1603090 Cannistraci Irene

Team name: Irene Cannistraci
Score: 0.11309

Language: Python

Data tidying

1. Removed outliers
2. Replaced the following values of Exterior2nd with the correct value ['CmentBd': 'CemntBd', 'Wd Shng': 'WdShing', 'Brk Cmn': 'BrkComm'].
3. Replace nan values of the following categorical features values with with the correct value (e.g for Alley I used 'NoAlleyAccess'), since in these cases nan doesn't mean missing value [Alley, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, Fireplaces, FireplaceQu, GarageType].

Feature engineering

I decided to merge the train set and the test set and to analyze, starting from the data description file, every single feature in order to understand what was the best way to treat them.

Missing values:

1. Most of the nan values were wrong values that I previously adjusted (explained in Data tidying).
2. Some features like MSZoning, LotFrontage, BsmtFullBath and BsmtHalfBath were replaced with mode and mean.
3. All the other values were replaced using a different criteria based on the specific feature. In most cases the case I used the most frequent value but for example for the set of features about Garage I used to put them together and check if the nan values of them means that they have no garage, so in these cases I filled the missing field with 0 (or with the corrispective categorical value).

New features:

1. I created many boolean features (0,1). For example the feature '2ndFlrSF' if is equal to 1, the corrispective boolean feature 'Has2ndFlr' will have value 1, otherwise 0. [HasAlley, HasRegularShape, HasCondition2, IsRemodeled, HasMoreMaterials, HasMasVnr, HasMultipleBsmtType, HasBsmt, Has2ndFlr, HasFireplaces, HasGarage, HasPavedDrive, HasWoodDeck, HasOpenPorch, HasEnclPorch, 'Has3SsnPorch', 'HasScreenPorch', HasPool].
2. I created new numerical features like TotalSF that represents total floor square feet [TotalSF, FoundationQual, FunctionalQual]
3. I created numerical features for the one that had both a value for quality and condition. I did the mean between them [OverallQualCondMean, ExterQualCondMean, BsmtQualCondMean, GarageQualCondMean].

Categorical features:

1. I transformed features with only 2 possible values into boolean (0,1) [Street, CentralAir]
2. I transformed features that represents a 'vote' into numerical features (ordinal) where every value had a corresponding integer that is the same for all these features [Po: 1, Fa: 2, TA: 3, Gd: 4, Ex: 5] and [Unf: 1, LwQ: 2, Rec: 3, BLQ: 4, ALQ: 5, GLQ: 6] where increasing value means better.
3. Other categorical features were transformed using one hot encoding (pandas getDummies). Then I removed values 'Metal', 'Membran' and 'Roll' of 'Roof Matl', 'Mix' of 'Electrical' and 'OthW' and 'Floor' of 'Heating' feature since there are no rows with that values in the test set.

Deleted features:

1. I deleted the following features: LotShape because I replaced it with HasRegularShape and PavedDrive because I replaced it with HasPavedDrive, Utilities since 2908 rows over 2909 had the same value (AllPub) and PoolQC since 2902 rows over 2909 had null values.

Training

1. Scaled data with RobustScaler from sklearn.preprocessing.
2. Prediction with StackingCVRegressor from mlxtend (`http://rasbt.github.io/mlxtend/user_guide/regressor/StackingCVRegressor/`) with Lasso, GradientBoostingRegressor, ElasticNet, Ridge, BayesianRidge and RandomForestRegressor as meta regressor. Using a 20-Folds cross-validation.
3. Prediction with XGBRegressor
4. Mean between the two regressors attributing 77 percentile to the StackingCVRegressor prediction and 23 percentile to the XGBRegressor prediction.