In [2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [3]:
```python
#importing dataset file
df= pd.read_csv('train.csv')
```

In [10]:
```python
#rows and coloumn present
df.shape
```

Out[10]: (103594, 25)

In [5]:
```python
# top 2 rows and coloumns
df.head(2)
```

Out[5]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cle
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	4	5	
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...	1	1	5	3	1	4	

2 rows × 25 columns

In [21]:
```python
#last 2 rows and coloumns
df.tail(2)
```

Out[21]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service
103902	103902	54173	Female	disloyal Customer	22	Business travel	Eco	1000	1	1	...	1	4	5	1	5	4
103903	103903	62567	Male	Loyal Customer	27	Business travel	Business	1723	1	3	...	1	1	1	4	4	3

2 rows × 25 columns

In [11]:
```python
#datatypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 103594 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   Unnamed: 0                         103594 non-null  int64  
 1   id                                 103594 non-null  int64  
 2   Gender                             103594 non-null  object 
 3   Customer Type                      103594 non-null  object 
 4   Age                                103594 non-null  int64  
 5   Type of Travel                     103594 non-null  object 
 6   Class                              103594 non-null  object 
 7   Flight Distance                    103594 non-null  int64  
 8   Inflight wifi service              103594 non-null  int64  
 9   Departure/Arrival time convenient  103594 non-null  int64  
 10  Ease of Online booking             103594 non-null  int64  
 11  Gate location                      103594 non-null  int64  
 12  Food and drink                     103594 non-null  int64  
 13  Online boarding                    103594 non-null  int64  
 14  Seat comfort                       103594 non-null  int64  
 15  Inflight entertainment             103594 non-null  int64  
 16  On-board service                   103594 non-null  int64  
 17  Leg room service                   103594 non-null  int64  
 18  Baggage handling                   103594 non-null  int64  
 19  Checkin service                    103594 non-null  int64  
 20  Inflight service                   103594 non-null  int64  
 21  Cleanliness                        103594 non-null  int64  
 22  Departure Delay in Minutes         103594 non-null  int64  
 23  Arrival Delay in Minutes           103594 non-null  float64
 24  satisfaction                       103594 non-null  object 
dtypes: float64(1), int64(19), object(5)
memory usage: 20.5+ MB
```

In [6]:
```python
#sum of datatypes
df.isnull().sum()
```

Out[6]:
```
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
```

In [9]:
```python
df.dropna(inplace=True)
```

In [19]:
```python
#duplicate values
df[df.duplicated()]
```

Out[19]:

Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness

0 rows × 25 columns

In [5]:
```python
#replacing values from rows
df.loc[1,'id']=2345
```

In [24]:
```python
# names of columns
df.columns
```

Out[24]:
```
Index(['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Inflight wifi service',
       'Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Inflight service',
       'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
       'satisfaction'],
      dtype='object')
```

In [27]:
```python
#calculating stastical values
df.describe(include=object)
```

Out[27]:

	Gender	Customer Type	Type of Travel	Class	satisfaction
count	103594	103594	103594	103594	103594
unique	2	2	2	3	2
top	Female	Loyal Customer	Business travel	Business	neutral or dissatisfied
freq	52576	84662	71465	49533	58697

In [28]:
```python
#finding average of same coloumns
df.groupby('Gender')['Age'].sum()
```

Out[28]:
```
Gender
Female    2063485
Male      2016095
Name: Age, dtype: int64
```

In [73]:
```python
#finding columns within particular age group
df[(df['Gender']=='Male') & (df['Age'].between(13,15))]
```

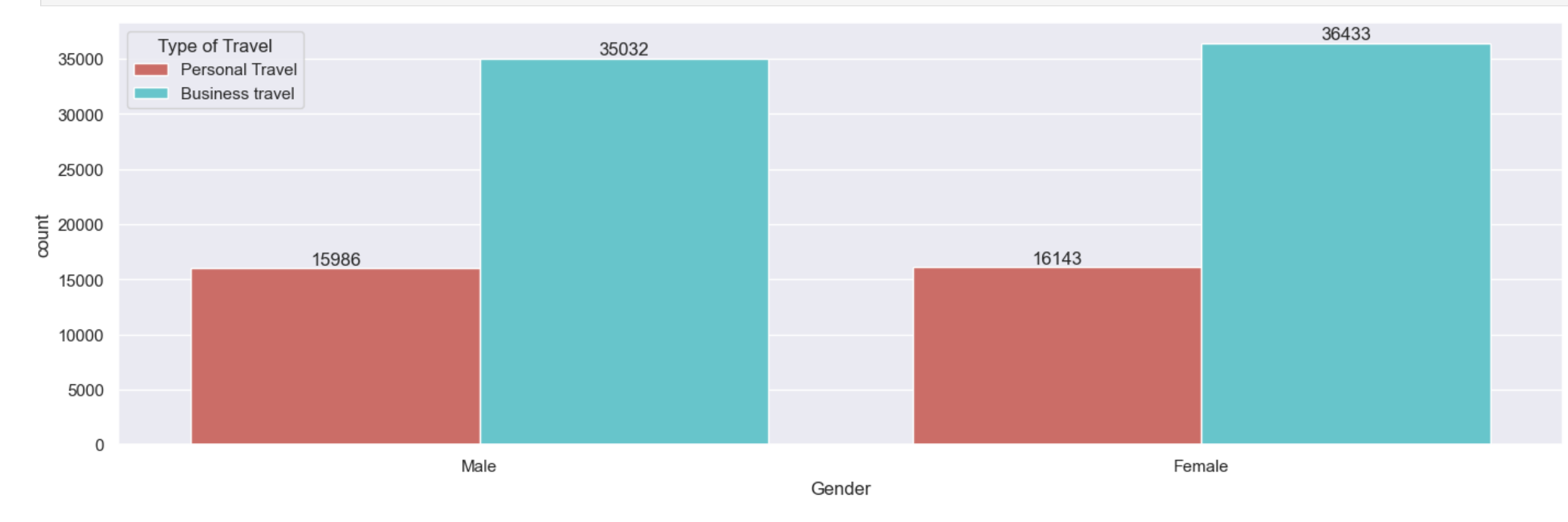
Out[73]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflig servic
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	4	
15	15	100580	Male	disloyal Customer	13	Business travel	Eco	486	2	1	...	4	2	1	4	1	
89	89	40017	Male	disloyal Customer	13	Business travel	Eco	525	2	2	...	3	3	1	3	1	
115	115	57513	Male	Loyal Customer	15	Personal Travel	Eco	235	2	4	...	1	3	3	4	4	
156	156	112483	Male	Loyal Customer	13	Personal Travel	Eco	853	1	4	...	4	5	2	5	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
103264	103264	28957	Male	Loyal Customer	14	Business travel	Business	2961	1	3	...	1	1	3	2	3	
103405	103405	122614	Male	Loyal Customer	14	Personal Travel	Eco	728	2	5	...	2	1	4	1	2	
103550	103550	113626	Male	Loyal Customer	14	Personal Travel	Eco	197	4	1	...	1	2	1	3	4	
103631	103631	123932	Male	Loyal Customer	13	Personal Travel	Eco	544	2	3	...	5	4	2	2	1	
103760	103760	28755	Male	disloyal Customer	15	Business travel	Eco	1024	2	2	...	5	4	4	4	4	

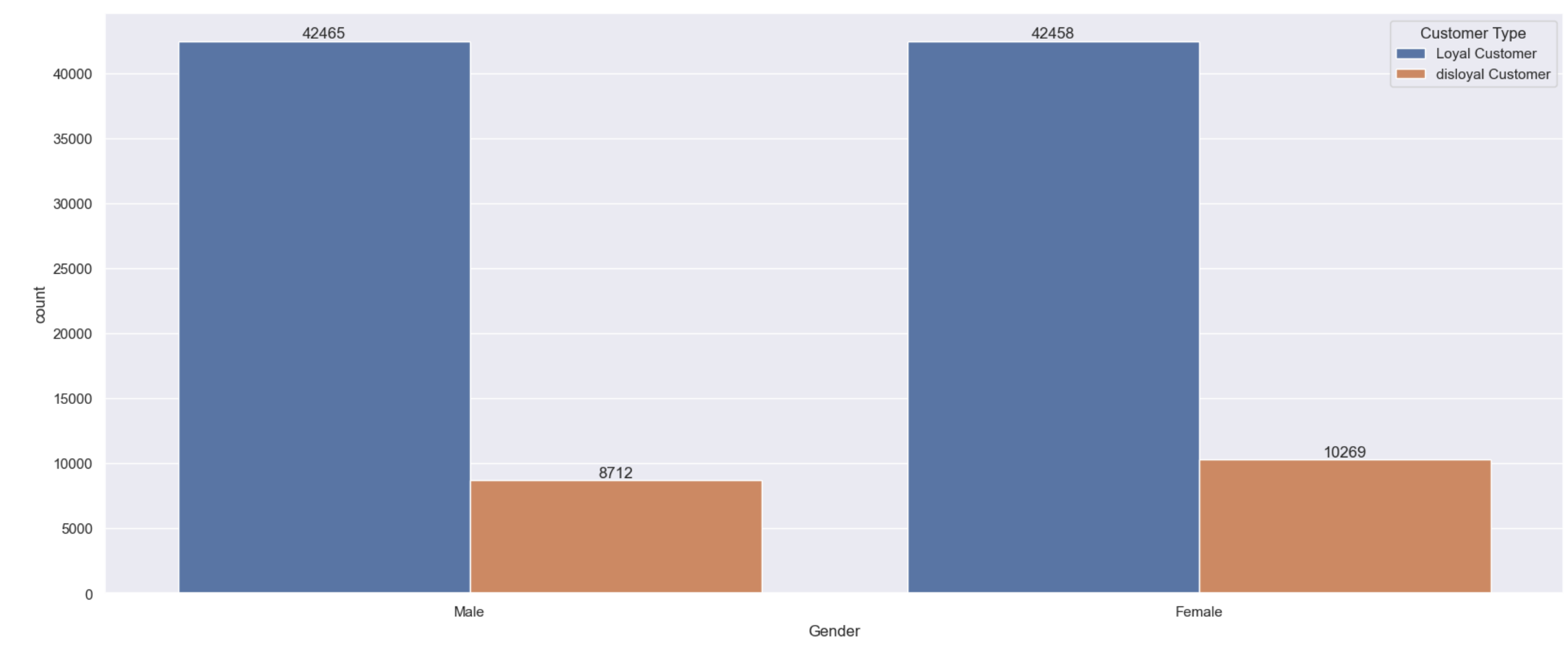
1058 rows × 25 columns

Exploratory Data Analysis

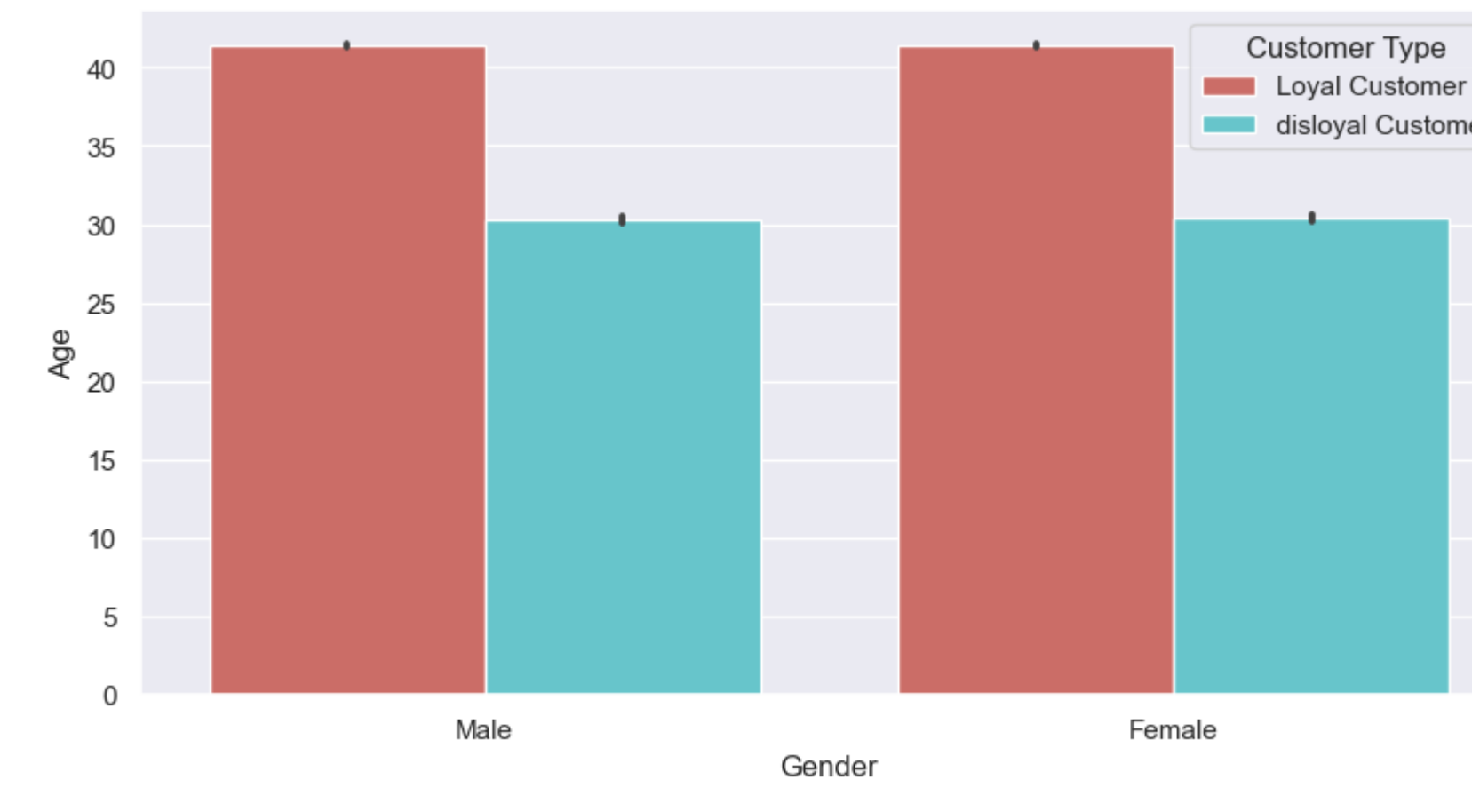
In [38]:
```python
# finding numbers of male and female passengers based on their travel type
ax = sns.countplot(x='Gender', data=df ,hue='Type of Travel',color='purple',palette = 'hls' )
sns.set(rc={'figure.figsize':(10,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



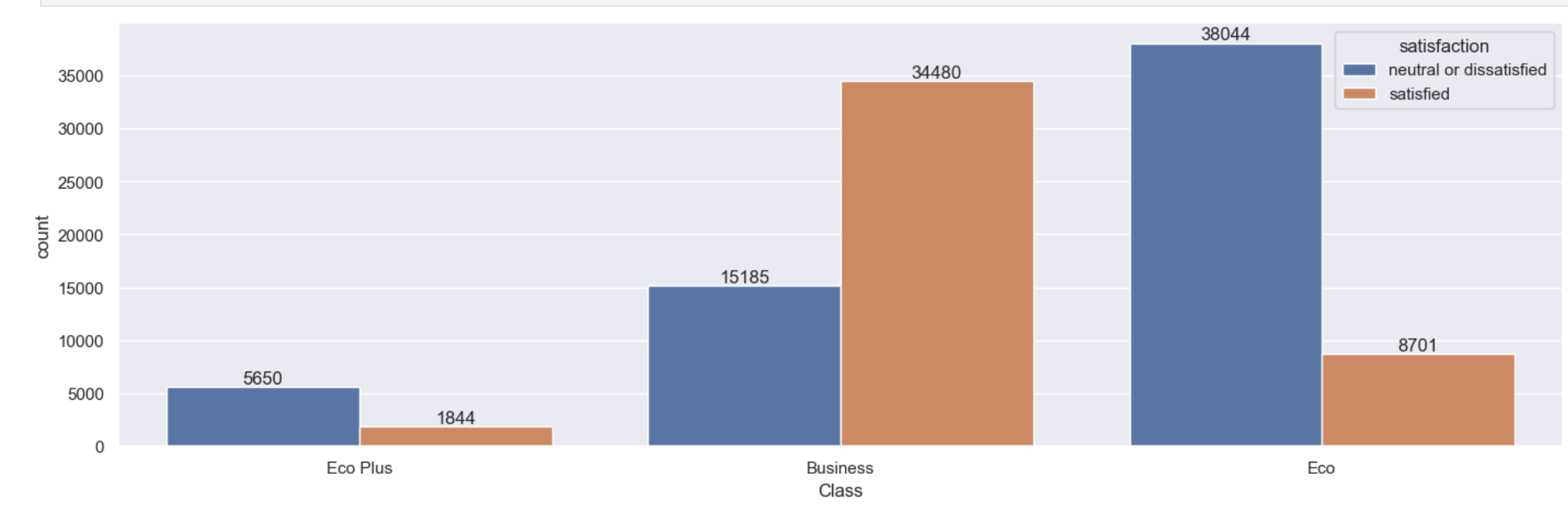
In [8]:
```python
# finding numbers of male and female passengers based on customer type
ax = sns.countplot(x='Gender', data=df ,hue='Customer Type' )
sns.set(rc={'figure.figsize':(15,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



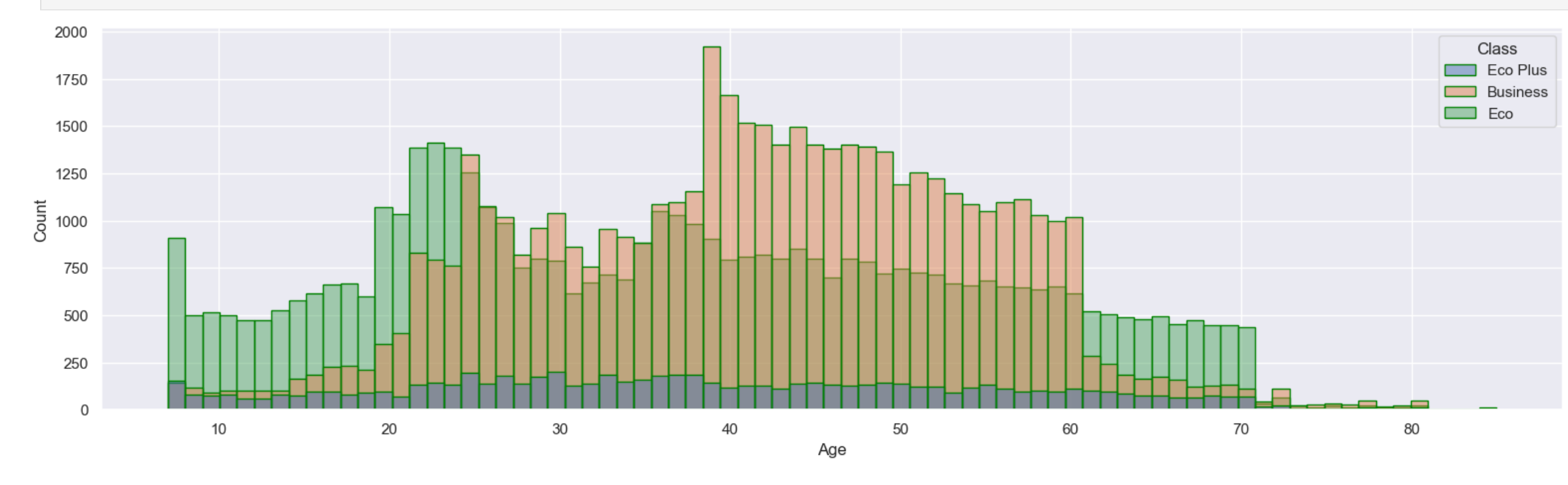
In [39]:
```python
# finding numbers of male and female passengers with their ages based on customer type
sns.barplot(x='Gender',y='Age' , data=df ,hue='Customer Type',color='purple',palette = 'hls')
sns.set(rc={'figure.figsize':(17,5)})
```



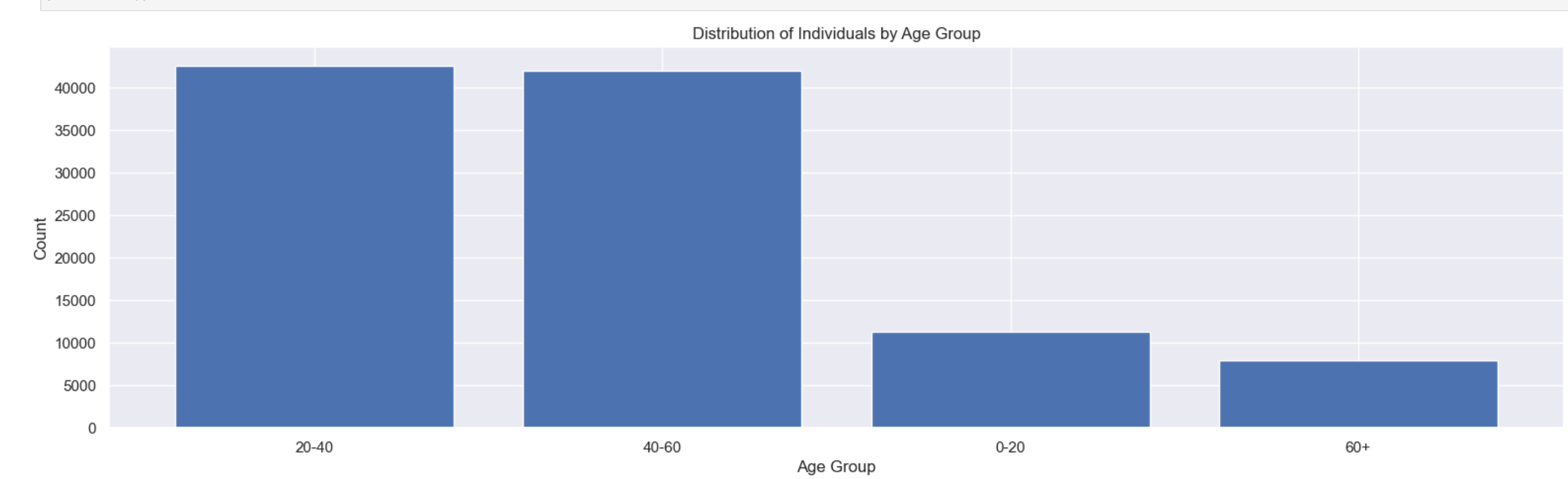
In [10]:
```python
# finding numbers of male and female passengers with their Class based on their satisafactory level
ax = sns.countplot(x='Class', data=df ,hue='satisfaction')
sns.set(rc={'figure.figsize':(19,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



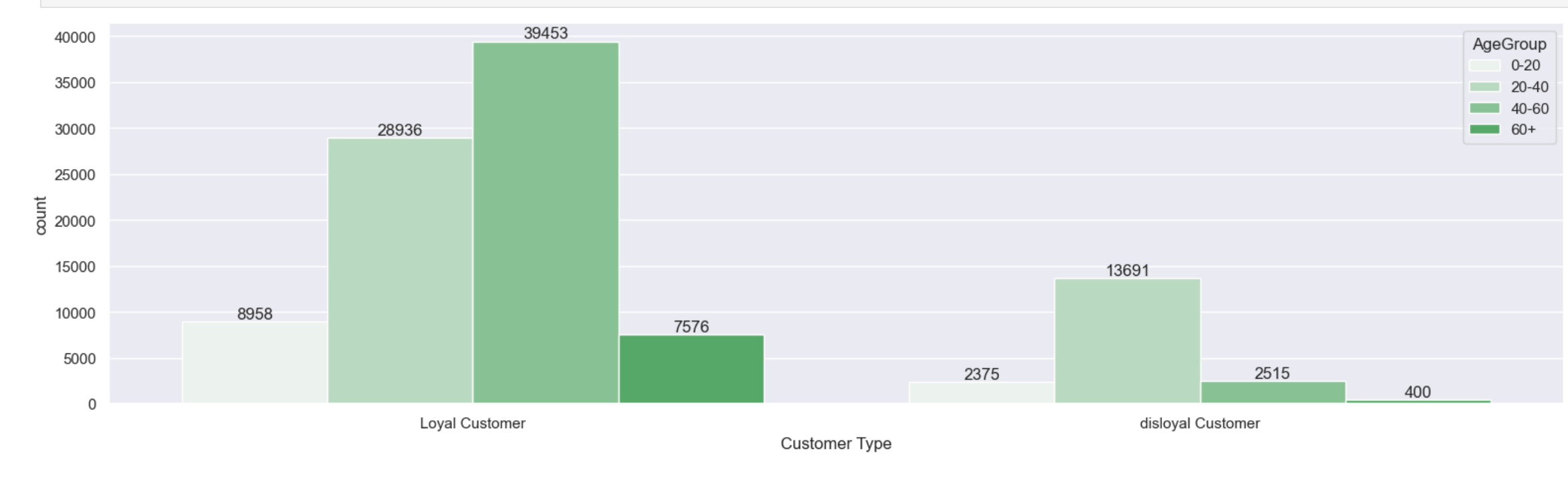
In [11]:
```python
# finding passengers with their age to class
sns.histplot(data=df, x="Age", hue="Class",edgecolor='green')
plt.show()
```



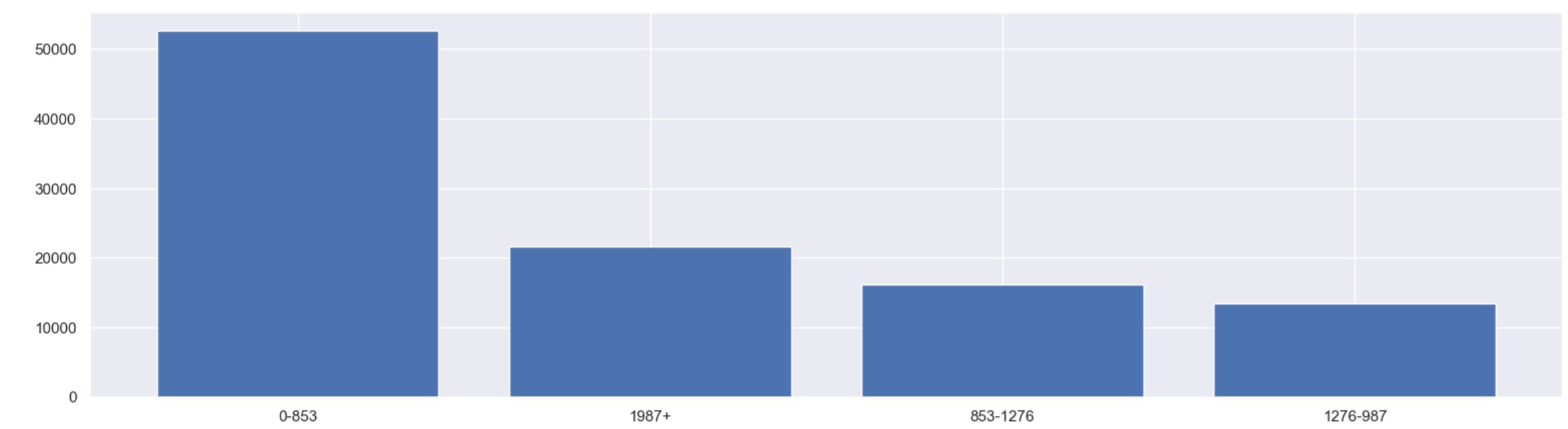
In [14]:
```python
#seprating passenger with their age group
