

```
In [40]: import pandas as pd
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize

import re
import csv
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# NLTK verilerini indir (ilk çalıştırmada gerekli)
nltk.download('punkt', quiet=True)
```

Out[40]: True

```
In [39]: !pip install wordcloud
```

```
Collecting wordcloud
  Obtaining dependency information for wordcloud from https://files.pythonhosted.org/packages/00/09/abb305dce85911b8fba382926cfc57f2f257729e25937fdcc63f3a1a67f9/wordcloud-1.9.4-cp311-cp311-win_amd64.whl.metadata
  Downloading wordcloud-1.9.4-cp311-cp311-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: numpy>=1.6.1 in c:\users\ademt\anaconda3\lib\site-packages (from wordcloud) (1.24.3)
Requirement already satisfied: pillow in c:\users\ademt\anaconda3\lib\site-packages (from wordcloud) (10.0.1)
Requirement already satisfied: matplotlib in c:\users\ademt\anaconda3\lib\site-packages (from wordcloud) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (23.1)
Requirement already satisfied: pyparsing<3.1, >=2.3.1 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ademt\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\ademt\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Downloading wordcloud-1.9.4-cp311-cp311-win_amd64.whl (299 kB)
----- 0.0/299.9 kB ? eta -:-:--
- ----- 10.2/299.9 kB ? eta -:-:--
- ----- 10.2/299.9 kB ? eta -:-:--
----- 41.0/299.9 kB 326.8 kB/s eta 0:00:01
----- 92.2/299.9 kB 655.4 kB/s eta 0:00:01
----- 174.1/299.9 kB 952.6 kB/s eta 0:00:01
----- 256.0/299.9 kB 1.1 MB/s eta 0:00:01
----- 299.9/299.9 kB 1.2 MB/s eta 0:00:00
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.4
```

```
In [41]: # Türkçe stopwords listesi
turkish_stopwords = set([
    've', 'ile', 'de', 'da', 'ki', 'kadar', 'için', 'ama', 'ya', 'veya',
    'bir', 'bu', 'şu', 'o', 'ne', 'nasıl', 'niye', 'hangi', 'her', 'tüm',
    'mi', 'mı', 'mu', 'mü', 'ise', 'değil', 'çok', 'az'
])
print("Türkçe Stopwords (ilk 50):")
print(list(turkish_stopwords)[:50])
```

```
Türkçe Stopwords (ilk 50):
['ama', 'veya', 'nasıl', 'mü', 'her', 'tüm', 'için', 'o', 'ile', 'de', 'niye', 'hangi', 'ki', 'mı', 'da', 'kadar', 'bu', 'çok', 'şu', 'mu', 'ise', 'az', 'değil', 'mi', 'ya', 'ne', 'bir', 've']
```

```
In [45]: # Basit Türkçe stemleme fonksiyonu
def simple_turkish_stem(token):
    # Yaygın Türkçe son ekleri kaldır
    suffixes = ['ler', 'lar', 'in', 'ın', 'un', 'ün', 'de', 'da', 'ki', 'e', 'a']
    for suffix in suffixes:
        if token.endswith(suffix):
            return token[:-len(suffix)]
    return token
```

```
In [46]: # Veri setini yükle
df = pd.read_csv('C:/Users/ademt/Desktop/tarim_problemleri_veriseti.csv', encoding='utf-8')
texts = df['sorun_metin'].tolist()
```

```
In [49]: df
```

Out [49]:		id	sorun_metin
	0	1	Bu yaz sezonunda şeftali tarlamızda yapraklard...
	1	2	mısır tarlamızda genç fidelerde çekiş kaybı ya...
	2	3	patates tarlamızda genç fidelerde genç sürgünl...
	3	4	Bu yaz kurak geçtiği için sığırların genç sürg...
	4	5	Son hasatta traktörümüzün çekiş kaybı yaşanması...
	...	...	...
	195	196	Bu yaz kurak geçtiği için sığırların genç fide...
	196	197	elma bahçemde genç fidelerde sararma ve gelişi...
	197	198	patates bahçemde yapraklarda kahverengileşme v...
	198	199	üzüm tarlamızda genç fidelerde yapraklarda kah...
	199	200	elma tarlamızda bıçakların sık sık tıkanması. ...

200 rows × 2 columns

```
In [47]: # İlk 500 karakteri göster
print("\nVerinin ilk 500 karakteri:")
print(''.join([str(text)[:500] for text in texts if isinstance(text, str)][1:]))
```

Verinin ilk 500 karakteri:

Bu yaz sezonunda şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme. Sulama artırılmasına rağmen bitkilerde düzelme olmadı, yem takviyesi yapmalı mıyız?

```
In [50]: # Cümlelere ayır
sentences = []
for text in texts:
    if isinstance(text, str):
        sentences.extend(sent_tokenize(text))
    else:
        continue
```

```
In [51]: # İlk 10 cümleyi göster
print("\nİlk 10 cümle:")
print(sentences[:10])
```

İlk 10 cümle:

['Bu yaz sezonunda şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme.', 'Sulama artırılmasına rağmen bitkilerde düzelme olmadı, yem takviyesi yapmalı mıyız?', 'mısır tarlamızda genç fidelerde çekiş kaybı yaşanması.', 'aktarma organlarında sorun kaynaklı olabilir mi?', 'patates tarlamızda genç fidelerde genç sürgünlerde kuru malar ve yaprak dökümü.', 'azot eksikliği kaynaklı olabilir mi?', 'Bu yaz kurak geçtiği için sığırların genç sürgünlerde kurumalar ve yaprak dökümü.', 'gübreleme programını yeniden düzenlemeli miyim?', 'Son hasatta traktörümüzün çekiş kaybı yaşanması.', 'gübreleme programını yeniden düzenlemeli miyim?']

```
In [52]: # Ön işleme fonksiyonu (girinti düzeltilmiş)
def preprocess_sentence(sentence):
    if not isinstance(sentence, str):
        return [], []

    tokens = word_tokenize(sentence)
    filtered_tokens = [
        token.lower() for token in tokens
        if token.isalpha() and token.lower() not in turkish_stopwords
    ]

    # Lemmatizasyon (Zembereksiz: filtrelenmiş kelimeler)
    lemmatized_tokens = filtered_tokens

    # Stemleme (basit kural tabanlı)
    stemmed_tokens = [simple_turkish_stem(token) for token in filtered_tokens]

    return lemmatized_tokens, stemmed_tokens
```

```
In [53]: # Örnek cümleyle fonksiyon çıktısını göster
ornek_cumle = "Bu yaz şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme."
lemmatized, stemmed = preprocess_sentence(ornek_cumle)
print("\nÖrnek Cümle Çıktısı:")
print(f"Ham Cümle: {ornek_cumle}")
print(f"Lemmatized Tokens: {lemmatized}")
print(f"Stemmed Tokens: {stemmed}")
```

Örnek Cümle Çıktısı:

Ham Cümle: Bu yaz şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme.

Lemmatized Tokens: ['yaz', 'şeftali', 'tarlamızda', 'yapraklarda', 'kahverengileşme', 'büzüşme']

Stemmed Tokens: ['yaz', 'şeftali', 'tarlamız', 'yapraklar', 'kahverengileşm', 'büzüşm']

```
In [54]: # Cümleleri işle (fonksiyonlu yaklaşım)
tokenized_corpus_lemmatized = []
tokenized_corpus_stemmed = []
for sentence in sentences:
    lemmatized_tokens, stemmed_tokens = preprocess_sentence(sentence)
```

```
if lemmatized_tokens:
    tokenized_corpus_lemmatized.append(lemmatized_tokens)
    tokenized_corpus_stemmed.append(stemmed_tokens)
```

```
# İlk işlenen cümlelerin çıktısını göster (fonksiyonlu)
print("\nİlk İşlenen Cümle (Fonksiyonlu):")
print(f"Ham: {sentences[0]}")
print(f"Lemmatized: {tokenized_corpus_lemmatized[0]}")
print(f"Stemmed: {tokenized_corpus_stemmed[0]}")
```

İlk İşlenen Cümle (Fonksiyonlu):

Ham: Bu yaz sezonunda şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme.

Lemmatized: ['yaz', 'sezonunda', 'şeftali', 'tarlamızda', 'yapraklarda', 'kahverengileşme', 'büzüşme']

Stemmed: ['yaz', 'sezonun', 'şeftali', 'tarlamız', 'yapraklar', 'kahverengileşm', 'büzüşm']

In [55]: # Ayrıntılı for döngüsü ile işleme

```
filtered_sentences = []
for sentence in sentences:
    tokens = word_tokenize(sentence)
    filtered_tokens = []
    for token in tokens:
        if token.isalpha():
            token_lower = token.lower()
            if token_lower not in turkish_stopwords:
                filtered_tokens.append(token_lower)
    filtered_sentences.append(filtered_tokens)
```

In [56]: # Lemmatizasyon (ayrıntılı, Zembereksiz: filtrelenmiş kelimeler)

```
tokenized_corpus_lemmatized_detailed = filtered_sentences
```

```
# Stemleme (ayrıntılı)
```

```
tokenized_corpus_stemmed_detailed = []
for filtered_tokens in filtered_sentences:
    stemmed_tokens = [simple_turkish_stem(token) for token in filtered_tokens]
    if stemmed_tokens:
        tokenized_corpus_stemmed_detailed.append(stemmed_tokens)
```

In [57]: # İlk işlenen cümlelerin çıktısını göster (ayrıntılı)

```
print("\nİlk İşlenen Cümle (Ayrıntılı):")
print(f"Ham: {sentences[0]}")
print(f"Lemmatized: {tokenized_corpus_lemmatized_detailed[0]}")
print(f"Stemmed: {tokenized_corpus_stemmed_detailed[0]}")
```

İlk İşlenen Cümle (Ayrıntılı):

Ham: Bu yaz sezonunda şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme.

Lemmatized: ['yaz', 'sezonunda', 'şeftali', 'tarlamızda', 'yapraklarda', 'kahverengileşme', 'büzüşme']

Stemmed: ['yaz', 'sezonun', 'şeftali', 'tarlamız', 'yapraklar', 'kahverengileşm', 'büzüşm']

In [58]: # Lemmatize edilmiş cümleleri CSV'ye kaydet

```
with open("lemmatized_sentences.csv", mode="w", newline="", encoding="utf-8") as file:
    writer = csv.writer(file)
    for tokens in tokenized_corpus_lemmatized:
        writer.writerow([' '.join(tokens)])
```

In [59]: # Stemlenmiş cümleleri CSV'ye kaydet

```
with open("stemmed_sentences.csv", mode="w", newline="", encoding="utf-8") as file:
    writer = csv.writer(file)
    for tokens in tokenized_corpus_stemmed:
        writer.writerow([' '.join(tokens)])
```

In [60]: # İlk 5 cümleyi yazdır

```
print("\nİlk 5 Cümle Karşılaştırması:")
for i in range(min(5, len(sentences))):
    print(f"Cümle {i+1} - Ham: {sentences[i]}")
    print(f"Cümle {i+1} - Lemmatized: {tokenized_corpus_lemmatized[i]}")
    print(f"Cümle {i+1} - Stemmed: {tokenized_corpus_stemmed[i]}")
    print()
```

İlk 5 Cümle Karşılaştırması:

Cümle 1 - Ham: Bu yaz sezonunda şeftali tarlamızda yapraklarda kahverengileşme ve büzüşme.

Cümle 1 - Lemmatized: ['yaz', 'sezonunda', 'şeftali', 'tarlamızda', 'yapraklarda', 'kahverengileşme', 'büzüşme']

Cümle 1 - Stemmed: ['yaz', 'sezonun', 'şeftali', 'tarlamız', 'yapraklar', 'kahverengileşm', 'büzüşm']

Cümle 2 - Ham: Sulama artırılmasına rağmen bitkilerde düzelme olmadı, yem takviyesi yapmalı mıyız?

Cümle 2 - Lemmatized: ['sulama', 'artırılmasına', 'rağmen', 'bitkilerde', 'düzelm', 'olmadı', 'yem', 'takviyesi', 'yapmalı', 'mıyız']

Cümle 2 - Stemmed: ['sulam', 'artırılmasın', 'rağmen', 'bitkiler', 'düzelm', 'olmadı', 'yem', 'takviyesi', 'yapmalı', 'mıyız']

Cümle 3 - Ham: mısır tarlamızda genç fidelerde çekiş kaybı yaşanması.

Cümle 3 - Lemmatized: ['mısır', 'tarlamızda', 'genç', 'fidelerde', 'çekiş', 'kayı', 'yaşanması']

Cümle 3 - Stemmed: ['mısır', 'tarlamız', 'genç', 'fideler', 'çekiş', 'kayı', 'yaşanması']

Cümle 4 - Ham: aktarma organlarında sorun kaynaklı olabilir mi?

Cümle 4 - Lemmatized: ['aktarma', 'organlarında', 'sorun', 'kaynaklı', 'olabilir']

Cümle 4 - Stemmed: ['aktarm', 'organların', 'sor', 'kaynaklı', 'olabilir']

Cümle 5 - Ham: patates tarlamızda genç fidelerde genç sürgünlerde kurumalar ve yaprak dökümü.

Cümle 5 - Lemmatized: ['patates', 'tarlamızda', 'genç', 'fidelerde', 'genç', 'sürgünlerde', 'kurumalar', 'yaprak', 'dökümü']

Cümle 5 - Stemmed: ['patates', 'tarlamız', 'genç', 'fideler', 'genç', 'sürgünler', 'kuruma', 'yaprak', 'dökümü']

```
In [61]: # Kelime bulutu oluştur
all_lemmatized_text = ' '.join([' '.join(tokens) for tokens in tokenized_corpus_lemmatized])
wordcloud = WordCloud(width=800, height=400, background_color='white', font_path=None).generate(all_lemmatized_text)
```

```
In [62]: # Kelime bulutunu kaydet
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.savefig('kelime_bulut.png')
plt.close()
```

```
In [63]: # Veri boyutu raporlama
raw_size = sum(len(text) for text in texts if isinstance(text, str))
cleaned_size = len(all_lemmatized_text)
print(f"Ham veri boyutu (karakter): {raw_size}")
print(f"Temizlenmiş veri boyutu (karakter): {cleaned_size}")
print(f"Boyut azalması: {(raw_size - cleaned_size) / raw_size * 100}%.2f%")
```

Ham veri boyutu (karakter): 23877

Temizlenmiş veri boyutu (karakter): 22235

Boyut azalması: 6.88%

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js