In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
# load titanic dataset
url = r"C:\Users\jayes\OneDrive\Desktop\NareshIT\3_apr\3th- EDA Automation Mistral, gradio\3th- EDA Automation Mistra
df = pd.read_csv(url)
df
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [3]:

```
print(df.describe())
```

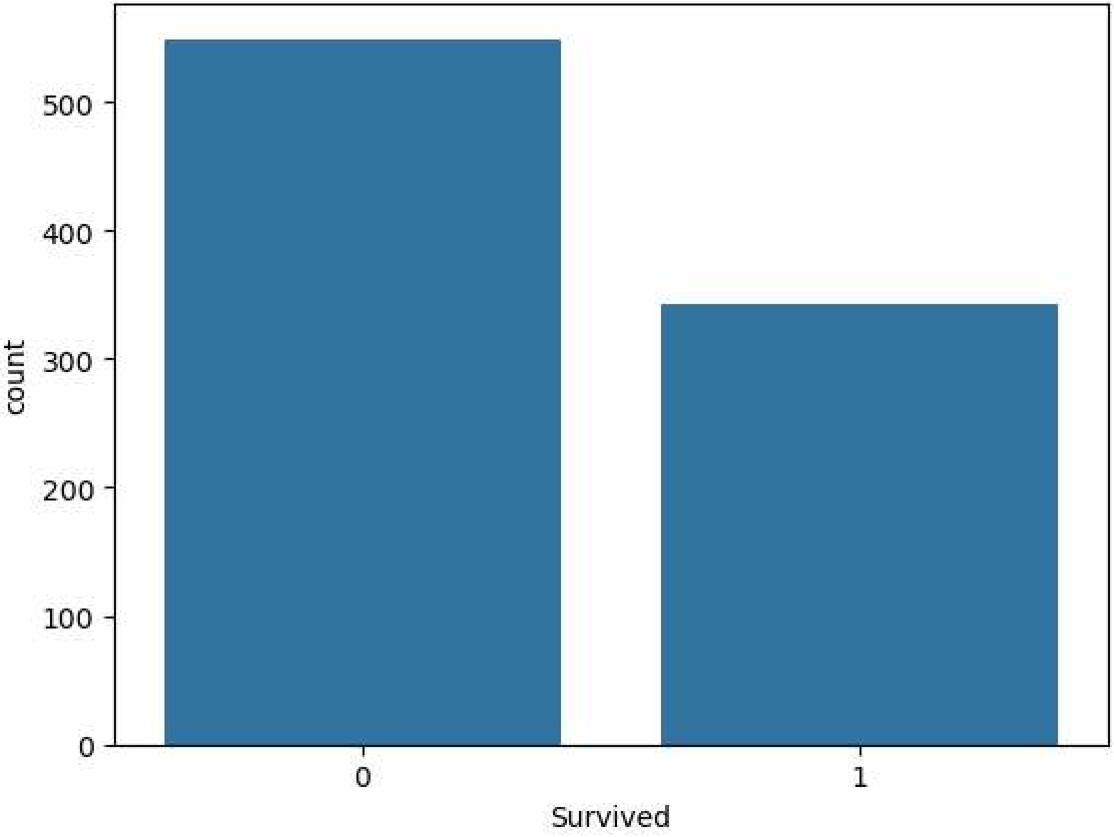
```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000   
mean    446.000000    0.383838    2.308642   29.699118    0.523008   
std     257.353842    0.486592    0.836071   14.526497    1.102743   
min       1.000000    0.000000    1.000000    0.420000    0.000000   
25%     223.500000    0.000000    2.000000   20.125000    0.000000   
50%     446.000000    0.000000    3.000000   28.000000    0.000000   
75%     668.500000    1.000000    3.000000   38.000000    1.000000   
max     891.000000    1.000000    3.000000   80.000000    8.000000   

            Parch        Fare  
count  891.000000  891.000000  
mean     0.381594   32.204208  
std      0.806057   49.693429  
min      0.000000    0.000000  
25%      0.000000    7.910400  
50%      0.000000   14.454200  
75%      0.000000   31.000000  
max      6.000000  512.329200  
```

In [4]:
```python
print('\n Missing values :\n', df.isnull().sum())
```

```
 Missing values :
 PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [5]:
```python
# survival rate visulisation
sns.countplot(x='Survived', data = df)
plt.title = 'survival count'
plt.show()
```



In [6]:

```
import ollama

def generate_insights(df_summary):
    prompt = f"analys data set summary and provide insights : \n\n{df_summary}"
    response = ollama.chat(model="deepseek-coder", messages=[{'role':'user','content' : prompt}])
    return response['message']['content']

#generate AI insights
summary = df.describe().to_string()
insights = generate_insights(summary)
print('\n AI generated insights: \n', insights)
```

```
 AI generated insights:
 The given dataset consists of information about passengers in various airline passenger transport flights, such as t
heir age and class (class is a factor that could be important for differentiating between classes), the number of sib
lings/spouses aboard PassengerId(SibSp) or parent children at home Parch. The fare they paid would also influence if 
someone survived in this dataset from 'Fare'.

Here's an analysis summary:
- There are a total count (891 samples), mean of all values, standard deviation for each feature and min/max value ra
nge among other information about the data set.
   - Mean Age is around average age as they were aboard during their flight in 'Age'. This could be important to know
if we should aim at predicting survival rates based on this variable or not (assuming that passengers above an averag
e of 30-45 years old are more likely).
   - Mean Fare indicates the cost paid by Passengers. It can potentially help us understand where higher fares were i
n relation to class, as well a direct correlation between fare and survival rate since high fares may have led passan
gers away from being aboard (assuming that passengers who pay more had lower chances of surviving).
   - The other features such Age(Pclass) also contribute significantly but with varying effects on Survival Rate. 4th
class passenger could be a higher risk to survive than the first three classes as they are likely older and paid less
fare for same distance traveled, if any (assuming that those who pay more would have been aboard in lower class).
- There is also no missing data points which should generally not exist. However here we do see 'SibSp' & Parch', it
might indicate a passenger had siblings/spouses or parents as well; however, they didn’t contribute to the survival r
ate (assuming their presence in airplane resulted in higher chance of surviving).
- The dataset is highly imbalanced with only 891 samples for survived. So you would need an extensive analysis and po
ssibly a data cleansing process before using this as your primary predictor model due that it represents about one fo
urth or less than half the survival rate when we consider all passengers, which may affect performance of algorithms 
such as Logistic Regression if not handled properly (assuming class imbalance).
- The dataset could be further analyzed by visualizing data via histograms and box plots to better understand distrib
ution pattern. A correlation matrix can also provide additional insight into the relationship between different varia
bles in relation with survival rate, like how fare is affected or passenger’s siblings/spouses are related etc..  (as
suming all factors could be predictors).
```

In [7]:

```python
import gradio as gr

def eda_analysis(file):
    df = pd.read_csv(file.name)
    summary = df.describe().to_string()
    insights = generate_insights(summary)
    return insights

# Create Web Interface
demo = gr.Interface(fn=eda_analysis, inputs="file", outputs="text", title="AI-Powered EDA with deepseek-coder")
```

```python
# Launch App
demo.launch(share=True)  # Use share=True for Google Colab
```

```
* Running on local URL:  http://127.0.0.1:7861
* Running on public URL: https://f3c02d977073f68941.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from the termin
al in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)
```

Out[7]:

In [ ]: