

هدف از این پروژه این است که بتوانیم قیمت‌های خانه را بر اساس ویژگی‌ها و داده‌هایی که در `training set` است، تخمین بزنیم. این یک مساله رگرسیون است که ما با استفاده از مدل‌ها و الگوریتم‌های مختلف سعی در تخمین زدن دقیق این قیمت‌ها داریم. در اینجا ما از سه مدل استفاده می‌کنیم که عبارت است از `linear regression`، `SVM` و `Randomforest` و بعد از اینکه مدل خود را آموزش دادیم داده‌های تست را به مدل خود می‌دهیم و دقت را بررسی می‌کنیم که در اینجا دقت بر اساس معیار

`RMSE(Root Mean Square Error)` بررسی می‌شود.

برای اینکه بتوانیم مدل خود را بسازیم ابتدا باید عملیات `cleaning` و `preprocessing` را انجام دهیم تا مدل ما با دقت بالا بتواند پیش‌بینی کند و داده اشتباه ندهد.

Preprocessing & cleaning

برای این منظور ما ابتدا باید داده‌های خود را بشناسیم و ویژگی‌ها را به درستی شناسایی کنیم و بعضی از آنها باید حذف شوند و بعضی از آنها باید ترکیب شوند تا ویژگی تازه‌ای بوجود بیاد و به مدل ما در پیش‌بینی کمک کند. در ابتدا ما همبستگی و ارتباط بین ویژگی‌ها را در می‌آوریم تا ببینیم کدام ویژگی بیشترین تاثیر را روی `SalePrice` دارند و بیشتر روی آنها بررسی انجام دهیم برای اینکار از

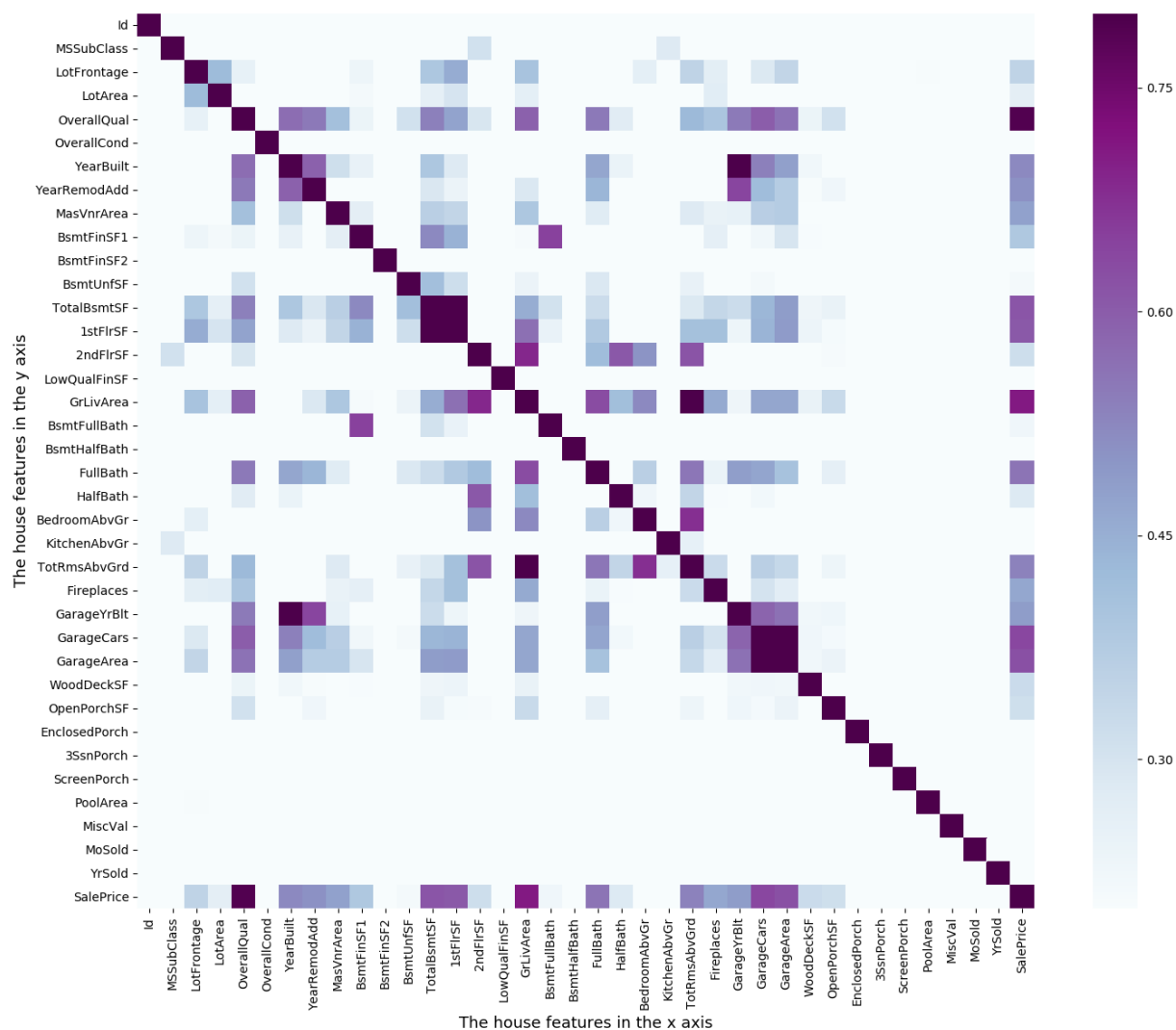
```
correlation_matrice = data_aux.corr()
```

