**Predicting Housing Prices Using Machine Learning Models**

Paul Stevens

November 21, 2020

**Abstract**

The purpose behind this study of machine learning is to get a general idea of an end-to-end machine learning process. It explains the major components of examining data, preprocessing the data to prepare for the training of models, and finally the evaluation of the performance of the models. In this study, a set of housing data which included 19 features was used to train three different linear machine learning algorithms and the resulting models were then subject to three different performance evaluations. Although the data showed a linear correlation between different features and the target price feature, the performance evaluations of each model weren’t able to provide a strong enough outcome to suggest that they could be used to predict future prices of houses.

**1 Introduction and Background**

Machine learning is a subset of Artificial Intelligence and can be broadly defined as the computational methods using experience to improve performance or to make accurate predictions. In this case, experience refers to the past information which is available to the learner (Mohri et al., 2018, p. 3). The study done on the housing data in this report can be classified as a supervised learning machine learning system. The term supervised learning system comes from the fact that all training is done with many examples of data including both their features and their target labels. This is a form of supervised learning known as model-based learning where a model is trained and then is used to make predictions.

The model trains itself with the different features and builds a hypothesis function that maps each feature to a coefficient. These coefficients are what is used by the predictor to calculate future prices. The linear regression algorithms that are used to train our models will have their performance based on cost functions which determine how bad their predictions are. The cost function evaluation measures the distance between the model’s predictions and the actual training examples (Géron, 2019, p. 20). Observing each of the linear regression model’s performance we can then compare the models and start to formulate a conclusion to the study.

**1.1 The Problem You Tried to Solve**

The study was focused around trying to use linear regression algorithms to train a model on a set of housing data collected from King County, Washington from May 2014 – 2015. The problem of the study was to use the model that was trained from the existing data to predict the prices of future houses that are to be sold. The assumption was that if there was a clear linear correlation between the features in the data and the target price label then there could be some linear regression model that could be trained that would produce a hypothesis function that would map future house features to a price. If a suitable model could be trained to predict house prices with minimal error then the model could be released for public use in that specific country to help potential house sellers to get an estimate on how much their house could possibly be sold for.

**1.2 Results from the Literature**

The first book used as a reference source for this project is a machine learning textbook titled, “*Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*”. The books provides a guide of how end-to-end machine learning projects are conducted. Specific attention paid to the chapters in this book were to chapters 1, 2, 4, 6, and 7. The second book used as a reference source for model selection is titled, “*Foundations of Machine Learning*”. Chapter 11 focuses on regression in general which provided information on the hypothesis functions. It also gives a general discussion on the model-based learning algorithms used in linear regression.

**1.3 What Tools and Programs Are Already Available for the Problem, or for Closely Related Ones?**

Zillow provides a service which is a home valuation model that estimates a home’s market value. The “Zestimate” incorporates public and user-submitted data, taking into account home facts, location, and market conditions when providing this estimate. The Zestimate accounts for different variables such as the characteristics and unique features. Zillow uses a sophisticated and proprietary algorithm that incorporates data from county and tax assessor records and direct feeds from hundreds of multiple listing services and brokerages (Zillow, n.d.). The model used by Zillow is what this project tried to replicate by on a smaller scale.

The models trained using machine learning could be used in conjunction with this service to help validate the Zestimates from Zillow on a particular house. Features can be fed into the machine learning model for a house that is listed on Zillow to see the difference price estimates between the model and the Zestimate. Although the model trained in this project is specific to one county, different models would need to be trained a specific county before you could try and use it to predict housing prices there.

**2 Data Collection**

The data was brought to my attention while searching for dataset and was downloaded from [https://geodacenter.github.io/data-and-lab//KingCounty-HouseSales2015/](https://geodacenter.github.io/data-and-lab/KingCounty-HouseSales2015/) and was collected from May 2014 – 2015 and has the home sale prices and features for 21,613 houses in King County, Washington. These features include:

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **Description** | **Feature** | **Description** |
| Price | Sale price | Sqft\_above | Square feet above ground |
| Bedrooms | Number of bedrooms | Sqft\_ment | Square feet below ground |
| Bathrooms | Number of bathrooms | Yr\_built | Year built |
| Livingsq | Size of living area in square feet | Yr\_re\_ated | If renovated, the year it was done |
| Sqft\_lot | Size of lot in square feet | Zipcode | 5 digit zipcode |
| Floors | Number of floors | Lat | Latitude of the house |
| Waterfront | Waterfront property or not | Long | Longitude of the house |
| View | Index of how good the view on the property is | Sqft\_ng15 | Average size of interior living space for the closest 15 houses |
| Condition | Index of the condition of the house | Sqft\_lot15 | Average size of the lot for the closest 15 houses |
| Grade | Index of the types of materials used and the quality of the construction of the house |  |  |

**3 Overview of the Architecture**

The project is programmed in Python 3.X using the Anaconda scientific distribution which includes support for running Jupyter notebooks and using the Scientific Python Library compute stack. Scikit-Learn is a python library that provides the efficient tools, and predictive data analysis functions that are built on NumPy, SciPy, and matplotlib. All of the function calls are done with the Scikit-Learn. It is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. The instances of data and the data target labels are housed in Pandas DataFrames and NumPy arrays.

**3.1 Finished Work: Running Modules**

The running modules in our notebook compose a series of steps that guide the project through a machine learning system. The data is loaded into a Pandas DataFrame then we use matplotlib to visualize the data in various forms. A correlation matrix is calculated against our target label which shows how a change in each feature might have influence over a change in the prediction price. The next few modules then show how data is preprocessed by dropping features that won’t be useful to training the model and then splitting the data into a set of training instances and testing instances. The last half of the modules in the notebook cover the training of three different models with various linear regression algorithms. The models are then evaluated using three different performance measures and results the model’s evaluations are then compared.

**3.2 Work In Progress: Modules Designed But Not Implemented**

A number of modules didn’t make the final notebook that were programmed and tested but that didn’t provide any significant improvement to the models. During preprocessing of the data, both normalization and standardization techniques were applied to the data features to help improve the scale and standard deviation but both techniques showed no improvement. Modules were also designed to include Ridge or Lasso regularization on the models. These techniques that are used to slightly tune models to help improve the performance would not have netted any significant change to the final performance measure.

There was a design for a k-fold cross-validation performance evaluation of each model and although it was implemented for just one of the models. The results were of the evaluation were so for from the other performance measures that there was no need to run them on the other models. This performance measure can be brought back to evaluate future models that look promising.

**3.3 Future work: Modules a Future Continuation May Have**

There is a lot of modules that are included in the notebook that pave the groundwork for much continuation in the project. Mainly those that are related to data exploration and visualization which might show promise in a decent model especially if it turns out that the data isn’t linearly correlated. The exploration and cleaning of the data is the part of a machine learning system that takes the longest to achieve desired results. If the project had more time to complete, these is where we could search for the most growth and improvement.

**4 Results and Evaluation**

The models that were trained in the project were evaluated using three different performance measures. The R-squared score is a measurement of how scattered the predicted prices are the fitted linear hypothesis function. A higher R-squared score represents an overall smaller difference between the predicted prices and the fitted values. The scores are interpreted as a percentage of the variance in the dependent variables that model can explain (Hamilton et al., 2015). This performance measure alone is not a good evaluate of a model which is why two others are used in conjunction.

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are two different computations but can be thought of as a single performance measure. The MAE is the sum of the difference between the distance between the model’s predictions and the fitted hypothesis function values (Géron, 2019, p. 39-40). While MAE gives us the absolute error, RMSE squares all of the residual values and can give a better picture of the performance of the model. The sum is made over all of the squared residual values so that under-predicted prices add on to the error instead of working against over-predicted prices like with the MAE performance measure.

Lastly, 95% confidence intervals can give real values to how such a model can predict a price. These intervals compute with 95% certainty that a predicted price will be within a generalization error range and do no worse that the upper bound of that range (Géron, 2019, p. 80). Combining these three performance measure can give a generalized view of how well the models are performing when faced with data it hasn’t seen before.

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Performance Measure** | **Result** |
| Linear Regression | R-squared Score | 0.6955638499670669 ≈ 69.6% |
| MAE/RMSE | 128673.0942688452 / 214531.24773355035 |
| 95% Confidence Interval | (193852.91520895 - 233384.57483994) |
| DecisionTree Regressor | R-squared Score | 0.7115009595915565 ≈ 71.2% |
| MAE/RMSE | 105698.45327318991 / 208840.45685213033 |
| 95% Confidence Interval | (171272.3383085 - 240612.67416067) |
| RandomForest Regressor | R-squared Score | 0.8510543751897715 ≈ 85.5% |
| MAE/RMSE | 73710.66701127964 / 150057.02204113491 |
| 95% Confidence Interval | (123850.01432183 - 172323.51458866) |

The RandomForest Regressor showed the best performance out of the three models with a big performance improvement in the RMSE over both Linear Regression and DecisionTree Regressor. Further improvement on the RandomForest Regressor may not actually be possible with this dataset without introducing some bias into the model which would not be optimal. The 95% confidence intervals of the best model still shows that a prediction of a house for sale would do no worse than an error of $172,323.

**5 Discussion and Conclusions**

A machine learning project is an ever-evolving process of testing and tuning the model before and after the system is put into production to keep it at an optimal performance level. This models that were trained and tested on the housing data set did give us measureable performance evaluations such that we were able to determine that there was one model that outperformed the other two. The three performance measures that were used in evaluating the models showed clear improvement over all the measure in the RandomForest Regressor model.

Further investigation to the best model could prove that it is not powerful enough to perform to the level that is needed when tuned correctly. The next steps to take if the project is continued is to evaluate the data set to determine any further linear or non-linear relationships. A polynomial fit between the features of the data and the target label might reveal that there a polynomial fit. Introducing a polynomial set of features should improve the model but at the cost of introducing variance and overfitting. Increasing the degree complexity of a model lets the model conform too easily to the data (overfitting) and will produce good training performance measures but will perform poorly on the unseen portion of the data.

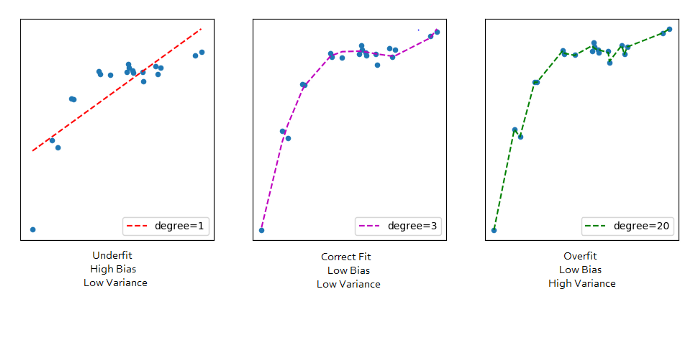


Figure 1 - Examples of increasing the degree complexity of a model and the problems they pose.

Achieving a polynomial fit using a linear regression model gives the hypothesized function more room in fitting the data to a better predictor function. This can be done by scaling the features to some determined n-th degree. Scaling the features will transform the original features into polynomial features such that features x1, x2, x3, etc…, will become x12, x1x2, x22, etc…, then normal linear regression is performed on the new set of features. The next modules to be designed would be to go back to the data to dive deeper into correlation and to scale to polynomial features to simply test if a model will perform on the level of the RandomForest Regressor.

**6 References**

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). Sebastopol, CA: OReilly Media.

Hamilton DF, Ghert M, Simpson AH. Interpreting regression models in clinical outcome studies. *Bone Joint Res*. 2015;4(9):152-153. doi:10.1302/2046-3758.49.2000571

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning* (2nd ed.). Cambridge, MA: The MIT Press.

Zillow, I. (n.d.). What is a Zestimate? Zillow's Zestimate Accuracy. Retrieved November 21, 2020, from <https://www.zillow.com/zestimate/>