ALLEGHENY COUNTY HOME PRICES

RAHUL GUPTA: 7/13/2021

INTRODUCTION: MY BACKGROUND

- Education:
  - Undergraduate degree in Engineering, IIT-BHU, India.
  - Masters in Financial Engineering, Columbia University, New York City.
  - CFA charter holder
  - FRM charter holder

- Work experience:
  - 6+ years in buy-side: quantitative investing across equities, fixed income and FX
  - 4+ years in sell-side: global markets trading, investment technology

- Machine learning education and experience:
  - Introductory course during MFE
  - Self-learning via MOOC
  - Applications at my current role

DATA AND PROBLEM DESCRIPTION

Dataset:

- Allegheny county home price
- This database is used by the Office of Property Assessments (OPA) to administer the property assessment system in Allegheny County, PA used for taxation purposes.
- The electronic format came into use beginning with the 2002 county reassessment with Sabre Systems
- Parcel ID numbers are the primary key linking assessment data with Allegheny County's parcel GIS files.
- Assessed Values are often not the same as the Market Value of a property, and the two may NOT be used interchangeably.
- Deeds have been recorded in Allegheny County since 1788. Early sales may not be seen in the electronic system, and may default to a 1950 sale date
- Sales recorded in deed records and captured in the property sales dataset may not appear in the assessment dataset if these properties have not yet been assigned an assessed value for taxation purposes.

Main problem statement:

  - Are housing prices a martingale?
  - How to create a house price index?
  - Create an investment strategy?

INTRODUCTION: CONTENTS

- Exploratory Data Analysis
  - Column list
  - Filtering out for right type of houses
  - Cleaning up dates
  - Cleaning up sale prices
  - Cleaning up cleaning up previous sale prices

- Analyzing house counts
  - Over time
  - By different characteristics

- Analyzing house prices
  - Over time
  - By different characteristics

- House price index
  - Removing seasonality
  - Comparing breakdowns

- Investment engine
  - Appreciation prediction model
  - Tuning the ML models
  - Running OOS models and compare performance

- Improvement areas
- Time for questions

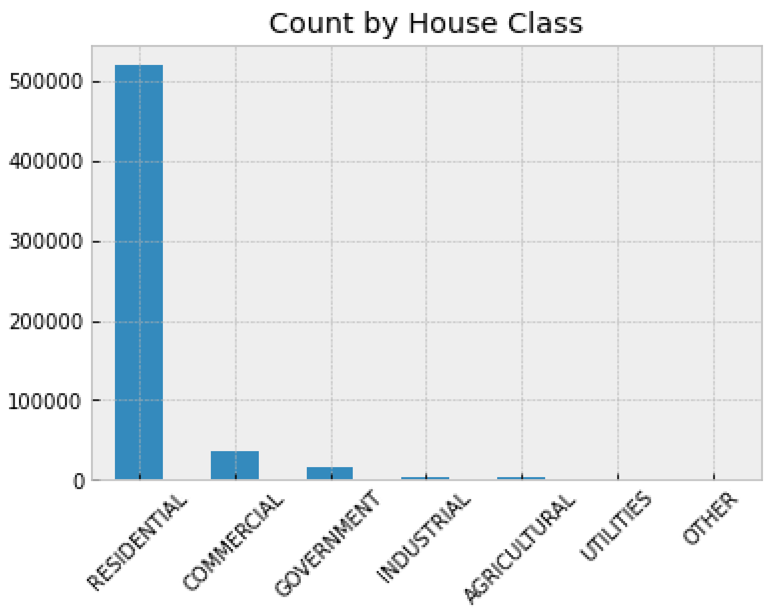
COLUMN LIST

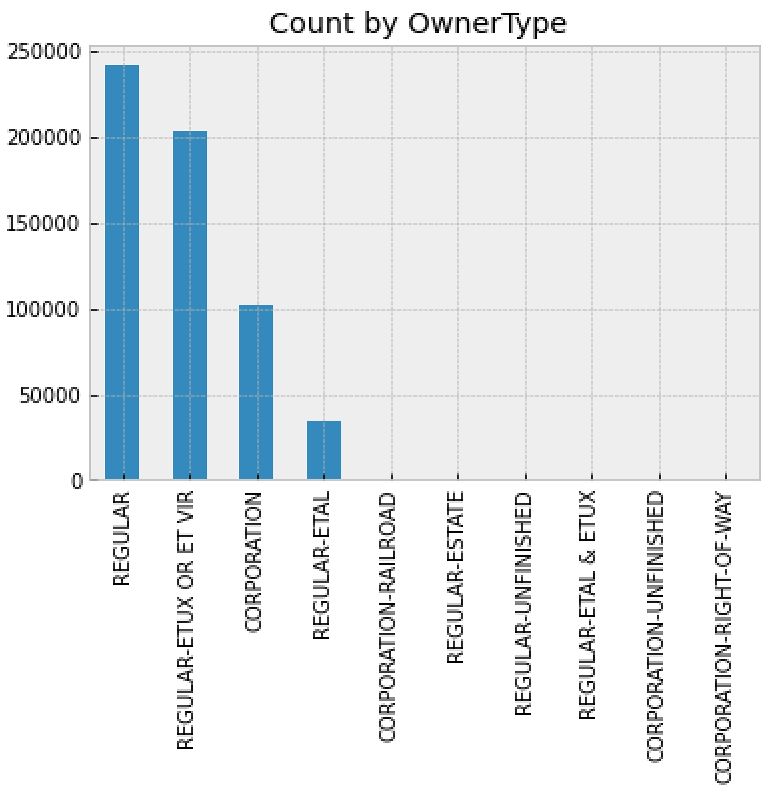
- Total 86 columns
- Total 581,123 rows
- Each data row is a unique house specified by PARID
- Most of the columns have good coverage
- Highlighted columns are focused on in the study
- Can technically use some of the columns that were ignored.

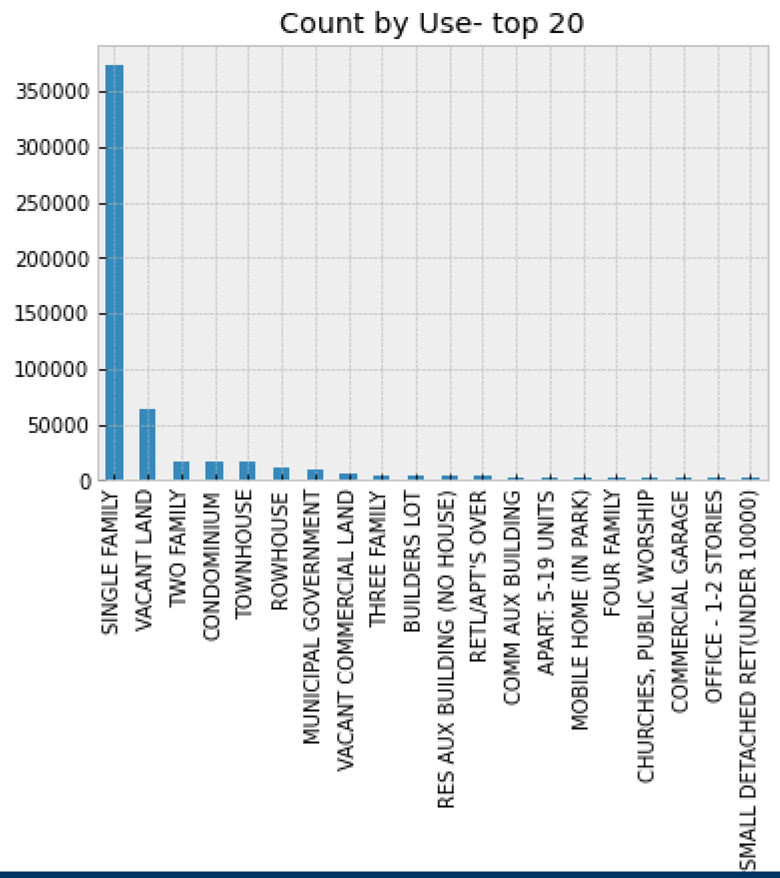
#	Column	Non-Null
0	PARID	581123
1	PROPERTYHOUSENUM	581097
2	PROPERTYFRACTION	581123
3	PROPERTYADDRESS	581123
4	PROPERTYCITY	581101
5	PROPERTYSTATE	581123
6	PROPERTYUNIT	581123
7	PROPERTYZIP	581102
8	MUNICODE	581123
9	MUNIDESC	581123
10	SCHOOLCODE	581123
11	SCHOOLDESC	581123
12	LEGAL1	581101
13	LEGAL2	517225
14	LEGAL3	337815
15	NEIGHCODE	581123
16	NEIGHDESC	580206
17	TAXCODE	581123
18	TAXDESC	581123
19	TAXSUBCODE	1718
20	TAXSUBCODE_DESC	1718
21	OWNERCODE	581123
22	OWNERDESC	581123
23	CLASS	581123
24	CLASSDESC	581123
25	USECODE	581123
26	USEDESC	581122
27	LOTAREA	581123
28	HOMESTEADFLAG	304917
29	FARMSTEADFLAG	6
30	CLEANGREEN	1813
31	ABATEMENTFLAG	594
32	RECORDDATE	351568
33	SALEDATE	574279
34	SALEPRICE	571639
35	SALECODE	573321
36	SALEDESC	571846
37	DEEDBOOK	531332
38	DEEDPAGE	531114
39	PREVSALEDATE	382339
40	PREVSALEPRICE	380619
41	PREVSALEDATE2	206550
42	PREVSALEPRICE2	205957
43	CHANGENOTICEADDRESS1	581123
44	CHANGENOTICEADDRESS2	581123
45	CHANGENOTICEADDRESS3	581123
46	CHANGENOTICEADDRESS4	568925
47	COUNTYBUILDING	581123
48	COUNTYLAND	581123
49	COUNTYTOTAL	581123
50	COUNTYEXEMPTBLDG	581123
51	LOCALBUILDING	581123
52	LOCALLAND	581123
53	LOCALTOTAL	581123
54	FAIRMARKETBUILDING	581123
55	FAIRMARKETLAND	581123
56	FAIRMARKETTOTAL	581123
57	STYLE	446084
58	STYLEDESC	446084
59	STORIES	446075
60	YEARBLT	446091
61	EXTERIORFINISH	446078
62	EXTFINISH_DESC	446078
63	ROOF	445356
64	ROOFDESC	445356
65	BASEMENT	445985
66	BASEMENTDESC	445985
67	GRADE	446080
68	GRADEDESC	446080
69	CONDITION	446014
70	CONDITIONDESC	446014
71	CDU	446014
72	CDUDESC	446014
73	TOTALROOMS	446009
74	BEDROOMS	446041
75	FULLBATHS	445961
76	HALFBATHS	442191
77	HEATINGCOOLING	445904
78	HEATINGCOOLINGDESC	445904
79	FIREPLACES	411111
80	BSMTGARAGE	425627
81	FINISHEDLIVINGAREA	446091
82	CARDNUMBER	446091
83	ALT_ID	26200
84	TAXYEAR	581123
85	ASOFDATE	581123

FILTERING FOR RELEVANT PROPERTIES

- The main data contains records for commercial, govt properties etc.
- It also contains information – vacant lot, parking lot
- Further, some are corporation owned.
- We filter for: Residential properties, with “REGULAR” ownership and used for family housing.
- Row count = 404,191

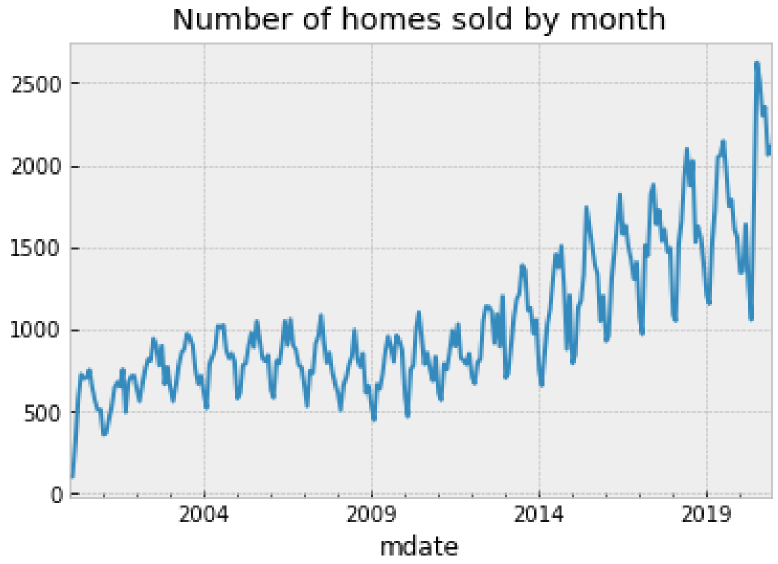


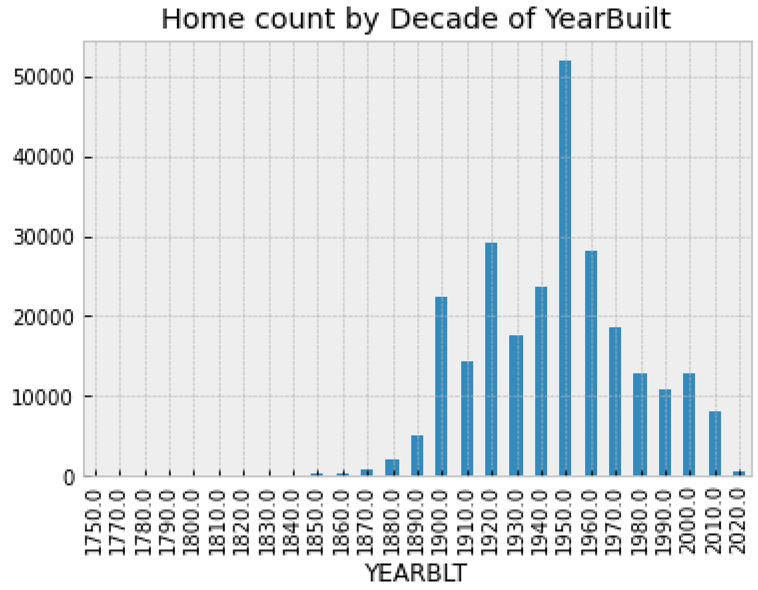




CLEAN UP DATES

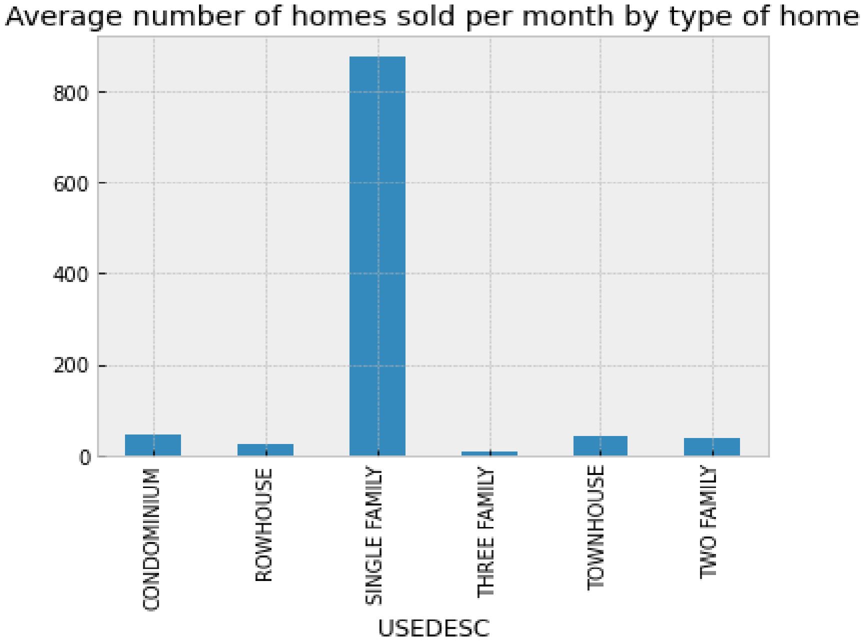
- Main two date columns are SALEDATE and PREVSALEDATE
- Because we will analyze at monthly levels, we convert the dates to month end values
- We ignore data which has either of those variables as NAN
  - Row count = 279,339
- We also filter to choose data only after SALEDATE > 1/1/2000 for clean dataset
  - Row count = 259,588

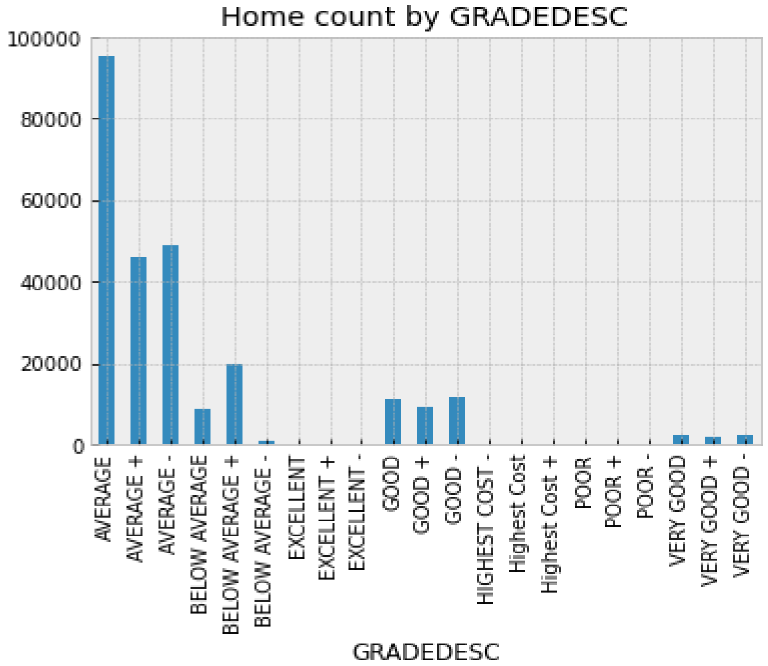


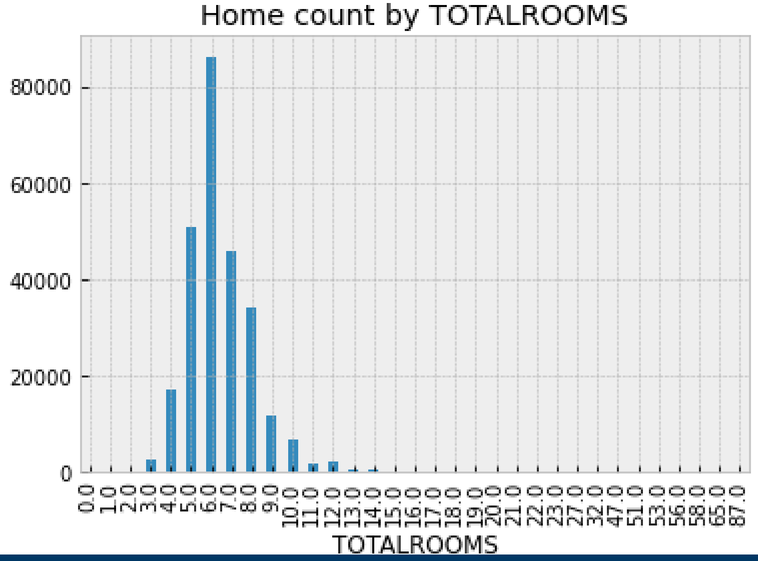


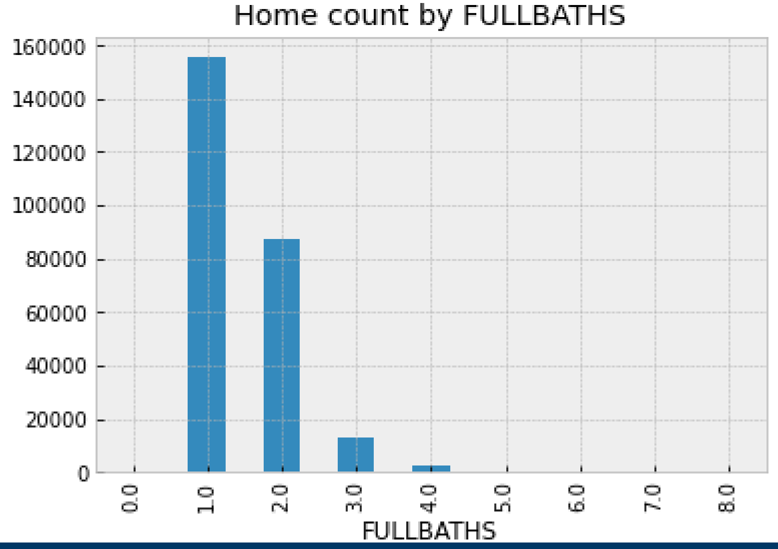
HOUSE COUNTS BY CHARACTERISTICS

- Most of the houses are Single Family with about 800 of those sold per month
- Most of them have between 4-7 rooms in and 1-2 full baths
- Further, they are either grades of average or good quality









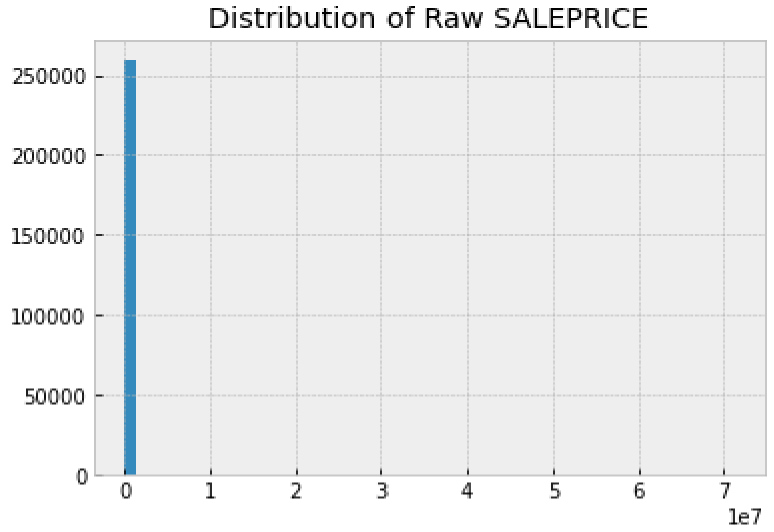
SALE PRICE DATA IS PRETTY BAD

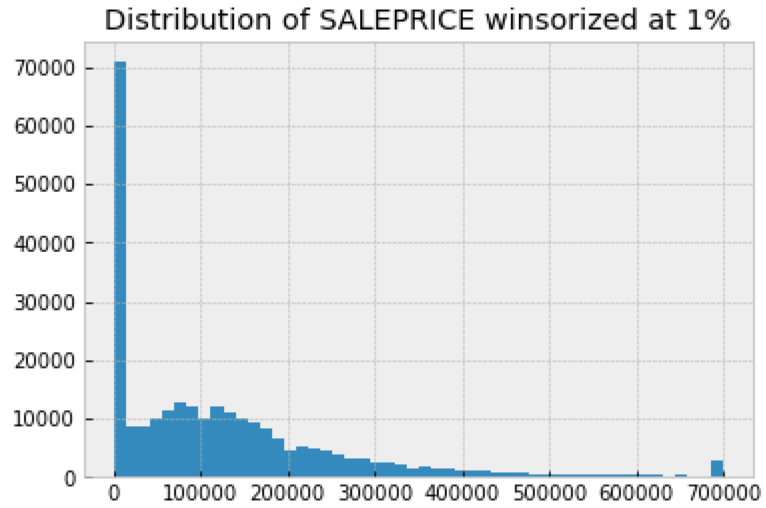
- Tons of houses with SALEPRICE at $1
- Winsorizing at 1% shows almost 70000 houses are left tail outliers
- Few right tails too

```
In [19]: df2.loc[df2.SALEPRICE < 10, ['PROPERTYHOUSENUM','PROPERTYADDRESS','PROPERTYCITY','PROPERTYUNIT
```

