Vi Le_Project 2

December 3, 2024

1 Project 2 - Airline Customer Satisfaction

1.1 Vi Le | DAT 402: Machine Learning- Data Science, Fall 2024

Customer satisfaction is a cornerstone of success for businesses in industries such as retail, telecommunications, and airlines. Satisfied customers are more likely to become loyal advocates, reflecting a brand's quality and trustworthiness.

With over four years of experience in retail and customer service, I have seen firsthand how critical it is to meet and exceed customer expectations. Inspired by this, I chose to analyze and predict customer satisfaction levels in the airline industry for this project. By leveraging data insights, my goal is to better understand the factors that drive satisfaction and loyalty, as well as to build a model that could classify happy and unhappy customers based on different factors.

Credits: - The following data set is from TJ Klein (link: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction) - The data set provided is modified from John D. (link: https://www.kaggle.com/datasets/johnddddddd/customer-satisfaction)

In this project, I will use 3 different classification methods on the same data set, and compare their peformances: - Naive Bayes - Logistic Regression - Decision Tree

First thing first, I will import all necessary libraries and packages.

[9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
from sklearn.naive_bayes import MultinomialNB, CategoricalNB, GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score
```

Then, I will load the data set, and see a first few rows of it.

[11]:
```
df = pd.read_csv("airline.csv")
df = pd.DataFrame(df)
df.head()
```

[11]:
```
   Unnamed: 0     id  Gender      Customer Type  Age   Type of Travel  \
0           0  70172    Male     Loyal Customer   13  Personal Travel
```

```
1           1    5047    Male  disloyal Customer   25  Business travel
2           2  110028  Female     Loyal Customer   26  Business travel
3           3   24026  Female     Loyal Customer   25  Business travel
4           4  119299    Male     Loyal Customer   61  Business travel

      Class  Flight Distance  Inflight wifi service  \
0  Eco Plus              460                      3
1  Business              235                      3
2  Business             1142                      2
3  Business              562                      2
4  Business              214                      3

   Departure/Arrival time convenient  …  Inflight entertainment  \
0                                  4  …                       5
1                                  2  …                       1
2                                  2  …                       5
3                                  5  …                       2
4                                  3  …                       3

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

   Inflight service  Cleanliness  Departure Delay in Minutes  \
0                 5            5                          25
1                 4            1                           1
2                 4            5                           0
3                 4            2                          11
4                 3            3                           0

   Arrival Delay in Minutes             satisfaction
0                      18.0  neutral or dissatisfied
1                       6.0  neutral or dissatisfied
2                       0.0                satisfied
3                       9.0  neutral or dissatisfied
4                       0.0                satisfied

[5 rows x 25 columns]
```

[12]:
```
print(df.dtypes)
```

```
Unnamed: 0                            int64
id                                    int64
Gender                               object
Customer Type                        object
```

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

Explanation of column names: - Gender: Gender of the passengers (Female, Male) - Customer Type: The customer type (Loyal customer, disloyal customer) - Age: The actual age of the passengers - Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel) - Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus) - Flight distance: The flight distance of this journey - Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5) - Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient - Ease of Online booking: Satisfaction level of online booking - Gate location: Satisfaction level of Gate location - Food and drink: Satisfaction level of Food and drink - Online boarding: Satisfaction level of online boarding - Seat comfort: Satisfaction level of Seat comfort - Inflight entertainment: Satisfaction level of inflight entertainment - On-board service: Satisfaction level of On-board service - Leg room service: Satisfaction level of Leg room service - Baggage handling: Satisfaction level of baggage handling - Check-in service: Satisfaction level of Check-in service - Inflight service: Satisfaction level of inflight service - Cleanliness: Satisfaction level of Cleanliness - Departure Delay in Minutes: Minutes delayed when departure - Arrival Delay in Minutes: Minutes delayed when Arrival - Satisfaction: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

1.1.1 Step 2: Data Preprocessing

[19]:
```python
# Check for NA values
print(df.isna().sum())
```

```
Unnamed: 0                           0
id                                   0
```

```
Gender                                 0
Customer Type                          0
Age                                    0
Type of Travel                         0
Class                                  0
Flight Distance                        0
Inflight wifi service                  0
Departure/Arrival time convenient      0
Ease of Online booking                 0
Gate location                          0
Food and drink                         0
Online boarding                        0
Seat comfort                           0
Inflight entertainment                 0
On-board service                       0
Leg room service                       0
Baggage handling                       0
Checkin service                        0
Inflight service                       0
Cleanliness                            0
Departure Delay in Minutes             0
Arrival Delay in Minutes             310
satisfaction                           0
dtype: int64
```

[21]:
```
df = df.dropna()
```

[23]:
```
# Check statistical aspect of data
df.describe()
```

[23]:
```
          Unnamed: 0             id            Age  Flight Distance  \
count  103594.000000  103594.000000  103594.000000    103594.000000
mean    51950.102274   64942.428625      39.380466      1189.325202
std     29997.914016   37460.816597      15.113125       997.297235
min         0.000000       1.000000       7.000000        31.000000
25%     25960.250000   32562.250000      27.000000       414.000000
50%     51955.500000   64890.000000      40.000000       842.000000
75%     77924.750000   97370.500000      51.000000      1743.000000
max    103903.000000  129880.000000      85.000000      4983.000000

       Inflight wifi service  Departure/Arrival time convenient  \
count          103594.000000                      103594.000000
mean                2.729753                           3.060081
std                 1.327866                           1.525233
min                 0.000000                           0.000000
25%                 2.000000                           2.000000
50%                 3.000000                           3.000000
```

```
75%               4.000000                                  4.000000
max               5.000000                                  5.000000

       Ease of Online booking  Gate location  Food and drink  Online boarding  \
count           103594.000000  103594.000000   103594.000000    103594.000000
mean                 2.756984       2.977026        3.202126         3.250497
std                  1.398934       1.277723        1.329401         1.349433
min                  0.000000       0.000000        0.000000         0.000000
25%                  2.000000       2.000000        2.000000         2.000000
50%                  3.000000       3.000000        3.000000         3.000000
75%                  4.000000       4.000000        4.000000         4.000000
max                  5.000000       5.000000        5.000000         5.000000

        Seat comfort  Inflight entertainment  On-board service  \
count  103594.000000           103594.000000     103594.000000
mean        3.439765                3.358341          3.382609
std         1.318896                1.333030          1.288284
min         0.000000                0.000000          0.000000
25%         2.000000                2.000000          2.000000
50%         4.000000                4.000000          4.000000
75%         5.000000                4.000000          4.000000
max         5.000000                5.000000          5.000000

       Leg room service  Baggage handling  Checkin service  Inflight service  \
count     103594.000000     103594.000000    103594.000000     103594.000000
mean           3.351401          3.631687         3.304323          3.640761
std            1.315409          1.181051         1.265396          1.175603
min            0.000000          1.000000         0.000000          0.000000
25%            2.000000          3.000000         3.000000          3.000000
50%            4.000000          4.000000         3.000000          4.000000
75%            4.000000          5.000000         4.000000          5.000000
max            5.000000          5.000000         5.000000          5.000000

         Cleanliness  Departure Delay in Minutes  Arrival Delay in Minutes
count  103594.000000               103594.000000             103594.000000
mean        3.286397                   14.747939                 15.178678
std         1.312194                   38.116737                 38.698682
min         0.000000                    0.000000                  0.000000
25%         2.000000                    0.000000                  0.000000
50%         3.000000                    0.000000                  0.000000
75%         4.000000                   12.000000                 13.000000
max         5.000000                 1592.000000               1584.000000
```

"Customer Type", "Type of Travle", "Class", "Gender", and "satisfaction" are categorical types and contain text values. For best use of data for machine learning models, I will turn those values into numerical type using **OrdinalEnconder()**.

[28]:
```python
from sklearn.preprocessing import OrdinalEncoder
ordinal = OrdinalEncoder()
df['Customer Type'] = ordinal.fit_transform(df[['Customer Type']])
df['Type of Travel'] = ordinal.fit_transform(df[['Type of Travel']])
df['Class'] = ordinal.fit_transform(df[['Class']])
df['satisfaction'] = ordinal.fit_transform(df[['satisfaction']])
df['Gender'] = ordinal.fit_transform(df[['Gender']])
df.head()
```

[28]:
```
   Unnamed: 0      id  Gender  Customer Type  Age  Type of Travel  Class  \
0           0   70172     1.0            0.0   13             1.0    2.0
1           1    5047     1.0            1.0   25             0.0    0.0
2           2  110028     0.0            0.0   26             0.0    0.0
3           3   24026     0.0            0.0   25             0.0    0.0
4           4  119299     1.0            0.0   61             0.0    0.0

   Flight Distance  Inflight wifi service  Departure/Arrival time convenient  \
0              460                      3                                  4
1              235                      3                                  2
2             1142                      2                                  2
3              562                      2                                  5
4              214                      3                                  3

   …  Inflight entertainment  On-board service  Leg room service  \
0  …                       5                 4                 3
1  …                       1                 1                 5
2  …                       5                 4                 3
3  …                       2                 2                 5
4  …                       3                 3                 4

   Baggage handling  Checkin service  Inflight service  Cleanliness  \
0                 4                4                 5            5
1                 3                1                 4            1
2                 4                4                 4            5
3                 3                1                 4            2
4                 4                3                 3            3

   Departure Delay in Minutes  Arrival Delay in Minutes  satisfaction
0                          25                      18.0           0.0
1                           1                       6.0           0.0
2                           0                       0.0           1.0
3                          11                       9.0           0.0
4                           0                       0.0           1.0

[5 rows x 25 columns]
```

1.1.2 Step 3: Visualize data for insights

[32]:
```
corr_matrix = df.corr()
corr_matrix["satisfaction"].sort_values(ascending=False)
```

[32]:
```
satisfaction                         1.000000
Online boarding                      0.503447
Inflight entertainment               0.398203
Seat comfort                         0.349112
On-board service                     0.322450
Leg room service                     0.313182
Cleanliness                          0.305050
Flight Distance                      0.298915
Inflight wifi service                0.284163
Baggage handling                     0.247819
Inflight service                     0.244852
Checkin service                      0.235914
Food and drink                       0.209659
Ease of Online booking               0.171507
Age                                  0.137040
id                                   0.013680
Gender                               0.012356
Gate location                        0.000449
Unnamed: 0                          -0.004552
Departure Delay in Minutes          -0.050515
Departure/Arrival time convenient   -0.051718
Arrival Delay in Minutes            -0.057582
Customer Type                       -0.187558
Type of Travel                      -0.448995
Class                               -0.449466
Name: satisfaction, dtype: float64
```

[35]:
```
corr_matrix.style.background_gradient(cmap='coolwarm')
```

[35]:
```
<pandas.io.formats.style.Styler at 0x11bbcc770>
```

1.1.3 Step 3: Split data into training set and test set

[39]:
```
# Randomize the dataset
data_randomized = df.sample(frac=1, random_state=1)

# Calculate index for split - take first 80% of the data for test set
training_test_index = round(len(data_randomized) * 0.8)

# Split into training and test sets
training_set = data_randomized[:training_test_index].reset_index(drop=True)
```

```
test_set = data_randomized[training_test_index:].reset_index(drop=True)

print(training_set.shape)
print(test_set.shape)
```

```
(82875, 25)
(20719, 25)
```

Test to see if training set and test set proportions are approximately similar to original data set.

[42]:
```
df['satisfaction'].value_counts(normalize=True)
```

[42]:
```
satisfaction
0.0    0.566606
1.0    0.433394
Name: proportion, dtype: float64
```

[44]:
```
training_set['satisfaction'].value_counts(normalize=True)
```

[44]:
```
satisfaction
0.0    0.566287
1.0    0.433713
Name: proportion, dtype: float64
```

[46]:
```
test_set['satisfaction'].value_counts(normalize=True)
```

[46]:
```
satisfaction
0.0    0.567885
1.0    0.432115
Name: proportion, dtype: float64
```

1.1.4 Step 4: Apply machine learning methods and performance metrics

- **Naive Bayes**

[51]:
```
trainX = training_set.iloc[:,:-1]
trainy = training_set['satisfaction']

testX = test_set.iloc[:,:-1]

testy = test_set['satisfaction']
NB_model = GaussianNB()
NB_model.fit(trainX, trainy)
y_pred = NB_model.predict(testX)

NB_accuracy = accuracy_score(testy, y_pred)
print(NB_accuracy)
```

0.8041893913798929

[53]:
```
confusion_matrix(testy, y_pred)
```

[53]:
```
array([[9564, 2202],
       [1855, 7098]])
```

Even though the model has pretty good accuracy score, it seems the model is struggle with Fasle Negative.

- **Logistic Regression**

[59]:
```
from sklearn.linear_model import LogisticRegression
log_reg_model = LogisticRegression(random_state=42)
log_reg_model.fit(trainX, trainy)
y_pred = log_reg_model.predict(testX)
log_reg_accuracy = accuracy_score(testy, y_pred)
print(log_reg_accuracy)
```

```
0.6843959650562286
```

```
/opt/anaconda3/lib/python3.12/site-
packages/sklearn/linear_model/_logistic.py:469: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

- **Decision Tree**

[61]:
```
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier(random_state=42, max_depth=5)
dt_model.fit(trainX, trainy)
y_pred = dt_model.predict(testX)
dt_accuracy = accuracy_score(testy, y_pred)
print(dt_accuracy)
```

```
0.90873111636662
```

