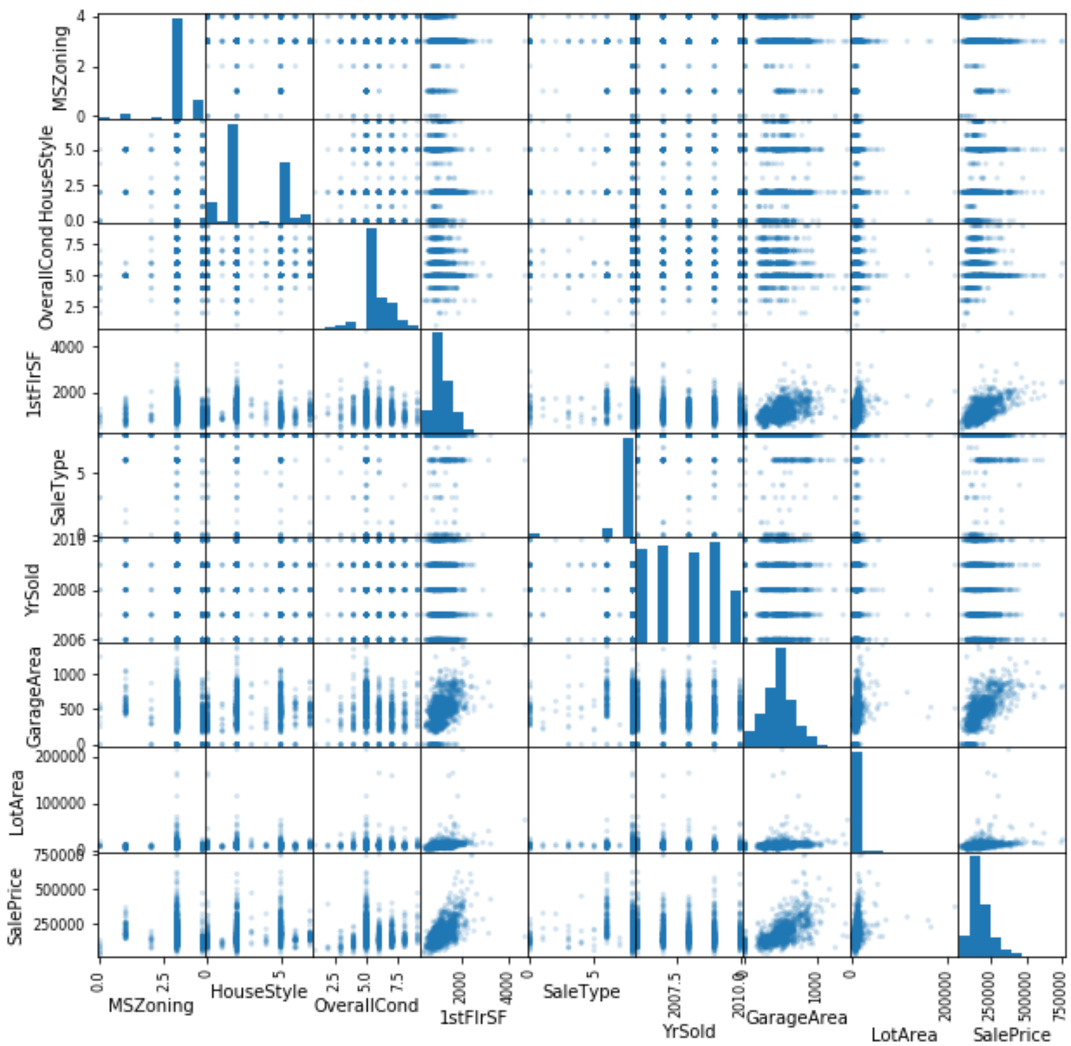
1

(a)

(b) There are 1460 samples and 79 features in the training set. MSSubClass, MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood are some of the categorical features.

(c) "1stFlrSF", "GarageArea", "LotArea" seems to have be highly co-related with SalePrice.



(d) The statsmodel shows that 37 of the all 79 features have a 95% confidence interval that do not contain 0. In other words, 37 coefficients are significant.

	coef	std err	t	P>|t|	[0.025	0.975]	Significant
MSSubClass	-66.73	42.37	-1.58	0.12	-149.85	16.38	No
MSZoning	-2086.23	1456.68	-1.43	0.15	-4943.78	771.32	No
LotFrontage	-130.55	47.15	-2.77	0.01	-223.04	-38.06	Yes
LotArea	0.42	0.10	4.14	0.00	0.22	0.61	Yes

Street	34320.00	13200.0 0	2.60	0.01	8436.47	60200.00	Yes
Alley	-2820.06	2447.86	-1.15	0.25	-7621.97	1981.86	No
LotShape	-758.48	617.06	-1.23	0.22	-1968.96	451.99	No
LandContour	1752.05	1259.33	1.39	0.16	-718.34	4222.45	No
Utilities	-46690.00	31000.0 0	-1.51	0.13	- 108000.0 0	14200.00	No
LotConfig	1.04	510.55	0.00	1.00	-1000.50	1002.58	No
LandSlope	4609.02	3571.45	1.29	0.20	-2397.03	11600.00	No
Neighborhood	379.42	146.10	2.60	0.01	92.83	666.02	Yes
Condition1	-659.72	948.05	-0.70	0.49	-2519.50	1200.05	No
Condition2	-9405.49	3129.98	-3.01	0.00	-15500.00	-3265.47	Yes
BldgType	-2936.09	1399.45	-2.10	0.04	-5681.37	-190.81	Yes
HouseStyle	-672.43	610.81	-1.10	0.27	-1870.65	525.79	No
OverallQual	10230.00	1108.09	9.23	0.00	8054.18	12400.00	Yes
OverallCond	5799.21	978.93	5.92	0.00	3878.87	7719.55	Yes
YearBuilt	237.82	73.73	3.23	0.00	93.19	382.46	Yes
YearRemodAd d	15.48	63.90	0.24	0.81	-109.88	140.84	No
RoofStyle	1029.48	1050.29	0.98	0.33	-1030.85	3089.81	No
RoofMatl	4563.46	1403.48	3.25	0.00	1810.29	7316.63	Yes
Exterior1st	-1093.27	489.01	-2.24	0.03	-2052.54	-134.00	Yes
Exterior2nd	673.80	442.19	1.52	0.13	-193.63	1541.23	No
MasVnrType	3779.43	1377.90	2.74	0.01	1076.43	6482.44	Yes
MasVnrArea	32.16	5.66	5.68	0.00	21.06	43.25	Yes
ExterQual	-9895.16	1837.88	-5.38	0.00	-13500.00	-6289.82	Yes
ExterCond	777.43	1170.89	0.66	0.51	-1519.47	3074.33	No
Foundation	50.42	1571.04	0.03	0.97	-3031.46	3132.30	No
BsmtQual	-7165.04	1265.24	-5.66	0.00	-9647.04	-4683.05	Yes
BsmtCond	2332.21	1223.01	1.91	0.06	-66.94	4731.36	No
BsmtExposure	-2923.30	817.65	-3.58	0.00	-4527.26	-1319.34	Yes
BsmtFinType1	-325.54	581.74	-0.56	0.58	-1466.74	815.65	No
BsmtFinSF1	7.86	2.76	2.85	0.00	2.45	13.27	Yes
BsmtFinType2	2320.40	1045.69	2.22	0.03	269.09	4371.72	Yes
BsmtFinSF2	12.67	5.04	2.52	0.01	2.79	22.55	Yes
BsmtUnfSF	-2.62	2.76	-0.95	0.34	-8.04	2.80	No
TotalBsmtSF	17.91	3.47	5.16	0.00	11.10	24.73	Yes
Heating	-1873.05	2995.47	-0.63	0.53	-7749.20	4003.11	No
HeatingQC	-449.09	574.29	-0.78	0.43	-1575.67	677.48	No
CentralAir	845.05	4188.38	0.20	0.84	-7371.21	9061.31	No

Electrical	-576.96	847.90	-0.68	0.50	-2240.28	1086.35	No
1stFlrSF	20.85	5.72	3.65	0.00	9.63	32.08	Yes
2ndFlrSF	23.95	5.12	4.68	0.00	13.90	33.99	Yes
LowQualFinSF	-25.15	13.08	-1.92	0.06	-50.80	0.51	No
GrLivArea	19.65	5.13	3.83	0.00	9.60	29.71	Yes
BsmtFullBath	5393.97	2278.67	2.37	0.02	923.95	9864.00	Yes
BsmtHalfBath	-853.71	3593.54	-0.24	0.81	-7903.09	6195.67	No
FullBath	1687.25	2515.19	0.67	0.50	-3246.75	6621.24	No
HalfBath	-320.91	2365.33	-0.14	0.89	-4960.94	4319.12	No
BedroomAbvGr	-4281.96	1555.79	-2.75	0.01	-7333.93	-1229.99	Yes
KitchenAbvGr	-18440.00	4708.94	-3.92	0.00	-27700.00	-9206.48	Yes
KitchenQual	-7336.03	1350.68	-5.43	0.00	-9985.64	-4686.43	Yes
TotRmsAbvGrd	4420.21	1097.11	4.03	0.00	2268.03	6572.40	Yes
Functional	3573.87	882.95	4.05	0.00	1841.81	5305.93	Yes
Fireplaces	8196.07	2459.16	3.33	0.00	3371.98	13000.00	Yes
FireplaceQu	-1347.16	738.64	-1.82	0.07	-2796.13	101.81	No
GarageType	372.98	586.13	0.64	0.53	-776.83	1522.79	No
GarageYrBlt	-25.99	65.58	-0.40	0.69	-154.64	102.65	No
GarageFinish	-2270.87	1332.59	-1.70	0.09	-4884.98	343.24	No
GarageCars	9827.77	2620.54	3.75	0.00	4687.10	15000.00	Yes
GarageArea	6.61	9.06	0.73	0.47	-11.17	24.38	No
GarageQual	-2214.90	1558.92	-1.42	0.16	-5273.00	843.20	No
GarageCond	230.18	1640.49	0.14	0.89	-2987.94	3448.29	No
PavedDrive	2173.76	1933.23	1.12	0.26	-1618.61	5966.13	No
WoodDeckSF	20.07	7.02	2.86	0.00	6.30	33.85	Yes
OpenPorchSF	-6.04	13.39	-0.45	0.65	-32.30	20.23	No
EnclosedPorch	-3.67	14.64	-0.25	0.80	-32.39	25.06	No
3SsnPorch	28.83	27.10	1.06	0.29	-24.33	81.99	No
ScreenPorch	47.53	15.01	3.17	0.00	18.09	76.97	Yes
PoolArea	752.16	59.47	12.65	0.00	635.50	868.82	Yes
PoolQC	-209600.00	15100.00	-13.85	0.00	-239000.00	-180000.00	Yes
Fence	122.77	836.62	0.15	0.88	-1518.41	1763.94	No
MiscFeature	-1606.44	1567.59	-1.03	0.31	-4681.55	1468.68	No
MiscVal	0.41	1.72	0.24	0.81	-2.96	3.78	No
MoSold	-75.86	294.39	-0.26	0.80	-653.36	501.65	No
YrSold	-338.16	82.32	-4.11	0.00	-499.65	-176.68	Yes
SaleType	-686.75	538.27	-1.28	0.20	-1742.66	369.16	No
SaleCondition	3655.03	774.29	4.72	0.00	2136.11	5173.95	Yes

(e) According to the result, the backward stepwise regression with 10-fold cross validation performs the best. The relevant parameter and accuracy of these regression methods are as follows:

	OLS	k-NN	Ridge-10 fold	Lasso-10 fold	BSR-10 fold	FSR-10 fold
parameter		K=9	$\lambda = 10$	$\lambda = 222379$	p=30	p=27
Accuracy	0.884	0.668	0.821	0.757	0.832	0.831

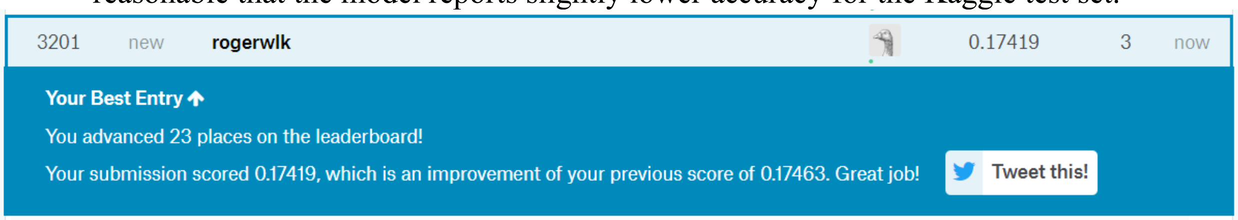
(f) After adding the quadratic features, the number of features is bigger than the number of training samples. This means that the adjusted-R^2 is negative. We haven't figure out a proper way to deal with this.

(g) The following table shows the variables retained by FSR and Lasso 10-fold. In most cases, they match our intuitions.

	Retained by FSR 10 Fold	Retained by Lasso 10 Fold
OverallQual	1	
GrLivArea	1	1
BsmtFinSF1	1	1
ExterQual	1	
GarageCars	1	
MSSubClass	1	1
KitchenQual	1	
YearBuilt	1	1
OverallCond	1	
LotArea	1	1
BsmtQual	1	
BsmtCond	1	
Fireplaces	1	
Functional	1	
MasVnrArea	1	1
BsmtFullBath	1	
BsmtExposure	1	
ScreenPorch	1	1
GarageCond	1	
WoodDeckSF	1	1
SaleCondition	1	
MasVnrType	1	
Street	1	
KitchenAbvGr	1	
Neighborhood	1	
MiscFeature	1	

PavedDrive	1	
YearRemodAdd		1
BsmtFinSF2		1
TotalBsmtSF		1
GarageArea		1
MiscVal		1

(h) For the Kaggle test set, the accuracy is 0.82581, which is lower than the hold-out validation accuracy. The hold-out validation test set is used to search for the best features, possibly resulting in overfitting. Because of the intrinsic variance of real test data, it's reasonable that the model reports slightly lower accuracy for the Kaggle test set.



2

(a) Yes, the labels are balanced. there are 500 positive and 500 negative reviews in each of the three files. We read the files, strip and split them line by line.

(b) We did all of the processing.

   a. By using only the lowercase and lemmatized words, the algorithm avoids treating the same words as different ones.

   b. We strip punctuation because, given the special way the algorithm works, it doesn't contribute to better understanding of the meaning of the review.

   c. We use a dictionary to strip the stop words to reduce the noise, so that the words that convey important information get deserved attention.

(c) We put the 2400 training reviews in list "“train_comments" and the 600 testing reviews in "test_comments".

(d) We printed the feature vector of the first and last of the training review set.

(e) We adopted the $l1-normalization$ because it's not sensitive to outliers and is easy to compute.

(f) The accuracy rate is 0.807. The confusion matrix is:

	Predicted positive	Predicted negative
Actual positive	233	67
Actual negative	49	251

(g)

   a. For the logistic regression with L2 (Ridge) penalty, the accuracy is 0.812. The most important words are:

```
ridge_reg score:  0.8116666666666666
-9.618895246292924 bad
-9.495534206943493 poor
-8.630034135120853 worst
-7.557491602815726 suck
-7.253623428010545 stupid

8.171356284118554 nice
8.701367610592152 delici
9.211282662997126 excel
11.07492346385661 great
11.131933906667545 love
```

b. With L1 (Lasso) penalty, the accuracy is 0.802. The most important words are:

```
lasso_reg score:  0.8016666666666666
-38.21068197450742 suck
-37.3495866984668 poor
-35.180229009615005 starter
-34.13712351085633 rude
-33.103730818035 stupid

30.494646521108994 wonder
31.93954286486569 soundtrack
32.534540788461214 beauti
33.99297262808296 15
34.75609218292082 rang
```

(h) For the 2-gram model, we also use the $l1$-normalization. The accuracy is 0.838. The confusion matrix is as follows.

	Predicted positive	Predicted negative
Actual positive	240	60
Actual negative	37	263

Also, both the ridge and lasso regularization get higher accuracy than the previous model.

```
ridge_reg bigram score:  0.8183333
-11.082743477338376 bad
-10.29417288468737 poor
-9.46343241220174 worst
-8.135534492865625 suck
-7.453715011476128 minut

8.708656298632569 nice
8.73474364492587 delici
10.74248730054831 excel
12.0859002328477 love
13.456966134340739 great
```

```
lasso_reg bigram score:  0.80333333
-38.22170134071952 suck
-37.353931124427035 poor
-35.17958414585297 starter
-34.14469960225314 rude
-33.11211540110329 stupid

30.489052831120798 wonder
31.94344069227162 soundtrack
32.52645487946976 beauti
33.99174183520786 15
34.77567562155982 rang
```

(i) According to the results above, the logistic regression with 2-grams performs the best in the prediction task. First of all, 2-grams sequences are more independent with each other. Secondly, they also provide additional information. For the language used in the reviews, we found that strong words such as “bad” and ”great” are the most significant indicator of positive or negative reviews.

Written Exercise

1.

$$X_{aug}^T X_{aug} = \begin{bmatrix} n & \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i2} & \cdots & \sum_{i=1}^{n} x_{ip} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i}^{n} x_{i1}^2 + \lambda & \sum_{i}^{n} x_{i1}x_{i2} & \cdots & \sum_{i}^{n} x_{i1}x_{ip} \\ \sum_{i=1}^{n} x_{i2} & \sum_{i}^{n} x_{i1}x_{i2} & \sum_{i}^{n} x_{i2}^2 + \lambda & \cdots & \sum_{i}^{n} x_{i2}x_{ip} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_{ip} & \sum_{i}^{n} x_{i1}x_{ip} & \sum_{i}^{n} x_{i2}x_{ip} & \cdots & \sum_{i}^{n} x_{ip}^2 + \lambda \end{bmatrix} = \left[\begin{array}{c:c} n & M \\ \hdashline M & X^T X + \lambda \end{array}\right]$$

$$X_{aug}^T X_{aug} \hat{\beta} = X_{aug}^T X_{aug} \bullet \begin{bmatrix} 0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{p} (\beta_j \sum_{i}^{n} x_{ij}) \\ (X^T X + \lambda I)\hat{\beta} \end{bmatrix}$$

$$X_{aug}^T Y = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ X^T Y \end{bmatrix}$$

Let $X_{aug}^T X_{aug} \hat{\beta} = X_{aug}^T Y$, we have $(X^T X + \lambda I)\hat{\beta} = X^T Y$. That is:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T Y$$

2.

(a) According to Bayes optimal classifier, the probability of pneumonia is 0, the probability of flu is 1/3 and the probability of healthy is 1/3.

$$P[Class = Pneumonia \mid (fever = T, headache = F)]$$

$$= \frac{P(Pneumonia)P(T,F \mid Pneumonia)}{P(T,F)}$$

$$= \frac{\frac{1}{10} \times 0}{\frac{9}{100}} = 0$$

$$P[Class = Flu \mid (fever = T, headache = F)]$$
$$= \frac{P(Flu)P(T, F \mid Flu)}{P(T, F)}$$
$$= \frac{2}{3}$$

$$P[Class = Healthy \mid (fever = T, headache = F)]$$
$$= \frac{P(Healthy)P(T, F \mid Healthy)}{P(T, F)}$$
$$= \frac{1}{3}$$

(b)

$$P(pneumonia \mid T, F) = \frac{P(pneumonia)P(T, F \mid pneumonia)}{P(T, F)}$$
$$= \frac{P(pneumonia)P(fever = T \mid pneumonia)P(headache = F \mid pneumonia)}{P(T, F)}$$
$$= \frac{\frac{1}{10} \times \frac{1}{2} \times \frac{1}{10}}{\frac{9}{100}} = \frac{1}{18}$$

$$P(flu \mid T, F) = \frac{P(flu)P(T, F \mid flu)}{P(T, F)}$$
$$= \frac{P(flu)P(fever = T \mid flu)P(headache = F \mid flu)}{P(T, F)}$$
$$= \frac{\frac{2}{10} \times \frac{15}{20} \times \frac{8}{20}}{\frac{9}{100}} = \frac{2}{3}$$

$$P(healthy \mid T, F) = \frac{P(healthy)P(T, F \mid healthy)}{P(T, F)}$$
$$= \frac{P(healthy)P(fever = T \mid healthy)P(headache = F \mid healthy)}{P(T, F)}$$
$$= \frac{\frac{7}{10} \times \frac{5}{70} \times \frac{61}{70}}{\frac{9}{100}} = \frac{61}{126}$$

We can force these three values sum to 1 by normalizing them.

3.

$$P(yes \mid 30, medium, yes, fair) \propto P(yes)P(\leq 30 \mid yes)P(medium \mid yes)P(yes \mid yes)P(fair \mid yes)$$
$$= \frac{9}{14} \times \frac{2}{9} \times \frac{4}{9} \times \frac{6}{9} \times \frac{6}{9}$$

$$P(no \mid 30, medium, yes, fair) \propto P(no)P(\leq 30 \mid no)P(medium \mid no)P(yes \mid no)P(fair \mid no)$$
$$= \frac{5}{14} \times \frac{3}{9} \times \frac{2}{9} \times \frac{1}{9} \times \frac{2}{9}$$

Obviously, $P(yes \mid 30, medium, yes, fair) > P(no \mid 30, medium, yes, fair)$. Therefore, we predict the new example will buy a computer.