Practical 1 – ID5059

Date: 5th March 2021

Student Id: 200028225

The objective of this project is to construct reasonable machine learning models, evaluate them and measure their performance. The UK government land registry dataset from 1995 to 2020 was being used to predicts the selling price for the property. The aim is to focus on the main 3 features such as estate type, property type, and whether or not a property is in London. The subsample of the original dataset was made by randomly taking the 3% observation from each year to maintain the consistency of the number of observations each year as the original dataset. The subsampling helps to reduce the computational expense and still provides similar results as the original dataset. The unwanted columns were being removed from the dataset while creating subsampled data and stored as a .csv file for further analysis.

The data from 1995 to 2016 years was being used as training data, whereas the years 2017 to 2020 as test data. As property type (D- detached, S- semi-detached, T-terraced, F-flat/maisonette, O- other), estate type (F- freehold and L- leasehold), and location are a categorical attribute. Thus, binary encoding (if a location is London then 1, else 0) was done for the location column. Whereas the one-hot-encoding was done for the property type and estate type columns using pd.get_dummies function. The estate type column has one addition type called 'U', it has been removed because it is an unidentified value for the estate type. The data does not have any null values, it had been checked using the isnull() function. The correlation matrix shows a positive correlation with the 'inLondonOrNot' attribute meaning the property price goes up if the property is in London. And negative correction with property type T meaning property price goes down with terraced houses.

The encoded data was being fitted in 6 different types of models to achieve sensible performance. The fitted models are linear regression, decision tree regression, random forest regression, lasso regression, Elastic net regression, and Ridge Regression. None of the model able to make accurate predictions but the decision tree and the random forest can make somewhat close predication to labels. The table shows that all models have quite high root means square error (prediction error). which means that all models are underfitting and features (property type, estate type, and property in London or not) are not providing enough evidence to make good predictions. The same results were found with 5-fold cross-validation (refer to table), all models are performing equally worse. However, the decision tree and random forest model is performing slightly better than the others, so both models were short-listed for fine turning with different parameter settings. Fine-tuning models show even worse prediction errors (around 410000£). Seems like I have done a lot of hyperparameter tuning. So, the random forest regression was taken as the final model to do the prediction for test data, because it has the lower RMSE scores using CV compare to others and it has done better predications than the other models. As expected, the final model did not perform well on testing data. It gave the prediction error of 1645210.0£ for predicting the selling price for a property, which is worse. While comparing with the mean of labels it gives around 470%, which is horrible. The random forest model did not predict well on unseen data at all. The error distribution is somewhat normally distributed and slightly right-skewed.

Fitted regression models	RMSE	Mean of RMSE using CV
Linear	403433.0£	300067.0£
Decision tree	397189.544£	298864.0£
Random forest	397217.666£	298920.0£
Lasso	403432.681£	300069.0£
Elastic net	415792.063£	302596.0£
Ridge	403432.682£	300067.0£

References:

- https://scikit-learn.org/stable/modules/model_evaluation.html
- https://towardsdatascience.com/what-is-one-hot-encoding-and-how-to-use-pandas-get-dummies-function-922eb9bd4970
- https://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_and_elasticnet.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html