Unit 19: Logistic Regression Variable Selection

Case Study: Building a Model that is Good at Predicting Approval for the President's Foreign Policy with Age, Sex, and Political Affiliation *with New Data*

Suppose we work at a political advertising agency. Rather than seek to **understand the relationship** between approval for the president's foreign policy with sex, age, and political affiliation, we would like build a model that will give us the **best predictions** for adults living in the U.S. in which we *don't know what they think about the president's foreign policy*.

We can assume that this agency has the age, sex, political affiliation, and address of all registered voters in the state. So one goal that this political advertising agency might have is to use this data to make predictions about whether a given person that lives at a particular house approves of the president's foreign policy. They could then use that information to decide whether to mail political advertising pamphplets to this address.

Data Preliminaries

In [1]:
```
import numpy as np
import pandas as pd
import zipfile as zp
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

In [2]:
```
missing_values = ["NaN", "nan", "Don't know/Refused (VOL.)"]
df = pd.read_csv('Feb17public.csv',
                 na_values=missing_values)[['age', 'sex', 'q5cf1', 'party']]
```

In [3]:
```
# reduce q52 responses to two categories
# and create binary reponse variable
df['y'] = df['q5cf1'].map({'Disapprove':0,'Approve':1})
# use cleaned data without records that have missing values
dfclean = df.dropna()
```

In [4]:
```
dfclean.head()
```

Out[4]:

	age	sex	q5cf1	party	y
1	70.0	Female	Disapprove	Democrat	0.0
2	69.0	Female	Disapprove	Independent	0.0
4	70.0	Female	Disapprove	Democrat	0.0
6	89.0	Female	Disapprove	Independent	0.0
7	92.0	Female	Approve	Republican	1.0

In [5]:
```
dfclean['party'].value_counts()
```

Out[5]:
```
Democrat                256
Independent             235
Republican              172
No preference (VOL.)     12
Other party (VOL.)        4
Name: party, dtype: int64
```

In [6]:
```
dfclean['sex'].value_counts()
```

Out[6]:
```
Female    352
Male      327
Name: sex, dtype: int64
```

In [7]:
```
dfclean.describe()
```

Out[7]:

	age	y
count	679.000000	679.000000
mean	50.338733	0.366716
std	17.951594	0.482263
min	18.000000	0.000000
25%	35.000000	0.000000
50%	52.000000	0.000000
75%	65.000000	1.000000
max	94.000000	1.000000

1. Overfitting by Using Too Many Uninformative Explanatory Variables

See Unit 19 slides section 1

2. Some Pros and Cons of Overfitting vs. Underfitting a Model (via too Many or too Little Explanatory Variables

See Unit 19 slides section 2

3. Theory: Overfitting vs. Underfitting a Model

See Unit 19 slides section 3

3.1. A General Goal of Machine Learning

See Unit 19 slides section 3.1

3.2. Properties of the Estimation Function

See Unit 19 slides section 3.2

3.3 Estimation Function Definitions

See Unit 19 slides section 3.3

3.4. Relationship between Bias, Variance, Overfitting, Underfitting, and Mean Squared Error of a Model

See Unit 19 slides section 3.4

3.5. Goal of Selecting a Model that will Make Good Predictions on *New Data*

See Unit 19 slides section 3.5

4. Goal: Find a "Parsimonious" Model

See Unit 19 slides section 4

5. More about Fitting a Logistic Regression Model

See Unit 19 slides section 5

5.1. How are the optimal values of $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ determined in a logistic regression model.

See Unit 19 slides section 5.1

5.2. Where do we find the optimal log-likelihood function value for a given logistic regression model?

Let's first fit a logistic regression model that predicts the log likelihood that a adult living in the U.S. supports the president's foreign policy given the following explanatory variables:

- party,
- age, and
- sex.

In [8]:
```
pewmod = smf.logit('y ~ party + age + sex',\
    data=dfclean).fit()
pewmod.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.419649
         Iterations 7
```

Out[8]:

Logit Regression Results

Dep. Variable:	y	**No. Observations:**	679
Model:	Logit	**Df Residuals:**	672
Method:	MLE	**Df Model:**	6
Date:	Tue, 27 Apr 2021	**Pseudo R-squ.:**	0.3614
Time:	11:34:30	**Log-Likelihood:**	-284.94
converged:	True	**LL-Null:**	-446.23
Covariance Type:	nonrobust	**LLR p-value:**	1.185e-66

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.5635	0.465	-9.807	0.000	-5.475	-3.651
party[T.Independent]	2.2604	0.312	7.236	0.000	1.648	2.873
party[T.No preference (VOL.)]	2.5881	0.680	3.808	0.000	1.256	3.920
party[T.Other party (VOL.)]	4.0865	1.212	3.372	0.001	1.711	6.462
party[T.Republican]	4.2985	0.341	12.592	0.000	3.629	4.968
sex[T.Male]	0.7288	0.217	3.363	0.001	0.304	1.154
age	0.0272	0.006	4.443	0.000	0.015	0.039

6. Model Selection with Log Likelihood Ratio Test

Using Log-likelihood Ratio Test For Comparing two Logistic Regression Models

In an earlier section we considered two models for predicting a favorable opinion of border wall construction in the Pew Research Survey of February 2017. Let's load the data and the two models and first see how we can test between the two models. The idea is analogous to the ANOVA method for comparing two linear regression models.

<u>Descriptive Analytics Question:</u> Is the proportion of people that support the president's foreign policy different for at least one pair of political parties *in the sample*?

