبکه های اجتماعی بییشینه است.
2- اخباری که شامل تعداد بیشتری از عکس ها و ویدیو ها هستند محبوبیت بیشتری دارند.
3- در حال حاضر نسبت به قدیم اخباری که فضای احساسی بیشتری دارند محبوب ترند.
4- در حال حاضر نسبت به قدیم نرخ استفاده از کلمات منفی در اخبار کاهش یافته است.
5- اگر سابجکتیو بودن عنوان خبر بیشتر باشد احساسی تر بودن آن هم بیشتر است.

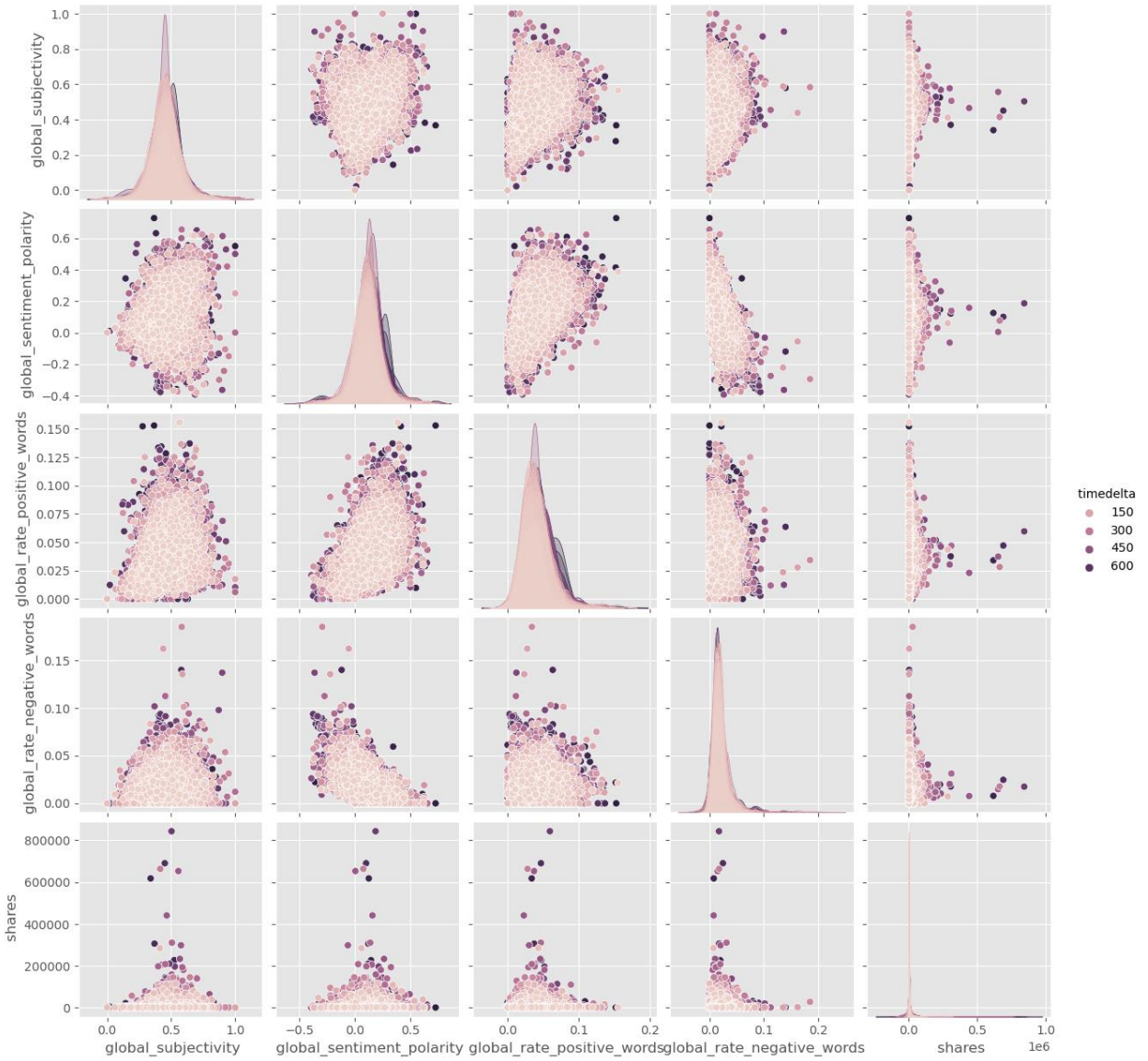
در ادامه به بررسی سه فرض اول می پردازیم.

فرض تست اول: میانگین shares در ستون هایی که هر دیتا چنل و آخرهفته بودن 1 است را بدست می آوریم.

```
entertainment_weekend = 3647.272925764192
tech_weekend = 3753.14332247557
socmed_weekend = 3948.0788643533124
world_weekend = 2679.4235727440146
bus_weekend = 3909.9897610921503
```

نتایج نشان می دهد بیشترین محبوبیت اخبار از چنل شبکه های مجازی است اما اختلاف بین محبوبیت انواع چنل ها چندان زیاد نیست.

نمودار برخی گروه فیچر ها:



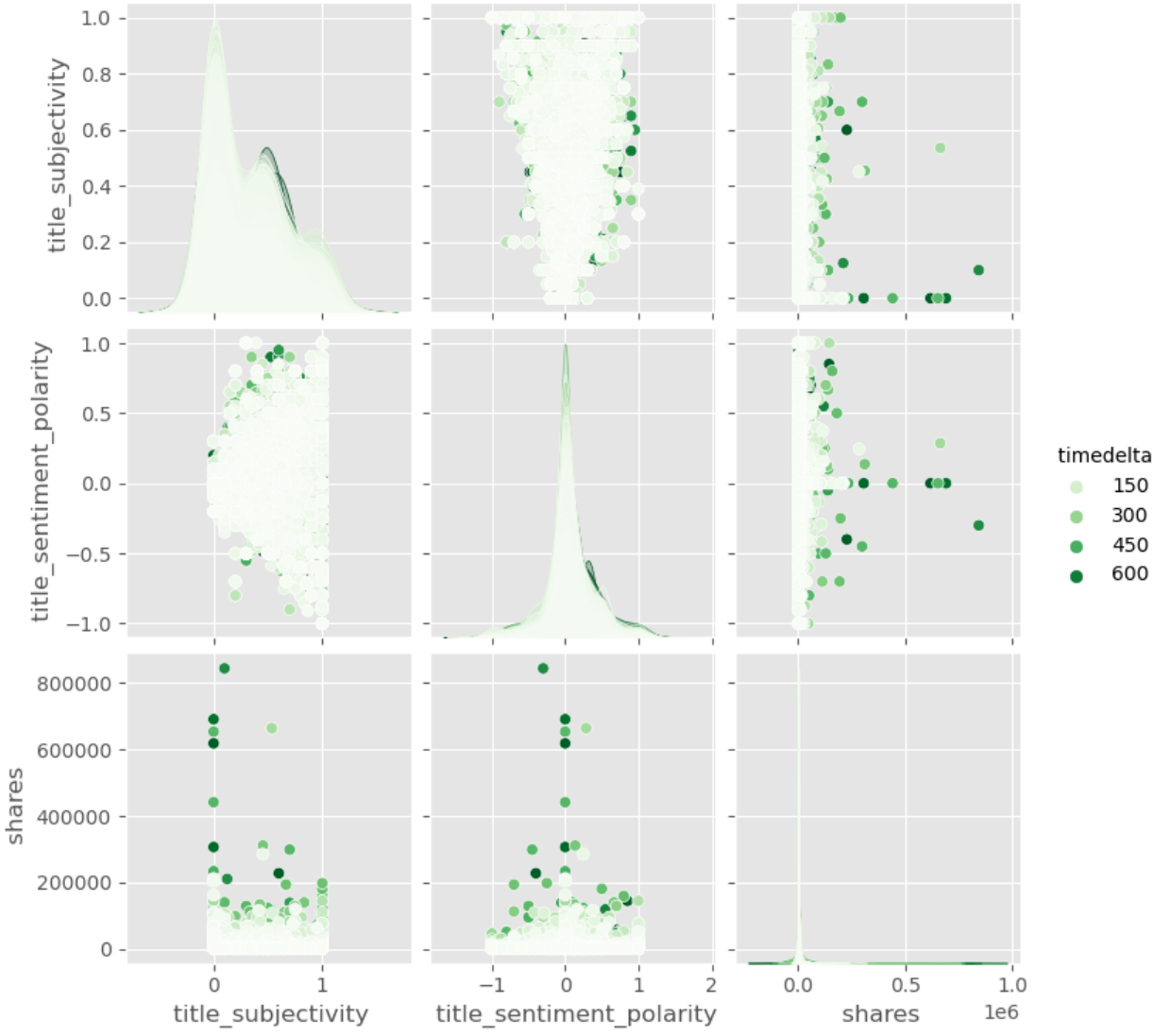
در این نمودار scatter فیچرهای تعداد کلمات مثبت و منفی ، فضای احساسی و تعداد شیر کردن ها. که بر حسب زمان تغیر رنگ می دهند به این ترتیب که رنگ پر رنگ تر نشان دهنده قدیمی تر بودن است. این نمودار شامل پاسخ ما به تست فرض سوم می باشد . با دقت به نمودار shares/global_sentiment_polarity مشاهده می کنیم که اخباری که فضای احساسی ندارند یا خنثی هستند محبوبیت بیشتری دارند و از آنجایی که عموم اخبار قدیمی تر حالت خنثی ای داشتند ، از محبوبیت بیشتری برخوردار بودند.

نتایج دیگری که از این نمودار می توان گرفت :

از مقایسه نمودار sharesو سایر فیچر ها مشاهده می کنیم که گویا داده های قدیمی تر اکثرا به طور میانگین از محبوبیت بالاتری برخوردار بودند اما با توجه به این نکته که تعداد آن ها به نسبت داده های جدید تر بسیار کمتر است می توان حدس زد که میزان محبوبیت بالاتر هر داده قدیمی بالاتر می رود که این را می توان به عنوان فرض آماری دیگری هم تست کرد.

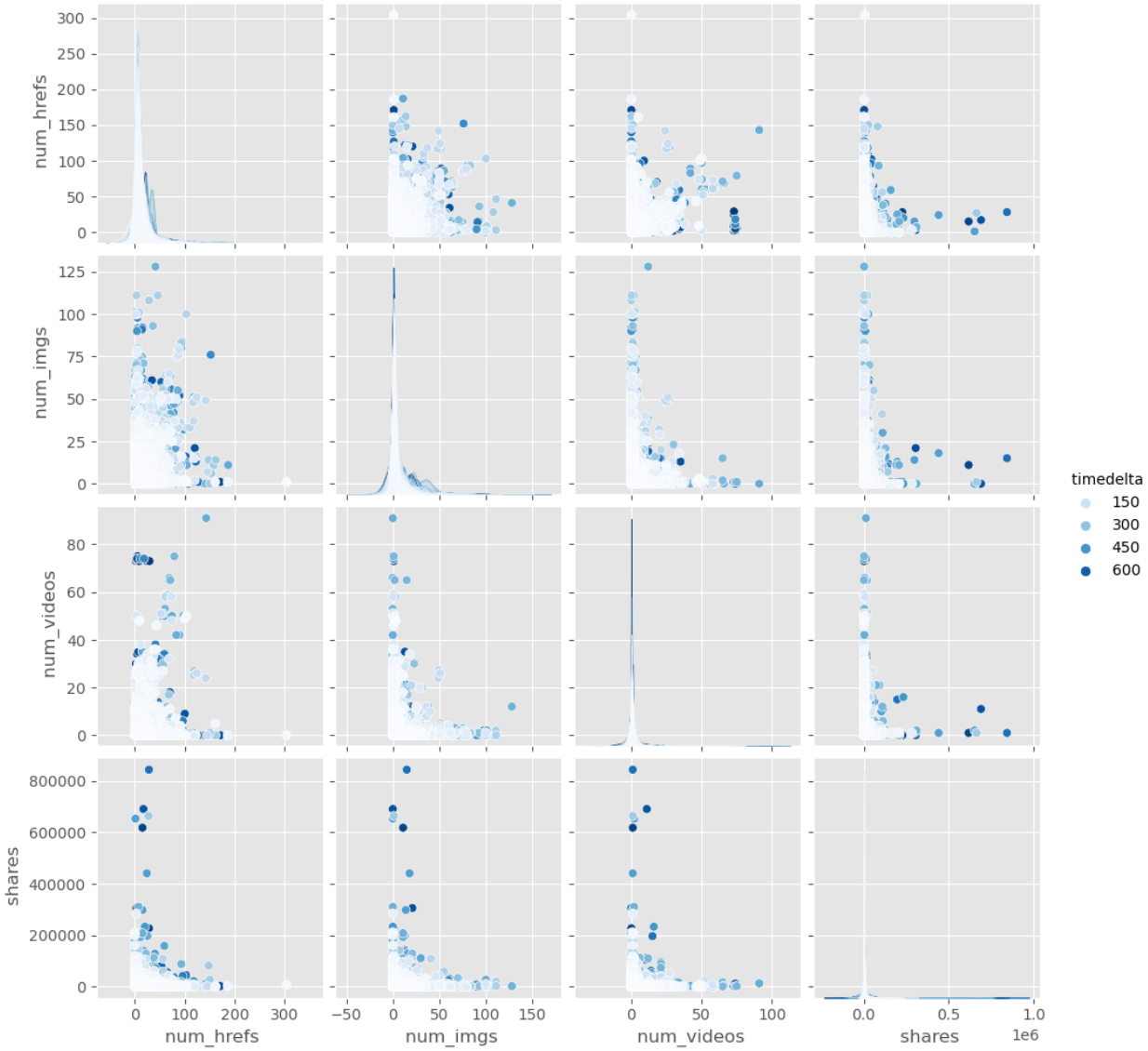
از سایر بخش های نمودار به جز ارتباط عکس بدیهی بین تعداد کلمات مثبت و منفی نکته دیگری نمیتوان دریافت.

