

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

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

```
df.head()
```

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
0	2022-05-18	2	10000	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2	Contact Owner
1	2022-05-13	2	20000	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1	Contact Owner
2	2022-05-16	2	17000	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1	Contact Owner
3	2022-07-04	2	10000	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1	Contact Owner
4	2022-05-09	2	7500	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1	Contact Owner

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
df = df.drop(columns=['Posted On' , 'Floor','Area Locality'])
```

```
df.head()
```

	BHK	Rent	Size	Area Type	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
0	2	10000	1100	Super Area	Kolkata	Unfurnished	Bachelors/Family	2	Contact Owner
1	2	20000	800	Super Area	Kolkata	Semi-Furnished	Bachelors/Family	1	Contact Owner
2	2	17000	1000	Super Area	Kolkata	Semi-Furnished	Bachelors/Family	1	Contact Owner
3	2	10000	800	Super Area	Kolkata	Unfurnished	Bachelors/Family	1	Contact Owner
4	2	7500	850	Carpet Area	Kolkata	Unfurnished	Bachelors	1	Contact Owner

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
from sklearn.model_selection import train_test_split
x_train , x_test , y_train, y_test = train_test_split(df.drop(columns=['Rent']) , df['Rent'],test_size=0.2 , random_state=0)
```

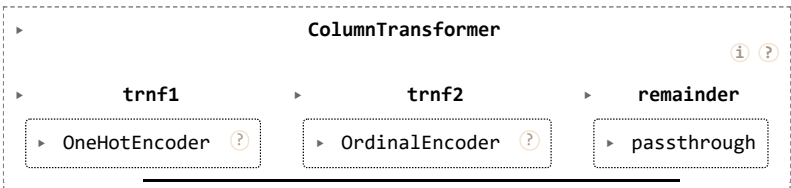
```
x_train.head()
```

	BHK	Size	Area Type	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
4681	2	700	Super Area	Hyderabad	Unfurnished	Bachelors/Family	2	Contact Owner
630	2	650	Carpet Area	Mumbai	Furnished	Family	2	Contact Agent
1742	3	1200	Super Area	Bangalore	Semi-Furnished	Bachelors/Family	2	Contact Owner
3077	1	560	Super Area	Chennai	Unfurnished	Bachelors/Family	1	Contact Owner
2996	2	600	Super Area	Chennai	Furnished	Bachelors/Family	2	Contact Owner

Next steps: [Generate code with x_train](#) [New interactive sheet](#)

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder , OrdinalEncoder
transformer = ColumnTransformer(transformers=(
    ('trnf1' , OneHotEncoder(sparse_output=False ,drop='first'),['City' , 'Tenant Preferred','Point of Contact']),
    ('trnf2' , OrdinalEncoder(categories=[
        ['Built Area' , 'Carpet Area','Super Area'],
        ['Unfurnished','Semi-Furnished','Furnished']]),
    ['Area Type' , 'Furnishing Status']
    )
),remainder= 'passthrough'
)
```

```
transformer
```



```
x_train_new = transformer.fit_transform(x_train)
```

```
x_test_new = transformer.transform(x_test)
```

```
from sklearn.pipeline import Pipeline , make_pipeline
```

```
pipe = Pipeline([
    ('trfn' , transformer)
])
```

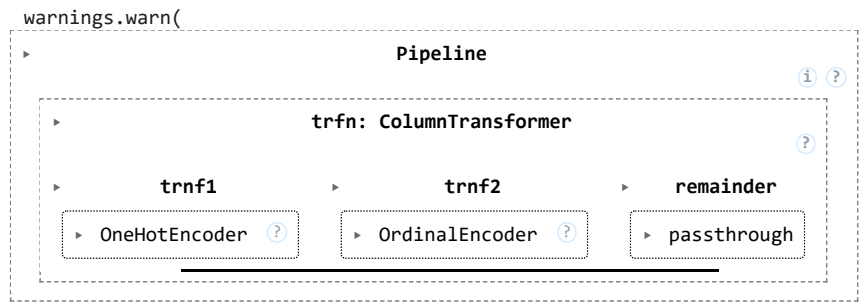
```
pipe.fit(x_train,y_train)
```

McAfee | WebAdvisor

×

Your download's being scanned.
We'll let you know if there's an issue.

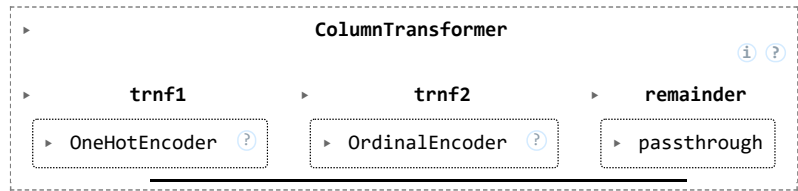
/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transformer.py:1667: FutureWarning:
The format of the columns of the 'remainder' transformer in ColumnTransformer.transformers_ will change in version 1.7 to match the format of the other transformers.
At the moment the remainder columns are stored as indices (of type int). With the same ColumnTransformer configuration, in the future they will be stored as column name
To use the new behavior now and suppress this warning, use ColumnTransformer(force_int_remainder_cols=False).



pipe.named_steps

```
{'trfn': ColumnTransformer(remainder='passthrough',
    transformers=((('trnf1',
        OneHotEncoder(drop='first',
                      sparse_output=False),
        ['City', 'Tenant Preferred',
         'Point of Contact']),
        ('trnf2',
        OrdinalEncoder(categories=[['Built Area',
                                   'Carpet Area',
                                   'Super Area'],
                               ['Unfurnished',
                                'Semi-Furnished',
                                'Furnished']]),
        ['Area Type', 'Furnishing Status'])))})
```


pipe.named_steps['trfn']




pipe.named_steps['trfn'].transformers_

```
[('trnf1',
  OneHotEncoder(drop='first', sparse_output=False),
  ['City', 'Tenant Preferred', 'Point of Contact']),
 ('trnf2',
  OrdinalEncoder(categories=[['Built Area', 'Carpet Area', 'Super Area'],
                             ['Unfurnished', 'Semi-Furnished', 'Furnished']]),
  ['Area Type', 'Furnishing Status']),
 ('remainder',
  FunctionTransformer(accept_sparse=True, check_inverse=False,
                      feature_names_out='one-to-one'),
  [0, 1, 6])]
```

```
import pickle
pickle.dump(pipe,open('pipe.pkl','wb'))
```

 **McAfee** | WebAdvisor



×

Your download's being scanned.
We'll let you know if there's an issue.