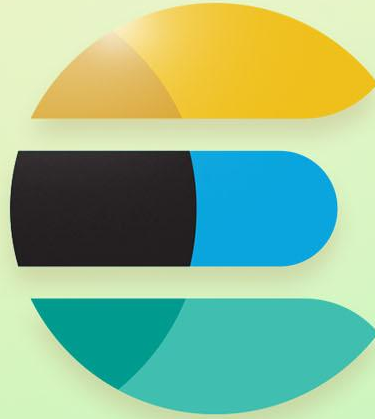


Introduction to relevance scoring



```
GET /products/_search
{
  "query": {
    "term": {
      "tags.keyword": "Vegetable"
    }
  }
}
```



Does the document contain
the "Vegetable" tag?



`_score = 1.0`



The document doesn't match

Query

```
GET /products/_search
{
  "query": {
    "match": {
      "name": "pasta chicken"
    }
  }
}
```

Matching documents

```
{
  "name": "Pasta Penne",
  "_score": 3.76910
}
```

```
{
  "name": "Chicken Breast",
  "_score": 2.53271
}
```

```
{
  "name": "Spinach Pasta",
  "_score": 2.19671
}
```

```
{
  "name": "Pasta with Chicken",
  "_score": 7.26801
}
```

Query results (sorted by _score)

```
{
  "name": "Pasta with Chicken",
  "_score": 7.26801
}
```

```
{
  "name": "Pasta Penne",
  "_score": 3.76910
}
```

```
{
  "name": "Chicken Breast",
  "_score": 2.53271
}
```

```
{
  "name": "Spinach Pasta",
  "_score": 2.19671
}
```

Query

```
GET /products/_search
{
  "query": {
    "match": {
      "name": "Pasta"
    }
  }
}
```

Inverted index for name field

TERM	DOCUMENT #1	DOCUMENT #2	DOCUMENT #3
"with"		X	
"pasta"	X	X	X
"penne"	X		
"chicken"		X	
"spinach"			X

Matching documents

```
{
  "name": "Pasta Penne",
  "_score": 3.76910
}
```

```
{
  "name": "Pasta with Chicken",
  "_score": 2.53271
}
```

```
{
  "name": "Spinach Pasta",
  "_score": 1.95713
}
```

↑↓ sorted by `_score`

Lecture summary

- Query results are sorted descendingly by the `_score` metadata field
 - A floating point number of how well a document matches a query
- Documents matching term level queries are generally scored 1.0
 - Either a document matches, or it doesn't (simply filtered out)
- Full text queries are not for exact matching
 - *How well* a document matches is now a factor
 - The most relevant results are placed highest (e.g. like on Google)