df['AgeGroup'] = pd.cut(df['Age'], bins=[0, 20, 40, 60,float('inf')], labels=[ "0-20", "20-40", "40-60", "60+"])
age_group_counts = df['AgeGroup'].value_counts()
plt.bar(age_group_counts.index, age_group_counts.values)
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.title('Distribution of Individuals by Age Group')
plt.show()
```



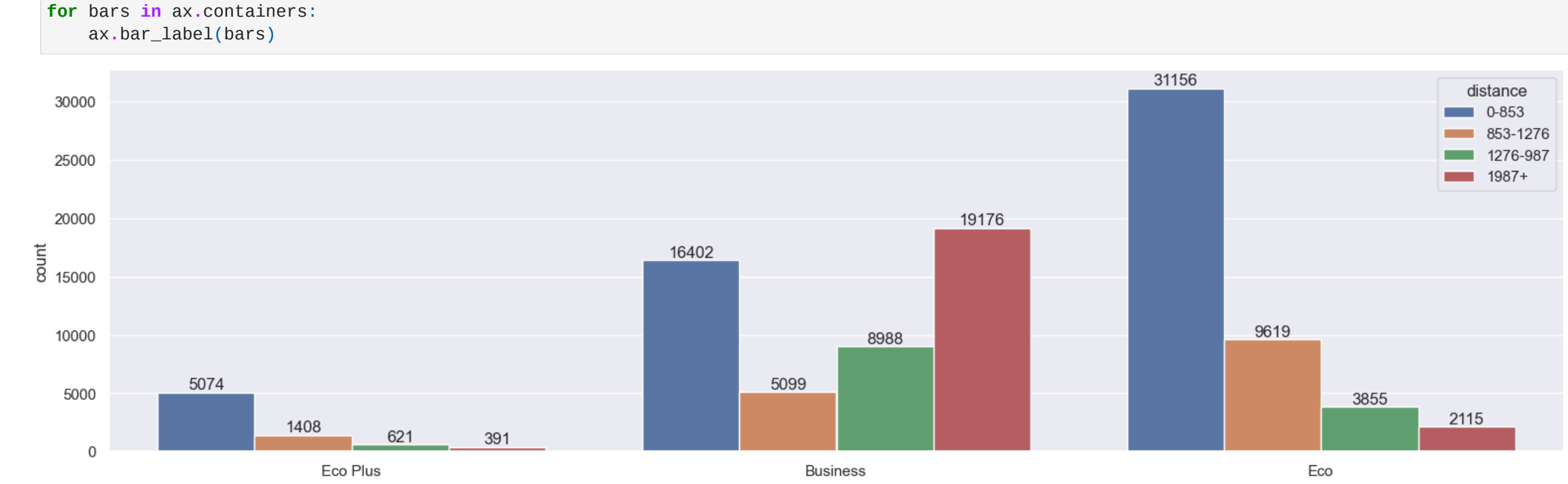
In [15]:
```python
#distribution of passengers type by their age groups
ax = sns.countplot(x='Customer Type', data=df ,hue='AgeGroup' ,color='g',saturation=5)
sns.set(rc={'figure.figsize':(19,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



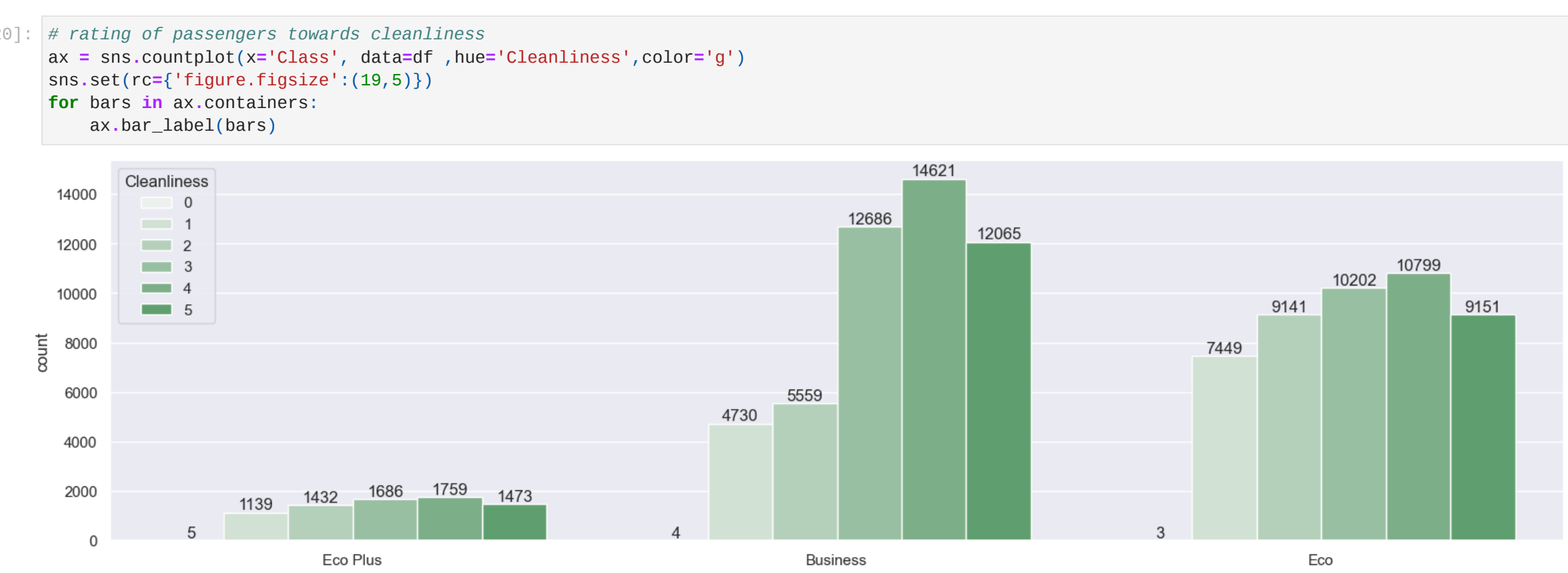
In [18]:
```python
#seprating passenger with their travel distances
df['distance'] = pd.cut(df['Flight Distance'], bins=[0, 853, 1276, 1987,float('inf')], labels=[ "0-853", "853-1276", "1276-987", "1987+"])
age_group_counts = df['distance'].value_counts()
plt.bar(age_group_counts.index, age_group_counts.values)
plt.show()
```



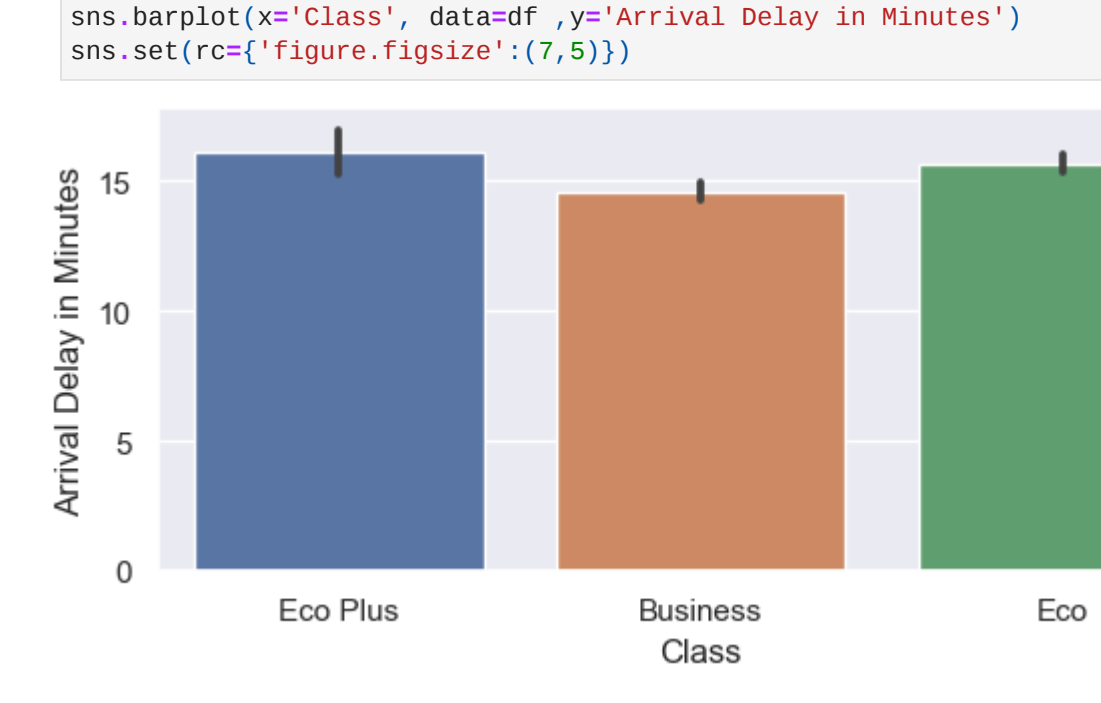
In [19]:
```python
# passengers in diferent classes with their travel distances
ax = sns.countplot(x='Class', data=df ,hue='distance')
sns.set(rc={'figure.figsize':(19,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



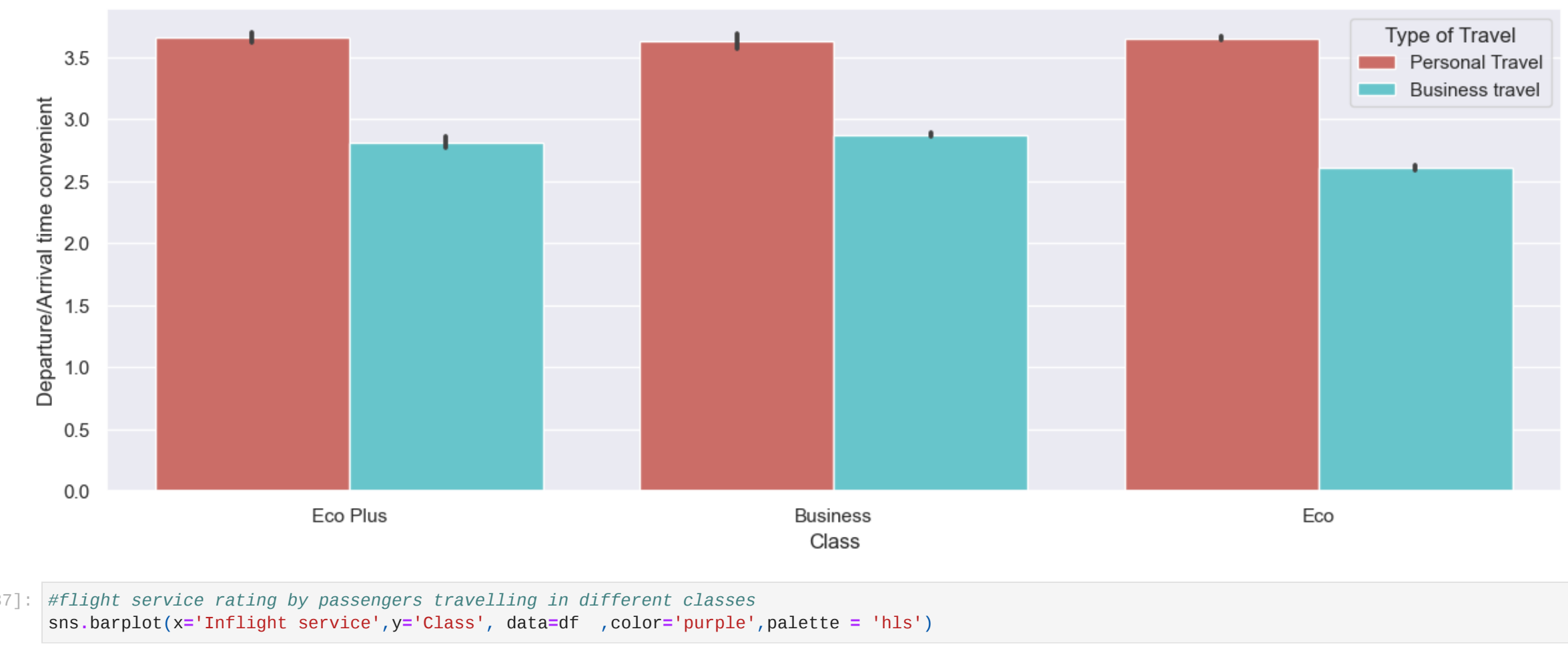
In [20]:
```python
# rating of passengers towards cleanliness
ax = sns.countplot(x='Class', data=df ,hue='Cleanliness',color='g')
sns.set(rc={'figure.figsize':(19,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [23]:
```python
# delay in flights
sns.barplot(x='Class', data=df ,y='Arrival Delay in Minutes')
sns.set(rc={'figure.figsize':(7,5)})
```

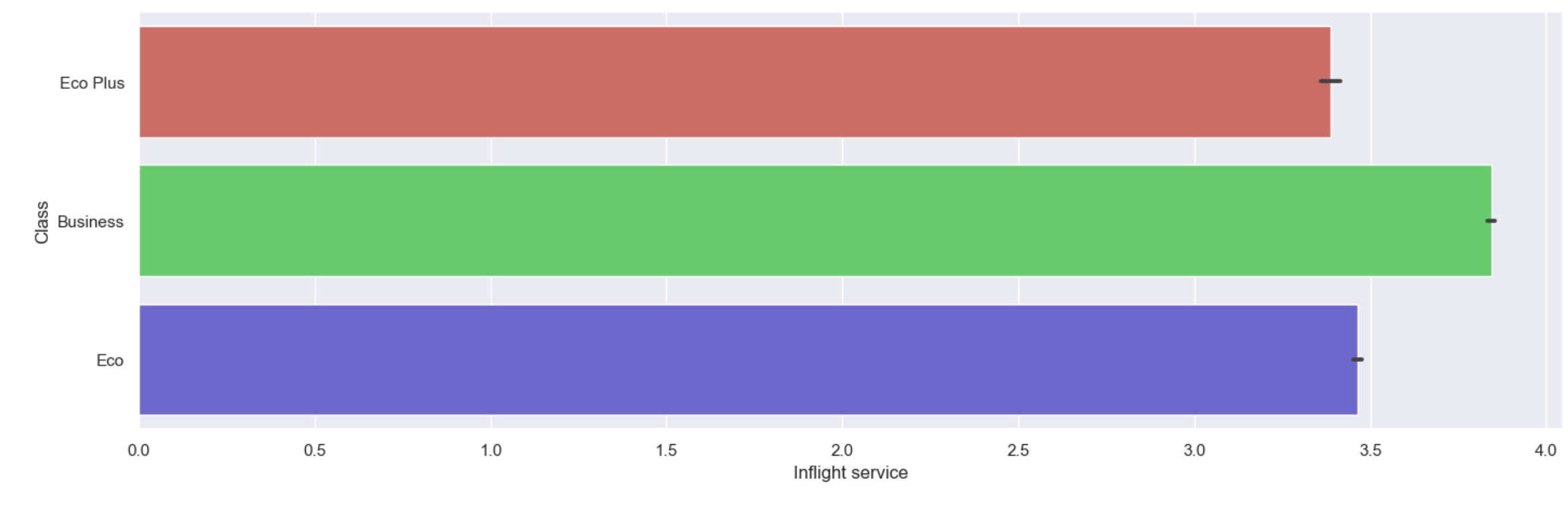


In [39]:
```python
# rating with respect to arrival/departure timming
sns.barplot(x='Class',y='Departure/Arrival time convenient',hue='Type of Travel' ,color='violet',palette = 'hls')
sns.set(rc={'figure.figsize':(15,6)})
```



In [37]:
```python
#flight service rating by passengers travelling in different classes
sns.barplot(x='Inflight service',y='Class', data=df  ,color='purple',palette = 'hls')
```

Out[37]: <Axes: xlabel='Inflight service', ylabel='Class'>



In [ ]: