In [342…

```python
# 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
```

In [348…

```python
df_original = pd.read_csv(r'C:\Users\HP\Desktop\Advance Data Analyst\5. Simplify Complex Data Relationships\5. Module 5\3. Interpret Logistic Regression\Files\Invistico_Airline.csv')
df_original.head(10)
```

Out[348…

	satisfaction	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	...	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Online boarding	Departure Delay in Minutes	Arrival Delay in Minutes
0	satisfied	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	2	...	2	3	3	0	3	5	3	2	0	0.0
1	satisfied	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	3	...	2	3	4	4	4	2	3	2	310	305.0
2	satisfied	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	3	...	2	2	3	3	4	4	4	2	0	0.0
3	satisfied	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	3	...	3	1	1	0	1	4	1	3	0	0.0
4	satisfied	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	3	...	4	2	2	0	2	4	2	5	0	0.0
5	satisfied	Loyal Customer	30	Personal Travel	Eco	1894	0	0	0	3	...	2	2	5	4	5	5	4	2	0	0.0
6	satisfied	Loyal Customer	66	Personal Travel	Eco	227	0	0	0	3	...	5	5	5	0	5	5	5	3	17	15.0
7	satisfied	Loyal Customer	10	Personal Travel	Eco	1812	0	0	0	3	...	2	2	3	3	4	5	4	2	0	0.0
8	satisfied	Loyal Customer	56	Personal Travel	Business	73	0	0	0	3	...	5	4	4	0	1	5	4	4	0	0.0
9	satisfied	Loyal Customer	22	Personal Travel	Eco	1556	0	0	0	3	...	2	2	2	4	5	3	4	2	30	26.0

10 rows × 22 columns

In [350…

```python
# Explore the data
df_original.dtypes
```

Out[350…

```
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
```

In [352…

```python
# Check the number of satisfied customers in the dataset
df_original['satisfaction'].value_counts(dropna = False)
```

Out[352…

```
satisfaction
satisfied       71087
dissatisfied    58793
Name: count, dtype: int64
```

In [354…

```python
# Check for missing values
df_original.isnull().sum()
```

Out[354…

```
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
```

In [356…

```python
# Drop the rows with missing values
df_subset = df_original.dropna(axis=0).reset_index(drop = True)
```

In [358…

```python
# Prepare the data
df_subset = df_subset.astype({"Inflight entertainment": float})
```

In [360…

```python
# Convert the categorical column satisfaction into numeric
df_subset['satisfaction'] = OneHotEncoder(drop='first').fit_transform(df_subset[['satisfaction']]).toarray()
```

In [362…

```python
# Output the data
df_subset.head(10)
```

Out[362…

	satisfaction	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	...	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Online boarding	Departure Delay in Minutes	Arrival Delay in Minutes
0	1.0	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	2	...	2	3	3	0	3	5	3	2	0	0.0
1	1.0	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	3	...	2	3	4	4	4	2	3	2	310	305.0
2	1.0	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	3	...	2	2	3	3	4	4	4	2	0	0.0
3	1.0	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	3	...	3	1	1	0	1	4	1	3	0	0.0
4	1.0	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	3	...	4	2	2	0	2	4	2	5	0	0.0
5	1.0	Loyal Customer	30	Personal Travel	Eco	1894	0	0	0	3	...	2	2	5	4	5	5	4	2	0	0.0
6	1.0	Loyal Customer	66	Personal Travel	Eco	227	0	0	0	3	...	5	5	5	0	5	5	5	3	17	15.0
7	1.0	Loyal Customer	10	Personal Travel	Eco	1812	0	0	0	3	...	2	2	3	3	4	5	4	2	0	0.0
8	1.0	Loyal Customer	56	Personal Travel	Business	73	0	0	0	3	...	5	4	4	0	1	5	4	4	0	0.0
9	1.0	Loyal Customer	22	Personal Travel	Eco	1556	0	0	0	3	...	2	2	2	4	5	3	4	2	30	26.0

10 rows × 22 columns

In [364…

```python
# Create the training and testing data
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)
```

In [366…

```python
# Fit a LogisticRegression model to the data
clf = LogisticRegression().fit(X_train,y_train)
```

In [374…

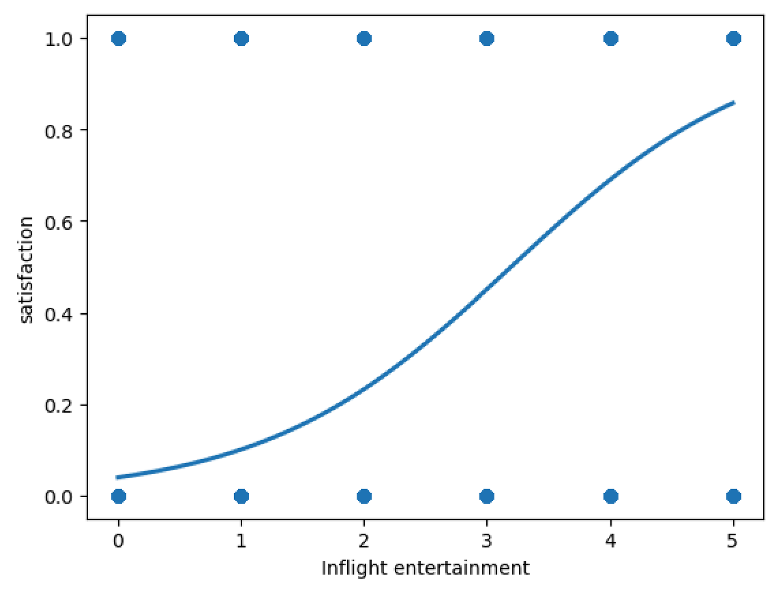
```python
# Obtain parameter estimates
print(clf.coef_)
print(clf.intercept_)
```

```
[[0.99752883]]
[-3.19359054]
```

In [376…

```python
# Create a plot of your model
sns.regplot(x="Inflight entertainment", y="satisfaction", data=df_subset, logistic=True, ci=None)
```

Out[376… <Axes: xlabel='Inflight entertainment', ylabel='satisfaction'>



In [380…

```python
# Predict the outcome for the test dataset
y_pred = clf.predict(X_test)
print(y_pred)
```

```
[1. 0. 0. ... 0. 0. 0.]
```

In [382…

```python
# Use the predict_proba and predict functions on X_test
# Use predict_proba to output a probability.

clf.predict_proba(X_test)
```

Out[382…

```
array([[0.14257646, 0.85742354],
       [0.55008251, 0.44991749],
       [0.89989529, 0.10010471],
       ...,
       [0.89989529, 0.10010471],
       [0.76826369, 0.23173631],
       [0.55008251, 0.44991749]])
```

In [384…

```python
# Use predict to output 0's and 1's.

clf.predict(X_test)
```

Out[384…

```
array([1., 0., 0., ..., 0., 0., 0.])
```

In [388…

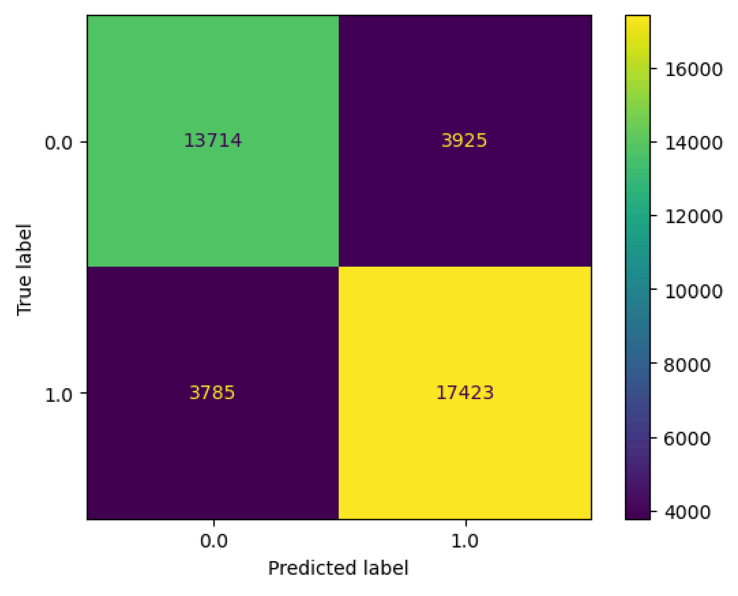
```python
# Analyze the results
print("Accuracy:", "%.6f" % metrics.accuracy_score(y_test, y_pred))
print("Precision:", "%.6f" % metrics.precision_score(y_test, y_pred))
print("Recall:", "%.6f" % metrics.recall_score(y_test, y_pred))
print("F1 Score:", "%.6f" % metrics.f1_score(y_test, y_pred))
```

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

In [390…

```python
# Produce a 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()
```

Out[390… <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x27bf9a6cb90>



In [ ]:

```python
# some key takeaways

# 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.
```