For the given dataset, apply the suitable data preprocessing techniques (any one) and visualize the details of original as well as preprocessed dataset with the help of visualization techniques (min.5)

In [46]:
```
import pandas as pd
```

In [47]:
```
import matplotlib.pyplot as plt
```

In [48]:
```
import seaborn as sns
```

In [49]:
```
import numpy as np
```

In [50]:
```
df = pd.read_csv("prac_1.csv")
```

In [51]:
```
df
```

Out[51]:

	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 [52]:
```
df.shape
```

Out[52]: (891, 12)

In [53]:
```
df.describe()
```

Out[53]:

	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 [54]:
```
df.dtypes
```

Out[54]:
```
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 [55]:
```
df.isnull().sum()
```

Out[55]:
```
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 [56]:
```
df = df.drop(['PassengerId'], axis=1)
```

In [57]:
```
df
```

Out[57]:

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

891 rows × 11 columns

In [58]:
```
df.isnull().sum()
```

Out[58]:
```
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 [59]:
```
df['Cabin'].unique()
```

Out[59]:
```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

In [60]:
```
df['Cabin'].nunique()
```

Out[60]:
```
147
```

In [61]:
```
dct = { "A": 1, "B": 2, "C": 3, "D": 4,"E":5, "F":6, "G":7, "T": 8, "U":9 }
data_list = [df]
```

In [62]:
```
 for dataset in data_list:
        dataset['Cabin'] = dataset['Cabin'].fillna("U0")
```

```
        dataset['Cabin_updated'] = dataset['Cabin'].str[0] #extracting first character from the cabin column
        dataset['Cabin_updated'] = dataset['Cabin_updated'].map(dct) #mapping dictionary on 'Cabin' column
        dataset['Cabin_updated'] = dataset['Cabin_updated'].fillna(0) # any left-out value filling to 0
        dataset['Cabin_updated'] = dataset['Cabin_updated'].astype(int) # typecasting to Integer type
```

In [63]:
```
df.head()
```

Out[63]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Cabin_updated
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	U0	S	9
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	3
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	U0	S	9
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	3
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	U0	S	9

In [64]:
```
df.isnull().sum()
```

Out[64]:
```
Survived          0
Pclass            0
Name              0
Sex               0
Age             177
SibSp             0
Parch             0
Ticket            0
Fare              0
Cabin             0
Embarked          2
Cabin_updated     0
dtype: int64
```

In [65]:
```
data_list = [df]

for dataset in data_list:
    mean = df["Age"].mean()
    std = df["Age"].std()
    is_null = dataset["Age"].isnull().sum()

    random_number = np.random.randint(mean - std, mean + std, size = is_null)
    age_updated = dataset["Age"].copy()
```

```
    age_updated[np.isnan(age_updated)] = random_number
    dataset["Age"] = age_updated
    dataset["Age"] = df["Age"].astype(int)
```

In [66]:
```
df.isnull().sum()
```

Out[66]:
```
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin            0
Embarked         2
Cabin_updated    0
dtype: int64
```

In [67]:
```
df["Embarked"].value_counts()
```

Out[67]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [68]:
```
df["Embarked"] = df["Embarked"].fillna('S')
```

In [69]:
```
df.isnull().sum()
```

Out[69]:
```
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin            0
Embarked         0
Cabin_updated    0
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Survived       891 non-null    int64
 1   Pclass         891 non-null    int64
 2   Name           891 non-null    object
 3   Sex            891 non-null    object
 4   Age            891 non-null    int32
 5   SibSp          891 non-null    int64
 6   Parch          891 non-null    int64
 7   Ticket         891 non-null    object
 8   Fare           891 non-null    float64
 9   Cabin          891 non-null    object
 10  Embarked       891 non-null    object
 11  Cabin_updated  891 non-null    int32
dtypes: float64(1), int32(2), int64(4), object(5)
memory usage: 76.7+ KB
```

In [71]:
```
df['Prefix'] = df['Name'].str.extract('([A-Za-z]+)\.')
df['Prefix'].value_counts()
```

Out[71]:
```
Mr          517
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Mlle          2
Major         2
Col           2
Countess      1
Capt          1
Ms            1
Sir           1
Lady          1
Mme           1
Don           1
Jonkheer      1
Name: Prefix, dtype: int64
```

In [72]:
```
data_list = [df]
prefixes = {'Mr': 1, 'Miss': 2, 'Mrs': 3, 'Master': 4, 'Others': 5}

for dataset in data_list:
    dataset['Prefix'].replace(['Lady'] , 'Mrs', inplace = True)
    dataset['Prefix'].replace(['Sir', 'Rev'] , 'Mr', inplace = True)
    dataset['Prefix'].replace(['Mlle', 'Ms', 'Countless', 'Mme'], 'Mrs' , inplace = True)
    dataset['Prefix'].replace(['Dr', 'Master', 'Major', 'Col', 'Capt','Don', 'Jonkheer'], 'Others', inplace= True)
    dataset['Prefix'] = dataset['Prefix'].map(prefixes)
```

In [73]:
```
df['Prefix'].dtype
```

Out[73]:
```
dtype('float64')
```

In [74]:
```
df['Prefix'] = df['Prefix'].round().astype('Int64')
```

In [75]:
```
df.head(13)
```

Out[75]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Cabin_updated	Prefix
0	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	U0	S	9	1
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C	3	3
2	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	U0	S	9	2
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S	3	3
4	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	U0	S	9	1
5	0	3	Moran, Mr. James	male	41	0	0	330877	8.4583	U0	Q	9	1
6	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S	5	1
7	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.0750	U0	S	9	5
8	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333	U0	S	9	3
9	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708	U0	C	9	3
10	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7000	G6	S	7	2
11	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.5500	C103	S	3	2
12	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.0500	U0	S	9	1

In [76]:
```
df['Fare'] = df['Fare'].fillna(0)
df['Fare'] = df['Fare'].astype(int)
```

In [77]:
```
#Sex : convert 'sex' feature into numeric
gender = {"male": 1, "female":2}
data_list = [df]

for dataset in data_list:
    dataset['Sex'] = dataset['Sex'].map(gender)
```

In [78]:
```
#Embarked column
data_list = [df]
new_emb = {"S": 0, "C": 1, "Q":2}
```

```
for dataset in data_list:
    dataset['Embarked'] = dataset['Embarked'].map(new_emb)
```

In [79]:
```
#now finally we can drop the column like: 'Ticket' , 'Cabin' , 'Name' etc.
df.head()
```

Out[79]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Cabin_updated	Prefix
0	0	3	Braund, Mr. Owen Harris	1	22	1	0	A/5 21171	7	U0	0	9	1
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	2	38	1	0	PC 17599	71	C85	1	3	3
2	1	3	Heikkinen, Miss. Laina	2	26	0	0	STON/O2. 3101282	7	U0	0	9	2
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	2	35	1	0	113803	53	C123	0	3	3
4	0	3	Allen, Mr. William Henry	1	35	0	0	373450	8	U0	0	9	1

In [80]:
```
data_list = [df]
for dataset in data_list:
    dataset = dataset.drop(['Ticket', 'Cabin', 'Name'], axis = 1, inplace = True)
```

In [81]:
```
df.head()
```

Out[81]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Cabin_updated	Prefix
0	0	3	1	22	1	0	7	0	9	1
1	1	1	2	38	1	0	71	1	3	3
2	1	3	2	26	0	0	7	0	9	2
3	1	1	2	35	1	0	53	0	3	3
4	0	3	1	35	0	0	8	0	9	1

In [82]:
```
data_list = [df]

for dataset in data_list:
    dataset['Age'] = dataset['Age'].astype(int)
    dataset.loc[dataset['Age'] <=11, 'Age'] = 0
    dataset.loc[(dataset['Age'] >=12) & (dataset['Age'] <=18), 'Age'] = 1
    dataset.loc[(dataset['Age'] >=19) & (dataset['Age'] <=22), 'Age'] = 2
    dataset.loc[(dataset['Age'] >=23) & (dataset['Age'] <=27), 'Age'] = 3
    dataset.loc[(dataset['Age'] >=28) & (dataset['Age'] <=33), 'Age'] = 4
```

```
    dataset.loc[(dataset['Age'] >=33) & (dataset['Age'] <=40), 'Age'] = 5
    dataset.loc[dataset['Age'] >40, 'Age'] = 6
```

In [83]:
```
df.head()
```

Out[83]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Cabin_updated	Prefix
0	0	3	1	2	1	0	7	0	9	1
1	1	1	2	5	1	0	71	1	3	3
2	1	3	2	3	0	0	7	0	9	2
3	1	1	2	5	1	0	53	0	3	3
4	0	3	1	5	0	0	8	0	9	1

In [84]:
```
data_list = [df]

for dataset in data_list:
    dataset['Fare'] = dataset['Fare'].astype(int)
    dataset.loc[dataset['Fare'] <=8, 'Fare'] = 0
    dataset.loc[(dataset['Fare'] >8) & (dataset['Fare'] <=15), 'Fare'] = 1
    dataset.loc[(dataset['Fare'] >=15) & (dataset['Fare'] <=31), 'Fare'] = 2
    dataset.loc[(dataset['Fare'] >=32) & (dataset['Fare'] <=99), 'Fare'] = 3
    dataset.loc[(dataset['Fare'] >=100) & (dataset['Fare'] <=250), 'Fare'] = 4
   # dataset.loc[(dataset['Fare'] >=33) & (dataset['Fare'] <=40), 'Fare'] = 5
    dataset.loc[dataset['Fare'] >250, 'Fare'] = 5
```

