Mining Real Estate Listings Using ORACLE Data Warehousing and Predictive Regression

Wuri Wedyawati, Meiliu Lu
Department of Computer Science
California State University
Sacramento, CA 95819-6021
mlu@csus.edu

Abstract

The object of this paper is to introduce our development experience of a data mining system for prospective real estate sellers and buyers to determine properties price. The prediction of continuous values of properties selling prices is modeled by a statistical technique called predictive regression.

The prerequisite of this data mining process is to design a data warehouse that contains a wide variety of real estate listings in the related areas. The data source is extracted from Multiple Listing Services (MLS) database. It is cleansed and transformed at a staging area. The data warehouse design for this system is a star schema with one large fact table surrounded by three dimension tables. Loading data into the warehouse is the final step in creating data warehouse as preparation for data mining. Oracle data warehousing tool kits ware used in the data warehouse construction. Visual Basic .NET was used to implement the data mining system.

Keywords: *data warehousing, predictive data mining, regression, real estate Multiple Listing Service*

1. Introduction

The real estate and financial services industries are finding that a mother lode of information lies just underfoot. Data mining is the latest strategy for transforming the way old-line industries will make future business decisions. The recent work in this rapid developing data mining application area has been on providing in-depth information on how to dramatically boost cross-selling opportunities, ramp up customer share and rapidly expand customer bases by leveraging the data a company and others collect. Until recently, very few have been done on helping individual prospective real estate sellers and buyers to determine current properties market value using data mining technology.

In the real estate world, key players are brokers, agents, sellers, and buyers. When a seller decides to sell his/her house, he/she has to be represented by an agent. An agent who represents a seller is called a seller agent. Seller agent will put the listing in the Multiple Listings Services (MLS) as an active listing. The active listing will become pending listing if the seller has accepted a buyer agent's offer. A buyer agent represents a buyer. The pending listing will become sold listing if the transaction is in the closed escrow state. All of those states of each real estate transaction could be found in the MLS.

It is easy for prospective buyers to find their desired houses in the MLS, but is not easy for them to answer a question, "How much money should I spend to buy that house?" On the other side, it is also difficult for prospective sellers to answer a question, "How much is my house worth now?" Help people in answering questions like these is the objective of this project. We use data mining technique to discover valuable, hidden business "intelligence" from MLS data for such decision support.

The mission of this system is to "learn" real estate price estimation model for assigning a current market price to a real estate property [16, 17, 18]. The data mining method used here is statistical linear regression. To assure the result of prediction as accurate as possible, we need to learn the model from most recent MLS data of related areas. Typically, this involves running decision-support queries that require significant computing power and which summarize thousands of rows from various distributed databases. It is difficult to do data mining using operational production system, such as MLS, without impacting its performance [9]. As a solution, a data warehouse is built based on MLS data. A data warehouse is a copy of transaction data specifically structured for query and analysis.

Building a data warehouse is a prerequisite for efficient mining of large and operational data [1, 2, 11]. In our case, a large amount of data is extracted from MLS operational production system to a staging area. Field selection, data transformation and cleansing of error entry

0-7803-8819-4/04/$20.00 ©2004 IEEE.

are preformed at the staging area. A data warehouse schema is designed first and then implemented using Oracle data management system. Once a data warehouse is created using Oracle warehousing tools, data is ready to be loaded or transported to the warehouse.

We use Visual Basic .NET as an implementation tool for transformation and cleansing, preparing data to be loaded to the warehouse, data mining, and GUI. When we load data to the warehouse we need to make connection to the Oracle data warehouse first and then pass queries from VB.NET framework to the Oracle warehouse. We choose Oracle Provider for OLE DB (OraOLEDB) component to establish this connection [19, 20, 21]. Once the connection is established, queries generated (based on user criteria) are passed to the warehouse to collect training data for data mining. As final step, a statistical regression model is built with the training data, and used to estimate the real estate property price today according to user's criteria.

We will discuss design and implementation of the data warehouse in next section, and describe predictive regression modeling in section 3. We conclude the paper in section 4 with remarks on the limitations and future enhancement.

2. Building data warehouse

A data warehouse is a collection of data gathered and organized so that it can easily be analyzed, extracted, synthesized for the purposes of further understanding the data [2]. There are four major tasks in building a data warehouse: (1) Extraction, (2) Transformation and cleansing, (3) Modeling, and 4) Transport. We will describe our experience of building the data warehouse step by step in this section.

2.1. Extraction

The first step of building a data warehouse is to obtain data from an operational source directly. The source of data needs to be documented. This task involves not only identifying the databases and files containing the data of interest, but also analyzing and documenting the business meaning of the data, data relationships and business rules. The extraction process will typically use either an unload utility or data manipulation language statements to extract the required source data. After getting the data, we copy the data to a temporary location where the data will be cleansed, transformed, and prepared for the warehouse. The temporary location is usually called as *staging area.*

We use Oracle 8i Data Warehousing tools to build a data warehouse called MASTERDW in this project. The process of building the MASTERDW data warehouse is shown in Figure 1. The data source extracted from MLS operational database system is in the flat file form. All data processed at the staging area are in the flat file form too. The resulting four flat files are loaded to the MASTERDW data warehouse and become the OFFICES table, AGENTS table, RESIDENTIAL table, and AREA table.

In the real estate world, each area has its own Multiple Listings Services (MLS). The area could be a subset of a county, a group of cities, a group of counties, or even a state, depending on how real estate agents want their associations to be. The MLS is a place for seller and buyer agents to exchange their listing information and is updated every minute whenever there are listing transactions. The operational data source used by this project is extracted from the Sacramento, El Dorado, Placer, and Yolo Counties Multiple Listings Services (MLS) database. The three counties are grouped together into one MLS database.

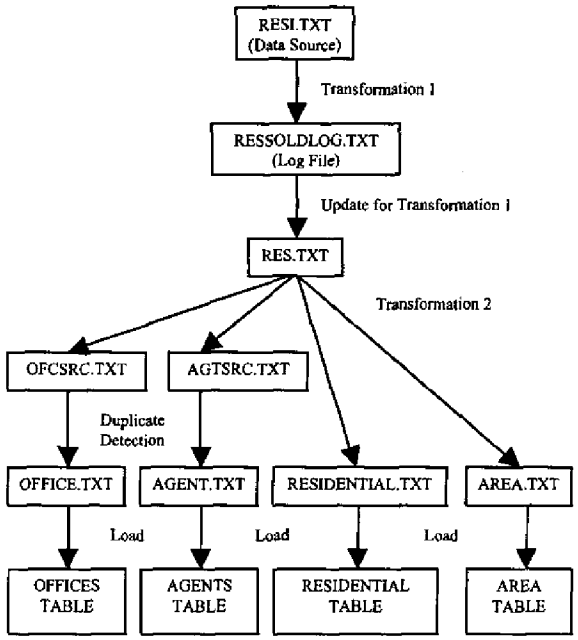

Figure 1. Building MASTERDW Data Warehouse Process

The MLS has an incremental database backup that is updated everyday. The database backup consists of six property types: residential, mobile home, residential income, land, commercial, and business opportunity. Each property type is backed up into a zip file. The data source used by this project is from the residential MLS database backup that is updated in January 9, 2004. It captures all the residential data in the source system since January 1, 1998 until January 9, 2004. The source data is

in the "|" delimited flat file and contains of 191 fields and 295787 rows. It is called "RESI.TXT".

