**Can Machine Learning   
Predict the Sale Price   
of a House?**

**By Michel Maniguet**

**Project Overview**

An ever-increasing number of problems are being solved using Machine Learning (ML) solutions. This paper will describe an attempt to apply ML to a problem, faced by real estate professionals on a regular if not daily basis: “What is the fair market value of a particular property?” This question will be referred to as “The Primary Question”, or TPQ in this report.

Realtors working as agents for buyers and sellers must ask this question, but for different reasons.

When representing buyers, it important for the agent to establish what the fair market value (FMV) of a given property is before placing an offer on behalf of a client, so that the client will not overpay for the property. Similarly, if the offer made is too low, the client is at risk of losing the property because the seller will reject the offer, or, in a multiple offer situation (a common occurrence in the Dallas-Fort Worth market at the current time), lose the home to another buyer. When representing sellers, the agent must advise the seller as to the FMV of the property. A wrong valuation risks the property not selling in a reasonable amount of time if it is overpriced, or the seller suffering financial loss if the property is sold at under FMV.

In both cases, the correct answer to TPQ ultimately results in dollars in a client’s pocket. The more dollars in the clients pocket after closing, the more legitimate the commission earned by the agent, and the more likely repeat business, as well as referrals will ensue through that client. Such is the case for desiring and answer for the question, “Can Machine Learning be used to predict the sale price of a home?” If so, how accurate is it? In order to be useful for realtors, the accuracy of any ML based pricing predictor must be very close to 5%, an acceptable range of accuracy reflected in Sales Price to List Price Ratio data available from the MLS. Finally, if the prediction is not adequately accurate, what can be done to improve the accuracy? These issues form the basis and foundation for this project.

**Project Roadmap**

The hypothesis driving this project is that ML can be used to predict market prices, and that the accuracy of those predictions can be made to fall within an acceptable accuracy range if the data and ML model are adequate. An overview of the strategy used in this project to address the hypothesis and thus to answer TPQ is as follows:

1. Extract historical real estate market data from a reliable source
2. Transform that data into a form usable for training an ML engine how to predict market price
3. Apply the ML model toward giving an estimate on a specific property

**Technology Stack**

A list follows of the primary technologies employed in the course of this project, with a brief description of their use or purpose:

* Python – general purpose processing
* Tabula – extracting data PDF (installed Java run time libraries required)
* Pandas & Numpy – Python libraries for data frame manipulation, large arrays & extended math
* Matplotlib with pyplot – Python plotting library
* Sci Kit Learn – Python Machine Learning library
* Joblib – library to support storage to and retrieval from disk of ML models

**Data Extraction**

It was initially intended to obtain primary source data by extraction via scraping from the North Texas Real Estate Information System (NTREIS), which is the DFW area Multiple Listing Service (MLS) data service. However, the timeframe available to complete the project was too short for this option. Therefore, to save time, the same data was extracted manually by downloading PDF files from the NTREIS system. The contents of the data files were obtained by selecting search parameters of Sold Date, within the previous 30 days, and the address of the sold property being within Denton County, TX. Below is a partial representation of the PDF file:

A screenshot of a cell phone

Description automatically generated

**Data Transformation**

The essential data needed from the PDF data above, being in a format designed to support cross platform document portability and portability, was not readily accessible using standard Python. Each line of data consists of information describing a unique home, also known as a “listing”, which sold within the last 30 days in Denton County, TX. Each column heading description the significance of the data beneath it.

All the data from each line must be extracted from the PDF. Tabula was used to facilitate this extraction, with the output of Tabula’s “read PDF” function being placed into a Pandas data frame. However, the structure of the data in this data frame as extracted required significant further processing before it could be usable for the next step of data analysis. The further processing primarily involved numerous splits and joins to parse the data from the original Tabula output data frame into a new data frame that was almost identical in structure and appearance to the PDF file pictured above. Below is a sample image of the result of this step:

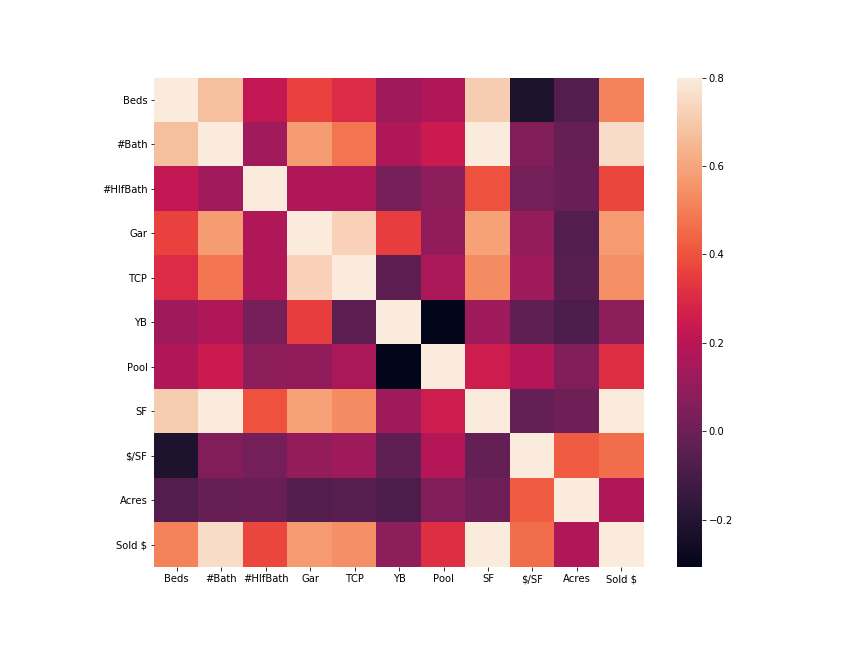
A screenshot of a cell phone

Description automatically generated

The final step in Transformation was to replace “Not a Number” entries (aka “Nan”) with zeros (or delete the listing entry completely, depending on where it occurred), to convert strings to numerical equivalents and to drop columns unneeded for later steps such as MLS Number, Address, City, etc.

**Data Analysis**

A number of graphical analyses were perform on the transformed data to understand and validate the data set before moving to the processing stage. For example, a heat map showing correlation relationships within the data is shown below, and demonstrates that normal and expected relationships exist within the data set:



For additional graphical analysis, see the appendix.

**Machine Learning**

Upon completion of extraction, transformation and analysis of the data, it was now ready to be processed using ML model. The models were expected to produce prediction for home prices. The following models were produced and tested:

1. Multivariate Linear Regression (MLR)
2. MLR Scaled
3. Random Forest
4. Gradient Boosting
5. Gradient Boosting w/drop

The following is the general procedure for generating each ML model for testing:

1. Prepare the dataset
2. Create the input (feature) and output datasets
3. Split the feature dataset into the train and learn data subsets
4. Create the model
5. Train the model
6. Test the model

Upon completion of testing of all models, individual model performance were then compared to each to determine which model demonstrated the highest performance.

Below is a sample of the data related code as it is prepared for training and testing in a model:

A screenshot of a social media post

Description automatically generated

A sample of the Python code used for creating and testing a model is shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A summary of the performance results of the ML models that were tested is shown below:

\_\_\_\_\_\_\_\_\_\_\_\_Model Type Train Test

MLR 0.8363 0.7216

MLR Scaled 0.8363 0.7216

Random Forest 0.9773 0.8499

Gradient Boosting 0.9960 0.8137

Gradient Boosting w/drop 0.9961 0.7965

**Evaluation**

As can be seen table above showing overall performance scores, the Random Forest ML model demonstrated the best performance when tested. For this reason, the other models would not be considered viable alternatives for predicting home prices.

In a proof of concept exercise, and in order to confirm the validity of the above score, the information for two other sold listings (not previously used for either training or testing of the ML models) was used as a secondary test against the completed Random Forest Model. Here are the results of that test:

* Test sample 1: predicted price: $233376, actual: $214000, difference = +9.05%
* Test sample 2: predicted price: $275244, actual price: $259900, difference = +5.9%

Thus the secondary test was consistent with the expectations for performance derived from the original scoring of the model.

Ideally, and in order to confirm the validity of the above score, a new dataset, of similar size order as the original train & test dataset, should be run through the model, and the results compared against the performance of the model produced by the score. The performance of the Random Forest Model against this secondary dataset would be expected to be very close to the overall performance score from the table above.

As stated in the project overview, the performance of price prediction would need to be within 5%. With an overall performance score of 85%, the Random Forest model predictor’s performance is 10% less than required. For this reason, accuracy would need to be improved to be usable.

**Recommendations**

In order to improve the performance of the price predictor ML model, the following development should be considered:

1. Sentiment Analysis of MLS description as input to the model
2. A larger dataset for training and testing
3. Location is guaranteed to improve performance. Incorporation of geographical information: GPS, subdivision ID (including subdivision “phases” where applicable, and clustering of either or both of these. #2 above would especially apply in this case.
   1. An expansion of this concept would be to create an app that determines subdivision boundaries using Central Appraisal District (CAD) data from each county, for every home (address & Subdivision Name). Those subdivision boundaries would be defined in GeoJSON format and used to improve the performance of future ML model predictors of any type.
4. While incorporation of all the above has a high probability of producing significant improvements in performance, a transition to a Neural Network architecture may produce even better performance.

**Conclusions**

Machine learning can indeed be used to predict sales price of a home. That fact is no longer in question. However, it is necessary to add that in the case of this project, accuracy performance was inadequate for the ML price prediction to be usable by real estate professionals. However, there is reason to believe that with improvements to the model, data available to the model, increased training iterations, and possibly a move to Neural Network based model architecture, price prediction performance can equal or exceed the range of the accuracy reflected in the Sales Price to List Price Ratio data available from the MLS.

**Appendix**

Project Files

Description Filename .

Pandas DataFrame of cleaned data 0smallListings.pkl

Jupyter Notebook Python source to clean data 1rempx.ipynb

Jupyter Notebook Python source data analysis 2rempan.ipynb

Jupyter Notebook Python source Multiple Linear 3rempest\_MLR.ipynb

Regression ML model

Jupyter Notebook Python source Multiple Linear 4rempest\_MLRscaling.ipynb

Regression with Scaling ML model

Jupyter Notebook Python source Random Forest 5rempest\_RanForReg.ipynb

ML model

Jupyter Notebook Python source Random Forest 6rempest\_RanForRegXGB.ipynb

ML model

<DIR> directory containing documentation documentation

\ This document \ Final Project Report.docx

\ Power Point Presentation \ Final Project Report.docx

<DIR> directory containing PDF source data pdfs

\Source of data for post-score testing \Denton\_County\_Sold\_072619.pdf

\Larger source data set \ QCMA6832\_1000\_entries.pdf

\Data source cleaned for train/test split \QuickCMA4925.pdf

Best performing ML model (Random Forest: “RF”) random\_forest\_model.joblib

Python source to run “RF” model from the shell rempest\_cmd.py

command line

How To Run The Model

1. Place the following files into a common directory “Path”>
   1. rempest\_cmd.py
   2. 0smallListings.pkl
   3. random\_forest\_model.joblib
2. Execute at the command prompt “Path”>
   1. “Path”> python rempest\_cmd.py C:\> python rempest\_cmd.py <Beds> <Baths> <HlfBath> \

<Gar> <TCP> <YB> <Pool: yes = 1, no = 0> <Square Feet> <Acres>

* 1. After processing, resulting output will appear: [Estimated Home Price]

Data Analysis Graphics

**Figure 1: Number of Sales vs Sale Price (Binned)**

A picture containing text

Description automatically generated

**Figure 2: Sale Price vs Square Feet (SF)**

A close up of a map

Description automatically generated

**Figure 3: Price Per Square Foot vs Square Feet**

A close up of a map

Description automatically generated

**Figure 4: Sale Price vs Year Built (all sold within the last 0-30 days)**

A screenshot of a cell phone

Description automatically generated

**Figure 5: Sale Price vs Garage Spaces**

A screenshot of a video game

Description automatically generated