# Óra 2019.02.05 Gáspár Csaba 8:30tól lesz innentől, szerda 10:15 től

DAP a tárgy mozaikneve (?talán data analysis platforms?)

Követelmények, CRISP-DM Gáspár Csaba

Adatelemzési platformok a tárgy címe

Ez egy specializáció része, “adat és media informatika” mellékspeci=szakirány az msc-n belül

Adatelemzés a cél, érdekes története van hogy hogyan jött létre a kurzus

De igazából data science-t tettek bele

Ez az alapozó tárgy a 3 ból, köv félévben szövegelemzés, log elemzés; 3.félév hang és video elemzés—ez ca 6 udemy kurzus, az elmélyedéshez még kell más is, pl gyakorlati hely, labor

Mo-n most épp nem nagyon tanít ilyet más egyetem, szándékig jutottak

Bevezetés-data scientist skillset

-stat, machine learning, optimization: itt machine learning kerül elő a kurzuson

-programming, cs fundamentals: itt python kerül elő picit, algoritmus elmélet viszont nem

-visualisation, off the shelf toolboxes: itt nem sok minden lesz, mert nem találtak oktatót aki speci értene ehhez

-business and domain knowledge: ehhez inkább storyzás lesz, de hitelbírálni ebből még nem tudsz

-big data, cloud computing: Mo-n ez fut legjobban, a lényeg h hogy érd el h ne csak a saját laptopodon fusson amit előállítasz

-communication, story telling: be tud mutatni azokat az összefüggéseket a laymen audiencnek amire rájöttél, úgy hogy megértse

Btw ehhez a 6hoz egyszerre nem igazán ért senki jól

A HR találta ki, hogy ilyen ember létezzen 1 személyben. Azóta ők is találkoztak a realitással, a mai unikornis pár: 1. Story-telling, értelmezés, összefüggéskeresés- újabban ez a data scientist; 2. Big data, működtetés, infrastruktúra működtetése-ez a data engineer

Pl. hívhatod magadat AI szakembernek is akár, mert e percben everything goes and no one knows exactly what any of this is supposed to mean

Nem hisz a ppt-ben, szóval ő táblára ír inkább

Rapid miner és python lesz először

A szakirányon 23 hallgató van hivatalosan, a dm lab szereti a külsősöket az óráin, kb ugyanennyi külsős van

Informatikusok alapjaira építenek

A dm lab az oktogonon van, szóval nem a tanáriban az egyetemen

Elmélet-gyakorlat egybeöntve lesz

Sok-sok tutorial online elérhető, de

- szerinte ezt rossz egyedül tanulni, mert sok a buktató, ami frusztrál, így könnyű abbahagyni

- ő dolgoztatna minket csoportban, de nincs meg a skillset a csapatmunkához a magyar egyetemistáknál

“magyar megy usa egyetem story, a céghez azért vesznek fel amihez értesz, de azért rúgnak ki ahogyan a többiekkel viselkedsz”

A tutorial mindig egy nagyságrenddel jobban működik, mint a mindennapi adathalmaz

Mesélni fog arról, hogy mindez, hogy ágyazódik bele az üzleti világba, az üzlet-technológia határán közvetítő embereket akarnak igazából a HR-en

6db beadandó, kis házi lesz, opcionális; 30-120 perc nagyságrendben, moodle ben csinálják a hallgatóik, a külsősök még nem tudni hova tölthetik fel, ha külön kéred még ki is javítják, 0 vagy 4 pont/házi, lehet javítani és újra beküldeni

+1 nagy házi Kaggle.com ők adatbányászati versenyeket írnak ki, ez a data science történelmi eredete is, 2 éve megvette a google, itt lesz kiírva egy verseny; pl. előre jelezni h ki fog bejárni az utolsó óráig az első x óra megjelenése/gépelési adatai alapján

fura az immediate feedback, mert az adathalmaz 30%án teszteli a megoldást, de a végén 100%ra adják a jegyet és a két adathalmaz ugye másképp viselkedhet…. Deadline utolsó hét vasárnap dél

Baseline megoldás: amit ő összedob 1óra alatt, az lesz a legjobb megoldás amit vki beküld (50 pont), a kettő közt féluton lesz 10 pont, exponenciálisan nő utána, 10 és 50 pont közt félúton indul a 20 pont, aztán 20 és 50 pont közt félúton indul a 30—ez csak a neptunkódos hallgatóknak verseny így, ők neptun kóddal mennek ide

ZH, saját géppel, elméletből, de nem tudják elolvasni a hallgatók kézírását, szóval google form, esszékérdések, 50 pont

Év végén 40pont től 2es

Egy google doc a tárgy info honlapja és lesz slack, kooperáció oké, másolás nem

Intro vége itt

Mi volt a data science előtt?

Data mining, üzleti intelligencia (business intelligence)-hírszerzés lenne, de arra nem akarták fordítani a 90-es évek Mo-ján, ennek érdekes unintended következményei lettek aztán

Adatelemzés 8 lépcsője (by SAS)

1. Standard reportot akarsz az adatodból pl. neptun kiírja az átlagokat vagy távközlési jóslás; köv 1-3 hóban hány ügyfele fog szolgáltatót váltani=churn 🡪de ha kész van, már nem ezt akarod látni, több kérdésed lesz tőle
2. Ad-hoc reportokat akarsz: hány kártyás előfizető, hány céges előfizető ment el (2019 ben már ez szokott lenni), önkiszolgáló BI (business intelligence)
3. Amit kitalálsz lekérdezést, az egyből jelenjen meg online (ez régen volt téma)
4. Mi lenne, ha értesítést kapnál a furcsa eseményről (alerting) pl google alerts
5. Meglévő adatok vizualizációja (erről itt nem n lesz szó)
6. Statisztikai elemzés, hipotézis vizsgálat (ezt ma kevesen csinálják, sok az adat és nem kell p<0.05 ahhoz h lásd, vagy nem bírja el a stat szoftver az elemzést ekkora adattömbökkel)
7. Forecasting (controllinghoz kapcsolódóan), ehhez kell a szakértő keze is, nem lehet teljesen automatizálni még
8. Prediktív analitika, erről lesz itt a legtöbbször szó: rövid távon belülre elemi előrejelzés, nem az érdekel h összesen hány ügyfél lesz, hanem minden egyes ügyfélre előrejelezni h ő ügyfél lesz-e 6 hónap múlva

A data science jelentős része itt van

+1 gép felismerés, itt is elemi események vannak pl van-e macska a képen, itt más metrikával értékelnek, noha a műveletek alapja hasonló

+2 optimalizáció: ha már nagyon ismered a működését, paraméter térben keresed a legjobb megoldást, nagy számítási kapacitás és kevés helyen fordul elő, pl minden eszköz a cégen belül van

Egy ember előrejelezhetősége: nagy tömegben elég jól becsülhető—de visszacsatol a rendszer, nem lehet elérni a 100%ot; ez az elv picit megrogyott az elmúlt években, mert annyi adat van… hogy etikai oldala is van, hogy mi jöhet ki abból amit te számolsz, pl. ki milyen valószínűség szerint vándorol el story: target.com; terhes magazint kapott a család mielőtt tudta volna a családfő hogy terhes a 13 éves lánya—honnan látták (akár hamarabb mint a lány), termékváltás pl színtelen-szagtalan piperére vált x hónappal előtte; de ha a magazin költsége kicsi… 6 % esélyre már kimehet az újság) ez itt rendszer szintű probléma, mert sehol nem volt 1 ember akinek ez a felelőssége.

Néha el kell gondolkozni, hogy „de miért akarjátok ezt tudni”; nekik a lista akivel nem dolgozik a dmlab: dohánycég, a másik vonal hive mind döntés pl no go volt az h mi a felső korhatár ahol gazdaságilag megéri szürkehályog műtétet finanszírozni

A tapasztalat az, hogy ezeket a lépcsőket alulról lehet mászni, amíg nem megy az 1. nem érdemes 6. szinttel próbálkozni annyira

Story az utazási irodáról, akiknek nem machine learning kellett hanem csak az a diagram hogy a hírlevélből jövő ügyfél jobb

Techincal announcement

Rapid miner kell, akadémiai licenc-szel; python 1.órát kihagyja, online tutorial-al készülsz fel

# Óra 2019/02/12 Nagy Rácz István

Register on kaggle

A dap2019 slack csatorna closed, szóval csak a tanár meghívásával tudsz bekerülni

Ma már lesz kisházi, ki van írva a honlapon, jövő kedd délig várják a megoldásokat

A kaggle-s feladaton a csapatnév a konkrét versenyhez az, aminél megkötés lesz a névkonvenció, addig mindegy

INTRO

Miért pont ezt a két eszközt tanítják? Az első gyakorlatok is rapid minerben lesznek már

A, enterprise eszközök: szoftverek, szolgáltatások, amiket a cégek pénzért vesznek, itt van support meg minden a pénzért

Gartner figure kdnuggets.com

Rapidminer egy leader e percben a ds (enterprise)software-k között, ingyenesen elérhető, hozzáférhető (academic licence), a 3. ok pedig hogy az előadóknak van személyes kötődése is a céghez, a magyar fejlesztői teammel közös irodájuk van 😊

B, ingyenes eszközök

Python vs r; a python tör fel most az r rovására épp 2016-2018 táján

Az ő véleményük az h az R nem is annyira jó :D web kdnuggets (12% használ both)

