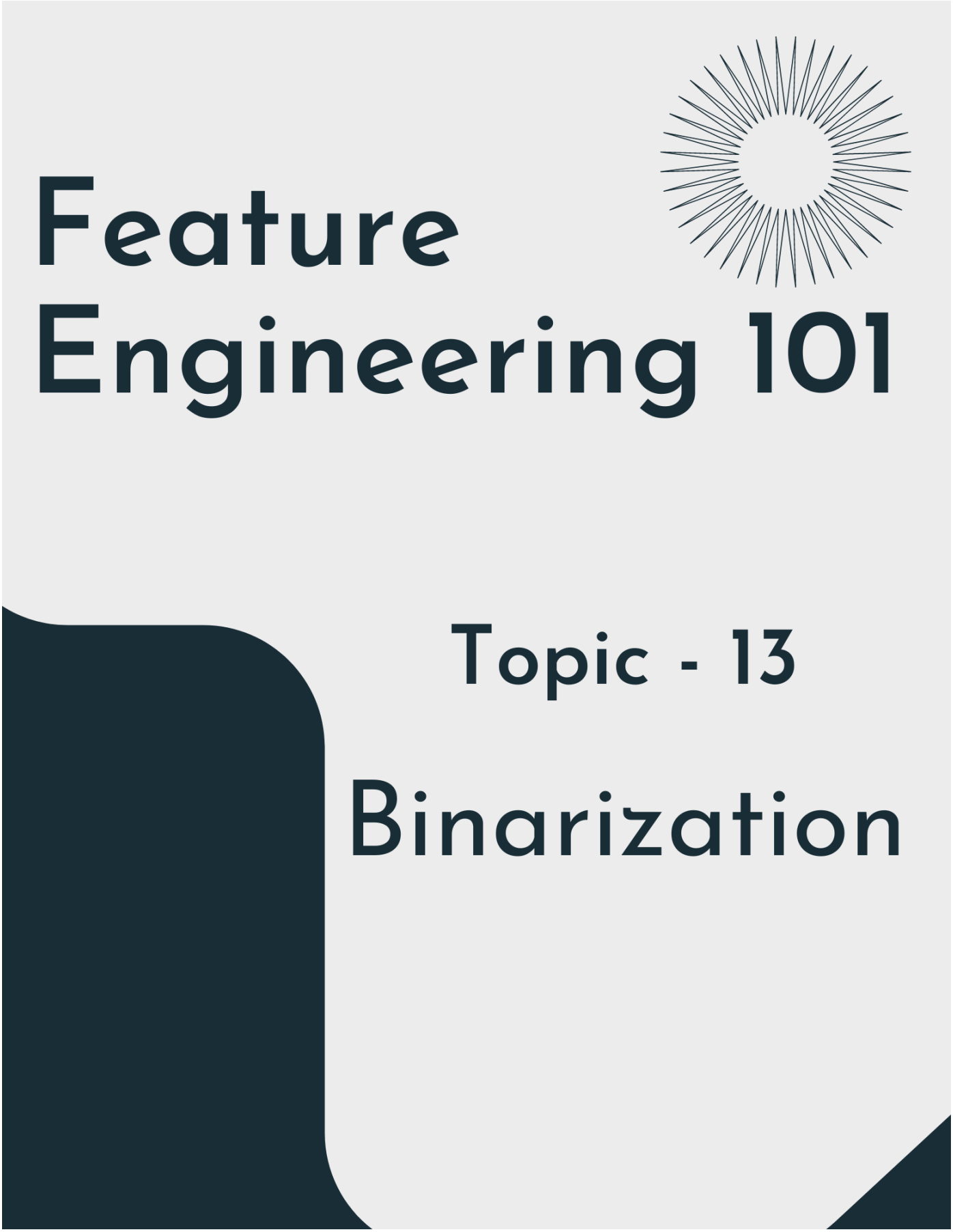


Binarization

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

In [2]:
```
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score

from sklearn.preprocessing import KBinsDiscretizer
from sklearn.compose import ColumnTransformer
```

In [3]:
```
df = pd.read_csv('train.csv',usecols=['Age','Fare','Survived'])
```

In [4]:
```
df.dropna(inplace=True)
```

In [5]:
```
df.shape
```

Out[5]:
```
(714, 3)
```

In [6]:
```
df.head()
```

Out[6]:

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

In [7]:
```
X = df.iloc[:,1:]
y = df.iloc[:,0]
```

In [8]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

In [9]:
```
X_train.head(2)
```

Out[9]:

	Age	Fare
328	31.0	20.5250
73	26.0	14.4542

In [10]:
```
clf = DecisionTreeClassifier()
```

In [11]:
```
clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
```

In [12]:
```
accuracy_score(y_test,y_pred)
```

Out[12]:
```
0.6433566433566433
```

In [13]:
```
np.mean(cross_val_score(DecisionTreeClassifier(),X,y,cv=10,scoring='accuracy'))
```

Out[13]:
```
0.6289319248826291
```

In [14]:
```
kbin_age = KBinsDiscretizer(n_bins=15,encode='ordinal',strategy='quantile')
kbin_fare = KBinsDiscretizer(n_bins=15,encode='ordinal',strategy='quantile')
```

In [15]:
```
trf = ColumnTransformer([
    ('first',kbin_age,[0]),
    ('second',kbin_fare,[1])
])
```

In [16]:
```
X_train_trf = trf.fit_transform(X_train)
X_test_trf = trf.transform(X_test)
```

In [17]:
```
trf.named_transformers_['first'].bin_edges_
```

Out[17]:
```
array([array([ 0.42,  6.  , 16.  , 19.  , 21.  , 23.  , 25.  , 28.  , 30.  ,
              32.  , 35.  , 38.  , 42.  , 47.  , 54.  , 80.  ])             ],
      dtype=object)
```

In [18]:
```
trf.named_transformers_['first'].bin_edges_
```

Out[18]:
```
array([array([ 0.42,  6.  , 16.  , 19.  , 21.  , 23.  , 25.  , 28.  , 30.  ,
              32.  , 35.  , 38.  , 42.  , 47.  , 54.  , 80.  ])             ],
      dtype=object)
```

In [19]:
```
output = pd.DataFrame({
    'age':X_train['Age'],
    'age_trf':X_train_trf[:,0],
    'fare':X_train['Fare'],
    'fare_trf':X_train_trf[:,1]
})
```

In [20]:
```
output['age_labels'] = pd.cut(x=X_train['Age'],
                                        bins=trf.named_transformers_['first'].bin_edges_[0].to
output['fare_labels'] = pd.cut(x=X_train['Fare'],
                                        bins=trf.named_transformers_['second'].bin_edges_[0].t
```

In [21]:
```
output.sample(5)
```

Out[21]:

	age	age_trf	fare	fare_trf	age_labels	fare_labels
821	27.0	6.0	8.6625	4.0	(25.0, 28.0]	(8.158, 10.5]
230	35.0	10.0	83.4750	13.0	(32.0, 35.0]	(76.292, 108.9]
784	25.0	6.0	7.0500	0.0	(23.0, 25.0]	(0.0, 7.25]
660	50.0	13.0	133.6500	14.0	(47.0, 54.0]	(108.9, 512.329]

	age	age_trf	fare	fare_trf	age_labels	fare_labels
621	42.0	12.0	52.5542	12.0	(38.0, 42.0]	(51.479, 76.292]

In [22]:
```
clf = DecisionTreeClassifier()
clf.fit(X_train_trf,y_train)
y_pred2 = clf.predict(X_test_trf)
```

In [23]:
```
accuracy_score(y_test,y_pred2)
```

Out[23]:
```
0.6363636363636364
```

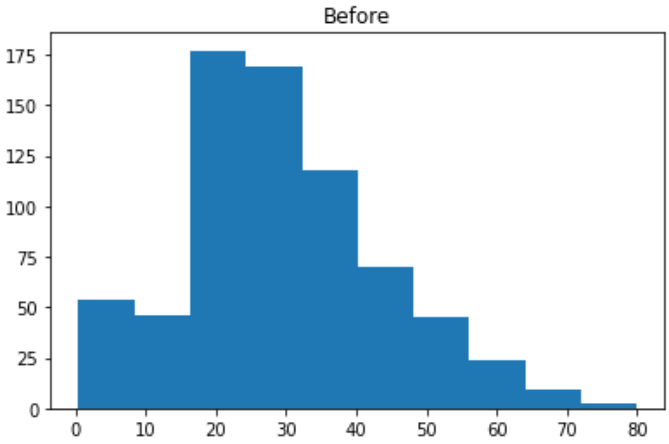
In [24]:
```
X_trf = trf.fit_transform(X)
np.mean(cross_val_score(DecisionTreeClassifier(),X,y,cv=10,scoring='accuracy'))
```

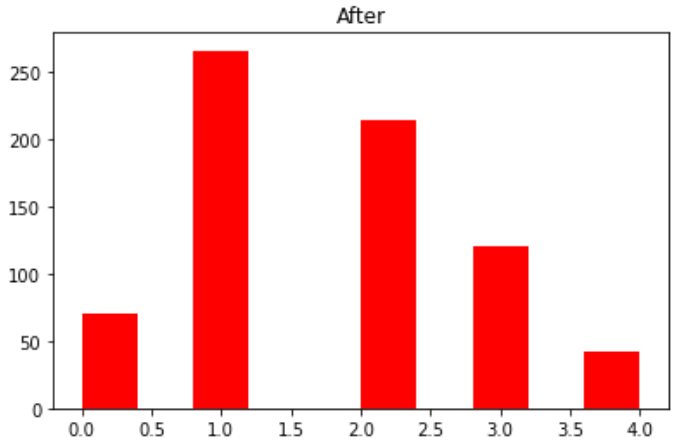
Out[24]:
```
0.6303012519561815
```

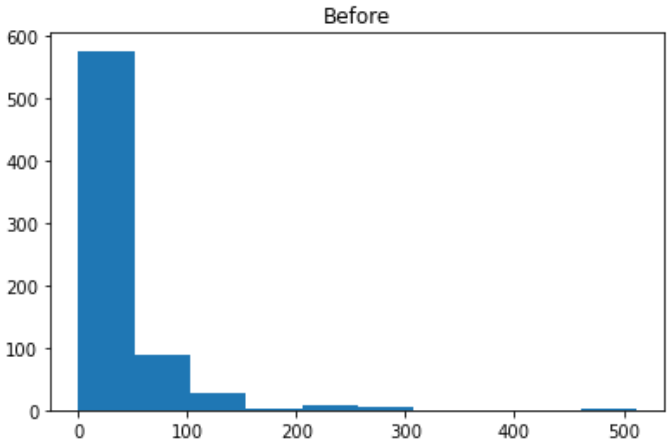
In [25]:
```
def discretize(bins,strategy):
    kbin_age = KBinsDiscretizer(n_bins=bins,encode='ordinal',strategy=strategy)
    kbin_fare = KBinsDiscretizer(n_bins=bins,encode='ordinal',strategy=strategy)

    trf = ColumnTransformer([
        ('first',kbin_age,[0]),
        ('second',kbin_fare,[1])
    ])

    X_trf = trf.fit_transform(X)
    print(np.mean(cross_val_score(DecisionTreeClassifier(),X,y,cv=10,scoring='accuracy')))

    plt.figure(figsize=(14,4))
    plt.subplot(121)
    plt.hist(X['Age'])
    plt.title("Before")

    plt.subplot(122)
    plt.hist(X_trf[:,0],color='red')
    plt.title("After")

    plt.show()

    plt.figure(figsize=(14,4))
    plt.subplot(121)
    plt.hist(X['Fare'])
    plt.title("Before")

    plt.subplot(122)
    plt.hist(X_trf[:,1],color='red')
    plt.title("Fare")

    plt.show()
```

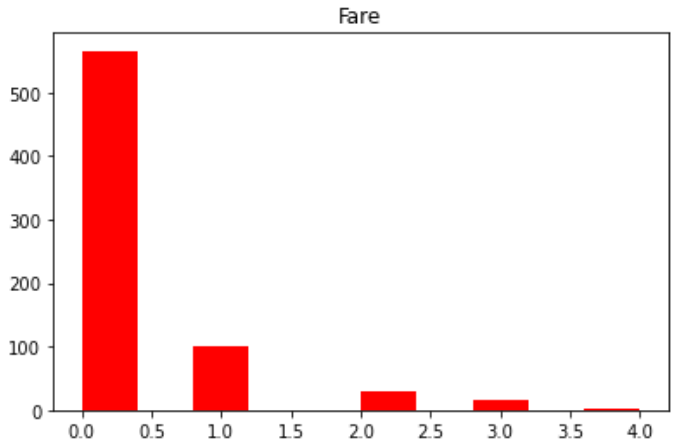
In [26]:
```
discretize(5,'kmeans')
```

```
0.6288928012519561
```


Before
175
150
125
100
75
50
25
0
0
10
20
30
40
50
60
70
80


After
250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Before
600
500
400
300
200
100
0
0
100
200
300
400
500


Fare
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

In [ ]: