Homework 1 Report

Jawahar Sai Nathani

1

(a) Explain what the function train_valid_split does and why we need this step.
(Ans) The function train_valid_split splits the whole data set into 2 parts train_set and valid_set which are used to train and test the machine learning model respectively.

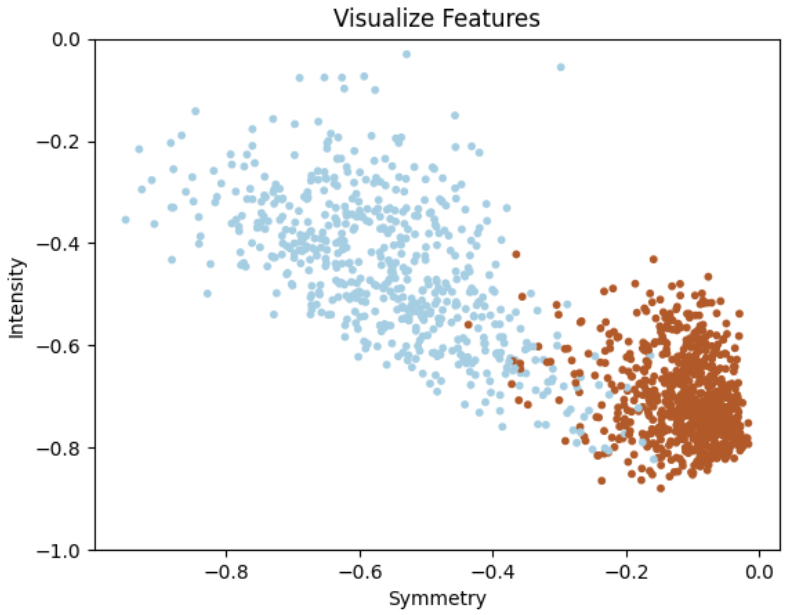
We need this step because of the following reasons

- Analysis on Unseen Data - The ultimate goal of a machine learning model is to predict unseen data. Separating the validation data from training data can help test how our model predicts unseen data.
- Overfitting Prevention - Sometimes machine learning models might capture the noise and irrelevant patterns from the training data very well. This is also called Overfitting. Overfitting can be prevented by testing our model on validation data and checking if the accuracy is similar to training accuracy.
- Hyperparameter Tuning - After training the model using different hyperparameters, all the models can then be tested on the validation data. By comparing the accuracies of different models we can decide the best parameter set for our model.

(b) Before testing, is it correct to re-train the model on the whole training set? Explain your answer.
(Ans) If the training dataset is already large and is similar to the testing dataset without any major differences, it might not be necessary to re-train the model on the whole training set. But if the training data is small or there is a major difference in training and testing sets, then re-training the model on the whole training set might be useful. Because training on more data always helps the model understand the data better and improve the model's accuracy. But there is a chance of overfitting and other problems, hence always double-check the model before using it.

(d) In the function prepare X, there is a third feature which is always 1. Explain why we need it.
(Ans) The third feature is also called bias or intercept. The bias term in the logistic regression equation ensures the model can capture relationships between the independent variables and the probability of the outcome occurring, even when all predictors are at their baseline values, by shifting the logistic curve horizontally. It represents the log odds of the probability that the dependent variable takes on the value of 1 when all independent variables are set to zero.

(f) visualize the training data from class 1 and 2
(Ans)



2

(a) Write the loss function $E(w)$ for one training data sample (x, y). Note that the binary labels are 1 and -1.
(Ans) Cross Entropy loss function for a single data point (x_i, y_i) is

$$E(w) = \ln(1 + e^{-y_i * w^T * x_i})$$

(b) Compute the gradient ∇ E(w). Please provide intermediate steps of derivation
(Ans)

$$\nabla E(w) = \frac{d}{dw}E(w) = \frac{d}{dw}(\ln(1 + e^{-y_i * w^T * x_i}))$$

$$= \frac{\frac{d}{dw}(1 + e^{-y_i * w^T * x_i})}{1 + e^{-y_i * w^T * x_i}} \quad as \quad [\frac{d}{dx}\ln(f(x)) = \frac{\frac{d}{dx}(f(x))}{f(x)}]$$

$$= \frac{\frac{d}{dw}(1) + \frac{d}{dw}(e^{-y_i * w^T * x_i})}{1 + e^{-y_i * w^T * x_i}} = \frac{-y_i * x_i * (e^{-y_i * w^T * x_i})}{1 + e^{-y_i * w^T * x_i}} \quad as \quad [\frac{d}{dx}e^{(f(x))} = e^{(f(x))} * \frac{d}{dx}(f(x))]$$

$$= \frac{-y_i x_i}{(e^{y_i * w^T * x_i}) * (1 + e^{-y_i * w^T * x_i})} = \frac{-y_i x_i}{1 + e^{y_i * w^T * x_i}}$$

$$\therefore \nabla E(w) = \frac{-y_i x_i}{1 + e^{y_i * w^T * x_i}}$$

(c) However, this is not the most efficient way since the decision boundary is linear. Why? Explain it. When will we need to use the sigmoid function in prediction?
(Ans) The sigmoid function introduces non-linearity to the model. If the decision boundary is linear and the data can be separated using that boundary then there is no point in introducing non-linearity, as it only makes the model more complex.

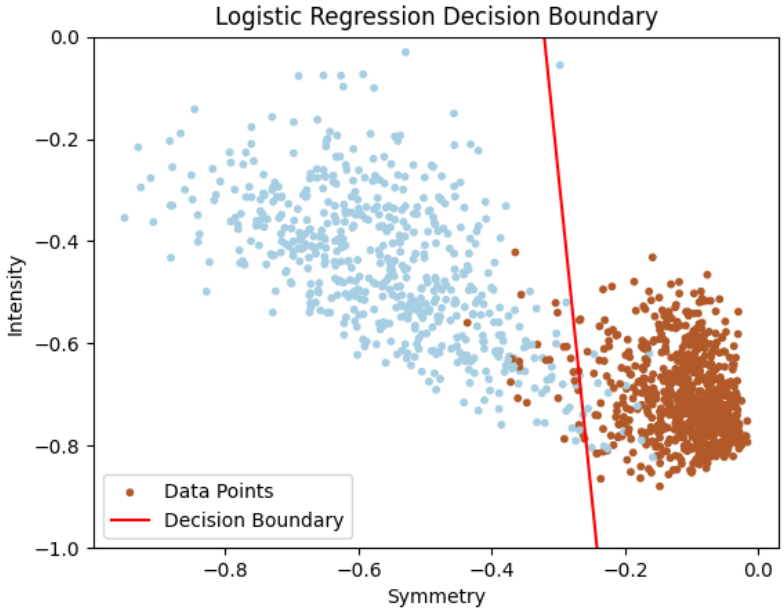
Using the sigmoid function would be useful when the decision boundary is not strictly linear and requires capturing more complex relationships between features and the target values. It can also be used if we are doing binary classification where we need to calculate probabilities of each class, as sigmoid outputs probabilities between 0 and 1.

(d) Is the decision boundary still linear if the prediction threshold is 0.9?
(Ans) The decision boundary in logistic regression is always linear irrespective of the prediction threshold. The decision boundary is determined as $w^T x$, which represents the linear combination of weights and features. For Example $w_0 + w_1 x_1 + w_2 x_2 = 0$. Prediction threshold only alters the point at which the model predicts one class over another based on the predicted probabilities.

(e) In light of your answers to the above two questions, what is the essential property of logistic regression that results in the linear decision boundary?
(Ans) The essential property of logistic regression that results in the linear decision boundary is the use of the sigmoid function to predict the probabilities. As the sigmoid function is monotonically increasing, there always exists a linear representation of weights and features, which results in a linear decision boundary.

3

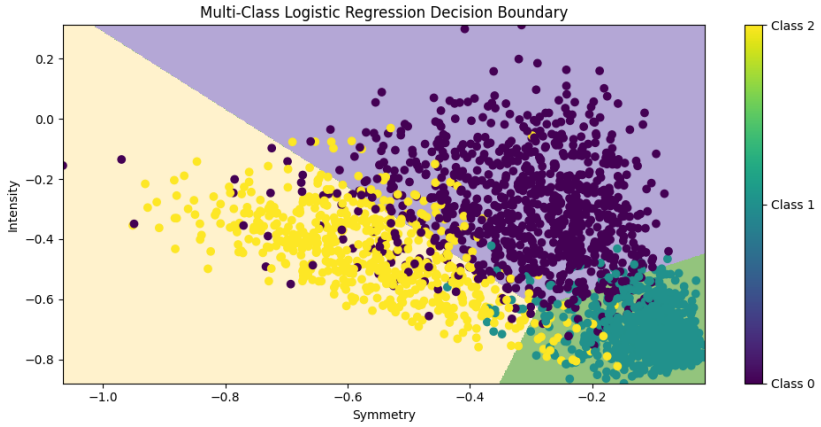
(d)visualize the results after training by using the function visualize results. Include the figure in your submission.



(e) Implement the testing process and report the test accuracy of your best logistic regression model.
(Ans) Testing Accuracy - 93.93939393939394

4

(d)visualize the results after training by using the function visualize results multi. Include the figure in your submission.



(e) Implement the testing process and report the test accuracy of your best logistic regression model.
(Ans) Testing Accuracy - 87.21071863580999

5

(a) Train the softmax logistic classifier and the sigmoid logistic classifier using the same data until convergence. Compare these two classifiers and report your observations and insights
(Ans) After training both models until convergence, these are the training, validation and testing accuracies.
Softmax Training Acc - 0.9718518518518519
Softmax Validation Acc - 0.9788359788359788
Softmax Testing Acc - 0.935064935064935
Sigmoid Training Acc - 0.9718518518518519
Sigmoid Validation Acc - 0.9788359788359788
Sigmoid Testing Acc - 0.935064935064935

Weights of both classifiers after convergence -
Softmax Params - [[-5.1183263 5.1183263] , [-15.27904326 15.27904326], [-1.10255845, 1.10255845]]
Sigmoid Params - [10.35356036 30.58445728 2.14420773]
From the above observations, we can say that both the classifiers are the same for 2 classes. And w1 - w2 = w also satisfies.

(b) Explore the training of these two classifiers and monitor the graidents/weights. How can we set the learning rates so that w1 - w2 = w holds for all training steps?
(Ans) Since the loss function is the same for both the classifiers when no of classes are 2, that is softmax function for k = 2 is the same as the sigmoid function. After observing the gradient values of both these classifiers after the first batch of the first epoch, the w was 1.5 times w1 - w2. Hence if the learning rate of softmax is α_1 and learning rate of sigmoid is α_2 then $\alpha_1 = 1.5 * \alpha_2$.