In [85]:
```
df.head()
```

Out[85]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Cabin_updated	Prefix
0	0	3	1	2	1	0	0	0	9	1
1	1	1	2	5	1	0	3	1	3	3
2	1	3	2	3	0	0	0	0	9	2
3	1	1	2	5	1	0	3	0	3	3
4	0	3	1	5	0	0	0	0	9	1

In [86]:
```
df.drop(['Cabin_updated', 'Prefix','SibSp','Parch','Embarked'], axis = 1, inplace = True)
```

In [87]:
```
#implementation of naive bayes algorithm
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

In [89]:
```
X = df.drop('Survived', axis = 1)
```

In [90]:
```
y = df.Survived
```

In [91]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size= 0.30, random_state = 1)
```

In [92]:
```
model = GaussianNB()
```

In [93]:
```
model.fit(X_train, y_train)
```

Out[93]:
```
▾ GaussianNB
GaussianNB()
```

In [94]:
```
model.score(X_test,y_test)
```

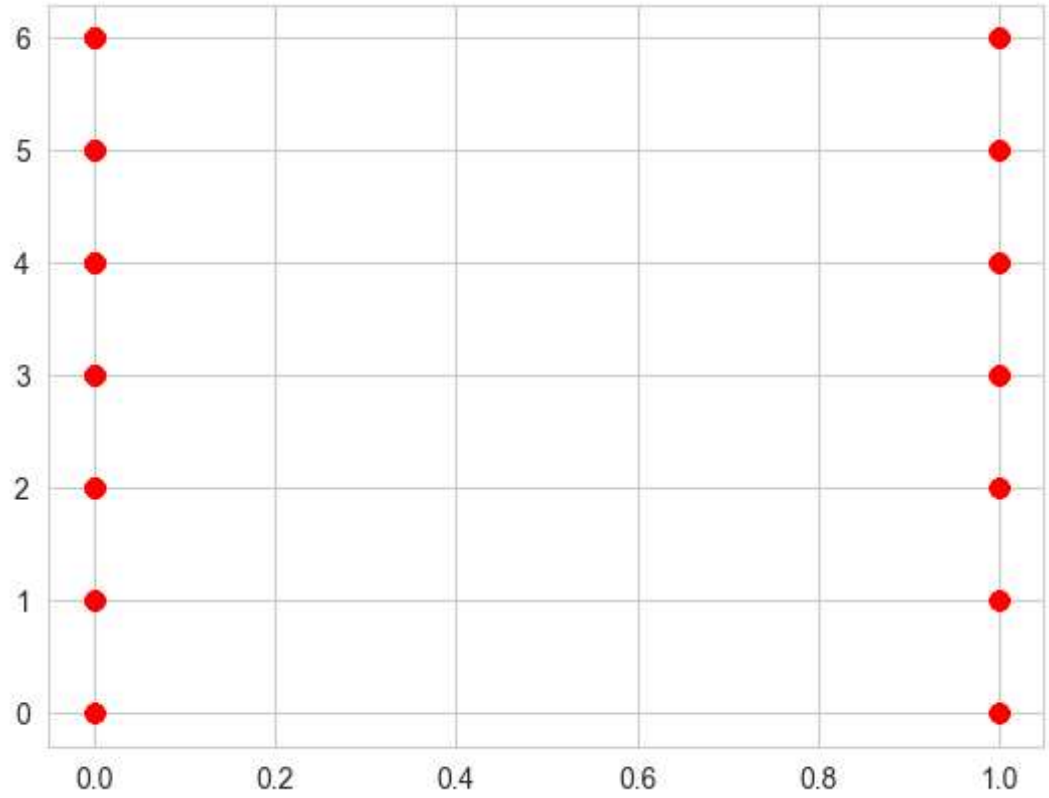
Out[94]:
```
0.7388059701492538
```

In [108…
```
#Kmeans clustering algorithm
from sklearn.cluster import KMeans
clf = KMeans(n_clusters = 2)
clf.fit(X)
```

Out[108]:
```
▾ KMeans
KMeans(n_clusters=2)
```

In [110…
```
plt.scatter(df['Survived'],df['Age'],color = 'red')
```

Out[110]:
```
<matplotlib.collections.PathCollection at 0x182a5d40910>
```



In [ ]:

In [ ]: