Machine Learning Engineer Nanodegree

Capstone Proposal

Xiao Wei

Citations in footnotes

Domain Background

Real-estate in the United States is estimated to be worth $30 trillion dollars at the end of 2016, which is more than the combined market capitalization of every publicly listed company in United States[1]. In addition, much of middle class households in United States (and around the world) have their net worths tied closely to the real estate market. This means that fluctuations in the price of real-estate affects many more people than fluctuations in the securities market (which are mostly owned by wealthy individuals).

The price of a house is made up of the attributes of the house and the surrounding areas. There is a rich history in academia of predicting housing prices. Recently, tech firms have been moving into this space and accurate prediction of housing prices is something that would disrupt the real estate broker industry that is plagued by inefficiencies and misaligned incentives (as detailed in Freakonomics). To help guide my prediction algorithm, I will be using an academic paper that predicted housing prices in Singapore[2].

Problem Statement

At the writing of this proposal there is a competition on Kaggle that is sponsored by Zillow that seeks to predict housing prices in the Los Angeles area. The competition provides the training data and the metric measurement criteria. Zillow predicts prices for almost every listing that they have on their website. Any model that is created can be trained on more data or used to predict more test cases. This is a regression problem where instead of predicting a housing price, the response is prediction of the error of the Zillow prediction model.

The inputs are the housing features provided by Zillow as well as sources found on the internet described in the “Datasets and Inputs” section below. Output of the model is a continuous variable that is supposed to represent the log error that the ZIllow model would predict for a given observation for a given time period. Another way of putting it is that this model predicts the error of another prediction model.

---

[1] https://www.zillow.com/research/2016-total-home-value-rents-14028/

[2] Jiang, Liang and Phillips, Peter C. B. and Yu, Jun, A New Hedonic Regression for Real Estate Prices Applied to the Singapore Residential Market (December 2, 2014). Cowles Foundation Discussion Paper No. 1969. Available at SSRN: https://ssrn.com/abstract=2533017 or http://dx.doi.org/10.2139/ssrn.2533017

Datasets and Inputs

The training data is provided by Zillow includes features for the houses. The actual log errors (y-label) for the training data is also provided alongside a data dictionary. In addition to the provided data, I also plan to use the S&P Case Shiller Index for the greater Los Angeles Area[3]. Another source of data would be Chinese foreign exchange reserves, since Chinese buyers have been snatching up American real estate, especially in Los Angeles[4].

- Properties_2016.csv - provided by Zillow. Contains features of properties in greater Los Angeles area
- Train_2016.csv - provided by Zillow. The y-labels for the properties. This is log(actualprice - predictedprice) on the Zillow website
- LXXRSA.csv -S&P Shiller Case price index from the internet
- Chinese foreign reserves - instrument variable for Chinese buyers from the internet

Features from Properties_2016.csv contain relevant features describing houses themselves such as square footage, number of fireplaces, pools, basement, etc. I am not sure which features will drop out of the model. You can see the features in the zillow_data_dictionary.xlsx file. The LXXRSA.csv contains one feature for the S&P Shinner Case price index, a continuous variable for the price index of homes in the LA area. Chinese foreign reserves is a continuous variable for the amount of foreign currency held by Chinese government. It is an instrument variable for the amount of Chinese money in the LA real estate market.

Solution Statement

Before the model I will need to clean/impute data. Shiller price index and Chinese currency reserves need to be merged into property_features data. Transformation of heteroscedastic variables may be necessary. For cross validation, time_series_split will be used instead of the normal cross validation k-fold split so that the validation set will always be in the future of the training set.

I plan to use a generalized linear model. I will most likely use elasticnet model of sklearn library since there will be many features that may be correlated and the elasticnet should help filter them out (or shrink them to 0). Regression is the most used type of model for prediction of continuous response variable. Predictions will be compared to actual using the mean absolute error metric. Other models that will be explored include XGBoost, which is popular boosting method for Kagglers, and a multilayer neural network. Regression and XGboost models will go

[3] https://fred.stlouisfed.org/series/LXXRSA

[4]http://www.safe.gov.cn/wps/portal/!ut/p/c5/04_SB8K8xLLM9MSSzPy8xBz9CP0os3gPZxdnX293QwP30FAnA8_AEBc3C1NjIxMjA6B8JE55dzMDArrDQfbhVhFsjFcebD5I3gAHcDTQ9_PIz03VL8iNMMgMSFcEAP5jfwo!/dl3/d3/L2dJQSEvUUt3QS9ZQnZ3LzZfSENEQ01LRzEwT085RTBJNkE1U1NDRzNMTDQ!/?WCM_GLOBAL_CONTEXT=/wps/wcm/connect/safe_web_store/state+administration+of+foreign+exchange/data+and+statistics/forex+reserves/foreign+exchange+reserves/bb4353804c420bf6aa0aaefd3fd7c3dc

through cross validation. Manual cross validation code may have to be written for the neural network model. Model averaging may be used to produce the final predictor.

Benchmark Model

The benchmark model will be taking the average of the response variable as the predictor for all observations. This is equivalent to a regression model with only the bias term. The mean absolute error evaluation metric will be used to evaluate the actual model vs benchmark. Another possibility is to use an OLS model on the same data that is fed into the final model. Either benchmark will use the same evaluation metric/comparison.

Evaluation Metrics

Zillow provides the attributes for all the observations for the training set as well as what the evaluation metric is going to be. The dependent variable is pretty weird (difference between mean absolute error of the predicted log error and actual log error (actual log error is the log error of what Zillow predicted and the actual selling price). This means that the goal isn't to predict housing sales but rather predict what is inaccurate about Zillow's prediction algorithm.

Housing prices isn't actually provided by Kaggle so this project can't actually predict housing prices. The evaluation metric will be mean absolute error.

Project Design

First step to any model is to conduct exploratory data analysis on the data. In this case the Properties_2016.csv. Any features with more than 10% missing values will probably be discarded. Important features with some missing values will need to have those values imputed, continuous variables will probably use median while binary variables will be 0. One important step for regression models is to account for heteroscedasticity. Heteroscedasticity is more prevalent than the homoscedasticity assumption that linear regression models make. Correcting for heteroscedasticity usually involves log transformation.

Some of the data (such as chinese foreign reserves will need to be interpolated for months in 2017 of which there is no data. I will just use the latest month of data that is available and set that for the months in the future). The foreign reserve and S&P Shiller Case Price index will also need to be manually merged into the Properties_2016 data by date.

It is possible that PCA will be done as part of the EDA if there are features that could be combined so that number of dimensions can shrink before the model.

After preprocessing is finished I will run elasticnet regression alongside maybe Huber regression and use cross-validation to select the best hyperparameters. The winner of elasticnet vs Huber will be used as part of the final model. I will explore using XGBoost. If XGBoost performs better than regression I may use that instead. Both the regression models as well as XGBoost will be subject to cross-validation to tune the models.

I will then use a neural network. I do not plan to use any convolutional layers. They will all be fully featured layers with varying number of nodes. I will experiment with using 2 or 3 layers and also the number of nodes in each hidden layer. Tuning the neural network parameters will require much time since I don't have much experience with the method. The two models' outputs will be averaged to produce the final prediction. I may also weight the model predictions depending on how they do in cross validation.