```
import numpy as np
import pandas as pd
lst={'Name':['Ram','Sham','Geeta','Sita','XX','YY','ZZ'],'Age':[14,3,12,20,21,60,65],'Gender':['M','M','F','F','M','M',np.nan]}
df1=pd.DataFrame(lst)
print(df1)
```

```
    Name  Age Gender
0    Ram   14      M
1   Sham    3      M
2  Geeta   12      F
3   Sita   20      F
4     XX   21      M
5     YY   60      M
6     ZZ   65    NaN
```

```
#Find the categorical column and separate it out with a different dataframe. To do so, use the select_dtypes()
df_cat = df1.select_dtypes(exclude=[np.number])
print(df_cat)
```

```
    Name Gender
0    Ram      M
1   Sham      M
2  Geeta      F
3   Sita      F
4     XX      M
5     YY      M
6     ZZ    NaN
```

```
#Find the distinct unique values in the Grade column using unique() function
df_cat['Gender'].unique()
```

```
array([nan, 'F', 'M'], dtype=object)
```

```
#Find the frequency distribution of each categorical column using value_counts()
df_cat.Gender.value_counts()
```

```
M    3
F    2
Name: Gender, dtype: int64
```

```
#Replace the entries in the Gender column. Replace Male with 0 and Female with 1. To do so, use the replace() function:
df_cat.replace(['M','F'],[1,0])
```

	Name	Gender
0	Ram	1.0
1	Sham	1.0
2	Geeta	0.0
3	Sita	0.0
4	XX	1.0
5	YY	1.0
6	ZZ	NaN

```
print(df_cat)
```

```
    Name Gender
0    Ram      M
1   Sham      M
2  Geeta      F
3   Sita      F
4     XX      M
5     YY      M
6     ZZ    NaN
```

```
df_cat.Gender.replace(['M','F'],[0,1])
```

```
0    0.0
1    0.0
2    1.0
3    1.0
```

```
4    0.0
5    0.0
6    NaN
Name: Gender, dtype: float64
```

```
df_cat.Name.replace(['XX','YY','ZZ'],[0,1,2])
```

```
0      Ram
1     Sham
2    Geeta
3     Sita
4        0
5        1
6        2
Name: Name, dtype: object
```

```
from sklearn.preprocessing import LabelEncoder                  # Import label encoder
label_encoder = LabelEncoder()                                  # label_encoder object knows how to understand word labels
df1['Encoded_Name']=label_encoder.fit_transform(df1['Name'])     #Fit label encoder and return encoded labels
df1['Gender']=label_encoder.fit_transform(df1['Gender'])
print(df1)
```

```
    Name  Age  Gender  Encoded_Name
0    Ram   14       1             1
1   Sham    3       1             2
2  Geeta   12       0             0
3   Sita   20       0             3
4     XX   21       1             4
5     YY   60       1             5
6     ZZ   65       2             6
```

```
from sklearn.preprocessing import LabelEncoder                  # Import label encoder
label_encoder = LabelEncoder()                                  # label_encoder object knows how to understand word labels
df1['Name']=label_encoder.fit_transform(df1['Name'])            #Fit label encoder and return encoded labels
print(df1)
```

```
   Name  Age Gender
0     1   14      M
1     2    3      M
2     0   12      F
3     3   20      F
4     4   21      M
5     5   60      M
6     6   65    NaN
```

```
from sklearn.preprocessing import LabelEncoder                  # Import label encoder
label_encoder = LabelEncoder()                                  # label_encoder object knows how to understand word labels

print(df1)
```

```
   Name  Age Gender
0     1   14      M
1     2    3      M
2     0   12      F
3     3   20      F
4     4   21      M
5     5   60      M
6     6   65    NaN
```

```
from sklearn.preprocessing import LabelEncoder
lst=['X','Y','Z','A','B']
le = LabelEncoder()
le.fit(lst)
#print(lst)
```

```
LabelEncoder()
```

```
le.transform(lst)
```

```
array([2, 3, 4, 0, 1])
```

```
print(lst)
```

```
['X', 'Y', 'Z', 'A', 'B']
```

```
le.fit_transform(lst)

     array([2, 3, 4, 0, 1])
```

```
lst1=le.inverse_transform([2, 3, 4, 0, 1])
print(lst1)


     ['X' 'Y' 'Z' 'A' 'B']
```

```
print(lst)

     ['X', 'Y', 'Z', 'A', 'B']
```

```
list=['Distinction','Pass','fail','Second','First']
le.fit_transform(list)

     array([0, 2, 4, 3, 1])
```

```
list=['Pass','Second','First','Distinction','fail']
le.fit_transform(list)

     array([2, 3, 1, 0, 4])
```

```
import numpy as np
import pandas as pd
df_ohe= pd.get_dummies(df1,columns=['Gender'])
print(df_ohe)

        Name  Age  Gender_F  Gender_M
     0     1   14         0         1
     1     2    3         0         1
     2     0   12         1         0
     3     3   20         1         0
     4     4   21         0         1
     5     5   60         0         1
     6     6   65         0         0
```

```
df_ohe= pd.get_dummies(df1,columns=['Name','Gender'])
print(df_ohe)
```

```
import numpy as np
import pandas as pd
#df_ohe= pd.get_dummies(df1,columns=['Gender'],prefix='Gender')
df_ohe= pd.get_dummies(df1,columns=['Name'],prefix='Name')
print(df_ohe)

        Age Gender  Name_Geeta  Name_Ram  ...  Name_Sita  Name_XX  Name_YY  Name_ZZ
     0   14      M           0         1  ...          0        0        0        0
     1    3      M           0         0  ...          0        0        0        0
     2   12      F           1         0  ...          0        0        0        0
     3   20      F           0         0  ...          1        0        0        0
     4   21      M           0         0  ...          0        1        0        0
     5   60      M           0         0  ...          0        0        1        0
     6   65    NaN           0         0  ...          0        0        0        1

     [7 rows x 9 columns]
```

```
import numpy as np
import pandas as pd
df_ohe= pd.get_dummies(df1,columns=['Name'], drop_first=True)
print(df_ohe)

        Age Gender  Name_Ram  Name_Sham  Name_Sita  Name_XX  Name_YY  Name_ZZ
     0   14      M         1          0          0        0        0        0
     1    3      M         0          1          0        0        0        0
     2   12      F         0          0          0        0        0        0
     3   20      F         0          0          1        0        0        0
     4   21      M         0          0          0        1        0        0
     5   60      M         0          0          0        0        1        0
     6   65    NaN         0          0          0        0        0        1
```

```
from sklearn.preprocessing import OneHotEncoder

ohe=OneHotEncoder(sparse=False,handle_unknown='error')

ohe_df=ohe.fit_transform(df1[['Gender']])
print(ohe_df)
```

```
[[0. 1. 0.]
 [0. 1. 0.]
 [1. 0. 0.]
 [1. 0. 0.]
 [0. 1. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

```
print(df1)
```

```
    Name  Age Gender
0    Ram   14      M
1   Sham    3      M
2  Geeta   12      F
3   Sita   20      F
4     XX   21      M
5     YY   60      M
6     ZZ   65    NaN
```

```
import pandas as pd
df1['Age']=pd.cut(df1['Age'],bins = [0,13,30,80],labels = [0, 1, 2])
df1
```

	Name	Age	Gender
0	Ram	1	M
1	Sham	0	M
2	Geeta	0	F
3	Sita	1	F
4	XX	1	M
5	YY	2	M
6	ZZ	2	NaN

```
df1['Age']=pd.cut(df1['Age'],bins = 2,labels =['child','adult'])
df1
```

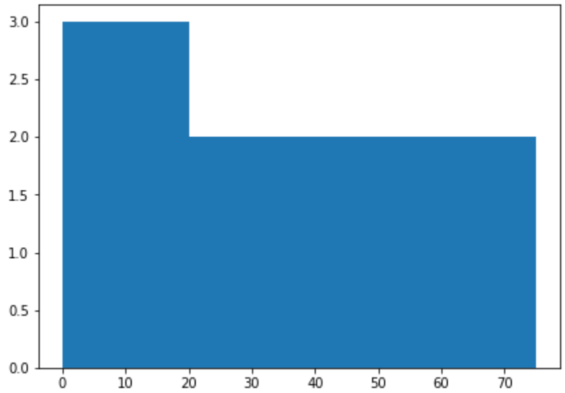
	Name	Age	Gender
0	Ram	child	M
1	Sham	child	M
2	Geeta	child	F
3	Sita	child	F
4	XX	child	M
5	YY	adult	M
6	ZZ	adult	NaN

```
from matplotlib import pyplot as plt
import numpy as np
```

```
# Creating dataset
```

```
# Creating histogram
fig, ax = plt.subplots(figsize =(7, 5))
ax.hist(df1['Age'], bins = [0,20,40, 75])

# Show plot
plt.show()
```

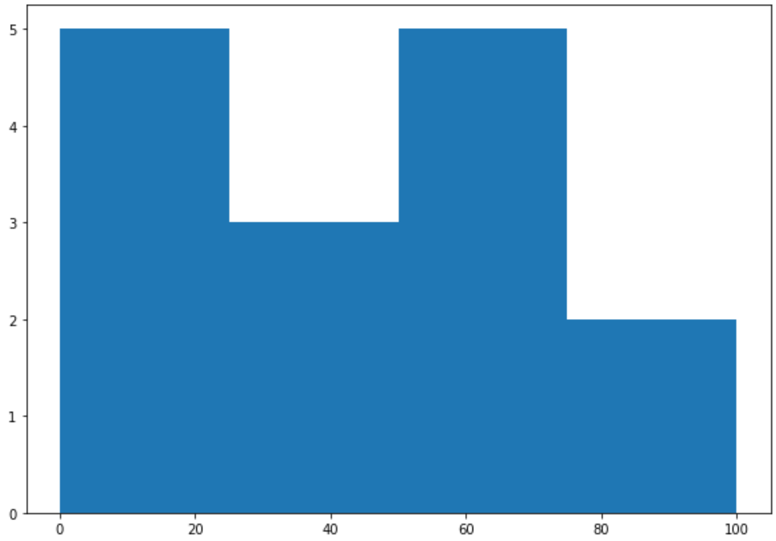


```
from matplotlib import pyplot as plt
import numpy as np


# Creating dataset
a = np.array([22, 87, 5, 43, 56,
              73, 55, 54, 11,
              20, 51, 5, 79, 31,
              27])

# Creating histogram
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a, bins = [0, 25, 50, 75, 100])

# Show plot
plt.show()
```


5
4
3
2
1
0
0
20
40
60
80
100