In [13]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

In [14]:
```
df = pd.read_csv('telecom_churn.csv')
df.head()
```

Out[14]:

	customer_id	telecom_partner	gender	age	state	city	pincode	date_of_registrati
0	1	Reliance Jio	F	25	Karnataka	Kolkata	755597	2020-01
1	2	Reliance Jio	F	55	Mizoram	Mumbai	125926	2020-01
2	3	Vodafone	F	57	Arunachal Pradesh	Delhi	423976	2020-01
3	4	BSNL	M	46	Tamil Nadu	Kolkata	522841	2020-01
4	5	BSNL	F	26	Tripura	Delhi	740247	2020-01

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243553 entries, 0 to 243552
Data columns (total 14 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   customer_id           243553 non-null  int64
 1   telecom_partner       243553 non-null  object
 2   gender                243553 non-null  object
 3   age                   243553 non-null  int64
 4   state                 243553 non-null  object
 5   city                  243553 non-null  object
 6   pincode               243553 non-null  int64
 7   date_of_registration  243553 non-null  object
 8   num_dependents        243553 non-null  int64
 9   estimated_salary      243553 non-null  int64
 10  calls_made            243553 non-null  int64
 11  sms_sent              243553 non-null  int64
 12  data_used             243553 non-null  int64
 13  churn                 243553 non-null  int64
dtypes: int64(9), object(5)
memory usage: 26.0+ MB
```

In [16]:
```
df.describe()
```

Out[16]:

	customer_id	age	pincode	num_dependents	estimated_salary	243
count	243553.000000	243553.000000	243553.000000	243553.000000	243553.000000	
mean	121777.000000	46.077609	549501.270541	1.997500	85021.137839	
std	70307.839393	16.444029	259808.860574	1.414941	37508.963233	
min	1.000000	18.000000	100006.000000	0.000000	20000.000000	
25%	60889.000000	32.000000	324586.000000	1.000000	52585.000000	
50%	121777.000000	46.000000	548112.000000	2.000000	84990.000000	
75%	182665.000000	60.000000	774994.000000	3.000000	117488.000000	
max	243553.000000	74.000000	999987.000000	4.000000	149999.000000	

In [17]:
```
df.shape
```

Out[17]:
```
(243553, 14)
```

In [18]:
```
df.isna().sum()
```

Out[18]:
```
customer_id             0
telecom_partner         0
gender                  0
age                     0
state                   0
city                    0
pincode                 0
date_of_registration    0
num_dependents          0
estimated_salary        0
calls_made              0
sms_sent                0
data_used               0
churn                   0
dtype: int64
```

In [19]:
```
df.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
df.columns
```

Out[20]:
```
Index(['customer_id', 'telecom_partner', 'gender', 'age', 'state', 'city',
       'pincode', 'date_of_registration', 'num_dependents', 'estimated_sal
ary',
       'calls_made', 'sms_sent', 'data_used', 'churn'],
      dtype='object')
```

In [21]:
```
df.drop(['customer_id','state','city','pincode','telecom_partner','date_of_
df.head()
```

Out[21]:

	gender	age	num_dependents	estimated_salary	calls_made	sms_sent	data_used	churn
0	F	25	4	124962	44	45	-361	0
1	F	55	2	130556	62	39	5973	0
2	F	57	0	148828	49	24	193	1
3	M	46	1	38722	80	25	9377	1
4	F	26	2	55098	78	15	1393	0

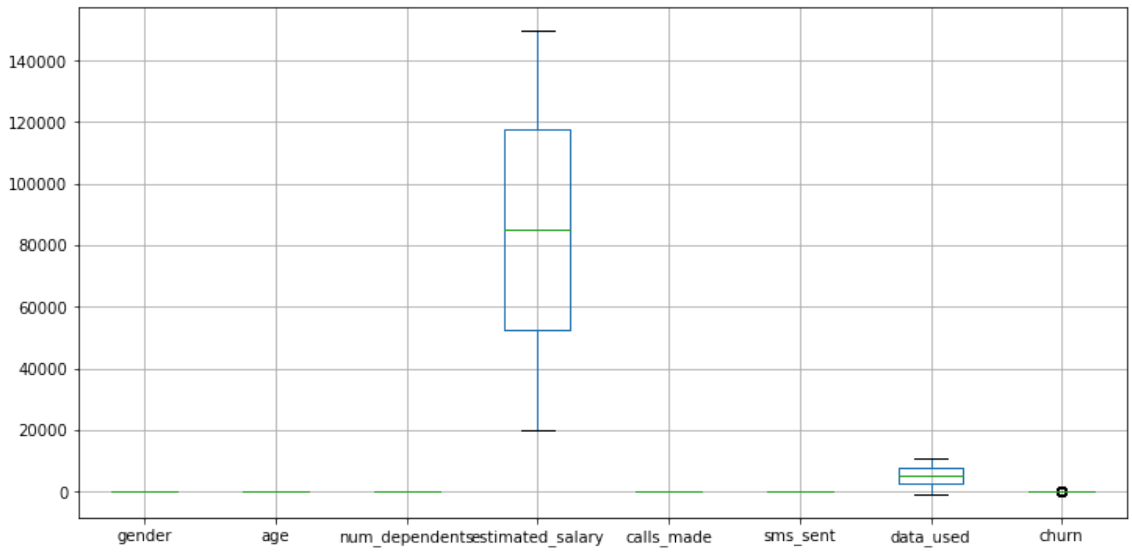
In [22]:
```
le = LabelEncoder()
df['gender'] = le.fit_transform(df['gender'])

df.head()
```

Out[22]:

	gender	age	num_dependents	estimated_salary	calls_made	sms_sent	data_used	churn
0	0	25	4	124962	44	45	-361	0
1	0	55	2	130556	62	39	5973	0
2	0	57	0	148828	49	24	193	1
3	1	46	1	38722	80	25	9377	1
4	0	26	2	55098	78	15	1393	0

In [24]:
```
plt.figure(figsize=(12,6))
df.boxplot()
plt.show()
```



In [25]:
```
df.dtypes
```

Out[25]:
```
gender              int32
age                 int64
num_dependents      int64
estimated_salary    int64
calls_made          int64
sms_sent            int64
data_used           int64
churn               int64
dtype: object
```

In [28]:
```
X = df.drop(columns = ['churn'])
y = df['churn']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
```

In [30]:
```
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [31]:
```
df.to_csv('Cleaned_Telecom_Customer_Churn.csv',index=False)
```

In [ ]:
```

```