Data Mining: Techniques and Applications (IST 557)

Sweekar Sudhakara (sks6492)

Project: "House Price Prediction"

1) **Introduction**:

The project is about house price prediction i.e., it deals with accurately estimating/predicting the house prices for a home buyer based on the aspects or features described regarding the home using machine learning techniques. For this project, the competition has provided 79 exploratory variables/features which describe the aspect of residential homes in Ames and Iowa. The following files have been given by the competition to perform the task:

- **train.csv** - the training set, that contains 79 exploratory variables/features describing the residential homes and the corresponding final prices of the homes (1460 data samples).
- **test.csv** - the testing set, that contains 79 exploratory variables/features describing the residential homes and we must predict the corresponding final prices of the homes (1459 data samples).
- **data_description.txt** - the file contains text description of each of the 70 exploratory variables/features.
- **sample_submission.csv** - the file contains sample final prices predicted for the home description presented in test.csv to understand the format for submission.

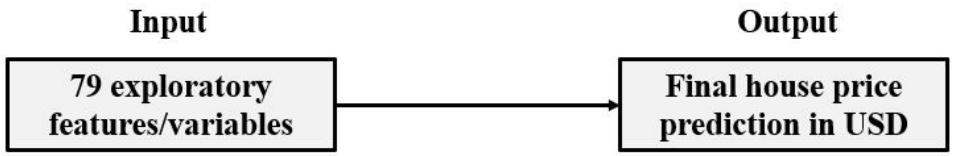
Description of the exploratory 79 features/variables:

Here we briefly describe some of the exploratory features/variables that are key to estimating the final prices of the homes.

- MSSubClass: Identifies the type of dwelling involved in the sale.
- LotFrontage: Linear feet of street connected to property
- Street: Type of road access to property
- LotShape: General shape of property
- HouseStyle: Style of dwelling
- LandContour: Flatness of the property, etc.

The data fields present in the dataset are:

- **Id**: a unique field that represents a particular home.
- **SalePrice**: this field represents the final house price based on the given features.
- **Features**: this field is not directly present in the dataset but it illustrates all the 70 exploratory variables/features as a whole.



2) **Method**:

The best method for solving the house price prediction problem is **Ensemble/Blend based Regression method** with following pre-processing inspired from other notebooks i.e.

1. Feature Engineering:

a. Three different types of features variables have been considered which includes: 1) Numerical, and 2) Categorical type features.
b. Before processing the above features, the "Id" is excluded both from the training & testing data and the "SalePrice" is excluded from the training data as it is unwanted information.
c. Prior to processing the categorical features, the 'YrSold', 'MoSold' and 'MsSubClass' features which are in numerical format are converted to string as they are lesser in number in terms of unique items and it is easy to convert to categorical format.
d. In the numerical column features, the NaN's are replaced with statistical median values as it helps reduce the importance given to the extreme values.
e. In the categorical column features, the NaN's are replaced with string "Missing" that helps indicate an empty place/NaN. However, in the previous checkpoint, the NaN's were replaced with 'None' and there is no significant difference between the two strings.
f. Basic new numerical features are generated based on the Pearson correlation coefficient value obtained against the 'SalePrice'. The reason behind introducing these new features is that they are meaningful and capture the information from multiple features and convey it together. The new features are:

    1) 'LotCompAr': the overall Lot area of the houses.
    2) 'BsmtFinComp': the complete basement floor area.
    3) 'BathComp': the total number of half and full bathrooms.
    4) 'PorchComp': the total number of porches in the house.
    5) 'HouseSFOverall': the total house square feet area.

g. Certain column features for example 'FireplaceQu', 'BstmQual', 'BstmCond', etc, are further label encoded that is based on the total number of unique items in the column, each unique item is assigned a label.
h. Conversion of 'SalePrice' (Training data predictions) into a logarithmic scale accustomed to the loss function.
i. Removal of outliers based on a threshold on the feature 'GrLivArea' which describes the above grade (ground) living area square feet. Removal of these outliers contribute significantly in the performance improvement.
j. For each of the categorical column features, the values or the items with the lowest occurrence i.e. less than 0.02% among all items are assigned with the string 'Sparse' indicating rarely occurring feature items among all other items.

k. The skewness of feature values are fixed with a threshold value of 0.75 along with log transformation with a lambda value of 0.15. These values have been picked based on an iterative study.
l. Further all the categorical variables are converted to dummy/indicator variables as they are more meaningful for the network to process.
m. The data has been split considering 4 folds and evaluated the model using Root-Mean Squared Error (RMSE) function. The k-fold methodology used is Stratified k-fold as it helps preserve the percentage of samples for each class.

As a summary, a total of 167 features were employed in the training and testing of the machine learning model. Some of these features were pre-existing among the 79 exploratory variables/features and some were generated as discussed in this section based on the details given. Some of these ideas of feature generation and processing ideas were adapted from 'Kaggle' and some were self thought.

2. Model, Parameters and Score:

There are several individual models built and trained on the data and these models collectively combine to generate the 'SalePrice' predictions. The models are:

Model	Parameters	Validation RMSLE
XGBoost Regressor	Learning rate=0.02, Base score=0.5, Maximum depth=4, Number of estimators=1000, Subsample value=0.8.	0.1071
Gradient Boosting Regressor	Learning rate=0.05, Base score=0.5, Maximum depth=4, Number of estimators=3000, Minimum number of samples leaf=15.	0.1189
Elastic Net	Maximum iteration= $1 \times e^{+7}$, Alpha value=[0.0001, 0.0002, 0.0004, 0.0005, 0.0007].	0.1357
Support Vector Regressor	Gamma=0.0003, C=20, Epsilon= 0.008.	0.1417
Ridge Regressor	Alpha value=[14.5, 14.7, 14.9, 15, 15.1, 15.3, 15.5].	0.1381
Lasso Regressor	Maximum iteration= $1 \times e^{+7}$, Alpha value=[0.00005, 0.0001, 0.0003, 0.0005, 0.0007, 0.0008]	0.1354

Stacking CV Regressor	Combination of each of the models in the table as regressors excluding XGBoost and XGBoost as the meta-regressor.	0.1105
Overall Model (Ensemble/Blended)	- All the models stated in the table are trained and evaluated on the training data. - A weight is assigned to each of the model's predictions based on the performance of each of these models on the training set. - By using these weights the predictions for the testing data is generated.	**0.11513**

- The performance of all these models are combined together with weights assigned based on their performance on the validation data and together in the form of blended/ensemble models perform well on the testing data.
- The advantage of this ensemble/blended model is that it is robust to outliers and works well on non-linear data. However, the disadvantage of using this type of blended model is that the training time is high.

3. Packages employed:

a. Pandas,
b. Numpy,
c. Skew & Boxcox1p from Scipy-Stats,
d. StratifiedKFold from Sklearn-Model Selection,
e. LinearRegression, RidgeCV, LassoCV, ElasticNetCV and Lasso from Sklearn linear model,
f. SVR from Sklearn-SVM, and
g. LabelEncoder and Make_pipeline from sklearn.

3) Results:

a) Description of other methods tried:

The baseline methods or the other methods tried apart from the best performing method is illustrated in detail in the table presented in the next section. The methods include some of them which have been presented in the checkpoint 1, 2 & 3. The models that have been experimented on include the following:

- **Checkpoint 1**: Different models like XGBoost, LightGBM, ElasticNet, Ridge and Lasso Regression methods were individually trained and tested with the given dataset but only

the XGBoost machine learning method on the feature processed dataset yielded good results among the other state-of-the-art methods with a RMSLE error of 0.12477 as a result it was chosen and submitted for the checkpoint 1. The result of each of these individually trained and tested have been depicted in the table with their respective parameters and results.

- **Checkpoint 2**: Further since none of the individual models were capable enough of predicting the house prices accurately even after basic feature engineering, thus, the ensemble or the blended model which gives combines all the individual models with pre-allocated weights based on their performance on the validation set was employed and it yielded good results on the dataset with a RMSLE error of 0.11756.

- **Checkpoint 3**: Based on the relatively good performance by the weighted combination of all the individual models on the dataset we further improved the model's performance by removing outliers, statistical feature engineering and introduced new mathematical based features which resulted in an overall performance improvement by 0.11624 (RMSLE error).
- Other methods tried but failed include the feature dimensional reduction strategies like PCA and t-SNE which aid in reducing the overall parameters and the model complexity but did not contribute positively in improving the model's performance.

b) Other methods performance analysis:

The performance of each of the models described in the previous section is depicted in the following table.

Other Models	Parameters	Validation RMSLE
XGBoost Regressor	Learning rate=0.02, Base score=0.5, Maximum depth=4, Number of estimators=1000, Subsample value=0.8.	0.1071
Advantages & Disadvantages	- Works better with complex data. - Not very interpretable.	
Gradient Boosting Regressor	Learning rate=0.05, Base score=0.5, Maximum depth=4, Number of estimators=3000, Minimum number of samples leaf=15.	0.1189
Advantages & Disadvantages	- Uses advanced Gradient descent algorithm. - Takes time to train.	

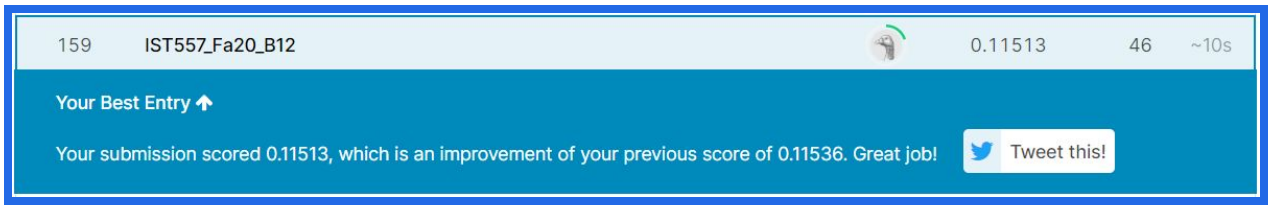
Elastic Net	Maximum iteration= $1 \times e^{+7}$, Alpha value=[0.0001, 0.0002, 0.0004, 0.0005, 0.0007].	0.1357
Advantages & Disadvantages	- Works well when number of features > number of samples. - Very complex to interpret.	
Support Vector Regressor	Gamma=0.0003, C=20, Epsilon= 0.008.	0.1417
Advantages & Disadvantages	- Memory efficient. - Does not work well with large data.	
Lasso Regressor	Maximum iteration= $1 \times e^{+7}$, Alpha value=[0.00005, 0.0001, 0.0003, 0.0005, 0.0007, 0.0008]	0.1354
Advantages & Disadvantages	- Works better with sparse data. - Not very interpretable.	
Best Method	Blended/ensemble based method that combines all the above stated models with a weight based on the performance on the validation data.	**0.11513**
Advantages	- Better performance in comparison to any other method. - Works with complex dataset effectively.	

c) Discussion regarding the performance:

a) The reason behind XGBoost based regression method's overall good performance in the individual model setup could be because it uses boosting based strategy to learn complex non-linear curve fitting.
b) The Lasso and ridge regression based method performs poorly unlike the XGBoost based regression method because the former methods cannot interpret complex data and fails to fit a line accurately.
c) The SVR regression method even though it is a memory efficient algorithm fails to work with sparse and complex data resulting in poor performance.
d) The stacked algorithm with XGBoost as the meta-regressor works well on the dataset because it is an ensemble machine learning algorithm that combines several models with a fixed weight.
e) The blended/ensemble based regression method performs better than any other regression method as it uses a variable weight structure for the individual models based on the individual performance on the validation set.

d) Best Method Performance:

The following figure depicts the screenshot of the best performance obtained so far after several modifications to the code which includes feature engineering and model building. The performance obtained in the **final checkpoint** is **0.11513** which is an improvement over the score obtained in the 3rd checkpoint i.e. 0.11624.



4) Summary:

a) Technical side learning:

- Exploring different state-of-the-art machine learning models along with hyper-parameter tuning to extract the best out of the models.
- Performed feature engineering i.e. built new features based on statistical analysis and eliminated certain outliers. Learned how to avoid over-fitting, analyze the dataset and do corresponding statistical repairs to enhance the dataset.
- Built blended models which uses the capability of individual weak models to build a strong model by allocating weights based on their performance.

b) Learnings throughout the project:

- Learned new competitive strategies to compete with some of the best machine learning researchers in Kaggle to successfully rank among the top 4% in over 5,000 participants.
- Learned from fellow students on how they approached and the methods they used to successfully achieve good scores through their presentations and adapted similar strategies to my work to improve the performance.
- To try different strategies for the same feature engineered dataset to get an enhanced performance with the models.

c) Summary of the performance throughout the project:

Checkpoint	Description	Scores
Checkpoint 1	XGBoost Regressor with basic feature engineering.	0.12477
Checkpoint 2	Blended/ensemble model with feature engineering.	0.11756

Checkpoint 3	Tuned blended/ensemble model with additional feature engineering.	0.11624
Final Checkpoint	Tuned blended/ensemble model with further feature engineering and hyper-parameter tuning.	**0.11513**

d) Future plan:

- To apply the machine learning strategies learnt from this project along with the feature engineering methodology on other competitive projects.