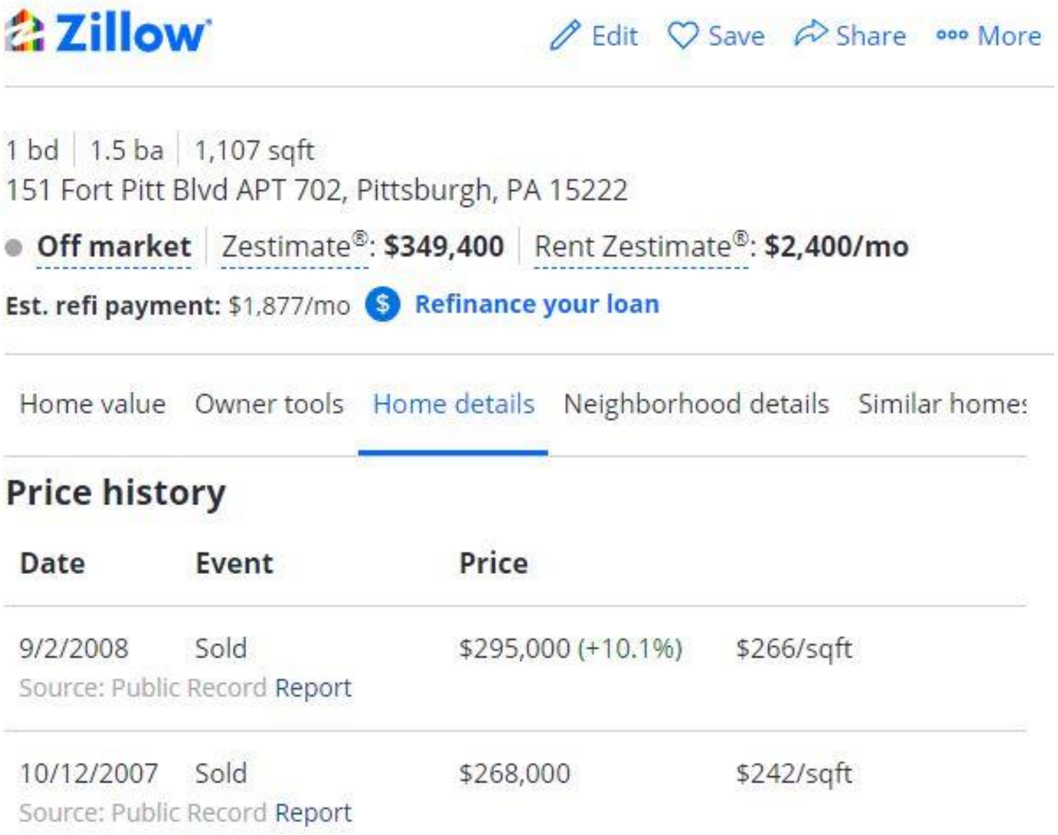
Out[19]:

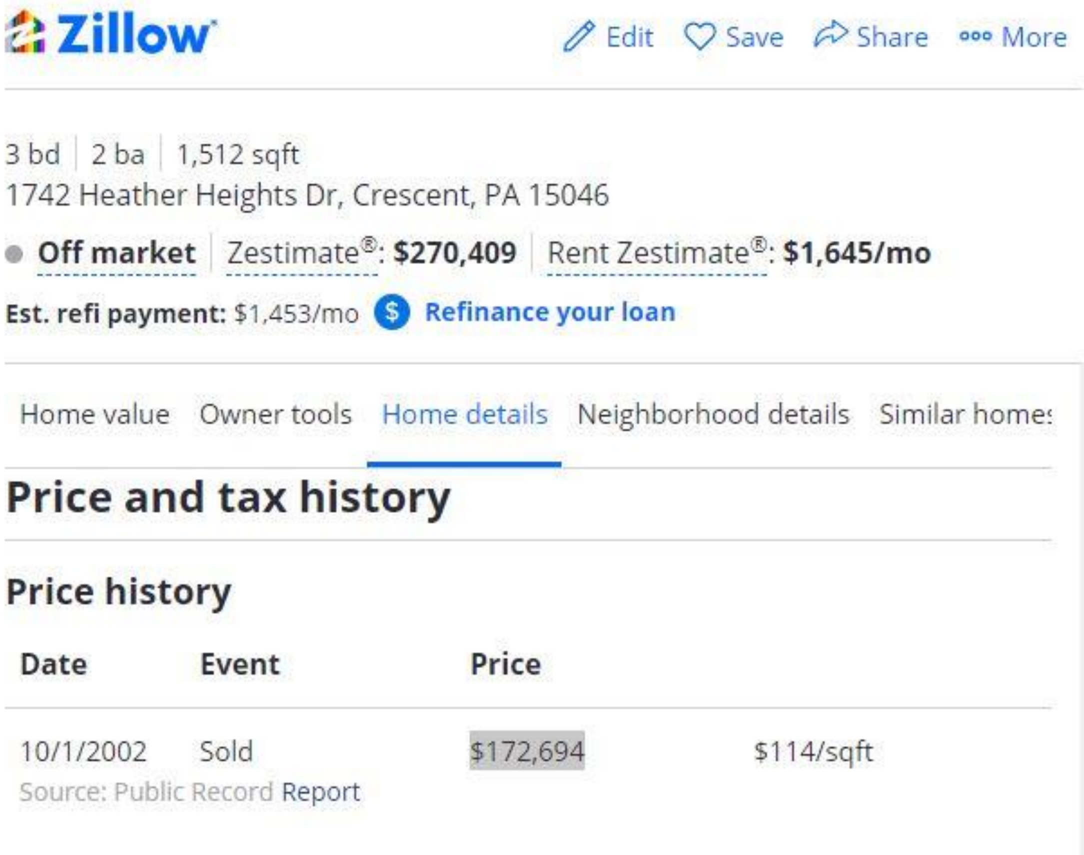
	PROPERTYHOUSENUM	PROPERTYADDRESS	PROPERTYCITY	PROPERTYUNIT	SALEPRICE	SALEDATE	F
30	151.0	FORT PITT BLVD	PITTSBURGH	UNIT 702	1.0	04-27-2020	
194	306.0	4TH AVE	PITTSBURGH	UNIT 302	1.0	02-08-2018	
436	429.0	1ST AVE	PITTSBURGH		1.0	12-31-2001	
611	1.0	MARION ST	PITTSBURGH		0.0	04-17-2014	
657	117.0	VAN BRAAM ST	PITTSBURGH		0.0	04-20-2010	
...	...	...	...	...	...	...	
581060	236.0	CARYL DR	PITTSBURGH		1.0	04-29-2005	
581068	4860.0	ELMWOOD DR	PITTSBURGH		1.0	05-29-2012	
581079	1519.0	KING JOHN DR	PITTSBURGH		1.0	05-14-2004	
581091	1742.0	HEATHER HEIGHTS DR	CRESCENT		1.0	12-10-2007	
581120	7558.0	NOBLESTOWN RD	MC DONALD		1.0	05-09-2020	





PRICE IN DATASET DOES NOT MATCH OTHER PUBLIC SOURCES





CLEANING FOR PRICES

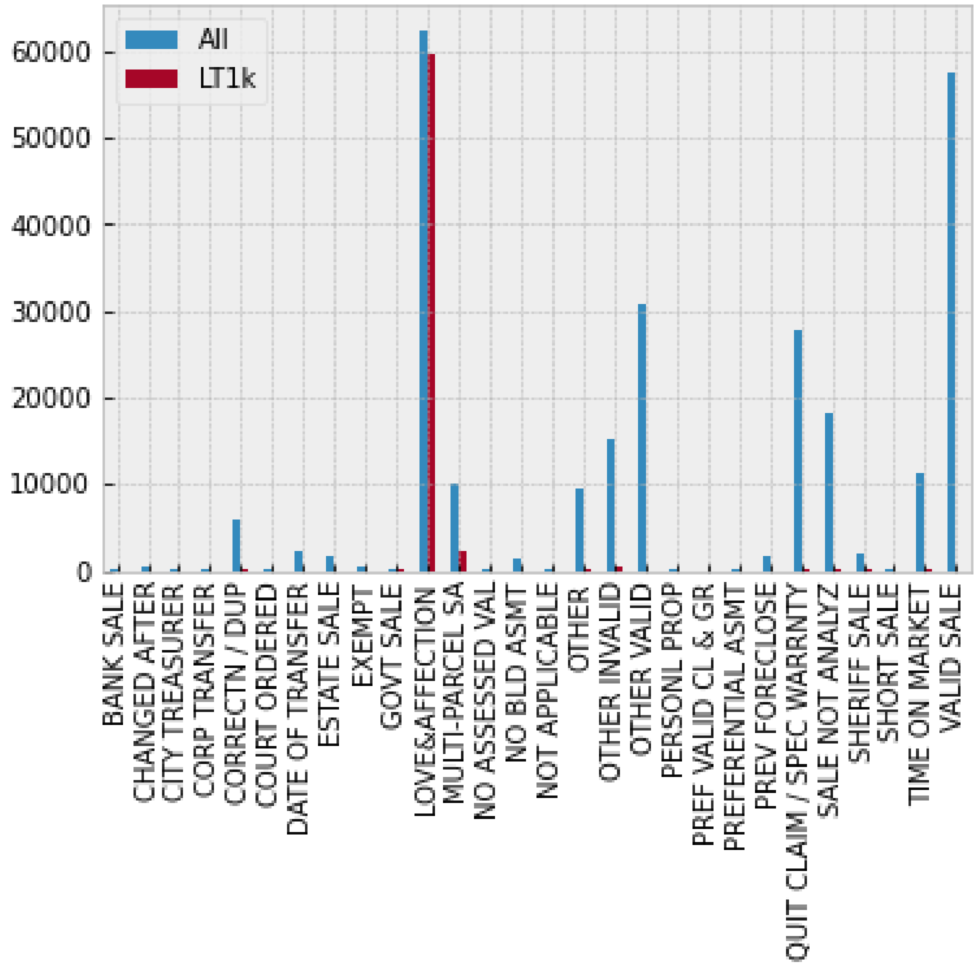
- SALEDESC: “A subjective categorization (description) as to whether or not the sale price was representative of current market value.”

- We look at SALEDESC and find that most of the prices less than $1k driven by LOVE&AFFECTION category.

- We also find that this contains “VALID” sales tagged. We filter for these tags.

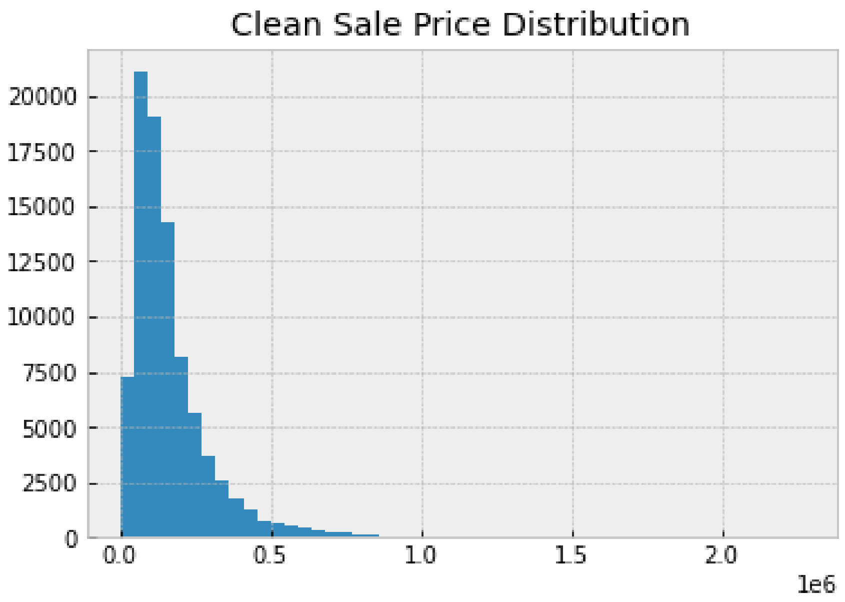
- Row count = 88305

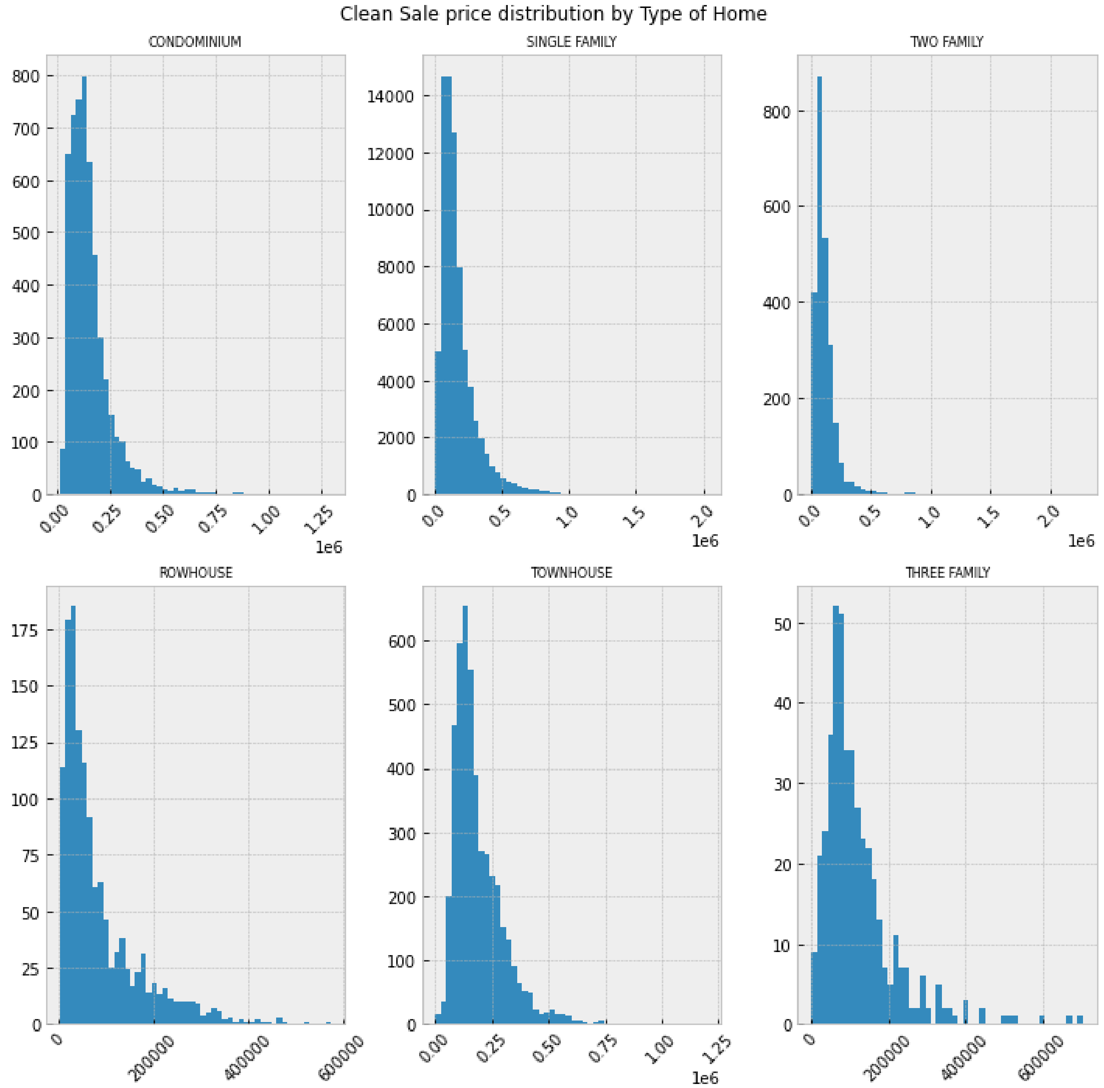
- We also winsorize at +-1% at a monthly level to take care of any remaining outliers at this point



CLEAN SALE PRICE DISTRIBUTION

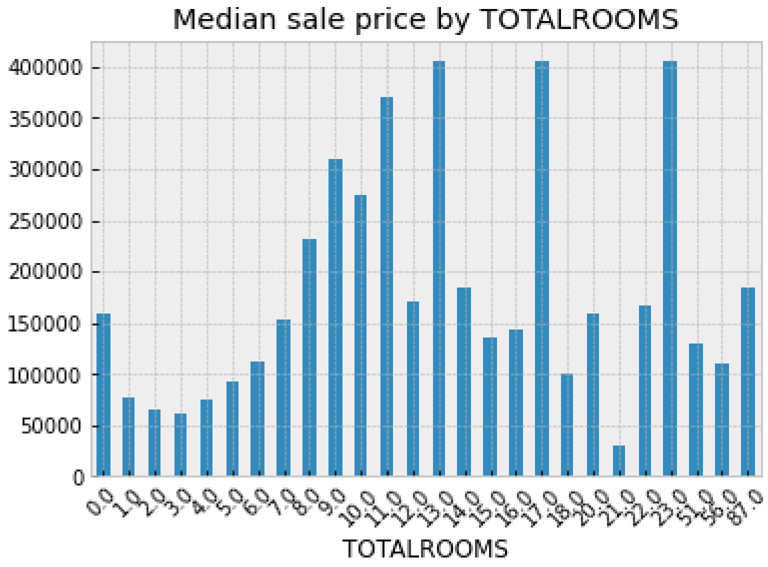
- Cleaned up price distribution looks much better
- Shown at aggregate level and also broken down by type of home

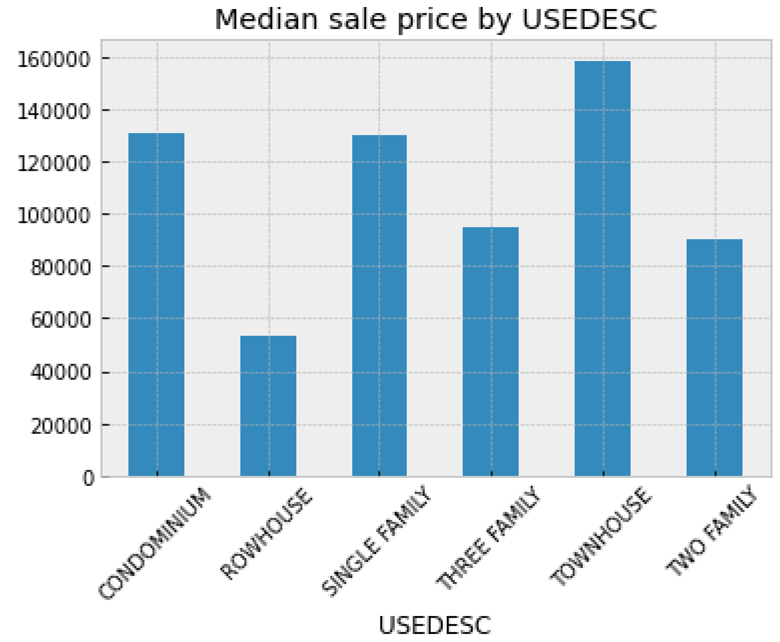


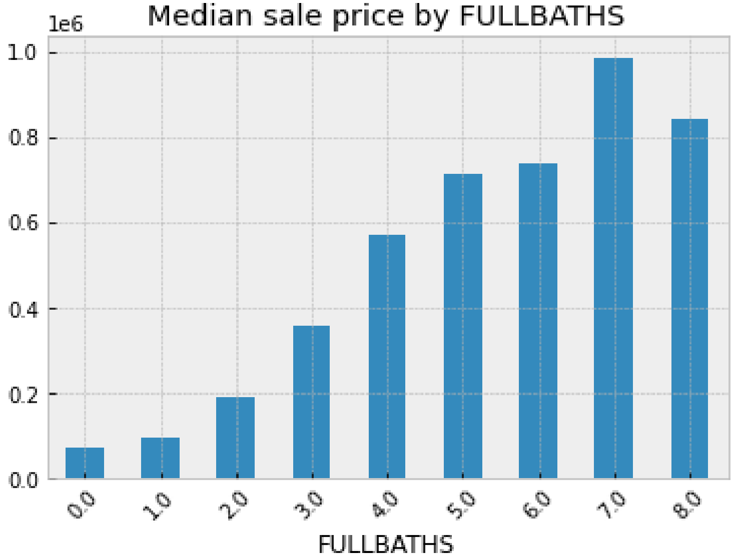


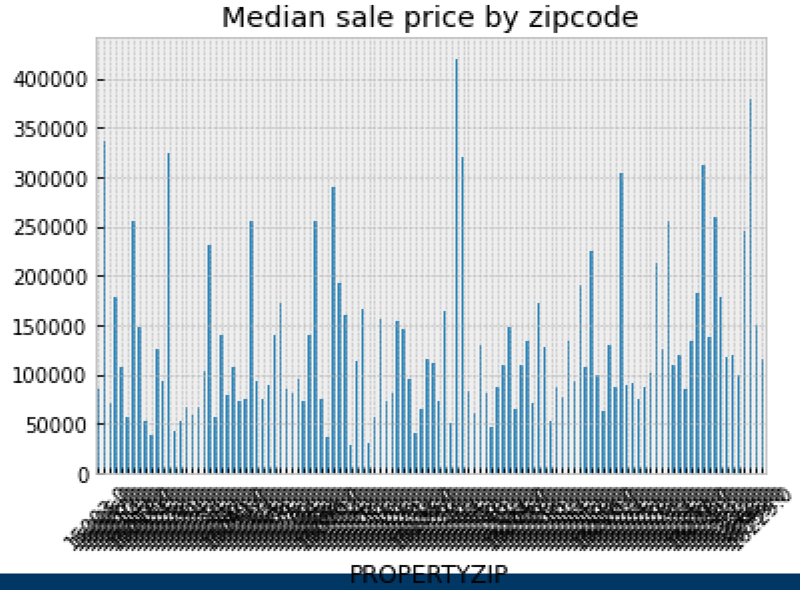
SALE PRICES CROSS SECTION

- Sale price varies by zipcode
- Generally increases by # of rooms and bathrooms
- Townhouses seem to be more expensive (counterintuitive!!)
- Yes, some houses are listed with ZERO rooms and bathrooms ☺!









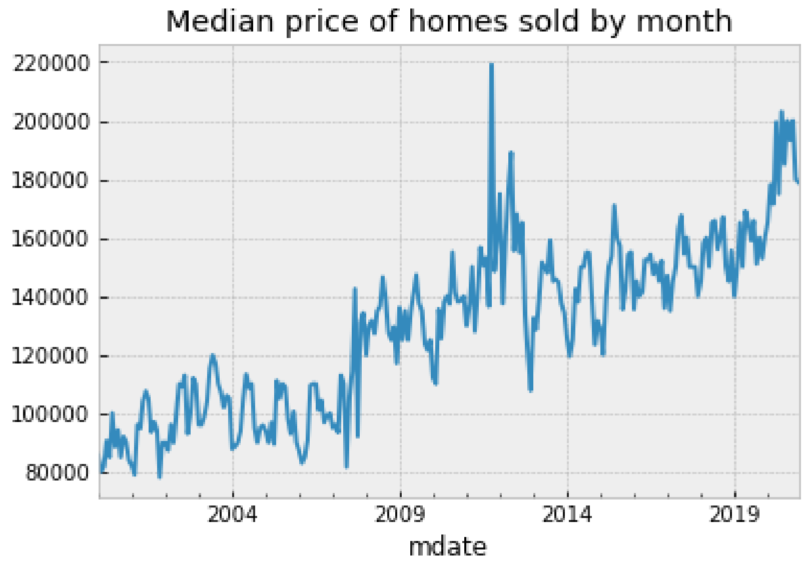
SALE PRICE TIME SERIES

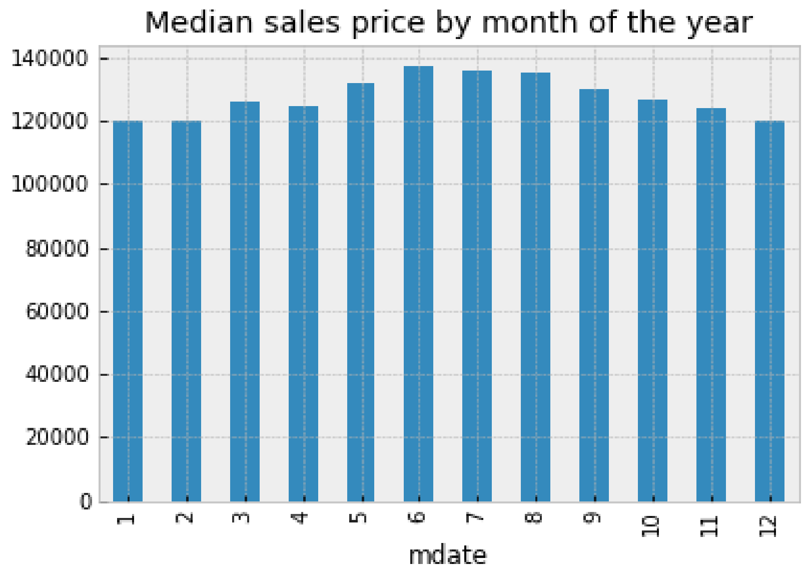
- Median prices have increased over time

- We observe that the prices have seasonality present. They seem to have peaks and troughs at regular intervals

- Upon looking at median price of aggregated months of the year, we find that summer months have higher prices

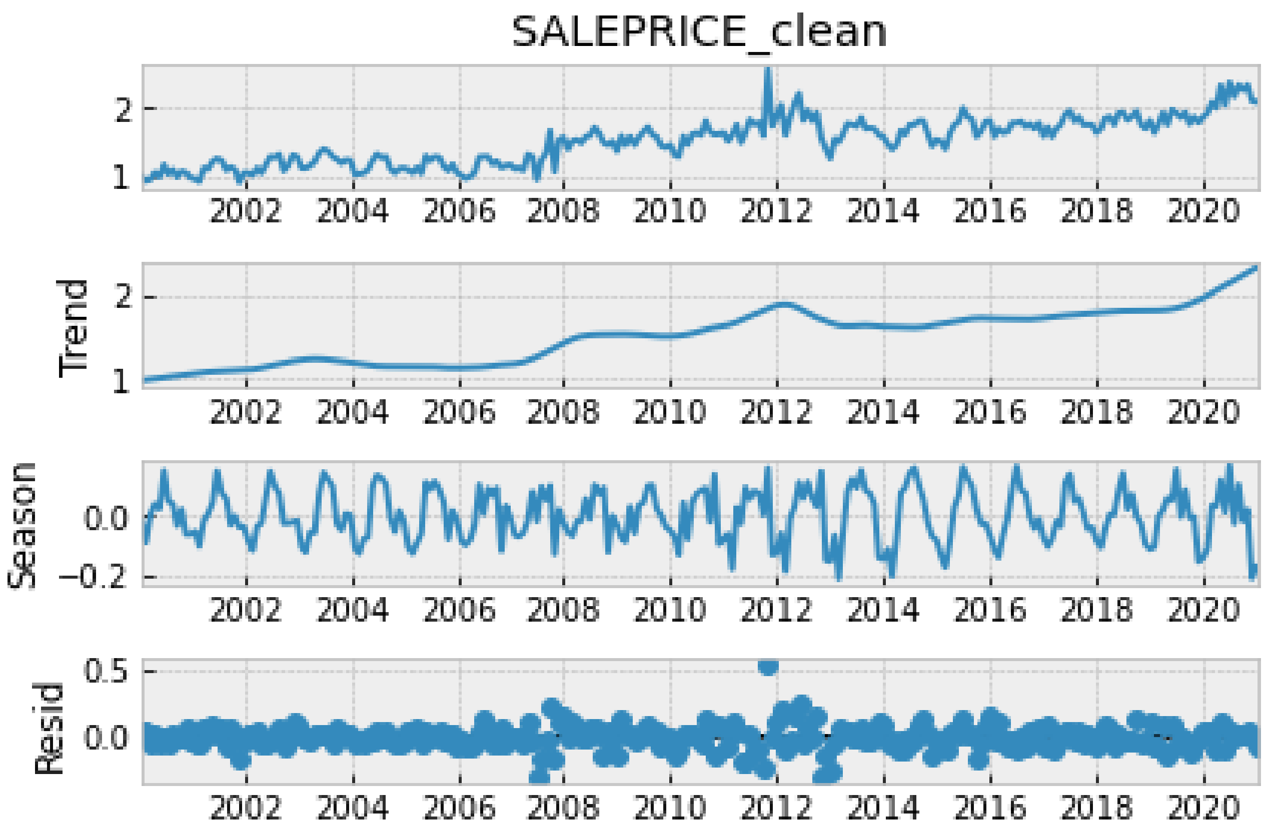
- This is a well known phenomenon and has been observed by academic researchers:
    - Hot and Cold Seasons in the Housing Market, Ngai & Tenreyro (2014)





SEASONALITY ADJUSTMENT OF SALESPRICE

- Need to adjust seasonality for the summer spikes in the prices and
- Use the Zillow* approach suggested: "a LOESS-based seasonal decomposition".
- Implemented using statsmodels.tsa.seasonal.STL
- It basically breaks a time series into
  - Trend: similar to moving average
  - Seasonality: seasonal mean of detrended time series
  - And residual



STL plot for seasonality adjustment of housing prices

*https://www.zillow.com/research/zhvi-methodology-2019-deep-26226/

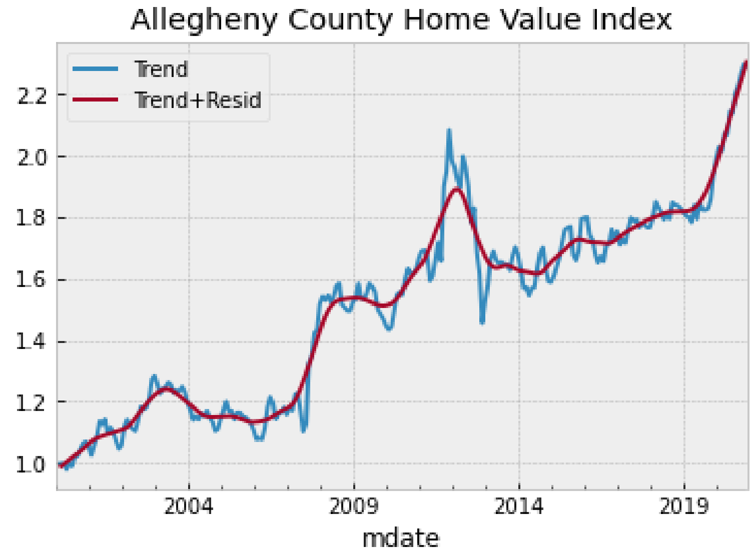
ALLEGHENY COUNTY HOME PRICE INDEX

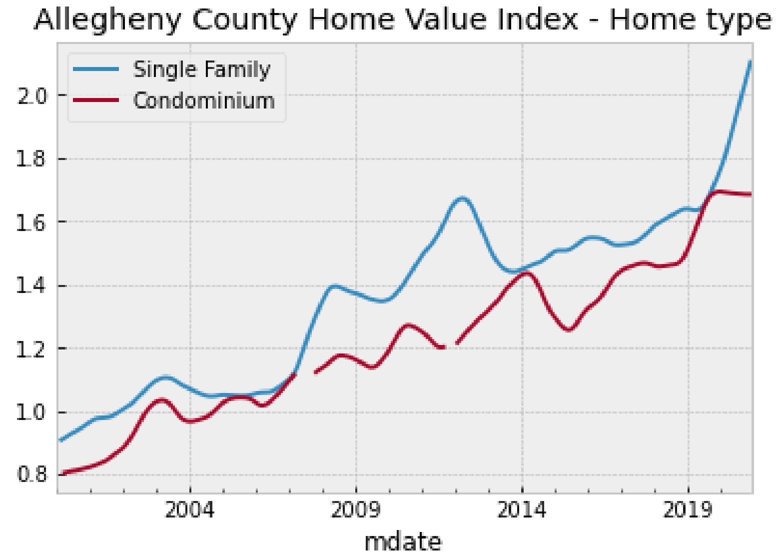
- The home index is basically defined as the seasonally adjusted median price

- Definitely noisy, so have included the pure trend as well

- Note the residuals from this approach can be used to exclude outliers

- Index by sub-groups are also shown:
    - Some groups appreciate differently than others. e.g., suburban houses appreciated more than city houses during COVID-19
    - Houses built before 1950 vs after 1950
    - Less than 5 rooms vs more than 5 rooms





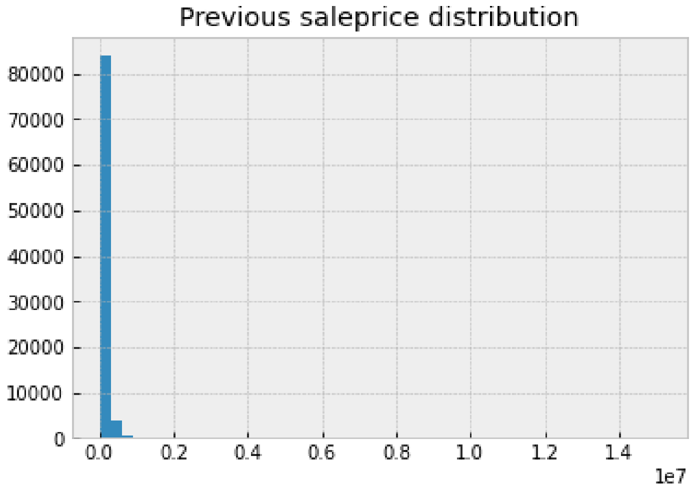
DATA FOR INVESTMENT STRATEGY

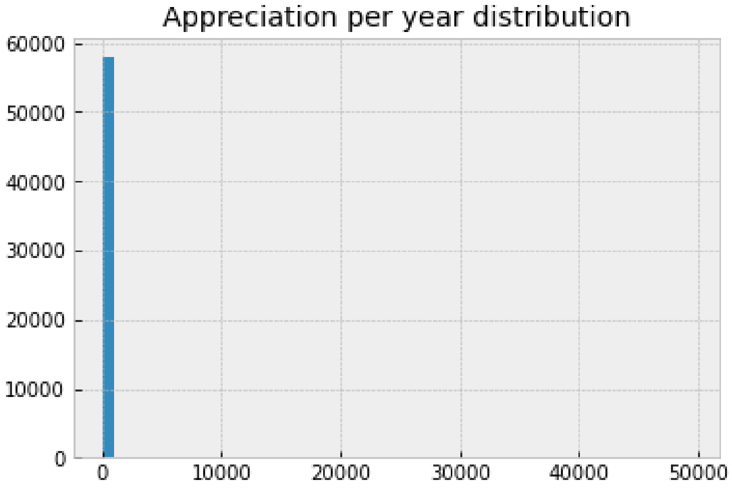
- To create our investment strategy, we need to be able to have buy and sell prices for the same securities
- Now, because we only have 1 row per unique house, we do not have a full record of its sale
- However, we do have records of previous two sales which we can use.
- Note that we are able to use only one previous sale because the PREVSALE2 has a ton of missing values
- Thus we next analyze the columns PREVSALEDATE and PREVSALEPRICE

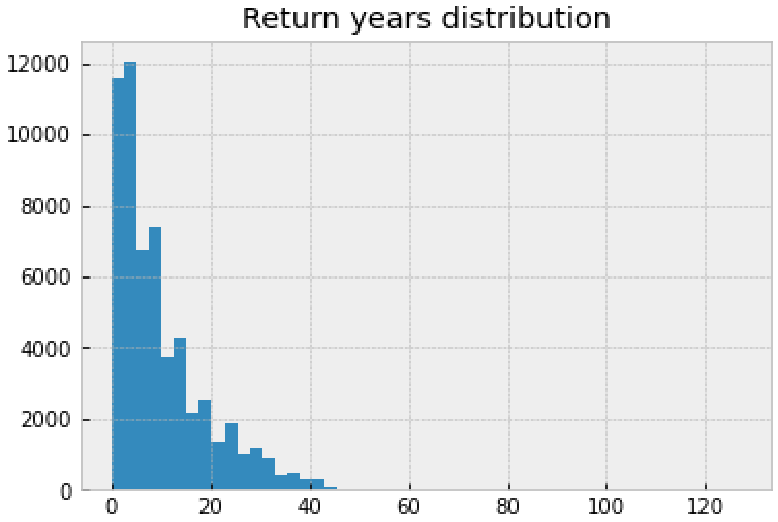
PARID	0001G00224060500	0001G00224070400
PROPERTYHOUSENUM	151	151
PROPERTYFRACTION		
PROPERTYADDRESS	FORT PITT BLVD	FORT PITT BLVD
PROPERTYCITY	PITTSBURGH	PITTSBURGH
PROPERTYSTATE	PA	PA
PROPERTYUNIT	UNIT 605	UNIT 704
PROPERTYZIP	15222	15222
OWNERDESC	REGULAR-ETUX	REGULAR
CLASSDESC	RESIDENTIAL	RESIDENTIAL
USEDESC	CONDOMINIUM	CONDOMINIUM
RECORDDATE	18-Sep-09	3-Jul-19
SALEDATE	18-Sep-09	22-Jun-19
SALEPRICE	265000	425000
SALEDESC	OTHER VALID	VALID SALE
PREVSALEDATE	30-Jul-07	13-Dec-07
PREVSALEPRICE	235000	365000
STYLEDESC	CONDO HR	CONDO HR
YEARBLT	2007	2007
EXTFINISH_DESC	Concrete	Concrete
ROOFDESC	ROLL	ROLL
GRADEDESC	VERY GOOD +	VERY GOOD +
CONDITIONDESC	AVERAGE	AVERAGE
TOTALROOMS	3	4
BEDROOMS	1	2
HEATINGCOOLING	B	B
HEATINGCOOLINGDESC	Central Heat with AC	Central Heat with AC

