

Reddit Analytics

Reddit platformundan çekilen verilerle veri analizi gerçekleştirme işlemleri için Aşağıdaki adımların takip edilmesi gerekmektedir.

Siteler ve kaynaklar:

Tutorial video: <https://www.youtube.com/watch?v=nssOuD9EcVk>

Reddit developer: <https://www.reddit.com/prefs/apps>

Reddit API Documentation: <https://www.reddit.com/dev/api/>

Pawg Library Documentation: <https://praw.readthedocs.io/en/latest/>

Proje kodları:



reddit.rar

Adımlar

1-Tutorial video bağlantısını takip ederek ilerleyiniz.

2-Öncelikle bir Reddit hesabınızın olması gerekmektedir. Websitesinde basitçe hesap açabilirsiniz.



3-Sonrasında arka planda Jupyter Notebook açık iken Reddit developer hesabına girip yeni bir proje oluşturuyoruz. Create applicaion dedikten sonra ismini koyup alttaki butonu “script” seçeneğinde çekip açık jupyter notebook adresini de redirect url kısmına kopyalıyoruz.

manage my riar
Developer: reddit

create application

By creating an app, you agree to Reddit's Developer Terms and Data Api Terms. You must also register to use the API.

name my-app

☐ web app A web based application
☐ installed app An app intended for installation, such as on a mobile phone
☒ script Script for personal use. Will only have access to the developers accounts

description

about url

redirect uri http://localhost:

create app

http://localhost:8080

Get Reddit API key 0.58

about
blog
about
advertising


help
site rules
Reddit help center
reddiquette

apps & tools
Reddit for iPhone
Reddit for Android
mobile website

<3
reddit premium
reddit coins

Ve API sayfamız oluşuyor. Burada persone api adreslerimiz ve isimimiz gibi bilgiler yer almaktadır. Bunları ipynb python dosyamıza belirtile kısımlara yazacağız.

developed applications

 **my-app**
personal use script
R355m7JOp6-TswgScCzkvQ

change icon

secret 9D3hVOcM4WmXCa8uytGOFT_6_BqbRg

developers Brilliant_Sock_2545 (that's you!) remove

name my-app

add developer:

description

about url

redirect uri http://localhost:8888

update app delete app

4-Python dosyamıza “pip install praw” yazıp çalıştırarak gerekli kütüphaneyi indirip import ediyoruz. Sonrasında gerekli kodları yazıyoruz. İmporttan sonraki hücrede

açtığımız proje api'sinde kendi reddit api'mizi bağliyoruz gerekli bilgileri doldurarak.

```
pip install praw

import praw

reddit = praw.Reddit(
    client_id = "R355m7j0p6-TswgScCzkvQ",
    client_secret = "9D3hVOcM4WmXCa8uytGOfT_6_BqbRg",
    user_agent = "my-app by u/Brilliant_Sock_2545 ",
    username = "Brilliant_Sock_2545",
    password = "Brilliant_Sock_2545",
)

subreddit = reddit.subreddit("python")

top_posts = subreddit.top(limit = 10)
new_posts = subreddit.new(limit = 10)

for post in top_posts:
    print("Title -", post.title)
    print("ID-", post.id)
    print("Author -", post.author)
    print("URL - ", post.url)
    print("Score - ", post.score)
    print("Comment count - ", post.num_comments)
    print("Created - ", post.created_utc)
    print("\n")
    print("*****")
```

5.Eğer api bilgileriniz doğru ise sonraki hücrede çekmek istediğimiz verilerin sayısını türü gibi bilgileri istiyoruz kurduğumuz kütüphane yardımı ile. Sonrasında başarılı şekilde verileri çekiyoruz.

```
Title - Lad wrote a Python script to download Alexa voice recordings, he didn't expect this email.
ID- g53lxf
Author - iFslam
URL - https://i.redd.it/2s0dj8ob12u41.png
Score - 12341
Comment count - 133
Created - 1587424299.0

*****

Title - This post has:
ID- hoolsm
Author - Krukerfluk
URL - https://www.reddit.com/r/Python/comments/hoolsm/this_post_has/
Score - 9234
Comment count - 435
Created - 1594386373.0

*****

Title - I redesign the Python logo to make it more modern
ID- gftej
Author - jessjwilliamson
URL - https://i.redd.it/rxezjyf4ojx41.png
Score - 7864
Comment count - 265
Created - 1588945149.0
```

Sonraki kısımda twitter analitiğinde istenen görevleri verileri çekerek praw kütüphanesinin uygun fonksiyonları ile birebir aynısını yapmaya çalışacağız. Bunun için dökümantasyonlardan faydalanabilirsiniz.

