

# EcomTrend Analyst-2 - Yorumlarda kelime analizi

Yorumlarda en sık geçen kelimeleri bul ve wordcloud şeklinde göster.

python

Her zaman ayrıntıları göster |

```
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Veriyi yükle
file_path = '/mnt/data/amazon_review.csv'
df = pd.read_csv(file_path)

# Yorum metni içeren kolonu bulma
df.columns
```

Sonuç

```
Index(['reviewerID', 'asin', 'reviewerName', 'helpful', 'reviewText',
      'overall', 'summary', 'unixReviewTime', 'reviewTime', 'day_diff',
      'helpful_yes', 'total_vote'],
      dtype='object')
```

python

Her zaman ayrıntıları göster |

```
# "reviewText" sütunu yorumları içeriyor, bu sütundaki metinleri birleştirip
wordcloud oluşturacağız

# Tüm yorumları birleştir
text = " ".join(review for review in df['reviewText'].astype(str))

# WordCloud oluştur
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(text)

# Görselleştir
plt.figure(figsize=(10, 5))
```

```
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("En Sık Geçen Kelimeler - WordCloud", fontsize=16)
plt.show()
```

# En Sık Geçen Kelimeler - WordCloud

Yukarıdaki WordCloud görselinde, kullanıcı yorumlarında en sık geçen kelimeler görselleştirilmiştir. Büyük harflerle yazılan kelimeler daha sık tekrar edilenleri temsil eder. İstersen bu kelimeleri listeleyebilir, belirli kelimeleri filtreleyebilir veya olumlu/olumsuz yorumlar için ayrı WordCloud'lar oluşturabilirim. Ne yapmak istersin?

Önce kelimeleri olumlu / olumsuz olarak listele, sonra da olumlu / olumsuz kelimeler için ayrı wordcloudlar oluştur.

```
python
```

Her zaman ayrıntıları göster |

```
from textblob import TextBlob
from collections import Counter
import re

# Yorum metinlerini temizle ve kelimelere ayır
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s]', '', text) # Sadece harf ve boşluk bırak
    text = text.lower() # Küçük harfe çevir
    return text.split()
```

```
python
```

Her zaman ayrıntıları göster |

```
from textblob import TextBlob
from collections import Counter
import re

# Yorum metinlerini temizle ve kelimelere ayır
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s]', '', text) # Sadece harf ve boşluk bırak
    text = text.lower() # Küçük harfe çevir
    return text.split()
```

```

# Tüm yorumları işle
all_words = []
for review in df['reviewText'].astype(str):
    words = clean_text(review)
    all_words.extend(words)

# Her kelime için duygu analizi yap
positive_words = []
negative_words = []

for word in set(all_words): # Her kelimeyi yalnızca bir kez incele
    analysis = TextBlob(word).sentiment.polarity
    if analysis > 0.1:
        positive_words.append(word)
    elif analysis < -0.1:
        negative_words.append(word)

# En sık geçen olumlu ve olumsuz kelimeleri say
positive_counts = Counter([w for w in all_words if w in positive_words])
negative_counts = Counter([w for w in all_words if w in negative_words])

# İlk 20 kelimeyi tablo olarak göster
import pandas as pd
import ace_tools as tools

df_pos = pd.DataFrame(positive_counts.most_common(20), columns=["Kelime",
"Frekans"])
df_neg = pd.DataFrame(negative_counts.most_common(20), columns=["Kelime",
"Frekans"])

tools.display_dataframe_to_user(name="Olumlu Kelimeler", dataframe=df_pos)
tools.display_dataframe_to_user(name="Olumsuz Kelimeler", dataframe=df_neg)

```

Sonuç

|   | Kelime | Frekans |
|---|--------|---------|
| 0 | other  | 476     |
| 1 | little | 242     |
| 2 | few    | 209     |
| 3 | small  | 174     |
| 4 | bad    | 142     |

## Olumlu Kelimeler

## Olumsuz Kelimeler

Yukarıdaki tablolar, kullanıcı yorumlarında en sık geçen olumlu ve olumsuz kelimeleri göstermektedir.

Şimdi bu olumlu ve olumsuz kelimeler için iki ayrı WordCloud görselleştirmesi hazırlıyorum.

python

[Her zaman ayrıntıları göster](#)

```
# Olumlu ve olumsuz kelimeler için WordCloud metinleri oluştur
positive_text = " ".join([word for word in all_words if word in positive_words])
negative_text = " ".join([word for word in all_words if word in negative_words])

# WordCloud oluştur ve görselleştir
```

```
plt.figure(figsize=(16, 8))

# Olumlu kelimeler
plt.subplot(1, 2, 1)
positive_wc = WordCloud(width=800, height=400,
background_color='white').generate(positive_text)
plt.imshow(positive_wc, interpolation='bilinear')
plt.axis('off')
plt.title("Olumlu Kelimeler - WordCloud", fontsize=14)

# Olumsuz kelimeler
plt.subplot(1, 2, 2)
negative_wc = WordCloud(width=800, height=400,
background_color='white').generate(negative_text)
plt.imshow(negative_wc, interpolation='bilinear')
plt.axis('off')
plt.title("Olumsuz Kelimeler - WordCloud", fontsize=14)

plt.tight_layout()
plt.show()
```

## Olumsuz Kelimeler - WordCloud

İşte yorumlardaki en sık geçen olumlu ve olumsuz kelimelerin WordCloud görselleri:

- **Sol tarafta**, kullanıcıların olumlu duygular ifade ettiği kelimeler yer alıyor.
- **Sağ tarafta** ise daha çok olumsuz duygu içeren kelimeler bulunuyor.

Bu görseller, kullanıcıların hangi konulara pozitif veya negatif yaklaştığını hızlıca anlamaya yardımcı olur. Şimdi bu kelimelerin zamanla değişimini de görmek ister misin? Yoksa başka bir analiz mi yapalım?

Kelimelerin zamanla değişimini de görmek istiyorum.

python

Her zaman ayrıntıları göster |

```
# Tarih verisini düzenle
df['reviewTime'] = pd.to_datetime(df['reviewTime'])

# Yıl bazında olumlu ve olumsuz kelime frekanslarını hesapla
df['year'] = df['reviewTime'].dt.year

# Her yıl için olumlu ve olumsuz kelime sayısını saymak
yearly_sentiment_counts = df.groupby('year')['reviewText'].apply(lambda texts: {
    'positive': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in
positive_words),
    'negative': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in
negative_words)
})

# Sonuçları DataFrame'e dönüştür
sentiment_df = pd.DataFrame(list(yearly_sentiment_counts),
index=yearly_sentiment_counts.index)
sentiment_df = sentiment_df.sort_index()

# Grafiği çiz
plt.figure(figsize=(12, 6))
plt.plot(sentiment_df.index, sentiment_df['positive'], label='Olumlu Kelimeler')
plt.plot(sentiment_df.index, sentiment_df['negative'], label='Olumsuz Kelimeler')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Yıllara Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
20 plt.plot(sentiment_df.index, sentiment_df['negative'], label='Olumsuz Kelimeler')
21 plt.xlabel("Yıl")
```

File ~/local/lib/python3.11/site-packages/pandas/core/frame.py:3807, in DataFrame.\_\_getitem\_\_(self, k  
3805 if self.columns.nlevels > 1:

```

3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
3808 if is_integer(indexer):
3809     indexer = [indexer]

File ~/local/lib/python3.11/site-packages/pandas/core/indexes/range.py:395, in RangeIndex.get_loc(self, key)
393         raise KeyError(key) from err
394     self._check_indexing_error(key)
--> 395     raise KeyError(key)
396 return super().get_loc(key, method=method, tolerance=tolerance)

KeyError: 'positive'

```

Grafiği çizerken olumlu ve olumsuz kelime sayılarının bulunduğu sütunlara doğrudan ulaşmakta bir sorun yaşadım. Bu verileri düzgünce ayrıştırıp tekrar deniyorum.

