Múltkori gyakorlathoz

Loop operator

- Loop operator, egy makróban tartja, hogy épp hol jár.
- Így lehet lehivatkozni: `%{macro_name}`
- String szinten helyettesíti be a makró értékét.

SSB/SSW

Itt újra kiszámoljuk a loop operátoron belül, de itt már normalizálva

Handle Exception

- Recall-lal visszahozzuk a `Remember` eredményét
- Megpróbálja (try) lefuttatni a parancsot. Ha nem megy, egy másik utasítást végez.
- Hasonló, mint a Java 'try catch'

Klaszterezés egyéb értékelési módjai

Rapidminerben vannak olyan operátorok, amelyek erre vannak kitatlálva (clustering performance...), pl külső/belső távolság, sűrűség.

Kiugró értékek kezelése

Lépések

1. Megkeresés
2. Kezelés

1. Kiugró keresési módok

1. **Vizualizáljuk**: de ez maximum 2-3 változóban működik
2. **Statisztikai** megközelítés
    1. $\bar{x} \pm 3\sigma_x$
    2. IQR módszer (Interquartilis): A sokaság tagjai alapján négy egyenlő részre osztjuk fel. $IQR = Q3 - Q1$.
        - $a < Q1 - 1.5IQR$
        - $a > Q3 + 1.5IQR$
3. **Távolság** alapú:
    - **Középponttól** való távolság: A pontoknak a ponthalmaz közepétől való távolságainak sorbarendezése alapján.
        - Ez viszont feltételezi, hogy a pontok sűrűsége gömb alakú.
        - Ha a középpont környékén vannak a nem-normális értékek (mert a széle a 'normális', pl. 'fánk' alakban), azt már nem találja meg.
    - Legközelebbi szomszédok (**K-nearest** neighbors)

$$\frac{\sum_i^n d_i}{n}$$

    - **Hierarchikus** klaszterzés segítségével
4. **Sűrűség** alapú: Minden pontra vesszük az ϵ távolságú környezetét
5. **Kiugró** pontok: A pontok egy hányadánál kevesebbszer vannak az ϵ távolságban

A legtöbb módszernek azonban nagyon nagy a számítási igénye.

A Rapidminernek vannak többé-kevésbé használható processzei ezek számítására.

2. Kiugrók kezelése

Módszerek

1. Üzletileg értelmezzük
2. Kitöröljük
3. Helyettesítjük
4. Átlag
5. Medián
6. Átlag +/- 3 szórás

Az, hogy melyiket használjuk az erősen függ a használni kívánt algoritmustól. (Például a 3

szórással szélre tolás hasznos lehet a lineáris regressziónál).

Itt fontos a különbség az olyan modellek között, amelyek tudnak extrapolálni (lineáris regresszió) és amelyek nem (döntési fa).

Modeltanítás esetében, azonban sokszor érdemes bennhagyni a kiugrókat a tanuló adathalmazban.

Adatelőkészítés

CRISP-DM módszertan

Adatelemzés lépéseinek a egy lehetséges sztenderdje. Alapvetően mi is ezt követjük.

Lehetséges problémák

1. Hiányzó értékek
2. Kiugró értékek
3. Hibás értékek
4. Skálák, normalizálás
    - A különböző változók értéktartományaira
    - Egy adott változón belüli mérékegységekre
5. Ismétlődő sorok
6. Korreláló/összefüggő változók
7. Változó típusok adott modelhez való alkalmazhatósága
8. Túl sok változó
9. Túl sok sor
10. Jelentés nélküli változók
11. Kiegyensúlyozattlan adathalmaz vagy a célváltozó

Hiányzó értékek

- Sorok törlése
- Oszlopok törlése
- Helyettesítés

    - Átlag
    - Medián
    - Módusz
    - Default érték
    - Modelalapú: A hiányzó értékeket mint célváltozót megpróbálom előrejelezni egy modellel a már meglévő értékek alapján

Viszont a helyettesítéssel torzítjuk a modelt, a törlés néha célravezetőbb.

Hibás értékek

Létező példa

Bank ügyfelek nemét 'F', 'N', 'M' kódokkal jelölte, ahol az 'F' egyszerre lehet 'Férfi' és 'Female'.

A CRISP-DM-ben az egyik lépés, hogy megvizsgáljuk, hogy az adott értékek tényleg azt mutatják-e, amit nekik gondolunk.

Ismétlődő sorok

Túldominálják a saját hatásukat. A legveszélyesebb, ha nem teljesen ismétlődik, hanem csak a célváltozó tekintetében.

Korreláló, összefüggő változók

Ki kell próbálni, és megnézni, hogy melyik működik jobban.

Változótípusok modelhez való alkalmazhatósága

- Dummizás
- Weight of evidence [Erről van fenn egy példa]

Jelentés nélküli változók

- Leginkább a célváltozóval való korrelálás alapján
  - De ez csak egy darab változóval való kapcsolatát vizsgálja.
  - A változó(csoport)-kombinációk is számíthatnak (pl a döntési fa esetében).
- Inkább egy szakértő döntse el, hogy van-e jelentése, vagy nincs.

Túl sok változó

- **Forward selection**: hozzáadjuk azt a változót, ami a legtöbbet adja a modelhez.
- **Backward selection**: elvesszük azt a változót, ami a legkevesebbet adja a modelhez.
- **Step-wise selection**: mind a két irányba dolgozunk

A első kettő mohó, túlhangsúlyozza a kezdeti értékeket.

Túl sok sor és kiegyensúlyozatlan adathalmaz

Mintavételezéssel lehet ezt esetleg megoldani.

- **Random mintavétel**: Ha egyenletes az eloszlás, akkor ezt nem zárja ki, vagy akár fel is nagyíthatja az egyensúlytalanságot.
- **Rétegzett mintavételezés** (stratified sampling)
  - Megpróbálja megőrizni az eredeti struktúrát
  - Mind a két halmazból külön-külön mintát vesz
- **Sorrend szerinti** mintavételezés (és egyáltalán nem random): ez leginkább idősoroknál hasznos.
  - A random mintavátel az időben később előforduló adatokat is beteheti a tanuló adathalmazba, aminek nincs értelme. (Például a hitelbírálatos példában az egy adott ideje már hitelt felvevő ügyfeleket vizsgáltuk.)