MINI-PROJECT 1 REPORT

Edmund Wu
261033501

Tianyi Xu
261082272

Santosh Passoubady
261098017

October 8, 2023

ABSTRACT

At the basic level, machine learning stems from statistical learning. We implement two types of statistical models: linear regression and logistic regression, and optimize them on two benchmark datasets. After training, we evaluate their performance following a standard presentation. After testing, we observe the phenomena of overfitting and discuss on ways to prevent it. We also discuss on different factors contributing to different model performances. Furthermore, we optimize the two models in several ways. We implement mini-batch gradient descent, we add a penalty, and we include momentum as optimizer.

1 Introduction

It is in our interest to evaluate a common and perhaps the most basic types of model: linear regression and logistic classification. Although simple, both models are very thoroughly studied. Both models are basic building blocks to more structured deep neural networks. Indeed, we can analogously think of linear regression and logistic classification respectively as a linear layer and a softmax activation layer (in this case, with a linear layer before) in a machine learning context. A thorough understanding of linear regression and logistic classification hence allows us to have a strong base to develop even slightly more complicated models.

This mini-project consists of three items: (1) an analytic linear least squares, (2) a gradient descent algorithm for logistic classification, and (3) an optimization through mini-batch gradient descent. All three supervised learning algorithms are then tested on two separate datasets of small size, (i) a dataset on Boston housing and (ii) a dataset on wine classification. Performance observations are then noted. A list of our peripheral tasks is as follows. We have processed that datasets, split into train/test sets, normalized the data, trained the models, implemented a 5-fold cross-validation, tested different hyper-parameters, and evaluated model performance under different hyper-parameters. We have also removed outliers and included an intercept/bias. In addition, even though the L1 penalty induces sparsity and is more relevant, we have decided to add an L2 penalty to the cost function in classification. We have additionally extended mini-batch stochastic gradient descent with momentum. We should also precise that rather than implementing a logistic classifier, a more correct term in our opinion is a soft-max classifier as the dataset consisted of 3 classes.

The two datasets used are cited in the references. Both datasets have been thoroughly exploited, and numerous works that use the datasets already exist, and can be found directly from the link in the references 1 2.

2 Datasets

We have tested the linear regression model on a dataset called *Boston Housing Prices* and the logistic classification model on a dataset called *Wine*.

2.1 Dataset 1: Boston Housing dataset

The *Boston Housing Prices* dataset originally sourced from an economics and management textbook, is made of 13 attributes and 1 target response with a sample size of 506. As mentioned, we removed one column ('B') for ethical reasons, leaving us with 12 attributes. We would like to use linear regression to relate these 12 attributes with the response denoted 'MEDV', the median value of owner-occupied homes in thousands of dollars.

2.2 Dataset 2: Wine dataset

This dataset comprises 178 data points, each characterized by 13 attributes with a total of 3 classes. Our goal is to predict the origin/class of these wine samples, which is a multi-class classification task with three distinct classes.

2.3 Data Processing

Both datasets are processed in a similar manner, differing only at the end.

- **Data Preparation and Exploration for both Datasets**
  First, data were loaded and basic data exploration on the dataset dimensions was performed, which informed us of no missing values. Then, further analysis of each feature column of both datasets was visualized through density bar plots.

  For the Boston dataset, KDE plots on each feature were created to further explore the skewness in the data.
- **Feature-Target Exploration**
  Since the target variable of the Boston dataset ('MEDV') is continuous whereas that of the wine dataset('class') is discrete, we visualized their feature-target relationship by plotting scatter plots for Boston and bar plots for wine respectively.
- **Outlier handling**
  From previous steps, we found some features to be skewed in the Boston dataset. By creating box plots, we visualized outliers for each feature. We then applied **Tukey's rule 3 to remove the outliers** and enhance data quality.
- **Data Splitting and Normalization**
  We first split both datasets into training and testing sets before normalizing them for feature scaling to avoid contaminating the dataset. Normalization is performed so as to speed up training. Then, we separated the target variable from the input features.

2.4 Ethical concerns

We have removed column feature 'B' that keeps track of the proportion of black people in town in the Boston housing dataset as well as a very small amount of outliers in the datasets. The significance of removing these data lies in the fact that such data could induce our model to be biased for future use and may give rise to societal concerns.

3 Results

We used the mean-squared error method to evaluate our linear regression model performance. A lower MSE value indicates a better model performance. We evaluated our logistic regression model through four separate metrics: accuracy, precision, recall, and F-1 score. In all four aspects, a higher score corresponds to a better model performance.

We acknowledge that during this mini-project, we referenced Pedro Domingos' paper titled "A Few Useful Things to Know About Machine Learning". 4

The set of parameters that were used consistently for the corresponding models across the experiment are defined in Appendix A.

3.1 Experiment 1

(a) **Table 2** Mean Squared Error (MSE) Of Training and Test Sets for Linear Regression Model on Boston Data set

Set	MSE
Training	18.677
Test	28.968

(b) **Table 3** Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Logistic Regression Model on Wine Data set

Set	Accuracy	Precision	Recall	F-1 score
Training	0.990	0.986	0.986	0.986
Test	0.981	0.972	0.972	0.972

From the results shown in Table 2, we see that the training M.S.E. is much lower than the test M.S.E. for our linear regression model. From the results shown in Table 3, our logistic regression model achieved higher scores in all four evaluation metrics on the training data as opposed to the testing data.

We extended our experimentation to Mini-batch Stochastic Gradient Descent on both the Boston Data set (B.1) and the Wine Data set (B.4). Furthermore, we investigated the performance enhancement achieved by incorporating Momentum, as demonstrated in Appendix B.3 and Appendix B.6. Due to the high loss variability of these models, it was challenging to decisively establish their superiority, but they consistently achieved comparable losses to the analytical solution.

Later, we ran the Linear Regression and Logistic Regression model with and without Regularization (B.2 and B.5)using Mini-Batch Stochastic Gradient Descent, but saw no significant changes in the results.

3.2 Experiment 2

(a) **Table 4** 5-Fold Cross-Validation: Average Training, Validation, and Test MSE for Linear Regression on Boston dataset

Training Set	Validation Set	Test Set
18.427	20.992	29.253

(b) **Table 5** 5-Fold Cross-Validation: Average Training, Validation, and Test Accuracy for Logistic Regression on Wine dataset

Training Set	Validation Set	Test Set
0.982	0.957	0.967

From the result shown in Table 4, we see that the MSE is the smallest in the training set, larger in the validation set, and the biggest in the test set. From the result shown in Table 5, we notice that the accuracy score is largest in the training set, smaller in the validation set, and lowest in the test set.

From results in our Google Colab, the best logistic regression model is obtained with a learning rate equal to 0.002230. Testing the model right after, the model performs well, giving $138/140$ correct predictions in the training set, and $35/36$ correct predictions in the test set.

Similarly, the best learning rate for linear regression obtained is 0.006667. In this case, we obtain a 5-fold average mean squared error of 5.583697, and predicted values are fairly accurate as shown in the Appendix C.

3.3 Experiment 3

(a) Influence of Training Data Size on Linear Regression Loss

(b) Influence of Training Data Size on Soft-max Loss

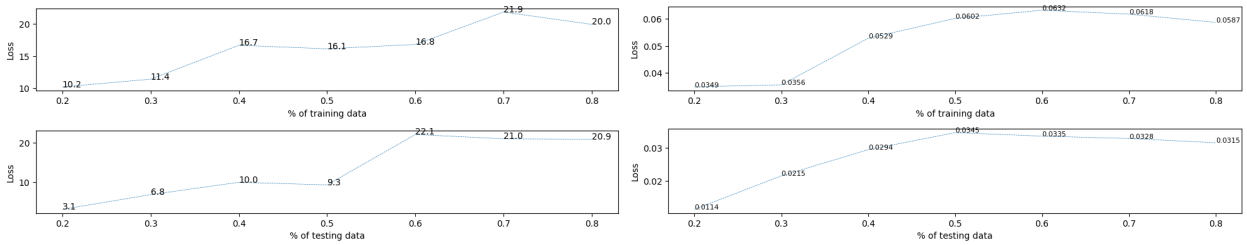


Figure 1: Influence of Training Data Size on Loss for both Data sets

In Figures 1a and 1b, a consistent pattern emerges: when assessing loss with an expanding subset of training data, it initially exhibits a low value. The initial low loss signifies over fitting, indicating that the model fits the limited dataset well but struggles with effective generalization.

