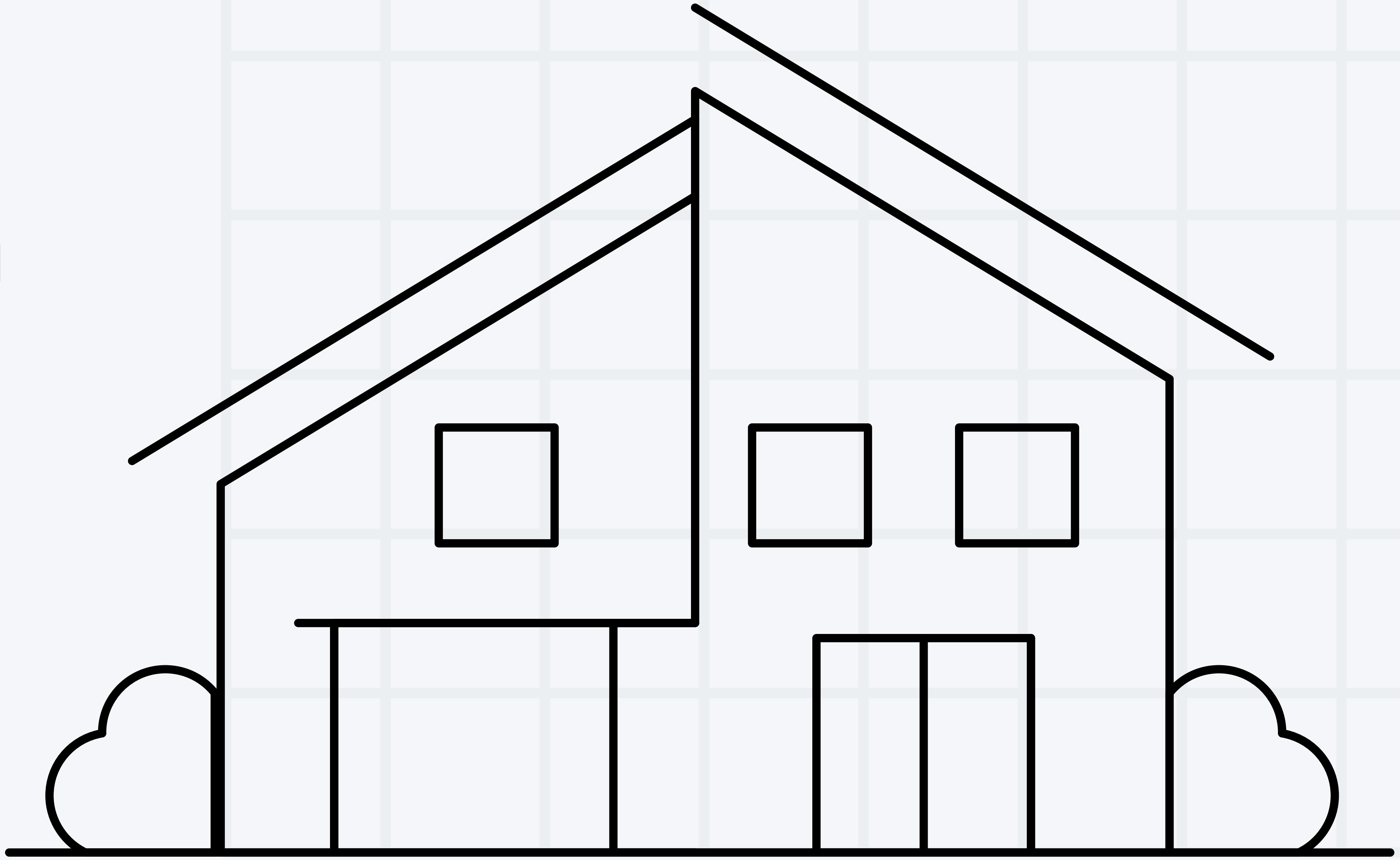
DSi -Project 2

Bangkok housing price



Ponparis Gurdsapsri

Background

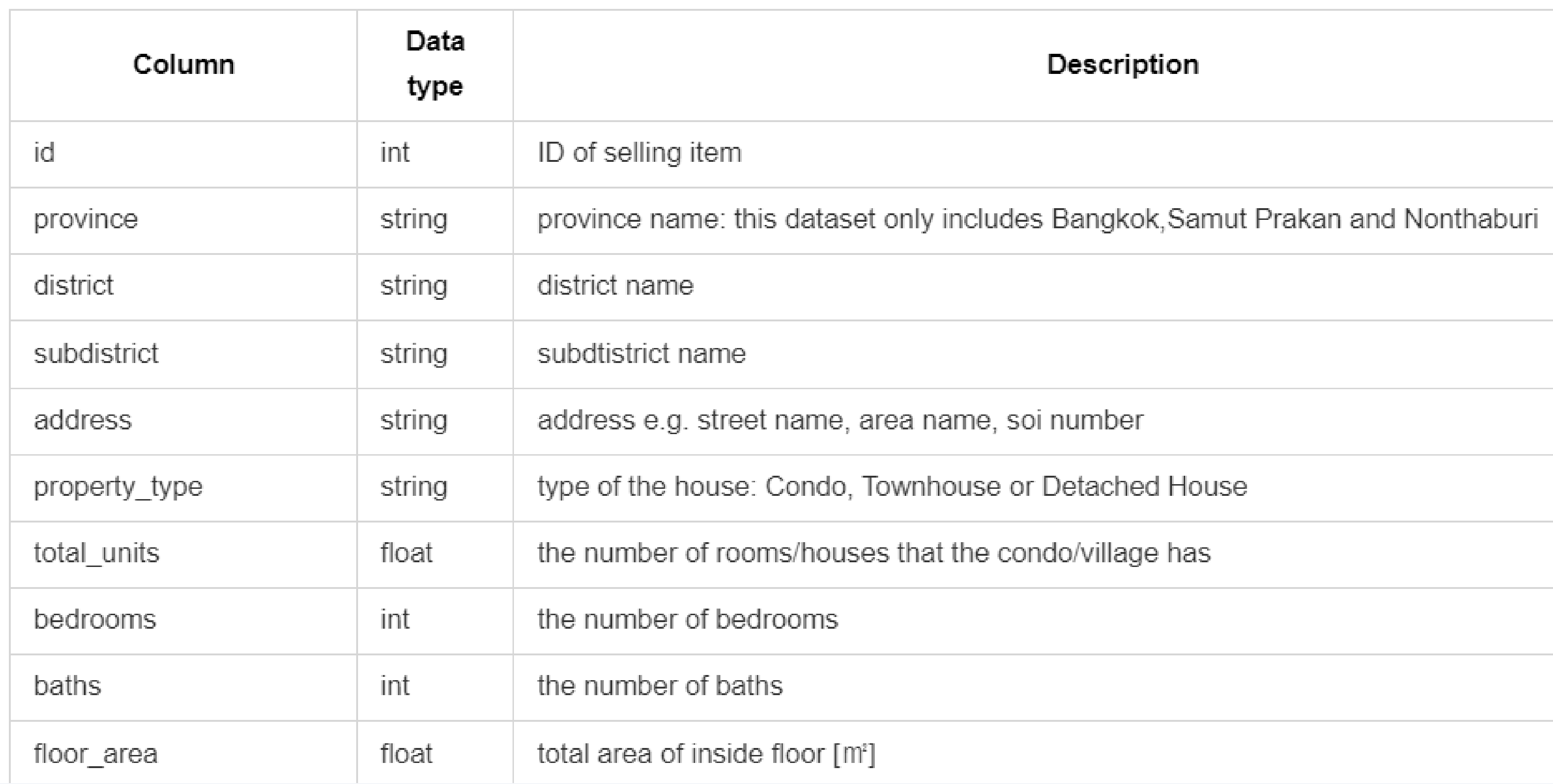
Home mortgage is the essential tool in making the dream of home ownership to reality for millions of individuals and families. As for the financial institution or bank, home mortgage is also one of the lowest risk financial products across the board, but there is still risk. To mitigate the risk, the approval credit should reflect the market value of the real estate. So in case of payment default and the real estate need to be sell by auction, the value should be equivalent or greater than the approved credit, to minimize loss.

Problem

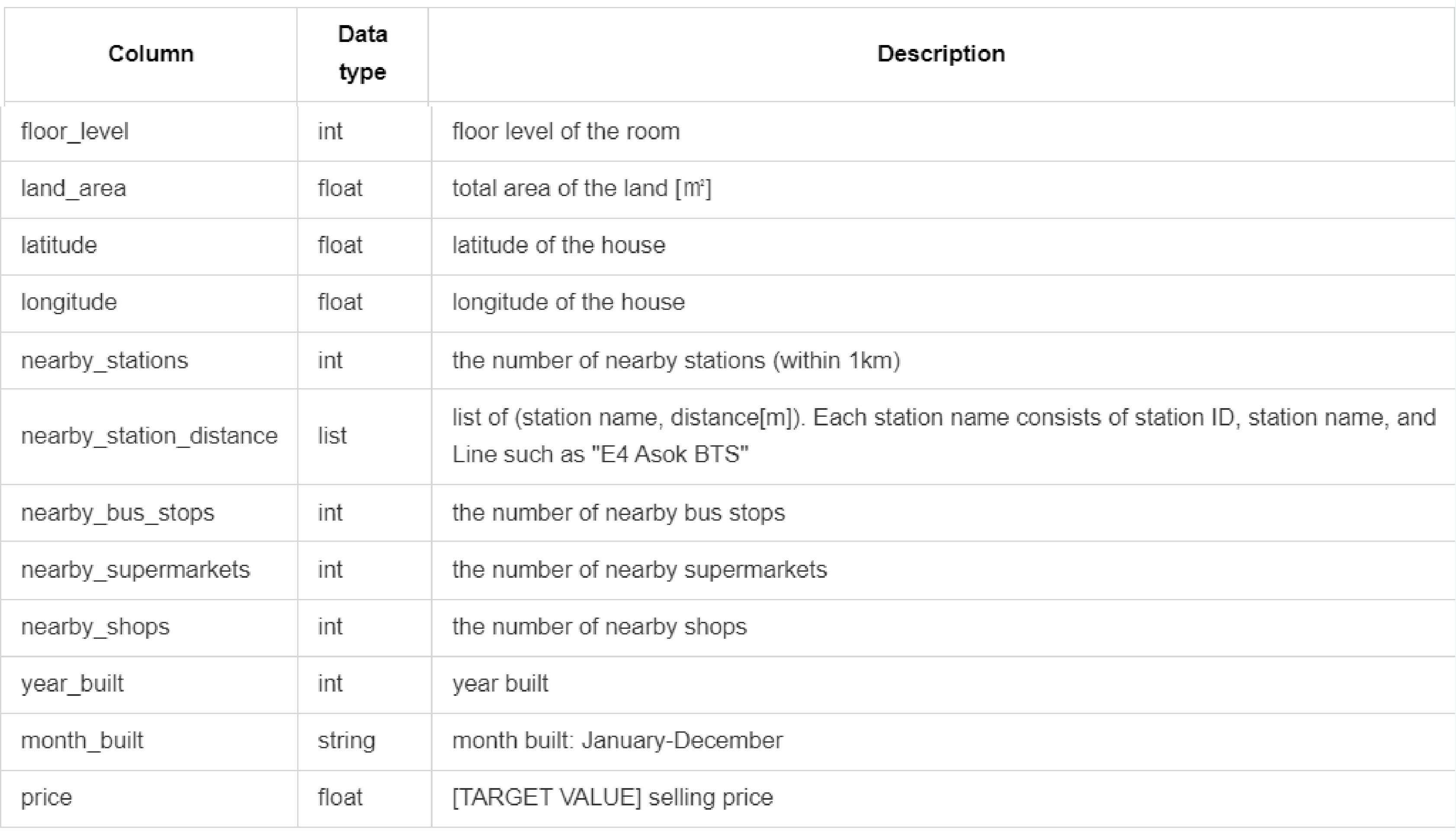
The task is to develop the tool to predict the market price from variety of input, such as location, size, number of train station nearby, etc. So the banker will have estimated value of property, based on the property itself.

Data

12,470 record of housing price in Bangkok, Nonthaburi, and Samutprakarn is provide. The data also contain feature such province name, property type, number of bedrooms, etc. All 22 (+ a property id) features explanation are provided in Data dictionary below.



Column	Data type	Description
id	int	ID of selling item
province	string	province name: this dataset only includes Bangkok,Samut Prakan and Nonthaburi
district	string	district name
subdistrict	string	subdtistrict name
address	string	address e.g. street name, area name, soi number
property_type	string	type of the house: Condo, Townhouse or Detached House
total_units	float	the number of rooms/houses that the condo/village has
bedrooms	int	the number of bedrooms
baths	int	the number of baths
floor_area	float	total area of inside floor [㎡]

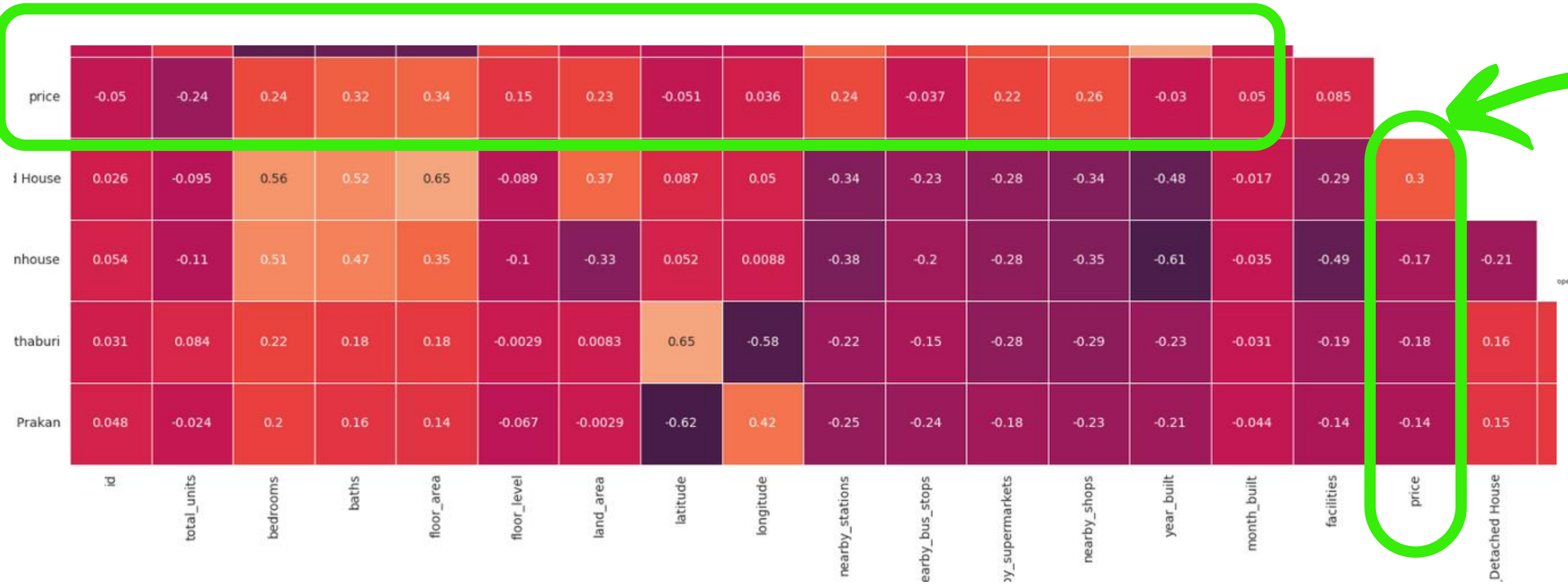


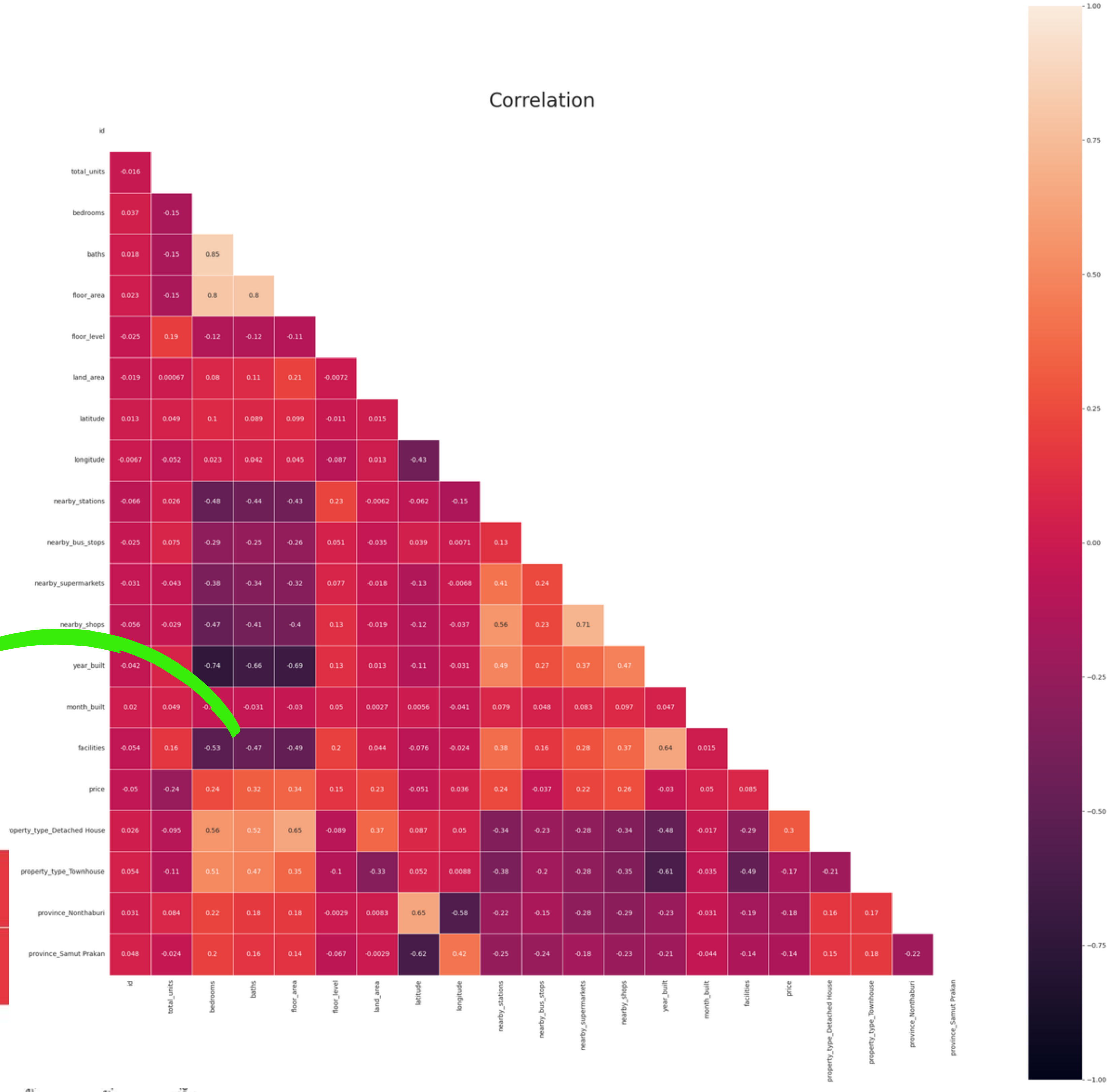
Column	Data type	Description
floor_level	int	floor level of the room
land_area	float	total area of the land [㎡]
latitude	float	latitude of the house
longitude	float	longitude of the house
nearby_stations	int	the number of nearby stations (within 1km)
nearby_station_distance	list	list of (station name, distance[m]). Each station name consists of station ID, station name, and Line such as "E4 Asok BTS"
nearby_bus_stops	int	the number of nearby bus stops
nearby_supermarkets	int	the number of nearby supermarkets
nearby_shops	int	the number of nearby shops
year_built	int	year built
month_built	string	month built: January-December
price	float	[TARGET VALUE] selling price

Data Exploration and Analysis

Finding:

- There is no strong correlation between price and features
- The most correlation efficient is 0.34, which is floor_area
- Bath_room is similar at 0.32

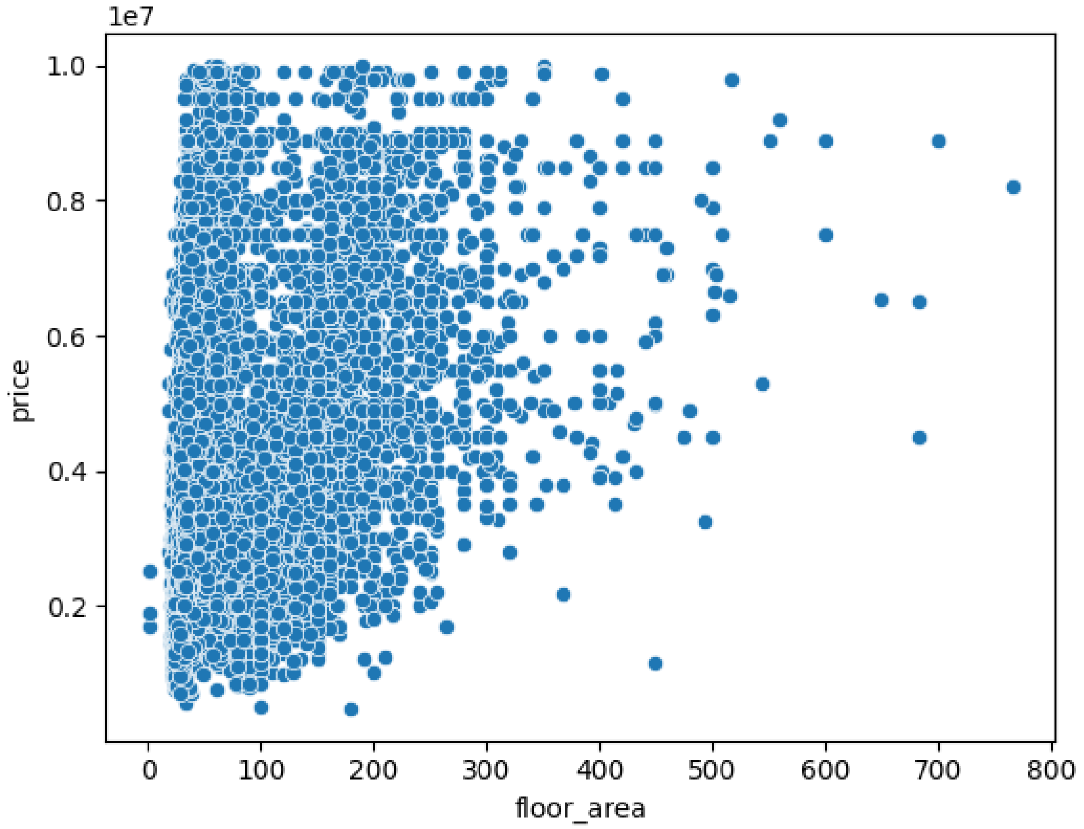


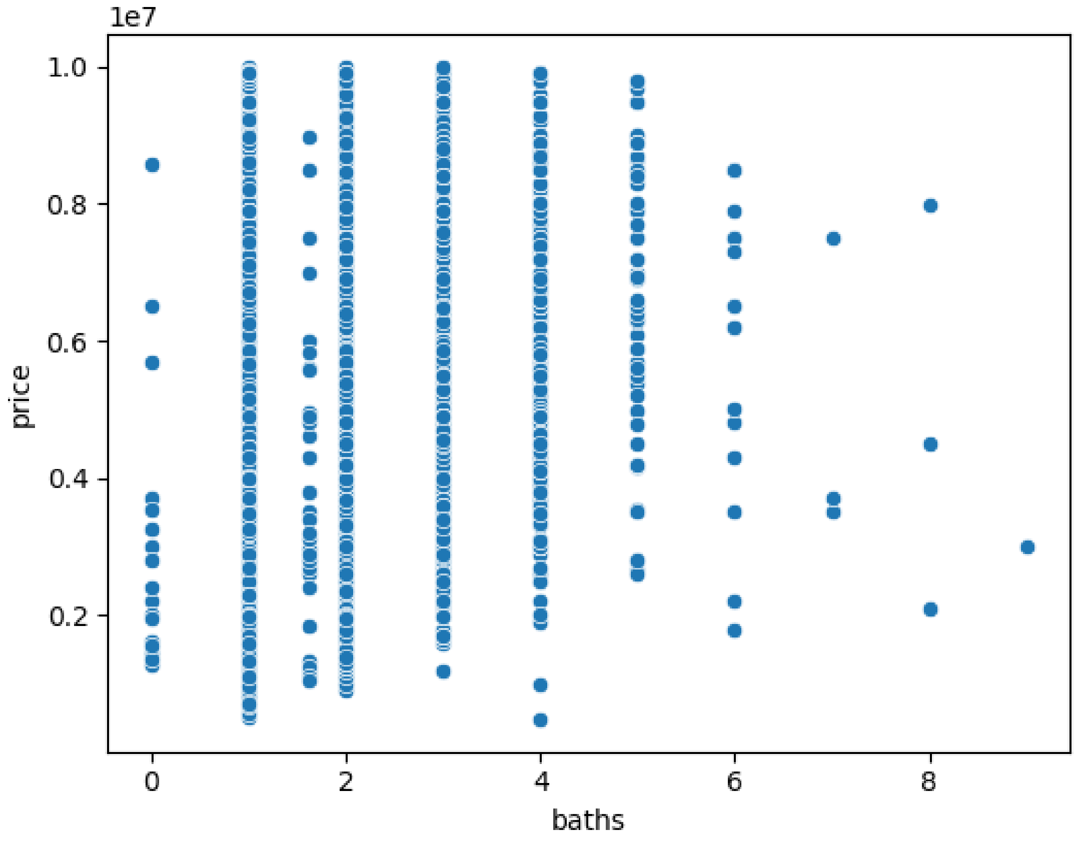


Data Exploration and Analysis (cont.)

Finding:

- Can't see the clear pattern here.
- There is no clear linear relationship between these 2 features and price

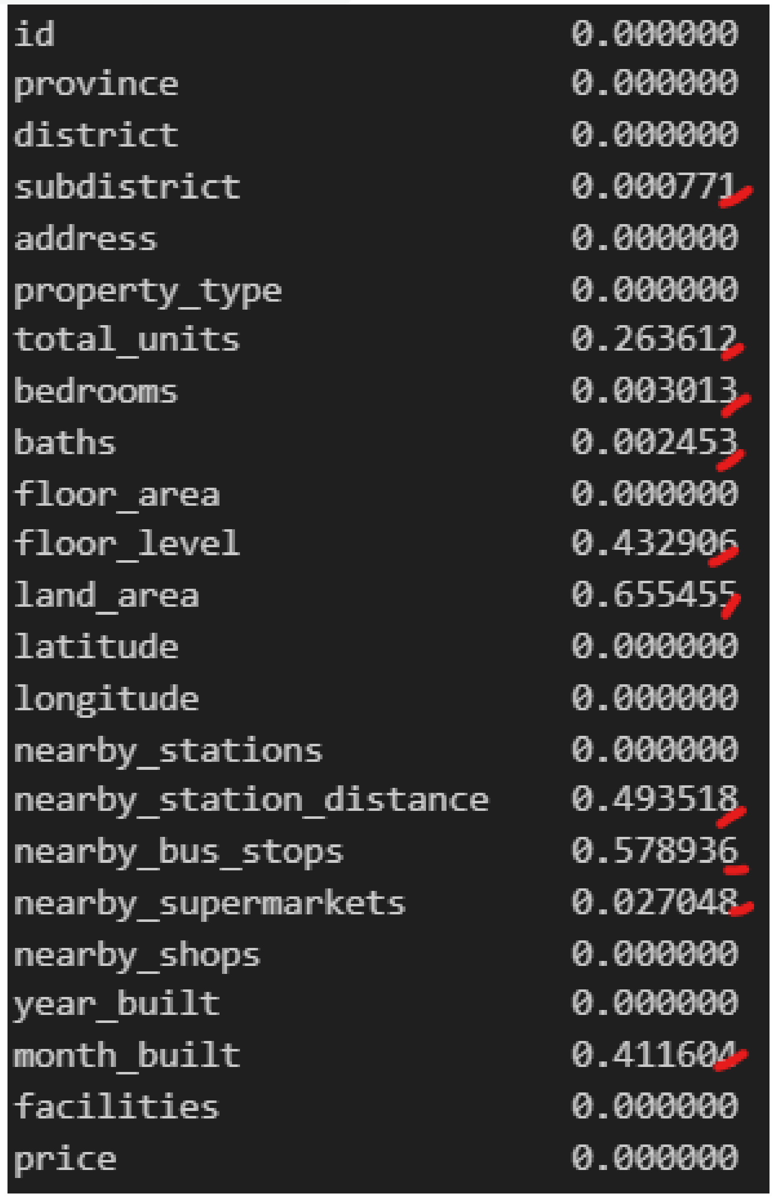




Data Exploration and Analysis (cont.)

Missing Data

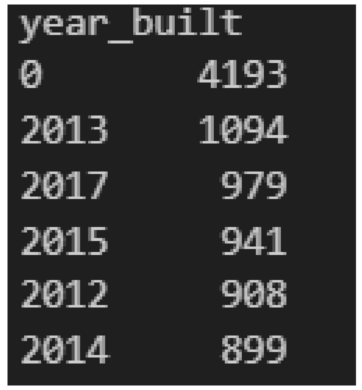
- 10 out of 22 features (columns) are missing some data (is null).
- 5 of them missing over 40% of the data



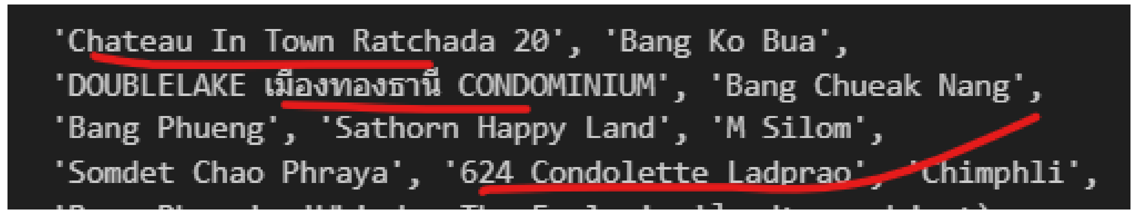
```
id                         0.000000
province                   0.000000
district                   0.000000
subdistrict                0.000771
address                    0.000000
property_type              0.000000
total_units                0.263612
bedrooms                   0.003013
baths                      0.002453
floor_area                 0.000000
floor_level                0.432906
land_area                  0.655455
latitude                   0.000000
longitude                  0.000000
nearby_stations            0.000000
nearby_station_distance    0.493518
nearby_bus_stops           0.578936
nearby_supermarkets        0.027048
nearby_shops               0.000000
year_built                 0.000000
month_built                0.411604
facilities                 0.000000
price                      0.000000
```

Data not align with its meaning from Data dict

- Year of building can't be 0 (2000+ years old??)
- Subdistrict name after condo name??



```
year_built
0        4193
2013     1094
2017      979
2015      941
2012      908
2014      899
```



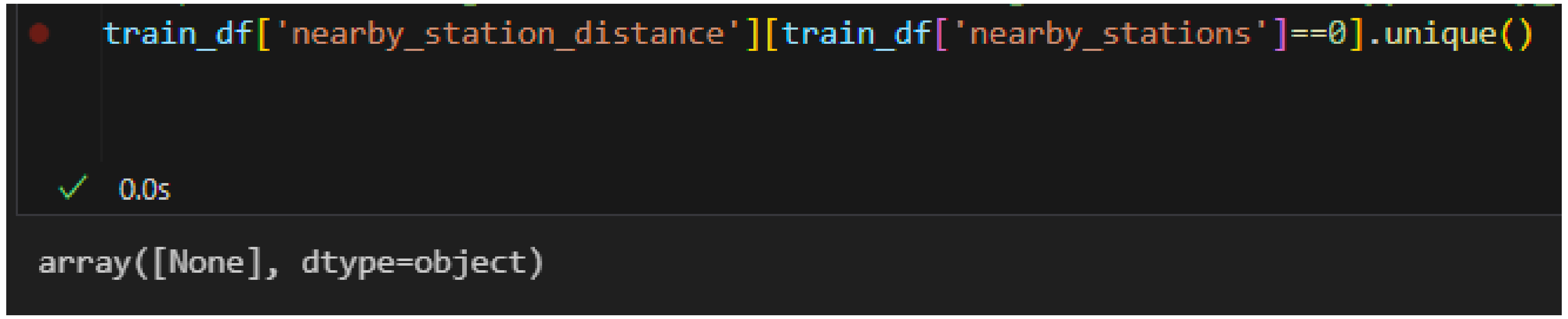
```
 'Chateau In Town Ratchada 20', 'Bang Ko Bua',
 'DOUBLELAKE เมืองทองธานี CONDOMINIUM', 'Bang Chueak Nang',
 'Bang Phueng', 'Sathorn Happy Land', 'M Silom',
 'Somdet Chao Phraya', '624 Condolette Ladprao', 'Chimphli',
```

Data Exploration and Analysis (cont.)

Filling in the missing data

- not at random
  - nearby_station_distance - can replace with 0
  - land_area - consider to rule this out

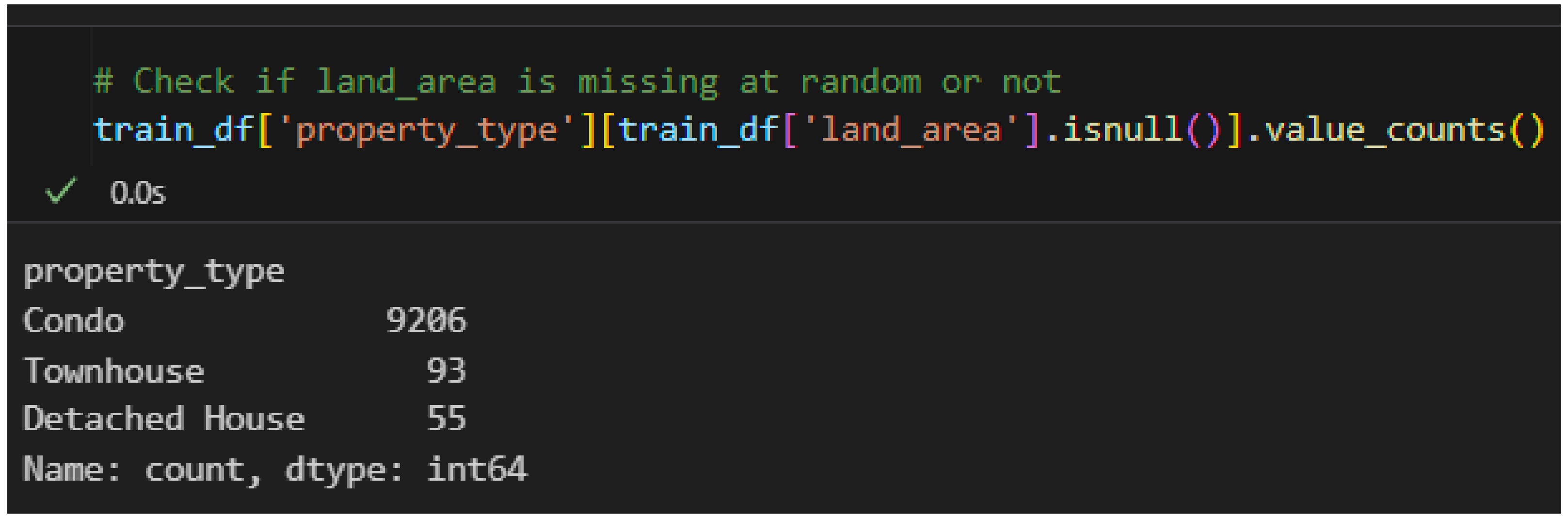
- missing at random
  - the rest - replace outlier with mean and apply multiple imputation method, aiming to maintain the mean and distribution to original as possible



```
train_df['nearby_station_distance'][train_df['nearby_stations']==0].unique()
```

✓ 0.0s

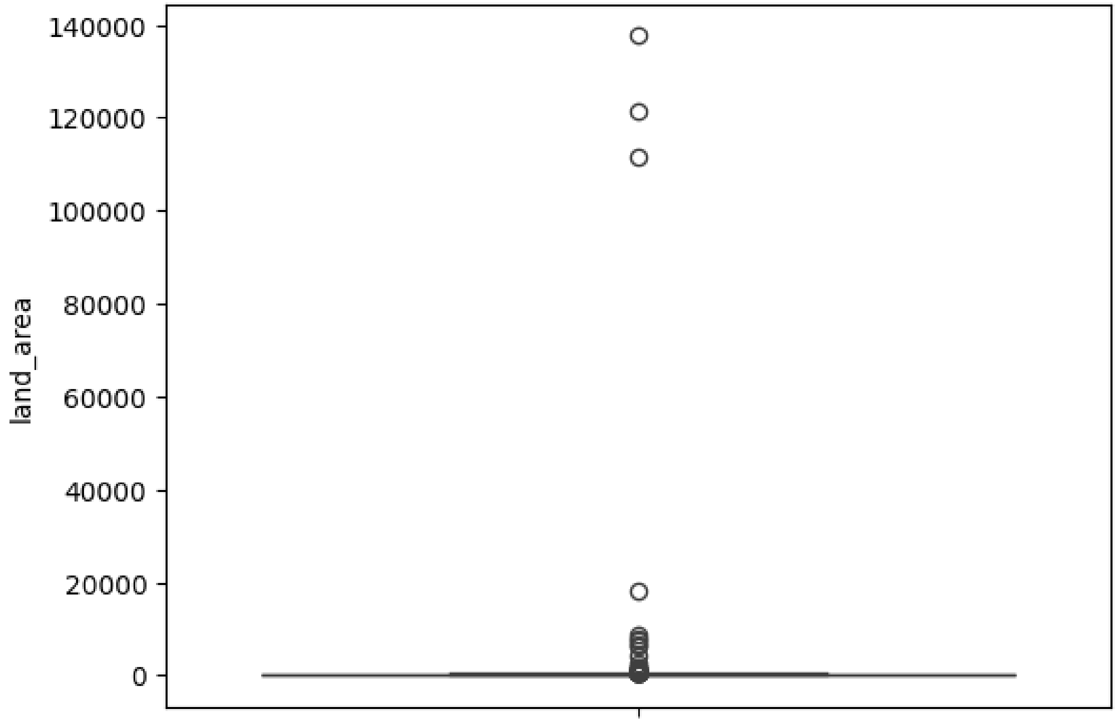
```
array([None], dtype=object)
```

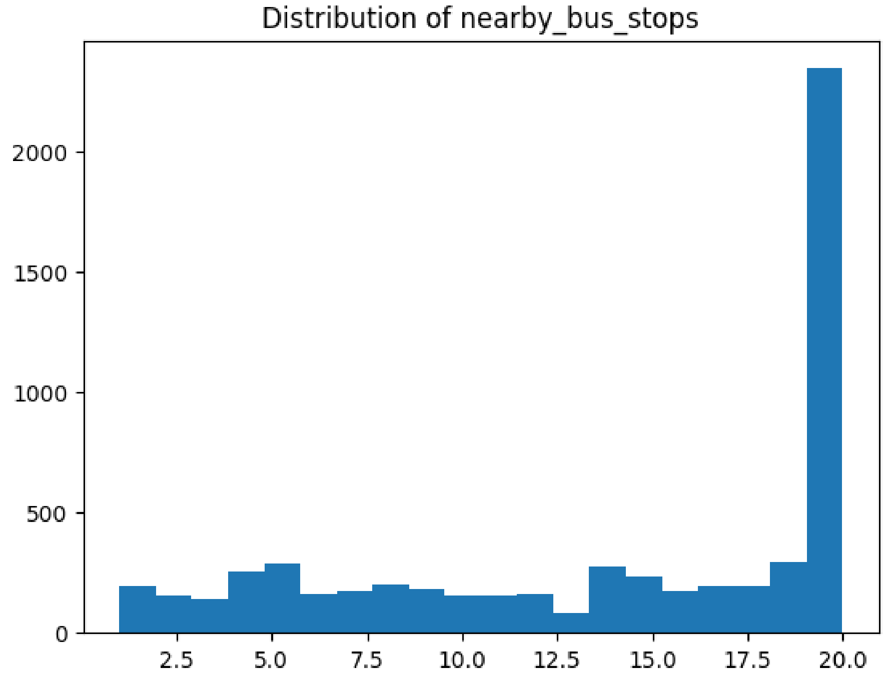


```
# Check if land_area is missing at random or not
train_df['property_type'][train_df['land_area'].isnull()].value_counts()
```

✓ 0.0s

```
property_type
Condo              9206
Townhouse            93
Detached House       55
Name: count, dtype: int64
```

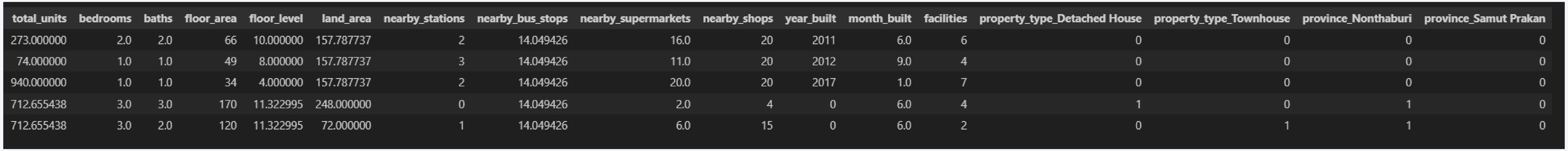




Preprocessing

Preparing data for model training

- 17 features is selected to be used for training
- One-hot encode property type and provinces
- dropped district and subdistrict due to their complexity when hot-coded
- dropped the rest because they barely have correlation with price
- Standardize the data
- split train-test using 20%-> test



total_units	bedrooms	baths	floor_area	floor_level	land_area	nearby_stations	nearby_bus_stops	nearby_supermarkets	nearby_shops	year_built	month_built	facilities	property_type_Detached House	property_type_Townhouse	province_Nonthaburi	province_Samut Prakan
273.000000	2.0	2.0	66	10.000000	157.787737	2	14.049426	16.0	20	2011	6.0	6	0	0	0	0
74.000000	1.0	1.0	49	8.000000	157.787737	3	14.049426	11.0	20	2012	9.0	4	0	0	0	0
940.000000	1.0	1.0	34	4.000000	157.787737	2	14.049426	20.0	20	2017	1.0	7	0	0	0	0
712.655438	3.0	3.0	170	11.322995	248.000000	0	14.049426	2.0	4	0	6.0	4	1	0	1	0
712.655438	3.0	2.0	120	11.322995	72.000000	1	14.049426	6.0	15	0	6.0	2	0	1	1	0

Making Model

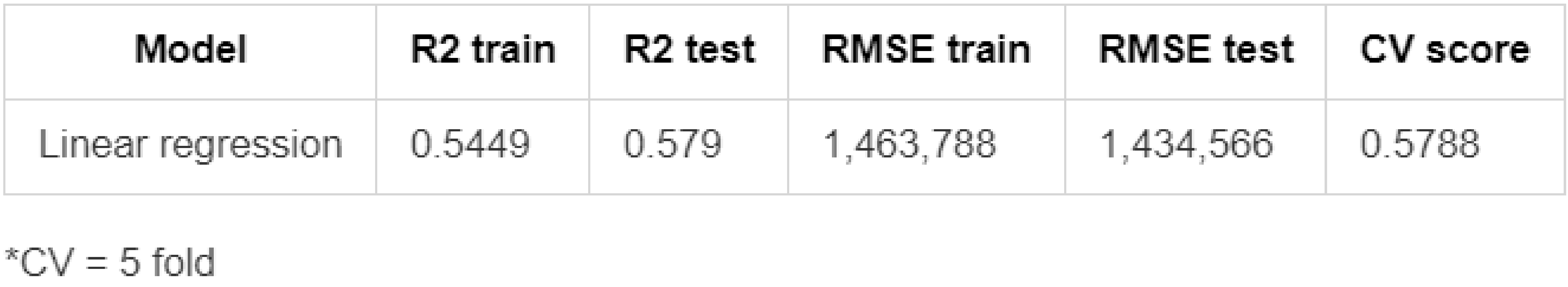
Strategy

- start with linear regression, using prepared train data
- evaluate the result. Increase or decrease feature depends on the result
- incase still high bias, use feature engineering such one-hot encode or polynomial
- test with other model -> Ridge, Lasso, Elastic net
- select the best performance model

Actual

- total 3 tries with 9 model created

Model Evaluation



Model	R2 train	R2 test	RMSE train	RMSE test	CV score
Linear regression	0.5449	0.579	1,463,788	1,434,566	0.5788

*CV = 5 fold

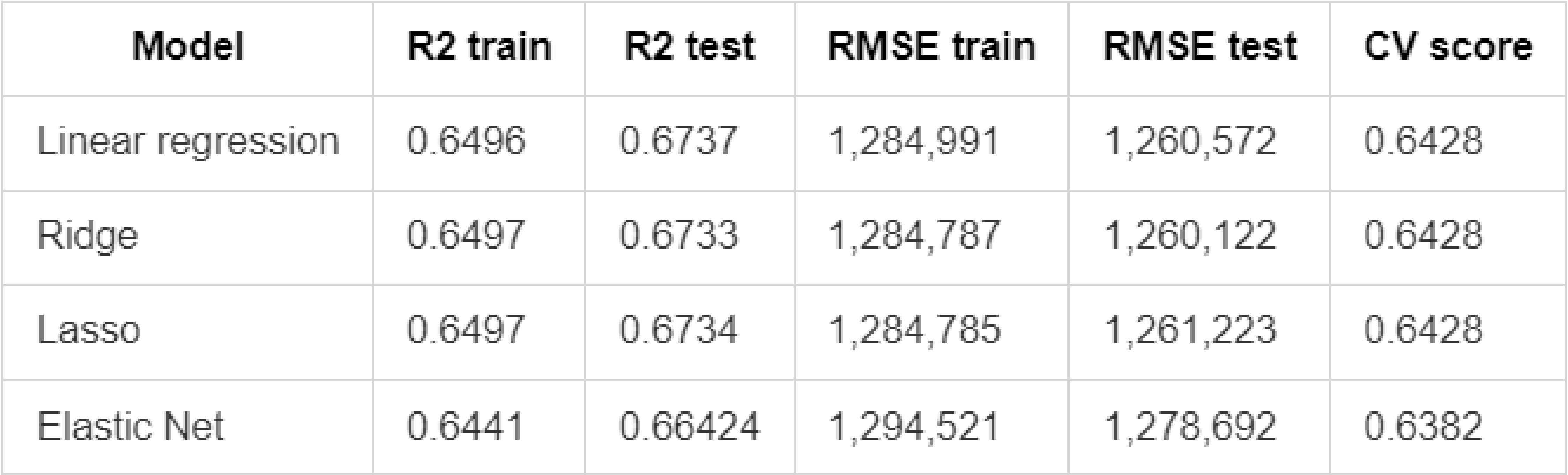
First trial summary

- using linear regression
- no sign of overfitting
- but score is not very good
- consider to add more features to model

Prepare data for next model

- One-hot encode district column
- this added 57 more features to train data

Model Evaluation (cont.)



Model	R2 train	R2 test	RMSE train	RMSE test	CV score
Linear regression	0.6496	0.6737	1,284,991	1,260,572	0.6428
Ridge	0.6497	0.6733	1,284,787	1,260,122	0.6428
Lasso	0.6497	0.6734	1,284,785	1,261,223	0.6428
Elastic Net	0.6441	0.66424	1,294,521	1,278,692	0.6382

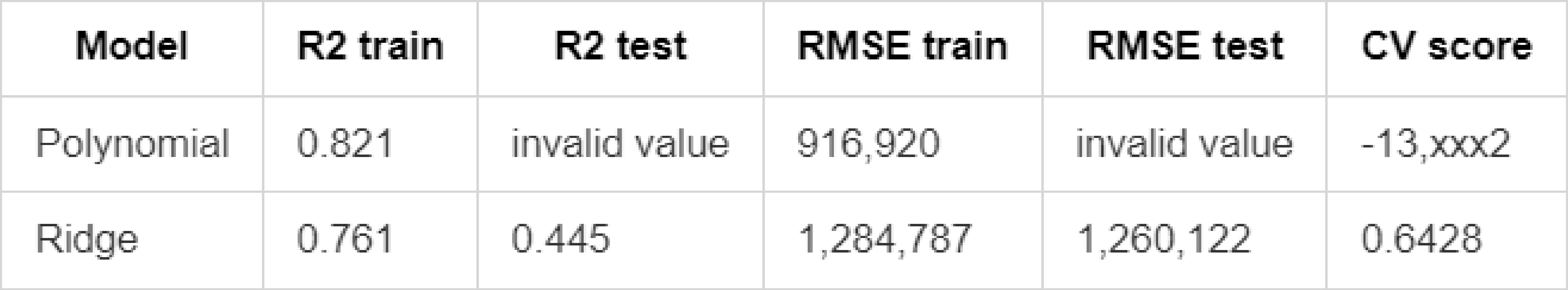
Second trial summary

- no sign of overfitting
- R2 and RMSE improve
- may be there are rooms to improve the features further

Prepare data for next model

- Apply one of feature engineering method, polynomial to add more complexity, aiming to reduce b

Model Evaluation (cont.)



Model	R2 train	R2 test	RMSE train	RMSE test	CV score
Polynomial	0.821	invalid value	916,920	invalid value	-13,xxx2
Ridge	0.761	0.445	1,284,787	1,260,122	0.6428

Third trial summary

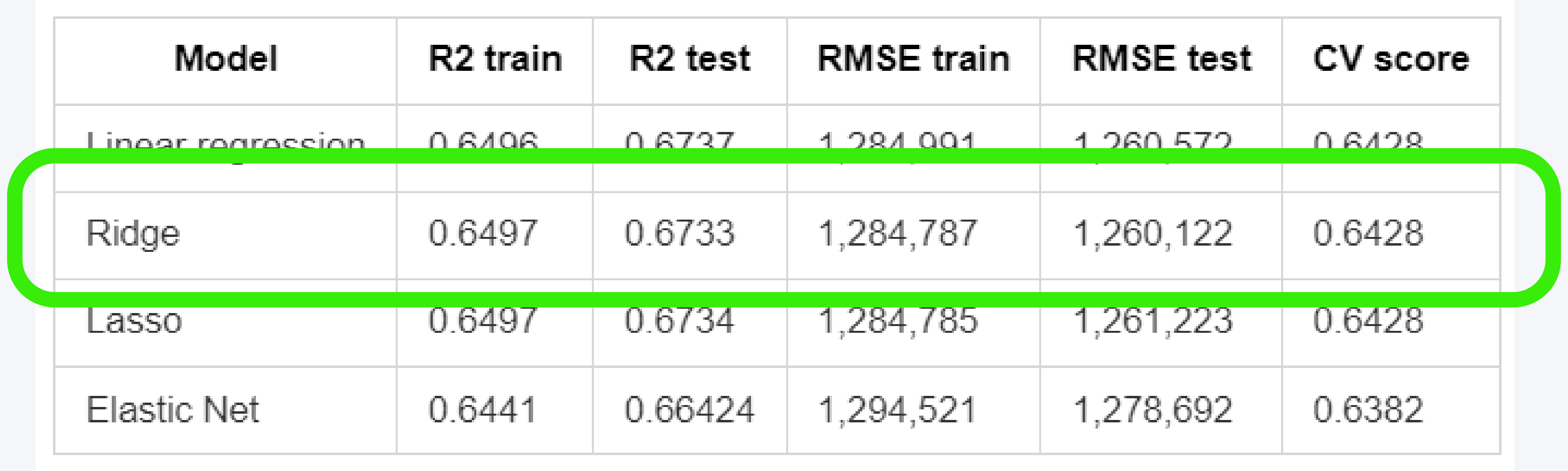
- Sign show strongly overfitting
- Regularization method such Ridge, Lasso, and Elastic net still couldn’t help overfitting
- Fall back to second trial

Hit PC’s limitation

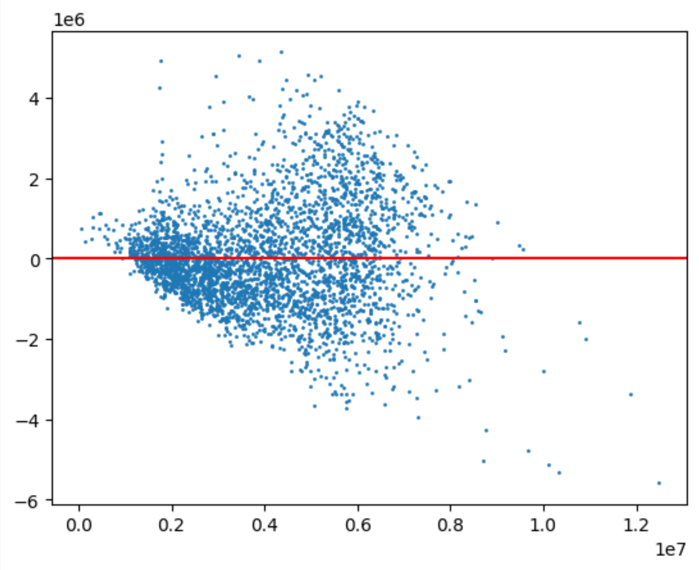
- Lasso and Elastic net can’t be performed due to PC limitation

Model selection outcome

Ridge model



Model	R2 train	R2 test	RMSE train	RMSE test	CV score
Linear regression	0.6496	0.6737	1,284,991	1,260,572	0.6428
Ridge	0.6497	0.6733	1,284,787	1,260,122	0.6428
Lasso	0.6497	0.6734	1,284,785	1,261,223	0.6428
Elastic Net	0.6441	0.66424	1,294,521	1,278,692	0.6382

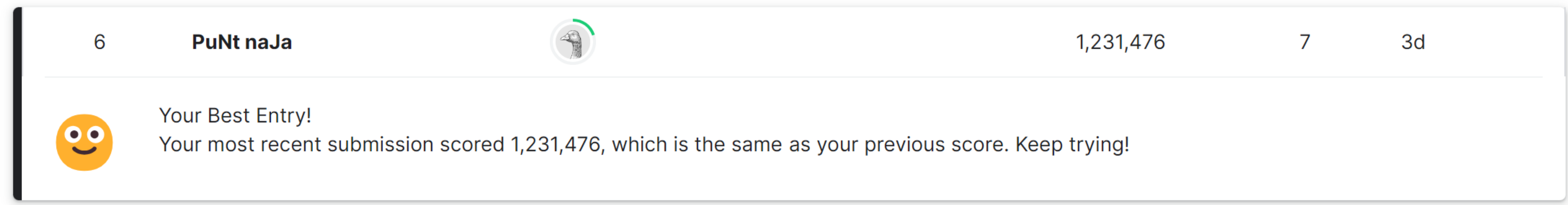


Why?

From the metric evaluation

Model selection is "Ridge" Model

- Not overfit
- Lowest RMSE
- Highest R2 score
- baseline not much worse than the best model



6 PuNt naJa 1,231,476 7 3d

Your Best Entry!
Your most recent submission scored 1,231,476, which is the same as your previous score. Keep trying!

Implication

Bank now has mode to predict the market value of applicant’s target property by input the following:

- province of property
- type of property
- number of unit/houses in the condo/village
- number of bedrooms and bathrooms
- floor area
- floor level
- land area
- nearby station
- nearby bus stop
- nearby supermarket
- nearby shop
- when it was built (month/year)
- number of facility (gym, swimming pool, etc.)

*** remarks

- model is able to explain about 65% of data
- it may has price error up to 1.2 MTHB

Implication (cont.)

Recommendation on high value property. it should contain the following feature

- more bathroom - each additional bathroom can increase value by 700k
- located in Watthana district - if it is located in this district, it is likely to have value 480k more
- located on the higher floor - each higher floor, it is likely to increase the value by 350k

Feature to avoid, because it may lower the property value.

- **not located in Bangkok - it is likely that the value is lower by 400k**
- **high total unit - each of additional total unit is likely to lower value by 200k**
- **townhouse - this type of property is likely to lower value by 150k**

Thank You