Szóval nagy a python community és a python machine learning sokkal nagyobbat fut a trend keresőszavak alapján

Kaggle-n is elhúzott a python a kernelek között (kernel itt a megosztott példa megoldás a felületükön)

Githubon a javascript után már a python jön (java előtt)- ez minden fejlesztő, nem csak ds már—mert ez egy általános célú programozási nyelv—az R itt vérzik el, az egy speci adatelmezésre létrehozott nyelv, így sokkal nehezebb integrálni amit ott hozol létre

És stackoverflow-n is ő a leggyakoribb hit az elmút években

Most elindulunk egy kaggle versenyen és beadunk 1 megoldást rapidmineren

Célja, hogy átrugdosson minket az első pár inflexiós ponton a tanulásban

A knowledge versenyek a kaggle-n az ami kb tutorial, a tanulás a célja

Jön a titanic dataset…—de nem ő csak azért is a house prices-t mutatja be

Data description- a variables fül gyakorlatilag, a változók tulajdonságai, hogy néz ki az adathalmaz

Az ingatlannak van ára, ezt kell modellezni az ingatlan tulajdonságai alapján

A modell kialakításához a train.csv-t kell használni

Rapid miner-gui – data flow szemlélet=rapid miner process az, amit az operátorokból építesz

* Operátor, ami adatelemzési alapműveletet hajt vége, kijön belőle egy másik adathalmaz

Ki kell találni h milyen operátorokat milyen sorrendben akarsz

Adatbeolvasás-operátor tab, drag and drop a kívánt operátor a vászonra

Ettől a parameters tab nézet megváltozik, itt tudod beállítani az operátor tulajdonságait

Import configuration wizard (de van show advanced parameters is)

A változó típusa (numerikus vs nominális=kategória változó) és szerepe

Az irányítószám pl. nem numerikus, mert nem alkalmazható rá semmilyen aritmetikai művelet, az nem elég h számként lehet ábrázolni

Kategorikus pl. város, fűtés típus

A numerikus változónak „végtelen” lehetséges értéke, a kategorikus változónál előre megadott fix értékkészlet van

Műveletek, amit lehet kategorikus változóval csinálni: súlyozni, egyenlőséget vizsgálni

Rendezés—pl átlagos ár a városban, méret szerint, abc—nem minden nominálist tudsz (sorba)rendezni

Rapidminer takes an educated guess regarding the type of the variable (de csak az első 100 sor alapján)

Valóságban egyesével végig mész a változókon, hogy ellenőrizd a tippjeit (pl irányítószámot benézné számnak)

Milyen szerepe lehet a változóknak?

Eredmény=cél-ezt becsülöd a modelleddel

magyarázó=bemeneti-ezt használod fel a modelben

id változó

Rapidmineren a label a cél változó neve

Operátornak van portja, ezzel kötöd össze őket, ahhoz h megkapd az eredményt, az utolsó operátor portját ki kell kötni a result pontra

Polynom változótípus- itt max egyezőséget vizsgálsz (quick fix, mert idő hiányában kihagyjuk az adattakarítás lépést a példában)

Statistcis- ránéhetsz pl a változók értékkészletére

Visualisations-adatok grafikus megjelenítése, csak itt

Adabányászati modell, algorimtus- pl döntési fa, neurális modell, knn(k-nearest neighbour), base, lineáris regresszió

Lin reg a legegyszerűbb algoritmus amit használunk, nem tud nominális változókkal mit kezdeni, vagy kihagyod vagy kódolod őket vhogy

Rapidminerben az oszlop neve attribútum (select attributes)

-attribute filter type-value type ez szűr típus alapján

-value type ban adod meg h mit tartasz meg (numeric)

Coefficiens oszlop a lin regben—ez a becslésnél a num változóhoz rendelt súly

Az együtthatót úgy találja ki h az így képzett súlyozott összeg minél pontosabban becsülje a válaszváltozót

Most jön be a test csv, külön operátorban, a struktúrája megegyezik (a tartalma más, más adatpontok itt), de itt nincs ár—megbecsülöd, ezt küldöd be, erre jelez vissza a kaggle h mennyire jó a tipp, ez a model kimenete

Apply model operátor; unl port az unlabel, az az adathalmaz amin futtatod

Itt elvesztettem hogy mit kötsz hova most gyorsan, hibák vannak az NaN-k miatt rengeteg oszlopban

Nem jó ha a train és a test adat túlságosan különbözik egymástól

Lett vmi labeled datasetem a végén, a lemaradt házi az az, hogy lakás id és predicted price-t kimented write csv re és submit to kaggle

# Óra 2019/02/13 Nagy Rácz István

Crisp-dm: egy adatelemzési módszertan, a projekt, főképp prediktív analitikai projektre ad keretet, hogy miképp hajtsd végre

**Cr**oss *I*ndustry **S**tandard **P**rocess for data mining: szóval alkalmazási területtől független, általános projetkmódszertan, gondolati keretet ad

6 fő fázis

1, business understanding: meg kell érteni az üzleti célokat, az üzleti problémát h ezt át lehessen fordítani adatelemzési céllá, problémává; definiált az adatelemzési célok üzleti sikerkritériumát-ezt átfordítani matematikai visszamérési függvénnyé

2, data understanding: feltárni azokat az adathalmazokat, amiket fel lehet használni a problémához, minőség, h lehet összekapcsolni őket, hogy frissülnek, hol vannak

3, data preparation: a cél h létrejöjjön az a táblázatos adathalmaz, elemzési adattábla ami tartalmazza a célváltozót és az összes olyan bemeneti változót, amit a célváltozó szempontjából fontosnak gondolsz, a bemeneti változók ált csak egy kiindulási halmaz, ebből szelektálsz, származtatott plusz változót hozol létre

4, modelling: modellt építesz és ellenőrzöd, h matematikailag a modell mennyire oké

5, evaluation: kiértékelés fázisa, NEM a modellek matematikai kiértékelése, hanem az üzleti kiértékelést tartalmazza, mennyire teljesíti a kitűzött üzleti kritériumokat

6, deployment: ha van már egy üzletileg kiértékelt megoldásod akkor h fogod ezt felhasználni az adott üzleti környezetben, a gyakorlatban

A munka 1-3 fázisra megy 80%a a maradék 20% a vége

A módszertan iteratív, tehát ennél körkörösebb, lehet h vissza kell ugrani később mégis

Mai példa a gyakorlaton 🡪 alkalmazási terület a hitelbírálat, classic adatbányászat felhasználási terület

Konkrét példa-credit scoring, már az 50es években is mentek ebbe az irányba

Olyan tanuló adattömb ahol megvan adva az is h mennyire sikeresen törlesztett, tehát múltbéli adatok, ez lehet binary is

Tehát a modell múltbéli összefüggéseken alapul (és reménykedsz, hogy azóta ezek az összefüggések nem változtak, továbbra is érvényesek)

Credit scorecard-a modell speci megjelenési formája, ez alapján zajlik az ügyfelek hitelminősítése—táblázat, benne tulajdonságok, numerikus változók értékkészlete intervallumokra osztva, minden kategóriához pontszám—adott ügyfél, aki ezzel a tulajdonsággal rendelkezik—pontozzák a táblázat alapján h mennyire jó adós

Az a modell kell, ami nagyobb arányban találja el? —vagy amelyik nagyobb profitot hozott? , a modell 1 soronként dolgozik

Az ület nem találati arányt fog mérni, hanem várható profitgörbét fog rajzolni---ez alapján állapítják majd meg a tresholdot

Visszafizetési valszín alapján sorbarendezed az ügyfeleket-vágási érték pl 0,9—megnézed h valós adatokon mi történt—ki tudod számolni a profitot, bukást pontosan –kiszámolod 0.9-re, 0.8-ra stb (- akik nem fizették vissza, +akik visszafizették ténylegesen is)

Nem pontosság kell itt, hanem a sorrendezésben kell h jó legyen a modelled

Mi legyen a célváltozó?

Elmarad a törlesztőrészlettel; a bedőlés azt jelenti náluk ált h 90 nap fölötti elmaradás volt bármikor—ő default os, azaz rossz adós-ezt kell becsülniük a bankoknak

A célváltozó a legritkább esetben olyan, ami természetesen előfordul az eredeti adattömbben

A credit scoringnak 2 vállfaja van

1. application scoring: a bekért adatok a nyomtatványon, kereset, kor stb.
2. behavioural scoring: már ügyfél egyéb terméken, pl rálátsz a folyószámlájára, hitelkártyájára

a behav data-t még gyúrni kell h beférjen a táblázatos adatokba (mert ott 1 tranzakció 1 sor, nem 1 ügyfél)—ez egy veszteséges adattömörítés ha a tranzakciósból csinálsz ügyfél szintű táblázatot

lehet olyan modellt is ami a tranzakció adatokat is bemenetiként kezeli, de akkor ez már másik irány, ha nem aggregálsz akkor búcsút mondhatsz a táblázatos adat modellnek

innen data prep

hogy fogod kiválogatni azt az ügyfélhalmazt amin tréningezed a modelled? (hozd létre az elemzési adattáblát)

lesz egy szűrés az ügyfelek között—csak a lezárt hitelesek? Vagy a még nyitottak közül is (ezek frissebbek) de kell olyan is aki nem fizetett vissza, vagy összes 3 hónaposnál régebben kezdett hitel?

best practice

megnézi az összes 1 hónapos hitelt, ebből megnézi h mennyi a defaultosok aránya ((0%) ahogy egyre öregebb a hitel úgy egyre növekszik a defaultosok aránya—telítődési görbét vársz—azt keresed, h hol van ahol már stabilabbá válik és az efölötti, de még élő hiteleket fogja beválogatni

itt open thread remained!!!!

Modellezés

-ha nominális akkor v eltalálja v nem (itt csak összehasonlítani tudsz)—ez az osztályozási modell;

-ha numerikus akkor kis hibával közelíti az értéket-ez a regressziós modell, itt nincsenek határok, bármilyen valós számot tippelhet válaszként

---a banki modell egy osztályozási probléma, mert ezek nem csak a kategóriát adják meg hanem a valószínűséget is a tipphez

# 4. óra 2019/02/19

Credits scoring folytatása

Üzleti cél: hitelképesség eldöntése kérelmezéskor

Üzleti értékeléskor: cél a profitgörbe felrajzolása

Célváltozó: 90 napon túli késedelembe esett-e az ügyfél

Eddig tartott a business understanding rész

Innen most data understanding (application vs behav scoring)

Aztán data prep amit mi itt ugrottunk, de igazából azt is kell csinálnod

Majd modelling: osztályozás vs regresszió: mindkettő felügyelt tanulási módszer, azaz a tréning set-nél mind a bemeneti, mind a cél változók értéke ismert; nem minden algoritmus alkalmas mindkét féle változó kezelésére; más a modell sikerességének kiértékelési módszere is (találati arány vs minél kisebb hibával jelezzen előre)

Vissza a modellezéshez:

Először regresszióval próbálkozunk, azon belül is egyváltozós regresszió

-Egy fv szerű determinációt keres, ami korreláción alapul

a görbe pontoktól vett távolsága legyen minimális y=w0+w1\*x1, az együtthatókat keresi, a tanulási adathalmazban minimalizálja a hibát, pontosabban a hibanégyzet átlagát minimalizálja—ettől ez a típusús modell érzékenyebb a kiugró értékekre

Többváltozós lineáris regresszió: bemeneti változók lineáris kombinációjaként; eltolás és súlyok

Pozitív együttható-> gerjesztő hatása van a célváltozóra, míg a negatívnak gátló

A lineáris reg-re hat, hogy mi a bemeneti változó értékkészlete, ettől a koefficiens nem = a súllyal!, mert lehet h x1 értékkészlete 0-1 míg x2 10-100

Normalizált változókkal dolgozni: minden változó értékét transzformálja 0 és 1 közé---így a kapott együtthatók már kezelhetőek súlyokként

Bináris célváltozó

Sorba rendezéses modell🡪mekkora a valószínűsége az 1-nek

Logit transzformáció

Ez volt a logisztikus regresszió, a p az valszín p érték

A problémái, korlátai ennek a modelltípusnak:

Nominális bemeneti változók? Lehetőségeid:

Kihagyod

ha fix az értékkészlete újrakódolod számokká, lehetőleg ne 0-val

—de ha nem ordinális akkor ezt ne akard csinálni, az ordinálisnál se akard ezt ha van más ötleted, mert az arányok ott se stimmelnek majd jó eséllyel

Dummy változó képzés: annyi változót képzel a bemeneti változóból ahány értékel felvehet (mínusz 1, ha zavar a redundancia) és 0-1 be kódolod le

|  |  |  |  |
| --- | --- | --- | --- |
| Eredeti változó | A1 | A2 | A3 |
| \* | 1 | 0 | 0 |
| # | 0 | 1 | 0 |
| @ | 0 | 0 | 1 |

Itt is van „probléma”, mert x-1 változóból már képezhető az x-ik dummy változó, redundáns az info tárolás, elég lenne itt pl 2 változó, a 3 változó nem lesz független—tehát érdemes csökkenteni ezeket az összefüggéseket és (értékkészlet-1) db dummy változót hozol létre

Lin reg és log reg is független bemeneti változókat feltételez

Másik megoldás: az adott nominális változó értéket cseréld le a hozzátartozó célváltozók átlagával a tanuló szetten (pl északra néző lakások); bináris bemenetnél pl 1 gyakorisági eloszlását (1esek aránya)

Weight of evidence (WoE) ~ az előbbihez (ez egy hiányos táblázat lenne, ne zavarjon ha nem jön ki 😊

Age, csoportosítja: missing,18-22,23-26,27-29,30-35,36-44,44+

Count 1000,4000,6000,10000,10000,3000

GoodS -,3040, sum36160

BadS -, 960, sum3840

WoE -, ln ((3040/36160)/(960/3840))=-1.09

WoE=ln(dist good/dist bad), ha a WoE érték mindenhol, minden értéknél ugyanannyi (0) akkor elkedvetlenedsz; az a változó jó ahol ezek az értékek nagyon különbözőek (-∞ és +∞ között lehet), ha az érték negatív akkor ebben a példában a részhalmazban a rossz adósok aránya nbb mint a sokaság egészében

Azt akarod h a különböző kategóriák WoE értéke eltérjen

Mennyire magyarázza célváltozó értékét???

Ez neki ahhoz kell, hogy ezzel alakítsa ki az elvárt scorecard formát

Dist alatt az eloszlást érti

Az adathalmazban 9:1 a jó és rossz adós arány,

A score egyik kiszámítási módja: Veszi a változóhoz tartozó együtthatót a regressziós modellből és ezt megszorozza az adott kategóriához tartozó WoE értékkel—voilá a score!

Numerikus változóknak intervallumokat képez ezt megcsinálja nominálisra is, ott is WoE

A WoE táblázaton épít egy regr modellt ebből lesz neki Wi—azért WoE-n futtat modellt mert ezekkel a paraméterekkel nem lehet lin modellt ha előtte nem szereled meg a változót, mert intervall meg nomin az eredeti

🡪ez előtt érdemes még

Vedd ki a nem értelmes, hasznos változót; mert a lin és log regr minden bemeneti változóhoz fog együtthatót számolni, ahhoz is ami nem tartalmaz összefüggést a célváltozóval, ez zajként jelenik meg az előrejelzésében és csökkenti a többi, valóban fontos változó erejét

A log transzformáció felnagyítja a kis hatást/összefüggést, de amíg a nem hasznosak kiszűrésére használod és nem az összefüggés erősségét viszonyítod egymáshoz addig oké

# 5. óra 2019/02/26 rapidminer gyak

Credit scoring modelt építünk, WoE számítás rapidminerrel

Change column separator if needed

Set status=célváltozó, ez lesz a label

Nem kötjük be az első portot, mert akkor összeomlik gyakran

Within parameteres dataset, vmi metadata, edit list gomb parameterrs set var20 from integer to binom-itt átállítottuk egy változó típusát

Logistic regression wants numeric, not nominal

Select attributes a szűrés, filter type value numeric változókra szűrsz így

Outputban coefficient a W

Intercept—a W0 eltolás a képletben

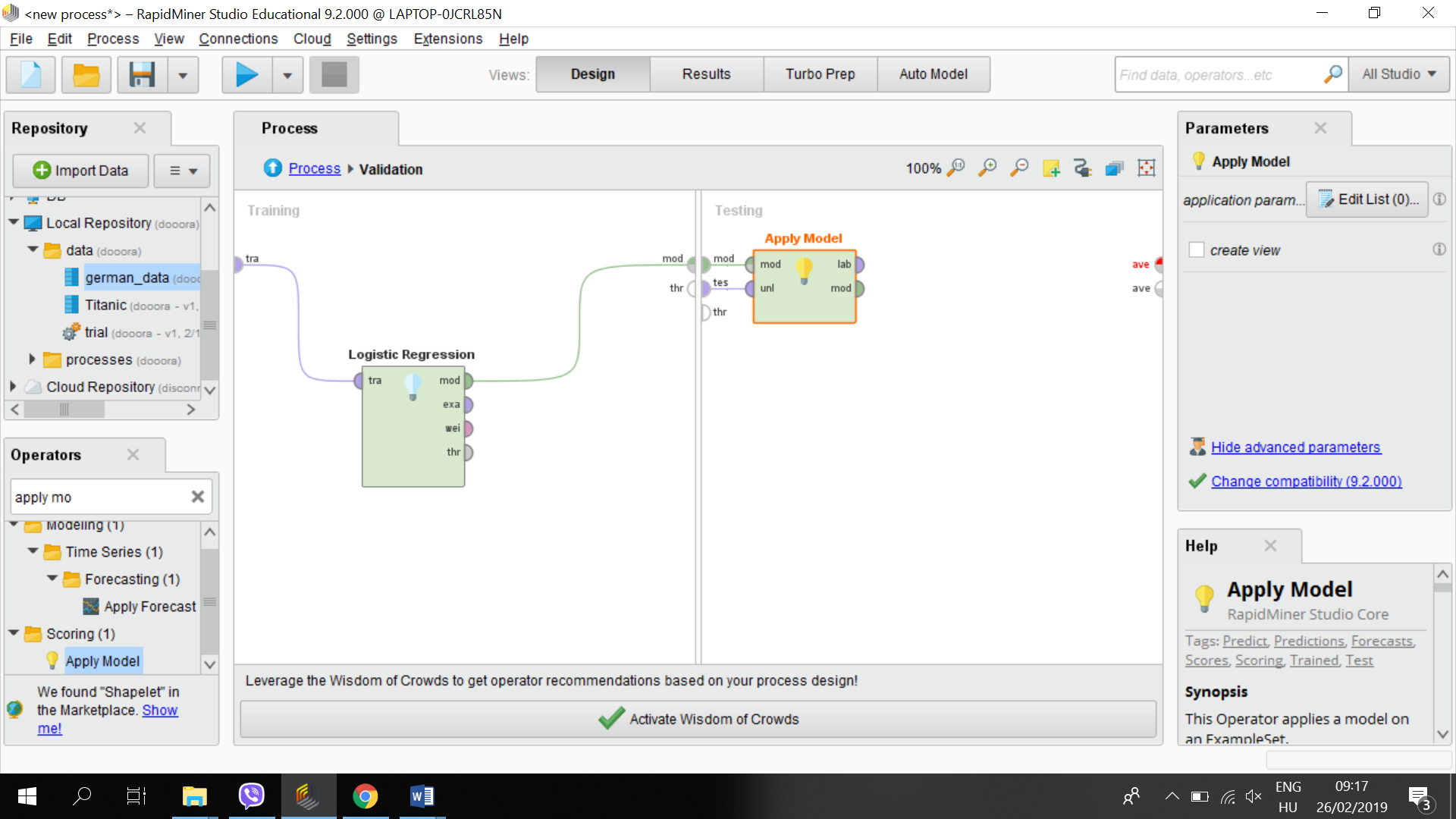
Nézzük meg h mennyire jó a modell pl cross validálás

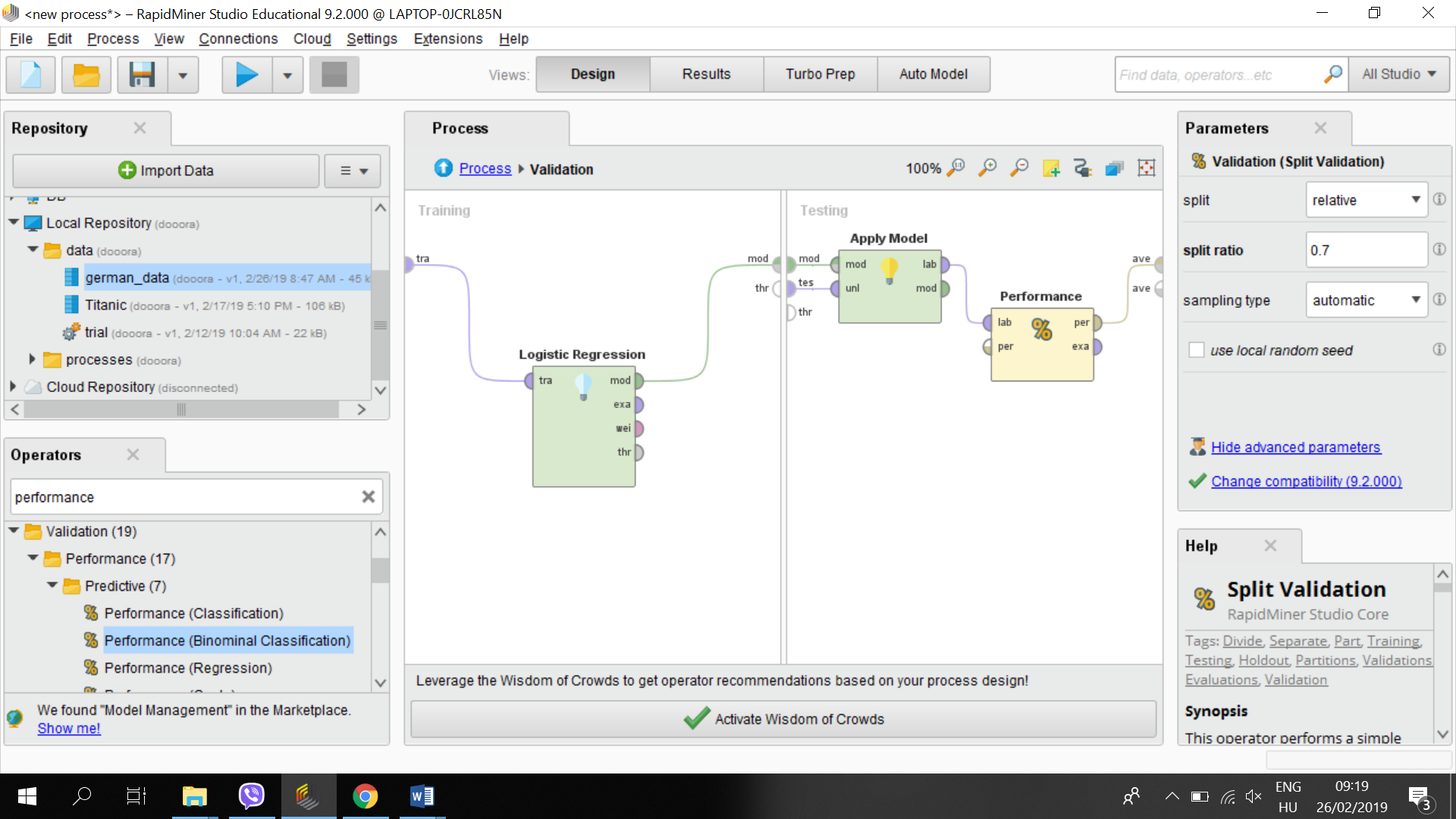
Megnézni h mennyit hibázik—de ehhez split validation kell csinálnod, azaz adattömbböl training és test setre leosztani

Míg a cross validation az adathalmazt felvágja mondjuk 4 részre, beletesz 4 db tanulási iterációt 1.első negyed a teszt halmaz—épít modellt, van hatékonyság, 2. iterációban 2. adatnegyedddel uezt stb.🡪 végül lesz 4 modell, 4 hatékonysággal ez kicsit jobb mint a split validation mert az adathalmaz minden részét felhasználja a tanuláshoz. A 4 hatékonyságnak ideális esetben közel azonos értékűnek kell lennie—nem mindegy hogyan vágod szét pl. ha túl kevés data point, v sorba volt rendezve—az baj és ezt ki kell akkor javítani

Sorrendezéssel akarod ellenőrizni, hogy milyen jó a modell

Split validation operátor, ennek van belseje, dupla klikk

performance binomial classification az az operátor, ami teszteli a modell teljesítményét

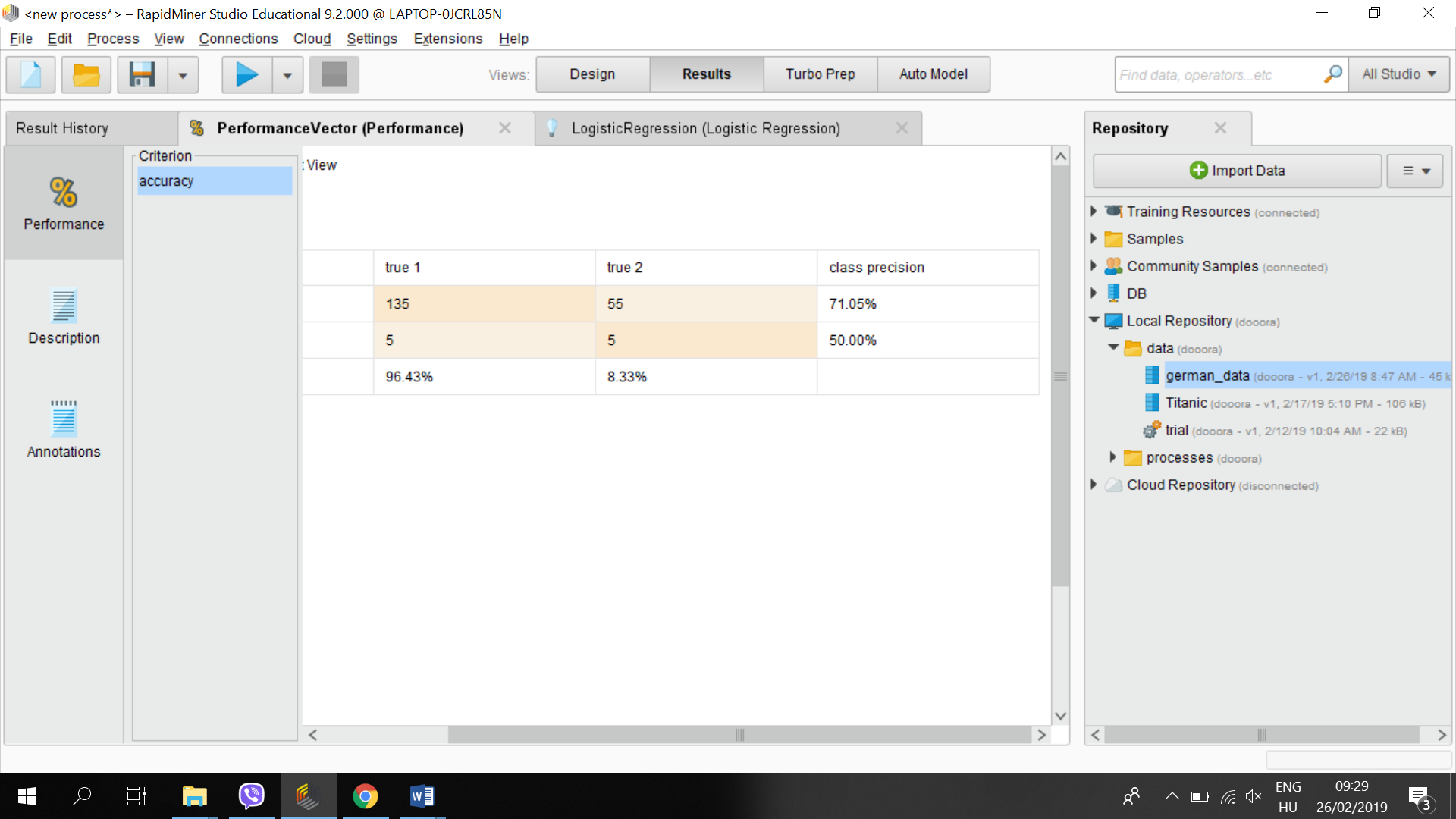
relative split -aránnyal dolgozik, van abszolút is ami sor db számra megy

Sampling type- random (shuffled) az automata, de lehet lineárist, meg stratified (rétegzett) is—ez arra jó, ha adott változó eloszlását akarod megőrizni mindkét szettben

Use local random seed-azért kell h mindenki ugyanazt kapja közülünk—mert mindenkinél ugyanúgy vágja szét training és tesztre

Average portot is ki kell kötni

Ha a modell mindenkire azt tippeli h jó adós—akkor 90%os lesz 10%rossz adós esetén… de ez nem az amit te keresel megoldásként mégse

 ez így még gyenge teljesítmény

Hogyan lehet felokosítani? Be kell adni nominálisokat, woe t kell számolni

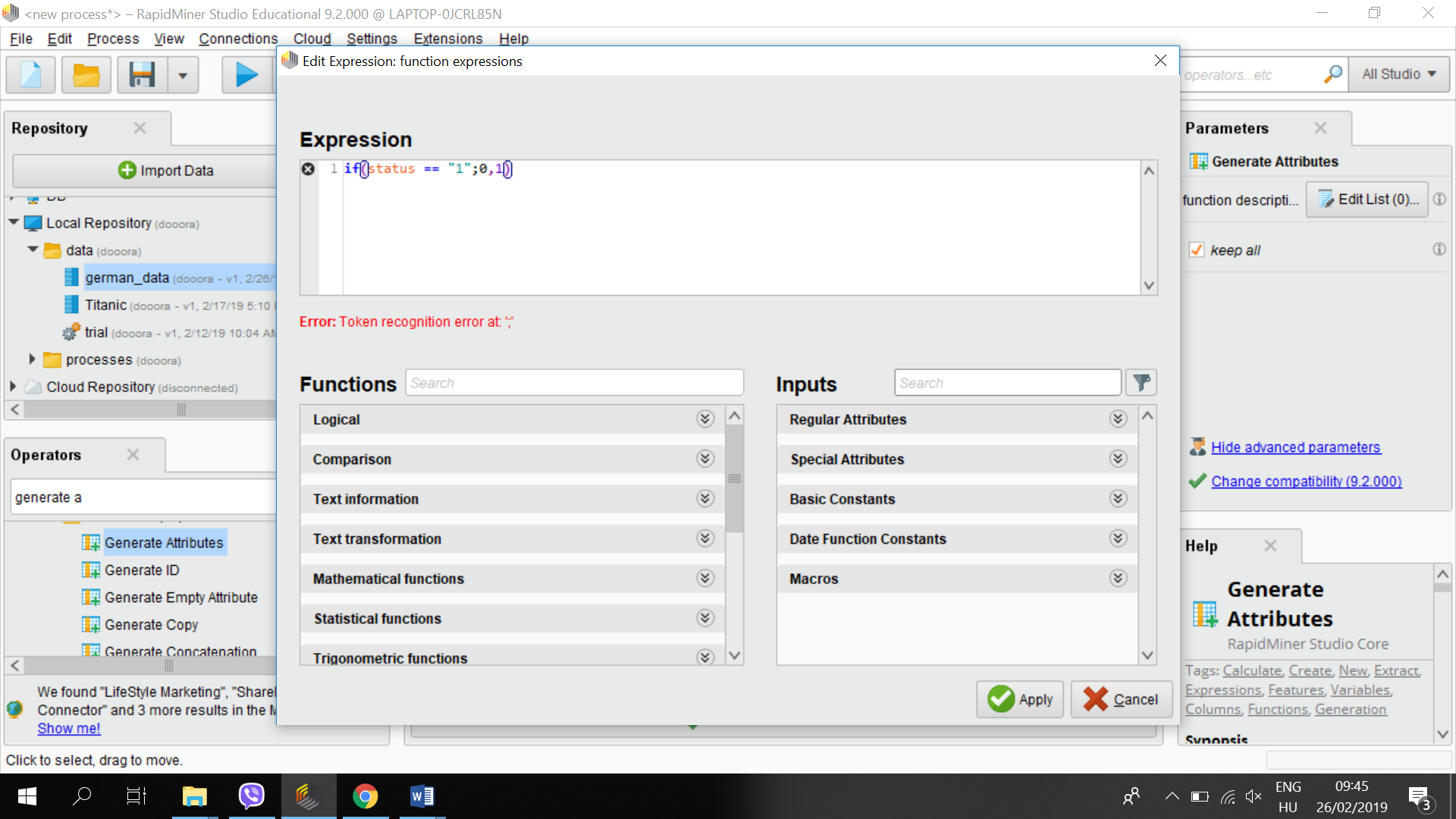
WoE-t kell számolni

WoE=ln(no of good adós/all\_good)/(no of bad/all\_bad)—ezt minden kategóriára külön kiszámolni

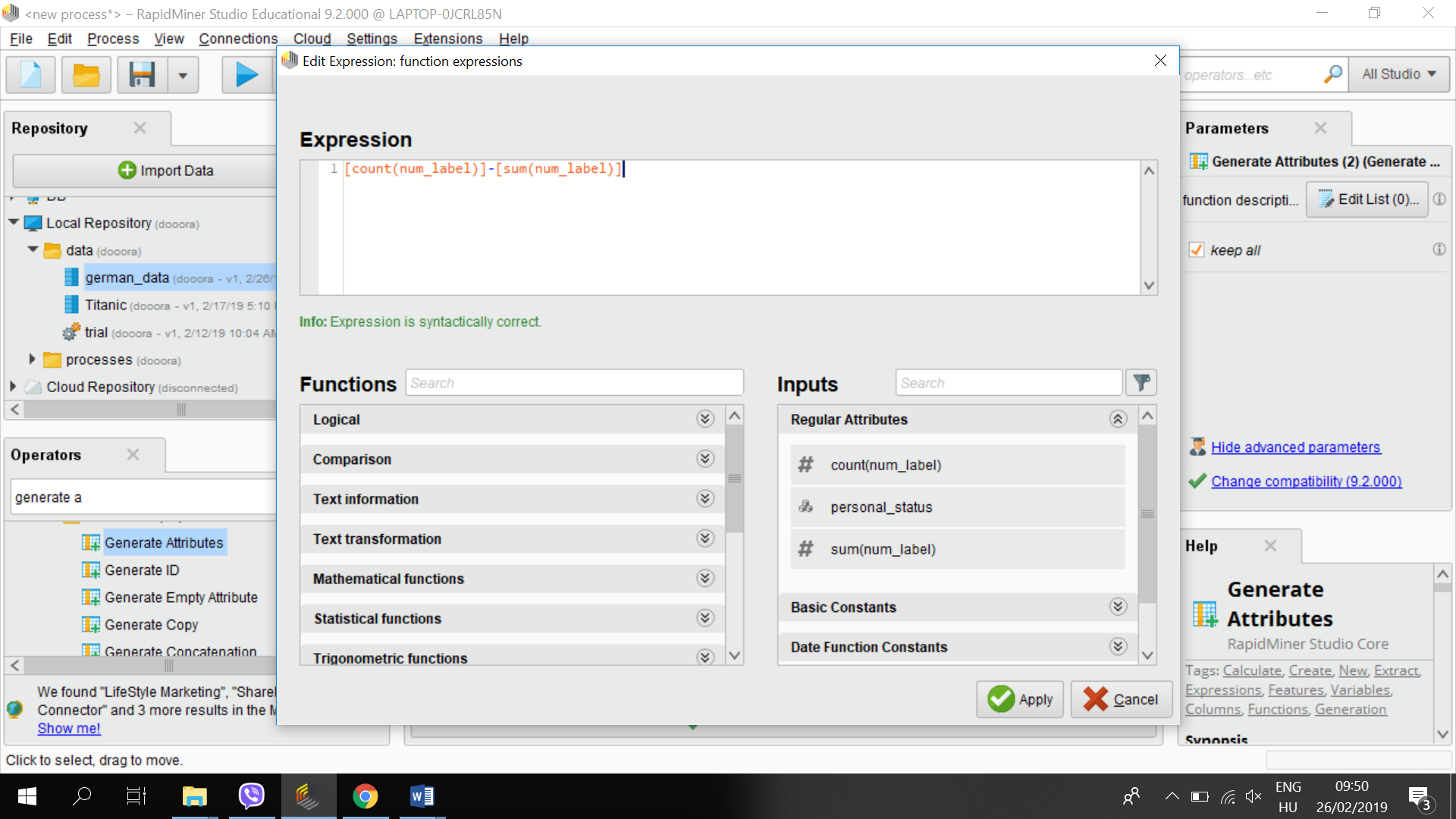
Subprocess process, h átlátható maradjon a rendszer—ez egy sub tab végülis vizuálisan

Aggregate~ mint a sql groupby

Binomiális változót nem tudsz összeadni, csinálunk egy numerikus célváltozót



Stringként értékeli az 1 et, ezt meg kell oldani itt



A generate attributes csak 1 adathalmazon belül tud dolgozni, szal össze kell illeszteni a két adathalmazt

# 6.óra 2019/02/27

A kisházi ennek a gyakorlatnak a folytatása

Sum adta meg a rosszak számát

A count adta meg az összes entry számát

R click show example set results

Stg is going on w my process, need to correct it based on updated version (vmiért kb megnégyszereztem az elemszámot???)

100% accuracy lett a modell, ez mindig gyanús🡪 a num label a bűnös, ez a log regr specialitása,h rátanult erre

Mit kezdjünk azokkal a változókkal, ahol 0 volt a coefficiens

Az algoritmus nem tudja kiszedni h melyik bemeneti változó értelmes, pl ha horoszkópot adod bele—a log reg ehhez is képez súlyt, emiatt érdemes kiválogatni a releváns változókat (a célváltozó szempontjából)🡪Ez a feature selection

Nem normalizált változóknál az együttható nem = a súllyal

A p value alapján is válogathatsz

Forward selection, egy változós log regt épít, kiválasztja azt az egyet, ami leghatékonyabb és ezt építi felfele egyesével, addig amíg a hatékonyság ettől nő

A backwardnál meg egyesével veszed ki azt, ami a legkevésbé rontja/még javítja

A stepwise egy lépésben vizsgálja h hozzáadsz meg h ki is veszel

(Moho algortimusnál az aktuálisan legoptimálisabb lépést teszti meg, de ez nem biztos h globálisan is az optimális lesz)

A lin reg nem reagál jól arra, ha a bemeneti változók korrelálnak és ideális esetben nem engedsz összefüggő bementei változókat

Rapidminerben van forward selection operátor

Ha kiveszed a count változókat a modellből akkor már 71%ra megy a modell le, szóval segített rajta

Kisházi válassz 1 nominális változót, amiből dummy változót készít pl personal statust és megnézni mit tesz ez a modell hatékonyságával, melyik a jobb?

Aztán válassz egy numerikust, abból csinálj nominálist és nézd meg h ez hogy segít többet, dummyként v woe-ként

A scorecardot kell leképezni még, de csak 1 változóra

A WoE számolást a teljes adathalmazra csináltuk—és csak utána szedtük szét 2 részre

WoE on training and test set should be very similar so this is the best practice, not a mistake, handle training and test together at this step—a kisháziban is!!

Normalize operátor—ha ezt a preprocessing-et alkalmazod akkor nem vesz ki infot a teszt – adathalmazból, cserébe a normalizált változó lehet 1nél nbb v 0nál kisebb a teszt halmazban! nagyházinál ezt érdemes figyelembe venni!!!

Nem minden modell képes extrapolálni (egyáltalán v jól)

ADATELMEZÉSI ALAPPROBLÉMÁK

A, felügyelt tanulási módszerek: olyan adathalmazokon, ahol mind a bemeneti, mind a célváltozó ismert és e kettő közötti kapcsolatot próbáljuk megtalálni, 2 aleset, 2 megoldást össze tudsz hasonlítani sikeresség szempontjából

a.1 regresszió: célváltozó numerikus, pl algoritmusra lin reg

a.2 osztályozás/calssification: célváltozó nominális, pl log regr

B, nem felügyelt tanulás

b.1 clusterezés: adathalmazban olyan csoportokat képezni, ahol a bementei változók szempontjából intragroup n hasonló és intergroup pedig n különböznek egymástól, az osztályozással ellentétben itt nincsenek labelek, nincs meghatározva h a célváltozó milyen /hány db értéket vehet fel, eredményességét nem tudod ugyanígy mérni, mert nem tudod a helyes választ

a1 és a2 bizonyos algoritmusok esetén feltételez ilyen intragroup hasonlóságot, de pl. a döntési fa nem ilyen, szóval nem feltétlenül!!!

**Clusterezési algoritmusok csoportosítása**

b1a particionáló algoritmusok: kezdeti feltételezést betáplálod, hány clustert sejtesz, ez alapján keres megadott db számú klasztert (jó kérdés h honnan tudhatnád…)

K-means: átmentem onenote ba kifejteni ezt ott

# 7.óra 2019.03.05

**K means tulajdonságai**

Ez egy távolság fv—numerikus változókra működik---ha nem az akkor dummy változót kell képezni h mehessen

Pl euklideszi távolság—koordináták különbségének négyzetösszegének a gyöke

Másik példa a manhattan távolság: különböző dimenziók mentén vett távolság összege

Outlierek kezelése---miért? 1 kiugró pont akár külön klaszter is maradhat, ha olyan a szerencséd és 1.iterációban ő középpontba kerül

Először meg kell találnod őket, aztán csinálni vele valamit pl. kidobod, de osztályozásnál ez nem annyira üdvös stratégia, főleg, ha az valós eset volt

Változók skálája~értékkészlet intervallum nagysága—ha más skálán vannak akkor a távolságszámítást hazavágja, nincs súlyozás pl eukledeszi távolságban—ezt reflektálva a klaszterek a domináns változó alapján fognak csoportosítani azt kész

Erre válasz lehet a standardizálás (x-mean(x))/SDx szórás—ez -végtelentől + végtelenig megy—de a várható érték 0 a szórás meg 1 lesz ezután; ha normál eloszlás volt akkor -3;3 között lesz az értékeke kilencvensok %a

Másik lehetséges válasz a normalizálás—

(x-min(x))/(max(x)-min(x))—értékkészlet 0-1

Milyen alakú klasztereket talál az algoritmus? Gömb jellegűeket—mert klaszter középponttól vett távolság alapján sorolod be őket

És mi van a sűrűséggel? Ahol több pont van?

Pl ha log veszed, hogy a kiugró nagy távolság ne vegyen akkor súllyal részt

Másik algoritmus a k medoid—ő k darab pontot felvesz, besorolja clusterekbe de a köv iterációnál az lesz az új klaszterközéppont aminek a clusterben lévő pontoktól vett távolsága minimális—így az outlier nem tudja maga felé rángatni a clusterközéppontot

Mikor nem ajánlott használni? Ennél nem figyeltem, szóval majd egyszer rájövök, vmi az idővel, clusterek számával…

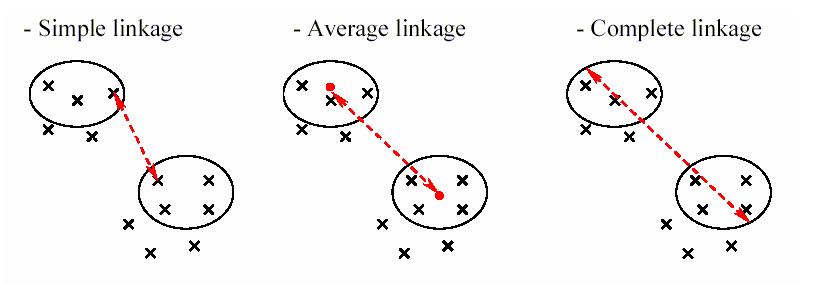
**Hiearchikus clasterező**

Lépésenként összevonja a legközelebbi kettőt—hogyan definiálsz cluster távolságot

Min-single linkage: 1 lépés 1 összekötés, dendogram, de nem látom mi a szabály/különbség a 3 verzióban

Max-complete linkage

Avg-average linkage



http://compbio.pbworks.com/f/linkages.JPG

A hierarchikus clusterezés egy „mohó” algoritmus-amit egyszer összevontál az itt már úgy is marad

Top-down vs bottom-up

Két részhalmaz távolsága vmi alapján maximális lesz, a top down-t ezért ne használd soha, mert számításigényesebb, bottomup a jó

Hogy hasonlítod össze az eredményt? Pl kmeans és hiererchikus clustering is lefut uazon az adaton

Pl az jó ha az outliereket elkülöníti külön clusterbe? Neeem

Cél lehet ezek közül 1 vagy mind akár: Kluszterközépponttól vett átlagos táv a cluster tagokra értve minimális, kluszterek egymástól vett távolsága pedig vhogy legyen maximális, legyen cost rate a clusterek elemszámára, ne legyen nagy különbség különböző clusterek elemszámában, ne legyen 1 elemű klaszter nagyon—szubjektív a visszamérése

Dimension reduction algorithm—PCA személyesen az eredeti változókból új dimenziókat hoz létre, úgy h infotartalom minél jobban megmaradjon, a régi változók lineáris kombinációi adják az új változó értékeit

# óra 2019/03/12 gyakorlat

ma is ad kisházit

03/26 az első python óra

03/19 nincs óra

Clusterezés continued

Particionáló algoritmusok vs hiearchikus clusterező

Először filter data type aztán k menas

Mi a label változó?

Filter example operátor- no missing attributes

Van egy plot fül rapidminerben ahol láthatod, hogy mi történt mi az „e”

Egyes változók kluszterközéppont értékei vannak felrajzolva

Agglomerative clustering=hierarchikus clusterező

Azért ilyen széles a dendogramm mert kiugró értékek, erre utal h ilyen alakja van a görbének

Flatten clustering operátor

Customer id a label

Azért nem megy, mert nem normalizáltak a változók, másképp vannak skálázva

Z transform=standarizálás, 0 várható érték 1 szórás

Range transformation lesz

1es klaszterben vannak a nagyon sokat telefonálok most, de még mindig sok az 1 elemű klaszter

???how to insert subprocess??? A data prep subprocess kéne

Visszamérési metrika (mind szubjektív, csak másképp)

Kisházi egy másik visszamérési metrika interpretálását kéri

Most SSB/SST hányadost használjuk erre

SSB- megtalált klaszterek középpontjai milyen sum távolságra vannak az egész adathalmaz közepétől

SSB minél nagyobb annál jobb a kluszterezés, ekkor volt értelme annak, hogy szétosztottad őket clusterekbe

minél kisebb, kulszertközpontok annál közelebb esnek grand meanhez, az X- az az adathalmaz tulajdonsága, kluszterezéstől független, x az összes pont

Minél nagyobb a hányados annál jobb

Pivot tábla/kimutatás segít ebben

Úgy számoltuk ki most, hogy bünteti a kevesebb clustert, mert sima sum távolságot számol

Ez alapján az a jobb modell, ahol 4 db 1 elemű és 1 óriás cluster van, de ez hülyeség attól még azért…

Outlier keresési technika

Csinált egy ilyen subprocesst

# 9.óra 2019/03/13

Osztályozó algoritmusok folytatása, felügyelt tanulás szóval van kimeneti változód is, ismert az értéke

Regressziótól más, mert itt a célváltozó nominális, ált bináris (regressziónál folytonos)

Logisztikus regresszió ismétlés, logit transzformáció, lineáris kombináció, p értéket becsül, bináris osztályozásra képes ez

Osztályozás hatékonyságának visszamérése: pontosság (többször találja el teszthalmazon a valós célváltozót)—de ha nagyon kevés a 0, és a modell mindenhova 1est tippel akkor csalókán nagy lehet a pontosság, pedig üzleti felhasználhatósága ennek csekély

Másik típus a **példányalapú algoritmus/Lusta osztályozók**

k-NN (nearest neighbours) algoritmus (!!!ne keverd a k meanssel ami kluszterezési algoritmus

1 NN: megkeresi azt a pontot ami legközelebb van a cián ponthoz, ami annak a célváltozója, annak fogja osztályozni ezt is

kNN pl. k=5 5 legközelebbi pontot megnézi, amelyik érték nyer, annak kategorizálja (amilyen érték leggyakoribb volt)

1. Implementáció a többségi szavazás elve alapján
   1. De ez a távolságot nem veszi figyelembe
2. Súlyozott—reciprok alapján—aminek a reciproka a legkisebb??? a két pötyi a szumma tetején piros és kék, a két kategória a példában k=i

Ha normáljuk a total distancre ezt akkor valószínűséget is meg tudunk adni

Lehet négyzetes távolsággal is súlyozni, hogy büntesse az outliereket

(K-t érdemes páratlannak tenni)

Eukledesz távolság minden bemeneti változót azonos súllyal vesz figyelembe

K NN-t alkalmazni regresszióra—súlyozott átlag és voila, használható rá

Azért lusta, mert nem épít modellt, nem határoz meg együtthatókat, a modell itt maga a tanuló adathalmaz!!

Ha nagy az adathalmaz akkor ez egy elég erőforrásigényes megközelítés azért, lassú

K értéket lehet empirikusan is pl 2-től 50 ig végigpróbálod, visszaméred hatékonyságot, és összehasonlítod

**Ő nem hisz abban h a k értékét a minta elemszámhoz képest határozd meg**

Csak numerikusat tud! Szóval dummy ha nincs más, WoE is lehet

Érdemes normalizálni (ld cluster) ha más nagyságrendű skálát akkor dominálja vmelyik a távolságot a szomszédkeresésnél

Hiányzó értékeket rosszul viseli

Döntési fa algoritmus (család)

A kész modell csomópontokból áll

Első Csomópont teljes adathalmazt tartalmazza

A vágások egyszerű esetben binárisak, adott változó adott értéke mentén

Azok a csomópontok, amikből már nem vezet ki vágás, levél—ez prediktál értéket—pl levélben a tanuló szettben többségi szavazás által kapott értéket

Hogyan találod meg a vágási pontot—olyan részhalmazokat eredményez, ahol a célváltozó szempontjából a részhalmazok homogénebbek lesznek, mint a vágás előtt—de ha 1:99 arányban vág akkor nem lesz túl informatív—figyelni kell a részhalmazok méretére

A modell általánosító képessége—ha erre nem figyelsz akkor overfitting/túltanulás

Leállási feltételek a faépítés során:

pl homogenitás elér adott szintet a csomópontban a célváltozóra nézve (pl. 90%), célváltozó értékeinek eloszlása alapján

szintek száma alapján—fa mélysége

vágás előtti és utána homogenitás—ha nem nőtt elég nagyot akkor mégse—nyesés (pruning)

adott csomópontba hány sor került pl 100 alatt már nem vágunk tovább akármi is van—csomópont elemszám

Maximális mérete a fának 🡪 1 levél 1 sor az adathalmazból

Vágások definiálása:

Hunt algoritmus, addig vágunk amíg a kapott részhalmazok homogének lesznek célváltozóra, összehasonlítja különböző outputokat; egyszerű, de könnyű összeomlasztani

Cart algoritmus: classification and regression tree

GINI – ki lehet számolni egy adott részhalmazra—ezzel össze lehet hasonlítani a különböző vágások ’jóságát’

Az elméleti legjobb vágás értéke 0 (GINI 0 és 1 közötti értéket vehet fel)

GINI calculation is on wikipedia, too

1 magyarázó változó szerepelhet több vágásban

A változó nem feltétlen attól fontosabb minél feljebbi szinten van, akkor már az jobban számít h a változó mekkora homogenitásnövekedést tud elérni, de egy változó több szinten is előfordulhat, mint vágási alap

Log regr: konfidencia érték és prediktált célváltozó értéket ad meg, ezt lehet hatékonyság mérésre is, meg sorba rendezésre is használni

Hogyan tudsz döntési fából konfidenciát kiszedni? Konfidencia—adott levélben mi a célváltozó eloszlása, ezt az arányszámot használod (pl x db 0 és y db 1 🡪conf(1)= (y/(y+x)

Ettől max annyi conf értéked lesz ahány levél, ettől a sorrend véletlenszerű lesz; míg log regr értékénél minden sorra számol confidence value-t

Levelek négyzeteket jelölnek ki a térrészbe, dimenziókkal párhuzamos egyenesek mentén vág

Míg k NN nél furi alakú, és még k értékétől is függ

Döntési fánál nem érdemes előszűrni a bementi változókat (míg log regnél igen); döntési fát az se zavarja, ha nem függetlenek a változók; a hiányzó értékek kezelése viszont esetleges; kiugró értékek se zavarják meg

A lin reg simán becsül olyan célváltozó értéket, amit nem látott a tanulás során, míg a döntési fa erre nem képes, az az értékkészlete ami a tanulás során volt, nem tud extrapolálni!!

De elvileg ezt meg lehet ugrani úgy, hogy a levélben építesz 1-1 lin regressziót és így már extrapolál neked

# 10. óra 2019/03/26

Gáspár Csaba innen python

4 alapfeladat, ebből kettő felügyelt tanulás—osztályozás és regresszió; unsupervised a clusterezés és az anomáliadetekció

Ez az utolsó kettő a trükkösebb, már a clusterezés is elég nehéz ám, van neki egy sötét oldala

Nála anomáliadetekció és idősor előrejelzés lesz

A rapidminerben mentek az elemi, egyszerű adatelemzések, pythonban a kevésbé triviális

Dataframe=example set az adattömböd

Azért pythont tanít, mert

Optimalizál a sum fejlesztési idő+futási időre ő

91-es, de 2012 óta hajlik a data science fele, a pandas óta, ezzel a data frame vonalat adta hozzá

Cél az olvashatóság volt

Vannak vizuális adatelemző eszközök, ami ugye szélesebb körben hozzáférhető, így könnyebb adatelemezni, olyanoknak is akik nem tudnak/akarnak programozni, de kiderült hogy fordítva lett, hogy inkább minden ITs tanult meg adatozni

Big data tecnológiák miatt is kellett a python programozhatóság, mert az elemzők nem tudtak javat

Az R régebbi volt ezen a téren, de az R t nem programozók csináltak, nagyon anarchikus volt és nehéz lett volna interpretert csinálni

R a másik programnyelv; python közelebb áll az IT agyhoz

Van még az ősi SAS, annyira öreg h nem változik már, 80as évekbeli, nagy rendszereket építenek vele

A python mivel general, nemcsak az adatok kezelésére jó, integrálni is lehet utána amit megcsináltál, ebben durván ráver az R re; hiába hiszik azt h átviszik java-ba, nagy eséllyel nem fog menni nekik 1 éven belül

Learn by doing hívő, szóval ő így tanít pythont is, főképp időhiányból

Ezért ő itt csúnya szintaktikát fog vinni, készülj fel

Pandas és sklearn lesz, meg az alap szókincs; kalapács üzemmód, azzal csinálsz valamit amit most épp tudsz

iPython notebook, jupytert használ ő is

ez egy játékfelület, nem a komoly munkára való, hanem ad hoc elemzésre, mert látszik a kód, az eredmény, meg a dokumentáció is

de programot írni már nem jó. Csak addig amíg tanulsz

mit használj fejlesztésre? Szerinte anaconda spider nem jó, mert túl light

ő szerinte pycharm free verziója, műegyetemi hallgatónak free a professional is, be lehet állítani h az interpreter szerveren fusson, ne a saját gépeden—de ezt csak a fizetős tudja, kb 40000 az ára

<http://notebook.dmlab.hu>

jelszó dataBook018

enyém: <https://notebook.dmlab.hu/notebooks/DAP2019/2019-03-26-Python-Repulorajt/Szab%C3%B3_D%C3%B3ra.ipynb>

a zzz\_PREP az az ő tutorialja

listába ebbe bármi befér, Rben egy listán belül csak 1 típus lehet

indent=blokkosítás

pass a nem csinál semmit, üres sor pl if en belül

függvény fv(1,2), ha megadsz default értéket, akkor annak hátulról/visszafele kell folytonosnak lennie, ha tudod a paraméter neveket akkor mindegy a sorrend)

paraméterek, függvényátadás

amit magyarul ír az ugye az ő intervenciója nem python syntax

csomagkezelés text file ba azt elmented py ként, utána ugyanabba a mappában van és meg tudod hívni a notebookba

import filename ide nem is kell már a py kiterjesztés, ha itt nem találja akkor elmegy a python install mappába—ha többedjére importálod akkor csak elsőre hajtja vége—ha azt szerkesztgeted amit importálsz aztán, akkor kernel restart kell ahhoz hogy updatelődjön

szdori.kedvenc\_szamom package.variable\_name ként tudsz hivatkozni rá úgy hogy megkülönböztetsd

# adatelemzos mantra

import pandas as pd

import numpy as np

import matplotlibplot.pyplot as plt

%matplotlib inline

Pythonnal gyors ábrát lehet gyártani, csúnyán, utána külön kell szépíteni

Pythonba nem keződhet % ez a matplotlib sajátja, van neki olyanja ami magick néven fut

Shift tabbal függvényre állva kiírja a dokumentációt, hogy lásd a paramétereket

../ shift+enter felhozza hogy milyen fájljaid vannak a mappában ha ezt a read.csv be nyomod

Jupyter notebook szépen megformázza a táblázatokat amikor megjelenít (ellentétben a core pythonnal, ami nem ilyen szép)

Trükk: a for ciklust nem szabad megírni sokszor, pl új oszlopot ne így csinálj

Holnap nincs óra

Pandas ha csak 1 string van megadva, akkor az oszlopok között kezd keresni

Ha vmi működik 1 elemre akkor ált többre is, itt pl listaként

Indexelős zárójel, lista generálós zárójel is []

Dataframe—ami min 2 oszlop és szépen van ábrázolva jupy ban

De a series az ocsmány ugyanitt—neki nincs is neve az ábrán, csak index és érték ehhez nincs megírva szép megjelenítő

Ha 1 oszlopos data frame-t akarsz a series helyett: df[['visit\_date']].head()

df.columns

#gyors módszer type convert re jupymban

Pyrthonban bármikor dinamikusan kiegészítheted végig az objektumot, ettől az elírásokra különösen érzékeny

Űjranevezheted a fv eket, de a for t már nem pl

# 11. óra 2019/04/02

Tescos adathalmaz melyik országból származik? Anomália detekció

Pl. kiszámolna napi szum forgalmat v vásárlók számát

Spark-ba lehet gondolkodni átugrani ha szorít a memoria és nem tudsz vmiért több hardvert venni

MAPE érték mean absulote percentage error

APE hány százalékot tévedsz hogy ha a hét napjara szamolt atlagos erteket tippeled,

**Anomális detekció**

Uzletileg furcsa(pl. MOL) vs elemzesileg/adatilag furcsa

Pl adatok jöhetnek: NAV, halozat/szerver

Adatelemző számára a definició az az hogy ’nem tipikus’, és akiből sok van „ők a jók”, normalitás a gyakoriságból származik, nem szabályalapú—>az az ad-hoc BI, sqlből kiszeded neki

Secunder tanulás—visszakapod hogy a jelzéseidből mi jött be, mi volt fals—ez alapján fine tune parameters to further optimize

# 12.óra 2019/04/03

Gáspár Csaba

Anomália detekció: ált unsupervised learning

Anomális working definition: nem magyarázódik az adatból az eset

Az üzleti def az más, pl csalókat flaggelni, hibás működés infrastruktúra társfelügyelete

A tipp ellenőrzése drága (false positive is costly)

Az üzleti értelemben vett anomáliákat keresik a valós életben

„drága függvény” mert expert kellett ahhoz, hogy eldöntse soronként hogy az most gyanús-e a tanító szettben

Ellonorzo fv 2 kimenete: ok, NaN

Az ok az úgy néz ki, mint a többiek, a NaN a hümhüm

Az adatsorral az a baj, hogy a legritkább esetben dátumoznak vissza megjegyzést egy másik team-ből hogy 2 hete xyz szerver hibás volt és most javítottuk meg…. Nem tudod ált, hogy a gyanúból mikor lett (ha lett) esemény

Ha mégis van adatod, akkor az oszlopban 1 és NaN van, az NaN-ok között is valószínűsíthető, hogy kell lennie 1eseknek igazából

Ez vmennyire hasonlít egy osztályozáshoz, de nem uaz

A teszt is ilyesmi szokott lenni, hogy a NaN-ok közül szedd ki azokat, amik még valószínűsíthetően 1-es az 1-esek alapján

Ez a setup arra jó, hogy a kevésbé gyakori történéseket vegye ki, ha túl sok a csaló…akkor az nem akad fenn, gondolhatsz rá úgy, hogy egy fajta csalást keresel

Osztályozás confidencia értékéhez analóg módon, az összes NaN mellé generálsz egy anomlália score-t (ez algoritmustól függ, hogy a kis v a nagy szám jelzi, hogy mennyire gyanús)

Az ellenorzo fv drága és még sokáig is tart, tehát pl 5 sor/hét az ellenőrzés—ezt beleteszed a másodlagos tanulási folyamatba (confirmed 1 or temporary 0, temporary because it fades away back to NaN)

De legalább az ellenorzo fv nem is egzakt, operátortól függ h szerinte ez gyanús v nem (ahogy az orvosok diagnózisa se az…)

Azzal jársz jól, ha minden entitásnak külön score-t adsz, tehát ellenőr33 szerint az eddigiek alapján ez gyanús, de nem adhatod neki azt, ami ellenőr16nál lenne a top3…

Ez egy döntéstámogatás, nem egy egzakt feladat! Emberek használják fel, a gép itt csak ötleteket dob fel

**Anomáliák 3 fő típusa**

1. Pontszerű anomália: sokdimenziós tér, az a tipikus, ahol sok adat van, az a kiugró ahol kevés pont van, nem illeszkedik az adatba
   1. Outlier detection az egyik minősített esete ennek, egyik dimenzióban az ő értéke durván kiugrik, ez általában adathibáról szól, nem illeszkedik az adatba. Pl 124m es testmagasság, 350mrd os ház
2. Kontextusfüggő anomália: reális az érték, de nem ezek között a feltételek között pl. fagyi forgalom télen vs nyáron; ezt lehet időben csinálni, típus alapján (pl. ilyen szerver nem így szokta), földrajzi lokáció alapon
3. Kooperatív anomália: a szereplők pontszerűen tök jól viselkednek, de együtt nagyon csalnak—ezt számítástechnikailag engedd el addig amíg nincs kvantumszámítógép, más megközelítést igényel, itt ne akarj machine learninget, adj inkább jó query felületet az elemzőjüknek a lekérdezéshez, még sql-ben is nehéz megírni egy ilyen kérdést ahol a körbeutalást keresed, létezik de nem ezen kurzus keretén belül

Az 1,2 vel foglalkozunk mi

Pl klusterezel és kéred a nagyon kis elemszámú klasztert, vagy a kis sűrűségűt ahol n távolság van a pontokon, v azt amelyik n messze van a clusterközépponttól

Feature-ök alapján építed meg a clustereket

Aztán stat, melyik clusterben mennyi volt az ismert 1 esek aránya

Stratégiákat tudsz ellenőrizni (?)

Így lesz ellenörző fv-ed, titkos fájlból 5-10 et kiveszel ellenőrzésre, de ettől ez még unsupervised módszer

Clusteren belül ne akarj osztályozni, sajnos nem működik, hiába szeretnénk hinni h vannak embercsoportok

Pl ezért utálja az iris (mert itt clusterezéssel lehet osztályozni, és a valóságban never ever) és a titanic adathalmazt (h azt miért azt majd később mondja el)

Sklearnben a legtöbb gépi tanulás fapados verziója megvan

Mivel unsupervised azért itt nincs test és train splits

from sklearn.cluster import KMeans

KMeans(algorithm='auto', copy\_x=True, init='k-means++', max\_iter=300,

n\_clusters=8, n\_init=10, n\_jobs=1, precompute\_distances='auto',

random\_state=None, tol=0.0001, verbose=0)

random\_state= adj meg konkrét értéket, ha jót akarsz

a model.fit az gyakorlatban a model.train-t jelenti de fitnek hívják mert csak

megmutatta az 1 soros if et lambda funkcióval

df['info\_2']=df['info'].apply(lambda x: 1 if x[0]=='d' else 0)

ha a string első betűje „d”…

van egy soros for ciklus is ilyesmivel amúgy

hétvégére fent lesz a kaggle verseny kb, AUC kelleni fog valszín hozzá