مرحله اول

استفاده می‌کنیم تا ماتریس میزان وابستگی ویژگی‌ها بهم را بیابیم که شکل زیر این ماتریس را نشان می‌دهد.

برای آنکه ما راحت‌تر بتوانیم با ویژگی‌ها کار کنیم و تغییرات را هم بتوانیم در `train` و `test` اعمال کنیم این دو را با هم ترکیب کرده و کار را ادامه می‌دهیم.

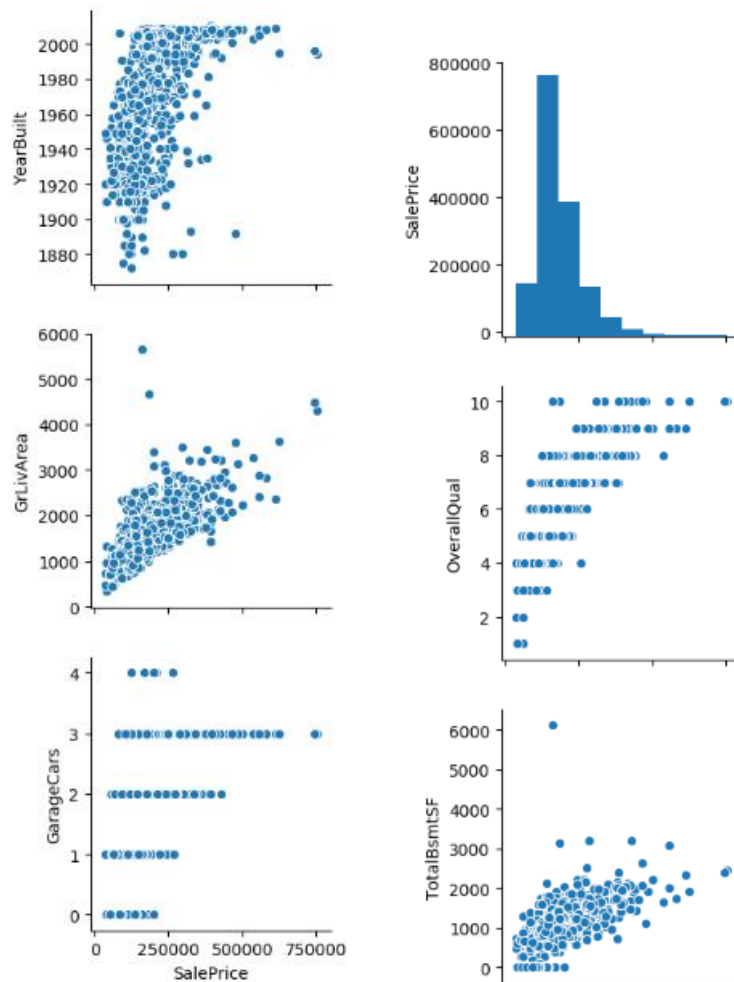
```
all_data = pd.concat(objs=[train_data, test_data], axis=0)
```



همان طور که مشخص است تعدادی از ویژگی ها ارتباط و همبستگی بیشتری با saleprice دارند که این شش ویژگی زیر از همه ارتباطشان بیشتر است.

