

25-08-25

Expt No. 5 - Experiment to understand feature scaling

Description: To understand the importance of feature scaling

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

Algorithm

1. Import necessary libraries (numpy, pandas, sklearn)

2. Load the dataset

3. Handle missing values

4. Encode categorical data

5. Combine preprocessing steps

6. Apply scaler on numeric features

7. Display scaled results

8. Save the scaled dataset

(1 = 100)

Program:

import numpy as np
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': [44, 27, 30, 28, 40, 35, np.nan,

48, 50, 37], 'Salary': [72000, 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[:, 1:3] = imputer.fit_transform(features[:, 1:3])

oh = OneHotEncoder(sparse_output=False)

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

final_set = np.concatenate((Country, features[:, 1:3]),

axis=1)

```
sc = StandardScaler()
standard_scaled = mmr.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.