Exemplar_Perform logistic regression

January 7, 2024

1 Exemplar: Perform logistic regression

1.1 Introduction

In this activity, you will complete an effective bionomial logistic regression. This exercise will help you better understand the value of using logistic regression to make predictions for a dependent variable based on one independent variable and help you build confidence in practicing logistic regression. Because logistic regression is leveraged across a wide array of industries, becoming proficient in this process will help you expand your skill set in a widely-applicable way.

For this activity, you work as a consultant for an airline. The airline is interested in knowing if a better in-flight entertainment experience leads to higher customer satisfaction. They would like you to construct and evaluate a model that predicts whether a future customer would be satisfied with their services given previous customer feedback about their flight experience.

The data for this activity is for a sample size of 129,880 customers. It includes data points such as class, flight distance, and in-flight entertainment, among others. Your goal will be to utilize a binomial logistic regression model to help the airline model and better understand this data.

Because this activity uses a dataset from the industry, you will need to conduct basic EDA, data cleaning, and other manipulations to prepare the data for modeling.

In this activity, you will practice the following skills:

- Importing packages and loading data
- Exploring the data and completing the cleaning process
- Building a binomial logistic regression model
- Evaluating a binomial logistic regression model using a confusion matrix

1.2 Step 1: Imports

1.2.1 Import packages

Import relevant Python packages. Use `train_test_split`, `LogisticRegression`, and various imports from `sklearn.metrics` to build, visualize, and evalute the model.

[1]:
```
### YOUR CODE HERE ###

# Standard operational package imports.
```

```
import numpy as np
import pandas as pd

# Important imports for preprocessing, modeling, and evaluation.
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import sklearn.metrics as metrics

# Visualization package imports.
import matplotlib.pyplot as plt
import seaborn as sns
```

1.2.2 Load the dataset

Load the **Invistico_Airline.csv** dataset. Save the resulting pandas DataFrame in a variable named `df_original`.

[2]:
```
### YOUR CODE HERE ###

df_original = pd.read_csv("Invistico_Airline.csv")
```

Hint 1

Use a function from the pandas library to read in the csv file.

Hint 2

Use the `read_csv` function and pass in the filename as a string.

Hint 3

Use `pd.read_csv("insertfilenamehere")`.

1.2.3 Output the first 10 rows

Output the first 10 rows of data.

[3]:
```
### YOUR CODE HERE ###

df_original.head(n = 10)
```

[3]:

	satisfaction	Customer Type	Age	Type of Travel	Class	\
0	satisfied	Loyal Customer	65	Personal Travel	Eco	
1	satisfied	Loyal Customer	47	Personal Travel	Business	
2	satisfied	Loyal Customer	15	Personal Travel	Eco	
3	satisfied	Loyal Customer	60	Personal Travel	Eco	
4	satisfied	Loyal Customer	70	Personal Travel	Eco	

```
5    satisfied  Loyal Customer   30  Personal Travel       Eco
6    satisfied  Loyal Customer   66  Personal Travel       Eco
7    satisfied  Loyal Customer   10  Personal Travel       Eco
8    satisfied  Loyal Customer   56  Personal Travel  Business
9    satisfied  Loyal Customer   22  Personal Travel       Eco

   Flight Distance  Seat comfort  Departure/Arrival time convenient  \
0              265             0                                  0
1             2464             0                                  0
2             2138             0                                  0
3              623             0                                  0
4              354             0                                  0
5             1894             0                                  0
6              227             0                                  0
7             1812             0                                  0
8               73             0                                  0
9             1556             0                                  0

   Food and drink  Gate location  …  Online support  Ease of Online booking  \
0               0              2  …               2                       3
1               0              3  …               2                       3
2               0              3  …               2                       2
3               0              3  …               3                       1
4               0              3  …               4                       2
5               0              3  …               2                       2
6               0              3  …               5                       5
7               0              3  …               2                       2
8               0              3  …               5                       4
9               0              3  …               2                       2

   On-board service  Leg room service  Baggage handling  Checkin service  \
0                 3                 0                 3                5
1                 4                 4                 4                2
2                 3                 3                 4                4
3                 1                 0                 1                4
4                 2                 0                 2                4
5                 5                 4                 5                5
6                 5                 0                 5                5
7                 3                 3                 4                5
8                 4                 0                 1                5
9                 2                 4                 5                3

   Cleanliness  Online boarding  Departure Delay in Minutes  \
0            3                2                           0
1            3                2                         310
2            4                2                           0
3            1                3                           0
```

```
4            2                5                          0
5            4                2                          0
6            5                3                         17
7            4                2                          0
8            4                4                          0
9            4                2                         30

   Arrival Delay in Minutes
0                       0.0
1                     305.0
2                       0.0
3                       0.0
4                       0.0
5                       0.0
6                      15.0
7                       0.0
8                       0.0
9                      26.0

[10 rows x 22 columns]
```

