

```
In [1]: import pandas as pd
import numpy as np
```

```
In [3]: df1=pd.read_csv(r"E:\DATASET\train.csv")
```

```
In [4]: df1.columns
```

```
Out[4]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
              'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
              dtype='object')
```

```
In [5]: df1=df1.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin'], axis=1)
```

```
In [6]: df1
```

```
Out[6]:
```

	Survived	Pclass	Sex	Age	Embarked
0	0	3	male	22.0	S
1	1	1	female	38.0	C
2	1	3	female	26.0	S
3	1	1	female	35.0	S
4	0	3	male	35.0	S
...	...	...	...	...	...
886	0	2	male	27.0	S
887	1	1	female	19.0	S
888	0	3	female	NaN	S
889	1	1	male	26.0	C
890	0	3	male	32.0	Q

891 rows × 5 columns

```
In [7]: df1.columns
```

```
Out[7]: Index(['Survived', 'Pclass', 'Sex', 'Age', 'Embarked'], dtype='object')
```

```
In [8]: X=df1.drop("Survived",axis=1)
```

```
In [9]: X.shape
```

```
Out[9]: (891, 4)
```

```
In [10]: y=df1["Survived"]
```

In [11]: `y.shape`

Out[11]: (891,)

In [13]: `X=pd.get_dummies(X,drop_first=True)`

In [27]: `X["Age"]=X["Age"].fillna(X["Age"].mean())`

In [ ]:

In [28]: `X`

Out[28]:

		Pclass	Age	Sex_male	Embarked_Q	Embarked_S
0	3	22.000000	1	0	1	
1	1	38.000000	0	0	0	
2	3	26.000000	0	0	1	
3	1	35.000000	0	0	1	
4	3	35.000000	1	0	1	
...	...	...	...	...	...	
886	2	27.000000	1	0	1	
887	1	19.000000	0	0	1	
888	3	29.699118	0	0	1	
889	1	26.000000	1	0	0	
890	3	32.000000	1	1	0	

891 rows × 5 columns

In [29]: `from sklearn.model_selection import train_test_split`

In [38]: `Xtrain,Xtest,ytrain,ytest=train_test_split(X,y,test_size=0.2)`

In [39]: `from sklearn.neighbors import KNeighborsClassifier`

In [40]: `knn=KNeighborsClassifier(n_neighbors=7)`

In [41]: `from sklearn.linear_model import LogisticRegression`

In [42]: `lreg=LogisticRegression()`

In [43]: `algo=[lreg,knn]`

```
In [44]: for i in algo:  
         print(i)
```

```
LogisticRegression()  
KNeighborsClassifier(n_neighbors=7)
```

```
In [45]: for z in algo:  
         z.fit(Xtrain,ytrain)  
         s=z.score(Xtest,ytest)  
         print(z, "Score==",s)
```

```
LogisticRegression() Score== 0.7430167597765364  
KNeighborsClassifier(n_neighbors=7) Score== 0.7374301675977654
```

###CROSS VALIDATION

```
In [ ]:
```