Manarat International University

Department of Computer Science and Engineering
Artificial Intelligence (CSE – 411)

House Prices: Advanced Regression Techniques

Team Information:

- **Name Of Our Team: Friends**
- **Contestants Name & Student ID**
  - **kazi Mushfiqur Rahman :: 1640CSE00465**
  - **Minhazul Zannat :: 1640CSE00466**
  - **Ashrafujjaman :: 1640CSE00537**

Introduction : It's an online project from kaggle called House Prices: Advanced Regression Techniques. It gives a task to predict the price of some unknown houses by examine theirs multi dimensional features of one specific place, base on the observation of some other houses features which prices are known.

It gives 1460 houses data (80 features) including their sale price as training data and gives 1459 houses data (79 features) without sale price as test data and also a data description to understand the data mapping. The task is to predict 1459 houses sale prices which are not given.

Data Preprocessing : Our team makes many steps to handle those data which overview is given bellow.

1. **Analyzing train-&-test data files & data-description to modify :** Most of the time there are some print mistakes and other problems too in data description which needs to maintain like changing wrong key-words (as : 'C' of 'MSZoning' to 'C (all)'. Some features value need to be converted too to handle that's

value priority properly, as like exchanging specific value of some categorical features with some numerical values (like : 'A' of 'MSZoning' ~ 2). Some specific feature’s attributes need to be modified or convert into others as like YrSold, MoSold have no effect on SalePrice (= can be string) on the other hand 'YearBuilt', 'YearRemodAdd' have serious effect on SalePrice (= need to be numeric).

1. [lower 'YearBuilt' ~ less SalePrice]
2. [lower 'YearRemodAdd' ~ less SalePrice] Also have to find relationships between some features to merge or use them as a group. [like : 'MasVnrArea' & 'MasVnrType']

2. **Handling outliers:** Using hitmap we find out the most effective features for prediction of SalePrice. Then we handle out-liars (numerical & categorical) focusing on the most effective features. We mostly handle the out-lies of a feature which cause too much noise in relationship curve with salePrice as we have to focus also in the relation (less data ~ lower performance).

3. **Handling missing data:** Using some techniques we first find the features which have missing data in both training and test file. Then we split the missing data set into two categories.

   a. **Single level:** using relationship of features[found before] we establish a csv file and then check them individually using 'MS excel > filter (ctrl + shift + L)' and handle the missing data.
   We also do some unrealistic data handling (like : 2590, 'GarageYrBlt' = 2207 convert into 2007)

b. **Multi level:** We do some categories and fill the blanks according to the respected conditions.
   i. 'NA' means Special value (like: Functional : Typ)
   ii. categorical(= need to be numerical) (like: LandSlope)
   iii. 'NA' means most frequent value (like: YearBuilt)
   iv. categorical 'NA' means 'None'
   v. numerical 'NA' means 0
   vi. 'NA' means most or recent common value according to
   (base on) other special groups (like: MSZoning + MSSubClass)
   vii. 'NA' means special value because of condition of data description (like: YearRemodAdd + YearBuilt)

4. Redundant samples: We do some redundant to some features which have too much missing values or complex to establish them in processing or even have a cheap interfere on salePrice.(like: utilities)

Feature Engineering:

We create some new features and also drop some too. We create them by combination of some relative features (like: YearBuilt and YearRemodAdd)
We also make some features which are just as definition of some features (like: hasBath defines that does the house have bathroom or not)

We also make some categorical features as numerical according to the data description file which was modified to handle feature’s value priority properly, as like exchanging specific value of some categorical features with some numerical values (like : 'A' of 'MSZoning' ~ 2).

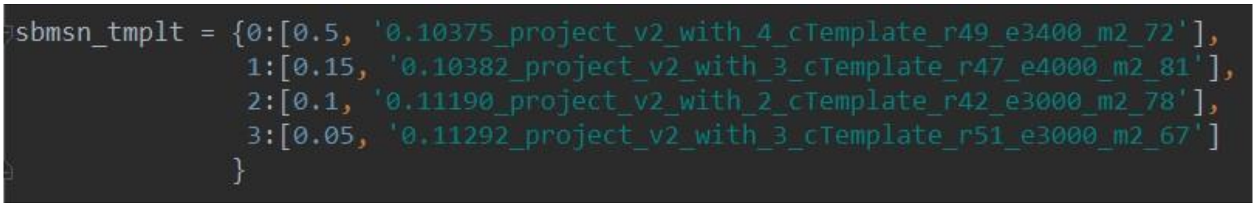
But we don’t reduce any feature dimensionally as the feature set is not too big to make the prediction system perfectly.

Model Methods: We use multiple methods to make our model. Some of them are.

- RobustScaler
- LassoCV
- ElasticNetCV
- SVM
- GradientBoostingRegressor
- LGBMRegressor
- XGBRegressor
- StackingCVRegressor

We use KFold as model selection and n_splits=10 as 10 subsets of the dataset while shuffle is activated.
And also we use merge technique to merge the models results to get the final output.

Results & Discussion: we submit one result in our main account of kaggle though we have submitted more than 64 times in our other accounts. We have to do that as a part of our model as we use neural network approach on our model. We keep record of our submissions and use them as nodes for our next output.

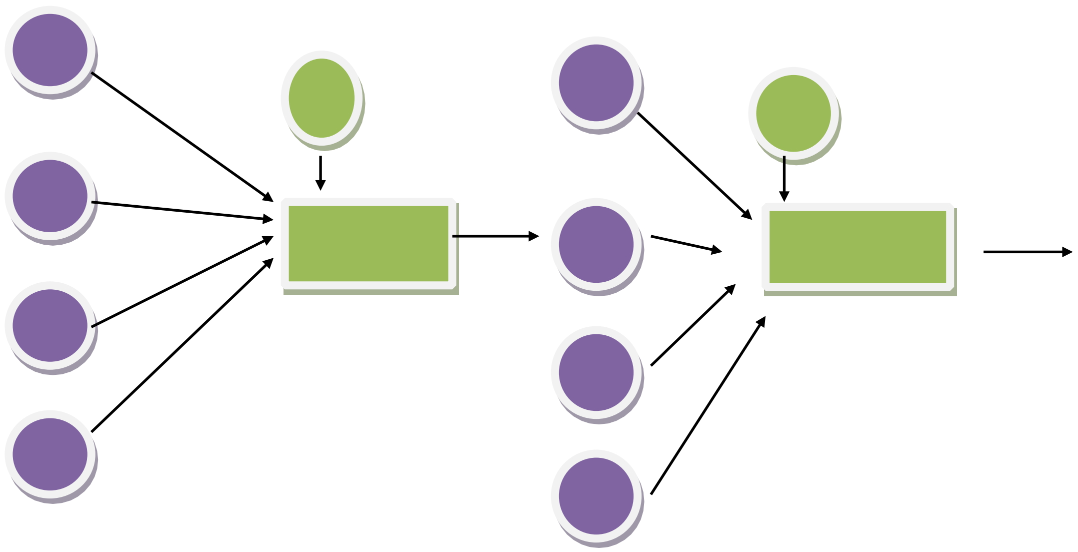


```
sbmsn_tmplt = {0:[0.5, '0.10375_project_v2_with_4_cTemplate_r49_e3400_m2_72'],
               1:[0.15, '0.10382_project_v2_with_3_cTemplate_r47_e4000_m2_81'],
               2:[0.1, '0.11190_project_v2_with_2_cTemplate_r42_e3000_m2_78'],
               3:[0.05, '0.11292_project_v2_with_3_cTemplate_r51_e3000_m2_67']
              }
```

These are some examples of our results which are entering in the model as 4 nodes with specific weights to combine with the result output getting from current stage which has the weight as (1 – total weight of outer nodes).

As we have to use the results of previous processes to create a new result that's why we need to submit a lot of time. And as the submissions are not our real intention to submit even a part of the target submission so we haven't use our main account to do those submission. But we know by doing so we break the rule of contest. For so I, Minhaz, on behalf of my team sorry to all my friends and our honorable teacher and hope you will forgive us.

Submission result: Position: 8, Score: 0.10375



Here, Blue circles indicates the previous results act as node for the process as green rectangular, where green circle is the node created by the current process. All together create a node of another process .