

```
In [17]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
pd.set_option("display.max_columns",500)
pd.set_option("display.max_rows",500)
```

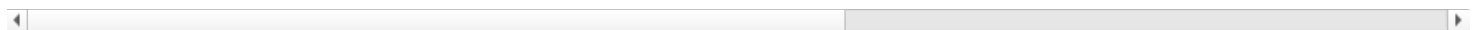
Load & Read Data

```
In [107]: df=pd.read_csv("Invistico_Airline.csv")
df.head(50)
```

Out[107]:

	satisfaction	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	Inflight wifi service	Inflight entertainment	Online support	baggage handling
0	satisfied	Loyal Customer	65	Personal Travel	Eco	265	0		0	0	2	2	4	2
1	satisfied	Loyal Customer	47	Personal Travel	Business	2464	0		0	0	3	0	2	2
2	satisfied	Loyal Customer	15	Personal Travel	Eco	2138	0		0	0	3	2	0	2
3	satisfied	Loyal Customer	60	Personal Travel	Eco	623	0		0	0	3	3	4	3
4	satisfied	Loyal Customer	70	Personal Travel	Eco	354	0		0	0	3	4	3	4
5	satisfied	Loyal Customer	30	Personal Travel	Eco	1894	0		0	0	3	2	0	2
6	satisfied	Loyal Customer	66	Personal Travel	Eco	227	0		0	0	3	2	5	5
7	satisfied	Loyal Customer	10	Personal Travel	Eco	1812	0		0	0	3	2	0	2
8	satisfied	Loyal Customer	56	Personal Travel	Business	73	0		0	0	3	5	3	5
9	satisfied	Loyal Customer	22	Personal Travel	Eco	1556	0		0	0	3	2	0	2
10	satisfied	Loyal Customer	58	Personal Travel	Eco	104	0		0	0	3	3	3	3
11	satisfied	Loyal Customer	34	Personal Travel	Eco	3633	0		0	0	4	2	0	2
12	satisfied	Loyal Customer	62	Personal Travel	Eco	1695	0		0	0	4	5	0	5
13	satisfied	Loyal Customer	35	Personal Travel	Eco	1766	0		1	0	1	4	0	4
14	satisfied	Loyal Customer	47	Personal Travel	Eco	84	0		1	0	1	5	2	1
15	satisfied	Loyal Customer	60	Personal Travel	Eco	1373	0		1	0	1	1	0	1
16	satisfied	Loyal Customer	13	Personal Travel	Eco	3693	0		1	0	2	4	0	4
17	satisfied	Loyal Customer	52	Personal Travel	Business	2610	0		1	0	2	1	2	2
18	satisfied	Loyal Customer	55	Personal Travel	Eco	2554	0		1	0	2	0	1	1
19	satisfied	Loyal Customer	28	Personal Travel	Eco	3095	0		1	0	2	3	0	3
20	satisfied	Loyal Customer	9	Personal Travel	Eco	3305	0		1	0	2	3	0	5
21	satisfied	Loyal Customer	10	Personal Travel	Eco	2090	0		1	0	2	1	0	1
22	satisfied	Loyal Customer	25	Personal Travel	Eco	2122	0		1	0	2	2	0	4
23	satisfied	Loyal Customer	53	Personal Travel	Business	1099	0		1	0	2	1	3	3
24	satisfied	Loyal Customer	16	Personal Travel	Eco Plus	1747	0		1	0	2	2	0	2
25	satisfied	Loyal Customer	30	Personal Travel	Eco	1817	0		1	0	2	4	0	4
26	satisfied	Loyal Customer	64	Personal Travel	Eco	1707	0		1	0	2	5	0	3
27	satisfied	Loyal Customer	42	Personal Travel	Eco	470	0		1	0	2	3	2	2

28	satisfied	Loyal Customer	9	Personal Travel	Eco	972	0	1	0	2	4	0	4
29	satisfied	Loyal Customer	35	Personal Travel	Eco	3695	0	1	0	3	0	4	4
30	satisfied	Loyal Customer	62	Personal Travel	Eco Plus	2948	0	1	0	3	5	0	5
31	satisfied	Loyal Customer	21	Personal Travel	Eco	2823	0	1	0	3	2	0	2
32	satisfied	Loyal Customer	20	Personal Travel	Eco	2485	0	1	0	3	2	0	2
33	satisfied	Loyal Customer	26	Personal Travel	Eco	2408	0	1	0	3	4	0	4
34	satisfied	Loyal Customer	20	Personal Travel	Eco	3009	0	1	0	3	4	0	3
35	satisfied	Loyal Customer	48	Personal Travel	Eco	1449	0	1	0	3	3	3	2
36	satisfied	Loyal Customer	10	Personal Travel	Eco	3209	0	1	0	3	4	0	4
37	satisfied	Loyal Customer	57	Personal Travel	Eco	2840	0	1	0	3	4	0	4
38	satisfied	Loyal Customer	25	Personal Travel	Eco	2534	0	1	0	3	2	0	2
39	satisfied	Loyal Customer	31	Personal Travel	Eco	2162	0	1	0	3	2	0	3
40	satisfied	Loyal Customer	22	Personal Travel	Eco	2352	0	1	0	3	3	0	3
41	satisfied	Loyal Customer	17	Personal Travel	Eco	2748	0	1	0	3	1	4	4
42	satisfied	Loyal Customer	33	Personal Travel	Eco	2045	0	1	0	3	1	0	4
43	satisfied	Loyal Customer	32	Personal Travel	Eco	2343	0	1	0	3	1	0	1
44	satisfied	Loyal Customer	60	Personal Travel	Eco	912	0	1	0	3	4	3	4
45	satisfied	Loyal Customer	38	Personal Travel	Eco Plus	2139	0	1	0	3	5	0	5
46	satisfied	Loyal Customer	29	Personal Travel	Eco	1918	0	1	0	3	5	0	5
47	satisfied	Loyal Customer	42	Personal Travel	Eco	2311	0	1	0	3	1	0	1
48	satisfied	Loyal Customer	31	Personal Travel	Business	1599	0	1	0	3	2	0	2
49	satisfied	Loyal Customer	34	Personal Travel	Eco	1816	0	1	0	3	4	0	4



In [4]: df.shape

Out[4]: (129880, 22)

In [6]: df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 22 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   satisfaction                          129880 non-null object
1   Customer Type                        129880 non-null object
2   Age                                  129880 non-null int64
3   Type of Travel                       129880 non-null object
4   Class                                129880 non-null object
5   Flight Distance                      129880 non-null int64
6   Seat comfort                         129880 non-null int64
7   Departure/Arrival time convenient    129880 non-null int64
8   Food and drink                      129880 non-null int64
9   Gate location                       129880 non-null int64
10  Inflight wifi service                129880 non-null int64
11  Inflight entertainment               129880 non-null int64
12  Online support                      129880 non-null int64
13  Ease of Online booking               129880 non-null int64
14  On-board service                    129880 non-null int64
15  Leg room service                    129880 non-null int64
16  Baggage handling                    129880 non-null int64
17  Checkin service                     129880 non-null int64
18  Cleanliness                         129880 non-null int64
19  Online boarding                     129880 non-null int64
20  Departure Delay in Minutes           129880 non-null int64
21  Arrival Delay in Minutes             129487 non-null float64
dtypes: float64(1), int64(17), object(4)
memory usage: 21.8+ MB

```

