📦 Import Required Libraries

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

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

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

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	Tenant Preferred
514	2022-06-28	2	10000	721	3 out of 4	Super Area	Belgharia Nimta	Kolkata	Furnished	Bachelors/Fa
1764	2022-06-23	2	10000	1000	1 out of 3	Super Area	Varthur	Bangalore	Semi-Furnished	Bachelors/Fa
2987	2022-07-06	2	15000	1100	1 out of 2	Super Area	Medavakkam	Chennai	Semi-Furnished	Bache
763	2022-05-31	2	53500	635	17 out of 20	Carpet Area	Nahar Jonquille and Jamaica, Chandivali	Mumbai	Unfurnished	Bachelors/Fa

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

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

	BHK	Rent	Size	Area Type	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
489	2	18000	900	Carpet Area	Kolkata	Semi-Furnished	Bachelors/Family	2	Contact Owner
4089	3	18000	1400	Super Area	Hyderabad	Unfurnished	Bachelors/Family	3	Contact Owner
4673	2	8000	800	Super Area	Hyderabad	Furnished	Bachelors/Family	2	Contact Owner

```
x=df.drop(columns=['Rent'])
```

```
y=df['Rent']
```

🔀 Train–Test Split

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
```

```
x_train.sample(10)
```

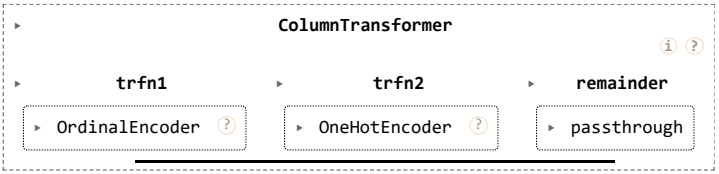
	BHK	Size	Area Type	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
211	1	500	Carpet Area	Kolkata	Furnished	Bachelors	1	Contact Agent
439	1	450	Carpet Area	Kolkata	Unfurnished	Bachelors/Family	1	Contact Owner
3906	2	100	Super Area	Hyderabad	Furnished	Family	2	Contact Owner
509	3	1700	Carpet Area	Kolkata	Semi-Furnished	Bachelors	3	Contact Agent
1020	1	500	Carpet Area	Mumbai	Furnished	Bachelors/Family	1	Contact Agent
2432	2	2200	Carpet Area	Delhi	Unfurnished	Bachelors	3	Contact Agent

ColumnTransformer: Categorical Feature Encoding

```
from sklearn.preprocessing import OneHotEncoder,OrdinalEncoder
```

```
from sklearn.compose import ColumnTransformer
transformer=ColumnTransformer(transformers=(
    ('trfn1',OrdinalEncoder(categories=[['Built Area','Carpet Area','Super Area'],
                                        ['Unfurnished','Semi-Furnished','Furnished']]),
                                        ['Area Type','Furnishing Status']),
    ('trfn2',OneHotEncoder(sparse_output=False,drop='first'),['City','Tenant Preferred','Poi
),remainder='passthrough')
```

```
transformer
```




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

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

Preprocessing Pipeline & Model Serialization

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

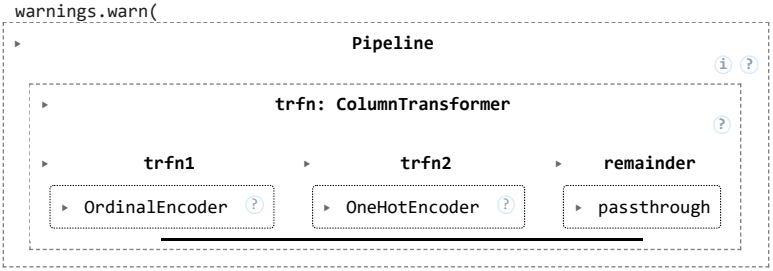
```
pipeline=Pipeline([
    ('trfn',transformer)
```

```
])
```

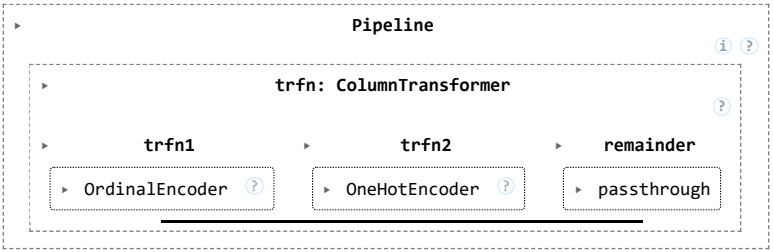
```
pipeline.fit(x_train,y_train)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transformer.py:1667: FutureWa
The format of the columns of the 'remainder' transformer in ColumnTransformer.transformers_ w
At the moment the remainder columns are stored as indices (of type int). With the same Column
To use the new behavior now and suppress this warning, use ColumnTransformer(force_int_remain

  warnings.warn(
```




```
pipeline
```




```
pipeline.named_steps['trfn'].transformers_
```

```
[('trfn1',
  OrdinalEncoder(categories=[['Built Area', 'Carpet Area', 'Super Area'],
                             ['Unfurnished', 'Semi-Furnished', 'Furnished']]),
  ['Area Type', 'Furnishing Status']),
 ('trfn2',
  OneHotEncoder(drop='first', sparse_output=False),
  ['City', 'Tenant Preferred', 'Point of Contact']),
 ('remainder',
  FunctionTransformer(accept_sparse=True, check_inverse=False,
                      feature_names_out='one-to-one'),
  [0, 1, 6])]
```

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

```
Start coding or generate with AI.
```