Fundamentos de Ingeniería de Datos

Un enfoque práctico

Felipe Ortega

2024-07-17

Tabla de contenidos

Prefacio

Este manual presenta los elementos y conceptos básicos para ingeniería de datos, desde un enfoque práctico. Por tanto, se presentan todos los fundamentos teóricos para el desarrollo de actividades de ingeniería de datos, incluyendo componentes clave como grafos de procesamiento de datos (DAG), *pipelines* de datos, *feature engineering*, gestión y despliegue de modelos, etc. Cada elemento o concepto clave estará acompañado de ejemplos prácticos que ilustran su implementación en proyectos reales, con una o varias tecnologías.

Este manual se ha creado con Quarto. Puedes consultar más información en el siguiente enlace de documentación: https://quarto.org/docs/books.

Parte I

Fundamentos de ingeniería de datos

1 Sistemas y aplicaciones de ingeniería de datos

En cada capítulo podemos integrar partes de código ejecutable en Python, R o Julia junto con contenido formateado en Markdown.

Ejemplo Knuth (1984) de una cita bibliográfica.

Contenidos:

- Ciclo completo de proyectos de ciencia de datos.
  - Esquemas de implementación: lab. vs producción.
  - Registro/catálogo de modelos.
  - CI/CD.
  - Monitorización y seguimiento.
- Arquitecturas de procesamiento de datos: propuestas y tendencias.
- Arquitectura *data mesh*.
  - Propuesta organizativa.
  - Propuesta tecnológica.
- El *data lake*.
  - Concepto y aplicaciones.
  - Ejemplos tecnológicos.
  - Riesgos de aplicaciones y buenas prácticas.
- Casos de ejemplo: algunas combinaciones tecnológicas para arquitecturas de procesamiento de datos.

2 *Pipelines* de procesamiento de datos

Contenidos:

- Concepto de *pipeline* de datos.
  - DAG: grafos de trabajo con datos.
  - Elementos constituyentes: *task*, *stage*, *lane*, *scheduler*, ...
  - Diseño de grafos de trabajo con datos.
- Aplicaciones:
  - Ingesta/obtención de datos.
  - Procesamiento de datos.
  - Ajuste/entrenamiento de modelos/algoritmos.
  - Reajuste/reentrenamiento de modelos/algoritmos.
  - Trasvase de datos streaming –> batch.
- Tecnologías: Apache Airflow.
- Tecnologías: Apache Beam.
- Otros ejemplos tecnológicos: Luigi (Python), Targets (R).

Parte II

Entornos de procesamiento distribuido de datos

3 Tecnologías de procesamiento distribuido de datos

- Apache Spark.
  - Arquitecturas de procesamiento distribuido en Spark.
    - Modo cliente vs. modo despliegue.
  - Detalles de procesamiento distribuido en Spark.
    - DAGs de procesamiento de datos.
    - Optimización a nivel lógico.
    - Optimización a nivel físico.
  - Ejemplos con Spark SQL y Spark Core.
- Apache Flink.
  - Procesamiento *streaming* estricto.
  - Comparativa con Apache Spark.
- Dask (Python).
  - Procesamiento de datos nativo en Python.
  - Modos de procesamiento de datos (multi-core vs. distribuido).
  - Comparativa con Spark y Flink.
- Multiprocesamiento en otros lenguajes.
  - Ejemplos con Go (sólo ilustrativos).

4 Entornos híbridos de procesamiento de datos

Contenidos:

- Soluciones de procesamiento *batch* vs *streaming*.
- Procesamiento streaming de datos estructurados en Spark.
  - Structured Streaming.
- Spark Structured Streaming.
  - Elementos constituyentes.
  - Catálogo de operaciones.
  - Modos de operación y salida.

Parte III

Desarrollo de aplicaciones de procesamiento de datos

5 Procesamiento de datos estructurados

Contenidos:

- Análisis avanzado de datos estructurados con Spark.
  - Operaciones con eventanado (*windows*).
  - *Watermarks* y obsolescencia de datos.
  - Funciones definidas por el usuario (UDFs).
- Ampliación de operaciones *shuffle* en Spark ??

6 Procesamiento de datos semi-estructurados y no estructurados

Contenidos:

- Utilización de bibliotecas/extensiones en Spark.
  - Tratamiento de datos JSON.
  - Tratamiento de datos XML.
- Conexiones con colas de mensajes.
  - Tecnología: Apache Kafka.
  - Conexión Kafka + Spark para procesamiento streaming.

Parte IV

Aprendizaje máquina escalable

7 Aplicaciones escalables de aprendizaje máquina

- Aprendizaje máquina escalable.
  - Diferencias respecto al ML en un solo nodo.
  - Retos y limitaciones.
  - Ejemplos tecnológicos.
    - Spark ML.
    - H2O.
- Spark ML.
  - Principales algoritmos.
  - Ejemplos de algoritmos de aprendizaje supervisado.
  - Ejemplos de algoritmos de aprendizaje no supervisado.
- Redes neuronales y modelos fundacionales.

8 Conclusiones

Resumen y principales conclusiones.

Referencias

Knuth, Donald E. 1984. «Literate Programming». *Comput. J.* 27 (2): 97-111. https://doi.org/10.1093/comjnl/27.2.97.