

The Data Engineering Cookbook

How to master the plumbing of data science

Andreas Kretz

January 6, 2019

v0.4

Contents

1	Introduction	6
2	What is Data Science?	7
2.1	Data Scientist	7
2.2	Data Engineer	8
2.3	Data Analyst	9
2.4	Who Companies Need	9
3	The Basic Skills	9
4	Learn to Write Code	9
4.1	Coding Basics	10
4.2	Learn To Use GitHub — available	10
4.3	Agile Development – available	11
4.3.1	Agile rules I learned over the years – available	12
4.3.2	Scrum	13
4.3.3	OKR	13
5	Computer Science Basics	14
5.1	Learn how a Computer Works	14
5.1.1	CPU,RAM,GPU,HDD	14
5.1.2	Differences between PCs and Servers	14
5.2	Computer Networking	14
5.2.1	ISO/OSI Model	14
5.2.2	IP Subnetting	14
5.2.3	Switch, Level 3 Switch	14
5.2.4	Router	14
5.2.5	Firewalls	14
5.3	Security and Privacy	15
5.3.1	SSL Public & Private Key Certificates	15
5.3.2	What is a certificate authority	15
5.3.3	JAVA Web Tokens	15
5.3.4	GDPR regulations	15
5.3.5	Privacy by design	15
5.4	Linux	15
5.4.1	OS Basics	15
5.4.2	Shell scripting	15
5.4.3	Cron jobs	15

5.4.4	Packet management	15
5.5	The Cloud	15
5.5.1	AWS,Azure, IBM, Google Cloud basics	15
5.5.2	cloud vs on premise	15
5.5.3	up & downsides	15
5.5.4	Security	15
6	My Big Data Platform Blueprint – available	16
6.1	Ingest – available	17
6.2	Store – available	17
6.3	Analyse / Process – available	18
6.4	Display – available	18
7	Data Science Platform	19
7.1	Security Zone Design	19
7.1.1	How to secure a multi layered application	19
7.1.2	Cluster security with Kerberos	19
7.2	Lambda Architecture	19
7.2.1	Stream and Batch processing – available	19
7.3	Three methods of streaming — available	21
7.4	Big Data	23
7.4.1	What is big data and where is the difference to data science and data analytics?	23
7.4.2	The 4Vs of Big Data — available	23
7.4.3	Why Big Data? — available	24
7.4.4	What are the tools associated?	28
7.5	What is the difference between a Data Warehouse and a Data Lake	28
7.6	Hadoop Platforms — available	28
7.6.1	What makes Hadoop so popular? — available	28
7.6.2	How does a Hadoop System architecture look like	32
7.6.3	What tools are usually in a with Hadoop Cluster	32
7.6.4	How to select Hadoop Cluster Hardware	32
7.7	Is ETL still relevant for Analytics?	32
7.8	Docker	32
7.8.1	What is docker and what do you use it for — available	32
7.8.2	How to create, start,stop a Container	34
7.8.3	Docker micro services?	34
7.8.4	Kubernetes	34
7.8.5	Why and how to do Docker container orchestration	34

8	How to Ingest Data	34
8.1	Application programming interfaces	34
8.1.1	REST APIs	35
8.1.2	HTTP Post/Get	35
8.1.3	API Design	35
8.1.4	Implementation	35
8.1.5	OAuth security	35
8.2	JSON	35
8.2.1	Super important for REST APIs	35
8.2.2	How is it used for logging and processing logs	35
9	Distributed Processing	35
9.1	MapReduce – available	35
9.1.1	How does MapReduce work – available	36
9.1.2	What is the limitation of MapReduce? – available	39
9.2	Apache Spark	39
9.2.1	Spark Basics	39
9.2.2	What is the difference to MapReduce? – available	39
9.2.3	How does Spark fit to Hadoop? – available	40
9.2.4	Available Languages – available	42
9.2.5	How to do stream processing	42
9.2.6	How to do batch processing	42
9.2.7	How does Spark use data from Hadoop – available	42
9.2.8	What is a RDD and what is a DataFrame?	43
9.2.9	Spark coding with Scala	43
9.2.10	Spark coding with Python	43
9.2.11	How and why to use SparkSQL?	43
9.2.12	Machine Learning on Spark? (Tensor Flow)	43
9.2.13	Spark Setup – available	43
9.2.14	Spark Resource Management – available	44
9.3	Message queues with Apache Kafka	45
9.3.1	Why a message queue tool?	45
9.3.2	Kakfa architecture	45
9.3.3	What are topics	45
9.3.4	What does Zookeeper have to do with Kafka	45
9.3.5	How to produce and consume messages	45
9.4	Machine Learning	45
9.4.1	Training and Applying models	45
9.4.2	What is deep learning	45

9.4.3	How to do Machine Learning in production — available	45
9.4.4	Why machine learning in production is harder then you think – available	46
9.4.5	How to convince people machine learning works — available . . .	47
10	How to Store Data	49
10.1	Data Modeling	49
10.1.1	How to find out how you need to store data for the business case .	49
10.1.2	How to decide what kind of storage you need to use	49
10.2	SQL	49
10.2.1	Database Design	49
10.2.2	SQL Queries	49
10.2.3	Stored Procedures	49
10.2.4	ODBC/JDBC Server Connections	49
10.3	NoSQL	49
10.3.1	KeyValue Stores (HBase)	49
10.3.2	Document Store HDFS — available	49
10.3.3	Document Store MongoDB	51
10.3.4	Hive Warehouse	51
10.3.5	Impala	51
10.3.6	Time Series Databases (?)	51
10.3.7	MPP Databases (Greenplum)	51
11	How to Visualize Data	51
11.1	Mobile Apps	51
11.1.1	Android & IOS basics	51
11.1.2	How to design APIs for mobile apps	51
11.2	How to use Webservers to display content	51
11.2.1	Tomcat	52
11.2.2	Jetty	52
11.2.3	NodeRED	52
11.2.4	React	52
11.3	Business Intelligence Tools	52
11.3.1	Tableau	52
11.3.2	PowerBI	52
11.3.3	Quliksense	52
11.4	Identity & Device Management	52
11.4.1	What is a digital twin?	52
11.4.2	Active Directory	52

12 Case Studies	52
12.1 7 Steps to Successfull Data Science Project	52
12.2 Data Science @Airbnb	52
12.3 Data Science @Netflix – available	52
12.4 Data Science @Uber	56
12.5 Data Sciecne @Zalando	56
13 DEV/OPS	56
13.1 Hadoop	56
13.1.1 Hadoop Cluster setup and management with Cloudera Manager (for example)	56
13.1.2 Spark code from coding to production	56
13.1.3 How to monitor and manage data processing pipelines	56
13.1.4 Oozie	56
13.1.5 Airflow Application management	56
13.1.6 Creating Statistics with Spark and Kafka	56

1 Introduction

What do you actually need to learn to become an awesome data engineer? Look no further, you find it here.

How to use this document: This is not a training! It's a collection of skills, that I value highly in my daily work as a data engineer. It's intended to be a starting point for you to find the topics to look into.

This project is a work in progress! Over the next weeks I am going to share with you my thoughts on why each topic is important. I also try to include links to useful resources.

How to find out what is new? You will always find the newest version on my Patreon <https://www.patreon.com/plumbersofds>

Help make this collection awesome! Join the discussion on Patreon or write me an email to andreaskayy@gmail.com. Tell me your thoughts, what you value, you think should be included, or where I am wrong.

– Andreas

2 What is Data Science?

2.1 Data Scientist

Data scientists aren't like every other scientist.

Data scientists do not wear white coats or work in high tech labs full of science fiction movie equipment. They work in offices just like you and me.

What differs them from most of us is that they are the math experts. They use linear algebra and multivariable calculus to create new insight from existing data.

How exactly does this insight look?

Here's an example:

An industrial company produces a lot of products that need to be tested before shipping.

Usually such tests take a lot of time because there are hundreds of things to be tested. All to make sure that your product is not broken.

Wouldn't it be great to know early if a test fails ten steps down the line? If you knew that you could skip the other tests and just trash the product or repair it.

That's exactly where a data scientist can help you, big-time. This field is called predictive analytics and the technique of choice is machine learning.

Machine what? Learning?

Yes, machine learning, it works like this:

You feed an algorithm with measurement data. It generates a model and optimises it based on the data you fed it with. That model basically represents a pattern of how your data is looking. You show that model new data and the model will tell you if the data still represents the data you have trained it with. This technique can also be used for predicting machine failure in advance with machine learning. Of course the whole process is not that simple.

The actual process of training and applying a model is not that hard. A lot of work for the data scientist is to figure out how to pre-process the data that gets fed to the algorithms.

Because to train a algorithm you need useful data. If you use any data for the training the produced model will be very unreliable.

A unreliable model for predicting machine failure would tell you that your machine is damaged even if it is not. Or even worse: It would tell you the machine is ok even when there is an malfunction.

Model outputs are very abstract. You also need to post-process the model outputs to receive health values from 0 to 100.