Hint 1

Use the `head()` function.

Hint 2

If only five rows are output, it is because the function by default returns five rows. To change this, specify how many rows `(n = )` you want to output.

1.3 Step 2: Data exploration, data cleaning, and model preparation

1.3.1 Prepare the data

After loading the dataset, prepare the data to be suitable for a logistic regression model. This includes:

- Exploring the data
- Checking for missing values
- Encoding the data
- Renaming a column
- Creating the training and testing data

1.3.2 Explore the data

Check the data type of each column. Note that logistic regression models expect numeric data.

[4]:
```
### YOUR CODE HERE ###

df_original.dtypes
```

[4]:
```
satisfaction                          object
Customer Type                         object
Age                                    int64
Type of Travel                        object
Class                                 object
Flight Distance                        int64
Seat comfort                           int64
Departure/Arrival time convenient      int64
Food and drink                         int64
Gate location                          int64
Inflight wifi service                  int64
Inflight entertainment                 int64
Online support                         int64
Ease of Online booking                 int64
On-board service                       int64
Leg room service                       int64
Baggage handling                       int64
Checkin service                        int64
Cleanliness                            int64
Online boarding                        int64
Departure Delay in Minutes             int64
Arrival Delay in Minutes             float64
dtype: object
```

Hint 1

Use the `dtypes` attribute on the DataFrame.

1.3.3 Check the number of satisfied customers in the dataset

To predict customer satisfaction, check how many customers in the dataset are satisfied before modeling.

[5]:
```
### YOUR CODE HERE ###

df_original['satisfaction'].value_counts(dropna = False)
```

[5]:
```
satisfied       71087
dissatisfied    58793
Name: satisfaction, dtype: int64
```

Hint 1

Use a function from the pandas library that returns a pandas series containing counts of unique values.

Hint 2

Use the `value_counts()` function. To examine how many NaN values there are, set the `dropna` parameter passed in to this function to `False`.

Question: How many satisfied and dissatisfied customers were there?

There were 71,087 satisfied customers and 58,793 dissatisfied customers.

Question: What percentage of customers were satisfied?

54.7 percent (71,087/129,880) of customers were satisfied. While this is a simple calculation, this value can be compared to a logistic regression model's accuracy.

1.3.4 Check for missing values

An assumption of logistic regression models is that there are no missing values. Check for missing values in the rows of the data.

[6]:
```
### YOUR CODE HERE ###

df_original.isnull().sum()
```

[6]:
```
satisfaction                          0
Customer Type                         0
Age                                   0
Type of Travel                        0
Class                                 0
Flight Distance                       0
Seat comfort                          0
Departure/Arrival time convenient     0
Food and drink                        0
Gate location                         0
Inflight wifi service                 0
Inflight entertainment                0
Online support                        0
Ease of Online booking                0
On-board service                      0
Leg room service                      0
Baggage handling                      0
Checkin service                       0
Cleanliness                           0
Online boarding                       0
Departure Delay in Minutes            0
Arrival Delay in Minutes            393
dtype: int64
```

Hint 1

To get the number of rows in the data with missing values, use the `isnull` function followed by the `sum` function.

Question: Should you remove rows where the `Arrival Delay in Minutes` column has missing values, even though the airline is more interested in the `inflight entertainment` column?

For this activity, the airline is specifically interested in knowing if a better in-flight entertainment experience leads to higher customer satisfaction. The `Arrival Delay in Minutes` column won't be included in the binomial logistic regression model; however, the airline might become interested in this column in the future.