Bu kısma kadar verileri çekti. Şimdi de Reddit üzerinden çektiğimiz verilerle Text mining işlemleri olan kelime frekansı, görselleştirme, wordcloud oluşturacağız ve sonlarda da yorum ve post metinlerine göre ufak bir duygu analizi projesi yapacağız.

6-Öncelikle belirlediğimiz bir kullanıcının gönderi ve yorumlarındaki metin verilerini çekip kelime frekansını gösteren bir proje ile başlayalım. Sırası ile önce bir username giriyoruz. Sonra kullanıcıdan yorum ve gönderilerin metinlerini topluyoruz. Sonraki aşamada bu metinleri birleştirip daha sonra parçalarına yani kelime tokenlarına ayırıyoruz. Tabi bu aşamada harf sayı gibi ifadeleri de temizliyoruz koddan. Son olarak Frekansını almak için en aşağıdaki fonksiyonu kullanıyoruz.

Kod ve sonuçlar alttaki görseldeki gibidir. Attığım dosyada da kodları inceleyebilirsiniz.

```
from collections import Counter
import re

#KELİME FREKANSI

# Add a username
username = 'Brilliant_Sock_2545'

# collect user's comments and responses
user = reddit.redditor(username)
posts = user.submissions.new(limit=None)
comments = user.comments.new(limit=None)

# merge all text from comments and responses
all_texts = ''
for post in posts:
    all_texts += post.title + ' ' + post.selftext + ' '
for comment in comments:
    all_texts += comment.body + ' '

words = re.findall(r'\b\w+\b', all_texts.lower())
word_counts = Counter(words)

# word frequency
for word, count in word_counts.most_common(20):
    print(f'{word}: {count}')

veri: 28
ve: 20
bilimi: 10
bu: 8
için: 5
bir: 5
elde: 4
gibi: 4
analiz: 4
analizi: 4
```

7-Şimdi aldığımız kelime frekans sonuçlarını daha iyi gözlemleyebilmek adına görselleştirelim. Bir önceki koddan yaptığımızı tekrardan yapıp bu sefer en yüksek tekrarı olan 20 kelimeyi alıp plot yardımı ile görselleştiriyoruz.

```
#Data Visualization

import matplotlib.pyplot as plt
import seaborn as sns

# add username
username = 'Brilliant_Sock_2545'

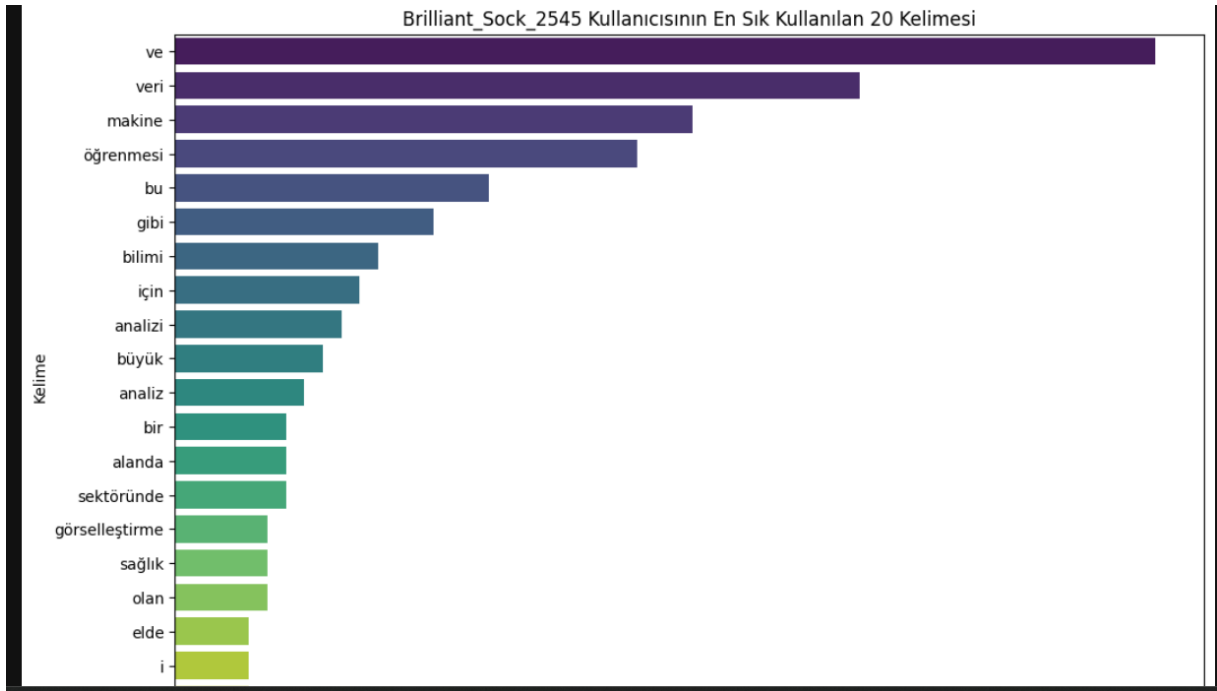
# collect all user's posts and comments
user = reddit.redditor(username)
posts = user.submissions.new(limit=None)
comments = user.comments.new(limit=None)

# merge all texts from comments and posts
all_texts = ''
for post in posts:
    all_texts += post.title + ' ' + post.selftext + ' '
for comment in comments:
    all_texts += comment.body + ' '

# split and clean words process
words = re.findall(r'\b\w+\b', all_texts.lower())
word_counts = Counter(words)

# take the most frequency 20 words
most_common_words = word_counts.most_common(20)
words, counts = zip(*most_common_words)

# Visualization word frequency
plt.figure(figsize=(12, 8))
sns.barplot(x=list(counts), y=list(words), palette='viridis')
plt.xlabel('Frekans')
plt.ylabel('Kelime')
plt.title(f'{username} Most used 20 words')
plt.show()
```



8.Şimdi de kelime tekrarlarından yola çıkarak wordcloud görselleştirmesi yapalım yine veri toplama ve ön işleme işlemleri ile aynı yere gelip bu sefer de wordcloud oluşturup kelimeleri son kısımda aktarıyoruz.

```
#Visualization with wordCloud

from wordcloud import WordCloud

username = 'Brilliant_Sock_2545'

# collect all user's comments and post as text
user = reddit.redditor(username)
posts = user.submissions.new(limit=None)
comments = user.comments.new(limit=None)

# Merge all collected text
all_texts = ''
for post in posts:
    all_texts += post.title + ' ' + post.selftext + ' '
for comment in comments:
    all_texts += comment.body + ' '

# Split and clean process
words = re.findall(r'\b\w+\b', all_texts.lower())
word_counts = Counter(words)

# create wordcloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(word_counts)

# Visualizate wordcloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title(f'{username} Kullanıcısının Word Cloud')
plt.show()
```



9.Bu seferki amacımız wordcloud'u bir resim dosyasının içine aktarma işlemidir. Özellikle alttaki kütüphane ile eksik dosyaları indirelim.

```
pip install wordcloud matplotlib numpy pillow
```

Şimdi de import işlemleri yapıp önceki gibi verileri çekip önışleyerek wordcloud oluşturalım. Öncekine göre fark resim dosyası eklemesi yapıp son koddaki görselleştirme işlemi uygulamaktır.(kullandığım görsel aynı rar dosyasının içindedir.)

10.Şimdi de Duygu Analizi kısmına gelebiliriz. Bunun için öncelikle textBlob ve vaderSentiment kütüphanelerini indirmemiz gerek.

```
pip install textblob vaderSentiment
```

Buradaki işlem ise veriyi text olarak çektikten hemen sonra emotion analyze fonksiyonunu çağırıp metinleri analiz etmek. Burada vader ile analiz edeceğiz. Sonrasında sonuçları yazdıracağız.

```
username = 'Unexpected'

# collect user's all comment and post as text
user = reddit.redditor(username)
posts = user.submissions.new(limit=None)
comments = user.comments.new(limit=None)

# Emotion Analyze function
def analyze_sentiment_vader(text):
    analyzer = SentimentIntensityAnalyzer()
    scores = analyzer.polarity_scores(text)
    return scores['compound'], scores['pos'], scores['neu'], scores['neg']

# Analyze post and comments texts
all_texts = ''
for post in posts:
    all_texts += post.title + ' ' + post.selftext + ' '
for comment in comments:
    all_texts += comment.body + ' '

# Emotion analyze with Vader
compound, pos, neu, neg = analyze_sentiment_vader(all_texts)

# results
print(f'Genel Duygu (Compound): {compound}')
print(f'Pozitif Duygu: {pos}')
print(f'Nötr Duygu: {neu}')
print(f'Negatif Duygu: {neg}')

# Generally emotion
if compound >= 0.05:
    genel_duygu = 'Pozitif'
elif compound <= -0.05:
    genel_duygu = 'Negatif'
else:
    genel_duygu = 'Nötr'

print(f'Genel Duygu Yorumu: {genel_duygu}')
```

```
Genel Duygu (Compound): 1.0
Pozitif Duygu: 0.135
Nötr Duygu: 0.788
Negatif Duygu: 0.077
Genel Duygu Yorumu: Pozitif
```


11.Şimdi de iki farklı kütüphane olan TextBlob ve Vader kullanarak duygu analizi yapalım. Kodlar ve sonuçlar aşağıdaki gibidir.

```
#Particularly emotion analyze

from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

username = 'Unexpected'

user = reddit.redditor(username)
posts = user.submissions.new(limit=None)
comments = user.comments.new(limit=None)

# emotion analyze dunction
def analyze_sentiment_textblob(text):
    analysis = TextBlob(text)
    return analysis.sentiment.polarity, analysis.sentiment.subjectivity

def analyze_sentiment_vader(text):
    analyzer = SentimentIntensityAnalyzer()
    scores = analyzer.polarity_scores(text)
    return scores['compound'], scores['pos'], scores['neu'], scores['neg']

# analyze text from comments and posts
all_texts = ''
for post in posts:
    all_texts += post.title + ' ' + post.selftext + ' '
for comment in comments:
    all_texts += comment.body + ' '

# emotion analyze with textblob
polarity, subjectivity = analyze_sentiment_textblob(all_texts)
print(f'TextBlob Polarity: {polarity}, Subjectivity: {subjectivity}')

# emotion analyze with vader
compound, pos, neu, neg = analyze_sentiment_vader(all_texts)
print(f'VADER Compound: {compound}, Positive: {pos}, Neutral: {neu}, Negative: {neg}')
```

```
TextBlob Polarity: 0.11961980872700215, Subjectivity: 0.48322713833184644
VADER Compound: 1.0, Positive: 0.135, Neutral: 0.788, Negative: 0.077
```

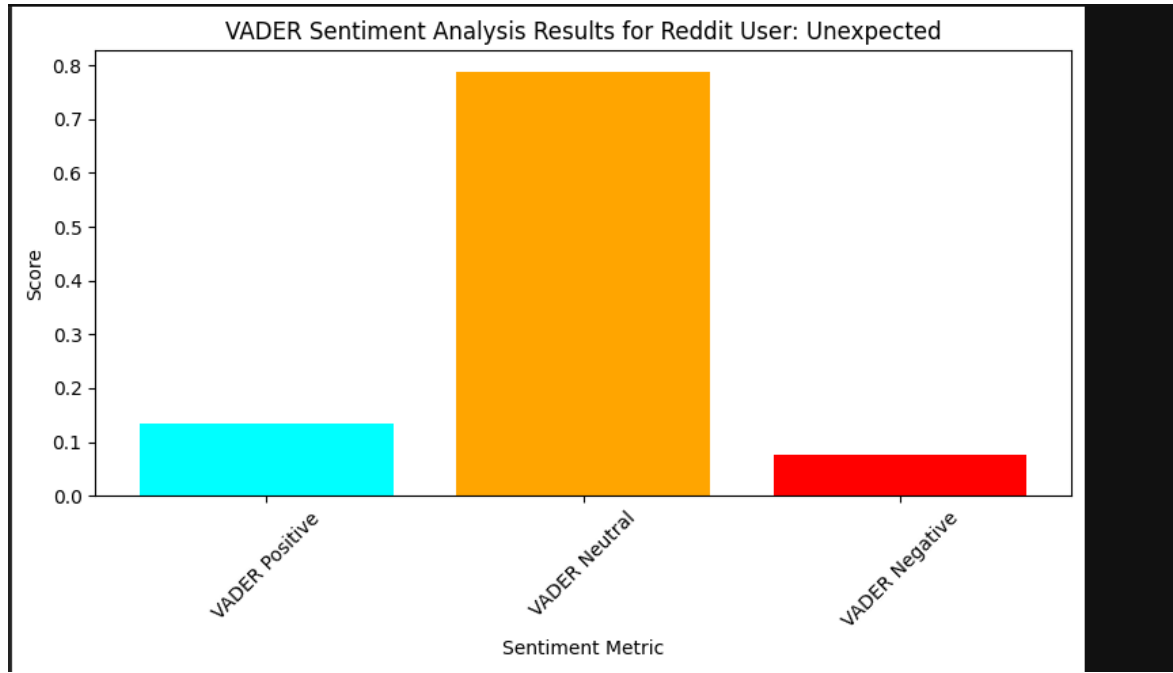
Bu sonuçları da görselleştirelim.

```
import matplotlib.pyplot as plt

# Kodunuzdan aldığınız VADER duygu analizi sonuçları
vader_compound = compound
vader_pos = pos
vader_neu = neu
vader_neg = neg

# Duygu analizi sonuçlarını görselleştirme
labels = [ 'VADER Positive', 'VADER Neutral', 'VADER Negative']
values = [ vader_pos, vader_neu, vader_neg]

plt.figure(figsize=(8, 5))
plt.bar(labels, values, color=[ 'cyan', 'orange', 'red'])
plt.title(f'VADER Sentiment Analysis Results for Reddit User: {username}')
plt.xlabel('Sentiment Metric')
plt.ylabel('Score')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



11.Son olarak Vader ile baştan yaptığımız Duygu Analizi ve sonuçlarını ayrıştırıp dataframe dönüştürerek daha basit bir görselleştirmeye evriltiyoruz

```
username = 'Unexpected'

user = reddit.redditor(username)
posts = user.submissions.new(limit=None)
comments = user.comments.new(limit=None)

# function for emotion analyze
def analyze_sentiment_vader(text):
    analyzer = SentimentIntensityAnalyzer()
    scores = analyzer.polarity_scores(text)
    return scores

# Analyze text from posts and comments and collect
results = []
for post in posts:
    sentiment_scores = analyze_sentiment_vader(post.title + ' ' + post.selftext)
    results.append({
        'Type': 'Post',
        'Content': post.title + ' ' + post.selftext,
        'Compound': sentiment_scores['compound'],
        'Positive': sentiment_scores['pos'],
        'Neutral': sentiment_scores['neu'],
        'Negative': sentiment_scores['neg']
    })

for comment in comments:
    sentiment_scores = analyze_sentiment_vader(comment.body)
    results.append({
        'Type': 'Comment',
        'Content': comment.body,
        'Compound': sentiment_scores['compound'],
        'Positive': sentiment_scores['pos'],
        'Neutral': sentiment_scores['neu'],
        'Negative': sentiment_scores['neg']
    })

# convert results to Dataframe
df = pd.DataFrame(results)

# Visualite
plt.figure(figsize=(14, 7))
sns.histplot(df, x='Compound', bins=30, kde=True, color='purple')
plt.title(f'{username} Kullanıcısının Duygu Dağılımı')
plt.xlabel('Duygu Skoru (Compound)')
plt.ylabel('Frekans')
plt.show()

# Emotion score(positive, negative, neutral)
df_mean = df[['Positive', 'Neutral', 'Negative']].mean().reset_index()
df_mean.columns = ['Duygu', 'Oran']
```

Sonuçlar:

