Recursive Forecasting of Housing Prices using Multi Output Regression with Statistical Analysis

Jayashree Selvan
Electrical & Computer Engineering
UMass - Amherst
jselvan@umass.edu

Sravya Ganugapati
Electrical & Computer Engineering
UMass - Amherst
sganugapati@umass.edu

1 Introduction

The housing market has always been a topic of attention globally. It is not just the trends in housing market which is of concern to buyers and owners, but it also reflects the present economic situation in the region. Besides affordability of a house, many other factors also affect the selling price of a house in a region. In our project, we try to forecast the selling price of a house in two states Massachusetts (MA) and New York (NY) for two months based on the history of 70 months. We also analyze how each factor out of the selected features affect the selling price of a house in a certain region.

The data is extracted from Zillow website, an online real estate database company that contains data about the houses in the United States with different factors and it offers services like forecasting these factors, allowing sellers and buyers to arrive at a decision. The creative point about our work is we try to forecast the selling price for two successive periods of time (months) at one go and repeating this recursively using Multi Output Regression. This is defined as multivariate regression where there is more than one target variable. We also perform statistical analysis that describes how each factor related to the median selling price of the houses.

Our project results present the best regression model coupled with best hyper parameter value and PCA for every dataset. The RMSE scores of forecasted target labels over two months for each dataset is also presented. Interestingly the way of handling missing values in the datasets resulted in huge difference in regression model performance, difference in the relation between median selling price and every other feature. On the whole, two datasets performed diversely from one another on the experiments performed which strongly prove the underlying observations, feature and history affect the forecasting problem.

2 Related Work

Our work is mainly focused on forecasting the selling price using Multi Output Regression and performing the Statistical Analysis. Zillow has performed many forecasts of many variables, but we try Multi Output Regression which is not used by Zillow. Multi Output Regression allows us to forecast the prices for the successive steps at once and any conventional regression model that does not support multivariate regression natively can be used with which the advantages of the best regression models can be exploited.

[1] used genetic algorithm to find the parameters of SVM and using that SVM model to forecast the housing price, whereas we used Cross validation technique to find the best hyperparameter values for the models used as the latter is more robust and robustness is needed as the data had missing values and therefore, some randomness is present in the data.

There were experiments performed on popular Boston dataset and Ames dataset. Most of the works were predicting the target labels given the current set of feature values. The real forecasting problem was under explored and only one target variable was always used. Statistical analysis were performed on Boston Dataset [2] which did not cover detailed

aspects of the correlation between the features and the single target label.

The works on Ames dataset focused on one specific regression model but did not consider using multi output regression model because again the target label was just one. Support vector model was extensively studied in almost all the prior works and forecasting univariate housing price model with ARIMA, ARMA, AR, and VAR models from statistical models were studied. These perform really well with Univariate time series where the target variable itself is used to train the regression model. The experiments were mainly focused on studying the reliability and prediction accuracy level of the regression models. The actual forecasting for more than one month/time period was limitedly looked upon.

3 Dataset

The data is collected from Zillow website. Each feature used in the experiment is present in a different csv file and they are integrated into a single file 'MA_dataset.csv' and 'NY_dataset.csv'. These files contain 490 data cases with 10 features, each data case pertaining to a day in a month. So, we have data for 70 months from October 2010 to July 2016 and 7 days (5, 10, 15, 18, 20, 25, and 28) in each month thus resulting in 490 data cases. The base features to forecast MSP are Median List Price per sqft.(MLP, value-$, type-continuous real float), Median Price Cut (MPC, value-%, type-continuous real float), Sold For Loss (SFL, value-%, type-continuous real float), Sold For Gain (SFG, value-%, type-continuous real float), Increasing Values (IV, value-%, type-continuous real float), Decreasing Values (DV, value-%, type-continuous real float), Turnover(TNV, value-%, type-continuous real float), Buyer Seller Index (BSI, type-continuous real float), Price To Rent Ratio (PTR, type-continuous real float), Market Health Index (MHI, value-%, type-continuous real float). For more information on the description, please refer to 'readme.txt' in the 'Code' folder.