For now, the missing values should be removed for two reasons:

- There are only 393 missing values out of the total of 129,880, so these are a small percentage of the total.
- This column might impact the relationship between entertainment and satisfaction.

1.3.5 Drop the rows with missing values

Drop the rows with missing values and save the resulting pandas DataFrame in a variable named `df_subset`.

[7]:
```
### YOUR CODE HERE ###

df_subset = df_original.dropna(axis=0).reset_index(drop = True)
```

Hint 1

Use the `dropna` function.

Hint 2

Set the axis parameter passed into the `dropna` function to `0` if you want to drop rows containing missing values, or `1` if you want to drop columns containing missing values. Optionally, use reset_index to avoid a SettingWithCopy warning later in the notebook.

1.3.6 Prepare the data

If you want to create a plot (`sns.regplot`) of your model to visualize results later in the notebook, the independent variable `Inflight entertainment` cannot be "of type int" and the dependent variable `satisfaction` cannot be "of type object."

Make the `Inflight entertainment` column "of type float."

[8]:
```
### YOUR CODE HERE ###

df_subset = df_subset.astype({"Inflight entertainment": float})
```

Hint 1

Use the `.astype()` function with the dictionary `{"Inflight entertainment": float}` as an input.

1.3.7 Convert the categorical column `satisfaction` into numeric

Convert the categorical column `satisfaction` into numeric through one-hot encoding.

[9]:
```
### YOUR CODE HERE ###

df_subset['satisfaction'] = OneHotEncoder(drop='first').
↪fit_transform(df_subset[['satisfaction']]).toarray()
```

Hint 1

Use `OneHotEncoder()` from `sklearn.preprocessing`.

Hint 2

Call `OneHotEncoder()`, specifying the `drop` argument as `'first'` in order to remove redundant columns from the output.

Call `.fit_transform()`, passing in the subset of the data that you want to encode (the subset consisting of `satisfaction`).

Call `.toarray()` in order to convert the sparse matrix that `.fit_transform()` returns into an array.

Hint 3

Index `df_subset` with a double pair of square brackets to get a DataFrame that consists of just `satisfaction`.

After getting the encoded values, update the `satisfaction` column (you can use reassignment).

1.3.8 Output the first 10 rows of `df_subset`

To examine what one-hot encoding did to the DataFrame, output the first 10 rows of `df_subset`.

[10]:
```
### YOUR CODE HERE ###

df_subset.head(10)
```

[10]:

	satisfaction	Customer Type	Age	Type of Travel	Class
0	1.0	Loyal Customer	65	Personal Travel	Eco
1	1.0	Loyal Customer	47	Personal Travel	Business
2	1.0	Loyal Customer	15	Personal Travel	Eco
3	1.0	Loyal Customer	60	Personal Travel	Eco
4	1.0	Loyal Customer	70	Personal Travel	Eco
5	1.0	Loyal Customer	30	Personal Travel	Eco
6	1.0	Loyal Customer	66	Personal Travel	Eco
7	1.0	Loyal Customer	10	Personal Travel	Eco

```
8           1.0  Loyal Customer   56  Personal Travel  Business
9           1.0  Loyal Customer   22  Personal Travel       Eco

   Flight Distance  Seat comfort  Departure/Arrival time convenient  \
0              265             0                                  0
1             2464             0                                  0
2             2138             0                                  0
3              623             0                                  0
4              354             0                                  0
5             1894             0                                  0
6              227             0                                  0
7             1812             0                                  0
8               73             0                                  0
9             1556             0                                  0

   Food and drink  Gate location  …  Online support  Ease of Online booking  \
0               0              2  …               2                       3
1               0              3  …               2                       3
2               0              3  …               2                       2
3               0              3  …               3                       1
4               0              3  …               4                       2
5               0              3  …               2                       2
6               0              3  …               5                       5
7               0              3  …               2                       2
8               0              3  …               5                       4
9               0              3  …               2                       2

   On-board service  Leg room service  Baggage handling  Checkin service  \
0                 3                 0                 3                5
1                 4                 4                 4                2
2                 3                 3                 4                4
3                 1                 0                 1                4
4                 2                 0                 2                4
5                 5                 4                 5                5
6                 5                 0                 5                5
7                 3                 3                 4                5
8                 4                 0                 1                5
9                 2                 4                 5                3

   Cleanliness  Online boarding  Departure Delay in Minutes  \
0            3                2                           0
1            3                2                         310
2            4                2                           0
3            1                3                           0
4            2                5                           0
5            4                2                           0
6            5                3                          17
```

```
7            4                2                          0
8            4                4                          0
9            4                2                         30

   Arrival Delay in Minutes
0                       0.0
1                     305.0
2                       0.0
3                       0.0
4                       0.0
5                       0.0
6                      15.0
7                       0.0
8                       0.0
9                      26.0

[10 rows x 22 columns]
```

