Assignment-2: A mathematical essay on Logistic Regression

Vasudev Gupta
Interdisciplinary Dual Degree - Data Science
Indian Institute of Technology Madras
Chennai, India
me18b182@smail.iitm.ac.in

***Abstract*—We will present a mathematical research on logistic regression models in this paper. We will use the logistic regression model to estimate the survival rate and will attempt to determine whether using the logistic regression model was the correct decision. We will also attempt exploratory data analysis and feature selection in order to give the correct data to the model. Finally, we'll check if our model can generalise effectively to unseen data.**

I. Introduction

Machine learning techniques are becoming increasingly popular, and they are assisting a number of organisations in tackling difficult challenges using data-driven approaches. Machine learning methods attempt to estimate the function on provided training data by capturing the underlying distribution of the training data and then predict on unseen data using the learned distribution. When building the model, one expects that the training data is comparable to previously unseen data and that the model will generalise well to future data.

The logistic regression model will be the focus of this essay. Logistic regression works on the assumption that the conditional distribution of the predicted variable with regard to the input data belongs of the Bernoulli distribution family. In the next part, we will go over a comprehensive mathematical analysis.

In this essay, we are given data on the Titanic disaster, and we must analyse which groups of passengers are more likely to survive and which groups of passengers died in the disaster. We are provided numerous attributes on which to base our predictions, but we must determine whether any of them can have a substantial impact on the model's performance.

In this article, we will deduce mathematical solutions to logistic regression models and use existing Python libraries to fit the logistic model to the provided data. We'll also talk about the data insights we've gleaned and why we opted to change or remove specific features.

II. Logistic Regression

Logistic regression is an extension of linear regression, and by making two adjustments, we may expand linear regression to the (binary) classification scenario. First, we substitute the Gaussian distribution for y with a Bernoulli distribution, which is better suited for binary responses, $y \in \{0, 1\}$. That is, we employ

$$p(y|x, w) = Ber(y|\mu(x))$$

where $\mu(x) = E[y|x] = p(y = 1|x)$. Second, we compute a linear combination of the inputs, as before, but then we pass this through a function that ensures $0 \leq \mu(x) \leq 1$ by defining

$$\mu(x) = sigm(w^T x)$$

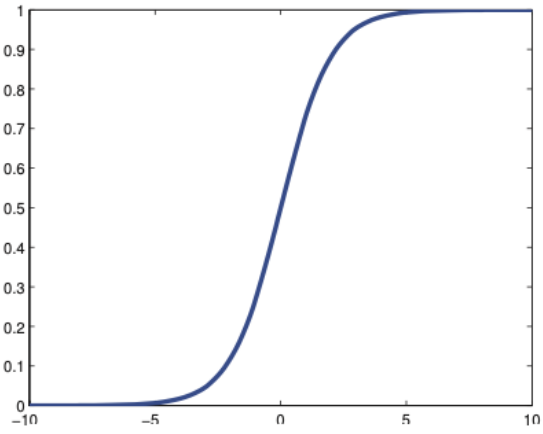
where $sigm(\eta)$ refers to the sigmoid function, also known as the logistic function. This is defined as

$$sigm(\eta) = \frac{\exp(\eta)}{\exp(\eta) + 1}$$

Putting these two steps together we get

$$p(y|x, w) = Ber(y|sigm(w^T x))$$

This is called logistic regression due to its similarity to linear regression (although it is a form of classification).



A simple example of logistic regression is shown in above figure, where we plot

$$p(y_i = 1|x_i, w) = sigm(w_0 + w_1 x_i)$$

If we threshold the output probability at 0.5, we can induce a decision rule of the form

$$\hat{y}(x) = 1 \iff p(y = 1|x) > 0.5$$

Logistic regression may be readily expanded to handle higher-dimensional inputs. If we set the threshold for these probabilities to 0.5, we have a linear decision boundary with the normal (perpendicular) provided by w. In the next part, we shall derive the loss function associated with logistic regression.

