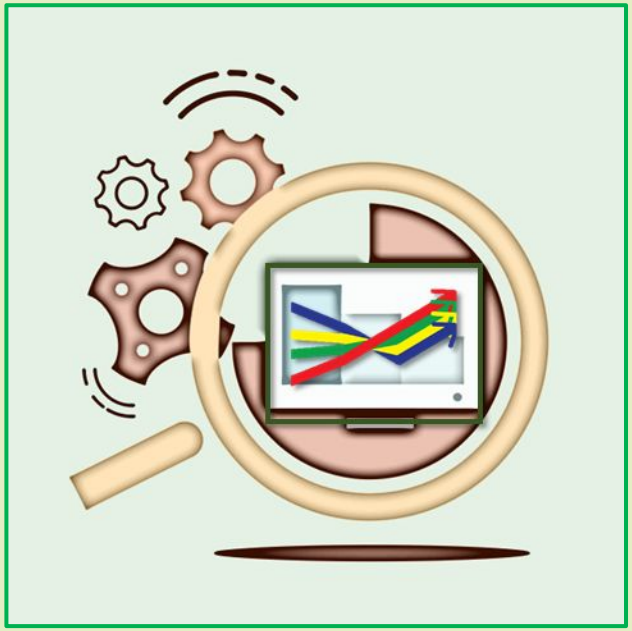
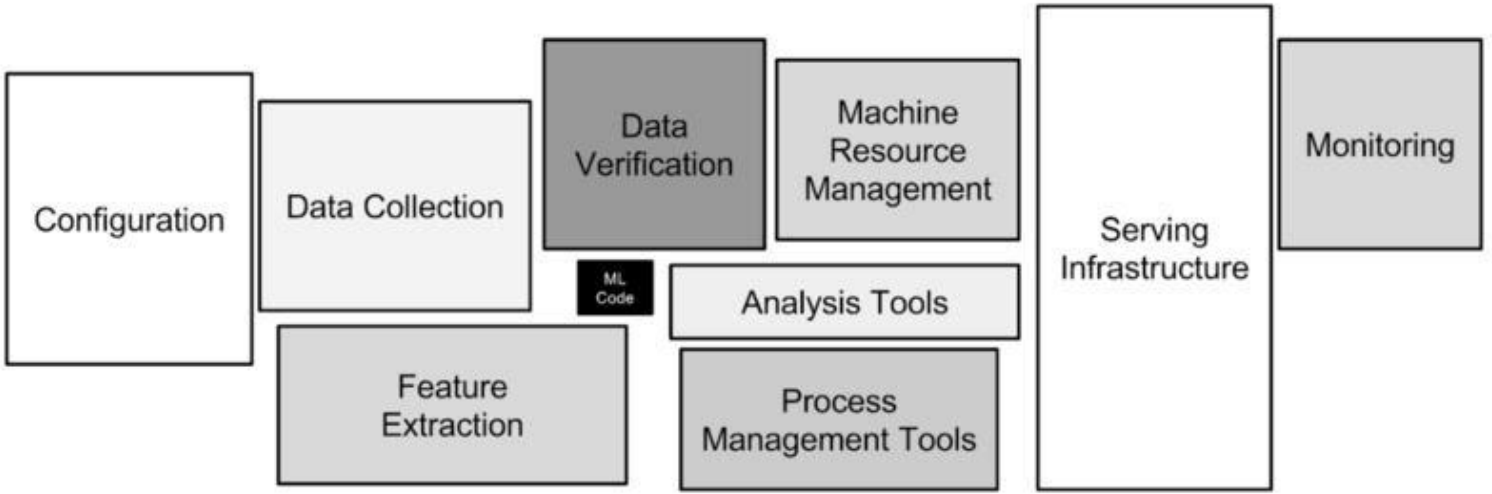
Train In Data



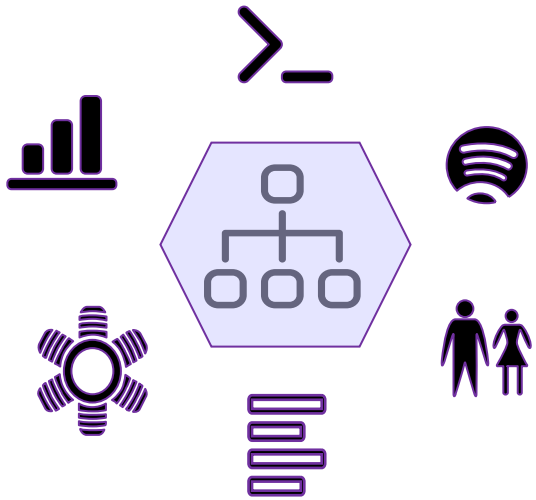
Challenges of ML Systems

ML Systems are Complex



Sculley et al. *(2014)*
https://papers.nips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

Challenges



- The need for reproducibility (versioning everywhere)

Data Dependencies

Models may be trained on data from many different sources

e.g. a house price prediction model which takes data from:

- An in-house SQL database with information on recent inquiries
- A second in-house NoSQL data store which contains historical house listings
- An external API with the latest crime statistics
- A base-line of features CSV prepared by a data scientist and updated on a weekly basis

Configuration issues

Model hyperparameters, versions, requirements, data sources can all be changed and modified via config.

e.g. a yaml file in your source code. Is this tested?

```yaml
# set train/test split
test_size: 0.1

# to set the random seed
random_state: 0

# The number of boosting stages to perform
n_estimators: 50

# the minimum frequency a label should have to be considered frequent
# and not be removed.
rare_label_tol: 0.01

# the minimum number of categories a variable should have in order for
# the encoder to find frequent labels
rare_label_n_categories: 5

# loss function to be optimized
loss: ls
allowed_loss_functions:
  - ls
  - huber
```

Data and Feature Preparation

The steps required to prepare data and transform it into features for the model may be complex.

e.g. a typical pipeline requires us to:

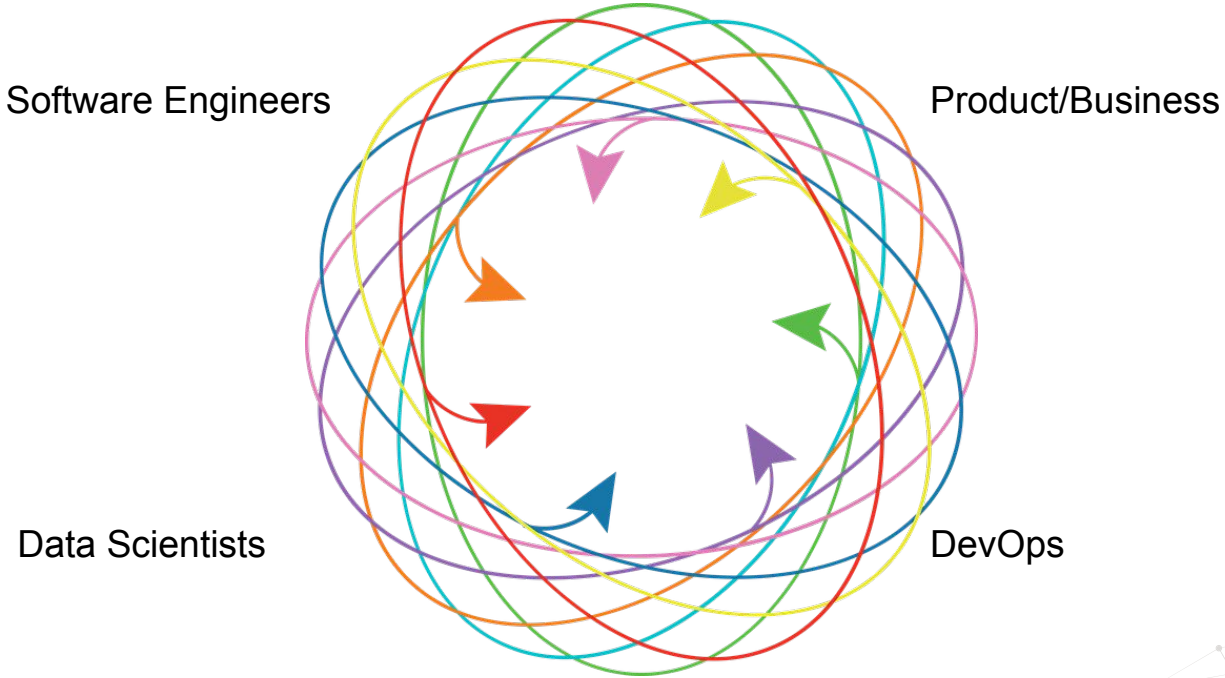
- Transform numerical data
- Transform categorical data
- Handle outliers
- Derive features from raw data
- Many other tasks

Detecting Model Errors

Traditional tests often do not detect errors in ML systems

When you deploy a model which performs worse, no exceptions are raised. Your API will not return any 500 status codes. Standard tests will not catch these sorts of mistakes.

ML System Contributors



Research vs. Production Environments

	Research	Production
Separate from customer facing software	✓	x
Reproducibility matters	Sometimes	Almost always
Scaling challenges	x	✓
Can be taken offline	✓	x
Infrastructure planning required	Sometimes	Almost always
Difficult to run experiments	x	✓