Hint 1

Use the `head()` function.

Hint 2

If only five rows are outputted, it is because the function by default returns five rows. To change this, specify how many rows `(n = )` you want.

1.3.9 Create the training and testing data

Put 70% of the data into a training set and the remaining 30% into a testing set. Create an X and y DataFrame with only the necessary variables.

[11]:
```
### YOUR CODE HERE ###

X = df_subset[["Inflight entertainment"]]
y = df_subset["satisfaction"]

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,␣
↪random_state=42)
```

Hint 1

Use `train_test_split`.

Hint 2

If you named your independent variable `X` and your dependent variable `y`, then it would be `train_test_split(X, y, test_size=0.30, random_state=42)`.

Hint 3

When you use `train_test_split`, pass in `42` to `random_state`. `random_state` is used so that if other data professionals run this code, they can get the same exact train test split. If you use a different random state, your results will differ.

Question: If you want to consider customer satisfaction with your model, should you train your model to use `inflight entertainment` as your sole independent variable?

Other variables, like `Departure Delay in Minutes` seem like they can be potentially influential to customer satisfaction. This is why only using one independent variable might not be ideal.

1.4 Step 3: Model building

1.4.1 Fit a LogisticRegression model to the data

Build a logistic regression model and fit the model to the training data.

[12]:
```
### YOUR CODE HERE ###

clf = LogisticRegression().fit(X_train,y_train)
```

Hint 1

Use `LogisticRegression()` and the `fit()` function on the training set. `LogisticRegression().fit(X_train,y_train)`.

1.4.2 Obtain parameter estimates

Make sure you output the two parameters from your model.

[13]:
```
### YOUR CODE HERE ###

clf.coef_
```

[13]:
```
array([[0.99751462]])
```

[14]:
```
### YOUR CODE HERE ###

clf.intercept_
```

[14]:
```
array([-3.19355406])
```

Hint 1

Refer to the content on obtaining the parameter estimates from a logistic regression model.

Hint 2

Call attributes to obtain the coefficient and intercept estimates.

