AI Seminar

Week 4

Overview

Feature Scaling

Normalization

Standardization

Regression Model Evaluation

Feature Scaling - Preprocessing of data

Features may be varying in magnitudes, units and range. (Kg, sq.ft, sq.mt, 1 millions, 10,00,000, $) (age and income)

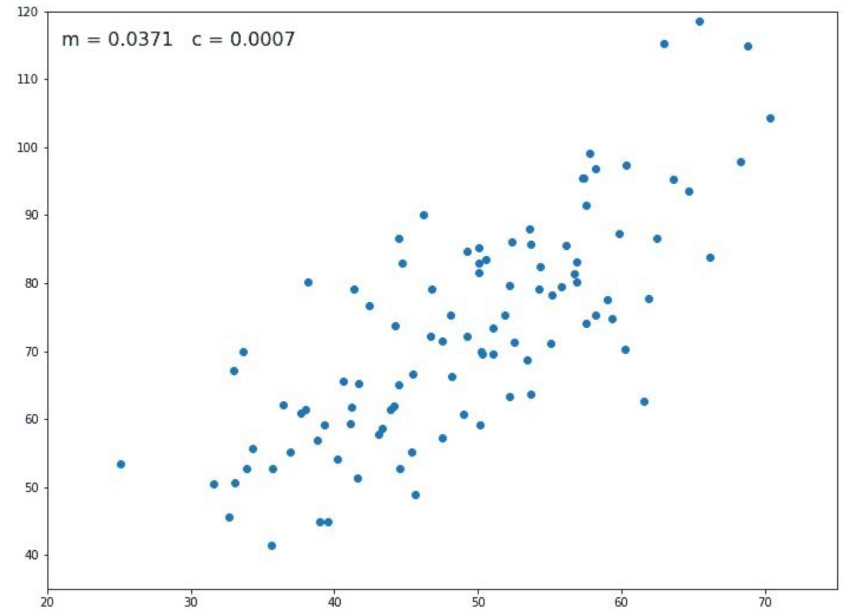
To avoid any variable from dominating over other variables

Can take longer learning time to build model.

Feature Scale Sensitive ML:

- Gradient Descent based (Linear Regression, Logistic Regression, Neural Networks)
- Distance Based algorithms (KNN, SVM, K-means)
- Principal Component Analysis (PCA)

Gradient Descent



The values of m and c are updated at each iteration to get the optimal solution

Credit:
https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931

> *In practice it is nearly always advantageous to apply pre-processing transformations to the input data before it is presented to a network. Similarly, the outputs of the network are often post-processed to give the required output values.*

— Page 296, Neural Networks for Pattern Recognition, 1995.

Feature Scaling

Normalization

Standardization

Normalization

Normalization aims to transform features to be on a similar scale.

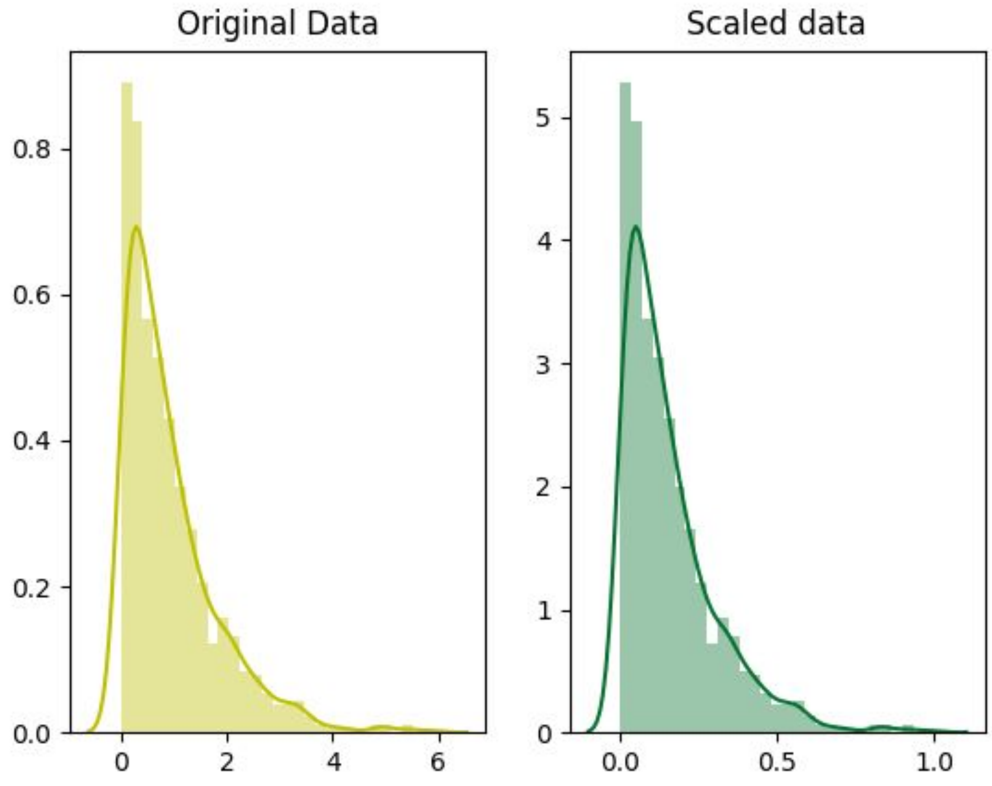
Min-Max

Rescaling of the data from the original range so that all values are within the range of 0 and 1.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, Xmax and Xmin are the maximum and the minimum values of the feature respectively.

Min-Max Scaling



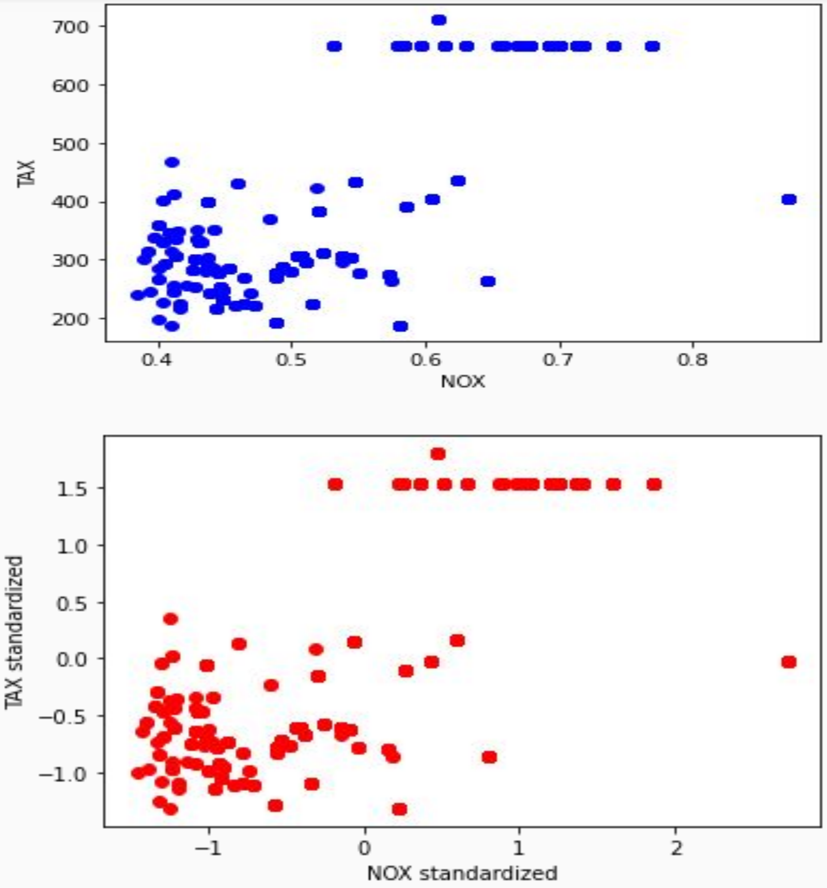
Standardization

Standardization (also called z-score) transforms data to a resulting distribution which has mean of 0 and a standard deviation of 1.

$$x' = \frac{x - \bar{x}}{\sigma}$$

Where σ is the standard deviation and x̄ is the mean.

Standardization



Scikit Learn API

```
from sklearn.preprocessing import StandardScaler
data = [[0, 0], [0, 0], [1, 1], [1, 1]]
scaler = StandardScaler()
print(scaler.fit(data))

print(scaler.mean_)

print(scaler.transform(data))




print(scaler.transform([[2, 2]]))
```

Methods

fit(self, X[, y])	Compute the mean and std to be used for later scaling.
fit_transform(self, X[, y])	Fit to data, then transform it.
get_params(self[, deep])	Get parameters for this estimator.
inverse_transform(self, X[, copy])	Scale back the data to the original representation
partial_fit(self, X[, y])	Online computation of mean and std on X for later scaling.
set_params(self, **params)	Set the parameters of this estimator.
transform(self, X[, copy])	Perform standardization by centering and scaling

Scikit Learn API

```
>>> from sklearn.preprocessing import MinMaxScaler
>>> data = [[-1, 2], [-0.5, 6], [0, 10], [1, 18]]
>>> scaler = MinMaxScaler()
>>> print(scaler.fit(data))
MinMaxScaler()
>>> print(scaler.data_max_)
[ 1. 18.]
>>> print(scaler.transform(data))
[[0.   0.  ]
 [0.25 0.25]
 [0.5  0.5 ]
 [1.   1.  ]]
>>> print(scaler.transform([[2, 2]]))
[[1.5 0. ]]
```

Methods

fit(self, X[, y])	Compute the minimum and maximum to be used for later scaling.
fit_transform(self, X[, y])	Fit to data, then transform it.
get_params(self[, deep])	Get parameters for this estimator.
inverse_transform(self, X)	Undo the scaling of X according to feature_range.
partial_fit(self, X[, y])	Online computation of min and max on X for later scaling.
set_params(self, **params)	Set the parameters of this estimator.
transform(self, X)	Scale features of X according to feature_range.

Regression Model Evaluation

https://www.coursera.org/lecture/machine-learning-with-python/evaluation-metrics-in-regression-models-5SxtZ

Hands-On

Week 4 Jupyter NoteBook:

Next Week

Classification

References

1) "Feature Scaling- Why it is required?"
https://medium.com/@rahul77349/feature-scaling-why-it-is-required-8a93df1af310

2) "Understand Data Normalization in Machine Learning":

https://towardsdatascience.com/understand-data-normalization-in-machine-learning-8ff3062101f0

3) "How To Prepare Your Data For Machine Learning in Python with Scikit-Learn ",
https://machinelearningmastery.com/prepare-data-machine-learning-python-scikit-learn/

4) "How to use Data Scaling Improve Deep Learning Model Stability and Performance",
https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/