The time series forecasting needs the data for the time to be forecasted in a different way unlike the usual predicting methods. As the features for the date of a month of which MSP is forecasted are not available until the end of that time period, it makes little sense to forecast the MSP at the end of the time period where the actual MSP could be already known. So, we need to perform lagging to extract the input that is to be given for forecasting. The features for forecasting the MSP of time period t are the features and the MSP of the previous time periods and number of lags is defined as the number of previous time periods whose features are given as input to forecast the MSP of t. The optimal number of lags is obtained by performing Recursive Feature Elimination and finding the optimal number of lags for each feature. Most of the features return 8 as the optimal number of lags and so, the data should be transformed only with 8 lags. So, this results in 88 features (11*8) and two target labels. We are performing multivariate regression to predict two target values, one being the MSP of t and the other is of t+1 i.e., the next step. This transformed data with lags is saved as 'MA_Lagged_data.csv' and 'NY_Lagged_data.csv'. PCA is performed on these 88 features to find the best 14 features and the input is transformed accordingly, saved as 'MA_New_Laggeddata.csv' and 'NY_New_Lagged_data.csv'.

4 Methodology

The methodology used for forecasting MSP is implementing a pipeline to find the best estimator object and then forecasting using this object on the features of the target period of time. The multiple stages of this experiment are as follows: Data Collection and Integration, Handling missing data, Feature creation, performing PCA on the created features and extracting the optimal features, transforming the input with the features, implement a pipeline on the input to perform cross validation to find the best hyperparameter and use the this best hyperprameter in the estimator object to finally forecast.

4.1 Preprocessing Data and Feature selection

The missing data is handled in a different way in the two datasets. In MA dataset, the

missing data is filled with random values that are in between the maximum and minimum of the values of that feature in a month. In NY dataset, the missing values are just replaced with zero.

Principal Component Analysis (PCA) is used to find the 14 best features by calculating the covariance matrix and those features are retained for which the absolute value of the correlation is the maximum as greater correlation value means both the features are not similar and they don't give the same information.

4.2 Pipeline

After extracting the 14 mostly correlated features from PCA, the input is transformed with these 14 features. Next, hyperparameter selection using GridSearchCV with a pipeline is performed to get the value of the best hyperparameter. Usually, the data is shuffled before splitting. But, that is not the case here as the data follows a time series and shuffling violates the natural expected order of the data.

4.3 Models Used

Three regression models, Lasso, KNN and Gradient Boosting Regressor, PCA for feature extraction and GridSearchCV for hyperparameter selection are used on both datasets. We also use Multi Output Regressor as these regression models don't support multivariate regression problem except Lasso.

4.3.1 Multi Output Regressor

This is the regression model for extending the regressors that do not support multi target regression natively. This works by performing regression on each target label independently with the given regression model object. This assumes there is no dependence between the target labels specified for a given input vector.

4.3.2 Lasso

Lasso is an L1 regularization of the basic Linear Regression model where the objective function is to reduce the Mean Squared Error of the predicted labels and the actual labels. The regularization is performed to avoid overfitting the data as part of capacity control. The hyperparameter that is tuned is the 'alpha' of the Lasso model. The mathematical objective function of the Lasso is shown below:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{x}_i \mathbf{w})^2 + \lambda ||\mathbf{w}||_1$$

4.3.3 KNN

K-nearest neighbor regression model fits the training data by memorizing it and predicting the labels by taking the mean of the labels returned by its k nearest neighbors, k being the hyperparameter for this model. The label predicting function is shown below:

$$f_{KNN}(\mathbf{x}) = \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x})} y_i$$

4.3.4 Gradient Boosting

Gradient Boosting regressor is an ensemble regression model that adds new models sequentially to minimize error. It identifies the data cases that produce large error and added new model that focuses on these data cases, thereby reducing the error sequentially.

4.3.5 PCA

Principal Component Analysis extracts those features that are more correlated to each other than the other. Given n features, an nxn covariance matrix is calculated with value in each

row gives the correlation coefficient for those two features. If we need k mostly correlated features (k<n), k principal component axes are obtained from which the features can be in turn extracted. In each principal component, the feature that has the maximum absolute correlation coefficient is considered the feature pertaining to that principal axis and likewise, k features are extracted.

4.3.6 GridSearchCV

GridSearchCV is used to perform cross validation on the training data to select the best hyperparameter value of the model that reduces the RMSE. It first divides the training data into the number of splits mentioned and one block is used as validation data and the other as training data and the model is fit and predicted and RMSE is calculated. This is repeated by selecting a different block as validation and the remaining as training for each hyperparameter value. The hyperparameter value for which the mean of RMSE's is the minimum is selected.

