The Analysis of Tour and Travel Compnay

Problem Statement

A tour & travels company wants to predict whether a customer will churn or not. Based on a few customer characteristics like their age, frequent flyer status, annual income class, services opted, account snick to social media, booked hotel or not, Target.

The analysis and forecasting are based on the customer churn's impact on yearly income, hotel reservations, and whether or not they were made in order to assist the business in developing predictive models, saving money, and performing fascinating EDAs.

Assumptions

1. No unusual occurrences will have a substantial impact on the data we used (Outliers).

2. The data we used is current data and can be used to analyse and make prediction for the organization.

3. There are no unanticipated negatives to the company employing any advised technique.

Research Questions

1. How Hotel Booking and Annual Income effect customer churn?

2. How can we prevent customer churn?

Importing Libraries

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loading the Dataset

In [3]:
```
data = pd.read_csv('customer churn data.csv')
```

Exploratory Data Analysis and Data Cleaning

In [4]:
```
data.head()
```

Out[4]:

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

In [5]:
```
data.tail()
```

Out[5]:

	Age	FrequentFlyer	AnnualIncomeClass	ServicesOpted	AccountSyncedToSocialMedia	BookedHotelOrNot	Target
949	31	Yes	Low Income	1	No	No	0
950	30	No	Middle Income	5	No	Yes	0
951	37	No	Middle Income	4	No	No	0
952	30	No	Low Income	1	Yes	Yes	0
953	31	Yes	High Income	1	No	No	0

In [6]:
```
#Data reading, checking dimensions and information of the data
print(data)

print('dimensions:')
print(data.shape)

print('Information:')
data.info()
```

```
     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
..   ...           ...                 ...            ...
949   31           Yes         Low Income              1
950   30            No      Middle Income              5
951   37            No      Middle Income              4
952   30            No         Low Income              1
953   31           Yes        High 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
..                         ...              ...     ...
949                         No               No       0
950                         No              Yes       0
951                         No               No       0
952                        Yes              Yes       0
953                         No               No       0

[954 rows x 7 columns]
dimensions:
(954, 7)
Information:
<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
```

In [7]:
```
#Cheking for duplicates and missing
data.isnull().values.any()
value = len(data[data.duplicated()])
print(value)
```

```
507
```

In [8]:
```
#Statistical summary
print("Statistical Summary")
data.describe().T
```

```
Statistical Summary
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
Age	954.0	32.109015	3.337388	27.0	30.0	31.0	35.0	38.0
ServicesOpted	954.0	2.437107	1.606233	1.0	1.0	2.0	4.0	6.0
Target	954.0	0.234801	0.424097	0.0	0.0	0.0	0.0	1.0

In [9]:
```
data.describe(include = 'object')
```

Out[9]:

	FrequentFlyer	AnnualIncomeClass	AccountSyncedToSocialMedia	BookedHotelOrNot
count	954	954	954	954
unique	3	3	2	2
top	No	Middle Income	No	No
freq	608	409	594	576

In [10]:
```
for col in data.describe(include = 'object') .columns:
    print(col)
    print(data[col].unique())
    print('-'*50)
```

```
FrequentFlyer
['No' 'Yes' 'No Record']
--------------------------------------------------
AnnualIncomeClass
['Middle Income' 'Low Income' 'High Income']
--------------------------------------------------
AccountSyncedToSocialMedia
['No' 'Yes']
--------------------------------------------------
BookedHotelOrNot
['Yes' 'No']
--------------------------------------------------
```

In [11]:
```
data.isnull().sum()
```

Out[11]:

```
Age                           0
FrequentFlyer                 0
AnnualIncomeClass             0
ServicesOpted                 0
AccountSyncedToSocialMedia    0
BookedHotelOrNot              0
Target                        0
dtype: int64
```

Data Analysis and Visualizations

In [12]:
```
from sklearn.preprocessing import LabelEncoder
number = LabelEncoder()
features = ['FrequentFlyer','AnnualIncomeClass','AccountSyncedToSocialMedia','BookedHotelOrNot']
for f in features:
    data[f] = number.fit_transform(data[f])
data.head()
```

Out[12]:

	Age	FrequentFlyer	AnnualIncomeClass	ServicesOpted	AccountSyncedToSocialMedia	BookedHotelOrNot	Target
0	34	0	2	6	0	1	0
1	34	2	1	5	1	0	1
2	37	0	2	3	1	0	0
3	30	0	2	2	0	0	0
4	30	0	1	1	0	0	0

Hotel Booking Visualization

In [21]:
```
churn_perc = data ['BookedHotelOrNot'].value_counts(normalize = 1)
print(churn_perc)

plt.figure(figsize=(5,6))
plt.bar(["Booked","NotBooked"], data ['BookedHotelOrNot'].value_counts(normalize = 1), edgecolor = 'black', width = 0.8, color = ['skyblue', 'purple'])
plt.title('How Hotel Bookings Affect Customer Churn')
plt.show()
```

```
0    0.603774
1    0.396226
Name: BookedHotelOrNot, dtype: float64
```

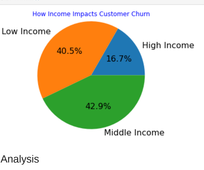
Annual Income Visualization

In [13]:
```
income = data.groupby("AnnualIncomeClass")
income = income.size()
income
```

Out[13]:

```
AnnualIncomeClass
0    159
1    386
2    409
dtype: int64
```

In [14]:
```
plt.pie(income.values , labels = ("High Income", "Low Income", "Middle Income" ), autopct='%1.1f%%', radius = 1.2, textprops = {"fontsize" : 16})
plt.title("How Income Impacts Customer Churn", c="b")
plt.show()
```



Analysis

Customers who have made hotel reservations account for 60% of the overall population, whereas those who have not made hotel reservations account for 39% of the population, which is a significant number.

We found that customers with the highest yearly income have the lowest proportion of customer churn, whereas those with the lowest (40%) and middle (42%), have the highest percentage of churn.

Insights

- The largest customer churn has been observed when a consumer has not made a hotel reservation. This may be due to a number of factors, including expensive hotel rates and longer waiting lists for reservations, both of which have a substantial impact on why people don't book hotels. This may help us understand how to run the campaign in order to shorten the waiting list.

- The highest customer churn have also seen in the situation where customer have low and middle income. This may give us an idea of the needs of the customers, and you can utilise the insights to work on how to lower your pricing without affecting your revenue, which you can then offer to your customers, which can prevent the customer churn.

In [ ]:
```

```