10. Logistic Regression

Bruce E. Shapiro

Getting Started in Machine Learning

Last revised: February 24, 2019

Goal: categorize data

- Suppose there are two categories C_0, C_1 determined by x

$$y = P(C_1|x)$$

Goal: categorize data

- Suppose there are two categories C_0, C_1 determined by x

$$y = P(C_1|x) \implies 1 - y = P(C_0|x)$$

Goal: categorize data

- Suppose there are two categories C_0, C_1 determined by x

$$y = P(C_1|x) \implies 1 - y = P(C_0|x)$$

- Odds ratio is defined as $\dfrac{y}{1-y}$

Goal: categorize data

- Suppose there are two categories C_0, C_1 determined by x

$$y = P(C_1|x) \implies 1 - y = P(C_0|x)$$

- Odds ratio is defined as $\dfrac{y}{1-y}$
- Fit a linear model to the natural log of odds function:

$$\ln \frac{y}{1-y} = a + bx$$

Goal: categorize data

- Suppose there are two categories C_0, C_1 determined by x

$$y = P(C_1|x) \implies 1 - y = P(C_0|x)$$

- Odds ratio is defined as $\dfrac{y}{1-y}$
- Fit a linear model to the natural log of odds function:

$$\ln \frac{y}{1-y} = a + bx \implies y = \frac{1}{1 + e^{-(a+bx)}}$$

here a, b are undetermined parameters

Method of Logistic Regression

- Use method of least squares to find a, b; then the function

$$y = \frac{1}{1 + e^{-(a+bx)}}$$

gives the probability that a given input x falls into category C_1

- For classification purposes, we normally assign those inputs with $y \geq 0.5$ to C_1 and those with $y < 0.5$ to C_0

$$\frac{1}{1 + e^{-(a+bx_0)}} = \frac{1}{2} \implies x_0 = -a/b$$

The value 0.5 is not set in stone and can be changed if desired.

- Result: If $x > -a/b$, assign to C_1; otherwise, assign to C_0.

Gas Guzzlers

- **Example.** Classify a car as a gas-guzzler based on weight
- In 2016, the mean fleet MPG was 24.7 MPG.

Gas Guzzlers

- **Example.** Classify a car as a gas-guzzler based on weight
- In 2016, the mean fleet MPG was 24.7 MPG.
- Using the UCI/ML MPG data set, define a car as a gas guzzler if its MPG is is below 24.7.

Gas Guzzlers

- **Example.** Classify a car as a gas-guzzler based on weight
- In 2016, the mean fleet MPG was 24.7 MPG.
- Using the UCI/ML MPG data set, define a car as a gas guzzler if its MPG is is below 24.7.
- Convert all the MPG values to 1's/0's to indicate gas guzzler (1) or not a gas guzzler (0)

Gas Guzzlers

- **Example.** Classify a car as a gas-guzzler based on weight
- In 2016, the mean fleet MPG was 24.7 MPG.
- Using the UCI/ML MPG data set, define a car as a gas guzzler if its MPG is is below 24.7.
- Convert all the MPG values to 1's/0's to indicate gas guzzler (1) or not a gas guzzler (0)
- Extract the weights from the UCI/ML MPG data set as the X-values

Gas Guzzlers

- **Example.** Classify a car as a gas-guzzler based on weight
- In 2016, the mean fleet MPG was 24.7 MPG.
- Using the UCI/ML MPG data set, define a car as a gas guzzler if its MPG is is below 24.7.
- Convert all the MPG values to 1's/0's to indicate gas guzzler (1) or not a gas guzzler (0)
- Extract the weights from the UCI/ML MPG data set as the X-values
- Use logistic regression to predict class membership (Y-value, 1/0) based on weight (X-value)