OverallQual, GrLivArea, TotalBsmtSF, 1stFLrSF, GarageCars, GrageArea

حال برای این ویژگی ها ما نمودارشان را با `saleprice` رسم میکنیم تا ببینیم به چه گونه پخش شده اند و ایا داده های پرتی وجود دارد یا نه. شکل های زیر این شش نمودار را نشان میدهد که با بررسی هر کدام عملیات مناسب را اتخاذ میکنیم.



مرحله دوم

بعد از اینکه این شش نمودار را کشیدیم و شکل ها را بررسی کردیم میفهمیم که دو ویژگی `GarageArea` و `GlivArea` داده پرت دارند و باید آن ها را حذف کنیم که به صورت زیر حذفشان میکنیم.

```
train_data = train_data[train_data['GrLivArea'] < 4500]
train_data = train_data[train_data['GarageArea'] < 1200]
```

مرحله سوم

ویژگی هایی که مقادیر مشخص ندارند را باید پیدا کنیم و به صورت مناسب پر کنیم. با استفاده از دستور زیر ویژگی هایی که missing value دارند را پیدا میکنیم و بر اساس آنکه object یا Int64 هستند با مقدار مناسب پر میکنیم.

```
[all_data.columns[all_data.isna().any()].tolist()]
```

که این ویژگی ها مقادیر نامشخص داشتند.

```
Shape of training set: (1458, 82)
Missing values before remove NA:
Index(['LotFrontage', 'Alley', 'MasVnrType', 'MasVnrArea', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Electrical', 'FireplaceQu', 'GarageType', 'GarageYrBlt',
       'GarageFinish', 'GarageQual', 'GarageCond', 'PoolQC', 'Fence',
       'MiscFeature'],
      dtype='object')
```

برای object ها از دستور زیر استفاده میکنیم.

```
all_data['Fence'] = all_data['Fence'].fillna('None')
```

و برای ویژگی های عددی به صورت زیر:

```
all_data['GarageArea'] = all_data['GarageArea'].fillna(0)
```

برای ویژگی های که در توضیحات missing value توضیح داده نشده بود و برای اینکه ما این مقادیر نامشخص را پر کنیم تصمیم گرفتیم که از Mode برای پر کردن استفاده کنیم.

```
all_data['SaleType'] = all_data['SaleType'].fillna(all_data['SaleType'].mode()[0])
```

مرحله چهارم

حال به این موضوع میپردازیم که آیا باید ویژگی ای حذف شود یا اضافه شود. برای این کار باید ویژگی ها را به خوبی بررسی کرد تا ببینیم کدام ویژگی ها به درد نمیخورند و در پیش بینی اثری ندارند و همچنین در کمتر شدن ابعاد کمک میکند. ویژگی street و utilities به دلیل آنکه فقط یک مقدار دارند میتوانند حذف شوند

و ID ، Garagecars و garageYrBlt هم چون اطلاعاتی به ما نمیدهند باید حذف شوند و کمکی به ما نخواهند کرد پس با استفاده از دستور drop ستون های آنها را حذف میکنیم. همچنین alley به دلیل آنکه مقادیر نا مشخص زیادی دارد و بالای ۹۰ درصد مقادیر آن نا مشخص است میتواند حذف شود.

```
all_data.drop(['Street'], axis=1, inplace=True)
```

برای آنکه ویژگی ای اضافه شود باید بررسی کنیم چه ترکیبی از ویژگی ها میتواند ما را کمک کند. این ویژگی که اندازه کل خانه را به ما بدهد وجود ندارد و میتواند به ما بسیار کمک کند چون هر چه اندازه خانه بزرگتر باشد ارزش خانه بیشتر میشود پس با استفاده از دستور زیر این ویژگی را اضافه کردیم.

