PART 1

$$\begin{aligned}
\hat{\beta}_{\text{MAP}} &= \arg\max_{\beta} \; P(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\beta})P(\boldsymbol{\beta}) \\
&= \arg\max_{\beta} \; \log P(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\beta}) + \log P(\boldsymbol{\beta}) \\
&= \arg\max_{\beta} \; \sum_{i=1}^{m} [\log P(y_i|x_i;\beta_i) + \log P(\beta_i)] \\
&= \arg\max_{\beta} \; \sum_{i=1}^{m} [-\frac{(y_i-\boldsymbol{\beta}^{\mathrm{T}}x_i)^2}{2\sigma^2} - \tau|\beta_i|] \\
&= \arg\min_{\beta} \; \sum_{i=1}^{m} [(y_i-\boldsymbol{\beta}^{\mathrm{T}}x_i)^2 + 2\sigma^2\tau|\beta_i|] \\
&= \arg\min_{\beta} \; \sum_{i=1}^{m} (y_i-\boldsymbol{\beta}^{\mathrm{T}}x_i)^2 + \lambda||\boldsymbol{\beta}||_1 \quad (\lambda = 2\sigma^2\tau)
\end{aligned}$$

As we can see, the result can be expressed in the form of Lasso regression.

PART 2

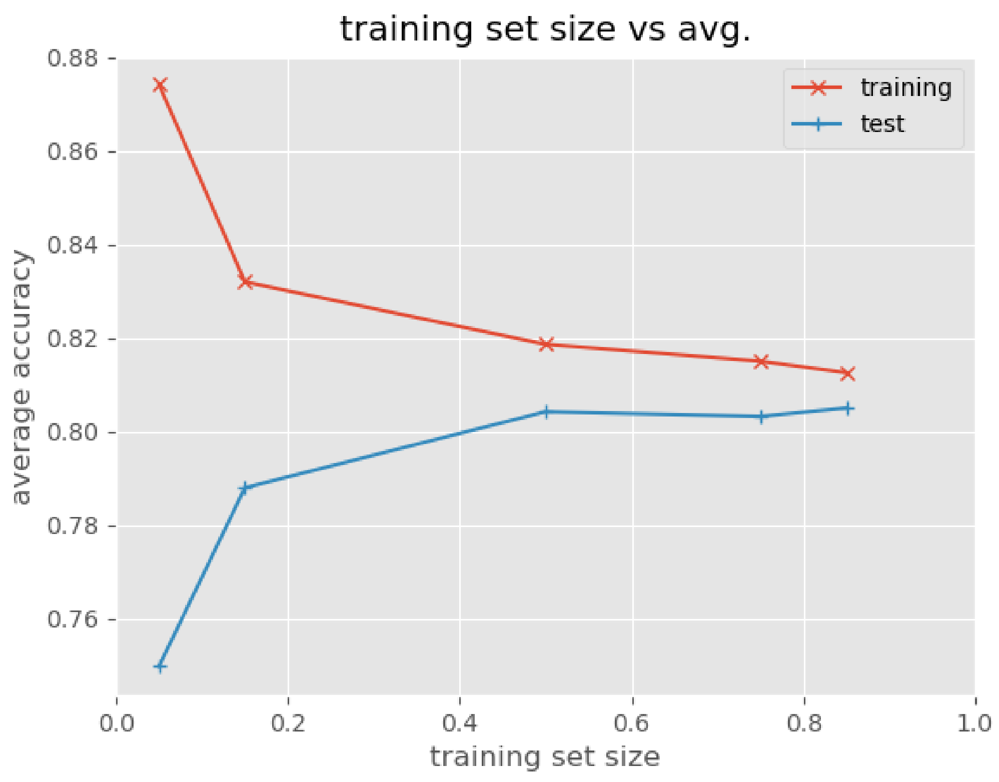
1. Here is the results for (a.):

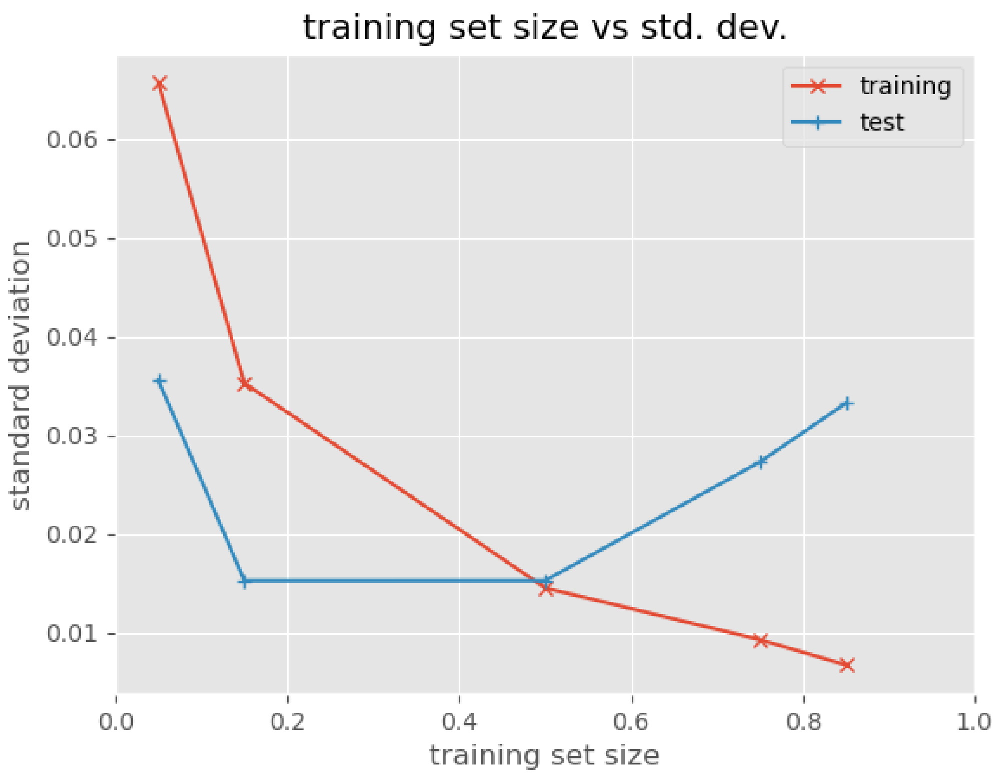
```
The average accuracy on training data:
[0.8831428571428571, 0.828952380952381, 0.8178571428571427,
0.8161333333333335, 0.8123737373737372]
The standard deviation of accuracy on training data:
[0.06615936233436501, 0.03545416382353847, 0.014178886286536425,
0.009847655890140165, 0.007215037277474644]
The average accuracy on test data:
[0.7505263157894736, 0.7898487394957983, 0.8062000000000001,
0.8030857142857144, 0.8065094339622643]
The standard deviation of accuracy on test data:
[0.02876586000438843, 0.014029174560901653, 0.01454089237359276,
0.02422237117431162, 0.033420601913392745]
```

I display the data in its initial form without rounding up to 2 decimal places because I found when I did that, something went wrong when plotting the curves. Also, the results may change slightly when running the code again, but that makes no difference.

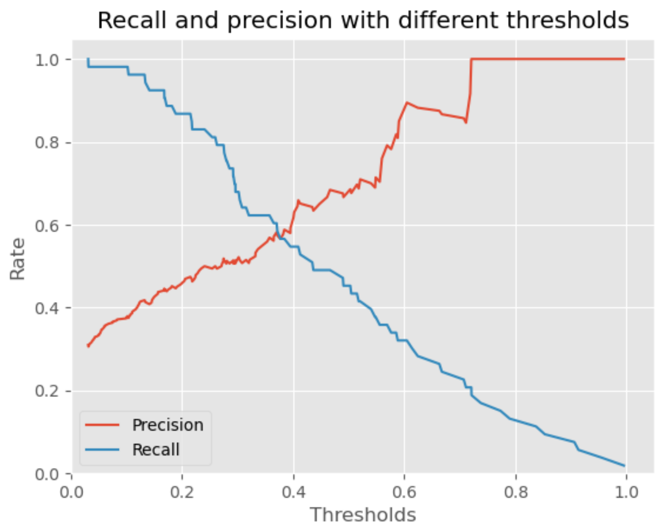
The following two pictures visualize the relationship between training set size and average accuracy (or standard deviation).

In each picture, there are two lines representing the training data set and test data set respectively. In the first picture, we can simply draw the conclusion that the average accuracy of training data decreases with the increase of training data set, while the average accuracy of test data set is just the opposite. In the second picture, we can observe that the std. dev. of training set also decreases with the increase of training data set, while the std. dev. of test set has no relationship with the proportion of training data set.



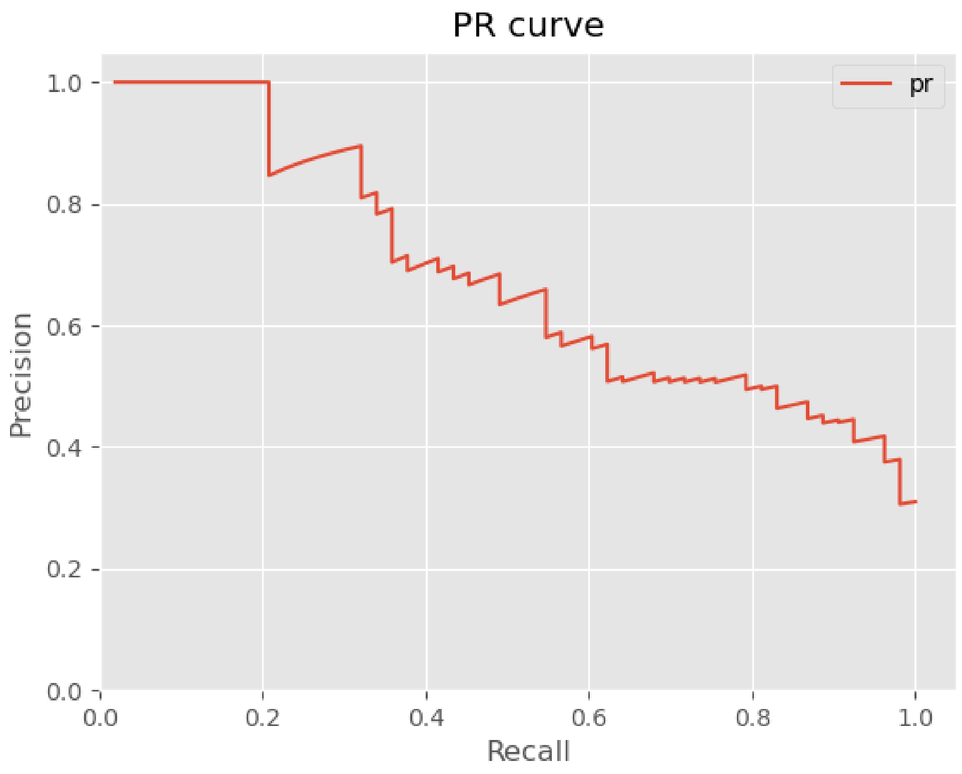


2. First I set the thresholds with different values stepping from 0 to 1, and plot the trend of recall and precision rate.

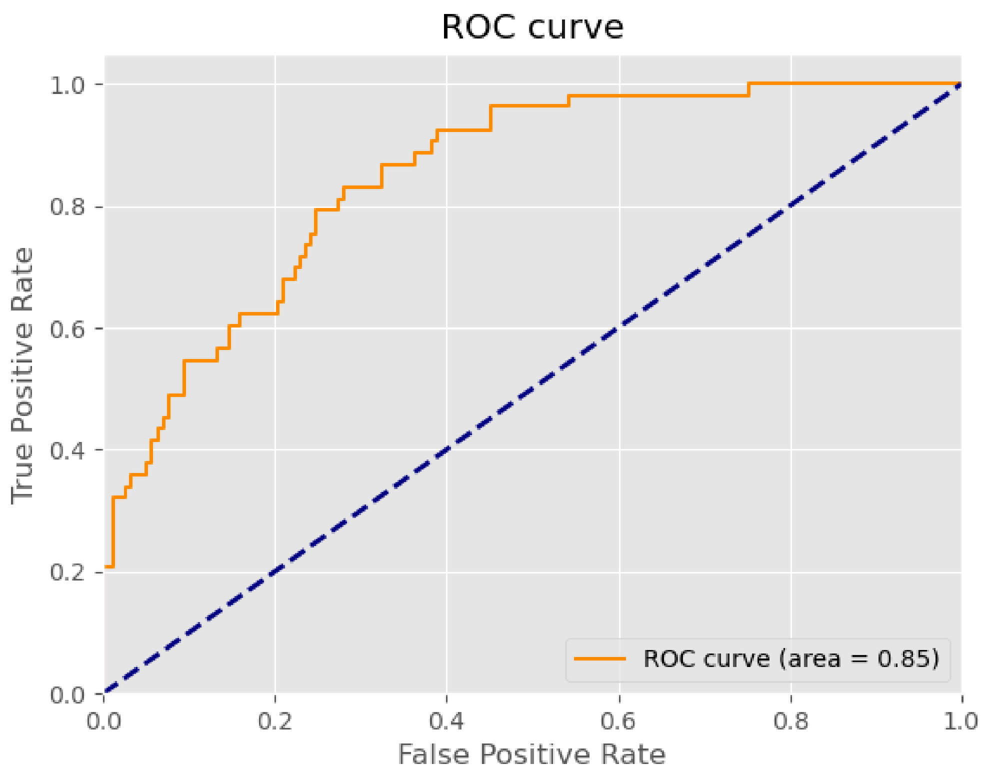


The result is easy to understand. When I set a threshold much higher than it should be (say, very close to 1), most of the samples will be labeled as 0, even if the probabilities are relatively high. Remember that the precision measures the proportion of all positive predictions that are correct, and recall measures the proportion of how many actual positive cases have been found out. In this case, the precision rate is approximately close to 1 and the recall rate close to 0.

Then I further plot the PR curve, which shows how the two metrics change simultaneously with the change of threshold.



Finally, I plot the ROC curve and calculate the AUC with the help of the functions provided by sklearn.



The ROC curve lies above the baseline and the AUC of this curve is 0.85. Usually, when the AUC is greater than 0.85, we can deduce that the model performs quite well. Therefore, I think the result is acceptable. Once we ensure that our algorithms will always converge to the optimal point, then the AUC is related to the data set itself. If the predicted value has a strong causality with the dependent variables (in this case for instance, the probability whether a person will default is negatively related to his wage, according to our empirical knowledge), then we'll get a higher AUC and a more meaningful result.