Sberbank Russian Housing Market - Top 1% Solution

Team
Chase Edge, Grant Webb
And Brandy Freitas

Project Overview

Project Overview

- Kaggle Competition
- Predict housing prices in Moscow during July 2015 to May 2016 using data from August 2011 to June 2015
- Data includes housing transaction information (e.g. square meter, number of rooms and build year), neighborhood details and macroeconomic information

Machine Learning Checklist

1. Frame the Problem and Look at the Big Picture
2. Get The Data
3. Explore the Data
4. Prepare the Data
5. Short List Promising Models
6. Fine-Tune the System

Look at the Big Picture

Value to Sberbank

- Mitigate Risk
  - Help avoid overlending (issuing mortgages in excess of the value of the home)
- Valuation
  - Help banks value their portfolio
- Predictions
  - To give confidence to renters, developers and lenders when they sign a lease or purchase a building

Data Exploration

Data Overview

Housing Data

- Number of Observations
  - Training - 30,471
  - Test - 7,662
- Number of Features - 290

Macro Data

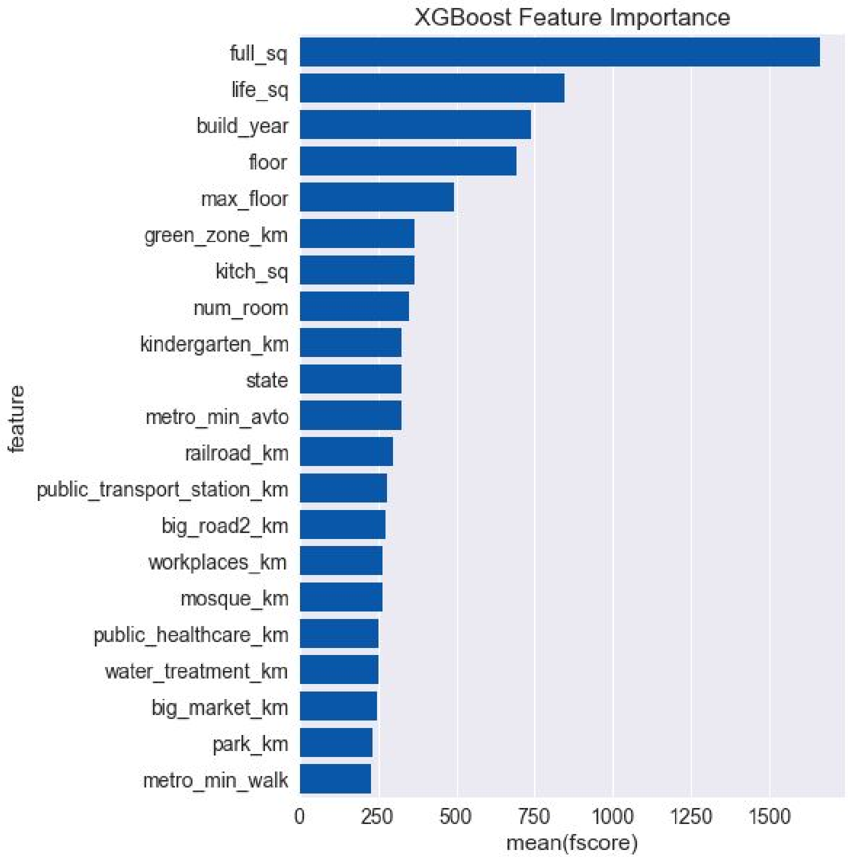
- Number of Observations - 2,484
- Number of Features - 100

Data Overview - Most Important Features

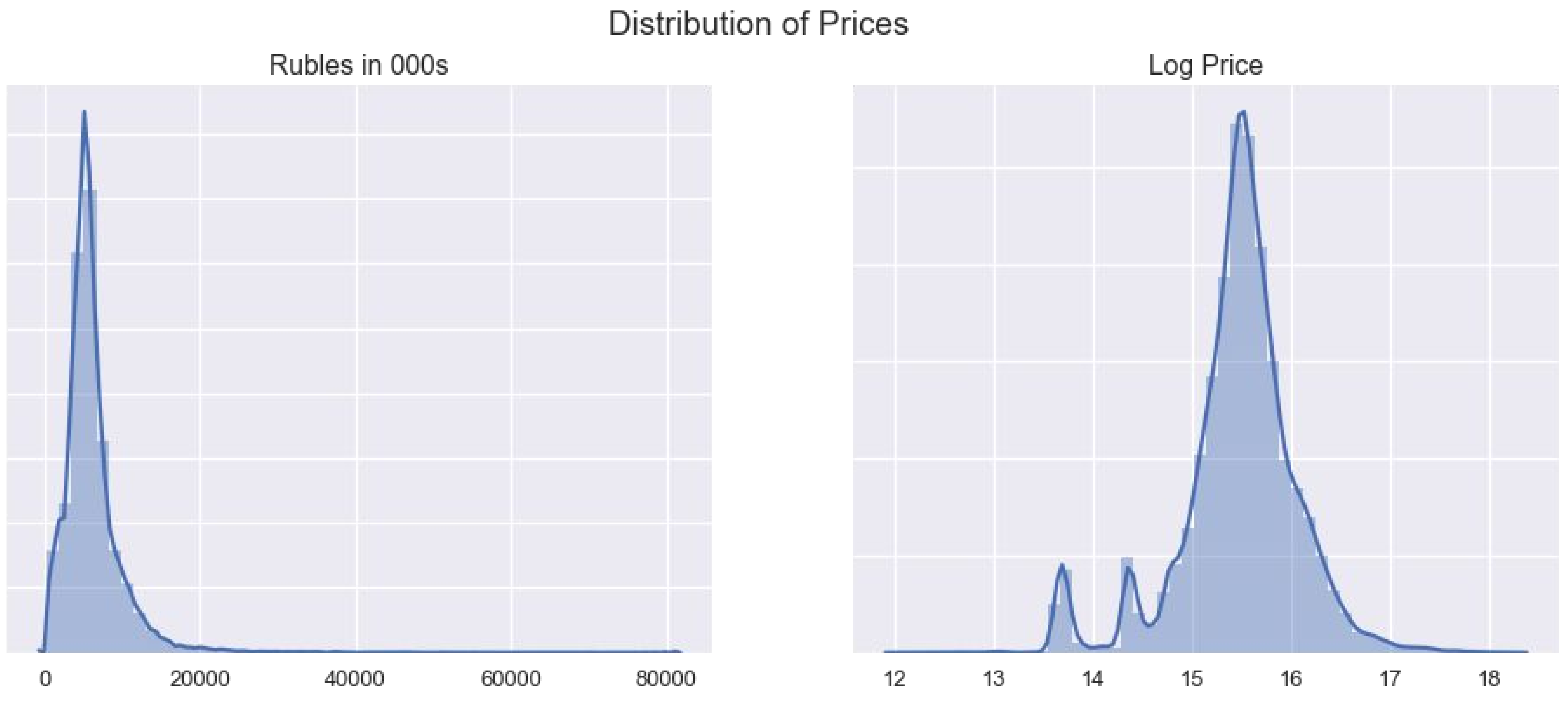
- Initial Thoughts
  - Square Meters
  - Number of Rooms
  - Build Year
  - Location