```
In [8]: df.dtypes
```

```

Out[8]: 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 [9]: df.isnull().sum()
```

```

Out[9]: 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 [14]: df["Arrival Delay in Minutes"].isnull()
```

```
Out[14]: 0      False
          1      False
          2      False
          3      False
          4      False
          ...
          129875 False
          129876 False
          129877 False
          129878 False
          129879 False
          Name: Arrival Delay in Minutes, Length: 129880, dtype: bool
```

```
In [15]: df.describe()
```

Out[15]:

	Age	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	Inflight wifi service	Inflight entertainment	Online support
count	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000
mean	39.427957	1981.409055	2.838597	2.990645	2.851994	2.990422	3.249130	3.383477	3.501750
std	15.119360	1027.115606	1.392983	1.527224	1.443729	1.305970	1.318818	1.346059	1.305970
min	7.000000	50.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	27.000000	1359.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	3.000000
50%	40.000000	1925.000000	3.000000	3.000000	3.000000	3.000000	3.000000	4.000000	4.000000
75%	51.000000	2544.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	5.000000
max	85.000000	6951.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000

Creating Age Groups

```
In [19]: df["Age"]
```

```
Out[19]: 0      65
          1      47
          2      15
          3      60
          4      70
          ..
          129875  29
          129876  63
          129877  69
          129878  66
          129879  38
          Name: Age, Length: 129880, dtype: int64
```

```
In [21]: ## Create the bucket <10, 10-20, 20-30, 30-40, 40-50, 50-60, 60-70, 70-80, 80+
          df["Age_Group"] = pd.cut(df.Age, bins=[0,10,20,30,40,50,60,70,80,9999], labels=["<10", "10-20", "20-30", "30-40", "40-50", "50-60", "60-70", "70-80", "80+"], dtype='object')
```

```
In [24]: df.columns
```

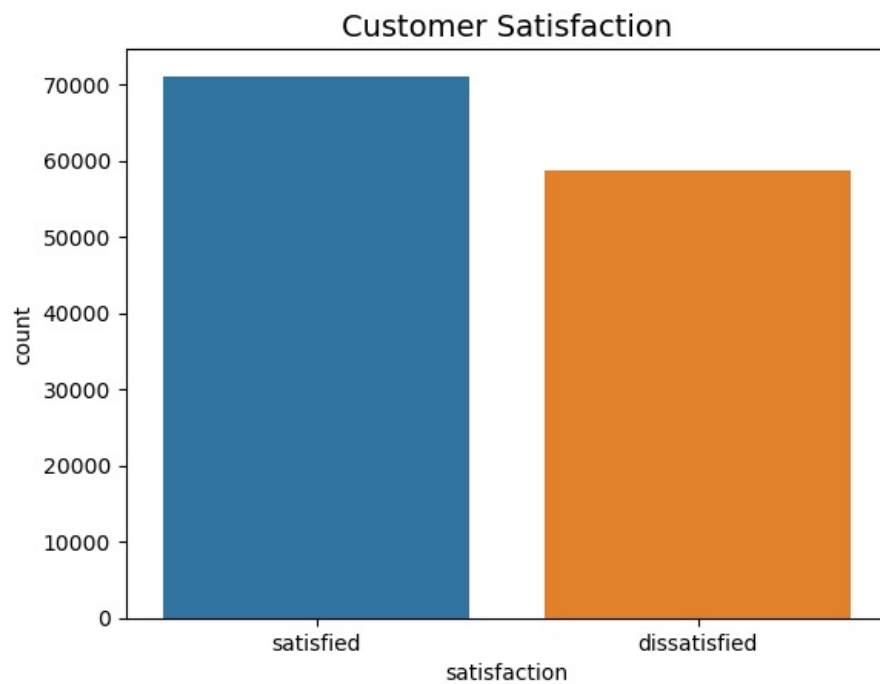
```
Out[24]: Index(['satisfaction', 'Customer Type', 'Age', 'Type of Travel', 'Class',
              'Flight Distance', 'Seat comfort', 'Departure/Arrival time convenient',
              'Food and drink', 'Gate location', 'Inflight wifi service',
              'Inflight entertainment', 'Online support', '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', 'Age_Group'],
              dtype='object')
```

Exploratory Data Analysis

Exploring the distribution of the first five columns, Satisfication, Customer Type, Type of Travel, Class, Age Group

```
In [111]: ### Exploring the distribution of Satisfaction of Customers
          sns.countplot(data=df, x="satisfaction")
          plt.title("Customer Satisfaction", size=14)
```

