

25-08-25

Expt No 5 - Experiment to understand feature scaling

Description: To understand the importance of feature scaling by applying standardization and normalization.

To understand the importance of feature scaling by applying standardization and normalization (Min-Max scaling) on dataset.

3. Handle missing values.

4. Encode categorical data.

5. Combine preprocessing steps.

6. Apply scaler on numeric features.

7. Display scaled results.

8. Print feature names.

9. Print feature scale.

10. Print feature names.

(1 = 0.12)

first & working) treatment of missing values

to import numpy as np and pandas as pd
import pandas as pd

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OneHotEncoder,
StandardScaler, MinMaxScaler

data = {'Country': ['France', 'Spain', 'Germany', 'Spain',
'Germany', 'France', 'Spain', 'France', np.nan,

'France'], 'Age': [49, 27, 30, 38, np.nan, 33, np.nan,
48, 50, 37], 'Salary': [20000, 45000, 54000, 61000,
np.nan, 58000, 52000, 79000, 83000, 62000],

'Purchased': ['No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No',
'Yes', 'No', 'Yes']}

df = pd.DataFrame(data)

df['Country'].fillna(df['Country'].mode()[0], inplace=True)

features = df.iloc[:, :-1].values

imputer = SimpleImputer(strategy='mean')

features[:, 1:] = imputer.fit_transform(features[:, 1:])

oh = OneHotEncoder(sparse=False)

Country = oh.fit_transform(features[:, [0]])

final_set = np.concatenate((Country, features[:, 1:]), axis=1)

axis=1)

```
sc = StandardScaler()
standard_scaled = mms.fit_transform(final_set)
print("Original Data: \n", final_set)
print("\n Standard Scaled Data: \n", standard_scaled)
print("\n Min-Max Scaled Data: \n", minmax_scaled)
```

Result: Thus, the dataset was preprocessed and scaled using standardization and min-max scaling.