نموداری مشابه نمودار قبل برای گروه دیگری از فیچر ها مشمول title_sentiment_polarity و title_subjectivity و shares را رسم می کنیم تا تفاوت بین این فیچر ها در حالت انحصار به عنوان خبر و حالت کلی را مشاهده کنیم.



طبق مشاهدات بر اساس این که نمدار ها تقریبا الگوی مشابهی دارند ، این دو حالت تفاوت چندانی ندارد.

در گروه بعدی فیچر های num_hrefs, num_imgs, num_videos و sharesرا به همین نحو نمودارکشی، بررسی می کنیم.



در این نمودار می توانیم تست فرض دوم را بررسی کنیم که به ارتباط مستقیم بین تعداد عکس ها و ویدیو ها در خبر و محبوبیت آن اشاره داشت. طبق نمودار shares/num_videos و shares/num_imgs به عکس این امر می رسیم که بنظر می آید هرچه تعداد عکس و فیلم کمتر باشد آن خبر محبوبیت بیشتری دارد.

نکات دیگری که از این نمودار می توان دریافت ارتباط عکس تعداد لینک ها و محبوبیت می باشد و ارتباط عکس تعداد عکس ها و لینک ها و تعداد ویدیو ها و عکس ها.

در قسمت بعدی به تمیز کردن دیتا می پردازیم تا دیتای بهتری برای مدل سازی داشته باشیم. برای این منظور ابتدا به بررسی اینکه آیا داده گم شده ای در دیتا وجود دارد یا خیر پرداختیم و کامل بودن دیتا سپس داده هایی که خبر خالی بوند و پرت بودند را حذف نمودیم که در نتیجه آن تعداد داده های ما به 38463 کاهش یافت سپس ستون هایی را از دیتافریم حذف کردیم این ستون ها شامل url و timedelta بودند که non predictive می باشند و مقادیر ماکسیسم و مینیمم قطبیت مثبت و منفی زیرا که با وجود داشتن میانگین این مقادیر به عنوان فیچر وجود ماکسیمم و مینیمم صرفا پیچیدگی دیتا را به همراه خواهد

داشت و حذف روز انتشار Sunday یا Saturday بودن که با وجود داشتن فیچر آخرهفته بودن ، حذف این دو هم به نظر آمد که دیتای بهتری به ما خواهد داد.

حال کل دیتا را به دو بخش X و y تقسیم که می کنیم که y همان targetاست پس فقط حاوی فیچر shares می باشد و باقی دیتا فیچر های X می شوند. سپس به کمک کتابخانه scikit-learn 20 درصد داده ها را برای تست بر می داریم و باقی را برای ترین مدل استفاده می کنیم.

مدل های ridgeو lasso را از کتابخانه scikit-learn ایمپورت می کنیم و در mse و r2_score را برای آن ها بکار می گیریم:

خروجی برای ridge:

```
mse : 54815087.948618345
r2_score 0.02974452233724545
```

خروجی برای lasso:

```
mse : 7403.905370657218
r2_score 0.02969625675763421
```

مشاهده می کنیم که میزان ارور mse برای lasso کمتر است اما ridge مقدار r2_score بیشتری دارد.

حال برای بهتر شدن نتایج min maxو standard normalization را بر روی دیتا اعمال می کنیم.