```
Out[111]: Text(0.5, 1.0, 'Customer Satisfaction')
```



In [120]: *##Data shows that there are more Invistico Airline customers who are datsified compared to dissatisfied, ##although there is a slightly different about +/-10 of the difference*

Customer Satisfaction By Age Group

In [85]: `Satisfication_By_Age_Group=df.groupby("Age_Group")["satisfaction"].value_counts().sort_values(ascending=False)`

In [86]: `Satisfication_By_Age_Group`

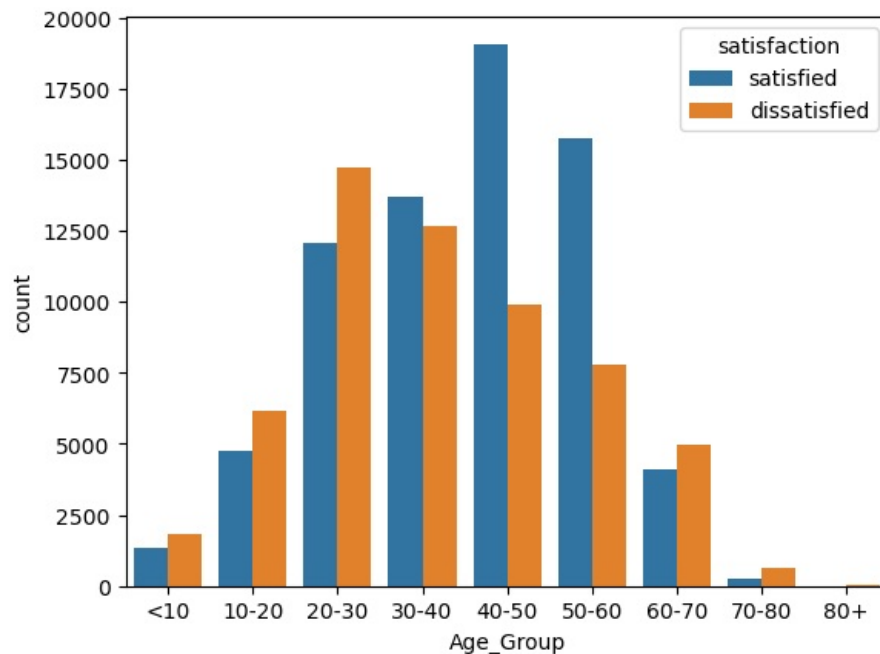
Out[86]:

Age_Group	satisfaction	count
40-50	satisfied	19084
50-60	satisfied	15755
20-30	dissatisfied	14731
30-40	satisfied	13695
	dissatisfied	12697
20-30	satisfied	12091
40-50	dissatisfied	9922
50-60	dissatisfied	7793
10-20	dissatisfied	6153
60-70	dissatisfied	4974
10-20	satisfied	4742
60-70	satisfied	4116
<10	dissatisfied	1845
	satisfied	1318
70-80	dissatisfied	659
	satisfied	280
80+	dissatisfied	19
	satisfied	6

Name: count, dtype: int64

In [103]: `sns.countplot(data=df, hue='satisfaction', x='Age_Group')`

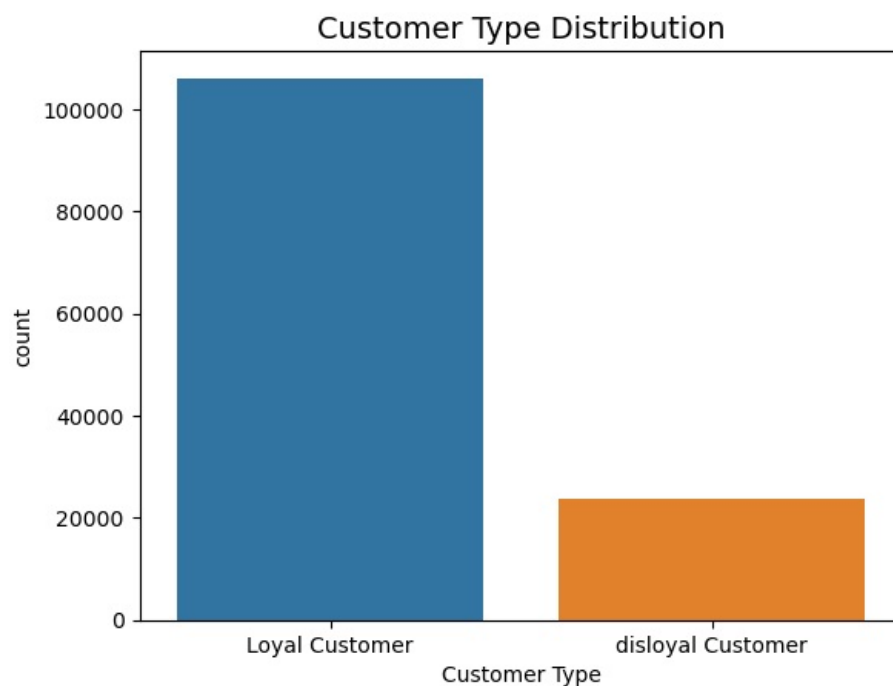
Out[103]: `<Axes: xlabel='Age_Group', ylabel='count'>`



In [122]: *##Age group from 30-50 shows a great number of customers who are satisfied, while <30 & >60 shows to be dissati*

In [26]: *## Exploring the distribution of Customer Type*
`sns.countplot(data=df, x="Customer Type")`
`plt.title("Customer Type Distribution", size=14)`

Out[26]: `Text(0.5, 1.0, 'Customer Type Distribution')`



In [123]: *##Loyal Customers are greater than Disloyal Customers*

The Distribution Of Loyal Customers By Age Group

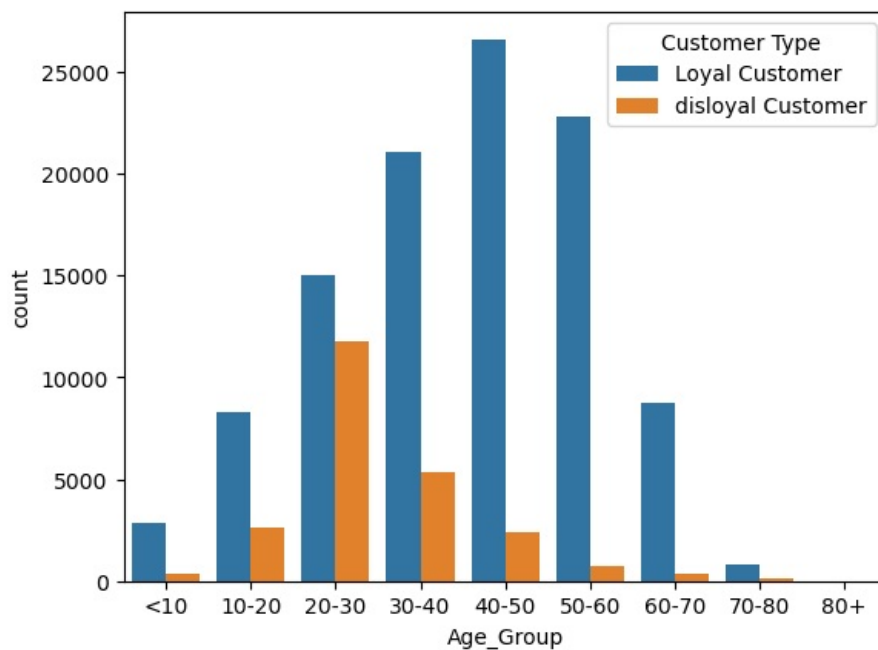
In [100]: `Customers_Type_By_Age_Group=df.groupby('Age_Group')['Customer Type'].value_counts().sort_values(ascending=False)`

```
In [101]: Customers_Type_By_Age_Group
```

```
Out[101]: Age_Group Customer Type
40-50      Loyal Customer      26586
50-60      Loyal Customer      22816
30-40      Loyal Customer      21031
20-30      Loyal Customer      15036
          disloyal Customer      11786
60-70      Loyal Customer       8714
10-20      Loyal Customer       8251
30-40      disloyal Customer     5361
<10       Loyal Customer       2829
10-20      disloyal Customer     2644
40-50      disloyal Customer     2420
70-80      Loyal Customer        822
50-60      disloyal Customer       732
60-70      disloyal Customer       376
<10       disloyal Customer       334
70-80      disloyal Customer       117
80+       Loyal Customer         15
          disloyal Customer         10
Name: count, dtype: int64
```

```
In [104]: sns.countplot(data=df, hue='Customer Type', x='Age_Group')
```

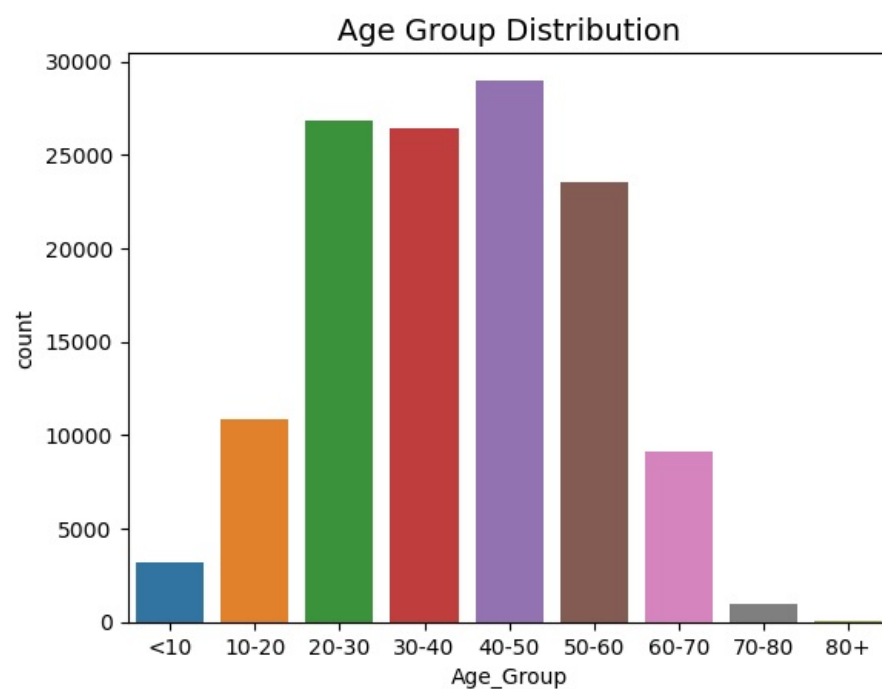
```
Out[104]: <Axes: xlabel='Age_Group', ylabel='count'>
```



