How to reduce dimentionality using PCA in Python?

This recipe helps you reduce dimentionality using PCA in Python

Last Updated: 08 Jul 2022

DATA MUNGING DATA CLEANING PYTHON MACHINE LEARNING RECIPES
PANDAS CHEATSHEET ALL TAGS

Recipe Objective

In many datasets we find that number of features are very large and if we want to train the model it take more computational cost. To decrease the number of features we can use Principal component analysis (PCA). PCA decrease the number of features by selecting dimension of features which have most of the variance.

So this recipe is a short example of how can reduce dimentionality using PCA in [Python.]

[Master the Art of Data Cleaning in Machine Learning]

Table of Contents

Step 1 - Import the library

```
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

Here we have imported various modules like PCA, datasets and StandardScale from differnt libraries. We will understand the use of these later while using it in the in the code snipet. For now just have a look on these imports.

Relevant Projects

Machine Learning Projects

Data Science Projects

Python Projects for Data Science

Data Science Projects in R

Machine Learning Projects for Beginners

Deep Learning Projects

Step 2 - Setup the Data

Here we have used datasets to load the inbuilt digits dataset.

```
digits = datasets.load_digits()
```

Step 3 - Using StandardScaler

StandardScaler is used to remove the outliners and scale the data by making the mean of the data 0 and standard deviation as 1.

```
X = StandardScaler().fit_transform(digits.data)
print(); print(X)
```

Step 4 - Using PCA

We are also using Principal Component Analysis(PCA) which will reduce the dimension of features by creating new features which have most of the varience of the original data. We have passed the parameter n_components as 0.85 which is the percentage of feature in final dataset. We have also printed shape of intial and final dataset.

```
pca = PCA(n_components=0.85, whiten=True)

X_pca = pca.fit_transform(X)
print(X_pca)

print("Original number of features:", X.shape[1])
print("Reduced number of features:", X_pca.shape[1])
```

Foe better understanding we are applying PCA again. Now We have passed the parameter n_components as 0.85 which is the percentage of feature in final dataset. We have also printed shape of intial and final dataset.

```
pca = PCA(n_components=2, whiten=True)

X_pca = pca.fit_transform(X)
print(X_pca)

print("Original number of features:", X.shape[1])
print("Reduced number of features:", X_pca.shape[1])
```

As an output we get:

[Neural Network Projects](#)
[Tensorflow Projects](#)
[NLP Projects](#)
[Kaggle Projects](#)
[IoT Projects](#)
[Big Data Projects](#)
[Hadoop Real-Time Projects Examples](#)
[Spark Projects](#)
[Data Analytics Projects for Students](#)

You might also like

[Data Science Tutorial](#)
[Data Scientist Salary](#)
[How to Become a Data Scientist](#)
[Data Analyst vs Data Scientist](#)
[Data Scientist Resume](#)
[Data Science Projects for Beginners](#)
[Machine Learning Engineer](#)
[Machine Learning Projects for Beginners](#)
[Datasets](#)
[Pandas Dataframe](#)
[Machine Learning Algorithms](#)
[Regression Analysis](#)
[MNIST Dataset](#)
[Data Science Interview Questions](#)
[Python Data Science Interview Questions](#)
[Spark Interview Questions](#)
[Hadoop Interview Questions](#)
[Data Analyst Interview Questions](#)
[Machine Learning Interview Questions](#)

```
[[ 0.         -0.33501649 -0.04308102 ... -1.14664746 -0.5056698
  -0.19600752]
 [ 0.         -0.33501649 -1.09493684 ...  0.54856067 -0.5056698
  -0.19600752]
 [ 0.         -0.33501649 -1.09493684 ...  1.56568555  1.6951369
  -0.19600752]
 ...
 [ 0.         -0.33501649 -0.88456568 ... -0.12952258 -0.5056698
  -0.19600752]
 [ 0.         -0.33501649 -0.67419451 ...  0.8876023  -0.5056698
  -0.19600752]
 [ 0.         -0.33501649  1.00877481 ...  0.8876023  -0.26113572
  -0.19600752]]

[[ 0.70631939 -0.39512814 -1.73816236 ...  0.60320435 -0.94455291
  -0.60204272]
 [ 0.21732591  0.38276482  1.72878893 ... -0.56722002  0.61131544
   1.02457999]
 [ 0.4804351  -0.13130437  1.33172761 ... -1.51284419 -0.48470912
  -0.52826811]
 ...
 [ 0.37732433 -0.0612296   1.0879821  ...  0.04925597  0.29271531
  -0.33891255]
 [ 0.39705007 -0.15768102 -1.08160094 ...  1.31785641  0.38883981
  -1.21854835]
 [-0.46407544 -0.92213976  0.12493334 ... -1.27242756 -0.34190284
  -1.17852306]]
Original number of features: 64
Reduced number of features: 25

[[ 0.70634542 -0.39504744]
 [ 0.21730901  0.38270788]
 [ 0.48044955 -0.13126596]
 ...
 [ 0.37733004 -0.06120936]
 [ 0.39703595 -0.15774013]
 [-0.46406594 -0.92210953]]
Original number of features: 64
Reduced number of features: 2
```

Download Materials

iPython Notebook

[view run code]

AWS vs Azure

Hadoop Architecture

Spark Architecture

Relevant Projects

End-to-End Speech Emotion Recognition Project using ANN

Speech Emotion Recognition using RAVDESS Audio Dataset - Build an Artificial Neural Network Model to Classify Audio Data into various Emotions like Sad, Happy, Angry, and Neutral

View Project Details

Time Series Analysis Project in R on Stock Market forecasting

In this time series project, you will build a model to predict the stock prices and identify the best time series forecasting model that gives reliable and authentic results for decision making.

View Project Details

Recommender System Machine Learning Project for Beginners-4

Collaborative Filtering Recommender System Project - Comparison of different model based and memory based methods to build recommendation system using collaborative filtering.

View Project Details

Loan Eligibility Prediction in Python using H2O.ai

In this loan prediction project you will build predictive models in Python using H2O.ai to predict if an applicant is able to repay the loan or not.

[View Project Details](#)

ML Model Deployment on AWS for Customer Churn Prediction

MLOps Project-Deploy Machine Learning Model to Production Python on AWS for Customer Churn Prediction

View Project Details

Time Series Forecasting with LSTM Neural Network Python

Deep Learning Project- Learn to apply deep learning paradigm to forecast univariate time series data.

View Project Details

MLOps on GCP Project for Autoregression using uWSGI Flask

This Project Explains the Process to create an end to end Machine learning development to design, Build and manage reproducible, testable, and evolvable ML models using GCP for AutoRegressor

View Project Details

Build Classification Algorithms for Digital Transformation[Banking]

Implement a machine learning approach using various classification techniques in Python to examine the digitalisation process of bank customers.

View Project Details

Deep Learning Project- Real-Time Fruit Detection using YOLOv4

In this deep learning project, you will learn to build an accurate, fast, and reliable real-time fruit detection system using the YOLOv4 object detection model for robotic harvesting platforms.

View Project Details

Build a Churn Prediction Model using Ensemble Learning

Learn how to build ensemble machine learning models like Random Forest, Adaboost, and Gradient Boosting for Customer Churn Prediction using Python

View Project Details

Trending Project Categories

- Machine Learning Projects
- Data Science Projects
- Deep Learning Projects
- Big Data Projects
- Apache Hadoop Projects
- Apache Spark Projects
- Show more

Trending Projects

- Walmart Sales Forecasting Data Science Project
- BigMart Sales Prediction ML Project
- Music Recommender System Project
- Credit Card Fraud Detection Using Machine Learning
- Resume Parser Python Project for Data Science
- Time Series Forecasting Projects
- Show more

Trending Blogs

- Machine Learning Projects for Beginners with Source Code
- Data Science Projects for Beginners with Source Code
- Big Data Projects for Beginners with Source Code
- IoT Projects for Beginners with Source Code
- Data Analyst vs Data Scientist
- Data Science Interview Questions and Answers
- Show more

Trending Recipes

- Search for a Value in Pandas DataFrame
- Pandas Create New Column based on Multiple Condition
- LSTM vs GRU
- Plot ROC Curve in Python
- Python Upload File to Google Drive
- Optimize Logistic Regression Hyper Parameters
- Show more

Trending Tutorials

- PCA in Machine Learning Tutorial
- PySpark Tutorial
- Hive Commands Tutorial
- MapReduce in Hadoop Tutorial
- Apache Hive Tutorial -Tables
- Linear Regression Tutorial
- Show more

ProjectPro

About us

Contact us

Privacy policy

User policy

Write for ProjectPro