Data Overview - Most Important Features

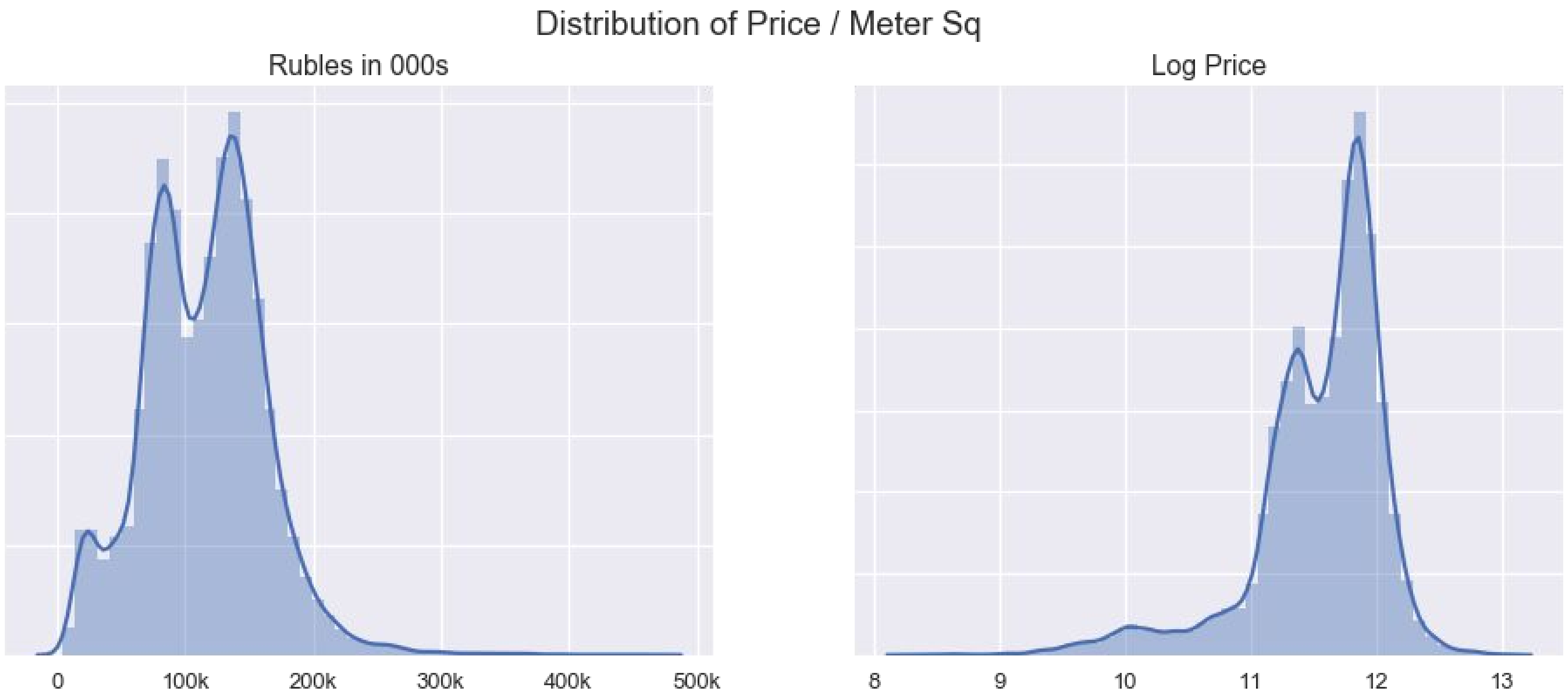
- Initial Thoughts
  - Square Meters
  - Number of Rooms
  - Build Year
  - Location



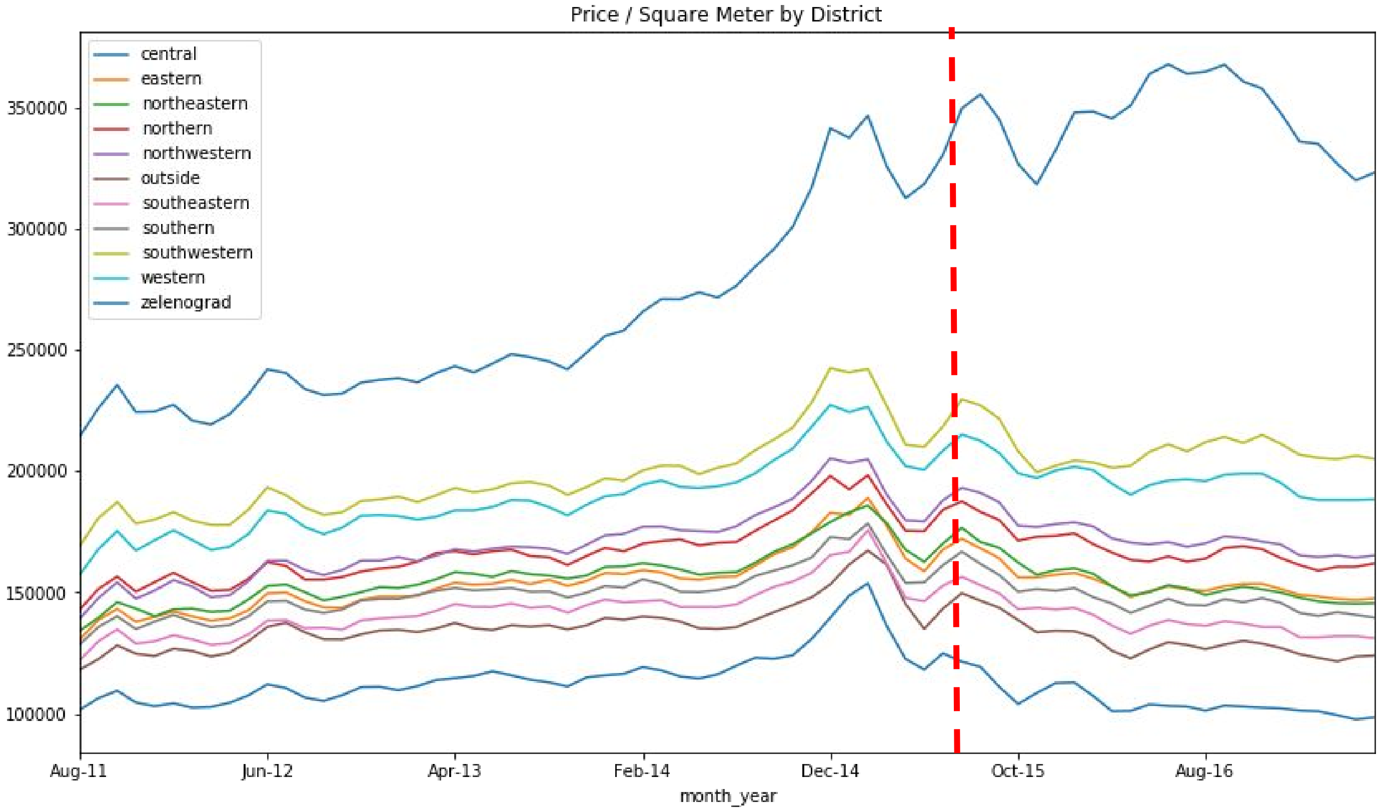
Housing Price



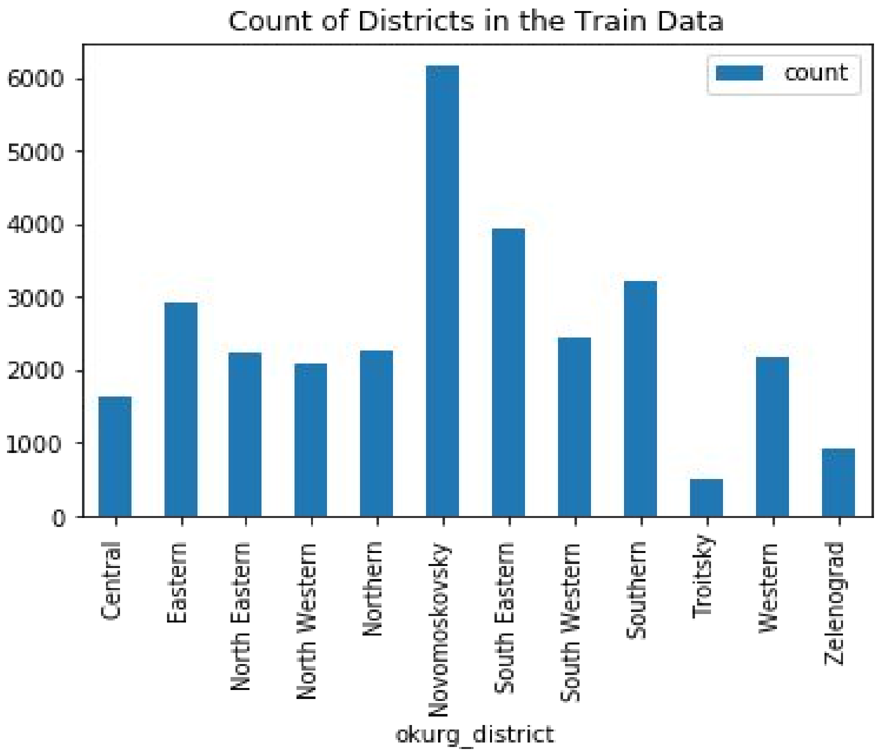
Housing Price

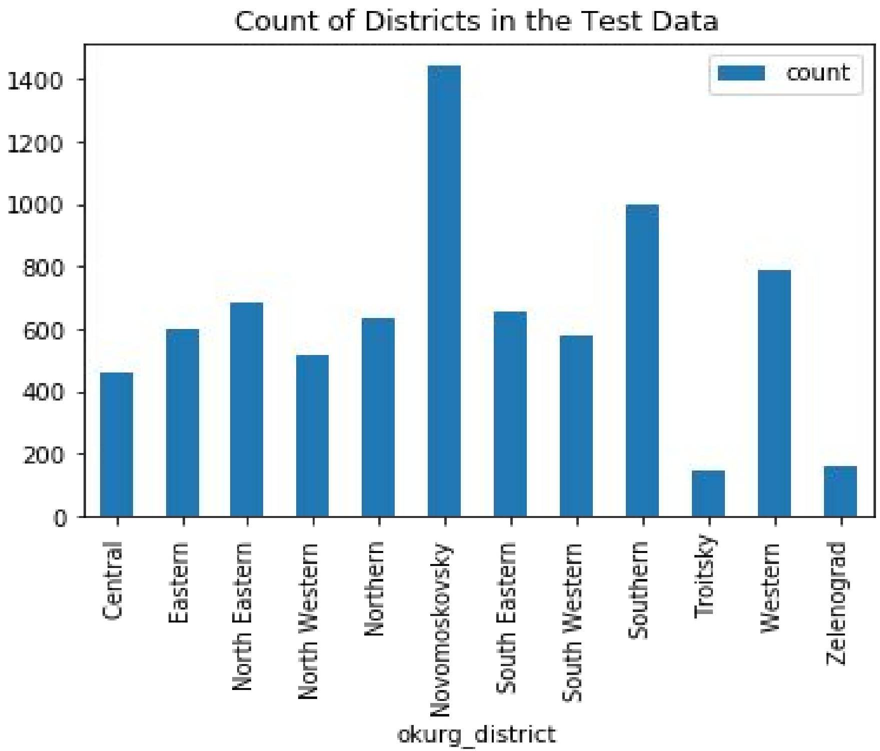


Prices by District



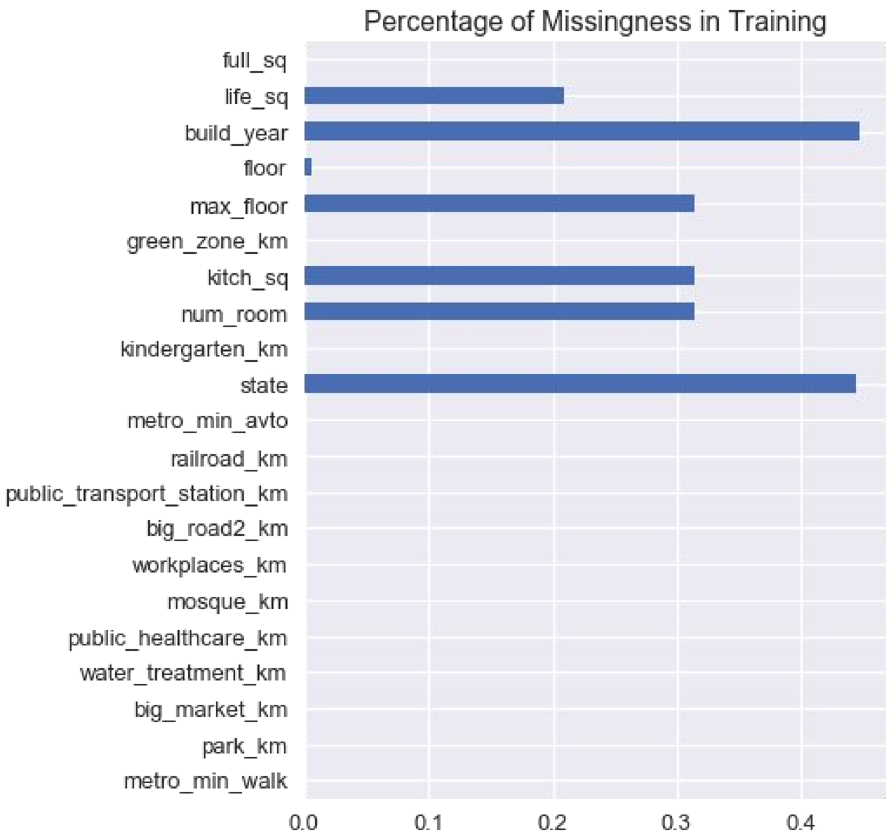
Observations by District

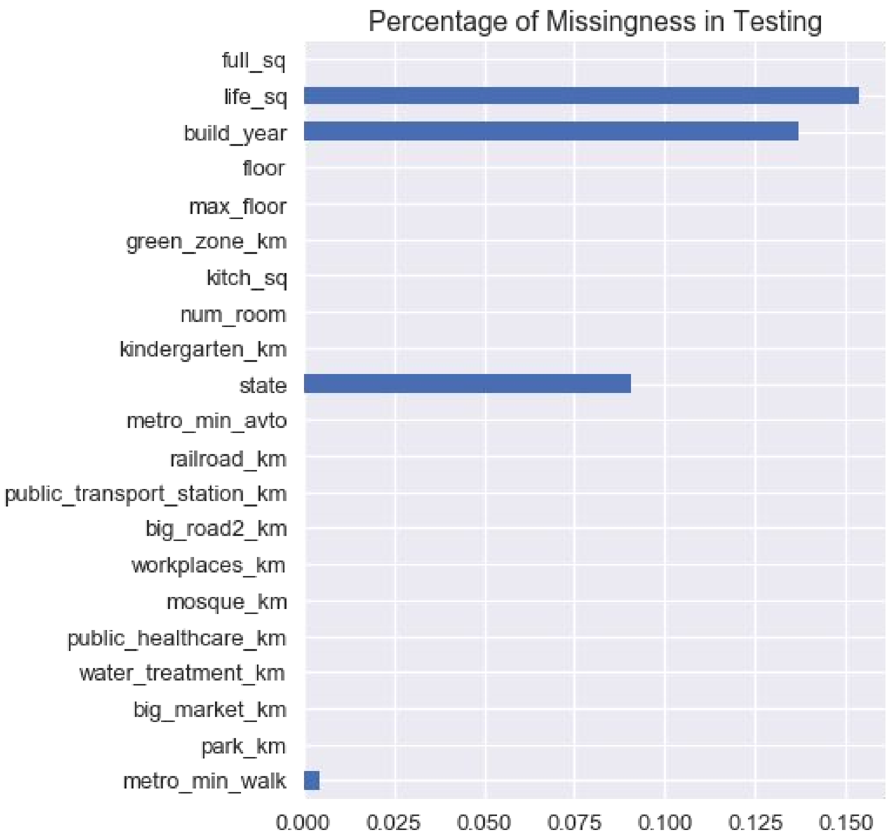




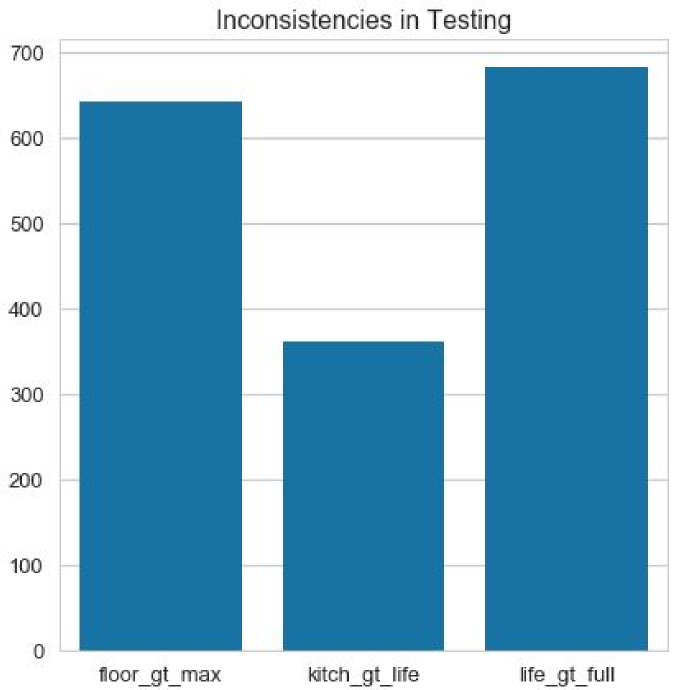
Data Preparation

Missingness in Top Features

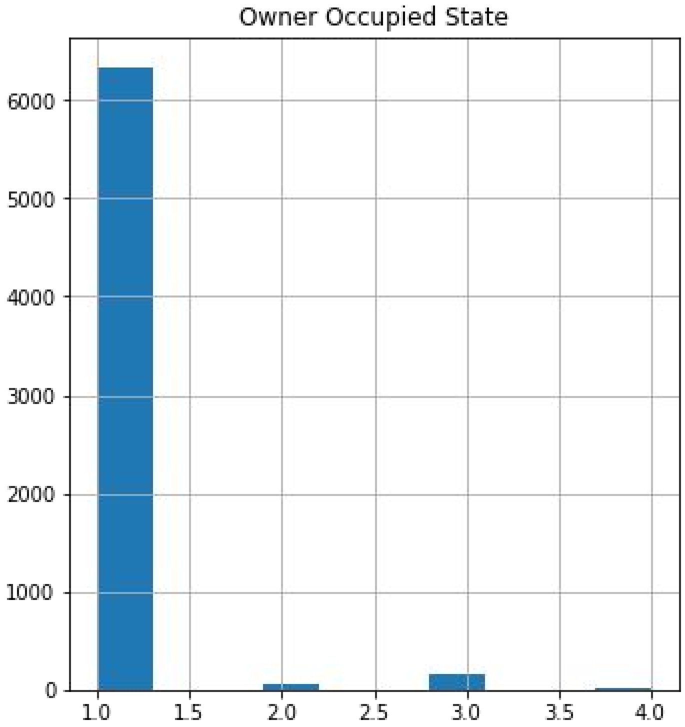


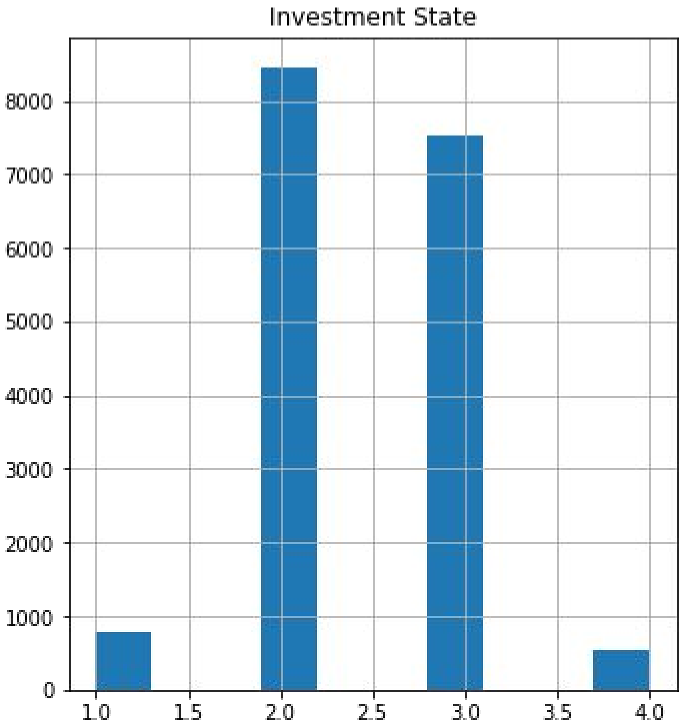


Data Inconsistencies

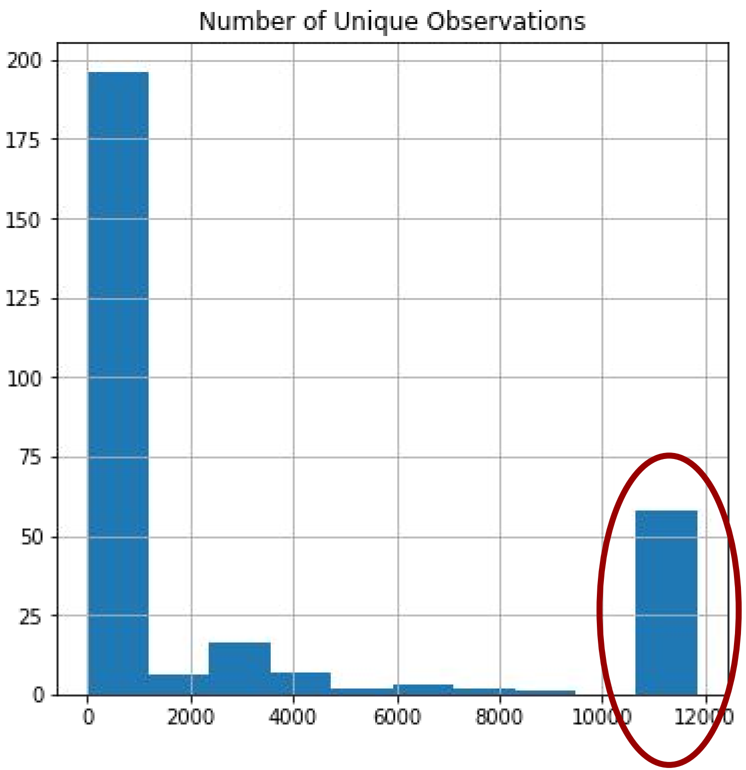


Data Inconsistencies





Imputations - KNN by Neighborhood



kindergarten_km	11852
park_km	11852
public_transport_station_km	11851
public_transport_station_min_walk	11852
water_km	11851
mkad_km	11852
ttk_km	11852
sadovoe_km	11852
bulvar_ring_km	11852
kremlin_km	11852
big_road1_km	11852
big_road2_km	11852
railroad_km	11852
oil_chemistry_km	11852
nuclear_reactor_km	11852