- Read data set, label columns, drop rows with missing data

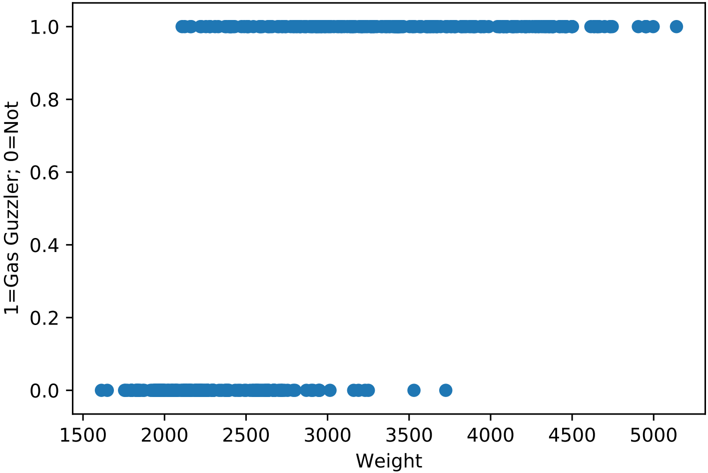
```
import pandas as pd

data=pd.read_fwf("https://archive.ics.uci.edu/ml/
  machine-learning-databases/auto-mpg/auto-mpg.data",
  header=None,na_values="?")
data.columns=("mpg","cyl","displ","hp","weight","accel",
  "model","origin","carname")
data = data.dropna(axis=0)

weight=data["weight"]
mpg=data["mpg"]
```

- Define class membership

```
mpg2016=24.7
gas_guzzler=[1 if z<mpg2016 else 0 for z in mpg]
```


1.0
0.8
0.6
0.4
0.2
0.0
1=Gas Guzzler; 0=Not
1500
2000
2500
3000
3500
4000
4500
5000
Weight

Define X and Y arrays:

```
X=np.array(weight).reshape(-1,1)
Y=np.array(gas_guzzler)
```

Define X and Y arrays:

```
X=np.array(weight).reshape(-1,1)
Y=np.array(gas_guzzler)
```

Do Logistic Regression on Training Set:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression as LR
XTRAIN,XTEST,YTRAIN,YTEST=train_test_split(X,Y)

r=LR().fit(XTRAIN,YTRAIN)
b=r.coef_[0,0]
a=r.intercept_[0]
x0=-a/b
print("a (intercept)=",a)
print("b (slope)=",b)
print("x0 (50% point)=",x0)
```

```
a (intercept)= -4.534251286801502
b (slope)= 0.0017664751497155893
x0 (50% point)= 2566.83559207246
```

Interpretation of result:

```
a (intercept)= -4.534251286801502
b (slope)= 0.0017664751497155893
x0 (50% point)= 2566.83559207246
```

The probability function is (approximately):

$$y \approx \frac{1}{1 + e^{-4.53425 + 0.00176647 \times \text{weight}}}$$

where y is the probability that a vehicle with a given weight is a gas guzzler, and $y = \dfrac{1}{2}$ when weight ≈ 2566.8

Evaluation - accuracy (Proportion Correct)

```
from sklearn.metrics import accuracy_score
YP=r.predict(XTEST)
accuracy_score(YP,YTEST)
```

```
0.8571428571428571
```

Evaluation - accuracy (Proportion Correct)

```
from sklearn.metrics import accuracy_score
YP=r.predict(XTEST)
accuracy_score(YP,YTEST)
```

```
0.8571428571428571
```

Evaluation - confusion matrix (See Chapter 11)
In a perfect prediction, it will be diagonal

```
from sklearn.metrics import confusion_matrix
confusion_matrix(YTEST,YP)
```

```
array([[38,  6],
       [ 8, 46]])
```

Multiple Feature Logistic Regression

```
xdata=data[["displ","hp","weight","accel"]]
xdata[:5]
X2=np.array(xdata)
X2TRAIN,X2TEST,Y2TRAIN,Y2TEST=train_test_split(xdata,Y)
b=r2.coef_
a=r2.intercept_

