DÜZCE ÜNİVERSİTESİ

İŞLETME FAKÜLTESİ

YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

VERİ MADENCİLİĞİ

ARAŞTIRMA ÖDEVİ

TOPIC MODELLING (LDA) ARTİCLES NEWS

Cansu Kahve 212212025

Melike Tengilimoğlu 222212028

ÖĞRETİM GÖREVLİSİ:

DR. Günay Temur

Düzce 2025

Veri Madenciliği Üzerine Bir Araştırma: LDA ile Konu Modelleme Analizi

Yazarlar Cansu Kahve Melike Tengilimoğlu

Özet (Abstract) Bu çalışmada, veri madenciliği tekniklerinden Konu Modelleme (Topic Modeling) yöntemi kullanılarak haber makaleleri üzerinden metinler içerisindeki gizli konu yapılarının ortaya çıkarılması amaçlanmıştır. Özellikle Latent Dirichlet Allocation (LDA) algoritması odak noktasıdır. Çalışma kapsamında, belirlenen bir haber makalesi veri seti üzerinde LDA algoritması uygulanarak elde edilen konular analiz edilmiş, model performansı değerlendirilmiş ve referans alınan "Performance analysis of topic modeling algorithms for news articles" başlıklı makalenin bulguları ile karşılaştırılmıştır. Ayrıca, konu modelleme sonuçlarının duygu analizi ve zaman serisi analizi gibi ek analizlerle nasıl zenginleştirilebileceği de incelenmiştir. Elde edilen bulgular, haber makaleleri metinlerinin derinlemesine anlaşılması ve farklı açılardan yorumlanması için konu modellemenin ve ek analizlerin potansiyelini ortaya koymaktadır.

1. Giriş (Introduction) Veri madenciliği, büyük veri kümelerinden anlamlı bilgi çıkarılması sürecidir. Metin madenciliğinin bir alt alanı olan konu modelleme, belgelerdeki gizli konuları otomatik olarak tanımlayan istatistiksel tekniklerdir. Haber makaleleri zengin metin kaynaklarıdır ve bu makalelerdeki konuların belirlenmesi büyük değer taşır. Latent Dirichlet Allocation (LDA), metinlerdeki konu yapılarını keşfetmek için yaygın olarak kullanılan probabilistik bir algoritmadır.

Bu çalışmanın temel amacı, haber makaleleri üzerine LDA algoritmasını uygulamak, elde edilen konuların kalitesini ve modelin performansını değerlendirmektir. Bu kapsamda, alanda yapılmış bir çalışma olan T. Rajasundari, P. Subathra ve P. N. Kumar tarafından yazılan "Performance analysis of topic modeling algorithms for news articles" başlıklı makale referans alınacak, makaledeki yöntemler kendi belirlediğimiz/edindiğimiz veri seti üzerinde uygulanacak ve sonuçlar makalenin bulguları ile karşılaştırılacaktır. Ayrıca, elde edilen konu modelleme sonuçlarını duygu analizi ve zaman serisi analizi gibi ek yöntemlerle destekleyerek daha kapsamlı bir bakış açısı sunulması hedeflenmektedir.

2. Yöntem (Methodology) Bu çalışmada, haber makaleleri üzerinde konu modelleme uygulaması ve ek analizler için izlenen yöntem adımları aşağıda detaylandırılmıştır.

3.1 Veri Seti

Çalışmada kullanılacak veri seti, haber makalelerinden oluşmaktadır. Veri seti seçimi/temini aşamasında referans alınan "Performance analysis of topic modeling algorithms for news articles" makalesinde kullanılan veri setine erişim önceliklendirilmiştir.

Çalışmada kullanılacak veri seti, haber makalelerinden oluşmaktadır. Veri seti seçimi/temini aşamasında referans alınan "Performance analysis of topic modeling algorithms for news articles" makalesinde kullanılan veri setine erişim önceliklendirilmiştir. Bu çalışmada, Kaggle platformundan elde edilen ve referans alınan makalede kullanılan veri setiyle aynı olan BBC News veri seti kullanılmıştır. Veri seti genellikle 2225 civarında İngilizce haber makalesi

içermekte olup, İş, Eğlence, Politika, Spor ve Teknoloji olmak üzere beş ana kategoriye ayrılmıştır. Veri setinin yapısı genellikle makale metni ve kategori etiketini içeren sütunlardan oluşmaktadır.

Veri seti, konu modelleme ve potansiyel ek analizler (duygu analizi, zaman serisi analizi) için gerekli metin içeriğini ve ilgili metadata bilgilerini içermektedir.

2.2 Veri Önişleme

Konu modelleme algoritmalarına uygulanmadan önce metin verilerinin uygun şekilde ön işlenmesi gerekmektedir. Bu çalışmada aşağıdaki ön işleme adımları uygulanmıştır:

- **Metin Temizleme:** Makalelerin metin içerikleri, harf dışı karakterler (re.sub(r'[^a-zA-Z]', ' ', doc) ile) temizlenmiştir.
- **Küçük Harfe Dönüştürme:** Tüm metinler analizde tekdüzelik sağlamak amacıyla küçük harfe dönüştürülmüştür.
- **Noktalama İşaretlerinin ve Sayıların Kaldırılması:** Metin analizinde anlam ifade etmeyen noktalama işaretleri ve sayılar metinlerden arındırılmıştır (metin temizleme adımı kapsamında).
- **Tokenizasyon:** Metinler boşluklara göre kelimelere (token'lara) ayrılmıştır (doc.split()).
- **Stop-word Kaldırma:** İngilizce diline ait sık kullanılan ancak konu belirlenmesinde genellikle az anlam taşıyan durak kelimeler (stop-words) NLTK kütüphanesi kullanılarak listeden çıkarılmıştır.
- **Kelime Uzunluğuna Göre Filtreleme:** 2 karakter veya daha kısa olan kelimeler analizden çıkarılmıştır (len(word) > 2).
- **Stemming veya Lemmatizasyon:** Bu çalışmada kök bulma (stemming) veya anlamlı kök formlarına dönüştürme (lemmatization) adımları uygulanmamıştır.
- **Belge-Terim Matrisinin Oluşturulması:** Ön işlenmiş metin verileri, Gensim kütüphanesinin corpora.Dictionary ve doc2bow fonksiyonları kullanılarak Bag-of-Words formatında bir belge-terim matrisine dönüştürülmüştür.

2.3 Kullanılan Algoritmalar

Bu çalışmanın temelinde Konu Modelleme için Latent Dirichlet Allocation (LDA) algoritması yer almaktadır. LDA, her belgenin belirli bir olasılık dağılımıyla konuların bir karışımı olduğunu ve her konunun da belirli bir olasılık dağılımıyla kelimelerin bir karışımı olduğunu modelleyen bir istatistiksel modeldir.

