Statistics Foundation

Machine Learning – Logistic Regression

Digital Vidya

Classification Problem

- **What is a Classification Problem ?**
  - We identify problem as classification problem when independent variables are continuous in nature and dependent variable is in categorical form i.e. in classes like positive class and negative class

  - The real life example of classification example would be
    - to categorize the mail as spam or not spam
    - to categorize the tumor as malignant or benign
    - to categorize the transaction as fraudulent or genuine

	Two Class Classification	
$y \in \{0, 1\}$	**1 or Positive Class**	**0 or Negative Class**
Email	Spam	Not Spam
Tumor	Malignant	Benign
Transaction	Fraudulent	Not Fraudulent

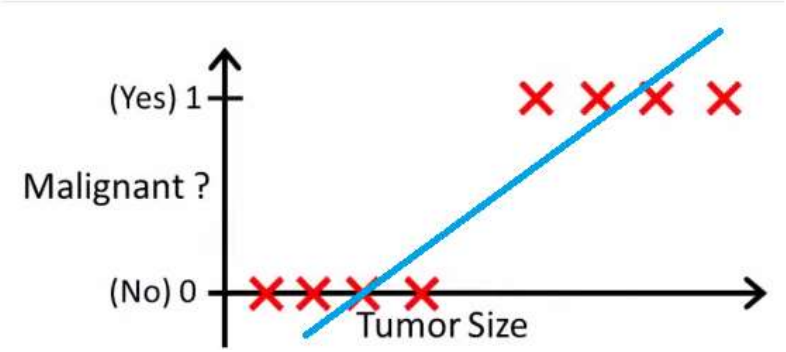
  - All these problem's answers are in categorical form i.e. Yes or No and that is why they are two class classification problems

  - Although, sometime we come across more than 2 classes and still it is a classification problem. These types of problems are known as multi class classification problems.

Digital Vidya

Classification Problem

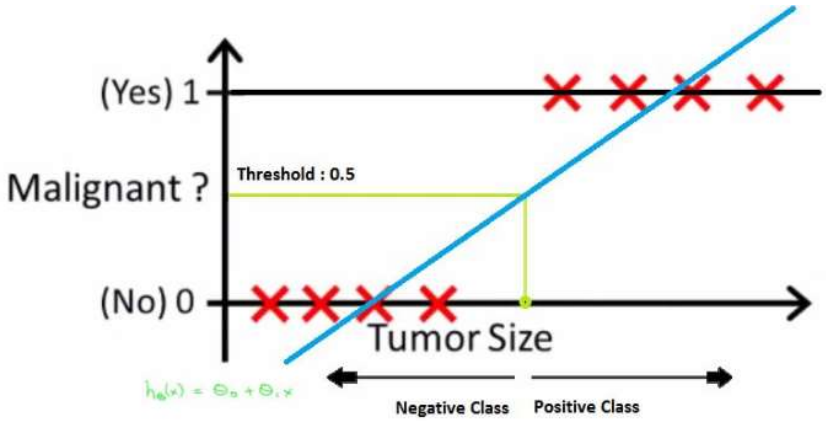
- **Why not use Linear Regression ?**
  - Suppose we have a data of tumour size vs its malignancy.
  - As it is a classification problem, if we plot, we can see, all the values will lie on 0 and 1.
  - And if we fit best found regression line, by assuming the threshold at 0.5, we can do line pretty reasonable job.



Digital Vidya

Classification Problem

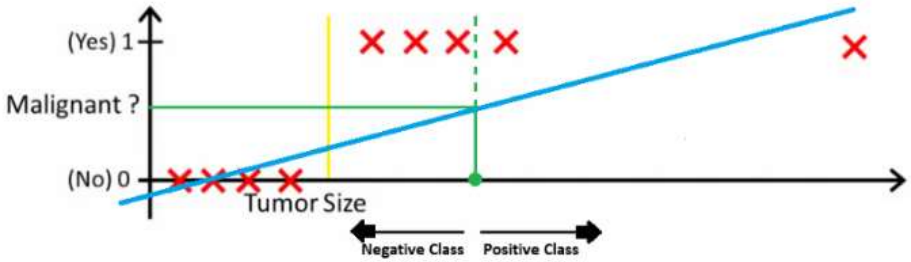
- **Why not use Linear Regression ?**
  - We can decide the point on the x axis from where:
  - All the values lie to its left side are considered as negative class
  - All the values lie to its right side are positive class



Digital Vidya

Classification Problem

- **Why not use Linear Regression ?**
  - What if there is an outlier in the data
  - Things would get pretty messy
  - For example, for 0.5 threshold

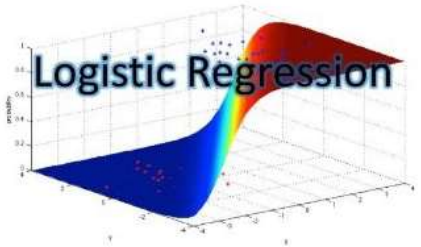


  - If we fit best found regression line, it still won't be enough to decide any point by which we can differentiate classes
  - It will put some positive class examples into negative class
  - The green dotted line (Decision Boundary) is dividing malignant tumours from benign tumours but the line should have been at a yellow line which is clearly dividing the positive and negative examples
  - So just a single outlier is disturbing the whole linear regression predictions
  - And that is where logistic regression comes into a picture

Digital Vidya

Logistic Regression

- **What is Logistic Regression**
  - Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable
  - In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.)
  - In other words, the logistic regression model predicts P(Y=1) as a function of X
  - It is named as 'Logistic Regression', because it's underlying technique is quite the same as Linear Regression. The term "Logistic" is taken from the Logit function that is used in this method of classification

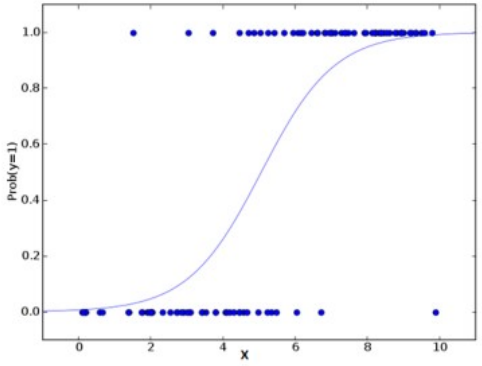


Digital Vidya

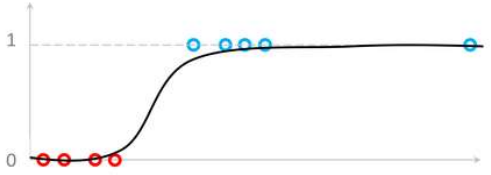
Logistic Regression

- **Logistic Regression**
  - Logistic Regression uses Sigmoid function
  - The logistic function is a Sigmoid function, which takes any real value between 0 and 1

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$



  - Let's consider t as linear function in a univariate regression model $t = \beta_0 + \beta_1 x$
  - So the Logistic Equation will become $p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$
  - Now, when logistic regression model come across an outlier, it will take care of it



Digital Vidya

Logistic Regression

- **Logistic Regression Equation**
  - The underlying algorithm of Maximum Likelihood Estimation (MLE) determines the regression coefficient for the model that accurately predicts the probability of the binary dependent variable
  - The algorithm stops when the convergence criterion is met or maximum number of iterations are reached
  - Since the probability of any event lies between 0 and 1 (or 0% to 100%), when we plot the probability of dependent variable by independent factors, it will demonstrate an 'S' shape curve.

$$\text{Logit} = \text{Log}\ (p/1\text{-}p) = \log\ (\text{probability of event happening}/\ \text{probability of event not happening}) = \log\ (\text{Odds})$$

Digital Vidya

Logistic Regression

- **Logistic Regression Example**
  - We are provided a sample of 1000 customers. We need to predict the probability whether a customer will buy (y) a particular magazine or not. As we've a categorical outcome variable, we'll use logistic regression
  - To start with logistic regression, first write the simple linear regression equation with dependent variable enclosed in a link function:

$$g(y) = \beta o + \beta(Age)$$

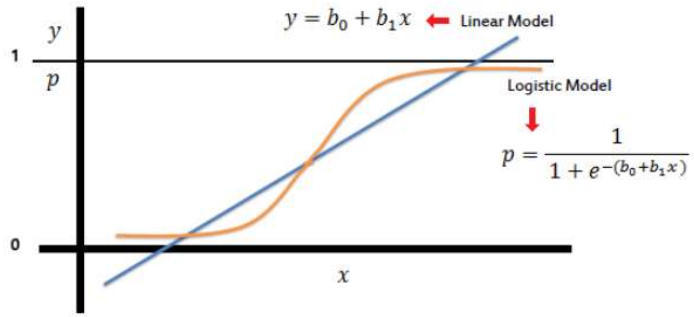
  - For understanding, consider 'Age' as independent variable
  - In logistic regression, we are only concerned about the probability of outcome dependent variable (success or failure).
  - g() is the link function. This function is established using two things:
  - Probability of Success(p) and Probability of Failure(1-p)
  - p should meet following criteria:
    - It must always be positive (since p >= 0)
    - It must always be less than equals to 1 (since p <= 1)
  - Final equation for Logistic Regression:

$$\log(p/(1\text{-}p)) = \beta o + \beta(Age)$$

Digital Vidya

Logistic Regression

- **Logistic Regression key facts**
  - (p/1-p) is the odd ratio
  - Whenever the log of odd ratio is found to be positive, the probability of success is always more than 50%
  - A typical logistic model plot is shown below
  - It shows probability never goes below 0 and above 1
  - Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the "odds" of the target variable, rather than the probability
  - Moreover, the predictors do not have to be normally distributed or have equal variance in each group
  - Logistic regression can handle any number of numerical and/or categorical variables

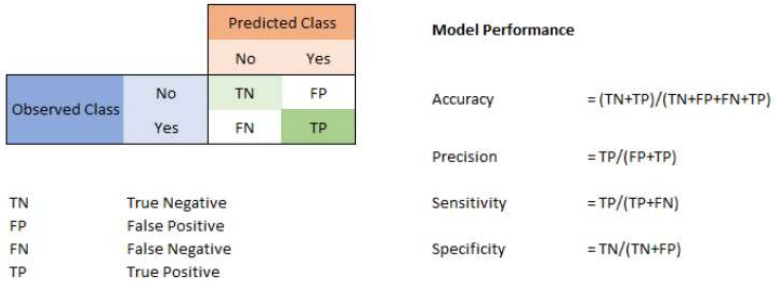


$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

Digital Vidya

Logistic Regression

- **Logistic Regression Evaluation**
  - **AIC** (Akaike Information Criteria) – The analogous metric of adjusted R^2 in logistic regression is AIC. AIC is the measure of fit which penalizes model for the number of model coefficients. Therefore, we always prefer model with minimum AIC value
  - **Null Deviance and Residual Deviance** – Null Deviance indicates the response predicted by a model with nothing but an intercept. Lower the value, better the model. Residual deviance indicates the response predicted by a model on adding independent variables. Lower the value, better the model
  - **Confusion Matrix**: This is a tabular representation of Actual vs Predicted values. This helps us to find the accuracy of the model and avoid over-fitting

