**Data Visualization with Python**

28.04.2023

**Data Visualization introsu**

**Matplotlib**

**Seaborn**

**Plotly-3D**

**Visualization project**

**Explorative Data Analysis(EDA) icin kullanilan gorsellestirme kutuphaneleri**

**Matplotlib:** create figure, axes, subplots (**import matplotlib.pyplot as plt**)

**Seaborn:** built on matplotlib and can be used together with it (**import seaborn as sns**)

2002’de John Hunter Matlab arayuz icin ilk surumunu yayinliyor.

**Functional:**

Plt.plot(age, salary) : burda age x ekseni, y ekseni salary

Plt.xlabel(“age”)

Plt.ylabel(“salary”)

Plt.title(“Salary by Age”)

\*

\*plotlamak kavrami kullanilir.

**Object-oriented:**

Fig, ax **=** plt.subplots() figure axes giriyorum.

ax.plot(age, salary, “r”)

ax.set\_xlabel(“Age”)

ax.set\_ylabel(“Salary”)

Ax.set\_title(“Salary by Age”)

Fig=plt.figure()

Burasi axes ve figure ile oynayabilecegim tek yer olacak.

Object oriented’de functional’dan farkli olarak bir atama var.

Her bir x,y ekseni birer axis, bunlardan olusan grafik axes olacak benim icin. Yani plot axes.

Figure siniftaki tahta, tahtaya asilan panolar axes gibi dusun.

(Axes axis’in cogulu. )

**Figure>Axes>Axis**

**Seaborn:** Matplotlib uzerine insa edilmistir. Istatisteksel analizlerde kullanilir. Yazilimi daha kolay. Gorselleri daha kuvvetlidir.

seaborn’de subplots olacak.

Figure=fisize(6,4)

**Scatter Plot**

Bir datanin normal dagilim gosterip gostermediginin goruntusunu verir.

x eksenini ve y eksenini array olarak tanimladim diyelim:

import matplotlib.pyplot as plt

import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])

y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

**plt.scatter**(x,y)

plt.show

Dedigim zaman, scatter grafigini veriyor.

plt.show yerine, **;** kullanirsam da bir satirlik aciklama yazisini gormem.

Indekse gore siralama yaptigindan, x ve y eksenindeki eleman sayilarinin esit olmasi gerekiyor. Yani bire bir orten.

**Bar Chart**

Kategorik veri kullaniyorsam bir data icinde bar chart kullaniyorum.

En azindan x ekseninde class olan yerlerde kullaniyorum. Classlama var yani.

Unique degerlerine, frekanslarina, moduna, count, meanine, bakabilirim ben bu kategorik datanin.

**Histogram**

Histogram da bar chart’la cok benziyor. Birbiri ile karistiriliyor. Ama histogram numeric

Aralik olarak datalar varsa.

Histogram normal dagilimi gosteren grafiktir. Mesela can egrisi normal dagilimdir.

**Box Plot**

x normal dagilimli bir array olsun diyelim

plt.boxplot(x,)

plt.show

boxplot’da outlier’a bakarim. Cok guzel gorunur.

ML=prediction

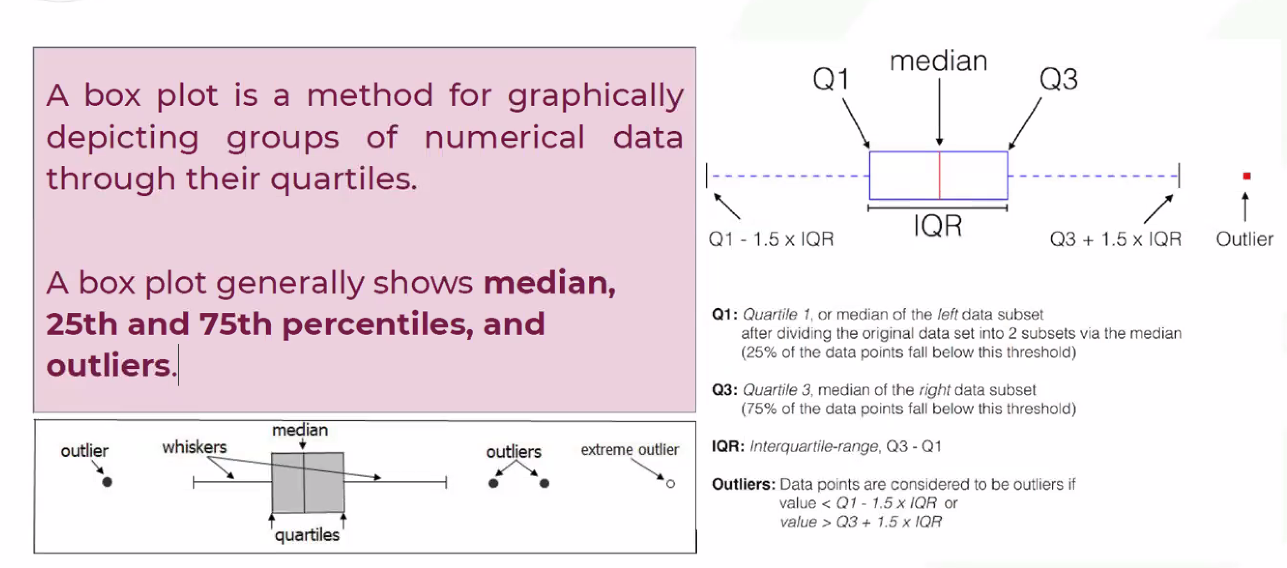
ML’de egitim datasi ile test datasini kiyasliyorum.

Makineye ogretiyorum, %98 accuracy ile predict ediyor mesela.

Piskolojiyi olcmek anketlerde ozellikle piskometri ile yapiliyor. En gercekci sonuclari veriyor.

Box’in icindeki cizgi median

Q1-1.5IQR ve Q3+ 1.5 IQR disindakiler outlier.



**Pie Chart**

En fazla 5-6’yi gecmeyecek sekilde kategorik data icin kullanirim.

**Distributions plots (dagilimlarda bakiyorum)**

kdeplot

rugplot

displot

Histplot

**Categorical Plots**

Barplot

Countplot

Boxplot

Swarmplot

Violinplot

**Comparison Plots**

Joinplot

**Pairplot (en cok bunu kullaniyoruz)**

Catplot

Matrix plot

Grid plot

|  |
| --- |
| Import seaborn as sns  tips=sns.load\_dataset(“tips”)  **sns.rugplot(x=”total\_bill”, data=tips)**  **sns.rugplot(tips[”total\_bill”])**  **#bu ikisi ayni gorseli verir.** |

**Distributions plots**

**sns.rugplot(x,y,data)**

(y yerine tek degiskenle bakmak istersem sadece x veriyorum. Bu durumda bana y ekseninde yogunlugu veriyor (density))

Datamin icinde bulunan

**columns,feature**

**Row, index, satir, observation**

Displot kutuphanede dagilimlari gostermek icin ilk cikan fonksiyon. Zamanla gelistikce yerine hisplot gibi baskaca seyler geldi. Displot yazinca icerisinde hangi func calissin istedigimi yazabildigim baskaca plot’lar var.

**Rugplot’da y ekseninde yogunluk cikiyordu.**

**hisplot’da ise y ekseninde count cikiyor.**

**sns.hisplot(x=”total\_bill”, data=tips, kde=True)**

Default da olsa icine yazmakta fayda var projeyi bir yerde paylasacaksan

sns.hisplot(tips[“total\_bill”])

**Distribution plots-kde**

kdeplot da rugplot gibi calisiyor. Sadece x verdigimde, y’de yogunlugu gosteriyor.

**Categorical Plots**

**Categorical Plots -countplot**

Sns.countplot (x= ‘day’, data=tips)

Sns.countplot(tips[‘day’]

Bir de **hue= ‘sex’** dedigimde mesela, icine verdigimiz kategorik degiskene gore renklendiriyor datayi.

Eger targetimiz kategorik bir degiskense, hue icin mutlaka o target’taki kategorik degiskeni veririrm. Yani hue train ettigimiz sutuna gore olmali.

Yani mesela smoker’lari tahmin edeceksem modelimde. hue= “smoker” derim.

**Categorical Plots- barplots**

sns.barplot(x=’day’, y=’total\_bill’, data=tips)

**sns.barplot(**x=’day’, y=’total\_bill’, data=tips, hue=’sex’) bu sekilde yazdigimda cinsiyete gore renklendirerek getiriyor grafigi

**Categorical Plots- barplots**

Gunlere gore toplam bahsisin dagilimini gosteren grafik

**sns.boxplot(**x=’day’, y=’total\_bill’, data=tips)

Median cizgisi bir tarafa daha yakinsa, carpiklik var demektir. Tam ortadaysa normal dagiliyor demektir.

Outlier tespit edince napiyorsun:

Gercek dunya datalari outlierlardan olusur. Hemen atamazsin.

**Comparison Plots**

**Comparison Plots-pairplot**

Birseyin aynisiyla kendisiyle korelasyonu 1’dir. Fahreneit ile celcius korelasyonu da 1 cikar mesela. Ayni seyi olcenlerden hangisinde daha az missing value varsa onu tutabiliriz aslinda.

sns.pairplot(tips, hue= “smoker”)

#corner=True

**Grids**

**Pairgrid**

Pairplottan farki ustte hangisi olsun, altta hangisi olsun secebiliyorum.

Yukariya scatter koy diyorum mesela

**g=g.map\_diag(sns.histplot)**

**FacetGrid**

Satirda ne yer alsin, sutunda ne yer alsin onu seciyorum bir de bunda.

**#facetgrid ile pairgrid cok kullanilmaz.**

**Matrix Plots**

**Heatmap (isi haritasi) (onemli grafik)**

Korelasyon tablolarinda bakilir. Hangi feature’un hangi feature ile corelasyonu oldugunu korelasyonun derecesini gosterir.

1’e yaklasanin correlation’i yuksek demektir. Hem 1 hem -1 icin.

Birlikte hareket etme durumu.

**29.04.2023**

**Matplotlib**

#matplotlib.org sitesinde grafikler var.

#ilgili gorselin codelari da var.

#burdan da alabilirsin

|  |
| --- |
| **!pip install matplotlib**  **!pip install seaborn**  **import matplotlib as mpl** #versiyon kontrol, default degerlere donme  #normalde pyplot ile kullanacagim matplotlibi, ama default degerlere donmesini istedigim zaman mpl kullanacagim. ya da versiyon kontrol edeceksem mpl kullaniyorum.  **print(mpl.\_\_version\_\_)**  #versiyonu gosterir  #diger durumlarda plt kullanacagim:  **import matplotlib.pyplot as plt**  **%matplotlib inline**  #bu kodu butun gorselleri ust uste bir satirda basmasini engellemek icin kullaniyorum  **%matplotlib notebook komutu var bir de.**  #bir de gereksiz yorumlari aciklamalari almamak icin yazdigimiz bir kod var:  **import warnings;**  **warnings.filterwarnings("ignore")**  **figure > axes> axis**  **Axes:plot** |

**Mesela bir gorseli jupytere aktarmak istiyorsun:**

#eger gorseli sag tiklayip copy image yapip, kod satirini markdowna cevirip copy paste yaparsan gorsel satira yapistirilir.

#shift enter yaparsan da gorseli gosterir.

**Plot olusturmak:**

x ve y’yi tanimladim

**plt.plot(x, y)**

**plt.show()**

**plotun ustunda [<matlotlib.. diye baslayan aciklamayi gormemek icin:**

#jupyter lab'da ; kullanirim

# ya da plt.show() diyebilirim.

#grafigin beyaz gorunen arka planini modifiye edebiliriz, renklendirebiliriz:

**print(plt.style.available)**

#plt.style bana arka plana donuk opsiyonlari gosteriyor

#available style options

**Style belirlemek icin:**

**plt.style.use('seaborn-dark')**

**Figure size belirlemek icin:**

**plt.figure(figsize=(10,5))**

#figsize ile figure'un size'ini ayarlayabiliyorum.

#parantezin icine shift tab tab yaparsam icine hangi parametreleri girebilecegimi anlayabilirim

**Plot modifiye:**

**plt.plot(x, y, 'bo')**

**plt.show()**

#mesela plot'dan sonraki parantezin icine girdim

#bana x,y,'bo' diye bir parametre gosterdi. mavi dot yapiyormus.

#bu sekilde plotun cizgisini belirleyebiliyorum.

#'ro' olani da red dot mesela

**plt.figure(figsize=(10,5))**

**plt.plot(x,y, "+g")**

**plt.plot(x,y, "g+")**

#yerleri degistirdiginde de calisiyor.

#python'da positional argument degilse calisiyor genelde

#pano figure du, tablolar axes'di. x ve y de axis

#yani **figure > axes> axis**

**Label ve title belirlemek:**

**plt.figure(figsize=(10,5))**

**plt.plot(x,y, "go")**

**plt.plot(x,y, "go")**

**plt.xlabel("X-Axis")**

**plt.ylabel("Y-Axis")**

**plt.title(label="AXES");**

**Coklu data ve legend vermek**

**age= np.arange(25,44,2.5)**

**salary=np.arange(40000, 76000, 5000)**

**salary2=[45372,46852,69401, 56790, 73000, 75000,80000, 83454]**

#mesela age ve salary gibi 2 eksen belirledim

#indeksler esit olmazsa hata alirim grafigi cizmez.

#ama arrayler yerine dataframe ile çalışırsak index kaynaklı hata almayız. null değer olsa bile.

**Plotlamak:**

**plt.plot(age, salary, "b:", label= "Salary")**

**plt.plot(age, salary2, "r-", label= "Salary2")**

#bu iki kodla plotluyoruz. X ve y degerimi verdim, rengini belirledim. Label adi verdim.

#tek axes icinde 2 plot yapiyoruz. biri salary digeri salary2 icin.

**plt.xlabel("Age")**

**plt.ylabel("Salary")**

**plt.legend()**

**plt.title("SALARY BY AGE");**

#bunlar da axisimizi sekillendirdigimiz kodlar

#y axis'de salary var. salary de salary2 de maas cunku. Ortak kullaniyorlar y axisi.

**#plt.legend()** iki salary'i birbirinden ayirmak icin kullaniyorum.

#renk, sekil ve degisken ismi olarak legend bize gosteriyor sol ust kosede (default).

#ben legend'a bakarak, mavi cizgi salary'i, kirmizi da salary2'yi gosteriyormus derim

#**plt.legend(["First Line", "Second Line")]** seklinde de yazilabilirdi.

**Interaktif hale getirmek icin:**

**%matplotlib notebook**

**%matplotlib inline**

# basina %matplotlib notebook eklersem, grafik interaktif hale geliyor

#Javascript error verirse, javascript install etmen gerekiyor

#aynisini %matplotlib inline olarak da yapabilirsin.

**Legend:**  (etiketleme)

#legend'in icine yazabilecegim parametrelere de baktigimda,

#**bbox\_to anchor** var mesela, location girebiliyorum. loc='best' dedim mesela

#**loc='upper right'** dersem sag uste tasir, **loc='center'** dersem ortalar.

#ya da bbox\_to anchor diyip koordinat girebiliyorum.

# ilk iki deger ile figure icinde gezerken, sonraki iki degerle axes'te gezebilirsin ince ayar icin. 1.0’dan fazla yazarsan axes disina cikiyor legend.

**Linewidt ve linestyle belirlemek:**

#cizgilerin kalinligi ile de oynayabiliriz:

#linewidth=8 diyerek mesela

**plt.plot(age, salary, "b-", linewidth= 4, label = "Salary")**

# setting up legends **linewidth = lw** de yazabilirsin

**plt.plot(age, salary2, "r", ls="-.", label = "Salary2")**

# setting up legends **linestyle=ls** de yazabilirsin.

#linestyle da stilini degistiriyor

**plt.plot(age, salary2, "r-", label= "Salary2", linestyle="-.")**

**Functional Method and Object Oriented Methods**

**Functional Method**

|  |
| --- |
| **plt.style.use("seaborn-darkgrid")**  **plt.plot(age, salary)**  **plt.xlabel("Age")**  **plt.ylabel("Salary")**  **plt.title("SALARY BY AGE")**  **plt.grid(color="purple")** |

**plt.grid(color="purple")**

#bunla da style'i gridli olanlarda gridleri modifiye edebiliyorum

#gridin rengini degistiriyor mesela.

**OOM**

|  |
| --- |
| **fig, ax= plt.subplots()**  **ax.plot(age, salary, "r")**  **ax.set\_xlabel("Age")**  **ax.set\_ylabel("Salary")**  **ax.set\_title("SALARY BY AGE")**  #bu da object oriented  #set ile yapiyorum. |

**Functional Method Multi Plot (2 satir 1 sutun)**

**plt.subplot(2,1,1)**

**plt.plot(age, salary, "r")**

**plt.subplot(2,1,2)**

**plt.plot(age, salary, "g")**

#2 satirlik 1 sutunluk grafigimdeki 1. veya 2. plot demis oluyorum basitce.

#tek bir figure'de 2 plot olmus oldu. 1 tahta 2 pano gibi

**OOM Multi Plot (2 satir 1 sutun)**

**fig, ax= plt.subplots(2,1)**

**ax[0].plot(age, salary, "r")**

**ax[1].plot(age, salary2, "g")**

**plt.tight\_layout();**

#bu son kodu yazilarin birbiri icine girmesini onlemek icin yaziyorum.

**Functional Method Multi Plot (2 satir 3 sutun, toplam 6 axes)**

**plt.subplot(2,3,1)**

**plt.plot(age, salary, "r")**

**plt.subplot(2,3,2)**

**plt.plot(age, salary, "g")**

**plt.subplot(2,3,3)**

**plt.plot(age, salary, "b")**

**plt.subplot(2,3,4)**

**plt.plot(age, salary, "bo")**

**plt.subplot(2,3,5)**

**plt.plot(age, salary, "y")**

**plt.subplot(2,3,6)**

**plt.plot(age, salary, "p")**

**plt.tight\_layout();**

#boylece 2 satir 3 sutunluk, 6 tane plot'u tek bir figur icinde cizdirmis olduk.

**mpl.rcParams.update(mpl.rcParamsDefault)**

## grafiklerdeki tum modifikasyonlari default degerlere dondurduk.

#fig size tum notebookda esit olmasini istiyorsam onu da mpl ile yaparim

**OOM Multi Plot (2 satir 3 sutun 6 axes)**

#2 satir 3 sutun varsa bunlarin nasil kombinasyonlarini alirim

**fig, axs = plt.subplots(nrows=2, ncols=3)**

#2 satir 3 sutun halinde subplotlar seklinde yapiyor burda figure’u.

**axs[0, 0].plot(age, salary, "r")**

**axs[0, 1].plot(age, salary2, "g")**

**axs[0, 2].plot(age, salary, "b")**

**axs[1, 0].plot(age, salary2, "pink")**

**axs[1, 1].plot(age, salary2, "purple")**

**axs[1, 2].plot(age, salary2, "orange")**

**fig.tight\_layout()**

**plt.show()**

#fig, axs = plt.subplots(nrows=2, ncols=3) olarak icine, nrows ve ncolumsn da yazdim

#burda indeksleri tek [ ] icine yazdim.

#ayri ayri da tirnak icine yazarak yapabilirim indeksleri:

**fig, axs = plt.subplots(2, 3)**

**axs[0][0].plot(age, salary, "r")**

**axs[0][1].plot(age, salary2, "g")**

**axs[0][2].plot(age, salary, "b")**

**axs[1][0].plot(age, salary2, "purple")**

**axs[1][1].plot(age, salary2, "y")**

**axs[1][2].plot(age, salary2, "g")**

**fig.tight\_layout()**

**plt.show()**

#ozetle object oriented'da 5 satir daha tasarruf ettim.

**bir grafikte 4-5 tane nasil cizdiririz peki:**

**x=np.arange(0,30,0.2)**

**fig, ax= plt.subplots(figsize= (14, 8))**

**ax.plot(x, x-1, color= "green", linewidth=3, linestyle= "-")**

**ax.plot(x, x-2, color= "red", lw=3, linestyle= "-.")**

**ax.plot(x, x-3, color= "yellow", lw=3, ls= ":")**

**ax.plot(x, x-4, color= "purple", lw=3, ls= "--")**

**plt.show()**

#maas kalibi icinde 4 tane class varsa, yani 4 derece maas varsa yani

#4’unu de farkli bir cizgi ile tek bir grafikte gosteriyor.

fig, ax = plt.subplots(2, 2, figsize = (12,6))

plt.suptitle("LineStyles & Markers & Markersize & Color", fontsize = 16, y = 0.95)

#y=.. dedigim yer basligin konumuyla ilgili

#bir sey yazmazsam orjinal yerini korur

**Farkli grafiklerde gostersin istersem:**

**fig, ax = plt.subplots(2, 2, figsize = (12,6))**

**plt.suptitle("LineStyles & Markers & Markersize & Color", fontsize = 16, y = 0.95)**

#y=.. dedigim yer basligin konumuyla ilgili

#bir sey yazmazsam orjinal yerini korur.

**ax[0, 0].plot(x, "r", marker= "+", markersize = 5)**

**ax[0, 1].plot(x-2, "y", marker= "o", ms = 5)**

**ax[1, 0].plot(x-3, "black", marker= "s", ms = 5, lw = 5)**

**ax[1, 1].plot(x-4, "g", marker= "1", ms = 5)**

**plt.show()**

#farkli farkli grafiklerde gorunmesini istersem boyle yapiyorum.