As we increase the amount of data, the loss initially rises, reflecting the model's challenges in generalizing effectively. Subsequently, it enters a phase of more gradual decline. In the case of Dataset 1, this occurs when approximately 70% of the training data is used, while for Dataset 2, it happens at around 60% of the training data. We expect the loss to steadily decrease until it approaches an optimal minimum.

3.4 Experiment 4

(a) Influence of Mini-batch Size on our Linear Regression Loss (smoothed) with Mini-Batch Stochastic Gradient Descent

(b) Influence of Mini-batch Size on our Logistic Regression Loss (smoothed) with Mini-Batch Stochastic Gradient Descent

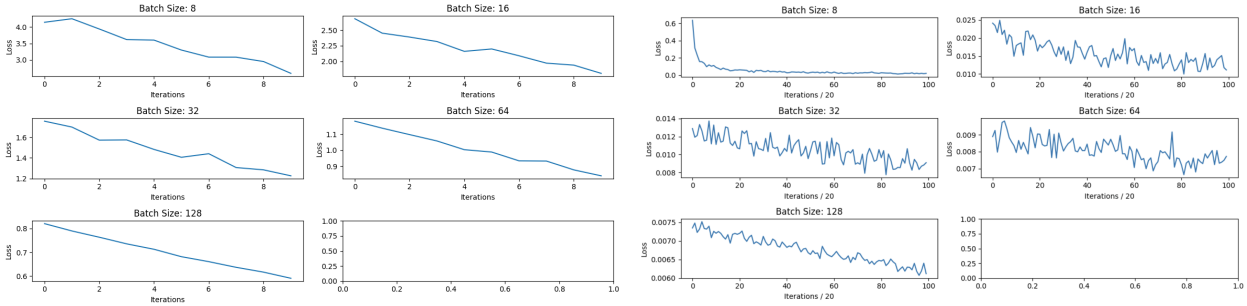


Figure 2: Influence of Mini-Batch Size on Loss for both Data sets

From Figures 2a and 2b, we notice that as we increase the mini-batch size, the noise in the loss of each iteration decreases. Moreover, we observe that there exists a general negative correlation between the iteration and the loss value for all graphs in Figures 2a and 2b.

Generally, mini-batch gradient descent shows faster convergence than batch gradient descent. A small batch size increases convergence rate, but also gives a more approximate update step. This may not be clearly observed from our experiments, perhaps due to the simplicity of the model.

3.5 Experiment 5

(a) Smoothed Loss Function of Linear Regression with Mini-Batch Stochastic Gradient Descent with Learning Rate: 1e-1, 1e-3, 1e-5

(b) Smoothed Loss Function of Logistic Regression with Mini-Batch Stochastic Gradient Descent with Learning Rate: 1e-1, 1e-3, 1e-5

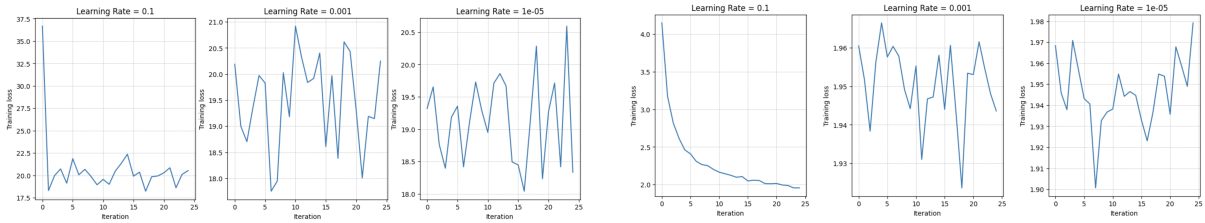


Figure 3: Influence of Learning Rate on Loss for both Data sets

In both 3a and 3b, one may observe that an optimal learning rate for both linear and logistic regression seems to lie somewhere between 1e-1 and 1e-3. The figures plot the smoothed-out loss curves taking the mean over 200 losses in a total of 5000 iterations.

3.6 Experiment 6

We performed cross-validation using different learning rates ranging from 1e-2 to 1e-5 and picked the one that gave us the lowest average cross-validation error, as is reported in experiment 2.

We chose the average cross-validation error as our performance metric because it provides a robust assessment of a model's generalization ability across different parameter configurations. By selecting the learning rate that yields the lowest average cross-validation error, we prioritize a parameter choice that optimizes the model's overall predictive accuracy while avoiding overfitting the training set.

(a) **Table 6** Mean Squared Error (MSE) Of Training and Test Sets for Our Linear Regression Model with Mini-Batch Stochastic Gradient Descent Using Optimal Parameter

Set	MSE
Training	20.258
Test	29.789

(b) **Table 7** Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Our Logistic Regression Model Using Optimal Parameter

Set	Accuracy	Precision	Recall	F-1 score
Training	0.990	0.986	0.986	0.986
Test	0.981	0.972	0.972	0.972

In Table 6, we observe that the optimal model's MSE is a bit higher than that of our baseline model from Experiment 1. This can be attributed to the inherent randomness within the cross-validation process and the model's training on a reduced dataset. It is noteworthy that Mini-Batch Stochastic Gradient Descent led to superior training results, albeit with a slightly elevated test MSE, as demonstrated in the Appendix D.1.

In Table 7, we achieve performance similar to the already high baseline model from Experiment 1. Notably, the use of both Mini-Batch Stochastic Gradient Descent D.2 and Mini-Batch Stochastic Gradient Descent with Momentum D.3 yields identical results in the test set, although it leads to a perfect score in the training set, indicating a modest improvement.

3.7 Experiment 7

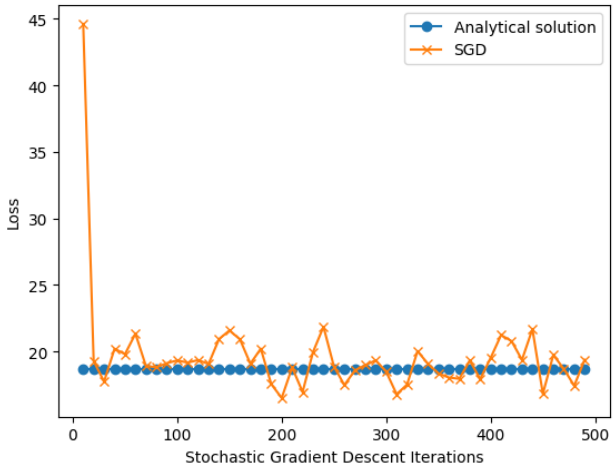
Table 8: Linear Regression MSE: Original vs. Enriched Training Data with Gaussian Basis

Set	MSE
Original training set	18.677
Enriched training set	16.802

From the results shown in Table 8, we see that the MSE under the original Boston dataset is higher than that under the Boston dataset that is enriched with Gaussian basis functions.

3.8 Experiment 8

Figure 4: Loss (MSE) Comparison: Linear Regression vs. Mini-Batch Stochastic Gradient Descent Across Iterations



From Figure 8, we observe that the loss value for our analytical solution remains the same. The loss value for our stochastic gradient descent starts off very high and then oscillates around the loss value found through the analytical solution as the number of iterations increases.

4 Discussion and Conclusion

Here are our takeaways from all the experiments we have made:

- **Experiment 1**
  We set baseline values for future comparisons with both models. The removal of regularization did not notably impact model performance for both datasets. Additionally, employing Mini-Batch Stochastic Gradient Descent (even along with the addition of Momentum) did not substantially enhance model outcomes based on our observations.
- **Experiment 2**
  Cross-validation results indicated that while the linear regression model showed a slightly higher test MSE compared to the baseline, this could be due to random chance and a smaller training dataset. The logistic model had lower accuracy but may generalize better for future data, suggesting that the original high baseline accuracy might indicate overfitting.
- **Experiment 3**
  As the training dataset size grows, models tend to initially overfit, fitting the limited data well but struggling to generalize. Yet, with more data, the loss increases due to better generalization before gradually decreasing, leading to improved model performance and an eventual minimum loss.
- **Experiment 4**
  Using larger batch sizes resulted in diminished noise in loss updates, reduced final loss values for both linear and logistic regression models, and ultimately accelerated convergence, emphasizing the benefits of using larger batches during training.
- **Experiment 5**
  Properly selecting the learning rate is crucial; too small of a learning rate leads to slow convergence, while too large of a learning rate can cause overshooting the minimum during optimization.
- **Experiment 6**
  Opting for the average cross-validation error as our performance metric proved to be a sound choice, yielding satisfactory results. While Linear Regression exhibited a slightly higher MSE than the baseline, for the same reason as in Experiment 2, Logistic Regression matched our previously identified performance, which was already close to a perfect score.
- **Experiment 7**
  Using Gaussian basis functions can enhance model performance, especially when dealing with non-linear relationships between input and output, enabling a linear model to capture more complex patterns.
- **Experiment 8**
  Mini-batch Stochastic Gradient Descent may oscillate around the analytical solution, but it's a practical choice for large datasets where the analytical solution is computationally expensive, offering a scalable optimization method. This is particularly evident since Linear Regression with SGD's time complexity is $\mathcal{O}(ND^2)$ compared to the analytical solution's $\mathcal{O}(ND^2 + D^3)$ (as seen in class 5), highlighting SGD's scalability advantages.

Future investigations could involve delving into deep neural networks to leverage their capability to capture complex patterns and perform comparisons with our current models. Additionally, probing into parameter tuning for these deep learning models, including factors such as learning rate, training data size, and mini-batch size, holds promise for enhancing their robustness on the datasets.

5 Statement of Contributions

Each member contributed equally to this project. Each of us completed this lab individually and wrote the report by merging our responses and code.

References

[1] The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics and Management, vol.5, 81-102, 1978. http://lib.stat.cmu.edu/datasets/boston

[2] Wine dataset for classfication. 1991. https://archive.ics.uci.edu/dataset/109/wine

[3] Tukey, J. W. (1977). Exploratory Data Analysis. Addison-Wesley.

[4] Domingos, P. (2012). A Few Useful Things to Know About Machine Learning. Communications of the ACM, 55(10).

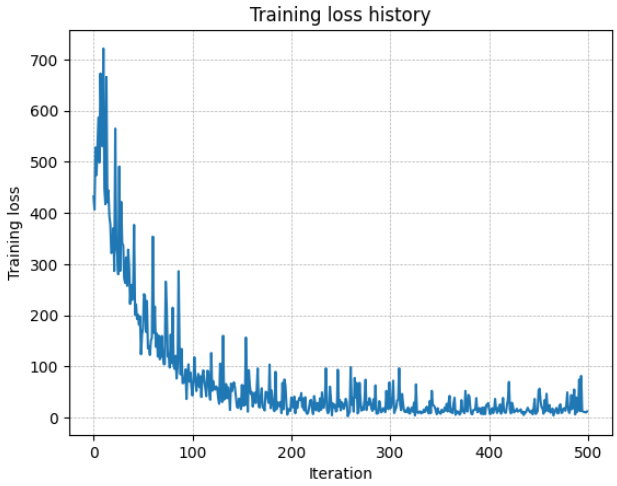
[5] Prémont-Schwarz, I. (2023). Gradient Descent. COMP 551, McGill University.

Appendix A Default Model Parameters

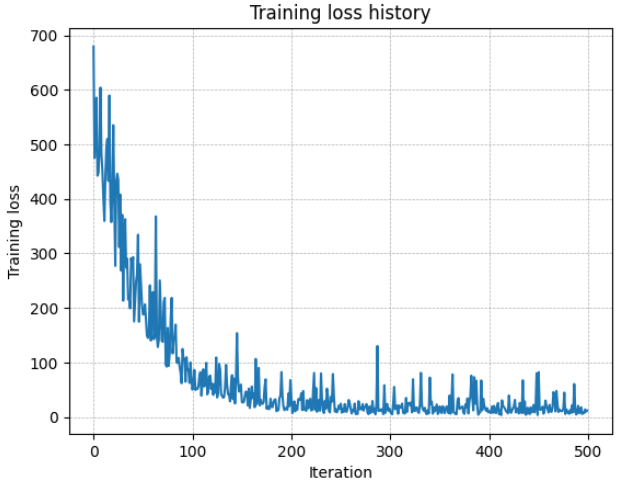
Model	Max iteration	Batch Size	Learning Rate	Epsilon	Regularization Term
Linear Regression with SGD	500	16	1e-2	1e-5	0
Linear Regression with SGD and Momentum	500	16	1e-2	1e-5	0
Logistic Regression	5000	16	1e-3	1e-7	0.1
Logistic Regression with SGD	5000	16	1e-3	1e-7	0.1
Logistic Regression with SGD and Momentum	5000	16	1e-3	1e-7	0.1

Appendix B Experiment 1

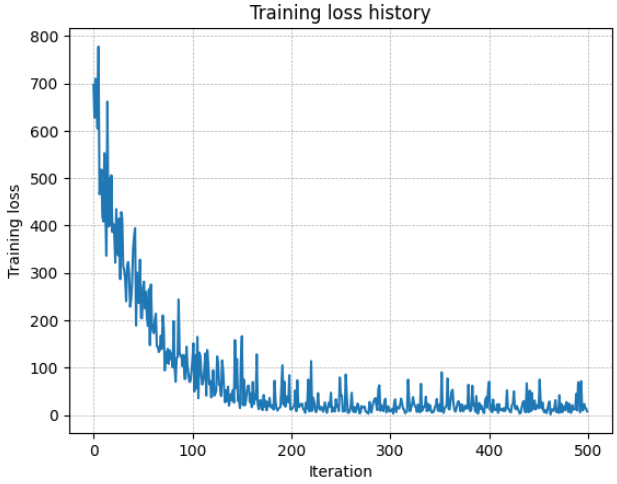
B.1 Training Loss History of Linear Regression with Mini-Batch Stochastic Gradient Descent



B.2 Training Loss History of Linear Regression with Mini-Batch Stochastic Gradient Descent Without Regularization Term



B.3 Training Loss History of Linear Regression with Mini-Batch Stochastic Gradient Descent and Momentum



B.4 Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Our Logistic Regression Model with Mini-Batch Stochastic Gradient Descent

Set	Accuracy	Precision	Recall	F-1 score
Training	0.995	0.993	0.993	0.993
Test	0.981	0.972	0.972	0.972

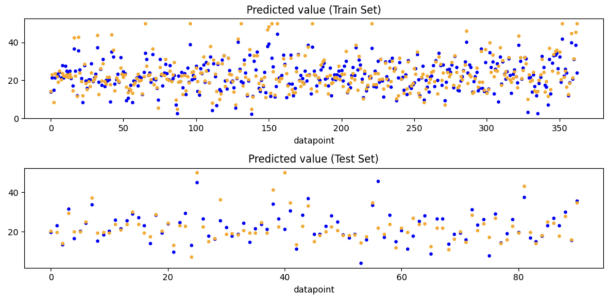
B.5 Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Our Logistic Regression Model with Mini-Batch Stochastic Gradient Descent Without Regularization Term

Set	Accuracy	Precision	Recall	F-1 score
Training	0.995	0.993	0.993	0.993
Test	0.981	0.972	0.972	0.972

B.6 Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Our Logistic Regression Model with Mini-Batch Stochastic Gradient Descent and Momentum

Set	Accuracy	Precision	Recall	F-1 score
Training	0.990	0.986	0.986	0.986
Test	0.981	0.972	0.972	0.972

Appendix C Experiment 2



Blue dots are predicted values, orange are true

Appendix D Experiment 6

D.1 Mean Squared Error (MSE) Of Training and Test Sets for Our Linear Regression Model with Mini-Batch Stochastic Gradient Descent with Momentum Using Optimal Parameter

Set	MSE
Training	19.935
Test	29.840

D.2 Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Our Logistic Regression Model with Mini-Batch Stochastic Gradient Descent Using Optimal Parameter

Set	Accuracy	Precision	Recall	F-1 score
Training	1.0	1.0	1.0	1.0
Test	0.981	0.972	0.972	0.972

D.3 Accuracy, Precision, Recall, and F1-score Of Training and Test Sets for Our Logistic Regression Model with Mini-Batch Stochastic Gradient Descent with Momentum Using Optimal Parameter

Set	Accuracy	Precision	Recall	F-1 score
Training	1.0	1.0	1.0	1.0
Test	0.981	0.972	0.972	0.972