

KOLLIPARA SAI NAGA PRANATHI

21BCE8008 (VIT-AP)

import libraries

In [1]:

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

load dataset

In [2]:

```
dataset=pd.read_csv("Titanic-Dataset.csv")
dataset
```

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]:

```
dataset.head()
```

Out[3]:

	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

In [4]:

```
dataset.shape
```

Out[4]: (891, 12)

In [5]:

```
dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
```

		# Non-Null	Count	Column 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			
	889	non-null		object	Embarked		
memory	usage:	83.7+	KB		dtypes:	float64(2), int64(5),	object(5)

```
In [6]: dataset.describe()
```

Out[6]:

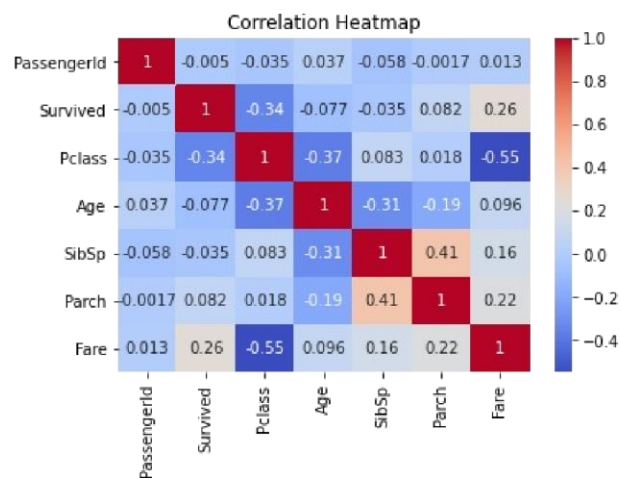
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	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 [7]: corr=dataset.corr()  
corr
```

Out[7]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
In [8]: corr_matrix = dataset.corr()  
sns.heatmap(corr_matrix, annot=True,cmap='coolwarm')  
plt.title('Correlation Heatmap')  
plt.show()
```



```
In [9]: dataset.Survived.value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
dataset.isnull().any()
```

In [10]: **handling null values**

```
Out[10]: PassengerId      False
Survived      False
Pclass        False
Name          False
Sex           False
Age           False
SibSp         True
Parch         False
Ticket        False
Fare          False
Cabin         False
dtype: bool
```

```
dataset.isnull().sum()
```

bool

```
In [11]:
```

```
Out[11]: PassengerId      0
Survived      0
Pclass        0
Name          0
Sex           0
Age           0
SibSp         177
Parch         0
Ticket        0
Fare          0
Cabin         0
dtype: int64
```

```
#median method
dataset['Age'].fillna(dataset['Age'].median(), inplace=True)
```

```
687 Embarked
dtype: int64
```

Null values are present in age, cabin and embarked that we have to handle.

```
#imputing data
```

```
dataset['Cabin'].fillna('Unknown', inplace=True)
```

```
#mode      method
dataset['Embarked'].fillna(dataset['Embarked'].mode()[0], inplace=True)
```

```
dataset['Embarked'].fillna(dataset['Embarked'].mode()[0], inplace=True)
```

In [12]:

`dataset.isnull().sum()` # null values are handled successfully

null values are handled successfully

In [13]:

Survived		0				
Pclass				0		
Name						
Sex	0					
Age	0					
SibSp	0					0
Parch						0
Ticket				0		
Fare						
Cabin	0					

0

C

C

In [14]:

```
0 Embarked
0 dtype: int64
```

```
0 dtype: int64
```

In [15]:

```
Out[15]:
```

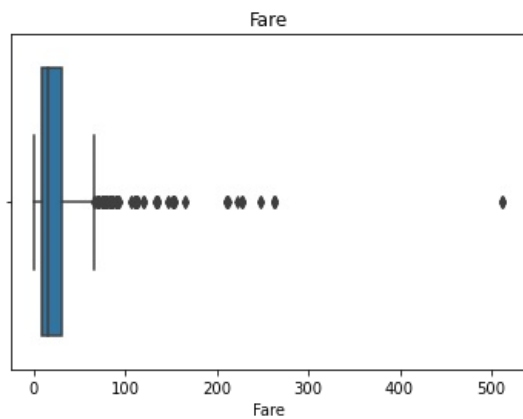
0

0

C

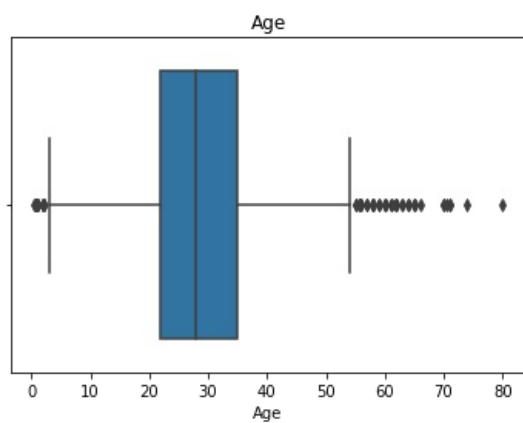
outlier detection and handling

In [16]:



```
sns.boxplot(data=dataset, x='Fare')  
plt.title("Fare") plt.show()
```

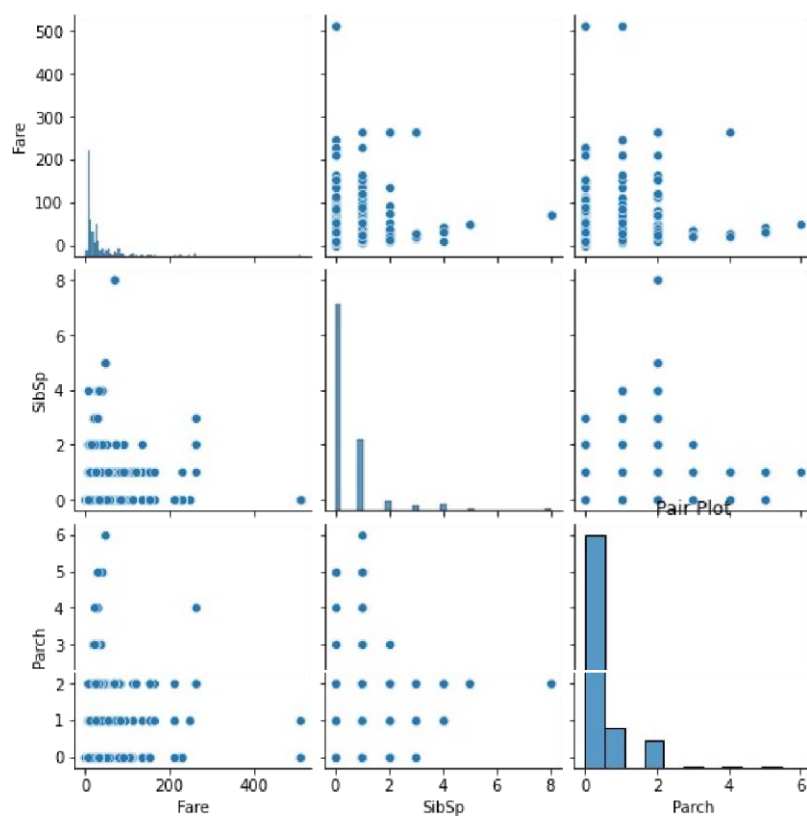
In [17]:



```
sns.boxplot(data=dataset, x='Age')  
plt.title("Age") plt.show()
```

In [18]:

```
#Pair plot for selected numerical  
sns.pairplot(data=dataset[['Fare', 'SibSp', 'Parch']])  
plt.title('Pair Plot')  
plt.show()
```



```
Outliers from scipy import stats

z_scores = np.abs(stats.zscore(dataset['Age'])) max_threshold=3
outliers = dataset['Age'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:") print(outliers)
```

96 detected using Z-Score: 71.0

116 70.5

```
z_scores = np.abs(stats.zscore(dataset['Fare'])) max_threshold=3
outliers = dataset['Fare'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:") print(outliers)
```

In [21]:

493 71.0

630 80.0

672 70.0

745 70.0

851 74.0

Name: Age, dtype: float64

Outliers detected using Z-Score:

27 263.0000

88 263.0000

118 247.5208

258 512.3292

299 247.5208

311 262.3750

341 263.0000

377 211.5000

380 227.5250

438 263.0000

527 221.7792

557 227.5250

679 512.3292

689 211.3375

700 227.5250

716 227.5250

730 211.3375

737 512.3292

742 262.3750

779 211.3375

Name: Fare, dtype: float64

Seperate dependent and independent variables

```
In [22]: x=dataset.iloc[:,2:13]
y = dataset['Survived']
```

```
In [23]: x.head()
```

Out[23]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	Braund, Mr.	male	22.0	1	0	A/5	7.2500	Unknown	S
1	1	Cumings, Mrs. John Bradley	female	35.0	1	0	PC	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Unknown	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Unknown	S

In [24]: y.head()

0 0
1 1

Out[24]:

```
from sklearn.preprocessing import LabelEncoder
```

2 1
3 1
4 0 Name: Survived, dtype: int64

type(x)

In [25]:

Out[25]: pandas.core.frame.DataFrame

type(y)

le=LabelEncoder()

x['Sex'] = le.fit_transform(x['Sex'])

x['Sex']

0 1
1 0
2 0
3 0
4 1

In [27]:

In [28]:

886 ..
887 1
888 0
889 0
890 1
891 1
Name: SexLength, dtype: int32

In [29]:

In [30]:

Out[30]:

x["Sex"].value_counts()

In [26]:

Out[26]:

pandas.core.series.Series

In [31]: encoding

Out[31]:

1	577
0	314

Name: Sex, dtype: int64

```
x["Sex"].nunique()
```

In [32]:

Out[32]:

2

In [33]:

```
x.head()
```

Out[33]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	Braund, Mr.	Male	22.0	1	0	A/5	7.2500	Unknown	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs)	Female	38.0	1	0	PC	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	Female	26.0	0	0	STON/O2. 3101282	7.9250	Unknown	S
3	1	Futrelle, Mrs. Jacques Heath (Lily)	Female	35.0	1	0	113803	53.1000	C123	S
4	3	Allen, Mr. William	Male	35.0	0	0	373450	8.0500	Unknown	S

```
print(x.columns)
```

In [34]:

one hot encoding

```
Index(['Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare',  
      'Cabin', 'Embarked'],  
      dtype='object')
```

```
In [35]: x.head()
```

Out[35]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	Unknown	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	Unknown	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	S
4	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	Unknown	S

```
In [36]: x.shape
```

Out[36]: (891, 10)

```
In [37]: Embarked=pd.get_dummies(x["Embarked"])
```

```
In [38]: Embarked
```

Out[38]:

	C	Q	S
0	0	0	1
1	1	0	0
2	0	0	1
3	0	0	1
4	0	0	1
...
886	0	0	1
887	0	0	1
888	0	0	1
889	1	0	0
890	0	1	0

891 rows × 3 columns

```
In [39]: #concat
x=pd.concat([x,Embarked],axis=1)
```

```
In [40]: x.head()
```

Out[40]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	C	Q	S
0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	Unknown	S	0	0	1
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C	1	0	0
2	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	Unknown	S	0	0	1
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	S	0	0	1
4	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	Unknown	S	0	0	1

In [41]:

```
#dropping Embarked column
x.drop(["Embarked"],axis=1,inplace=True)
```

In [42]:

```
x.head(10)
```

Out[42]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S
0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	Unknown	0	0	1
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	1	0	0
2	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	Unknown	0	0	1
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	0	0	1
4	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	Unknown	0	0	1
5	3	Moran, Mr. James	1	28.0	0	0	330877	8.4583	Unknown	0	1	0
6	1	McCarthy, Mr. Timothy J	1	54.0	0	0	17463	51.8625	E46	0	0	1
7	3	Palsson, Master. Gosta Leonard	1	2.0	3	1	349909	21.0750	Unknown	0	0	1
8	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	0	27.0	0	2	347742	11.1333	Unknown	0	0	1
9	2	Nasser, Mrs. Nicholas (Adele Achem)	0	14.0	1	0	237736	30.0708	Unknown	1	0	0

In [43]:

```
x.shape
```

Out[43]: (891, 12)

splitting into training and testing set

In [44]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [45]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[45]: ((623, 12), (268, 12), (623,), (268,))

In [46]:

```
x_train
```

Out[46]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S
857	1	Daly, Mr. Peter Denis	1	51.0	0	0	113055	26.5500	E17	0	0	1
52	1	Harper, Mrs. Henry Sleeper (Myna Haxtun)	0	49.0	1	0	PC 17572	76.7292	D33	1	0	0
386	3	Goodwin, Master. Sidney Leonard	1	1.0	5	2	CA 2144	46.9000	Unknown	0	0	1
124	1	White, Mr. Percival Wayland	1	54.0	0	1	35281	77.2875	D26	0	0	1
578	3	Caram, Mrs. Joseph (Maria Elias)	0	28.0	1	0	2689	14.4583	Unknown	1	0	0
...
835	1	Compton, Miss. Sara Rebecca	0	39.0	1	1	PC 17756	83.1583	E49	1	0	0
192	3	Andersen-Jensen, Miss. Carla Christine Nielsine	0	19.0	1	0	350046	7.8542	Unknown	0	0	1
629	3	O'Connell, Mr. Patrick D	1	28.0	0	0	334912	7.7333	Unknown	0	1	0
559	3	de Messemaeker, Mrs. Guillaume Joseph (Emma)	0	36.0	1	0	345572	17.4000	Unknown	0	0	1

684 2 Brown, Mr. Thomas William Solomon 1 60.0 1 1 29750 39.0000 Unknown 0 0 1

623 rows × 12 columns

In [47]: x_test

Out[47]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S
495	3	Yousseff, Mr. Gerious	1	28.0	0	0	2627	14.4583	Unknown	1	0	0
648	3	Wiley, Mr. Edward	1	28.0	0	0	S.O./P.P. 751	7.5500	Unknown	0	0	1
278	3	Rice, Master. Eric	1	7.0	4	1	382652	29.1250	Unknown	0	1	0
31	1	Spencer, Mrs. William Augustus (Marie Eugenie)	0	28.0	1	0	PC 17569	146.5208	B78	1	0	0
255	3	Touma, Mrs. Darwis (Hanne Youssef Razi)	0	29.0	0	2	2650	15.2458	Unknown	1	0	0
...
263	1	Harrison, Mr. William	1	40.0	0	0	112059	0.0000	B94	0	0	1
718	3	McEvoy, Mr. Michael	1	28.0	0	0	36568	15.5000	Unknown	0	1	0
620	3	Yasbeck, Mr. Antoni	1	27.0	1	0	2659	14.4542	Unknown	1	0	0
786	3	Sjoblom, Miss. Anna Sofia	0	18.0	0	0	3101265	7.4958	Unknown	0	0	1
64	1	Stewart, Mr. Albert A	1	28.0	0	0	PC 17605	27.7208	Unknown	1	0	0

268 rows × 12 columns

In [48]: y_train

Out[48]:

857	1
52	1
386	0
124	0
578	0
..	
835	1
192	1
629	0
559	1
684	0

Name: Survived, Length: 623, dtype: int64

In [49]: y_test

Out[49]:

495	0
648	0
278	0
31	1
255	1
..	
263	0
718	0
620	0
786	1
64	0

Name: Survived, Length: 268, dtype: int64

feature scaling

```
from sklearn.preprocessing import StandardScaler
```

```
scale = StandardScaler()  
x[['Age', 'Fare']] = scale.fit_transform(x[['Age', 'Fare']])
```

In [55]:

x.head()

In [56]:

In [58]:

Out[58]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S		
0	3	Braund, Mr.	Owen	1	-0.565736	1	0	A/5	-0.502445	Unknown	0	0	1	
1	1	Cumings, Mrs.	John	Bradley	(Florence Briggs)	0	1	0	PC	0.786845	C85	1	0	0
2	3	Heikkinen, Miss.	Laina	0	-0.258337	0	0	STON/O2.	3101282	-0.488854	Unknown	0	0	1
3	1	Futrelle, Mrs.	Jacques	Heath	(Lily)	0	1	0	113803	0.420730	C123	0	0	1
4	3	Allen, Mr.	William	0	0.433312	0	0		373450	-0.486337	Unknown	0	0	1

In []:

Loading

[MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js