Assignment 4 Report

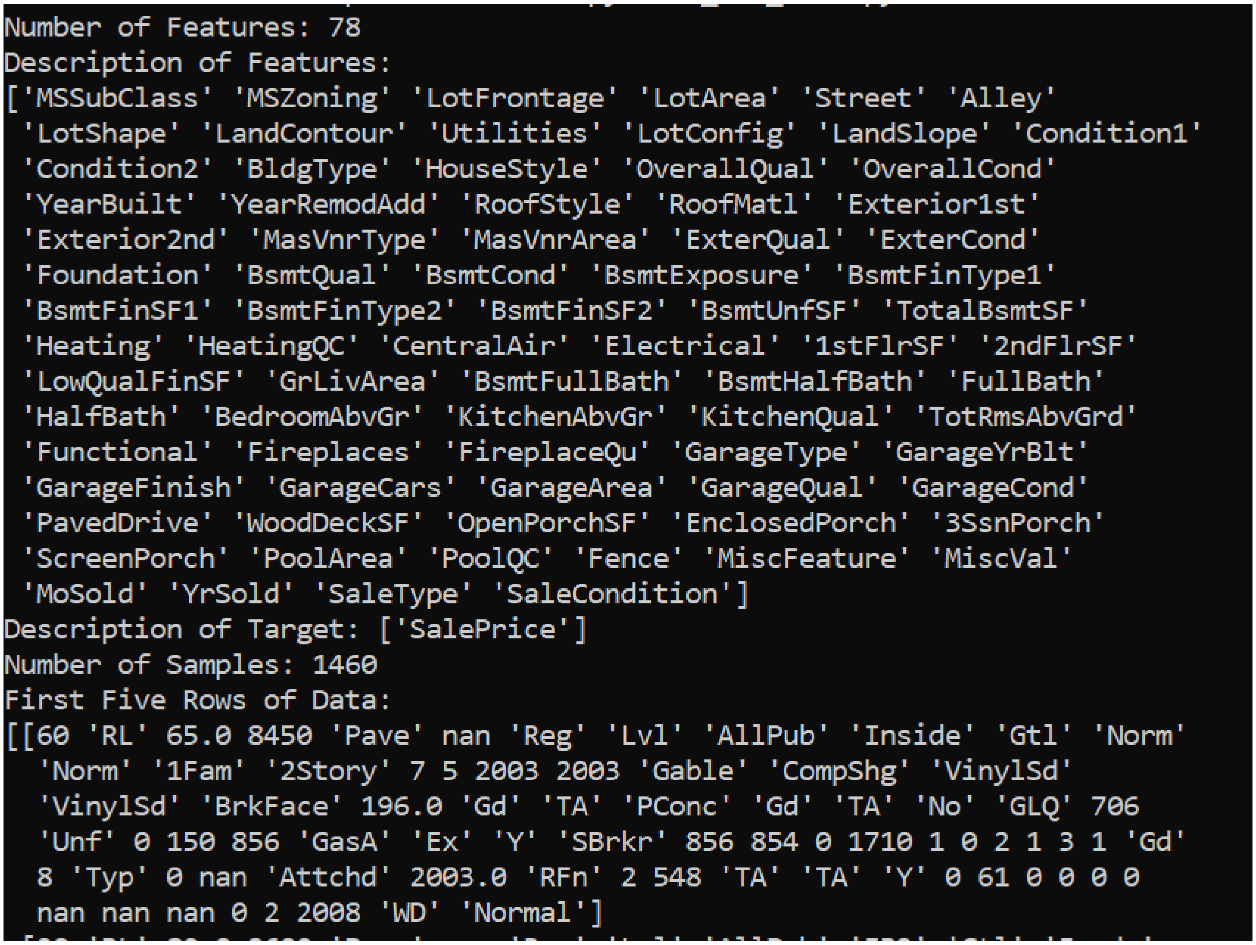
Names: Andrew Gabler & Kevin Spike

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.

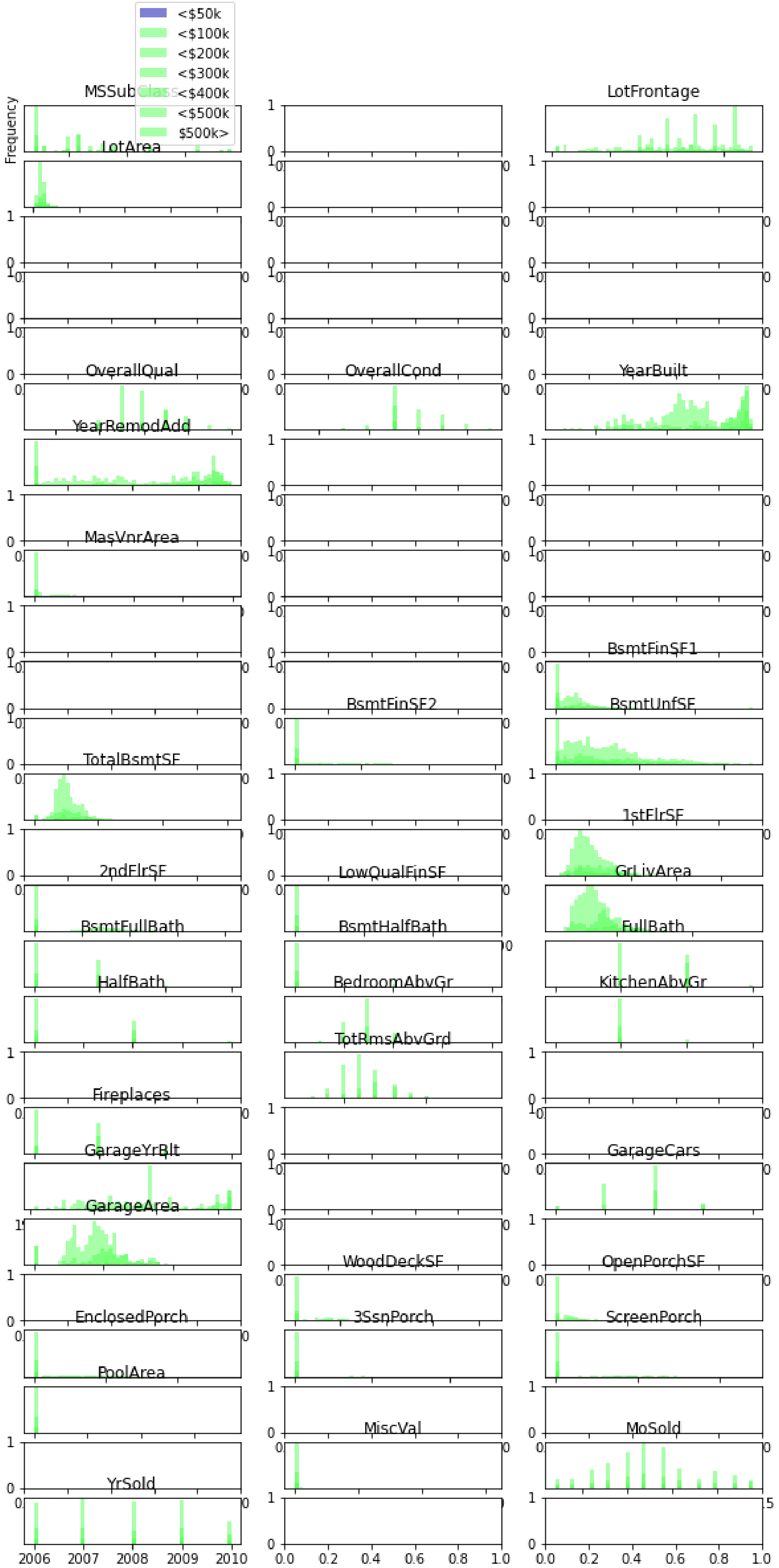
Task 2

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:



```
Number of Features: 78
Description of Features:
['MSSubClass' 'MSZoning' 'LotFrontage' 'LotArea' 'Street' 'Alley'
 'LotShape' 'LandContour' 'Utilities' 'LotConfig' 'LandSlope' 'Condition1'
 'Condition2' 'BldgType' 'HouseStyle' 'OverallQual' 'OverallCond'
 'YearBuilt' 'YearRemodAdd' 'RoofStyle' 'RoofMatl' 'Exterior1st'
 'Exterior2nd' 'MasVnrType' 'MasVnrArea' 'ExterQual' 'ExterCond'
 'Foundation' 'BsmtQual' 'BsmtCond' 'BsmtExposure' 'BsmtFinType1'
 'BsmtFinSF1' 'BsmtFinType2' 'BsmtFinSF2' 'BsmtUnfSF' 'TotalBsmtSF'
 'Heating' 'HeatingQC' 'CentralAir' 'Electrical' '1stFlrSF' '2ndFlrSF'
 'LowQualFinSF' 'GrLivArea' 'BsmtFullBath' 'BsmtHalfBath' 'FullBath'
 'HalfBath' 'BedroomAbvGr' 'KitchenAbvGr' 'KitchenQual' 'TotRmsAbvGrd'
 'Functional' 'Fireplaces' 'FireplaceQu' 'GarageType' 'GarageYrBlt'
 'GarageFinish' 'GarageCars' 'GarageArea' 'GarageQual' 'GarageCond'
 'PavedDrive' 'WoodDeckSF' 'OpenPorchSF' 'EnclosedPorch' '3SsnPorch'
 'ScreenPorch' 'PoolArea' 'PoolQC' 'Fence' 'MiscFeature' 'MiscVal'
 'MoSold' 'YrSold' 'SaleType' 'SaleCondition']
Description of Target: ['SalePrice']
Number of Samples: 1460
First Five Rows of Data:
[[60 'RL' 65.0 8450 'Pave' nan 'Reg' 'Lvl' 'AllPub' 'Inside' 'Gtl' 'Norm'
  'Norm' '1Fam' '2Story' 7 5 2003 2003 'Gable' 'CompShg' 'VinylSd'
  'VinylSd' 'BrkFace' 196.0 'Gd' 'TA' 'PConc' 'Gd' 'TA' 'No' 'GLQ' 706
  'Unf' 0 150 856 'GasA' 'Ex' 'Y' 'SBrkr' 856 854 0 1710 1 0 2 1 3 1 'Gd'
  8 'Typ' 0 nan 'Attchd' 2003.0 'RFn' 2 548 'TA' 'TA' 'Y' 0 61 0 0 0 0
  nan nan nan 0 2 2008 'WD' 'Normal']
```

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.



<$50k
<$100k
<$200k
<$300k
<$400k
<$500k
$500k>
Frequency
MSSubClass
LotFrontage
LotArea
OverallQual
OverallCond
YearBuilt
YearRemodAdd
MasVnrArea
BsmtFinSF1
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
1stFlrSF
2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
TotRmsAbvGrd
Fireplaces
GarageYrBlt
GarageCars
GarageArea
WoodDeckSF
OpenPorchSF
EnclosedPorch
3SsnPorch
ScreenPorch
PoolArea
MiscVal
MoSold
YrSold
0
1
2006
2007
2008
2009
2010
0.0
0.2
0.4
0.6
0.8
1.0

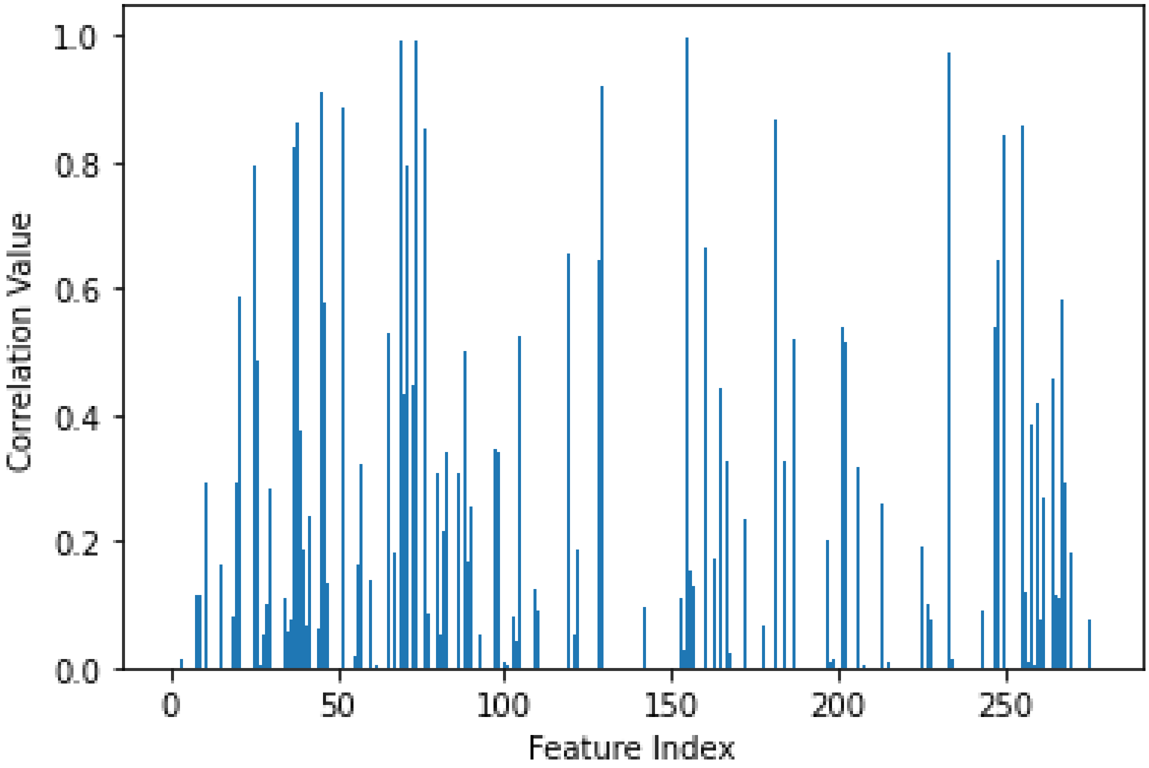
Task 3

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.

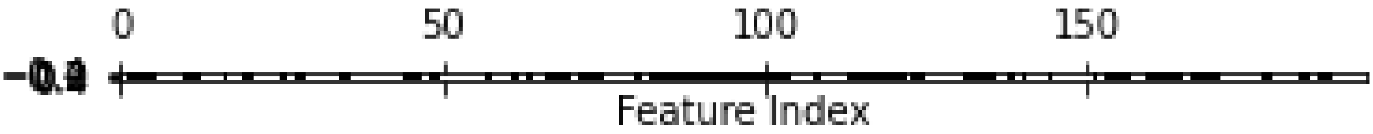


Task 5

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.

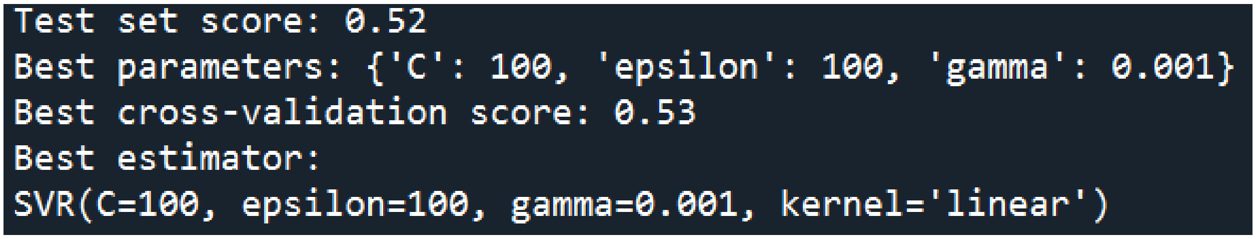


Task 7

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

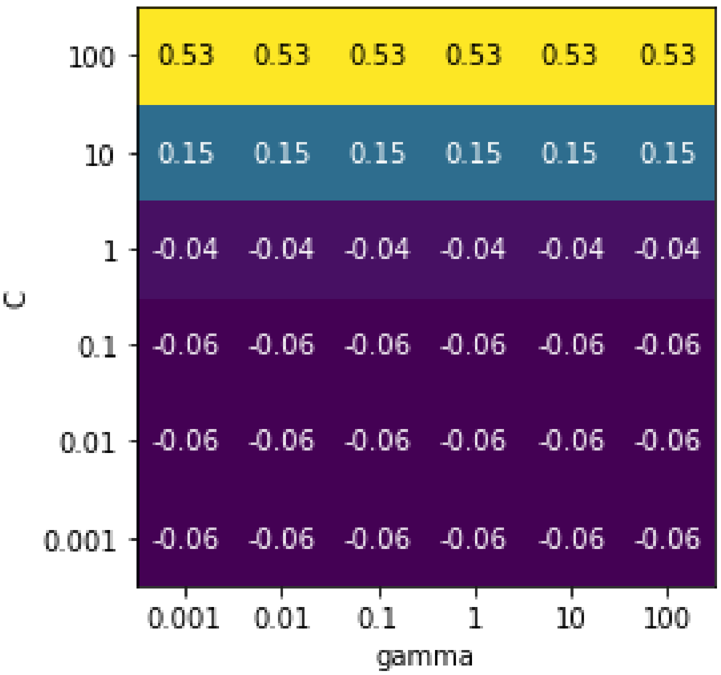
Task 8

In the file *model_development.py*, a GridSearch is used for the best SVR parameters.



```
Test set score: 0.52
Best parameters: {'C': 100, 'epsilon': 100, 'gamma': 0.001}
Best cross-validation score: 0.53
Best estimator:
SVR(C=100, epsilon=100, gamma=0.001, kernel='linear')
```

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


C
100
10
1
0.1
0.01
0.001
gamma
0.001
0.01
0.1
1
10
100
0.53 0.53 0.53 0.53 0.53 0.53
0.15 0.15 0.15 0.15 0.15 0.15
-0.04 -0.04 -0.04 -0.04 -0.04 -0.04
-0.06 -0.06 -0.06 -0.06 -0.06 -0.06
-0.06 -0.06 -0.06 -0.06 -0.06 -0.06
-0.06 -0.06 -0.06 -0.06 -0.06 -0.06

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.