
zühlke
empowering ideas

Who am I and why am I here?

I'm Wolfgang and I work for Zühlke's Data Analytics Team...

...and we're always looking for data engineering talents.

I have been here before: Enterprise Computing and "Fast and Furious".

You (so I've been told) are curious to hear real-life stories.

It appears we have a deal!

Resources:

https://github.com/smurve/HSR2019 (https://github.com/smurve/HSR2019)

https://github.com/Project-Ellie/home-in-time (https://github.com/Project-Ellie/home-in-time)

Data Engineering is Software Engineering

Data engineers write software that deals with data.

Data engineers are in high demand.

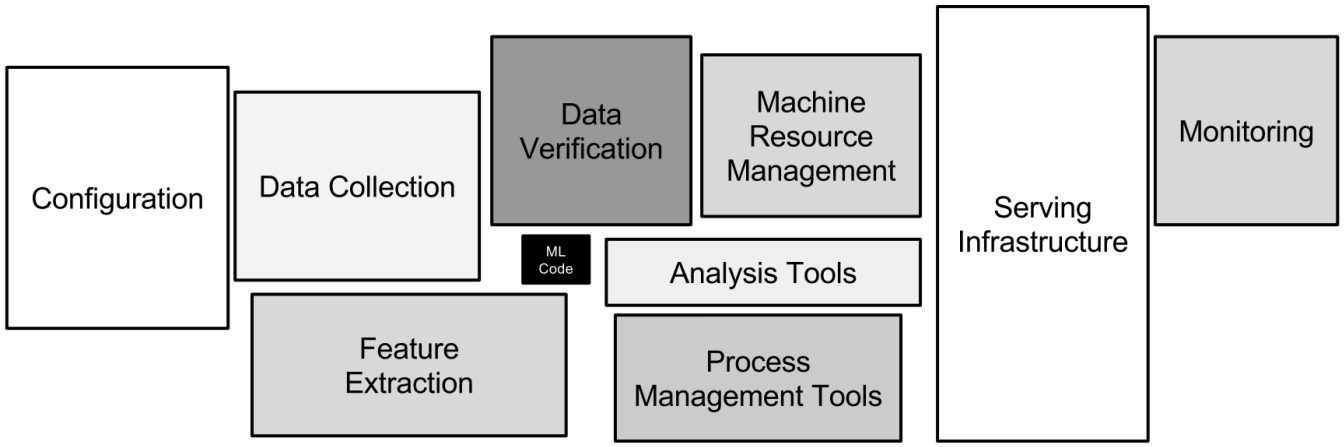
Data engineers sometimes get into ML, too!

Data engineer / ML engineer / Data scientist - ???

Skills of a Data Engineer

- Knows traditional DBs and SQL well
- Applies data visualization
- Has a basic understanding of statistics
- Has a good idea (if not more) about ML
- Can write distributable, efficient code
- Wants to automate everything
- Is always security-aware (GDPR, etc)

The hardest part of ML is not ML! (https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf)



What do you bring to the table already?

- Python?
- Tensorflow?
- TF 2.0 alpha?
- Lua, R, Julia, Torch, etc?
- Big data?
- Machine Learning?
- Deep Learning?

Our project: ”home in time”

Predicting flight delays

https://github.com/Project-Ellie/home-in-time (https://github.com/Project-Ellie/home-in-time)

We discuss the project and stray away into different topics.

Hardly any subject is in-depth.

Theoretical background (if any) through references.

More in-depth material in additional Jupyter notebooks.

Flight data from Atlanta

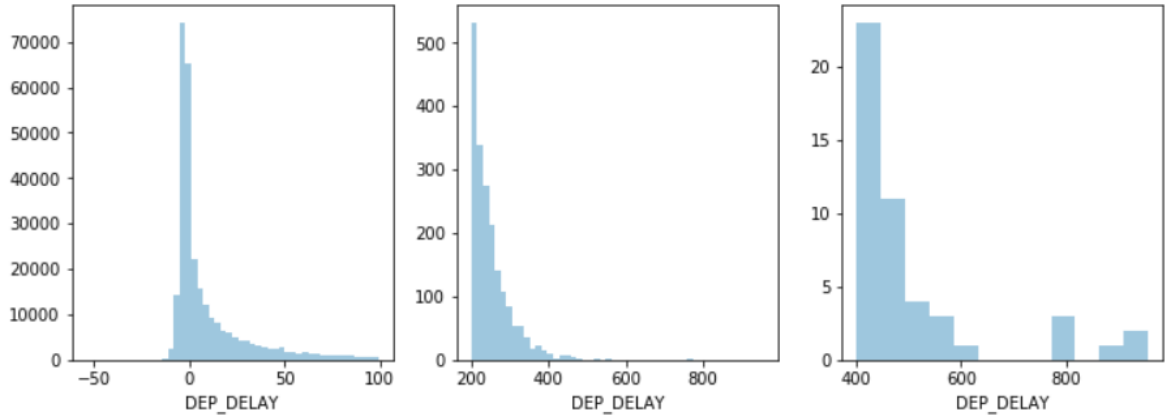
	DATE	AIRLINE	DEP_T	DEP_HOD	DEP	ARR_T	WND_SPD_DEP	DEP_LAT	DEP_LON	ARR_LAT	ARR_LON	MEAN_TEMP_ARR
0	2002-06-01	US	610	6.0	ATL	712	6.9	33.63	-84.42	35.21	-80.94	78.3
1	2002-06-01	DL	620	6.0	ATL	738	6.9	33.63	-84.42	27.97	-82.53	79.1
2	2002-06-01	DL	620	6.0	ATL	740	6.9	33.63	-84.42	28.42	-81.30	77.4
3	2002-06-01	DL	620	6.0	ATL	749	6.9	33.63	-84.42	36.89	-76.20	80.9
4	2002-06-01	UA	627	6.0	ATL	810	6.9	33.63	-84.42	38.94	-77.46	77.7
5	2002-06-01	DL	630	6.0	ATL	836	6.9	33.63	-84.42	40.77	-73.87	76.3
6	2002-06-01	DL	630	6.0	ATL	735	6.9	33.63	-84.42	32.89	-97.03	78.4
7	2002-06-01	DL	635	6.0	ATL	841	6.9	33.63	-84.42	40.69	-74.16	75.9
8	2002-06-01	DL	635	6.0	ATL	749	6.9	33.63	-84.42	35.87	-78.78	79.1
9	2002-06-01	DL	640	6.0	ATL	734	6.9	33.63	-84.42	34.89	-82.21	78.3

Predict flight delays - Really?

Flight delays are - unfortunately - unpredictable.

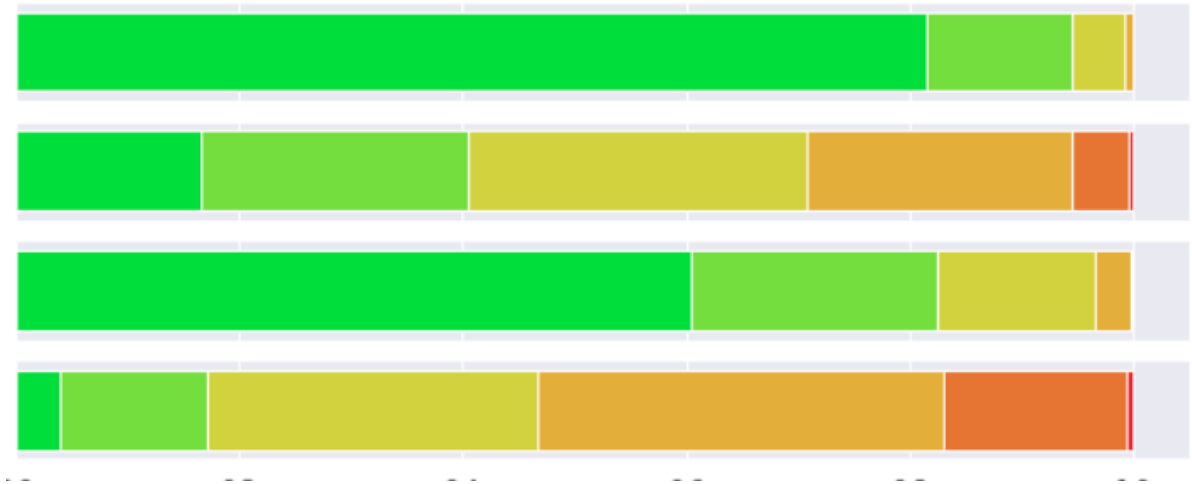
But still there are patterns: Wheather, airline reliability...

But flight delays have a fat tail:



Predict flight delays - Really?

"Smart" prediction: display the probability distribution. See *collateral* (https://github.com/smurve/HSR2019/blob/master/collateral/Fat_Tails.ipynb)



Data Exploration

- Play with billions of records?
- We need a fast analytical database.
- At any scale.
- We need SQL, still!
- Only a world-class cloud allows for (almost arbitrary) up-scaling.

Analytical Databases

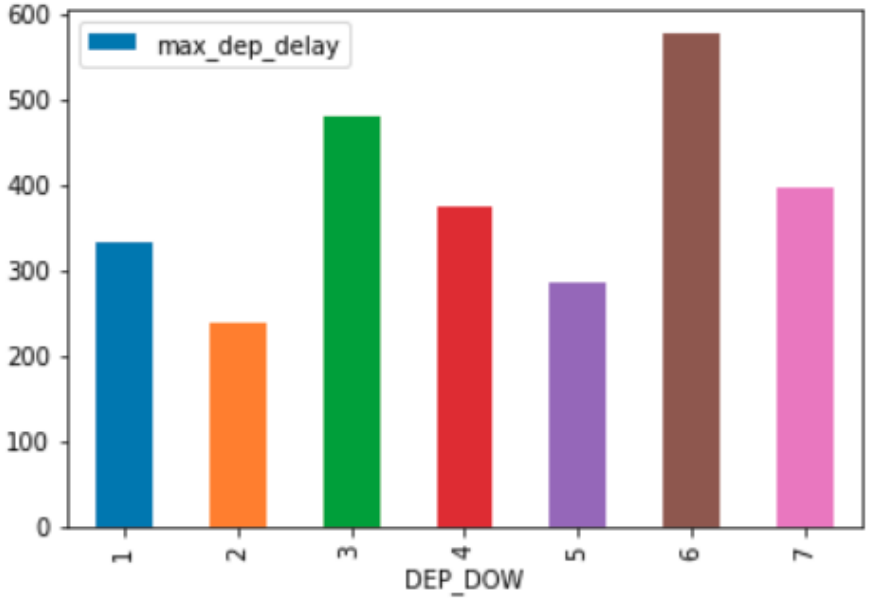
- Amazon Redshift
- Google BigQuery
- Azure Cosmos DB

Architecture:

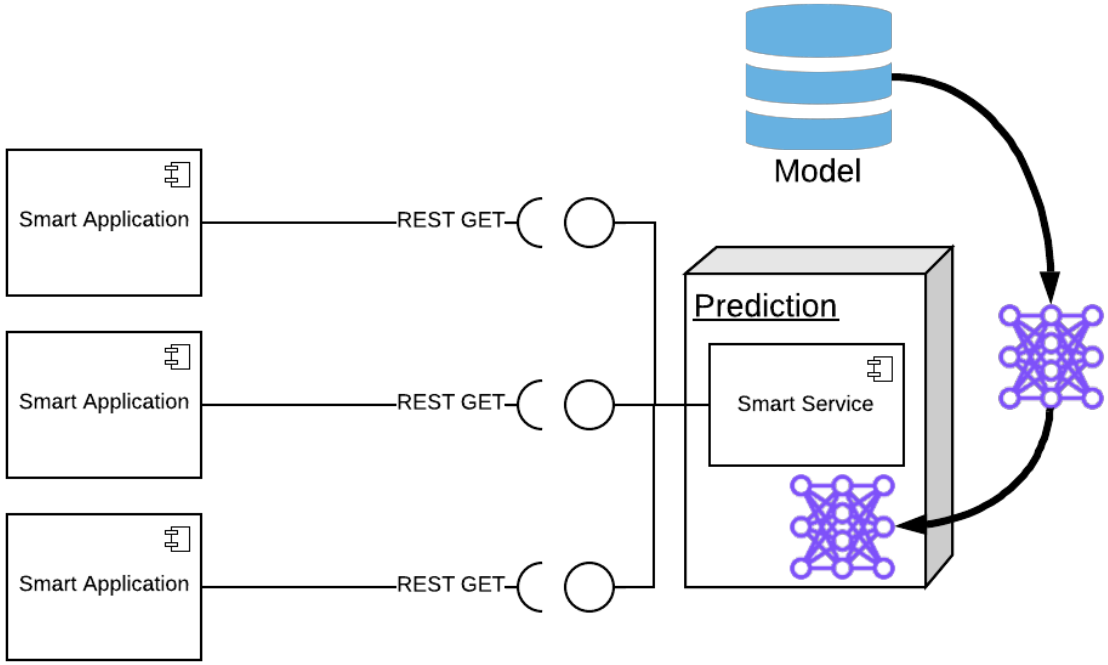
- Multi-core/distributed query execution
- Append-only
- Weaker consistency guarantees

Exploring Flight data (home-in-time)

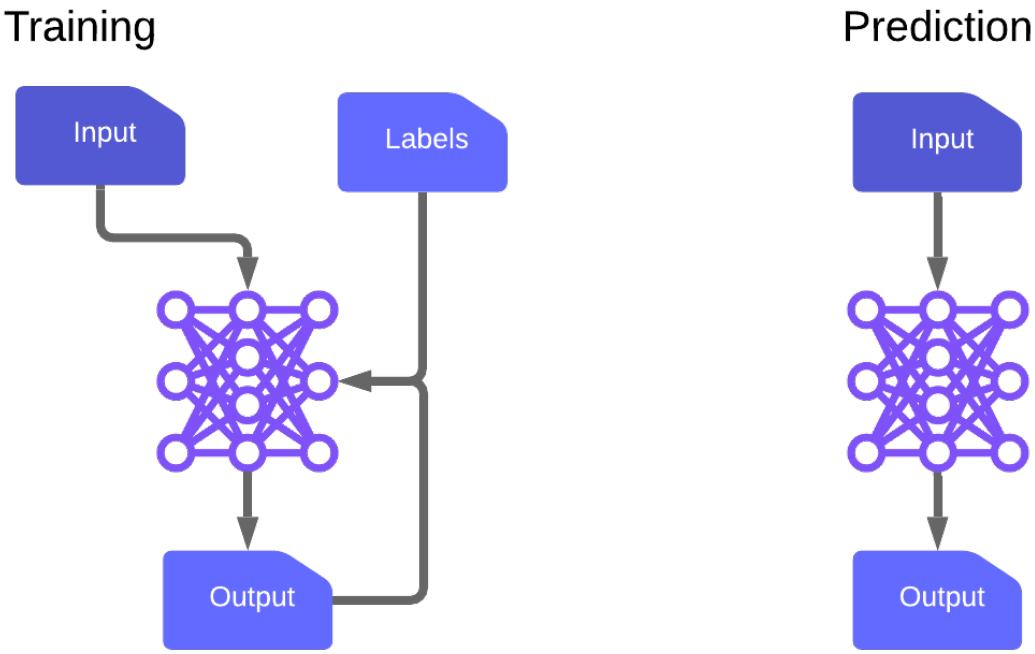
00_Data_Exploration (https://github.com/Project-Ellie/home-in-time/blob/master/00_Data_Exploration.ipynb)



Deployment Architecture



Training and Prediction

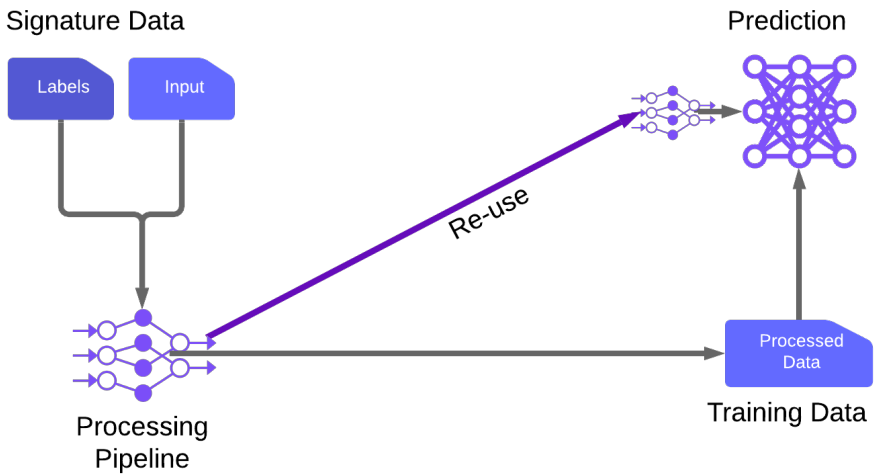


Training data

- Some models require millions or even billions of training records
- Training data needs to be
  - collected
  - cleansed
  - re-formatted
  - aggregated
  - preprocessed
  - combined from different sources

Signature and Training Stage

- Reproduce all pre-processing steps during prediction!
- Failure leads to "training-serving skew"



Fast Data Processing with Beam Pipelines

- Apache Beam is a de-facto standard
- Supports real-time and batch processing with the same code.
- Programming model: directed acyclic graphs
- Test execution local on any machine
- production-scale parallel execution on a cluster
- Map/Reduce/Shuffle automatically optimized

Programming a pipeline

In [ ]:

```
with beam.Pipeline('DirectRunner', PipelineOptions()) as p:

    csv_encoder = tft.coders.CsvCoder(ORDERED_TRAINING_COLUMNS, TRAINING_METADA
TA.schema)

    _ = (p
         | 'read_from_csv' >> beam.io.ReadFromText(
             file_pattern='../testdata/test.csv', coder=csv_encoder)

         | 'write_to_csv' >> beam.io.WriteToText(
             file_path_prefix='./out.csv', coder=csv_encoder)
        )
```

A Production Beam Pipeline in action

Fodder for the Model

See: Input Functions (https://github.com/Project-Ellie/home-in-time/blob/master/03_Input_Functions.ipynb)

- Process any number of files
- Create a continuous stream of decoded records
- Repeat the data stream (epochs)
- Shuffle the data to stabilize learning
- split the data in efficient batch sizes
- automatically iterate over those batches
- prefetch data, use multiple threads in parallel
- distribute data stream if possible.

Tensorflow

Fundamental concepts: Directed Graphs and Sessions

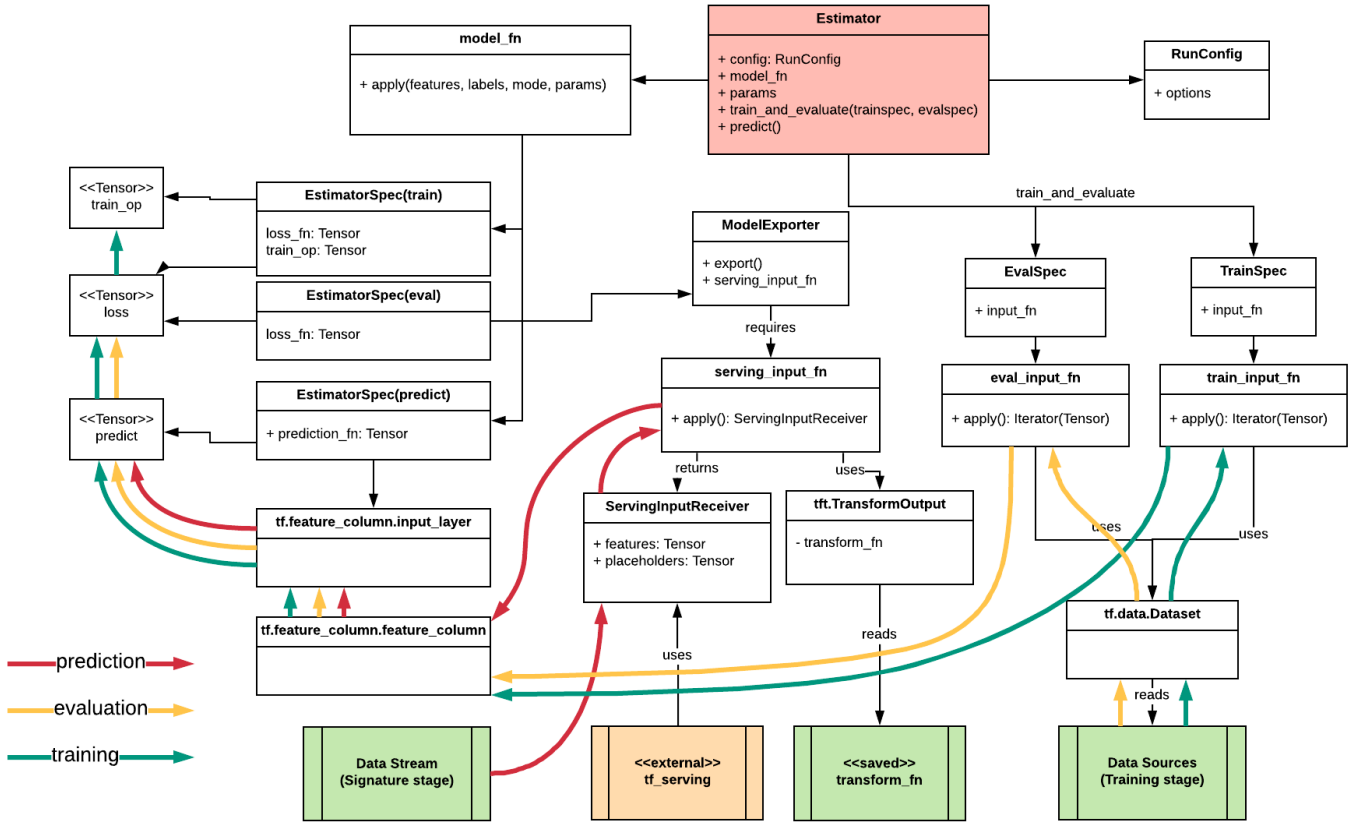
Hardware abstraction and optimal use of GPU/TPU resources

Distributable without code chance

Fully-featured DL Library

We'll learn to use Tensorflow in the exercises

Tensorflow: Programming model and data flow



Exercises

Tensorflow introduction (https://github.com/smurve/HSR2019/blob/master/exercises/TF_Introduction.ipynb)