```
In [9]: dfclean.groupby('party').mean()
```

Out[9]:

party	age	y
Democrat	51.085938	0.058594
Independent	45.370213	0.357447
No preference (VOL.)	45.666667	0.416667
Other party (VOL.)	39.500000	0.750000
Republican	56.593023	0.825581

We can see that the proportion of 'favor' responses varies quite a bit between party affiliations, by looking at the mean values for 'y'. In each subgroup, the sample mean of y equals the proportion who favored the |president's foreign policy.

<u>Inference Question:</u> Is the proportion of people that support the president's foreign policy different for at least one pair of political parties *in the population* of all adults that live in the U.S.?

Use a Full model and reduced model for log-likelihood-ratio test

Recall that 'party' is a categorical variable with 5 categories. If we wish to test the null hypothesis of no party effects, we need a 4 degree of freedom test. For this we can use the log-likelihood-ratio test.

Step 1: Set up a full model and a null model.

- **Reduced Model (Model 0)**:
  - Response = Support for president's foreign policy
  - Explanatory Variables:
    - age
    - sex
- **Full Model (Model 1)**:
  - Response = Support for president's foreign policy
  - Explanatory Variables:
    - age
    - sex
    - party

First we fit the reduced and full model:

In [10]:
```
mod_red = smf.logit('y ~ age + sex', data=dfclean).fit()
mod_full = smf.logit('y ~ party + age + sex', data=dfclean).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.616503
         Iterations 5
Optimization terminated successfully.
         Current function value: 0.419649
         Iterations 7
```

Step 2: Set up the null and alternative hypotheses.

H_0 : Reduced Model is correct,

H_A : Reduced Model is incorrect because the missing 'party' coefficient in Reduced Model is not

Step 3: Calculate the test statistic.

We don't need to display the summaries to perform the test, but it is informative to review the model summaries to understand the variables. The maximized log-likelihood is shown in the model summary as 'Log-Likelihood'.

Step 3a: Extract the log-likelihoods for the two models:

In [11]:
```
mod_red.summary()
```

Out[11]:

Logit Regression Results

Dep. Variable:	y	**No. Observations:**	679
Model:	Logit	**Df Residuals:**	676
Method:	MLE	**Df Model:**	2
Date:	Tue, 27 Apr 2021	**Pseudo R-squ.:**	0.06190
Time:	11:34:30	**Log-Likelihood:**	-418.61
converged:	True	**LL-Null:**	-446.23
Covariance Type:	nonrobust	**LLR p-value:**	1.008e-12

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-2.3139	0.289	-8.017	0.000	-2.880	-1.748
sex[T.Male]	0.8720	0.168	5.201	0.000	0.543	1.201
age	0.0258	0.005	5.379	0.000	0.016	0.035

In [12]:
```
mod_full.summary()
```

Out[12]:

Logit Regression Results

Dep. Variable:	y	**No. Observations:**	679
Model:	Logit	**Df Residuals:**	672
Method:	MLE	**Df Model:**	6
Date:	Tue, 27 Apr 2021	**Pseudo R-squ.:**	0.3614
Time:	11:34:30	**Log-Likelihood:**	-284.94
converged:	True	**LL-Null:**	-446.23
Covariance Type:	nonrobust	**LLR p-value:**	1.185e-66

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.5635	0.465	-9.807	0.000	-5.475	-3.651
party[T.Independent]	2.2604	0.312	7.236	0.000	1.648	2.873
party[T.No preference (VOL.)]	2.5881	0.680	3.808	0.000	1.256	3.920
party[T.Other party (VOL.)]	4.0865	1.212	3.372	0.001	1.711	6.462
party[T.Republican]	4.2985	0.341	12.592	0.000	3.629	4.968
sex[T.Male]	0.7288	0.217	3.363	0.001	0.304	1.154
age	0.0272	0.006	4.443	0.000	0.015	0.039

In [13]:
```
mod_red.llf, mod_full.llf
```

Out[13]:
```
(-418.6053096733376, -284.9418430874059)
```

In [14]:
```
mod_red.df_model, mod_full.df_model
```

Out[14]:
```
(2.0, 6.0)
```

Step 3b: Use these log-likelihoods to calculate the likelihood ratio test statistic.

Just be careful to get the multiplier (-2) right so the chi-sqaure approximation works correctly.

In [15]:
```
# Extract log-likelihood function values
# and model degrees of freedom from each model
llf_red, df_red = mod_red.llf, mod_red.df_model
llf_full, df_full = mod_full.llf, mod_full.df_model
# take differences
llr, dfdiff = -2*(llf_red - llf_full), df_full - df_red
# display results
pd.DataFrame({'-2*llf': [-2*llf_red, -2*llf_full, llr],
              'df_model': [df_red, df_full, dfdiff]},
             index=['reduced model','full model', 'diff'])
```

Out[15]:

	-2*llf	df_model
reduced model	837.210619	2.0
full model	569.883686	6.0
diff	267.326933	4.0

Step 4: Calculate the degrees of freedom for the chi-squared distribution that this test statistic is an observation from.

Why was df = 4 in this analysis?

Step 5: Calculate the p-value and make a conclusion.

In [16]:
```
# import chisquare function and compute p-value
from scipy.stats import chi2
1 - chi2.cdf(llr, df=dfdiff)
```

Out[16]: 0.0

Summarize the test with calculated p-value using chi-square distribution

In [17]:
```
# summarize test results
print('-2*llr:', round(llr, 2), \
      ' df:', dfdiff, ' p-value:', \
      1 - chi2.cdf(llr, df=dfdiff))
```