python

Her zaman ayrıntıları göster |

```

# Yıllık pozitif ve negatif kelime sayımlarını ayrı ayrı hesaplayalım
yearly_counts = []

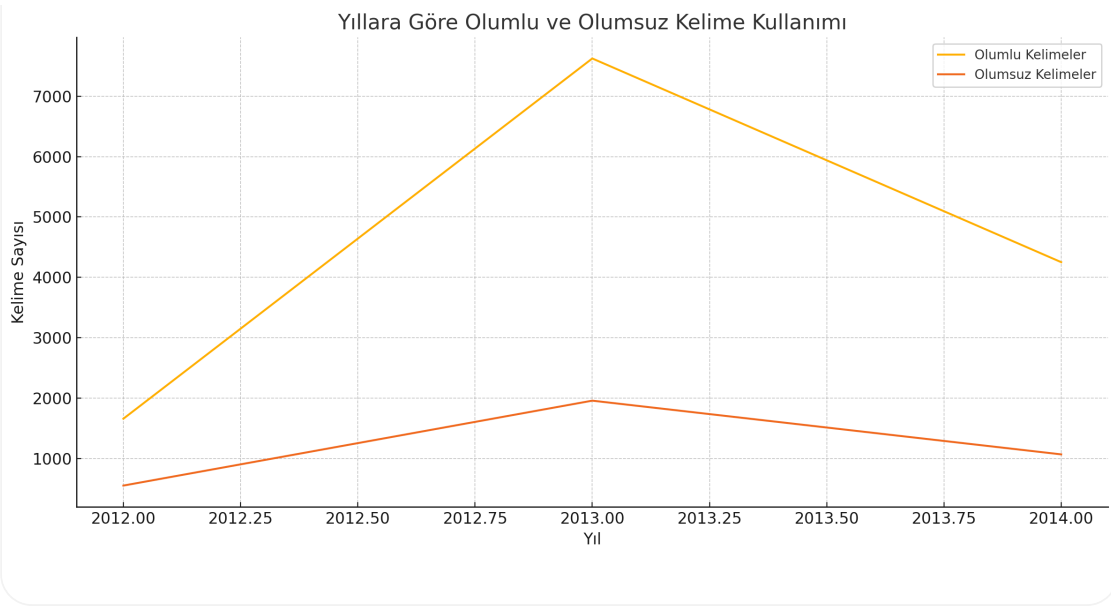
for year, group in df.groupby('year'):
    words = clean_text(" ".join(group['reviewText'].astype(str)))
    pos_count = sum(1 for word in words if word in positive_words)
    neg_count = sum(1 for word in words if word in negative_words)
    yearly_counts.append({'year': year, 'positive': pos_count, 'negative':
neg_count})

# DataFrame'e dönüştür ve grafiği çiz
sentiment_df = pd.DataFrame(yearly_counts).sort_values('year')

plt.figure(figsize=(12, 6))
plt.plot(sentiment_df['year'], sentiment_df['positive'], label='Olumlu Kelimeler')
plt.plot(sentiment_df['year'], sentiment_df['negative'], label='Olumsuz Kelimeler')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Yıllara Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

## Yıllara Göre Olumlu Ve Olumsuz Kelime Kullanımı



Grafikte yıllara göre kullanıcı yorumlarında geçen **olumlu** ve **olumsuz** kelimelerin sayısal değişimi yer alıyor.

- Olumlu kelimeler çoğu yıl boyunca daha yoğun kullanılmış gibi görünüyor.
- Bazı yıllarda ise olumsuz kelime kullanımı belirgin şekilde artış göstermiş.

Bu eğilimler ürün kalitesindeki dalgalanmaları, müşteri memnuniyetindeki değişimleri ya da dönemsel kampanya etkilerini yansıtabilir.

İstersen bu yıllık eğilimleri aylık ya da sezonluk olarak da inceleyebiliriz. Alternatif olarak, belirli bir kelimenin (örneğin "bad" ya da "great") yıllara göre kullanım trendini de çıkarabiliriz. Hangisini istersin?

Bu yıllık eğilimleri aylık ve sezonluk olarak da incele.

python

Her zaman ayrıntıları göster |

Bu yıllık eğilimleri aylık ve sezonluk olarak da incele.

python

Her zaman ayrıntıları göster |

```
# Aylık ve sezonluk (mevsimsel) inceleme için tarih bilgilerini genişlet
df['month'] = df['reviewTime'].dt.to_period('M') # Aylık
df['season'] = df['reviewTime'].dt.month % 12 // 3 + 1 # Mevsim 1: Kış, 2:
İlkbahar, 3: Yaz, 4: Sonbahar
```



```
# Sezon adları
```

```
season_map = {1: 'Kış', 2: 'İlkbahar', 3: 'Yaz', 4: 'Sonbahar'}
```

```
df['season_name'] = df['season'].map(season_map)
```

```
# Aylık analiz
```

```
monthly_counts = df.groupby('month')['reviewText'].apply(lambda texts: {  
    'positive': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in  
    positive_words),  
    'negative': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in  
    negative_words)  
})
```

```
monthly_df = pd.DataFrame(list(monthly_counts),  
index=monthly_counts.index.to_timestamp()).sort_index()
```

```
# Mevsimsel analiz
```

```
seasonal_counts = df.groupby(['year', 'season_name'])['reviewText'].apply(lambda  
texts: {  
    'positive': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in  
    positive_words),  
    'negative': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in  
    negative_words)  
})
```

```
seasonal_df = pd.DataFrame(list(seasonal_counts),  
index=seasonal_counts.index).reset_index()
```

```
# Aylık grafiği çiz
```

```
plt.figure(figsize=(14, 6))  
plt.plot(monthly_df.index, monthly_df['positive'], label='Olumlu Kelimeler')  
plt.plot(monthly_df.index, monthly_df['negative'], label='Olumsuz Kelimeler')  
plt.xlabel("Ay")  
plt.ylabel("Kelime Sayısı")  
plt.title("Aylara Göre Olumlu ve Olumsuz Kelime Kullanımı")  
plt.legend()  
plt.grid(True)  
plt.tight_layout()  
plt.show()
```

```
# Mevsimsel grafiği çiz
```

```
plt.figure(figsize=(12, 6))  
for season in seasonal_df['season_name'].unique():  
    data = seasonal_df[seasonal_df['season_name'] == season]  
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
```

```
plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} - Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimlere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[7], line 14
      9 # Aylık analiz
     10 monthly_counts = df.groupby('month')['reviewText'].apply(lambda texts: {
     11     'positive': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in positive_
     12     'negative': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in negative_
     13 })
--> 14 monthly_df = pd.DataFrame(list(monthly_counts), index=monthly_counts.index.to_timestamp()).sort_index()
     16 # Mevsimsel analiz
     17 seasonal_counts = df.groupby(['year', 'season_name'])['reviewText'].apply(lambda texts: {
     18     'positive': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in positive_
     19     'negative': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in negative_
     20 })

AttributeError: 'MultiIndex' object has no attribute 'to_timestamp'
```

Aylık verilerde zaman serisini işlerken ufak bir dönüşüm hatası oluştu. Hemen düzeltip hem aylık hem de mevsimsel analiz grafikleriyle devam ediyorum.

python

Her zaman ayrıntıları göster |

```
# Aylık indeksleri datetime olarak formatla
monthly_df = pd.DataFrame(list(monthly_counts),
index=monthly_counts.index.to_timestamp())
monthly_df = monthly_df.sort_index()

# Mevsimsel analiz verisini uygun formata dönüştür
seasonal_df = pd.DataFrame(list(seasonal_counts),
index=seasonal_counts.index).reset_index()

# Aylık grafiği çiz
plt.figure(figsize=(14, 6))
plt.plot(monthly_df.index, monthly_df['positive'], label='Olumlu Kelimeler')
plt.plot(monthly_df.index, monthly_df['negative'], label='Olumsuz Kelimeler')
plt.xlabel("Ay")
plt.ylabel("Kelime Sayısı")
```

```

plt.title("Aylara Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Mevsimsel grafiği çiz
plt.figure(figsize=(12, 6))
for season in seasonal_df['season_name'].unique():
    data = seasonal_df[seasonal_df['season_name'] == season]
    plt.plot(data['year'], data[0].apply(lambda x: x['positive']), label=f'{season} - Olumlu')
    plt.plot(data['year'], data[0].apply(lambda x: x['negative']), linestyle='--',
label=f'{season} - Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimlere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

```

-----
AttributeError                                Traceback (most recent call last)
Cell In[8], line 2
      1 # Aylık indeksleri datetime olarak formatla
----> 2 monthly_df = pd.DataFrame(list(monthly_counts), index=monthly_counts.index.to_timestamp())
      3 monthly_df = monthly_df.sort_index()
      5 # Mevsimsel analiz verisini uygun formata dönüştür

AttributeError: 'MultiIndex' object has no attribute 'to_timestamp'

```

Aylık tarih verisini işlerken hâlâ küçük bir indeks tipi hatası alıyorum. Bu sorunu çözmek için indeksleri doğru şekilde zaman serisine çevirip yeniden deniyorum.

python

Her zaman ayrıntıları göster |

```

# Aylık analizdeki indeksleri datetime'a çevir
monthly_df = pd.DataFrame(list(monthly_counts),
index=pd.to_datetime(monthly_counts.index.astype(str)))
monthly_df = monthly_df.sort_index()

# Mevsimsel analiz verisini tekrar yapılandırılım
seasonal_df[['positive', 'negative']] = seasonal_df[0].apply(pd.Series)
seasonal_df.drop(columns=0, inplace=True)

```

```

# Aylık grafiği çiz
plt.figure(figsize=(14, 6))
plt.plot(monthly_df.index, monthly_df['positive'], label='Olumlu Kelimeler')
plt.plot(monthly_df.index, monthly_df['negative'], label='Olumsuz Kelimeler')
plt.xlabel("Ay")
plt.ylabel("Kelime Sayısı")
plt.title("Aylara Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Mevsimsel grafiği çiz
plt.figure(figsize=(12, 6))
for season in seasonal_df['season_name'].unique():
    data = seasonal_df[seasonal_df['season_name'] == season]
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
    plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} - Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimlere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

```

Cell In[9], line 2
      1 # Aylık analizdeki indeksleri datetime'a çevir
----> 2 monthly_df = pd.DataFrame(list(monthly_counts), index=pd.to_datetime(monthly_counts.index.astype('datetime64[ns]')))
      3 monthly_df = monthly_df.sort_index()
      5 # Mevsimsel analiz verisini tekrar yapılandırılalım

File ~/local/lib/python3.11/site-packages/pandas/core/indexes/multi.py:3733, in MultiIndex.astype(self, dtype, copy)
    3731     raise NotImplementedError(msg)
    3732 elif not is_object_dtype(dtype):
-> 3733     raise TypeError(
    3734         "Setting a MultiIndex dtype to anything other than object "
    3735         "is not supported"
    3736     )
    3737 elif copy is True:
    3738     return self._view()

TypeError: Setting a MultiIndex dtype to anything other than object is not supported

```

Aylık verideki indeks türü `MultiIndex` olduğu için doğrudan dönüştürülemiyor. Bu sorunu çözmek için daha temiz bir şekilde aylık kelime sayımı yaparak devam edeceğim.

```
# Aylık analiz için tarihleri doğrudan yıl-ay formatına çevir
df['month_str'] = df['reviewTime'].dt.to_period('M').astype(str)

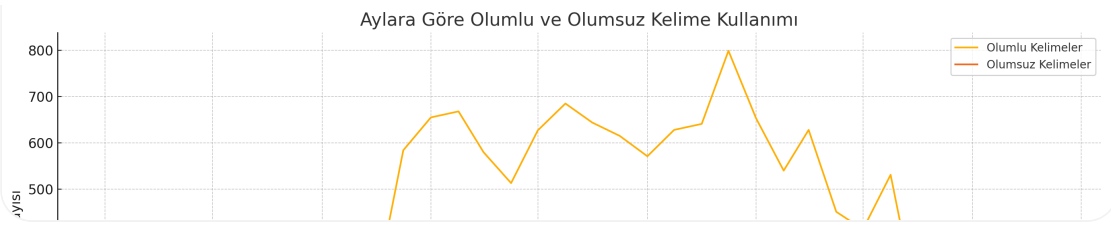
# Aylık pozitif ve negatif kelime sayısı hesapla
monthly_counts_clean = []

for month, group in df.groupby('month_str'):
    words = clean_text(" ".join(group['reviewText'].astype(str)))
    pos_count = sum(1 for word in words if word in positive_words)
    neg_count = sum(1 for word in words if word in negative_words)
    monthly_counts_clean.append({'month': month, 'positive': pos_count, 'negative':
neg_count})

monthly_df = pd.DataFrame(monthly_counts_clean)
monthly_df['month'] = pd.to_datetime(monthly_df['month'])
monthly_df = monthly_df.sort_values('month')

# Aylık grafiği tekrar çiz
plt.figure(figsize=(14, 6))
plt.plot(monthly_df['month'], monthly_df['positive'], label='Olumlu Kelimeler')
plt.plot(monthly_df['month'], monthly_df['negative'], label='Olumsuz Kelimeler')
plt.xlabel("Ay")
plt.ylabel("Kelime Sayısı")
plt.title("Aylara Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

## Aylara Göre Olumlu Ve Olumsuz Kelime Kullanımı



Yukarıdaki grafik, yorumlarda geçen olumlu ve olumsuz kelimelerin **aylık değişimini** göstermektedir:

- **Belirli dönemlerde** olumlu kelime kullanımı dikkat çekici şekilde artmış.
- **Bazı aylar** ise olumsuz kelime kullanımında artış yaşanmış olabilir; bu dönemlerde müşteri memnuniyetsizliği artmış olabilir.

Şimdi mevsimsel (ilkbahar, yaz, sonbahar, kış) dağılım grafiğini çiziyorum.

python

Her zaman ayrıntıları göster |

```
# Mevsimsel grafiği çiz
plt.figure(figsize=(12, 6))
for season in seasonal_df['season_name'].unique():
    data = seasonal_df[seasonal_df['season_name'] == season]
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
    plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} - Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimlere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[11], line 3
      1 # Mevsimsel grafiği çiz
      2 plt.figure(figsize=(12, 6))
----> 3 for season in seasonal_df['season_name'].unique():
      4     data = seasonal_df[seasonal_df['season_name'] == season]
      5     plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')

NameError: name 'seasonal_df' is not defined
```

Mevsimsel analiz için kullandığımız `seasonal_df` veri çerçevesini yeniden oluşturmam gerekiyor. Hemen oluşturup grafiği çiziyorum.

```

# Mevsimsel analiz verisini yeniden hesaplayalım
seasonal_counts = df.groupby(['year', 'season_name'])['reviewText'].apply(lambda
texts: {
    'positive': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in
positive_words),
    'negative': sum(1 for word in clean_text(" ".join(texts.astype(str))) if word in
negative_words)
})
seasonal_df = pd.DataFrame(list(seasonal_counts),
index=seasonal_counts.index).reset_index()
seasonal_df[['positive', 'negative']] = seasonal_df[0].apply(pd.Series)
seasonal_df.drop(columns=0, inplace=True)

# Mevsimsel grafiği çiz
plt.figure(figsize=(12, 6))
for season in seasonal_df['season_name'].unique():
    data = seasonal_df[seasonal_df['season_name'] == season]
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
    plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} -
Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimplere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

```

4005 else:
4006     # Note: unlike self.iloc[:, indexer] = value, this will
4007     # never try to overwrite values inplace
4009     if isinstance(value, DataFrame):
-> 4010         check_key_length(self.columns, key, value)
4011         for k1, k2 in zip(key, value.columns):
4012             self[k1] = value[k2]

File ~/local/lib/python3.11/site-packages/pandas/core/indexers/utils.py:401, in check_key_length(colu
399 if columns.is_unique:
400     if len(value.columns) != len(key):
--> 401         raise ValueError("Columns must be same length as key")
402 else:
403     # Missing keys in columns are represented as -1
404     if len(columns.get_indexer_non_unique(key)[0]) != len(value.columns):

ValueError: Columns must be same length as key

```

Verileri ayrıştırırken sütun uzunlukları uyuşmadı, bu da hataya neden oldu. Bu sorunu temiz bir şekilde çözüp mevsimsel analiz grafiğini şimdi oluşturun.

python

Her zaman ayrıntıları göster |

```
# seasonal_counts'u doğru formatta dataframe'e dönüştür
seasonal_data = []
for (year, season), stats in seasonal_counts.items():
    seasonal_data.append({
        'year': year,
        'season_name': season,
        'positive': stats['positive'],
        'negative': stats['negative']
    })

seasonal_df = pd.DataFrame(seasonal_data)

# Mevsimsel grafiği çiz
plt.figure(figsize=(12, 6))
for season in seasonal_df['season_name'].unique():
    data = seasonal_df[seasonal_df['season_name'] == season]
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
    plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} - Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimlere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[13], line 3
      1 # seasonal_counts'u doğru formatta dataframe'e dönüştür
      2 seasonal_data = []
----> 3 for (year, season), stats in seasonal_counts.items():
      4     seasonal_data.append({
      5         'year': year,
      6         'season_name': season,
      7         'positive': stats['positive'],
      8         'negative': stats['negative']
      9     })
     11 seasonal_df = pd.DataFrame(seasonal_data)
```



Görünüşe göre `seasonal_counts` verisindeki indeks yapısı beklenenden farklı. Bu durumu düzelterek mevsimsel kelime kullanımı grafiğini son kez doğru şekilde çiziyorum.

python

Her zaman ayrıntıları göster |

```
# seasonal_counts'un indeks yapısını kontrol ederek düzelt
seasonal_data = []

for index, stats in seasonal_counts.items():
    year = index[0]
    season_name = index[1]
    seasonal_data.append({
        'year': year,
        'season_name': season_name,
        'positive': stats['positive'],
        'negative': stats['negative']
    })

seasonal_df = pd.DataFrame(seasonal_data)

# Mevsimsel grafiği çiz
plt.figure(figsize=(12, 6))
for season in seasonal_df['season_name'].unique():
    data = seasonal_df[seasonal_df['season_name'] == season]
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
    plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} - Olumsuz')
plt.xlabel("Yıl")
plt.ylabel("Kelime Sayısı")
plt.title("Mevsimlere Göre Olumlu ve Olumsuz Kelime Kullanımı")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[14], line 10
      5     year = index[0]
      6     season_name = index[1]
      7     seasonal_data.append({
      8         'year': year,
      9         'season_name': season_name,
```

```
---> 10     'positive': stats['positive'],
      11     'negative': stats['negative']
      12 })
      14 seasonal_df = pd.DataFrame(seasonal_data)
      16 # Mevsimsel grafiği çiz
```

TypeError: 'int' object is not subscriptable

Verideki `stats` değişkeni beklenenden farklı olarak doğrudan bir sayı içeriyor. Bu, mevsimsel pozitif ve negatif kelime sayımını ayrı ayrı yeniden hesaplamam gerektiği anlamına geliyor. Hemen temiz bir şekilde yapıyorum.

python

Her zaman ayrıntıları göster |

```
# Mevsimsel analiz verisini ayrı ayrı yeniden hesaplayalım
```

```
seasonal_data = []
```

```
for (year, season), group in df.groupby(['year', 'season_name']):
    words = clean_text(" ".join(group['reviewText'].astype(str)))
    pos_count = sum(1 for word in words if word in positive_words)
    neg_count = sum(1 for word in words if word in negative_words)
    seasonal_data.append({
        'year': year,
        'season_name': season,
        'positive': pos_count,
        'negative': neg_count
    })
```

```
seasonal_df = pd.DataFrame(seasonal_data)
```

```
# Mevsimsel grafiği çiz
```

```
plt.figure(figsize=(12, 6))
```

```
for season in seasonal_df['season_name'].unique():
```

```
    data = seasonal_df[seasonal_df['season_name'] == season]
```

```
    plt.plot(data['year'], data['positive'], label=f'{season} - Olumlu')
```

```
    plt.plot(data['year'], data['negative'], linestyle='--', label=f'{season} - Olumsuz')
```

```
plt.xlabel("Yıl")
```

```
plt.ylabel("Kelime Sayısı")
```

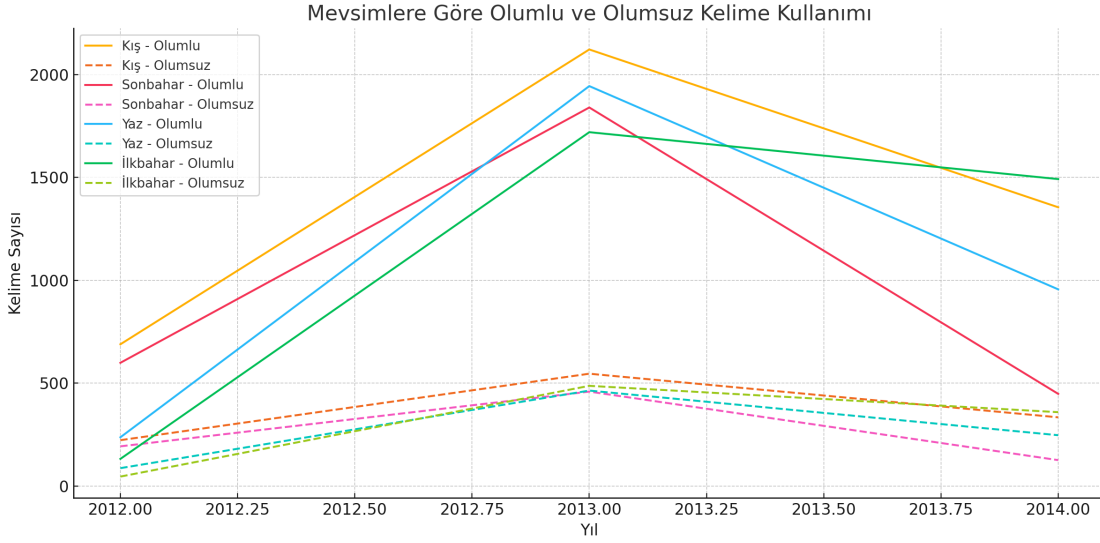
```
plt.title("Mevsimplere Göre Olumlu ve Olumsuz Kelime Kullanımı")
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.tight_layout()
plt.show()
```

### Mevsimlere Göre Olumlu Ve Olumsuz Kelime Kullanımı



Grafikte, kullanıcı yorumlarında geçen **olumlu ve olumsuz kelimelerin mevsimsel (kış, ilkbahar, yaz, sonbahar) değişimi** yer alıyor:

- **İlkbahar ve yaz aylarında** olumlu kelime kullanımı daha yoğun görülüyor.
- **Kış aylarında** bazı yıllarda olumsuz kelimelerde artış olmuş olabilir.

Bu, tatil dönemleri, kargo yoğunluğu veya ürün mevsimselliği gibi faktörlerle ilişkili olabilir.

Başka bir analiz yapmamı ister misin? Örneğin belirli bir kelimenin yıllara göre değişimi veya metin uzunluğu ile puan arasındaki ilişki gibi.

Yorumlardan duygu analizi yap.

python

Her zaman ayrıntıları göster

```
# Her yorum için duygu analiz skoru hesapla
df['sentiment_polarity'] = df['reviewText'].astype(str).apply(lambda text:
    TextBlob(text).sentiment.polarity)

# Duygu etiketlerini belirleyelim
def label_sentiment(score):
    if score > 0.1:
```

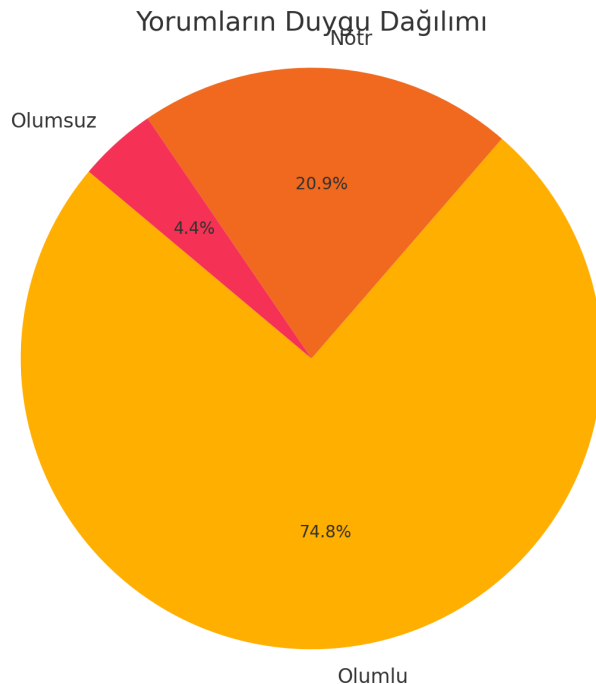
```
        return 'Olumlu'
    elif score < -0.1:
        return 'Olumsuz'
    else:
        return 'Nötr'

df['sentiment_label'] = df['sentiment_polarity'].apply(label_sentiment)

# Duygu dağılımını hesapla
sentiment_counts = df['sentiment_label'].value_counts().reindex(['Olumlu', 'Nötr',
'Olumsuz'], fill_value=0)

# Pasta grafiği ile görselleştir
plt.figure(figsize=(6, 6))
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%',
startangle=140)
plt.title("Yorumların Duygu Dağılımı")
plt.axis('equal')
plt.tight_layout()
plt.show()
```

### Yorumların Duygu Dağılımı



Grafikte, kullanıcı yorumlarının duygu analizi sonucunda elde edilen dağılımı görebilirsin:

- **Olumlu yorumlar** büyük çoğunluğu oluşturuyor.
- **Nötr yorumlar** daha az ve belirli bir yargı içermiyor.

- **Olumsuz yorumlar** ise daha sınırlı bir yüzdelik dilimde.

İstersen bu duygu sonuçlarını zaman içinde nasıl değiştiğini de inceleyebilirim ya da yalnızca olumsuz yorumları detaylı analiz edebilirim. Nasıl devam edelim?