In [19]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.tree import DecisionTreeClassifier
```

In [4]:

```
#importing same data again
data=pd.read_csv('titanic.csv',usecols=['Pclass','Survived','Sex','Age','SibSp','Parch','Fare',
data.sample(4)
```

Out[4]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
439	0	2	male	31.0	0	0	10.5000	S
687	0	3	male	19.0	0	0	10.1708	S
472	1	2	female	33.0	1	2	27.7500	S
67	0	3	male	19.0	0	0	8.1583	S

In [21]:

```
data['Embarked'].nunique()
```

Out[21]:

```
3
```

In [5]:

```
# doing train test and split of the data
X_train,X_test,y_train,y_test=train_test_split(data.drop('Survived',axis=1),data['Survived'],te
```

In [6]:

```
X_train.head()
```

Out[6]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
331	1	male	45.5	0	0	28.5000	S
733	2	male	23.0	0	0	13.0000	S
382	3	male	32.0	0	0	7.9250	S
704	3	male	26.0	1	0	7.8542	S
813	3	female	6.0	4	2	31.2750	S

In [8]:

```
y_train.sample(5)
```

Out[8]:

```
598    0
195    1
428    0
376    1
403    0
Name: Survived, dtype: int64
```

```
In [13]: # filling the missing values through column transformer
         Transformer1=ColumnTransformer(transformers=[
             ('trans1_age',SimpleImputer(),[2]),
             ('trans2_embarked',SimpleImputer(strategy='most_frequent'), [6]) # idex or name le call gar
         ],remainder='passthrough')
```

```
In [14]: data.isna().sum()
```

```
Out[14]: Survived      0
         Pclass        0
         Sex           0
         Age         177
         SibSp         0
         Parch         0
         Fare          0
         Embarked      2
         dtype: int64
```

One hot encoding

```
In [18]: Transformer2=ColumnTransformer(transformers=[
             ('onehotenconding', OneHotEncoder(sparse_output=False,handle_unknown='ignore'),[1,6])
         ], remainder='passthrough')
```

Scaling

```
In [27]: Transformer3=ColumnTransformer(transformers=[('scale',MinMaxScaler(),slice(0,10))]) # 0 dekhi 8
```

why did i write slice(0,10) ? because out of 7 columns in X_train, from colummn transformer 2 will be droped. remained 5 columns. Now from one hot encoding of sex and embarked 2 from sex and 3 from embarked will be formed adding total 5 columns to initial 5 columns which gives total of 10 columns so using 0, 10 in sciling

Feature selection

```
In [28]: Transformer4=SelectKBest(score_func=chi2,k=8)
```

####Train the model

```
In [32]: Transformer5=DecisionTreeClassifier()
```

Create a pipeline object

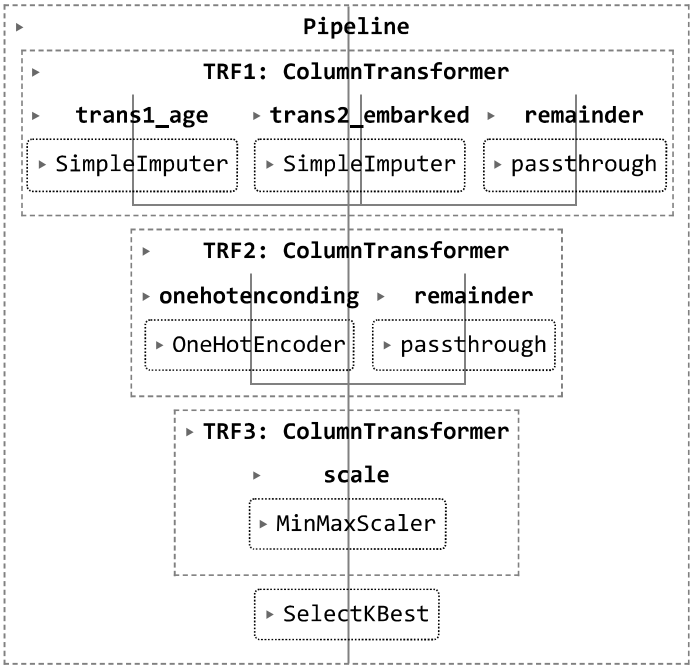
```
In [30]: pipe=Pipeline([('TRF1',Transformer1),
                       ('TRF2',Transformer2),
                       ('TRF3',Transformer3),
                       ('TRF4',Transformer4)])
```

train under pipeline model

In [31]:

```
pipe.fit(X_train,y_train)
```

Out[31]:



Exploring pipe function

In [ ]:

```
```