

Veri Hazırlığı

- events.json ve meta.json dosyalarını read_json() ile okuduktan sonra iterasyon ile dataframe'e dönüştürüldü.
- Değeri Nan olan product id satırları atıldı
- Df_event dataframe'indeki verinin %10'l test olarak için kullanılmak üzere veriden ayrıldı

NOT 1: Aynı session_id'deki ürünlerin, sepete beraber eklendikleri varsayılarak işlem yapıldı

NOT 2: Aynı session'da aynı üründen birden fazla olanlar tekilleştirildi

Yöntem

Çalışmada iki farklı yöntem kullanıldı. Bu yöntemlerden birincisi, ürünlerin beraberliğini kullanan apriori yöntemi. Yöntemde ürünlerin diğer ürünlerle kaç sefer beraber satıldığıнын bulunması gerekiyor. Yanlış yapılsa oldukça maliyetli olan bu işlemler için, beraber satılan ürünlerin tüm ikili permütasyonları alındı (Şekil 1) ve dataframe bu ikililere göre açıldı (explode).

	productid_unique	productid_unique__perm
1	[HBV00000NE1LU, HBV00000NVZGQ]	[(HBV00000NE1LU, HBV00000NVZGQ), (HBV00000NVZGQ, HBV00000NE1LU)]
2	[HBV00000OEL9W, HBV00000OELAO]	[(HBV00000OEL9W, HBV00000OELAO), (HBV00000OELAO, HBV00000OEL9W)]
7	[HBV00000NE1T6, HBV00000NVZGU]	[(HBV00000NE1T6, HBV00000NVZGU), (HBV00000NVZGU, HBV00000NE1T6)]
11	[HBV00000NFGSE, MTFID52052]	[(HBV00000NFGSE, MTFID52052), (MTFID52052, HBV00000NFGSE)]
14	[HBV00000PQLIY, ZYBENC8148244]	[(HBV00000PQLIY, ZYBENC8148244), (ZYBENC8148244, HBV00000PQLIY)]

Şekil 1

Sonrasında bu ikililer ayrılıp tek bir ürün bazında gruplanınca, her bir ürünün beraber satıldığı ürünlere ve toplam satılma sayılarına ulaşıldı (Şekil 2).

productid_unique__perm	associated_count
(HBV00000PV6O6, HBV00000OE7D4)	544
(HBV00000OE7D4, HBV00000PV6O6)	544
(HBV00000NVZBY, HBV00000NVZBI)	518
(HBV00000NVZBI, HBV00000NVZBY)	518
(HBV00000O3680, HBV00000OE7D4)	515
(HBV00000OE7D4, HBV00000O3680)	515
(HBV00000OE7QQ, HBV00000OE7D4)	513
(HBV00000OE7D4, HBV00000OE7QQ)	513
(HBV00000OE7QQ, HBV00000NVZBI)	512
(HBV00000NVZBI, HBV00000OE7QQ)	512

Şekil 2

Apriori parametreleri olan support, confidence ve lift hesaplandıktan sonra veri, lift değeri 3'ten büyük olacak şekilde filitrelendi (Şekil 3).

productid	secondary_productid	main_count	secondary_count	associated_count	main_support	secondary_support	confidence	lift
HBV00000PVQ5H	HBV00000PVQ5L	11	5	5	0.001075	0.000489	0.454545	930.454545
HBV00000U271V	HBV00000U271X	11	5	5	0.001075	0.000489	0.454545	930.454545
HBV00000U271X	HBV00000U271V	5	11	5	0.000489	0.001075	1.000000	930.454545
HBV00000PVQ5L	HBV00000PVQ5H	5	11	5	0.000489	0.001075	1.000000	930.454545
HBV000002O5N3	HBV000002O5N5	13	10	6	0.001270	0.000977	0.461538	472.384615
HBV000002O5N5	HBV000002O5N3	10	13	6	0.000977	0.001270	0.600000	472.384615
HBV00000PVR1G	HBV00000PVQZA	15	10	6	0.001466	0.000977	0.400000	409.400000
HBV00000PVQZA	HBV00000PVR1G	10	15	6	0.000977	0.001466	0.600000	409.400000

Şekil 3

Lift [0, 1] arasına normalize edildikten sonra, ürün bazında gruplanarak, her bir ürün için, beraber satıldığı diğer ürünler puanlarıyla beraber listelendi (Şekil 4).

productid	suggestion
AILEBIZISMTLDGY54	[HBV00000GYMOJ - 0.9626222008528484, HBV00000NGXGG - 0.9603626130137258, AILEMTTDMT57 - 0.9491556650436266, AILEBIZISMTLDHB18 - 0.9468241728890646, HBV00000PV5VU - 0.9448591437428155, HBV00000NGXFE - 0.9418980736597291, OYUNMTTW2087 - 0.9278187254663753, HBV00000PVBjY - 0.8711684466211331]
AILEBIZISMTLDHB18	[AILEBIZISMTLDGY54 - 0.9468241728890646, HBV00000PV5VU - 0.9197106132810778]
AILEBSHSB22037	[HBV0000080SK3 - 0.9247598894953776, HBV00000NGXFL - 0.8352868228456015]
AILEETI1811300	[HBV00000PQJLN - 0.7568302483309046]
AILEETI5311300	[HBV00000PQJLN - 0.6575801983796288]
AILEFIMAN5299094	[HBV00000PVBjY - 0.7551345889851345, HBV00000NGXFL - 0.7507803056493331]
AILEHDMAR8967772	[AILEHDMAR8967775 - 0.9551990935267259, AILEUCIL6835 - 0.9392878828915716, AILEHDMAR8967777 - 0.9025558848349363]
AILEHDMAR8967774	[AILEMLP5003562 - 0.9624446290335206]
AILEHDMAR8967775	[AILEHDMAR8967772 - 0.9551990935267259, AILEHDMAR8967777 - 0.9367580423352642, AILEUCIL6835 - 0.9365599498964181]

Şekil 4

Çalışmada ikinci bir yöntem olarak FastText kullanıldı. Normalde doğal dil işlemede kullanılan yöntemin, embedding özelliği uygulandı. Bir cümledeki kelimelerin birbirlerine yakınlıklarına göre uzayda dizilmelerini sağlayan yöntemle cümle yerine aynı session'daki ürün id'leri verildi. Buna göre, ürünler diğerlerine ne kadar fazla ve art arda eklendiklerine göre benzerliklere sahip oldular.

Bu yöntemi apriori'den farklı olarak ürünlerin eklenme sırasını da göz önüne aldı.

"HBV00000U273W" id'li "Tadelle Sutlu Cıkolatalı Dondurma 36 gr" ürünü için sırasıyla apriori ve fastText yöntemlerinin bulduğu yakın ürünler (Şekil 5).

productid	name	category	subcategory	similarity	source
HBV00000U27AV	Pernigotti Sade Fıstıklı Kornet Dondurma 125 ml	Dondurma	Kap Dondurma	0.7610994159770398	apriori
HBV00000NFI46	Carte Dor Selection Toblerone 800 ml	Dondurma	Kap Dondurma	0.6343918694458681	apriori
HBV00000NFHXJ	C. Dor Straciatella-Vşne-Bftr Çik	Dondurma	Kap Dondurma	0.6053060893574258	apriori
HBV00000SPIEB	Algida Maraş Usulü Sade & Çikolata & Bademli Karamel 475 ml	Dondurma	Kap Dondurma	0.5912047916889125	apriori
ZYBICN9293428	Viennetta Vanilya-Çikolata 800 ml	Dondurma	Kap Dondurma	0.5588678774910434	apriori
HBV00000SPBCM	Algida Cornetto Disc Vanilya Karamel 140 ml	Dondurma	Kap Dondurma	0.5284734773855276	apriori
HBV00000SPBC4	Algida Cornetto Classico Çikolata 125 ml	Dondurma	Kap Dondurma	0.479695719339553	apriori
HBV00000PLHDK	Golf Maraşım Kesme Çikolata-Kaymak 500 MI	Dondurma	Kap Dondurma	0.46119622642709013	apriori
HBV00000SPIEN	Carte D'or Kat Kat Çikolata & Vanilya & Islak Kek 485 ml	Dondurma	Kap Dondurma	0.4123174100188411	apriori
HBV00000SP6Z4	Carte D'Or Cornetto Disc Milka 140 ml	Dondurma	Kap Dondurma	0.2097636000249	apriori

productid	name	category	subcategory	similarity	source
HBV00000NFHXV	Carte Dor Cls BI Bdm-Krml-Kymk 925ml	Dondurma	Kap Dondurma	0.8534789085388184	fastText
HBV00000SP6Z6	Carte D'Or Uni Cornetto 135 ml	Dondurma	Kap Dondurma	0.8524455428123474	fastText
HBV00000SP84A	Magnum Yakut 100 ml	Dondurma	Kap Dondurma	0.8493083715438843	fastText
HBV00000SPIEB	Algida Maraş Usulü Sade & Çikolata & Bademli Karamel 475 ml	Dondurma	Kap Dondurma	0.8371350765228271	fastText
HBV00000SP6Z4	Carte D'Or Cornetto Disc Milka 140 ml	Dondurma	Kap Dondurma	0.8299534320831299	fastText
HBV00000SP6UL	Algida Frootie 80 ml	Dondurma	Kap Dondurma	0.8255515694618225	fastText
HBV00000SPBCM	Algida Cornetto Disc Vanilya Karamel 140 ml	Dondurma	Kap Dondurma	0.8252162933349609	fastText
HBV00000SP842	Magnum Mini Cookie & Badem 6'lı	Dondurma	Kap Dondurma	0.8251237869262695	fastText
HBV00000SP848	Magnum Cookie 95 ml	Dondurma	Kap Dondurma	0.8200174570083618	fastText
HBV00000U2740	Tadelle Bitter Dondurma 36 gr	Dondurma	Kap Dondurma	0.8173852562904358	fastText
HBV00000SPBCJ	Algida Cornetto Disc Oreo 140 ml	Dondurma	Kap Dondurma	0.8138348460197449	fastText

