1595881 Maiorca Valentino
1594551 Moschella Luca

team name: Maiorca_1595881_Moschella_1594551
score: 0.11246

Language: Python

Data tidying:

1. Removed outliers
2. Corrected typos in the values of the Exterior* features, replaced ‘CmentBd’, ‘Wd Shng’, ‘Brk Cmn’ with ‘CemntBd’, ‘Wd Sdng’, ‘BrkComm’.

Features engineering:

First of all, the general approach has been to consider the features one by one with dedicated utilities functions we wrote to explore their distribution and other aspects.

1. The missing values are handled in different ways. If it is possible to deduce the missing value from the context it is done, if the missing value is from a categorical feature the most frequent value is used and, finally, if the missing value is from a numerical feature we use a 10-NN approach (with the package fancy impute https://pypi.org/project/fancyimpute/).
2. Created many boolean features with a self-explanatory meaning and source feature: 'HasAlley', 'IsGoodNeighborhood', 'IsRemodeled', 'IsRemodelRecent', 'IsNewHouse', 'IsBsmtFinType1Unf', 'IsBsmtFinType2Unf', 'BsmtIsPresent', CentralAir, '2ndFloorIsPresent', ‘HasFireplace’, 'GarageIsPresent', 'HasWoodDeck', 'HasOpenPorch', 'HasEnclosedPorch', 'Has3SsnPorch', 'HasScreenPorch', ‘HasPool’, 'HasShed'
3. Created some numerical features which have been useful to improve the score (features ending with ‘_int’)
4. Created ‘TotalArea’, a feature which is the sum of all the area features in the dataset: 'LotFrontage', 'LotArea', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'LowQualFinSF', 'PoolArea'
5. Created a new feature to keep count of the bathrooms available (half bathrooms are counted as 0.5)
6. Merged a couple of features into a single one: (‘Condition1’, ‘Condition2’) -> ‘Condition’. This has been possible due to their values and has improved the score too.
7. Deleted some features because of their distribution (their values are completely irrelevant since they’re almost all missing or they are almost always the same)
8. Added some features as a combination of a ‘Quality’ and ‘Condition’ feature with the same meaning (i.e. ‘GarageCondition’, ‘GarageQuality’ -> ‘GarageCondQual’)
9. Categorical features (with no intrinsic order) have been modeled using the One Hot Encoding (using the get_dummies method from Pandas)
10. Categorical features (with intrinsic order) have been modeled with integers mapping
11. Numerical features have been analyzed to check their skewness which has been resolved when needed (with BoxCox transformations)
12. Some categorical features have values in the training dataset which are not present in the test dataset. So we decided to remove their column after the One Hot Encoding to improve the score: 'HouseStyle_2.5Fin', 'RoofMatl_Roll', 'RoofMatl_Metal', 'RoofMatl_Membran', 'Heating_Floor', 'Heating_OthW', 'Electrical_Mix', 'MSSubClass_150', 'MSSubClass_90', 'MSZoning_C (all)'

Training:

1. Performing the predictions on the log of the price to normalize the exponential distribution of the houses price.
2. Geometric mean of the predictions from the following models: RidgeCV, LassoCV, ElasticNetCV, GradientBoostingRegressor, BayesianRidge, StackingCVRegressor. In the stacking regressor the predictors are all the previous (without the cross validation) and the meta predictor is a Lasso model. The cross validation is a 20-folds cross validation, and we perform a refit on all the data after selecting the best parameters.
3. Normalizing the outliers (price too high or too low) of the predicted price
4. Rounding the predicted price to multiples of 1000