Hint 3

Use `.coef_` and `.intercept_`

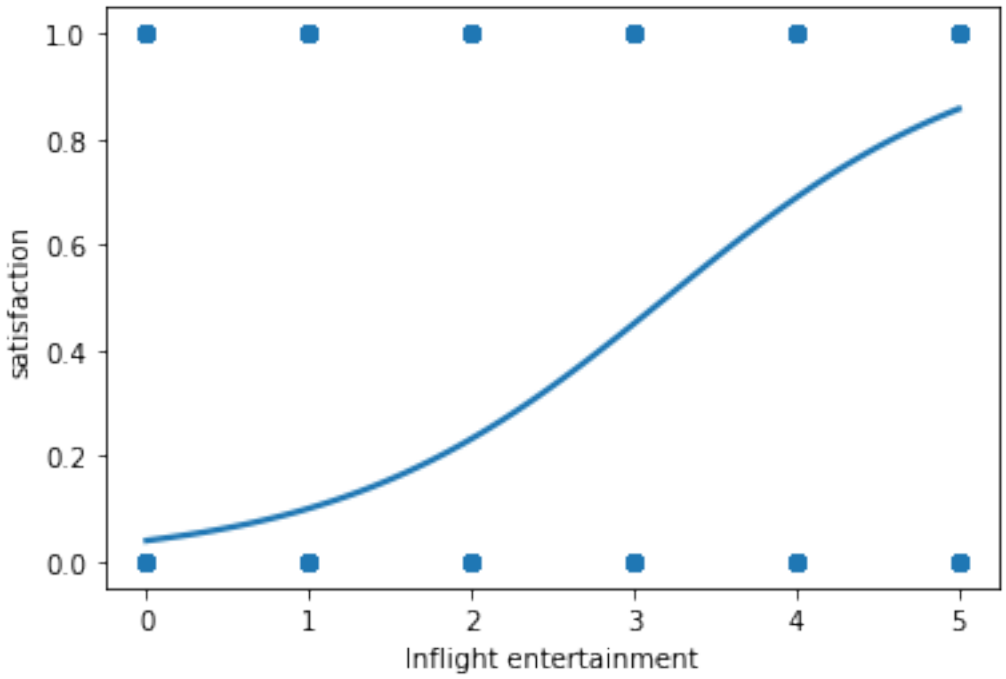
1.4.3 Create a plot of your model

Create a plot of your model to visualize results using the seaborn package.

[15]:
```
### YOUR CODE HERE ###

sns.regplot(x="Inflight entertainment", y="satisfaction", data=df_subset,␣
↪logistic=True, ci=None)
```

[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd05d8dff50>



Hint 1

Use a function from the seaborn library that can plot data and a logistic regression model fit.

Hint 2

Use the `regplot` function.

Hint 3

Set the `logistic` parameter passed in to this function to `True` to estimate a logistic regression model.

Question: What can you tell from the graph?

The graph seems to indicate that the higher the `inflight entertainment` value, the higher the customer satisfaction, though this is currently not the most informative plot. The graph currently doesn't provide much insight into the data points, as `Inflight entertainment` is categorical.

1.5 Step 4: Results and evaluation

1.5.1 Predict the outcome for the test dataset

Now that you've completed your regression, review and analyze your results. First, input the holdout dataset into the `predict` function to get the predicted labels from the model. Save these predictions as a variable called `y_pred`.

[16]:
```
### YOUR CODE HERE ###

# Save predictions.

y_pred = clf.predict(X_test)
```

1.5.2 Print out `y_pred`

In order to examine the predictions, print out `y_pred`.

[17]:
```
### YOUR CODE HERE ###
print(y_pred)
```

```
[1. 0. 0. … 0. 0. 0.]
```

1.5.3 Use the `predict_proba` and `predict` functions on `X_test`

[18]:
```
# Use predict_proba to output a probability.

### YOUR CODE HERE ###
clf.predict_proba(X_test)
```

[18]:
```
array([[0.14258068, 0.85741932],
       [0.55008402, 0.44991598],
       [0.89989329, 0.10010671],
       …,
       [0.89989329, 0.10010671],
       [0.76826225, 0.23173775],
       [0.55008402, 0.44991598]])
```

Hint 1

Using the `predict_proba` function on `X_test` will produce the probability that each observation is a 0 or 1.

[19]:
```
# Use predict to output 0's and 1's.

### YOUR CODE HERE ###

clf.predict(X_test)
```

[19]:
```
array([1., 0., 0., …, 0., 0., 0.])
```

Hint 2

`clf.predict` outputs an array of 0's and 1's, where 0's are not satisfied and 1's are satisfied.

1.5.4 Analyze the results

Print out the model's accuracy, precision, recall, and F1 score.

[20]:
```
### YOUR CODE HERE ###

print("Accuracy:", "%.6f" % metrics.accuracy_score(y_test, y_pred))
print("Precision:", "%.6f" % metrics.precision_score(y_test, y_pred))
print("Recall:", "%.6f" % metrics.recall_score(y_test, y_pred))
print("F1 Score:", "%.6f" % metrics.f1_score(y_test, y_pred))
```

```
Accuracy: 0.801529
Precision: 0.816142
Recall: 0.821530
F1 Score: 0.818827
```

Hint 1

Use four different functions from `metrics` to get the accuracy, precision, recall, and F1 score.

Hint 2

Input `y_test` and `y_pred` into the `metrics.accuracy_score`, `metrics.precision_score`, `metrics.recall_score`, and `metrics.f1_score` functions.

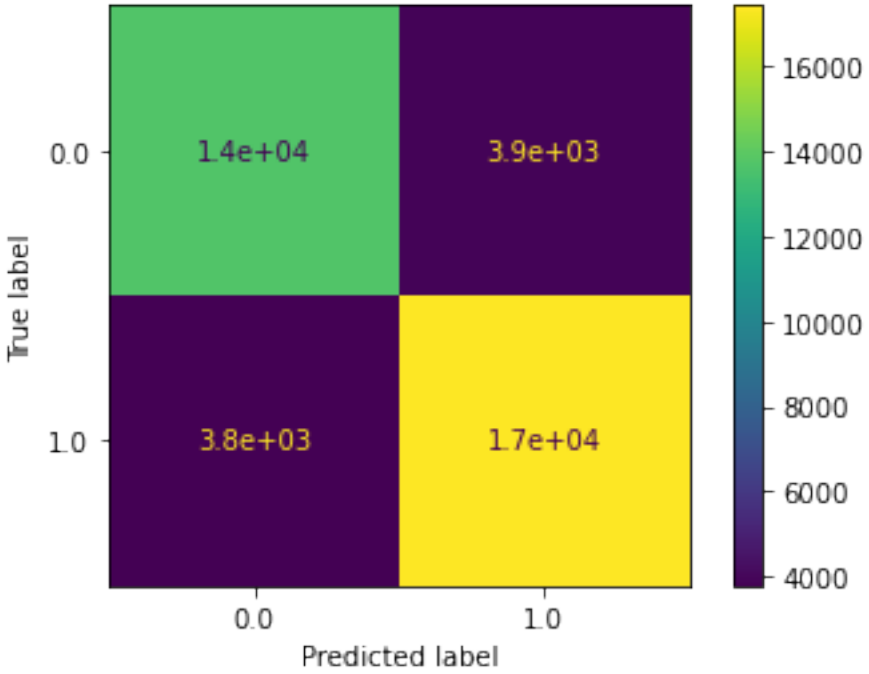
1.5.5 Produce a confusion matrix

Data professionals often like to know the types of errors made by an algorithm. To obtain this information, produce a confusion matrix.

[21]:
```
### YOUR CODE HERE ###

cm = metrics.confusion_matrix(y_test, y_pred, labels = clf.classes_)
disp = metrics.ConfusionMatrixDisplay(confusion_matrix = cm,display_labels =␣
↪clf.classes_)
disp.plot()
```

[21]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fd05ed7d690>



Question: What stands out to you about the confusion matrix?

Two of the quadrants are under 4,000, which are relatively low numbers. Based on what we know from the data and interpreting the matrix, it's clear that these numbers relate to false positives and false negatives.

Additionally, the other two quadrants—the true positives and true negatives—are both high numbers above 13,000.

Hint 1

Refer to the content about plotting a confusion matrix.

Question: Did you notice any difference in the number of false positives or false negatives that the model produced?

There isn't a large difference in the number of false positives and false negatives.

Question: What do you think could be done to improve model performance?

Using more than a single independent variable in the model training process could improve model performance. This is because other variables, like `Departure Delay in Minutes`, seem like they could potentially influence customer satisfaction.

1.6 Considerations

What are some key takeaways that you learned from this lab? * A lot of machine learning workflows are about cleaning, encoding, and scaling data. * The approach you use to plot or graph your data may depend on the type of variable you are evaluating. * Training a logistic regression model on a single independent variable can produce a relatively good model (80.2 percent accuracy).

What findings would you share with others? * Logistic regression accurately predicted satisfaction 80.2 percent of the time.
* The confusion matrix is useful, as it displays a similar amount of true positives and true negatives.

What would you recommend to stakeholders? * Customers who rated in-flight entertainment highly were more likely to be satisfied. Improving in-flight entertainment should lead to better customer satisfaction. * The model is 80.2 percent accurate. This is an improvement over the dataset's customer satisfaction rate of 54.7 percent. * The success of the model suggests that the airline should invest more in model developement to examine if adding more independent variables leads to better results. Building this model could not only be useful in predicting whether or not a customer would be satisfied but also lead to a better understanding of what independent variables lead to happier customers.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.