Şekil 5

Sonrasında bu iki yöntemin çıktıları birleştirilmiştir. Bazı durumlarda 10 tane çıktı üretilemediği durumlar olduğundan, bu durumlarda, gerekli ürün için o ürünün alt kategorisinden, fiyatı o ürüne yakın olan rasgele ürünler seçilmiştir.

Birden fazla ürün için tavsiye istendiğinde ise, her bir ürün için ayrı ayrı tavsiyeler alınmıştır. Sepetteki farklı ürünler için, aynı ürünün tavsiye edildiği durumlara öncelik verilecek şekilde sonuçlar seçilmiştir.

Kurulan model'e Flask ile api üzerinden de erişilebilmektedir.

Test

Verinin test için ayrılan %10'luk kısmı üzerinden denemeler yapılmıştır. 3848 farklı session için, sepete eklenen ilk 2 ürüne göre tavsiye listelenmiştir. 2 ürün için gelen tavsiyelerden, diğer ürünler için doğru olarak bulunan satır sayısı ise 918 olmuştur.

Eklenebilecek Özellikler

- Kod genel olarak daha verimli yazılabilirdi. Çoğu yerde, sonrasında anlatabilmek için kod açık bir şekilde yazıldı
- Json verisini dataframe'e dönüştürürken daha doğru bir yöntem bulunabilir döngü yerine
- Zaman etkisi dahil edilebilir
- Çoklu ürünler için farklı birçok kombinasyon ile modeller eğitilebilir (daha uzun sürecek olsa da)
- Alt kategorilerden faydalandım ama bunun ürün isimlerindeki anlamsal ilişkileri de kullanmak isterdim
- Kodu daha dinamik hale getirebilirdim
- Yeni ürünleri tanıtmak için ve sürekli aynı ürünlerin tavsiye edilmemesi için, arada rasgele olacak şekilde alakalı alt kategoriden birkaç ürün tavsiye edilebilir

Kodun Çalıştırılması

- Kodu github'ta kendi hesabımda açık olarak paylaşmak yerine, yeni anonim bir hesap üzerinden paylaştım. Projeye <https://github.com/tempuser21/recommendation> sayfasından erişilebilir
- Requirements.txt dosyasındaki paketler kurulur
- recommendation_api.py çalıştırıldıktan sonra yönlendirilen link ile çağrı yapılabilir
- api çağrısı için örnek istek;

```
[
  ["HBV00000NE0SY", "HBV00000QU3Z9", "HBV00000NE25T", "HBV00000NG8T3"]
]
```

- api çağrısı için örnek cevap;

```
[{"productid":"HBV00000NEOSS","name":"Nesquik Kakaolu S\u00fct 6x180 ml","category":"Kahvalt\u0131l\u0131k ve S\u00fct","subcategory":"S\u00fct","similarity":0.3014592153,"source":"apriori","product_number":"1","order":1.0,"same_candidate_number":1},{ "productid":"HBV00000QU3ZB","name":"Filiz Makarna Fiyonk 500 Gr","category":"Temel G\u0131da","subcategory":"Bakliyat, Pirin\u00e7, Makarna","similarity":0.1582040927,"source":"apriori","product_number":"2","order":1.0,"same_candidate_number":1}]
```

NOT 1: Bazı işlemlerin ön işleme adımları uzun sürdüğünden, işlem sonuçları csv olarak kaydedildi

NOT 2: Gensim kütüphanesindeki paket uyumsuzluklarından FastText modelini local'imde uygulayamadım. Modeli yüklemekle ilgili cevabını bulamadığım bir sorunla karşılaştım, fazla uğraşma fırsatım da olmadı bu kısımda. Bu yüzden, api çağrılarında sadece apriori'ye göre tavsiye yapılacak. Aynı kodun notebook versiyonunda (recommendation.ipynb) ise FastText de var, Colab ortamında sorunsuz çalıştırdım. Notebook çalıştırıldığında hem apriori hem de FastText için tavsiye yapıyor.