

Adatok analízációja és adatbányászat Python nyelven



Szabó Ákos Dániel

GEIAL526-ML Adatelemzési és adatbányászati módszerek
tárgyból

Beadandó II.

Tartalomjegyzék

Feladat specifikáció.....	3
Használt technológiák, adatkészletek bemutatása.....	4
Adatkészlet.....	4
Python	5
Könyvtárak és csomagok:.....	5
Adatkészlet beolvasás és tisztítás	6
Tisztítás	6
Adatok elemzése	7
Hisztogrammok	9
Korrelációs Mátrix.....	13
Döntési Fa	15
Adatok felosztása	15
Tanítás.....	16
Megjelenítés	17
Gyökér (Root) node.....	17
Elágazás a bal ágon (feature_4 <= 14.91)	17
Elágazás a jobb ágon (feature_4 > 14.91)	17
További elágazások	18
Confusion Matrix.....	19

Feladat specifikáció

Adatelemzés és adatbányászati módszerek

2. féléves feladat

Válasszon ki egy adathalmazt (kaggle.com, seaborn adatcsomag, saját adatok) (kivéve iris.csv) és

végezze el az adatok elemzését Python-ban az alábbi módszertant alkalmazva.

1. Töltse be az adatokat és végezze el a szükséges adattisztítási lépéseket.
2. Elemezze az adathalmazt leíró statisztikai mutatókkal, grafikonokkal. Szűrje ki a kiugró értékeket.
3. Keresse meg a változók között fennálló korrelációs viszonyokat.
4. Klaszterezze az adatokat egy kiválasztott algoritmussal, vagy készítsen osztályozó modellt az adathalmazhoz egy kiválasztott módszerrel és mutassa meg a modell pontosságát.

Az elemzésről készítsen pdf dokumentumot, amely tartalmazza: az adathalmaz forrását és leírását,

az alkalmazott eljárások rövid ismertetését, a futási eredményeket és az eredmények kiértékelését. A

feladathoz csatolni kell a Python kódot is magyarázattal ellátva (kommentezve).

Leadási határidő: 2023. december 8.

A feladat leadása az aláírás megszerzésének a feltétele.



Feladat leadás módja: MS Teams feladatkiíráshoz feltölteni

Használt technológiák, adatkészletek bemutatása

Adatkészlet

Kaggle oldalon választottam a feladat specifikációban ismertetett forrásból ezt az adatkészletet: [Drug Classification \(kaggle.com\)](https://www.kaggle.com/datasets/abdulqaderkhan/drug-classification)

Ezeket az adatokat egy gyógyszergyártó cégtől gyűjtötték, amely címkézte a gyógyszerek adatkészletét és az azt befolyásoló paramétereket. Ezért a betegség és a beteg típusa alapján meglehetősen határozni milyen gyógyszer ajánlott.

# Age	Sex	BP	Cholesterol	# Na_to_K	Drug
Age of the Patient	Gender of the patients	Blood Pressure Levels	Cholesterol Levels	Sodium to potassium Ration in Blood	Drug Type
	M 52% F 48%	HIGH 39% LOW 32% Other (59) 30%	HIGH 52% NORMAL 49%		DrugY 46% drugX 27% Other (55) 28%
23	F	HIGH	HIGH	25.355	DrugY
47	M	LOW	HIGH	13.093	drugC
47	M	LOW	HIGH	10.114	drugC
28	F	NORMAL	HIGH	7.798	drugX
61	F	LOW	HIGH	18.043	DrugY
22	F	NORMAL	HIGH	8.607	drugX
49	F	NORMAL	HIGH	16.275	DrugY
41	M	LOW	HIGH	11.037	drugC
60	M	NORMAL	HIGH	15.171	DrugY
43	M	LOW	NORMAL	19.368	DrugY
47	F	LOW	HIGH	11.767	drugC
34	F	HIGH	NORMAL	19.199	DrugY
43	M	LOW	HIGH	15.376	DrugY

1. Figure Adathalmaz

Python

Python egy magasszintű, interpretált programozási nyelv. Széles körben használják adatelemzés, mesterséges intelligencia, webfejlesztés és hardverprogramozás területén. Az egyszerűsége és könnyen érthetősége miatt ideális választás kezdők számára. Számos külső könyvtár érhető el, ami további funkcionalitást biztosít.

Könyvtárak és csomagok:

pandas:

Felhasználása: Adatok importálására, tisztítására, feldolgozására és statisztikai műveletek végrehajtására szolgál.

Példa alkalmazás: A beadandó feladatban az adatkészlet beolvasására és tisztítására használtam.

seaborn:

Felhasználása: Statisztikai grafikonok készítésére alkalmas.

Példa alkalmazás: A beadandó feladatban ezzel a csomaggal készítettem grafikonokat az adatkészletből.

scikit-learn:

Felhasználása: Gépi tanulási algoritmusok implementálására használható, például regresszió, csoportosítás stb.

Példa alkalmazás: A beadandó feladatban ezt a csomagot alkalmaztam korrelációs elemzésre és kiugró értékek azonosítására az adatkészletben.

Adatkészlet beolvasás és tisztítás

A használt könyvtár csomagok bemutatásánál ismertettem a pandas csomagot, amivel beolvastam az általam használt adatkészletet. Mivel konstans az adatkészlet, és egy szinten van a python forráskóddal, így beégetve megadtam a .CSV fájl nevét és elmentettem egy *data* változóba.

Tisztítás

Az eredeti dokumentum nem tartalmazott hibákat ezért az adattisztítás bemutatása érdekében én generáltam bele plusz hibákat a fájlba.

```
,F,NORMAL,HIGH,15,DrugY      #Hiányzó Kor
23,M,NORMAL,NORMAL,,drugX     #Hiányzó Koleszterol
46,F,HIGH,HIGH,100,DrugY      #Kiugró Koleszterol
110,M,NORMAL,HIGH,15.969,DrugY #Kiugró Kor
17,,HIGH,NORMAL,12.766,drugA   #Hiányzó Nem
55,M,,HIGH,11.537,drugC        #Hiányzó Vérnyomás
36,F,NORMAL,,9.065,drugX       #Hiányzó Koleszterol
47,M,HIGH,NORMAL,15.436,      #Hiányzó gyógyszer
```

2. Figure DirtyData

A korral és Koleszterollal lehet kezdeni valamit ha hiányzik vagy kiugró az adat akkor a mediánnal pótlom. Viszont a többi adat kritikus és éppen ha valamelyik ezek közül hiányzik akkor az adatsort eltávolítom.

```
# Delete rows where Date is missing
df.dropna(subset=['Sex'], inplace=True)
print(df.isnull().sum())

df.dropna(subset=['BP'], inplace=True)
print(df.isnull().sum())

df.dropna(subset=['Cholesterol'], inplace=True)
print(df.isnull().sum())

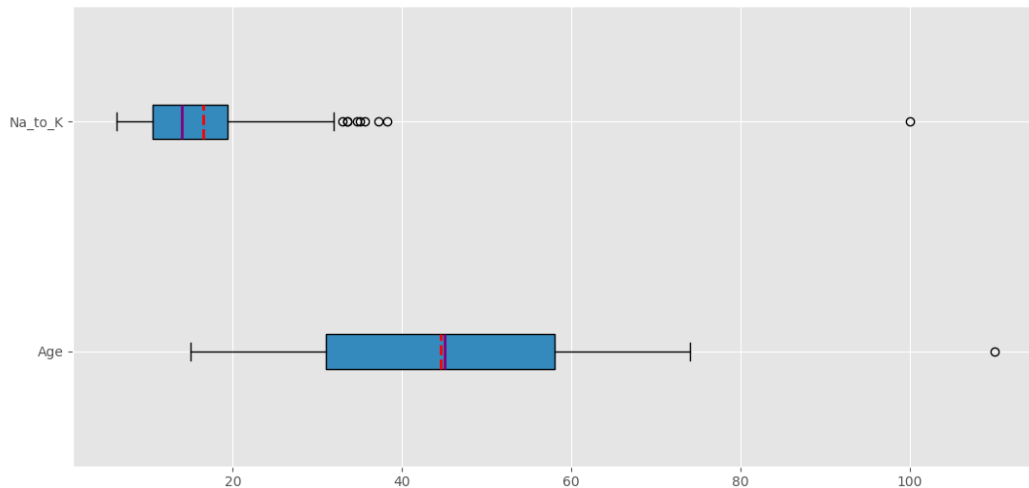
df.dropna(subset=['Drug'], inplace=True)
print(df.isnull().sum())

# Replace missing and incorrect Age,Na_to_K values using median
median = df['Age'].median()
df['Age'].fillna(median, inplace=True)
print(df.isnull().sum())

median = df['Na_to_K'].median()
df['Na_to_K'].fillna(median, inplace=True)
print(df.isnull().sum())
```

3. Figure Kód részlet adattisztítás

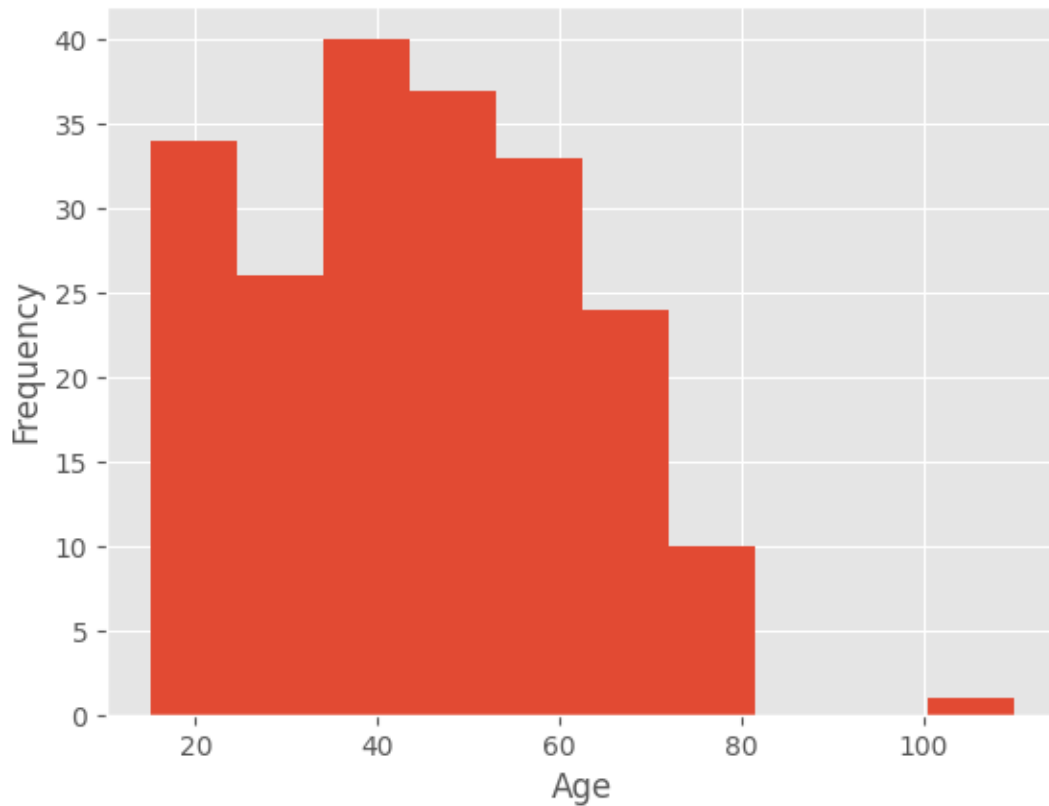
Adatok elemzése



4. Figure Doboz diagramm

A kód egy dobozdiagramot, más néven boxplotot, hoz létre két numerikus változó, azaz 'Age' és 'Na_to_K' oszlopok alapján. A dobozdiagramot gyakran használják a statisztikai adatok szemléltetésére, különösen az adatok eloszlásának és kiugró értékek azonosítására.

A dobozdiagramnak két fő része van: a dobozok és az esetleges kiugró értékek. A dobozok a változók interkvartilis tartományát (Q1 és Q3 közötti területet) mutatják, míg a középső vonal a mediánt reprezentálja. Az esetleges kiugró értékek (outlierek) azok az értékek, amelyek messzebb vannak a dobozoktól, és lehetnek fontosak az adathalmaz eloszlásának megértése szempontjából. Az ábra színekkel és egyéb vizuális elemekkel segíti a könnyebb értelmezést.

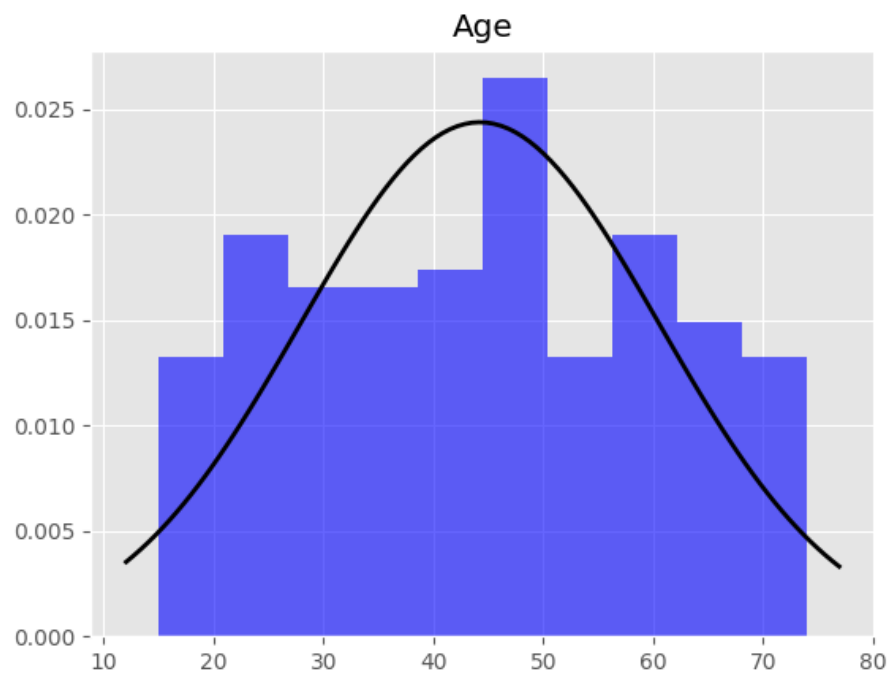


5. Figure Kor gyakoriság

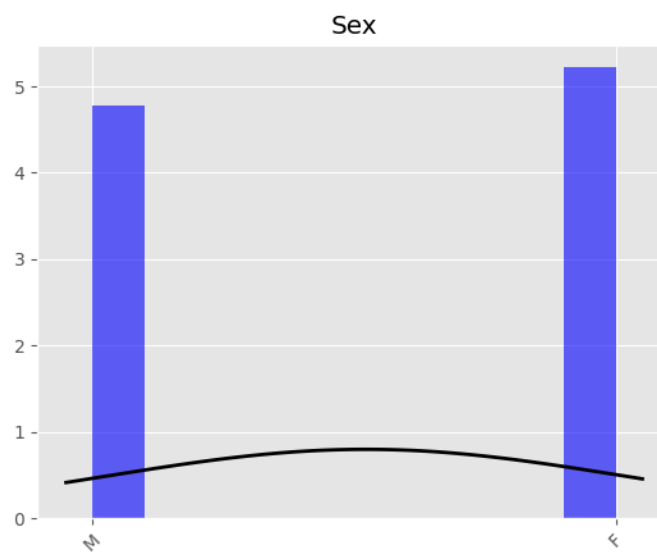
A kód egy hisztogramot is készít az 'Age' (kor) oszlop alapján, és aztán megjeleníti azt. A hisztogram egy olyan diagram, amely bemutatja az értékek gyakoriságát különböző intervallumokban.

Ez a hisztogram segít megérteni az 'Age' oszlop eloszlását, vagyis hogy az életkorok hogyan vannak elosztva az adathalmazban. Az x tengelyen az életkor intervallumok vannak, míg az y tengelyen a gyakoriság (az adott intervallumban található értékek száma). A hisztogram segíthet az életkorok eloszlásának és eloszlásának vizuális értelmezésében.

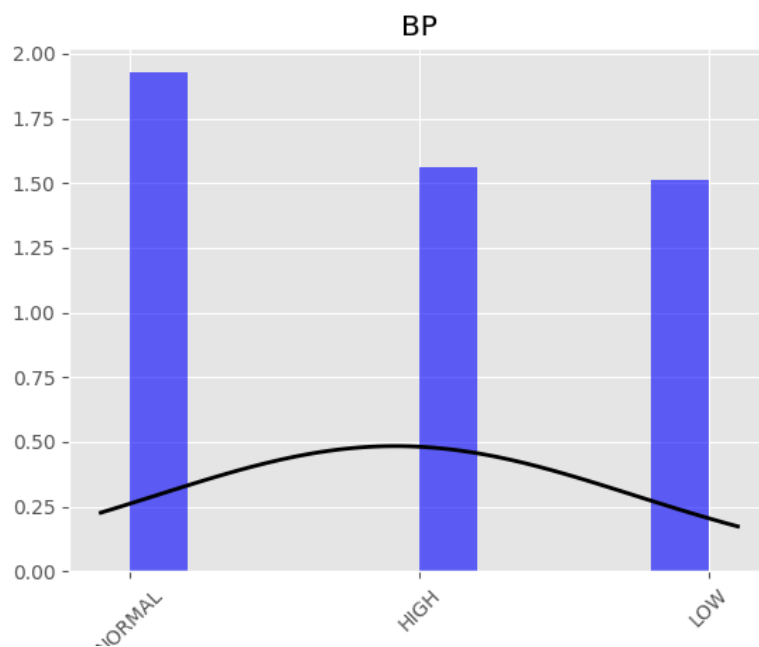
Hisztogrammok



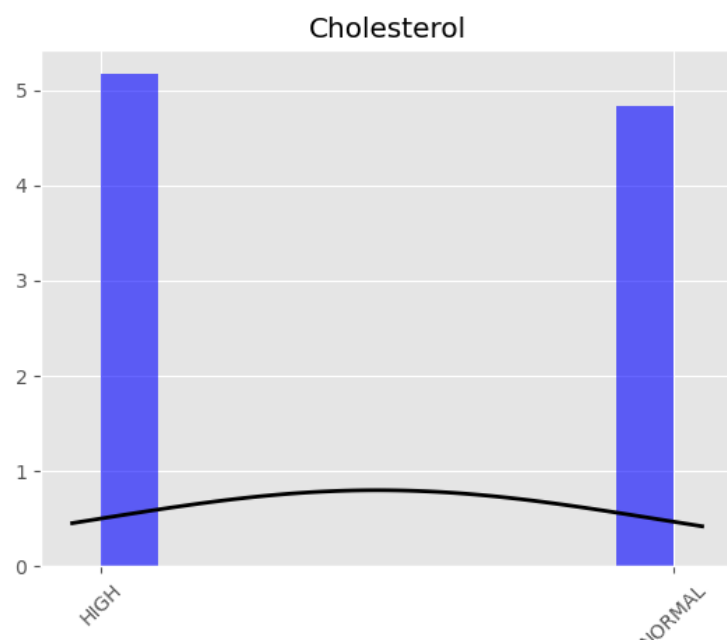
6. Figure Kor Hisztogram



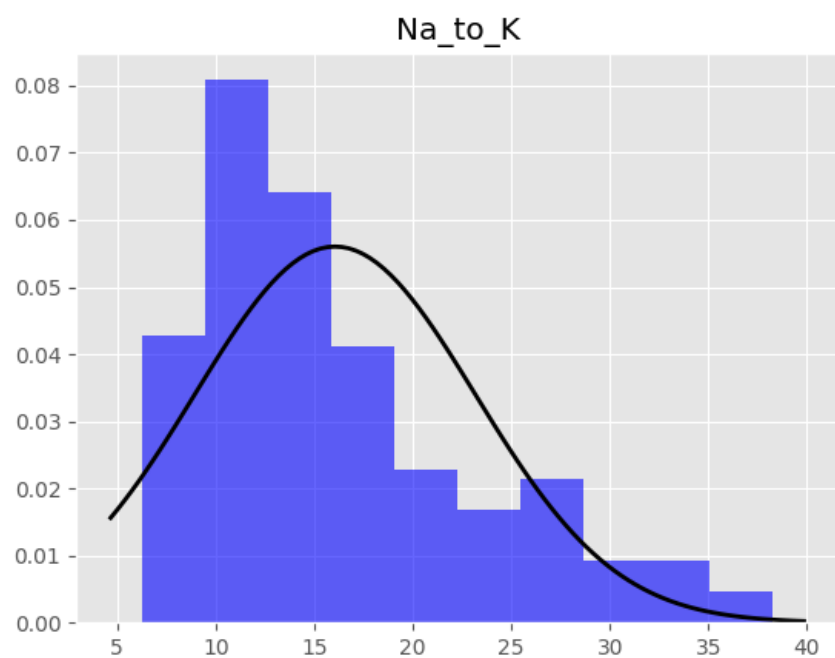
7. Figure Nem Hisztogram



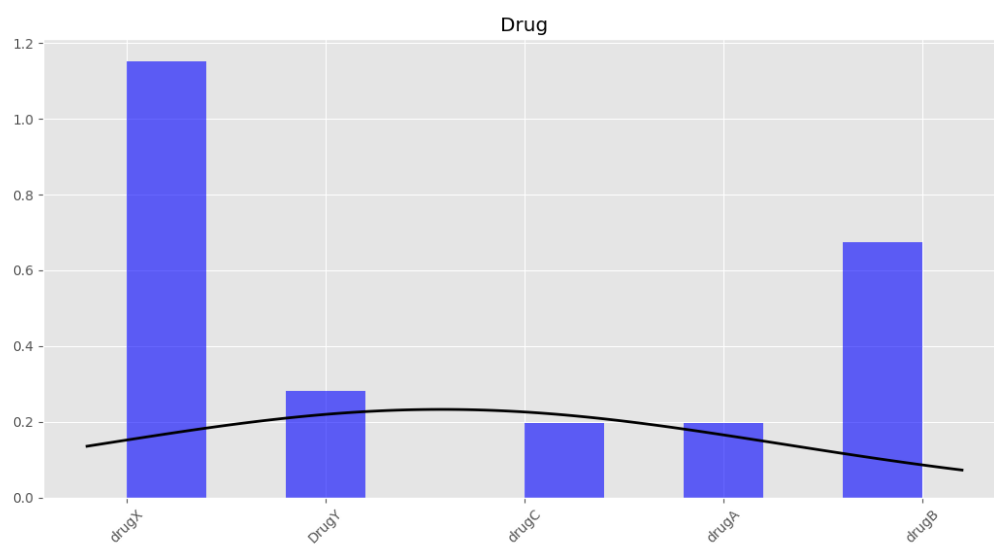
8. Figure Vérnyomás Hisztogram



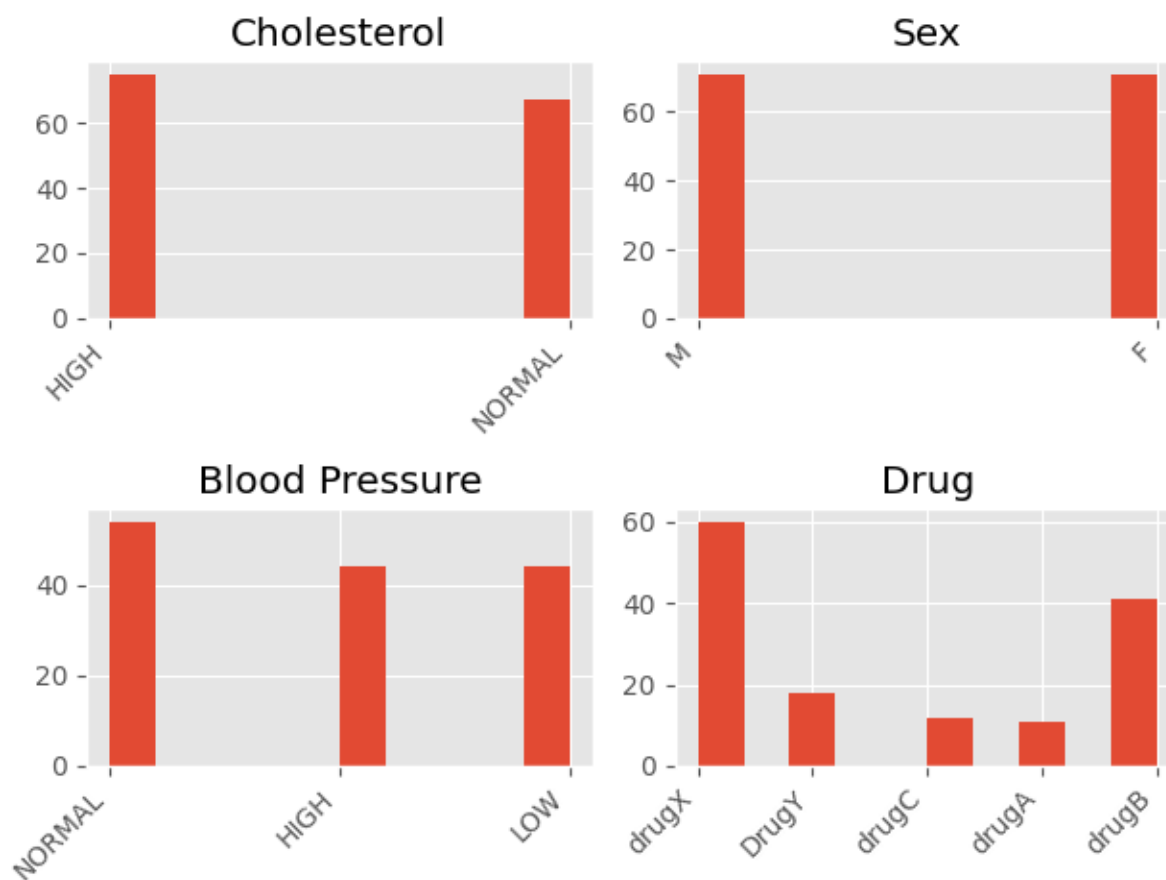
9. Figure Koleszterol Hisztogram



10. Figure Na to K Híztogram



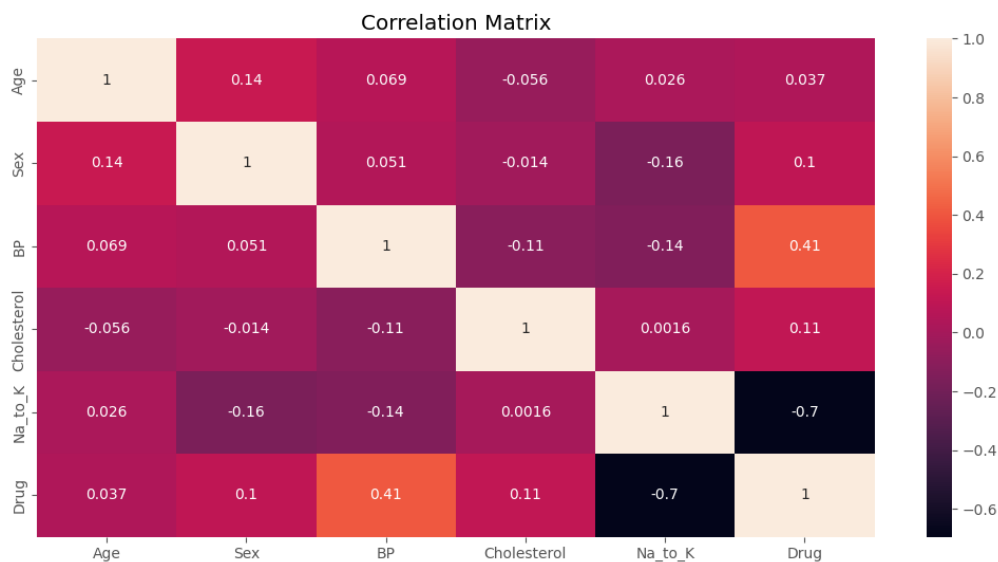
11. Figure Gyógyszer Híztogram



12. Figure Gyakoriság

A hisztogramok hasznosak lehetnek az adathalmaz jellemzéséhez és annak megértéséhez, hogy az attribútumok milyen értéktartományban és eloszlásban vannak. Ezen információk segíthetnek a modellalkotás során az attribútumok kezelésében és az esetleges további adatfeldolgozási lépések kiválasztásában.

Korrelációs Mátrix



13. Figure Correlation Matrix

A korreláció egy statisztikai mutató, amely azt méri, hogy két változó milyen mértékben mozog együtt. A korrelációs mátrix egy táblázat, amely megmutatja a változók közötti lineáris kapcsolatokat, vagyis hogy milyen erős és irányú az egyik változó változása a másikkal összefüggésben.

A korrelációs mátrixban található értékek közül a leggyakrabban használt a Pearson-korreláció, amely -1 és 1 közötti értékeket vehet fel:

- Ha egy érték 1-hez közelít, az azt jelenti, hogy két változó között erős pozitív lineáris kapcsolat van: amikor az egyik változó nő, a másik is nő.
- Ha egy érték -1-hez közelít, akkor erős negatív lineáris kapcsolat áll fenn: amikor az egyik változó nő, a másik csökken.
- Ha az érték közel van 0-hoz, az azt jelenti, hogy nincs vagy csak gyenge lineáris kapcsolat a változók között.

A fenti példában a következőket láthatod a mátrixban:

- 'Age' és 'Sex' közötti korreláció 0.142803, ami egy gyenge pozitív kapcsolatot jelent.
- 'Na_to_K' és 'Drug' közötti korreláció -0.695237, ami egy erős negatív kapcsolatot jelent. Ez azt mondja nekünk, hogy minél magasabb a nátrium-kálium arány, annál kisebb az esélye annak, hogy egy adott gyógyszert felírjanak.
- Ez azt mutatja, hogy van valamiféle összefüggés a vérnyomás ('Blood Pressure') és a gyógyszerválasztás ('Drug') között. A pozitív korreláció azt sugallja, hogy általában magasabb vérnyomás esetén bizonyos típusú gyógyszerek gyakrabban előfordulhatnak, vagy fordítva

Ez a mátrix segít abban, hogy lássuk, milyen irányban és milyen erősségben követik egymást a változók, ami hasznos lehet az adatok megértése és a további elemzések tervezése szempontjából.

Döntési Fa

Adatok felosztása

```
# Splitting the dataset  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

14. Figure Adatok felosztása

Ez a kód a dataset-et (adathalmazt) két részre osztja: tanító (training) és tesztelő (testing) részekre. A `train_test_split` függvényt a Scikit-Learn könyvtár `model_selection` moduljából importáljuk. Ez a módszer hasznos azért, hogy a gépi tanulási modellek teljesítményét értékeljük.

A paraméterek rövid magyarázata:

- `X`: A független változókat tartalmazó mátrix (input features).
- `y`: A célváltozót tartalmazó vektor (target variable).
- `test_size`: A tesztelő halmaz mérete a teljes adathalmazhoz viszonyítva. Ebben az esetben a 0.3 azt jelenti, hogy a tesztelő halmaz 30% -át teszi ki az összes adatnak, és a tanító halmaz 70% -át.
- `random_state`: A véletlenszám-generátor kezdőértéke. Ezt azért használjuk, hogy a program minden futtatáskor ugyanazokat az adatokat válassza ki a tesztelő és tanító halmazokból. Ez segít reprodukálható eredmények elérésében.

A függvény visszatérési értékei négy részlet:

- `X_train`: A tanító halmaz független változói.
- `X_test`: A tesztelő halmaz független változói.
- `y_train`: A tanító halmaz célváltozói.
- `y_test`: A tesztelő halmaz célváltozói.

Ezután a `DecisionTreeClassifier` vagy más gépi tanulási algoritmusok segítségével megtanítjuk a modellt (`mod_dt`), majd teszteljük a modellt a tesztelő halmazon, és kiértékeljük a teljesítményét. A tesztelés során a modeltől függően a kimeneteket (például a predikciókat) összehasonlítjuk a valóságos célváltozókkal (tesztelő halmaz).

Tanítás

A döntési fa tanítása során a cél az, hogy a modell megtanulja az adathalmaz mintáit és azokat a címkékhez rendelje. Az alábbiakban részletezem a döntési fa tanításának lépéseit a kódban:

```
# Decision Tree
mod_dt = DecisionTreeClassifier()
mod_dt.fit(X_train, y_train)
prediction = mod_dt.predict(X_test)

# Print accuracy
print('The accuracy of the Decision Tree is', "{:.3f}".format(metrics.accuracy_score(prediction, y_test)))
```

15. Figure Tanítás

Modell létrehozása: A DecisionTreeClassifier osztályból létrehozok egy döntési fa modellt. Ez az osztály az alapvető döntési fa algoritmust implementálja.

Modell tanítása: A fit metódust használom a modell tanítására. A fit függvény két fő bemenettel rendelkezik:

- X_train: A tanító adathalmaz, amely az attribútumokat tartalmazza.
- y_train: A tanító adathalmaz címkéit tartalmazó vektor.

Ezután a modell tanulni fogja az adathalmaz mintáit és azok címkéit. A döntési fa a be- és kimeneti változók közötti összefüggéseket fogja megtanulni, és egy fastruktúrát fog kialakítani, amely az attribútumok alapján vezet az osztálycímkékhez.

A tanítás után a modell készen áll az előrejelzésekre, és a tesztadathalmazon történő alkalmazáshoz a predict metódust használhatod.

Ez a kód az előrejelzéseket hozza létre a tesztadathalmazon, és azokat a prediction változóban tárolja.

A tanítás utáni fázisban érdemes kiértékelni a modell teljesítményét, például a pontosságát, ami a következőképpen történik:

```
# Print accuracy
print('The accuracy of the Decision Tree is', "{:.3f}".format(metrics.accuracy_score(prediction, y_test)))
```

16. Figure Accury calculation

Az én esetemben a pontosság:

```
The accuracy of the Decision Tree is 0.984
```

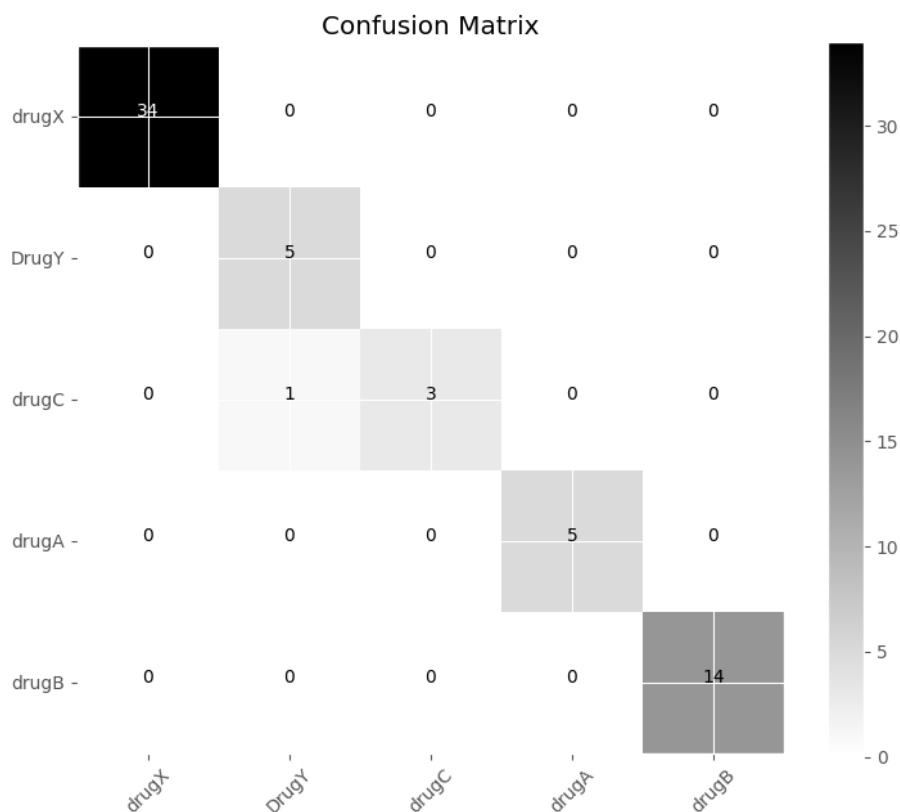
17. Figure Accuracy

További elágazások

- Az feature_2 értéke szerint további elágazások következnek.
- Például, ha feature_2 kisebb vagy egyenlő 0.50-gyel, akkor az feature_3 értékét vizsgálja.
- A különböző feltételek és a levélcsomópontok tartalmazzák az osztályokat.

Ez a fa tükrözi a gépi tanulási modell gondolkodását. A fa egyes ágai és levélcsomópontjai alapján a modell döntéseket hoz az adatpontok osztályozása során. Ezáltal könnyen értelmezhető, és a fa struktúrájának elemzésével jobban megérthető, hogyan hoz döntéseket a modell az adott jellemzők alapján.

Confusion Matrix



19. Figure Confusion Matrix

A kapott confusion matrix (zavarási mátrix) egy olyan táblázat, amely bemutatja a gépi tanulási modell teljesítményét az osztályozási feladatban. A mátrix fő átlós elemei azok az értékek, amelyek a modell által helyesen vagy helytelenül osztályozott példányok számát mutatják meg az egyes osztályokban. Az oszlopok általában a modell által tett előrejelzéseket, míg a sorok a valóságban lévő osztályokat jelentik.

Ahol a sorok a valós osztályokat, az oszlopok pedig a modell által tett előrejelzéseket jelképezik. Néhány kulcsfontosságú információ a mátrixból:

A [0, 0] elem (34): 34 példányt helyesen osztályozott a modell az első osztályba ('drugX').

A [1, 1] elem (5): 5 példányt helyesen osztályozott a modell a második osztályba ('DrugY').

A [2, 2] elem (3): 3 példányt helyesen osztályozott a modell a harmadik osztályba ('drugC').

A [3, 3] elem (5): 5 példányt helyesen osztályozott a modell a negyedik osztályba ('drugA').

A [4, 4] elem (14): 14 példányt helyesen osztályozott a modell az ötödik osztályba ('drugB').

Az átló felett és alatt található értékek (0) azt mutatják, hogy a modell nem tett helytelen előrejelzéseket azokban az osztályokban, ahol valóban nem voltak példányok.

A modell teljes pontosságát a diagonális elemek (az átló) összege adja meg az összes példányból. Az esetek egyesítése azonban további fontos információkat nyújthat a modell teljesítményéről, különösen akkor, ha az osztályok egyenetlenül vannak elosztva.

A confusion matrix harmadik sorának második eleme (2. oszlopban) a [2, 1] cella tartalma mutatja, hogy a modell egy példányt helytelenül osztályozott. Tehát a harmadik osztályban ('drugC') lévő példányok közül egyet rosszul prediktált, azt hiszi, hogy a második osztályba ('DrugY') tartozik.