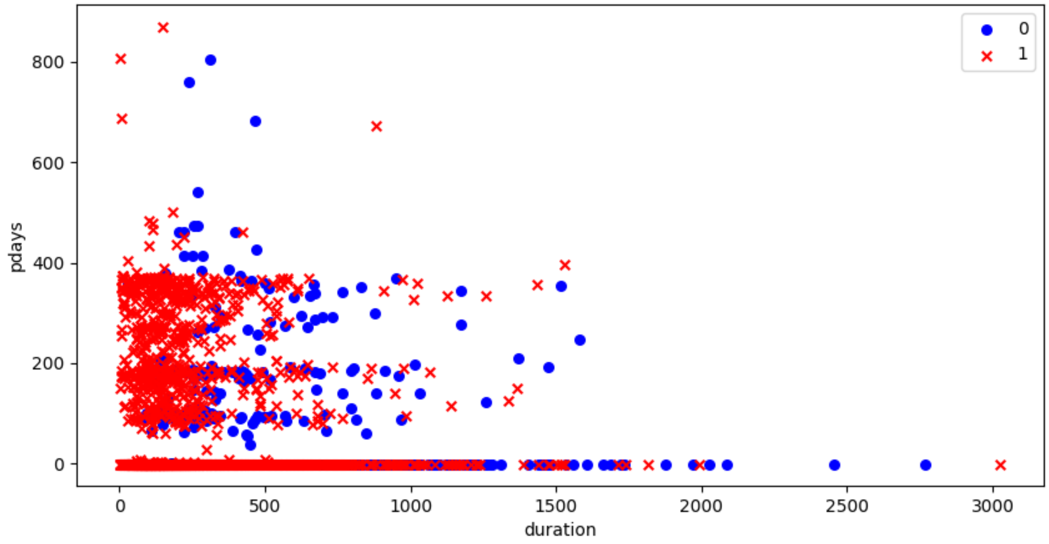
This assignment was completed by Mei Zhengkun, Yiqiu Wang, and Iva Dimitrova.

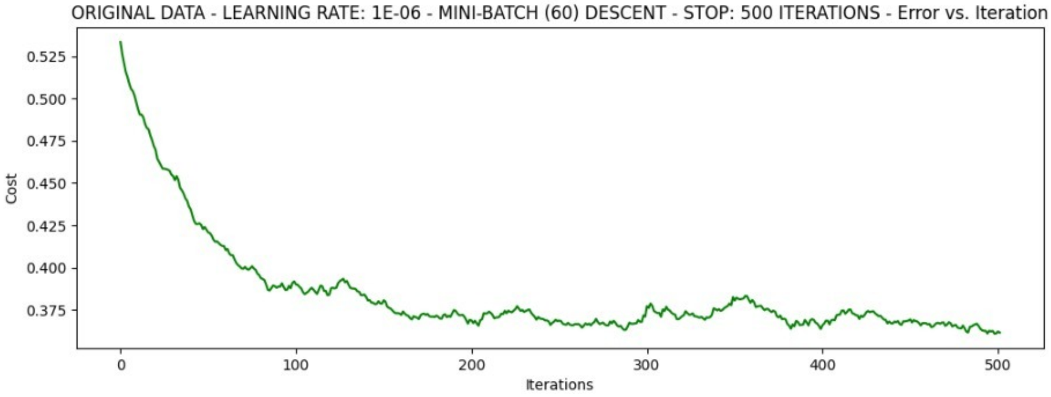
For the classification task, we decided to use the “Bank Marketing” database from a Portuguese bank institution. It contained in total 45211 samples and 16 features. However, we decided to select only two features, the number of days that passed by after the client was last contacted from a previous campaign and the duration of the last interaction with the client. We decided to select these two features amongst all because we judged they would affect the output of the classification task the most. The goal of this binary classification task was to predict if the client was going to subscribe to a term deposit or not. A term deposit refers to a certain amount of money deposited in a bank account that grows with a specific interest rate. However, the money must be kept in the account for a fixed period in order to receive the interest in full.

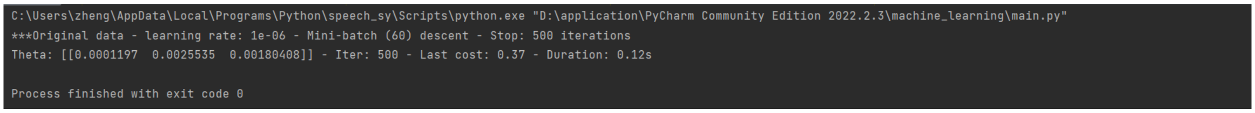
For this classification task we used both a Support vector machine algorithm(which only using the SVM function provided by the python itself) as well as a Logistic Regression(which we create for ourselves). The reason why we make the small coding line SVM model is that we also want to compare our result produced by the Logistical regression model with the model others created and the model we create for ourselves. Because the other model may not use the features(duration and pdays) as we choose, so the SVM model was designed just using the same features as we did in logistical one, which controls the variable. That is why we create this simple SVM model too.



For less difficulty of designing the code(cause I think we have more flexibility to deal with numbers rather than the string), we change the last line(yes and no question) to zero and one. That is why we have two dataset(bank and bank1) in our zip file, the bank data is the original one and the bank1 is the dataset changing the final column. In the logistical one model, we use the gradient descent method to let the machine learn how to decrease the cost.

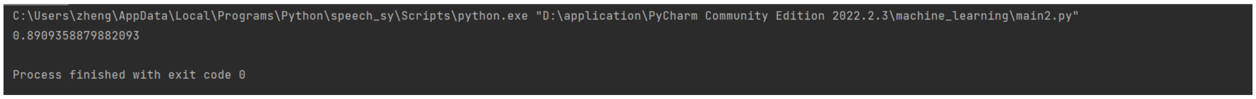
For comparing the result from logistical regression model with the simple SVM model created by ourselves, the SVM algorithm showed an accuracy of approximately 89% while the Logistic Regression algorithm had a cost function of 0.37 and an accuracy of 63%. In conclusion, the SVM results were significantly higher than the Logistic Regression for this dataset, maybe the logistical one is not very suitable for solving the binary classification problem, cause the accuracy for logistical model is just too low, I know most of my classmates just got the accuracy at least more than 70 percent, and we also tried to change the iteration time but it didn't work, I think this means we should improve our model afterwards. The following graph shows the reduction of the cost function.





```
C:\Users\zheng\AppData\Local\Programs\Python\speech_sy\Scripts\python.exe "D:\application\PyCharm Community Edition 2022.2.3\machine_learning\main.py"
***Original data - learning rate: 1e-06 - Mini-batch (60) descent - Stop: 500 iterations
Theta: [[0.0001197  0.0025535  0.00180408]] - Iter: 500 - Last cost: 0.37 - Duration: 0.12s

Process finished with exit code 0
```



```
C:\Users\zheng\AppData\Local\Programs\Python\speech_sy\Scripts\python.exe "D:\application\PyCharm Community Edition 2022.2.3\machine_learning\main2.py"
0.8909358879882093

Process finished with exit code 0
```

For comparing our result with the model developed by other one, the model we choose is alcompa(https://github.com/alcompa/bank-deposit-classification) Our results showed a lower accuracy rate in comparison to the ones reported by Alberto

Compagnoni who conducted a Logistic Regression on the same database. We also add the report file from this model to our zipfile and you can find it. He reported an accuracy of 81% with the Logistic Regression algorithm. Moreover, he used more algorithms to compare the results. He concluded 89% accuracy with KNN, 86% accuracy with DT, and 88% with Naive Bayees. The accuracy works better for his logistical model, but we did not use the same features, we only use two features and if the features we pick influence less on the final result, the performance of our model could be very bad. That is the reason why I assume our logistical model have the lower accuracy. And for our SVM model, it used the SVM function provided by python, it has the similar accuracy compared to the KNN, Naive Bayees and Logistical Regression model, and I think the KNN one has the best performance. So we assume maybe for the simple binary classification problem, the lazy algorithm would have a better performance, but it is only a guess.

In conclusion, the SVM model we built had the similar accuracy rate as the KNN, DT and logistical model built by others, among them the KNN have the best performance which is about 89%. For the logistical model we built, the performance is not expected, having a clear lower accuracy than the other one, and than the same logistical model built by others. To improve the classification performance of our model, we should consider more features of the dataset which could make the machine classify the dataset easily.