**MSDS\_422 - Practical Machine Learning**

**Module 2 – Evaluating Regression Models**

**Introduction**

The Ames Housing Dataset compiled by Dean De Dock is a rich dataset that contains over 79 explanatory variables that describe almost all the aspects of residential homes in Ames, Iowa. Traditionally, you could use some basic features known about homes to estimate its value, but this is often unreliable since these features while important, do not describe all the aspects of a home that can contribute to the value.

To get a better sense of the value of a home we can employ machine learning techniques to predict the value and even create additional features based on the 79 provided in the dataset. This is a more modern approach and can reduce the overall cost to manually gather and analyze the value of homes. In addition, as more data is collected about the housing market the model we employ will improve and can be calibrated to provide a more accurate prediction.

**Research and Design Method**

The dataset contained 79 features about the residential homes and the sale price. First, I reviewed the distribution of sale prices and found that the variable we were trying to predict was skewed to the right. In addition, the kurtosis and skewness were not close to zero, so I used a log transformation (numpy log1p) to improve it to be normally distributed.

Next, all the numerical features were plotted against the sale price using a scatter chart which identified several outliers. The above grade living area square feet (GrLivArea) variable had a positive correlation but a few outliers existed when the value was greater than 4500. First floor square feet (1stFlrSF), Type 1 finished square feet (BsmtFinSF1), and total square feet of basement area (TotalBsmtSF) had a similar trend and these outliers were removed.

After removing the outliers, it was identified that there were several missing values and they were either imputed using the ‘None’ if it was categorial or 0 if it was numerical. In some case such as kitchen quality (KitchenQual) I decided to use the mode if it was categorial or median if it was numerical.

Lastly, to prepare the dataset for the machine learning model, I used a Box-Cox transformation to better distribute skewed data greater than 0.5 and created several new features such as total square feet, total bathrooms, has a pool, etc. that was based on the existing 79 features. After completing the feature creation and transformation, the last thing I did was use the numpy get dummies function to translate the categorial features into columns with a 1 for yes or 0 for no. These transformations created over 200 features and several good correlations to the sale price that the model used to make the prediction.

**Machine Models Used in the Research**

To identify the best model to predict the sale price of homes I compared linear, ridge, lasso, and elastic net regression models to predict the sale price. The lasso model used several alphas and the elastic net model used several alphas and l1 ratios to compute the accuracy, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). The best model with the lowest RMSE score on the test data in Kaggle.com was the elastic net model using an alpha of 0.0007 and l1 ratio of 0.95 and achieved an RMSE score of 0.12420. This was followed by lasso with a 0.12434, ridge with a 0.12591, and the standard linear model did not perform well and was not submitted.

After doing additional research I found several other models: Support Vector Machine (SVR), Gradient Boosting Regressor, Light GBM, XGBOOST, and StackingRegressor. These models blended together with ridge, lasso, and elastic net gave the best prediction achieving a RMSE score of 0.11997. More details on the additional models can be found in the appendix.

**Conclusion**

My recommendation to management is to move ahead with using the blended model developed in this study to predict the sale price of homes in Ames, Idaho. If the blended model is not an option, I would recommend elastic net, then lasso, and finally ridge. The RMSE using the blended approach is good enough and would provide a more cost-effective approach than manually doing the research to identify the values of homes. Lastly, the funds used to do the research can be redirected to collect more data about homes in Ames to improve the model since the model will learn and become more accurate with more data.

For further information on this study you can find the full source code and approach on GitHub: <https://github.com/chrisfesta/NWU_MSDS422>

**Supporting Material**

**Model comparison**

|  |  |
| --- | --- |
| **A close up of a map  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |

**Appendix**

**See Jupyter Notebook for the research and analysis**

**Best score on Kaggle**

UserID: Chris Festa / <https://www.kaggle.com/csfesta>

A screenshot of a cell phone

Description automatically generated

Additional score can be found in the conclusion in the Jupyter Notebook

**Additional Regression Models**

* Support Vector Machine - Regression (SVR): Epsilon-Support Vector Regression <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>
* Gradient Boosting Regressor: GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>
* LightGBM: is a gradient boosting framework that uses tree based learning algorithms <https://lightgbm.readthedocs.io/en/latest/>
* XGBOOST: is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. <https://xgboost.readthedocs.io/en/latest/>
* StackingRegressor: Stacking regression is an ensemble learning technique to combine multiple regression models via a meta-regressor. The individual regression models are trained based on the complete training set; then, the meta-regressor is fitted based on the outputs -- meta-features -- of the individual regression models in the ensemble. <http://rasbt.github.io/mlxtend/user_guide/regressor/StackingRegressor/>