5 Experiments and Results

5.1 Experiments Perfomed

The target label of interest in this project is the MSP per sq. ft. Here, the observations are recorded for seven different dates every month and our aim is to train the regression models to forecast two months MSP values i.e. 14 individual forecasts. Once the initial dataset was created, as the first step we derived correlation between the features using a correlation matrix heatmap. Then we obtained scatter plot of every feature with the MSP to compare features as they relate to the MSP by statistical analysis. In order to find the optimal lag history that would result in more accurate forecast of MSP, we iterated through 12 lags i.e. shifted the data 12 steps behind, creating (11*12) features, to create historic data and generated a lag rank plot for every feature. Basically, the lag with lowest rank has higher importance.

After manual observation of the lag rank plots, 8 lag steps were considered optimal and a new dataset with 8 step lag was created. This dataset now consisted of (11*8) features which has to be reduced to avoid overfitting or underfitting of the test data. Hence, performed dimensionality reduction with PCA which reduces the number of features to 14 but retains the number of observations. All the above steps are performed for each dataset separately. Then hyper parameter optimization is performed on this transformed data with GridSearchCV. We are comparing the prediction performances of three different regression models namely Lasso, KNN and GradientBoosting for the two datasets. Once the initial prediction of MSP is obtained for t and t+1 time period, we then use this value along with the previous history feature recursively to predict t+2, t+3 and so on until t+14 time period which together account for 2 months future prediction each with seven individual day's prediction.

5.2 Statistical Analysis

Statistical analysis can be used for a variety of reasons. In this project we performed this analysis to understand the relation between the observations. Specifically to understand how every feature is related to the value of MSP in that observation. Analyzing each data point individually is not scalable hence we generated scatter plots for every feature namely MLP, MPC, BSI, MHI etc. against MSP. These plots have the ability to provide an idea of the underlying correlation between the individual features with MSP. Whenever the plots move upwards with the X-axis value, that depicts a positive correlation of the X and Y variables. On the other hand, if the plot moves downwards then it indicates a negative correlation. With this plots, we can also understand if there is a linear relation between the X and Y variables and can derive a linear regression model based on the plot.

In our experiment, for the NY dataset the plots are lumped almost for all features with low to no noise i.e. outliers, whereas in MA dataset every scatter plot has lot of outliers and the correlation cannot be concluded as either positive or negative. This difference among the

data can be due to the way missing values are handled. Since MA dataset's missing values are replaced with mean values, the noise is suppressed.

Deriving inference about effect of other features on MSP, for MA dataset from the scatter plots the features BSI, DV, IV and MHI would be the best to forecast MSP based on their positive relation with MSP. MPC feature can also be included and it does not affect the quality of forecast. In NY dataset, features namely DV, IV, MLP, PTR and MPC establish a positive correlation with MSP and would be best suited to perform the forecast. The rest of the features also seem to have predictive power as they are negatively correlated with MSP.

We also plotted a correlation matrix heatmap that shows how stronger or weaker the level of correlation is between each feature and every other feature.

5.3 Learning and Forecast with Regression Models

Apart from statistical analysis, we trained three different regression models namely Lasso, KNN, GradientBoosting without any dimensionality reduction or hyper parameter optimization. Obtained RMSE scores for predictions by these models. Then performed PCA for dimensionality reduction and GridSearchCV for hyper parameter optimization, obtained RMSE scores for these and it was evident that the RMSE scores for the plain regression models were greater than the regression models with PCA and hyper parameter optimization. The RMSE scores are included in the figures.

An interesting observation was that for MA dataset KNN regression model with hyper parameter optimization and PCA performed better in comparison to other models. But for NY dataset, GradientBoosting regression model outperformed both KNN and Lasso in terms of forecasting. The understanding from this difference in the regression model can be due to the level of outliers and noise which were created based on the way the missing values were handled.

After identifying the best model for each dataset manually, we generated three plots namely 'Timeline vs MSP', 'Timeline Forecast vs MSP' and 'Timeline Forecast vs RMSE' with the corresponding regression model that produces lower RMSE for each dataset. The first plot 'Timeline vs MSP' is obtained from initial prediction done by passing the test feature set. The 'Timeline Forecast vs MSP' is obtained by using the initial predicted target labels as history to forecast MSP for next two months i.e. 14 days recursively. This is the second novel part of the project where we use the first prediction of timeline t and t+1 to forecast MSP from t+2 to t+14 time period recursively using the forecasted value in every step as the history for the next forecast.

