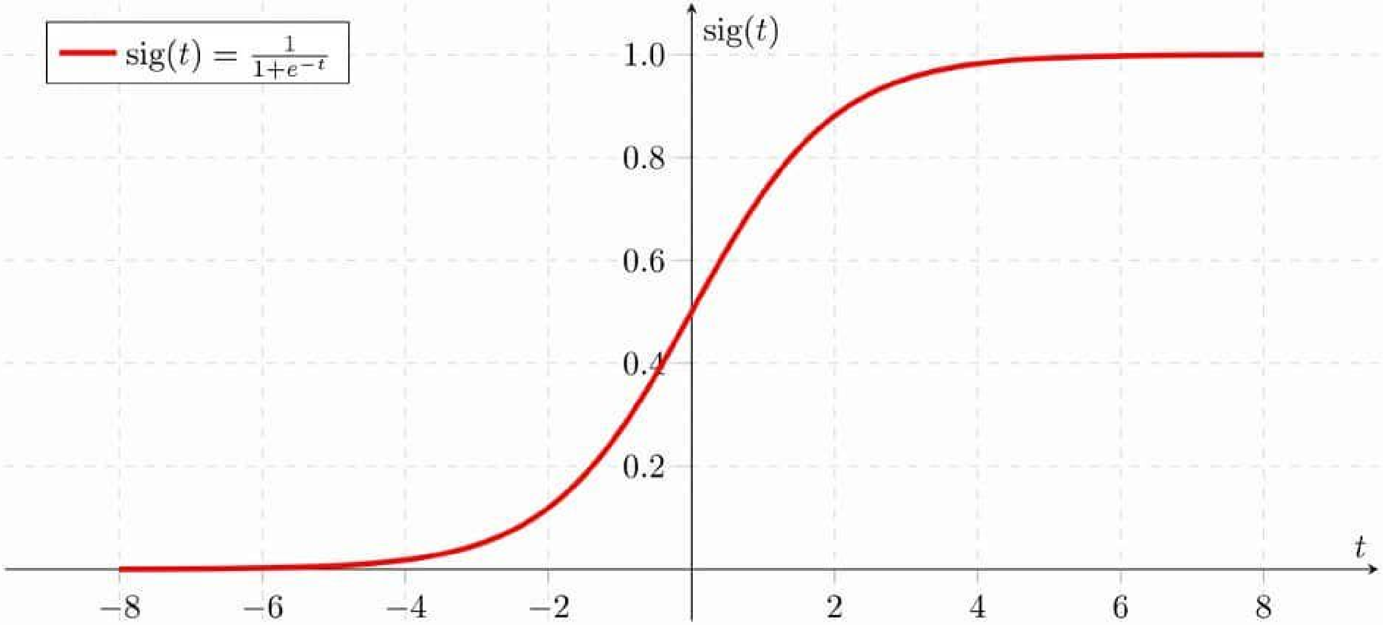
Logistic Regression

- By name, everyone thinks this is a regression algorithm but not.
- Logistic Regression is used when the dependent variable(target) is categorical.
- logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.
- sigmoid function = 1/(1+e(-x))
- It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.



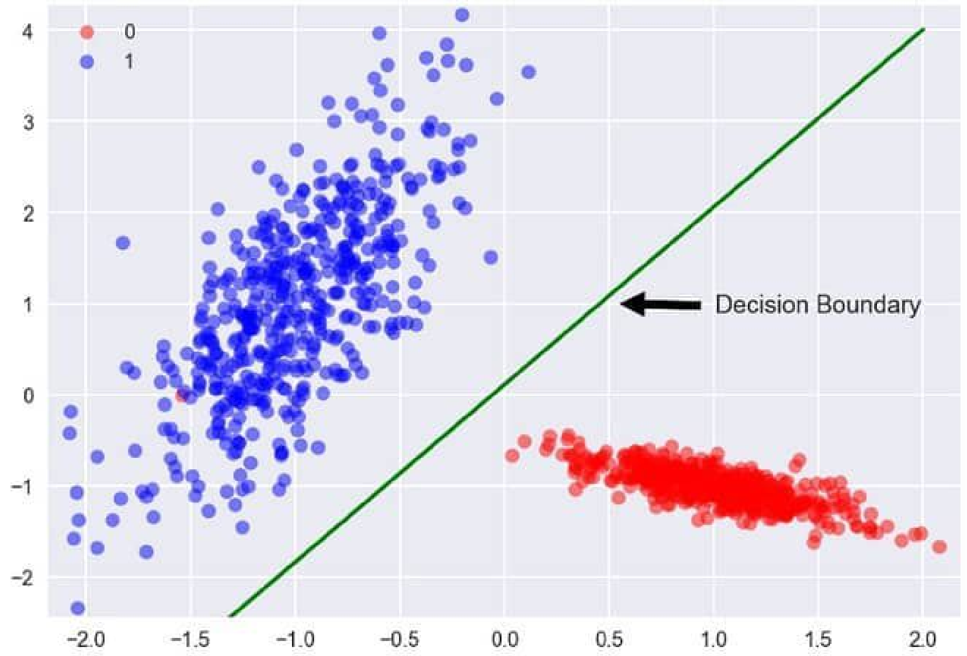
@learn.machinelearning

Logistic Regression

- You can keep your own threshold value as a separator for predicted classes.
- So here the sigmoid returns a number between 0 and 1.
- For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction was .2 we would classify the observation as negative.
- For logistic regression with multiple classes, we could select the class with the highest predicted probability.
- If we have multiple classes more than 2 then we build multiple logistic regression models with 2 classes and pics the highest avg probability as a class.
- This is just an outline of how an algorithm works. from the next post, I will teach in detail with Math.

Logistic Regression

- Let's see why we are using a sigmoid function instead of just how we are doing linear regression.



- The algorithm assumes that data should be almost separable.
- So we will draw a plan to separate the 2 classes like blue points and red points as shown in the above figure. And the green line is the decision boundary.
- The equation can be written as W*T x X + b. W is the weight vector or coefficient vector, X is the input values and b is the intersect.

@learn.machinelearning

Logistic Regression

- So how do we find the line which separates the data?
- We find the distance between the point and the plane and if we get W^T x X > 0 then it is correctly classified or else incorrectly classified.
- For example, we have seen data in the last figure where blue points are positive and red points are negative and of course, we will have some misclassified points in the training dataset.
- So here we check for the accuracy by sum(yi x W^T x Xi > 0) for all points. if the point is correctly classified by the line we get yi x W^T x Xi > 0 as positive or else we get as yi x W^T x Xi < 0. yi is the original label.
- let's check a few cases if the point is positive and it is above the plain. so the distance between the point and plan will be positive and the label is positive so we get the result as > 0. if the point is negative and the point lies below the plane and we get distance as a negative and original label is negative and we get result as > 0

Logistic Regression

- If the point is positive and lies below the line so we get the result as < 0 which is misclassified and if the point is negative and lies above the line we get the result as < 0 which is also misclassified.

  So here we want to maximize the output. if every point is correctly classified we get highest score sum(yi x W^T x Xi).

  We find the line using gradient descent as how we find the line in linear regression.

  So the problem with this method is when we have extreme outliers. which will change the shape of the line because we can get high value as output because of one extreme point and we think this the perfect line but intuitively we misclassified most of the points.

  To solve this problem we introduce a concept called sigmoid and a loss function to calculate the loss of a model.

Logistic Regression

- Why sigmoid?
- To get rid of outliers in logistic regression we are using a sigmoid function which helps to bring all the values to lie between 0 and 1.
- sigmoid function = 1/(1 + e^(-x))
- We choose sigmoid because it has a nice probabilistic interpretation, normalizing the values.
- so the formula becomes maximizing the sum(1/(1+ e^(-yi*WAT*xi)))
- Maximizing any function will be hectic so we try to minimize the problem. We apply log to make it easy so when we apply log formula becomes maximizing of sum(log(1/(1+ e^(-yi*WAT*xi)))).
- log(1/x) = - log(x)
- = maximizing of sum(-log(1+ e^(-yi*WAT*xi))).
- And in geometry maximising f(x) = minimizing -f(x)

Logistic Regression

- This is because of let's take a function of X^2 where it is a monotonic function. the minimising factor in this function will be the same as maximising the (-1)*X^2.
- so our final formula will be.

  minimising sum(log(1+e^(-yi*w^T*Xi)))

- We use gradient descent to find the best weight vector or coefficient vector.
- This method will also face the same error which is faced by linear regression are overfitting or underfitting problems.
- We can apply the L1 or L2 regularization to reduce the overfitting.
- Here L1 regularizers give another advantage is we can find the best features which are important for modelling.
- This is because L1 will make some weights to zero and which weight has the highest value will be the best value.

Logistic Regression

- Lets Solve LR using the Logistic function.
- The logistic function also called the sigmoid function.
- $y = 1 / (1 + e^{-Z})$, $Z = b_0 + b_1 * x_1 + \ldots\ldots + b_n * x_n$.
- Here y is the predicted output of an input features.
- Logistic regression models the probability of the default class eg first class.
- Here we will keep a threshold value to get the class label using the probability we got from sigmoid function.
- For example, if we have 0.5 as a threshold value and we got 0.6 as P as output from a sigmoid function then we can define this as a first class or second class based on the problem and you can see as the probability goes towards 1 we can say it is more confident in predicting the class label as first class or second class.
- To get the probability of another class we do 1-P.

Logistic Regression

- And how can we check whether our model is performing well or not?
- we use a cost function called Cross-Entropy, also known as Log Loss.
- log loss = - (1/n) sum(yi * log(pi) + (1-yi) * log(1-pi)).
- yi is the original class label and pi is the probability value.
- Multiplying by y and (1−y) in the above equation is a sneaky trick that lets us use the same equation to solve for both y=1 and y=0 cases. If y=0, the first side cancels out. If y=1, the second side cancels out. In both cases we only perform the operation we need to perform.
- As you can see if the probability values go high towards 1 for one class or low towards 0 for other class then we get the low values when you apply log to it so if the probabilities are high, we get minimum log loss value.

Logistic Regression

Multicollinearity in LR

- Multicollinearity is a statistical phenomenon in which predictor variables in a logistic regression model are highly correlated.
- This is the most important problem in LR as if the features are collinear then the weight vector will change arbitrarily where we can use that to define best features.
- To check whether the features are collinear or not we can use the perturbation technique(There are many to do but we go with this).Steps to do this test.
- Add very small value to the complete dataset and find the weight vector before adding value and after adding a small value.
- Find the difference between weight vectors if it changes a lot then there is collinearity or else there is no collinearity.

Logistic Regression

Important points

- Every algorithm assumes something about the data LR Assumes the data is linearly separable or almost linearly separable.
- We can find feature importance in this algorithm using weight vectors.
- If we are using sigmoid the algorithm is less prone to the outliers. we can also remove outliers using the iteration process like find the weight vector and find the distance between all the points and weight vector, remove the points which are far and repeat the process until you think there are no more outliers.
- We can't use this algorithm for multiclass but what we can do is use one vs Rest method.

Logistic Regression

Prepare Data for Logistic Regression

- **Binary Output Variable:** This might be obvious as we have already mentioned it, but logistic regression is intended for binary (two-class) classification problems. It will predict the probability of an instance belonging to the default class, which can be snapped into a 0 or 1 classification.
- **Remove Noise:** Logistic regression assumes no error in the output variable (y), consider removing outliers and possibly misclassified instances from your training data.
- **Remove Correlated Inputs:** Like linear regression, the model can overfit if you have multiple highly-correlated inputs. Consider calculating the pairwise correlations between all inputs and removing highly correlated inputs.

Logistic Regression

Prepare Data for Logistic Regression

- **Gaussian Distribution:** Logistic regression is a linear algorithm (with a non-linear transform on output). It does assume a linear relationship between the input variables with the output. Data transform of your input variables that better expose this linear relationship can result in a more accurate model. For example, you can use log, root, Box-Cox and other univariate transforms to better expose this relationship.
- **Fail to Converge:** It is possible for the expected likelihood estimation process that learns the coefficients to fail to converge. This can happen if there are many highly correlated inputs in your data or the data is very sparse (e.g. lots of zeros in your input data).