Predpovedaj budúcnosť trhu

Ukážka využitia Elasticsearch na analýzu dátovej sady z New Yorskej burzy s využitím vstavaných nástrojov strojového učenia, ktoré elastic podporuje.

technology

Spark
kafka
Java
elastic
spring
aws
kubernetes
docker
Liferay

ableneo in numbers

135% Our growth in 2017.

40 Innovation enablers in team.

06 Years of experience.

03 Locations.

ableneo

young talent

ableneo's young talent program is an opportunity for students or graduates who have a clear vision for their future career and are looking for experience.

During the 6 months program we focus on:

- **Training / Mentoring focused on gaining engineering skills**
- **Project / Customer experience**
- **Innovation Enabling, helping you bring ideas to reality**

about ableneo

5 years of experience

With adoption of complex organizational change. ableneo was spinned-off from mimacom group as separate brand in 2018.

3 countries

Where ableneo enables innovation in customer projects, another 2 in pipeline.

40 innovation enablers in our team

Consisting of software architects, engineers, scrum masters and coaches.

135% business growth in 2017

ableneo strategically focuses on building key competences committed to maximum efficiency and delivering business value.

technology stack

frontend
angular
react
typescript
webpack

backend
spring boot
java

data
elastic stack
kafka
hadoop
spark

platform
liferay dxp

infrastructure
aws
cloud foundry
docker
kubernetes
spring cloud

If you're interested contact us at youngtalent@ableneo.com, or visit www.ableneo.com

ableneo

ableneo

0 Requirements

- Docker installed
- git.io/fjesW

1 Theory

Elastic stack

- **Elasticsearch**
- Search & analytics engine
- Easily scalable and very fast

- **Kibana**
- user interface for elasticsearch

- **Logstash & Beats**
- data shipping & transformation

Machine learning in elastic

Elasticsearch data insights

01

Search for transactions for a user in real time

02

Use aggregations and visualisations:

- top ten selling products
- trends in transactions over time

03

Use machine learning and go deeper:

- Changes in behavior
- Unusual processes running on host

Anomaly detection

Elastic scoring system

Record scoring

The scoring for an individual anomaly a "record"

Record scoring

- Scoring at the lowest level of the hierarchy
- Absolute "unusualness" of a specific instance of something occurring.
- Example:
    - Rate of failed logins for user=admin was observed to be 300 fails in the last minute
    - Value of the response time for a specific middleware call just jumped to be 300% larger than usual
    - Number of orders being processed this afternoon is much lower than what it is for a typical Thursday afternoon
    - The amount of data being transferred to a remote IP address is much more than the amount being transferred to other remote IPs
- These occurrences has a calculated probability (as small as 1e-308)
- Based on past behavior -> Baseline probability model

Record scoring

- However, this probability value, while certainly useful, lack some contextual information
    - How does the current anomalous behavior compare to past anomalies?
    - Is it more or less unusual than past anomalies?
    - How does this item’s anomalousness compare to other potentially anomalous items (users, IPs, etc.)?
- ML normalizes the probability -> ranks an item’s anomalousness on a scale 0-100 (the anomaly_score)
- Sereverity labels according to score

Example – ML's API

```
GET /_xpack/ml/anomaly_detectors/farequote_count/results/records?human
{
  "sort": "record_score",
  "desc": true,
  "start": "2016-02-09T16:15:00.000Z",
  "end" :"2016-02-09T16:20:00.000Z"
}
```

- 5-minute interval (the bucket_span of the job)
- the record_score is 90.6954 (out of 100)
- the raw probability is 1.75744e-11.

Conclusion
Very unlikely that the volume of data in this particular 5 minute interval should have an actual rate of 179 documents because "typically" it is much lower, closer to 60.


```
{
  "count": 1,
  "records": [
    {
      "job_id": "farequote_count",
      "result_type": "record",
      "probability": 1.75744e-11,
      "record_score": 90.6954,
      "initial_record_score": 85.0643,
      "bucket_span": 300,
      "detector_index": 0,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000,
      "function": "count",
      "function_description": "count",
      "typical": [
        59.9827
      ],
      "actual": [
        179
      ]
    }
  ]
}
```

Example – ML's API

```
GET /_xpack/ml/anomaly_detectors/farequote_count/results/records?human
{
  "sort": "record_score",
  "desc": true,
  "start": "2016-02-09T16:15:00.000Z",
  "end" :"2016-02-09T16:20:00.000Z"
}
```

The projection of probability 1.75744e-11 into record_score is 90.6954 (out of 100)

- Roughly based on a quantile analysis
- Probability values historicaly seen for anomalies (in this job) are ranked against each other.
- Lowest probabilities historically for the job get the highest score


```
{
  "count": 1,
  "records": [
    {
      "job_id": "farequote_count",
      "result_type": "record",
      "probability": 1.75744e-11,
      "record_score": 90.6954,
      "initial_record_score": 85.0643,
      "bucket_span": 300,
      "detector_index": 0,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000,
      "function": "count",
      "function_description": "count",
      "typical": [
        59.9827
      ],
      "actual": [
        179
      ]
    }
  ]
}
```

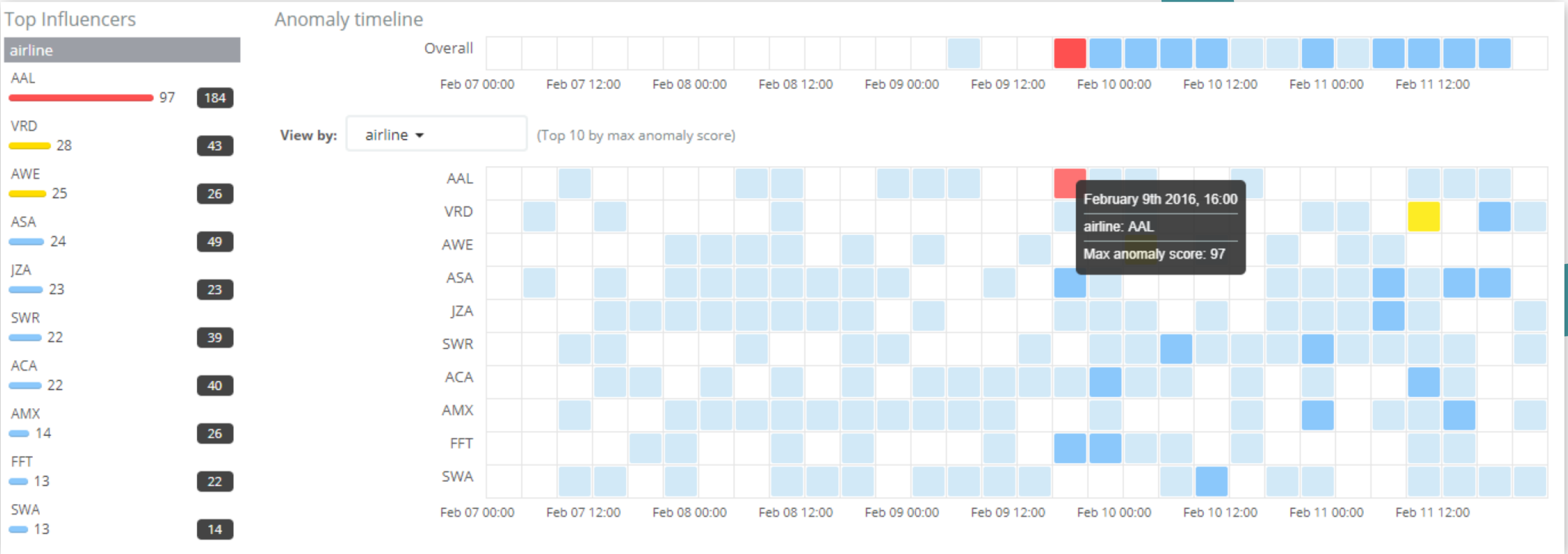
02

Influencer scoring

The scoring for an entity such as a user or IP address an "influencer"

Record scoring

- Rank entities that may have contributed to an anomaly
  - In ML these contributors are called influencers
  - In previous examples it was just a single time series
- Example
  - Analysis of a population of users' internet activity
    - ML job looks at unusual bytes sent and unusual domains visited
    - You could specify "user" as a possible influencer since that is the entity that is "causing" the anomaly to exist
    - An influencer score will be given to each user, dependent on how anomalous each was considered in one or both of these areas (bytes sent and domains visited) during each time interval.
    - The higher the influencer score, the more that entity will have contributed to or is to blame for, the anomalies.