A. Maximum likelihood estimation

The negative log-likelihood for logistic regression is given by

$$NLL(w) = -\sum_{i=0}^{N} \log[\mu_i^{I(y_i=1)} \times (1-\mu_i)^{I(y_i=0)}]$$

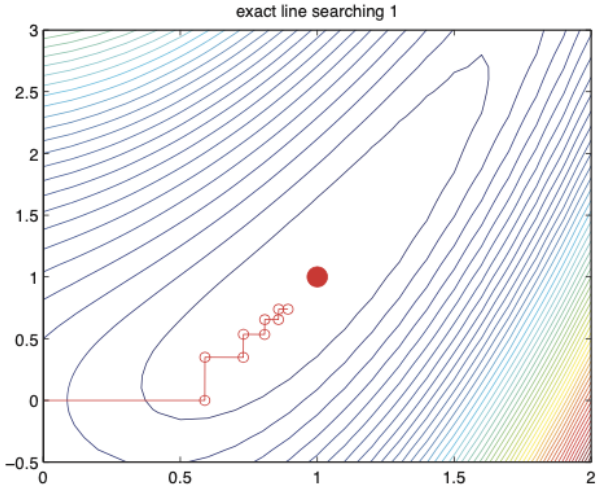
$$= -\sum_{i=0}^{N} [y_i \log \mu_i + (1-y_i)\log(1-\mu_i)]$$

This is also called the cross-entropy error function. Unlike linear regression, we can no longer write down the Maximum Likelihood Estimation in closed form. Instead, we need to use an optimization algorithm to compute it. For this, we need to derive the gradient and Hessian. In the case of logistic regression, the gradient and Hessian are given by the following

$$g = \frac{d}{dw} f(w) = \sum_i (\mu_i - y_i) x_i = X^T(\mu - y)$$

$$H = \frac{d}{dw} g(w)^T = \sum_i (\Delta_w \mu_i) x_i^T = X^T S X$$

where $S = diag(\mu_i(1-\mu_i))$. Here the NLL is convex and has a unique global minimum.



The simplest algorithm for unconstrained optimization is gradient descent (as shown in above figure), also known as steepest descent. This can be written as follows:

$$\Theta_{k+1} = \Theta_k - \eta_k g_k$$

where η_k is the learning rate.

III. The Problem

In the following sections, we will analyse and clean the Titanic dataset. We will also fit the machine learning model to the data and test if the model can make accurate predictions on previously unknown data.

A. Data Analysis and Feature Engineering

There were 10 feature columns and 1 target column in the given data. Given ten input variables, we must predict the target variable.

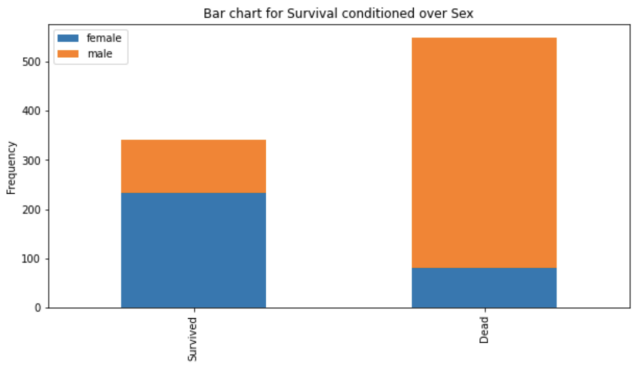
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

We imputed the columns with the median of their respective columns because some of the samples had `NaN` values.

We analysed the data by classifying characteristics according to the target variable (i.e. `Survival`) and displaying bar graphs to evaluate the importance of various attributes in predicting survival rate.

We plotted bar chart based of the Gender in the following figure:



We found that more percentage of females survived compared to the male passengers.

Further, we tried to categorize the passengers based on their age group and marital status. For extracting this information

from the data, we relied on `Age` and `Name` columns. We used `Name` column by extracting prefix and further dividing it's classes into `Mr`, `Mrs`, `Miss`, `Master` and `Other` category.

We calculated the probability of survival based on this column in the next figure:

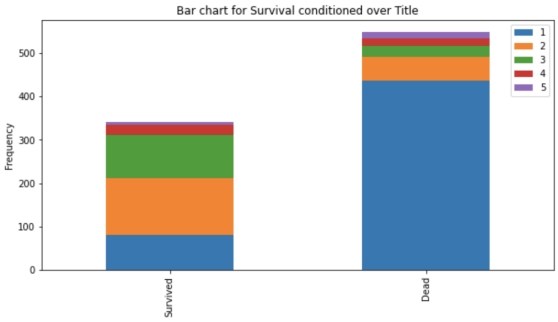
```
for dataset in [train, test]:
    dataset['Title'] = dataset['Title'].replace(['Lady', 'Countess','Ca
    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')

train[['Title', 'Survived']].groupby(['Title'], as_index=False).mean()
```

	Title	Survived
0	Master	0.575000
1	Miss	0.702703
2	Mr	0.156673
3	Mrs	0.793651
4	Other	0.347826

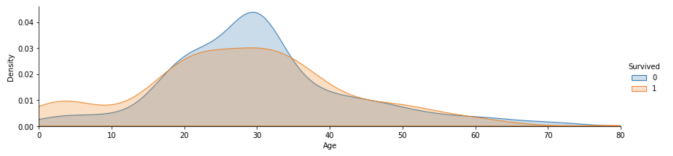
```
title_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Other": 5}
for dataset in [train, test]:
    dataset['Title'] = dataset['Title'].map(title_mapping)
    dataset['Title'] = dataset['Title'].fillna(0)

bar_chart('Title')
```



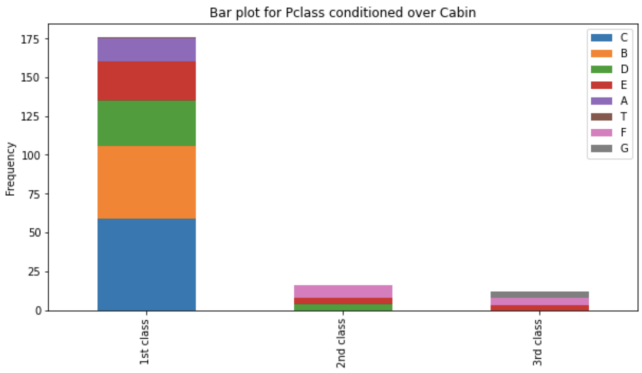
We concluded that married females outlived unmarried females and that married females outlived married males by a large margin. We hypothesised that married men died protecting their families.

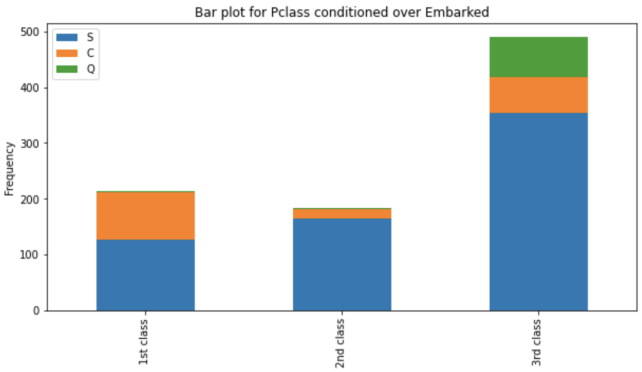
For understanding the age group of the survived people, we plotted the density plots of `Age` conditioned over `Survival`.



We discovered that persons between the ages of 18 and 35 died at a higher rate than people of other ages.

Finally, we attempted to comprehend the class of individuals by visualising which class of people dwelt in which cabin and where they embarked.





B. *Model Fitting*

We trained a Logistic regression model on the data using the `sklearn` library and achieved accuracy of `80%`

We have created a confusion matrix to better comprehend the ratio of `false negatives` and `false positives`.

```
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(target, model.predict(train_data))

index = ["Prediction positive", "Prediction negative"]
columns = ["Label positive", "Label negative"]
pd.DataFrame(matrix, columns=columns, index=index)
```

	Label positive	Label negative
Prediction positive	469	80
Prediction negative	94	248

IV. Conclusions

In this essay, we derived the solution of logistic regression mathematically based on some assumptions. We also applied Logistic regression techniques to real-world data and tried to inspect if the fitted model can be predict well. We conclude that provided the underlying assumptions are met, a basic model like logistic regression may be used on real-world complicated datasets.

References

[1] Christopher M. Bishop, "Pattern Recognition and Machine Learning"
[2] Kevin P. Murphy, "Machine Learning, A Probabilistic Perspective"
[3] Yunus Koloğlu, Hasan Birinci, Sevde Ilgaz Kanalmaz, Burhan Özyılmaz, "A Multiple Linear Regression Approach For Estimating the Market Value of Football Players in Forward Position"