radiation_km	11852
power_transmission_line_km	11852
thermal_power_plant_km	11852
ts_km	11849
basketball_km	11852
hospice_morgue_km	11852
big_church_km	11852
church_synagogue_km	11852
mosque_km	11852
museum_km	11852
exhibition_km	11852
catering_km	11852

	count
kremlin_km	
20.549464	976
0.072897	603
23.373697	582
20.666814	364
22.222434	319
29.133765	288
25.735256	282
15.869044	275
21.609733	254
18.752843	232
25.595974	229
19.763938	215
22.567655	202

Feature Engineering & Insights

Feature Engineering

Sq_diff = Full_sq - Kitch_sq

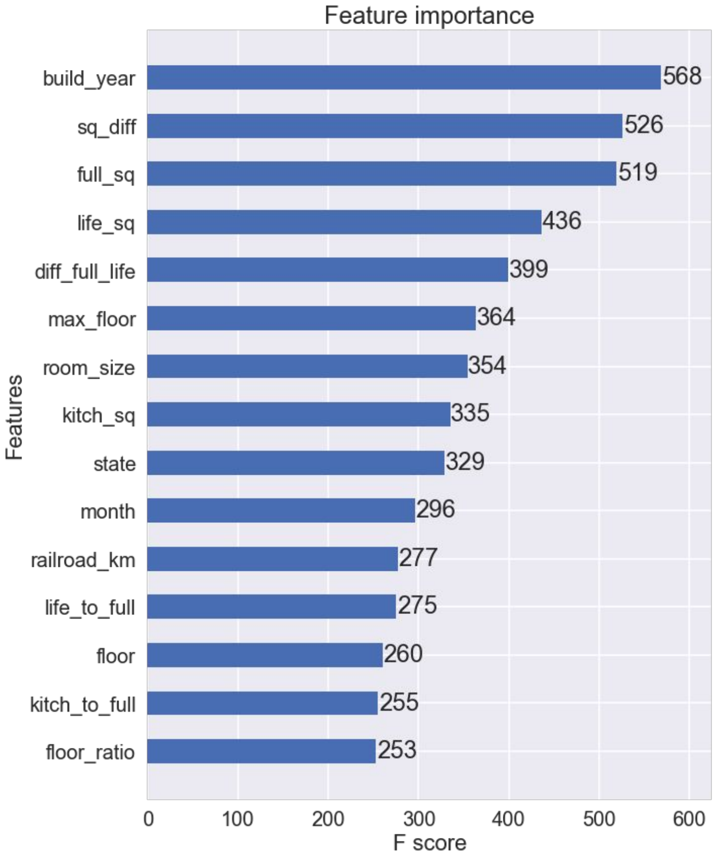
Floor_ratio = Floor / Max_floor

Life_to_full = Life_sq / Full_sq

Kitch_to_full = Kitch_sq / Full_sq

Room_size = Life_sq / Num_room

Month = month of sale



Decline in Russian Housing Prices

Dealing with the Decline

- **Problem**
  - Housing prices declined in 2015 and 2016
  - Model predicts values that are too high

- **Solutions**
  - Incorporate economic data
  - Make downward adjustments to predicted values
  - *Adjust prices for fluctuations in the market based on a price index*

Price Index

- The Data
  - Russian government statistics on the monthly rental prices within Moscow
- The Index
  - 3 month rolling average of 3 bed, 2 bed and 1 bed rentals in Moscow
  - Indexed to the start of the training data (August 2011)
  - Averaged the 3 indices
- The Application
  - Adjust all prices in the training data for changes in the index (divide by index)
  - Model predicts prices as if they occurred in August 2011
  - Adjust the predicted values for the index (multiply by index)

Price Index

rent_index
1.20
1.15
1.10
1.05
1.00
08/2011
06/2012
04/2013
02/2014
12/2014
10/2015
08/2016
month_year

Price Index - Example

- Training Transaction
  - Date - April 2013
  - Price - RUB 5,693,972
  - Index - 1.13
  - Adjusted Price - RUB 5,038,913
- Predicted Value
  - Date - April 2016
  - Price - RUB 3,902,007
  - Index - 1.06
  - Adjusted Price - RUB 4,136,127

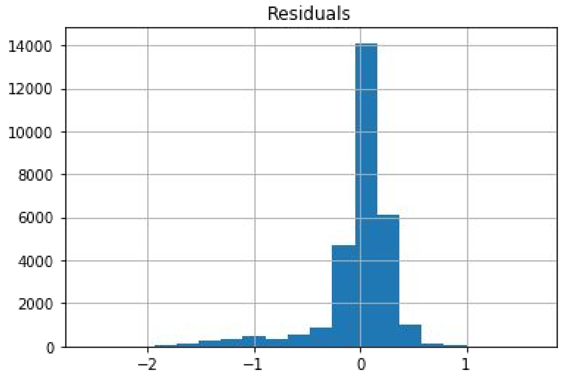
Short-list Promising Models

KISS - OLS with 2 Variables

Features = full_sq, sub_area

Response = log_price

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              log_price   R-squared:                       0.574
Model:                            OLS   Adj. R-squared:                  0.572
Method:                 Least Squares   F-statistic:                     270.5
Date:                Tue, 30 May 2017   Prob (F-statistic):               0.00
Time:                        18:13:54   Log-Likelihood:                -9345.5
No. Observations:               29096   AIC:                         1.898e+04
Df Residuals:                   28951   BIC:                         2.018e+04
Df Model:                         144
Covariance Type:            nonrobust
===========================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
Intercept                     14.8414      0.022    682.397      0.000      14.799      14.884
full_sq                        0.0146   9.91e-05    147.232      0.000       0.014       0.015
sub_area_Ajeroport             0.1418      0.037      3.827      0.000       0.069       0.214
sub_area_Akademicheskoe        0.2088      0.032      6.483      0.000       0.146       0.272
sub_area_Alekseevskoe          0.1792      0.040      4.427      0.000       0.100       0.258
```

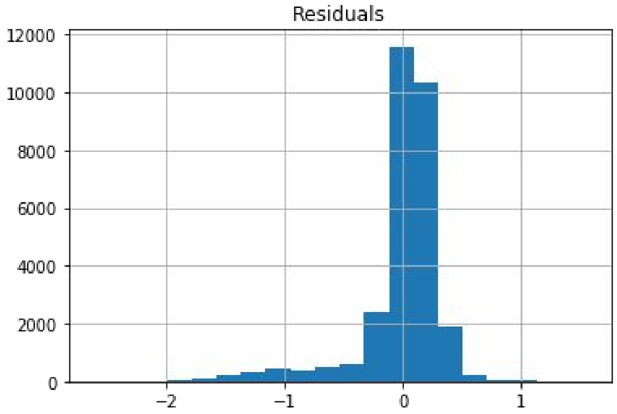


Your submission scored 0.38249, which is not an improvement of your best score. Keep trying!

KISS - OLS with 7 Variables

Features = full_sq, floor, max_floor, life_to_full, kitch_to_life, product_type, sub_area

Response = log_price



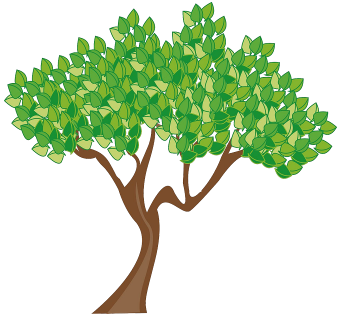

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              log_price   R-squared:                       0.577
Model:                            OLS   Adj. R-squared:                  0.575
Method:                 Least Squares   F-statistic:                     265.3
Date:                Tue, 30 May 2017   Prob (F-statistic):               0.00
Time:                        23:59:56   Log-Likelihood:                -9220.6
No. Observations:               29096   AIC:                         1.874e+04
Df Residuals:                   28946   BIC:                         1.998e+04
Df Model:                         149
Covariance Type:            nonrobust
============================================================================================
                                coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------
Intercept                     9.9379      0.021    462.806      0.000       9.896       9.980
full_sq                       0.0142      0.000    133.472      0.000       0.014       0.014
floor                         0.0013      0.000      2.687      0.007       0.000       0.002
max_floor                     0.0050      0.000     10.345      0.000       0.004       0.006
life_to_full                 -0.1116      0.024     -4.733      0.000      -0.158      -0.065
kitch_to_life                -0.1586      0.028     -5.650      0.000      -0.214      -0.104
product_type_Investment       4.9614      0.009    553.891      0.000       4.944       4.979
product_type_OwnerOccupier    4.9764      0.014    349.646      0.000       4.949       5.004
```

simple_linear_052716_log_multiples.csv **0.34644**

Tree-Based Models

Decision Tree

Unrestrained Decision Trees are known to overfit as shown:



```
Accuracy on training set 1.000
Accuracy on test set 0.224
The severe overfitting of the unrestrained Decision Tree!!
```

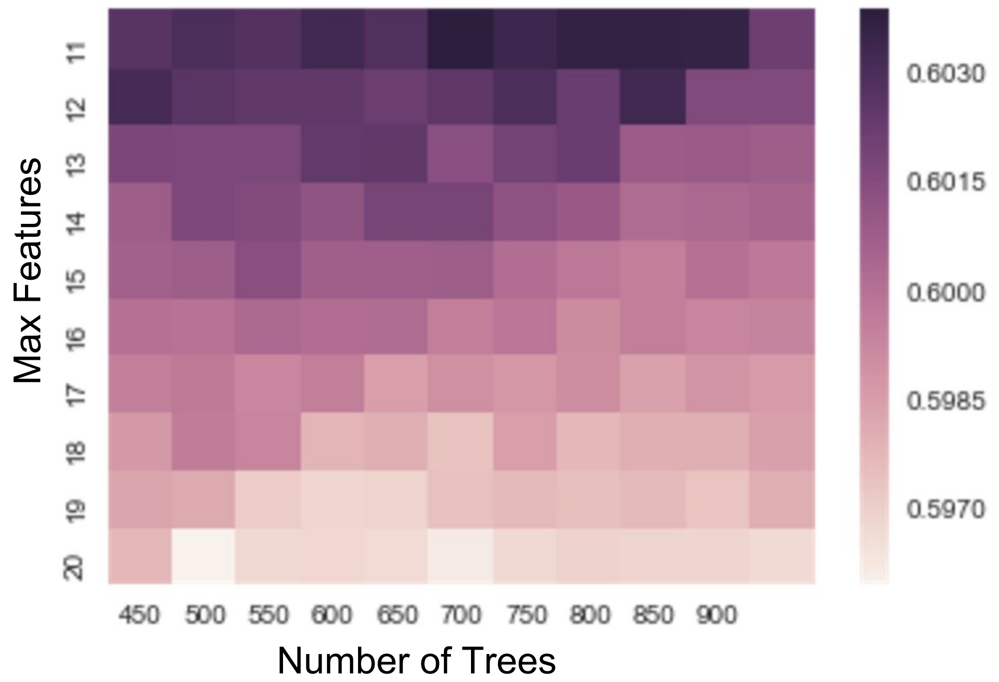
```
-- Grid Parameter Search via 5-fold CV

GridSearchCV took 106.79 seconds for 144 candidate parameter settings.
Model with rank: 1
Mean validation score: 0.537 (std: 0.006)
Parameters: {'min_samples_split': 2, 'max_leaf_nodes': None, 'max_depth': 10,
af': 10}

Model with rank: 2
Mean validation score: 0.537 (std: 0.006)
Parameters: {'min_samples_split': 20, 'max_leaf_nodes': None, 'max_depth': 10, 'min_samples_l
eaf': 10}

Model with rank: 3
Mean validation score: 0.537 (std: 0.006)
Parameters: {'min_samples_split': 10, 'ma
eaf': 10}
```

```python
# set of parameters to test
param_grid = {"min_samples_split": [2, 10, 20],
              "max_depth": [None, 2, 5, 10],
              "min_samples_leaf": [1, 5, 10],
              "max_leaf_nodes": [None, 5, 10, 20],
              }
```

```
Accuracy on training set 0.628
Accuracy on test set 0.568
No longer overfitting of the Decision Tree!!
```

Random Forest

```
Accuracy on training set 0.922
Accuracy on test set 0.581
Also overfitting of the unrestrained Random Forest!!
```



```
Accuracy on training set 0.947
Accuracy on test set 0.626
No longer overfitting of the Random Forest, as much!!
```

8 hrs to run with 3 cores processing!

XGBoost

- Log(Scaled Price / Meter Sq) as Target
- Fast - Training time 1 min 43 sec
- Best Results
- Less Interpretable

```
xgb_params = {
    'eta': 0.01,
    'max_depth': 5,
    'subsample': 0.7,
    'colsample_bytree': 0.3,
    'objective': 'reg:linear',
    'eval_metric': 'rmse',
    'silent': 1,
    'n_jobs': -1
}


xgb.train(xgb_params,
          dtrain,
          num_boost_round=1000,
          evals=[(dval, 'val')],
          early_stopping_rounds=20, verbose_eval=20)
```

14	▲ 6	RunningWolf	0.31193	124	1mo
15	▲ 901	Chase Edge	0.31212	35	2mo
16	▲ 1092	yrtchn	0.31227	125	1mo

Future Work

Next Steps

- Model Ensembling
- Feature Engineering
- Time Series Analysis on Pricing Index
- Cluster Analysis on Neighborhoods

Questions