BİL 401 - Büyük Veriye Giriş Proje Raporu

Arda Ege İsker	Taner Furkan Göztok
Mühendislik Fakültesi Bilgisayar Mühendisliği	*Mühendislik Fakültesi Bilgisayar Mühendisliği*
TOBB Ekonomi ve Teknoloji Üniversitesi	*TOBB Ekonomi ve Teknoloji Üniversitesi*
Söğütözü, Çankaya Cd. No:43 06510, Çankaya/Ankara	*Söğütözü, Çankaya Cd. No:43 06510, Çankaya/Ankara*
aisker@etu.edu.tr	tfgoztok@etu.edu.tr

Öz

1997-2018 yılları arasında Amazon.com'a konulan elektronik ürünlerin incelendiği veri kümesi kullanılarak anlamlı veriler elde etmek.

I. Giriş

Apache Spark teknolojisi ile kullanılarak, BİL401 dersinde öğrenilen bilgiler dahilinde, dağıtık veri işlemesi teknolojisiyle sahip olduğumuz "Amazon Electronic Product Reviews" veri setine anlam kazandırılmaya çalışılmıştır. Docker ortamında çalıştırılan bu proje, veri setiyle ilgili çıkarımlar yapmamıza olanak sağlamış, grafik ve listelere dökülen tablolarla herkesin anlayabileceği duruma getirilmiştir.

II. İlgili Projeler

Amazon ürün incelemeleri üstüne bir çok açık kaynaklı veri projeleri mevcut [1] [2] [3].Bu projelerin bir çoğu makine öğrenmesi alanında ürünlerin sınıflandırılması, tavsiye mekanizmaları ya da ürünün skoruna dayalı tahmin projeleri. Bu proje ile benzer yönleri hepsinin sabit ve yapılı bir veri kümesi üstünde işlem yapıyor olması.

III. Problem Açıklaması

Amazon üzerinde bulunan elektronik ürünlerin 1997-2018 yılları arasındaki 20 milyondan fazla incelemeye anlam kazandırılması, ve bu işlemin kısa bir sürede verimli metotlarla yapılması gerekmektedir. Böylece kullanıcılar ve ürünler hakkında bilgi edilebilir, bu bilgiler pazarlanabilir veya ürün üreticilerinin örnek alması gereken toplu bir şemaya dönüştürülebilir. Projenin tek amacı bunlarla kalmayıp; öğrenciler için dağıtık veri yapısı, sanallaştırma gibi teknolojilerin nasıl kullanılabileceği hakkında bilgi sahibi olmasını sağlamaktır.

IV. Veri Kümesi Kaynağı

Projemizde kullandığımız veri kümesini San Diego Üniversitesi'nin 2018 yılında gerçekleştirdiği açık kaynaklı veri madenciliği projesinden elde ettik. Veri .JSON formatında yapılı bir veri, içerisinde toplam 20994353 review bulunmaktadır. Dosyanın toplam boyutu 11.51 GB 'tır. Veri kümesi bağlantı linkinden indirilebilir. Her kategori için ayrıştırılmış incelemelerin bulunduğu indirme opsiyonu vardır, ayrıca verinin işlenmemiş hali, aynı incelemelerin teke düşürülmüş hali, en az 5 incelemeye sahip ürünlerin filtrelenmiş hali gibi işlenmiş veri kümeleri de vardır. Bu projede 1997-2018 yılları arasındaki elektronik kategorisindeki işlenmemiş hali kullanılmıştır.Veri kümesini tıklayarak indirebilirsiniz.

V. Kullanılan Teknolojiler

Projede Apache Spark (Pyspark), Jupyter Notebooks on Docker ve Python kullanılmıştır. Apache Spark dağıtık veri yapısı oluşturma ve veri işlemleri için, Jupyter Notebooks bu işlemlerin sanal bir ortamda istenilen cihaz kaynaklarının kolayca ayarlanabilmesi için, Python da ana programlama dili olarak kullanılmıştır.

VI. Dataflow Diagramları

Proje tek bir veri seti üzerinden hazırlanmış olup, tablolar sabittir. Bundan ötürü bir Dataflow bulunmamaktadır. Ancak veri setinin şeması aşağıda görülebilir.Veri kümesi her ürünün bir kimliği niteliğinde olan "ASIN" adlı kolon vardır. Projede inceleme - ürün dönüşümü, incelemede yer alan ASIN degeri ile sağlanır.

```
[('asin', 'string'),
 ('image', 'array<string>'),
 ('overall', 'double'),
 ('reviewText', 'string'),
 ('reviewTime', 'string'),
 ('reviewerID', 'string'),
 ('reviewerName', 'string'),
 ('style',
  'struct<Capacity::string,Color Name::string,Color::string,Colorj::string,Colour::string,Configuration::string,Des
ign::string,Digital Storage Capacity::string,Edition::string,Flavor Name::string,Format::string,Grip Type::string,I
tem Display Length::string,Item Package Quantity::string,Length Range::string,Length::string,Material Type::string,
Metal Type::string,Model Name::string,Model Number::string,Model::string,Number of Items::string,Offer Type::string
,Package Quantity::string,Package Type::string,Pattern::string,Platform for Display::string,Platform::string,Preamp
lifier Output Channel Quantity::string,Processor Description::string,Product Packaging::string,Service plan term::s
tring,Shape::string,Size Name::string,Size::string,Style Name::string,Style::string,Team Name::string,Tension Suppo
rted::string,Wattage::string,Width::string,processor_description::string,style::string,style_name::string>'),
 ('summary', 'string'),
 ('unixReviewTime', 'bigint'),
 ('verified', 'boolean'),
 ('vote', 'string')]
```

VII. Hesaplama Algoritmalarının Detayları

Hesaplamalarımızı üç farklı zaman dilimi için gerçekleştirdik. Bunlar;

1) Tüm Zamanlar
2) Yıl Bazlı Hesaplamalar
3) Mevsim Bazlı Hesaplamalar

Tüm zamanları içeren hesaplamaları yaparken tüm veri kümesini bir "Spark Dataframe" 'e dönüştürüp bu dataframe üstünden hesaplamaları gerçekleştirdik. Yıl bazlı hesaplamalar yapılırken Dataframe'e yeni bir kolon "year" kolonu eklenmiştir. Bu kolon, veri setinde hali hazırda bulunan reviewTime kolonu kullanılarak yapılmıştır. Öncelikle string olarak tutulan reviewTime kolonu date'e çevrilmiş, date'e çevrildikten sonra her bir satırın yılı "year" kolonuna yazılmış ve groupBy işleminde kullanılmıştır.Mevsim bazlı hesaplamaları yaparken ise Spark SQL kütüphanesinden yararlanılarak SQL'in WHERE sorgusu ile mevsimlere ait yıllar bir aralık olarak verilmiştir.

```
#Reformatting reviewTime column, String to Date in order to make things work efficiently and well.
from pyspark.sql.types import IntegerType

df = df.withColumn(
            'reviewTime',
                F.to_date(
                    F.unix_timestamp('reviewTime', 'MM d, yyyy').cast('timestamp')))\
        .withColumn(
            'vote',
                df["vote"].cast(IntegerType())
        )
```

Yukarıda görülen kod ile string'den date'e çevirme işlemi gerçekleştirilmiştir. Daha sonra da yeni oluşturulan bir Dataframe'e "year" kolonu eklenmiş ve groupBy işlemleri bu kolon kullanılarak yapılmıştır.

```
adjustedDf = df.withColumn("reviewTime", to_date(col("reviewTime"), "yyyy-MM-dd"))\
    .withColumn('year', year("reviewTime"))
```

Mevsim bazlı filtreleme işlemi yaparken ise :

```
spark.sql("SELECT * FROM TABLE where reviewTime between '"+str(year) +"-01-01' and '"+str(year) +"-03-01' ")
```

SQL sorguları ile yapılıyor.

