

## Play Store Uygulama dağılımı Bitirme Projesi

Ben mobil oyun yapımı üzerine çalıştığım için, üstünde çalışmak istediğim verinin de bu konuda olmasını istedim. Üzerinde çalışacak veri bulmakta gerçekten zorlandım. Çünkü, olan veri setlerinde ya çok az veri oluyordu ya da hiç faydalı bilgi olmuyordu. Ücretsiz bu alanda kompleks bir veri seti bulamadım. Bulduğum bu veri play store'daki uygulamaların tümünü, ne kadar indirilmiş - oyunun adı - kim yayınlamış - kaç yıldız almış gibi özellikleri bulunan 2 milyon verili bir set. Fakat, bu uygulamanın ne kadar kazandığı bilgisi maalesef yok. Bu sebepten ötürü, makine öğrenmesi yaparken, çıkarım yapacak bir konu bulamadım, daha doğrusu çok tutarlı olmadı çıkan sonuçlar. Ayrıca, bu veri setinde "Oyun", "Uygulama" gibi genel bir ayrım olmadığından direkt olarak oyunları göremiyordum. Örneğin, "sports" kategorisi içinde maç uygulamaları da var, futbol oyunları da.

İlk önce verileri tek tek inceledim. İlk olarak, veri üzerinde kullanmayacağımı düşündüğüm bilgilerin bulunduğu sütunları çıkardım ki veri setim daha temiz olsun. Örneğin, "developer Email", "Developer Website"... gibi. Bu bilgiler veriyi incelememde birşey katmayacağı için bunların hepsini kaldırdım. Verime düzgün bir şekilde genel incelemek için ise, info(), describe() gibi yöntemlere başvurdum. Bu incelemeler sonrası aklıma gelen soruları sırayla cevaplamaya başladım ve bu cevaplar veri üzerindeki çıkarımlarıma yardım etti.

Öncelikle uygulamaların hangi kategoride daha çok olduğunu bulmak için kategorilerine göre sıraladım.

Play store'da en çok "Education" kategorisinde uygulama bulunduğunu öğrendim. Bu bilgiden sonra aklıma ilk olarak şu soru geldi, peki en çok indirilen uygulamalar (yani başarılı olanlar) hangi kategoride daha çok. Bunun için kendime referans olarak bir indirme sayısı belirledim. 100 Milyon olarak. Ve bu sayıdan çok olan kategorilere baktım. Sonuç olarak en çok indirmeye sahip olan kategori "Tools" kategorisi. Demek ki insanlar en çok işe yarar uygulamalar kullanmayı tercih ediyorlar.

Mağazadaki indirme sayılarına oranla uygulama sayılarını karşılaştırdım.

Az sayıda indirmesi olan çok fazla uygulama olduğunu gördüm. Bu da başarılı olan uygulamaların azınlıkta olduğunu söyledi.

Kullanıcılar peki, paralı mı yoksa ücretsiz uygulamayı mı daha çok tercih ediyor diye baktığımda ise, Çok büyük oranda ücretsiz uygulama tercih ettiklerini gördüm.

Başarılı uygulamalar için, indirme sayısını 100milyon olarak belirledikten sonra çok fazla veri olduğunu görüp 1milyar olarak güncelledim bu sınırı.

Bu 1 milyar üstü indirme almış uygulamaları listeledikten sonra bunlardan kaç tanesinin oyun olduğu bulmak için kategorilerini kullanarak diğerlerinden ayırdım. ve sadece 2 tanesi oyunmuş. Bu oyunlar kalan başarılı uygulamalara oranını hesapladım ve %2.85'i sadece oyunlardan oluştuğunu öğrendim.

Mağazadaki tüm uygulamaların %1.94'ü ücretli, %98.05'i ise ücretsiz. Ücretlendirmenin indirmeler üzerindeki etkisine bakmak istedim çünkü, yapacak olduğum bir oyun veya uygulamanın ücretlendirmesini belirlerken insanların neyi tercih ettiğini bilmek avantaj sağlayacaktır bana. Sonuç olarak, ücretsiz uygulamalar'ın 1000 kat daha fazla indiriliyor.

Mağaza içindeki yıldız ile puanlamanın indirmelere etkisini merak ettim. Bunun üzerine, öncelikle uygulama indiren insanların yüzde kaçını girip puanlama yapıyor ona bakmak istedim. Tüm kullanıcıların sadece %0.88'i puan vermiş uygulamalara. Yine de, yüksek puan alan ve düşük puan alan uygulamaların indirme oranlarını karşılaştırdım. Düşüğü ve yükseğin sınırını 3 olarak belirledim. 3 yıldız ortalama bir yıldızdır çünkü. Çok yüksek bir fark beklerken buradaki fark sadece 4 katı gibi bir farktı. Bu incelemelerin sonucunda indirme sayısına hem puanlamanın hem de ücretlendirmenin ciddi ölçüde etkisi olduğunu öğrendim.

Makine öğrenmesi için verimi setimi temizlemeye başladım. Bunun için öncelikle sayısal olmayan sütunları çıkardım. Ama bu sette karşılaştırılabileceğim keyifli veriler yok aslında bu yüzden pekte güzel sonuçlar çıkamadı predictionlardan.

Bool değerleri int olarak değiştirdim.

Sonrasında veri setimin içinde NaN değerleri bulup bu satırları silerek veri setimi temizledim. Veri setimi Maximum Installs'a göre değerlendirmek istedim. İndirme üzerindeki tahminler etkiler neler bunları öğrenmek istedim açıkçası. Eğer, veri setimde kazanç olsaydı muhtemelen kazanca etki eden faktörler bana daha çok yardımcı olurdu.

train\_test\_split kütüphanesini alıp veri setimi parçalara ayırdım ve büyük kısmını train için kalan kısmı da testi için ayırdım.

1.717.545 veri train için

572.516 veri ise test için

Linear Regressyon metodu uyguladım, önce veriyi fitleyip sonrasında train verisini kontrol ettim.

Test verileriyle predict oluşturarak bu predictlerin tutarlılığını görebilmek için görselleştirme metodu uyguladım. Verilerin bir kısmı tutarlı, bir kısmı tutarsız sonuç verdi. Dediğim gibi veri setimin bu test için verimsizliğinden kaynaklandığını düşünüyorum. Ardından Decision Tree metodu uyguladım fakat fitedikten sonra prediction verirken Notebook memory hatası verdiği için sonucunu göremedim accuracy değerini vs. Bu yüzden son satırlarımdaki sonuç görünmüyor ama kodlar yazılı. Hata fotoğrafını da ekte görebilirsiniz.

Bitirme Projesi Draft saved

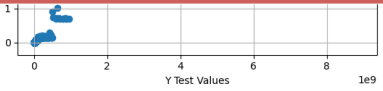
File Edit View Run Add-ons Help

Share Save Version 3

+ Cancel Run Code

Draft Session (4h:58m)

Your notebook tried to allocate more memory than is available. It has restarted.



A scatter plot showing 'Y Test Values' on the x-axis (ranging from 0 to 1e9) and an unlabeled y-axis (ranging from 0 to 1). The data points are clustered into two distinct groups: one near the origin (0,0) and another near the top-left (low x, high y).

[52]:  
from sklearn.tree import DecisionTreeClassifier

[53]:  
classifier = DecisionTreeClassifier(max\_depth = 6)

[54]:  
classifier.fit(X\_train, y\_train)

[54]:  
DecisionTreeClassifier  
DecisionTreeClassifier(max\_depth=6)

[55]:  
y\_pred2 = classifier.predict(X\_train)

[56]:  
from sklearn.metrics import accuracy\_score