Graded Assignment 1 - Report

-Archit Sangal

Assumptions -

1. In some model not all possible combination are tested as they would take hours for computation, and we got sufficiently efficient model using some features. As done under multivariate section of Gradient Descent.
2. Under multivariate section of Gradient Descent, all the code is there regarding all permutations and combination but in other sections, I ran a loop to get the best possible feature in case of univariate and constructed a model according to it, and then I remove that loop.
3. Due to above reason, it was very difficult to make a plot of cost function vs iterations graph or MLE vs iteration graph, but the data can be printed and plotted easily.
4. Matrices whose inverse needs to be found out have been assumed to be non-singular.
5. Mean and standard deviation calculated during training data is a contributor to model generation. Hence, this same mean and standard deviation is used for testing data as well.
6. Data is randomly distributed before splitting to ensure non-biased properties. Keeping in mind that the aim is not to maximize the training accuracy, but the accuracy on test data that the model has not seen during training. Hence, we prefer the model with better testing data accuracy than training data accuracy.

Methods Used -

1. Pre-Processing -
    - Checking For Null Values
    - Checking For Duplicates
    - One-Hot Encoding
2. Linear Regression -
    - Gradient Descent
    - Closed-form method
    - Newton's Method
3. Classification Problem -
    - Naive Bayes
    - Logistic Regression using-
        1. Gradient Method
        2. Newton's Method

Above Methods are used for both univariate and multivariate.

Results Obtained -

Linear Regression -

Multivariate -

Gradient Descent	Training Error	Testing Error
Mean Squared Error	22.03558259649766	22.58080104224621
Mean Absolute Error	3.238419581720472	3.309702517731528
Mean Absolute Relative Error	16.006038542706797	17.372705914776727

Closed-form method	Training Error	Testing Error
Mean Squared Error	22.033326145902194	22.610248378908178
Mean Absolute Error	3.2410948545018243	3.3147537521930097
Mean Absolute Relative Error	16.026690340512914	17.390370196549863

Newton's Method	Training Error	Testing Error
Mean Squared Error	22.1992313174984	23.252034897651153
Mean Absolute Error	3.2342083405440247	3.3620848286968767
Mean Absolute Relative Error	16.048282676199584	17.611126351087865

Univariate -

Gradient Descent	Training Error	Testing Error
Mean Squared Error	36.69973451507083	41.73681217429911
Mean Absolute Error	4.330540042488694	4.5660072494405854
Mean Absolute Relative Error	20.5386774483734	20.96435989900394

Closed-form method	Training Error	Testing Error
Mean Squared Error	36.69923607550484	41.695856854656114
Mean Absolute Error	4.335996575406678	4.57005932209336
Mean Absolute Relative Error	20.58155949020036	21.00356132617038

Newton's Method	Training Error	Testing Error
Mean Squared Error	36.69953805007493	41.72764548804307
Mean Absolute Error	4.331749443878168	4.566905362284432
Mean Absolute Relative Error	20.54818194541364	20.973048614200387

Classification Problem -

Multivariate -

Naive Bayes	Training	Testing
Accuracy Score	0.9405204460966543	0.9666666666666667

Gradient Method	Training	Testing
Accuracy Score	0.9256505576208178	0.9333333333333333

Newton's Method	Training	Testing
Accuracy Score	0.9256505576208178	0.9333333333333333

Univariate -

Naive Bayes	Training	Testing
Accuracy Score	0.895910780669145	0.8944444444444445

Gradient Method	Training	Testing
Accuracy Score	0.7992565055762082	0.9055555555555556

Newton's Method	Training	Testing
Accuracy Score	0.7992565055762082	0.9055555555555556

Hyperparameter (Learning Rate)

Linear Regression (Gradient Descent) - Multivariate → 0.05 ; Univariate → 0.001
Logistic Regression (Gradient Descent) - Multivariate → 0.01 ; Univariate → 0.01

Observations -

1. Sometimes, time taken by a method depends heavily on the initial values taken. Sometimes, if we use some specific initial values, we may not even get correct outputs.
2. In classification problem, we must remove unique IDs as-
    - This may lead to overfitting of the data
    - It may create problems in removing duplicate rows.
    - It may not be directly related to the data class prediction. This is not a rule, but is often true.
3. Closed Form in linear regression and Naive Bayes in Classification problem are the most efficient with respect to time.
4. For gradient methods, the time taken by the computation decreases with increase in learning rate. But if it is taken too high, we may miss the optimal value we are trying to find.
5. One-hot encoding can be tricky to deal with as for the methods where inverse of matrices are involved, they may produce linearly dependent columns leading to singular matrices. For example, we usually add column of 1 in the matrix **X,** and if we do an operation like C_i replaced with summation of columns of one hot encoding will result in column of 1, leading to a singular matrix.
6. Machine learning is also about experience and practise. We can improve efficiency with these experience and practise.