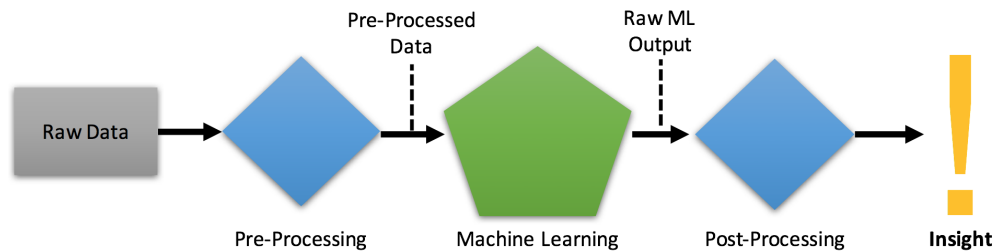


Figure 1: The Machine Learning Pipeline

2.2 Data Engineer

Data Engineers are the link between the management’s big data strategy and the data scientists that need to work with data.

What they do is building the platforms that enable data scientists to do their magic.

These platforms are usually used in four different ways:

- Data ingestion and storage of large amounts of data
- Algorithm creation by data scientists
- Automation of the data scientist’s machine learning models and algorithms for production use
- Data visualisation for employees and customers

– Most of the time these guys start as traditional solution architects for systems that involve SQL databases, web servers, SAP installations and other “standard” systems.

But to create big data platforms the engineer needs to be an expert in specifying, setting up and maintaining big data technologies like: Hadoop, Spark, HBase, Cassandra, MongoDB, Kafka, Redis and more.

What they also need is experience on how to deploy systems on cloud infrastructure like at Amazon or Google or on premise hardware.

2.3 Data Analyst

2.4 Who Companies Need

For a good company it is absolutely important to get well trained data engineers and data scientists.

Think of the data scientist as the professional race car driver. A fit athlete with talent and driving skills like you have never seen.

What he needs to win races is someone who will provide him the perfect race car to drive. That's what the solution architect is for.

Like the driver and his team the data scientist and the data engineer need to work closely together. They need to know the different big data tools Inside and out.

That's why companies are looking for people with Spark experience. It is a common ground between both that drives innovation.

Spark gives data scientists the tools to do analytics and helps engineers to bring the data scientist's algorithms into production.

After all, those two decide how good the data platform is, how good the analytics insight is and how fast the whole system gets into a production ready state.

3 The Basic Skills

4 Learn to Write Code

Why this is important: Without coding you cannot do much in data engineering. I cannot count the number of times I needed a quick Java hack.

The possibilities are endless:

- Writing or quickly getting some data out of a SQL DB
- Testing to produce messages to a Kafka topic
- Understanding Source code of a Java Webservice
- Reading counter statistics out of a HBase key value store

6 My Big Data Platform Blueprint – available

Some time ago I have created a simple and modular big data platform blueprint for myself. It is based on what I have seen in the field and read in tech blogs all over the internet.

Today I am going to share it with you.

Why do I believe it will be super useful to you?

Because, unlike other blueprints it is not focused on technology. It is based on four common big data platform design patterns.

Following my blueprint will allow you to create the big data platform that fits exactly your needs. Building the perfect platform will allow data scientists to discover new insights.

It will enable you to perfectly handle big data and allow you to make data driven decisions.

THE BLUEPRINT – available The blueprint is focused on the four key areas: Ingest, store, analyse and display.

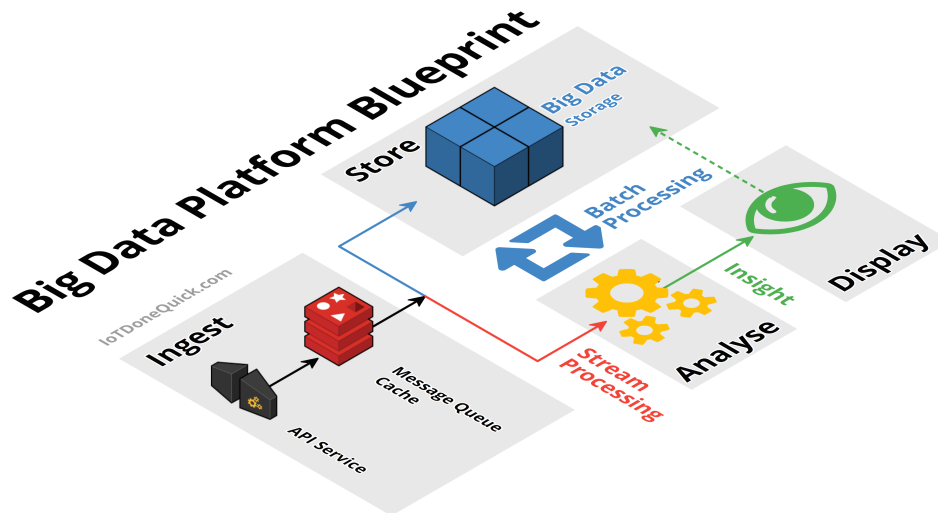


Figure 2: Platform Blueprint

Having the platform split like this turns it into a modular platform with loosely coupled interfaces.

Why is it so important to have a modular platform?

If you have a platform that is not modular you end up with something that is fixed or

hard to modify. This means you can not adjust the platform to changing requirements of the company.

Because of modularity it is possible to switch out every component, if you need it.

Now, lets talk more about each key area.

6.1 Ingest – available

Ingestion is all about getting the data in from the source and making it available to later stages. Sources can be everything from tweets, server logs to IoT sensor data like from cars.

Sources send data to your API Services. The API is going to push the data into a temporary storage.

The temporary storage allows other stages simple and fast access to incoming data.

A great solution is to use messaging queue systems like Apache Kafka, RabbitMQ or AWS Kinesis. Sometimes people also use caches for specialised applications like Redis.

A good practice is that the temporary storage follows the publish, subscribe pattern. This way APIs can publish messages and Analytics can quickly consume them.

6.2 Store – available

This is the typical big data storage where you just store everything. It enables you to analyse the big picture.

Most of the data might seem useless for now, but it is of upmost importance to keep it. Throwing data away is a big no no.

Why not throw something away when it is useless?

Although it seems useless for now, data scientists can work with the data. They might find new ways to analyse the data and generate valuable insight from it.

What kind of systems can be used to store big data?

Systems like Hadoop HDFS, Hbase, Amazon S3 or DynamoDB are a perfect fit to store big data.

6.3 Analyse / Process – available

The analyse stage is where the actual analytics is done. Analytics, in the form of stream and batch processing.

Streaming data is taken from ingest and fed into analytics. Streaming analyses the “live” data thus, so generates fast results.

As the central and most important stage, analytics also has access to the big data storage. Because of that connection, analytics can take a big chunk of data and analyse it.

This type of analysis is called batch processing. It will deliver you answers for the big questions.

To learn more about stream and batch processing read my blog post: [How to Create New and Exciting Big Data Aided Products](#)

The analytics process, batch or streaming, is not a one way process. Analytics also can write data back to the big data storage.

Often times writing data back to the storage makes sense. It allows you to combine previous analytics outputs with the raw data.

Analytics insight can give meaning to the raw data when you combine them. This combination will often times allow you to create even more useful insight.

A wide variety of analytics tools are available. Ranging from MapReduce or AWS Elastic MapReduce to Apache Spark and AWS lambda.

6.4 Display – available

Displaying data is as important as ingesting, storing and analysing it. People need to be able to make data driven decisions.

This is why it is important to have a good visual presentation of the data. Sometimes you have a lot of different use cases or projects using the platform.

It might not be possible for you to build the perfect UI that fits everyone. What you should do in this case is enable others to build the perfect UI themselves.

How to do that? By creating APIs to access the data and making them available to developers.

Either way, UI or API the trick is to give the display stage direct access to the data in the big data cluster. This kind of access will allow the developers to use analytics results as well as raw data to build the the perfect application.

7 Data Science Platform

7.1 Security Zone Design

7.1.1 How to secure a multi layered application

(UI in different zone then SQL DB)

7.1.2 Cluster security with Kerberos

I talked about security zone design and lambda architecture in this podcast: <https://anchor.fm/andreask-to-Design-Security-Zones-and-Lambda-Architecture-PoDS-032-e248q2>

7.2 Lambda Architecture

7.2.1 Stream and Batch processing – available

Batch Processing: Ask the big questions. Remember your last yearly tax statement?

You break out the folders. You run around the house searching for the receipts.

All that fun stuff.

When you finally found everything you fill out the form and send it on its way.

Doing the tax statement is a prime example of a batch process.

Data comes in and gets stored, analytics loads the data from storage and creates an output (insight):



Figure 3: Batch Processing Pipeline

References

- [1] J. Ely and I. Stavrov, *Analyzing chalk dust and writing speeds: computational and geometric approaches*, BoDine Journal of Mathematics **3** (2001), 14-159.

List of Figures

1	The Machine Learning Pipeline	8
2	Platform Blueprint	16
3	Batch Processing Pipeline	19
4	Stream Processing Pipeline	20
5	Common SQL Platform Architecture	25
6	Scaling up a SQL Database	26
7	Scaling out a SQL Database	27
8	Hadoop Ecosystem Components	29
9	Connections between tools	30
10	Flume Integration	31
11	Mapping of input files and reducing of mapped records	36
12	MapReduce Example of Time Series Data	38
13	The Map Reduce Process	39
14	Hadoop vs Spark capabilities	40
15	Spark Using Hadoop Data Locality	43
16	Spark Resource Management With YARN	44
17	HDFS Master and Data Nodes	50
18	Distribution of Blocks for a 512MB File	51
19	Old Netflix Batch Processing Pipeline	53
20	Netflix Trending Now Feature	55
21	Netflix Streaming Pipeline	56

List of Tables