Activity_Perform logistic regression

November 16, 2023

1 Activity: 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.

[35]:
```
# Standard operational package imports.
import pandas as pd
import numpy as np
```

```
from sklearn.preprocessing import OneHotEncoder

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

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

1.2.2 Load the dataset

The dataset **Invistico_Airline.csv** is loaded. The resulting pandas DataFrame is saved as a variable named `df_original`. As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

[2]:
```
# RUN THIS CELL TO IMPORT YOUR DATA.

### 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 file name 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.

[4]:
```
df_original.head(10)
```

[4]:

	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.

[7]:
```
df_original.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 22 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   satisfaction                      129880 non-null  object
 1   Customer Type                     129880 non-null  object
 2   Age                               129880 non-null  int64
 3   Type of Travel                    129880 non-null  object
 4   Class                             129880 non-null  object
 5   Flight Distance                   129880 non-null  int64
 6   Seat comfort                      129880 non-null  int64
 7   Departure/Arrival time convenient 129880 non-null  int64
 8   Food and drink                    129880 non-null  int64
 9   Gate location                     129880 non-null  int64
 10  Inflight wifi service             129880 non-null  int64
 11  Inflight entertainment            129880 non-null  int64
 12  Online support                    129880 non-null  int64
 13  Ease of Online booking            129880 non-null  int64
 14  On-board service                  129880 non-null  int64
 15  Leg room service                  129880 non-null  int64
 16  Baggage handling                  129880 non-null  int64
 17  Checkin service                   129880 non-null  int64
 18  Cleanliness                       129880 non-null  int64
 19  Online boarding                   129880 non-null  int64
 20  Departure Delay in Minutes        129880 non-null  int64
 21  Arrival Delay in Minutes          129487 non-null  float64
dtypes: float64(1), int64(17), object(4)
memory usage: 21.8+ MB
```

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.

[13]:
```
# Total counts of satisfied and dissatisfied

df_original.value_counts('satisfaction')
```

[13]:
```
satisfaction
satisfied       71087
dissatisfied    58793
dtype: int64
```

[15]:
```
# percentages of satisfied and dissatisfied

df_original.satisfaction.value_counts(normalize=True)
```

[15]:
```
satisfied       0.547328
dissatisfied    0.452672
Name: satisfaction, dtype: float64
```

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 are 71087 satisfied and 58793 dissatisfied

Question: What percentage of customers were satisfied?

54.7% of customers were satisfied

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.

[22]:
```
df_original.isnull().sum()
```

[22]:
```
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?

Yes to ensure accuracy of this model specifically, but it is important information if the airlines had further questions about the data.

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`.

[37]:
```
df_subset = df_original.dropna(axis = 0).reset_index(drop = True)
```

[37]:
```
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           0
dtype: int64
```

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."

[27]:
```
#Converting inflight entertainment from an int to a float

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

[27]:
```
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            float64
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 `.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.

[41]:
```
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`.

[42]:
```
df_subset.head(10)
```

[42]:
```
   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.

[47]:
```
X = df_subset[["Inflight entertainment"]]
y = df_subset["satisfaction"]
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.30,␣
↪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?

There are many reasons as to why customers could be dissatisfied, I think there should be more variables involved than just one.

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.

[49]:
```
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.

[50]:
```
clf.coef_
```

[50]:
```
array([[0.99751462]])
```

[51]:
```
clf.intercept_
```

[51]:
```
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

