## **Assignment 4 Report**

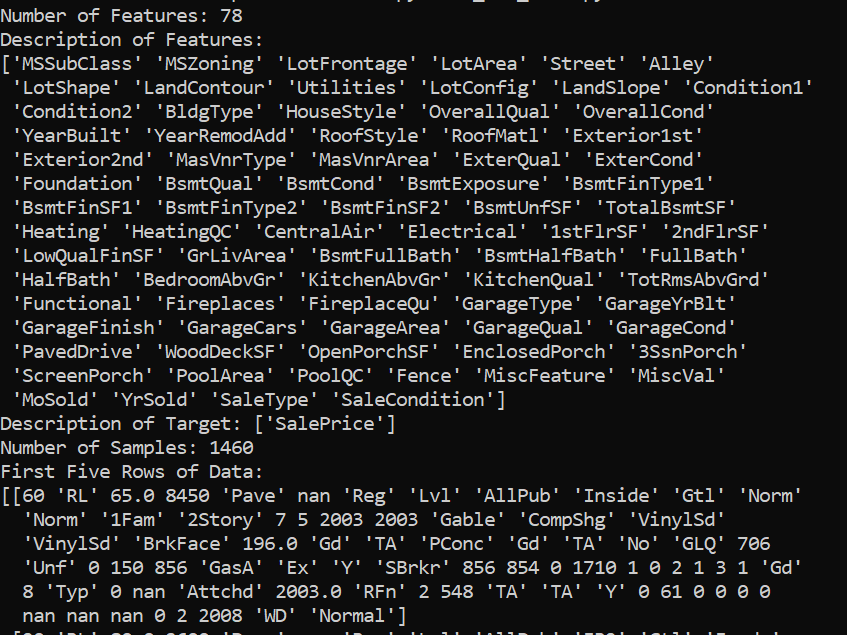
**Names:** Andrew Gabler & Kevin Spike

### **Task 1**

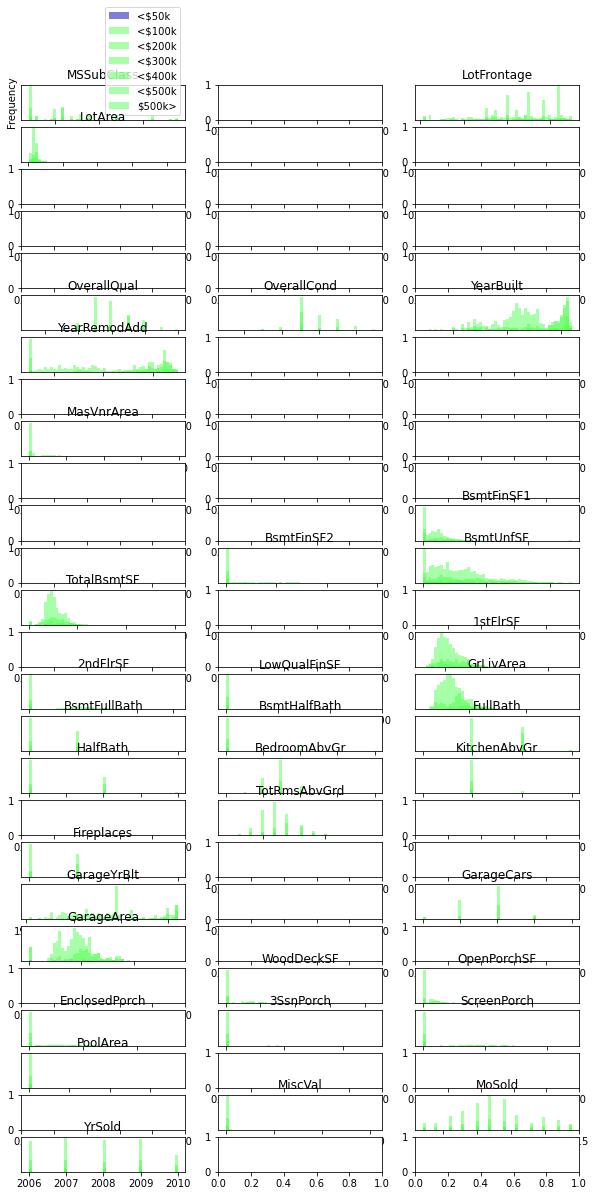
For the first step, correlating to Step 1 on the assignment document, we wrote something to read the CSV file attached. This can be found in *data\_loader.py*. We treated the last column, the sale price, as the target.

### **Task 2**

For the second step, we wrote a simple script in *meet\_the\_data.py* that prints out some basic information about each feature. This is the basic information:



This script generates the histograms shown below. Just a note, the blanks are for those histograms that contain non-numeric characters; they are simply left blank.



### **Task 3**

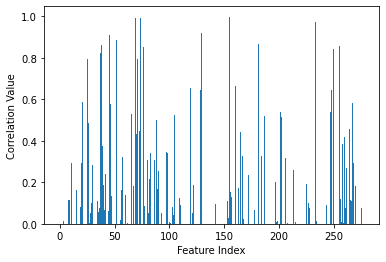
Filling in missing values was done in *information\_fill.py*. This file also performs One-Hot-Encoding on the dataset; this was originally done with the intention that a model would be needed to fill in values. However, the columns with missing values either had missing values because they were meant to be a “None” classification or they were NaN because of not having a feature. An example is “GarageYrBlt”, which does not make sense if the house had no garage. These were just filled in with zeroes. This was done for “MasVnrArea”, “LotFrontage” as well.

### **Task 4**

Univariate feature selection was done in *univariate\_feature\_selection.py*. This uses a simple SelectPercentile with 70% retention. The figure below shows the indices of features that are kept.



This is not quite visible unless expanded; as such, a proper bar graph was used that showed the correlation for each feature.



### **Task 5**

Column transformation mostly included scaling a few columns. This was done in *transformation.py*. A StandardScaler was applied to “LotArea”, “1stFlrSF”, “YearBuilt” and “YearRemodAdd” since these were features where unit variance was important. These features also could deal with 0 being a non-indicator value.

### **Task 6**

Graph below shows in white which features are kept after the script *model\_based\_selection.py* is applied to the dataset.

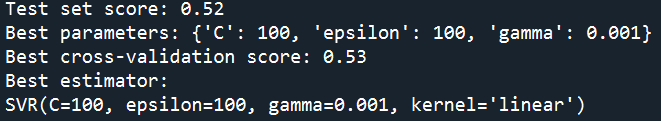


### **Task 7**

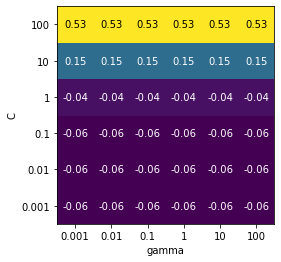
The file used to do PCA analysis on the features was *pca.py*.

### **Task 8**

In the file *model\_development.py*, a GridSearch is used for the best SVR parameters.



The following graph was generated that shows the scoring of each candidate value.



#### File Manifest

**data\_description.txt**

Downloaded description of data.

**data\_loader.py**

Responsible for loading the data so that we can use it.

**houseSalePrices.csv**

The data as downloaded.

**information\_fill.py**

Responsible for filling in missing values.

**meet\_the\_data.py**

Gives cursory glance of the data.

**model\_based\_selection.py**

Uses linear models to select features.

**model\_devopment.py**

Building and running of actual model.

**pca.py**

Responsible for PCA.

**transformation.py**

Does data transformations like scaling.

**univariate\_feature\_selection.py**

Responsible for Univariate Feature Selection.