PREVIOUS SALE PRICE IS NOISY. DOES NOT MATCH ZILLOW

- We remove anything with saleprice less than $5 (bad data points as earlier)
- $\text{appreciation} = \frac{\text{SALESPRICE}_{\text{clean}}}{PREVSALPRICE} - 1$
- We also remove any row where $prev_{sale_{date}} > sale_{date}$
- $return_{years} = year(sale_{date}) - year(prev_{sale_{date}})$
- $appreciation_{per_{year}} = \frac{appreciation}{return_{years}+1}$







PREVIOUS SALE PRICE ADJUSTMENT

- Clip the $appreciation_{per_{year}}$ at +-25%

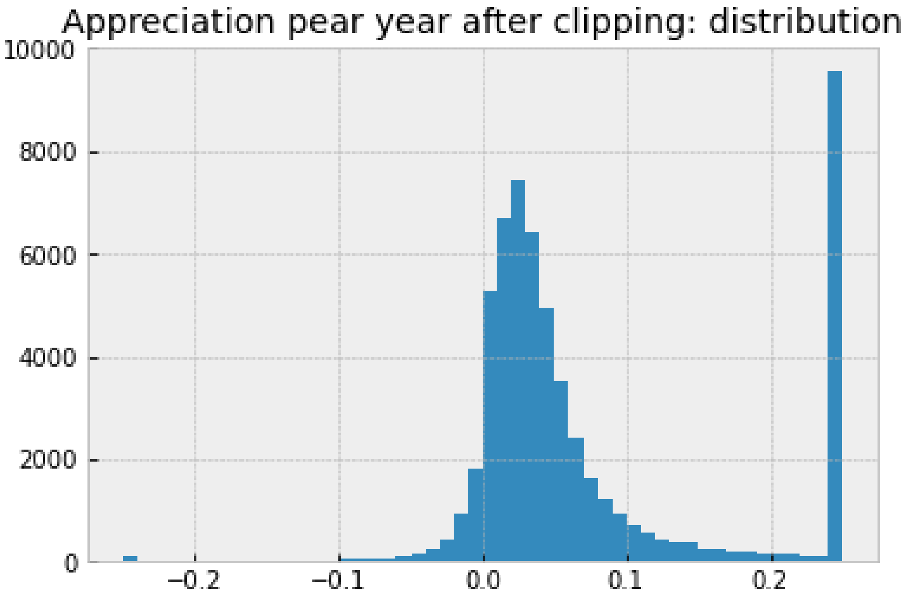
- We get a bimodal distribution with a LOT of values on the right tail. However we so not want to get rid of another 10k datarows so continue with this for now.

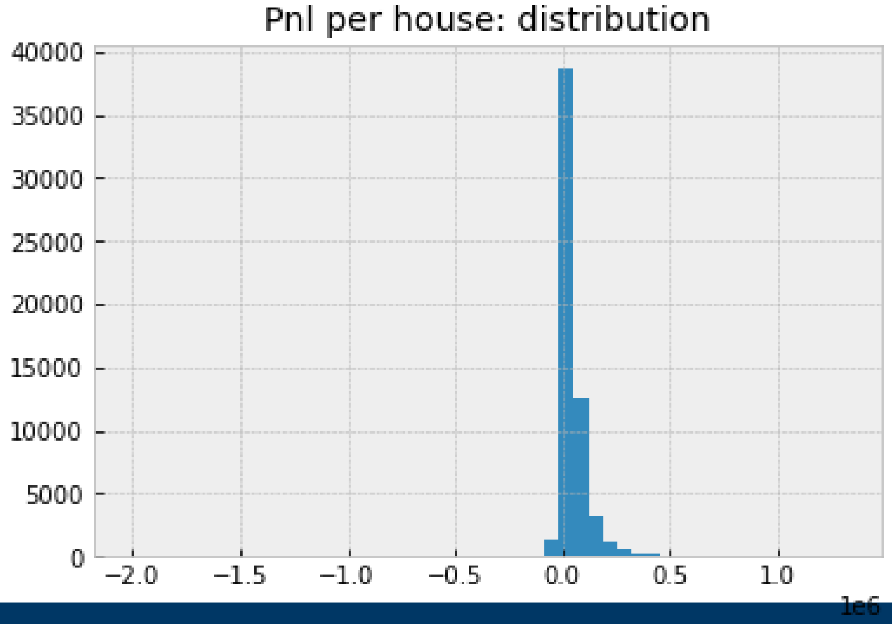
- Note the final investment strategy does not change if we remove these rows from our dataset.

- Compute back the $prevsaleprice_{clean}$ using the clipped per year appreciations

- We also normalize the clipped appreciation by converting to z-scores and clipped at +-3 on a monthly basis ($appreciationz$)

- Define $house_{pnl} = saleprice_{clean} - prevsaleprice_{clean}$





APPRECIATION PREDICTION MODEL – DATA SET UP

- Once we have a clean dataset we build the prediction model.
- We aim to model $appreciationz$ using various features and will use the forecasted values in our investment strategy
- Note, a lot of these variable are correlated (e.g., totalrooms, bedrooms) so we exclude some of the columns
- We convert categorical features into one-hot encodings
- Split the data into train and test as:
  - Train period: 2000-2015
  - Test: 2016-2020

Numerical	Categorical
LOTAREA	PROPERTYZIP
YEARBLT	USEDESC
TOTALROOMS	STYLEDESC
FINISHEDLIVINGAREA	EXTFINISH_DESC
FULLBATHS	BASEMENTDESC
	GRADEDESC
	HEATINGCOOLINGDESC

	TOTAL ROOMS	BED ROOMS	FULL BATHS	FINISHED LIVINGAREA
TOTAL ROOMS	1.00	0.83	0.58	0.76
BED ROOMS	0.83	1.00	0.53	0.71
FULL BATHS	0.58	0.53	1.00	0.66
FINISHED LIVINGAREA	0.76	0.71	0.66	1.00

APPRECIATION PREDICTION MODEL - APPROACHES

- Given the feature space, expected non-linear relationships, and multi-co-linearity, we expect non-linear models ensemble models to be most suitable

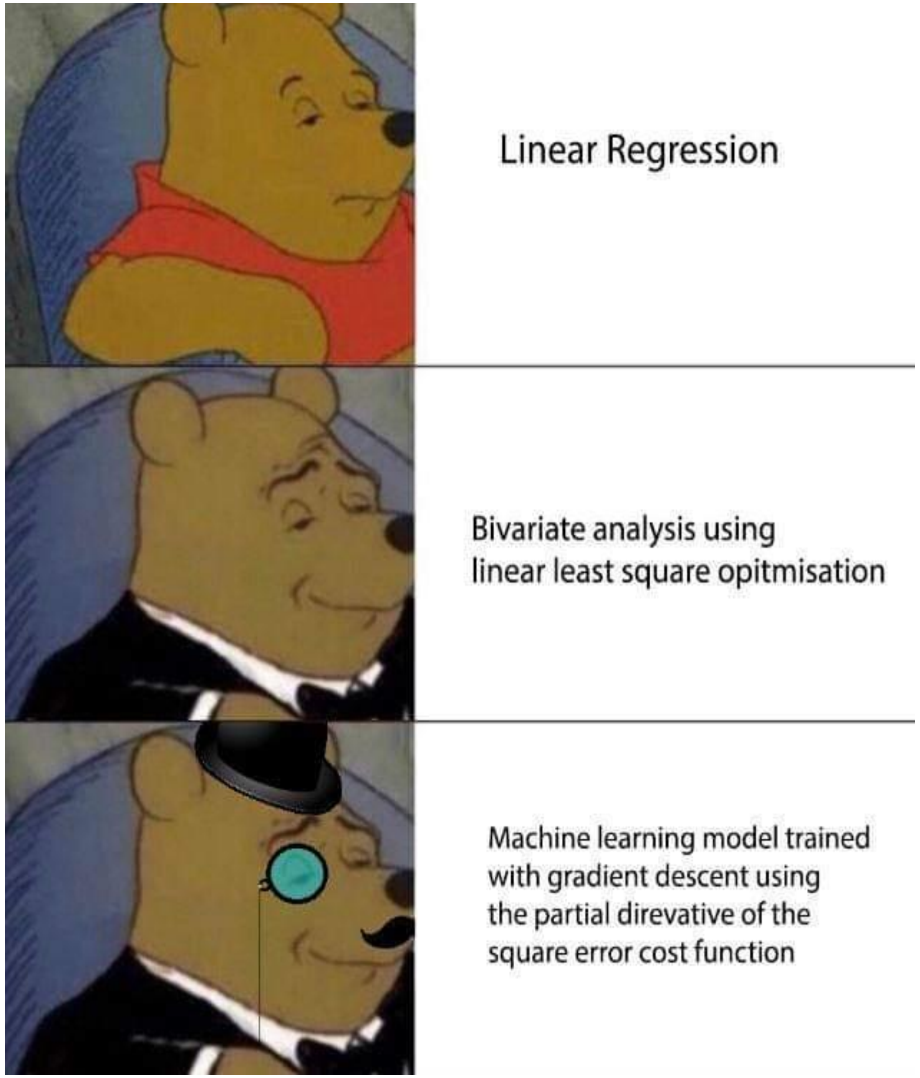
- We implement three models:

  - OLS regression: primarily as the baseline model

  - Ridge regression: given that we have correlated features, this will give better beta loadings

  - Random Forest regression: we expect this to perform best due to
    - Non-linear relationships
    - Better handling of multi-collinearity
    - Ensemble approach better suited for noisy data

- Next we look at some results

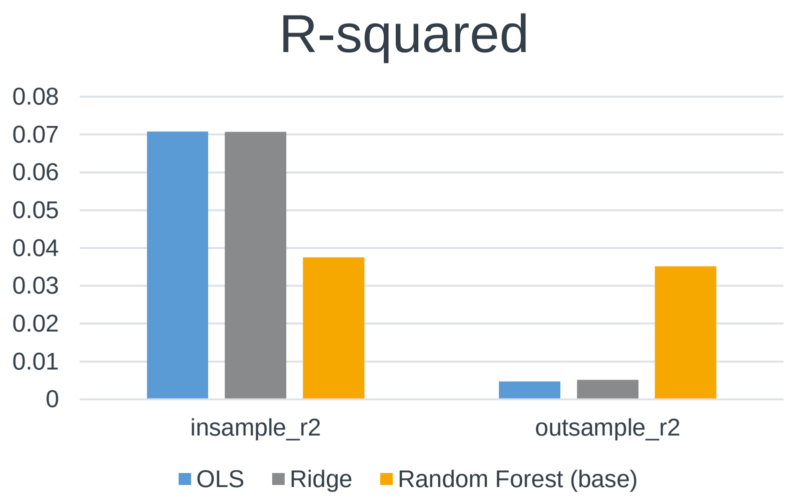


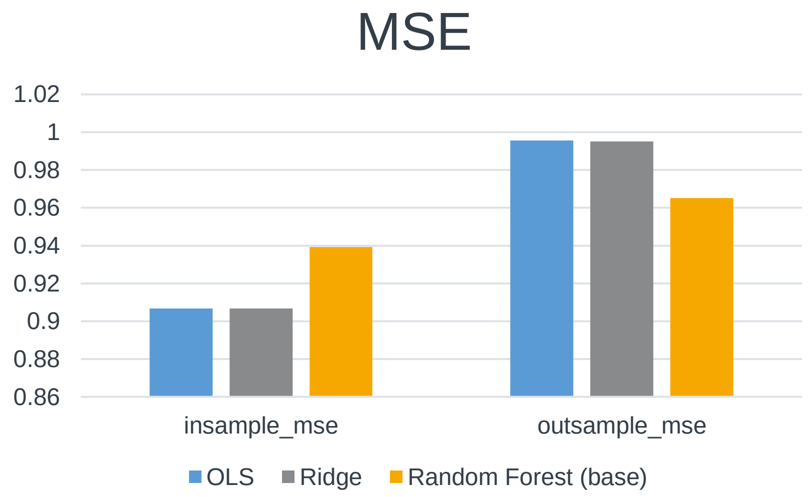
APPRECIATION PREDICTION MODEL – RESULTS

- We present a simple in-sample out-sample results.
  - Train period: 2000-2015
  - Test: 2016-2020

- Note this is not our “proposed model”. We obviously need to perform hyperparameters tuning for the Tree based model.

- Just highlights performance of the base models:
  - RFR performs best and holds its performance OOS
  - Interestingly linear models perform significantly worse OOS vs in-sample.



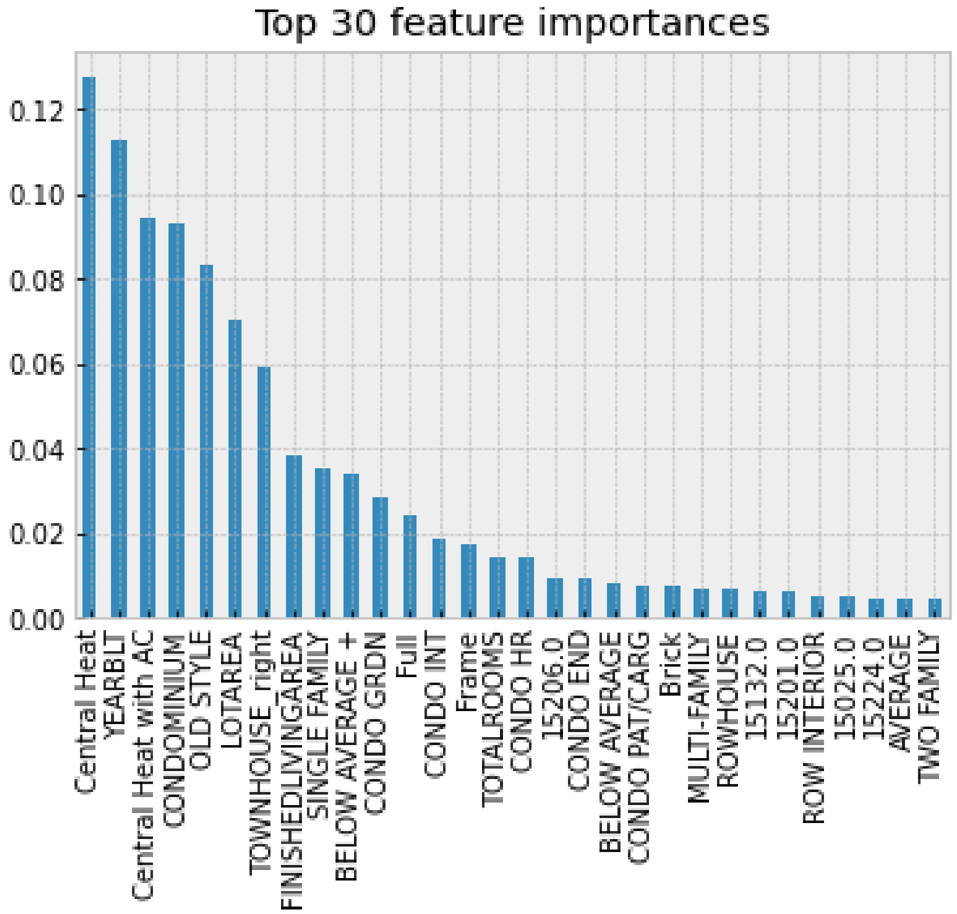


FEATURE IMPORTANCE

- Analyze the main features that affect the appreciation using feature_importance function

- YEARBLT, LOT_AREA, FINISHED_LIVING_AREA rank among the top features

- Further, houses with central heat, and house types (condominiums, single family etc.) are also important.



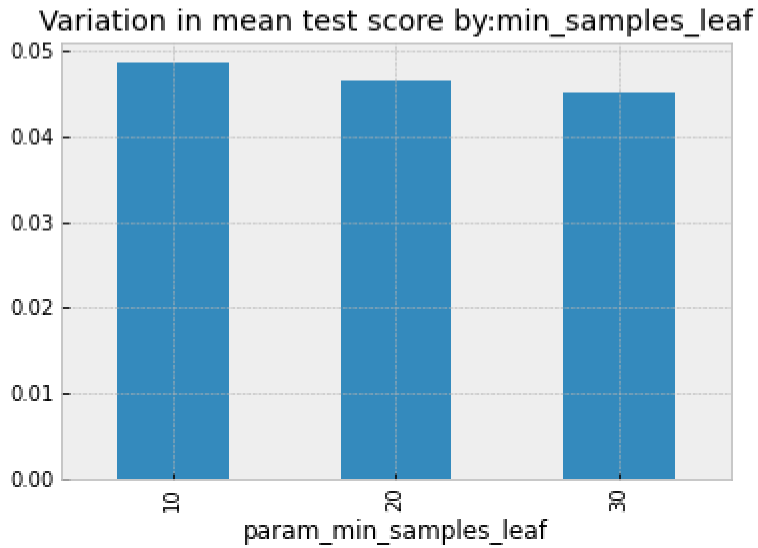
RANDOM FOREST MODEL: HYPERPARAMETER TUNING

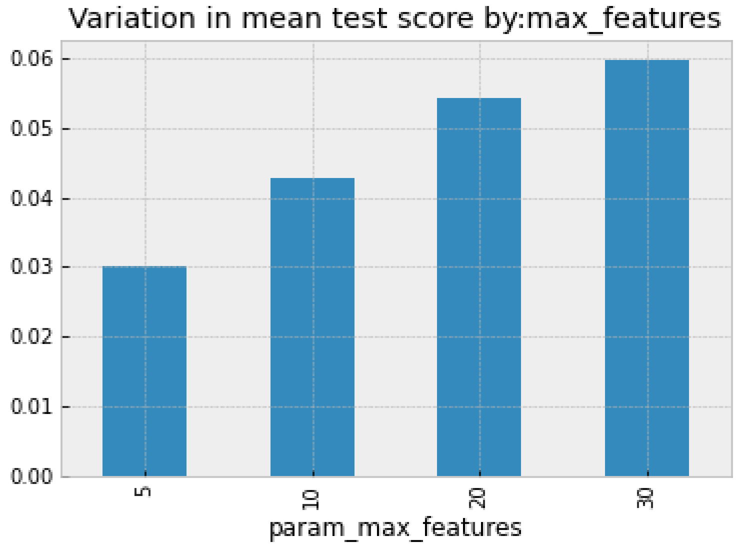
- We perform hyper-parameter turning using GridSearchCV
- Also employ TimeSeriesSplit for cross-validation
- Note its critical to use TimeSeriesSplit in financial data due to time dependency.
- Do not use a separate validation set

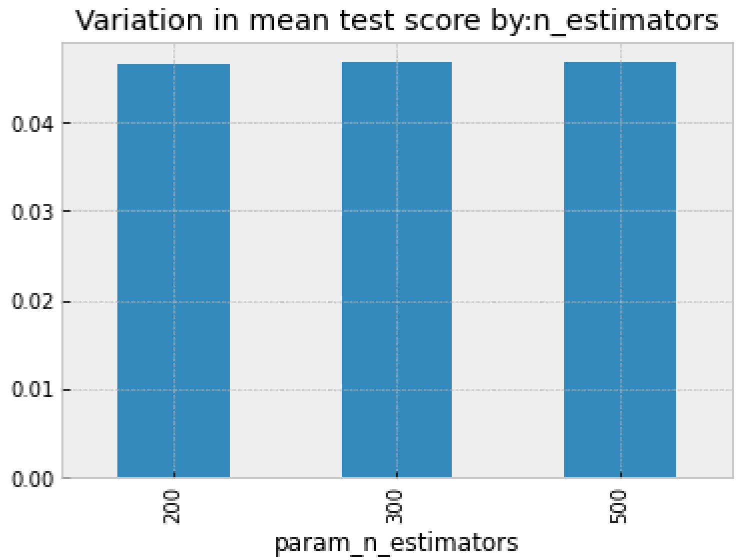
```
rfr_grid = RandomForestRegressor()

param_search = {
    'max_depth': [4,8,16,32],
    'max_features': [5,10,20,30],
    'min_samples_leaf': [10,20,30],
    'n_estimators': [200,300,500],
    'n_jobs' : [-1]
}

start_time =  time.time()

tscv = TimeSeriesSplit(n_splits=5)
gridsearch_rfr = GridSearchCV(estimator=rfr_grid, cv=tscv,
                              param_grid=param_search)
gridsearch_rfr.fit(x_train, y_train)
```

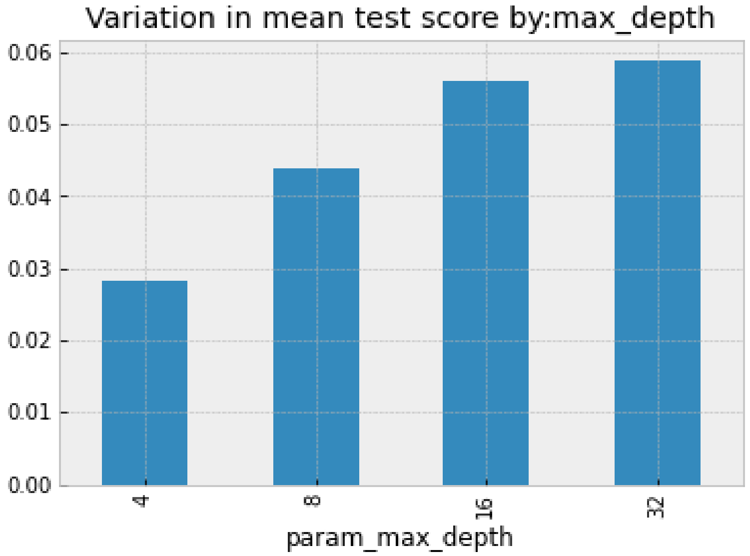
RANDOM FOREST MODEL: HYPERPARAMETER TUNING

- Test scores get better with increasing max_features and with max_depth
- Test scores get slightly worse with increasing min_sample_leaf
- Not significantly affected by #trees
- The test scores here is the default used by the GridSearchCV which is R^2 in case of regressions.









GRIDSEARCHCV BEST_ESTIMATOR – PRACTICAL ISSUES WITH ML!

```
Default: RandomForestRegressor(max_depth=3, max_features='sqrt', min_samples_leaf=10, n_estimators=200)

Optimized: RandomForestRegressor(max_depth=16, max_features=30, min_samples_leaf=10, n_estimators=500)
```

- In the standard machine learning approach we use the hyperparameter tuning and choose the best estimator in our final proposed model.

- However, if the model is overfit, the out of sample performance will be bad even after using a validation set.

- The downsides of an overfit model in the financial application are significantly higher than in other applications. Thus its never a good idea to blindly accept the best_estimator from a grid search style tuning approach.

- In our analysis the default model performs slightly better than the optimized model. This could be due to overfitting in the max_depth and max_features parameters (comparison not shown).

- In an ideal setting, we would spend <u>significantly</u> more time on this step, however given the time constraint, we continue to use the forecast from our default RFR model.