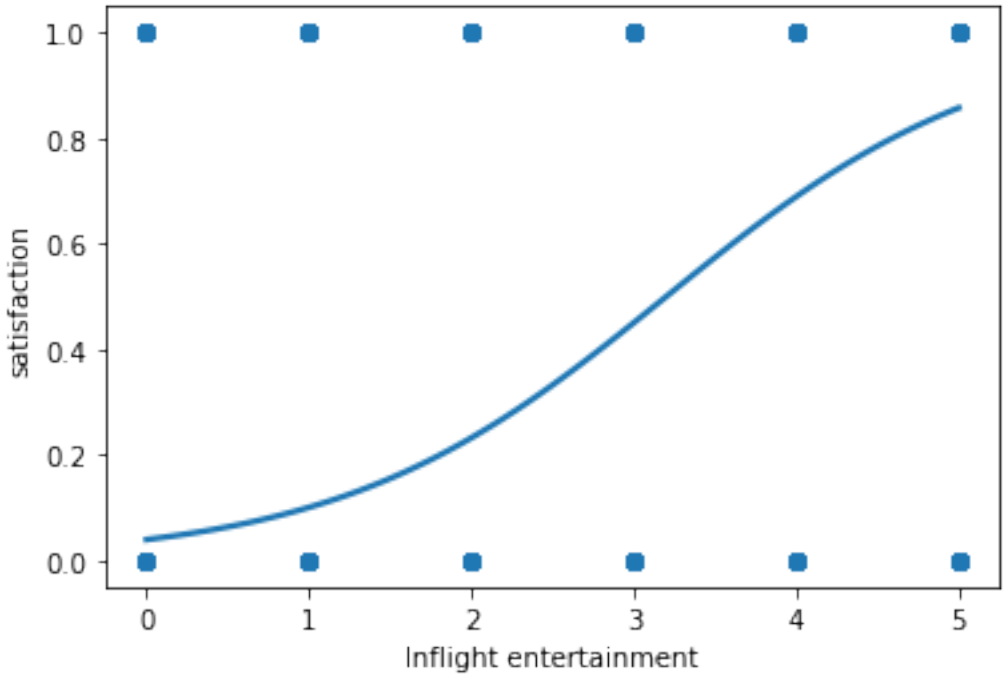
Use `.coef_` and `.intercept_`

1.4.3 Create a plot of your model

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

[54]:
```
sns.regplot(x="Inflight entertainment", y="satisfaction", data=df_subset,␣
↪logistic=True, ci=None)
```

[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ab4402d50>
```



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 is showing that higher inflight entertainment increases customer satisfaction. But this doesn't show much information about what specifcally makes the customer satisfaction increase.

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`.

[55]:
```
# 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`.

[56]:
```
print(y_pred)
```

```
[1. 0. 0. … 0. 0. 0.]
```

1.5.3 Use the predict_proba and predict functions on X_test

[57]:
```
# Use predict_proba to output a probability.

clf.predict_proba(X_test)
```

[57]:
```
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.

[58]:
```
# Use predict to output 0's and 1's.

clf.predict(X_test)
```

[58]:
```
array([1., 0., 0., …, 0., 0., 0.])
```

Hint 2

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

1.5.4 Analyze the results

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

[60]:
```
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.

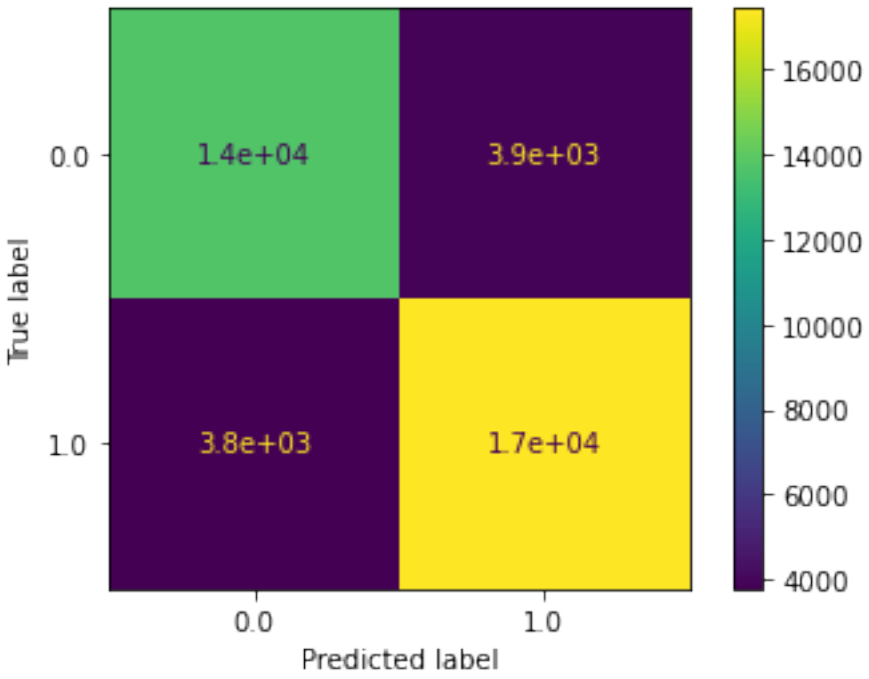
1.5.5 Produce a confusion matrix

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

[63]:
```
# Making confusion matrix

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

[63]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f8ab48172d0>
```



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

From the confusion matrix it seems this model predicts well. Less than 4000 incorrect predictions and over 14000 for correct predictions.

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?

Both are nearly the same.

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

Including more variables as to why customers may or may not be dissatisfied.

1.6 Considerations

What are some key takeaways that you learned from this lab?

Machine learning can be used in many ways to help us understand data better and see it from new perspectives with real accuracy

What findings would you share with others?

This model is a good fit for following customer satisfaction ratings, with predictions at 80% with more variables involved in the future it could become even better.

What would you recommend to stakeholders?

Most customers are satisfied with the airline and they should continue on this trend of implementing inflight entertainment to keep customers happy and giving them more of what makes them satisfied.

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.