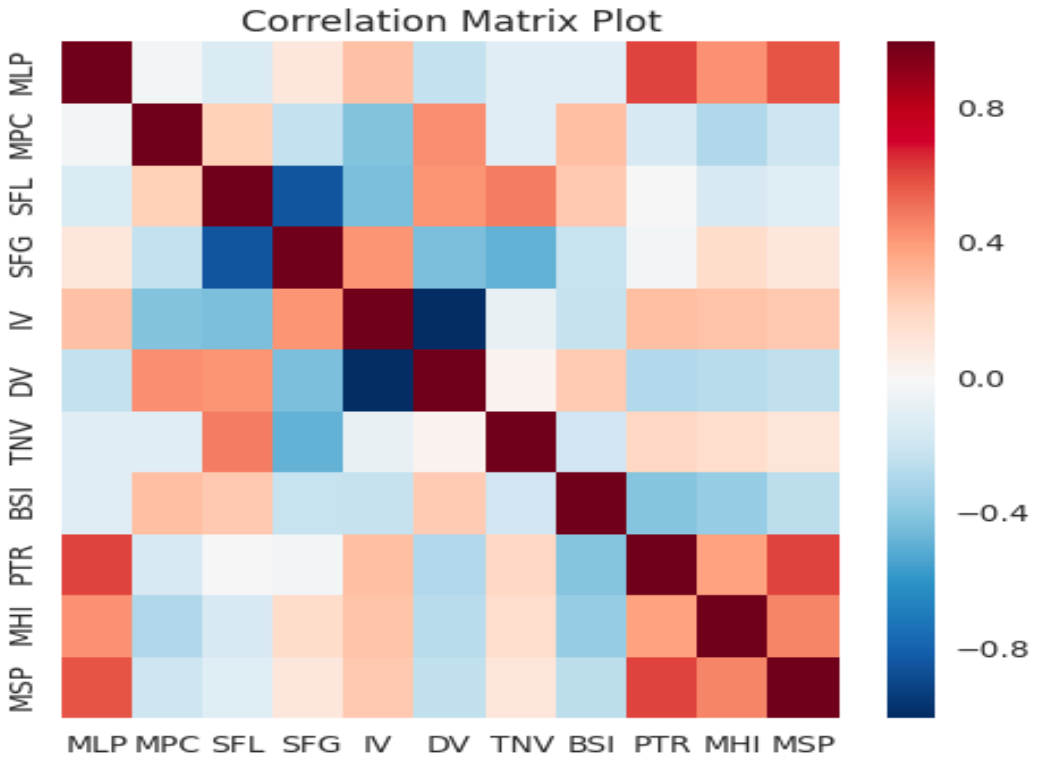
5.4 Plots and Figures

The various plots and figures generated and studied from are given below.

MA Dataset

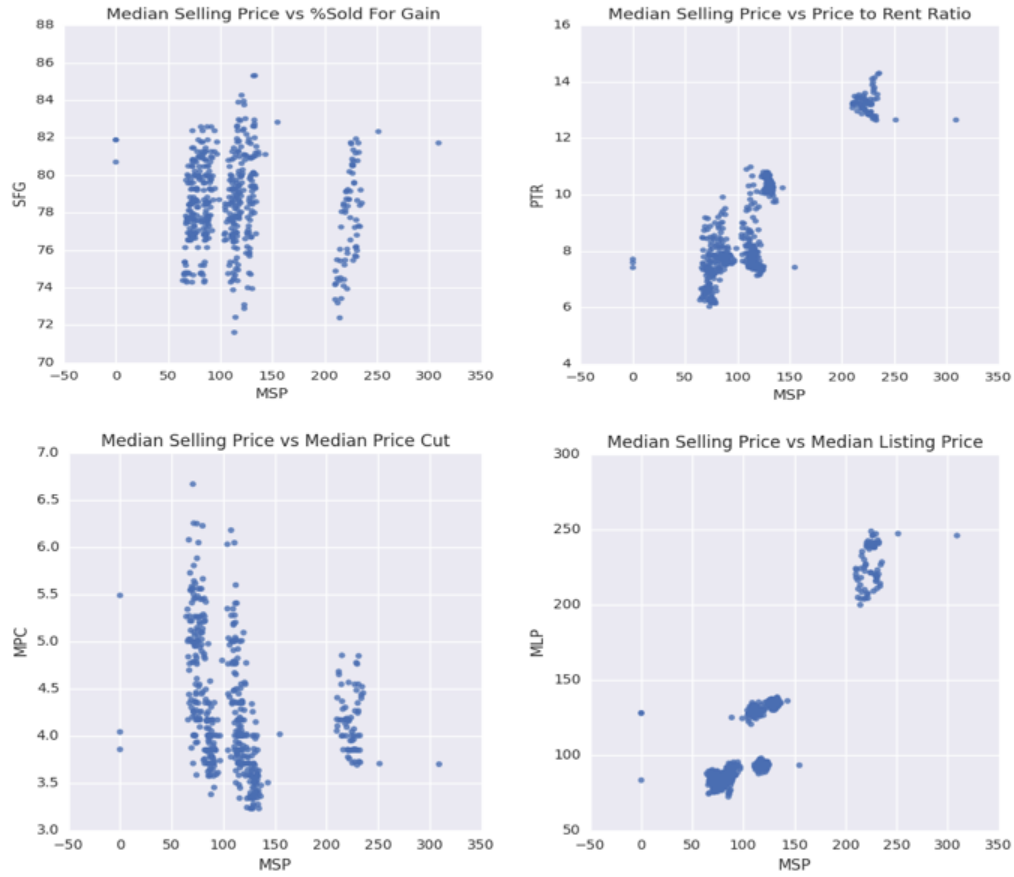

Median Selling Price vs Median Price Cut
MPC
MSP
Median Selling Price vs Market Health Index
MHI
MSP
Median Selling Price vs % Increasing Value
IV
MSP
Median Selling Price vs Buyer Seller Index
BSI
MSP

246


Correlation Matrix Plot
MLP
MPC
SFL
SFG
IV
DV
TNV
BSI
PTR
MHI
MSP
0.8
0.4
0.0
−0.4
−0.8

247

NY Dataset



5 Discussion and Conclusions

In previous works done on housing data, multivariate regression was used almost negligibly to none. Most works featured univariate models and the usage of multi output regressor was never explored. These unexplored things were experimented through our project.

Based on our project experiments on two different datasets, we inferred that every regression model performs differently for different datasets. In general ensemble methods are said to perform well compared to regular regression models. But in here interestingly we observed that KNN outperformed GradientBoosting with respect to MA dataset. This behavior might be influenced by the way missing values are being substituted with.

Looking at the RMSE of recursively forecasted MSP values; the expectation was for the

error score to go up. Surprisingly, for both datasets, the RMSE reduced as we increased the number of forecasts ahead. This can be attributed to the independency between the target labels assumed in the multi output regression.

The biggest challenge in this project was creating the dataset. None or not even part of the data was readily available. We had to create by putting together lot of elements of the dataset from scratch. Also, when we decided to convert the time series into a supervised learning model, we faced lot of issues during the course of the project.

With more time, we would have tried to deploy the ARIMA models which we were facing issues with for this multivariate time series forecast dataset.

6 Future Scope

The future scope of the project experiment would be to explore further models that can perform better to predict multiple target labels with lesser history.

References

[1] Jiron Gu, Mingcang Zhu, Liuguangyan Jiang (2005) *Housing price forecasting based on genetic algorithm and support vector machine*.

[2] Yonas B. Dibike, Slavco Velickov, Dimitri Solomatine, Michael B. Abbott, *Model Induction with support vector machines: Introduction and Application*.

[3] B. Park, Jae Kwon Bae, *Using Machine Learning Algorithms for housing price prediction: The case of Fairfax county, Virginia housing data*.

[4] https://www.hindawi.com/journals/aaa/2014/648047/

[5] http://www.doc.ic.ac.uk/~mpd37/theses/2015_beng_aaron-ng.pdf

[6] D E Rapach, JK, Strauss, *Differences in Housing Prices Forecastability acorss US States*

[7] ECM Hui, S Yue, *Housing Price Bubbles in Hong Kong, Shanghai, Beijing, A Comparative Study*

[8] N Miller, L Peng, *Exploring Metropolitran Housing Price Volatility*

[9] http://zillow.com