House Price

REPRICE

Analysis Class Report

Revision History

Authors	Description of Change	Sections	Rev	Date
Mohammad H., Kunal M., and Don F.	Initial Release	All	0	2/18/2019

Table of Contents

1 Team Description

Team Member Name	Email Address
Mohammadreza Hajy Heydary	Mheydary@csu.fullerton.edu
Kunal Matthews	Amos-m2345@csu.fullerton.edu
Don Feng	donfeng97@csu.fullerton.edu

2 Project Description

Statement of Purpose:

Design and implement a web-based housing price estimator based on physical and geographical attributes of a property.

Detailed Description:

REPRICE is a web-based housing price estimator based on physical and geographical attributes of a property. The system leverages machine learning to predict the estimated median value of a property at different levels. With the results, the user will be able to decide on whether to buy, sell or wait for a better market price.

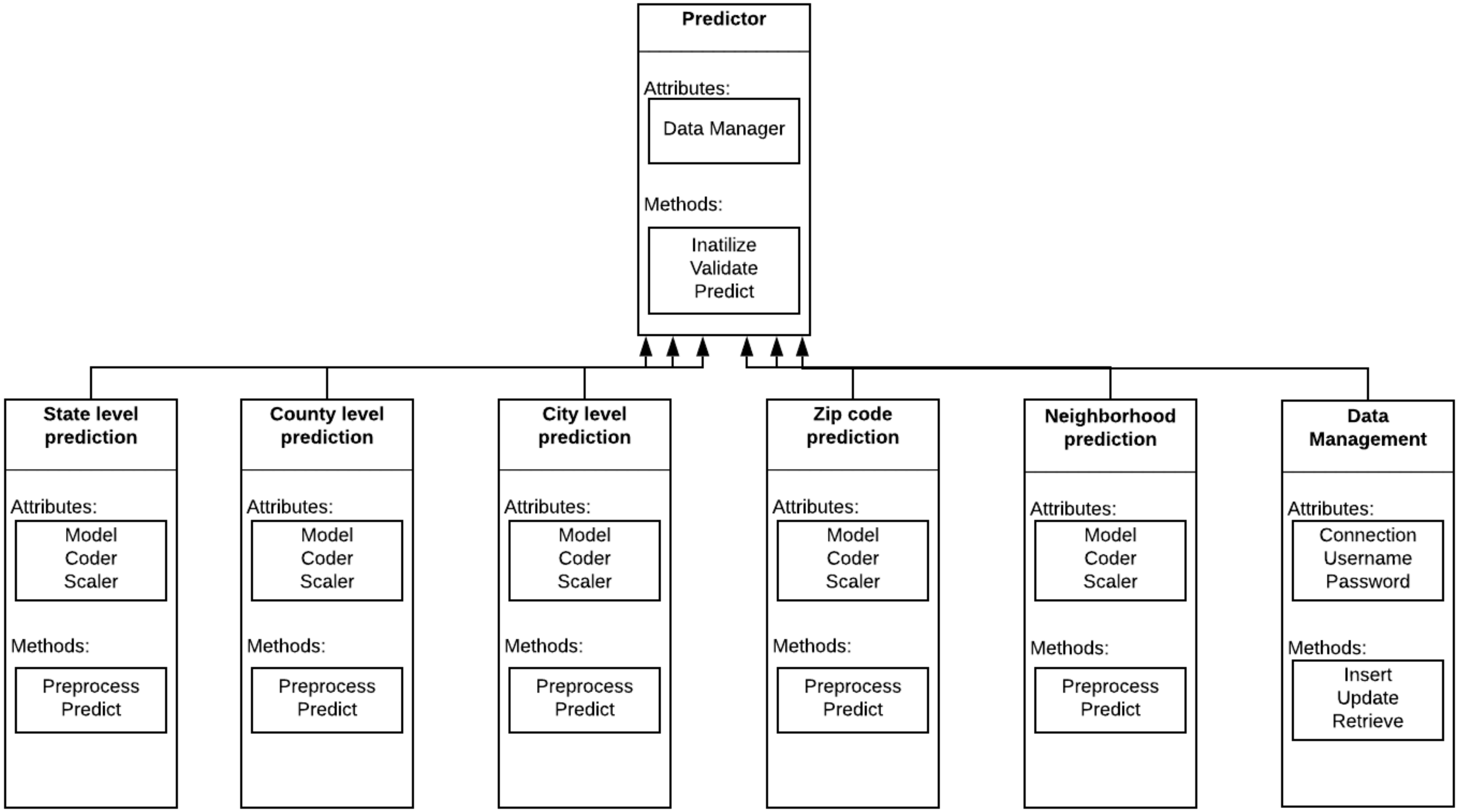
Modeling Detail:

REPRICE utilizes the Zillow research dataset which is freely available to the public. Zillow's data is broken into multiple subsets each of which addresses an attribute that is thought to be significant in determining the value of a property. REPRICE will use the following subsets:

1. State time series: a list of historical median property values in a state through time
2. County time series: a list of historical median property values in a county through time
3. City time series: a list of historical median property values in a city through time
4. Zip code time series: a list of historical median property values in a zip code through time
5. Neighborhood time series: a list of historical median property values in a neighborhood through time
6. House information: property specific attributes such as the number of bedrooms and square footage

The dataset provided by Zillow contains historical data going back to 1998, however many values are missing. Thus, we shall only use the more recent data that has minimal missing values to make the analysis feasible. For modeling, we plan to break the process into multiple models each predicting at one level in a serial format (i.e., one model's output is appended to the input of the next model in the series). A unified modeling approach is not feasible due to the chaotic and complex behavior of the housing market time series at each level. The final prediction of the series of models along with previous predictions will be mapped to a human-readable format and presented to the end user. The details of modeling are subject to change as we do more experimentation on the data and discover potential changes that could reduce the overall prediction error.

2.1 Analysis Class Diagram



2.2 Analysis Class List

Sequence Number	Analysis Class
1	Predictor
2	State-Level Prediction
3	County-Level Prediction
4	City-Level Prediction
5	Zip-Code Level Prediction
6	Neighborhood-Level Prediction
7	Data Management

2.2.1 Predictor

Description: This class acts as interface for the proceeding prediction classes and to simplify interaction with the chain of predictor models, input requirements, database communication, and user input validation

Methods:

- Initialize: instantiate the basic variables required to process user input such as connection to the database and opening log file.
- Validate: this private method is used to check the user input and make sure the user entries could be parsed and fed to the model chain
- Predict: instantiate an instance of the prediction classes and call them in order to predict the respective level of data and feed the results along with required inputs to the next element in the chain of models. Once done, return the adjusted predictions to the user

Attributes:

- Data manager: an instance of the database class used to communicate with database, retrieve the data required for each level of modeling and store logs

2.2.2 State Level Prediction

Description: The state-level prediction class encapsulates loading process of the main units from the database, preprocessing new data, and predicting the state-level median property value. Upon instantiation, the class requires inputting the state model, the set of standard scaler parameters (mean and standard deviation), and the set of encoder maps for each attribute if the attribute is categorical (e.g., California is mapped to the unique numerical value 1).

Methods:

- Preprocess: this is a private attribute called by the predict method to preprocess the new query based on the parameters loaded upon instantiation.
- Predict: this method inputs one object with the format: <state_name, the vector of past *m* true observations> and returns a point estimate of the property median value for the given state in the current period.

Attributes:

- Model: the previously trained state-level model loaded from the database
- Coder: the previously fitted state-level encoder/decoder loaded from the database
- Scaler: the previously fitted state-level standard scaler loaded from the database

2.2.3 County Level Prediction

Description: The county-level prediction class encapsulates loading process of the main units from the database, preprocessing new data, and predicting the county-level median property value. Upon instantiation, the class requires inputting the county model, the set of standard scaler parameters (mean and standard deviation), and the set of encoder maps for each attribute if the attribute is categorical (e.g., Orange is mapped to the unique numerical value 1).

Methods:

- Preprocess: this is a private attribute called by the predict method to preprocess the new query based on the parameters loaded upon instantiation.
- Predict: this method inputs one object with the format: <county_name, state_level_prediction, the vector of past *m* true observations> and returns a point estimate of the property median value for the given county in the current period.

Attributes:

- Model: the previously trained county-level model loaded from the database
- Coder: the previously fitted county-level encoder/decoder loaded from the database
- Scaler: the previously fitted county-level standard scaler loaded from the database

2.2.4 City Level Prediction

Description: The city-level prediction class encapsulates loading process of the main units from the database, preprocessing new data, and predicting the city -level median property value. Upon instantiation, the class requires inputting the city model, the set of standard scaler parameters (mean and standard deviation), and the set of encoder maps for each attribute if the attribute is categorical (e.g., Fullerton is mapped to the unique numerical value 1).

Methods:

- Preprocess: this is a private attribute called by the predict method to preprocess the new query based on the parameters loaded upon instantiation.
- Predict: this method inputs one object with the format: < city _name, county_level_prediction, the vector of past *m* true observations> and returns a point estimate of the property median value for the given city in the current period.

Attributes:

- Model: the previously trained city-level model loaded from the database
- Coder: the previously fitted city-level encoder/decoder loaded from the database
- Scaler: the previously fitted city-level standard scaler loaded from the database

2.2.5 Zip-Code Level Prediction

Description: The zip code level prediction class encapsulates loading process of the main units from the database, preprocessing new data, and predicting the zip code level median property value. Upon instantiation, the class requires inputting the zip code model, and the set of standard scaler parameters (mean and standard deviation)

Methods:

- Preprocess: this is a private attribute called by the predict method to preprocess the new query based on the parameters loaded upon instantiation.

- Predict: this method inputs one object with the format: < zip_code, county_level_prediction, city_level_prediction, the vector of past *m* true observations> and returns a point estimate of the property median value for the given zip code in the current period.

Attributes:

- Model: the previously trained zip code level model loaded from the database
- Coder: the previously fitted zip code level encoder/decoder loaded from the database
- Scaler: the previously fitted zip code level standard scaler loaded from the database

2.2.6 Neighborhood Level Prediction

Description: The neighborhood-level prediction class encapsulates loading process of the main units from the database, preprocessing new data, and predicting the neighborhood-level median property value. Upon instantiation, the class requires inputting the neighborhood model, the set of standard scaler parameters (mean and standard deviation), and the set of encoder maps for each attribute if the attribute is categorical (e.g., *South Los Angeles* is mapped to the unique numerical value 1).

Methods:

- Preprocess: this is a private attribute called by the predict method to preprocess the new query based on the parameters loaded upon instantiation.
- Predict: this method inputs one object with the format: < RegionID, county_level_prediction, city_level_prediction, zip_code_level_prediction, the vector of past *m* true observations> and returns a point estimate of the property median value for the given neighborhood in the current period.

Attributes:

- Model: the previously trained neighborhood-level model loaded from the database
- Coder: the previously fitted neighborhood-level encoder/decoder loaded from the database
- Scaler: the previously fitted neighborhood-level standard scaler loaded from the database

2.2.7 Data Management

Description: All the house price data will be pull from Zillow and storage into our database. All data will connect to the classes above (State, County, City, Zip-code, neighborhood) when insert and Retrieve them. Each month the database will update by pull over newest data from Zillow.

Methods:

- Insert: append new records to the database.
- Update: Check Zillow for most recent records and update database accordingly.
- Retrieve: return user specified data.

Attributes:

- Connection: the connection to communicate with the database.