```
-2*llr: 267.33  df: 4.0  p-value: 0.0
```

Conclusion:

Because the $p - value < 0.0001 < \alpha = 0.05$, we reject the null hypothesis. Thus there is sufficient evidence to suggest that the reduced Model is incorrect because the missing 'party' coefficient in Reduced Model is not zero.

Or in other words, there is sufficient evidence to suggest that party affiliation is a significant factor associated with support for the president's foreign policy.

7. Model Selection with AIC and BIC

See Unit 19 slides section 7

In the previous section we used a hypothesis test to evaluate whether there was evidence to suggest that the full model with the party slopes (as well as the age and sex slope) was a better fit of the data then the reduced model which just had the age and sex slopes.

Now, let's use the AIC and BIC scores of the full and reduced model to see which of the two models is closer to being a parsimonious model.

In [18]:
```
mod_full.summary()
```

Out[18]:

Logit Regression Results

Dep. Variable:	y	**No. Observations:**	679
Model:	Logit	**Df Residuals:**	672
Method:	MLE	**Df Model:**	6
Date:	Tue, 27 Apr 2021	**Pseudo R-squ.:**	0.3614
Time:	11:34:31	**Log-Likelihood:**	-284.94
converged:	True	**LL-Null:**	-446.23
Covariance Type:	nonrobust	**LLR p-value:**	1.185e-66

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.5635	0.465	-9.807	0.000	-5.475	-3.651
party[T.Independent]	2.2604	0.312	7.236	0.000	1.648	2.873
party[T.No preference (VOL.)]	2.5881	0.680	3.808	0.000	1.256	3.920
party[T.Other party (VOL.)]	4.0865	1.212	3.372	0.001	1.711	6.462
party[T.Republican]	4.2985	0.341	12.592	0.000	3.629	4.968
sex[T.Male]	0.7288	0.217	3.363	0.001	0.304	1.154
age	0.0272	0.006	4.443	0.000	0.015	0.039

In [19]:
```
mod_red.summary()
```

Out[19]:

Logit Regression Results

Dep. Variable:	y	**No. Observations:**	679
Model:	Logit	**Df Residuals:**	676
Method:	MLE	**Df Model:**	2
Date:	Tue, 27 Apr 2021	**Pseudo R-squ.:**	0.06190
Time:	11:34:31	**Log-Likelihood:**	-418.61
converged:	True	**LL-Null:**	-446.23
Covariance Type:	nonrobust	**LLR p-value:**	1.008e-12

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-2.3139	0.289	-8.017	0.000	-2.880	-1.748
sex[T.Male]	0.8720	0.168	5.201	0.000	0.543	1.201
age	0.0258	0.005	5.379	0.000	0.016	0.035

Extracing the AIC and BIC

We can extract the AIC from a given model output by using the **.aic** parameter.

In [20]:
```
print('Full Model AIC:',mod_full.aic)
print('Reduced Model AIC:',mod_red.aic)
```

```
Full Model AIC: 583.8836861748118
Reduced Model AIC: 843.2106193466752
```

Similarly, we can extract the BIC from a given model output by using the **.bic** parameter.

In [21]:
```
print('Full Model BIC:',mod_full.bic)
print('Reduced Model BIC:',mod_red.bic)
```

```
Full Model BIC: 615.5280340677227
Reduced Model BIC: 856.7724827293513
```

Conclusion:

Both AIC and BIC favor the full model. This suggests that the reduced model is too simple, so the bias due to omitted variables is too large for this model compared to the full model.

Or in other words, by adding the party explanatory variable to the full model, the improvment (ie. increase) in the optimal log likelihood function value (ie. predictive power) was great enough to counterbalance the addition of another slope (which contributes more to overfitting).

In [ ]:
```
```