

بخش ۱

A و Unsupervised Learning بدون هیچ است، الگوها یا ساختارهای پنهان را در

داده ها پیدا می کند مثل خوشه بندی یا کاهش ابعاد در حال که Supervised Learning با داده های

برچسب کار می کند و هدفش پیش بین خروجی بزرگ داده های جدید است.

B: ضروری است زیرا:

۱- بهبود عملکرد، سریع به جواب رسیدن می رسد الگوریتم های مثل Gradient Descent و

مقیاس ویژگی ها یکسان باشد.

۲- حساسیت به مقیاس، الگوریتم های مثل KNN و SVM به فاصله بین نقاط وابسته اند،

اگر مقیاس ویژگی ها متفاوت باشد ویژگی با مقیاس بزرگتر تأثیر ناعادلانه ای روی نتیجه می گذارد.

۳- عملکرد بهتر مدل، در روش های مثل رگرسیون خطی یا شبکه های عصبی مقیاس های نابرابر

غضایب یا وزن ها را این ثابت می کنند و فقط مدل را کاهش می دهند.

۴- استاندارد سازی داده ها، با نرمال سازی یا استاندارد سازی، مدل ها داده ها را منصفانه تر

مقیاس و پردازش می کنند.

C: Standardization داده‌ها را به شکلی تغییر می‌دهد که میانگین صفر و انحراف معیار یک شود.

ولی در Normalization داده‌ها را به یک بازه مشخص مثل (معمولاً 0 تا 1) می‌برد.

D: زیرا داده‌ها را به یک بازه مشخص (معمولاً 0 تا 1) می‌برد، فاصله‌ها و روابط بین داده‌ها

را حفظ می‌کند و فقط مقیاس را تغییر می‌دهد، محاسبات آن بسیار ساده و با سرعت بالا انجام می‌شود.

لنگ می‌کند اگر سیستم‌های مثل گزاینان نزدیک به صفر و یا دورتر از یک جواب می‌دهند.

E: داده‌ها را استاندارد می‌کند تا عمل‌ها دقیق‌تر و منصفانه‌تر عمل کنند، به ویژه در روش‌های حساس

به توزیع و مقیاس

کامبرد، داده‌ها را استاندارد سازی می‌کند، نسبت به توزیع حساسیت بالا دارد، حذف اثر مقیاس

بهیچ‌گونه عمل نمی‌کند

F: تکنیکی است که به مدل‌های یادگیری ماشین اجازه می‌دهد تا از بیش‌ترین برداش جلودیگر کنند این

کار با افزودن یک Penalty به Loss function انجام می‌شود تا بهیچ‌گونه مدل (مثل ضرایب بزرگ)

محدود شود.

تذکره ۱- $overfitting$ (بیش‌برازش) : مدل بیش از حد به داده‌های آموزشی وابسته می‌شود

و نیز ویژگی‌های غیر ضروری را هم یاد می‌گیرد که نتیجه آن هم روی داده‌های آموزشی خوب کار می‌کند ولی روی داده‌های جدید دقت پایین و تعمیم‌پذیری ضعیف می‌شود.

۲- $Underfitting$ (کم‌برازش) : مدل به اندازه کافی پیچیدگی ندارد و نمی‌تواند الگوهای مهم

داده‌های آموزشی را یاد بگیرد که نتیجه آن هم بر داده‌های آموزشی و هم جدید عملکرد ضعیف دارد.

H: زیرا روشی است که داده‌ها را به چند بخش ($Fold$) تقسیم می‌کند و مدل را چندین بار آموزش و آزمایش می‌کند تا عملکردش را بهتر ارزیابی کند.

کاربرد در $Train/Test$: دقت ارزیابی مدل را بالا می‌برد، داده‌ها را بهتر استفاده می‌کند و از

مشکلات تقسیم تصادفی در $Train/Test$ جلوگیری می‌کند.

$Gradient Descent$: ابتدا با مقادیر اولیه تصادفی برای پارامترها (مثل وزن‌ها) آغاز می‌شود

پس مشتق تابع هزینه نسبت به هر پارامتری سبب می‌شود که نشانه دهد هزینه در چه جهتی بیشتر تغییر می‌کند

این فرآیند بارها تکرار می‌شود تا به کمینه تابع هزینه (نقطه بهینه) برسیم با تغییرات خیلی کوچک شوند.

نکته: به خاطر توانایی یادگیری خودکار، مدل‌سازی پیچیدگی‌ها و عملکرد قوی در داده‌های بزرگ برای

بخش دوم :

A. به دلیل ترکیب نادر و قدرت واکه سیستم غرض یا بیرون با یک شبکه که به یک استاندارد در

علم داده تبدیل شده.

NumPy یک کتابخانه است برای محاسبات عددی در پایتون که بیشتر با آرایه‌های چند

بعدی کار می‌کند و برای عملیات ریاضی سریع و تسهیل عمل چیر خطی یا محاسبات بردار عالی و داده

محکم‌ال و سرعش بالاست.

Pandas یک کتابخانه است ولی برای تحلیل و مدیریت داده‌ها طراحی شده با جدول dataframe

و سری‌ها کار می‌کند داده‌های نامشمار را پشتیبانی می‌کند و برای کارهای مانند فیلتر کردن و مرتب‌سازی

خواندن فایل‌های داده‌ها بسیار مناسب است.

۵:

به دلایل : ۱- انعطاف پذیری (می‌تواند متنوع نمودار را با تنظیمات دلخواه بسازیم).

۲- کنترل کامل به رنگ‌ها، برجسته‌ها، مقیاس محورها داریم به اشتیاق از علم داده‌ها را می‌توان

و متن‌ها را به‌راستایی بسیار استفاده و یادگیری سریع

12. 1- زیجاس پرفیورمنس 2- سادس پرفیورمنس 3- ادفا 4- Pandas 5- تکرر آجاس 6- تنظیمات

E

1- از لایه لایه Def استفاده می کنیم.

2- سپس اسم تابع را می نویسیم.

3- داخل پرانتز پارامترها را می گذاریم مثال :
def salam (name) :
 Print ("سلام " + name)

4- با دو نقطه خط را تمام می کنیم. (:)

5- بدنه تابع را با indent می نویسیم.

F : 1- کوتاه خوانایی بالا 2- سرعت بالا 3- فقط ساخت لیست (تغییر فشرده نشانی نمی دهد)

6- انعطاف پذیری

G : می توانیم آن را به صورت 1- از کتابخانه Pandas 2- با ماژول CSV خواند

1) import pandas as pd

data = pd.read_csv("File.csv")

Print(data)

2) import csv

with open("File.csv", "r", encoding="utf-8") as file:

 reader = csv.DictReader(file)

 for row in reader:

13

:h

ساختار ۱

JSON : ساده تر و شبیه به لی و آرایه است
XML : جتنی بزرگ و پیچیده تر است

جمع د سرعت

- حجم کمتر و پردازش سریع تری دارد
- بدلیل تک صای افزاینده شلین تر است

کاربرد:

- بیشتر در API ها و وب سرویس های مدرن استفاده می شود
- در سرویس های قدیمی، اسناد و تنظیمات سیستم ها کاربرد دارد

خوانایی و پردازش

- خوانایی و مستقیماً در جاوا اسکریپت پردازش می شود

- نیاز به ابزارهای خاص برای پردازش دارد