

Tugas 3 Machine Learning

Hari Julian Putra

1. Try to execute the code from DataCamp.

4/30/25, 7:47 AM

DataLab | AI-powered data notebook for all skill levels

SVM_CANCER



Support Vector Machines with Scikit-learn

In this tutorial, you'll learn about Support Vector Machines, one of the most popular and widely used supervised machine learning algorithms.

```
# !pip install -r requirements.txt
```

```
# Import scikit-learn dataset library
from sklearn import datasets

# Load dataset
cancer = datasets.load_breast_cancer()
```

Exploring Data

After you have loaded the dataset, you might want to know a little bit more about it. You can check feature and target names.

```
# print the names of the 13 features
print("Features: ", cancer.feature_names)

# print the label type of cancer ('malignant' 'benign')
print("Labels: ", cancer.target_names)
```

Features: ['mean radius' 'mean texture' 'mean perimeter' 'mean area' 'mean smoothness' 'mean compactness' 'mean concavity' 'mean concave points' 'mean symmetry' 'mean fractal dimension' 'radius error' 'texture error' 'perimeter error' 'area error' 'smoothness error' 'compactness error' 'concavity error' 'concave points error' 'symmetry error' 'fractal dimension error' 'worst radius' 'worst texture' 'worst perimeter' 'worst area' 'worst smoothness' 'worst compactness' 'worst concavity' 'worst concave points' 'worst symmetry' 'worst fractal dimension']

Labels: ['malignant' 'benign']

Let's explore it for a bit more. You can also check the shape of the dataset using shape.

```
# print data(feature) shape
cancer.data.shape
```

(569, 30)

Let's check top 5 records of the feature set.

2. If succeed online, then try to run it locally, which might be a real challenge with Python and related packages installation.

```
#Import scikit-learn dataset library
from sklearn import datasets
#Load dataset
cancer = datasets.load_breast_cancer()
# print the names of the 13 features
print("Features: ", cancer.feature_names)
# print the label type of cancer('malignant' 'benign')
print("Labels: ", cancer.target_names)
# print data(feature)shape
cancer.data.shape
# print the cancer data features (top 5 records)
print(cancer.data[0:5])
# print the cancer labels (0:malignant, 1:benign)
print(cancer.target)
# Import train_test_split function
from sklearn.model_selection import train_test_split
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data,
cancer.target, test_size=0.3,random_state=109) # 70% training and
30% test
#Import svm model
from sklearn import svm
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel
#Train the model using the training sets
clf.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Accuracy: how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

- Find information how to create and use virtual environment, that might be helpful to scikit-learn.

3. Create and Use Virtual Environment

Creating virtual environment “venv”

```
PS C:\Users\ASUS> python -m venv venv
PS C:\Users\ASUS> venv\Scripts\activate
(venv) PS C:\Users\ASUS> pip install scikit-learn pandas matplotlib
Collecting scikit-learn
  Using cached scikit_learn-1.6.1-cp313-cp313-win_amd64.whl.metadata (15 kB)
Collecting pandas
  Using cached pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Collecting matplotlib
  Downloading matplotlib-3.10.1-cp313-cp313-win_amd64.whl.metadata (11 kB)
Collecting numpy>=1.19.5 (from scikit-learn)
  Downloading numpy-2.2.5-cp313-cp313-win_amd64.whl.metadata (60 kB)
Collecting scipy>=1.6.0 (from scikit-learn)
  Downloading scipy-1.15.2-cp313-cp313-win_amd64.whl.metadata (60 kB)
Collecting joblib>=1.2.0 (from scikit-learn)
  Using cached joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Collecting python-dateutil>=2.8.2 (from pandas)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.2-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
```

Create a requirements.txt file to list all packages (scikit-learn, pandas, matplotlib) for easy reinstallation later.

```
Downloading contourpy-1.3.2-cp313-cp313-win_amd64.whl (223 kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.57.0-cp313-cp313-win_amd64.whl (2.2 MB)
  2.2/2.2 MB 3.1 MB/s eta 0:00:00
Using cached joblib-1.4.2-py3-none-any.whl (301 kB)
Downloading kiwisolver-1.4.8-cp313-cp313-win_amd64.whl (71 kB)
Downloading numpy-2.2.5-cp313-cp313-win_amd64.whl (12.6 MB)
  12.6/12.6 MB 3.0 MB/s eta 0:00:00
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading pillow-11.2.1-cp313-cp313-win_amd64.whl (2.7 MB)
  2.7/2.7 MB 2.4 MB/s eta 0:00:00
Downloading pyparsing-3.2.3-py3-none-any.whl (111 kB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
Downloading scipy-1.15.2-cp313-cp313-win_amd64.whl (41.0 MB)
  41.0/41.0 MB 3.9 MB/s eta 0:00:00
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, tzdata, threadpoolctl, six, pyparsing, pillow, packaging, numpy, kiwisolver, joblib, fonttools, cy-
ler, scipy, python-dateutil, contourpy, scikit-learn, pandas, matplotlib
Successfully installed contourpy-1.3.2 cycler-0.12.1 fonttools-4.57.0 joblib-1.4.2 kiwisolver-1.4.8 matplotlib-3.10.1 numpy-2.2.5 packa
ing-25.0 pandas-2.2.3 pillow-11.2.1 pyparsing-3.2.3 python-dateutil-2.9.0.post0 pytz-2025.2 scikit-learn-1.6.1 scipy-1.15.2 six-1.17.0 t
hreadpoolctl-3.6.0 tzdata-2025.2
(venv) PS C:\Users\ASUS> pip freeze > requirements.txt
```