خروجی برای min max به شرح زیر می باشد:

```
ridge mse: 7.709063617278568e-05
ridge r2_score: 0.02959999905737065
lasso mse: 7.709063617278568e-05
lasso r2_score: -0.0011376682534236515
```

همانطور که میبینیم اعمال min max بر روی هیچکدام تاثیر مثبتی نگذاشته است . اما تاثیر منفی آن روی lasso بیشتر است.

خروجی برای standard normalization:

```
ridge mse: 0.40836304158061276
ridge r2_score: 0.02974512383670247
lasso mse: 0.40836304158061276
lasso r2_score: 0.031006286813436423
```

دریافت نتایج بهتر نسبت به حالتی که فیچر ها اسکیل نشده بودند در میبایم که اعمال standard normalization بر روی مدل lasso تا اینجا بهترین نتیجه را داشته.

در قسمت بعدی دو متد forward feature selection و backward feature selection را که در فایل Sequentialfeatureselection.py پیاده سازی کردیم را بر روی دیتا اعمال می کند .

خروجی forward:

```
ridge mse: 56548712.706162535
ridge r2_score: -0.0009415347351378056
```

خروجی backward:

```
ridge mse: 54815053.96659359
ridge r2_score: 0.029745123836702914
```

با دریافت نتایج نابهینه در پیاده سازی forward به چند احتمال می رسیم که پررنگ ترین و منطقی ترین آن ها مناسب نبودن مدل رگرشن برای این دیتا می باشد احتمال دیگر اشکال در پیاده سازی است و احتمال دیگر scale نبودن دیتا. اینکه backward هم نتایج بهتری را می دهد نشان دهنده این است که فیچر سلکشن بهتری برای این دیتا می باشد.

حال برای lasso از forward feature selection کتابخانه scikit-learn استفاده می کنیم که خروجی آن به شرح زیر است:

```
lasso mse: 66433588.45456741
lasso r2_score: 0.031044360879728727
```

که تا اینجا بهترین خروجی را به ما داده است. البته متد این کتابختنه این تفاوت را دارد که در ورودی تعداد فیچر هایی که می خواهیم باقی بماند را مشخص می کنیم.

در ادامه فیچر سلکشن دیگری هم از کتابخانه scikit-learn استفاده می کنیم که SelectKBest نام دارد . مقدار r2_score را برای lasso حساب می کنیم.

```
0.031053987980391162
```

به مقدار بهتری رسیده ایم.

سپس برای lasso از فیچر های چندجمله ای استفاده می کنیم ، alpha را مساوی 0.1 و درجه را 2 در نظر می گیریم.
R2_score آن به شرح زیر است.

```
0.02663647712177819
```

با وجود اینکه انتظار می رفت استفاده از فیچر های چندجمله ای نتیجه را بهتری کند اما این اتفاق نیوفتاد ، از جکله کارهایی که برای بررسی دوباره می توان کرد تغییر درجه و آلفا است.

با تغییر درجه ریزالت بدتر شده که به دلیل اورفیت با افزایش درجه و اندرفیت با کاهش درجه است. اما با افزایش alphaبه مقدار 100 به بهترین ریزالت رسیده ایم:

```
0.032065433801704346
```

این مقدار r2_score بهترین ریزالت ما تا این مرحله با امتحان حالت های مختلف بوده.

می توان شیوه های تخمین ارور دیگری را هم استفاده کرد. برای مثال absolute error که عبارت است از مجموع اختلاف تمام y_test ها وy_pred ها.

برای همین مدل نهایی absolute error آن برابر

```
21942357.190672915
```

می باشد. این مقدار همیشه از mae کمتر است زیرا mae تعریف absolute erro است بر روی تعداد داده ها.

نتیجه:

با هدف های معینی که از ابتدای کار داشتیم که از جمله استفاده از lasso یا rodge regression برای مدل بوده در بهرین حالت به استفاده از lasso به عنوان مدل ، استفاده از standard normalization به عنوان متد اسکل کردن ، SelectKBest به عنوان فیچر سلکشن و بهره گیری از فیچر های چندجمله ای با درجه 2 رسیدیم. یعنی بهترین پیشبینی ما برای دیتای تست 3.2% مطابقت با مقادیر واقعی آن دارد. این مقدار به طور کلی این منظور را می رساند که استفاده از مدل های رگرشن معین شده برای این دیتا اشتباه است.