BIG DATA ANALYTICS USING (PY)SPARK FOR ANALYZING IPO TWEETS

Dirk Van den Poel (Special Thanks to: Giselle van Dongen & Sacha Dubrulle Jolien Jackers, Sami Schyvinck & Diederick Van Damme)

UGENT DATA ANALYTICS TEAM

• Data Analytics
  www.dataanalytics.UGent.be
• Big Data
  www.bigdata.UGent.be
RESEARCH QUESTIONS

1. Are we able to predict, based on tweet characteristics, whether a message will be retweeted or not?

2. Is there a relationship between the buzz on social media during an IPO (Initial Public Offering) and the evolution of the (post-)IPO price?

HARDWARE

• Commodity hardware: Tweet streaming & fetching retweets
• HP Proliant C7000 Blade cluster (16x BL460c G6, 32x Quad core CPUs, 512 GB DDR3): ETL big data operations, mainly filtering and joining original tweets with retweets
• HP Z600 workstation to handle tweet analysis
• Databricks: Student projects
SOFTWARE STACK

- Python
- Jupyter
- Java
- Spark
- Parquet
- Databricks

STARTING POINT: DATA ACQUISITION

- Single-threaded java application on commodity hardware
- Saved as JSON files per 300 tweets, about 2 - 3 JSON files per minute ➔ about 600,000 in total
- List of keywords related to IPO’s
  - Company names: Humana, Editas
  - Tickers: HUM, EDIT
  - General: #ipo, ipo
ETL

- Streaming application
- Python script
- .json.zip files
- Batches .json.gz
- Spark script in batches per month
- Read in

- Zip files not supported by Spark
- Too many and too small files

PARQUET

- Distributed columnar data storage format
- Excellent integration with Spark
- High compression ratio
- Fast read
TIME DISTRIBUTION

FILTERING

– 263 M tweets (03/30/2016 – 03/30/2017)
– Problems:
  – Keywords are common words
  – Keywords are part of common words
I was wrong so many times in life but I do not think I deserve the neighbor who’s an improvised drummer at 21.15
FILTERING

- This leaves us with 263 M tweets (03/30/2016 – 03/30/2017)
  Problems:
  -Tickers/company names are (part of) common words
  -ticker ➔ cashtags: $SNAP, $EDIT
-Regex filtering in Spark
  ➔ 1.7 M tweets remaining
  -Contains ipo
  -Contains cashtag
  -Contains name

```python
# returns 0 if column is empty (no regexp match found), 1 otherwise (at least one regexp match found)
def non_empty_column(ipo_cashtag_column):
    if len(ipo_cashtag_column) > 0:
        return 1
    else:
        return 0
non_empty_column_udf = udf(non_empty_column)

# REGEX to extract tickers or presence of IPO
cashtag_expression = '"\$[a-z][1-6]"'
ipo_expression = '" #ipo | ipo "'
augmented_df = (df.select("tweet_id_str", 'text', lower(df.text).alias("text_lower"))
               .select("="
                        non_empty_column_udf(regexp_extract("text_lower", ipo_expression, 0)).alias("contains_ipo"),
                        non_empty_column_udf(regexp_extract("text_lower", cashtag_expression, 0)).alias("contains_cashtag"),
                        udf_contains_company_name("text_lower").alias("contains_company_name")
               )
)
filtered_df = augmented_df.filter((augmented_df.contains_ipo == 1) |
                                   (augmented_df.contains_cashtag == 1) |
                                   (augmented_df.contains_company_name == 1))
```
FILTERING, IN RETROSPECT

- Loose constraints
  e.g. EDIT, ABS, HUM, SNAP, IPO
  + Catch all buzz, filtering afterwards still possible
  - High storage volume

- Tight constraints
  e.g. $EDIT, $ABS, $HUM, $SNAP, #IPO
  + Limited storage volume
  - Dataset limited to financial tweets

FILTERING, RESULT

```r
ML_BTWsample(withReplacement=FALSE, fraction = 0.1).select("text").show(10, truncate = FALSE)
```

<table>
<thead>
<tr>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>A new article on vaccination by Mapi and Sanofi Pasteur: <a href="https://t.co/kN3puxRjje">Link</a> <a href="https://t.co/goePzgKdt">Link</a></td>
</tr>
<tr>
<td>#Bestseller Vizio E32-C1 32-Inch 1080p LED Full-Array Smart TV for $400</td>
</tr>
<tr>
<td>#Business</td>
</tr>
<tr>
<td>#Cisco</td>
</tr>
<tr>
<td>#Deals</td>
</tr>
</tbody>
</table>

only showing top 10 rows
RETWEETS

- At the time of streaming, the number of retweets is not yet available
  => Fetch the tweet again after waiting period for tweets to accumulate retweets
- 8 CentOS machines with multiple Authentication Keys each to overcome rate limits using simple Python scripts
- BUT:
  - Account deleted/blocked?
  - Account not publicly available anymore
  - Tweet deleted

RETWEETS, IN RETROSPECT

- Depending on the goal of the analysis, keeping all tweet information at initial streaming might have been redundant
MODELING RETWEETS

RESEARCH QUESTION

1. Are we able to predict, based on tweet characteristics, whether a message will be retweeted or not?
FEATURES OVERVIEW

TEXT FEATURES
- Contains cashtag
- Cashtag count
- Contains hashtag
- Hashtag count
- Language
- Contains URL
- Contains cashtag with spaces

NON-TEXT FEATURES
- User seniority
- User favorite count
- User friends count
- Users follower count
- Day of week

FEATURE PREPROCESSING

Numeric input variables
- StringIndexer
- OneHotEncoder
- VectorAssembler
- Input vector

Categorical input variables
- StringIndexer
- OneHotEncoder
- VectorAssembler
- Input vector

Dependent variable
- Binarizer
- Label
DATA MINING METHODOLOGY

- Logistic regression
- Random forest

MODEL DESCRIPTIONS

```python
def RF_classification(model_description, df, cat_columns, num_columns, output_column = 'retweet_binary':
    stringIndexers = []; indexed_cols = []; encoders = []; encoded_cols = []

    for column in cat_columns:
        outCol = column + '_indexed'
        stringIndexer = StringIndexer(inputCol=column, outputCol=outCol)
        stringIndexers.append(stringIndexer)
        indexed_cols.append(outCol)

    for column in indexed_cols:
        outCol = column + '_encoded'
        encoder = OneHotEncoder(inputCol=column, outputCol=outCol)
        encoders.append(encoder)
        encoded_cols.append(outCol)

    binarizer = Binarizer(inputCol = 'retweet count double', outputCol = output_column, threshold = 0.5)
    assembler = VectorAssembler(inputCols = num_columns + encoded_cols, outputCol='features')

    pipeline = Pipeline(stages= stringIndexers + encoders + [assembler] + [binarizer])
    pipelineModel = pipeline.fit(df)

    transformed_df = (pipelineModel.transform(df)
                           .select('features', col('retweet_binary'), cast('integer'), 'retweet_count'))

    train, validation, test = transformed_df.randomSplit([0.6, 0.1, 0.3], seed = 42)

    rf = RandomForestClassifier(numTrees = 500, labelCol = 'retweet_binary',
                                featureCols = list(transformed_df.columns[-len(stringIndexers) - 3:-1]),
                                maxBins = 32)
    rfModel = rf.fit(train)

    train_predictions = rfModel.transform(train)
    validation_predictions = rfModel.transform(validation)

    AUC_evaluator = BinaryClassificationEvaluator(randomSeed = 42, labelCol = 'retweet_binary', metricName = 'areaUnderROC')
    AUC_train = AUC_evaluator.evaluate(train_predictions)
    AUC_validation = AUC_evaluator.evaluate(validation_predictions)

    print('Train AUC: {}, Validation AUC: {}'.format(AUC_train, AUC_validation))
```

StringIndexer and OneHotEncoder

Binarizer & VectorAssembler

Pipeline

Data split, train, predict and evaluate
MODEL DESCRIPTIONS

MODEL PERFORMANCE

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Features</th>
<th>Train AUC</th>
<th>Test AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>Text</td>
<td>0.6340</td>
<td>0.6290</td>
</tr>
<tr>
<td>Random Forest</td>
<td>Text</td>
<td>0.6469</td>
<td>0.6416</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>Non-text</td>
<td>0.6688</td>
<td>0.6661</td>
</tr>
<tr>
<td>Random Forest</td>
<td>Non-text</td>
<td>0.7948</td>
<td>0.7923</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>Text + Non-text</td>
<td>0.7090</td>
<td>0.7049</td>
</tr>
<tr>
<td>Random Forest</td>
<td>Text + Non-text</td>
<td>0.7994</td>
<td>0.7970</td>
</tr>
</tbody>
</table>
LINK: TWITTER & STOCK RETURNS

RESEARCH QUESTION

2. Is there a relationship between the buzz on social media during an IPO (Initial Public Offering) and the evolution of the (post-)IPO price?
LITERATURE REVIEW: TWITTER -> RETURNS

Behavioral economics: Emotions can profoundly affect individual behavior and decision making. But also collective behavior (such as the stock market)?

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Model(s)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bollen, J., Mao, H., &amp; Zeng, X. (2011)***</td>
<td>Log. R NN SVM SVR** SOFNN NB</td>
<td>X</td>
</tr>
<tr>
<td>Au, Benjamin; Zhang, Qian; Zhang, Wanlu (2013) X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paglia, S. a M. (2013)</td>
<td></td>
<td>X*</td>
</tr>
<tr>
<td>Sumbureru, P.T. (2015)</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

LITERATURE REVIEW: IPO TWEETS -> RETURNS

Julie Zhang (Nov. 2015, Univ. of Massachusetts): ‘Social Media Whispers During The IPO Quiet Period’, INFORMS-conference in Philadelphia.
STOCKS

ARA: AMERICAN RENAL ASSOCIATES HOLDINGS

IPO date: April 21, 2016

<table>
<thead>
<tr>
<th>IPO</th>
<th>IPO - CP</th>
<th>IPO + 3M</th>
<th>IPO + 6M</th>
</tr>
</thead>
<tbody>
<tr>
<td>$22,00</td>
<td>$26,50</td>
<td>$26,59</td>
<td>$18,15</td>
</tr>
</tbody>
</table>

Sentiment score: 84  
Prediction:  

Sentiment score: 137  
Prediction:  

Sentiment score: 63  
Prediction:  

GHENT UNIVERSITY
SNAP

ANSWERS TO RESEARCH QUESTION
## RESEARCH QUESTION: IPO VS FIRST-DAY CP

<table>
<thead>
<tr>
<th>Stock</th>
<th>IPO</th>
<th>IPO - CP</th>
<th>UP_1/DOWN_0 Sentiment score</th>
<th>POS_1/DOWN_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARA</td>
<td>$22.00</td>
<td>$26.50</td>
<td>1</td>
<td>84</td>
</tr>
<tr>
<td>BOLD</td>
<td>$15.00</td>
<td>$15.13</td>
<td>1</td>
<td>45</td>
</tr>
<tr>
<td>MGP</td>
<td>$21.00</td>
<td>$22.01</td>
<td>1</td>
<td>71</td>
</tr>
<tr>
<td>NTLA</td>
<td>$18.00</td>
<td>$22.10</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>PLSE</td>
<td>$4.00</td>
<td>$4.17</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>RETA</td>
<td>$11.00</td>
<td>$13.07</td>
<td>1</td>
<td>-6</td>
</tr>
<tr>
<td>RRR</td>
<td>$19.50</td>
<td>$18.70</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>SBPH</td>
<td>$12.00</td>
<td>$11.10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SCWX</td>
<td>$14.00</td>
<td>$14.00</td>
<td>0</td>
<td>56</td>
</tr>
<tr>
<td>AZRE</td>
<td>$18.00</td>
<td>$14.60</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>SUPV</td>
<td>$11.00</td>
<td>$11.50</td>
<td>1</td>
<td>51</td>
</tr>
<tr>
<td>TUSK</td>
<td>$15.00</td>
<td>$13.26</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>TRHC</td>
<td>$12.00</td>
<td>$14.88</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TCMD</td>
<td>$15.00</td>
<td>$11.08</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>SNAP</td>
<td>$17.00</td>
<td>$24.48</td>
<td>1</td>
<td>6087</td>
</tr>
</tbody>
</table>

### Up / Down
- Pos: 7 / 5 (71% / 33%)
- Neg: 2 / 1 (13% / 7%)

## RESEARCH QUESTION: FIRST-DAY CP VS 3M LATER

<table>
<thead>
<tr>
<th>Stock</th>
<th>IPO - CP</th>
<th>IPO - CP</th>
<th>PRICE 3M AFTER IPO</th>
<th>Sentiment score</th>
<th>POS_1/DOWN_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARA</td>
<td>$26.50</td>
<td>$26.59</td>
<td>1</td>
<td>137</td>
<td>1</td>
</tr>
<tr>
<td>BOLD</td>
<td>$15.13</td>
<td>$17.42</td>
<td>1</td>
<td>103</td>
<td>0</td>
</tr>
<tr>
<td>MGP</td>
<td>$22.01</td>
<td>$26.70</td>
<td>1</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>NTLA</td>
<td>$22.10</td>
<td>$20.19</td>
<td>0</td>
<td>111</td>
<td>1</td>
</tr>
<tr>
<td>PLSE</td>
<td>$4.17</td>
<td>$4.62</td>
<td>1</td>
<td>-9</td>
<td>0</td>
</tr>
<tr>
<td>RETA</td>
<td>$13.07</td>
<td>$19.21</td>
<td>1</td>
<td>294</td>
<td>1</td>
</tr>
<tr>
<td>RRR</td>
<td>$18.70</td>
<td>$23.14</td>
<td>1</td>
<td>167</td>
<td>1</td>
</tr>
<tr>
<td>SBPH</td>
<td>$11.10</td>
<td>$10.77</td>
<td>0</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>SCWX</td>
<td>$14.00</td>
<td>$14.51</td>
<td>1</td>
<td>238</td>
<td>1</td>
</tr>
<tr>
<td>AZRE</td>
<td>$14.60</td>
<td>$16.10</td>
<td>1</td>
<td>254</td>
<td>1</td>
</tr>
<tr>
<td>SUPV</td>
<td>$11.50</td>
<td>$13.50</td>
<td>1</td>
<td>89</td>
<td>1</td>
</tr>
<tr>
<td>TUSK</td>
<td>$13.26</td>
<td>$17.46</td>
<td>1</td>
<td>269</td>
<td>1</td>
</tr>
<tr>
<td>TRHC</td>
<td>$14.88</td>
<td>$14.63</td>
<td>0</td>
<td>452</td>
<td>1</td>
</tr>
<tr>
<td>TCMD</td>
<td>$11.08</td>
<td>$17.52</td>
<td>1</td>
<td>258</td>
<td>1</td>
</tr>
</tbody>
</table>

### Up / Down
- Pos: 10 / 3 (71% / 21%)
- Neg: 1 / 0 (7% / 0%)
RESEARCH QUESTION: 3M VS 6M LATER

<table>
<thead>
<tr>
<th></th>
<th>Price 6M after IPO</th>
<th>IPO - 3M later</th>
<th>IPO - 6M later</th>
<th>UP 1/DOWN 0</th>
<th>Sentiment score</th>
<th>POS 1/DOWN 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARA</td>
<td>26.59</td>
<td>18.15</td>
<td>0</td>
<td>63</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MGP</td>
<td>26.70</td>
<td>25.90</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NTLA</td>
<td>20.19</td>
<td>14.67</td>
<td>0</td>
<td>63</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PLSE</td>
<td>4.62</td>
<td>5.97</td>
<td>1</td>
<td>66</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>RETA</td>
<td>19.21</td>
<td>27.83</td>
<td>1</td>
<td>123</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>RRR</td>
<td>23.14</td>
<td>22.12</td>
<td>0</td>
<td>365</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SBPH</td>
<td>10.77</td>
<td>7.85</td>
<td>0</td>
<td>206</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SCWX</td>
<td>14.51</td>
<td>11.29</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Up</th>
<th>Down</th>
<th>%</th>
<th>Up</th>
<th>Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos</td>
<td>2</td>
<td>6</td>
<td></td>
<td>Pos</td>
<td>25%</td>
</tr>
<tr>
<td>Neg</td>
<td>0</td>
<td>0</td>
<td></td>
<td>Neg</td>
<td>0%</td>
</tr>
</tbody>
</table>

FUTURE RESEARCH & RECOMMENDATIONS

---
FUTURE RESEARCH

– Take into account who is tweeting (cf. pump and dump)
– Deep learning approach to focus on text of the tweet
– Recommendations for tweet success
– Can we extend stock-trading models using twitter information, either in the IPO of in a more general stock-trading perspective?

IMPROVEMENTS OVER LAST YEAR

– Random Forest received major upgrade
– Findspark Python package

```python
import findspark
findspark.init(spark_home = HOME_DIR + "/spark-2.1.0")
import pyspark

from pyspark.sql import SQLContext, SparkSession

try:
    sc = pyspark.SparkContext(appName = 'App')
    spark = SparkSession(sparkContext=sc)
    print "SparkSession initialized"
except ValueError:
    print "SparkSession already initialized"
```
RECOMMENDATIONS

• Extending number of state-of-the-art data mining methodologies in Spark: e.g. XGBoost, Hybrid Ensembles
• Facilitating the interpretation of output (e.g. variables importances)
BIG DATA COURSE @ GHENT UNIVERSITY

- Target audience:
  - Business Engineering students
  - Exchange students
  - Students taking course as elective
- Heterogeneous backgrounds
- SQL + Python + Spark

DATABRICKS STUDENT GROUP ASSIGNMENT

- Previously, job submits on HPC cluster
- Stable
- Easy notebook import
- Availability without need of powerful machine
- Notebook environment with nice GUI + nice graphs
Dirk Van den Poel
Senior Full Professor of Data Analytics / Big Data

DEPARTMENT OF MARKETING

E Dirk.VandenPoel@UGent.Be

www.crm.ugent.be
www.bigdata.ugent.be
www.mma.ugent.be
www.dataanalytics.ugent.be