



TEXT CLASSIFICATION WITH ELASTICSEARCH



DATA SCIENCE
COURSE OUTLINE

Overview

Text analysis, tokenization, and stop word removal is as much a function of search as it is of classification, making Elasticsearch the perfect tool for text classification. This course will teach you how to build a classification system using the same features that make Elasticsearch so powerful for search. After exploring precision, recall, F1 scoring, fallback mechanisms and more, you will learn how to use tools like percolation and significant terms aggregation to build better text classification systems in your own use case.

Audience

Data Scientists, Data Architects, Data Engineers, Software Developers

Duration

2–3 hours

Language

English

Prerequisites

- We recommend you have taken [Elasticsearch Engineer I](#) and [Elasticsearch Engineer II](#) or possess equivalent knowledge. Engineer I and Engineer II teach the concepts that are the foundation upon which all specializations are built.
- Familiarity with Python and basic statistics

Requirements

- Stable internet connection
- Mac, Linux, or Windows
- Latest version of Chrome or Firefox (Safari and IE are not 100% supported)
- Due to virtual classroom JavaScript requirements, we recommend that you disable any ad-blockers and restart your browser before class.



TEXT CLASSIFICATION WITH ELASTICSEARCH

Modules

Unit 1

- Learn how to use Elasticsearch to solve classification problems, and how to design classification systems using the Elastic Stack.

- **Hands-On Lab**

Unit 2

- Optimizing a query is not an easy task. In this unit, you will learn how to build a better query using the result of a significant terms aggregations, as well as how to optimize queries using a metaheuristic approach.

- **Hands-on Lab**

Unit 3

- Learn how to use percolators and an optimized query to solve classification problems, as well as the limits of each approach.

- **Hands-on Lab**

Unit 4

- If a percolator fails you should always have some kind of fallback. In this training we are going to see how you could implement one.

- **Hands-on Lab**