In [207…

```
import numpy as np
import pandas as pd
```

In [208…

```
from sklearn.model_selection import train_test
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncode
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline,make_pip
from sklearn.feature_selection import SelectKB
from sklearn.tree import DecisionTreeClassifie
```

In [209…

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

In [210…

```
df.head()
```

Out[210…

	PassengerId	Survived	Pclass	Name	Sex	Ag
0	1	0	3	Braund, Mr. Owen Harris	male	22.
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.
2	3	1	3	Heikkinen, Miss. Laina	female	26.
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.
4	5	0	3	Allen, Mr. William Henry	male	35.

Let's Plan

In [211…

```
df.drop(columns=['PassengerId','Name','Ticket'
```

In [212…

```
# Step 1 -> train/test/split
```

```
X_train,X_test,y_train,y_test = train_test_spl
```

In [213…

```
X_train.head()
```

Out[213…

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

In [197…

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

Out[197…

```
492    0
265    0
239    0
386    0
240    0
Name: Survived, dtype: int64
```

In [214…

```
# imputation transformer
trf1 = ColumnTransformer([
    ('impute_age',SimpleImputer(),[2]),
    ('impute_embarked',SimpleImputer(strategy=
],remainder='passthrough')
```

In [215…

```
# one hot encoding
trf2 = ColumnTransformer([
    ('ohe_sex_embarked',OneHotEncoder(sparse=F
],remainder='passthrough')
```

In [216…

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

In [217…

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

In [218…

```
# train the model
trf5 = DecisionTreeClassifier()
```

Create Pipeline

In [219…

```
pipe = Pipeline([
    ('trf1',trf1),
    ('trf2',trf2),
    ('trf3',trf3),
    ('trf4',trf4),
    ('trf5',trf5)
])
```

Pipeline Vs make_pipeline

Pipeline requires naming of steps, make_pipeline does not.

(Same applies to ColumnTransformer vs make_column_transformer)

In [ ]:

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

In [220…

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

Out[220…

```
Pipeline
Pipeline(steps=[('trf1',
                 ColumnTransformer(remain
der='passthrough',
                                   transf
ormers=[('impute_age', SimpleImputer(),

[2]),

('impute_embarked',

SimpleImputer(strategy='most_frequent'),

[6])])),
                ('trf2',
                 ColumnTransformer(remain
der='passthrough',
                                   transf
ormers=[('ohe_sex_embarked',

OneHotEncoder(handle_unknown='ignore',
```

```
sparse=False),

[1, 6])])),
                ('trf3',
                 ColumnTransformer(transf
ormers=[('scale', MinMaxScaler(),

slice(0, 10, None))])),
                ('trf4',
                 SelectKBest(k=8,
                             score_func
=)),
                ('trf5', DecisionTreeClas
sifier())])
```

trf1: ColumnTransformer

```
ColumnTransformer(remainder='passthroug
h',
                  transformers=[('impute_
age', SimpleImputer(), [2]),
                                ('impute_
embarked',
                                 SimpleIm
puter(strategy='most_frequent'),
                                 [6])])
```

impute_age

```
[2]
```

SimpleImputer

```
SimpleImputer()
```

impute_embarked

```
[6]
```

SimpleImputer

```
SimpleImputer(strategy='most_frequent')
```

trf2: ColumnTransformer

```
ColumnTransformer(remainder='passthroug
h',
                  transformers=[('ohe_sex
_embarked',
                                 OneHotEn
coder(handle_unknown='ignore',

sparse=False),
                                 [1,
6])])
```

ohe_sex_embarked

```
[1, 6]
```

OneHotEncoder

```
OneHotEncoder(handle_unknown='ignore', sp
arse=False)
```

trf3: ColumnTransformer

```
ColumnTransformer(transformers=[('scale',
MinMaxScaler(), slice(0, 10, None))])
```

scale
slice(0, 10, None)
MinMaxScaler
MinMaxScaler()
SelectKBest
SelectKBest(k=8, score_func=)
DecisionTreeClassifier
DecisionTreeClassifier()

Explore the Pipeline

In [232…
```
# Code here
pipe.named_steps
```

Out[232…
```
{'trf1': ColumnTransformer(remainder='passthrou
gh',
                   transformers=[('impute_age',
SimpleImputer(), [2]),
                                 ('impute_embar
ked',
                                  SimpleImputer
(strategy='most_frequent'),
                                  [6])]),
 'trf2': ColumnTransformer(remainder='passthrou
gh',
                   transformers=[('ohe_sex_emba
rked',
                                  OneHotEncoder
(handle_unknown='ignore',

sparse=False),
                                  [1, 6])]),
 'trf3': ColumnTransformer(transformers=[('scal
e', MinMaxScaler(), slice(0, 10, None))]),
 'trf4': SelectKBest(k=8, score_func=),
 'trf5': DecisionTreeClassifier()}
```

In [204…
```
# Display Pipeline

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

In [233…
```
# Predict
y_pred = pipe.predict(X_test)
```

In [234…
```
y_pred
```

Out[234…
```
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], dtype=int64)
```

In [235…

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[235…

```
0.6256983240223464
```

Cross Validation using Pipeline

In [236…

```
# cross validation using cross_val_score
from sklearn.model_selection import cross_val_
cross_val_score(pipe, X_train, y_train, cv=5,
```

Out[236…

```
0.6391214419383433
```

GridSearch using Pipeline

In [237…

```
# gridsearchcv
params = {
    'trf5__max_depth':[1,2,3,4,5,None]
}
```

In [238…

```
from sklearn.model_selection import GridSearch
grid = GridSearchCV(pipe, params, cv=5, scorin
grid.fit(X_train, y_train)
```

Out[238…

GridSearchCV

```
GridSearchCV(cv=5,
             estimator=Pipeline(steps=
[('trf1',
                                       C
olumnTransformer(remainder='passthrough',

transformers=[('impute_age',
```

```
            SimpleImputer(),
            [2]),
            ('impute_embarked',
            SimpleImputer(strategy='most_frequent'),
            [6])])),
            ('trf2',
                                                C
            olumnTransformer(remainder='passthrough',
```