

A.Ü. ZİRAAT FAKÜLTESİ
TARIM MAKİNALARI VE TEKNOLOJİLERİ
MÜHENDİSLİĞİ BÖLÜMÜ
TARIMDA VERİ BİLİMİNE GİRİŞ

Doç. Dr. Necati Çetin

1



KAYNAKLAR

Python ile Makine Öğrenmesi
(Prof. Dr. Engin Sorhun)

Makine Öğrenmesi Teorik Yönleri ve Python Uygulamaları
(Dr. Öğr. Üyesi Sinan Uğuz)

Uygulamalarla Veri Bilimi
(Doç. Dr. Deniz Kılınç – Nezahat Başeğmez)

Python ile Makine Öğrenmesi
(Prof. Dr. Sadi Evren Şeker)

İnternet Kaynakları

necati.cetin@ankara.edu.tr
necaticetin1990@gmail.com

1) Veri Toplama (Data Collection)

Makine öğrenmesinin ana amacı, *ne olacağını yüksek doğrulukla tahmin* etmektir. Makine öğrenmesi algoritmaları ancak yeterli kalitede verilerle eğitildiklerinde iyi performans gösteren tahmin modelleri üretir. Bu nedenle bu süreç verilerin toplanması ile başlar.



MAKİNE ÖĞRENMESİNİN AŞAMALARI



Veri, makine öğrenmesiyle çalışan yapay zekânın gıdasıdır.

Modellere ne kadar çok veri sağlarsak o kadar yüksek performans elde ederiz. Çünkü, model incelenen durumu daha iyi performans ile tahmin edecektir.

Eğer veri hem **hacim** hem de **kalite** açısından yeterli değilse tahmin performansı iyi olsa bile model hakkında şüpheler oluşur.

Verilerin toplanmasında karşılaşılan en yaygın sorunlar:

Hatalı veriler: Toplanan veriler incelenen olay ya da problem ile ilgisiz olabilir.

Veri yapısı: Toplanan veriler her zaman yapılandırılmış veriler (sistemik olarak satırlar ve sütunlar halinde sıralanmış) olmayabilir.

Eksik veriler: Değişkenlerin içerdiği değerlerde eksikler olabilir.



**DATA
COLLECTION
CHALLENGES**

Veri dengesizliği: Verilerdeki bazı sınıflar veya kategoriler orantısız şekilde yüksek veya düşük sayıda örneğe sahip olabilir. Sonuç olarak, model yetersiz temsil edilme riski taşır.

Yanlı veri: Model, verilerin ve etiketlerin nasıl seçildiğine bağlı olarak, cinsiyet, yaş veya bölge gibi önyargılar ihtiva edebilir. Bu tür durumların ilk bakışta tespit edilmesi ve kaldırılması zordur.

Data Preparation



2) Veri Hazırlama (Data Preparation)

Toplanan ham veriler genellikle eksik gözlemlere sahip, farklı formatlarda kaydedilmiş veya istatistiki olarak belirli davranış veya eğilimlerden yoksun olabilir. Ayrıca, birçok hata içermesi muhtemeldir. Bu nedenle modelleme işlemi için verilerin düzenlenip belli bir formatta sunulması gerekir.

Veri hazırlama aşamaları

- ✓ Keşfi veri analizi (Exploratory data analysis)
- ✓ Veri önişleme (Data preprocessing)
- ✓ Veri seti bölümlenmesi (Data split)

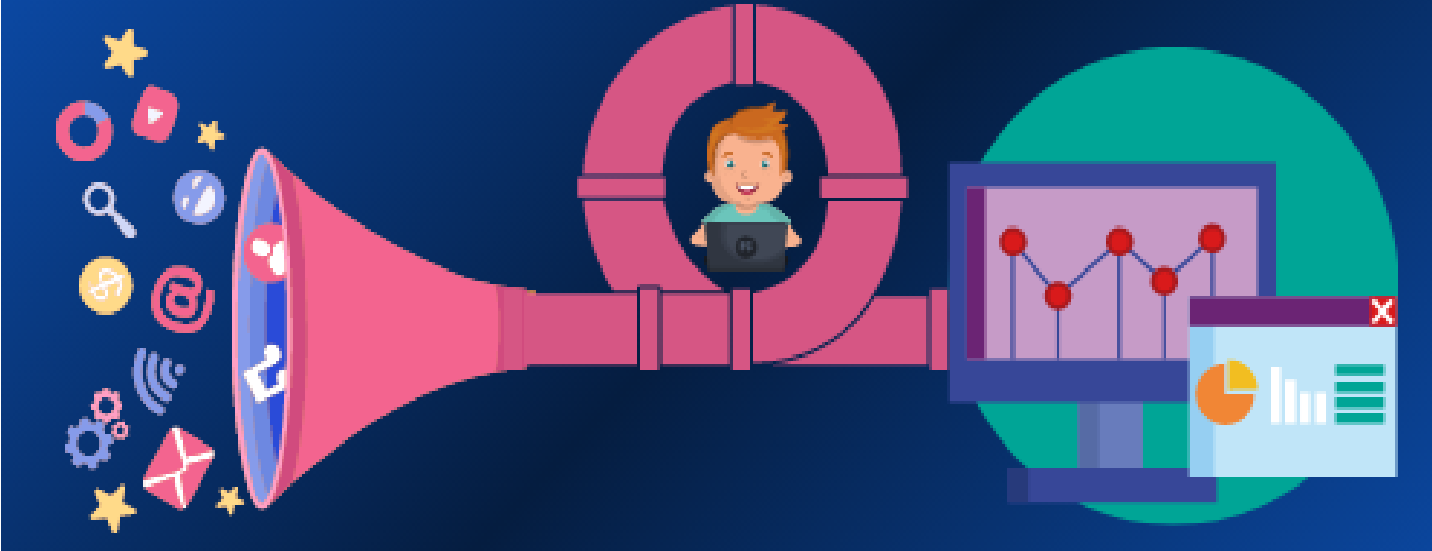
Exploratory Data Analysis

Keşfi veri analizi, veri setlerinin temel özelliklerini, genellikle *görsel* ve *istatistiki* yöntemlerle özetlemek için analiz eder. Hipotezi test edilmeden önce veri setindeki **değişkenlerin dağılımları**, **aykırı değerler** ve **anormallikler** tespit edilir. Bu sayede, verilerin model oluşturmaya uygun olup olmadığını ya da ne tür modeller için uygun olduğunu anlamamızı sağlar.

Keşfi veri analizinde,

- 1) Veri setinin tanımlayıcı istatistiği (descriptive statistics) kullanılarak ortalama, standart hata, minimum ve maksimum değerler elde edilir,
- 2) Histogram grafikleri kullanılarak normal dağılıma uygunluğuna bakılır,
- 3) Serpilme diyagramı, zaman serisi grafiği vb. grafiklerle değişkenler arasındaki ilişki görselleştirilir,
- 4) Aykırı değerler görselleştirilerek tespit edilir,
- 5) Hedef ve öznitelik değişkenleri belirlenir,
- 6) Hedef değişkenin her bir öznitelik değişkeniyle ilişkisi (doğrusal ya da doğrusal olmayan) tespit edilir.

DATA PREPROCESSING



Veri ön işleme, makine öğrenmesinde en fazla zaman alan kısımdır.

Veri bilimciler, zamanlarının %60'ını verileri temizlemek ve düzenlemek için harcarlar.

Verilerin iyi modellenmesi bu adımın sağlıklı bir şekilde yürütülmesine bağlıdır.

Veri önışleme birkaç teknik adıma dayanır:

Veri temizleme: Yanlış eklenen veya sınıflandırılan veriler, manuel (tek tek filtrelenerek) veya otomatik (algoritma kullanılarak) olarak kaldırılır.

Veri yapılandırma: Yapılandırılmamış veriler yapılandırılmış verilere dönüştürülür. Örneğın, metın fotoğraf, ses ya da video kaydı gibi yapılandırılmamış veriler, algoritmalar yardımıyla kantitatif verilere dönüştürüldükten sonra satır ve sütun düzenine getirilir.

Eksik veri tamamlama: Sınıf ve kategorilerde orantısız verileri dengelemek ve eksik verileri doldurmak için geliştirilmiş yöntemler kullanılır ve veri setindeki bilgi kaybı en aza indirilir.

Aşırı örnekleme (Oversampling): Veri kümesindeki yanlılık (mesela erkek çalışanların sayısının kadın sayısından çok fazla olması) veya dengesizlik, daha fazla gözlem/örneklem oluşturularak düzeltilebilir.

Veri entegrasyonu: Daha geniş bir veri seti elde etmek için birden fazla veri setini birleştirmek, tek bir veri kümesindeki eksikliğin üstesinden gelebilir.

Veri normalleştirme: Veri setindeki öznitelik değişkenleri ve hedef değişken farklı ölçeklerde olduğunda (mesela cm cinsinden yükseklik ve m^3 cinsinden hacim vb.) çoğu makine öğrenmesi algoritması iyi performans göstermez.

Bu nedenle, verileri **normalleştirmek** (değişkenleri 0 ve 1 veya -1 ve 1 aralığında olmaları için değerlerini yeniden ölçeklendirmek) veya **standartlaştırmak** (verileri ortalama 0 ve standart sapması bir olacak şekilde yeniden ölçeklendirir) bu sorunu çözmeye yardımcı olabilir.

Min-Max Normalization

Min-Max Normalization transforms x to x' by converting each value of features to a range between 0 and 1, and this is also known as (0-1) Normalization. If the data has negative values the range would have been between -1 and 1.

The formula for Min-Max Normalization is:

Normalized Value $x' = \frac{\text{Original Value } x - \text{Minimum Value of } x}{\text{Maximum Value of } x - \text{Minimum Value of } x}$

Formula for Min-Max Normalization

Standardization

Standardized value $z = \frac{\text{Original value } x - \text{Mean } \mu}{\text{Standard deviation } \sigma}$

Öznitelik seçimi (Feature selection): Öznitelik sayısı ne kadar fazlaysa veri seti de o kadar büyük olmaktadır. Ancak yüksek boyutlu bir veri setinde alakasız ve önemsiz bazı öznitelikler olabilir. Bu tür özniteliklerin modele katkısı diğer özniteliklere göre çok düşük ya da sıfırdır. Bu öznitelikler, verimli tahmine dayalı modelleme sürecini engelleyen bir dizi soruna sebep olabilir.

Öznitelik seçimi bir veri setinde en önemli öznitelikleri seçme işlemidir. Bu yöntem, modelleme süresini kısaltırken tahmin performansını da çoğu zaman artırmaktadır.

All Features



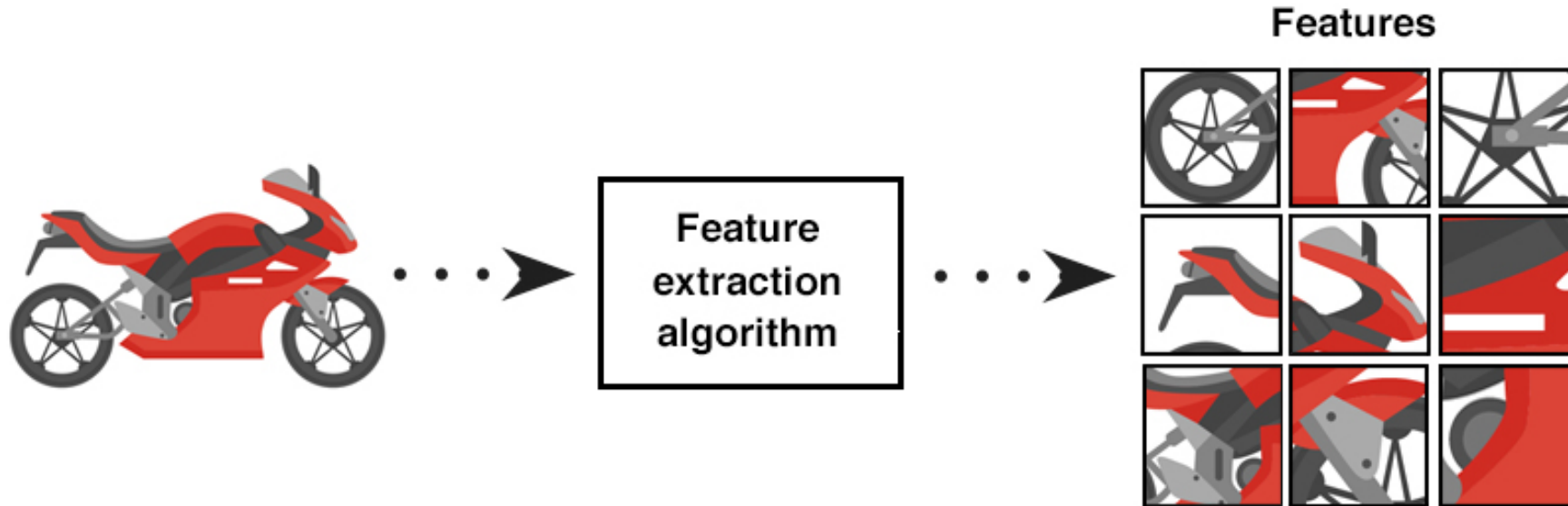
Feature Selection



Final Features



Öznitelik çıkarımı (Feature extraction): Öznitelik seçimi veri setindeki alakasız veya gereksiz öznitelikleri filtreleyerek azaltmak ya da ayıklamak için kullanılırken, öznitelik çıkarımı çok sayıda özneliğin içerdiği değerleri dönüştürerek az sayıda özneliğe indirgemek için kullanılır.

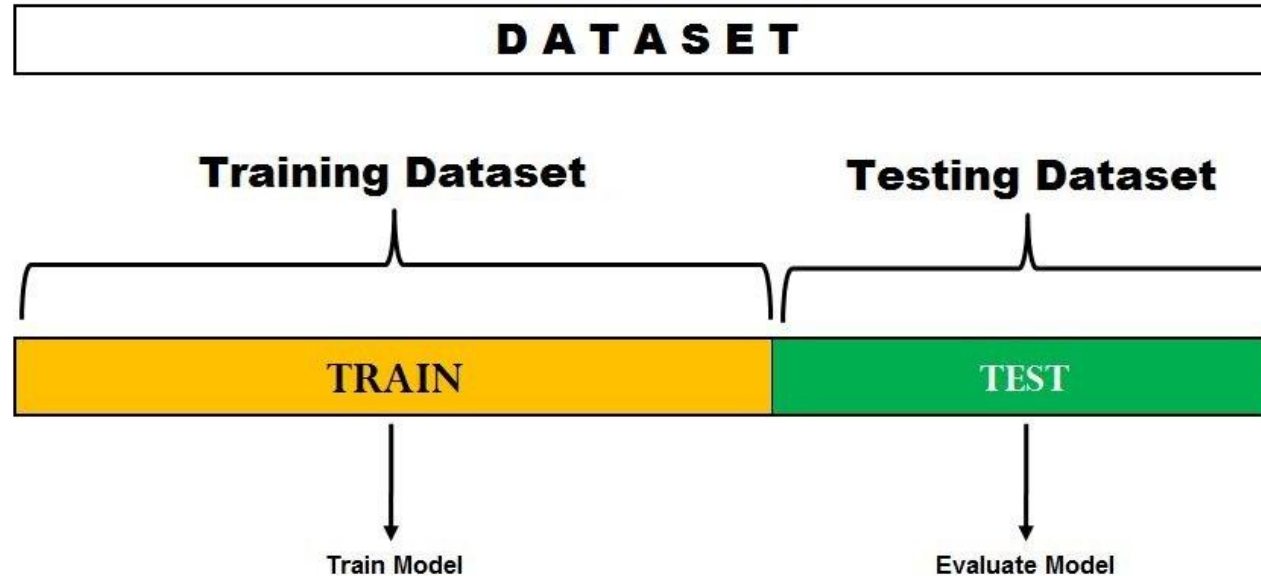




Öznitelik çıkarımında, özniteliklerin değerleri matematiksel olarak daha az sayıda yeni özneliğe dönüştürülmektedir.

Ancak, bu az sayıdaki yeni öznelik, önceki çok sayıdaki özneliğin içerdiği bilgiyi içermektedir. Öznitelik çıkarımında, dönüştürme sırasında az miktarda da olsa bilgi kaybı söz konusu olabilir.

Veri bölümlenme ya da **eğitim-test bölümlenmesi** adımında veri seti eğitim ve test seti olarak iki ayrı gruba bölünür. Eğitim verileri, algoritmanın verileri üzerinden eğitilmesi ve tahmin modeli oluşturması için kullanılır. Üretilen modelin performansını ve yeni durumları öngörüp öngöremediği test veri seti kullanılarak sınılanır.



Eğitim-test seti bölümlenmesi genellikle rassal (rastgele) olarak yapılır. Örneğin 10 gözlemlik bir veri setinde 1, 4, 6, 7, 8, 10. gözlemler eğitim seti olarak rassal olarak seçilirken arta kalan 2, 3, 5, 9. gözlemler ise test seti olarak atanır.



Initial Dataset



Data clean
Data cu
Remove



Pre-
Data

Data | split



80%

SVM
Learning
algorithms

Output
variable

X ₅	Y
2.9	A
1.3.1	A
2.6.1	B

HOW TO BUILD A MACHINE LEARNING MODEL

MODEL OLUŞTURMA

Algoritmanın verilerle eğitilerek model oluşturmaktan kasıt,

Denetimli öğrenmede özniteliklerle hedef değişken arasındaki ilişkiyi,

Denetimsiz öğrenmede öznitelikler arasındaki benzerlik ya da mesafeyi (farklılıkları),

Pekiştirmeli öğrenmede ise özniteliklerin dayandığı sebep-sonuç ilişkisinin kurallarını olabildiğince gerçeğe yakın olarak tanımlayan fonksiyonları tahmin etmektir.

Makine öğrenmesinde model oluşturma aşaması:

Algoritmanın Eğitilmesi (training)

Modelin Test Edilmesi (testing) adımlarını içermektedir.

ALGORİTMANIN EĞİTİLMESİ

Bir kısmı iyileşmiş, bir kısmı halen tedavi gören **100 tane X hastasına ait veri** olsun. Veri setinde bu kişilere ait kan değerleri vb. hastalığı teşhise dair **20 kadar öznitelik değişkeni** bulunsun. Bilgisayarın anlaması için etiketleme işlemini **0 (X hastası değil)** ve **1 (X hastası)** şeklinde rakamlarla yapacağız. Buradaki veri problemi tipik bir **sınıflandırma** problemidir: Hastalığı teşhiste kullanılan 20 özneliğin değerine göre bir gözlemin (hastanın) **X hastası olup olmadığını tahmin eden** bir model üretmek.

ALGORİTMANIN EĞİTİLMESİ

Veri seti içinden **rastgele seçilen 80 tane gözlemi** algoritmayı eğitmek için ayıralım. Algoritma her bir gözlemin öznitelik değerleri ile etiketini inceleyip öğrenecek ve **özniteliklerle sınıflandırma etiketleri arasındaki ilişkiyi formüleştirecek**. Bunun anlamı her bir öznitelik için bir katsayı hesaplamasıdır.

Her bir öznitelik hastalığın teşhisinde aynı derecede öneme sahip olmadığından algoritma da teşhis göstergeleri için **farklı katsayılar hesaplayacaktır**. Her bir özneliği hastayı teşhiste bir parametre olarak kabul edersek bunlar için hesaplanmış katsayılarda parametre katsayıları olacaktır.

ALGORİTMANIN EĞİTİLMESİ

Model üretmekten kasıt, parametre katsayılarını hesaplamaktır.

Yani algoritma **20 teşhis göstergesi için 20 tane katsayı** hesaplayacaktır.

Hesaplanan parametre değerleri öyle hesaplanmalı ki her bir gözlemin (hastanın) 20 teşhis değerlerini **bu katsayılarla çarptığımızda** bu kişinin X hastası olup olmadığını tahmin edebilsin.

ALGORİTMANIN EĞİTİLMESİ

Burada şunu vurgulamakta fayda var:

Algoritma her bir gözlem için ayrı parametre değeri hesaplamıyor.

Gözlemlerin tamamı için **genelleştirilebilir 20 katsayı** hesaplıyor.

Diğer bir ifadeyle, algoritma hastayı teşhis için değil **hastalığı teşhis için model (parametre katsayıları) üretiyor.**

ALGORİTMANIN EĞİTİLMESİ

Algoritmik model ya da hastalığı teşhis eden, **eldeki veriler (öznitelikler)** doğrultusunda **gözlemlerin sınıfını (kişilerin X hastası olup olmadıklarını)** tahmin eden modelin başarısı ilk etapta **hastaların tahmin edilen sınıfları** ile **gerçek sınıflarının** karşılaştırılması sonucu anlaşılacaktır.

Eğitim setindeki 80 kişinin 80'inin de sınıf etiketini tahmin etmiş mi?

Model; X hastası olanların tamamını **1** olarak,

X hastası olmayanların tamamını da **0** olarak sınıflandırmış mı?

Yani, model özniteliklerden yola çıkarak hedef değişkeni yeniden üretebilmiş mi?

ALGORİTMANIN TEST EDİLMESİ

En yalın haliyle ifade edecek olursak bu adımda **eğitim seti üzerinden oluşturulan modelin test setini yeniden üretip üretmediği, tahmin edip etmediği sorusuna cevap aranır.**

Veri setinin bir kısmının test seti olarak ayrılmasının amacı eğitim seti üzerinden oluşturulan modelin **'yeni durumları' tahmin edip edemeyeceğinin anlaşılabilmesi** içindir.

ALGORİTMANIN TEST EDİLMESİ

Algoritma, eğitim setinde **hem hedef değişkeni** hem de **öznitelik değişkenleri birlikte verilerek** eğitilir.

Algoritma öğrendiklerini, bu ikisi arasında keşfettiği ilişkiyi, hesapladığı bir fonksiyonla ifade eder.

Bu fonksiyon ya da model ile eğitim setindeki hedef değişkenini tahmin eder ve sonuçları **hedef değişkenin gerçek verileriyle karşılaştırırsak** modelin tahmin performansıyla ilgili fikir verir.

ALGORİTMANIN TEST EDİLMESİ

Ancak, modelin özneliliklerle, zaten bildiği/tanıdığı hedef etiketlerini yeniden üretebilmesi ya da tahmin etmesi maharet değildir.

Makine öğrenmesinde amaç **yeni durumları öngörebilen, olabilecekleri tahmin edebilen ve karar alıcıların önündeki belirsizlik bulutlarını dağıtabilen** model üretmektedir.

ALGORİTMANIN TEST EDİLMESİ

Test sürecinde **modele sadece test setindeki öznitelikler verilir** hedef değişkeni verilmez.

Test setindeki öznitelik değişkenleri verirken, eğitim sürecinde öğrendikleriyle oluşturduğu modeli kullanarak test setindeki hedef değişkenini tahmin etmesi, yeniden üretmesi istenir.

Diğer bir ifadeyle, **modelin tahmin parametreleri ile test setindeki özniteliklerin değerleri çarpılarak** gözlemlerin sınıf etiketleri tahmin edilir.

ALGORİTMANIN TEST EDİLMESİ

Burada maharet ise, modelin **test setindeki ipuçlarını kullanarak** (özniteliklerden hareketle) **bilmediği, tanımadığı hedef etiketlerini tanımlayabilmesidir.**

Eğer önceki adımda oluşturulan model test sürecini de başarıyla atlatabiliyorsa artık yeni durumları tahmin etmek için kullanılabilir.

PERFORMANS ÖLÇME KRİTERLERİ

Model Değerlendirmenin amacı, **modelin tahmin hatası** ve **başarısının ölçülmesidir.**

Bir modelin performansı **hata** ya da **kayıp fonksiyonu** üzerinden değerlendirilir.

Bunun için, makine öğrenmesi algoritmasının türüne bağlı olarak **çeşitli metrikler ya da performans ölçütleri** geliştirilmiştir.

PERFORMANS ÖLÇME KRİTERLERİ

Regresyon	Sınıflandırma	Kümeleme
MSE	Doğruluk	ARI
RMSE	Kesinlik	MIS
MAE	Duyarlılık	V-measure
R	F1	Elbow
R^2	ROC-AUC	
Düzeltilmiş R^2	Kesinlik-Duyarlık Grafiği	

ÇAPRAZ-DOĞRULAMA

Bazı durumlarda model eğitildiği veri setine duyarlı olabilir.

Örneğin rassal olarak seçilmiş gözlemlerden bir eğitim seti oluşturduğumuzu ve bu eğitim seti üzerinden bir algoritma eğiterek model oluşturduğumuzu düşünelim.

Modelin performansını ölçmek için kullandığımız **metrik yüksek tahmin performansına işaret ediyor olsun**. Acaba gözlemleri **karıştırıp** rassal olarak yeniden bir eğitim seti oluşturarak algoritmayı eğitirsek bu modelin performansı ölçtüğümüz aynı yüksek seviyeyi verir mi?

ÇAPRAZ-DOĞRULAMA

Her defasında **farklı rassal seçimlerle** eğitim seti oluşturup modelin performansını ölçtüğümüzde, modelin performans değerleri istikrarlı değilse ve dalgalanıyorsa **model sonucu veri setine aşırı duyarlı demektir.**

Böyle bir modeli, veri setine genelleştirmek, yeni durumları tahmin etmek için kullanmak **riskli** olacaktır.

ÇAPRAZ-DOĞRULAMA

Modelin eğitim aşamasında karşılaşılan sorunlardan birisi de **aşırı öğrenme (over-fitting)** ya da **abartılı taklit** sorunudur.

Bu durum modelin veriye duyarlı olmasıyla alakalı olabilir.

Algoritma eğitim aşamasında kendisine sunulan veriyi ezberlememiş, adeta yalayıp yutmuştur.

Ancak, bunun dışında karşılaştığı her yeni durumda ezberi bozulmaktadır.

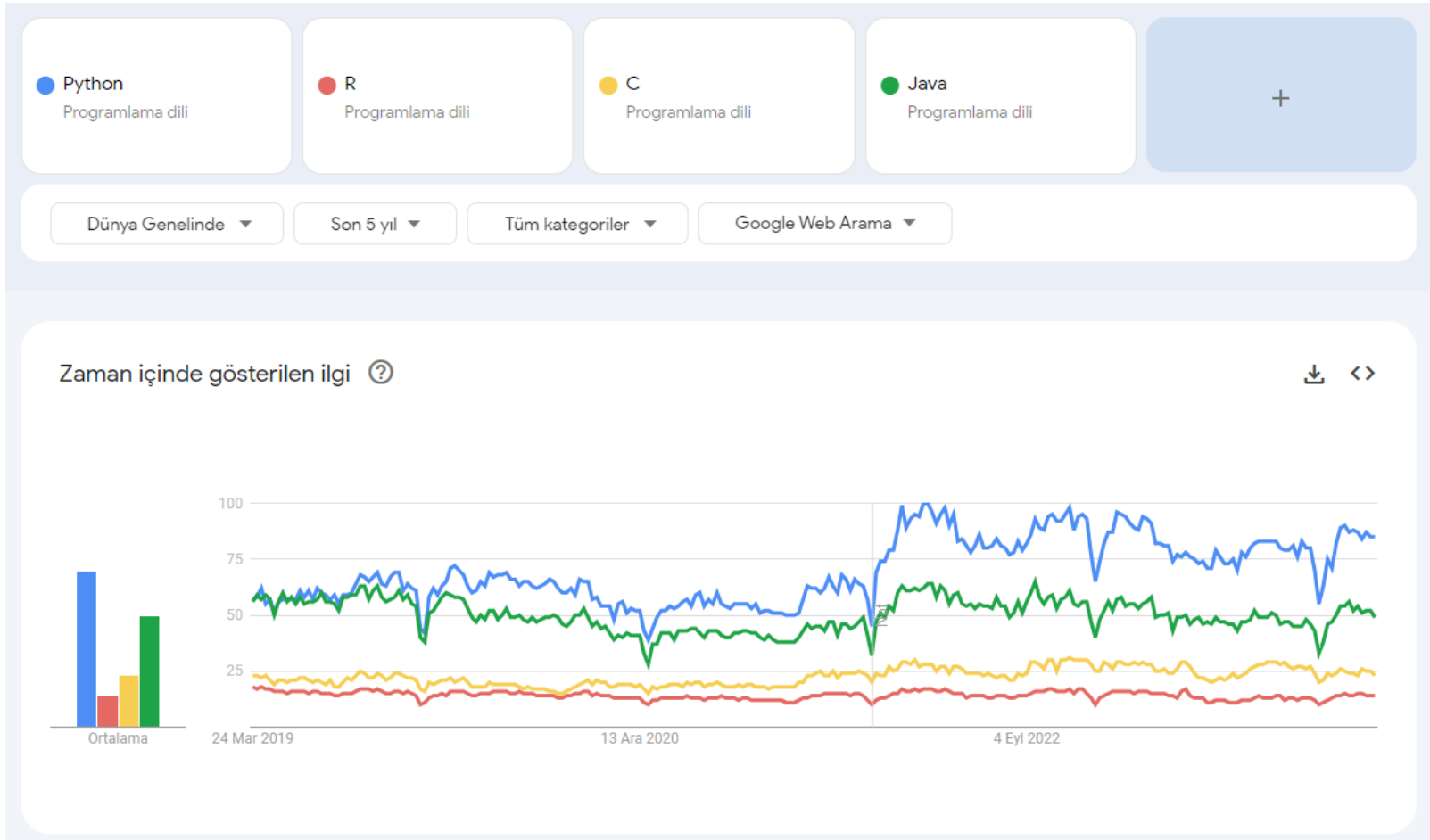
ÇAPRAZ-DOĞRULAMA

Başka bir ifadeyle abartılı taklit, eğitim setini öğrenip uygulama, tahmin, öngörü alanı dışına çıkmayan, öğrendiğini genelleştiremeyen, her duruma uyarlayamayan bir modelin durumudur.

Bunun sonucu olarak, model eğitim setindeki hedef değişkeni başarıyla tahmin ederken test setindeki hedef değişkeni aynı derecede tahmin edememektedir.

VERİ BİLİMİ VE PYTHON

Neden Python?



● Makine öğrenimi
Konu

● Veri Bilimi
Çalışma alanı

● Veri Analizi
Konu

● Derin öğrenme
Çalışma alanı

+

Dünya Genelinde ▾

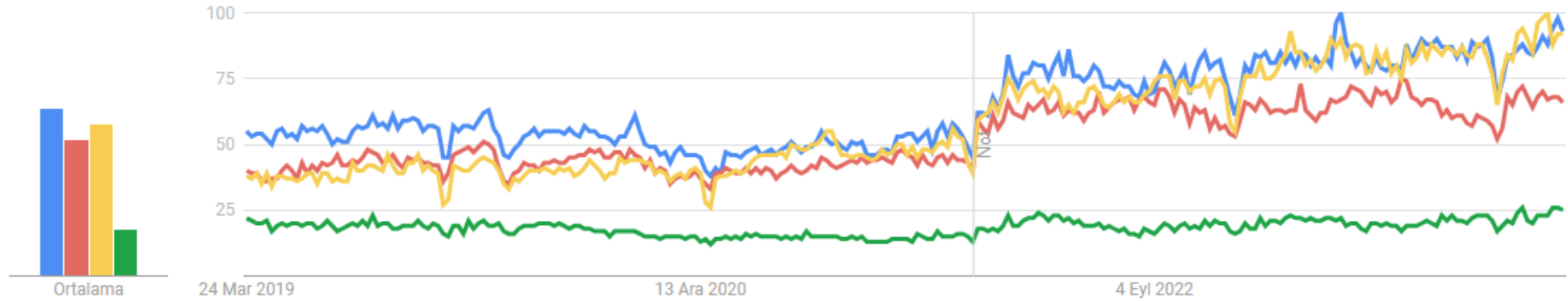
Son 5 yıl ▾

Tüm kategoriler ▾

Google Web Arama ▾

Zaman içinde gösterilen ilgi ?

↓ <>





Anaconda Distribution

Free Download

Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms



Get Additional Installers





Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Upgrade Now

Connect

Home

Environments

Learning

Community

Anaconda Toolbox
Supercharged local notebooks. Click the Toolbox tile to install.

[Read the Docs](#)

Documentation

Anaconda Blog



All applications

on

base (root)

Channels



DataSpell

DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

Install



CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch



JupyterLab

3.4.4

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Notebook

6.4.12

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated



Qt Console

5.2.2

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.



Spyder

5.2.2

Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing,



VS Code

1.75.1

Streamlined code editor with support for development operations like debugging, task running and version control.



Home

Environments

Learning

Community

NEW

Anaconda Toolbox
Supercharged local notebooks. Click the Toolbox tile to Install.

[Read the Docs](#)

[Documentation](#)

[Anaconda Blog](#)



All applications

on **base (root)**

Channels



anaconda-toolbox

0.4.0

Install



console_shortcut_miniconda

0.1.1

Install



Glueviz

1.2.4

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



Orange 3

3.34.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



powershell_shortcut_miniconda

0.0.1



PyCharm Professional

A Full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.



RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.



SPYDER TÜMLEŞİK GELİŞTİRME ORTAMI (IDE: INTEGRATED DEVELOPMENT ENVIRONMENT)

Scientific **PY**thon **D**evelopment **EnviR**onment

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\gawe_

C:\Users\gawe_\SKLEARN\dogrusalregresyon.py

dogrusalregresyon.py x temp.py x

```
1  -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7
8  #1 kütüphaneler bölümü
9  import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 #2 veri önışleme
14 #2.1 veri yükleme
15 veriler = pd.read_csv('satislar.csv')
16 #test
17 print(veriler)
18
19 #veri önışleme
20
21 aylar = veriler[['Aylar']]
22 print(aylar)
23
24 satislar = veriler [['Satislar']]
25 print(satislar)
26
27 satislar2 = veriler.iloc[:,0:1].values
28 print(satislar2)
29
30 #verilerin eğitim ve test için bölünmesi
31 from sklearn.model_selection import train_test_split
32
33 x_train, x_test, y_train, y_test = train_test_split(aylar, satislar, test_size=0.33, random_state=0)
34
35 #verilere standarization işleminin yapılması
36 from sklearn.preprocessing import StandardScaler
```

Çalışma ortamı
(Script)

Nar Type Size Value

Dosya ve
değişken
yöneticisi

Help Variable Explorer Plots Files

Console 1/A x

Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916
64 bit (AMD64)]
Type "copyright", "credits" or "license" for more
informati

IPython 7.31.0 -- An enhanced Interactive Python.

In [1]:

Konsol

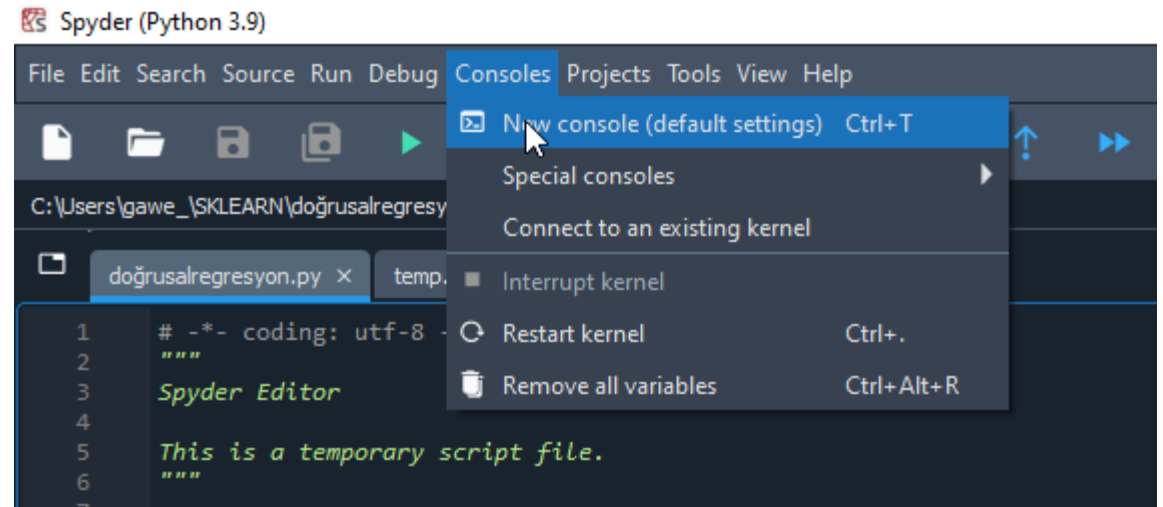
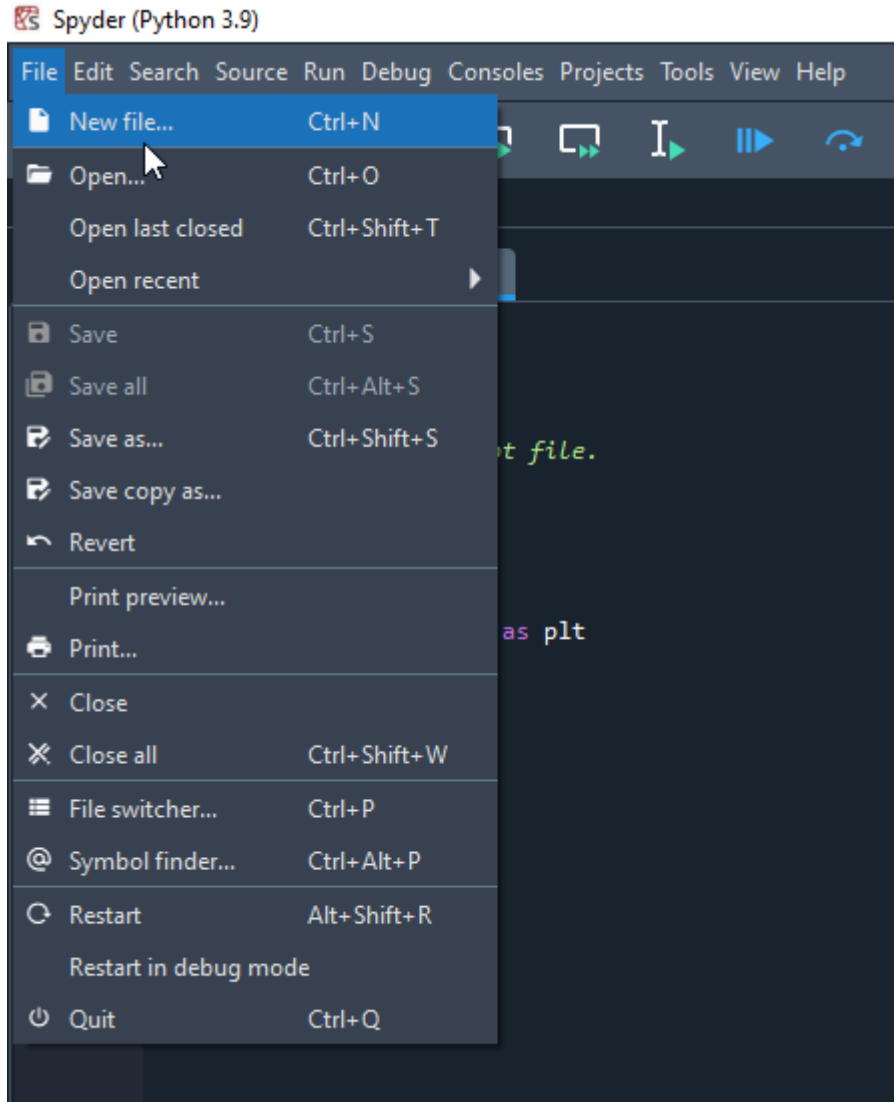
IPython Console History

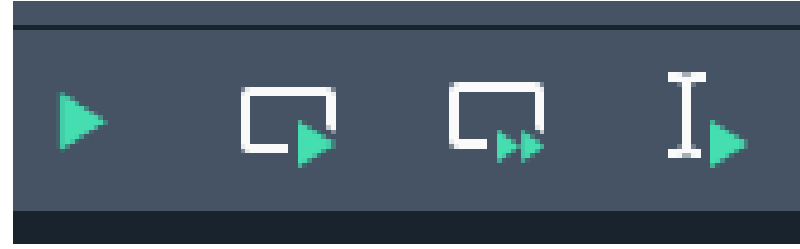
LSP Python: ready conda: base (Python 3.9.13) Line 1, Col 1 UTF-8 CRLF RW Mem 92%



Spyder ekranında üç ana pencere bulunmaktadır. Bunlar Çalışma Dosyası (Script), Dosya ve Değişken Yöneticisi ve Konsol'dur.

```
doğrusalregresyon.py × temp.py ×
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 #1 kütüphaneler bölümü
9 import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 #2 veri önışleme
14 #2.1 veri yükleme
15 veriler = pd.read_csv('satislar.csv')
16 #test
17 print(veriler)
18
19 #veri önışleme
20
21 aylar = veriler[['Aylar']]
22 print(aylar)
23
24 satislar = veriler [['Satislar']]
25 print(satislar)
26
27 satislar2 = veriler.iloc[:,0:1].values
28 print(satislar2)
29
30 #verilerin eğitim ve test için bölünmesi
31 from sklearn.model_selection import train_test_split
32
33 x_train, x_test, y_train, y_test = train_test_split(aylar, satislar, test_size=0.33, random_state=0)
34
35 #verilere standarization işleminin yapılması
36 from sklearn.preprocessing import StandardScaler
```





Spyder (Python 3.9)

File Edit Search Source Run Debug Console Help

C:\Users\gawe_\SKLEARN\dogrusalregresyon.py

doğrusalregresyon.py x temp.py x

```
1  -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7
8  #1 kütüphaneler bölümü
9  import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 #2 veri ön işleme
14 #2.1 veri yükleme
15 veriler = pd.read_csv('satislar.csv')
16 #test
17 print(veriler)
18
19 #veri ön işleme
20
21 aylar = veriler[['Aylar']]
22 print(aylar)
23
24 satislar = veriler [['Satislar']]
25 print(satislar)
26
27 satislar2 = veriler.iloc[:,0:1].values
28 print(satislar2)
29
30 #verilerin eğitim ve test için bölünmesi
31 from sklearn.model_selection import train_test_split
32
33 x_train, x_test, y_train, y_test = train_test_split(aylar, satislar, test_size=0.33, random_state=0)
34
35 #verilere standarization işleminin yapılması
36 from sklearn.preprocessing import StandardScaler
```

Name	Type	Size	Value
------	------	------	-------

Help Variable Explorer Plots Files

Console 1/A x

Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]:

IPython Console History

LSP Python: ready conda: base (Python 3.9.13) Line 1, Col 1 UTF-8 CRLF RW Mem 92%

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\gawe_\spyder-py3\autosave

C:\Users\gawe_\spyder-py3\autosave\untitled0.py

doğrusalregresyon.py × temp.py × untitled0.py* ×

```
1 # -*- coding: utf-8 -*-  
2 """  
3 Created on Tue Mar 26 20:44:59 2024  
4  
5 @author: gawe_  
6 """  
7  
8 degisken = [1, 5, 6, 7]  
9
```

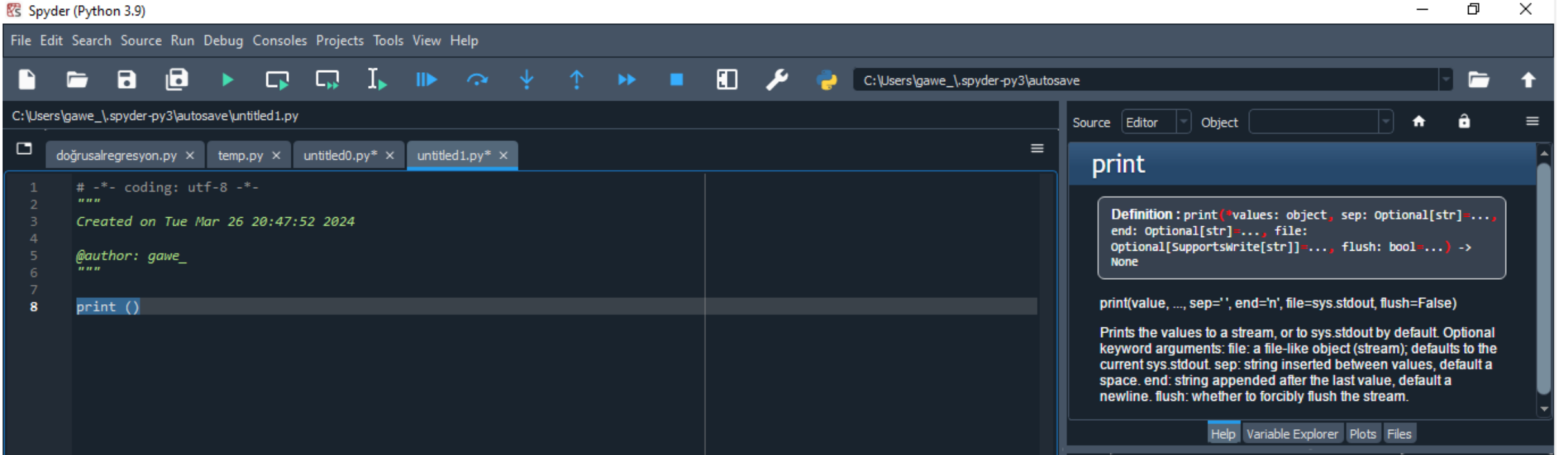
Name	Type	Size	Value
degisken	list	4	[1, 5, 6, 7]

Help Variable Explorer Plots Files

The screenshot displays a Python IDE interface. At the top right, a variable explorer shows a variable named 'degisken' of type 'list' with a size of 4, containing the elements [1, 5, 6, 7]. Below this, a window titled 'degisken - List (4 elements)' is open, showing a detailed view of the list with the following data:

Ind	Type	Size	Value
0	int	1	1
1	int	1	5
2	int	1	6
3	int	1	7

At the bottom right, a console window shows the output of a print statement: [1, 5, 6, 7]. The console also displays an error message: 'NameError: name 'degisken' is not defined'. The IDE interface includes a 'Save and Close' button and a 'Close' button at the bottom.



The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script with the following content:

```
1 # -*- coding: utf-8 -*-  
2 """  
3 Created on Tue Mar 26 20:47:52 2024  
4  
5 @author: gawe_  
6 """  
7  
8 print ()
```

The right-hand pane shows the help documentation for the `print` function. The title is `print`. The definition is:

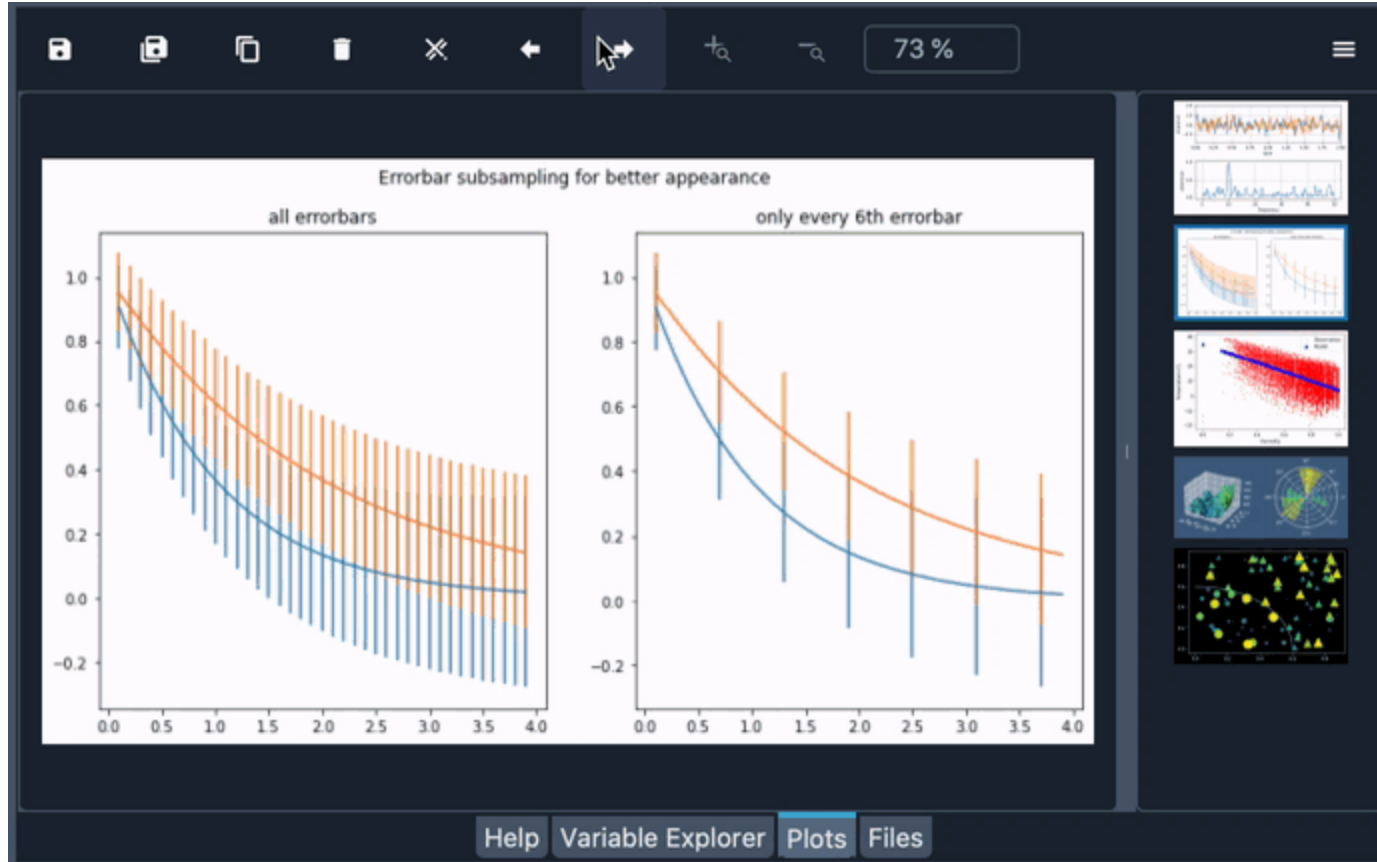
```
Definition: print(*values: object, sep: Optional[str]=...,  
end: Optional[str]=..., file:  
Optional[SupportsWrite[str]]=..., flush: bool=...) ->  
None
```

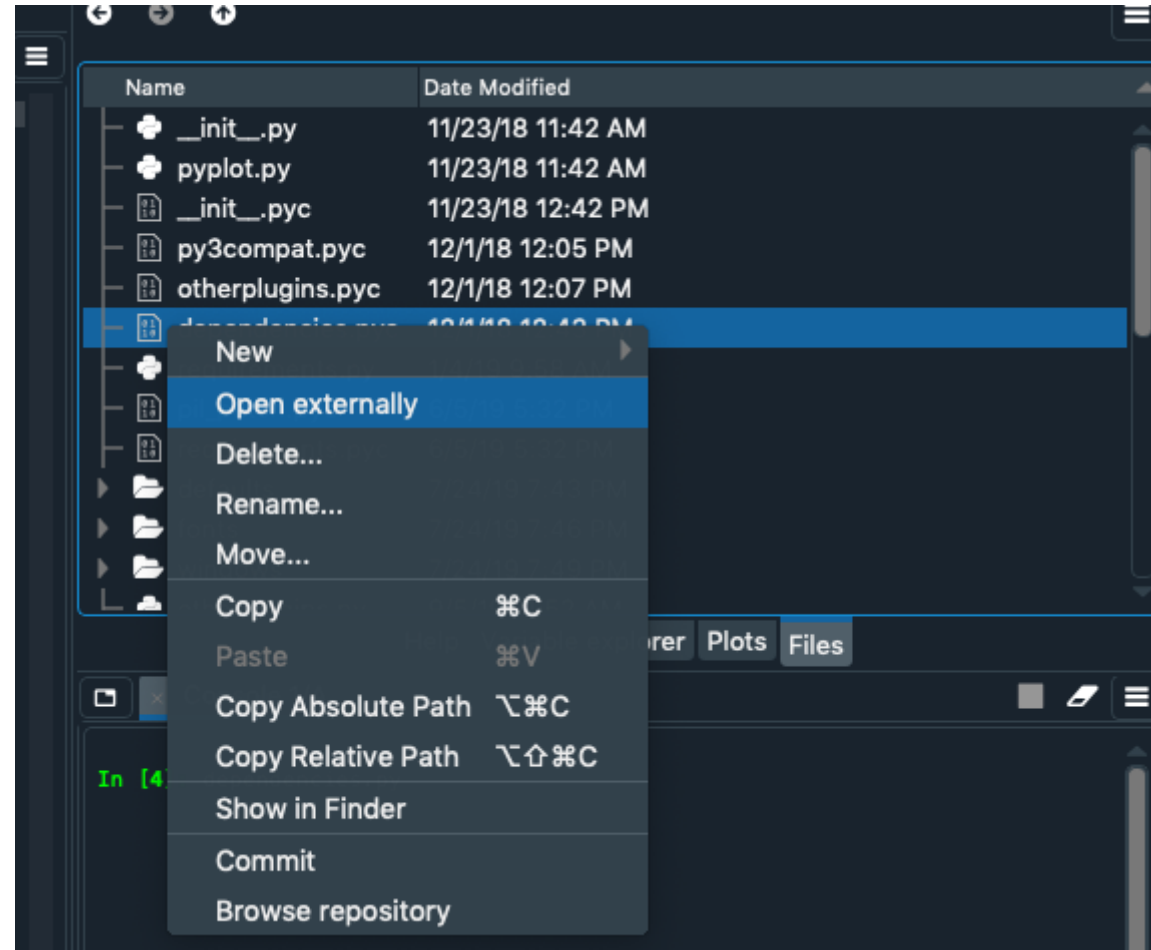
The function signature is:

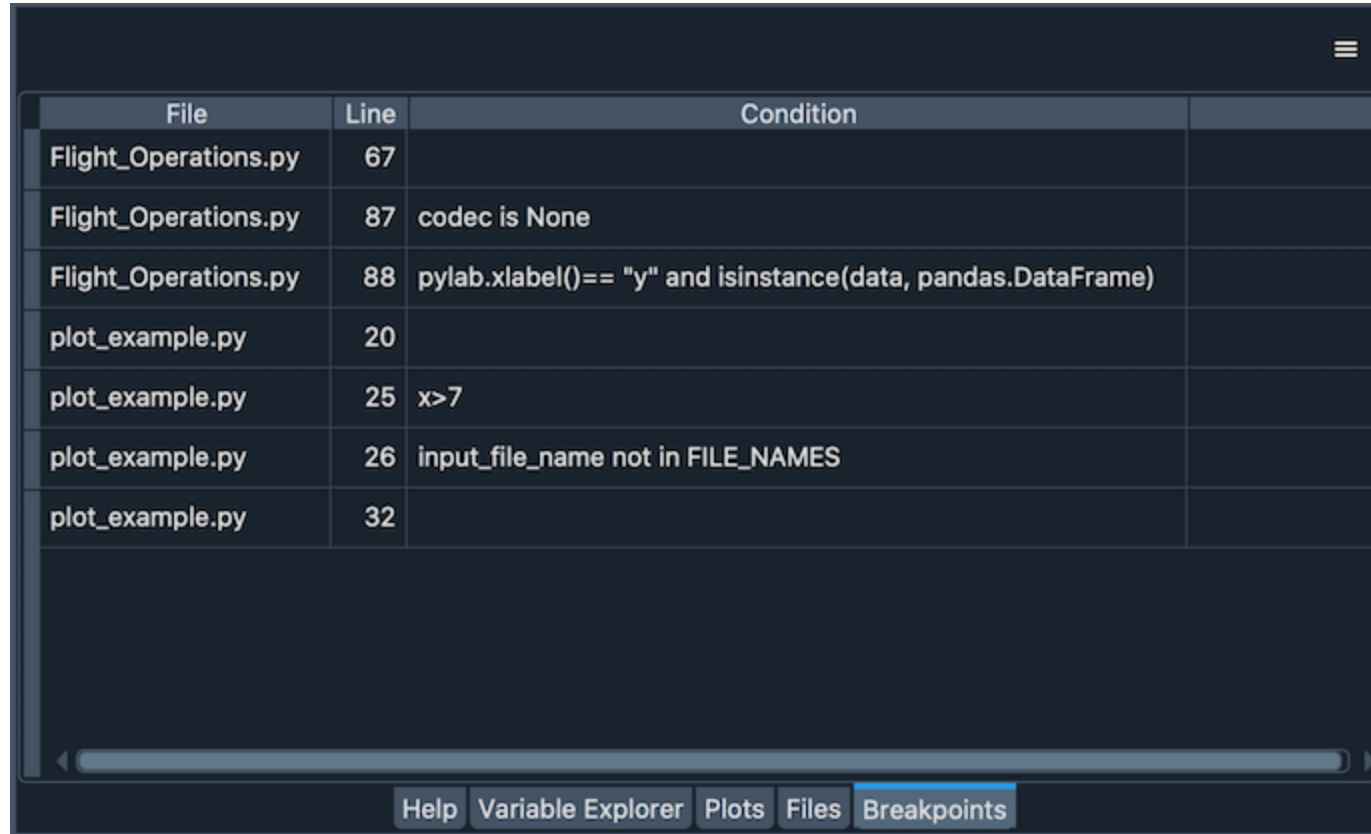
```
print(value, ..., sep=' ', end='n', file=sys.stdout, flush=False)
```

The description states: "Prints the values to a stream, or to sys.stdout by default. Optional keyword arguments: file: a file-like object (stream); defaults to the current sys.stdout. sep: string inserted between values, default a space. end: string appended after the last value, default a newline. flush: whether to forcibly flush the stream."

Ctrl+I = Yardım => Kod hakkında bilgi verir







File	Line	Condition
Flight_Operations.py	67	
Flight_Operations.py	87	codec is None
Flight_Operations.py	88	pylab.xlabel() == "y" and isinstance(data, pandas.DataFrame)
plot_example.py	20	
plot_example.py	25	x>7
plot_example.py	26	input_file_name not in FILE_NAMES
plot_example.py	32	

Help Variable Explorer Plots Files Breakpoints

```
doğrusalregresyon.py × temp.py × untitled0.py* × untitled1.py* ×
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 #1 kütüphaneler bölümü
9 import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12
```

```
C:\Users\gawe_\spyder-py3\autosave\untitled1.py
doğrusalregresyon.py* × temp.py × untitled0.py* × untitled1.py* ×
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 26 20:47:52 2024
4
5 @author: gawe_
6 """
7
8 import sklearn.metrics as mts
```

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\gawe_\spyder-py3\autosave\untitled1.py
doğrusalregresyon.py* x temp.py x untitled0.py* x untitled1.py* x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 26 20:47:52 2024
4
5 @author: gawe_
6 """
7
8 import sklearn.metrics as mts
9
10 dir(mts)
11
12 help(mts.f1_score)
```

```
Console 1/A x
untitled1.py , width= C:/Users/gawe_/spyder-py3/autosave')
Help on function f1_score in module sklearn.metrics._classification:
f1_score(y_true, y_pred, *, labels=None, pos_label=1, average='binary', sample_weight=None, zero_division='warn')
    Compute the F1 score, also known as balanced F-score or F-measure.
    The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to
Python Console History
conda: base (Python 3.9.13) Line 12, Col 19 UTF-8 CRLF RW Mem 85%
```




technical analysis 🔍

Filter by [classifier](#)

10,000+ projects for "technical analysis"

Order by Relevance ▾

- Framework
- Topic
- Development Status
- License
- Programming Language
- Operating System
- Environment
- Intended Audience
- Natural Language
- Typing

	technical-analysis 0.0.4 Technical Analysis with Python	Nov 21, 2023
	odoo-technical-analysis 0.1.6 OdoO Technical Analysis	Dec 3, 2023
	technical 1.4.3 Technical Indicators for Financial Analysis	Feb 5, 2024
	finbright-technical 0.0.1 finbright private utils package	Feb 7, 2023
	technical_indicators 0.0.16 This module provides some technical indicators for analysing stocks.	Jun 3, 2014
	technical-indicator 1.0.3 Technical Indicator is a Python package for calculating technical indicators from financial time series datasets	Apr 7, 2023
	openbb-technical 1.1.4 Technical Analysis extension for OpenBB	Mar 11, 2024

[Help](#)[Sponsors](#)[Log in](#)[Register](#)

technical-analysis 0.0.4

[Latest version](#)

```
pip install technical-analysis
```



Released: Nov 21, 2023

Technical Analysis with Python

Navigation

[Project description](#)[Release history](#)[Download files](#)

Project links

[Homepage](#)

Statistics

GitHub statistics:

★ Stars: 4

🔗 Forks: 3

🔔 Open issues: 0

Project description

Technical Analysis for Python

Technical Analysis (TA) is the study of price movements.

This package aims to provide an extensible framework for working with various TA tools. This includes, but is not limited to: candlestick patterns, technical overlays, technical indicators, statistical analysis, and automated strategy backtesting.


Why Use This Library?


The Technical Analysis Library is still in its early days, but already has the following:

1. Recognition for 30+ Candlestick Patterns
2. 10+ technical indicators
3. 10+ moving average types (including adaptive)
4. Technical overlays
5. Automated backtests and strategies

Tümü Okul Uygulamalar Belgeler Web Diğer

En iyi eşleşme

 **Anaconda Prompt (anaconda3)**
Uygulama



Anaconda Prompt (anaconda3)
Uygulama

Aç
Yönetici olarak çalıştır
Dosya konumunu aç
Başlangıç'a sabitle
Görev çubuğuna sabitle
Kaldır

```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.19045.4170]  
(c) Microsoft Corporation. Tüm hakları saklıdır.  
(base) C:\Users\gawe_>pip install technical-analysis_
```

anaconda prompt (anaconda3)

```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.19045.4170]  
(c) Microsoft Corporation. Tüm hakları saklıdır.  
(base) C:\Users\gawe_>pip install technical-analysis_
```




CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Launch

in one user-friendly environment.

Install

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  
(base) PS C:\Users\gawe_> pip install technical-analysis
```



Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated