

DATA SCIENCE *(adattudomány)*

A LIDL-nél

Bevezető / gyors összefoglaló:

Az éves projekt célja az volt, hogy szerezzek és elmélyítsem a tudásomat a Python programozási nyelvben és az adatelemzési területen. A Python egy tág és többbéteggű programozási nyelv, melyben a játékkészítéstől a különböző adatvédelmekig bármit meg tud csinálni az ember. Ebben a projektben adatelemzéssel és előrejelzések gyártásával foglalkoztam. A projekt során főként a NumPy, Pandas, Matplotlib és Seaborn könyvtárakat használtunk. Ezek a könyvtárak nagy segítséget nyújtanak a nagy adatbázisok gyors és hatékony elemzésében. Használtuk még a Scikit-learn könyvtárat is, mely inkább a modellekkel segít. Használtuk még a Microsoftnak az LGBM-jét (light gradient-boosting machine) is.

Indulás:

Az egész akkor kezdődött, amikor egy május végei napon, Szilva egy lehetőséggel kínálkozott. Érkezett, egy a területében kiváló szakember, Törköly Tamás az iskolába, aki egyetemi diákokat toborzott, hogy megmutassa nekik az adattudomány világát. Ez egy apró bökkenő volt, de az találkozói alkalmával többen is nagy érdeklődést mutattunk, így a korábbi digitális tanulásunk miatt beleegyezett. A következő tanév elején volt az újabb meeting, melyre az évfolyamról 4-5-en elmentünk. Itt már nagyobb részletességgel bemutatta a különböző ágakat az adattudományon és az ő feladataikon belül. Nyáron még gondolkoztam, hogy esetleg inkább kezdenék valamit a sakkal, esetleg tanulmányokat írnék a különböző megnyitások elnevezésének folyamatairól és történeteiről, de a meeting végével bebizonyosodott, hogy én ezt szeretném inkább csinálni, mert ez egy jól körvonalazódó projekt volt, viszonylag egyértelmű munkarészekkel és ezért rendesen be tudtuk osztani az időt (ez aztán később teljesen másképp bizonyosodott.) Az élő meeting után, néhány online beszélgetés folyamán kisebb csoportokra váltunk, ahol az érdeklődési körünkben sokkal jobban el tudtunk mélyedni. Nekünk ez a Botival a különböző boltok eladásait kellett termék-családonkénti lebontásban előrevetítenünk. Ezenkívül minden csoport kapott egy mentort, aki már a LIDL-nél programozik és ők kísérték minket végig a folyamatokon, támogattak minket, ahol szükséges volt. A mi konzulensünk Wurdits Ádám volt, aki erőfeszítést és fáradságot nem kímélve segített nekünk (számtalan alkalommal, a saját egyetemi tanulmányai és munkája mellett 3-4-5 óra hosszan egyeztetett Botival és velem.)

A munka első része:

Egy viszonylag hamar kinyilvánuló nehézség számunkra az volt, hogy nem tudtuk programozni. Erre a Tamásék már az elején felkészültek így intéztek számunkra a Udemy felületén kurzusokat. Két kurzusból volt lehetőségünk tanulni egy 40 órás, valamint egy 17 órás. A hosszabb a Python for Data Science and Machine Learning Bootcamp, melyben Jose Portilla az adattudomány egyik legnagyobb klasszisa tanított minket. Ebbe a 40 órába csak maga a videóanyag számított bele, azt nem tartalmazta,

hogyan leállítod és megpróbálsz értelmezni, esetleg visszatekersz, ami a szükséges időt nagyon gyorsan többszörözte. Ezen kívül még minden leckéhez járt egy Jupyter notebook (Python kódok lefuttatására szolgáló jegyzet-, munkafüzet), melyben feladatokat kellett megoldani. Leckénként 1-2 órányi feladat. Összesen 26 lecke van, de ebből egyet-egyet kihagyunk, mert nem volt teljesen releváns. A tanulási folyamat elég sokáig tartott: szeptember végétől január közepéig. Teljesen új volt az egész, így kicsit döcögösen indult be, de kitartóan csináltuk. Év közben néhány helyen kimaradtak dolgok, de ezeket december végére realizáltuk és be tudtuk pótolni. A karácsony előtti napok intenzív kurzusvégzéssel folytak. 11-től 7-ig (természetesen evés és rövidebb pihenési szünetekkel ékelve) nagyjából 7-8 napon keresztül csak programoztam. Ez a korábbi hiányosságokat bepótolta és frissítette a már megtanultakat.

