EDA on titanic data set

In [2]:
```
!pip install pyforest
```

```
Collecting pyforest
  Downloading pyforest-1.1.0.tar.gz (15 kB)
Building wheels for collected packages: pyforest
  Building wheel for pyforest (setup.py): started
  Building wheel for pyforest (setup.py): finished with status 'done'
  Created wheel for pyforest: filename=pyforest-1.1.0-py2.py3-none-any.whl size=14607 sh
a256=b0be22dca46381029ed0f8c0cfd596ca2e09d622658d1a6bc6455fcb9599cb60
  Stored in directory: c:\users\vikin\appdata\local\pip\cache\wheels\d5\1a\3e\6193fe1c56
168f5df4aef57d8411033ba4611881135d495727
Successfully built pyforest
Installing collected packages: pyforest
Successfully installed pyforest-1.1.0
```

In [4]:
```
import pyforest
```

In [5]:
```
data=pd.read_csv('Titanic-Train-Data.csv')
```

In [6]:
```
data
```

Out[6]:

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

891 rows × 12 columns

In [8]:
```
data.shape
```

Out[8]:
```
(891, 12)
```

In [9]:
```
data.isna().sum()
```

Out[9]:
```
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 [10]:
```
data.describe()
```

Out[10]:

	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 [11]:
```
data.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 [13]:
```
data.dtypes
```

Out[13]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [22]:
```python
from sklearn.preprocessing import LabelEncoder
```

In [24]:
```python
from sklearn import preprocessing

label_encoder=preprocessing.LabelEncoder()

data['Sex']= label_encoder.fit_transform(data['Sex'])

data['Sex'].value_counts()
```

Out[24]:
```
1    577
0    314
Name: Sex, dtype: int64
```

In [25]:
```python
data
```

Out[25]:

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

891 rows × 12 columns

In [26]:

```
data=data.drop(['Ticket','Name','Cabin'],axis=1)
data
```

Out[26]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	S
1	2	1	1	0	38.0	1	0	71.2833	C
2	3	1	3	0	26.0	0	0	7.9250	S
3	4	1	1	0	35.0	1	0	53.1000	S
4	5	0	3	1	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
886	887	0	2	1	27.0	0	0	13.0000	S
887	888	1	1	0	19.0	0	0	30.0000	S
888	889	0	3	0	NaN	1	2	23.4500	S
889	890	1	1	1	26.0	0	0	30.0000	C
890	891	0	3	1	32.0	0	0	7.7500	Q

891 rows × 9 columns

In [28]:
```
data['Age'].median()
```

Out[28]:
```
28.0
```

In [32]:
```
data['Age']=data['Age'].fillna(value=28)
data
```

Out[32]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	S
1	2	1	1	0	38.0	1	0	71.2833	C
2	3	1	3	0	26.0	0	0	7.9250	S
3	4	1	1	0	35.0	1	0	53.1000	S
4	5	0	3	1	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
886	887	0	2	1	27.0	0	0	13.0000	S
887	888	1	1	0	19.0	0	0	30.0000	S
888	889	0	3	0	28.0	1	2	23.4500	S
889	890	1	1	1	26.0	0	0	30.0000	C
890	891	0	3	1	32.0	0	0	7.7500	Q

891 rows × 9 columns

In [34]:
```
data['Age'].isna().sum()
```

Out[34]:
```
0
```

In [35]:
```
data.isna().sum()
```

Loading [MathJax]/extensions/Safe.js

Out[35]:
```
PassengerId    0
Survived       0
Pclass         0
Sex            0
Age            0
SibSp          0
Parch          0
Fare           0
Embarked       2
dtype: int64
```

In [36]:
```
data['Embarked'].value_counts()
```

Out[36]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [38]:
```
g=data.groupby('Survived')
g['Embarked'].value_counts()
```

Out[38]:
```
Survived  Embarked
0         S           427
          C            75
          Q            47
1         S           217
          C            93
          Q            30
Name: Embarked, dtype: int64
```

In [42]:
```
data['Embarked']=data['Embarked'].fillna(value='S')
```

In [43]:
```
data
```

Out[43]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	S
1	2	1	1	0	38.0	1	0	71.2833	C
2	3	1	3	0	26.0	0	0	7.9250	S
3	4	1	1	0	35.0	1	0	53.1000	S
4	5	0	3	1	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
886	887	0	2	1	27.0	0	0	13.0000	S
887	888	1	1	0	19.0	0	0	30.0000	S
888	889	0	3	0	28.0	1	2	23.4500	S
889	890	1	1	1	26.0	0	0	30.0000	C
890	891	0	3	1	32.0	0	0	7.7500	Q

891 rows × 9 columns

In [46]:
```
from sklearn.preprocessing import LabelEncoder

from sklearn import preprocessing

label_encoder=preprocessing.LabelEncoder()

data['Embarked']=label_encoder.fit_transform(data['Embarked'])
```

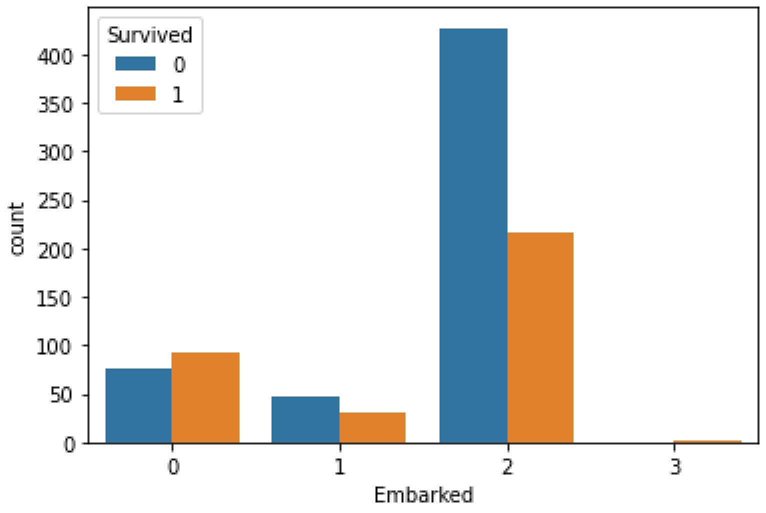
Loading [MathJax]/extensions/Safe.js

```
data['Embarked'].value_counts()
```

Out[46]:
```
2    644
0    168
1     77
3      2
Name: Embarked, dtype: int64
```

In [47]:
```
sns.countplot(data['Embarked'],hue=data['Survived'])
plt.show()
```

```
C:\Users\vikin\anaconda3\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



In [49]:
```
data['Embarked'].value_counts()
```

Out[49]:
```
2    644
0    168
1     77
3      2
Name: Embarked, dtype: int64
```

In [50]:
```
data
```

Loading [MathJax]/extensions/Safe.js

Out[50]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	2
1	2	1	1	0	38.0	1	0	71.2833	0
2	3	1	3	0	26.0	0	0	7.9250	2
3	4	1	1	0	35.0	1	0	53.1000	2
4	5	0	3	1	35.0	0	0	8.0500	2
...	...	...	...	...	...	...	...	...	...
886	887	0	2	1	27.0	0	0	13.0000	2
887	888	1	1	0	19.0	0	0	30.0000	2
888	889	0	3	0	28.0	1	2	23.4500	2
889	890	1	1	1	26.0	0	0	30.0000	0
890	891	0	3	1	32.0	0	0	7.7500	1

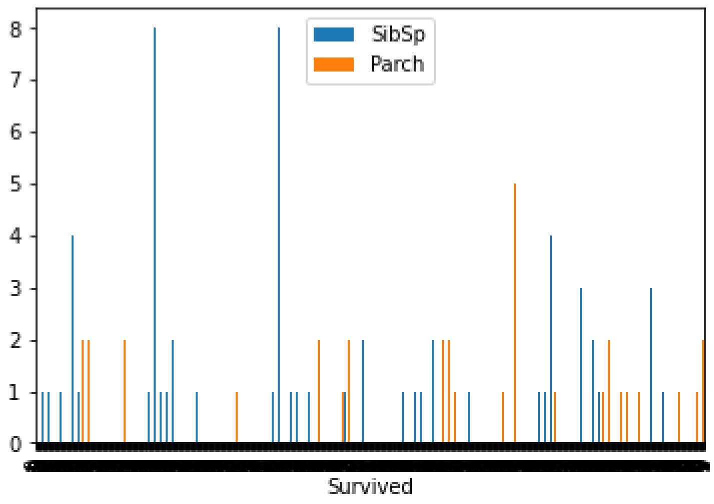
891 rows × 9 columns

In [51]:
```python
data.corr()
```

Out[51]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarke
PassengerId	1.000000	-0.005007	-0.035144	0.042939	0.034212	-0.057527	-0.001652	0.012658	0.01308
Survived	-0.005007	1.000000	-0.338481	-0.543351	-0.064910	-0.035322	0.081629	0.257307	-0.16351
Pclass	-0.035144	-0.338481	1.000000	0.131900	-0.339898	0.083081	0.018443	-0.549500	0.15711
Sex	0.042939	-0.543351	0.131900	1.000000	0.081163	-0.114631	-0.245489	-0.182333	0.10405
Age	0.034212	-0.064910	-0.339898	0.081163	1.000000	-0.233296	-0.172482	0.096688	-0.01420
SibSp	-0.057527	-0.035322	0.083081	-0.114631	-0.233296	1.000000	0.414838	0.159651	0.06665
Parch	-0.001652	0.081629	0.018443	-0.245489	-0.172482	0.414838	1.000000	0.216225	0.03832
Fare	0.012658	0.257307	-0.549500	-0.182333	0.096688	0.159651	0.216225	1.000000	-0.22122
Embarked	0.013083	-0.163517	0.157112	0.104057	-0.014205	0.066654	0.038322	-0.221226	1.00000

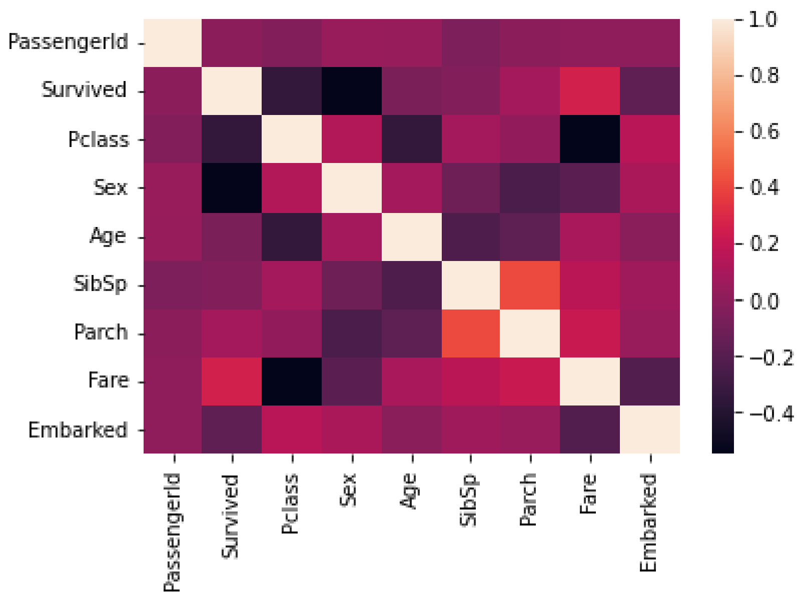
In [57]:
```python
data.plot(x="Survived", y=['SibSp','Parch'], kind='bar')
plt.show()
```



In [58]:
```python
sns.heatmap(data.corr())
```

Loading [MathJax]/extensions/Safe.js

Out[58]: <AxesSubplot:>



In [60]:
```
data['Family']=data['SibSp']+data['Parch']+1
data=data.drop(['SibSp','Parch'],axis=1)
data=data.drop('Embarked',axis=1)
data=data.drop('PassengerId',axis=1)
data
```

Out[60]:

	Survived	Pclass	Sex	Age	Fare	Family
0	0	3	1	22.0	7.2500	2
1	1	1	0	38.0	71.2833	2
2	1	3	0	26.0	7.9250	1
3	1	1	0	35.0	53.1000	2
4	0	3	1	35.0	8.0500	1
...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	1
887	1	1	0	19.0	30.0000	1
888	0	3	0	28.0	23.4500	4
889	1	1	1	26.0	30.0000	1
890	0	3	1	32.0	7.7500	1

891 rows × 6 columns

In [61]:
```
x=data.drop('Survived',axis=1).values
y=data['Survived'].values
```

In [62]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [64]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=100)
```

In [65]:
```
from sklearn.metrics import accuracy_score
```

In [68]:
```
lr=LogisticRegression()
```

Loading [MathJax]/extensions/Safe.js

```
n,y_train)
```

```
lrpred=lr.predict(x_test)
```

In [70]:
```
accuracy_score(y_test,lrpred)
```

Out[70]:
```
0.7910447761194029
```

In [72]:
```
from sklearn.model_selection import GridSearchCV
```

In [74]:
```
c_space=np.logspace(-5,8,15)
param_grid={'C':c_space}

logreg_cv=GridSearchCV(lr,param_grid,cv=5)

logreg_cv.fit(x_train,y_train)


print('T L R P:{}',format(logreg_cv.best_params_))

print('the best score is{}',format(logreg_cv.best_score_))
```

```
T L R P:{} {'C': 31.622776601683793}
the best score is{} 0.8042193548387097
```

In [ ]:
```
```

Loading [MathJax]/extensions/Safe.js