Housing Sales Price Prediction in North Dallas 2021

Introduction

The housing market in North Dallas has been growing rapidly over the last few years and even more so in the last 6 months. Several large corporations are relocating their headquarters to the area and moving large numbers of employees with them which is driving the market upward.

The goal of the project is to solve the problem of optimal housing pricing by using machine learning to estimate a sale price of homes in a small section of North Dallas suburbs. This project will use a small subset of the available data as a case study to prove effectiveness of the methodology. If the models show that a home could be sold higher this results in more money for the home sellers, a larger fee for the realty company, and a larger commission for the realtor. If the model shows that homes are overpriced then it can be suggested to lower the price if it is needed to move the home quickly.

The proprietary dataset was provided by Sandra Smith from Data Sample Realty in order to carry out the analysis

Data Wrangling

The approach for data wrangling was to make each remove all unnecessary features and investigate and fix all null values. After data wrangling, every row for each feature should have a relevant value and each feature should have the correct Dtype for analysis. The list of features and data types in the raw data can be seen below.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9117 entries, 0 to 2613
Data columns (total 140 columns):
 #  Column                    Non-Null Count  Dtype
---  ------                   --------------  -----
 0  APN/PIN/Tax 1             8527 non-null   object
 1  APN/PIN/Tax 2             8527 non-null   object
 2  APN/PIN/Tax 3             3774 non-null   object
 3  FIPS                      9117 non-null   int64
 4  Census Tract              9006 non-null   float64
 5  Property Address          9103 non-null   object
 6  House Number              9069 non-null   object
 7  Pre Direction             198 non-null    object
 8  Street                    9103 non-null   object
 9  Street Suffix             8968 non-null   object
 10 Post Direction            9 non-null      object
 11 Unit Type                 450 non-null    object
 12 Unit Number               450 non-null    object
 13 City                      9103 non-null   object
 14 State                     9117 non-null   object
 15 ZIP 5                     9094 non-null   float64
 16 ZIP 4                     9028 non-null   float64
 17 Mailing Address           9101 non-null   object
 18 Mailing House Number      8925 non-null   object
 19 Mailing Pre Direction     384 non-null    object
 20 Mailing Street            9101 non-null   object
 21 Mailing Street Suffix     8724 non-null   object
 22 Mailing Post Direction    57 non-null     object
 23 Mailing Unit Type         899 non-null    object
 24 Mailing Unit Number       889 non-null    object
 25 Mailing City              9085 non-null   object
 26 Mailing State             9074 non-null   object
 27 Mailing ZIP 5             9073 non-null   object
 28 Mailing ZIP 4             8944 non-null   float64
 29 MLS ID                    9117 non-null   int64
 30 Listing Type              9117 non-null   object
 31 Status                    9117 non-null   object
 32 List Price                9117 non-null   int64
 33 Days on Market            9115 non-null   float64
 34 Contract Status Change Date 9117 non-null object
 35 Township Name             9114 non-null   object
 36 Subdivision Code          1115 non-null   object
 37 Subdivision               9114 non-null   object
 38 Current Year Tax          9117 non-null   int64
 39 Tax Amount                9110 non-null   float64
 40 Tax Rate Code Area        5444 non-null   object
 41 Current Year Assessment   9117 non-null   float64
 42 Total Assessed Value      9117 non-null   int64
 43 Assessed Land             9117 non-null   int64
 44 Assessed Improvement      9117 non-null   int64
 45 Total Market Value        9117 non-null   int64
 46 Market Value Land         9117 non-null   int64
 47 Market Value Improvement  9117 non-null   int64
 48 Estimated Value           8819 non-null   float64
 49 Absentee Status           9117 non-null   object
 50 Exemption-Veterans        9117 non-null   bool
 51 Exemption-Disabled        9117 non-null   bool
 52 Exemption-Senior          9117 non-null   bool
 53 Exemption-Widow           9117 non-null   bool
 54 Exemption-Homestead       9117 non-null   bool
 55 Block #                   8816 non-null   object
 56 Lot #                     9008 non-null   object
 57 Section #                 0 non-null      float64
 58 County Use Code           9113 non-null   object
 59 Universal Land Use        9117 non-null   object
 60 State Land Use Code       0 non-null      float64
 61 Zoning                    2318 non-null   object
 62 Plat Map Reference        3730 non-null   object
 63 Lot Acres                 9117 non-null   float64
 64 Lot SQFT                  9117 non-null   float64
 65 New Construction          8847 non-null   object
 66 Beds                      9013 non-null   float64
 67 Baths                     9117 non-null   float64
 68 Total Building Area       9117 non-null   int64
 69 Living Area SQFT          9041 non-null   float64
 70 Gross Area SQFT           355 non-null    float64
 71 Basement SQFT             7 non-null      float64
 72 Garage SQFT               8301 non-null   float64
 73 Basement                  1551 non-null   object
 74 Flooring Cover            2968 non-null   object
 75 Stories                   8915 non-null   object
 76 Style                     1086 non-null   object
 77 Year Built                8847 non-null   float64
 78 Air conditioning          8311 non-null   object
 79 Heat Type                 8174 non-null   object
 80 Fireplace indicator       9117 non-null   bool
 81 Construction Type         4996 non-null   object
 82 Exterior Wall             8312 non-null   object
 83 Roof Material Type        6676 non-null   object
 84 Porch Type                5825 non-null   object
 85 Has Pool                  1631 non-null   object
 86 Parking spaces            9117 non-null   int64
 87 Parking Type              8436 non-null   object
 88 Total Units               9117 non-null   int64
 89 Sell Score                0 non-null      float64
 90 Distressed Indicator      1 non-null      object
 91 Distressed Recording Date 191 non-null    object
 92 Distressed Case Number    3 non-null      object
 93 Auction Date              84 non-null     object
 94 Default Date              0 non-null      float64
 95 Recording Date            8945 non-null   object
 96 Document Type             8789 non-null   object
 97 Sales Price               6153 non-null   float64
 98 Equity                    8819 non-null   float64
 99 Equity Percentage         8956 non-null   object
 100 Number Of Mortgages      9117 non-null   int64
 101 Mortgage Loan Balance    9117 non-null   float64
 102 Mortgage 1 Lender Name   6647 non-null   object
 103 Mortgage 1 Loan Type     6647 non-null   object
 104 Mortgage 1 Amount        6647 non-null   float64
 105 Mortgage 1 Loan Date     6635 non-null   object
 106 Mortgage 1 Rate          6647 non-null   object
 107 Estimated Rate           6647 non-null   object
 108 Estimated Rate.1         6647 non-null   object
 109 Mortgage 1 Age           6635 non-null   object
 110 Mortgage 2 Lender Name   260 non-null    object
 111 Mortgage 2 Loan Type     260 non-null    object
 112 Mortgage 2 Amount        231 non-null    float64
 113 Mortgage 2 Loan Date     259 non-null    object
 114 Mortgage 2 Rate          260 non-null    object
 115 Mortgage 2 Age           259 non-null    object
 116 Mortgage 3 Lender Name   3 non-null      object
 117 Mortgage 3 Loan Type     3 non-null      object
 118 Mortgage 3 Amount        1 non-null      float64
 119 Mortgage 3 Loan Date     3 non-null      object
 120 Mortgage 3 Rate          3 non-null      object
 121 Mortgage 3 Age           3 non-null      object
 122 Mortgage 4 Lender Name   2 non-null      object
 123 Mortgage 4 Loan Type     2 non-null      object
 124 Mortgage 4 Amount        0 non-null      float64
 125 Mortgage 4 Loan Date     2 non-null      object
 126 Mortgage 4 Rate          2 non-null      object
 127 Mortgage 4 Age           2 non-null      object
 128 Owner 1 First Name       8122 non-null   object
 129 Owner 1 Middle Name      4869 non-null   object
 130 Owner 1 Last Name        8123 non-null   object
 131 Owner 1 Full Name        9117 non-null   object
 132 Owner 1 Email Addresses  2578 non-null   object
 133 Owner 1 Phone Numbers    2349 non-null   object
 134 Owner 2 First Name       4809 non-null   object
 135 Owner 2 Middle Name      2683 non-null   object
 136 Owner 2 Last Name        4809 non-null   object
 137 Owner 2 Full Name        4953 non-null   object
 138 Owner 2 Email Addresses  690 non-null    object
 139 Owner 2 Phone Numbers    635 non-null    object
dtypes: bool(6), float64(28), int64(14), object(92)
```

Features From Raw Input Data

There is a lot of personal information in the raw dataset so the first task is to remove features such as names, phone numbers, and email addresses. After the personal data was scrubbed, the next step was to look at certain types of real estate to concentrate on family sized homes not commercial properties.

```
cleaning = cleaning[cleaning['Universal Land Use'] != 'Condominium Unit (Residential)']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Duplex (2 units - any combination)']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Multi-Family Dwellings (Generic - 2+)']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Apartment house (5+ units)']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Commercial Building']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Vacant Land (General)']
```

```
cleaning = cleaning[cleaning['Listing Type'] != 'lease']
```

```
cleaning = cleaning[cleaning['Universal Land Use'] != 'Commercial (General)']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Commercial-Vacant Land']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Condominium Offices']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Neighborhood Shopping Center - Strip Center/Mall - Enterprise Zone']
cleaning = cleaning[cleaning['Universal Land Use'] != 'Under Construction']
```

Code Snippet Showing Removal of Commercial Properties

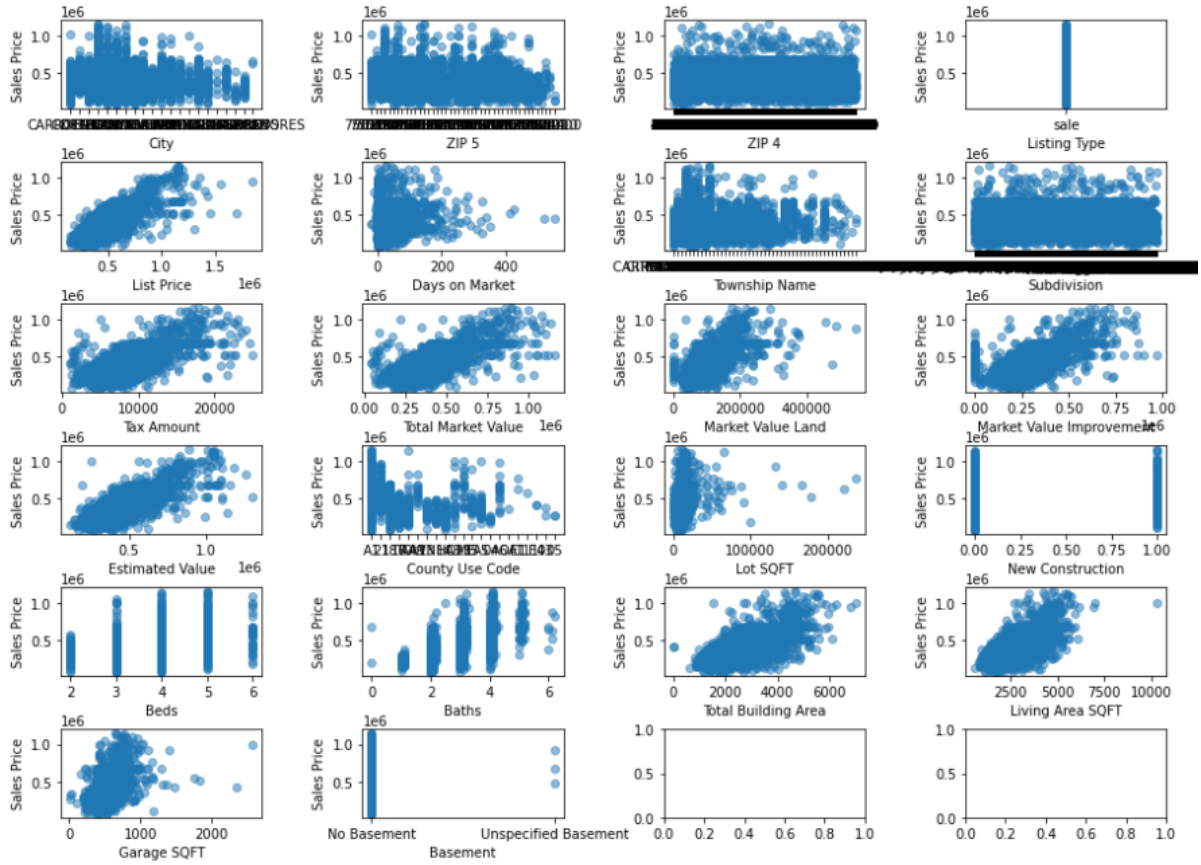
The last portion of data wrangling was to fill null values for the categorical data. I assumed that unless specified there was not a pool. The vast majority of roof types was composition shingles so I used that to fill nulls for roof type.

```
cleaning['Has Pool'].fillna('None', inplace = True)
cleaning['Basement'].fillna('No Basement', inplace = True)
cleaning['Porch Type'].fillna('None', inplace = True)
cleaning['Heat Type'].fillna('Yes', inplace = True)
cleaning['Roof Material Type'].fillna('Composition Shingle', inplace = True)
cleaning['Air conditioning'].fillna('Yes', inplace = True)
cleaning = cleaning.drop(['Basement SQFT', 'Total Assessed Value', 'Assessed Land', 'Assessed Improvement', 'State', 'Status', '
```

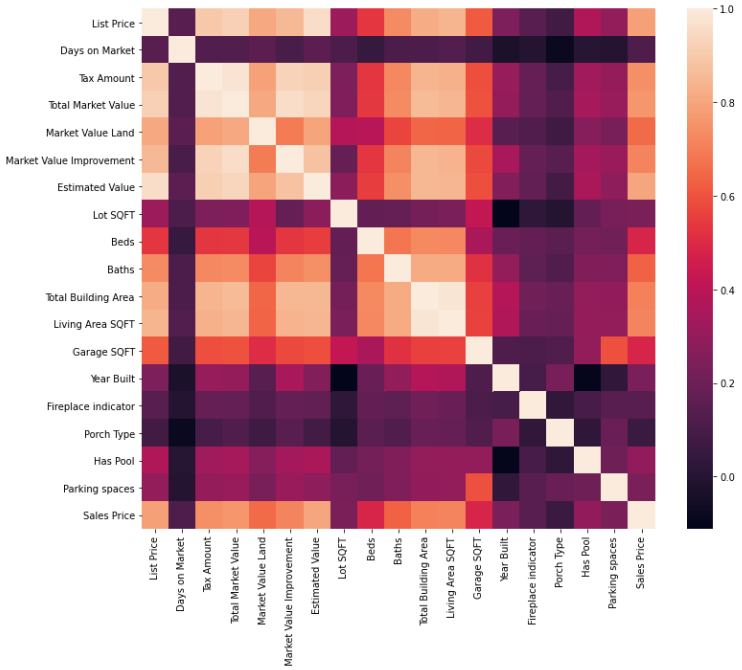
Code Snippet Showing Filling Null Categorical Data

Exploratory Data Analysis

For exploratory data analysis I did preliminary plotting to see the relationships between the features and the two dataframes. This is very easily done with pandas profiling to see histograms of each feature as well to see if you have any remaining null or extreme values..



Seeing How Features Correlate vs Sales Price



Correlation Heatmap of Features

Preprocessing

The first step I took in preprocessing was to one hot encode the categorical features. This made the number of features in the dataset go from 29 to 5893. This seemed extreme until I looked further into the categories. There were very few properties that had duplicate 4 digit zip codes so there were over 3400 new features created for these. There were so many they couldn't reasonably be correlated to sales price so I removed them.

Before splitting into training and test sets I created several different sets of X datasets based on varying types of features. I made one with only home features, one with a smaller set of home features, and one with only location data to compare predictions between different feature sets. Later down the road I found that the estimated price was the most important feature so I circled back and created new features of 90% and 110% of this value to see if it would predict better.

Once all of these sub datasets were created I split them each up into train/test datasets from modeling.

Modeling

When I got to the modeling phase I planned to test a wide variety of model types in order to compare results. Below is a summary table of the models and their results.

Model	Train MAE ($)	Test MAE ($)

Support Vector Regression	116,322	109,472
Random Forest Regression	67,369	74,960
Lasso Regression	64,532	66,370
Ridge Regression	64,574	66,138
XGBoost Regression	60,930	64,929

It is plain to see the XGBoost model had the lowest error and would be the recommended model to use for production. All of the models seem to generalize well to the testing data and did not overfit on the training data. The code for the hyperparameter tuning for the XGBoost model can be seen below as well as the top 5 most important features from the model.

```
start = time.time()

steps = [('scaler', StandardScaler()), ('xgbr', XGBRegressor())]
pipe = Pipeline(steps)

n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
max_depth = [1,2,3,4,5,6,7,8,9,10]
eta = [.001,.005,.01,.025,.05,.1,.2,.3]
subsample = [.25,.5,.75,1]
colsample_bytree = [.25,.5,.75,1]

random_grid = {'xgbr__n_estimators': n_estimators,
               'xgbr__max_depth': max_depth,
               'xgbr__eta': eta,
               'xgbr__subsample': subsample,
               'xgbr__colsample_bytree': colsample_bytree}

xgboost = RandomizedSearchCV(
        estimator=pipe,
        param_distributions = random_grid, n_iter = 100, cv = 5, verbose=2, random_state=42, n_jobs = -1)

grid_result = xgboost.fit(X_big_train, y_big_train)
best_params = xgboost.best_params_
print(best_params)

y_pred_xgboost = xgboost.predict(X_big_test)
y_tr_pred_xgboost = xgboost.predict(X_big_train)



print('It takes %s minutes' % ((time.time() - start)/60))
median_mae_xgboost = mean_absolute_error(y_big_train, y_tr_pred_xgboost), mean_absolute_error(y_big_test, y_pred_xgboost)
median_mae_xgboost

print(median_mae_xgboost)


r2_score(y_big_train, y_tr_pred_xgboost), r2_score(y_big_test, y_pred_xgboost)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
{'xgbr__subsample': 0.5, 'xgbr__n_estimators': 1800, 'xgbr__max_depth': 2, 'xgbr__eta': 0.005, 'xgbr__colsample_bytree': 0.75}
It takes 75.25825051069259 minutes
(60930.19888094893, 64929.436901653615)
```

Code Snippet of Best Performing Model Tuning - XGBoost

Feature	Importance
Estimated Value	0.181957
Total Market Value	0.107437
List Price	0.079475
Tax Amount	0.029974
Market Value Improvement	0.023867

Most Important Features according to XGBoost