07RealEstatePractice

March 18, 2019

1 Please start the test in Kaggle

In [1]:
```
# import pandas as pd
# import os
# import urllib.request
# from sklearn.model_selection import train_test_split
```

In [2]:
```
# if 'data' not in os.listdir():
#     os.mkdir('data')

# if 'df_realestate.csv' not in os.listdir('data'):
#     url = 'https://s3.amazonaws.com/datasets-jeremy/df_realestate.csv'
#     urllib.request.urlretrieve(url, os.path.join('data', 'df_realestate.csv'))

# if 'df_realestate_processed.csv' not in os.listdir('data'):
#     url = 'https://s3.amazonaws.com/datasets-jeremy/df_realestate_processed.csv'
#     urllib.request.urlretrieve(url, os.path.join('data', 'df_realestate_processed.cs
```

In [87]:
```
# # before preprocessing
# file = os.path.join('data', 'df_realestate.csv')
# df_realestate = pd.read_csv(file, encoding='big5')
# df_realestate

# # processed
# path = "data//df_realestate_processed.csv"
# df_realestate_processed = pd.read_csv(path)
# X = df_realestate_processed.drop(["price_per_meter", "total_price"], axis=1)
# Y = df_realestate_processed['total_price']
```

In [88]:
```
# df_realestate['price_per_ping'] = df_realestate[' 單價 (元/平方公尺)']
# showing_cols = [
#     ' 主要建材',
#     ' 主要用途', ' 交易年月日', ' 交易標的', ' 交易筆棟數', ' 備註', ' 土地區段位置/建
#     ' 土地移轉總面積 (平方公尺)', ' 建物型態', ' 建物現況格局-廳', ' 建物現況格局-房',
#     ' 建物現況格局-隔間', ' 建物移轉總面積 (平方公尺)', ' 建築完成年月', ' 有無管理組織
#     ' 總樓層數', ' 車位移轉總面積 (平方公尺)', ' 車位類別', ' 都市土地使用分區', ' 鄉鎮
#     ' 非都市土地使用分區', ' 非都市土地使用編定',
#     'num_of_bus_stations_in_100m', 'income_avg', 'income_var',
```

```
#     'location_type', 'low_use_electricity',
#     'nearest_tarin_station', 'nearest_tarin_station_distance',
#     'lat', 'lng', 'price_per_ping'
# ]

# df_realestate = df_realestate[showing_cols]
```

In [89]:
```
# print(len(df_realestate))
# df_realestate = df_realestate[pd.notnull(df_realestate['price_per_ping'])]
# print(len(df_realestate))
```

```
75203
70670
```

In [90]:
```
# train, test = train_test_split(df_realestate, random_state=4242, test_size=1500)
# train = train.reset_index()
# train.loc[:, 'index'] = train.index
# test = test.reset_index()
# test.loc[:, 'index'] = test.index
# answer = test[['index', 'price_per_ping']]
# submission = test[['index', 'price_per_ping']]
# submission.loc[:, 'price_per_ping'] = 0.0
# test = test.drop('price_per_ping', axis=1)
```

In [94]:
```
# train.to_csv(os.path.join('data', 'train.csv'), index=False)
# test.to_csv(os.path.join('data', 'test.csv'), index=False)
# answer.to_csv(os.path.join('data', 'answer.csv'), index=False)
# submission.to_csv(os.path.join('data', 'submission.csv'), index=False)
```

In [96]:
```
# test_submission = answer
# test_submission['price_per_ping'] = answer['price_per_ping'].mean()
```

In [99]:
```
# test_submission.to_csv(os.path.join('data', 'test_submission.csv'), index=False)
```