

Moduł 11

Przypadek użycia 1

Giełda i analiza tweetów



iBigWorld:
Innovations for Big Data in a Real World

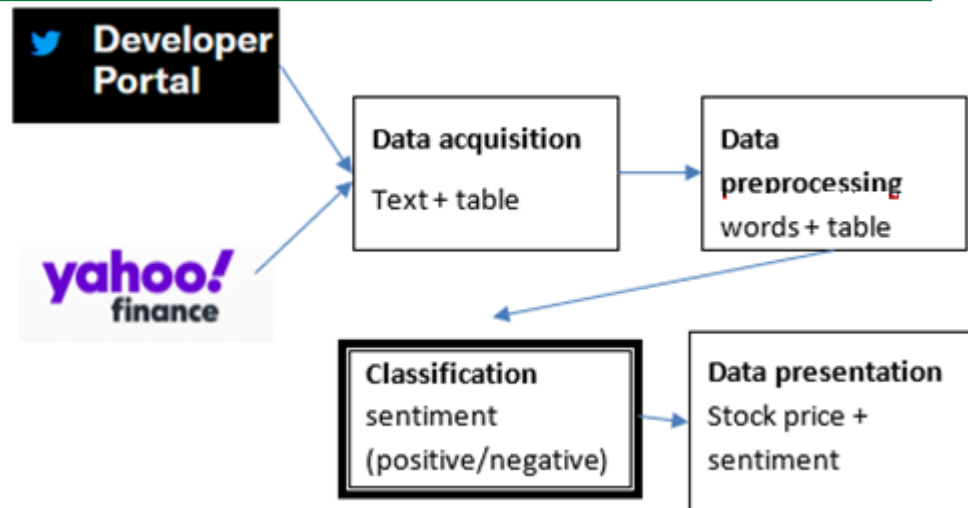
Zespół UBB

Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the National Agency (NA). Neither the European Union nor NA can be held responsible for them.



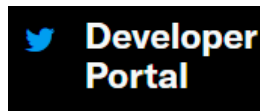
Analiza tweetów i giełdy

- Cena rynkowa akcji jest silnie powiązana z
 - podstawami (takimi jak zarobki lub wartość księgowa),
 - sentymentem akcjonariuszy do firmy - jej ogólny wizerunek

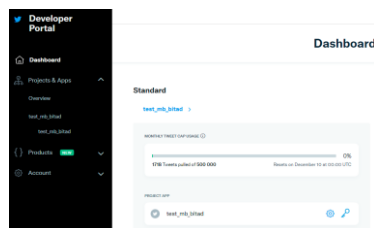


Pozyskiwanie danych

- Portal twittera



- Tworzenie konta developerskiego



- Tworzenie tokena dla aplikacji

Consumer Keys ①

API Key and Secret	Regenerate
--------------------	------------

Authentication Tokens ①

Bearer Token Generated November 12, 2021	Regenerate Revoke
---	----------------------

Access Token and Secret For @marcin_bernas	Generate
---	----------

Wykorzystanie API do pozyskania danych

- Instalacja biblioteki tweepy

```
✓ [1] !pip3 install tweepy
      !pip3 install tweepy --upgrade
```

- Wykonanie kwerendy

```
# Replace with your own search query
query = 'Musk Tesla'

tweets = client.search_recent_tweets(query=query, tweet_fields=['context_annotations', 'created_at'],

for tweet in tweets.data:
    print(tweet.text)
```

```
RT @Reuters: Tesla Chief Executive Elon Musk got into a spat with Bernie Sanders after the U.S. senator demanded the wealthy pay the
$TSLA - Elon Musk is up to $6.9B worth of Tesla stock sales and trolling Bernie Sanders about selling more https://t.co/Iow0jDuSW2
RT @Fattoquotidiano: Bernie Sanders chiede più tasse per i super ricchi ed Elon Musk di Tesla lo attacca: "Mi ero dimenticato fossi i
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
RT @RevChuckCurrie: @elonmusk @SenSanders I'm old enough to remember when Barack Obama and Joe Biden saved @elonmusk and @Tesla with
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
RT @RevChuckCurrie: @elonmusk @SenSanders I'm old enough to remember when Barack Obama and Joe Biden saved @elonmusk and @Tesla with
RT @RevChuckCurrie: @elonmusk @SenSanders I'm old enough to remember when Barack Obama and Joe Biden saved @elonmusk and @Tesla with
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
https://t.co/ZwdPbv7DE
RT @RBReich: So let me get this straight: Elon Musk accuses senators who want to increase taxes on billionaires of socialism, but he
RT @TheRealHoarse: Elon Musk is an asshole.
```



Pobranie danych rynkowych

- Import API do komunikacji z serwisem

```
!pip install yfinance
```

```
!pip install plotly
```

- Pobranie bibliotek:

- `# Raw Package`
- `import numpy as np`
- `import pandas as pd`
- `#Data Source`
- `import yfinance as yf`
- `#Data viz`
- `import plotly.graph_objs as go`

Wstępne przetwarzanie

- Dane mogą być wczytane do potoku:
- `#Wymagany interwał 5 min.`
- `data = yf.download(tickers='TSLA', period='5d', interval='5m')`
- `#Wyswietl dane`
- `data`

Datetime						
2022-01-18 09:30:00-05:00	1026.604980	1032.000000	1023.000000	1025.349976	1025.349976	1466741
2022-01-18 09:35:00-05:00	1025.000000	1025.999878	1018.380676	1020.150024	1020.150024	609419
2022-01-18 09:40:00-05:00	1021.000000	1035.849976	1016.059998	1034.969971	1034.969971	697473
2022-01-18 09:45:00-05:00	1032.500000	1037.625000	1031.000122	1037.625000	1037.625000	439527
2022-01-18 09:50:00-05:00	1040.000000	1045.699951	1036.640015	1043.672363	1043.672363	519753
...	...	...	...	...	...	...
2022-01-24 15:35:00-05:00	911.455811	914.000000	906.400024	910.809998	910.809998	412382
2022-01-24 15:40:00-05:00	910.049988	911.489990	900.109985	902.730591	902.730591	449806
2022-01-24 15:45:00-05:00	902.140015	917.419922	901.270081	917.200012	917.200012	607290
2022-01-24 15:50:00-05:00	917.429993	929.679993	914.609985	927.619995	927.619995	785036
2022-01-24 15:55:00-05:00	927.840027	933.513123	924.010071	930.130005	930.130005	928936

390 rows × 6 columns

Spark - inicjalizacja



- Instalacja sparka

```
[1]: !pip install pyspark
```

```
Requirement already satisfied: pyspark in c:\users\admin\anaconda3\lib\site-packages (3.1.2)
```

```
Requirement already satisfied: py4j==0.10.9 in c:\users\admin\anaconda3\lib\site-packages (from pyspark)
```

- Import modułów

```
[4]: from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAssembler
from pyspark.ml import Pipeline
from pyspark.sql.types import StructType, StructField, IntegerType, StringType
from pyspark.sql import functions as f
```



Spark - inicjalizacja



- Tworzenie sesji i import danych

```
[5]: spark = SparkSession.builder.appName("bitad example").getOrCreate()
```

```
[6]: df = spark.read.csv("tweet.csv",header=True)
```

```
[7]: df.show()
```

```
+-----+-----+
|          date|          tweet|
+-----+-----+
|2021-11-15 12:11:...|b'RT @ecomarxi: M...|
|2021-11-15 12:11:...|"b""RT @business:...|
|2021-11-15 12:11:...|b'Musk argues \xe...|
|2021-11-15 12:11:...|"b""RT @business:...|
|2021-11-15 12:11:...|b'RT @RBReich: So...|
|2021-11-15 12:11:...|b'@elonmusk @SenS...|
|2021-11-15 12:11:...|b'RT @SpiritofLen...|
|2021-11-15 12:11:...|b'RT @RevChuckCur...|
|2021-11-15 12:11:...|b'RT @_SemaHernan...|
|2021-11-15 12:11:...|"b'Trending on #L...|
|2021-11-15 12:11:...|b'@Rotshasa @_Anu...|
```

- Różne źródła: Hadoop, S3, another



Pre-processing – czyszczenie danych

- Usuwanie niechcianych znaków:

```
def preprocessing(lines):  
    words = lines.select(explode(split(lines.value,  
    "t_end"))).alias("word")  
    words = words.na.replace('', None)  
    words = words.na.drop()  
    words = words.withColumn('word', F.regexp_re  
place('word', r'http\S+', ''))  
    words = words.withColumn('word', F.regexp_re  
place('word', '.*\@w+', ''))  
    words = words.withColumn('word', F.regexp_re  
place('word', '#', ''))  
    words = words.withColumn('word', F.regexp_re  
place('word', 'RT', ''))  
    return words
```



Analiza sentymentu za pomocą funkcji UDF

- Creating sentiment function:

```
# text classification
def polarity_detection(text):
    return TextBlob(text).sentiment.polarity

def subjectivity_detection(text):
    return TextBlob(text).sentiment.subjectivity

def text_classification(words):
    # polarity detection
    polarity_detection_udf = udf(polarity_detection, StringType())
    words = words.withColumn("polarity", polarity_detection_udf(col("word")))

    # subjectivity detection
    subjectivity_detection_udf = udf(subjectivity_detection, StringType())
    words = words.withColumn("subjectivity", subjectivity_detection_udf("word"))

    return words
```

Analiza sentymentu

- Do połączenia wyników konstruujemy funkcję

```
def calcSent(row) :  
    if (row>0) :  
        return 1  
    elif (row<0) :  
        return -1  
    else: return 0
```

- Ostatecznie inna funkcja UDF łączy sentyment w finałowy wynik:

```
udfSentiment = udf(calcSent,StringType())  
tweets = words.withColumn("Sentiment",udfSentiment(f.  
col("polarity")+f.col("subjectivity")))
```

Prezentacja danych

- Wykres giełdowy prezentowany wraz z sentymentem.



summary	word	polarity	subjectivity	Sentiment
count	201	201	201	201
mean	null	0.06758114190950017	0.2736339227384003	0.373134328358209
stddev	null	0.1767709106810668	0.38519889662969664	0.48484495136659067
min		-0.25	0.0	0
max	tweet	0.85	1.0	1

Wyniki w praktyce

At&T

```
mean| -0.08003519777713337| 0.3732086551441|
stddev| 0.3205156324523829| 0.381136956641|
min| -0.05|
max| 1.0| 0.9166666666666666|
```

```
client = tweepy.Client(bearer_token="AAAAAAAAAAAA")
# Replace with your own search query
query = 'xxxx stocks'
tweets = client.search_recent_tweets(query=query,
```

Tesla

```
mean| 0.04365062946039919| 0.3065865832809452|
stddev| 0.23524532525350564| 0.3482713236024659|
min| -0.01759259259259...| 0.0|
max| 1.0| 1.0|
```

Stopy zwrotu: 1 dzień: **+4.08%**, YTD: **+49.47%**



Microsoft

```
mean| -0.0119470851728916| 0.5605974312425916|
stddev| 0.1804408159539326| 0.3986481860619351|
min| -0.05000000000000002| 0.0|
max| 1.0| 1.0|
```

Stopy zwrotu: 1 dzień: **-0.56%**, YTD: **-7.87%**



Stopy zwrotu: 1 dzień: **+1.02%**, YTD: **+53.64%**



Gamestop

```
mean| 0.2660398064063477| 0.33207730726527723|
stddev| 0.27181145155479447| 0.2524707073281747|
min| -0.03333333333333...| 0.0|
max| 0.5| 1.0|
```

Stopy zwrotu: 1 dzień: **-0.94%**, YTD: **+999.68%**

