In [1]:

```
#3 GENDER SUBMISSION
import re
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import metrics
%matplotlib inline
```

In [2]:

```
df=pd.read_csv(r"C:\Users\DHEEPAK\Desktop\gender_submission.csv")
print(df)
```

```
     PassengerId  Survived
0            892         0
1            893         1
2            894         0
3            895         0
4            896         1
..           ...       ...
413         1305         0
414         1306         1
415         1307         0
416         1308         0
417         1309         0

[418 rows x 2 columns]
```

In [3]:

```
plt.figure(figsize=(20,2))
```

Out[3]:

```
<Figure size 2000x200 with 0 Axes>
```

```
<Figure size 2000x200 with 0 Axes>
```

In [4]:

```
df.describe()
```

Out[4]:

	PassengerId	Survived
count	418.000000	418.000000
mean	1100.500000	0.363636
std	120.810458	0.481622
min	892.000000	0.000000
25%	996.250000	0.000000
50%	1100.500000	0.000000
75%	1204.750000	1.000000
max	1309.000000	1.000000

In [5]:

```
df.isnull().any()
```

Out[5]:

```
PassengerId    False
Survived       False
dtype: bool
```

In [6]:

```
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
print('This dataframe has %d rows and %d columns'%(df.shape))
```

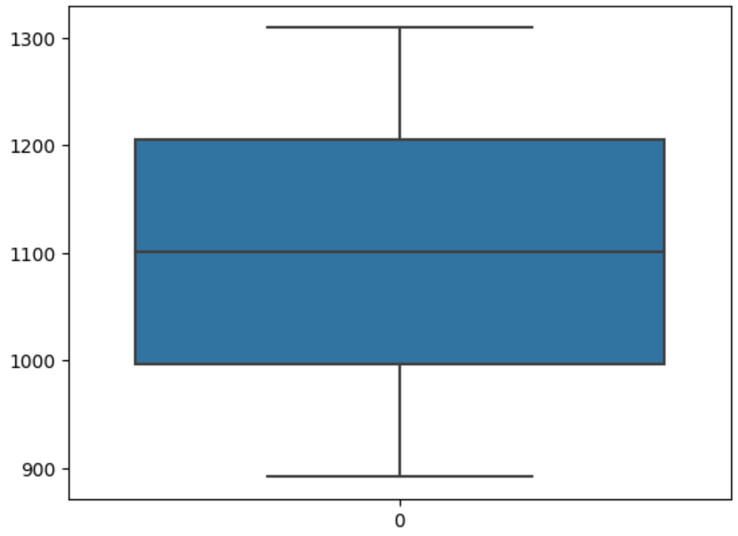
```
This dataframe has 418 rows and 2 columns
```

In [7]:

```
sns.boxplot(df['PassengerId'])
```

Out[7]:

```
<Axes: >
```



In [8]:

```
df.head(10)
```

Out[8]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1
5	897	0
6	898	1
7	899	0
8	900	1
9	901	0

In [9]:

```
features_matrix=df.iloc[:,0:1]
```

In [10]:

```
target_vector=df.iloc[:,-1]
```

In [11]:

```
print('The features matrix has %d rows and %d columns'%(features_matrix.shape))
```

```
The features matrix has 418 rows and 1 columns
```

In [12]:

```
print('The target matrix has %d rows and %d columns'%(np.array(target_vector).reshape(-1,1).shape))
```

```
The target matrix has 418 rows and 1 columns
```

In [13]:

```
from sklearn.preprocessing import StandardScaler
features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

In [17]:

```
algorithm=LogisticRegression(penalty='l2',dual=False,tol=1e-4,C=1.0,fit_intercept=True,intercept_scaling=1,class_weight=None,random_state
```

In [18]:

```
Logistic_Regression_Model=algorithm.fit(features_matrix_standardized,target_vector)
```

In [19]:

```
observation=[[1]]
```

In [20]:

```
predictions=Logistic_Regression_Model.predict(observation)
```

In [21]:

```
print('The model predicted the observation to belong to class %d'%(predictions))
```

```
The model predicted the observation to belong to class 0
```

In [25]:

```
print('The algorithm was trained to predict one of the two classes :'%(algorithm.classes_))
```

```
The algorithm was trained to predict one of the two classes :
```

In [26]:

```
he model says the probability of the observation we passed belonging to class['0'] is %s"""%(algorithm.predict_proba(observation)[0][0]))
```

```
The model says the probability of the observation we passed belonging to class['0'] is 0.6474324251144166
```

In [27]:

```
he model says the probability of the observation we passed belonging to class['1'] is %s"""%(algorithm.predict_proba(observation)[0][1]))
```

```
The model says the probability of the observation we passed belonging to class['1'] is 0.35256757488558343
```

In [ ]: