In [39]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score,classification_report
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_excel('customer_booking.xlsx')
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   num_passengers         50000 non-null  int64  
 1   sales_channel          50000 non-null  object 
 2   trip_type              50000 non-null  object 
 3   purchase_lead          50000 non-null  int64  
 4   length_of_stay         50000 non-null  int64  
 5   flight_hour            50000 non-null  int64  
 6   flight_day             50000 non-null  object 
 7   route                  50000 non-null  object 
 8   booking_origin         50000 non-null  object 
 9   wants_extra_baggage    50000 non-null  int64  
 10  wants_preferred_seat   50000 non-null  int64  
 11  wants_in_flight_meals  50000 non-null  int64  
 12  flight_duration        50000 non-null  float64
 13  booking_complete       50000 non-null  int64  
dtypes: float64(1), int64(8), object(5)
memory usage: 5.3+ MB
```

In [6]:
```
df.describe()
```

Out[6]:

	num_passengers	purchase_lead	length_of_stay	flight_hour	wants_extra_baggage	v
count	50000.000000	50000.000000	50000.00000	50000.00000	50000.000000	
mean	1.591240	84.940480	23.04456	9.06634	0.668780	
std	1.020165	90.451378	33.88767	5.41266	0.470657	
min	1.000000	0.000000	0.00000	0.00000	0.000000	
25%	1.000000	21.000000	5.00000	5.00000	0.000000	
50%	1.000000	51.000000	17.00000	9.00000	1.000000	
75%	2.000000	115.000000	28.00000	13.00000	1.000000	
max	9.000000	867.000000	778.00000	23.00000	1.000000	

In [7]:
```
df.booking_complete.unique()
```

Out[7]:
```
array([0, 1])
```

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

Out[4]:

	num_passengers	sales_channel	trip_type	purchase_lead	length_of_stay	flight_hour	flig
0	2	Internet	RoundTrip	262	19	7	
1	1	Internet	RoundTrip	112	20	3	
2	2	Internet	RoundTrip	243	22	17	
3	1	Internet	RoundTrip	96	31	4	
4	2	Internet	RoundTrip	68	22	15	

In [12]:
```
sales_channel = pd.get_dummies(df['sales_channel'])
```

In [13]:
```
trip_type = pd.get_dummies(df['trip_type'])
```

In [14]:
```
flight_day = pd.get_dummies(df['flight_day'])
```

In [15]:
```
route = pd.get_dummies(df['route'])
```

In [16]:
```
booking_origin = pd.get_dummies(df['booking_origin'])
```

In [17]:
```
df.drop(columns=['sales_channel','trip_type','flight_day','route','bookin
```

In [18]:
```
df = pd.concat([df,sales_channel,trip_type,flight_day,route,booking_origi
```

In [20]:
```
df['booking_complete'].unique()
```

Out[20]:
```
array([0, 1])
```

In [24]:
```
len(df)
```

Out[24]:
```
50000
```

In [25]:
```
x_train,x_test,y_train,y_test = train_test_split(df.drop(columns='booking_
```

In [32]:
```
model = LogisticRegression(class_weight='balanced',solver='lbfgs',max_ite
```

In [33]:
```
model.fit(x_train,y_train)
```

Out[33]:
```
LogisticRegression(class_weight='balanced', max_iter=10000)
```

In [34]:
```
predictions = model.predict(x_test)
```

In [37]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.94      0.71      0.81     10653
           1       0.30      0.74      0.43      1847

    accuracy                           0.71     12500
   macro avg       0.62      0.72      0.62     12500
weighted avg       0.85      0.71      0.75     12500
```

In [40]:
```
fimp = model.coef_.flatten()
```

In [43]:
```
df.columns.drop('booking_complete')
```

Out[43]:
```
Index(['num_passengers', 'purchase_lead', 'length_of_stay', 'flight_hour
',
       'wants_extra_baggage', 'wants_preferred_seat', 'wants_in_flight_me
als',
       'flight_duration', 'Internet', 'Mobile',
       ...
       'Timor-Leste', 'Tonga', 'Tunisia', 'Turkey', 'Ukraine',
       'United Arab Emirates', 'United Kingdom', 'United States', 'Vanuat
u',
       'Vietnam'],
      dtype='object', length=923)
```

```
plt.figure(figsize=(16,9))
plt.barh(df.columns.drop('booking_complete'),fimp)
```