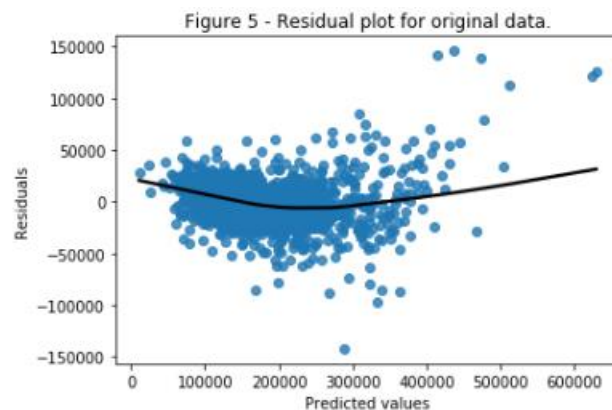
```
all_data['TotalSF'] = all_data['TotalBsmtSF'] + all_data['1stFlrSF'] + all_data['2ndFlrSF'] +  
all_data['GarageArea']
```

مرحله پنجم

در این مرحله باید عملیات encoding را انجام دهیم به دلیل آنکه ما تایپ های مختلفی داریم و همه را باید یکپارچه کنیم تا مدل های ما بدرستی کار کنند برای این کار از one-hot encoding استفاده میکنیم و ابتدا ویژگی هایی که به صورت object هستند را انکد میکنیم تا به صورت عددی در بیایند و سپس کل مقادیر را به one-hot encoder میدهم که با استفاده از تابع get_dummies() این کار صورت میپذیرد.

Regression models

مدل اول ما رگرسیون خطی است که با استفاده از کتابخانه `sklearn.linear_model` از رگرسیون خطی استفاده کردیم ولی دقت ما خوب نشد و بعد از اینکه داده های `train` را به مدل خود دادیم و مدل ما آموزش دید، داده های تست را دادیم و `RMSE` ما 0.65363 شد که دقت مناسبی نیست و این به دلیل این است که ویژگی های ما با مقادیر پیشبینی شده رابطه خطی ندارند و شکل به صورت زیر است .



همان طور که مشاهده میشود داده های ما غیر خطی هستند و این روش خوب جواب نداد.

حال مدل بعدی ما `SVM` است که این مدل دقت ما خیلی بهتر کرد و ما با استفاده از کتابخانه `sklearn.svm`

این کار را انجام دادیم و دقت بدست آمده در شکل زیر مشخص است.

3716	pouya khorsandi		0.19264	2	-10s
------	-----------------	--	---------	---	------

Your Best Entry

You advanced 637 places on the leaderboard!

Your submission scored 0.19264, which is an improvement of your previous score of 0.65363. Great job!

Tweet this!


همان طور که در شکل بالا مشخص است ما از خطای 0.65 به خطای 0.19 رسیدیم که اختلاف فاحشی وجود دارد.

مدل بعدی ما `Randomforest` است جزو الگوریتم های `ensemble` است و از چندین درخت استفاده میکند تا داده را تخمین بزند. برای اینکه با این مدل کار کنیم ابتدا پارامتر های آن را ست میکنیم.

```
n_estimators=750, criterion='mse'
```

```
max_depth=15, min_samples_split=5, min_samples_leaf=1
```


با ست کردن این پارامتر ها حال میتوانیم از این مدل استفاده کنیم که از کتابخانه `sklearn.ensemble` استفاده از کرده و `Randomforest` را اجرا میکنیم. با دادن داد های تست دقت زیر بدست آمد.

2671	pouya khorsandi		0.14261	3	-10s
------	-----------------	---	---------	---	------

Your Best Entry ↑

You advanced 1,046 places on the leaderboard!

Your submission scored 0.14261, which is an improvement of your previous score of 0.19264. Great job!

 **Tweet this!**

دقت ما به 0.14 رسید که به نسبت دقت بدی نیست و از مدل قبل پیشرفت بیشتری داشتیم.