CS838 Data Science Course Project
Stage Five

Group 7
Hongyi Wang, Hao Fu, Miao Yang

April 6, 2017

1 Statistics on Table E: specifically, what is the schema of Table E, how many tuples are in Table E? Give at least four sample tuples from Table E.

In this stage, we use the integrated table E output from stage 4, which describe house condition and sale prices (the data table can be find at https://github.com/samfu1994/cs838webpage/blob/master/stage5/data_set/data_to_analyze.csv). There are **1461** tuples contained in table E, each tuple has **36** attributes. The schema of table E is like:

{*MSSubClass, MSZoning, LotArea, LotConfig, Neighborhood, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, RoofStyle, Exterior1st, ExterQual, Foundation, BsmtFinSF1, BsmtUnfSF, TotalBsmtSF, Heating, 1stFlrSF, 2ndFlrSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, KitchenQual, GarageCars, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, MoSold, YrSold, SalePrice*}

Since it's hard to provide detail for each attribute, we only give details of several attributes here, the full description of attributes in table E can be found at
https://github.com/samfu1994/cs838webpage/blob/master/stage5/data_set/data_description.txt
We now provide description of several important attributes here.

- **MSZoning**: (*categorical feature*) Identifies the general zoning classification of the sale.
  - A: Agriculture
  - C: Commercial
  - FV: Floating Village Residential
  - I: Industrial
  - RH: Residential High Density
  - RL: Residential Low Density
- **LotArea**: (*numerical feature*) Lot size in square feet
- **BldgType**: (*categorical feature*) Type of dwelling
  - 1Fam: Single-family Detached
  - 2FmCon: Two-family Conversion; originally built as one-family dwelling
  - Duplx: Duplex
  - TwnhsE: Townhouse End Unit
  - TwnhsI: Townhouse Inside Unit
- **OverallQual**: (*categorical feature*) Rates the overall material and finish of the house from 1 (Very Poor) to 10 (Very Excellent).
- **1st/2nd FlrSF**: (*numerical feature*) First/Second Floor square feet

- **GarageArea**: (*numerical feature*) Size of garage in square feet
- **YearBuilt**: (*numerical feature*) Original construction date
- **Heating**: (*categorical feature*) Type of heating
  - Floor: Floor Furnace
  - GasA: Gas forced warm air furnace
  - GasW: Gas hot water or steam heat
  - Grav: Gravity furnace
  - OthW: Hot water or steam heat other than gas
  - Wall: Wall furnace
- **SalePrice**: (*numerical feature*) sale price of the house

We now give several **samples of tuples** from table E:

1. 60, RL, 8450, Inside, CollgCr, 1Fam, 2Story, 7, 5, 2003, Gable, VinylSd, Gd, PConc, 706, 150, 856, GasA, 856, 854, 1710, 1, 0, 2, 1, 3, 1, Gd, 2, 548, 0, 61, 0, 2, 2008, 208500
2. 20, RL, 9600, FR2, Veenker, 1Fam, 1Story, 6, 8, 1976, Gable, MetalSd, TA, CBlock, 978, 284, 1262, GasA, 1262, 0, 1262, 0, 1, 2, 0, 3, 1, TA, 2, 460, 298, 0, 0, 5, 2007, 181500
3. 60, RL, 11250, Inside, CollgCr, 1Fam, 2Story, 7, 5, 2001, Gable, VinylSd, Gd, PConc, 486, 434, 920, GasA, 920, 866, 1786, 1, 0, 2, 1, 3, 1, Gd, 2, 608, 0, 42, 0, 9, 2008, 223500
4. 70, RL, 9550, Corner, Crawfor, 1Fam, 2Story, 7, 5, 1915, Gable, Wd, Sdng, TA, BrkTil, 216, 540, 756, GasA, 961, 756, 1717, 1, 0, 1, 0, 3, 1, Gd, 3, 642, 0, 35, 272, 2, 2006, 140000
5. 60, RL, 14260, FR2, NoRidge, 1Fam, 2Story, 8, 5, 2000, Gable, VinylSd, Gd, PConc, 655, 490, 1145, GasA, 1145, 1053, 2198, 1, 0, 2, 1, 4, 1, Gd, 3, 836, 192, 84, 0, 12, 2008, 250000

2 What was the data analysis task that you wanted to do? For that task, describe in detail the data analysis process that you went through.

We did four data analysis tasks in this stage:

- We want to see statistical distribution of house sale price with respect to different attributes and find distribution of outliers for each attribute.
- We want to know if we can use the rest of the attributes to accurately predict the value of the attribute SalePrice. It's a quite classical regression problem, we measure the accuracy by mean-squared-error (MSE), mean-absolute-error(MAE), median-absolute-error, and R^2 score to measure the accuracy
- We want to know if we can use the rest of the attributes to accurately classify if the house was built earlier than 1975
- We imported the table E to MongoDB, and do several database related analysis through SQL

We now describe detailed process we went through.

2.1 SalePrice Distribution wrt Different Attributes

For this analysis, we want to see distribution of house sale price with respect to different attributes (categorical attributes). In particular, for each attribute we first went through all instances and extracted the sale price value based on each possible value of a given attribute. e.g. for categorical attribute **MSZoning**, we extract sale prices for each tuple whose value is **A, C, FV, ..., RL** respectively.
Using these data, we drew box-plot for each categorical attribute in table E, which gave us mean, median, min, max values as well as distribution of outliers. We show one box-plot example here, detailed discussion will be given in Section 4.

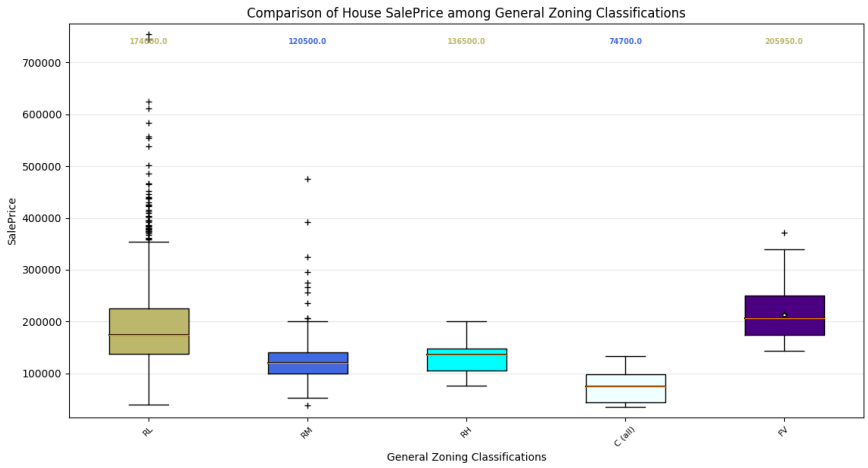

Figure 1: Box-plot wrt MSZoning Attribute

2.2 SalePrice Prediction (Regression Task)

In this data analysis task, we want to use attribute **SalePrice** as label, and use all the rest attributes as features to train a regression model, and solve a classical house price prediction problem.
Firstly, we did several data pre-processing and cleaning work. For data cleaning, there are several missing values in the raw table E. Since we can do nothing for missing values of categorical features, we just removed those attributes with missing values. For numerical features, we took mean value of the whole feature(column) to fill the missing values.
In pre-processing, we basically did two things, the one is we normalize(scale) all numerical features. For categorical features, we use "one-of-K" scheme to encode the categorical feature values to fit the model training (e.g. if attribute A has potential value $\{a, b, c\}$, then for a, we encode it as [1,0,0], and b as [0,1,0], c as [0,0,1]).
We split the data set into training and test sets, in which training set contains **1022** tuples, and test set contains **438** tuples.
During the training process, we trained following models:

- LS with Ridge regularization
- LS with Lasso regularization
- SVM for regression
- Multi-layer Neural Network for regression

To report the accuracy, we use the following metrics for regression tasks, all of the following metrics are follow the definition is scikit-learn(http://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics):

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- Median Absolute Error
- R^2 score

2.3 Build-Year Prediction (Classification Task)

For this task, we want to show if we can accurately classify in which year the house was built, using other attributes (with the SalePrice included).
First, we labeled each tuples based on the attribute **YearBuilt**, if the value is higher than 1975, we labeled it as **"1"**, else we label it **"0"**. Why we choose 1975 as the threshold? It turns out this threshold will give us approximately equal number of positive and negative training instances.
Except for label, we used the same training and test sets from the regression task. During the training process, we trained following models:

- SVM with RBF kernel
- Decision Tree
- Random Forest
- Logistic Regression
- Multi-layer Neural Network

To measure the classification accuracy, we report **Precision, Recall, F1** for this task.

2.4 Database Related Data Analysis

We imported the table E to database(we used MongoDB for this stage), through several queries e.g. groupby, we got several intuition from the data the results are similar to what we got from the box-plot.

3 Give any accuracy numbers that you have obtained (such as precision and recall for your classification scheme).

3.1 Accuracy for Regression

The metrics we used for regression are discussed in Section 2. Now we report the metrics for each model we used in regression task.

Evaluations Among ML Models				
ML Models	MSE	MAE	MedianAE	R^2 score
Ridge	0.25815	0.31562	0.20674	0.72948
Lasso	0.22952	0.28841	0.19683	0.75948
SVM Regression	0.11071	0.19775	0.16268	0.88399
Neural Net	0.14204	0.24236	0.16520	0.85114

From the foregoing results, we can tell that the SVM model gave better regression performance, it predict the house sale price more accurately.

3.2 Accuracy for Classification

For classification task, we report precision, recall, and F1 scores, basically they are like what we did in stage 2.

Evaluations Among Ml Models			
ML Models	Precision	Recall	F_1 Score
Decision Tree	0.86224	0.78241	0.82039
SVM	0.95698	0.82407	0.88557
Random Forest	0.94444	0.78703	0.85858
Logistic Regression	0.91	0.84259	0.87499
Neural Network	0.86538	0.83334	0.84905

From these result, we can tell that the SVM classifier with RBF kernel gave us better classification result.

4 What did you learn/conclude from your data analysis? Were there any problems with the analysis process and with the data?

4.1 What we learn/conclude from the data analysis?

For classification task, what we learned was quite simple, that we can totally use the rest of attributes to predict if a certain house was built before 1975 (or say, a certain year).
For regression task, box-plot, and the database related tasks, we can get more intuition from the data.
Let's discuss the regression task first, after predicting the sale price of house, we also plotted the coefficients (weight vector) in Ridge and Lasso model, since this will show us which attributes are more important in house sale price prediction. The results are given in Figure 2 and Figure 3.

