

```
In [76]: 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.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
```

```
In [77]: from matplotlib import rcParams
rcParams['figure.figsize'] = 15, 5 # this controls the size of graphs
sns.set_style('darkgrid')
```

```
In [78]: titanic = pd.read_csv('train.csv')
titanic.head()
```

```
Out[78]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

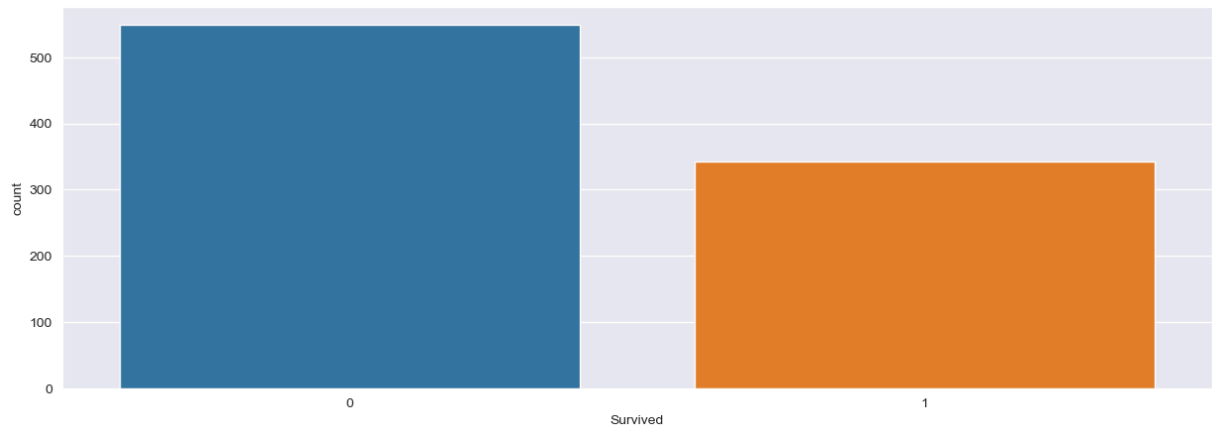


```
In [79]: titanic.describe()
```

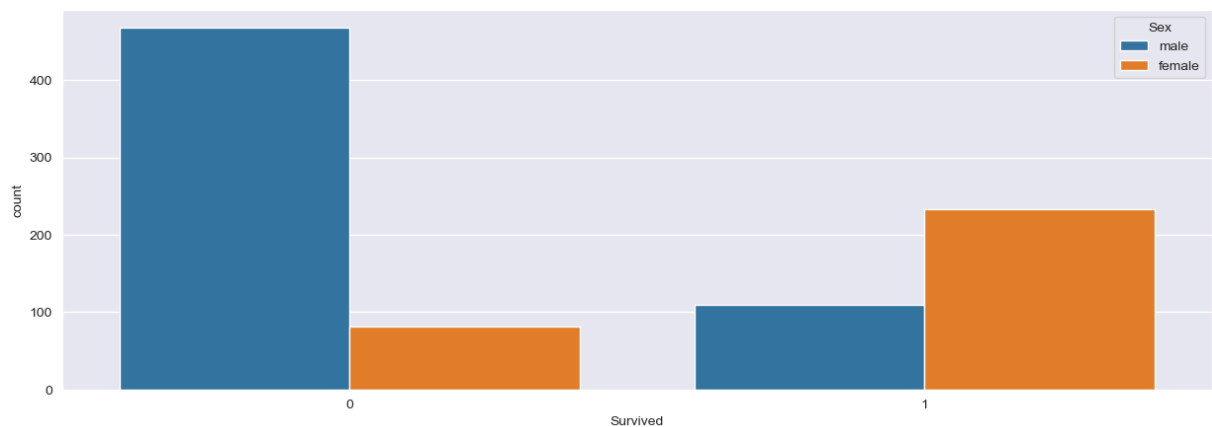
Out[79]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
In [80]: sns.countplot(x=titanic['Survived'])
plt.show()
```



```
In [81]: sns.countplot(x=titanic['Survived'], hue='Sex', data=titanic)
plt.show()
```



```
In [83]: sns.countplot(x='Survived', hue='Pclass', data=titanic)

# Show the plot
plt.title('Survival Counts by Pclass')
```

```
plt.xlabel('Survived')  
plt.ylabel('Count')  
plt.legend(title='Pclass')  
plt.show()
```

```

-----
AttributeError                                Traceback (most recent call last)
Cell In[83], line 1
----> 1 sns.countplot(x='Survived', hue='Pclass', data=titanic)
      2 # Show the plot
      3 plt.title('Survival Counts by Pclass')

File c:\Users\thinh\anaconda3\Lib\site-packages\seaborn\categorical.py:2955, in count
plot(data, x, y, hue, order, hue_order, orient, color, palette, saturation, width,
dodge, ax, **kwargs)
    2952 if ax is None:
    2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
    2956 return ax

File c:\Users\thinh\anaconda3\Lib\site-packages\seaborn\categorical.py:1587, in _Bar
Plotter.plot(self, ax, bar_kws)
    1585 """Make the plot."""
    1586 self.drawBars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
    1588 if self.orient == "h":
    1589     ax.invert_yaxis()

File c:\Users\thinh\anaconda3\Lib\site-packages\seaborn\categorical.py:767, in _Cate
goricalPlotter.annotate_axes(self, ax)
    764 ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
    766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

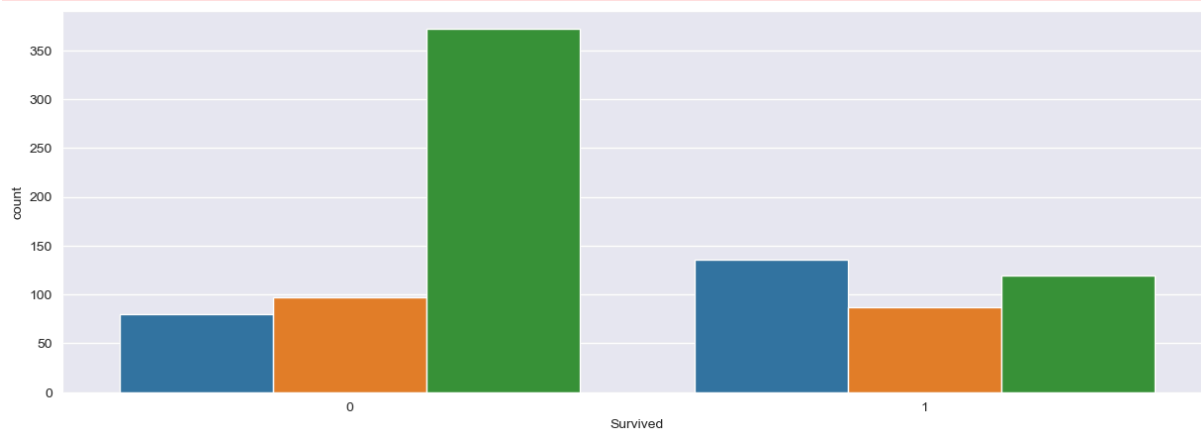
File c:\Users\thinh\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:322, in Axe
s.legend(self, *args, **kwargs)
    204 @_docstring.dedent_interpd
    205 def legend(self, *args, **kwargs):
    206     """
    207     Place a legend on the Axes.
    208
    (...)
    320     .. plot:: gallery/text_labels_and_annotations/legend.py
    321     """
--> 322     handles, labels, kwargs = mlegend._parse_legend_args([self], *args, **kw
args)
    323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    324     self.legend._remove_method = self._remove_legend

File c:\Users\thinh\anaconda3\Lib\site-packages\matplotlib\legend.py:1361, in _parse
_legend_args(axs, handles, labels, *args, **kwargs)
    1357 handles = [handle for handle, label
    1358               in zip(_get_legend_handles(axs, handlers), labels)]
    1360 elif len(args) == 0: # 0 args: automatically detect labels and handles.
-> 1361     handles, labels = _get_legend_handles_labels(axs, handlers)
    1362     if not handles:
    1363         log.warning(
    1364             "No artists with labels found to put in legend. Note that "
    1365             "artists whose label start with an underscore are ignored "
    1366             "when legend() is called with no argument.")

