New Model Output Results:

```
--- Test Set ---
Mean Absolute Error: ... 6.078747834396562
Mean Squared Error:..... 67.18
RMSE: .................. 8.196381335337657
Coeff of det (R^2):..... 0.809 (1.4 % better)

--- Val Set ---
Mean Absolute Error: ... 4.498476271962494
Mean Squared Error:..... 38.02
RMSE: .................. 6.166026942280251
Coeff of det (R^2):..... 0.904 (0.7 % better)

--- Train Set ---
Mean Absolute Error: ... 2.339212417385334
Mean Squared Error:..... 10.24
RMSE: .................. 3.200258043072929
Coeff of det (R^2):..... 0.965
```

Current Production Model R^2:

```
   Test:       0.795
   Validation: 0.897
   Train:      0.962
```

New Model Hyperparameters (XGBoost-based)

```
    max_depth=7
    min_child_weight=6
    gamma = 10
    subsample=0.75
    colsample_bytree = 0.5
    reg_alpha = 100
    reg_lambda = 1
    n_estimators=750  (can add more if desired)
    learning_rate=0.16
    seed=42
    tree_method='hist
```

Features

+ storey_range_avg
(derived from storey_range_max and
storey_range_min)

Keep Historic

This seemed to be pretty good…

```
--- Test Set ---
Mean Absolute Error: ... 6.078747834396562
Mean Squared Error:..... 67.18
RMSE: .................. 8.196381335337657
Coeff of det (R^2):..... 0.809

--- Val Set ---
Mean Absolute Error: ... 4.498476271962494
Mean Squared Error:..... 38.02
RMSE: .................. 6.166026942280251
Coeff of det (R^2):..... 0.904

--- Train Set ---
Mean Absolute Error: ... 2.339212417385334
Mean Squared Error:..... 10.24
RMSE: .................. 3.200258043072929
Coeff of det (R^2):..... 0.965
```

```
xg = xgb.XGBRegressor(
      objective='reg:squarederror',
      max_depth=7,
      min_child_weight=6,
      gamma = 10,
      subsample=0.75,
      colsample_bytree = 0.5,
      reg_alpha = 100,
      reg_lambda = 1,
      n_estimators=1000,
      learning_rate=0.16,
      nthread=4,
      seed=42,
      tree_method='hist')
```