```
In [131]: ## Age Group between 20-60, shows a greater number of Loyal Customers
```

```
In [30]: sns.countplot(data=df, x="Age_Group")
plt.title("Age Group Distribution", size=14)
```

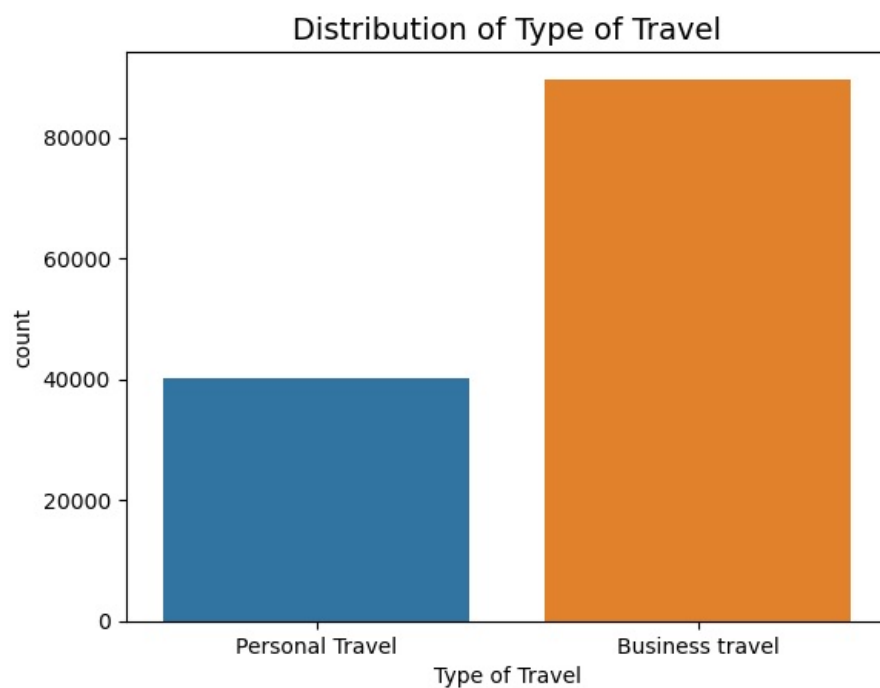
```
Out[30]: Text(0.5, 1.0, 'Age Group Distribution')
```



```
In [130]: ## Invistico Airlines Customer Shows A Great Numbe of Age Group Between 20-60
```

```
In [31]: ## Exploring the distribution of Type of Travel
sns.countplot(data=df, x="Type of Travel")
plt.title("Distribution of Type of Travel", size=14)
```

```
Out[31]: Text(0.5, 1.0, 'Distribution of Type of Travel')
```

```
In [129]: ## 60% of Customers Uses Invistico Airlines For Business Travel.  
## 40% of Customers Uses Invistico Airlines For Personal Travel.
```

```
In [79]: Travel_Type_by_Age_Group=df.groupby("Age_Group")["Type of Travel"].value_counts().sort_values(ascending=False)
```

```
In [80]: Travel_Type_by_Age_Group
```

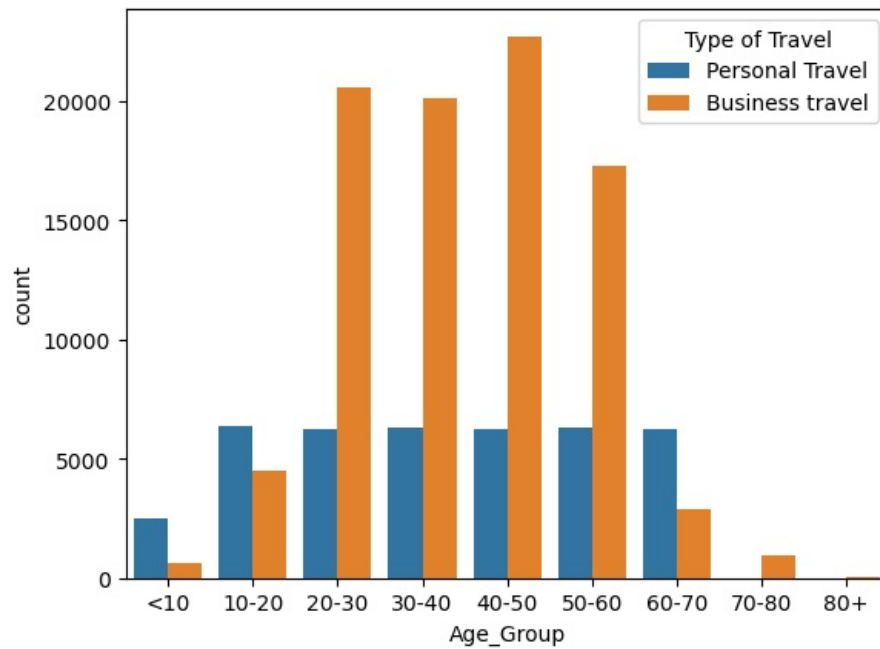
```
Out[80]:
```

Age Group	Type of Travel	count
40-50	Business travel	22729
20-30	Business travel	20601
30-40	Business travel	20103
50-60	Business travel	17266
10-20	Personal Travel	6368
30-40	Personal Travel	6289
50-60	Personal Travel	6282
40-50	Personal Travel	6277
60-70	Personal Travel	6222
20-30	Personal Travel	6221
10-20	Business travel	4527
60-70	Business travel	2868
<10	Personal Travel	2528
70-80	Business travel	939
<10	Business travel	635
80+	Business travel	25
70-80	Personal Travel	0
80+	Personal Travel	0

Name: count, dtype: int64

```
In [106]: sns.countplot(data=df, hue='Type of Travel', x='Age_Group')
```

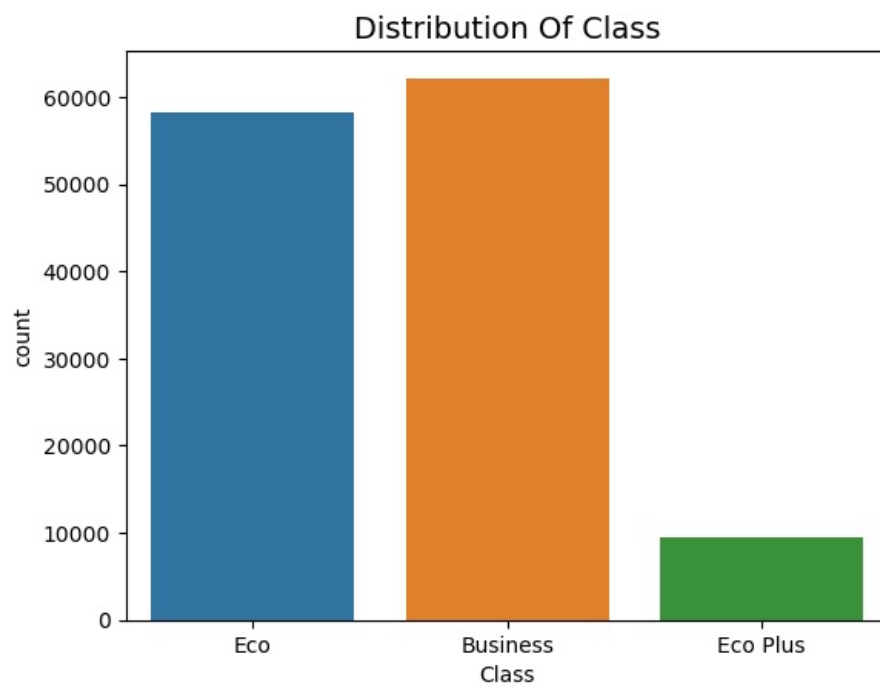
```
Out[106]: <Axes: xlabel='Age_Group', ylabel='count'>
```



In [132]: *## Age Groups between 20-60 uses Invistico Airlines for Business Travel more compared to other age groups*

In [32]: *### Exploring the Distribution of Class*
`sns.countplot(data=df, x="Class")`
`plt.title("Distribution Of Class", size=14)`

Out[32]: Text(0.5, 1.0, 'Distribution Of Class')



Distribution Of Class By Age Group

In [89]: `Class_By_Age_Group=df.groupby('Age_Group')['Class'].value_counts()`

In [90]: `Class_By_Age_Group`

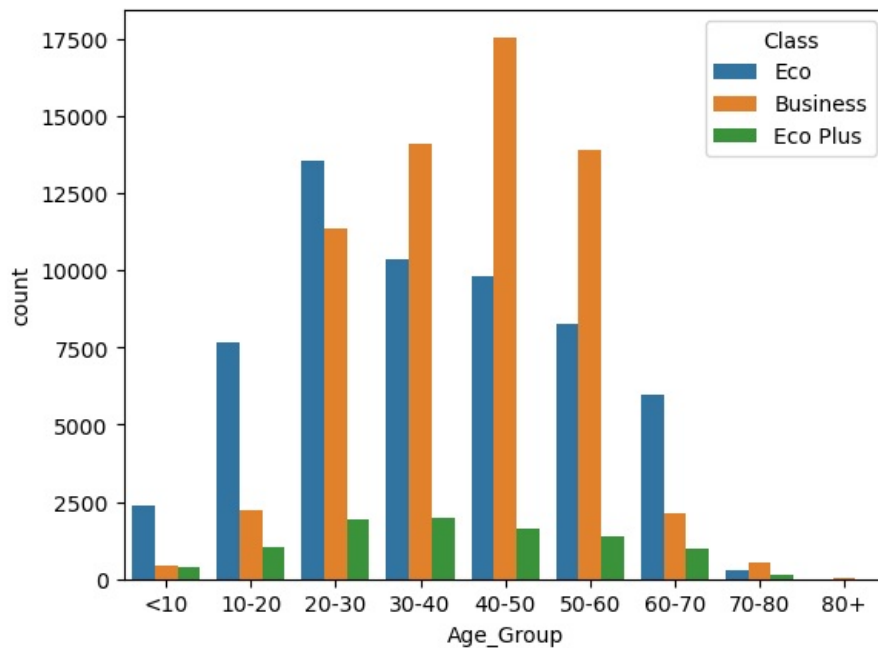
```
Out[90]:
```

Age_Group	Class	Count
<10	Eco	2383
	Business	417
	Eco Plus	363
10-20	Eco	7662
	Business	2205
	Eco Plus	1028
20-30	Eco	13553
	Business	11352
	Eco Plus	1917
30-40	Business	14075
	Eco	10361
	Eco Plus	1956
40-50	Business	17547
	Eco	9825
	Eco Plus	1634
50-60	Business	13911
	Eco	8239
	Eco Plus	1398
60-70	Eco	5989
	Business	2115
	Eco Plus	986
70-80	Business	518
	Eco	294
	Eco Plus	127
80+	Business	20
	Eco	3
	Eco Plus	2

Name: count, dtype: int64

```
In [105]: sns.countplot(data=df, hue='Class', x='Age_Group')
```

```
Out[105]: <Axes: xlabel='Age_Group', ylabel='count'>
```



```
In [133]: ## Age Group Between 30-60 years of age uses business class More,
## While Age groups of <10-70 uses Economic Class More
```

Creating Bins To Convert Departure/Arrival time convenient, Food and drink, Gate location, Inflight wifi service, Inflight entertainment, Online support, Ease of Online booking, On-board service, Leg room service, Baggage handling, Checkin service, Cleanliness, Online boarding Into Categorical Labels

```
In [34]: ## Converting Departure/Arrival Time Convenient into Categorical Labels
df["Rating_On_D/A_Time_Convenient"]=pd.cut(df["Departure/Arrival time convenient"],[0,1,2,3,4,5],labels=["Very
```

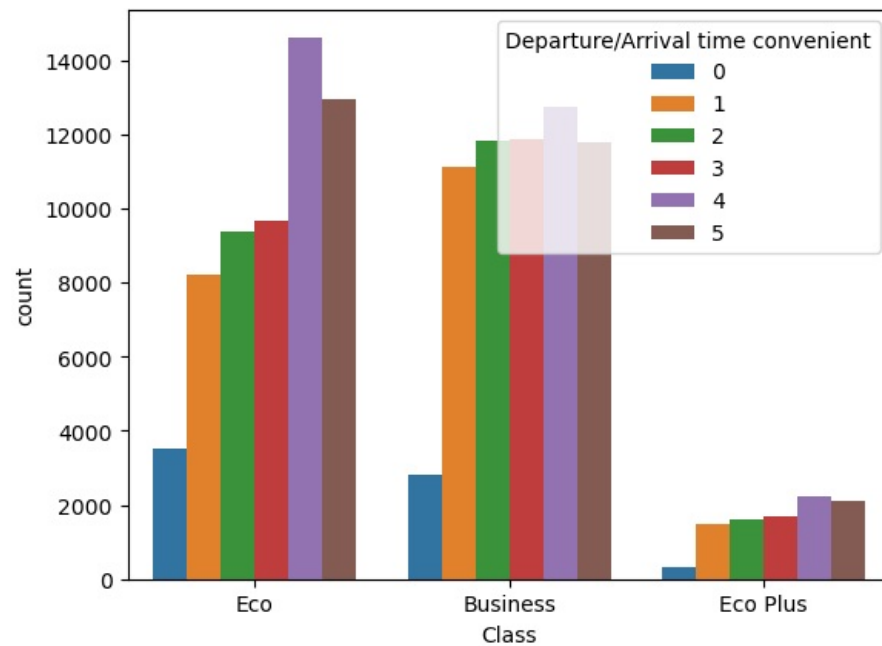
```
In [54]: sns.countplot(data=df, x="Rating_On_D/A_Time_Convenient")
plt.title("Customer Satisfaction On Departure/Arrival Time Convenient", size=14)
```

```
Out[54]: Text(0.5, 1.0, 'Customer Satisfaction On Departure/Arrival Time Convenient')
```



```
In [138]: sns.countplot(data=df, hue='Departure/Arrival time convenient', x='Class')
```

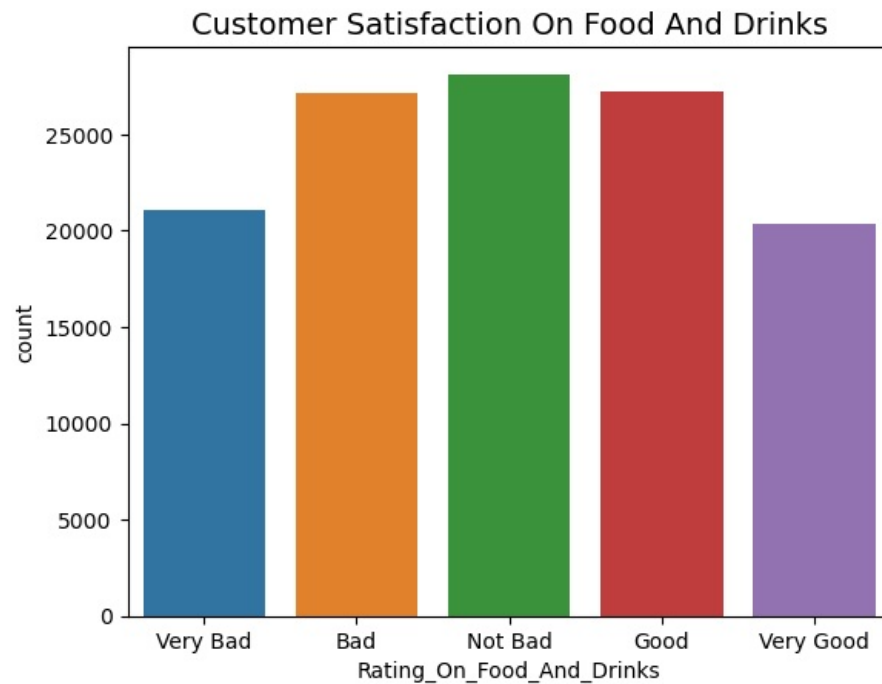
```
Out[138]: <Axes: xlabel='Class', ylabel='count'>
```



```
In [35]: ## Converting Food and Drink into Categorical Labels
df["Rating_On_Food_And_Drinks"] = pd.cut(df["Food and drink"], [0,1,2,3,4,5], labels=["Very Bad", "Bad", "Not Bad",
```

```
In [55]: sns.countplot(data=df, x="Rating_On_Food_And_Drinks")
plt.title("Customer Satisfaction On Food And Drinks", size=14)
```

```
Out[55]: Text(0.5, 1.0, 'Customer Satisfaction On Food And Drinks')
```



```
In [94]: Rating_On_Food_And_Drinks_By_Class=df.groupby('Rating_On_Food_And_Drinks')['Class'].value_counts()
```

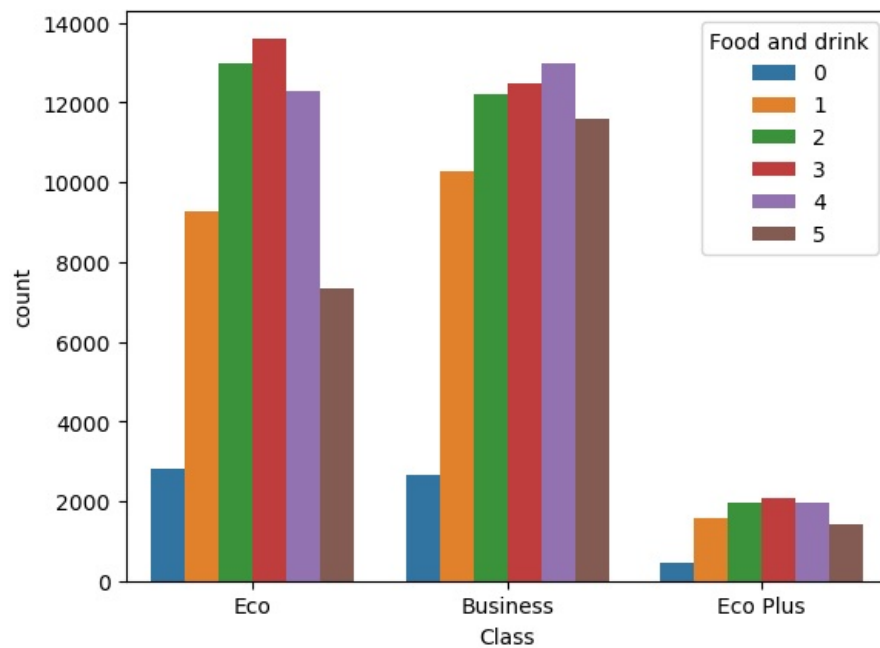
```
In [95]: Rating_On_Food_And_Drinks_By_Class
```

```
Out[95]: Rating_On_Food_And_Drinks  Class
Very Bad                          Business    10260
                                   Eco         9258
                                   Eco Plus    1558
Bad                                Eco        12981
                                   Business    12204
                                   Eco Plus     1961
Not Bad                           Eco        13607
                                   Business    12483
                                   Eco Plus     2060
Good                               Business    12964
                                   Eco        12299
                                   Eco Plus     1953
Very Good                         Business    11591
                                   Eco         7337
                                   Eco Plus     1419

Name: count, dtype: int64
```

```
In [128]: sns.countplot(data=df, hue='Food and drink', x='Class')
```

```
Out[128]: <Axes: xlabel='Class', ylabel='count'>
```



```
In [36]: ## Converting Gate Location into Categorical Labels
df["Rating_On_Gate_Location"] = pd.cut(df["Gate location"], [0,1,2,3,4,5], labels=["Very Bad", "Bad", "Not Bad", "G
```

```
In [56]: sns.countplot(data=df, x="Rating_On_Gate_Location")
plt.title("Customer Satisfaction On Gate Location")
```

```
Out[56]: Text(0.5, 1.0, 'Customer Satisfaction On Gate Location')
```



```
In [37]: ## Converting Inflight Wifi Service into Categorical Labels
```

```
df["Rating_On_Inflight_Wifi_Service"]=pd.cut(df["Inflight wifi service"],[0,1,2,3,4,5],labels=["Very Bad", "Bad", "Not Bad", "Good", "Very Good"])
```

```
In [57]: sns.countplot(data=df, x="Rating_On_Inflight_Wifi_Service")
plt.title("Customer Satisfaction On Inflight Wifi Service", size=14)
```

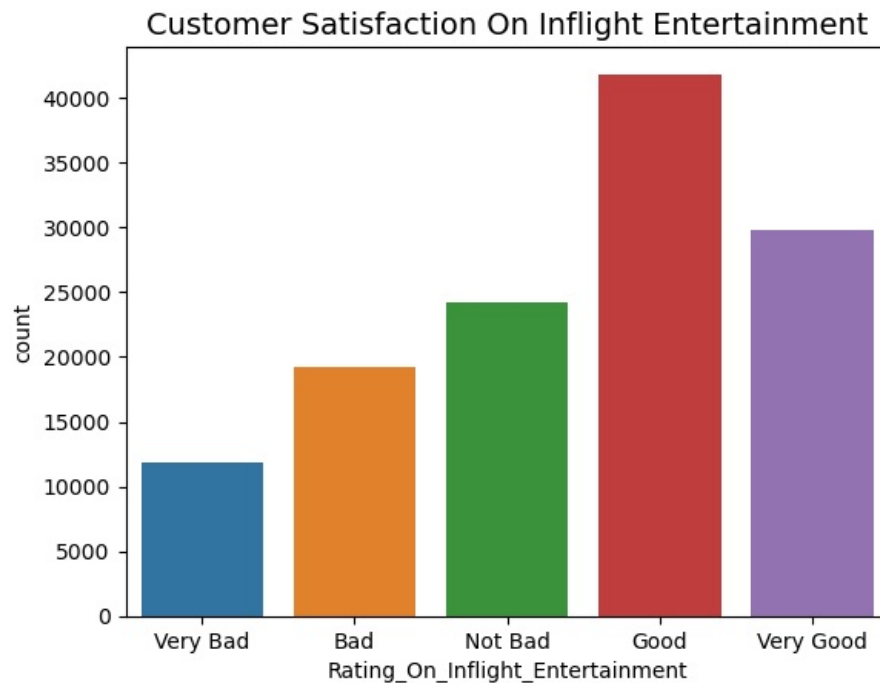
```
Out[57]: Text(0.5, 1.0, 'Customer Satisfaction On Inflight Wifi Service')
```



```
In [38]: ## Converting Inflight Entertainment into Categorical Labels
df["Rating_On_Inflight_Entertainment"]=pd.cut(df["Inflight entertainment"],[0,1,2,3,4,5],labels=["Very Bad", "Bad", "Not Bad", "Good", "Very Good"])
```

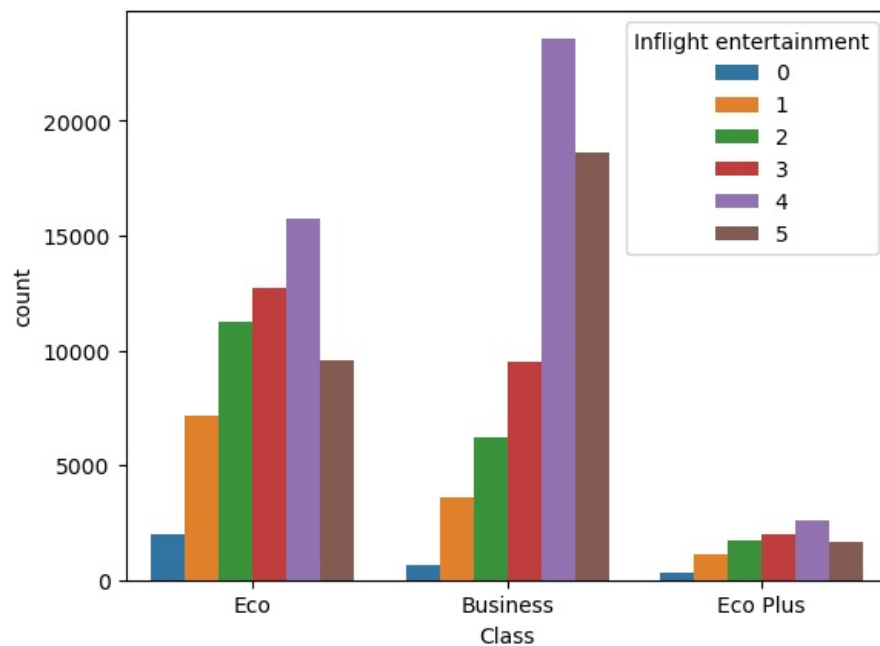
```
In [61]: sns.countplot(data=df, x="Rating_On_Inflight_Entertainment")
plt.title("Customer Satisfaction On Inflight Entertainment", size=14)
```

```
Out[61]: Text(0.5, 1.0, 'Customer Satisfaction On Inflight Entertainment')
```



```
In [135]: sns.countplot(data=df, hue='Inflight entertainment', x='Class')
```

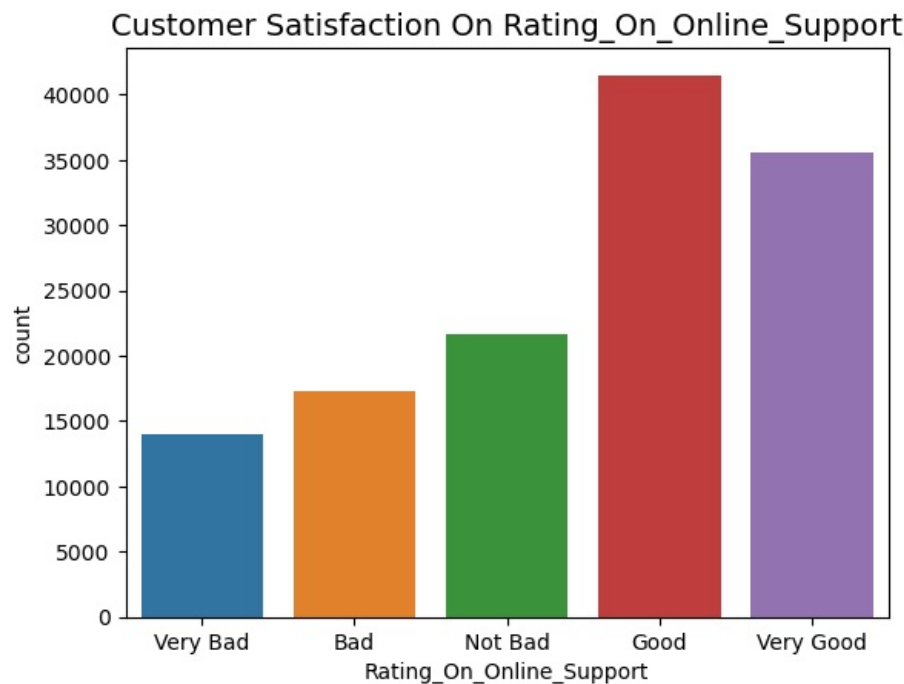
```
Out[135]: <Axes: xlabel='Class', ylabel='count'>
```



```
In [39]: ## Converting Online Support into Categorical Labels
df["Rating_On_Online_Support"] = pd.cut(df["Online support"], [0,1,2,3,4,5], labels=["Very Bad", "Bad", "Not Bad",
```

```
sns.countplot(data=df, x="Rating_On_Online_Support")
plt.title("Customer Satisfaction On Rating_On_Online_Support", size=14)
```

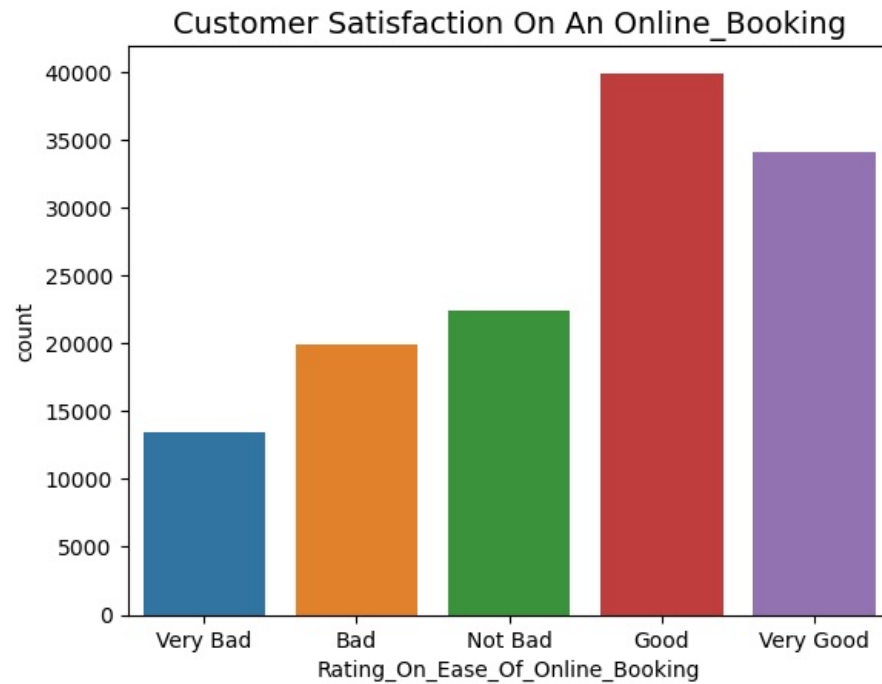
```
Out[62]: Text(0.5, 1.0, 'Customer Satisfaction On Rating_On_Online_Support')
```



```
In [41]: ## Converting Ease Of Online Booking into Categorical Labels
df["Rating_On_Ease_Of_Online_Booking"] = pd.cut(df["Ease of Online booking"], [0,1,2,3,4,5], labels=["Very Bad", "B
```

```
In [63]: sns.countplot(data=df, x="Rating_On_Ease_Of_Online_Booking")
plt.title("Customer Satisfaction On An Online_Booking", size=14)
```

```
Out[63]: Text(0.5, 1.0, 'Customer Satisfaction On An Online_Booking')
```

```
In [43]: ## Converting On-boarding Service into Categorical Labels
df["Rating_On_On_board_Service"] = pd.cut(df["On-board service"], [0,1,2,3,4,5], labels=["Very Bad", "Bad", "Not Ba
```

```
In [64]: sns.countplot(data=df, x="Rating_On_On_board_Service")
plt.title("Customer Satisfaction On An On-board Service", size=14)
```

```
Out[64]: Text(0.5, 1.0, 'Customer Satisfaction On An On-board Service')
```



```
In [44]: ## Converting Leg Room Service into Categorical Labels
df["Rating_On_Leg_Room_Service"] = pd.cut(df["Leg room service"], [0,1,2,3,4,5], labels=["Very Bad", "Bad", "Not Ba
```

```
In [67]: sns.countplot(data=df, x="Rating_On Leg Room Service")
plt.title("Customer Satisfaction On Leg_Room_Service", size=14)
```

```
Out[67]: Text(0.5, 1.0, 'Customer Satisfaction On Leg_Room_Service')
```



```
In [45]: ## Converting Baggage Handling into Categorical Labels
df["Rating_On_Baggage_Handling"]=pd.cut(df["Baggage handling"],[0,1,2,3,4,5],labels=["Very Bad", "Bad", "Not Ba
```

```
In [68]: sns.countplot(data=df, x="Rating_On_Baggage_Handling")
plt.title("Customer Satisfaction On Baggage Handling", size=14)
```

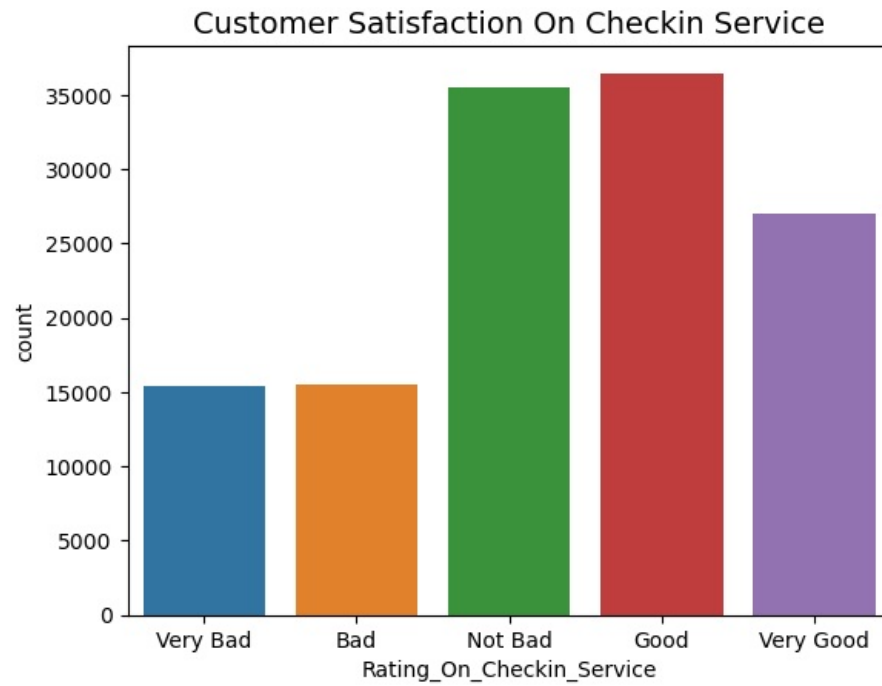
```
Out[68]: Text(0.5, 1.0, 'Customer Satisfaction On Baggage Handling')
```



```
In [46]: ## Converting Checkin Service into Categorical Labels
df["Rating_On_Checkin_Service"]=pd.cut(df["Checkin service"],[0,1,2,3,4,5],labels=["Very Bad", "Bad", "Not Bad"]
```

```
In [69]: sns.countplot(data=df, x="Rating_On_Checkin_Service")
plt.title("Customer Satisfaction On Checkin Service", size=14)
```

```
Out[69]: Text(0.5, 1.0, 'Customer Satisfaction On Checkin Service')
```



```
In [47]: ## Converting Cleanliness into Categorical Labels
df["Rating_On_Cleanliness"] = pd.cut(df["Cleanliness"], [0, 1, 2, 3, 4, 5], labels=["Very Bad", "Bad", "Not Bad", "Good"])
```

```
In [53]: sns.countplot(data=df, x="Rating_On Cleanliness")
plt.title("Customer Satisfaction On Cleanliness", size=14)
```

```
Out[53]: Text(0.5, 1.0, 'Customer Satisfaction On Cleanliness')
```



```
In [48]: ## Converting Online Boarding into Categorical Labels
df["Rating_On_Online_Boarding"] = pd.cut(df["Online boarding"], [0, 1, 2, 3, 4, 5], labels=["Very Bad", "Bad", "Not Bad"])
```

```
In [49]: sns.countplot(data=df, x="Rating_On Online Boarding")
plt.title("Customer Satisfaction On An Online Boarding", size=14)
```

```
Out[49]: Text(0.5, 1.0, 'Customer Satisfaction On An Online Boarding')
```



Clonclusions

In [143..

```
## Data shows that there are more Invistico Airline customers who are datisfied compared to dissatisfied,
## Although there is a slightly different about +/-10 of the difference

## Age group from 30-50 shows a great number of customers who are satisfied, while <30 & >60 shows to be dissat.

## Loyal Customers are greater than Disloyal Customers
## Age Group between 20-60, shows a greater number of Loyal Customers

## Invistico Airlines Customer Shows A Great Numbe of Age Group Between 20-60

## 60% of Customers Uses Invistico Airlines For Business Travel.
## 40% of Customers Uses Invistico Airlines For Personal Travel.
## Age Groups between 20-60 uses Invistico Airlines for Business Travel more compared to other age groups

## Age Group Between 30-60 years of age uses business class More,
## While Age groups of <10-70 uses Economic Class More

## Customer Satisfaction
## Invistico Airline needs to improve;
## food and drinks, a significant number of customers rated food and drinks to be very bad and bad.
## Departure/Arrival Time Convenient
## Inflight Wifi Service

## Overall Customer Satisfaction on other categories is very impressive.
```