Átmenet:

Január közepe felé visszaálltunk a kurzus végzésre (2-3 óra / hét), azonban ez nem tartott tovább pár hétnél, ugyanis elértünk egy olyan szintre, ahol már nagyjából magunktól tudunk programozni. Ekkor még nem voltak meetingjeink, inkább csak összetettebb feladatokat kaptunk az Ádámától. Ezeket a feladatokat kicsit kevesebb intenzitással csináltuk, kicsit elkalandozott a figyelmünk. Az Ádámnak ez gyorsan szemet szűrt és ezért egy új munkafolyamatot eszelt ki számunkra. Ennek következménye, hogy február közepére a munka, amit belefektettünk heti szinten hatványozódott.

Második része:

A munkánk központjába egy internetes kihívást / versenyt állítottunk. Ez a „Store Sales – Time Series Forecasting” (Bolti eladások – Idősoros előrejelzés). Ez a Kaggle (egy Jupyter alapú online munkafüzet, melyben akár többen is tudunk dolgozni) felületén volt. A kihívásban egy ecuadori bolthálózattal, a Favoritával foglalkoztunk. Több mint 50 boltból kaptunk eladási adatokat egy 5 éves periódus folyamán, majd ezek alapján kellett minél pontosabban megjósolnunk egy 16 napos időtartamra az eladásokat. Hamar bele is csaptunk a lecsóba. Alkalmaztuk az Ádám új módszereit: többek közt, elkezdtünk új tematika szerint hetente többször meetingelni, ezt a click-up segítette. Emellett létrehoztunk egy közös notebookot (munkafüzetet), amibe a Botival egységesíteni és összesíteni tudtuk az egyéni munkáinkat.

Új meetingelési tematika, ceremóniák:

Az egész munkánkat sprintekbe szerveztük. Ezek 1-2 hetes periódusok voltak, melyek a tisztánlátás végett hetekre előre volt tervezve. Egy sprintben klasszikusan hetente 2-4 meetingünk volt. Hétközben voltak a standupok, azaz egyéni meetingek, ahol az Ádámmal egy per egyben voltunk és a saját feladatrészeinkkel haladtunk. A hétvégén pedig volt egy workshop. Ezek voltak azok, ahol hárman közösen meetingeltünk. Ezt mindig egy sprint review / retrospective-vel nyitottuk, ahol elmeséltük, hogy milyen feladatokkal készültünk el, milyen érzéseink voltak a sprinttel kapcsolatban. Ezután a Botival elmeséltük egymásnak a heti munkánk lényegét és a kódokat elmagyaráztuk. A hétvégi meetingeken voltak még rövidebb „előadások” az Ádámától, ahol elmagyarázta az új anyagrészt és tanított hasznos kiegészítő dolgokat.

Projektmenedzsmenti eszköz és kis ismertető a feladattípusokról:

A feladatainkat egy ticket rendszerben kaptuk meg. Ez annyit takar, hogy minden feladatnak volt egy külön leírása, valamint kaptunk hozzájuk segédanyagokat, videókat, dokumentációt. A nehezebb feladatokhoz még egy „hint” is járt, ami rávezetett minket arra pl., hogy mivel kezdjük. Erre a feladat rendszerezésre, egy jó UI-val (user interface – felhasználói felület) rendelkező weboldalt, a click-upot használtuk.

Click-up:

A Click-up egy weboldal, ahol a feladatainkat tudtuk számontartani. Ide írta össze az Ádám a feladatokat, amiket aztán hozzárendelt az egyikünkhöz. Egy hasznos funkciója, hogy a feladatokat meg lehetett labelni/tagelni (jelölni, hogy pontosan mivel is foglalkozik). Ezen kívül prioritást is be lehetett állítani, valamint követni lehetett, hogy melyikünk hol tart, valamint, hogy melyik fázisában van éppen a feladat. Erre is külön rendszer van a click-upon. (To do - csinálni, in progress - folyamatban, under review – vizsgálat alatt és done – teljesített.)

Feladat állapota

Prioritás beállítása

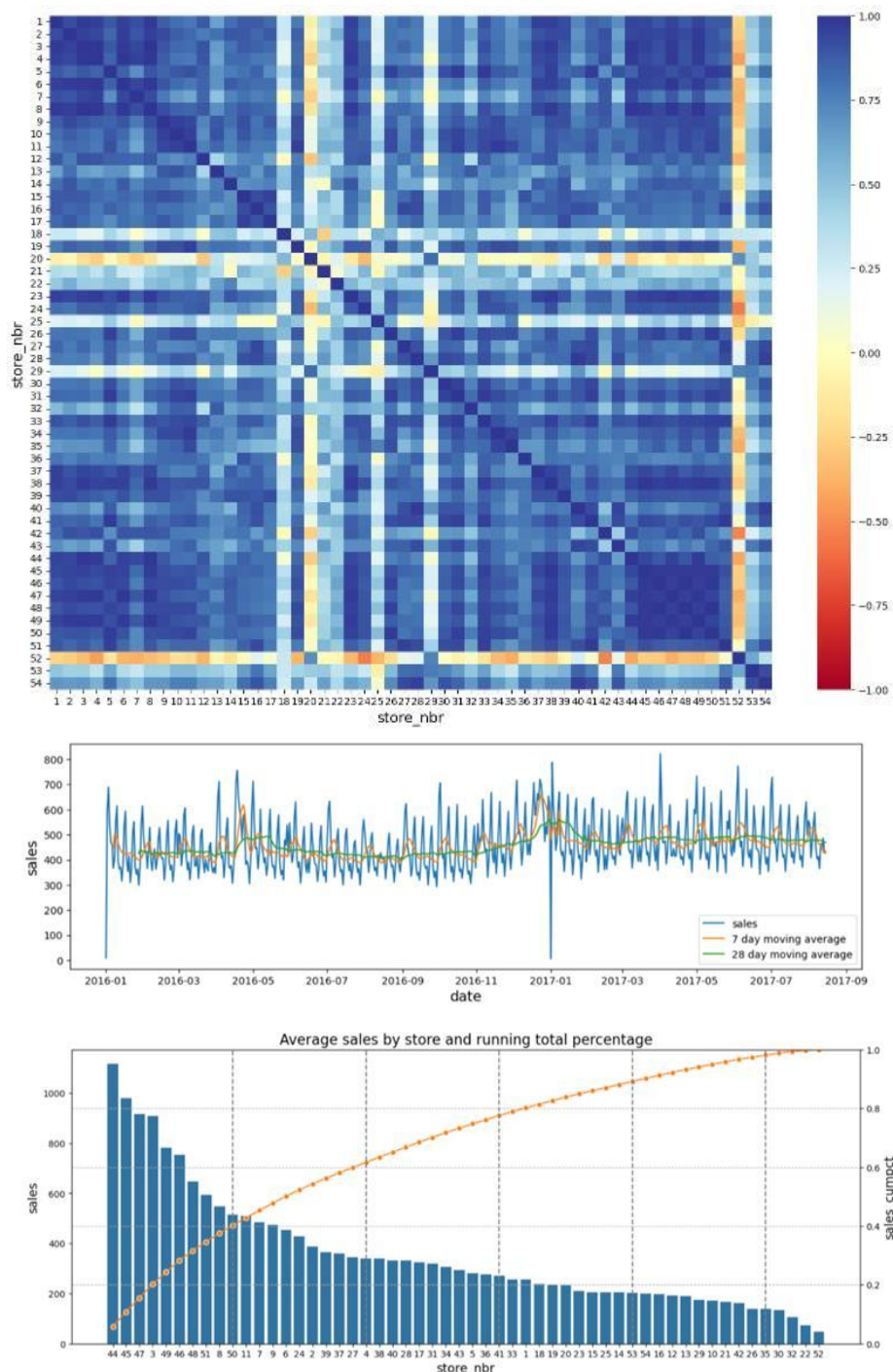
Tagek

Visszatekintések

STATUS	TASKS	ASSIGNEE	DUE DATE	PRIORITY
DONE	2 TASKS			
	Create a function that downcasts datatypes	BM		High
	Read, treat, and join external oil data	SB		Low
IN REVIEW	2 TASKS			
	Research one-hot encoding	BM, SB		Medium
	Research clustering	SB, BM		Low
IN PROGRESS	4 TASKS			
	Clean notebook	SB		Low

Labelek amiket a Click-uphoz hozzáadtunk, a kódolási feladatok típusai:

Eda – Exploratory Data Analysis, azaz felfedező adat elemzés. Itt nagyrészt gráfokat és diagrammokat kódolunk. Az a feladat, hogy észrevegyünk összefüggéseket, anomáliákat, adathibákat (amiket aztán majd kitisztítunk.) Néhány példa:



Evaluation – Értékelés. Ez a rész a kódunk sikerességének kiértékelésével foglalkozik. Különböző metrikák alapján tudjuk eldönteni, hogy az adott modell (vagy modellek, hibrid használata esetén) milyen eredményeket értek el. A kettő fő, amit mi használtunk a MAPE (Mean Absolute Percentage Error – átlagos abszolút százalékos hiba) valamint az RMSLE (Root Mean Squared Logarithmic Error – Logaritmikus hiba négyzetes átlagos négyzetgyöke.) A verseny RMSLE alapján értékelt, de mi inkább a MAPE-t vizsgáltuk a könnyebb érthetőség és magyarázhatóság miatt.

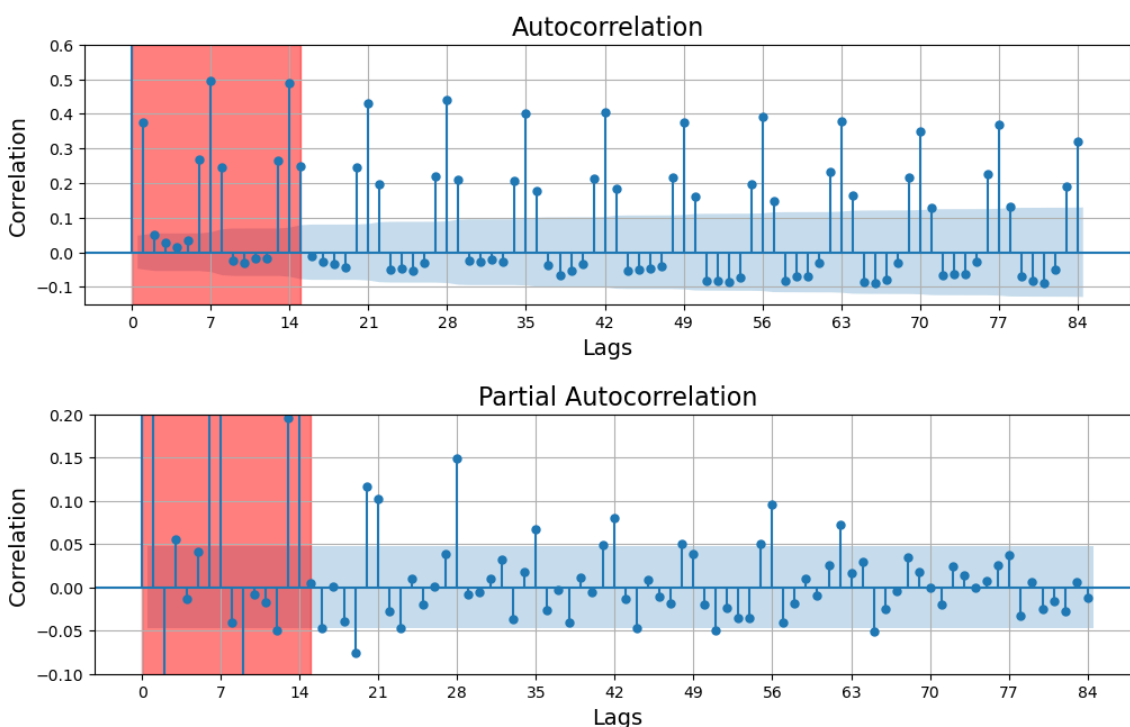
Feature engineering – Új változók létrehozása. Ez felelős a modell számára legjobban kezelhető adatkészlet előállításában. A különféle függvényeket a függvénykönyvtárban gyűjtjük, de erről később fog még szó esni.

Maintenance – Karbantartás. Ide tartoznak azok a feladatok, amik a jegyzetfüzet átláthatóságában, rendezettségében és érthetőségében segítenek. Ide tartozik még, hogy jegyzetfüzetekben ajánlatos egy kód írási stílust használni. Mi a PEP 8 standard stílust használjuk, mivel ez van a legszélesebb körökben elterjedve és a LIDL-nél az Ádámék is ezt használják. Ez olyasmi, mint egy arculati kódex, megszabja pl., hogy hány sortörés lehet két függvény között a könyvtárban, hány szóköz van a különböző jelek oldalain és stb....

Modeling – Ide azokat a függvényeket értjük, amelyek a modellezési folyamatban valamilyen szerepet vállalnak. Többek közt ide tartozik a `train_test_split`, ami elkészíti a tanítási és tesztelési folyamathoz szükséges 4 adatkészletet.

Preprocessing – Előfeldolgozás. Itt a hibás adatokat javítjuk ki, a hiányzókat feltöltjük különböző eljárások alapján. Itt kezeljük az EDA alapján felmerülő rendellenességeket.

Research – Ezek kutatási feladatok, olvastunk és néztünk videókat többek közt a klaszterezésről, hyper-paraméter optimalizálásról, one-hot encodingról, ACF és PACF-ről (autókorreláció és részleges/parciális autókorreláció). Még az olaj piaci árak mozgását és ennek hatását Ecuadorra is vizsgálta kicsit a Boti. Az ACF és PACF plotjaim:



Utility – Kisegítő függvények. Ezek azok a függvények, beállítások vagy éppen kiegészítések, amik az általános feladatok végzik el. Ide tartozik pl. az adatbeolvasó funkció. Egy funkció, ami az adatok méretét csökkenti az egyszerűbb feldolgozás és gyorsabb futási idő érdekében vagy éppen az idő-dekorátor.

Közös notebook:

Létrehoztunk még egy közös notebook-ot. A Kagglerek, van egy olyan előnye, hogy ha az ő felületükön dolgozol akkor a jegyzetfüzeteden keresztül, egyszerűen tudsz a versenyre próbálkozásokat benyújtani. Ezt a notebookot mi Store Sales Main-nek neveztük el. Ide mind a ketten be tudtuk rakni a kódjainkat, amiket a lokálisan saját jegyzetfüzeteinkben fejlesztettünk. A saját jegyzetfüzeteket minden alkalommal a frissített fő verzióról ágaztuk le.) Ha hozzá akartál adni a notebookhoz, akkor mindig figyelni kellett a megfelelő kódolási stílus használatára (PEP-8), valamint először egy PR-t (pull request – behúzási kérelem) kellett feladni. Ezt a főverzióba történő integrálási kérelmet az Ádám ellenőrizte, hogy lefut-e és ha sikerült akkor véglegesen hozzáadta a füzethez. Ez azért volt fontos, mert a Mainnek egy fontos aspektusa, hogy egy 'run all' gombnyomással az egész jegyzetfüzet hiba és fennakadás nélkül lefut.

A jegyzetfüzetben megtalálható még egy verziótörténet. Ahogy haladtunk előre (jelen állás szerint) 90 verziót írtunk, melyeket különféle módokon neveztünk el attól függően, hogy milyen szerepet játszanak. A név legeleje, mindig egy kulccsal kezdődött:

PR – Pull Request, az előbb már említettem.

Bug fix – ha valamilyen, probléma merült fel a futtatás közben, akkor ezt javítottuk itt ki.

EXP – Experiment. Itt a különböző funkciók eredményességét vizsgáltuk. Mindig, amikor valami újat adtunk hozzá a jegyzetfüzethez megnéztük, hogy ez milyen változást eredményezett a metrikán. Az alapvetőbb funkciókkal egy-két tizedesjegyi javulást vagy rontást véltünk felfedezni, de a Feature Importanceben (funkció fontosság) vizsgált, a modell számára leghasznosabb funkciók, ennek a 10-15-szörösét is javíthatták.

Notebook magyarázat:

A legelső lépés minden Python jegyzetfüzetben a könyvtárak importálása. Ezek azok a csomagok, melyek parancsokat és függvényeket tartalmaznak, amik segítenek a munkánkban. A főbb könyvtárak Pythonban a Pandas és Numpy, valamint vizualizációkra a Matplotlib és a Seaborn. Ha ez megvan, akkor jöhet az adatok beolvasása.

A verseny, amin részt vettünk hat táblát nyújtott segítségünkre, meg egy üreset. A Kaggle különlegessége, hogy külön train és test táblák vannak. Való-életű szituációkban nincsen test tábla, hiszen az még nem történt még. Tábláink:

Train.csv – ez volt a fő táblánk. Ezen volt a legtöbb adat, egy majdnem 5 éves perióduson belül szerepelt a dátum, a különböző Favorit boltok (store_nbr), hogy milyen termékcsalád (family), hogy mennyit adtak el (sales), hogy melyik termékek voltak leárazva (on_promotion) és egy azonosító oszlopot (id), de mi ezt azonnal eldobtuk. A train táblák legfontosabb aspektusa, hogy ezen tanítjuk/eddzük (ezért train) a modelleket.

Test.csv – ugyan azokat az oszlopokat tartalmazza, mint a train, csak a megtippelendő 16 napos ablakra és a sales értékeket hiányoznak, hiszen nem történt még meg.

Sample_submission.csv – ez az üres tábla. Ebben tudjuk benyújtani a versenyre az eredményeinket.

Stores.csv – itt információk vannak az egyes boltokról. Pl., hogy melyik város és államban helyezkedik el, milyen a típusa és hogy melyik klaszterbe tartozik.

Transactions.csv – itt napokra lebontva az egyes boltok összesített tranzakciószámát látjuk.

Oil.csv – tartalmazza az olaj aznapi árát. Ez azért fontos, mert Ecuador gazdaságának egy nagy részét teszi ki az olaj.

Holidays.csv – itt a fontos ünnepek és események vannak, melyek befolyásolhatják a boltok nyitvatartását és eladásait.

Ezután megadtunk néhány alap tulajdonságot a táblázatainknak és diagrammjainknak. Itt szerepel még a kapcsoló, amellyel állítjuk, hogy a futtatások alatt melyik részeket szeretnénk (EDA, Backtest, stb.) Ha ez igazra van állítva, akkor futtatjuk le, különben a program átugorja ezeket a részeket.

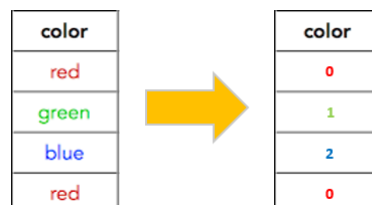
Ezután következik a függvénykönyvtár. Ez talán a legkomplikáltabb része az egész jegyzetfüzetnek és a kódfejlesztés 80%-ban ide írtunk. Itt van az összes utility és feature engineering függvény. Ezek felelősek a train tábla előkészítéséért olyan formában, melyet a modell a legjobban tud felhasználni. A következő részben a legfontosabb függvények fognak szerepelni. (Ezeket vegyesen írta a Boti meg én.)

Downcast – lecsökkenti az adatbázisban szereplő adatok méretét a lehető legkisebbre. Ezzel tárhelyet és futási időt spórolunk.

Read_data – beolvassa a kívánt táblákat.

Concat_dfs – összeilleszti a train és a test adatkészletet. Ez azért fontos, mert ugyan azokat az oszlopokat tartalmazzák és ezért ugyan azokat a függvényeket kell rajtuk lefuttatnunk. Így viszont nem függvényenként két táblán futtatunk, hanem egy nagyobb táblán, amivel időt spórolunk.

Label_encode – a modellek nem tudnak stringeket (betűket) felhasználni, és ezért minden nevet meg kategóriát, amik betűből állnak kódolnunk kell. Az encodeolás minden szöveg típusú kategóriához hozzárendel egy számot.



Orinal_encode – ez ugyanúgy kódolja a kategóriákat számok szerint, csak valami sorrend alapján.

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

Onehot_encode – ez úgy kódolja a kategóriákat, hogy minden kategóriának létrehoz egy oszlopot, és amelyik kategóriába tartozik éppen az adat, abba az oszlopba egy 1-et tesz, minden másik oszlopba 0-át.

id	color
1	red
2	blue
3	green
4	blue



id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Reverse_encoding – a kulcs alapján visszakódolja a kategóriákat stringekké.

Augment_data – ez egy tisztító funkció. Néhány helyen adathiba van és kevesebb az érték, mint valójában. Ezért azokon a részeken megszorozza az adatot és így egy reálisabb képet kap a modellünk.

Flag_missing_sales – megjelöli azokat a napokat, ahol az összes eladás 0.

Flag_open_stores – megjelöli, hogy az adott napon nyitva volt-e a bolt. Ezt a különböző holidayek és transactionök alapján tudjuk végrehajtani.

Add_date_feature – hozzáad az adatkészletünkhöz a jelenlegi dátum alapján egy év, negyed, hónap, hét nap, év-napja, hónap-napja és hét-napja oszlopokat.

Add_waves – hozzáad szinusz és koszinusz hullámokat. Ez azért fontos, hogy a modellünk tudja, hogy az év első és utolsó napja között ugyan annyi idő telik el, mint az első és második között. A hullámpárok frekvenciája megegyezik a hét/hónap/év periodicitásával és ezért tanulja meg, hogy a különbségek ugyan akkorák.

Split_df – ez a train és a test mentén szétválasztja az adatkészletünket.

Counter – számolja a napokat a következő fizetési naphoz.

Lag function – ez a napi adatokat eltolja megadott nappal. Ez azért fontos mert a PACF és ACF plotokban megfigyeltük, hogy a legnagyobb összefüggés az eladásban az előtte lévő hetekben a hét ugyan azon a napján van. Ezért elcsúsztatja pl. a keddi adatokhoz az előtte lévő 4 kedd adatát.

Add_oil/holiday/stores/transactions_df – ez hozzáadja a fő adatkészletünkhöz az ezekben a táblákban szereplő adatokat.

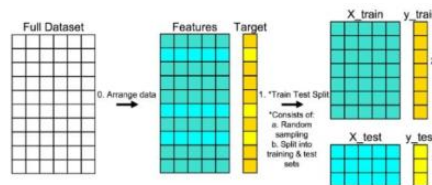
Train_test_split – ez szétválasztja az adatot tanuló és tesztelő adatkészletre. Később esik még róla szó.

Fit_predict – ez egyszerre illeszti a modellünket és becsül.

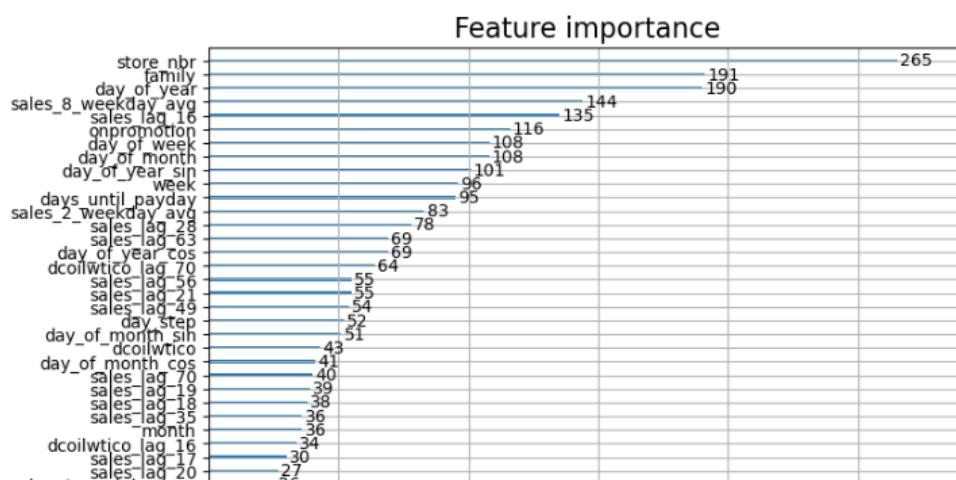
Prepare_data – végül jön az egésznek a kulcsa. Ez egy függvényben függvény, és amikor előkészítjük az adatunkat a modell számára akkor hívjuk meg. Ez az előbb felsorolt függvényeket tartalmazza és szép sorjában lefuttatja az összeset és a végére visszadobja nekünk azt az adatkészletet, amit már csak be kell adnunk a modellnek.

Ha megvan a függvénykönyvtárunk, következik az EDA. Erről már meséltem, de igazából itt plotok százai vannak. Csináltunk, barplotot, lineplotot, scatterplotot, boxplotot, heatmapet, és hasonlókat. Csináltunk feltáró adatelemzéseket pl. a termékcsaládokról. Itt megvizsgáltuk, hogy melyik termékcsaládból mennyit vesznek és melyik a legsikeresebb. Vizsgáltunk boltokat is. Itt például meg tudtuk nézni, hogy melyik bolt, hogy szerepel, mikor nyitottak új boltokat... Vizsgáltuk az eladásokat különböző módokon. Amit itt meg tudtunk állapítani többek közt, hogy pénteken és szombaton vásárolnak a legtöbbet az emberek és aztán ahogy haladunk előre, úgy egyre kevesebbet.

Ha megvagyunk az EDA-val jön a Modellezés. A modellezésnek három része van. Először a modellt szétválasztjuk: ezt hívják *train_test_split*-nek. Itt létrehozunk egy *train* adatsort, amin a modellünket eddük majd, valamint egy *testet*, amin ellenőrizzük. A működése:



A második részben illesztjük a modellünket. Itt megismertetjük őt a *train* adattal és ő különböző algoritmusok alapján ezekből információkat szűr le. Minden *feature*-t más erősséggel vesz figyelembe, ezt hívják *feature importance*-nek. Vannak olyanok, amelyeket jobban tud felhasználni és olyanok is, amelyeket kevésbé. Ezt mi minden futás után egy gráfban lekérjük a modelltől, hogy melyiket mennyire használta. Egy *feature importance* modell teteje (túl hosszú, hogy jól tudjam ábrázolni):



Ha az illesztés megvan, akkor utána jön a *predictelés* - becslés. Itt a modell az üres helyekre beírja a szerinte történni fogó eseményeket.

A predictelés után kiértékeljük a modellünket. Ez csak annyit takar, hogy összevetjük az aktuális eredményeket és valós eredményeket. Itt jön be a `train_test_split` fontossága. Mivel a való életben arra, amit meg akarunk becsülni ugye még nincs adatunk ezért az elején szétválasztottuk az adatkészletünket és így később majd a test adaton le tudjuk ellenőrizni. Különböző metrikák alapján tudjuk eldönteni, hogy mennyire volt sikeres, de erről már volt szó feljebb.

Nekünk még van egy backtestünk is. Ez összefoglalva annyit csinál, hogy guruló ablakokon hajtja végre az illesztést, predictelést és ezért a modellünk sokkal változatosabb adatokon van tesztelve és ezért még pontosabb lesz.

Tapasztalatok és tanulságok:

Az éves projekt egy hosszú, sikeres folyamat volt. Voltak benne bukkanók, nehézségek, megtorpanások, búsakodások, mérgeledések és bosszankodások, de a végére egy akadémiai értékekkel gazdag, élménydús utat jártunk be.

A nehézségek közé tartozott, hogy a programozás világa teljesen új volt számunkra. Semmit sem ismertünk belőle, és ez nekem sokszor el is vette az elejét a kedvemet attól, hogy dolgozzak. Szerencsére, ahogy egyre több videót néztem és haladtam előre a kurzussal, úgy kezdtem belecsöppenni ebbe a világba. A legelső fő sikerélmény, számomra az volt, amikor megírtam a `lag` functiont. Ez volt az első összetettebb kódsor, amit írtam és néhány óra bogarászás és hibajavítás után rendesen lefutott. Láttam, ahogy a meglévő `train` táblánk kibővült és miután leszámoltgattuk, hogy jól jött ki az eltolódás, eszméletlenül boldog lettem. Ez volt az a pont, ahol berántott magába ez a világ és a heti többalkalmas-többórás meetingekre már nem csak, hogy szívesen jártam, de meetingeken kívül is gondolkodtam azon, hogy tudnánk javítani a funkciókat, hogy tudnánk megoldani a problémákat.

Egy másik nehézség a munkánk folyamán az idő egyensúlyozása volt. Nyilván mivel nyelvi évünk van, főleg a szótanulás és a házi megírása volt az előtérben. Emellett még mindketten nagy rendszerességgel sportolunk, hétvégente meccsre járunk. Ezeken kívül még a barátainkkal is töltünk időt, éljük a szociális életünket meg ha már minden sok, pihenünk. Sajnos a 7 hónap alatt nem sikerült egyszer se megtalálnom a jó egyensúlyt. Az év elején többet tanultam, voltam a barátaimmal. Ez azt jelentette, hogy a kurzussal kevesebbet haladtam. Amikor december közepén realizáltuk, hogy kicsit el vagyunk csúszva az egész átbillent a túloldalra. A szünetben még pótoltam az elmaradásokat, de januártól kezdve a munka kicsit a nyelvtanulásom, de inkább a szociális életem rovására ment. Egyrészt ekkor adódtak már alkalmak, amikor nem csináltam meg a házit, a szavakat csak reggel kapkodva tanultam meg. Másrészt viszont nem találkoztam a barátaimmal. Nem mentem el velük kajálni, délutánonként nem dumálgattunk. Ez sajnos ahhoz vezetett, hogy az egyik legközelebbi barátommal elkezdtünk veszekedni és végül eltávolodtunk (de remélhetőleg, ezt majd kijavítjuk, hogy kevesebb munka lesz.)

Az egyensúly felborulásának sokkal veszesebb következményei is lehettek volna, de szerencsénkre az Ádám jól építette fel a meetingeket és rugalmas volt, ezért a felhasznált időt maximálisan ki tudtuk használni és nem volt sok kárba vesztett perc.

Egy nagyon hasznos velejárója a projektnek számomra azon kívül, hogy megtanultuk programozni és stb., hogy belecsöppentünk a felnőttvilágba. Hasonlóan alakította a

dolgokat az Ádám, mint ahogy ők is csinálják és ezért a saját bőrünkön tapasztalhattuk, milyen is ez a dolgozás, hogy működnek a munkafolyamatok. Néha még azt is megmutatta, hogy ők éppen min dolgoznak, melyik weboldalt használják a ticketekre stb.

Egy másik pozitívum, hogy a tanév során a munkánk előadását már gyakorolhattuk. Az első ilyen prezentálási lehetőségről lemaradtam, de a másodikon már sikerült részt vennem. Itt el kellett a Tamásnak mesélnünk, hogy mivel haladtunk eddig. Meg kellett mutatnunk neki a notebookunkat és hogy tisztában vagyunk a fogalmakkal, kifejezésekkel. Ez egy remek lehetőség volt a gyakorlásra, valamint rávilágított arra, hogy mit kell még egyszer átvennünk, mivel foglalkozunk még kicsi többet. Ahol kellett kijavított minket és hasznos tanácsokat is adott, hogy hogyan tovább.

Jövő: „*hogyan is tovább?*”

A Botival már beszélgettünk kicsit erről és arra jutottunk, hogy mindkettőnk folytatni akarja annak ellenére, hogy vége az „éves projektnek”. Meg is beszéltünk már egy alkalmat az Ádámmal, amikor bemegyünk az irodájába és találkozunk először élőben (izgi). Ekkor fogjuk megbeszélni, hogy hogyan fogjuk folytatni. Mind a hárman kicsit kevesebbet fogunk tudni vele foglalkozni, mint az elmúlt hónapokban, ez azért egy kemény időszak volt, más dolgokra nem nagyon maradt időnk. Amit én jónak találnék, hogy még egy néhány hétig dolgozzatunk a Store Salesen, próbálgatunk az eredményen kicsit javítani. Megpróbálhatjuk akár a Kaggle felületén publikálni a munkánkat. Ehhez meg kell majd tanulnunk a markdown cellákat használni és testre szabni. Ezután pedig, nekem még nagyon tetszene, ha egy teljesen új projekttel is foglalkoznánk. Megtanulnánk új modelleket használni, akár a neurális hálókkal is kísérletezhetnénk kicsit.

Végszó:

Bár a munkanaplót bejegyzéseinek minősége nem a legjobb és írhattam volna többet, úgy gondolom, hogy ebben az írásban kitettem magamért és megpróbáltam minden elvárást felülmúlni. És természetesen a befektetett órák és munka, valamint a szinte szakmabeli klasszisú végproduktum magáért beszél.

Nagyon hálás vagyok mindenkinek, aki segített ezt a lehetőséget megteremteni és kihasználni. Örökre emlékezetes lesz, és remélhetőleg sok előnyöm fog belőle származni.

- *Máté Bende, 2023*