



Rating Product & Sorting Reviews in Amazon

İş Problemi

E-ticaretteki en önemli problemlerden bir tanesi ürünlere satış sonrası verilen puanların doğru şekilde hesaplanmasıdır. Bu problemin çözümü e-ticaret sitesi için daha fazla müşteri memnuniyeti sağlamak, satıcılar için ürünün öne çıkması ve satın alanlar için sorunsuz bir alışveriş deneyimi demektir. Bir diğer problem ise ürünlere verilen yorumların doğru bir şekilde sıralanması olarak karşımıza çıkmaktadır. Yanıltıcı yorumların öne çıkması ürünün satışını doğrudan etkileyeceğinden dolayı hem maddi kayıp hem de müşteri kaybına neden olacaktır. Bu 2 temel problemin çözümünde e-ticaret sitesi ve satıcılar satışlarını arttırırken müşteriler ise satın alma yolculuğunu sorunsuz olarak tamamlayacaktır.



Veri Seti Hikayesi

Amazon ürün verilerini içeren bu veri seti ürün kategorileri ile çeşitli metadataları içermektedir. Elektronik kategorisindeki en fazla yorum alan ürünün kullanıcı puanları ve yorumları vardır.

12 Değişken

4915 Gözlem

71.9 MB

reviewerID	Kullanıcı ID'si
asin	Ürün ID'si
reviewerName	Kullanıcı Adı
helpful	Faydalı değerlendirme derecesi
reviewText	Değerlendirme
overall	Ürün rating'i
summary	Değerlendirme özeti
unixReviewTime	Değerlendirme zamanı
reviewTime	Değerlendirme zamanı Raw
day_diff	Değerlendirmeden itibaren geçen gün sayısı
helpful_yes	Değerlendirmenin faydalı bulunma sayısı
total_vote	Değerlendirmeye verilen oy sayısı



Proje Görevleri

Görev 1: Average Rating’i güncel yorumlara göre hesaplayınız ve var olan average rating ile kıyaslayınız.

Paylaşılan veri setinde kullanıcılar bir ürüne puanlar vermiş ve yorumlar yapmıştır. Bu görevde amacımız verilen puanları tarihe göre ağırlıklandırarak değerlendirmek. İlk ortalama puan ile elde edilecek tarihe göre ağırlıklı puanın karşılaştırılması gerekmektedir.

Adım 1: Ürünün ortalama puanını hesaplayınız.

Adım 2: Tarihe göre ağırlıklı puan ortalamasını hesaplayınız.

- reviewTime değişkenini tarih değişkeni olarak tanıtmanız
- reviewTime'in max değerini current_date olarak kabul etmeniz
- her bir puan-yorum tarihi ile current_date'in farkını gün cinsinden ifade ederek yeni değişken oluşturmanız ve gün cinsinden ifade edilen değişkeni quantile fonksiyonu ile 4'e bölüp (3 çeyrek verilirse 4 parça çıkar) çeyrekliklerden gelen değerlere göre ağırlıklandırma yapmanız gerekir. Örneğin q1 = 12 ise ağırlıklandırırken 12 günden az süre önce yapılan yorumların ortalamasını alıp bunlara yüksek ağırlık vermek gibi.

Adım 3: Ağırlıklandırılmış puanlamada her bir zaman diliminin ortalamasını karşılaştırıp yorumlayınız.

Görev 2: Ürün için ürün detay sayfasında görüntülenecek 20 review'i belirleyiniz.

Adım 1: helpful_no değişkenini üretiniz.

- total_vote bir yoruma verilen toplam up-down sayısıdır.
- up, helpful demektir.
- Veri setinde helpful_no değişkeni yoktur, var olan değişkenler üzerinden üretilmesi gerekmektedir.
- Toplam oy sayısından (total_vote) yararlı oy sayısı (helpful_yes) çıkarılarak yararlı bulunmayan oy sayılarını (helpful_no) bulunuz.

Adım 2: score_pos_neg_diff, score_average_rating ve wilson_lower_bound skorlarını hesaplayıp veriye ekleyiniz.

- score_pos_neg_diff, score_average_rating ve wilson_lower_bound skorlarını hesaplayabilmek için score_pos_neg_diff, score_average_rating ve wilson_lower_bound fonksiyonlarını tanımlayınız.
- **score_pos_neg_diff**'a göre skorlar oluşturunuz. Ardından; df içerisinde **score_pos_neg_diff** ismiyle kaydediniz.
- **score_average_rating**'a göre skorlar oluşturunuz. Ardından; df içerisinde **score_average_rating** ismiyle kaydediniz.
- **wilson_lower_bound**'a göre skorlar oluşturunuz. Ardından; df içerisinde **wilson_lower_bound** ismiyle kaydediniz.

Adım 3: 20 Yorumu belirleyiniz ve sonuçları Yorumlayınız.

- **wilson_lower_bound**'a göre ilk 20 yorumu belirleyip sıralayınız.
- Sonuçları yorumlayınız.

miuul

miuul.com