It looks like I could improve the accuracy score by remove some less important feature columns. Based on correlation coefficients, I will drop the least correlated columns to “satisfaction” column: “id”, “Gender”, “Unnamed: 0”, “Gate location”, “Departure Delay in Minutes”, “Departure/Arrival time convenient”, “Arrivael Delay in Minutes”

[67]:
```
training_set = training_set.drop(columns=['id', 'Unnamed: 0', 'Gender', 'Gate␣
↪location', 'Departure Delay in Minutes',
```

```
                     'Departure/Arrival time convenient', 'Arrival Delay in␣
 ↪Minutes'])
```

[69]:
```
test_set = test_set.drop(columns=['id', 'Unnamed: 0', 'Gender', 'Gate␣
 ↪location', 'Departure Delay in Minutes',
                     'Departure/Arrival time convenient', 'Arrival Delay in␣
 ↪Minutes'])
```

I will peform all three models again to see if the accuracy score improves.

[84]:
```
trainX = training_set.iloc[:,:-1]
trainy = training_set['satisfaction']
testX = test_set.iloc[:,:-1]
testy = test_set['satisfaction']
NB_model = GaussianNB()
NB_model.fit(trainX, trainy)
NB_y_pred = NB_model.predict(testX)
after_NB_accuracy = accuracy_score(testy, NB_y_pred)

log_reg_model = LogisticRegression(random_state=42)
log_reg_model.fit(trainX, trainy)
log_y_pred = log_reg_model.predict(testX)
after_log_reg_accuracy = accuracy_score(testy, log_y_pred)

dt_model = DecisionTreeClassifier(random_state=42, max_depth=5)
dt_model.fit(trainX, trainy)
dt_y_pred = dt_model.predict(testX)
after_dt_accuracy = accuracy_score(testy, dt_y_pred)

NB_cf = confusion_matrix(testy, NB_y_pred)
log_cf = confusion_matrix(testy, log_y_pred)
dt_cf = confusion_matrix(testy, dt_y_pred)
```

```
/opt/anaconda3/lib/python3.12/site-
packages/sklearn/linear_model/_logistic.py:469: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[73]:
```
print("Naive Bayes accuracy score:", NB_accuracy,
     "\nLogistic Regression accuracy score", log_reg_accuracy,
     "\nDecision Tree accuracy score", dt_accuracy,
```

```
      "\n","\nNaive Bayes accuracy score AFTER:", after_NB_accuracy,
     "\nLogistics Regression accuracy score AFTER:", after_log_reg_accuracy,
     "\nDecision Tree accuracy score AFTER:", after_dt_accuracy)
```

```
Naive Bayes accuracy score: 0.8041893913798929
Logistic Regression accuracy score 0.6843959650562286
Decision Tree accuracy score 0.90873111636662

Naive Bayes accuracy score AFTER: 0.8651479318499927
Logistics Regression accuracy score AFTER: 0.8375886867126792
Decision Tree accuracy score AFTER: 0.908007143201892
```

[111]:
```
NB_cf
```

[111]:
```
array([[10672,  1094],
       [ 1700,  7253]])
```

[92]:
```
log_cf
```

[92]:
```
array([[10099,  1667],
       [ 1698,  7255]])
```

[98]:
```
dt_cf
```

[98]:
```
array([[10905,   861],
       [ 1045,  7908]])
```

Whether before or after remove less important feature variables, Decision Tree still performs the best out of 3 models. To further improve the performance of Decision Tree model, I will perform bagging to have a better performance for test set.

[79]:
```
from sklearn.ensemble import BaggingClassifier
bagg = BaggingClassifier(DecisionTreeClassifier(), n_estimators=500,␣
 ↪max_samples=100, n_jobs= -1, random_state=42)
bagg.fit(trainX, trainy)
y_pred = bagg.predict(testX)
accuracy_score(testy, y_pred)
```

[79]:
```
0.9171292050774651
```

1.1.5 Conclusion

- High false negative or false positive in this case (airline industry) would not be as harmful as in medical field, where it could be dangerous and fatal. After removing less important features, the false positive is now higher than the false negative, even though the accuracy score is prettty high.

–> This could still greatly impact the airline business in many way: not addressing the dissatifaction of the customer in time could lead to the loss of loyalty, bad public reputation as customer might file complaint or address the issue to social media when they feel ignored.

- Specifically, Naives Bayes model changes from false negative > false positive to false poitive > false negative could indicate that even though the less important (less correlated) features to "Satisfaction level" are removed, the model might lack some context to distinguish between satisfy and dissatisfy classes.