IMPORT THE BASIC LIBRARIES

In [ ]:

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

Import the data file

In [2]:

```
df=pd.read_csv('titanic3.csv')
```

Separate the categorial and numerical columns in dataset

In [3]:

```
cat_var=df.select_dtypes(object)
num_var=df.select_dtypes(np.number)
```

Check the null values in categorial data

In [4]:

```
cat_var.isnull().sum()
```

Out[4]:

```
name            0
sex             0
ticket          0
cabin        1014
embarked        2
boat          823
home.dest     564
dtype: int64
```

In [5]:

```
cat_var.head(2)
```

Out[5]:

	name	sex	ticket	cabin	embarked	boat	home.dest
0	Allen, Miss. Elisabeth Walton	female	24160	B5	S	2	St Louis, MO
1	Allison, Master. Hudson Trevor	male	113781	C22 C26	S	11	Montreal, PQ / Chesterville, ON

DROP THE USELESS COLUMNS WHICH ARE NOT USEFULL

In [6]:

```
cat_var.drop(['name','ticket','home.dest'],axis=1,inplace=True)
```

```
C:\Users\sanja\anaconda3\lib\site-packages\pandas\core\frame.py:4906: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  return super().drop(
```

In [7]:

```
cat_var.head()
```

Out[7]:

	sex	cabin	embarked	boat
0	female	B5	S	2
1	male	C22 C26	S	11
2	female	C22 C26	S	NaN
3	male	C22 C26	S	NaN
4	female	C22 C26	S	NaN

In [8]:

```
cat_var.isnull().sum()
```

Out[8]:

```
sex            0
cabin       1014
embarked       2
boat         823
dtype: int64
```

In [9]:

```
cat_var.drop(['boat'],axis=1,inplace=True)
```

```
C:\Users\sanja\anaconda3\lib\site-packages\pandas\core\frame.py:4906: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  return super().drop(
```

In [10]:

```
cat_var.head(3)
```

Out[10]:

	sex	cabin	embarked
0	female	B5	S
1	male	C22 C26	S
2	female	C22 C26	S

In [11]:

```
cat_var.isnull().sum()
```

Out[11]:

```
sex            0
cabin       1014
embarked       2
dtype: int64
```

In [ ]:

In [ ]:

```

```

In [ ]:

```

```

Filling the null values with appropriate ones

In [12]:

```
cat_var.fillna(cat_var.cabin.value_counts().idxmax(),inplace=True)
cat_var.fillna(cat_var.embarked.value_counts().idxmax(),inplace=True)
```

```
C:\Users\sanja\anaconda3\lib\site-packages\pandas\core\frame.py:5176: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  return super().fillna(
```

In [13]:

```
cat_var.head()
```

Out[13]:

	sex	cabin	embarked
0	female	B5	S
1	male	C22 C26	S
2	female	C22 C26	S
3	male	C22 C26	S
4	female	C22 C26	S

In [14]:

```
cat_var.isnull().sum()
```

Out[14]:

```
sex         0
cabin       0
embarked    0
dtype: int64
```

Preprocess the independent categorial data by Label Encoding

In [15]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
cat_var=cat_var.apply(le.fit_transform)
cat_var.tail()
```

Out[15]:

	sex	cabin	embarked
1304	0	80	0
1305	0	80	0
1306	1	80	0
1307	1	80	0
1308	1	80	3

In [16]:

```
num_var.isnull().sum()
```

Out[16]:

```
pclass         0
survived       0
age          263
sibsp          0
parch          0
fare           1
body        1188
dtype: int64
```

In [17]:

```
num_var.drop(['body'],axis=1,inplace=True)
```

```
C:\Users\sanja\anaconda3\lib\site-packages\pandas\core\frame.py:4906: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  return super().drop(
```

In [18]:

```
num_var
```

Out[18]:

	pclass	survived	age	sibsp	parch	fare
0	1	1	29.0000	0	0	211.3375
1	1	1	0.9167	1	2	151.5500
2	1	0	2.0000	1	2	151.5500
3	1	0	30.0000	1	2	151.5500
4	1	0	25.0000	1	2	151.5500
...	...	...	...	...	...	...
1304	3	0	14.5000	1	0	14.4542
1305	3	0	NaN	1	0	14.4542
1306	3	0	26.5000	0	0	7.2250
1307	3	0	27.0000	0	0	7.2250
1308	3	0	29.0000	0	0	7.8750

1309 rows × 6 columns

In [19]:

```
num_var.isnull().sum()
```

Out[19]:

```
pclass        0
survived      0
age         263
sibsp         0
parch         0
fare          1
dtype: int64
```

In [ ]:

```

```

In [ ]:

```

```

Fill the numerical Null Values

In [ ]:

```
num_var['age'].fillna(df['age'].mean(),inplace=True)
num_var['fare'].fillna(df['fare'].mean(),inplace=True)
```

In [21]:

```
num_var.isnull().sum()
```

Out[21]:

```
pclass      0
survived    0
age         0
sibsp       0
parch       0
fare        0
dtype: int64
```

Merge the dataframes into final_dataset

In [22]:

```
final_dataset=pd.concat([num_var,cat_var],axis=1)
```

Classify the dependent and not dependent columns in the dataset

In [23]:

```
X=final_dataset.drop(['survived'],axis=1)
Y=final_dataset['survived']
```

Train and Test the Model

In [24]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,random_state=42,test_size=0.2)
len(X_train),len(X_test),len(Y_train),len(Y_test)
```

Out[24]:

```
(1047, 262, 1047, 262)
```

import the required classification models and metric measures

In [25]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [26]:

```
LR= LogisticRegression()
KNN=KNeighborsClassifier()
NB=GaussianNB()
LSVM=LinearSVC()
SVM=SVC()
DT=DecisionTreeClassifier()
RFC=RandomForestClassifier()
```

Fit those models

In [27]:

```
LR_fit=LR.fit(X_train,Y_train)
KNN_fit=KNN.fit(X_train,Y_train)
NB_fit=NB.fit(X_train,Y_train)
LSVM_fit=LSVM.fit(X_train,Y_train)
SVM_fit=SVM.fit(X_train,Y_train)
DT_fit=DT.fit(X_train,Y_train)
RFC_fit=RFC.fit(X_train,Y_train)
```

```
C:\Users\sanja\anaconda3\lib\site-packages\sklearn\linear_model\_logisti
c.py:444: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://s
cikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-re
gression (https://scikit-learn.org/stable/modules/linear_model.html#logis
tic-regression)
  n_iter_i = _check_optimize_result(
C:\Users\sanja\anaconda3\lib\site-packages\sklearn\svm\_base.py:1225: Con
vergenceWarning: Liblinear failed to converge, increase the number of ite
rations.
  warnings.warn(
```

Predict the models

In [28]:

```
LR_pred=LR_fit.predict(X_test)
KNN_pred=KNN_fit.predict(X_test)
NB_pred=NB_fit.predict(X_test)
LSVM_pred=LSVM_fit.predict(X_test)
SVM_pred=SVM_fit.predict(X_test)
DT_pred=DT_fit.predict(X_test)
RFC_pred=RFC_fit.predict(X_test)
```

Efficiency Metrics

In [29]:

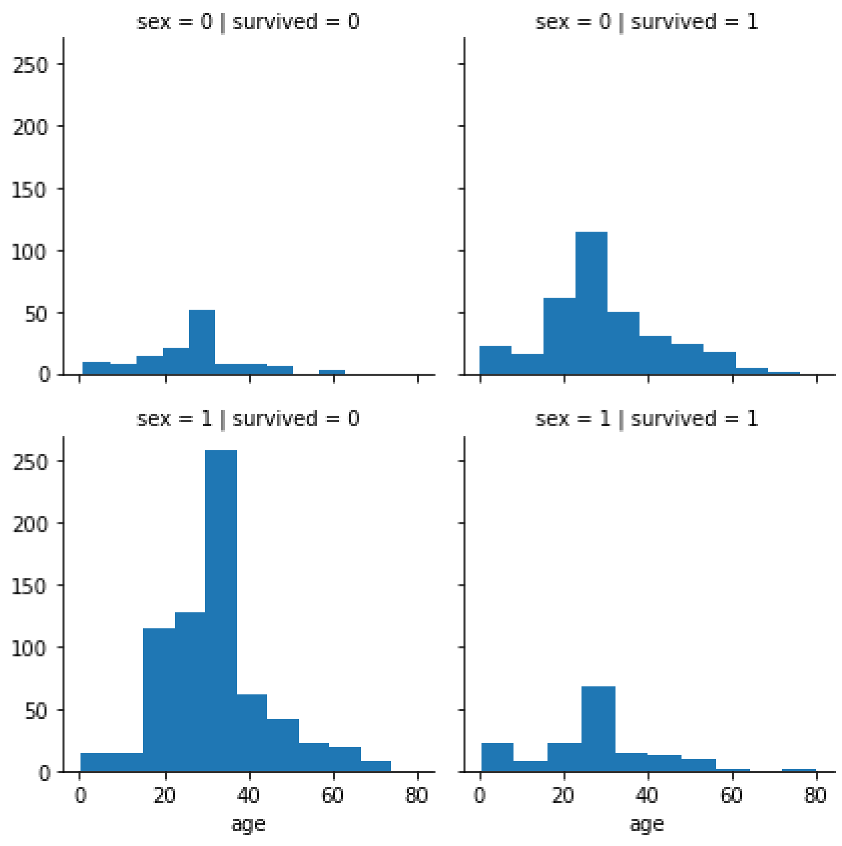
```
print("the accuracy score of Logistic Regression is %f percent accurate" %(accuracy_scor
print("the accuracy score of KneighborsClassification is %f percent accurate" %(accuracy
print("the accuracy score of NaiveBaiyes is %f percent accurate" %(accuracy_score(NB_pre
print("the accuracy score of Linear SVC is %f percent accurate" %(accuracy_score(LSVM_pr
print("the accuracy score of SVC is %f percent accurate" %(accuracy_score(SVM_pred,Y_tes
print("the accuracy score of Descision Tree Classifier is %f percent accurate" %(accurac
print("the accuracy score of Random Forest Classifier is %f percent accurate" %(accuracy
```

```
the accuracy score of Logistic Regression is 77.862595 percent accurate
the accuracy score of KneighborsClassification is 69.083969 percent accur
ate
the accuracy score of NaiveBaiyes is 73.282443 percent accurate
the accuracy score of Linear SVC is 74.045802 percent accurate
the accuracy score of SVC is 64.885496 percent accurate
the accuracy score of Descision Tree Classifier is 75.954198 percent accu
rate
the accuracy score of Random Forest Classifier is 78.244275 percent accur
ate
```

Visualizing the data

In [34]:

```
g=sns.FacetGrid(final_dataset,col="survived",row="sex")
g=g.map(plt.hist,"age")
```



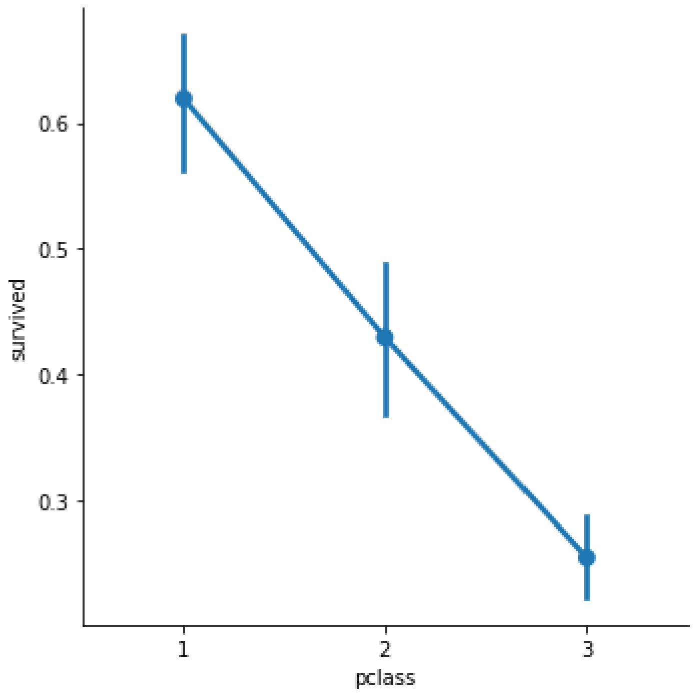
In [38]:

```
sns.factorplot(x="pclass",y="survived",data=final_dataset)
```

```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\categorical.py:3717: U
serWarning: The `factorplot` function has been renamed to `catplot`. The
original name will be removed in a future release. Please update your cod
e. Note that the default `kind` in `factorplot` (`'point'`) has changed
`'strip'` in `catplot`.
  warnings.warn(msg)
```

Out[38]:

```
<seaborn.axisgrid.FacetGrid at 0x1fe9783a370>
```

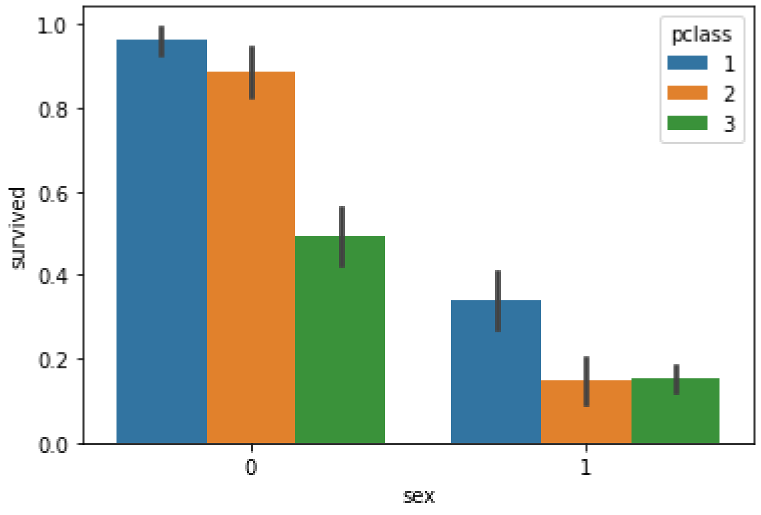


In [41]:

```
sns.barplot(x="sex",y="survived",hue="pclass",data=final_dataset)
```

Out[41]:

```
<AxesSubplot:xlabel='sex', ylabel='survived'>
```


survived
pclass
sex