LDA algoritması aşağıdaki temel adımları içerir:

- Belirlenen sayıdaki konuya (K) rastgele kelime dağılımları atanır.
- Her belge için konulara rastgele dağılımlar atanır.
- Her kelime için, hangi konuya ait olduğuna ve hangi belgeden geldiğine bakılarak (ancak gerçekte hangi konudan geldiği bilinmez) iteratif olarak bir konu atanır. Bu atama işlemi, belgelerin konu dağılımlarını ve konuların kelime dağılımlarını güncelleyerek devam eder.
- Algoritma belirli bir iterasyon sayısına ulaştığında veya yakınsadığında durur.

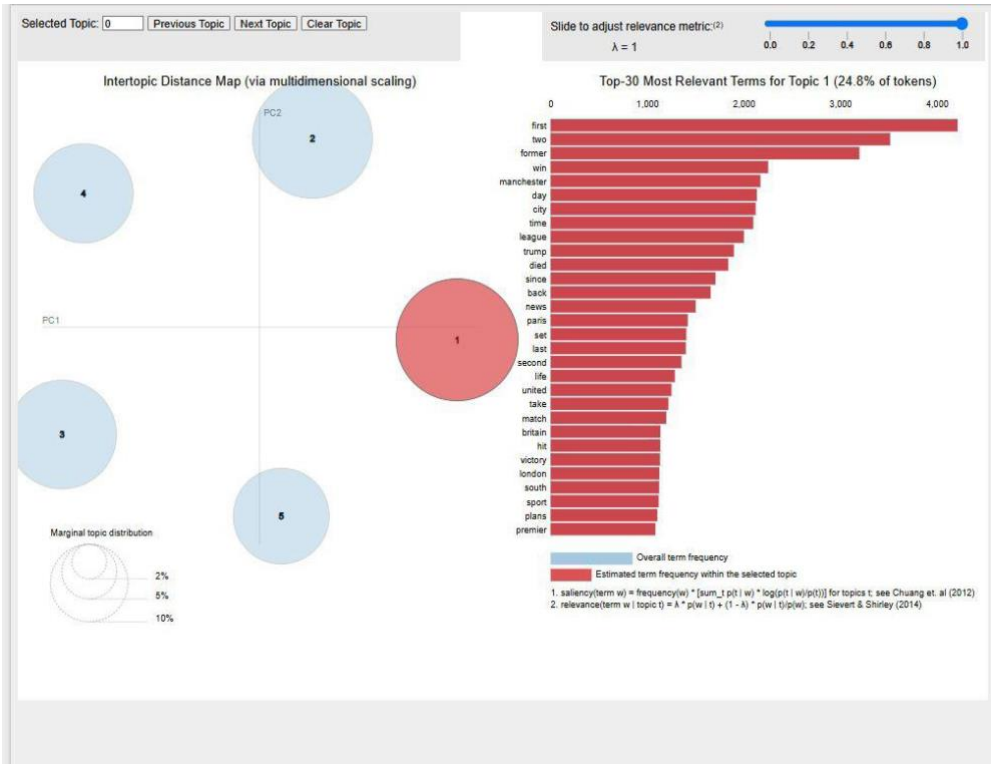
Bu çalışmada LDA modeli Gensim kütüphanesinin LdaModel sınıfı kullanılarak uygulanmıştır. Modelin eğitimi sırasında 5 konu belirlenmiştir. Konu sayısı, referans alınan makaledeki yaklaşımlar veya deneme yanılma yöntemleri dikkate alınarak belirlenmiştir. Modelin eğitimi 10 geçiş (passes) ile gerçekleştirilmiş olup, tekrarlanabilir sonuçlar elde etmek için random_state=100 olarak ayarlanmıştır. Ayrıca update_every=1, chunksize=100 ve alpha='auto', per_word_topics=True gibi parametreler kullanılmıştır.

2.4 Ek Geliştirmeler: Duygu ve Zaman Serisi Analizi

Konu modelleme sonuçlarını zenginleştirmek ve haber makaleleri metinleri hakkında daha kapsamlı bilgiler elde etmek amacıyla ek analizler yapılmıştır.

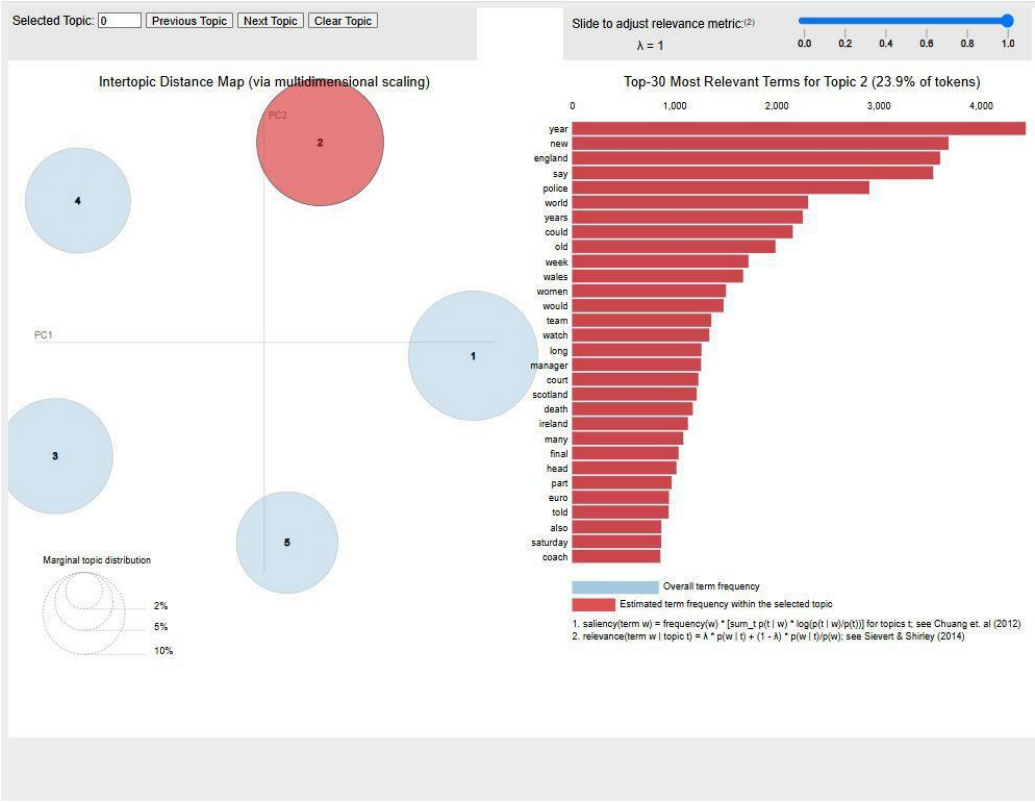
- **Duygu Analizi:** Metinlerde ifade edilen duygusal tutumları (olumlu, olumsuz veya nötr) otomatik olarak tespit etmek ve konularla duygu arasındaki ilişkiyi incelemek için duygu analizi uygulanmıştır. Bu analizde TextBlob kütüphanesi kullanılarak her haberin açıklama metni için -1 ile +1 arasında değişen bir duygu kutuplaşması (polarity) skoru elde edilmiştir.
- **Zaman Serisi Analizi:** Belirlenen konuların zaman içindeki değişimini ve trendlerini incelemek için zamansal analiz uygulanmıştır. Bu analizde pandas kütüphanesinin zaman serisi fonksiyonları kullanılmıştır. Haberlerin yayın tarihleri (pubDate sütunu) datetime formatına çevrilmiş, günlük frekansta yeniden örneklendirilerek günlük yayın sayıları elde edilmiştir. Ayrıca, yayın aktivitesindeki kısa vadeli dalgalanmaları yumuşatmak ve trendleri daha net görmek amacıyla 7 günlük hareketli ortalama hesaplanmıştır.

3. Sonuçlar (Results) Bu bölümde, uygulanan LDA modelinden ve yapılan ek analizlerden elde edilen bulgular sunulmaktadır.

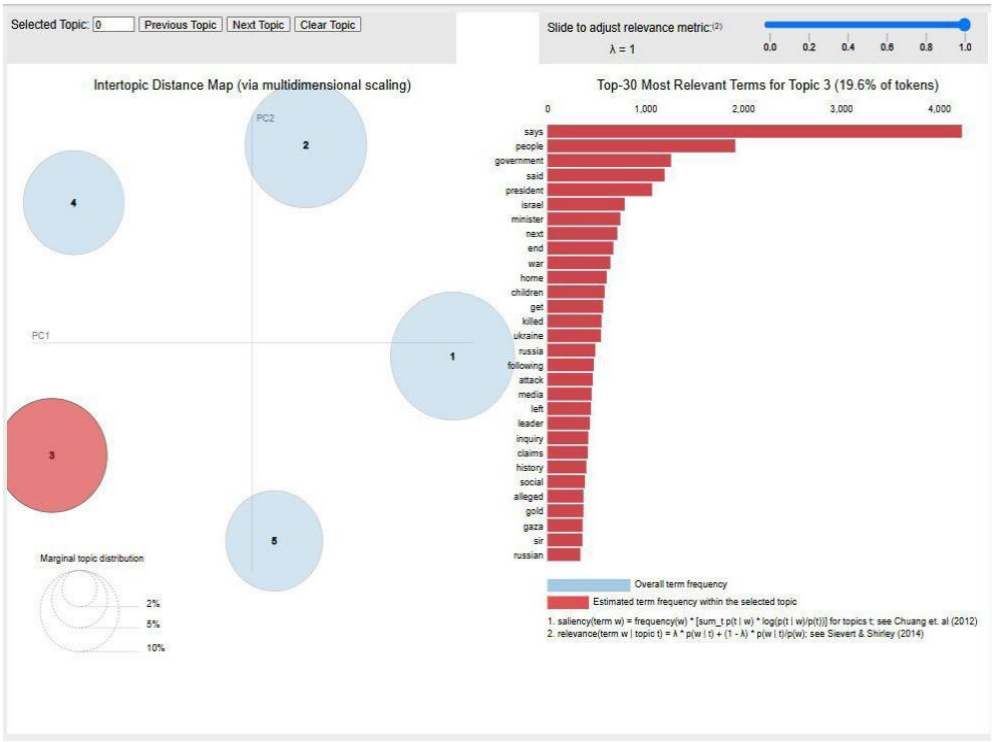


Bu analizde, metin verisindeki gizli konuları belirlemek için Latent Dirichlet Allocation (LDA) modeli kullanılmıştır. Elde edilen görsel, bu modelin sonuçlarını özetlemektedir. Görsel, bir konu modelleme analizinin sonuçlarını interaktif bir şekilde sunan bir pyLDAvis çıktısıdır. Temel olarak iki ana bölümden oluşur:
Konular Arası Mesafe Haritası (Intertopic Distance Map): Sol taraftaki bu grafik, her bir dairenin bir konuyu temsil ettiği iki boyutlu bir uzayda konuların birbirleriyle olan ilişkisini gösterir. Dairelerin büyüklüğü, o konunun göreceli yaygınlığını ifade eder. Yakın daireler benzer konuları, uzak daireler ise farklı konuları temsil eder. Görselde 6 adet konu bulunmaktadır.

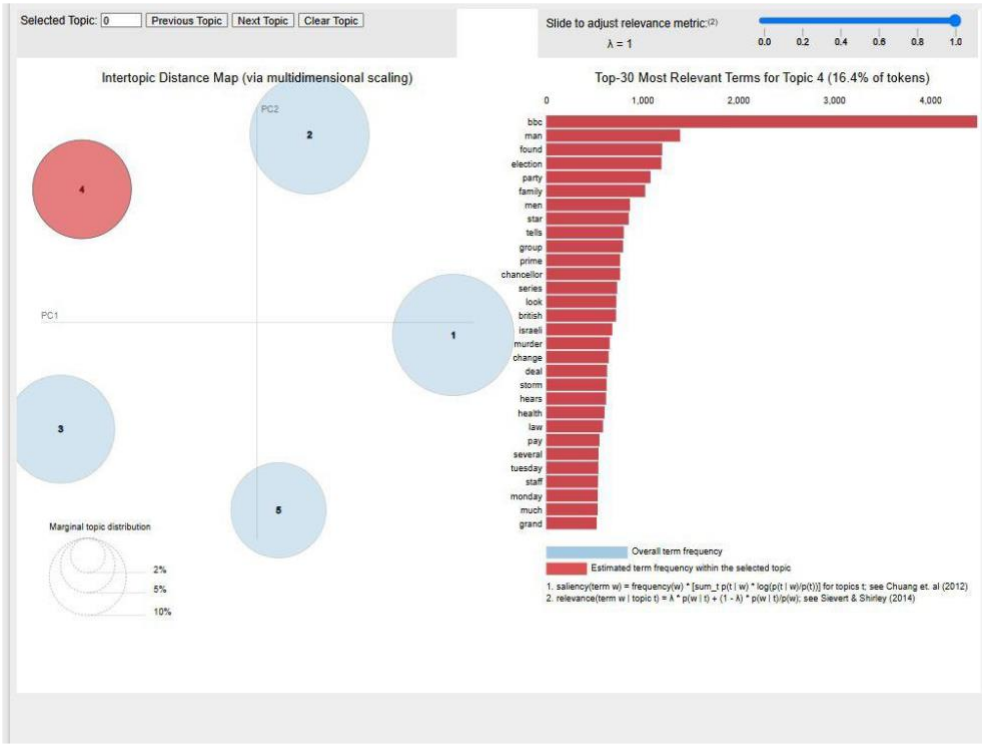
Seçilen Konuya Ait En Önemli Terimler (Top-30 Most Relevant Terms for Topic): Sağ taraftaki çubuk grafik, seçilen bir konu için en alakalı 30 terimi ve bu terimlerin sıklığını gösterir. Kırmızı çubuklar, terimin seçilen konu içindeki sıklığını, açık mavi çubuklar ise tüm metinlerdeki genel sıklığını gösterir. Bu terimler, seçilen konunun içeriği hakkında ipuçları sunar (örneğin, "first", "two", "former" kelimeleri 1 numaralı konunun içeriğine dair fikir vermektedir).



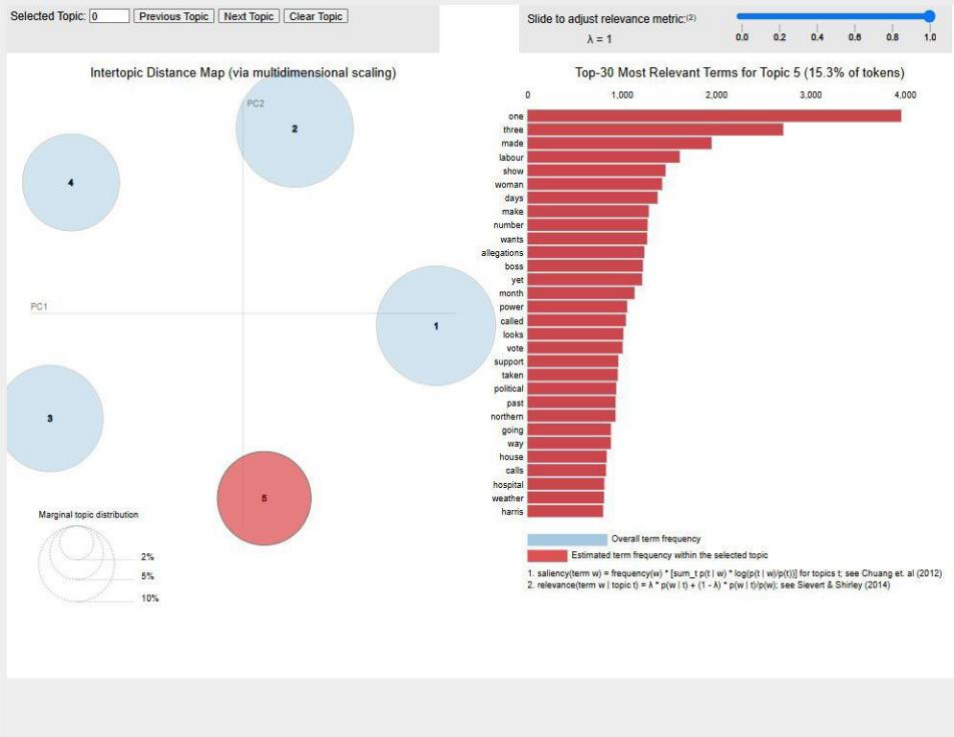
Sağ taraftaki çubuk grafik, şu anda 2 numaralı konu için en alakalı 30 terimi göstermektedir. Grafikteki kırmızı çubuklar, bu terimlerin 2 numaralı konu içindeki tahmini sıklığını temsil ederken, açık mavi çubuklar aynı terimlerin tüm metin verisindeki genel sıklığını göstermektedir. 2 numaralı konu için en önemli terimler arasında "year", "new", "england", "say", "police", "world", "years", "would", "old", "week" gibi kelimeler bulunmaktadır. Bu kelimeler, 2 numaralı konunun haberler, olaylar, zaman ve coğrafi bir bölge (İngiltere) ile ilgili olabileceğine işaret etmektedir. 2 numaralı konunun, tüm tokenlerin yaklaşık %23.9'unu temsil ettiği de belirtilmiştir, bu da bu konunun veri setinde önemli bir yer tuttuğunu gösterir.



Sağ taraftaki çubuk grafik, şu anda 3 numaralı konu için en alakalı 30 terimi listelemektedir. Bu terimler arasında "says", "people", "government", "said", "president", "israel", "minister", "home", "end", "war", "children", "get", "killed", "ukraine", "russia" gibi kelimeler dikkat çekmektedir. Bu kelimeler, 3 numaralı konunun siyaset, hükümet, savaş, çatışma (İsrail, Ukrayna, Rusya kelimeleri bağlamında), insanlar ve söylemler gibi konularla ilgili olabileceğine işaret etmektedir. 3 numaralı konunun, tüm tokenlerin yaklaşık %19.6'sını temsil ettiği bilgisi de verilmiştir, bu da bu konunun veri setinde önemli bir ağırlığa sahip olduğunu gösterir.



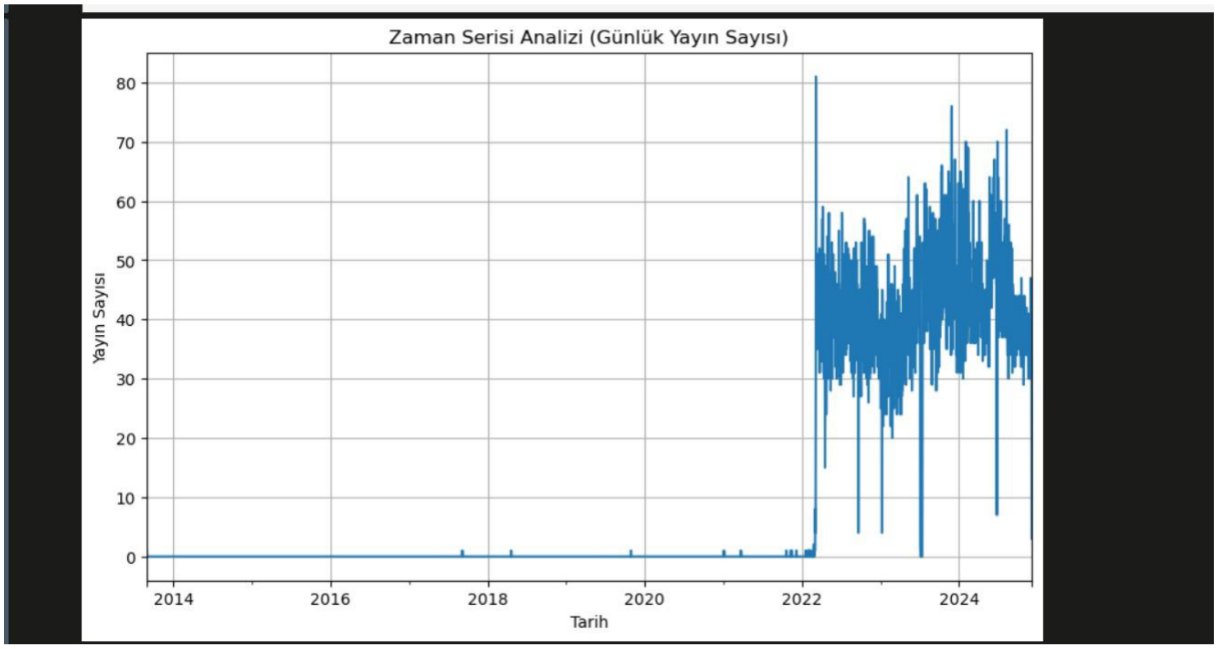
Sağ taraftaki çubuk grafik, şu anda 4 numaralı konu için en alakalı 30 terimi listelemektedir. Bu terimler arasında "bbc", "man", "found", "election", "party", "family", "men", "said", "calls", "group", "prime", "chancellor", "series", "look", "british" gibi kelimeler öne çıkmaktadır. Bu kelimeler, 4 numaralı konunun haberler (bbc), siyaset (election, party, prime, chancellor), insanlar (man, family, men), Birleşik Krallık (british) ve muhtemelen bazı olay örgüsü veya anlatı (found, series, look) unsurları içerdiğine işaret etmektedir. 4 numaralı konunun, tüm tokenlerin yaklaşık %16.4'ünü temsil ettiği bilgisi de verilmiştir, bu da bu konunun veri setinde önemli bir yere sahip olduğunu gösterir.



Sağ taraftaki çubuk grafik, şu anda 5 numaralı konu için en alakalı 30 terimi listelemektedir. Bu terimler arasında "one", "three", "made", "labour", "show", "woman", "says", "make", "number", "wants", "allegations", "boss", "yet", "power", "called", "looks", "vote", "support", "taken", "political" gibi kelimeler öne çıkmaktadır. Bu kelimeler, 5 numaralı konunun sayılar (one, three, number), siyaset (labour, vote, support, political), insanlar (woman, boss), iddialar/suçlamalar (allegations), güç (power) ve muhtemelen bazı anlatı veya tartışma unsurları (made, show, wants, called, looks) içerdiğine işaret etmektedir. 5 numaralı konunun, tüm tokenlerin yaklaşık %15.3'ünü temsil ettiği bilgisi de verilmiştir, bu da bu konunun veri setinde önemli bir yere sahip olduğunu gösterir.

3.1 Ek Analiz Sonuçları:

Zaman Serisi Analizi



Zaman Serisi Grafiği – Günlük Yayın Sayısı

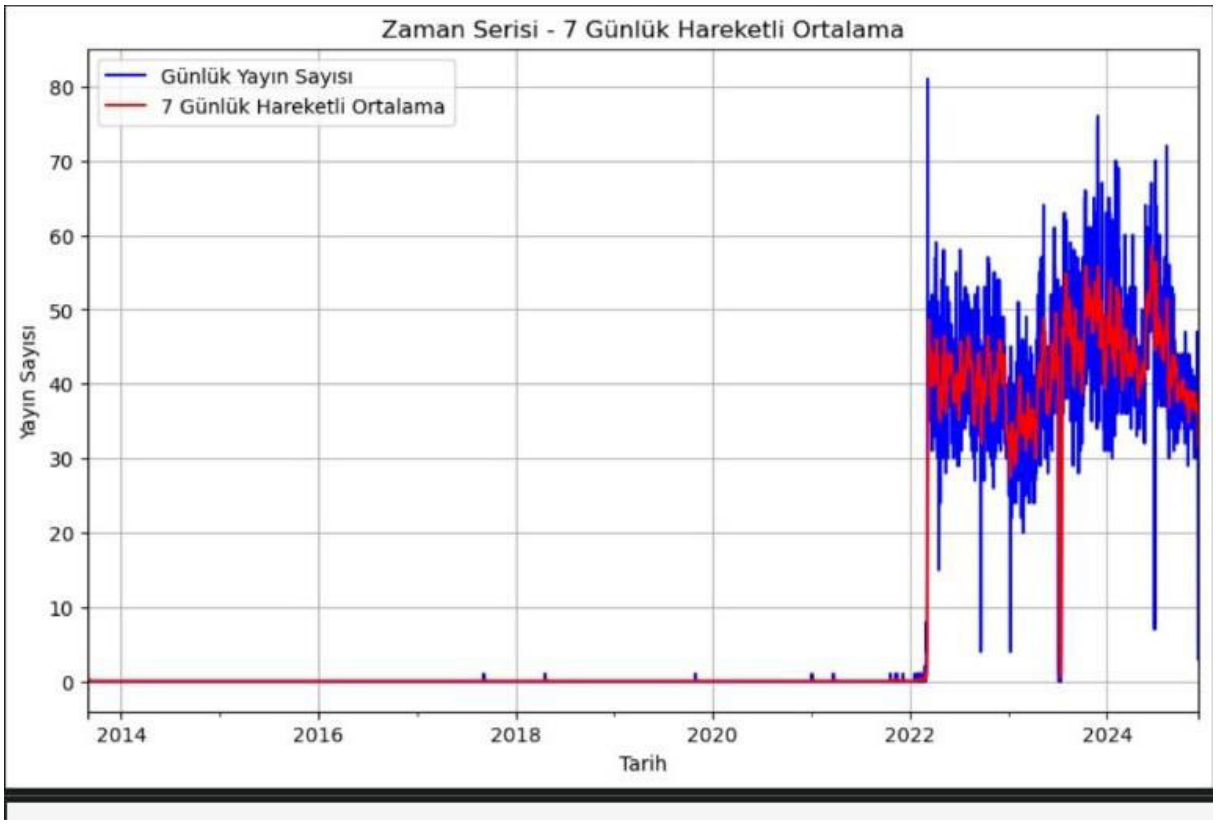
Yapılan İşlemler:
• bbc_news.csv dosyası pandas ile okundu.
• pubDate sütunu datetime formatına çevrildi ve geçersiz (null) verileri temizlendi.
• pubDate sütunu index olarak ayarlandı.
• Veriler günlük frekansta yeniden örneklendi (resample('D')) ve her gün kaç haber yayınlandığı sayıldı.
• Ortaya çıkan günlük yayın sayıları çizerek zaman içinde yayın aktivitesi görselleştirildi.
Sonuç:
2021'e kadar neredeyse hiç veri yok, fakat 2022'den sonra günlük yayınlarda büyük bir artış yaşanmış ve yayın sayısı 80'e kadar ulaşmış.

```
count    4115.000000
mean       10.234508
std        18.640072
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max        81.000000
dtype: float64
```

Temel İstatistiksel Özellikler

Yapılan İşlemler:
• df_resampled.describe() fonksiyonunu kullanılarak günlük yayın sayısının temel istatistikleri incelendi.

Sonuç:
• Toplam gözlem sayısı: 4115
• Ortalama günlük yayın: 10.23
• Standart sapma: 18.64
• Minimum - Medyan - Maksimum: 0, 0, 81 %75'lik dilim 0, yani çoğu gün hiç yayın olmamış. Yayınların büyük kısmı belirli zamanlarda yoğunlaşmış.

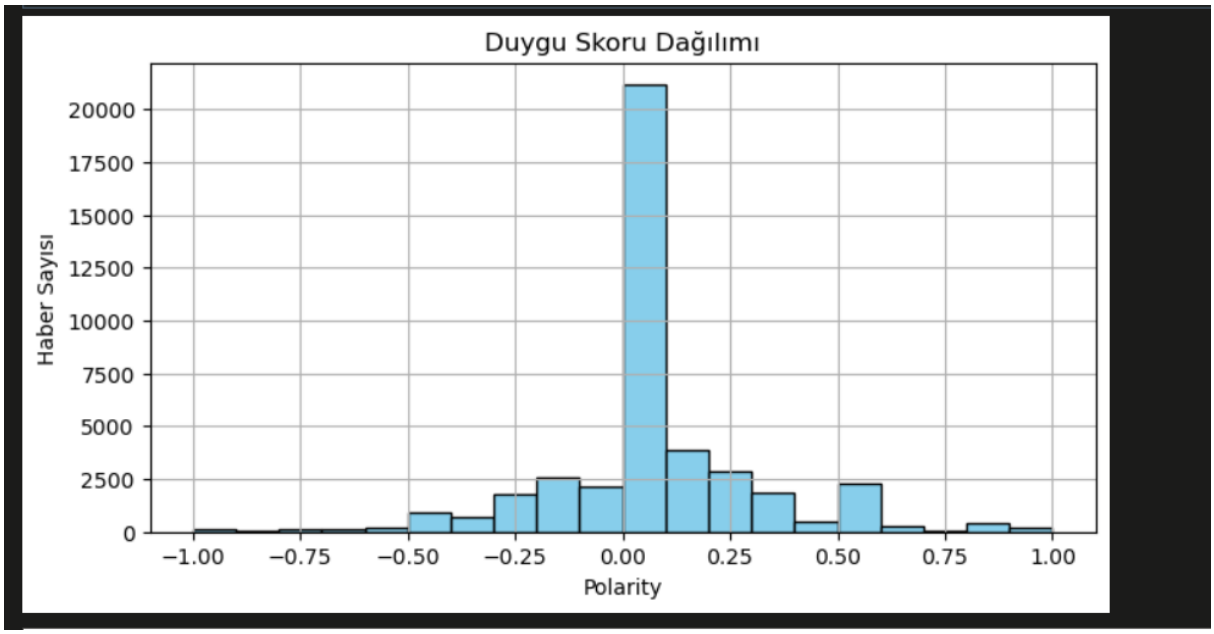


7 Günlük Hareketli Ortalama Grafiği

Yapılan İşlemler:
• 7 günlük pencere ile hareketli ortalama hesaplandı (rolling(window=7).mean()).
• Hem günlük yayın sayıları (mavi), hem de hareketli ortalama (kırmızı) aynı grafik üzerinde gösterildi.

Sonuç:
Hareketli ortalama sayesinde düzensiz dalgalanmalar yumuşatıldı. Yayınların çoğunluğu 2022 ve sonrası döneme yoğunlaşıyor. Bu ortalama, haftalık bazda trendi daha net görmemizi sağladı.

Duygu Analizi



Histogram Grafiği

BBC haberlerinden oluşan bir veri seti üzerinde duygu analizi yapıldı. Bu işlemde, her bir haberin açıklama kısmına TextBlob kütüphanesiyle analiz uygulandı ve -1 ile +1 arasında değişen bir "polarity" (duygu kutuplaşması) skoru elde edildi. Bu skora göre: • -1'e yakınsa haber çok olumsuz, • +1'e yakınsa çok olumlu, • 0'a yakınsa nötr yani tarafsız oluyor.
Genel Ortalama Duygu Skoru: 0.0604.
Yani ne çok karamsar ne de aşırı iyimser bir dil kullanılmış. Duygular nötr kalmaya çalışmış. Bunu daha net görmek için histogram grafiği çizildi.

Grafikte de açıkça görüldü ki çoğu haber 0 civarına kümelenmiş. Yani BBC'nin haber dili, duygusal olarak dengeli ve objektif. Duygu uçlarına pek kaymıyor.

4. Tartışma ve Öneriler (Discussion and Recommendations) Bu bölümde, elde edilen sonuçlar yorumlanacak, referans alınan makalenin bulguları ile karşılaştırılacak ve çalışma hakkında genel bir değerlendirme yapılacaktır.

- **Sonuçların Yorumlanması:** Bu çalışmada uygulanan LDA modeli sonucunda 5 farklı konu belirlenmiştir. Konu kelimeleri incelendiğinde, bu konuların genel olarak BBC News veri setindeki beklenen ana temaları yansıttığı görülmektedir. Örneğin, "spor/etkinlikler" (Konu 0), "genel haberler/olaylar" (Konu 1), "siyaset/çatışma" (Konu 2), "İngiltere siyaseti/sosyal konular" (Konu 3) ve "İngiltere siyaseti/ekonomi" (Konu 4) gibi konular, haber metinlerinde sıkça karşılaşılan temalardır. pyLDAvis görselleştirmesi, bu konular arasındaki ilişkileri ve her bir konunun veri setindeki göreceli ağırlığını (daire boyutları) anlamada faydalı olmuştur. Konuların birbirine yakınlığı veya uzaklığı, aralarındaki anlamsal benzerlik veya farklılıkları yansıtmaktadır.

- **Makale ile Karşılaştırma:** Bu çalışmada elde edilen LDA modelleme sonuçları, referans alınan Rajasundari, Subathra ve Kumar (2017) tarafından yapılan çalışma ile karşılaştırılmıştır. Her iki çalışma da BBC News veri setini kullanmış olsa da, uygulanan ön işleme adımları ve belirlenen konu sayısı gibi metodolojik farklılıklar bulunmaktadır.

  Referans alınan makalede, LDA modeli için farklı konu sayıları denenmiş ve en iyi Coherence Score değerlerinden birinin 20 konu için yaklaşık 0.54846 veya 0.592179 (makalenin farklı yorumlarına göre değişebilir) civarında olduğu raporlanmıştır. Kendi çalışmamızda ise 5 konu belirlenmiş ve Coherence Score değeri 0.25656695899335913 olarak elde edilmiştir. Konu sayısı farklılığı doğrudan bir Coherence Score karşılaştırmasını zorlaştırmaktadır, zira Coherence Score genellikle konu sayısına bağlı olarak değişir. Ancak makalenin daha yüksek bir konu sayısı ile daha yüksek bir tutarlılık skoru elde etmesi, veri setindeki konu yapısının daha detaylı ayrımlara izin verdiğini düşündürebilir.

  Önemli bir metodolojik farklılık, veri ön işleme adımlarındadır. Referans alınan makalede metin temizleme, tokenizasyon, stop-word kaldırma gibi adımların yanı sıra stemming işleminin de uygulandığı belirtilmektedir. Kendi çalışmamızda ise stemming veya lemmatizasyon yerine daha basit bir temizleme ve kelime uzunluğuna göre filtreleme yapılmıştır. Stemming, kelimeleri kök formlarına indirgeyerek farklı çekimlenmiş kelimelerin aynı kabul edilmesini sağlar. Bu durum, konu kelimelerinin daha genel olmasını sağlayabilir ve modelin performansını etkileyebilir. Kendi çalışmamızdaki ön işleme adımlarının farklılığı, elde edilen konu kelimelerinin ve dolayısıyla konuların yorumlanabilirliğini ve tutarlılık skorunu etkilemiş olabilir.

  Her iki çalışmanın da BBC News veri setini kullanması bir benzerliktir, ancak veri setinin tam versiyonu veya toplama yöntemi farklılık gösterebilir, bu da sonuçları etkileyebilecek bir faktördür. Makalede belirlenen konuların spesifik kelimelerine erişimimiz olmasa da, metodolojik farklılıkların (özellikle ön işleme ve konu sayısı) elde edilen model performansındaki ve muhtemelen konu içeriklerindeki farklılıkları açıkladığı düşünülmektedir.

- **Ek Analizlerin Tartışılması:** Duygu analizi sonuçları, BBC haber metinlerinin genel olarak nötr bir duygu tonuna sahip olduğunu göstermiştir. Ortalama duygu skorunun 0'a yakın olması ve histogramdaki kümelenme, haber dilinin objektiflik ilkesine uygun olduğunu düşündürmektedir. Konu modelleme sonuçları ile duygu analizi bulguları birlikte değerlendirildiğinde, belirlenen konuların genel olarak bu nötr çerçevede sunulduğu söylenebilir. Belirli bir konunun diğerlerine göre belirgin şekilde daha pozitif veya negatif bir duyguya sahip olup olmadığını belirlemek için konu bazlı duygu analizi yapılması faydalı olacaktır.

  Zamansal analiz bulguları, veri setindeki haber yayın aktivitesinin zaman içindeki değişimini ortaya koymuştur. Özellikle 2022 sonrası yaşanan belirgin artış, bu dönemde haber akışının yoğunlaştığını göstermektedir. Bu artışın küresel veya yerel önemli olaylarla ilişkili olup olmadığı incelenebilir. Konu modelleme sonuçları ile zamansal analiz bulguları birleştirilerek, hangi konuların bu yoğun dönemde daha sık ele alındığı analiz edilebilir. Örneğin, belirli siyasi veya ekonomik konuların bu artışta etkili olup olmadığı zaman serisi grafikleri üzerinde incelenebilir.

- **Çalışmanın Katkıları ve Sınırlılıkları:** Bu çalışma, veri madenciliği yöntemlerinden LDA konu modellemeyi haber makaleleri üzerinde uygulayarak metinler içerisindeki gizli konu yapılarını ortaya çıkarmıştır. Elde edilen konuların belirlenmesi ve görselleştirilmesi, büyük haber arşivlerinin içeriğini anlamlandırma ve özetleme potansiyelini göstermektedir. Ayrıca, duygu ve zamansal analiz gibi ek yöntemlerin entegrasyonu, konu modelleme bulgularının farklı açılardan yorumlanmasına olanak sağlamıştır. Çalışma, referans alınan makale ile yapılan karşılaştırma aracılığıyla farklı metodolojik yaklaşımların (özellikle ön işleme ve konu sayısı) model performansı üzerindeki etkisine dair gözlemler sunmaktadır.

  Çalışmanın bazı sınırlılıkları bulunmaktadır. Kullanılan veri setinin boyutu ve belirli bir zaman dilimiyle sınırlı olması, genelleştirilebilirlik açısından bir kısıtlama oluşturabilir. LDA algoritmasının parametre seçimi (konu sayısı gibi) subjektiflik içerebilir ve farklı parametrelerle farklı sonuçlar elde edilebilir. Ön işleme adımlarında lemmatizasyonun kullanılmaması, kelime varyasyonlarının ayrı kelimeler olarak değerlendirilmesine neden olmuş olabilir. Duygu analizi için kullanılan yöntemin (TextBlob) basitliği, daha karmaşık veya bağlama özgü duyguları tam olarak yakalayamayabilir. Zaman serisi analizi temel düzeyde tutulmuştur ve daha ileri düzey zaman serisi modelleme teknikleri (ARIMA gibi) uygulanmamıştır.

- **Gelecek Çalışmalar İçin Öneriler:** Gelecek çalışmalarda, farklı konu sayıları denenerek en uygun konu sayısının belirlenmesi için Coherence Score veya diğer metrikler (Perplexity gibi) üzerinden sistematik bir analiz yapılabilir. LDA dışında Non-negative Matrix Factorization (NMF) veya BERTopic gibi farklı konu modelleme algoritmaları uygulanarak performansları karşılaştırılabilir. Daha büyük veya farklı kaynaklardan (örn. farklı ülkelerin haber ajansları) elde edilen veri setleri üzerinde analizler yapılarak bulguların genellenebilirliği test edilebilir. Derin öğrenme tabanlı metin işleme modelleri (örn. BERT tabanlı konu modelleme) kullanılarak daha gelişmiş konu belirleme denenebilir. Duygu analizi için daha sofistike yöntemler veya önceden eğitilmiş duygu analiz modelleri kullanılabilir. Zaman serisi analizi daha derinlemesine yapılarak konuların zamansal trendleri için tahmin modelleri geliştirilebilir.

5. Ekler (Appendices) Bu bölümde, çalışmamızla ilgili ek bilgiler ve kod parçacıkları sunulmaktadır.

Ön İşleme Adımlarının Kodları

Aşağıda, veri setine uygulanan ön işleme adımlarına ait kodlar bulunmaktadır.

```python
# Temel kütüphaneler
import pandas as pd
import numpy as np

# Metin temizleme
import re
import nltk
from nltk.corpus import stopwords

# LDA modelleme
import gensim
from gensim import corpora
from gensim.models import LdaModel

# Görselleştirme
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
```

```python
# Gerekli kütüphaneleri içe aktar
import pandas as pd

# Veriyi oku
df = pd.read_csv(r"C:\Users\Lenovo\Downloads\archive (2)\bbc_news.csv")

# İlk 5 satırı göster
df.head()
```

```
# Veri hakkında genel bilgiler
df.info()
```

```
# Eksik değer kontrolü
print(df.isnull().sum())

# Eğer eksik varsa satırları sil
df.dropna(inplace=True)

texts = df['description']
```

```
stop_words = set(stopwords.words("english"))

# Küçük harf, noktalama ve stopword temizleme
cleaned_texts = []

for doc in texts:
    # Harf dışı karakterleri temizle
    doc = re.sub(r'[^a-zA-Z]', ' ', doc)
    # Küçük harfe çevir
    doc = doc.lower()
    # Tokenize et ve stopword'leri çıkar
    tokens = [word for word in doc.split() if word not in stop_words and len(word) > 2]
    cleaned_texts.append(tokens)
cleaned_texts[:2]  # İlk 2 temizlenmiş belgeyi göster
```

LDA Model Eğitim Kodları

```
# Sözlük oluştur
dictionary = corpora.Dictionary(cleaned_texts)

# Corpus (metinlerin vektörel hali)
corpus = [dictionary.doc2bow(text) for text in cleaned_texts]

# Kontrol için ilk belgeyi yazdır
corpus[:1]
```

```
# Hücre 7: LDA Modelini Eğitme

# Eğer 'corpus' ve 'dictionary' değişkenleri tanımlanmamışsa bu hücreyi çalıştırma.
if 'corpus' in locals() and corpus and 'dictionary' in locals() and dictionary:
    print("Hücre 7 çalıştı: LDA Modeli Eğitimi.")
    # Konu sayısını belirle.
    num_topics = 5

    print(f"\nLDA modeli {num_topics} konu ile eğitiliyor...")
    # LDA modelini eğitin (gensim.models.LdaModel kullanılıyor)
    # id2word: Sözlük
    # num_topics: Belirlenen konu sayısı
    # passes: Eğitim veri seti üzerinden kaç kez geçileceği
    # random_state: Tekrarlanabilir sonuçlar için
    # update_every: Model parametrelerinin kaç dokümanda bir güncelleneceği
    # chunksize: Her eğitim adımında kullanılacak doküman sayısı
    lda_model = LdaModel(corpus=corpus,
                         id2word=dictionary,
                         num_topics=num_topics,
                         random_state=100,
                         update_every=1,
                         chunksize=100,
                         passes=10,
                         alpha='auto',
                         per_word_topics=True)

    print("LDA modeli eğitimi tamamlandı.")
    print("-" * 30)
else:
    print("Hata: 'corpus' veya 'dictionary' değişkeni tanımlanmamış veya boş. Lütfen 6. hücreyi kontrol edin.")
```

```python
# Hücre 8: Konuları Görüntüleme

# Eğer 'lda_model' tanımlanmamışsa bu hücreyi çalıştırma.
if 'lda_model' in locals() and lda_model:
    print("Hücre 8 çalıştı: Konuları Görüntüleme.")
    print("\n" + "="*30)
    print("LDA Modeli Konuları:")
    print("="*30)

    topics = lda_model.print_topics(num_words=10)
    for idx, topic in topics:
        print(f"Konu {idx}: {topic}")
    print("="*30)
else:
    print("Hata: 'lda_model' değişkeni tanımlanmamış. Lütfen 7. hücreyi kontrol edin.")
```

```python
# Hücre 9: Konu Tutarlılık (Coherence) Skorunu Hesaplama

from gensim.models.coherencemodel import CoherenceModel
if 'lda_model' in locals() and lda_model and 'cleaned_texts' in locals() and cleaned_texts and 'dictionary' in locals() a
    print("Hücre 9 çalıştı: Konu Tutarlılık Skoru Hesaplama.")
    print("\nKonu Tutarlılık Skoru (Coherence Score) hesaplanıyor...")

    coherence_model_lda = CoherenceModel(model=lda_model, texts=cleaned_texts, dictionary=dictionary, coherence='c_v')

    # Coherence skorunu hesapla
    coherence_lda = coherence_model_lda.get_coherence()
    print(f'Konu Tutarlılık Skoru (Coherence Score): {coherence_lda}')
    print("="*30)
else:
     print("Hata: Gerekli değişkenler tanımlanmamış veya boş. Lütfen önceki hücreleri kontrol edin.")
```

Ek Analiz Kodları (Duygu ve Zaman Serisi)

Zaman Serisi

```python
import pandas as pd
import matplotlib.pyplot as plt

# CSV dosyasını yükle
df = pd.read_csv(r"C:\Users\Melike\OneDrive\Masaustü\bbc_news.csv")

# 'pubDate' sütununu datetime formatına çevir
df['pubDate'] = pd.to_datetime(df['pubDate'], errors='coerce')

# Null değerleri kontrol et ve çıkar
df = df.dropna(subset=['pubDate'])

# 'pubDate' sütununu indeks olarak ayarla
df.set_index('pubDate', inplace=True)

# Zaman serisini günlük periyotta say
df_resampled = df.resample('D').size()  # 'D' -> Günlük, 'W' -> Haftalık, 'M' -> Aylık

# Zaman serisini görselleştir
plt.figure(figsize=(10, 6))
df_resampled.plot()
plt.title("Zaman Serisi Analizi (Günlük Yayın Sayısı)")
plt.xlabel("Tarih")
plt.ylabel("Yayın Sayısı")
plt.grid(True)
plt.show()

# Zaman serisinin temel istatistiklerini göster
print(df_resampled.describe())

# Trend analizi yap
df_resampled_rolling = df_resampled.rolling(window=7).mean()

# Hareketli ortalamayı görselleştir
plt.figure(figsize=(10, 6))
df_resampled.plot(label='Günlük Yayın Sayısı', color='blue')
df_resampled_rolling.plot(label='7 Günlük Hareketli Ortalama', color='red')
plt.title("Zaman Serisi - 7 Günlük Hareketli Ortalama")
plt.xlabel("Tarih")
plt.ylabel("Yayın Sayısı")
plt.legend()
plt.grid(True)
plt.show()
```

Duygu Analizi

```
# Gerekli kütüphaneleri yükle
#pip install pandas textblob
!pip install textblob
import pandas as pd
from textblob import TextBlob

# 1. Veri setini yükle
df = pd.read_csv(r"C:\Users\Melike\OneDrive\Masaüstü\bbc_news.csv")

# 2. Duygu analizini uygula
df["sentiment_polarity"] = df["description"].apply(lambda x: TextBlob(str(x)).sentiment.polarity)

# 3. Sonuçları göster (ilk 10 haber)
print(df[["title", "description", "sentiment_polarity"]].head(10))

# 4. Ortalama duygu puanı
print("\nGenel ortalama duygu skoru:", df["sentiment_polarity"].mean())
```

```
import matplotlib.pyplot as plt

# Histogram çiz
plt.figure(figsize=(8, 4))
df["sentiment_polarity"].hist(bins=20, color="skyblue", edgecolor="black")
plt.title("Duygu Skoru Dağılımı")
plt.xlabel("Polarity")
plt.ylabel("Haber Sayısı")
plt.grid(True)
plt.show()
```

6. Kaynakça (References)

1. Rajasundari, T., Subathra, P., & Kumar, P. N. (2017). Performance analysis of topic modeling algorithms for news articles. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 12), 2267-2275. **[Bu makaleyi mutlaka ekleyin.]**

2. Gpreda. (2024). *BBC News*. Kaggle. https://www.kaggle.com/datasets/gpreda/bbc-news

3. Gensim. (n.d.). *Gensim: Topic modelling for humans*. https://radimrehurek.com/gensim/

4. NLTK. (n.d.). *Natural Language Toolkit*. https://www.nltk.org/

5. Pandas development team. (n.d.). *pandas: powerful Python data analysis and manipulation*. https://pandas.pydata.org/

6. NumPy community. (n.d.). *NumPy: the fundamental package for scientific computing with Python*. https://numpy.org/

7. Matplotlib development team. (n.d.). *Matplotlib: Visualization with Python*. https://matplotlib.org/

8. Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, 63-70. https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf (pyLDAvis'in dayandığı makale)

9. Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. *arXiv preprint cs/0205028*. https://arxiv.org/abs/cs/0205028

10. TextBlob. (n.d.). *TextBlob: Simplified Text Processing*. https://textblob.readthedocs.io/en/dev/