```

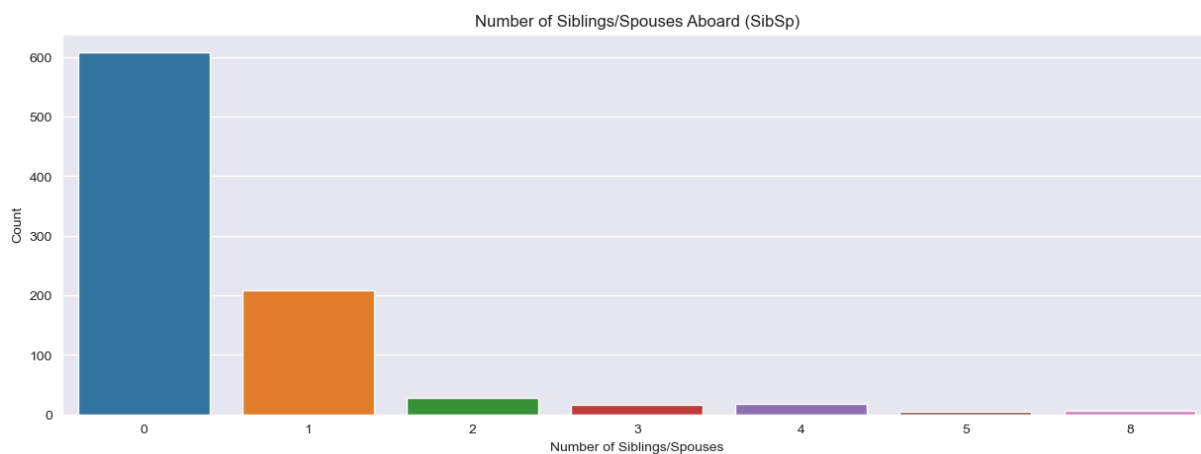
```
File c:\Users\thinh\anaconda3\Lib\site-packages\matplotlib\legend.py:1291, in _get_1
legend_handles_labels(axs, legend_handler_map)
    1289 for handle in _get_legend_handles(axs, legend_handler_map):
    1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
    1292         handles.append(handle)
    1293         labels.append(label)
```

AttributeError: 'numpy.int64' object has no attribute 'startswith'



```
In [84]: sns.countplot(x='SibSp', data=titanic)

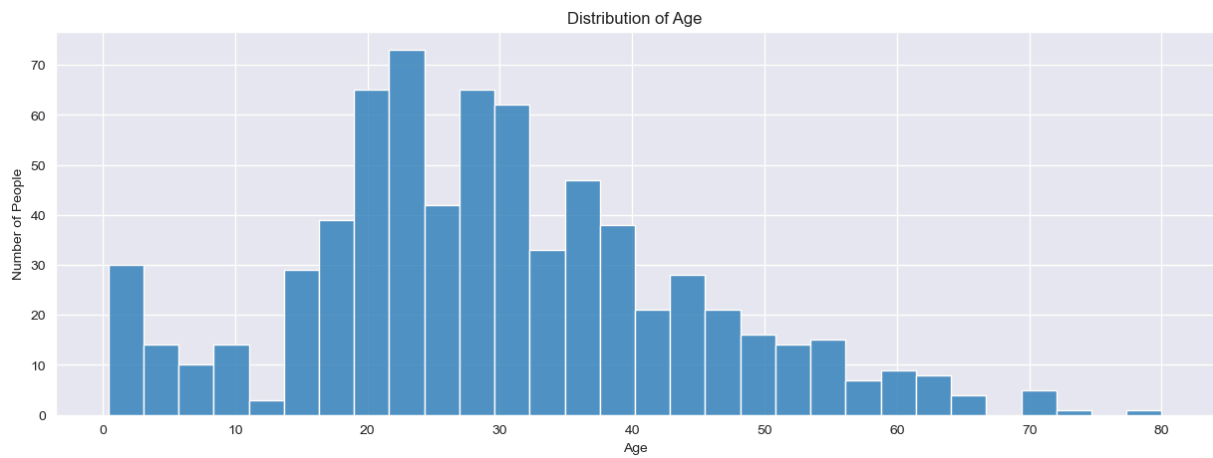
# Show the plot
plt.title('Number of Siblings/Spouses Aboard (SibSp)')
plt.xlabel('Number of Siblings/Spouses')
plt.ylabel('Count')
plt.show()
```



```
In [85]: sns.histplot(titanic['Age'].dropna(), kde=False, bins=30)

# Show the plot
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Number of People')
plt.show()

# Use the describe() command on the Age column, dropping null values
titanic['Age'].dropna().describe()
```

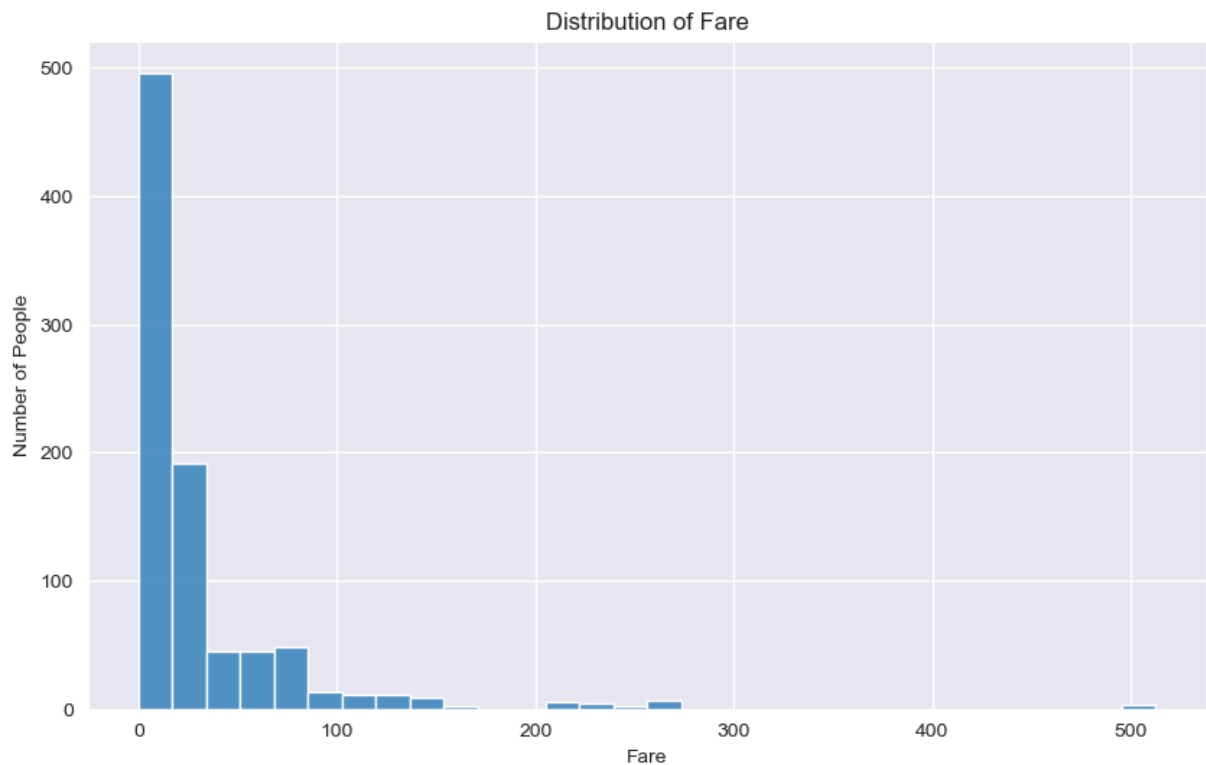


```
Out[85]: count    714.000000
         mean      29.699118
         std       14.526497
         min        0.420000
         25%       20.125000
         50%       28.000000
         75%       38.000000
         max       80.000000
         Name: Age, dtype: float64
```

```
In [86]: plt.figure(figsize=(10, 6))
         sns.histplot(titanic['Fare'].dropna(), kde=False, bins=30)

         # Show the plot
         plt.title('Distribution of Fare')
         plt.xlabel('Fare')
         plt.ylabel('Number of People')
         plt.show()

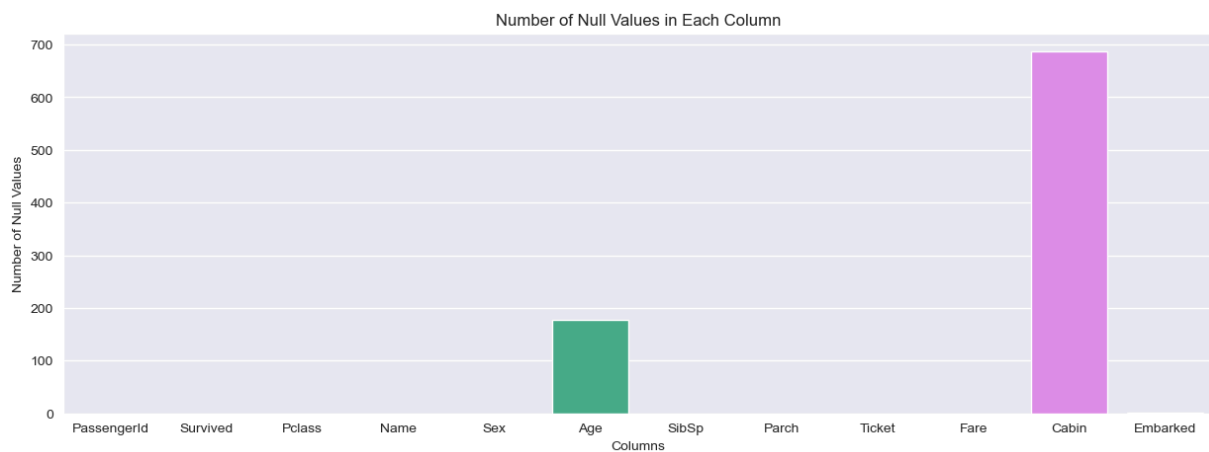
         # Use the describe() command on the Fare column, dropping null values
         titanic['Fare'].dropna().describe()
```



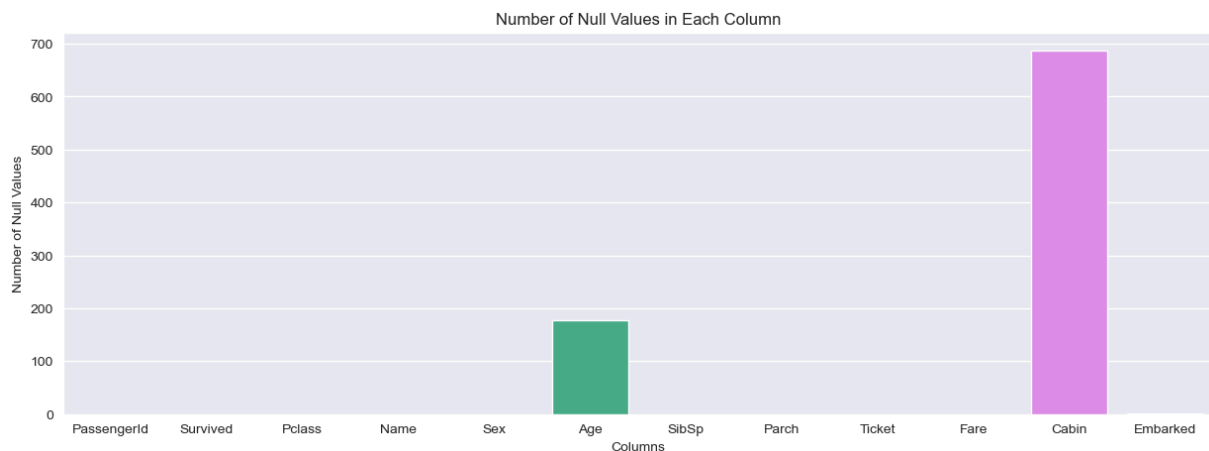
```
Out[86]: count    891.000000
         mean     32.204208
         std      49.693429
         min       0.000000
         25%       7.910400
         50%      14.454200
         75%      31.000000
         max      512.329200
         Name: Fare, dtype: float64
```

```
In [68]: sns.barplot(x=titanic.columns, y=titanic.isnull().sum())

# Show the plot
plt.title('Number of Null Values in Each Column')
plt.xlabel('Columns')
plt.ylabel('Number of Null Values')
plt.show()
```



```
In [69]: sns.barplot(x=titanic.columns, y=titanic.isnull().sum().values)
plt.title('Number of Null Values in Each Column')
plt.xlabel('Columns')
plt.ylabel('Number of Null Values')
plt.show()
```



```
In [70]: titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    object
4   Sex          891 non-null    object
5   Age          714 non-null    float64
6   SibSp        891 non-null    int64
7   Parch        891 non-null    int64
8   Ticket       891 non-null    object
9   Fare         891 non-null    float64
10  Cabin        204 non-null    object
11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
In [71]: titanic = titanic.drop(['Cabin'], axis=1)
titanic.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    object
4   Sex          891 non-null    object
5   Age          714 non-null    float64
6   SibSp        891 non-null    int64
7   Parch        891 non-null    int64
8   Ticket       891 non-null    object
9   Fare         891 non-null    float64
10  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

```
In [72]: titanic = pd.get_dummies(titanic, columns=['Sex', 'Embarked'], drop_first=True)
titanic.head()
```

Out[72]:

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Sex_m
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	T
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	PC 17599	71.2833	F
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	F
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	F
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	T

```
In [73]: titanic = titanic.drop(['PassengerId', 'Name', 'Ticket'], axis=1)
```

```
In [74]: titanic = titanic.dropna()

# Split the data into X (features) and y (target)
```

```
X = titanic.drop('Survived', axis=1)
y = titanic['Survived']
```

In [75]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

```
# Initialize the Logistic Regression model
log_reg = LogisticRegression(max_iter=1000)

# Train the model
log_reg.fit(X_train, y_train)

# Make predictions
y_pred = log_reg.predict(X_test)

# Evaluate the model
print("Classification Report:")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Classification Report:

	precision	recall	f1-score	support
0	0.77	0.83	0.80	126
1	0.72	0.65	0.69	89
accuracy			0.75	215
macro avg	0.75	0.74	0.74	215
weighted avg	0.75	0.75	0.75	215

Confusion Matrix:

```
[[104  22]
 [ 31  58]]
```