



elasticsearch  
for developer

Boosting

boost.txt

# Boost

INDEX

QUERY

# Index boost

แยกความสำคัญของแต่ละ index

**GET** thai,english/\_search

```
{  
  "indices_boost" : {  
    "thai" : 1.5,  
    "english" : 1.1  
  }  
}
```

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/search-request-index-boost.html>

# Query

- Boosting query clause
- Boosting by popularity

# Boost query clause

Problem :: ต้องการค้นหา **book**

Elasticsearch **book**

Lucene **book**

**book**



# Search book

GET store/\_search

```
{
  "query": {
    "term": {
      "name": {
        "value": "book"
      }
    }
  }
}
```



```
"hits": [
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU5N-hLXkZRCf3MRSdC",
    "_score": 0.5945348,
    "_source": {
      "name": "book",
      "price": 100
    }
  },
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU5N-DqXkZRCf3MRSdB",
    "_score": 0.37158427,
    "_source": {
      "name": "elasticsearch book",
      "price": 200
    }
  },
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU5N9o4XkZRCf3MRSdA",
    "_score": 0.19178301,
    "_source": {
      "name": "lucene book",
      "price": 400
    }
  }
]
```

# Boost query clause

Problem :: ต้องการให้หนังสือ Lucene ขึ้นอันดับหนึ่ง

Lucene **book**

Elastic search **book**

**book**



[http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/\\_boosting\\_query\\_clauses.html](http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/_boosting_query_clauses.html)

# What we need ?

```
"hits": [  
  {  
    "_index": "store",  
    "_type": "book",  
    "_id": "AU5N9o4XkZRCf3MRSdA",  
    "_score": 0.22567046,  
    "_source": {  
      "name": "lucene book",  
      "price": 400  
    }  
  },  
  {  
    "_index": "store",  
    "_type": "book",  
    "_id": "AU5N-hLXkZRCf3MRSdC",  
    "_score": 0.020825317,  
    "_source": {  
      "name": "book",  
      "price": 100  
    }  
  },  
  {  
    "_index": "store",  
    "_type": "book",  
    "_id": "AU5N-DqXkZRCf3MRSdB",  
    "_score": 0.013015823,  
    "_source": {  
      "name": "elasticsearch book",  
      "price": 200  
    }  
  }  
]
```



# Boost with boolean query

```
GET store/_search
{
  "query": {
    "bool": {
      "must": [{
        "match": {
          "name": {
            "query": "book"
          }
        }
      }],
      "should": [
        {
          "match": {
            "name": {
              "query": "lucene",
              "boost": 5
            }
          }
        }
      ]
    }
  }
}
```



```
"hits": [
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU5N9o4XkZRCf3MRSdA",
    "_score": 0.22567046,
    "_source": {
      "name": "lucene book",
      "price": 400
    }
  },
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU5N-hLXkZRCf3MRSdC",
    "_score": 0.020825317,
    "_source": {
      "name": "book",
      "price": 100
    }
  },
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU5N-DqXkZRCf3MRSdB",
    "_score": 0.013015823,
    "_source": {
      "name": "elasticsearch book",
      "price": 200
    }
  }
]
```

# Boost query

**“boost” 1 - 15**



# Workshop

- ค้นหาคำว่า book
- ความสำคัญของคำเรียกจาก field
  1. name
  2. detail

รายละเอียดในไฟล์ boost\_workshop.txt

# Bad search result !!

GET store/\_search

```
{
  "query": {
    "term": {
      "_all": {
        "value": "book"
      }
    }
  }
}
```



```
"hits": [
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU55YKabXkZRCf3MRSdG",
    "_score": 0.4459011,
    "_source": {
      "name": "lucene",
      "detail": "lucene book book book book",
      "price": 400
    }
  },
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU55XUcvXkZRCf3MRSdE",
    "_score": 0.36407673,
    "_source": {
      "name": "elasticsearch in action",
      "detail": "elasticsearch book book book book book book",
      "price": 200
    }
  },
  {
    "_index": "store",
    "_type": "book",
    "_id": "AU55XUDlXkZRCf3MRSdD",
    "_score": 0.21441993,
    "_source": {
      "name": "lucene book",
      "detail": "lucene book book book book",
      "price": 400
    }
  }
]
```

# We need better result !!

```
"hits": [  
  {  
    "_index": "store",  
    "_type": "book",  
    "_id": "AU5XU62XkZRCf3MRSdF",  
    "_score": 0.37080312,  
    "_source": {  
      "name": "book",  
      "detail": "my personal book",  
      "price": 100  
    }  
  },  
  {  
    "_index": "store",  
    "_type": "book",  
    "_id": "AU5XUD1XkZRCf3MRSdD",  
    "_score": 0.31226164,  
    "_source": {  
      "name": "lucene book",  
      "detail": "lucene book book book book",  
      "price": 400  
    }  
  },  
  {  
    "_index": "store",  
    "_type": "book",  
    "_id": "AU5YKabXkZRCf3MRSdG",  
    "_score": 0.06874621,  
    "_source": {  
      "name": "lucene",  
      "detail": "lucene book book book book",  
      "price": 400  
    }  
  }  
]
```

# Boost by popularity

Problem :: ต้องการใช้ข้อมูลจากจำนวนของ like

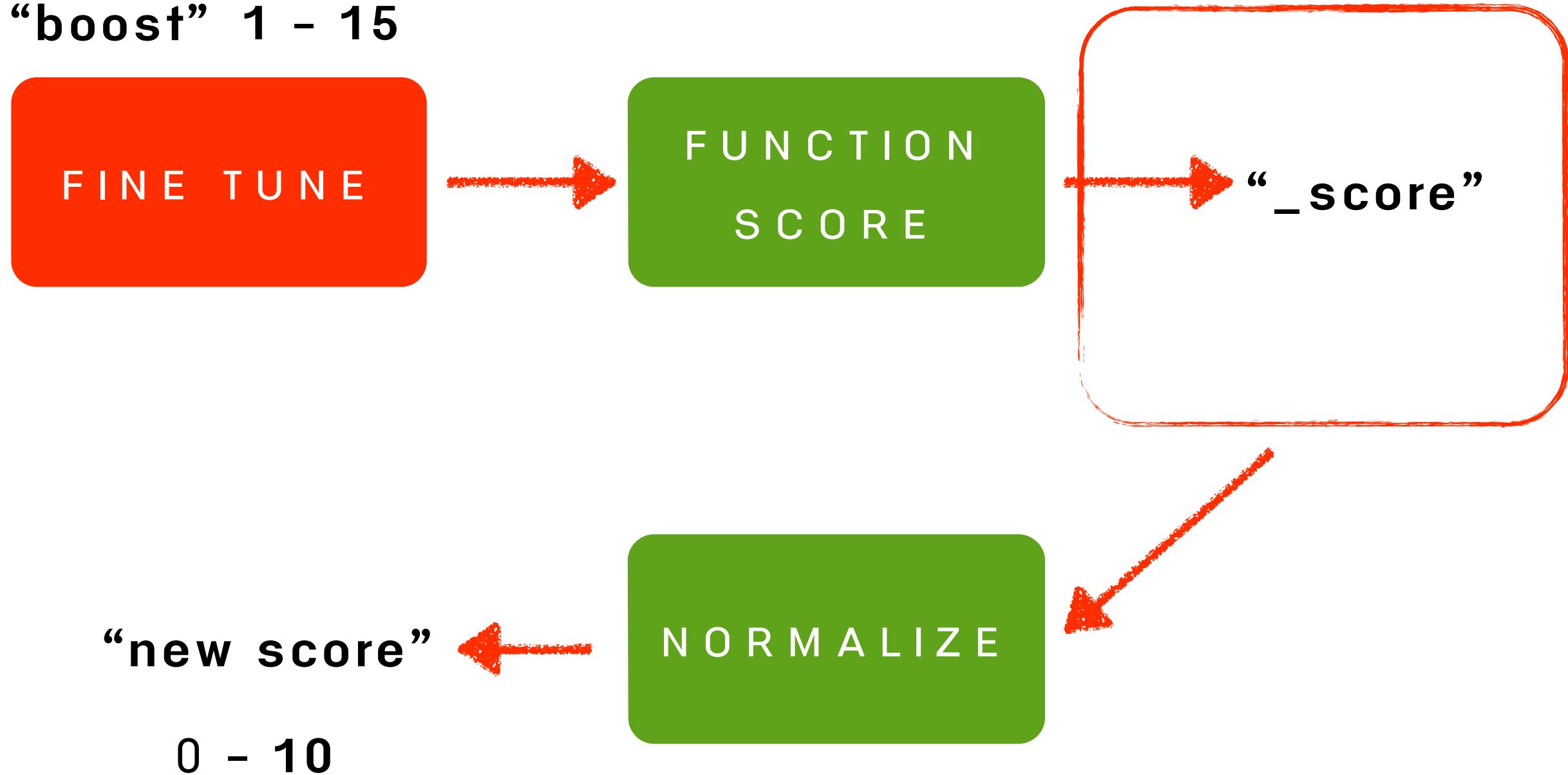
```
PUT /blogs/post/3
{
  "title": "Course elasticsearch for developer",
  "content": "In this post we will talk about...",
  "like": 30
}
```



<http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/boosting-by-popularity.html>

# Boost by popularity

**“boost” 1 - 15**

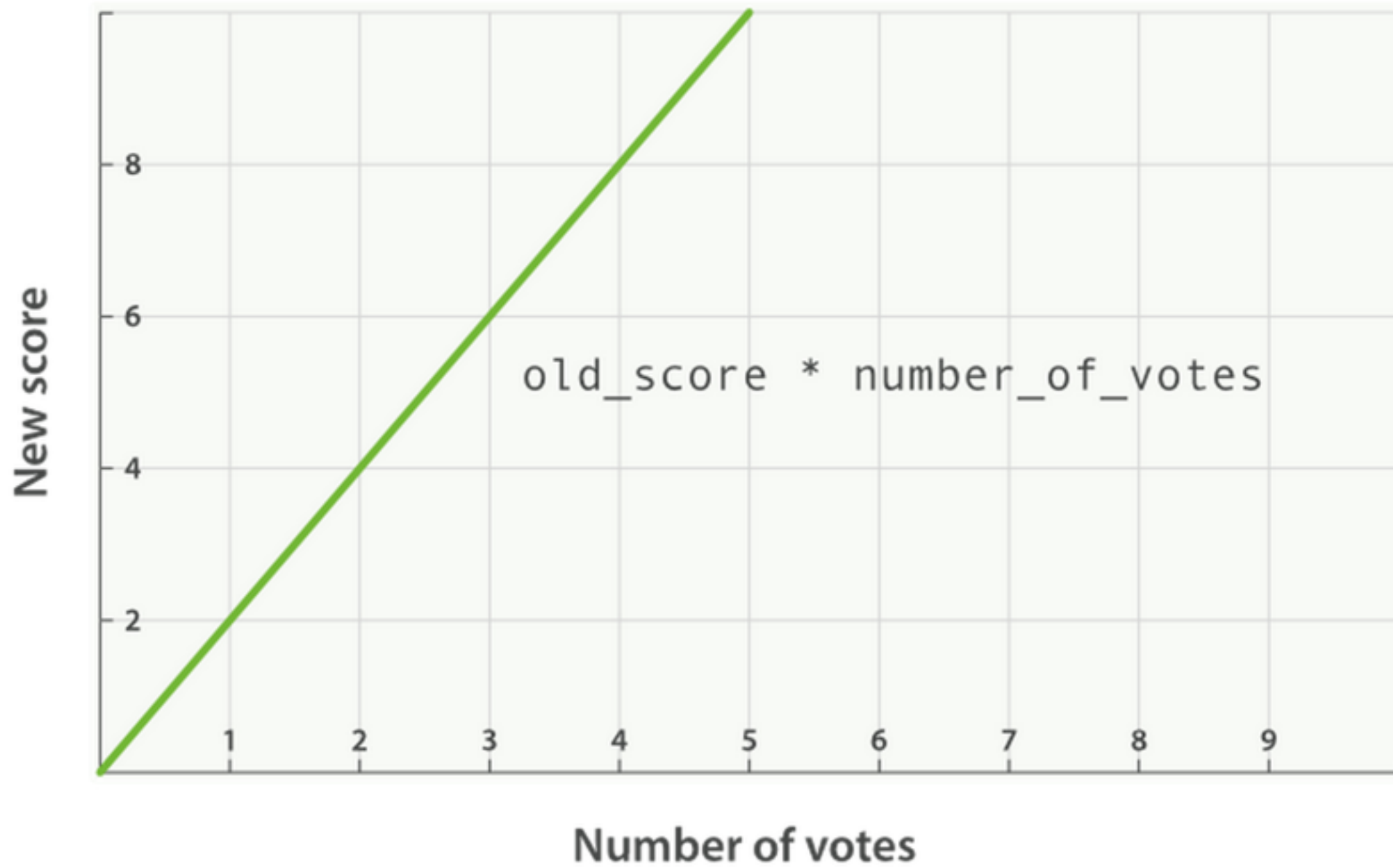


# Query with function\_score

```
GET blogs/_search
{
  "query": {
    "function_score": {
      "query": {
        "multi_match": {
          "query": "course",
          "fields": ["title", "content"]
        }
      },
      "field_value_factor": {
        "field": "like"
      }
    }
  }
}
```



# Linear popularity



# ถ้าต้องการเปลี่ยนล่ะ ?

- เปลี่ยน modifier ได้เช่น

log

log1p

log2p

sqrt

- กำหนด factor ของ modifier

# Modifier & Factor

**GET** blogs/\_search

{

"query": {

"function\_score": {

...

"field\_value\_factor": {

"field": "like",

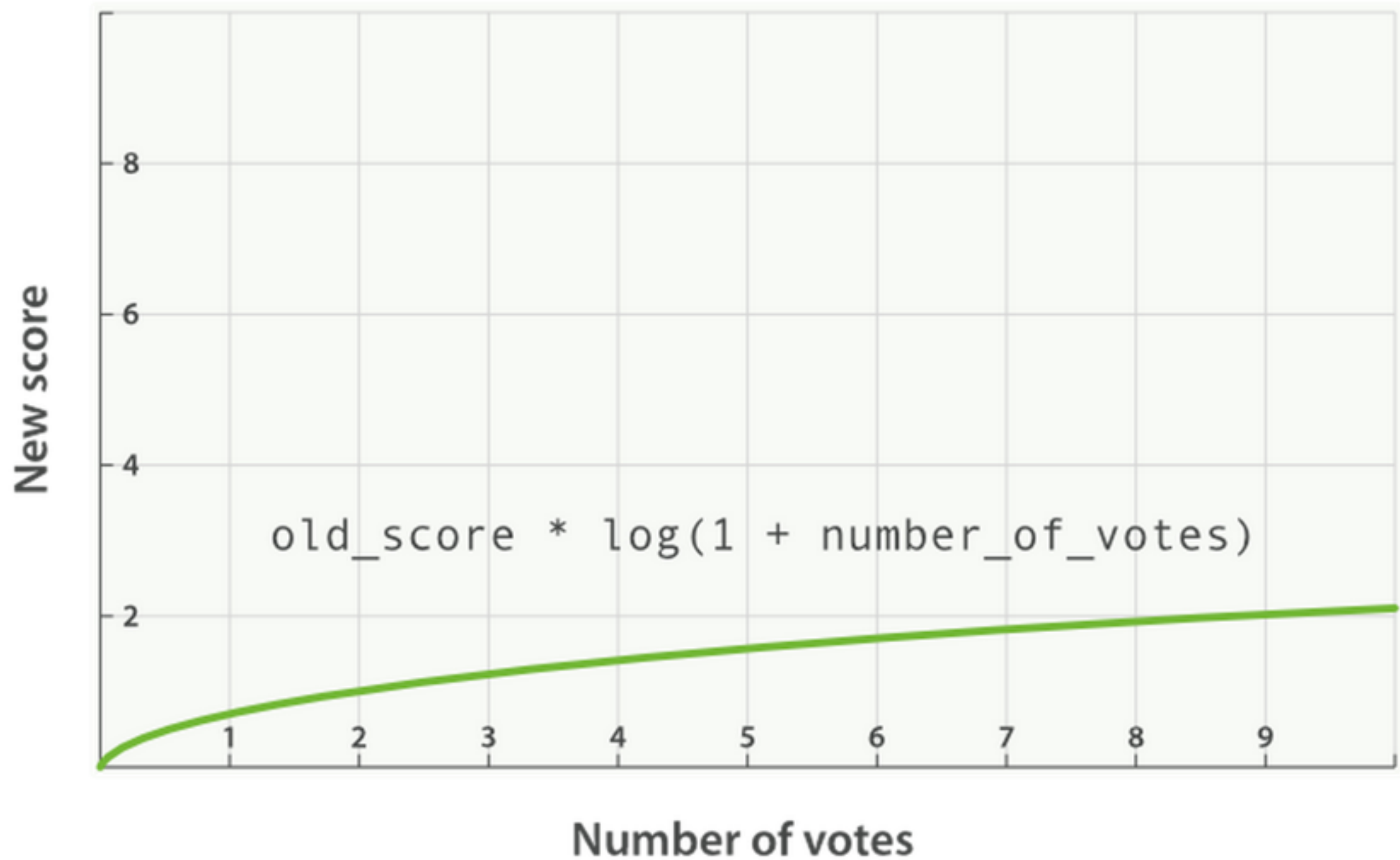
"modifier": "log",

"factor": 1.2

}

}} }

# Log popularity



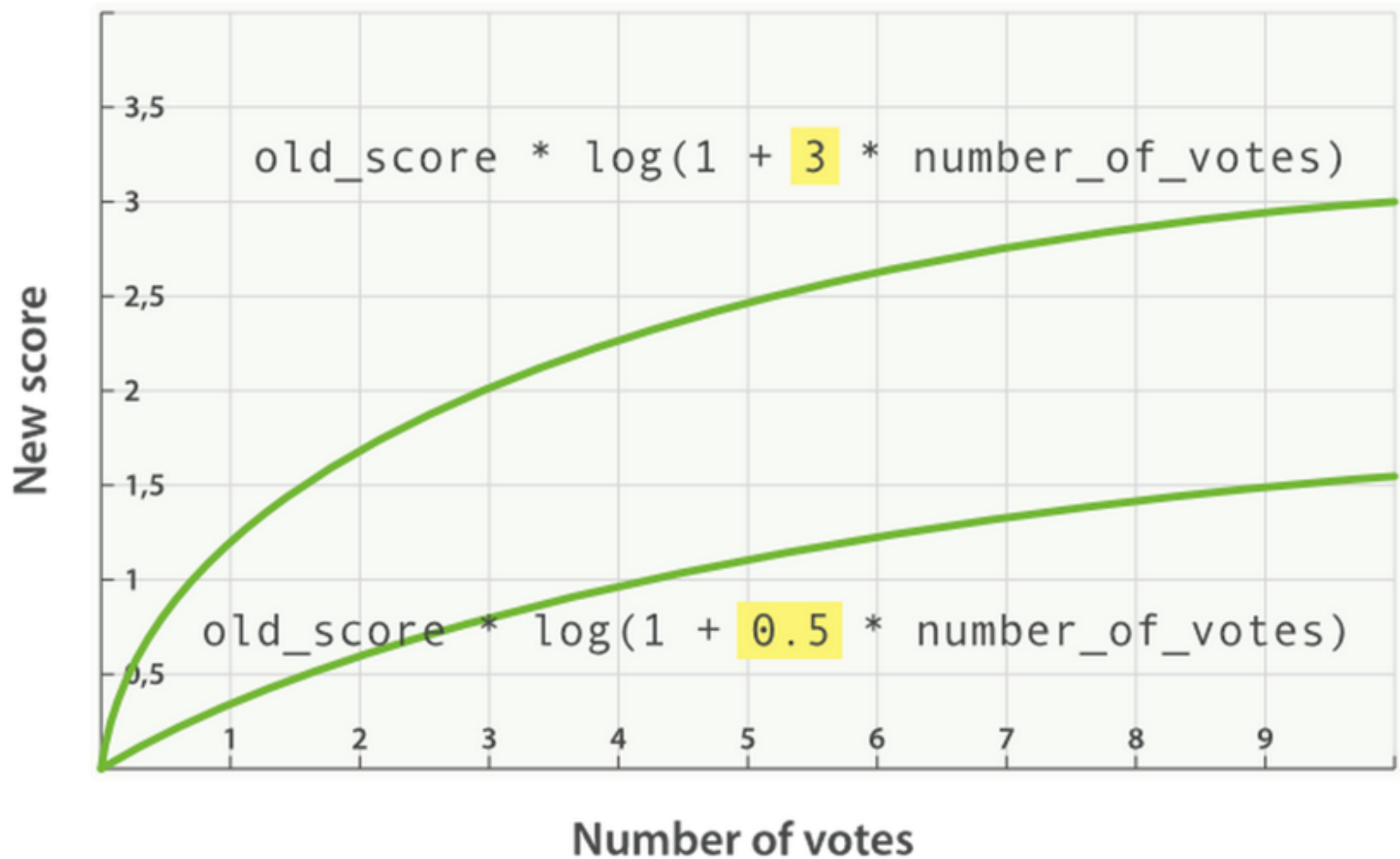
# Factor

old\_score \* **log** ( 1 + **factor** + # of like)



modifier

# Change Factor



# Boost mode

old\_score \* log ( 1 + **factor** + # of like)



Default = multiply

# Boost mode

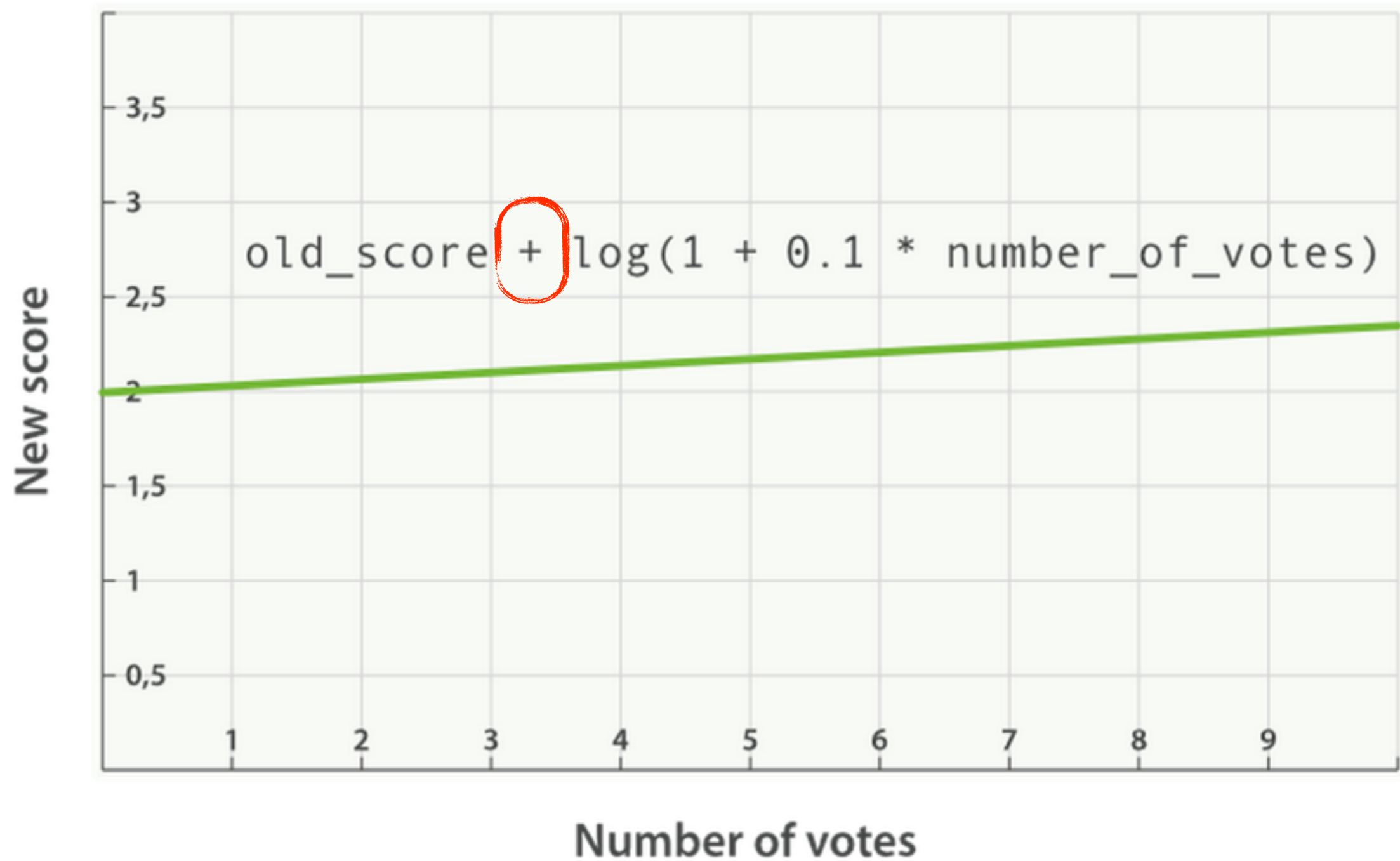
- multiply
- sum
- min
- max
- replace



# Sum boost mode

```
GET blogs/_search
{
  "query": {
    "function_score": {
      ...
      "field_value_factor": {
        "field": "like",
        "modifier": "log",
        "factor": 1.2
      },
      "boost_mode": "sum"
    }
  }
}
```

# Sum boost mode



# Max boost score

```
GET blogs/_search
{
  "query": {
    "function_score": {
      ...
      "field_value_factor": {
        "field": "like",
        "modifier": "log",
        "factor": 1.2
      },
      "boost_mode": "sum",
      "max_boost": 1.5
    }
  }
}
```