The Data Engineering Handbook

Build durable, scalable, orchestrated software in Go using Kubernetes

Lex Sheehan

Data Engineering Handbook

Build durable, scalable, orchestrated software in Go with Kubernetes

Lex Sheehan

This book is for sale at http://leanpub.com/dataengineeringhandbook

This version was published on 2020-03-04

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Preface (DRAFT)

A Data Engineer is a new type of role: a specialization of a software engineer with additional strengths in data modeling, databases, and distributed systems. It is similar to the difference between your family doctor and a Oncologist: they have the same fundamental training and practice, but the Oncologist specializes. Successful data engineers build durable, scaleable and orchestrated software.

This book covers current industry standard technologies for machine learning and data analytics and provides foundational understanding of the field of data science.

The chapters progress through various design, testing,implementation, deployment techniques; including creating ML platofrms, building analytica pipelines, the use of monitoring tools and developing Kubeflow solutions in Go.

We'll understand why data analysts prefer Python and R and data analysis tools and why we can typically do better using Go and Kubernetes.

1 Intro to Data Science and Machine Learning (DRAFT)

Data Science

Fueled by social media, online business, the increase in computing power and data from emerging markets, data is everywhere. The amount of digital data that exists is growing at a rapid rate, doubling every two years.

The goal of data science is to improve decision making by basing decisions on insights extracted from these large data sets.

Data Science deals with both structured–think "database"–and unstructured data, i.e., everything else. Data Science activites include data cleansing, preparation, analysis: extracting nonobvious and useful patterns from large data sets and communicating actionable insights to business stakeholders.

Data Science Modeling

Data Science Modeling encompasses all of the techniques used when trying to extract insights and information from data.

Steps To Create a Data Science Model

1. Identify the Goals
2. Data Acquisition
3. Data Preparation
4. Exploratory Data Analysis
5. Data Modeling
6. Visualization and Communication
7. Deployment and Maintenance

1. Identify the Goals

This is the planning phase where we identify, understand and define the goals and objectives of this project.

The Data Analyst asks a lot of questions to understand the business problem:

- What decisions do we expect to make from this data?
- What questions should we ask of this data?
- What level of confidence is required in our answers?
- What does the ideal result set look like?

2. Data Acquisition

We gather data from logs, databases, API's, web sites and online repositories.

3. Data Preparation

Data preparation includes data cleaning and data transformation.

Data cleaning can be very time consuming and includes correcting inconsistent datatypes, misspelled data and data attributes, correcting inconsistent formats, e.g., dates, locations, missing and duplicate data values.

Data transformation moves data from its source to a target destination and modifies the data, based on defined mapping rules.

Understanding the aspects of your data, e.g., its overall size, can help in deciding how to procede with your analyses.

Analysis of Small datasets might in memory using tools like Python, Jupyter, and Pandas, or R.

Larger datasets are better handled in an indexed, SQL database, e.g., PostgreSQL.

Consider using Hadoop and/or Apache Spark for extremely large datasets.

ETL Tools

Extract, Transform Load (ETL) tools like Talend[1] and Informatica PowerCenter[2] can be used to assist in performing complex data transformations that can help us to understand the data better.

4. Exploratory Data Analysis

Exploratory Data Analysis is where we try to understand what we can do with our data.

Garbage in Garbage out.

This is arguably the most important step because it's here that we define and refine the selection of feature variables that will be used in the model development.

The wrong variables will produce an inaccurate model.

[1] https://www.talend.com/
[2] https://www.informatica.com/products/data-integration/powercenter.html

Example of Feature Variables

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

There are 1460 instances of training data and 1460 of test data. Total number of attributes equals 81, of which 36 are numerical, 43 are categorical + Id and SalePrice.

Numerical Features: 1stFlrSF, 2ndFlrSF, 3SsnPorch, BedroomAbvGr, BsmtFinSF1, BsmtFinSF2, BsmtFullBath, BsmtHalfBath, BsmtUnfSF, EnclosedPorch, Fireplaces, FullBath, GarageArea, GarageCars, GarageYrBlt, GrLivArea, HalfBath, KitchenAbvGr, LotArea, LotFrontage, LowQualFinSF, MSSubClass, MasVnrArea, MiscVal, MoSold, OpenPorchSF, OverallCond, OverallQual, PoolArea, ScreenPorch, TotRmsAbvGrd, TotalBsmtSF, WoodDeckSF, YearBuilt, YearRemodAdd, YrSold

Categorical Features: Alley, BldgType, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, BsmtQual, CentralAir, Condition1, Condition2, Electrical, ExterCond, ExterQual, Exterior1st, Exterior2nd, Fence, FireplaceQu, Foundation, Functional, GarageCond, GarageFinish, GarageQual, GarageType, Heating, HeatingQC, HouseStyle, KitchenQual, LandContour, LandSlope, LotConfig, LotShape, MSZoning, MasVnrType, MiscFeature, Neighborhood, PavedDrive, PoolQC, RoofMatl, RoofStyle, SaleCondition, SaleType, Street, Utilitif

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Exploratory Data Analysis Tools

Jupyter Notebook[3] - to share documents containing live code, equations, and visualizations
Numpy[4] - Fundamental Python library
Pandas[5] - Python library and data analysis tool
Matplotlib[6] - 2D plotting library useful for finding invalid data
Seaborn[7] - Python data visualization library based on matplotlib

5. Data Modeling and Analysis

First, we apply Machine Learning techniques like Decision Tree, KNN, and Naive Bayes[8] to the data to get our data into a state that is ready for analysis.

Next, we train the models on the training dataset and test.

Then, we select the best performing model.

[3] https://jupyter.org/
[4] http://www.numpy.org/
[5] https://pandas.pydata.org/
[6] https://matplotlib.org/
[7] https://seaborn.pydata.org/
[8] https://www.datasciencecentral.com/profiles/blogs/comparing-classifiers-decision-trees-knn-naive-bayes

Data Modeling and Analysis Tasks

If there is time-based data, explore whether there exist trends in certain fields over time — usually using a time-based visualization software such as Superset or Grafana

If there is location-based data, explore the relationships of certain fields by area — usually using mapping software such as Leaflet JS, and spatial querying (we use PostgreSQL with PostGIS)

Explore correlations (r values) between different fields

Classify text using natural language processing methods (such as the bag of words model)

Implement various machine learning techniques in order to identify trends between multiple variables/fields — regression analyses can be useful

If there are many variables/fields, dimensionality reduction techniques (like Principle Component Analyses) can be used to reduce these to a smaller subset of variables that retain most of the information

Deep learning and neural networks have much potential, especially for much larger, structured datasets (though we have not yet made substantial use of this)

Data Modeling Tools

Tools used to model data include:

PostgreSQL[9] - querying, including spatial querying — for SQLite, see SpatiaLite
JetBrains DataGrip[10] - Database IDE
Datasette[11]- a tool for exploring and publishing data
Jupyter Notebook[12] - allows for sharing of documents containing live code, equations, and visualizations
SciPy[13] - Python library for advanced calculations
NumPy[14] & Pandas[15] - Python data analyses/manipulation libraries
Scikit-Learn[16] - Python machine learning library
Tensor Flow[17] - Python machine learning library generally used for deep learning and neural networks
Keras[18] - Python library for fast experimentation with neural networks

[9] https://www.postgresql.org/
[10] https://www.jetbrains.com/datagrip/
[11] https://datasette.readthedocs.io/en/stable/
[12] https://jupyter.org/
[13] https://www.scipy.org/
[14] https://numpy.org/
[15] https://pandas.pydata.org/
[16] https://scikit-learn.org/stable/
[17] https://www.tensorflow.org/
[18] https://pypi.org/project/Keras/

6. Visualization and Communication

The business analyst then communicates findings with the business owner using tools like Tableau, PowerBI, Looker and ClikView that help create powerful reports and dashboards.

7. Deploy and Maintain Data Model

This is arguably the most technically challenging phase of our project. Challenges include:

- Managing Data Science Languages
- Compute Power Requirements
- Portability
- Scalability
- Usage Spikes
- Version Control
- Tracking Effects of Configuration Changes
- Changing Shape of Ingress Data
- Testing and Validation Issues
- Degredation in Performance Over Time

ML models typically need to be updated more frequenty that typical software applications.

Release Strategies

Shadowing is the technique by which production traffic to any given service is captured and replayed against the newly deployed version of the service. Shadowing has no production impact. Since traffic is duplicated, any bugs in services that are processing shadow data have no impact on production.

Since there is no production impact, shadowing provides a powerful technique to test persistent services. You can configure your test service to store data in a test database, and shadow traffic to your test service for testing. Both blue/green deployments and canary deployments require more machinery for testing.

Shadowing lets you measure the behavior of your service and compare it with an expected output. A typical canary rollout catches exceptions,e.g., 402 data validation errors, but what happens when your service has a logic error and is not returning an exception?

Another release strategy, called a canary release, is to push changes to a small group of end users. Sincehe canary is only distributed to a small number of users, its impact is relatively small and changes can be reversed quickly should the new change prove to be buggy.

Both Shadowing and Canary tests are often automated and are run after testing in a sandbox environment has been completed.

Machine Learning

Machine Learning (ML) is an application of artificial intelligence (AI) that provides computer systems the ability to automatically learn and improve from experience without being explicitly programmed. The primary goal is to allow the computers learn automatically without human intervention.

The process of learning begins with observations of data. Data patterns are discovered to make better decisions in the future based on the example data provided.

Data Science

Data Science is the study of turning raw data into actionable information. Large amounts of structured and unstructured data are mined in order to identify patterns and can help an organization reduce costs, increase efficiencies, recognize new market opportunities and increase it's competitive advantage.

Computer Engineeering + Statistics = Data Science

Data Science:

- Unifies statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data
- Employs techniques from various fields such as mathematics, statistics, computer science, and information science
- Uses algorithms, processes and computational analysis of large data as its primary scientific method
- Extracts knowledge and insights from structured and unstructured data
- Is related to data mining and big data, but not the same in that an extremely large data set is not a requirement

Data Scientists

Data Scientists combine statistical skills and software engineering to make data insightful.

There are two main types of Data Scientists:

The Data Analyst

This type of Data Scientists is primarily concerned with making sense of data or working with using existing tooling and typically:

- possess a combination of analytic, machine learning, data mining and statistical skills as well as experience with algorithms and coding
- can perform data cleaning, methods for dealing with very large data sets
- regularly handles a wide variety of poorly understood and questionably obtained data
- understands experimental design, modeling, statistical validity, forecasting, interpretation, statistical inference
- makes observations about data that requires skills in domain knowledge, visualization, data engineering, database administration and statistical modeling
- uses the scientific method to glean value from data:

  makes observation > questions observaction and gathers data > forms a hypotheis to explain the observation and make predictions based on hypothesis > tests the hypothesis (and predictions) using a reproducable experiment

- uses data visualization to gain deep knowledge of a particular domain
- possesses expert skills in communication, logic, domain knowledge, research, statistics and mathematics
- may also be known as Statistician, Quantitative Analyst, Decision Support Engineering Analyst

The Software Engineer

This type of Data Scientist build production systems to deliver products that extract insight from data and typically:

- build models that capture behavioral data, e.g. user interactions or transactions with systems, sensor

logs, to transform the user's experience into insights

- creates tools and approaches to an unseen problems or troublesome data, as opposed to being an expert in applying a tool for similar results
- codes system components that use the feedback loop (data > algorithm > interface > action > REPEAT) to feed the data science algorithms with learning material
- works on the backend, administrating the operating system, building data pipelines, deploying and managing cloud infrastructure, implementing enterprise software packages and managing databases to ingest and store large scale data (Big Data)
- may also be a Data Analyst

Summary

There is a lot of valuable data in the digital universe, but it will take determination and skilled workforce to find and put to use. It will need to be protected, analyzed, and acted upon.

2 Unit Testing and Test Driven Development Methodologies

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

3 Data-intensive APIs Using a RESTful Approach

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

4 Designing and Developing Distributed Systems

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

5 Data Ingress from IoT Devices

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

6 Stream Processing Using Apache Kafka

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

7 Interacting with the SQS Distributed Queuing System

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

10 Securing Deployments and Cloud-Ops

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

11 Continuous Integration and Continuous Delivery

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

13 Google Cloud Dataflow SDK on Apache Beam

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

14 Applying Resource-Oriented Design Concepts

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

15 Creating Machine Learning (ML) Platforms

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

16 Building Analytical Pipelines

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

17 Big Data Algorithms and Data Structures

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

18 Analyzing Data with Python and R

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

19 Application of Quantitative Science to Solve Business Problems

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

20 Prometheus, Grafana, and other Application Monitoring Tools

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.

21 Develop Kubeflow Solutions in Go

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/dataengineeringhandbook.