Türkçe Metinlerde Duygu Analizi

Abdullatif Köksal

Abdullatif Köksal

Blog(İngilizce): akoksal.com

Konuşma Özeti

- Duygu Analizi
- Gözetimli Makine Öğrenmesi
- Doğal Dil İşleme
    a. Elle Oluşturulmuş Öznitelikler
    b. Word2Vec
    c. BERT
- Analizler

Duygu Analizi(Sentiment Analysis)

Duygu analizi aslında bir tavır tespitidir.

1. **Söyleyenin** tavrı
2. **Hedefe** olan tavır
3. Tavrın **türü**
    a. Sevgi, aşk, nefret, değer verme, istek duyma gibi olabilir.
    b. Daha sık olarak tavrın kutuplarını tespit etme: pozitif, negatif, nötr
4. Tavrı içeren **metin**
    a. Cümle veya döküman

https://web.stanford.edu/~jurafsky/slp3/slides/7_Sent.pdf Sayfa 13

Duygu Analizi

Basitçe metinde yazarın aktardığı duygunun *pozitif-negatif-nötr* diye sınıflandırılması.

Pozitif

@TK_TR @TurkishAirlines 13.02.2020 09:25 Malaga uçuşu, Tek.San. istedik gelen iki görevli mükemmel görev yaptılar, Uçaktaki Onur bey de harikaydı, bu üç personelinizi lütfen ödüllendirin,bu insanların sayısı artmalı, gurur duyuyoruz @TK_TR

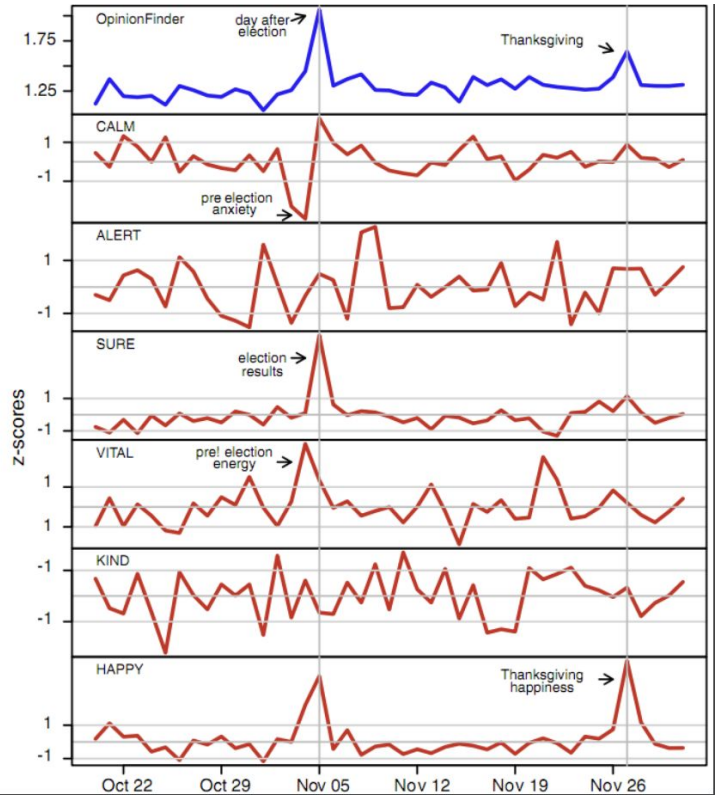
Negatif

Ayni rezervasyon kodundaki iki kisiye bilerek farkli koltuklar atayip parayla satin almaya zorlayan havayolu #thy #thyoa #thypismanliktir @TK_TR @TK_HelpDesk @TurkishAirlines

Duygu Analizi

Twitter sentiment:

Johan Bollen, Huina Mao, Xiaojun Zeng. 2011.
Twitter mood predicts the stock market,
Journal of Computational Science 2:1, 1-8.
10.1016/j.jocs.2010.12.007.

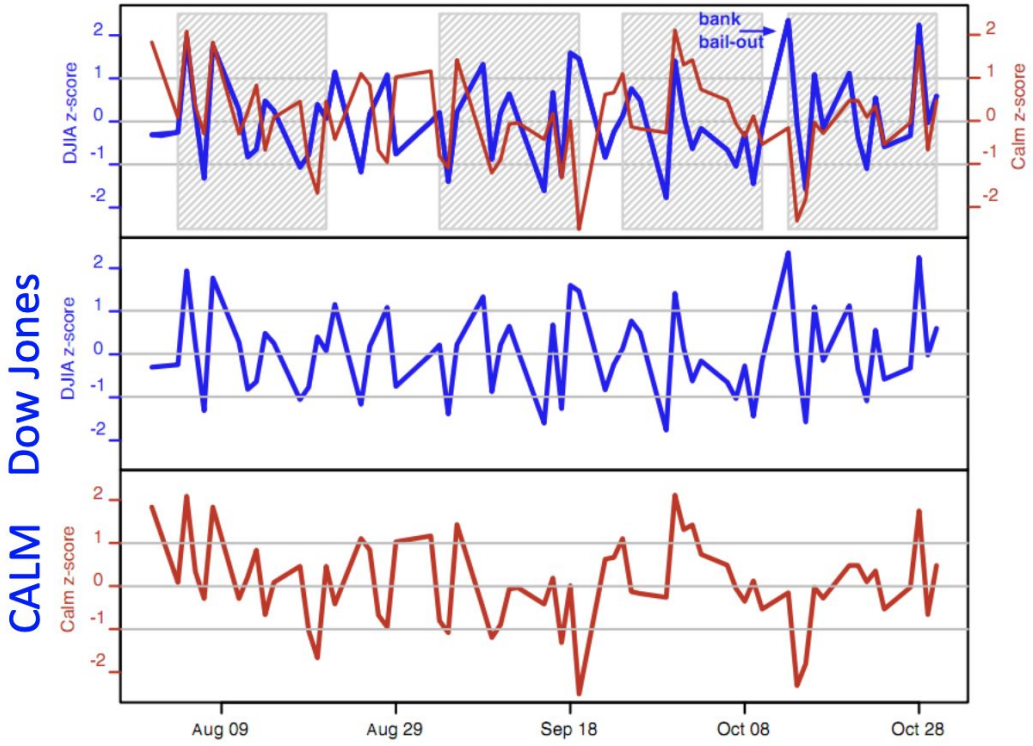


https://web.stanford.edu/~jurafsky/slp3/slides/7_Sent.pdf

Duygu Analizi

Bollen et al. (2011)

- CALM predicts DJIA 3 days later
- At least one current hedge fund uses this algorithm



https://web.stanford.edu/~jurafsky/slp3/slides/7_Sent.pdf

Türkçe Veriler

- Model oluşturabilmek için pozitif-nötr-negatif şeklinde ayrılmış verilere ihtiyaç var.
- Bu veriler ürün ve film reytinglerine göre otomatik olarak toplanabilir.

2018 Boğaziçi Verisi

- Twitter verisi olarak: 2018 Boğaziçi Verisi*
- Boğaziçi Üniversitesi’nde Information Retrieval(CMPE 493) dersinde 17 öğrenci grubu tarafından içinde ağırlıklı olarak Boğaziçi geçen twitler iki aşamalı olarak etiketlendi.
- 5733 training, 639 validation, 1592 test verisi mevcut
- %52 nötr, %30 pozitif, %18 negatif

*Abdullatif Köksal, Arda Çelebi, Arzucan Özgür, CMPE 493 - 2018 dönemi öğrencileri

2018 Boğaziçi Verisi

Pozitif:

Negatif:

Gözetimli Makine Öğrenmesi (Supervised)

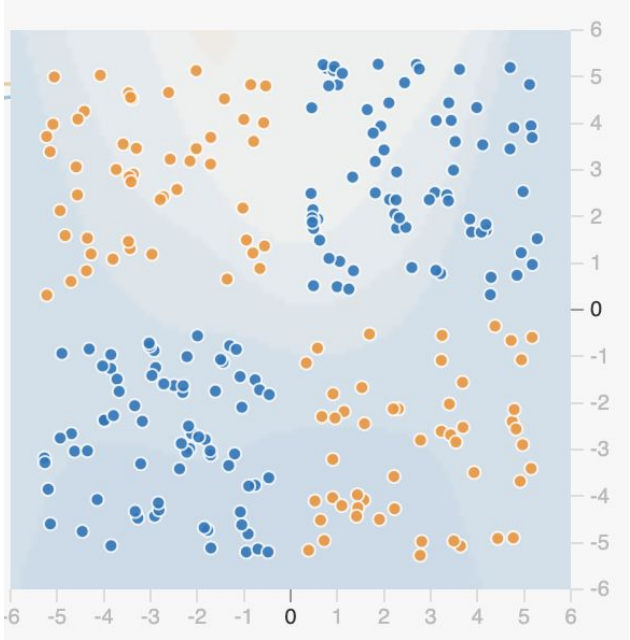
- Gözetimli öğrenme, örnek üzerinden öğrenmeye verilen isimdir.
- Örnek veriler üzerinden modellenir.
- Model yeni bir veriyle karşılaştığında eğitim verisi üzerinden öğrendiği bilgiyle tahminde bulunur.

Gözetimli Makine Öğrenmesi (Supervised)

- Gözetimli öğrenme, örnek üzerinden öğrenmeye verilen isimdir.

Basitçe:

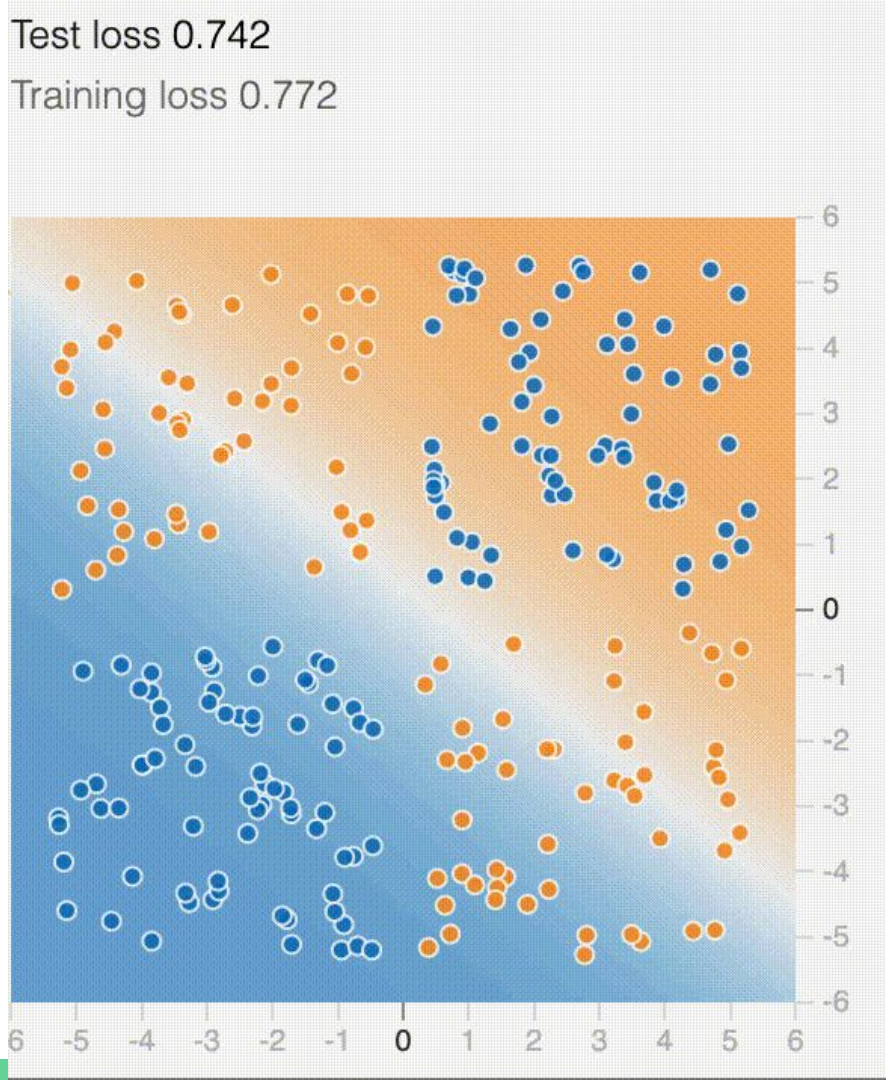
1. Veriler vektörel olarak ifade edilir.
2. Farklı modellemeler kullanılarak verilerden öğrenilir.



https://playground.tensorflow.org/

Gözetimli Makine Öğrenmesi

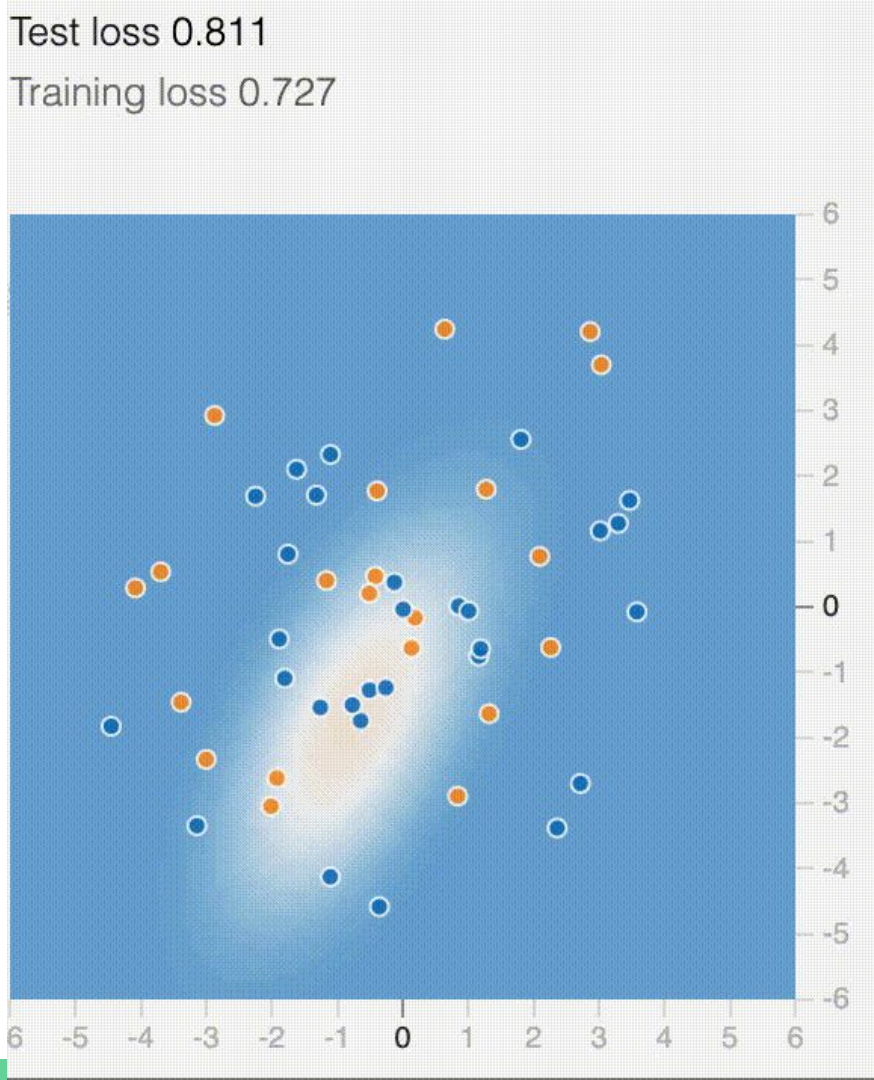
- Verimizi iki boyutlu bir vektör olarak ifade edebiliyoruz.
- **Kötü bir model**le öğrenmeye çalışsak:



https://playground.tensorflow.org/

Gözetimli Makine Öğrenmesi

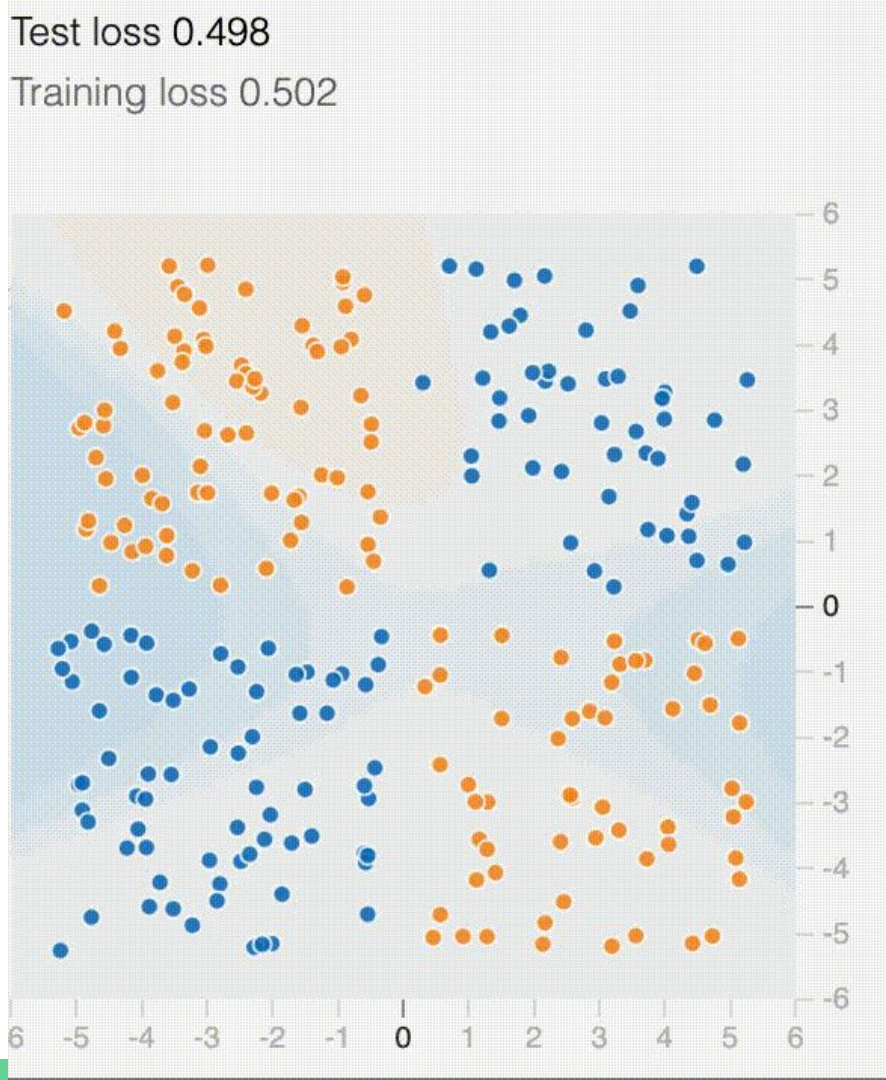
- Verimizi iki boyutlu bir vektör olarak ifade edebiliyoruz.
- **Kötü bir veri yapısı**yla öğrenmeye çalışsak:



https://playground.tensorflow.org/

Gözetimli Makine Öğrenmesi

- Verimizi iki boyutlu bir vektör olarak ifade edebiliyoruz.
- **Uygun veri ve uygun model**le yapsak:



https://playground.tensorflow.org/

Doğal Dil İşleme

- Doğal Dil İşleme tanım olarak büyük ölçekteki doğal dil verisinin bilgisayar tarafından anlaşılıp işlenmesidir.
- Alt dalları:
  - Makine Çevirisi
  - Soru Cevaplama
  - Metin Özetleme
  - Duygu Analizi
  - ...

Doğal Dil İşleme

Doğal Dil İşleme'deki bir sınıflandırma probleminin, bir Gözetimli Makine Öğrenmesi problemi olarak ayrıca ilgilenmesi gereken konular:

- Değişken uzunluktaki cümleler ve dökümanlar için nasıl modeller kullanabiliriz?

- Kelime ve cümleleri vektörlerle nasıl ifade edebiliriz?

Doğal Dil İşleme

- Değişken uzunluktaki cümleler ve dökümanlar için nasıl modeller kullanabiliriz?

Finansal verilerde de kullanılan değişken uzunluktaki veriler için çeşitli modeller geliştirilmiştir.

Doğal Dil İşleme için bu modeller özelleştirilmiştir. Detaylar için:

- RNN (LSTM, GRU)
- Attention Mechanism
- Transformers

Doğal Dil İşleme

- Kelime ve cümleleri vektörlerle nasıl ifade edebiliriz?

Bir cümleyi sabit uzunluklu bir vektörle nasıl ifade ederiz?

Bir kelimenin anlamını cümleden bağımsız olarak bir vektörle nasıl ifade ederiz?

Bir kelimenin anlamını cümleye bağlı olarak sayısal bir vektörle nasıl ifade ederiz?

Doğal Dil İşleme

- Kelime ve cümleleri vektörlerle nasıl ifade edebiliriz?

1. Elle Oluşturulmuş Öznitelikler
2. Word2Vec
3. BERT

Google Colab

Google Colab, Jupyter Notebook’ların Google Drive üzerinden ekran kartlı bir makinede kullanılmasına olanak tanıyan ücretsiz bir servistir.

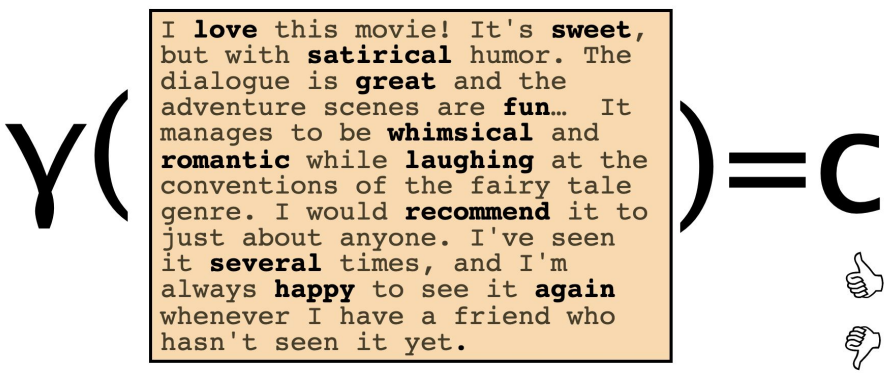
https://colab.research.google.com/

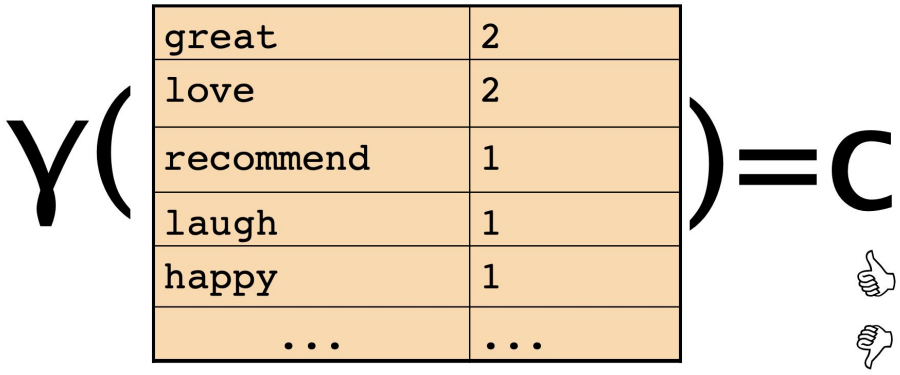
3 farklı yaklaşımı da Google Colab üzerinden çalıştırıp göreceğiz.

Elle Oluşturulmuş Öznitelikler

2013 yılına kadar DDİ'deki sınıflandırma problemlerinde, elle oluşturulmuş öznitelikler sıklıkla kullanılıyordu.

Bag of Words:





Resimler: https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

SentiTurkNet - Elle Oluşturulmuş Öznitelikler

- Şimdi örnek bir kod çalışmasıyla deneyeceğiz.

- SentiTurkNet'te yaklaşık 15000 kelimenin polarity skorları mevcut.

	pos	neg
güzel	1.000	0.000
çirkin	0.062	0.731
iyi	1.000	0.000
kötü	0.036	0.946
müthiş	0.068	0.720
berbat	0.000	1.000

SentiTurkNet: http://myweb.sabanciuniv.edu/rdehkharghani/sentiturknet-3/

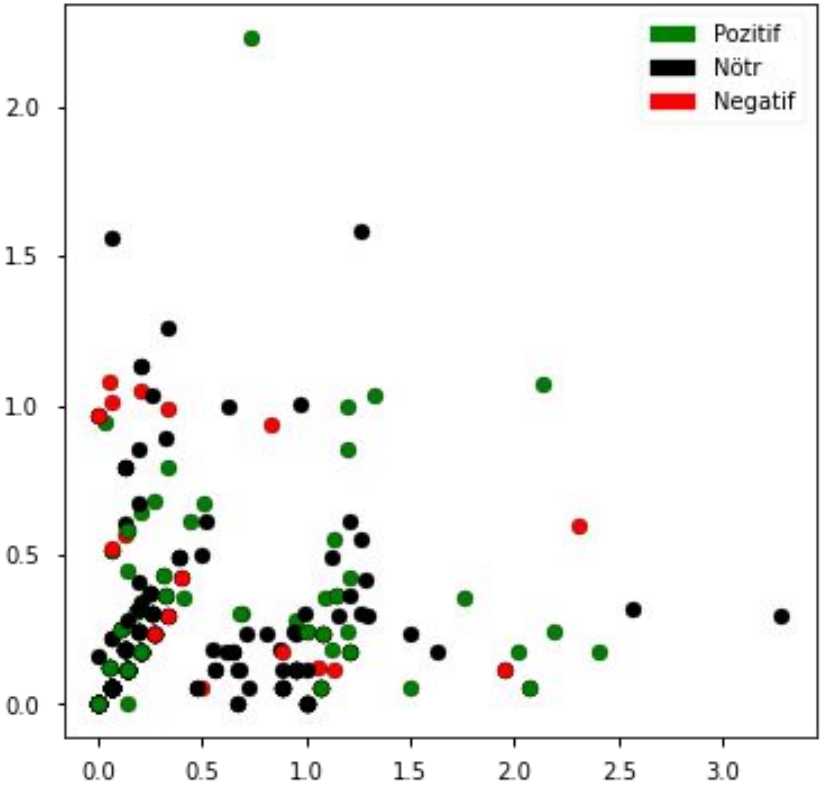
Elle Oluşturulmuş Öznitelikler

SentiTurkNet kullanılarak cümlenin pozitif kelime skorları ve negatif kelime skorları toplanacak.

```
def feature_extraction(text):
    pos_val = 0
    neg_val = 0
    for token in nlp(text):#splitting sentence into words by nlp(text)
        word = token.text.lower()
        if word in final_stn:
            pos_val+=final_stn[word]['pos']
            neg_val+=final_stn[word]['neg']
    return [pos_val, neg_val]
```

Elle Oluşturulmuş Öznitelikler

Cümle vektörleri:



Elle Oluşturulmuş Öznitelikler

- Skorlar

	Pozitif	Nötr	Negatif	**Ortalama**
Elle Oluşturulmuş	0.04	0.94	0.09	0.36

- Çok az kelimenin puanını bildiğimiz için cümleler iyi bir şekilde temsil edilemedi.

- Farklı bir şekilde cümleleri temsil ederek başarıyı arttırmamız gerekir.

Word2Vec

2013 yılında çıkan Word2Vec makalesi, benzerleri ile beraber alanda çok sık kullanılmaya başlandı. Çıkan model, kelimeleri anlamlarını barındıracak şekilde sayısal vektörlerle ifade edilmesini sağlıyor.

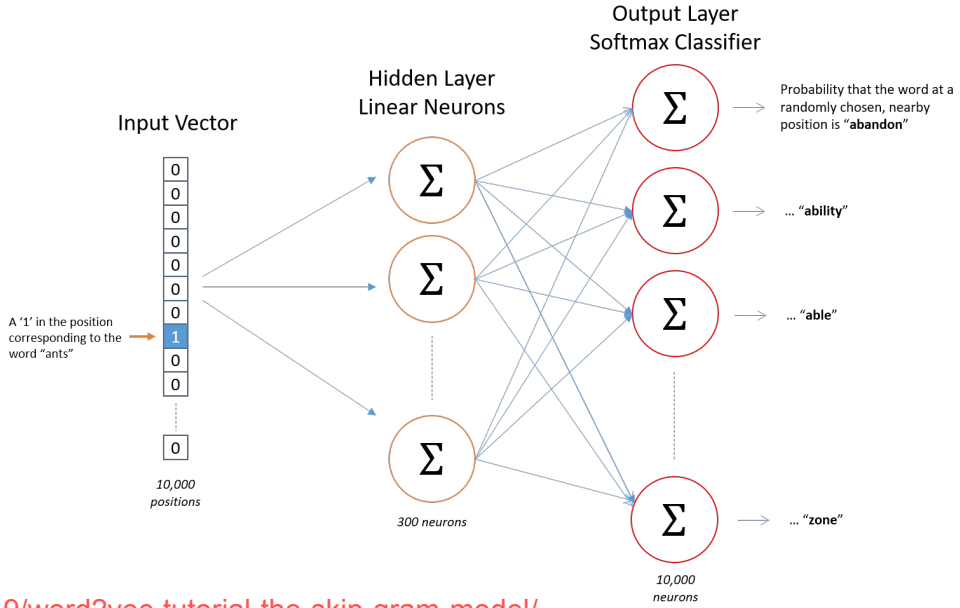
2019 yılına kadar çoğu DDİ projesi Word2Vec ve türevleri kullanılarak yazıldı.

Fikir: Dilin kendi yapısındaki anlamı kullanarak kelimeleri temsil etmeye çalışmak.

Word2Vec

Sahte Hedef: Bir kelimeyi çevresindeki kelimelerle tahmin etmeye çalışmak.

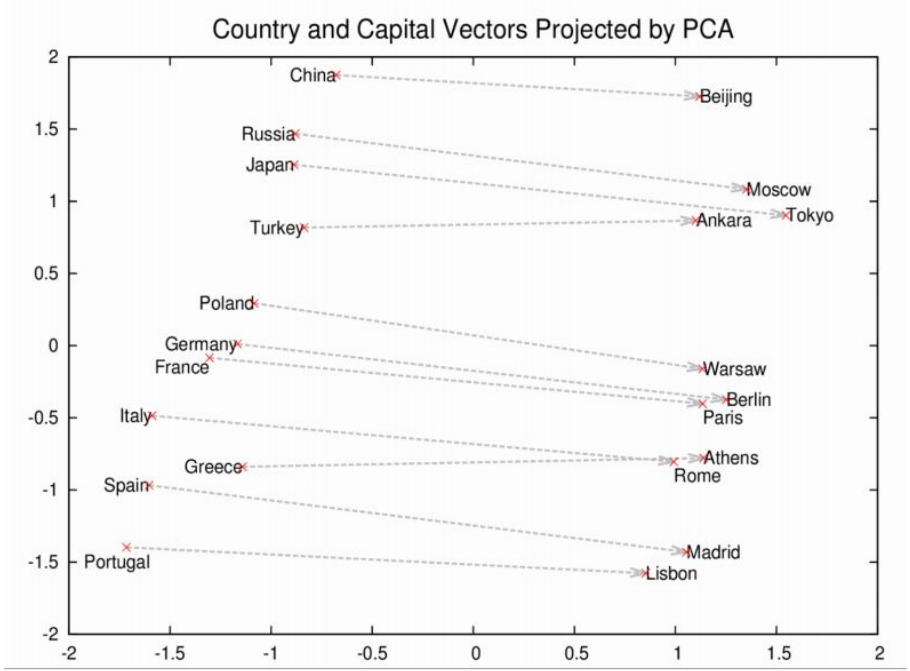
Esas Hedef: Kelimeyi temsil eden vektörleri bulmak.



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Word2Vec

Çıktı: Çevresindeki kelimelerin dağılımı üzerinden her kelimeyi temsil eden sabit uzunlukta bir vektör.



https://pathmind.com/wiki/word2vec

Word2Vec

Çıktı: Çevresindeki kelimelerin dağılımı üzerinden her kelimeyi temsil eden sabit uzunlukta bir vektör.

```
word_vectors.most_similar('kral')

[('kralı', 0.6476399898529053),
 ('kraliçe', 0.6076635718345642),
 ('kralın', 0.6053656935691833),
 ('prens', 0.5873309373855591),
 ('veliaht', 0.5570374131202698),
 ('hükümdar', 0.5513037443161011),
 ('tahtı', 0.5475605726242065),
 ('imparator', 0.5394125580787659),
 ('taht', 0.5144139528274536),
 ('kralların', 0.5098164081573486)]
```

```
word_vectors.doesnt_match(
  ['elma', 'ahududu', 'armut', 'kalem'])

'kalem'
```

https://github.com/akoksal/Turkish-Word2Vec

Word2Vec

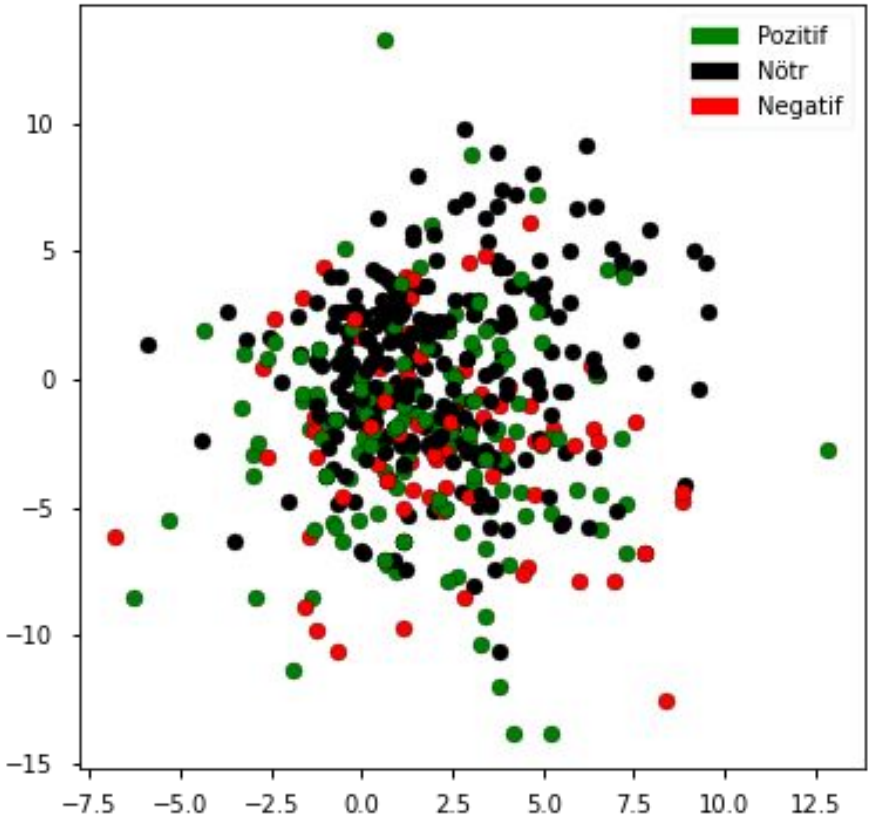
Cümlenin vektörünü kelime vektörlerinin toplamı olarak temsil ediyoruz.

```
from gensim.models import KeyedVectors
word2vec = KeyedVectors.load_word2vec_format(
        '/gdrive/My Drive/Turkish NLP/Word2Vec/trmodel',
        binary=True
        )

def feature_extraction(text):
    vector = np.zeros(400)
    for token in nlp(text):
        word = token.text.lower()
        if word in word2vec:
            vector+=word2vec[word]
    return vector
```

Word2Vec

Cümle vektörleri:



Aslında vektörler 400 uzunluğunda fakat görselleştirmek için 2 boyuta indirgendi

Word2Vec

- Skorlar

	Pozitif	Nötr	Negatif	**Ortalama**
Elle Oluşturulmuş	0.04	0.94	0.09	0.36
Word2Vec	0.37	0.69	0.47	0.51

- Kelimeleri daha iyi ve kelimelerin büyük bir kısmını temsil ettiğimiz için başarı ciddi bir şekilde arttı.

- Cümleleri daha iyi temsil etmek için RNN yapısı veya kelimelerin cümle içerisindeki değişimini gösteren bir model kullanmamız gerekiyor.

BERT

BERT 2018 yılının sonunda Google tarafından çıkarılan, **cümleleri** kelime anlamlarına uygun bir şekilde temsil eden kompleks bir eğitilmiş modeldir.

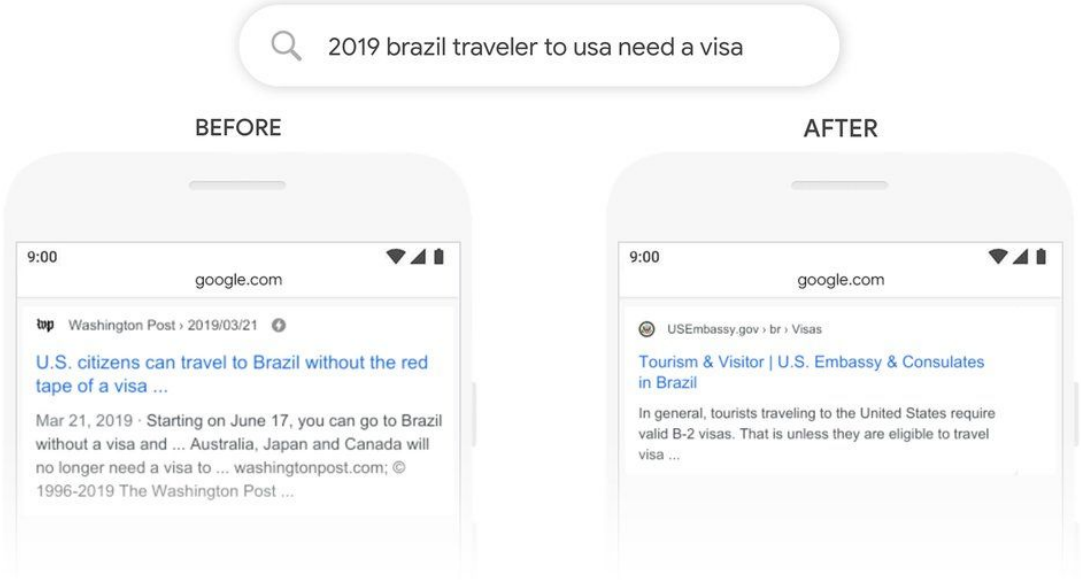
Cümleyi çok iyi bir şekilde temsil edebildiği için neredeyse DDİ'nin tüm alt alanlarında en iyi başarımı gösterdi.

Rank	Model	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	QNLI	RTE
1	BERT: 24-layers, 1024-hidden, 16-heads	80.4	60.5	94.9	85.4/89.3	87.6/86.5	89.3/72.1	86.7	91.1	70.1
2	Singletask Pretrain Transformer	72.8	45.4	91.3	75.7/82.3	82.0/80.0	88.5/70.3	82.1	88.1	56.0
3	BiLSTM+ELMo+Attn	70.5	36.0	90.4	77.9/84.9	75.1/73.3	84.7/64.8	76.4	79.9	56.8

https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

BERT

Google Arama’ya da makaleyle beraber entegre edildiği açıklandı:



https://www.blog.google/products/search/search-language-understanding-bert/

BERT

Sahte Hedef:

1. Cümlede maskelenmiş veya değiştirilmiş kelimeleri tahminlemek.
2. Bir cümlenin diğer cümlenin ardından gelip gelmediğini tahminlemek.

Esas Hedef:

Kelimelerin anlamını ve sırasını dikkate alacak şekilde **cümleleri** temsil eden, sabit uzunluklu bir vektör oluşturmak.

BERT

Cümleyi kelimelere ve alt kelimelere bölme - **BertTokenizer**
Cümleyi kelimelerin sırasını da dikkate alarak sabit uzunluklu bir vektör olarak temsil etme - **BertModel**

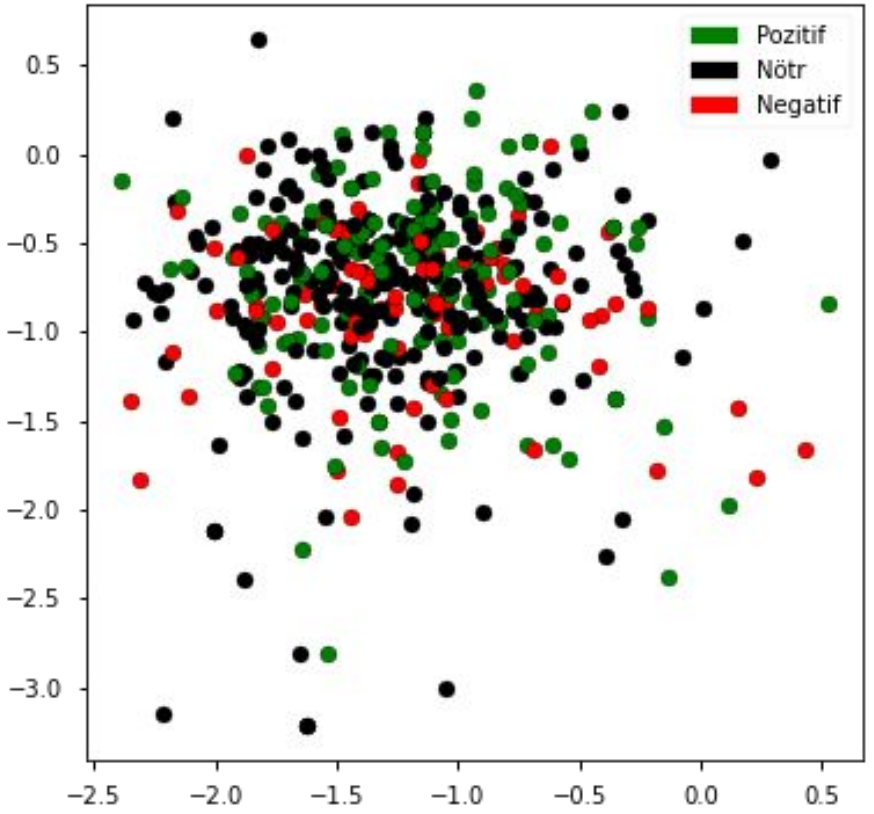
```
tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-128k-uncased")
bert = AutoModel.from_pretrained("dbmdz/bert-base-turkish-128k-uncased").to('cuda')

def feature_extraction(text):
    x = tokenizer.encode(filter(text))
    with torch.no_grad():
        x, _ = bert(torch.stack([torch.tensor(tokenizer.encode(text))]).to('cuda'))
        return list(x[0][0].cpu().numpy())
```

https://github.com/stefan-it/turkish-bert

BERT

Cümle vektörleri:



Aslında vektörler 768 uzunluğunda fakat görselleştirmek için 2 boyuta indirgendi

BERT

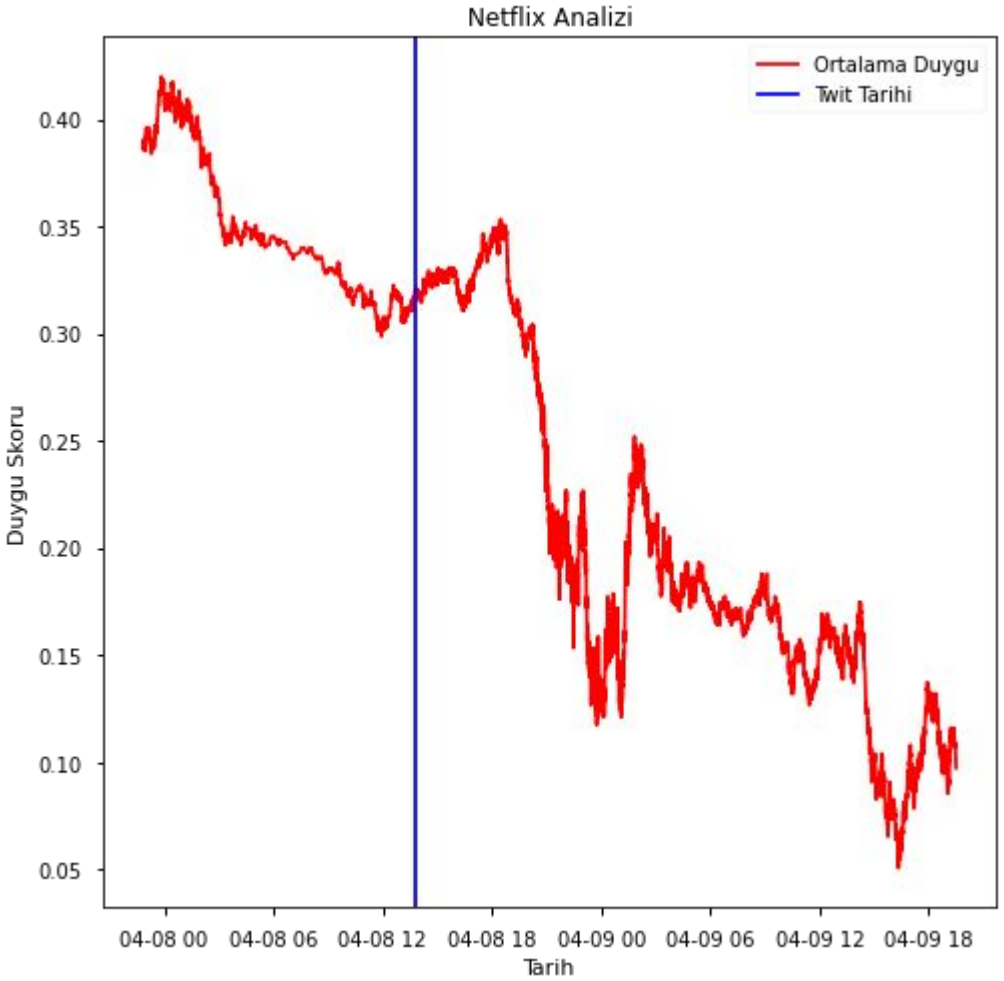
- Skorlar

	Pozitif	Nötr	Negatif	**Ortalama**
Elle Oluşturulmuş	0.04	0.94	0.09	0.36
Word2Vec	0.37	0.69	0.47	0.51
BERT	0.53	0.76	0.67	0.65

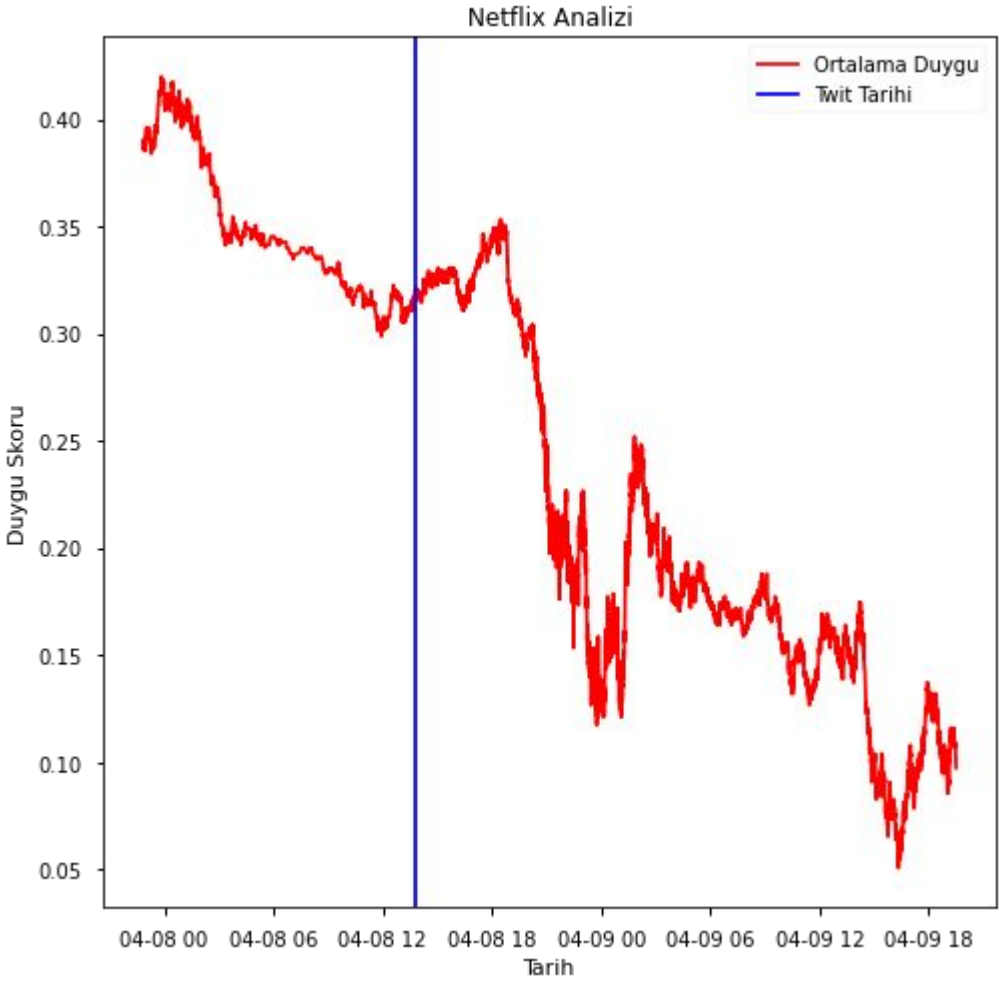
- Cümleyi kelimelerin sırasına da uygun bir şekilde kompleks bir modelle temsil etmiş olduk.

Netflix Analizi

Netflix hakkında atılmış Türkçe twitlerin ortalama skorları:

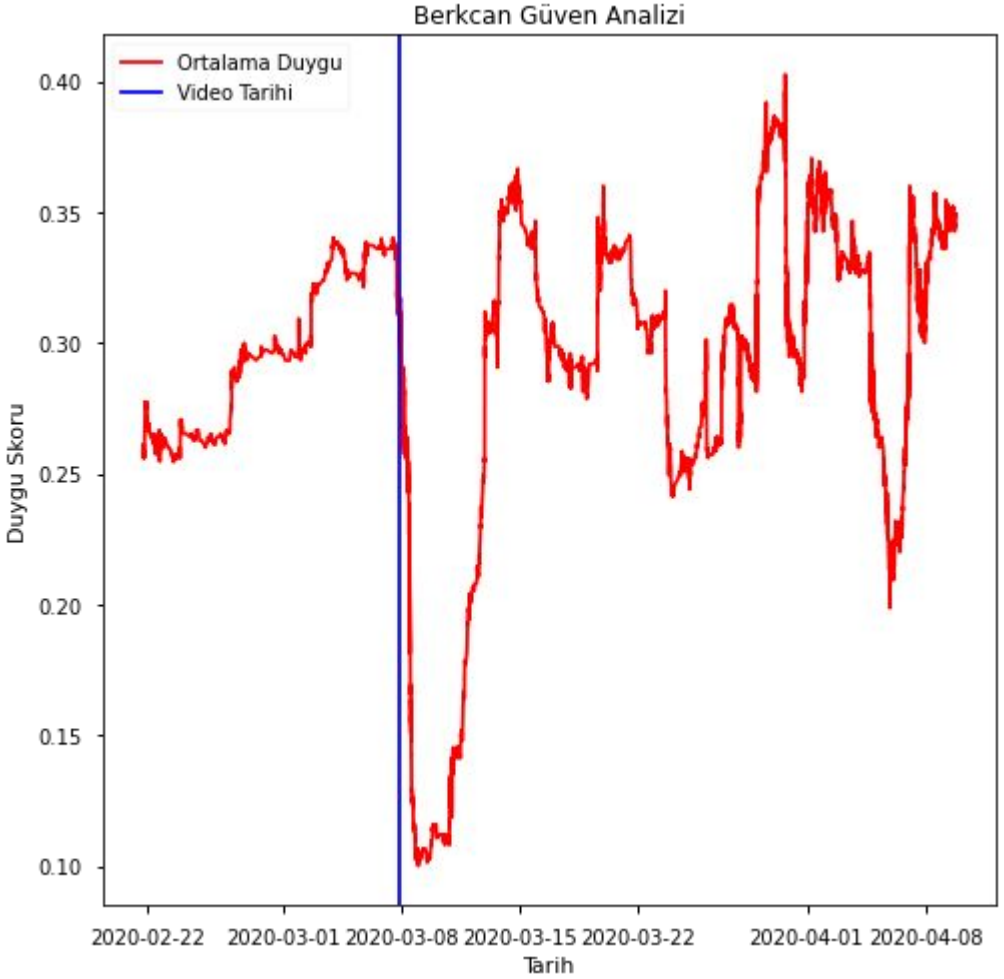


Netflix Analizi

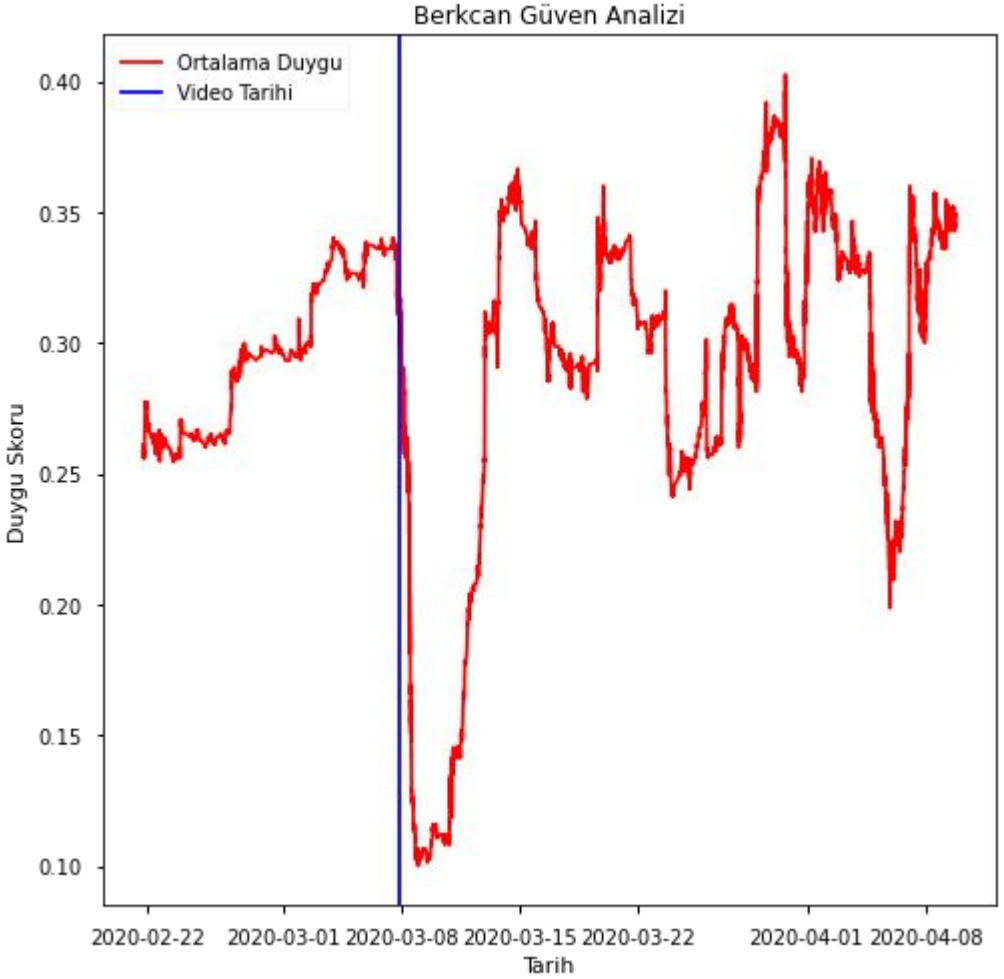


Berkcan Güven Analizi

Berkcan Güven hakkında atılmış Türkçe twitlerin ortalama skorları:

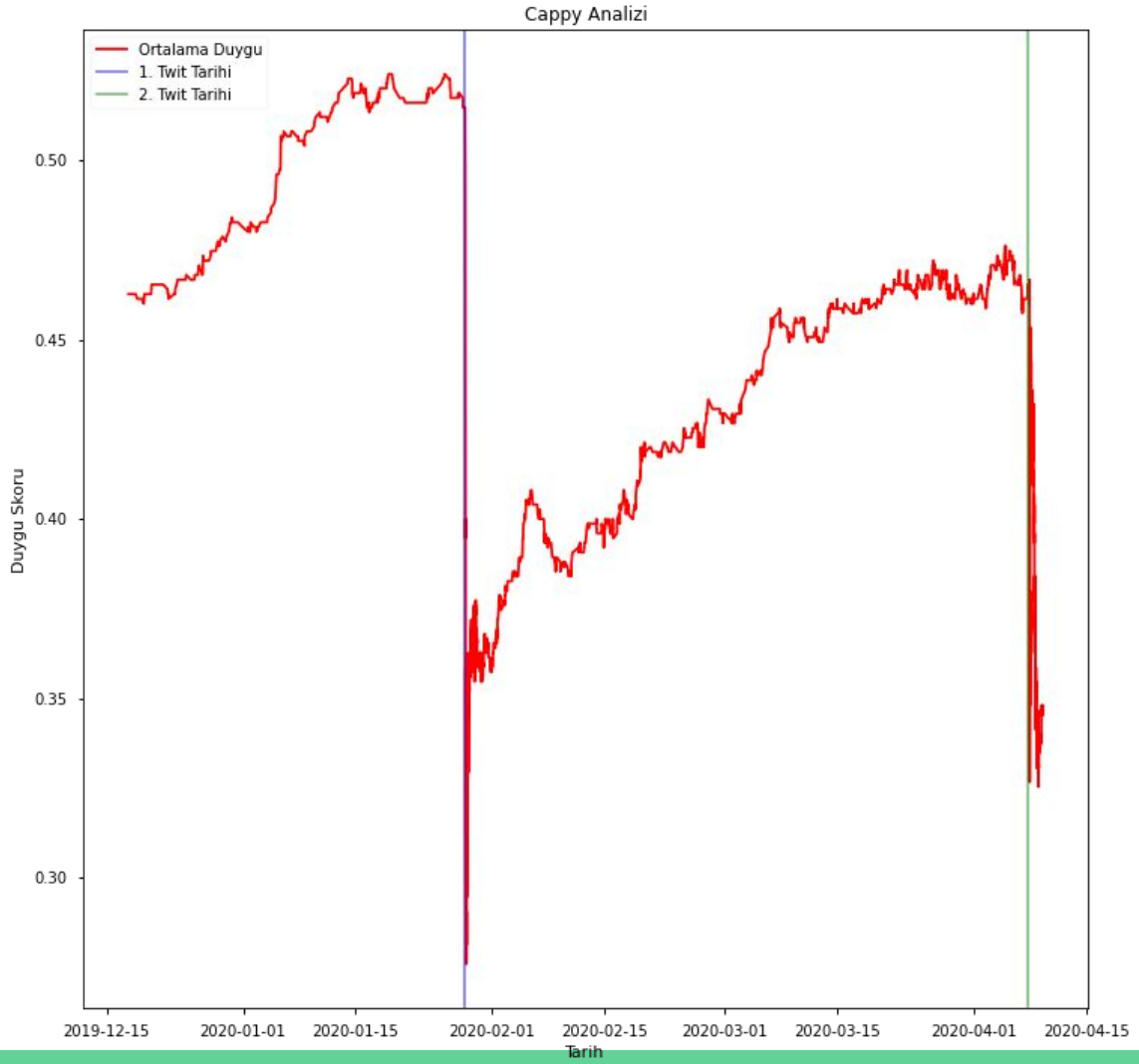


Berkcan Güven Analizi

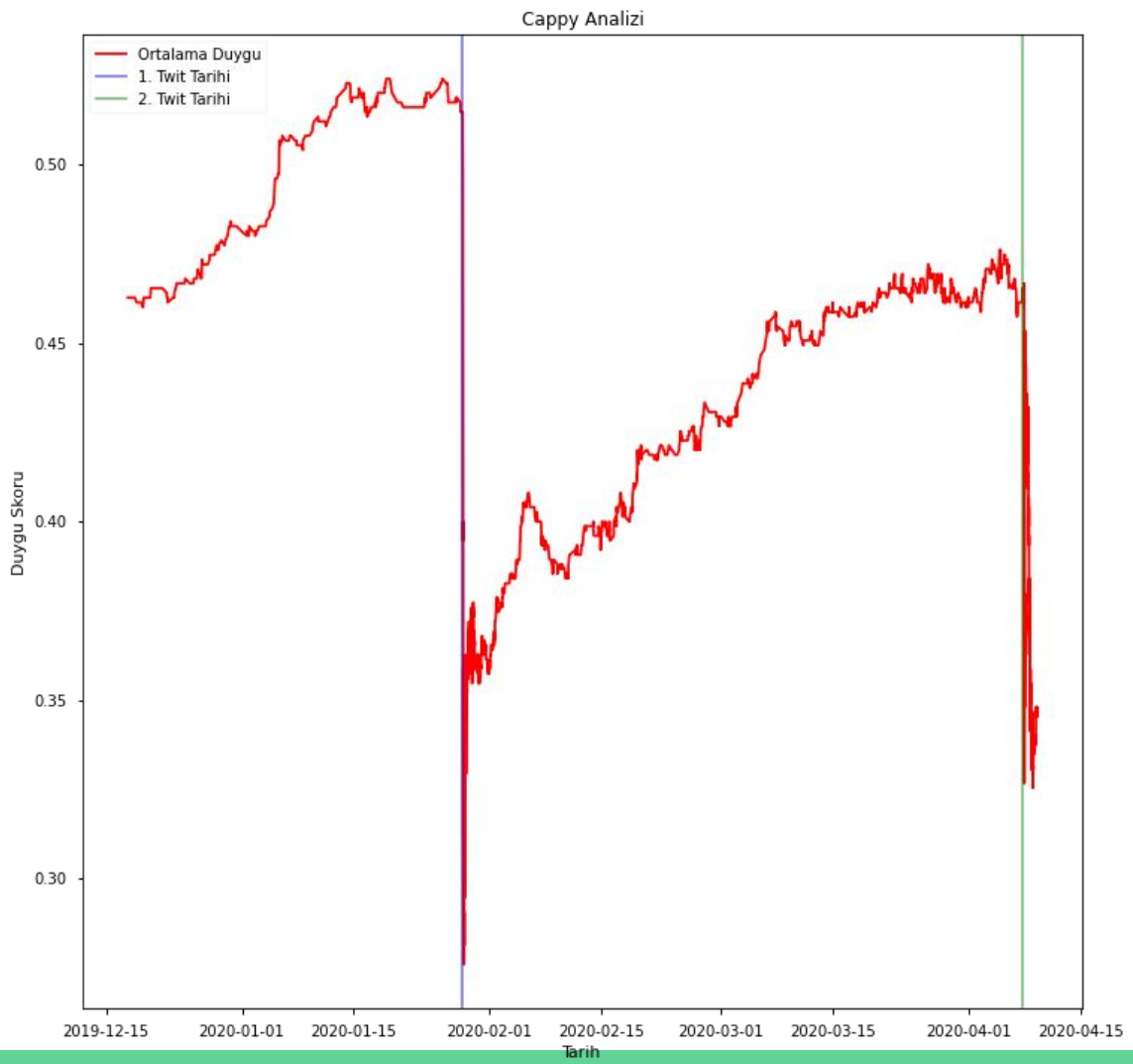


Cappy Analizi

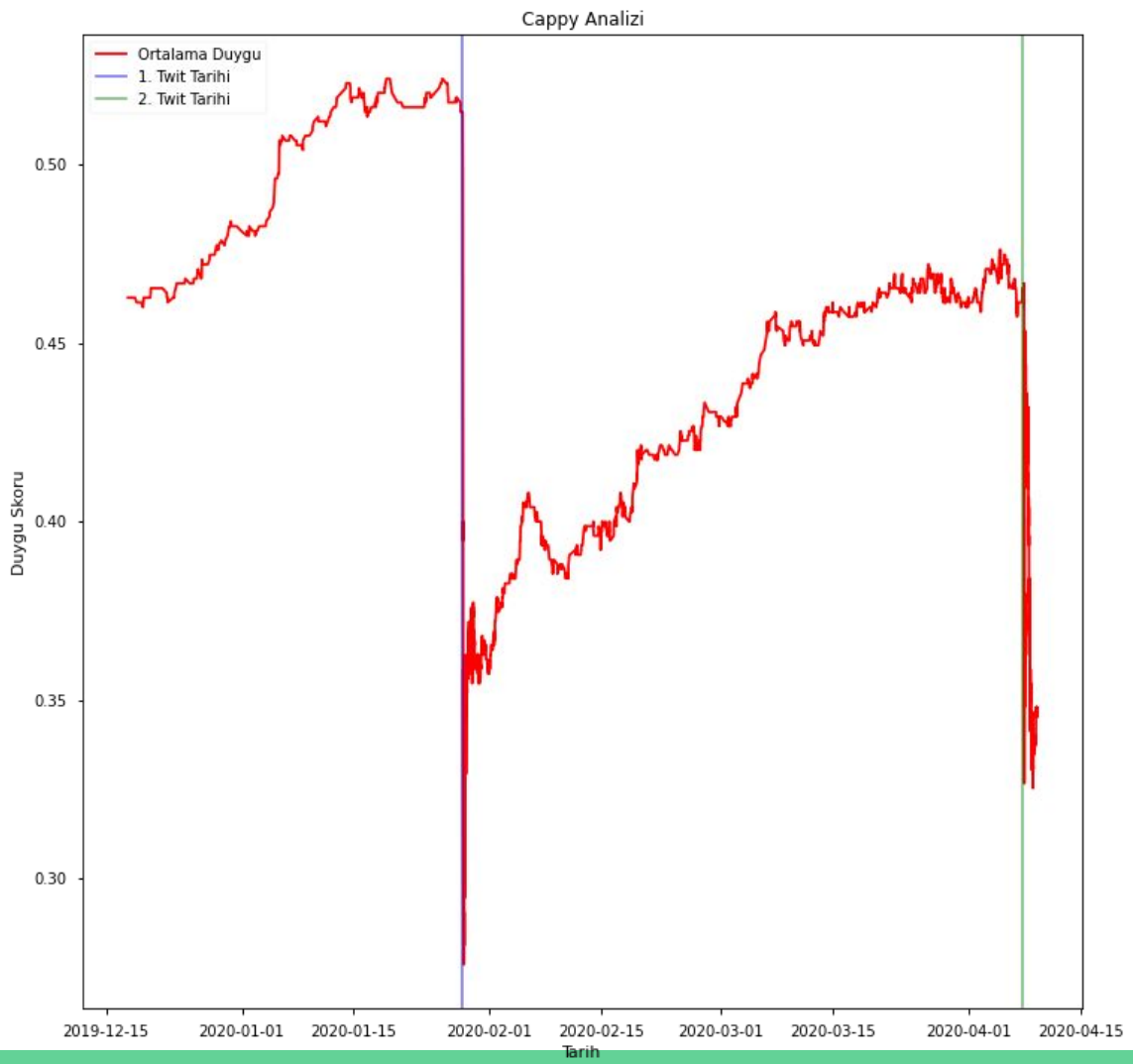
Cappy hakkında atılmış Türkçe twitlerin ortalama skorları:



Cappy Analizi



Cappy Analizi



Sonuç

- Duygu Analizi çok farklı yönlerden bakılması gereken bir çalışma alanıdır.
- Doğal Dil İşleme'de elle oluşturulmuş öznitelikler yerine kelime ve cümle için otomatik oluşturulan vektörler kullanılmaya başlandı.
- Otomatik duygu analizi ile büyük ölçekli marka/politikacı/kişi analizi yapılabilir.

Slayt, veriler ve kod için:

akoksal.com

https://github.com/akoksal