# Assignment 3 Option 2
# Quantity To Produce Quality

A.D. van der Merwe
Department of Computer Science
University of Stellenbosch
24923273
24923273@sun.ac.za

***Abstract*—**

## I. Introduction

## II. Background

This section presents background information on the gradient descent optimisation algorithm, logistic regression model, and ensemble learning. Additionally, background information on bootstrap aggregating, also known as bagging, basis functions, and performance metrics used in this report.

### A. Gradient Descent

The gradient descent algorithm was first introduced by Augustin-Louis Cauchy in 1847 [1]. Cauchy introduced gradient descent to solve optimisation systems of simultaneous equations through iterative optimisation to find the minimum of a function. Cauchy also introduced the step size parameter, now commonly referred to as the learning rate, to control how large the steps are for each iteration as the algorithm updates model parameters to reach an optimal solution.

The generic learning algorithm of gradient descent is represented by Algorithm 1.

**Algorithm 1** Gradient Descent Learning Algorithm

1: Preprocess the training set $D_T$ as necessary
2: Initialise parameter vector, $\mathbf{w}(t)$, $t = 0$
3: Initialise the learning rate $\eta$
4: **while** stopping condition not satisfied **do**
5:  **for** each $i = 1, ..., n_T$ **do**
6:  Calculate error signal, $\delta(t)$
7:  Calculate a search direction, $\mathbf{q}(t) = f\left(\mathbf{w}(t), \delta(t)\right)$
8:  Update parameter vector: $\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\mathbf{q}(t)$
9:  **end for**
10:  $t = t + 1$
11:  Compute prediction error
12: **end while**
13: Return $\mathbf{w}(t-1)$ as solution

### B. Logistic Regression

The logistic regression model was first introduced by David Cox in 1958 as a method to perform binary classification [2]. Cox specifically designed the logistic regression model to model the probability of a binary outcome as a function of descriptive features.

To construct a logistic regression model that makes use of gradient descent as an optimisation algorithm, a threshold function that is continuous, and therefore differentiable is needed. This function is known as the logistic function and is represented by the mathematical equation below.

$$Logistic(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

where $z$ is a numeric value.

Before the logistic regression model is constructed the binary target features are mapped to 0 or 1. The logistic regression model is then constructed by use of the equation that follows.

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}_i) = \frac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{d}_i}} \tag{2}$$

where $\mathbf{d}_i$ is a vector of the $i$-th descriptive features, with the bias term represented by $\mathbf{d}_0$ and equal to one, $\mathbf{w}$ is a vector of weights, where $\mathbf{w}_0$ represents the weight of the bias term, and the weights that remain corresponds to their respective descriptive features in $\mathbf{d}_i$. The term $\mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)$ represents the predicted output for the $i$-th instance of the logistic regression model. The output of the logistic regression model can be interpreted as probabilities of the occurrence of a target instance that belongs to a specific class. The probability the $i$-th target instance that belongs to class one is given by the equation below.

$$P(y_i = 1|\mathbf{d}_i) = \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i) \tag{3}$$

where $y_i$ is the true label for the $i$-th observation. Similarly, the probability of the $i$-th target instance that belongs to class zero is given by the equation below.

$$P(y_i = 0|\mathbf{d}_i) = 1 - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i) \tag{4}$$

To classify the $i$-th target instance, $\mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)$ is compared to a threshold of $0.5$. The equation used to classify the $i$-th target feature that belongs to either class zero or class one is given as follows.

$$\hat{y}_i = \begin{cases} 0 & \text{if } \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i) < 0.5 \\ 1 & \text{if } \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i) \geq 0.5 \end{cases} \tag{5}$$

where $\hat{y}_i$ is the predicted class of the $i$-th binary target variable.

Gradient descent is used as the optimisation algorithm to find the optimal decision boundary for a logistic regression model. The optimal decision boundary is defined as the set of weights that minimise the sum of squared error (SSE) based on the training set. The mathematical represntation of the SSE is as follows.

$$L_2\left(\mathbb{M}_{\mathbf{w}}, \mathcal{D}\right) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)\right)^2 \tag{6}$$

where $\mathcal{D}$ is the training dataset and $n$ is the number of instances in the training dataset and $L_2$ is the SSE of the training dataset.

The equation used to represent the error signal used in the gradient descent optimisation algorithm to update the weights of the logistic regression model is as follows.

$$\delta(\mathcal{D}, w_j) = \sum_{i=1}^{n}\left((y_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))\mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)(1 - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))d_{j,i}\right) \tag{7}$$

where $w_j$ is the $j$-th weight of the logistic regression model.

The equation used to update the weights of the logistic regression model by use of the gradient descent optimisation algorithm is as follows.

$$w_j = w_j + \eta\sum_{i=1}^{n}\left((y_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))\mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)(1 - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))d_{j,i}\right) \tag{8}$$

where $\eta$ is the learning rate.

### C. Basis Functions

### D. Ensemble Learning

### E. Bootstrap Aggregating

### F. Performance Metrics

Performance metrics are essential tools when the effectiveness of classification models are evaluated. Performance metrics provide a quantitative measure of how reliable and accurate a prediction model performs classification on a dataset. Key metrics include accuracy, precision, recall, and F1-score, each offering unique insights into different aspects of model performance [**?**].

*a) Accuracy:* Accuracy is a common method used to evaluate the performance of classification models. The accuracy of a predictive classification model is determined by the proportion of correctly predicted labels against the total number of predictions. The calculation of the accuracy of a predictive model is as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{9}$$

Accuracy is a popular choice of performance measure mainly beacause it is fairly easy to understand and compute. Accuracy generally perform well on well balanced datasets. On imbalanced datasets, accuracy can produce values that are misleading. If a model predicts

*b) Precision and Recall:* Precision is the proportion of () predictions against all of the and (). The equation to calculate the precision of a classification model is as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{10}$$

Recall is the proportion of predictions against all of the and (). The equation to calculate the recall of a classification model is as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{11}$$

*c) F1-score:* The F1-score, also known as the Dice similarity coefficient, is the harmonic mean of precision and recall, that provides a balance between the precision and recall [**?**]. In multiclass classificationprecision and recall are adapted from binary classification to handle multiple classes. Instead of focusing on a single positive and negative class, these metrics are calculated for each class, where each class is treated as the positive class while considering all others as negative. The F1-score is calculated for each clas and then averaged. There are three types of averaging F1-score metrics.

The first F1-score metric is the macro averaging F1-score, where the F1-score is calculated independently for each label and then these scores are averaged, giving each class the same weight. The equation used to calculate the macro F1-score is as follow:

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{12}$$

where $C$ is the number of classes.

## III. Implementation

## IV. Empirical Procedure

### A. Performance Metrics

### B. Data Preprocessing

### C. Experimental Setup

### D. Control Parameters

TABLE I: Logistic Regression Control Parameters

| Dataset | Control Parameters | | |
|---|---|---|---|
| | *eta* | *epochs* | *patience* |
| ***Breast Cancer*** | 0.00239 | 10678 | 6 |
| ***Diabetes Dataset*** | 0.01121 | 18690 | 9 |
| ***Banana Quality*** | 0.00023 | 4338 | 5 |
| ***Water Quality*** | 0.00601 | 25484 | 9 |
| ***Spiral Dataset*** | 0.06708 | 17335 | 7 |

### E. Statistical Significance and Analysis

## V. Research Results

## VI. Conclusion

## References

[1] J. Braet, M. Cristina, Hinojosa-Lee, and J. Springael. "Evaluating performance metrics in emotion lexicon distillation: a focus on F1 scores." (2024).

[2] A. Cauchy ”Méthode générale pour la résolution des systemes d’équations simultanées.” In: Comp. Rend. Sci. Paris (1847).
[3] D. R. Cox ”The regression analysis of binary sequences.” In: Journal of the Royal Statistical Society Series B: Statistical Methodology (1958).
[4] B. J. Erickson, and K. Felipe ”Magician’s corner: 9. Performance metrics for machine learning models.” In: Radiology: Artificial Intelligence 3, no. 3 (2021): e200126.