Report

Author: Xishi Chen

1. Data Process

From building_bio.xlsx, I extracted some predictors: ID, the number of units, the number of stories, built year and transportation_index, stored in matrix data.

$$trasp\_index = \sum_{i=1} \frac{1}{trasp_i}$$

Then from the folder apt_detail, I used the fisrt line of every csv file(if there is no useful information in the first line then choose the second line) and combined them into a matrix. From this matrix, I took information, like the number of bedrooms and bathrooms, the area, the price, the price is for rent or for sold, then got matrix data3.

Merge data3 and data into data_final(-999 means NAN in data.)

In [513]:
```
1 data_final.head()
```

Out[513]:

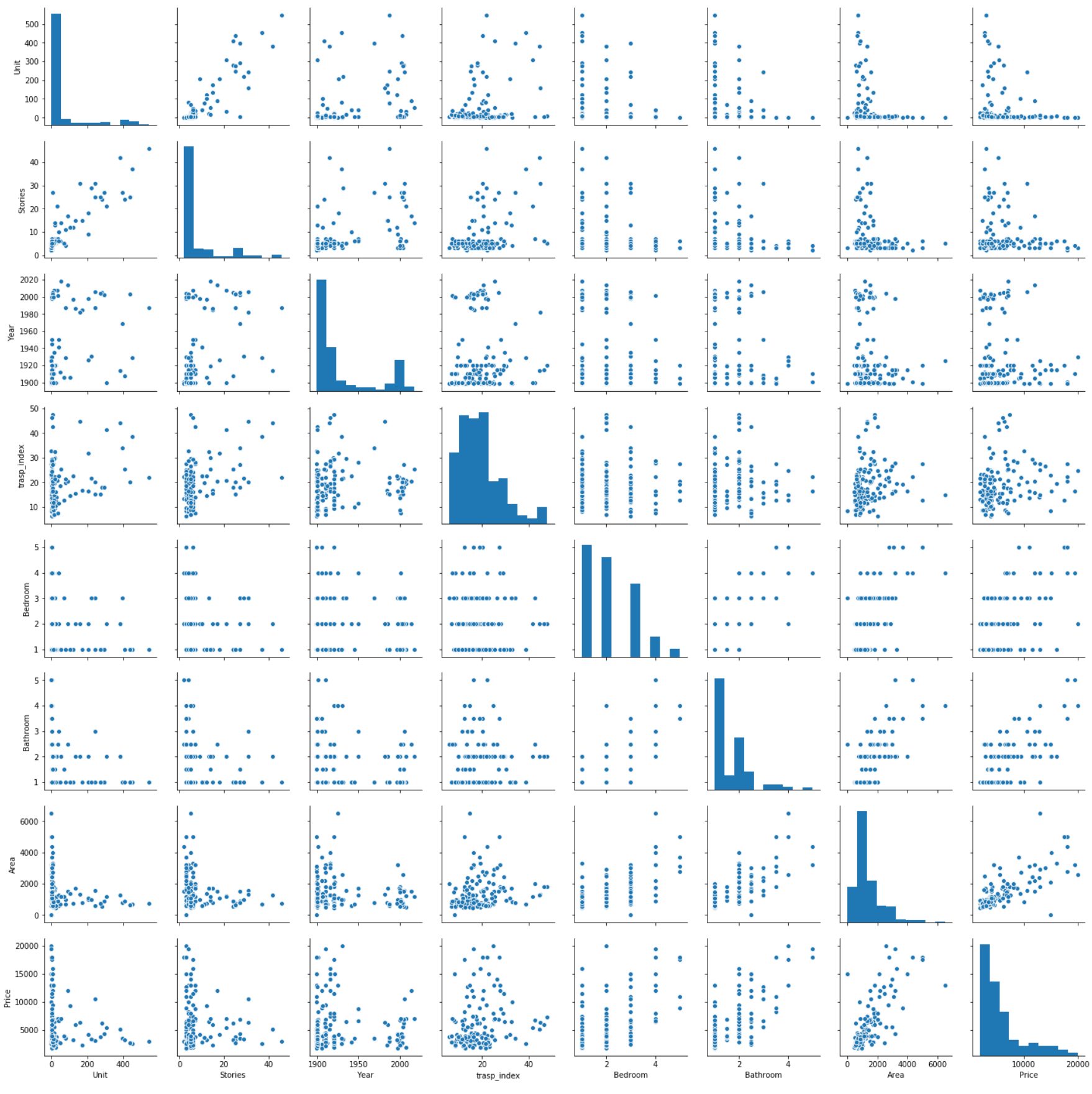
	id	Unit	Stories	Year	Type	trasp1	trasp2	trasp3	trasp4	trasp5	trasp_index	Price	sold_or_rent	Bedroom	Bathroom	Area
5	70968	2.0	3.0	1901.0	Rental Building	0.440	0.46	0.67	0.71	0.72	8.736517	6500.0	Rent	4.0	2.5	1700.0
6	70968	2.0	3.0	1901.0	Rental Building	0.440	0.46	0.67	0.71	0.72	8.736517	6500.0	Rent	4.0	2.5	1700.0
13	71080	3.0	5.0	1899.0	Rental Building	0.130	0.22	0.22	0.30	0.33	23.146853	2000.0	Rent	1.0	1.0	650.0
17	71118	1.0	3.0	1930.0	Two-Family Home	0.130	0.17	0.23	0.29	0.30	24.704096	19950.0	Rent	2.0	4.0	2600.0
19	71132	2.0	5.0	1910.0	Rental Building	0.095	0.25	0.25	0.28	0.33	25.128047	6850.0	Rent	1.0	2.0	1500.0

2.Data Visulization

I considered some houses with extremely high price as outliers and dropped them. Then ploted the scatter plot.

In [517]:
```
1 #explore data
2 X=data_final[['Unit','Stories','Year','trasp_index','Bedroom','Bathroom','Area','Price']]
3 Y=data_final['Price']
4 sns.pairplot(X)
```

Out[517]: `<seaborn.axisgrid.PairGrid at 0x1a2a2e4518>`



3. linear Model

In order to evaluate model, I chose the log error

$$logerror = \frac{\sum_{i=1} |\log(true_i) - \log(estimate_i)|}{n}$$

From the scatter plot above, I applied the following linear model:

$$Price \sim \frac{1}{Unit} + \frac{1}{Stories} + Year + Year^2 + transpindex + Bedroom + bathroom + Area$$

In [557]:
```
X=data_final[['Year','trasp_index','Bedroom','Bathroom','Area']]
X['Year2']=[pow(i,2) for i in data_final['Year']]
#X['Unit']=data_final['Unit']
X['Unit-inv']=1/data_final['Unit']
X['Stories-inv']=1/data_final['Stories']
Y=data_final['Price']
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
reg = LinearRegression().fit(X_train, y_train)
ans=reg.predict(X_test)
log_err(y_test,ans)
```

```
/Users/chenxishi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)

/Users/chenxishi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
  after removing the cwd from sys.path.
/Users/chenxishi/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
  """
```

Out[557]:
```
0.31331071902882596
```

In [552]:
```
ans[0:10] #estimate price
```

Out[552]:
```
array([ 3349.25112812,  1936.49620416,  3654.45393552,  4782.3198531 ,
        5345.40298828,  3944.20893383,  5506.26588184,  6575.76787103,
        2436.43176311, 14927.85940909])
```

In [554]:
```
y_test[0:10] #true price
```

Out[554]:
```
357     3700.0
167     2300.0
581     2800.0
205     3995.0
429    13000.0
267     3500.0
63      5000.0
362     7500.0
546     2600.0
25     18000.0
Name: Price, dtype: float64
```

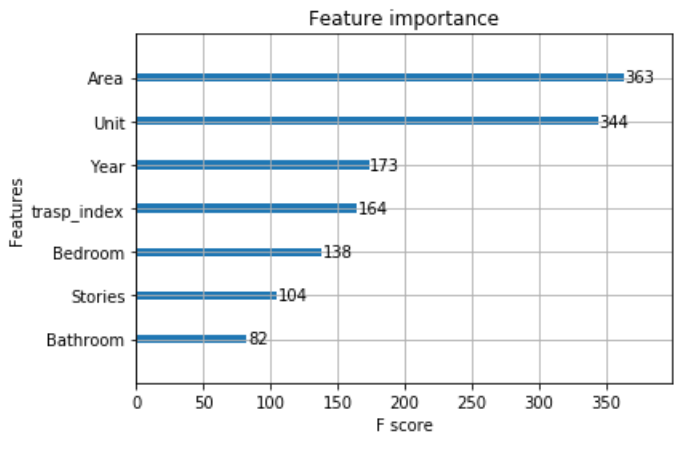
advantage: linear model is efficient when the response value has a strong linear relationship with predict values or the polynomial of predict values. The parameters of the model are easy to explain.

3. XGBoost

I also tried XGBoost model, but it didn't work well. However, from this model, I have some ideas about the importance of different predictors.

In [479]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)

model = xgb.XGBRegressor(max_depth=5, learning_rate=0.1, n_estimators=160, silent=True, objective='reg:gamma')
model.fit(X_train, y_train)

ans = model.predict(X_test)

plot_importance(model)
plt.show()

```

```
/Users/chenxishi/anaconda3/lib/python3.7/site-packages/xgboost/core.py:587: FutureWarning: Series.base is deprecated and will be removed in a future version
  if getattr(data, 'base', None) is not None and \
```



In [481]:
```
1 ans[0:10]
```

Out[481]:
```
array([  2555.3413,  16089.6   ,   9091.725 ,   3750.8875,   6521.8364,
         803016.4   ,  13583.345 ,   8141.1934,   7967.3604,   7114.7847],
      dtype=float32)
```

In [482]:
```
1 y_test[0:10]
```

Out[482]:
```
204       4500.0
257       5900.0
173       2350.0
259       3195.0
169       4495.0
295    1240000.0
251       4400.0
328       3800.0
375    2395000.0
504       3600.0
Name: Price, dtype: float64
```

In [526]:
```
1 log_err(y_test,ans)
```

Out[526]:
```
1.2634879291341337
```

4. Improvment

for data processing:

1. Design a zipcode_index depending on average house price in a small area
2. Apply time series model for houses with a lot of history price to predict price in the future
3. In calculating trasp_index, give different weights to different streets

for linear model:

1. Use leverage (h_{ii}) to decide outliers
2. Use hypothesis test to choose interaction terms
3. Include categorical predict value with 0-1

for XGBoost:

I am still thinking about it.

5. Other questions

1.How would you use this model to make money?

Price predict model can be used to estimate the price of a house and help landlords decide rent.

2.How do you think data science can bring revenue to a start-up company in real estate brokerage industry?

The company can learn the information from data science that which kind of house is easily to be leased and inapprorprriate price for some houses.

3.What is the ideal skillset a data scientist should have in such company?

a. Be familiar with models' theories and applications

b. Be familiar with data processing tools

c. Be good at data visulization

d. Know real estate industry well