6-data_preprocessing_with_column_transformer

October 20, 2024

1 Data Preprocessing With Column Transformer

The ColumnTransformer in Python's sklearn library is a powerful tool that allows you to apply different preprocessing steps to different columns of your dataset. This is particularly useful when you have a mixture of numerical and categorical data and need to apply distinct transformations to each type of data. Using the ColumnTransformer, you can combine transformations like scaling numerical features, encoding categorical variables, and imputing missing values, all in a single unified process.

Why Use ColumnTransformer?

1. **Multiple Data Types**: It handles datasets with mixed data types (numerical, categorical, text, etc.).
2. **Efficient Pipeline**: You can combine different preprocessing steps into a pipeline, making the code cleaner and easier to maintain.
3. **Selective Preprocessing**: Apply different preprocessing steps to specific columns, instead of preprocessing the entire dataset uniformly.
4. **Integration with Pipelines**: It can be combined with machine learning models in a Pipeline, allowing for a streamlined workflow from preprocessing to model training and evaluation.

[2]:
```
import pandas as pd

d = {'sales':␣
 ↪[100000,222000,1000000,522000,111111,222222,1111111,20000,75000,90000,1000000,10000],␣
 ↪'city':␣
 ↪['Tampa','Tampa','Orlando','Jacksonville','Miami','Jacksonville','Miami','Miami','Orlando',
 ↪'size': ['Small',␣
 ↪'Medium','Large','Large','Small','Medium','Large','Small','Medium','Medium','Medium','Small

df = pd.DataFrame(data=d)
df
```

[2]:

	sales	city	size
0	100000	Tampa	Small
1	222000	Tampa	Medium
2	1000000	Orlando	Large

```
3    522000  Jacksonville   Large
4    111111         Miami   Small
5    222222  Jacksonville  Medium
6   1111111         Miami   Large
7     20000         Miami   Small
8     75000       Orlando  Medium
9     90000       Orlando  Medium
10  1000000       Orlando  Medium
11    10000       Orlando   Small
```

[7]:
```
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.compose import make_column_transformer

ohe = OneHotEncoder(sparse_output=False)
ode = OrdinalEncoder()

ct = make_column_transformer(
    (ohe, ['city']),
    (ode, ['size']),
    remainder='passthrough'
)

ct.set_output(transform='pandas')
```

[7]:
```
ColumnTransformer(remainder='passthrough',
                  transformers=[('onehotencoder',
                                 OneHotEncoder(sparse_output=False), ['city']),
                                ('ordinalencoder', OrdinalEncoder(), ['size'])])
```

[8]:
```
df_pandas = ct.fit_transform(df)
df_pandas
```

[8]:
```
    onehotencoder__city_Jacksonville  onehotencoder__city_Miami  \
0                                0.0                        0.0
1                                0.0                        0.0
2                                0.0                        0.0
3                                1.0                        0.0
4                                0.0                        1.0
5                                1.0                        0.0
6                                0.0                        1.0
7                                0.0                        1.0
8                                0.0                        0.0
9                                0.0                        0.0
10                               0.0                        0.0
11                               0.0                        0.0

    onehotencoder__city_Orlando  onehotencoder__city_Tampa  \
```

```
0                            0.0                          1.0
1                            0.0                          1.0
2                            1.0                          0.0
3                            0.0                          0.0
4                            0.0                          0.0
5                            0.0                          0.0
6                            0.0                          0.0
7                            0.0                          0.0
8                            1.0                          0.0
9                            1.0                          0.0
10                           1.0                          0.0
11                           1.0                          0.0

    ordinalencoder__size  remainder__sales
0                    2.0            100000
1                    1.0            222000
2                    0.0           1000000
3                    0.0            522000
4                    2.0            111111
5                    1.0            222222
6                    0.0           1111111
7                    2.0             20000
8                    1.0             75000
9                    1.0             90000
10                   1.0           1000000
11                   2.0             10000
```

[9]:
```
ct2 = make_column_transformer(
    (ohe, [1]),
    (ode, [2]),
    remainder='drop'
)

ct2.set_output(transform='pandas')
```

[9]:
```
ColumnTransformer(transformers=[('onehotencoder',
                                 OneHotEncoder(sparse_output=False), [1]),
                                ('ordinalencoder', OrdinalEncoder(), [2])])
```

[11]:
```
df_pandas2 = ct2.fit_transform(df)
df_pandas2
```

[11]:
```
    onehotencoder__city_Jacksonville  onehotencoder__city_Miami  \
0                                0.0                        0.0
1                                0.0                        0.0
2                                0.0                        0.0
3                                1.0                        0.0
```

```
4                                 0.0                          1.0
5                                 1.0                          0.0
6                                 0.0                          1.0
7                                 0.0                          1.0
8                                 0.0                          0.0
9                                 0.0                          0.0
10                                0.0                          0.0
11                                0.0                          0.0

    onehotencoder__city_Orlando  onehotencoder__city_Tampa  \
0                           0.0                         1.0
1                           0.0                         1.0
2                           1.0                         0.0
3                           0.0                         0.0
4                           0.0                         0.0
5                           0.0                         0.0
6                           0.0                         0.0
7                           0.0                         0.0
8                           1.0                         0.0
9                           1.0                         0.0
10                          1.0                         0.0
11                          1.0                         0.0

    ordinalencoder__size
0                    2.0
1                    1.0
2                    0.0
3                    0.0
4                    2.0
5                    1.0
6                    0.0
7                    2.0
8                    1.0
9                    1.0
10                   1.0
11                   2.0
```

[12]:
```
ct3 = make_column_transformer(
    (ohe, [1]),
    ('passthrough', ['size']),
    remainder='drop'
)

ct3.set_output(transform='pandas')
```

[12]:
```
ColumnTransformer(transformers=[('onehotencoder',
                                 OneHotEncoder(sparse_output=False), [1]),
```

```
                                  ('passthrough', 'passthrough', ['size'])])
```

[13]:
```
ct3.fit_transform(df)
```

[13]:

	onehotencoder__city_Jacksonville	onehotencoder__city_Miami	onehotencoder__city_Orlando	onehotencoder__city_Tampa	passthrough__size
0	0.0	0.0	0.0	1.0	Small
1	0.0	0.0	0.0	1.0	Medium
2	0.0	0.0	1.0	0.0	Large
3	1.0	0.0	0.0	0.0	Large
4	0.0	1.0	0.0	0.0	Small
5	1.0	0.0	0.0	0.0	Medium
6	0.0	1.0	0.0	0.0	Large
7	0.0	1.0	0.0	0.0	Small
8	0.0	0.0	1.0	0.0	Medium
9	0.0	0.0	1.0	0.0	Medium
10	0.0	0.0	1.0	0.0	Medium
11	0.0	0.0	1.0	0.0	Small