2.2. Transformation and cleansing

After extraction from the operational system, source data needs to be transformed and cleansed [10]. There are four steps of transformation and cleansing here:

1. *Transformation 1* There are several statuses in the residential listings, such as active, expired, expired pending, pending, sold, temporary off market, and duplicate withdrawn. We use only entries with sold residential listing status for the purpose of real estate price prediction. For all the sold listings in the "RESI.TXT", this step checks ten key fields (from listing price to sold price) and writes any invalid entry into a log file ("RESSOLDLOG.TXT").

2. *Update for transformation 1* After running transformation 1 process, this step is updating the data source ("RESI.TXT") based on the log file ("RESSOLDLOG.TXT"). Most of the listings that are in the log file are error data entries detected by using common sense knowledge of real estate. To correct the error data entry of a field, the other corresponding fields in the same entry are considered during validation process. The updated listings are saved to "RES.TXT".

3. *Transformation 2* selects fields that are needed in the data warehouse. There are four tables to be created in the data warehouse: offices table, agents table, areas table, and residential table. It selects fields from the listings in "RES.TXT" and writes them into four "|" delimited files: "OFCSRC.TXT", "AGTSRC.TXT", "AREA.TXT", and "RESIDENTIAL.TXT".

4. *Duplication detection* The "AGTSRC.TXT" and "OFCSRC.TXT" contain duplicate records for agents and offices since they are created from "RES.TXT". An agent or an office can have more than one listing in "RES.TXT". Therefore, "AGTSRC.TXT" and "OFCSRC.TXT" need to be processed to eliminate the duplicate records, and the result will be saved into AGENT.TXT and OFFICE.TXT separately.

2.3. Modeling

Data warehouses and OLAP tools are based on a multidimensional data model, which is defined by dimensions and facts [8, 22]. *Dimensions* are the perspectives or entities with respect to which an organization wants to keep records. Each dimension may have a table associated with it, called a *dimension table.* A multidimensional data model is typically organized around a central theme, which is represented by a fact table. The *fact table* contains the names of the facts, or measures, as well as keys to each of the related dimension tables.

We approach the design of this dimensional database by considering four steps in the following order: (1) study the business process to model; (2) declare business grain specifying exactly what an individual fact table row represents; (3) choose the dimensions that help answering the questions; (4) identify the numeric facts that will populate each fact table row.

MASTERDW data warehouse uses star schema with one fact table and three dimension tables as shown in Figure 2.

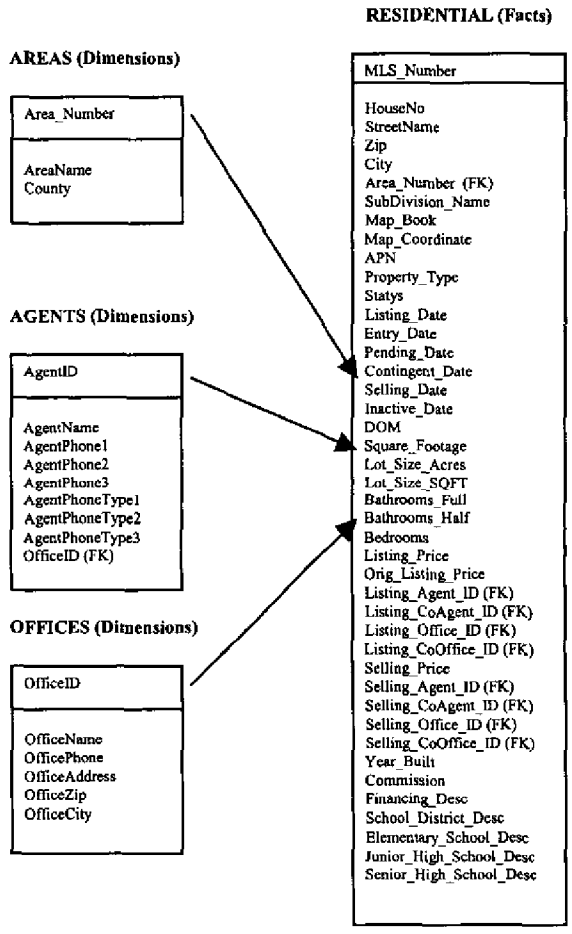

Figure 2. Star Schema for MASTERDW Data Warehouse

The fact table is RESIDENTIAL with MLS number or listing number as primary key. The dimension tables are:

- AGENTS with agent id as primary key.
- OFFICES with office id as primary key.
- AREAS with area number as primary key.

2.4. Transport

The final step of building MASTERDW data warehouse is to transport or load the data from the "|" delimited flat files to the data warehouse using SQL*Loader utility. There are four SQL*Loader statements to load four "|" delimited flat files to the data warehouse [3, 4, 5]:

1. Load "AREA.TXT" to AREAS dimension table with the following SQL*Loader statement:
   c:\>sqlldr MASTERDW/MASTERDW
   control=area.ctl log=area.log

2. Load "OFFICE.TXT" to OFFICE dimension table with the following SQL*Loader statement:
   c:\>sqlldr MASTERDW/MASTERDW
   control=office.ctl log=office.log

3. Load "AGENT.TXT" to AGENTS dimension table with the following SQL*Loader statement:
   c:\>sqlldr MASTERDW/MASTERDW
   control=agent.ctl log=agent.log

4. Load "RESIDENTIAL.TXT" to RESIDENTIAL dimension table with the following SQL*Loader statement:
   c:\>sqlldr MASTERDW/MASTERDW
   control=residential.ctl log=residential.log

The SQL*Loader statements above creates "xxx.BAD" file that contains bad records from "xxx.TXT" that cannot be loaded to xxx dimension table. For example, there is only one bad record in "RESIDENTIAL.BAD". The record is bad because there is a record with a field whose length is more than available length in RESIDENTIAL dimension table.

After the data are loaded to data warehouse, the table and indexes are to be analyzed. The purpose of "ANALYZE" command is to gather statistics that are used by the optimizer. Features like summary management will not available if there is no statistics. The SQL statements that are used to analyze statistics are:

```
SQL> ANALYZE TABLE AREAS COMPUTE
STATISTICS;
Table analyzed.
```

3. Predictive modeling

In this data mining system we use a linear regression model to predict the price of a real estate property today, on the basis of MLS data. Linear methods for regression are simple and often provide an adequate and interpretable description of how the inputs affect the output [13, 14, 15]. For prediction purposes they can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases [6, 7, 8, 9]. This is exactly the reason for us to choose regression method here since each search result from the data warehouse can only provide us a relatively small training set.

Predictive regression models a random variable Y (called a response variable) as a linear function of another random variable X (called a predictor variable) [12]. Given n samples or data points of the form (x_1, y_1), (x_2, y_2), …, (x_n, y_n), where $x_i \in X$ and $y_i \in Y$, predictive regression can be expressed as

$$Y = \alpha + \beta \cdot X \qquad (3.1)$$

where α and β are regression coefficients. With the assumption that the variance of Y is a constant, these coefficients can be solved by the method of least squares, which minimizes the error between the actual data points and the estimated line :

$$\beta = [\sum (x_i - mean_x)(y_i - mean_y)] / [\sum (x_i - mean_x)^2] \qquad (3.2)$$

$$\alpha = mean_y - \beta \cdot mean_x \qquad (3.3)$$

where $mean_x$ and $mean_y$ are the mean values for random variables X and Y given in a training data set.

There are two variables involved in the predictive regression method as shown in equation (3.1). They are X as input and Y as response output that depends on X. In this application, the response output (Y) is today's house price and the input (X) is today's date since the goal of this project is to estimate today's house price. Therefore, the calculation uses sold date as X and sold price as Y. In order to get the value of Y, equation (3.1) needs regression coefficients which are α and β. The value of α and β are generated using equations (3.2) and (3.3) from a training data set from the previous listings. The training data set is selected from the MASTERDW data warehouse based on a set of parameters entered by the user, such as status, area number, square footage, number of bedrooms, number of full bathrooms, number of half bathrooms, and year built. An example of the query selection is:

"Select * from Residential where (Status = 'Sold') and (Area_Number) = '10835' and (Square_Footage between '2000' and '3000') and (Bedrooms = '4') and (Bathrooms_Full = '2') and (Bathrooms_Half = '0') and (Year_Built = '2001')"

It returns 5 listings with all the fields in Residential table. The 5 listings with MLS number, sold date, and sold price fields are shown in Table 3.1.

Table 3.1. Result of An Example Query

MLS_Number	Sold_Date	Sold_Price
152301488	15-Apr-03	333000
152119745	19-Feb-02	262000
30038339	17-Sep-03	369900
30034997	27-Oct-03	337900
30016832	22-Jul-03	340000

The sold date field needs to be converted into integer. The steps to convert sold date field into integer are:

1. Get the minimum sold date.
2. Get differences between each sold date and the minimum sold date for each listings.
3. Add each difference by 1 and use it as X.

The minimum sold date of Table 3.1 is 19-Feb-02. Therefore, the value of X of 19-Feb-02 is 1. The value of X and Y of table 4.1 is shown in table 4.2.

The average of X ($mean_x$) is 425. The average of Y ($mean_y$) is 328560. The value of β is 150.98. The value of α is 264391.38. The equation (3.1) becomes

$$Y = 264391.38 + 150.98 * X \qquad (3.4)$$

Equation (3.4) is used to predict today's real estate price. If today is 28-Mar-04, then X is 767. For the given parameter values in our query selection example, today's real estate price estimate is 380196.86. This example screen output is shown in Figure 3.

With the help of this data mining system, individual prospective real estate sellers and buyers now are able to determine current properties market value based on recent market values of similar properties. This knowledge discovered from MLS database gives users of the system additional information in making important real estate decision. As many other decision support system, its success not only depends on the technology but also user's judgment and other information.

However, based on different architecture designs analysis [8], the degree of success for a data mining system can also depends on how close a data mining system is coupled with a DB/DW system. This loose coupling data mining system, though not as efficient as we expect it to be, is better than no coupling since it makes use of both MLS data and Oracle data warehousing tool kits.

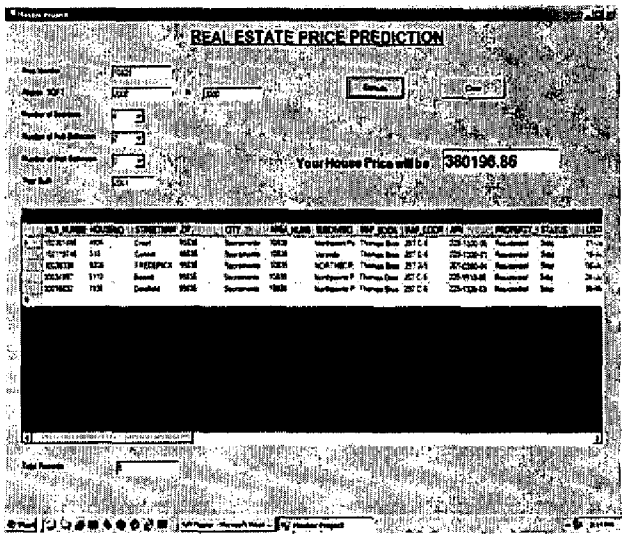

Figure 3. Real Estate Price Prediction Result Form

4. Conclusion

In this paper, we described a predictive data mining system for mining real estate Multiple Listings Services (MLS) data. The system offers several advantages: (1) it can be used to predict current real estate property price based on recent real estate market data; (2) an easy-to-use GUI allows the user to customize the relevant parameters; (3) training data for each predictive data mining activity is facilitated by a data warehouse MASTERDW to allow efficient data mining.

