Predicting Melbourne House Prices Using Ridge Regression and Decision Tree Regressor

1. Introduction

The aim of this project was to predict housing prices in Melbourne. An accurate price prediction is important for anyone looking to buy or sell a house, as well as anyone else participating in the real estate market, such as tax assessors, investors, and mortgage lenders. Accurate predictions may help individuals make better investment plans and help influence policy.

The formulation of the problem is explained in section 2. Section 3 contains information on the sourcing and preprocessing the data, model choice and evaluation. The results and conclusions are in sections 4 and 5, respectively.

2. Problem Formulation

Predicting house prices is a regression problem. Each data point represents the transaction of selling a property.

The label of a data point is the price the property was sold for in Australian dollars.

The 18 features used were:
Suburb
Number of rooms
House type(house, unit, or townhouse)
Selling method
Real estate agent
Date sold (in months since start of 2016)
Distance from central Melbourne
Number of bathrooms
Number of carspots
Land Area
Building Area
The year the house was built
Governing council for the area
Latitude
Longitude
Region
Number of properties that exist in the suburb

3. Method

The data was obtained from the full dataset of the Melbourne Housing Market dataset in Kaggle[1]. The original dataset had 21 columns and 34,857 rows.

The data was preprocessed and the models were built using python pandas[2] and scikit-learn[3] libraries.

Data points where data was missing from any of the columns were removed, as well as columns containing the address, postcode, and number of bedrooms from a different source (called “Bedroom2” in the original dataset). The date column was edited to contain the number of months since the start of 2016. The categorical variables (suburb, house type, selling method, real estate agent, governing council, and region) were one-hot encoded using the pandas get_dummies function. The remaining numeric variables were scaled using the scikit-learn StandardScaler function. The resulting dataset had

The dataset was split randomly into training and validation sets using the scikit-learn train_test_split function with 30% of the data reserved for validation. The resulting training set had 6,665 data points and the validation set had 2,222 data points.

The considered classifiers were ridge regression (with l2 regularization), knn regressor (with k=10, using minkowski distance), and decision tree regressor (using mean squared error to minimize loss). They were evaluated with mean square error and R^2 scores using 10-fold cross-validation. After this, the best performing algorithms were selected to be trained on the whole training set and evaluated using the training and validation sets.

4. Results

The performances of the predictive models in cross-validation were as follows:

Model	MSE	R^2
Ridge	0.35	0.67
10-nn	0.51	0.52
Decision Tree	0.34	0.68

Since ridge regression and decision tree had close to equal performance, both were chosen to be trained using the whole training dataset. These models were then evaluated on the test set and on the validation set, with results as follows:

Model	Training set	Validation set
Ridge	R^2 = 0.75 MSE = 0.27	R^2 = 0.75 MSE = 0.21
Decision tree	R^2 = 0.98 MSE = 0.02	R^2 = 0.64 MSE = 0.31

5. Conclusions

It seems that ridge regression is most suitable to make predictions on this data. Decision tree also seemed to give good results, but results on the test set were much better which suggests that the model is overfitted.

Improvements might be made by reducing the number of features. There were many features related to the house location which correlate with each other, and having all of these present may worsen prediction performance. Another way to improve predictions might be missing data imputation, as a lot of data points were removed due to a few missing features. Additional improvements could be made with hyperparameter tuning.

References

[1] data: https://www.kaggle.com/anthonypino/melbourne-housing-market?select=Melbourne_housing_FULL.csv

[2] McKinney, W. & others, 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*. pp. 51–56.

[3] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.