1836702 Nguyen Phu Hien Team name: We train on test set for FDS

1735340 Elshibiny Amr Score: .11180

Data tidying:

1. Remove outliers from the columns with most outstanding outliers, which were found to be SalePrice and GrLivArea.
2. Take the log of the label “SalePrice” in order to avoid the effect of the right skewness in the data.
3. Deal with the NAs in the dataset:
   1. Some of the missing data did not actually represent missing data, just a service or so that isn’t available in this house. The NAs here were substituted by a dummy variable.
   2. Others indicated zero (like zero area for basements in case the house doesn’t have one)
   3. Others were data that is actually missing, in which case we filled the with the mode or median of the data.

Feature engineering:

1. Convert some of the variables that were misclassified as numerical into categorical, such as years or MSSubClass.
2. Encode all categorical variables in order for it to be more easy to use for the models.
3. Created new feature:
   1. TotalBsmtSF + 1stFlrSF + 2ndFlrSF
4. Used boxcox transform on all features with skewness above .75 (59 in total)

Model Building:

* 1. Out of fold prediction (Python):
     1. We trained a large models with the training data and picked the ones with performance < .13 . Models chosen: Huber, Ridge, Bayesian Ridge, Gradient boosting, XGB regressor, Kernel Ridge, Lasso, Elastic net.
     2. New features were created where the model predictions on the training set served as features themselves.
     3. We then used the library stacking regressor to make out of fold predictions based on the previous data we created.
  2. Stacking regression:
     1. Stack the regressors of Kernel Ridge, GBoost, lasso, Huber, Bayesian Ridge and Ridge with Kernel Ridge as meta model using the library mlxtend.regressor. (this step gives different score to the oof prediction above)
     2. Predict the accuracy score on the train set
     3. Decide to leave out extreme gradient boost and light gradient boost, just using the stacked regressor.
  3. Mice + svm (R):
     1. We loaded our dataset and used mice package on it, which is a package that models the best replacement value for missing data against a model of our choice on a variable by variable basis. We made mice run based random forest modelling. (This idea is adopted from another kernel)
     2. Next, we used the SVM model to make our predictions on the resulting data.
  4. Combining: We combined the two previous models for our final predictions with:
     1. 60% weight for out of fold prediction
     2. 20% for stacked regressor
     3. 20% for mice+SVM model