

travel-churn-analysis

October 30, 2024

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

```
[ ]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
[ ]: data = '/content/drive/MyDrive/Customertavel.csv'
df = pd.read_csv(data)
```

```
[ ]: df.head()
```

```
[ ]: 
```

	Age	FrequentFlyer	AnnualIncomeClass	ServicesOpted	\
0	34	No	Middle Income	6	
1	34	Yes	Low Income	5	
2	37	No	Middle Income	3	
3	30	No	Middle Income	2	
4	30	No	Low Income	1	

	AccountSyncedToSocialMedia	BookedHotelOrNot	Target
0	No	Yes	0
1	Yes	No	1
2	Yes	No	0
3	No	No	0
4	No	No	0

```
[ ]: # Checking for missing values.
df.isnull().sum()
```

```
[ ]: Age
FrequentFlyer
AnnualIncomeClass
```

0
0
0

```

ServicesOpted          0
AccountSyncedToSocialMedia  0
BookedHotelOrNot       0
Target                 0
dtype: int64

```

```
[ ]: # Duplicate value in the data
df.duplicated().sum()
```

```
[ ]: 507
```

```
[ ]: # Dimensions of data
df.shape
```

```
[ ]: (954, 7)
```

```
[ ]: # Basic Info of the dataset
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 954 entries, 0 to 953
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Age                   954 non-null   int64
 1   FrequentFlyer         954 non-null   object
 2   AnnualIncomeClass     954 non-null   object
 3   ServicesOpted         954 non-null   int64
 4   AccountSyncedToSocialMedia 954 non-null   object
 5   BookedHotelOrNot      954 non-null   object
 6   Target                954 non-null   int64
dtypes: int64(3), object(4)
memory usage: 52.3+ KB

```

```
[ ]: # Description of Numerical columns.
df.describe()
```

```

[ ]:
count      Age  ServicesOpted      Target
mean    32.109015      2.437107    0.234801
std      3.337388      1.606233    0.424097
min     27.000000      1.000000    0.000000
25%     30.000000      1.000000    0.000000
50%     31.000000      2.000000    0.000000
75%     35.000000      4.000000    0.000000
max     38.000000      6.000000    1.000000

```

```
[ ]: # Column names
df.columns
```

```
[ ]: Index(['Age', 'FrequentFlyer', 'AnnualIncomeClass', 'ServicesOpted',
          'AccountSyncedToSocialMedia', 'BookedHotelOrNot', 'Target'],
          dtype='object')
```

```
[ ]: ''' Visulazing some insights'''
# AnnualIncomeClass

fig = px.histogram(df,x='AnnualIncomeClass',color='Target')
fig.show()
```

```
[ ]: '''Meaningful insights of the above plot'''

# Middle Income people travel more than High Income, there will be may reason
↳behind it like they don't have enough time to travel they are busy in their
↳work to earn more money
# or anything else also happens.

# Note : Company need to focus on how they can attract High Income family or
↳people to their travelling plan and also attract Low Income people by
↳providing jaw dropping deals or offers.
```

```
[ ]: fig = px.histogram(df,x='AnnualIncomeClass',color='FrequentFlyer')
fig.show()
```

```
[ ]: '''Meaningful insights of the above plot'''

# As per the data Middle Income class are not a FrequentFlyer
```

```
[ ]: df['ServicesOpted'].value_counts()
```

```
[ ]: ServicesOpted
1      404
2      176
3      124
4      117
5       69
6       64
Name: count, dtype: int64
```

```
[ ]: fig = px.histogram(df,x='AnnualIncomeClass',color='ServicesOpted')
fig.show()
```

```
[ ]: fig = px.histogram(df,x='AnnualIncomeClass',color='BookedHotelOrNot')
fig.show()
```

```
[ ]: '''Meaningful insights of the above plot'''

# Mainly Low Income Class people not booking hotel while travelling
# And very low amount of High Income Clss booked hotels
```

```
[ ]: '''Data Preprocessing'''
'''Encoding Categorical Columns'''

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
cat_col = df.select_dtypes(include=['object']).columns
for i in cat_col:
    df[i] = le.fit_transform(df[i])
```

```
[ ]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 954 entries, 0 to 953
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age                                   954 non-null    int64
1   FrequentFlyer                        954 non-null    int64
2   AnnualIncomeClass                   954 non-null    int64
3   ServicesOpted                       954 non-null    int64
4   AccountSyncedToSocialMedia          954 non-null    int64
5   BookedHotelOrNot                    954 non-null    int64
6   Target                               954 non-null    int64
dtypes: int64(7)
memory usage: 52.3 KB
```

```
[ ]: '''Standardizing the values'''
# Before standardizing first spilt the data
```

```
[ ]: from sklearn.model_selection import train_test_split

X = df.drop(columns=['Target'])
y = df['Target']

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.
↪3,random_state=42)
```

```
[ ]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

```
[ ]: X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.fit_transform(X_test)
```

```
[ ]: '''Fitting ML Algorithm'''  
  
from sklearn.linear_model import LogisticRegression  
lr = LogisticRegression()  
from sklearn.metrics import accuracy_score
```

```
[ ]: lr.fit(X_train_scaled,y_train)
```

```
[ ]: LogisticRegression()
```

```
[ ]: y_pred = lr.predict(X_test_scaled)
```

```
[ ]: accuracy_score(y_test,y_pred)
```

```
[ ]: 0.8432055749128919
```