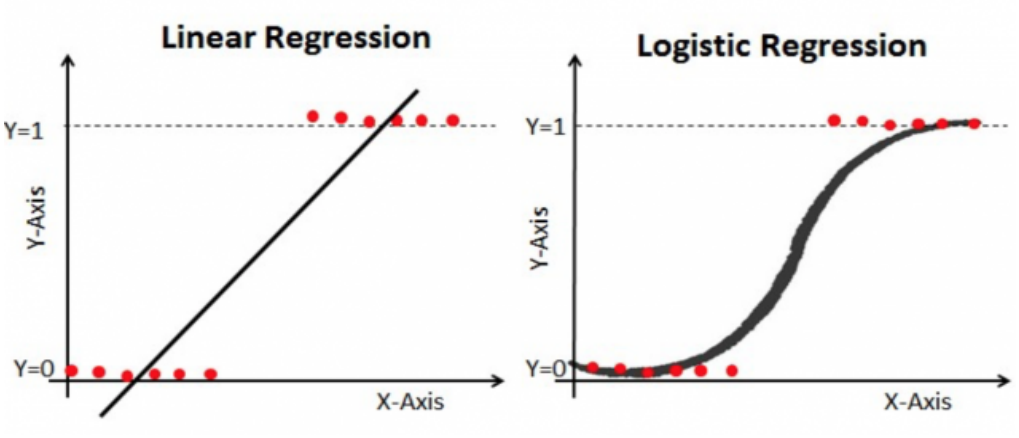
Spring 2021 Capstone
Mehrab Hafiz
Final Project Report

1. Logistic Regression

Logistic regression is the go-to method for binary classification problems. Similar to other regression methods, it is used to model the relationship between the response variable and a set of independent variables. Unlike linear regression where a continuous response value is predicted, logistic regression is used to compute the probabilities of outcomes for a dichotomous response variable. An example is predicting whether a customer purchased a product given a set of predictors such as gender, age and salary.



As we can see from the demonstration above, both linear and logistic regression can be used for predicting binary outcomes. A line is fitted in linear regression while a sigmoid function is used for logistic regression. However, the sigmoid function is much better at fitting the dataset.

2. Problem

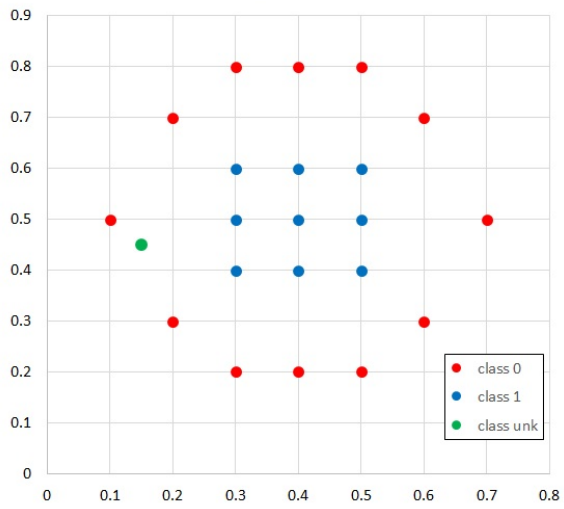
One weakness of logistic regression is that it is a generalized linear model. It might not appear that way at first sight due to the sigmoid function. However, the sigmoid function is used as an activation function. The actual z value is calculated by taking the dot product of parameters and their corresponding weights:

$$z = w_0 x_0 + w_1 x_1 + \ldots + w_m x_m = \sum_{j=1}^{m} w_j x_j = \mathbf{w}^T \mathbf{x}.$$

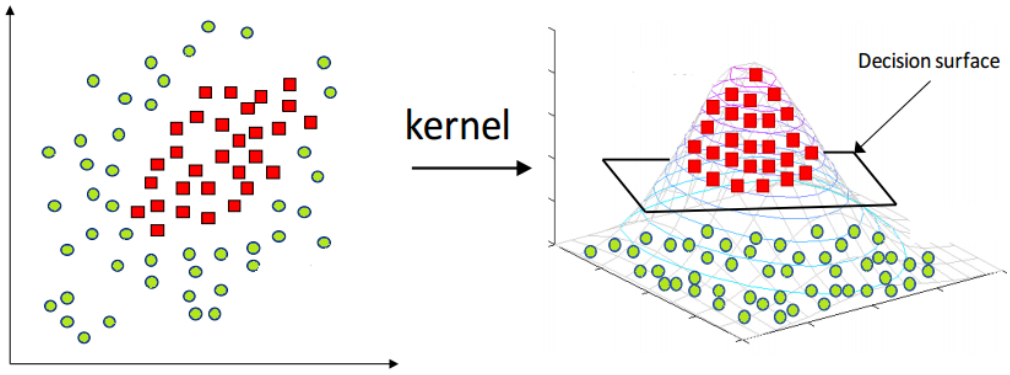
The outcome depends only on the sum of weighted parameters. Essentially, Logistic Regression constructs a linear boundary for classifying the dataset.

3. Kernel Logistic Regression

Logistic regression can produce great accuracy on linearly separable datasets. A dataset is linearly separable where the two classes can be separated by drawing a straight line. Some data is far from being linearly separable. Let's look at an example:



In this case, no straight line can be drawn to distinguish class 0 from class 1. Logistic regression would give around 60% accuracy which is not much better than random guesses. This is where kernels come into play. The dataset can be transformed into a 3rd dimension using a polynomial kernel:



A decision boundary can now be used to separate the classes.

4. The Dataset

Breast Cancer Wisconsin (Diagnostic) Data Set for non-linear models. Features are generated based on images of a fine needle aspirate (FNA) of a breast mass. The dataset can be found here.

Columns used:

- Mean radius
- Mean texture
- Mean symmetry
- Diagnosis (Benign or Malignant)

5. The Kernel Function

The kernel function accepts the feature matrix as input and computes kernel value for each of the rows in the matrix. For a given row, the kernel value k is calculated as:

$$k = \alpha_1 x_1^{p1} + \alpha_2 x_2^{p2} + \alpha_3 x_3^{p3} + \beta$$

A trained model will have alpha and beta values. In our case, there will be 3 alpha values and one bias value. The alpha values correspond to the three features we will be using to train the model. The beta value is a constant that is added to the dot product.

6. The Activation Function

Now we can apply a sigmoid function on the z value to obtain a probability.

$$\hat{y} = \frac{1}{1+e^{-k}}$$

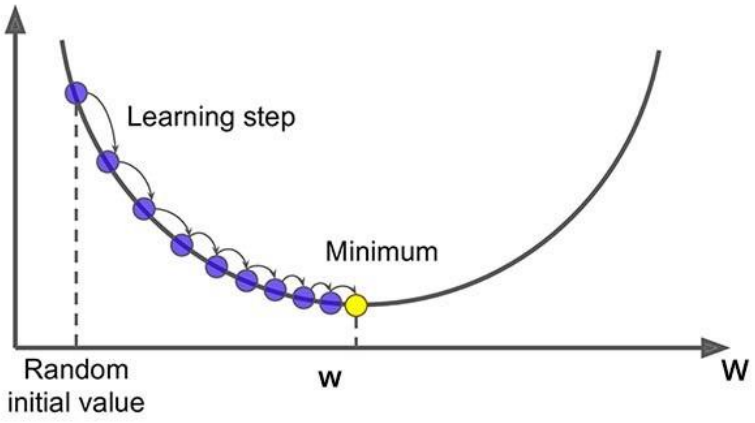
The resulting probability value will be between 0 and 1. We can set a decision boundary at .5. If the probability is less than .5, we predict 0. Otherwise we predict 1. This decision boundary can be tuned further through hyperparameter testing.

7. Training the Model

Training a logistic regression model comes down to finding proper alpha and beta values such that the loss function is minimized. For our model, we are using a cross entropy loss function. For a predicted value ŷ and an actual value y, the loss function is calculated as follows:

$$L(y, \hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)]$$

The advantage of using cross entropy loss is that it heavily penalizes predicted values that deviate from the true value. We use a simplified gradient descent to minimize this loss function. The weights are adjusted based on the derivatives of the loss function. The derivatives tell us the direction and the amount of weight adjustment that is needed to make the loss smaller.



8. Derivative of Cross Entropy Loss

We start with $-\ y\ *\ log(\hat{y})$ and $(1\ -\ y)\ *\ log(1\ -\ \hat{y})$
Then the derivative for loss is $-\ y\ *\ 1/\hat{y}\ *\ \hat{y}'$ and $-\ (1\ -\ y)\ *\ 1\ /(1\ -\ \hat{y})\ *\ \hat{y}'$
Combining both gives $(\hat{y}\ -\ y)\ /\ (\hat{y}\ -\ \hat{y}^2)\ *\ \hat{y}'$

Per chain rule, we have to find the derivative of ŷ. Now ŷ is based on a sigmoid function therefore its derivative is basically the derivative of a sigmoid function: $\hat{y}\ *\ (1\ -\ \hat{y})$ which simplifies to $(\hat{y}\ -\ \hat{y}^2)\ *\ z'$. Denominator is cancelled out leaving just $(\hat{y}\ -\ y)\ *\ z'$.

Similarly, we have to find the derivative of z that is contained in ŷ. If
$z\ =\ \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 +\ \beta$, then the derivative is just $z'\ =\ x_1 + x_2 + x_3 =\ x$.

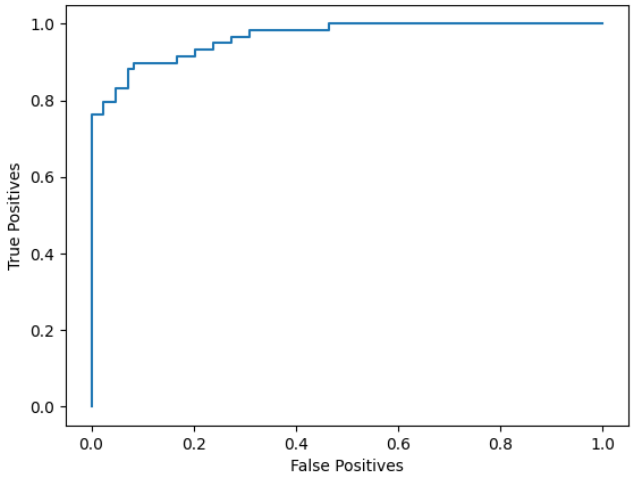
Thus, the final derivative is $(\hat{y} - y) * (x_1 + x_2 + k)$. We can now use these to update the weights for alpha and k values. For beta, we can just use $(\hat{y} - y)$.

9. Results & Parameter Tuning

In addition to gradient descent, we also need a learning rate factor. It is a hyperparameter with a small positive value that dictates how much the weights should change during each iteration. A smaller learning rate might overfit the training data while a large one will miss local minimums. 5 models were trained for 10000 iterations with the following rates:

Rate	Accuracy
.1	.80
.05	.90
.03	.85
.01	.83
.005	.75

As discussed earlier, logistic regression is not computing outcomes, but rather probabilities associated with those outcomes. A decision boundary needs to be set in order to convert those probabilities into classes. We can set the boundary at .5 but this doesn't always give the most accurate model. More often than not datasets are skewed towards certain outcomes. An ROC curve can be used to find the most optimal cutoff. The curve is generated by plotting the True Positive Rate (TPR) against False Positive Rate (FPR):



It appears that almost all the malignant tumors are correctly detected at p = .5. No adjustments are necessary. In a real world scenario it might be beneficial to sacrifice accuracy in exchange for increased cancer detection rates.