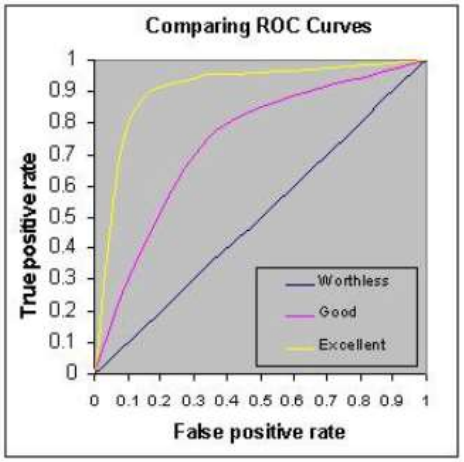


Confusion Matrix

Digital Vidya

Logistic Regression

- ## Logistic Regression Evaluation
  - **ROC Curve:** Receiver Operating Characteristic (ROC) summarizes the model's performance by evaluating the trade-offs between true positive rate (sensitivity) and false positive rate (1- specificity)
    - For plotting ROC, it is advisable to assume p > 0.5 since we are more concerned about success rate
    - ROC summarizes the predictive power for all possible values of p > 0.5
    - The area under curve (AUC), referred to as index of accuracy (A) or concordance index, is a perfect performance metric for ROC curve
    - Higher the area under curve, better the prediction power of the model
    - The ROC of a perfect predictive model has TP equals 1 and FP equals 0. This curve will touch the top left corner of the graph



Digital Vidya

Logistic Regression

- **Probability, Odds and Log of Odds**
  - Let's say that the probability of success is p=0.8, then the probability of failure is 1−p=0.2
  - The odds of success is p1−p=0.81−0.8=4, i.e. the odds of success is 4 to 1 and the odds of failure is 0.25 to 1
  - Note that:
    - Probability ranges from 0 to 1
    - Odds range from 0 to ∞
    - Log Odds range from −∞ to ∞

Digital Vidya

Logistic Regression

- **Interpretation of Logistic Output**
  - Consider sample dataset ‘honordata’

```
##   female read write math hon femalexmath
## 1      0   57    52   41   0           0
## 2      1   68    59   53   0          53
## 3      0   44    33   54   0           0
## 4      0   63    44   47   0           0
## 5      0   47    52   57   0           0
## 6      0   44    52   51   0           0
```

Digital Vidya

Logistic Regression

- **Interpretation of Logistic Output**

  Step1:
  - Logistic Regression with No Predictor Variables
  - Target variable is 'hon' (as shown sample data)

```
##   female read write math hon femalexmath
## 1      0   57    52   41   0           0
## 2      1   68    59   53   0          53
## 3      0   44    33   54   0           0
## 4      0   63    44   47   0           0
## 5      0   47    52   57   0           0
## 6      0   44    52   51   0           0
```

$$logit(p) = \beta_0$$

```
##             Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -1.12546    0.16441 -6.845446 7.623779e-12
```

The intercept= -1.12546 which corresponds to the log odds of the probability of being in an honor class p

We can go from the log odds to the odds by exponentiating the coefficient which gives us the odds O=0.3245

We can go backwards to the probability by calculating p=O/(1+O) = 0.245

Logistic Regression

- **Interpretation of Logistic Output**

Step2:

- Logistic Regression with a Single Dichotomous Predictor Variable
- Target variable is 'hon' (as shown sample data)
- Here we will use a binary predictor variable female in our model:

$$logit(p) = \beta_0 + \beta_1 * female$$

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -1.4708517  0.2689554 -5.468756 4.532047e-08
## female       0.5927822  0.3414293  1.736178 8.253231e-02
```

```
##   female read write math hon femalexmath
## 1      0   57    52   41   0           0
## 2      1   68    59   53   0          53
## 3      0   44    33   54   0           0
## 4      0   63    44   47   0           0
## 5      0   47    52   57   0           0
## 6      0   44    52   51   0           0
```

The intercept= -1.47085 which corresponds to the log odds for males being in an honor class
(since male is the reference group, female=0)
The coefficient for female= 0.59278 which corresponds to the log of odds ratio between the female group and male group
The odds ratio equals 1.81 which means the odds for females are about 81% higher than the odds for males

Logistic Regression

- **Interpretation of Logistic Output**

Step3:

- Logistic Regression with a Single Continuous Predictor Variable
- Target variable is 'hon' (as shown sample data)
- Here we will use a single continuous predictor variable math in our mode:

$$logit(p) = \beta_0 + \beta_1 * math$$

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -9.7939421 1.48174484 -6.609736 3.850061e-11
## math         0.1563404 0.02560948  6.104784 1.029399e-09
```

```
##   female read write math hon femalexmath
## 1      0   57    52   41   0           0
## 2      1   68    59   53   0          53
## 3      0   44    33   54   0           0
## 4      0   63    44   47   0           0
## 5      0   47    52   57   0           0
## 6      0   44    52   51   0           0
```

The intercept= -9.79394 which is interpreted as the log odds of a student with a math score of zero being in an honors class

The coefficient for math= 0.15634 which is interpreted as the expected change in log odds for a one-unit increase in math score
The odds ratio can be calculated by exponentiating this value to get 1.16922 which means we expect to see about 17% increase in the odds of being in an honors class, for a one-unit increase in math score

$$logit(p) = \frac{p}{1-p} = -9.79394 + 0.15634 * math$$

Digital Vidya

Logistic Regression

- **Python Implementation**

```
import pandas as pd
import numpy as np
Diabetes=pd.read_csv('diabetes.csv')
table1=np.mean(Diabetes,axis=0)
table2=np.std(Diabetes,axis=0)

#####The data are unbalanced with 35% of observations having diabetes.
#####The standard deviation of the different variables is also very
different, to compare the coefficient of the different variables the
coefficient will need to be standardized


inputData=Diabetes.iloc[:,:8]
outputData=Diabetes.iloc[:,8]
```

Ref: https://github.com/AntoineGuillot2/Logistic-Regression-Python/blob/master/LogisticReg.py

Logistic Regression

- **Python Implementation**

```
from sklearn.linear_model import LogisticRegression
logit1=LogisticRegression()
logit1.fit(inputData,outputData)


logit1.score(inputData,outputData)
```

Even if the logistic regression is a simple model around 78% of the observation are correctly classified!

####Due to class imbalance, we need to check the model performance on each class. Not being able to classify people with diabetes would be a major problem since this is the goal of the model.

Ref: https://github.com/AntoineGuillot2/Logistic-Regression-Python/blob/master/LogisticReg.py

Logistic Regression

- **Python Implementation**

```
##True positive
trueInput=Diabetes.ix[Diabetes['Outcome']==1].iloc[:,:8]
trueOutput=Diabetes.ix[Diabetes['Outcome']==1].iloc[:,8]
##True positive rate
np.mean(logit1.predict(trueInput)==trueOutput)
##Return around 55%

##True negative
falseInput=Diabetes.ix[Diabetes['Outcome']==0].iloc[:,:8]
falseOutput=Diabetes.ix[Diabetes['Outcome']==0].iloc[:,8]
##True negative rate
np.mean(logit1.predict(falseInput)==falseOutput)
##Return around 90%
```

Ref: https://github.com/AntoineGuillot2/Logistic-Regression-Python/blob/master/LogisticReg.py

Logistic Regression

- **Python Implementation**

```
###Confusion matrix with sklearn
from sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score
confusion_matrix(logit1.predict(inputData),outputData)
```

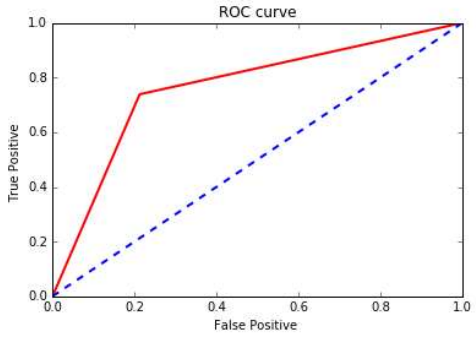
Ref: https://github.com/AntoineGuillot2/Logistic-Regression-Python/blob/master/LogisticReg.py

Logistic Regression

- **Python Implementation**

```
##Computing false and true positive rates
fpr, tpr,_=roc_curve(logit1.predict(inputData),outputData,drop_intermediate=False)

import matplotlib.pyplot as plt
plt.figure()
##Adding the ROC
plt.plot(fpr, tpr, color='red', lw=2, label='ROC curve')
##Random FPR and TPR
plt.plot([0, 1], [0, 1], color='blue', lw=2, linestyle='--')
##Title and label
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC curve')
plt.show()
roc_auc_score(logit1.predict(inputData),outputData)
```



The ROC curve of the model

Ref: https://github.com/AntoineGuillot2/Logistic-Regression-Python/blob/master/LogisticReg.py

Digital Vidya

Thank You

Digital Vidya