Top Influencers
airline
AAL 97 184
VRD 28 43
AWE 25 26
ASA 24 49
JZA 23 23
SWR 22 39
ACA 22 40
AMX 14 26
FFT 13 22
SWA 13 14
Anomaly timeline
Overall
Feb 07 00:00
Feb 07 12:00
Feb 08 00:00
Feb 08 12:00
Feb 09 00:00
Feb 09 12:00
Feb 10 00:00
Feb 10 12:00
Feb 11 00:00
Feb 11 12:00
View by: airline
(Top 10 by max anomaly score)
AAL
VRD
AWE
ASA
JZA
SWR
ACA
AMX
FFT
SWA
February 9th 2016, 16:00
airline: AAL
Max anomaly score: 97

Example – ML's API

```
GET _xpack/ml/anomaly_detectors/farequote_count_and_responsetime_by_airline/results/i
nfluencers?human
{
  "start": "2016-02-09T16:15:00.000Z",
  "end" :"2016-02-09T16:20:00.000Z"
}
```

Result for the influencing airline AAL

- the influencer_score of 97.1547
- displayed in the Anomaly Explorer UI (rounded to 97).
- The probability value of 6.56622e-40 is again the basis of the influencer_score (before it gets normalized) - it takes into account the the probabilities of the individual anomalies that particular airline influences, and the degree to which it influences them.


```
{
  "count": 2,
  "influencers": [
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "result_type": "influencer",
      "influencer_field_name": "airline",
      "influencer_field_value": "AAL",
      "airline": "AAL",
      "influencer_score": 97.1547,
      "initial_influencer_score": 98.5096,
      "probability": 6.56622e-40,
      "bucket_span": 300,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000
    },
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "result_type": "influencer",
      "influencer_field_name": "airline",
      "influencer_field_value": "AWE",
      "airline": "AWE",
      "influencer_score": 0,
      "initial_influencer_score": 0,
      "probability": 0.0499957,
      "bucket_span": 300,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000
    }
  ]
}
```

Example – ML's API

```
GET _xpack/ml/anomaly_detectors/farequote_count_and_responsetime_by_airline/results/i
nfluencers?human
{
  "start": "2016-02-09T16:15:00.000Z",
  "end" :"2016-02-09T16:20:00.000Z"
}
```

initial_influencer_score of 98.5096,

- score when the result was processed, before subsequent normalizations adjusted it slightly to 97.1547.
- This occurs because the ML job processes data in chronological order and never goes back to re-read older raw data to analyze/review it again.
- Also note that a second influencer, airline AWE, was also identified, but its influencer score is so low (rounded to 0) that it should be ignored in a practical sense.


```
{
  "count": 2,
  "influencers": [
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "result_type": "influencer",
      "influencer_field_name": "airline",
      "influencer_field_value": "AAL",
      "airline": "AAL",
      "influencer_score": 97.1547,
      "initial_influencer_score": 98.5096,
      "probability": 6.56622e-40,
      "bucket_span": 300,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000
    },
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "result_type": "influencer",
      "influencer_field_name": "airline",
      "influencer_field_value": "AWE",
      "airline": "AWE",
      "influencer_score": 0,
      "initial_influencer_score": 0,
      "probability": 0.0499957,
      "bucket_span": 300,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000
    }
  ]
}
```

Example – ML's API

```
GET _xpack/ml/anomaly_detectors/farequote_count_and_responsetime_by_airline/results/i
nfluencers?human
{
  "start": "2016-02-09T16:15:00.000Z",
  "end" :"2016-02-09T16:20:00.000Z"
}
```

Because the influencer_score is an aggregated view across multiple detectors, you will notice that the API does not return the actual or typical values for the count or the mean of response times.

If you need to access this detailed information, then it is still available for the same time period as a record result, as shown before.


```
{
  "count": 2,
  "influencers": [
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "result_type": "influencer",
      "influencer_field_name": "airline",
      "influencer_field_value": "AAL",
      "airline": "AAL",
      "influencer_score": 97.1547,
      "initial_influencer_score": 98.5096,
      "probability": 6.56622e-40,
      "bucket_span": 300,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000
    },
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "result_type": "influencer",
      "influencer_field_name": "airline",
      "influencer_field_value": "AWE",
      "airline": "AWE",
      "influencer_score": 0,
      "initial_influencer_score": 0,
      "probability": 0.0499957,
      "bucket_span": 300,
      "is_interim": false,
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000
    }
  ]
}
```

03

Bucket scoring

The scoring for a window of time a "bucket"

Record scoring

- At the top of the hierarchy is to focus on time
  - The bucket_span of the ML job
  - Unusual things happen at specific times and it is possible that one or more items can be unusual together at the same time (within the same bucket).
- Anomalousness of a time bucket is dependent on:
  - The magnitude of the individual anomalies (records) occurring within that bucket
  - The number of individual anomalies (records) occurring within that bucket. This could be many if the job has "splitting" using by_fields and/or partition_fields - or if there exist multiple detectors in the job.
- The calculation behind the bucket score is more complex than just a simple average of all the individual anomaly record scores, but will have a contribution from the influencer scores in each bucket.
- "overall" lane in the "Anomaly timeline"

Example – ML's API

```
GET _xpack/ml/anomaly_detectors/farequote_count_and_responsetime_by_airline/results/b
uckets?human
{
  "start": "2016-02-09T16:15:00.000Z",
  "end" :"2016-02-09T16:20:00.000Z"
}
```

- anomaly_score - the overall aggregated, normalized score
- initial_anomaly_score - the anomaly_score at the time the bucket was processed (again, in case later normalizations have changed the anomaly_score from its original value).
- bucket_influencers - an array of influencer types present in this bucket. As suspected, given our discussion of influencers above, this array contains entries for both
- influencer_field_name:airline and influencer_field_name:bucket_time (which is always added as a built-in influencer). The details of what specific influencers values (i.e. which airline) is available when one queries the API specifically for the influencer or record values, as shown earlier.


```
{
  "count": 1,
  "buckets": [
    {
      "job_id": "farequote_count_and_responsetime_by_airline",
      "timestamp_string": "2016-02-09T16:15:00.000Z",
      "timestamp": 1455034500000,
      "anomaly_score": 90.7,
      "bucket_span": 300,
      "initial_anomaly_score": 85.08,
      "event_count": 179,
      "is_interim": false,
      "bucket_influencers": [
        {
          "job_id": "farequote_count_and_responsetime_by_airline",
          "result_type": "bucket_influencer",
          "influencer_field_name": "airline",
          "initial_anomaly_score": 85.08,
          "anomaly_score": 90.7,
          "raw_anomaly_score": 37.3875,
          "probability": 6.92338e-39,
          "timestamp_string": "2016-02-09T16:15:00.000Z",
          "timestamp": 1455034500000,
          "bucket_span": 300,
          "is_interim": false
        },
        {
          "job_id": "farequote_count_and_responsetime_by_airline",
          "result_type": "bucket_influencer",
          "influencer_field_name": "bucket_time",
          "initial_anomaly_score": 85.08,
          "anomaly_score": 90.7,
          "raw_anomaly_score": 37.3875,
          "probability": 6.92338e-39,
          "timestamp_string": "2016-02-09T16:15:00.000Z",
          "timestamp": 1455034500000,
          "bucket_span": 300,
          "is_interim": false
        }
      ],
      "processing_time_ms": 17,
      "result_type": "bucket"
    }
  ]
}
```

Using Anomaly Scores for Alerting

- Which would be useful for alerting?
  - depends on what you are trying to accomplish
  - granularity,
  - rate, of alerts

bucket-based anomaly score – detect and alert upon significant deviations in the overall data set as a function of time

influencer_score – detect and alert on the most unusual entities over time

record_score - detect and alert upon the most unusual anomaly within a window of time

- **Recommend using the bucket-based** -> max 1 alert per bucket_span
- If using record_score - number of anomalous records per unit time is arbitrary (could be many!!)

Resources

https://discuss.elastic.co/t/what-is-the-machine-learning-algorithm-in-elastic/123527

https://www.elastic.co/blog/machine-learning-anomaly-scoring-elasticsearch-how-it-works

Economics

- **Stock market** is a place, where shares of companies are traded

- **Volume** is a measure of **how much** of a given financial asset **has been traded** in a given period of time

- **High/Low** – are stock price maximums during the day
- **Open/Close** – are prices at the start and at the end of the day

2 Import & analyze

3

Find the real-world event behind anomalies

4 Predict & get rich $$$

ableneo

The innovation enablers