```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('/content/train.csv')
```

```
df.sample(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
696	697	0	3	Kelly, Mr. James	male	44.0	0	0	363592	8.0500	NaN	S
727	728	1	3	Mannion, Miss. Margareth	female	NaN	0	0	36866	7.7375	NaN	Q
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S
118	119	0	1	Baxter, Mr. Quigg Edmond	male	24.0	0	1	PC 17558	247.5208	B58 B60	C
848	849	0	2	Harper, Rev. John	male	28.0	0	1	248727	33.0000	NaN	S

```
df.head()
```

	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

```
df=df.drop(['PassengerId','Name','Ticket','Cabin'],axis=1)
```

```
df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

```
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
```

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline,make_pipeline
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.feature_selection import SelectKBest,chi2
from sklearn.preprocessing import MinMaxScaler
```

let's plan

missing value impute --> onehotencoder --> scaling --> feature selection (top 8/5)--> decisiontree -->

```
x_train,x_test,y_train,y_test=train_test_split(df.drop(columns=['Survived']),df['Survived'],test_size=0.2,random_state=42)
```

```
x_train
```

	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
...	...	...	...	...	...	...	...
106	3	female	21.0	0	0	7.6500	S
270	1	male	NaN	0	0	31.0000	S
860	3	male	41.0	2	0	14.1083	S
435	1	female	14.0	1	2	120.0000	S
102	1	male	21.0	0	1	77.2875	S

712 rows × 7 columns

```
x_test.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
709	3	male	NaN	1	1	15.2458	C
439	2	male	31.0	0	0	10.5000	S
840	3	male	20.0	0	0	7.9250	S
720	2	female	6.0	0	1	33.0000	S
39	3	female	14.0	1	0	11.2417	C

```
y_train.head()
```

	Survived
331	0
733	0
382	0
704	0
813	0

dtype: int64

```
# imputation of null values
df.isnull().sum()
```

	0
Survived	0
Pclass	0
Sex	0
Age	177
SibSp	0
Parch	0
Fare	0
Embarked	2

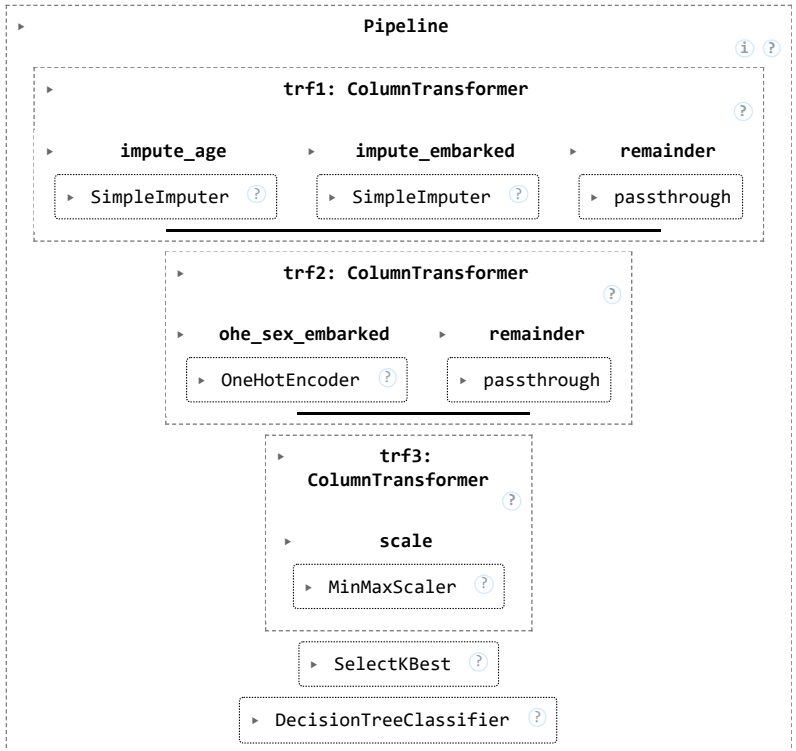
dtype: int64

```
trf1=ColumnTransformer([
    ('impute_age',SimpleImputer(),[2]),
    ('impute_embarked',SimpleImputer(strategy='most_frequent'),[6])
],remainder='passthrough')
```

```
# ohe
trf2=ColumnTransformer([
    ('ohe_sex_embarked',OneHotEncoder(sparse_output=False,handle_unknown='ignore'),[1,6])
],remainder='passthrough')
```

```
# scaling
trf3=ColumnTransformer([
    ('scale',MinMaxScaler(),slice(0,10))
])
```

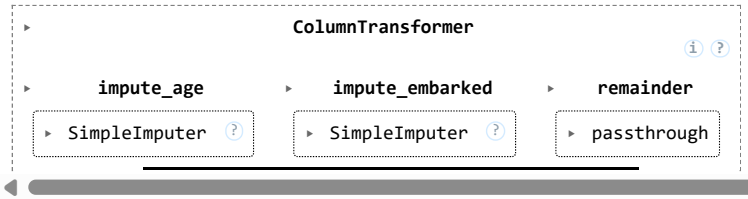
```
#feature selection
trf4=SelectKBest(score_func=chi2,k=8)


# decision tree
trf5=DecisionTreeClassifier()


# Display Pipeline

from sklearn import set_config
set_config(display='diagram')


# now apply the pipeline
pipe=Pipeline([
    ('trf1',trf1),
    ('trf2',trf2),
    ('trf3',trf3),
    ('trf4',trf4),
    ('trf5',trf5)
])


# Alternate Syntax
# pipe = make_pipeline(trf1,trf2,trf3,trf4,trf5)




# train
pipe.fit(x_train,y_train)
```




```
pipe.named_steps
```

```
{'trf1': ColumnTransformer(remainder='passthrough',
                   transformers=[('impute_age', SimpleImputer(), [2]),
                                 ('impute_embarked',
                                  SimpleImputer(strategy='most_frequent'),
                                  [6])]),
 'trf2': ColumnTransformer(remainder='passthrough',
                   transformers=[('ohe_sex_embarked',
                                  OneHotEncoder(handle_unknown='ignore',
                                                sparse_output=False),
                                  [1, 6])]),
 'trf3': ColumnTransformer(transformers=[('scale', MinMaxScaler(), slice(0, 10, None))]),
 'trf4': SelectKBest(k=8, score_func=<function chi2 at 0x7891dc4bb1a0>),
 'trf5': DecisionTreeClassifier()}
```

```
pipe.named_steps['trf1']
```



```
# Predict
y_pred = pipe.predict(x_test)
```

```
y_pred
```

```
array([1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1,
       0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
       0, 0, 0])
```

```
accuracy_score(y_test,y_pred)# low due to feature selection
```

```
0.6256983240223464
```

cross validation pipeline give mean of all accrurancy score

```
from sklearn.model_selection import cross_val_score
cross_val_score(pipe, x_train, y_train, cv=5, scoring='accuracy').mean()
```

```
np.float64(0.6391214419383433)
```

exporting pipeline