Chenyun Yu

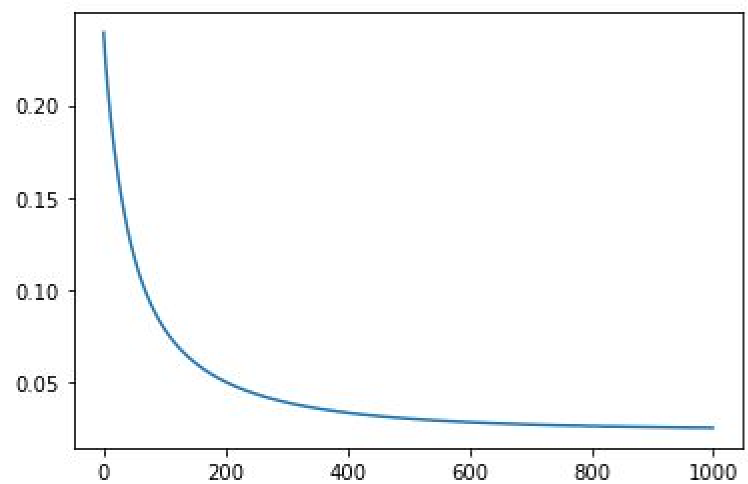
Prof. Fern

CS-575 MACHINE LEARNING

Implementation 2 Reporter

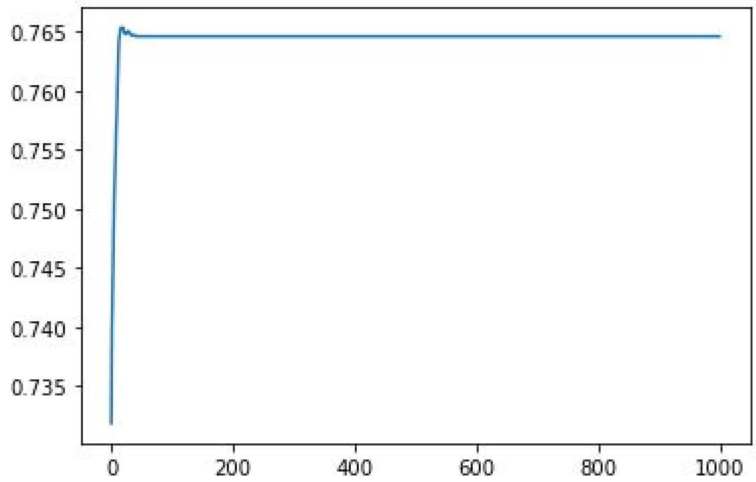
Part 1 logistic regression with L2 (Ridge) regularization.

1. Implement Algorithm 1 and experiment with different regularization parameters $\lambda \in \{10^{-i}: i \in [0,5]\}$.
   Implementation can be found in the *part1_final.py*, in this part, I believe the most urgent thing is to find a reasonable convergence condition for this regression. Based on the experience from the last experiment, I choose a relatively large learning rate(alpha), which is 10 ^-1; and a middle regularization parameter, which is 10^-2. The results of the convergence test are as follows, therefore I believe that when the gradient descent less than 10 ^-1 or 10^-2, the function gradually converges.

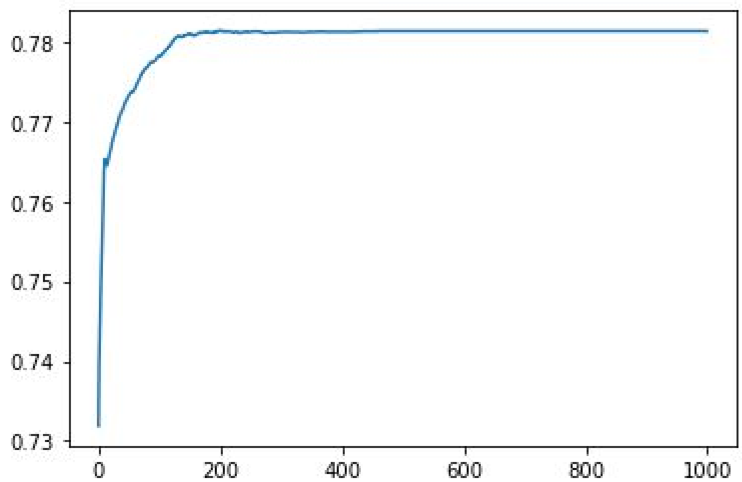


2. Plot the training accuracy and validation accuracy of the learned model as the λ value varies. What trend do you observe for the training accuracy as we increase λ? Why is this the case? What trend do you observe for the validation accuracy? What is the best λ value based on validation accuracy?

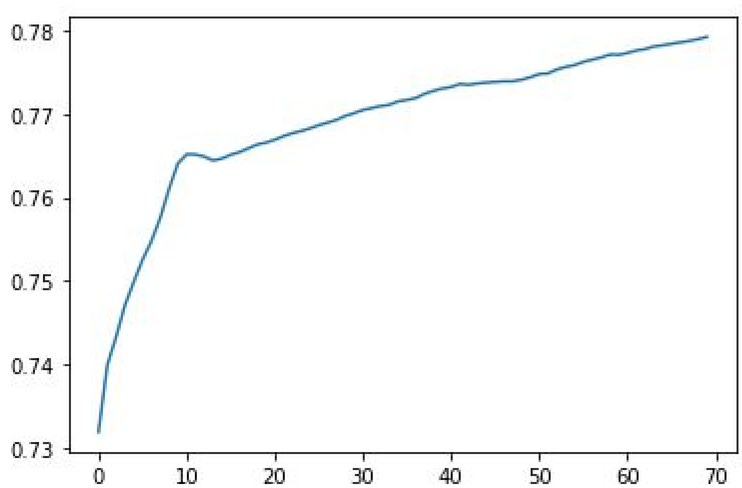
   a. Using the 10^-1 as the convergence mark for testing training accuracy



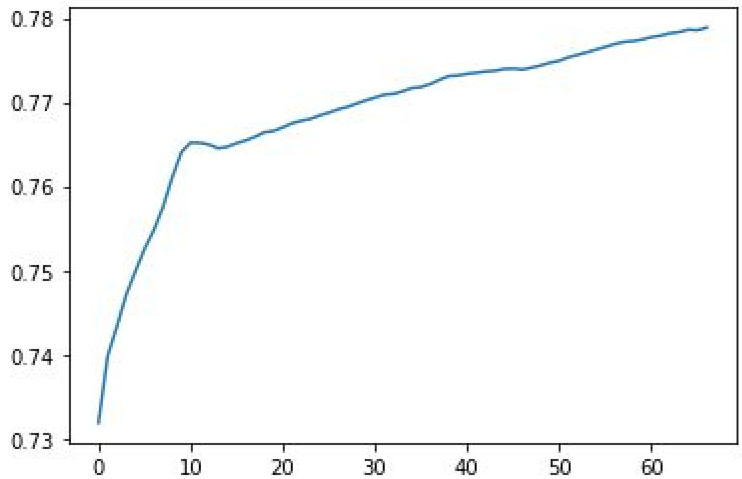
Regularization parameter = 10 ^ -1



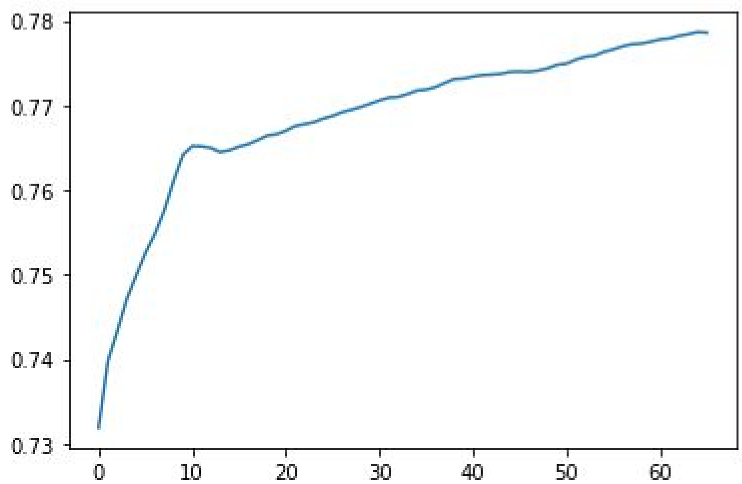
Regularization parameter = 10 ^ -2



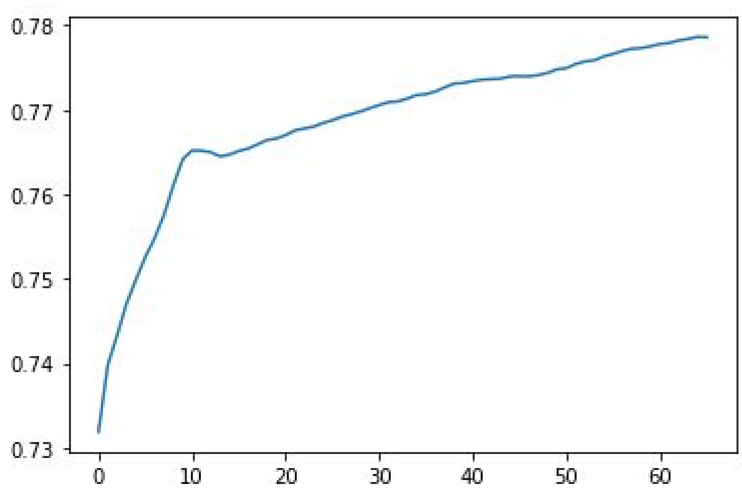
Regularization parameter = 10 ^ -3:



Regularization parameter = 10 ^ -4:



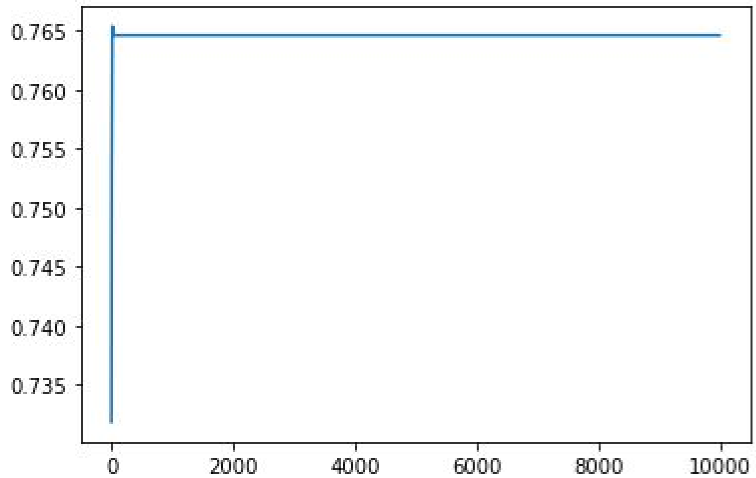
Regularization parameter = 10 ^ -5:



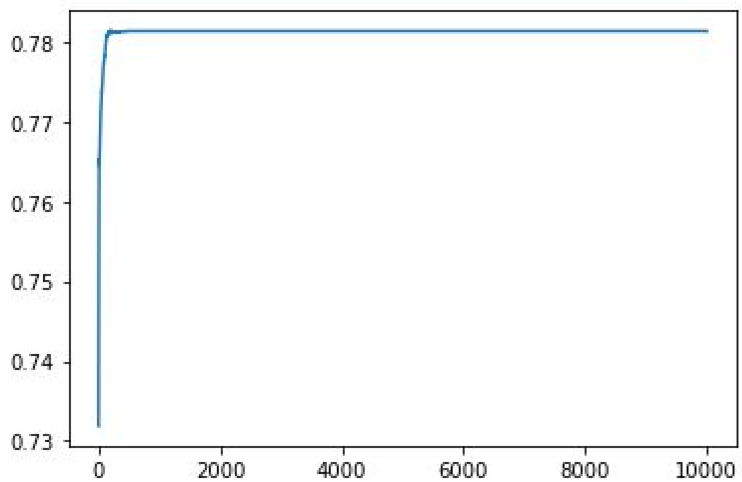
Based on my observations, the training accuracy is closer to 0,8, another conclusion that can be generated from the plots is that the 10 ^-1's convergence condition is too loose, because, in the last few plots, there is no sign of convergence. I will use 10 ^-2 as the convergence condition.

b. Using the 10^-2 as the convergence mark for testing validation accuracy

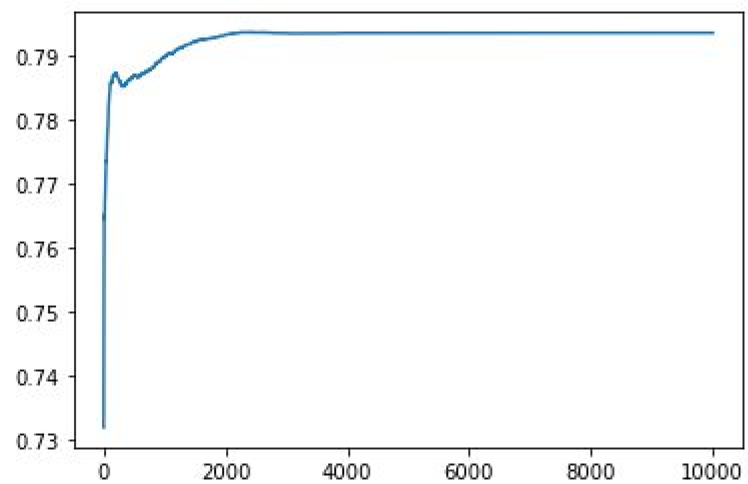
Regularization parameter = 10 ^ 0:



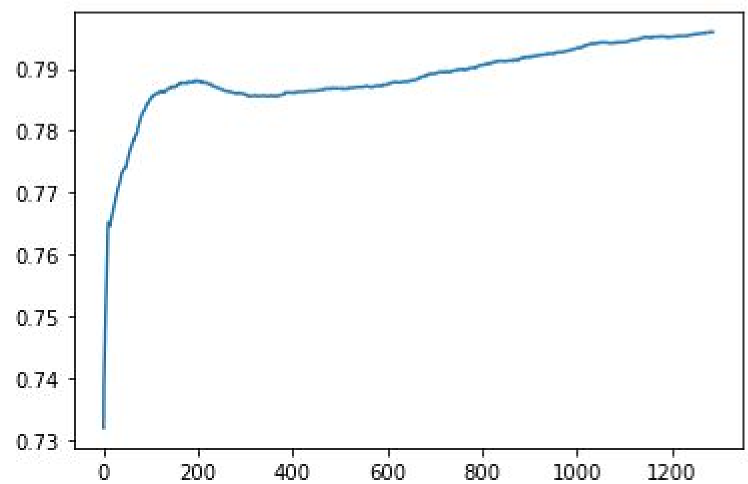
Regularization parameter = 10 ^ -1:



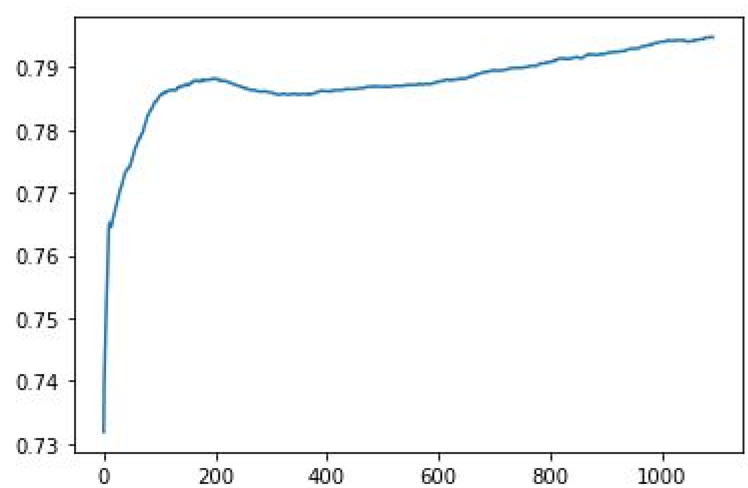
Regularization parameter = 10 ^ -2:



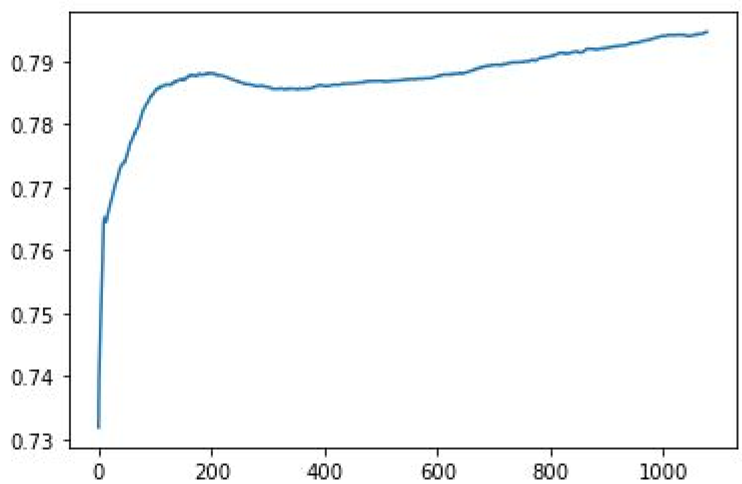
Regularization parameter = 10 ^ -3:



Regularization parameter = 10 ^ -4:



Regularization parameter = 10 ^ -5:



According to the above plots, the accuracy is also closer to 0.8, and when the Regularization parameter is smaller than 10 ^ -3, the iteration that is needed for convergence will dramatically decrease.

According to the above plots of training and validation, we can find that reducing the regularization parameter (λ) will slightly increase the accuracy of predictions, meanwhile, if the regularization parameter smaller than some thresholding, its performance of training will dramatically improve

I think the reason why this phenomenon show this trend is because the train data matrix is sparse. The norm of some vectors is pretty small, furthermore, this small norm will be used as the denominator in the sigmoid function, which causes a negative effect to convergence.

I think the sweet spot of λ and convergence is λ = 10^-5 and convergence = 10 ^ -2

3. For the best model selected in (b), sort the features based on|wj|. What are the top 5 features that are considered important according to the learned weights? How many features have wj= 0? If we use a larger λ value, do you expect more or fewer features to have wj= 0?

Top 5

4	2.32783	Previously_Insured
5	1.9015	Vehicle_Damage
2	0.590621	Age

195	0.581757	Policy_Sales_Channel_160
187	0.574234	Policy_Sales_Channel_152

features that wj= 0

89	0	Policy_Sales_Channel_28
152	0	Policy_Sales_Channel_110
163	0	Policy_Sales_Channel_123

The number of 0 does not seem to change much, here are 3 for any regularization rate in L2.

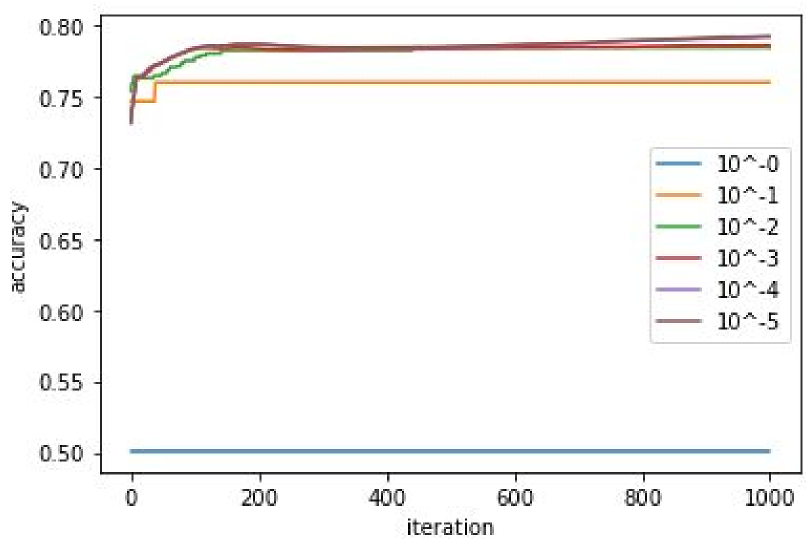
The full table can be found in *full_table.xlsx*

Part 2 Logistic Regression with L1 (Lasso) regularization

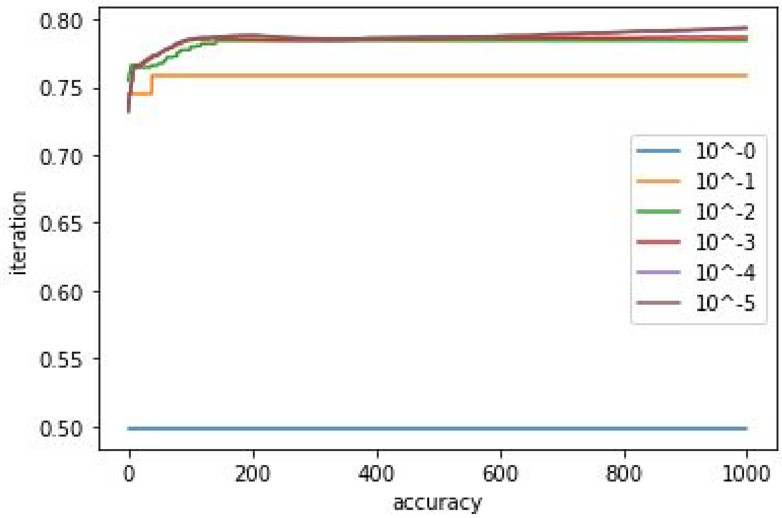
1. implement Algorithm 2 and experiment with different regularization parameters λ∈{10−i: I∈[0,5]}.
   Implementation can be found in the *part2_final.py*, experiment result will be shown in b)

2. Plot the training accuracy and validation accuracy of the learned model as the λ value varies. What trend do you observe for the training accuracy as we increase λ? Why is this the case? What trend do you observe for the validation accuracy? What is the best value based on the validation accuracy?

   Training accuracy (learning rate = 0.1)



Validation accuracy (learning rate = 0.1)



The result is somewhat similar to the former L2 experiment. When the regularization parameter is just 1(10^0) which means without regularization, the convergence speed is very low, which makes the awful accuracy of prediction in the small iteration. As the regularization rate decreases, the convergence rate will gradually increase, as well as the accuracy improvement.
I think the reason why this phenomenon show this trend is because the train data matrix is sparse. The norm of some vectors is pretty small, furthermore, this small norm will be used as the denominator in the sigmoid function, which causes a negative effect on convergence.
Based on the observation, similar to L2, 10 ^ -5 is my preferred regularization rage, because it is excellent in performance and accuracy.

3. For the best model, sort the features based on|wj|. What are the top 5 features that are considered important? How many features have wj= 0? If we use a largerλvalue, do you expect more or fewer features to havewj= 0?
Top5 for L1, λ = 10^-5

NO	Weight	Feature
4	2.27952	Previously_Insured
5	1.87347	Vehicle_Damage
187	0.571128	Policy_Sales_Channel_152
2	0.555273	Age
195	0.550357	Policy_Sales_Channel_160

Zero features (39) for L1, λ = 10^-5

65	0	Policy_Sales_Channel_2
88	0	Policy_Sales_Channel_27
89	0	Policy_Sales_Channel_28
93	0	Policy_Sales_Channel_32
98	0	Policy_Sales_Channel_39
99	0	Policy_Sales_Channel_40
101	0	Policy_Sales_Channel_43
103	0	Policy_Sales_Channel_45
104	0	Policy_Sales_Channel_46
106	0	Policy_Sales_Channel_48
107	0	Policy_Sales_Channel_49
115	0	Policy_Sales_Channel_57
116	0	Policy_Sales_Channel_58
120	0	Policy_Sales_Channel_62
121	0	Policy_Sales_Channel_63
125	0	Policy_Sales_Channel_68
126	0	Policy_Sales_Channel_69
129	0	Policy_Sales_Channel_78
131	0	Policy_Sales_Channel_81
133	0	Policy_Sales_Channel_87
135	0	Policy_Sales_Channel_89
138	0	Policy_Sales_Channel_92
139	0	Policy_Sales_Channel_93
140	0	Policy_Sales_Channel_94
142	0	Policy_Sales_Channel_96
145	0	Policy_Sales_Channel_100
147	0	Policy_Sales_Channel_103
148	0	Policy_Sales_Channel_106
149	0	Policy_Sales_Channel_107
150	0	Policy_Sales_Channel_108
152	0	Policy_Sales_Channel_110
154	0	Policy_Sales_Channel_113
163	0	Policy_Sales_Channel_123

166	0	Policy_Sales_Channel_126
169	0	Policy_Sales_Channel_129
170	0	Policy_Sales_Channel_130
173	0	Policy_Sales_Channel_133
174	0	Policy_Sales_Channel_134
182	0	Policy_Sales_Channel_146
183	0	Policy_Sales_Channel_147

The number of 0 will increase with the decreasing of the regularization, the number of the feature that weighed as 0 is only 39 when the regularization is 10^-5, but whenever the regularization increases to the 10^-3, the number of 0 weight explodes to 105, then there are just 2 non-zero weights when the regularization parameter is 10^-1.
The full table can be found in *full_table.xlsx.*

4. Compare and discuss the differences in your results for Part 1 and Part 2, both in terms of the performance and sparsity of the solution.
   Both L2 (ridge) and L1 (Lasso) has similar accuracy, which is closer to 80%. However, it is clear that the L1 has a better performance and utilization of memory(as far as I know, at least Python and Matlab provide sparse matrix type).