index

March 7, 2022

1 Fitting a Logistic Regression Model - Lab

1.1 Introduction

In the last lesson you were given a broad overview of logistic regression. This included an introduction to two separate packages for creating logistic regression models. In this lab, you'll be investigating fitting logistic regressions with `statsmodels`. For your first foray into logistic regression, you are going to attempt to build a model that classifies whether an individual survived the Titanic shipwreck or not (yes, it's a bit morbid).

1.2 Objectives

In this lab you will:

- Implement logistic regression with `statsmodels`
- Interpret the statistical results associated with model parameters

1.3 Import the data

Import the data stored in the file `'titanic.csv'` and print the first five rows of the DataFrame to check its contents.

[4]:
```
# Import the data
import pandas as pd

df = pd.read_csv("titanic.csv")
df.head()
```

[4]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
```

```
4                            Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

1.4 Define independent and target variables

Your target variable is in the column `'Survived'`. A `0` indicates that the passenger didn't survive the shipwreck. Print the total number of people who didn't survive the shipwreck. How many people survived?

[18]:
```
# Total number of people who survived/didn't survive
y = df["Survived"]
print(df["Survived"].value_counts())
print("Survived: ", len(df[df["Survived"]==1]))
print("Did not Survived: ", len(df[df["Survived"]==0]))
```

```
0    549
1    342
Name: Survived, dtype: int64
Survived:  342
Did not Survived:  549
```

Only consider the columns specified in `relevant_columns` when building your model. The next step is to create dummy variables from categorical variables. Remember to drop the first level for each categorical column and make sure all the values are of type `float`:

[19]:
```
# Create dummy variables
relevant_columns = ['Pclass', 'Age', 'SibSp', 'Fare', 'Sex', 'Embarked',␣
 ↪'Survived']
dummy_dataframe = pd.get_dummies(df[relevant_columns], drop_first = True,
                                 dtype = float)

dummy_dataframe.shape
```

[19]:
```
(891, 8)
```

Did you notice above that the DataFrame contains missing values? To keep things simple, simply delete all rows with missing values.

NOTE: You can use the `.dropna()` method to do this.

[20]:
```
# Drop missing rows
dummy_dataframe = dummy_dataframe.dropna()
dummy_dataframe.shape
```

[20]: (714, 8)

Finally, assign the independent variables to `X` and the target variable to `y`:

[21]:
```
# Split the data into X and y
y = dummy_dataframe["Survived"]
X = dummy_dataframe.drop(columns = ["Survived"], axis = 1)
```

1.5 Fit the model

Now with everything in place, you can build a logistic regression model using `statsmodels` (make sure you create an intercept term as we showed in the previous lesson).

> Warning: Did you receive an error of the form "LinAlgError: Singular matrix"? This means that `statsmodels` was unable to fit the model due to certain linear algebra computational problems. Specifically, the matrix was not invertible due to not being full rank. In other words, there was a lot of redundant, superfluous data. Try removing some features from the model and running it again.

[23]:
```
# Build a logistic regression model using statsmodels
import statsmodels.api as sm

X = sm.add_constant(X)
logit_model = sm.Logit(y,X)
result = logit_model.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.443267
         Iterations 6
```

1.6 Analyze results

Generate the summary table for your model. Then, comment on the p-values associated with the various features you chose.

[24]:
```
# Summary table
result.summary()
```

[24]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:                Survived   No. Observations:                  714
Model:                           Logit   Df Residuals:                      706
Method:                            MLE   Df Model:                            7
Date:                 Mon, 07 Mar 2022   Pseudo R-squ.:                 0.3437
Time:                         18:43:52   Log-Likelihood:                -316.49
converged:                       True   LL-Null:                       -482.26
Covariance Type:            nonrobust   LLR p-value:                 1.103e-67
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.6503      0.633      8.921      0.000       4.409       6.892
Pclass        -1.2118      0.163     -7.433      0.000      -1.531      -0.892
Age           -0.0431      0.008     -5.250      0.000      -0.059      -0.027
SibSp         -0.3806      0.125     -3.048      0.002      -0.625      -0.136
Fare           0.0012      0.002      0.474      0.636      -0.004       0.006
Sex_male      -2.6236      0.217    -12.081      0.000      -3.049      -2.198
Embarked_Q    -0.8260      0.598     -1.381      0.167      -1.999       0.347
Embarked_S    -0.4130      0.269     -1.533      0.125      -0.941       0.115
==============================================================================
"""
```

[25]:
```
# Your comments here
high_pval = ["Fare", "Embarked_Q", "Embarked_S"]
```

1.7 Level up (Optional)

Create a new model, this time only using those features you determined were influential based on your analysis of the results above. How does this model perform?

[29]:
```
# Your code here
to_drop = ["Fare", "Embarked_Q", "Embarked_S", "Survived"]
y = dummy_dataframe["Survived"]

X = dummy_dataframe.drop(columns = to_drop, axis = 1)

X = sm.add_constant(X)
second_model = sm.Logit(y,X)
results = second_model.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.445882
         Iterations 6
```

[29]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:                Survived   No. Observations:                  714
Model:                           Logit   Df Residuals:                      709
Method:                            MLE   Df Model:                            4
Date:                 Mon, 07 Mar 2022   Pseudo R-squ.:                 0.3399
Time:                         18:47:12   Log-Likelihood:                -318.36
converged:                       True   LL-Null:                       -482.26
```

```
Covariance Type:            nonrobust   LLR p-value:                 1.089e-69
================================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const          5.6008      0.543     10.306      0.000       4.536       6.666
Pclass        -1.3174      0.141     -9.350      0.000      -1.594      -1.041
Age           -0.0444      0.008     -5.442      0.000      -0.060      -0.028
SibSp         -0.3761      0.121     -3.106      0.002      -0.613      -0.139
Sex_male      -2.6235      0.215    -12.229      0.000      -3.044      -2.203
================================================================================
"""
```

[ ]:
```
# Your comments here
```

1.8 Summary

Well done! In this lab, you practiced using `statsmodels` to build a logistic regression model. You then interpreted the results, building upon your previous stats knowledge, similar to linear regression. Continue on to take a look at building logistic regression models in Scikit-learn!