APPLIED ARTIFICIAL INTELLIGENCE

CS 514

PROJECT 5

HOUSE PRICES : ADVANCED REGRESSION TECHNIQUES

House Prices : Advanced Regression Techniques

Username : DarkVoid
Language used : Python
Packages used : Pandas, Numpy, Scikit-learn, Matplotlib
Kaggle score : 0.13066

This project is my attempt at the Kaggle competition – 'House Prices : Advanced Regression Techniques'.

First, the training and testing datasets were loaded into Pandas dataframes for exploratory analysis. The contents were examined using the *head()* method which shows the first few rows of the table. The skewness of the *SalePrice* variable was determined to check the linearity of the distribution and improve it if the data was too positively or negatively skewed. In this case, the data was positively skewed and so logarithmic transform needs to be applied on the *SalePrice* attribute to reduce the skewness (skew value closer to 0 is desired).

Then, the features most positively correlated with the *SalePrice* were determined to see how they influence its value. These features were plotted against the *SalePrice* to investigate if there are any outliers that could tamper with our model, and these outliers were eliminated.

The next step was to engineer features for the learning model from the non-numeric (categorical) variables. This can be done by assigning numerical values in an intuitive way to the values under the features. So, the features with just two unique values were encoded as 1 or 0 based on the most dominant value with respect to *SalePrice*. For ordinal features, numbers were assigned corresponding to their relative ranking among themselves. E.g., for the *KitchenQual* feature, *Excellent* was assigned 4, *Good* 3 and so on. Likewise, many other features were engineered to improve the model.

Before building the model, the *NaN* values were interpolated with 0 and records missing values were dropped. The model used is the Ridge Regression model with regularization factor *alpha*. The feature vector *X* for the model is the training dataset with the target variable (*SalePrice*) and *Id* column removed (as it does not have any bearing on *SalePrice*). The target variable *Y* for the model was the log-transformed *SalePrice* data. Using the *train_test_split()* method, a training set and hold-out set were created, with corresponding target sets for each. Out of these, *x_train* and *y_train* were used to fit and train the model. The *x_test* and *y_test* could be used to test the model and calculate the RMSE value for evaluating the model.

The test dataset was transformed in the exact same way as the training dataset, so that the shape of the feature vectors are identical. With a suitably fine-tuned value of *alpha*, the model was run on the test data to generate a set of predictions for *SalePrice* for each of the records. Since the target variable for training was in log-space to begin with, the predicted output was also log-transformed. It was converted back into dollars using the *exp()* method. The output was attached to the Id column to generate the final dataframe, which was converted into a CSV file for submission.

List of features engineered

The following features were engineered to improve the performance of the regression model (Descriptions obtained from text file). As mentioned above, all the feature engineering carried out on the training data was re-applied on the test data.

1. Street : Type of road access to property
2. Alley : Type of alley access to property
3. Utilities : Type of utilities available
4. SaleCondition : Condition of sale

5. SaleType : Type of sale
6. CentralAir : Central air conditioning
7. RoofStyle : Type of roof
8. Condition1 : Proximity to various conditions
9. LotShape : General shape of property
10. KitchenQual : Kitchen quality
11. New house : House is considered ‘new’ if it was sold in the same year as it was built in