The design and implementation of this project is a successful integration of the following key elements: (1) thorough understanding of business meaning of the data, data relationships and business rules of real estate; (2) Oracle data warehousing tool kits; (3) predictive regression method; and (4) Microsoft Visual Basic .NET.

There are some obvious limitations and observations for this prototype system. First, the training data selection criteria are limited to a subset of features in MLS data only. Second, the model is based on a short-term bull market for real estate without consideration of other possible economic trend. Third, more exploration on other data mining method may be helpful. Last, the data warehouse is a stand-alone system without an automatic updating facility.

There are several ways to improve this data mining system: (1) add more selectable features for users, such as whether a house has a pool or not, how many car

garages a house has, how many square feet of a house lot is, and school district information; (2) include economic data in price prediction model; (3) study the possible extension of using k-nearest-neighbor prediction rule for this application; (4) streamline the data warehousing and data mining process to advance the degree of coupling to semitight coupling. Semitight coupling in this case means that besides linking a DM system (predictive regression) to a DB/DW (MLS database/ MASTERDW), efficient implementations of a few essential data mining primitives (frequently encountered data mining functions) and some frequently used intermediate mining results can be precomputed and stored ready to go, therefore it will enhance the performance of the data mining system.

References

1. Inmon, W.H., *Building the Data Warehouse (Third Edition)*, John Wiley & Sons, 2002.
2. Imhoff, C., Galemmo, N., and Geiger, J.G, *Mastering Data Warehouse Design*, John Wiley & Sons, 2003.
3. Hobbs, L. and Hillson, S., *Oracle 8i Data Warehousing*, Digital Press, 2000.
4. Dodge, G. and Gorman, T., *Essential Oracle 8i Data Warehousing: Designing, Building, and Managing Oracle Data Warehouses (Second Edition)*, John Wiley & Sons, 2000.
5. Corey, M.J., Abbey, M., Taub, B., and Abramson, I., *Oracle 8i Data Warehousing (Second Edition)*, McGraw-Hill Osborne Media, 2001.
6. Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag New York, Inc., 2003.
7. Dunham, M., *Data Mining: Introductory and Advanced Topics*, Prentice Hall, 2003.
8. Han, J. and Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001.
9. Witten, I.H. and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Jawa Implementations*, Morgan Kaufmann, 2000.
10. Pyle, D., *Data Preparation for Data Mining*, Morgan Kaufmann, 1999.
11. Pyle, D., *Business Modeling and Data Mining (First Edition)*, Morgan Kauffmann, 2003.
12. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., *Advances in Knowledge Discovery and Data Mining*, MIT Press, 1996.
13. Giudici, P., *Applied Data Mining: Statistical Methods for Business and Industry*, John Wiley & Sons, 2003.
14. Bozdogan, H., *Statistical Data Mining & Knowledge Discovery*, CRC Press, 2003.
15. Weiss, S. and Indurkhya, N., *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann, 1997.
16. Thuraisingham, B., *Data Mining: Technologies, Techniques, Tools, and Trends*, CRC Press, 1998.
17. Edelstein, H.A, *Introduction to Data Mining and Knowledge Discovery (Third Edition)*, Two Crows Corporation, 1999.
18. Klosgen, W., Zytkow, J.M., and Zyt, J., *Handbook of Data Mining and Knowledge Discovery (First Edition)*, Oxford University Press, 2002
19. Wakefield, C., Sonder, H., and Lee, W.M., *VB .NET Developer's Guide*, Syngress Publishing, Inc., 2001.
20. Balena, F., *Programming Microsoft Visual Basic® .NET™*, Microsoft Press, 2002.
21. Deitel, H.M., Deitel, P.J., Nieto, T.R., and Yaeger, C.H., *Visual Basic® .NET™ for Experienced Programmers*, Prentice Hall, 2003.
22. Kimball, R., Ross, M., *The Data Warehouse Toolkit (Second Edition)*, Wiley, 2002