TBOS_CH_4_Classification

October 4, 2023

1 ISLP - Classification

Brenden Siekman Python coding by Stasia Colgan 2023-10-03

1.1 Classification

Classification is a statistical method for categorizing observations into discrete classes. Classification aims to predict a qualitative response for an observation, given one or more predictors. Classification can be used for inference as well as prediction. For example, medical researchers may want to know what symptoms or life styles are risk factors for a particular disease. A simpler model, like logistic regression may be best for this use case, so we can have interpretable coefficients from our model. Algorithms that implement classification are known as classifiers.

1.1.1 Examples of classification problems

1. Predicting credit default given a balance (Logistic Regression)
2. Spam email detection (Naive Bayes)
3. Credit card fraud detection (Random Forest / Gradient Boosted Trees)
4. Diagnosing an arrivin patient with a set of symptoms
5. Whether or not a passenger survived on the Titanic given demographic and ticket info

1.1.2 Some types of classifiers (less to more complex)

1. Naive Bayes
2. Logistic Regression
3. K-Nearest Neighbors
4. Decision Trees (CH 8)
5. Boosted Trees (CH 8)
6. Random Forest (CH 8)
7. Neural Networks (CH 10)

1.1.3 Considerations for which model to use

1. Interpretability: do we need to make sense of the model coefficients?
2. Robustness: how robust the model is to variations and noise in the data.
3. Precision: accuracy, sensitivity, specificity.
4. Speed / ease of prediciton.

1.2 Logistic Regression

Logistic regression is a statistical method for modeling the relationship between one or more predictor variables and a binary response variable. Unlike linear regression, which predicts a continuous outcome, logistic regression predicts the probability of an event occuring.

[ ]:

```
!pip install -U scikit-learn
```

[14]:

```
Default = pd.read_csv('Default.csv')
```

[13]:

```
import sklearn
import numpy as np
from matplotlib.pyplot import subplots
import statsmodels.api as sm
import pandas as pd
```

[15]:

```
Default
```

[15]:

```
     default student      balance        income
0         No      No   729.526495  44361.625074
1         No     Yes   817.180407  12106.134700
2         No      No  1073.549164  31767.138947
3         No      No   529.250605  35704.493935
4         No      No   785.655883  38463.495879
...      ...     ...          ...           ...
9995      No      No   711.555020  52992.378914
9996      No      No   757.962918  19660.721768
9997      No      No   845.411989  58636.156984
9998      No      No  1569.009053  36669.112365
9999      No     Yes   200.922183  16862.952321

[10000 rows x 4 columns]
```

[22]:

```
from sklearn.discriminant_analysis import (LinearDiscriminantAnalysis as LDA,␣
 ↪QuadraticDiscriminantAnalysis as QDA)
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

1.3 When to Use Logistic Regression vs. Other Classification Methods?

- **Binary Outcome**: Logistic regression is ideal when your response variable is binary (e.g. Yes/No, 1/0, True/False)
- **Linear Relationship**: Logistic regression assumes that the log odds of the response variable is a linear combination of the predictor variables. Does the logit function fit the data?
- **Interpretability**: If you need to understand the influence of individual predictors, logistic regression offers clear coefficients for interpretation.

- **Baseline Method**: Logistic regression is often used as a baeline against which more complex classifiers like decision trees, random forests, and gradient boosting machines are compared.

1.4 Mathematics Behind Logistic Regression

The fundamental idea behind logistic regression is to model the log odds of the probability of the event using a linear combination of predictors. Mathematically:

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ...\beta_n X_n$$

Where: - p is the probability of the event occuring - $X_1, X_2, ..., X_n$ are the predictor variables - $\beta_0, \beta_1, \beta_2, ..., \beta_n$ are the coefficients

To get the probability of p:

$$p = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n}}$$

1.5 Interpreting the Coefficients

To interpret the coefficients of a logistic regression: - **Intercept (β_0)**: the log odds of the event where all predictor variables are 0. - **Predictor(β_i)**: The change in log odds for a one-unit increase in the predictor variable. Specifically, if β_i is the coefficient for X_i, then e^{β_i} is the odds ratio associated with a one-unit increase in X_i.

For example if $\beta_i = 0.5$ for a predictor, then a one0unit increase in that predictor increases the odds of the event by $e^0.5$ or about 1.65 times.

To put this in terms of probability, let's use a hypothetical scenario:

Assume that, before the one-unit increase in X, the odds of the event occurring are 2:1. This translates to:

$$P_{base} = \frac{2}{2+1} = \frac{2}{3} \approx 0.66667$$

or 66.67%

Now, after the one-unit increase in X, the odds become $1.65 \times 2 = 3.3$, or 3:3:1. The new probability, after the increase, becomes:

$$P_{new} = \frac{3.3}{3.3+1} = \frac{3.3}{4.3} \approx 0.7674$$

or 76.74%

Therefore, in terms of probability, a one-unit increase in the predictor X increases the likelihood of the event from 66.67% to 76.74%.

1.6 How Good is our Model?

To assess 'goodness' for a classification model, we use a Confusion Matrix.

Type 1 Error: False positive conclusion **Type 2 Error**: False negative conclusion

For the confusion matrix shown above, correctly predicted values are located on the diagonal, while the off-diagonal elements represent misclassifications.

1. **Accuracy**:

- **Definition**: Accuracy is the proportion of true results (both true positivies and true negatives) in the population. It measures how often the classifier makes the correct prediction.
- Out of 10,000 borrowers, if your model correctly identifies 9,000 borrowers (either as in default or not in default), then the accuracy is 90%.

2. **Sensitivity**:

- **Definition**: Sensitivity measures the proportion of actual positives that are correctly identified.
- Of all the actual borrowers, how many did we correctly label as in default?

3. **Specificity**:

- **Definition**: Specificity measures the proportion of actual negatives that are correctly identified.
- Of all the borrowers NOT in default, how many did we label as not in default?

4. **Kappa Value**:

- **Definition**: The Kappa statistic is a metric that compares the observed accuracy with the expected accuracy (random chance).
- Suppose you and a friend both watch a movie and have to classify it as either good or bad. If you both agree most of the time, but more than what would be expected by pure luck, the Kappa statistic would capture this agreement level.

```
[23]: #Insert code for confusion matrix here
```

We can also use a Receiver Operating Characteristics (ROC) curve in which the measure of the ROC curve is the 'Area Under the Curve' (AUC).

```
[24]: #Insert code for ROC curve here
```

5. **AUC** (Area Under the Curve):

- **Definition**: AUC represents the probability that a random positive instance (from the data) will rank higher than a random negative instance. It's the area under the ROC curve (a plot of sensitivity versus 1-specifically).
- Imagine all the classified items are lined up from the most likely to be positive to the least. The AUC tells us the lieklihood that we've correctly ranked a pair of positive and negative items. An AUC of 1 is a perfect score. We want the curve to be as high and to the left as possible. An AUC of 0.5 indicates we've done no better than chance. Represented by the dashed line.

```
[ ]: #Insert code for the three goodness of fit measures below here
```

6. **McFadden's R-squared**:

- **Definition**: It is a measure that compares the likelihood of our model to the likelihood of a null model (a model with no predictors).

- It's akin to saying, "Here's how much better our model is compared to making predictions without any information." An R-squared value close to 1 suggests our model is close to perfect, while a value close to 0 suggests our model is no better than the null model.

7. **Residual Deviance**:

- **Definition**: It represents the lack of fit of our model to the data. A smaller deviance indicates a better fit.
- Think of this as a measure of "how off" our predictions ar from the actual outcomes. Smaller valus are better.

8. **Null Deviance**:

- **Definition**: It represents the difference between a model with no predictors and the actual data.
- This is our baeline "badness of fit" before any predictors are added. It gives context to the residual deviance - how much our model improves the fit compared to not modeling at all.

2 Classification with NHANES Dataset.

```
[ ]:
```