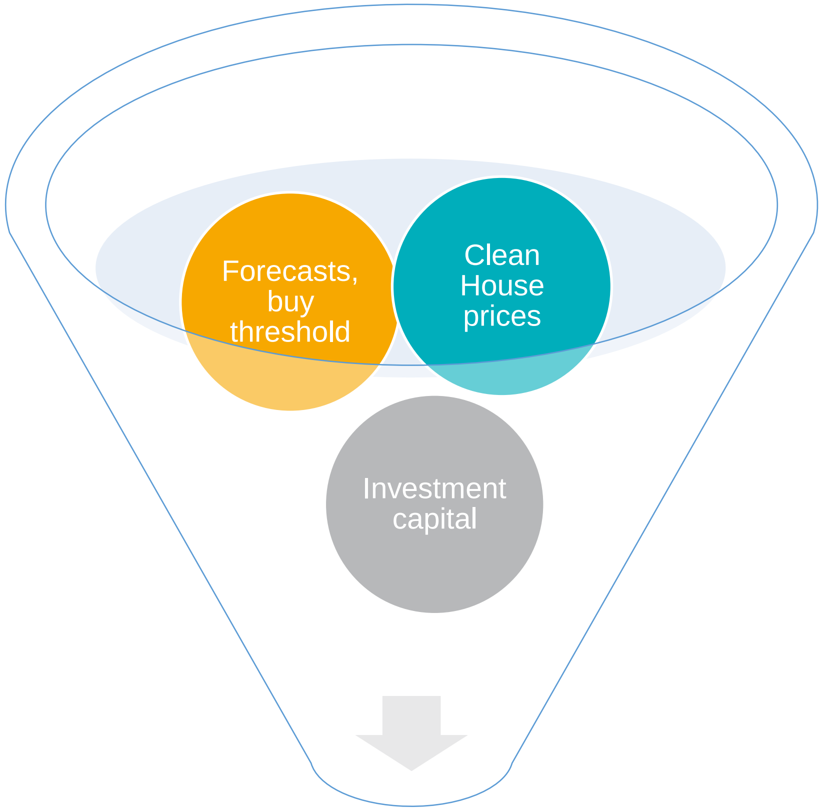
INVESTMENT STRATEGY: INPUTS

Forecast scores:

- Propose to use predictions from the random forest regression model.
- Convert the forecasted zscores to uniform distribution
- Compare with linear and random model

buy_threshold:

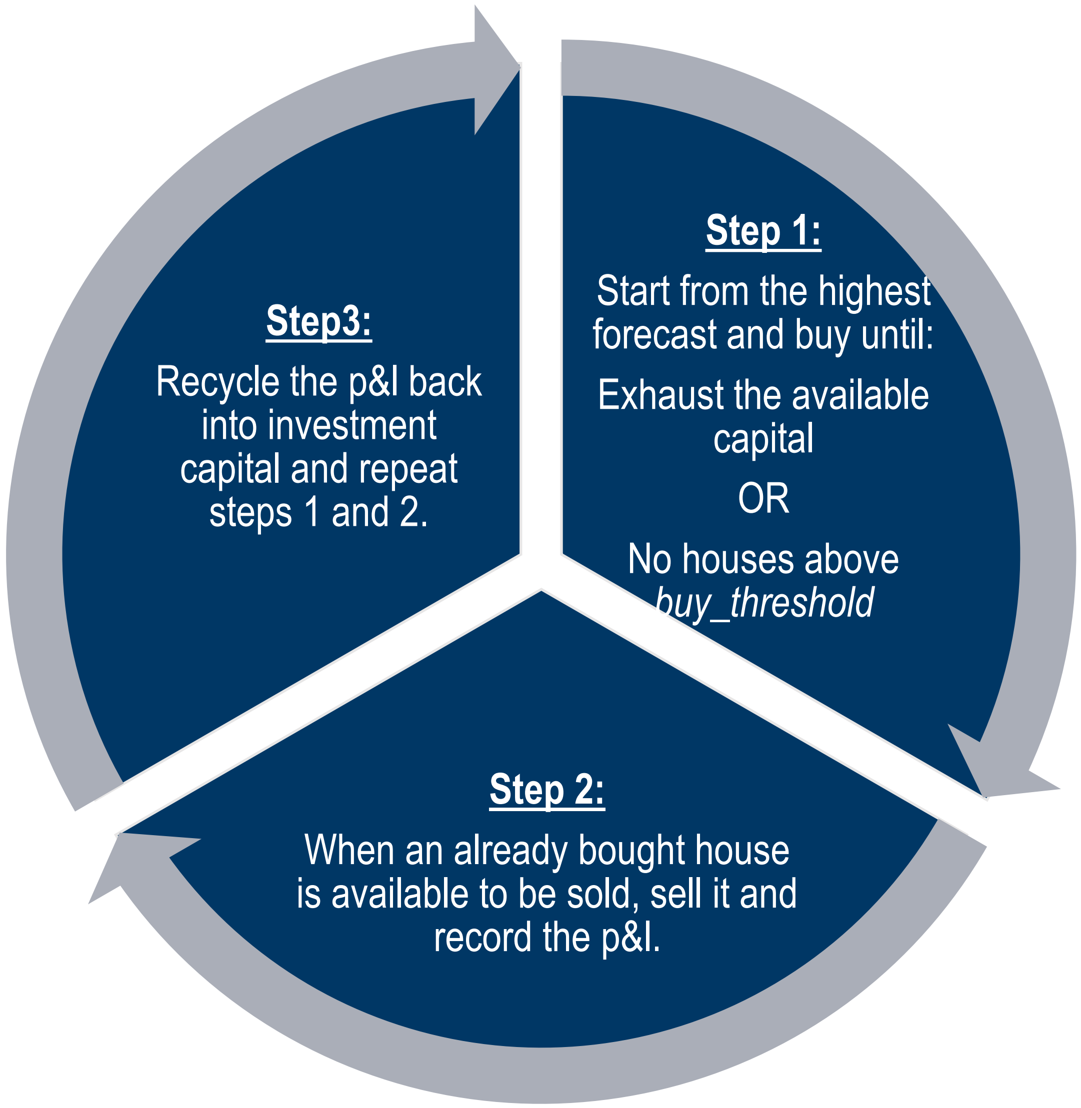
- Converting to uniform lets us choose top n-percent of opportunities
- A threshold of 0.3 means buy only the top 70% of the available houses per month
- Capital limit is the bounding condition (starts with $5mn) leading to results having low sensitivity to the threshold
- Can tune for the optimal threshold. To keep the presentation short we have excluded that analysis here and use a conservative 0.5.



Housing Investments

INVESTMENT STRATEGY : PROCESS

- Forecast: random forest model
- Buy_threshold: 0.5
- Outputs:
  - NAV over time
  - # houses bought/sold
  - Total PnL
  - Investment
  - Records of houses bought and sold



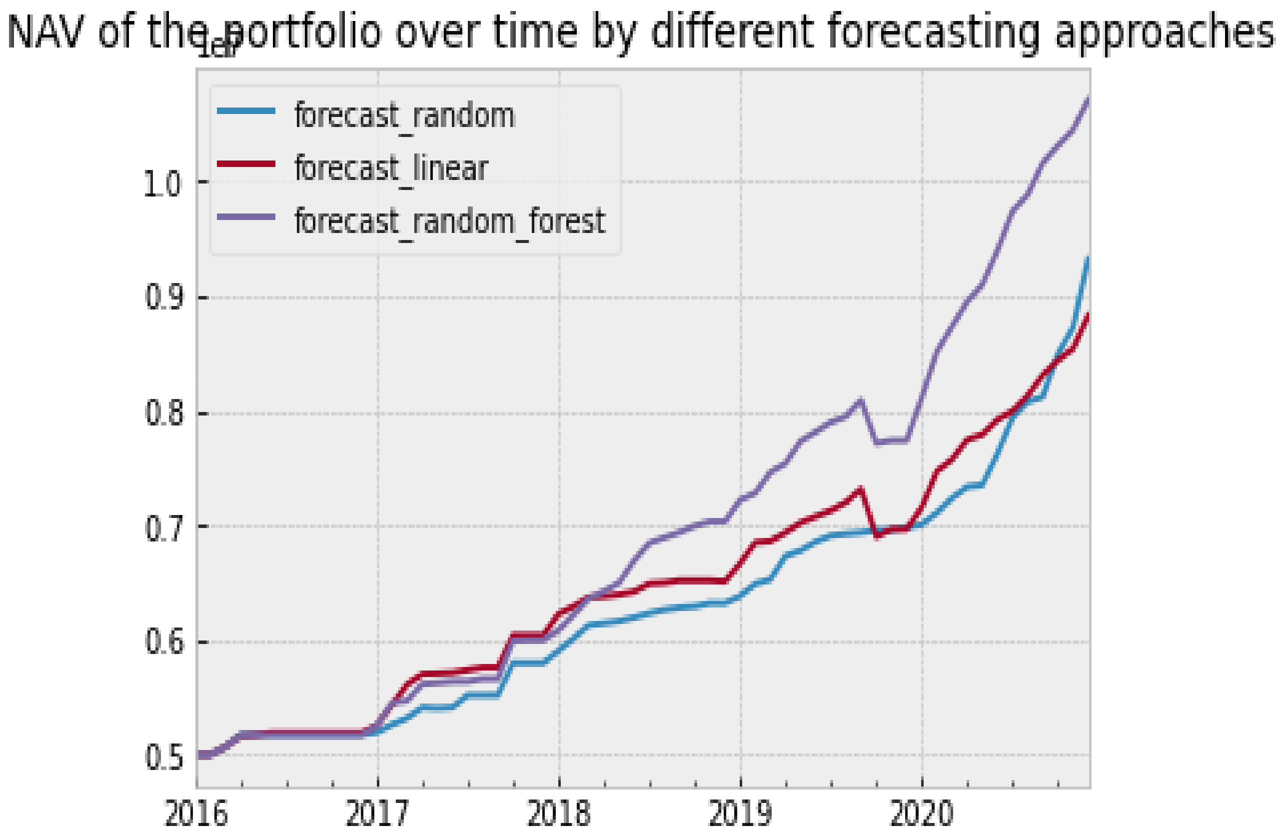
INVESTMENT ENGINE: SAMPLE OUTPUT

- The investment engine records detailed information each month which can be analyzed to improve the models.

	houses_bought	houses_sold	capital_invested	capital_redeemed	pnl_booked	capital_remaining	NAV
2016-01-31	23	0	4.936997e+06	0.0	0.000000	6.300251e+04	5.000000e+06
2016-02-29	2	0	6.266667e+04	0.0	0.000000	3.358444e+02	5.000000e+06
2016-03-31	6	1	3.130667e+05	343142.0	68628.400000	3.041118e+04	5.068628e+06
2016-04-30	4	1	4.882667e+05	466685.0	93337.000000	8.829511e+03	5.161965e+06
2016-05-31	2	1	5.000000e+04	42000.0	8400.000000	8.295111e+02	5.170365e+06
2016-06-30	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06
2016-07-31	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06
2016-08-31	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06
2016-09-30	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06
2016-10-31	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06
2016-11-30	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06
2016-12-31	0	0	0.000000e+00	0.0	0.000000	8.295111e+02	5.170365e+06

INVESTMENT STRATEGY PERFORMANCE

- We find that random forest model performs best with the final NAV above $10mn.
- We also include the performance of the linear and random models alongside.
- Note the random forecast has performed well because of the trend in the financial markets leads to risk premia harvesting or capturing the real-estate beta.
- If this were a long-short strategy, we would expect the random forecast to be pure noise and thus have zero long term expected return.



	Final NAV	# Bought
Random	$8,560,397.00	109
Linear	$8,825,877.00	150
Random Forest	$10,602,610.00	209

MARTINGALE?

- A martingale is defined as a sequence of random values: $x_0, x_1, x_2, \ldots\ldots x_N$ with the property:
  - The conditional expectation of $E[x_{N+1}|x_0, x_1, ..x_N] = x_N$
  - Hence, given a martingale, the best predictor of the value of the random variable at

  time $t + 1$ is the value of the random variable at time t.

- Conceptually we do not expect the housing price to be a martingale in the real world. We do know from experience that house prices have a relatively stable drift and trend up over time. The best predictor of housing price at t+1 is not the housing price at t, but rather a function of the house price at t and the risk premia embedded in the housing market.

- We empirically show this as well by analyzing the housing price index. Looking at the index it’s clear that the market has an average positive drift in it, thus proving that the housing price is not a martingale in the real world measure.

POTENTIAL IMPROVEMENTS

- Given the limited time and scope of the project there are a lot of areas of improvement
- Forecasting model:
  - Employ other ML models: Neural nets could work well with this kind of data
  - Tune the random forest regression model better: we spent limited time
  - Could employ here classifications here as they work better (than regression) in noisier data
  - Employ boosting approaches rather than bagging
  - High dimensionality due to one hot encoding of the zipcodes. Can aggregate them better.
- Investment engine:
  - No optimization done at the investment engine level. Used a base model
  - No way to take advantages of better opportunities in the future. Optimize for a level of cash held, assuming there would better opportunities in the future.
  - Can reduce net beta exposure to the real estate market depending on the forecast from a house price index model

Questions?