Certificate Course in Machine Learning using Python [6 Weeks]

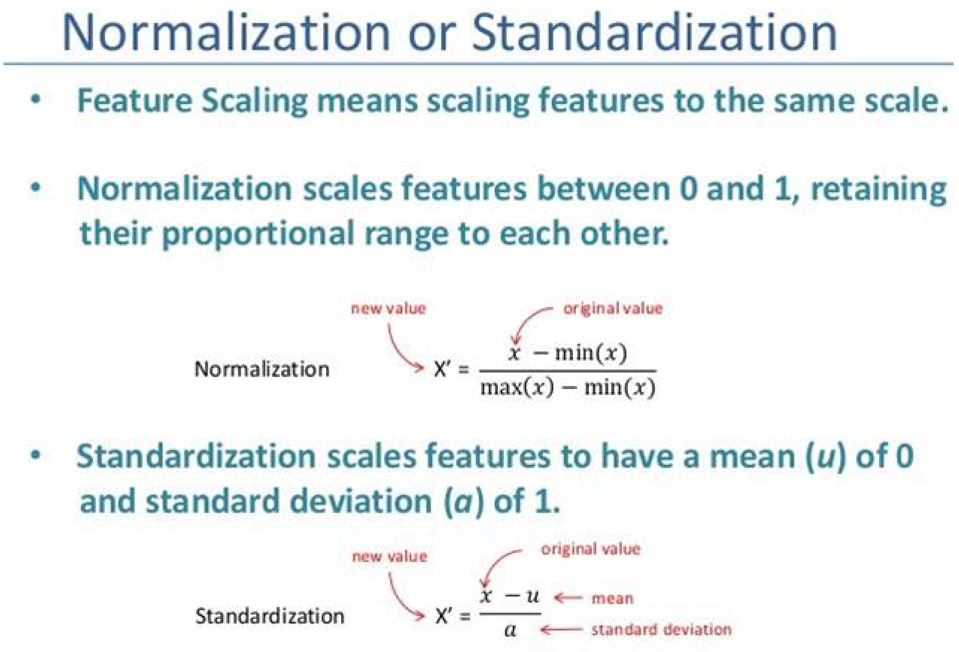
Dashboard › My courses › Certificate Course in Machine Learning using Python [6 Weeks] › Day 11 › Pre-processing of data for Machine Learning

Pre-processing of data for Machine Learning

Data Pre-processing for Machine Learning

Step 6: Feature Scaling

- Real world dataset contains features that highly vary in magnitudes, units, and range.
- Differences in the scales across input variables may increase the difficulty of the problem being modelled. An example of this is that large input values (e.g. a spread of hundreds or thousands of units) can result in a model that learns large weight values.
- A model with large weight values is often unstable, meaning that it may suffer from poor performance during learning and sensitivity to input values resulting in higher generalization error.
- Feature Scaling or Standardization is a step of Data Pre Processing which is applied to independent variables or features of data.
- It basically helps to normalise the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.



Normalization or Standardization

- Feature Scaling means scaling features to the same scale.
- Normalization scales features between 0 and 1, retaining their proportional range to each other.

Normalization
$$X' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

new value
original value

- Standardization scales features to have a mean (u) of 0 and standard deviation (a) of 1.

Standardization
$$X' = \frac{x - u}{a}$$

new value
original value
mean
standard deviation

StandardScaler

- StandardScaler performs the task of Standardization. Usually a dataset contains variables that are different in scale. For e.g. an Employee dataset will contain AGE column with values on scale 20-70 and SALARY column with values on scale 10000-80000.
- As these two columns are different in scale, they are Standardized to have common scale while building machine learning model.
- Scaling is done for numerical values only. Categorical values are not scaled.

Example:

```
 from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

#scaling ‘Age’ and ‘Salary’ Column only

X[['Age','Salary']] = scaler.fit_transform(X[['Age','Salary']])
```

MinMaxScaler

- Transform features by scaling each feature to a given range.
- This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

Example:

```
  from sklearn.preprocessing import MinMaxScaler

  scalerX = MinMaxScaler(feature_range=(0, 1))

  X[['Age','Salary']] = scalerX.fit_transform(X[['Age','Salary']])

 X
```

Step 7: Splitting the Dataset into Training set and Test Set

- One important aspect of all machine learning models is to determine their accuracy. Now, in order to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that model and hence, find the accuracy of the model.
- A better option is to split our data into two parts: first one for training our machine learning model, and second one for testing our model.
- Train the model on the training set.
- Test the model on the testing set, and evaluate how well our model did.

train_test_split: splits the data into two sets: train and test.

It returns four datasets: X_train, X_test, y_train, y_test.

Parameters:

- **test_size:** This parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.8 as the value, the dataset will be split 80% as the test dataset.
- **random_state:** Here you pass an integer, which will act as the seed for the random number generator during the split.

Example:

- **Now X and y is ready. Spilt the data in two parts: train data and test data as:**

```
 from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test =train_test_split(X,y,test_size = 0.20, random_state=42)
```

- The 80% of data will be assigned as training data and remaining 20% of data will be assigned as testing data.

Previous Next

PREVIOUS ACTIVITY

◀ Data File: Admission_Predict.csv: Day 10

Jump to...

NEXT ACTIVITY

Python code: Regression Problem:Apply Different Algorithm: Day 11 ▶

Stay in touch

Contact Us

http://nielit.gov.in/gorakhpur/

abhinav@nielit.gov.in or ajay.verma@nielit.gov.in