Çıktı olarak üretilen ürünlerin URL dönüşümleri "https://www.amazon.com/dp/" + Ürünün ASIN numarası eklenerek hesaplanmıştır.

```
def getProductURL(asin):
    return "https://www.amazon.com/dp/" + asin
```

Bir resmin web bağlantı linki kullanılarak resmin ekrana basılması için Python'un URLLib kütüphanesini kullandık.

```
urllib.request.urlretrieve(imageURLs[i],"image.png")
img = Image.open("image.png")
img.show()
```

"En" lere dair hesaplamaları , ürünün ASIN numarası ile groupBy yapıp daha sonra azalan sırada sıralayarak elde ettik.Daha sonra istediğimiz kadar satır alarak bir listeye atadık. Daha sonra her bir liste elemanını döngü ile gezerek ürüne dair bilgileri kullanıcıya sunduk.

```
voteDF = df.filter(df.vote.isNotNull()) #Task number 15
voteDF = voteDF.withColumn("vote", voteDF["vote"].cast(IntegerType()))
listOftopReviews = voteDF.orderBy('vote', ascending=False).take(10)
topReviewTexts=[]
topReviewTextsAsin=[]

for i in range(10):
    text = listOftopReviews[i].reviewText
    asin = listOftopReviews[i].asin
    topReviewTexts.append(text)
    topReviewTextsAsin.append(asin)

for j in range(10):
    print(str(j+1) + ")")
    print("Product ASIN No:" + topReviewTextsAsin[j])
    print("Product URL: " + getProductURL(topReviewTextsAsin[j]))
    print(topReviewTexts[j])
    print("////////////////////////////////////////////////////")
    print("////////////////////////////////////////////////////")
    print("////////////////////////////////////////////////////")
```

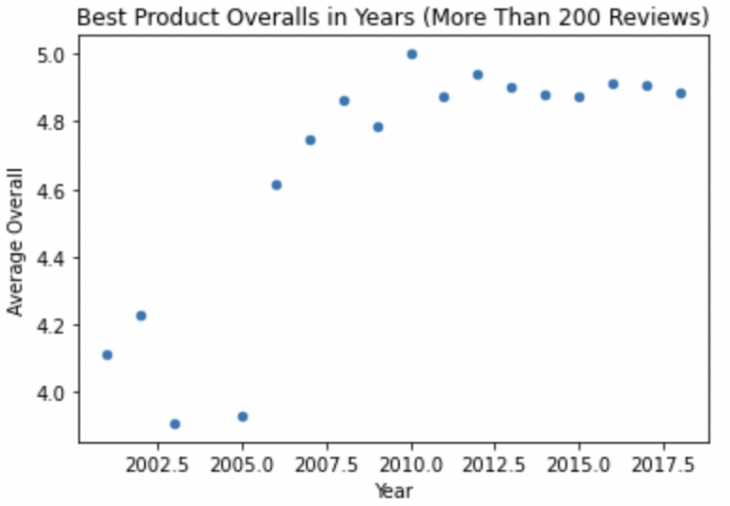
Kullandığımız her kolon string veri türünde olduğu için matematiksel hesaplamalar yapmamız gerektiği zaman string'den tam sayıya dönüşüm gerçekleştirdik.

VIII. Sonuçlar

"Amazon Electronic Product Review" veri kümesinden istenilen sonuçlar elde edilmiş, grafik ve listeler çıkarılmış ve veri setine anlam kazandırılmıştır. Aşağıda, proje kapsamında elde ettiğimiz sonuçları görebilirsiniz.

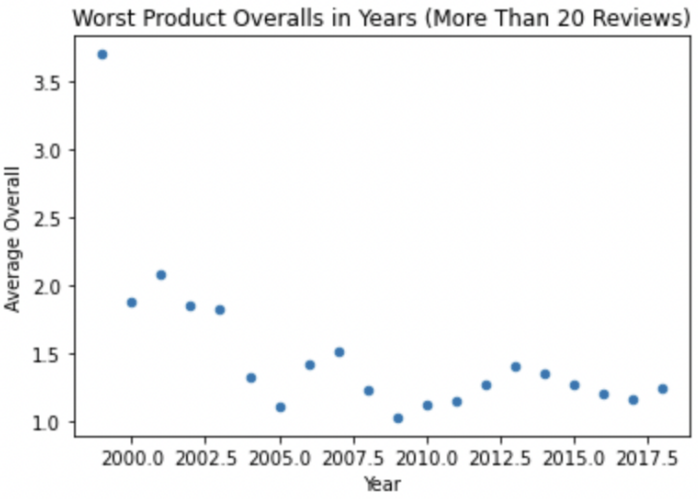
```
#HER YILA GORE EN IYI PUANA SAHIP URUNLERIN LISTESI VE GRAFIGI
```

Year	ASIN	Average Overall
2001	B00004SB92	4.112195
2002	B00004SB92	4.230047
2003	B00005ARK3	3.909091
2005	B00007KDVI	3.933071
2006	B000A3WS84	4.616858
2007	B0002IWC9C	4.744755
2008	B000A5TAT2	4.862595
2009	B00007E7JU	4.784000
2010	B000065BP9	5.000000
2011	B00004ZCJJ	4.876106
2012	B00001WRSJ	4.942222
2013	B00005N6KG	4.901288
2014	B00004T8R2	4.878378
2015	B000051ZOA	4.876251
2016	B00002EQCS	4.910769
2017	B000067S60	4.910000
2018	B00004T8R2	4.887613



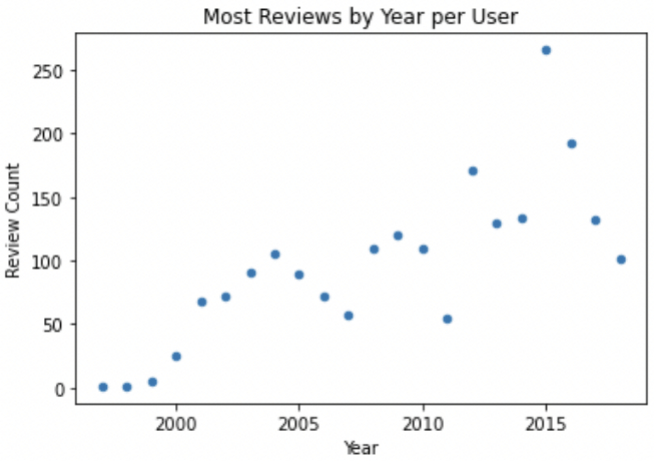
```
#HER YILA GÖRE EN KÖTÜ PUANA SAHİP ÜRÜNLERİN LİSTESİ VE GRAFİĞİ
```

Year	ASIN	Average Overall
1999	B00000JFMK	3.704545
2000	B00000JMUG	1.888889
2001	B00004YRVY	2.090909
2002	B000063V7N	1.857143
2003	B00001P4XA	1.833333
2004	B00000J061	1.333333
2005	B00005A9AU	1.115385
2006	B00004VUGJ	1.428571
2007	B000065BPB	1.521739
2008	B00000J1V3	1.238095
2009	B00004T1XE	1.034483
2010	B00005LEN4	1.125000
2011	B00004WINT	1.150000
2012	1400501466	1.275862
2013	B00004THCZ	1.413793
2014	B00004VX39	1.357143
2015	B000066R6M	1.280000
2016	998504780X	1.214286
2017	B00004SY4H	1.166667
2018	B00004SABB	1.250000



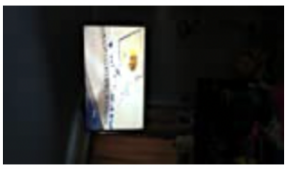
#HER YILIN EN ÇOK REVIEW YAZAN KULLANICISI VE GRAFİĞİ

Year	User ID	Review Count
1997	A3B12RM18TB8OQ	1
1998	A2CT25FY12PYNF	1
1999	APKNTDAV6EPI4	5
2000	AJIC64C630FTT	25
2001	A3NJ1W51XSRC29	68
2002	A30NZYWSJ5WT7I	72
2003	A3GQBTDQYMJYA0	91
2004	A3PWF2AQ5XYALW	106
2005	A7OR8OWWGK55W	89
2006	A221XWNYJZX863	72
2007	A1CQ0FUOKPQZOV	57
2008	A1CKDRO4VYQQQC	110
2009	AVFR88UIGCQS6	120
2010	AESUMN2OKV2H8	110
2011	ALKZQOZM9Y72H	54
2012	A2OFPBKQ6DTVW7	171
2013	A1HBJHGRLTM5XH	129
2014	A3OXZRN6VAXPBG	133
2015	A3HWHZNT6R7B3P	266
2016	A0008052W2M3QH4G1NJZ	192
2017	A1YY8PAYNR1CNG	132
2018	AVRDNNVGWOTZY	102



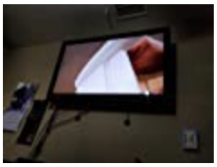
#Fotoğrafı olan ve en çok review alan 5 ürünün resimleri

Image reviews of most reviewed products that has imaged reviews.
1) https://www.amazon.com/dp/B00B5HE3UU

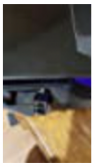


2) https://www.amazon.com/dp/B00B5HE3UU

3) https://www.amazon.com/dp/B00B5HE3UU



4) https://www.amazon.com/dp/B00B5HE3UU



5) https://www.amazon.com/dp/B00B5HE3UU

#ONAYLI ÜRÜNLERİN TÜM ÜRÜNLERE OLAN ORANI

```
Rate of verified products to all products: %88.58140091290262.
```

#2016 KIŞININ EN BEĞENİLEN ÜRÜNÜ

ASIN	Average Overall
B00004T8R2	4.914634

#2016 KIŞININ EN BEĞENİLMEYEN ÜRÜNÜ

ASIN	Average Overall
B00005ATMI	1.222222

Apache Spark aynı özelliklere sahip iki bilgisayardan çalıştırılmıştır (Macbook Air M1). Hesaplama performansı ölçülebilmesi için Docker Container' larına farklı kaynak kombinasyonları verilmiştir. Örneğin bir makinada 6 çekirdekli işlemci ve 8GB bellek konfigürasyonu ile, diğer makinada ise 4 çekirdekli işlemci ve 4GB bellek konfigürasyonu kullanılmıştır.Yürütme zamanında farkındalık gösterilmesini sağlayan en büyük etkenin bellek boyutu olduğu görülmüştür. Bellek boyutunun küçük olduğu ayarlamalarda Apache Spark' ın işlem yaptığı blok sayısının azaldığı, böylece 8GB ve 4GB bellek arasında %30'a yakın performans farkı olduğu gözlemlenmiştir.

Ayrıca bellek boyutunun yetersiz geldiği bazı işlemlerde "RowBasedKeyValueBatch: Calling spill() on RowBasedKeyValueBatch." hatası alınmıştır. Bu hata araştırıldığında ise, Spark' a ayrılan belleğin dolduğunun ve bellek kullanımı gerektiği için bellekte bulunan bazı verilerin diske yazıldığı görülmüştür. Bu da performansı etkileyen en önemli faktörlerden birisi olmuştur. Kullanılan cihazlardaki belleklerin hızı resmi olarak 204GB/s belirtilirken, disk hızı ortalama olarak 2400MB/s olarak belirtilmiştir.

IX. References

1. https://github.com/minimaxir/amazon-spark
2. https://github.com/Currie32/Text-Summarization-with-Amazon-Reviews
3. https://github.com/aws-samples/amazon-codeguru-reviewer-sample-app
4. https://spark.apache.org/docs/latest/api/python/
5. https://sparkbyexamples.com/pyspark-tutorial/
6. https://hub.docker.com/r/jupyter/pyspark-notebook
7. https://docs.docker.com/desktop/mac/install/
8. https://jupyter-notebook.readthedocs.io/en/stable/