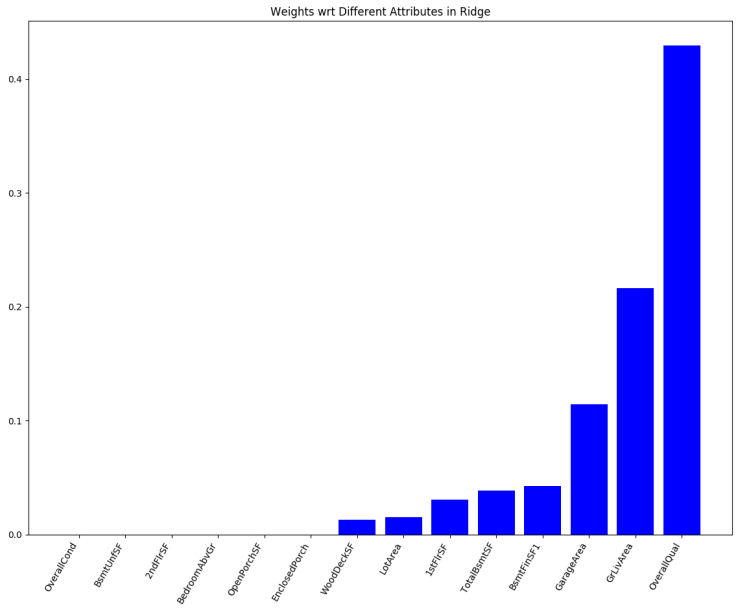

Figure 2: Weights wrt Different Attributes in Ridge

From the figures we can tell that, the attributes **GarageArea** (size of garage in square feet) and **OveralQual** seems more important than others in house price prediction.
Since **GarageArea** is a numerical attribute, we can't tell more about it from our box-plot and SQL query. But for **OveralQual**, here are several interesting results. We first show the box-plot for **OveralQual** in Figure 4. From this box-plot, we can tell that, the house sale price is monotone increasing wrt the overall quality of the house. Also, we can see that, for each attribute value, there are seldom outliers (the black dots), this indicate that the data for **OveralQual** have low variance and are more centered around their mean values.
To verify the foregoing statement, we show another box-plot(in Figure 5). We can see that, in the box-plot for attribute(type of dwelling). There are more outliers, and the variance of

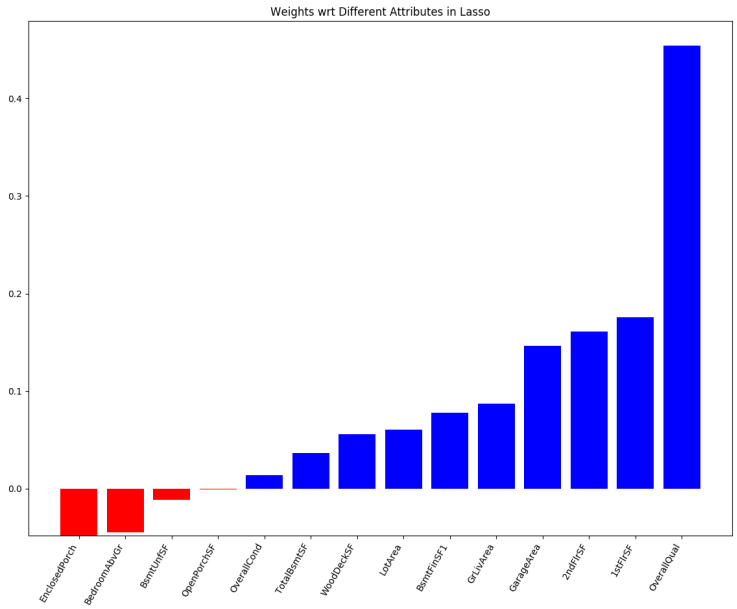

Figure 3: Weights wrt Different Attributes in Lasso

the data distribution is larger.
Finally, we showed the result from the SQL of MongoDB
(using command db.table1.aggregate([{$group:{_id: "$MSZoning", price: {$avg : "$SalePrice"}}}])), we got the same result that the house sale price is monotone increasing wrt the overall quality of the house.

```
{ "_id" : 3, "price" : 87473.75 }
{ "_id" : 2, "price" : 51770.333333333336 }
{ "_id" : 10, "price" : 438588.3888888889 }
{ "_id" : 1, "price" : 50150 }
{ "_id" : 4, "price" : 108420.6551724138 }
{ "_id" : 5, "price" : 133523.34760705288 }
{ "_id" : 8, "price" : 274735.53571428574 }
{ "_id" : 6, "price" : 161603.0347593583 }
{ "_id" : 9, "price" : 367513.0232558139 }
{ "_id" : 7, "price" : 207716.42319749217 }
```

Listing 1: Groupby result wrt OveralQual attribute

Also, to make a comparison to this result, we also show a SQL result wrt "OveralCond" attribute.
(command: db.table1.aggregate([{$group:{_id: "$OveralCond", price: {$avg : "$SalePrice"}}}]))
We can see that the house sale price is not monotone increasing wrt the "OveralCond" attribute.

```
{ "_id" : 3, "price" : 101929.4 }
{ "_id" : 2, "price" : 141986.4 }
{ "_id" : 9, "price" : 216004.54545454544 }
{ "_id" : 7, "price" : 158145.48780487804 }
{ "_id" : 6, "price" : 153961.59126984127 }
{ "_id" : 8, "price" : 155651.73611111112 }
{ "_id" : 1, "price" : 61000 }
{ "_id" : 4, "price" : 120438.43859649122 }
{ "_id" : 5, "price" : 203146.91473812424 }
```

Listing 2: Groupby result wrt OveralCond attribute

4.2 Problem faced during data analyze

- It's quite tricky to handle the missing value, and we didn't want to just delete the data tuples because of one or two missing values. But we really can do nothing about

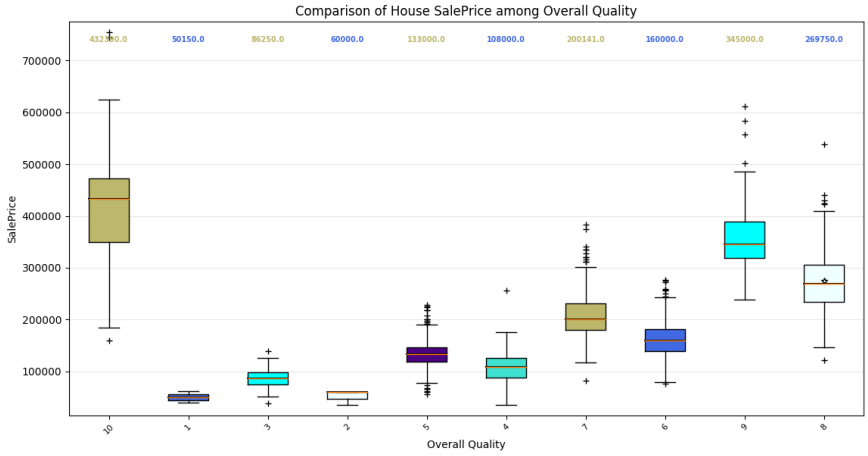

Figure 4: Box-plot for QveralQual Attribute

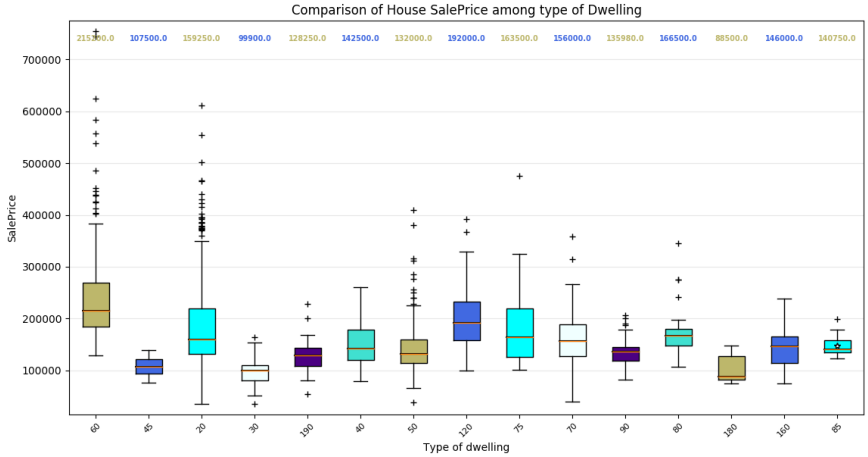

Figure 5: Box-plot for Type of Dwelling

the missing values in categorical attributes. To make a trade-off solution, we had no choice but delete several data tuples

- In regression, since our label is quite large numerical values, it's quite hard to tell whether our regression model is accurate enough. In this stage, we normalized our labels to get better intuition about our regression models

5 If you have more time, what would you propose you can do next?

- For classification task, one can tell that, our recalls are not perfect (actually for several classifier our precisions are also not perfect). If we have more time, we can further debug our classifiers and improve the recall of them through debug values of features, try to include more features, try to remove outliers, and etc.
- For regression task, we are also not "very" accurate, we can try more models, or even deep learning method to improve our accuracy. Actually, there is a house price prediction competition running on Kaggle. We may focus somehow on that in the further weeks.
- We can further develop more intuitions from this dataset and do more cool data visualization staff.