

## Secondo Modello

In questo notebook sono usati i dataset su cui è stata operata la fase preliminare di **preparazione dei dati**, la **divisione** in base alla regione e la **selezione delle feature**.

È costruito un **metamodello** che fa utilizzo di un dataset le cui feature sono le predizioni dei tre modelli costruiti, rispettivamente un **DecisionTreeRegressor**, un **AdaBoostRegressor** e un **RandomForestRegressor**.

Il metamodello è allenato e allenato con un **LinearRegressor** sulle stesse istanze di train usate per le predizioni.

```
In [2]: # Librerie
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings

from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
## Lettura dei dati ##

warnings.filterwarnings('ignore')
```

## Letture dei dati

```
In [3]: # local file path
dir_name = 'selezione'
region_names = np.array(['A', 'B', 'C'])

fp_Xtrain = []
fp_Xval = []
fp_Xtest = []
fp_Ytrain = []
fp_Yval = []
fp_Ytest = []

for i in range(3):
    fp_Xtrain.append(dir_name + f'/{X_train(region_names[i]).csv}')
    fp_Xval.append(dir_name + f'/{X_val(region_names[i]).csv}')
    fp_Xtest.append(dir_name + f'/{X_test(region_names[i]).csv}')
    fp_Ytrain.append(dir_name + f'/{Y_train(region_names[i]).csv}')
    fp_Yval.append(dir_name + f'/{Y_val(region_names[i]).csv}')
    fp_Ytest.append(dir_name + f'/{Y_test(region_names[i]).csv}')
```

```
In [4]: # Lettura dei dati

X_train = []
X_val = []
X_test = []
Y_train = []
Y_val = []
Y_test = []

for i in range(3):
    X_train.append(pd.read_csv(fp_Xtrain[i], low_memory=False))
    X_val.append(pd.read_csv(fp_Xval[i], low_memory=False))
    X_test.append(pd.read_csv(fp_Xtest[i], low_memory=False))
    Y_train.append(pd.read_csv(fp_Ytrain[i], low_memory=False))
    Y_val.append(pd.read_csv(fp_Yval[i], low_memory=False))
    Y_test.append(pd.read_csv(fp_Ytest[i], low_memory=False))

X_train = np.array(X_train, dtype=object)
X_val = np.array(X_val, dtype=object)
X_test = np.array(X_test, dtype=object)
Y_train = np.array(Y_train, dtype=object)
Y_val = np.array(Y_val, dtype=object)
Y_test = np.array(Y_test, dtype=object)
```

```
In [5]: def dimensionality(y=False):
    for i in range(3):
        print(f'X_train(region_names[i]): {X_train[i].shape}')
        print(f'X_val(region_names[i]): {X_val[i].shape}')
        print(f'X_test(region_names[i]): {X_test[i].shape}')
        if y:
            print(f'Y_train(region_names[i]): {Y_train[i].shape}')
            print(f'Y_val(region_names[i]): {Y_val[i].shape}')
            print(f'Y_test(region_names[i]): {Y_test[i].shape}')
            print()
```

```
In [6]: dimensionality(y=True)

X_train: (26819, 55)
X_val: (9006, 55)
X_test: (9085, 55)
Y_train: (26819, 1)
Y_val: (9006, 1)
Y_test: (9085, 1)

X_trainb: (8119, 32)
X_valb: (2688, 32)
X_testb: (2606, 32)
Y_trainb: (8119, 1)
Y_valb: (2688, 1)
Y_testb: (2606, 1)

X_trainc: (64771, 33)
X_valc: (21808, 33)
X_testc: (21876, 33)
Y_trainc: (64771, 1)
Y_valc: (21808, 1)
Y_testc: (21876, 1)
```

## Modelli precedentemente individuati

### Albero di decisione

```
In [7]: dt_model = np.array([
    DecisionTreeRegressor(max_leaf_nodes = 10),
    DecisionTreeRegressor(max_leaf_nodes = 10),
    DecisionTreeRegressor(max_leaf_nodes = 10)
])
```

### Boosting

```
In [8]: boost_model = np.array([
    AdaBoostRegressor(dt_model[0], n_estimators = 850),
    AdaBoostRegressor(dt_model[1], n_estimators = 700),
    AdaBoostRegressor(dt_model[1], n_estimators = 700)
])
```

### Foresta

```
In [9]: rf_model = np.array([
    RandomForestRegressor(n_estimators = 45, n_jobs = -1),
    RandomForestRegressor(n_estimators = 38, n_jobs = -1),
    RandomForestRegressor(n_estimators = 51, n_jobs = -1)
])
```

```
In [10]: models = np.array([
    dt_model,
    boost_model,
    rf_model
])
```

## Costruzione del Metamodello

Costruzione di un dataset Y cui ciascuna colonna contiene le predizioni dei modelli selezionati per le istanze di train. Le colonne hanno il nome del tipo di modello usato. Sono generati un Y di train a partire dagli X di train e allo stesso modo un Y di test a partire dagli X di test.

```
In [11]: def gen_Y(X, y, models, col_names, index):
    Y = pd.DataFrame()
    for i in range(len(X)):
        models[i][index].fit(X[index], y[index])
        Y[col_names[i]] = models[i][index].predict(X[index])
    return Y
```

```
In [12]: def gen_arr(X, y, models, col_names):
    Y = []
    for i in range(len(X)):
        Y.append(gen_Y(X, y, models, col_names, i))
    Y = np.array(Y, dtype=object)
    return Y
```

```
In [13]: col_names = ['DecisionTree', 'Boosting', 'RandomForest']

In [14]: Y_train = gen_arr(X_train, y_train, models, col_names)
Y_test = gen_arr(X_test, y_test, models, col_names)
```

```
In [15]: Y_train[0].head(20)

Out[15]:
```

|    | DecisionTree | Boosting  | RandomForest |
|----|--------------|-----------|--------------|
| 0  | 0.050485     | 0.266712  | 0.055412     |
| 1  | 0.007182     | 0.113340  | 0.025957     |
| 2  | 0.007182     | 0.056837  | -0.006884    |
| 3  | 0.007182     | 0.032949  | -0.019451    |
| 4  | 0.017906     | 0.095386  | 0.021818     |
| 5  | 0.007182     | 0.091944  | 0.023238     |
| 6  | 0.017906     | 0.201193  | 0.123257     |
| 7  | 0.007182     | 0.182836  | -0.028503    |
| 8  | 0.017906     | 0.182836  | 0.026966     |
| 9  | 0.007182     | 0.071312  | -0.002090    |
| 10 | 0.007182     | 0.188450  | 0.006948     |
| 11 | 0.007182     | 0.023291  | 0.015780     |
| 12 | 0.007182     | 0.201981  | -0.004285    |
| 13 | 0.007182     | 0.198270  | -0.027130    |
| 14 | 0.007182     | 0.103031  | -0.000350    |
| 15 | 0.017906     | 0.061509  | 0.035279     |
| 16 | 0.017906     | -0.023329 | 0.004317     |
| 17 | 0.007182     | 0.058858  | 0.000308     |
| 18 | 0.007182     | 0.023251  | -0.000682    |
| 19 | 0.017906     | 0.112282  | 0.097573     |

```
In [16]: Y_test[0].head(20)

Out[16]:
```

|    | DecisionTree | Boosting  | RandomForest |
|----|--------------|-----------|--------------|
| 0  | 0.007520     | 0.034881  | 0.019503     |
| 1  | 0.040975     | 0.659259  | 0.013394     |
| 2  | 0.007520     | -0.899670 | 0.017364     |
| 3  | 0.007520     | -0.025480 | -0.052040    |
| 4  | 0.040975     | -0.861412 | 0.002068     |
| 5  | 0.040975     | 0.921076  | -0.021154    |
| 6  | 0.007520     | -1.097865 | -0.000320    |
| 7  | 0.007520     | 0.007690  | -0.010995    |
| 8  | -0.011041    | 0.029852  | -0.023108    |
| 9  | 0.007520     | 0.007690  | -0.044424    |
| 10 | 0.011041     | 0.137659  | 0.010438     |
| 11 | 0.007520     | -0.050845 | 0.010550     |
| 12 | 0.011041     | -0.006302 | 0.023889     |
| 13 | 0.007520     | 0.592034  | 0.029549     |
| 14 | 0.007520     | 0.051201  | 0.028119     |
| 15 | 0.007520     | 0.005953  | -0.008425    |
| 16 | 0.011041     | 0.032760  | 0.070837     |
| 17 | 0.007520     | 0.673133  | 0.065244     |
| 18 | 0.011041     | 0.710624  | 0.019638     |
| 19 | 0.007520     | 0.420446  | 0.014377     |

```
In [17]: def dimensionality(Y):
    for i in range(3):
        print(region_names[i])
        print(f'X_train(region_names[i]): {X_train[i].shape}')
        print(f'X_val(region_names[i]): {X_val[i].shape}')
        print(f'Y_train(region_names[i]): {Y_train[i].shape}')
        print()
        print(f'X_test(region_names[i]): {X_test[i].shape}')
        print(f'Y_test(region_names[i]): {Y_test[i].shape}')
        print(f'Y_test(region_names[i]): {Y_test[i].shape}')
        print()
```

```
In [18]: dimensionality(Y)

A
X_traina: (26819, 55)
X_vala: (26819, 3)
Y_traina: (26819, 1)

X_testa: (9085, 55)
Y_testa: (9085, 3)
Y_testa: (9085, 1)

B
X_trainb: (8119, 32)
X_valb: (8119, 3)
Y_trainb: (8119, 1)

X_testb: (2606, 32)
Y_testb: (2606, 3)
Y_testb: (2606, 1)

C
X_trainc: (64771, 33)
X_valc: (64771, 3)
Y_trainc: (64771, 1)

X_testc: (21876, 33)
Y_testc: (21876, 3)
Y_testc: (21876, 1)
```

## Regressione Lineare

Modello di regressione lineare allenato sul nuovo dataset costruito con le predizioni per le **X di train** e le stesse **y di train** su cui sono stati allenati i modelli. Le predizioni sono calcolate sulla base del nuovo dataset costruito con le predizioni per le **X di test**.

```
In [19]: def linear_regression_preds(Y_train, y_train, Y_test):
    preds = []
    for Y_train, Y_test in zip(Y_train, Y_test):
        preds.append(
            LinearRegression(fit_intercept=True).fit(Y_train, y_train).predict(Y_test)
        )
    preds = np.array(preds, dtype = object)
    return preds
```

```
In [20]: y_preds = linear_regression_preds(Y_train, y_train, Y_test)

Sulla base del confronto tra predizioni e valori attesi calcolo l'errore.
```

```
In [21]: def get_mse(y_trues, y_preds):
    mse = []
    for y_true, y_pred in zip(y_trues, y_preds):
        mse.append(mean_squared_error(y_true, y_pred))
    mse = np.array(mse, dtype = object)
    return mse
```

```
In [22]: mse = get_mse(y_test, y_preds)
```

```
In [23]: for i in range(3):
    print(region_names[i])
    print(mse[i])
    print()

A
0.0016997970728566968

B
0.0021310291641643032

C
0.0022809873002168383
```

Confronto con l'errore del modello banale

```
In [24]: def mse_trivial(y_train, y):
    y_pred = np.repeat(y_train.mean(), len(y_train))
    return (y_pred - y)**2).mean()
```

```
In [25]: for i in range(3):
    print(region_names[i])
    print(mse_trivial(y_test[i].values.ravel(), y_preds[i]))
    print()

A
0.02757368566548814

B
0.019152250649440997

C
0.02754620685288015
```

## Analisi dei risultati

Rispetto al modello banale gli errori del metamodello sono circa dieci volte più precise: seppur il modello non sia molto preciso, combinare le informazioni di più modelli sembra essere risultato vantaggioso.

## Migliori Predizioni

```
In [46]: def best_prediction(X, y_trues, y_preds, index, nrow=10):
    err = abs(y_trues[index] - y_preds[index])
    err = err.values.ravel()
    args = err.argsort()
    for i in range(nrow):
        print(
            f'Row : {i}, logerror : {2.7*logerror: 2.7f} % {'
            f' {y_trues[index].iloc[args[i]]}'+'logerror',
            err[args[i]]
        )
    return X[index].iloc[args[nrow]].transpose()
```

## Prima Regione 1286

```
In [47]: best_prediction(X_test, y_test, y_preds, 0, nrow=5)

Row : 4204, logerror : 0.0469000 error: 0.0000018
Row : 7126, logerror : -0.0182000 error: 0.0000019
Row : 7059, logerror : -0.0050000 error: 0.0000040
Row : 7353, logerror : 0.0198000 error: 0.0000014
Row : 4263, logerror : 0.0046090 error: 0.0000059
```

|                                | 4204         | 7126         | 7055         | 7353         | 4263         |
|--------------------------------|--------------|--------------|--------------|--------------|--------------|
| bathtubroomt                   | 2.000000e+00 | 2.000000e+00 | 3.000000e+00 | 2.500000e+00 | 1.000000e+00 |
| bedroomt                       | 2.000000e+00 | 3.000000e+00 | 4.000000e+00 | 3.000000e+00 | 1.000000e+00 |
| buildingqualitytypeid          | 7.000000e+00 | 7.000000e+00 | 7.000000e+00 | 7.000000e+00 | 7.000000e+00 |
| calculatedbathnbr              | 2.000000e+00 | 2.000000e+00 | 3.000000e+00 | 2.500000e+00 | 1.000000e+00 |
| calculatedfinishesquarefeet    | 1.274000e+03 | 1.264000e+03 | 2.127000e+03 | 1.345000e+03 | 8.820000e+02 |
| finishedsquarefeet12           | 1.264000e+03 | 1.264000e+03 | 2.127000e+03 | 1.345000e+03 | 8.820000e+02 |
| fireplaceact                   | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| latitude                       | 3.366740e+07 | 3.372438e+07 | 3.361572e+07 | 3.358339e+07 | 3.377900e+07 |
| longitude                      | 1.178505e+08 | 1.180366e+08 | 1.176718e+08 | 1.177196e+08 | 1.179970e+08 |
| rawcensustractandblock         | 7.259000e+03 | 6.718800e+03 | 8.710000e+03 | 2.550000e+03 | 7.259000e+03 |
| loweansstructureblock          | 6.059063e+07 | 6.059100e+07 | 6.059032e+07 | 6.059063e+07 | 6.059088e+07 |
| regionidzip                    | 5.265000e+04 | 2.521800e+04 | 1.227300e+03 | 3.246380e+04 | 4.296700e+04 |
| roomcnt                        | 0.000000e+00 | 6.000000e+00 | 5.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| unitcnt                        | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| yearbuilt                      | 2.006000e+03 | 1.960000e+03 | 1.968000e+03 | 1.990000e+03 | 1.983000e+03 |
| structuraevaluatedollart       | 2.500940e+05 | 5.604800e+04 | 1.442240e+05 | 1.265110e+05 | 5.993700e+04 |
| taxvaluedollartot              | 4.617750e+05 | 4.641480e+05 | 3.221080e+05 | 3.187550e+05 | 2.017750e+05 |
| landtaxvaluedollartot          | 2.116810e+05 | 4.081000e+05 | 1.778400e+05 | 1.922440e+05 | 1.472380e+05 |
| taxamount                      | 5.697100e+03 | 5.336500e+03 | 3.316140e+03 | 5.697400e+04 | 9.620300e+04 |
| int_transactiondate            | 2.630000e+02 | 2.030000e+02 | 2.630000e+02 | 1.810000e+02 | 8.600000e+01 |
| period_mean_price              | 1.852407e-02 | 8.281493e-03 | 1.852407e-02 | 8.717589e-03 | 8.947002e-03 |
| neighborhood_mean_price        | 1.951526e-02 | 1.168289e-02 | 1.227379e-02 | 1.051390e-02 | 1.159592e-02 |
| living_area_prop               | 2.185004e-01 | 2.096186e-01 | 2.244021e-01 | 3.708600e-01 | 2.185004e-01 |
| tax_ratio                      | 8.105440e+01 | 8.697513e+01 | 9.713410e+01 | 9.818783e+01 | 8.742508e+01 |
| tax_prop                       | 1.181466e+00 | 1.373389e+01 | 8.107756e+01 | 6.580752e+01 | 4.077956e+01 |
| buildingqualitytypeid_na_flag  | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| unitcnt_na_flag                | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| assessmentyear_2015.0          | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.1.0      | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.6.0      | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.7.0      | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.10.0     | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.12.0     | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.13.0     | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.18.0     | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid.24.0     | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| poolcnt.1.0                    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode.0100 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode.122  | 0.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| propertycountylandusecode.34   | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.000000e+00 | 0.000000e+00 |
| propertycountylandusecode.rare | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.247.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.247.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.248.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.260.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.263.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.264.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.265.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.266.0    | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| propertylandusetypeid.267.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypeid.269.0    | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |

Condizionazioni:

- Il valore assoluto del logerror è relativamente piccolo.
- La grandezza di 5 delle 5 case è simile come indicano bedroomcnt, bathtubroomt, e lotsizesquarefeet. Una casa ha un numero contenuto di stanze e dunque è semplice decisamente più piccola rispetto alle altre.
- Tutte le case hanno lo stesso buildingqualitytypeid pari a 7.
- Tutte le case sono di città e regioni differenti.
- Il periodo\_mean\_price è molto variabile, mentre la variabilità del neighborhood\_mean\_price è contenuta.
- Tutte e cinque le case sono sprovviste di un impianto antincendio (fireplaceact = 0) e sono sprovviste di piscina (poolcnt = 0).
- Quattro case su cinque sono state assegnate nel 2015 (flag assessmentyear\_2015 = accessio).
- Tutte e cinque le case hanno il missing flag per buildingqualitytypeid e unitcnt presenti.
- Per tutte e cinque le case il flag heatingorsystemtypeid sono sempre presenti (valori non assenti nel train oppure altre feature eliminate durante la selezione).

## Seconda Regione 2061

```
In [48]: best_prediction(X_test, y_test, y_preds, 1, nrow=5)

Row : 1698, logerror : 0.0409842 error: 0.0000032
Row : 773, logerror : 0.1458000 error: 0.0000075
Row : 989, logerror : -0.0051408 error: 0.0000037
Row : 778, logerror : 0.0415100 error: 0.0000012
Row : 7794, logerror : -0.0243876 error: 0.0000058
```

|                             |              |              |              |              |              |
|-----------------------------|--------------|--------------|--------------|--------------|--------------|
| propertylandusetypepaid_240 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_246 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_247 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_248 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_260 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_261 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| propertylandusetypepaid_263 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_264 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_265 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_266 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_267 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusetypepaid_269 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |

Considerazioni:

- il valore assoluto dei `logerror` è positivo e piccolo per tre predizioni, negativo e piccolo per le altre due.
- ci sono due case il cui `bedrooment` e `bathrooment` è pari a 0, il che è altrettanto improbabile. Può essere sintomo di un dataset mal compilato o di un tipo di informazione mancante non intesa come Nan.
- `newcase` ha un importo anticipato (il `Fineplacement` pari a 0).
- fatta eccezione per una casa venduta l'8 gennaio, `int_transactiondate` suggerisce che le altre quattro siano state vendute nei mesi finali dell'anno.
- una casa ha una piscina (bit di `poolcnt` acceso).
- tutte le case sono di città e regioni differenti.
- tutte e cinque le case hanno il missing flag per `buildingsqtypeid` e `unitcnt` acceso.
- per nessuna casa ho informazione su `heatingorsystemtypeid` (non presente in train o eliminato dalla feature selection)



Row : 1085, logerror : -0.038600 error: 0.624503  
Row : 166, logerror : 2.2300000 error: 0.5950939  
Row : 1393, logerror : 2.1973627 error: 0.5317995  
Row : 1845, logerror : -2.238122 error: 0.4723513  
Row : 2238, logerror : 1.3631378 error: 0.4209428

|          | 1085                         | 56           | 1393         | 1845         | 2238         |
|----------|------------------------------|--------------|--------------|--------------|--------------|
| Out[56]: |                              |              |              |              |              |
|          | bathroomcnt                  |              |              |              |              |
|          | 2.000000e+00                 | 2.000000e+00 | 2.000000e+00 | 1.000000e+00 | 2.500000e+00 |
|          | bedroomcnt                   |              |              |              |              |
|          | 3.000000e+00                 | 3.000000e+00 | 5.000000e+00 | 3.000000e+00 | 2.000000e+00 |
|          | calculatedbathnbr            |              |              |              |              |
|          | 2.000000e+00                 | 2.000000e+00 | 2.000000e+00 | 1.000000e+00 | 2.500000e+00 |
|          | calculatedfinishedsquarefeet |              |              |              |              |
|          | 1.160000e+03                 | 1.696000e+03 | 1.860000e+03 | 1.759000e+03 | 1.240000e+03 |
|          | finishedsquarefeet12         |              |              |              |              |
|          | 1.160000e+03                 | 1.696000e+03 | 1.860000e+03 | 1.759000e+03 | 1.240000e+03 |
|          | fireplacecnt                 |              |              |              |              |
|          | 0.000000e+00                 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
|          | latitude                     |              |              |              |              |
|          | 3.425852e+07                 | 3.426906e+07 | 3.422859e+07 | 3.418204e+07 | 3.417732e+07 |
|          | longitude                    |              |              |              |              |
|          | 1.187694e+08                 | 1.188007e+08 | 1.191919e+08 | 1.191742e+08 | 1.187859e+08 |
|          | lotsizesquarefeet            |              |              |              |              |
|          | 7.000000e+03                 | 8.257000e+03 | 6.534000e+03 | 6.000000e+03 | 7.205000e+03 |
|          | rawcensustractandblock       |              |              |              |              |
|          | 6.111000e+07                 | 6.111000e+07 | 6.111000e+07 | 6.111000e+07 | 6.111000e+07 |
|          | regionidzip                  |              |              |              |              |
|          | 9.711000e+04                 | 9.711800e+04 | 9.710700e+04 | 9.710400e+04 | 9.639300e+04 |
|          | roomcnt                      |              |              |              |              |
|          | 5.000000e+00                 | 6.000000e+00 | 9.000000e+00 | 7.000000e+00 | 5.000000e+00 |
|          | unitcnt                      |              |              |              |              |
|          | 1.000000e+00                 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
|          | yearbuilt                    |              |              |              |              |
|          | 1.963000e+03                 | 1.976000e+03 | 1.977000e+03 | 1.951000e+03 | 1.990000e+03 |
|          | structuraevaluatedollarcnt   |              |              |              |              |
|          | 2.016460e+05                 | 1.871250e+05 | 2.430000e+05 | 1.227120e+05 | 2.537050e+05 |
|          | taxvaluedollarcnt            |              |              |              |              |
|          | 4.032920e+05                 | 4.775830e+05 | 4.680000e+05 | 2.197130e+05 | 4.224960e+05 |
|          | landtaxvaluedollarcnt        |              |              |              |              |
|          | 2.016460e+05                 | 2.904580e+05 | 2.450000e+05 | 9.700100e+04 | 1.687910e+05 |
|          | taxamount                    |              |              |              |              |
|          | 7.000000e+01                 | 7.000000e+01 | 5.589700e+03 | 2.643160e+03 | 5.680340e+03 |
|          | int_transactiondate          |              |              |              |              |
|          | 7.000000e+01                 | 7.000000e+01 | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 |
|          | period_mean_price            |              |              |              |              |
|          | 4.897452e+03                 | 4.897452e+03 | 1.617989e+02 | 1.080729e+02 | 1.993256e+02 |
|          | neighborhood_mean_price      |              |              |              |              |
|          | 1.740458e+02                 | 1.063207e+02 | 8.821141e+03 | 6.311392e+03 | 1.281764e+02 |
|          | living_area_prop             |              |              |              |              |
|          | 1.657143e+01                 | 2.054015e+01 | 2.846648e+01 | 2.931667e+01 | 2.185004e+01 |
|          | tax_ratio                    |              |              |              |              |
|          | 1.6574789e+01                | 8.735039e+01 | 8.730343e+01 | 8.312512e+01 | 7.437865e+01 |
|          | tax_prop                     |              |              |              |              |
|          | 1.000000e+00                 | 6.442412e-01 | 9.918367e-01 | 1.265059e+00 | 1.503072e+00 |
|          | assessmentsyear_2015.0       |              |              |              |              |
|          | 1.000000e+00                 | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
|          | poolcnt.1.0                  |              |              |              |              |
|          | 0.000000e+00                 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
|          | propertylandusetypeid_246.0  |              |              |              |              |
|          | 0.000000e+00                 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
|          | propertylandusetypeid_261.0  |              |              |              |              |
|          | 1.000000e+00                 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 0.000000e+00 |
|          | propertylandusetypeid_265.0  |              |              |              |              |
|          | 0.000000e+00                 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
|          | propertylandusetypeid_266.0  |              |              |              |              |
|          | 0.000000e+00                 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.000000e+00 |
|          | propertylandusetypeid_275.0  |              |              |              |              |
|          | 0.000000e+00                 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |

Condiderazioni:

- per tre istante il valore assoluto del `logerror`: è molto grande.
- il numero di stanze è molto simile: `bedroomcnt`: è pari a 2 o 3 (una casa ha ben 5 camere da letto), `bathroomcnt`: a 1 o 2, infatti `lotsizesquarefeet` è simile.
- quattro case su cinque hanno un impianto antiscandico (`Fireplacecnt`).
- ci sono due coppie di case della stessa città.
- quattro case su cinque sono state costruite tra gli anni '60 e '70.
- due case sono state vendute il 7 gennaio.
- tutte e cinque le case sono sprovviste di piscina (`poolcnt` = 0).
- quattro delle cinque case hanno `propertylandusetypeid` pari a 261.

### Terza Regione 3101

|                                |              |              |              |              |              |
|--------------------------------|--------------|--------------|--------------|--------------|--------------|
| heatingorsystemtypeid_2.0      | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.000000e+00 | 0.000000e+00 |
| heatingorsystemtypeid_7.0      | 0.000000e+00 | 1.000000e+00 | 1.000000e+00 | 0.000000e+00 | 1.000000e+00 |
| propertycountylandusecode_0100 | 0.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |
| propertycountylandusecode_0101 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0102 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0103 | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0104 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0105 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0106 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0107 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0108 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0109 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0110 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0111 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0112 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0113 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0114 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0115 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertycountylandusecode_0116 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertylandusecode_246.0      | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertyzoningdesc_LAR1        | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| propertyzoningdesc_rare        | 0.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |

Considerazioni:

- il valore assoluto del `logerror` è per tre case molto alto.
- due case sono decisamente molto grandi: `bathroomcnt` e `bedroomcnt` indicano che una casa ha 7 bagni e 11 camere da letto, l'altra 6 bagni e 12 camere. Anche `calculatedfinishedsquarefeet` è di conseguenza molto grande.
- `calculatedfinishedsquarefeet` e `finishedsquarefeet12` per tutte e cinque le case hanno lo stesso valore.
- la prima casa è stata costruita nel '50, la seconda addirittura nel 1909.
- due coppie di case sono della stessa città.
- due case hanno `heatingorsystemtypeid` pari a 2, le altre tre a 7.