print("a (intercept)=",a)
print("b (slope)=",b)
```

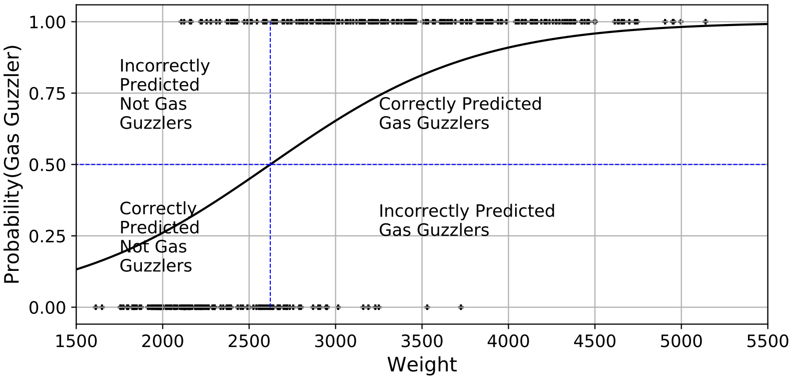
```
a (intercept)= [-0.2392631]
b (slope)= [[ 0.01192084 -0.02142693  0.00248871
  -0.38355612]]
```

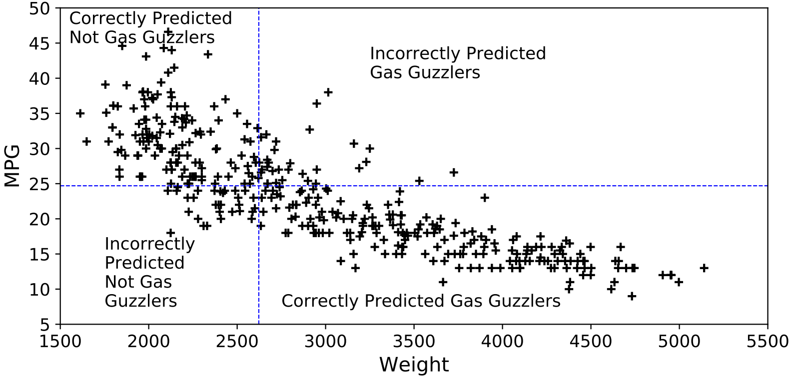
Interpretation of result:

```
a (intercept)= [-0.2392631]
b (slope)= [[ 0.01192084 -0.02142693  0.00248871
  -0.38355612]]
```

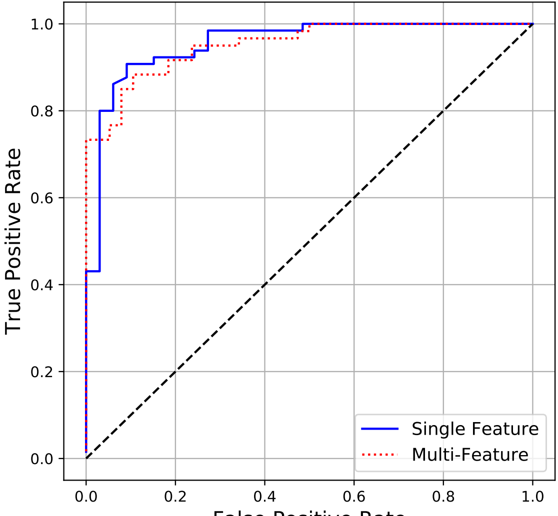
The probability that a car with a given **displ**, **hp**, **weight**, and **accel** will be a gas guzzler is

$$y \approx \frac{1}{1 + e^{-.24 + .012 \times \mathsf{displ} - .021 \times \mathsf{hp} + .0025 \times \mathsf{weight} - .38 \times \mathsf{accel}}}$$


Probability(Gas Guzzler)
1.00
0.75
0.50
0.25
0.00
Incorrectly Predicted Not Gas Guzzlers
Correctly Predicted Gas Guzzlers
Correctly Predicted Not Gas Guzzlers
Incorrectly Predicted Gas Guzzlers
1500
2000
2500
3000
3500
4000
4500
5000
5500
Weight


Correctly Predicted
Not Gas Guzzlers
Incorrectly Predicted
Gas Guzzlers
Incorrectly
Predicted
Not Gas
Guzzlers
Correctly Predicted Gas Guzzlers
MPG
Weight
50
45
40
35
30
25
20
15
10
5
1500
2000
2500
3000
3500
4000
4500
5000
5500

ROC Curve (See Chapter 11 for details)



References

- 2016 average fuel economy
  https://www.reuters.com/article/us-autos-emissions/
  u-s-vehicle-fuel-economy-rises-to-record-24-7-mpg-epa-idU
  downloaded 14 Dec 2018
- UCI MPG data set
  https://archive.ics.uci.edu/ml/datasets/auto+mpg;
  relevant paper: Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.
- Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science.