H2O Training

1. Introduction to ML
2. Introduction to H2O
3. H2O Data Munging
4. Machine Learning Concepts
5. H2O Algorithms
6. Productionalizing Models

Operationalizing Machine Learning:

PRODUCTIONALIZING MODELS

Productionalizing Models

- Goal:
  - Move from prototype to production

- Roadblock:
  - Prototyping Environment Cages Your:
    - Feature preprocessing
    - Models
    - Ideas

Case Study: Lending Club Dataset

- Loan data from 2007 up until 2015 including rejected applications and accepted applications.
- Of the 500k accepted applicants about 160k loans have either been completely paid off or defaulted.
- There are about 4 million applicants in the rejected loans dataset.
- **Use Case 1**: Predict the likelihood of a user defaulting based on the information supplied when applying for a loan.
- **Use Case 2**: Determine the interest rate Lending Club would have offered the user based on the information supplied when applying for a loan.
- Full Data: https://www.kaggle.com/wendykan/lending-club-loan-data
- H2O Subset: https://s3.amazonaws.com/h2o-public-test-data/bigdata/laptop/lending-club/loan.csv

H2O Generated POJO Model WebApp Example

https://github.com/h2oai/app-consumer-loan

H2O generated POJO model WebApp Example

This example shows a generated Java POJO being called using a REST API from a JavaScript Web application.

The application simulates the experience of a consumer applying for a loan. The consumer provides some information about themselves and is either offered a loan or denied.

H2O World 2015 Presentation

The "Building a Smarter Application" presentation given at H2O World 2015 references this repo.

- https://github.com/h2oai/h2o-world-2015-training/tree/master/tutorials/building-a-smarter-application

Pieces at work

Processes

(Front-end)

1. Web browser

(Back-end)

1. Jetty servlet container

> Note: Not to be confused with the H2O embedded web port (default 54321) which is also powered by Jetty.

Case Study: Lending Club Dataset

	Column Name	Description	Unit	Role
1	loan_amnt	Requested loan amount	US dollars	Predictor
2	term	Longest term length	Months	Predictor
3	**int_term**	**Recommended interest rate**	**Rate**	**Response**
4	emp_length	Employment length	Years	Predictor
5	home_ownership	Housing status	Categorical	Predictor
6	annual_inc	Annual income	US dollars	Predictor
7	purpose	Purpose for the loan	Categorical	Predictor
8	addr_state	State of residence	Categorical	Predictor
9	dti	Debt to income ratio	Percent	Predictor
10	delinq_2yrs	Number of delinquencies in the past 2 years	Count	Predictor
11	revol_util	Revolving credit line utilized	Percent	Predictor
12	total_acc	Number of active accounts	Count	Predictor
13	**bad_loan**	**Bad loan indicator**	**Boolean**	**Response**
14	longest_credit_length	Age of oldest active account	Years	Predictor
15	verification_status	Income verification status	Boolean	Predictor

Example Model Output

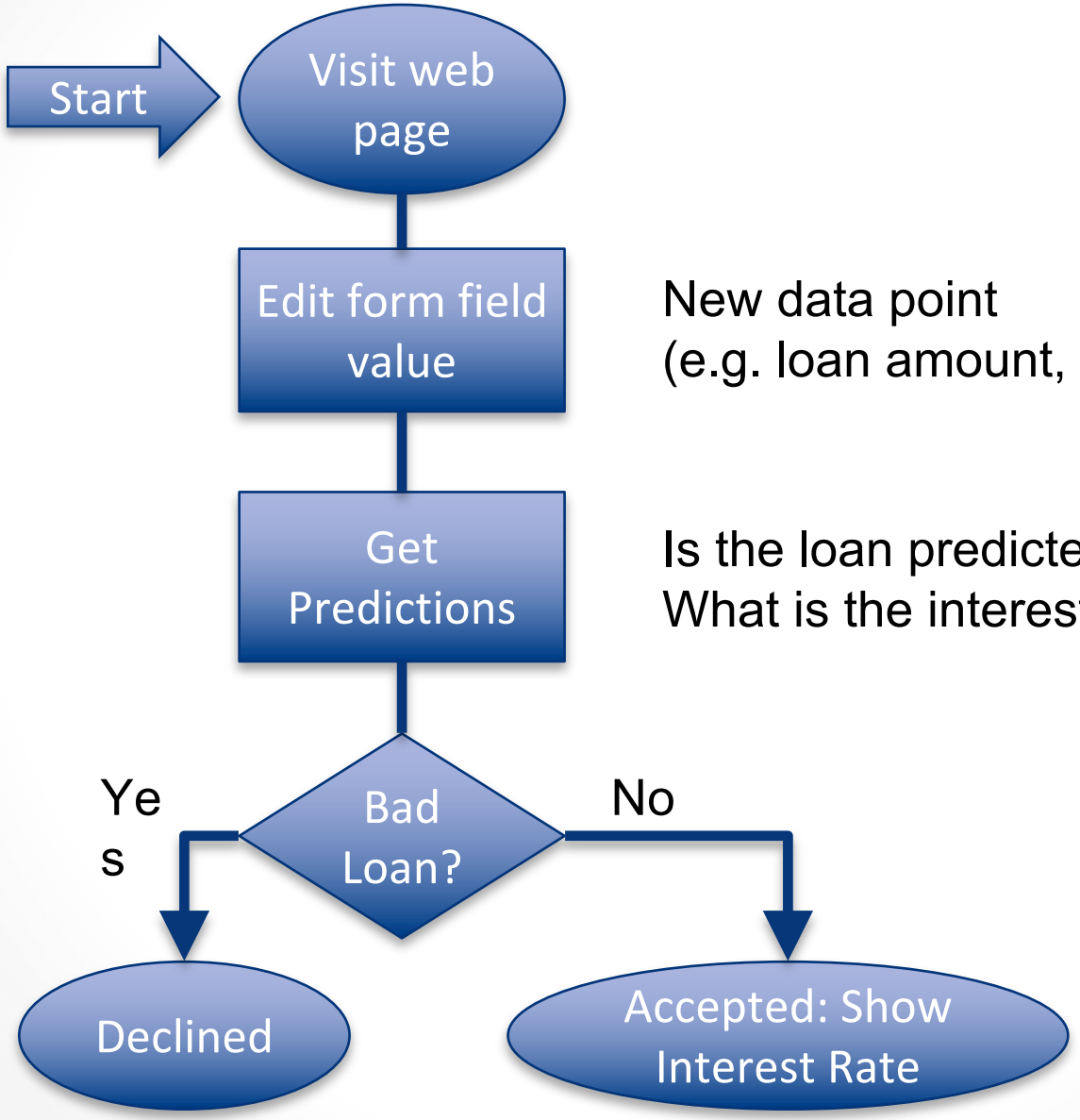
Bad Loan Model

Algorithm:	GBM
Model category:	Binary
	Classification
ntrees:	100
max_depth:	5
learn_rate:	0.05
AUC on valid:	.685
max F1:	0.202

Interest Rate Model

Algorithm:	GBM
Model category:	Regression
ntrees:	100
max_depth:	5
learn_rate:	0.05
MSE:	11.1
R2:	0.424

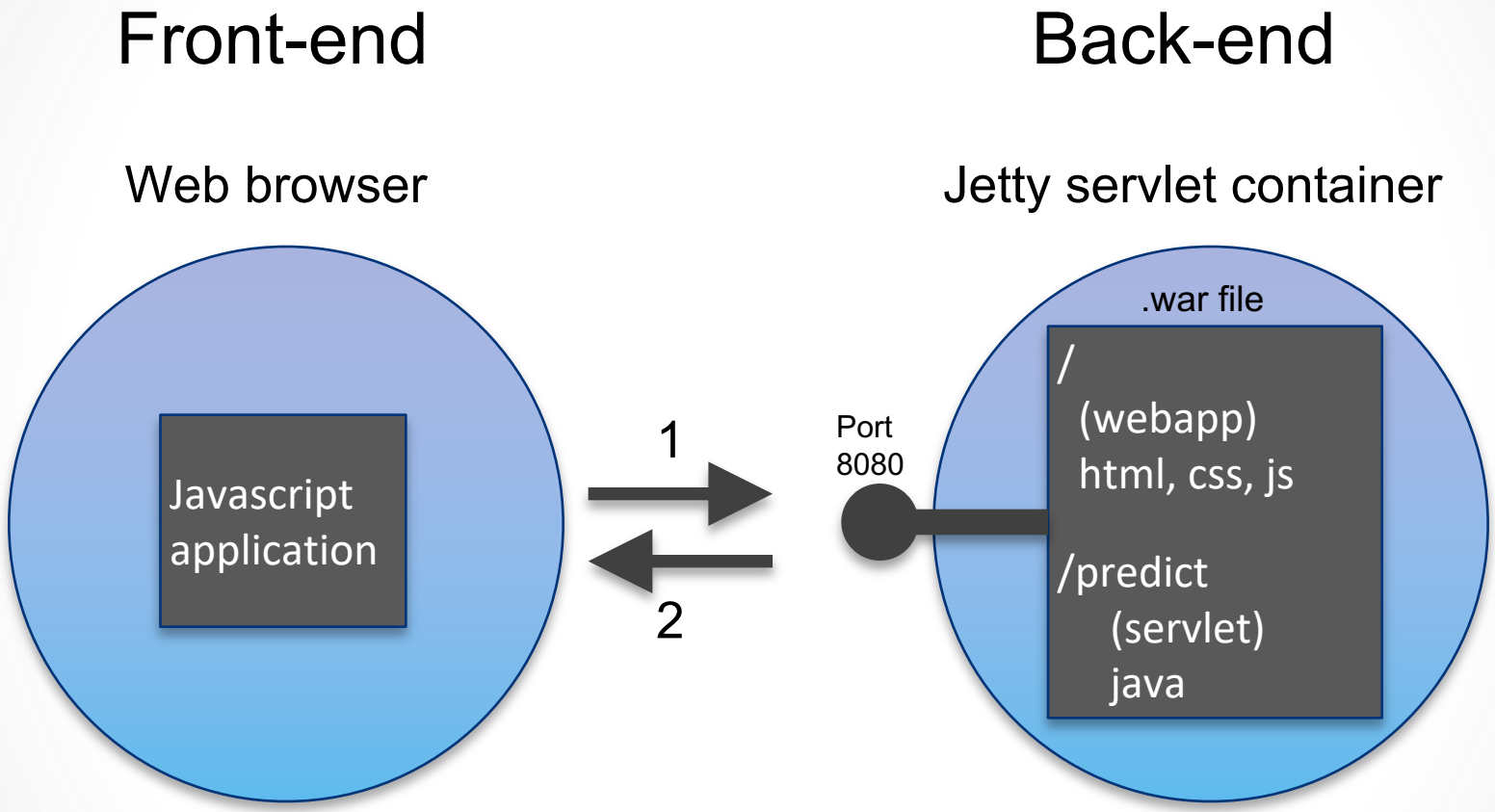
Workflow For This App



New data point
(e.g. loan amount, income)

Is the loan predicted to be bad?
What is the interest rate?

App Architecture Diagram

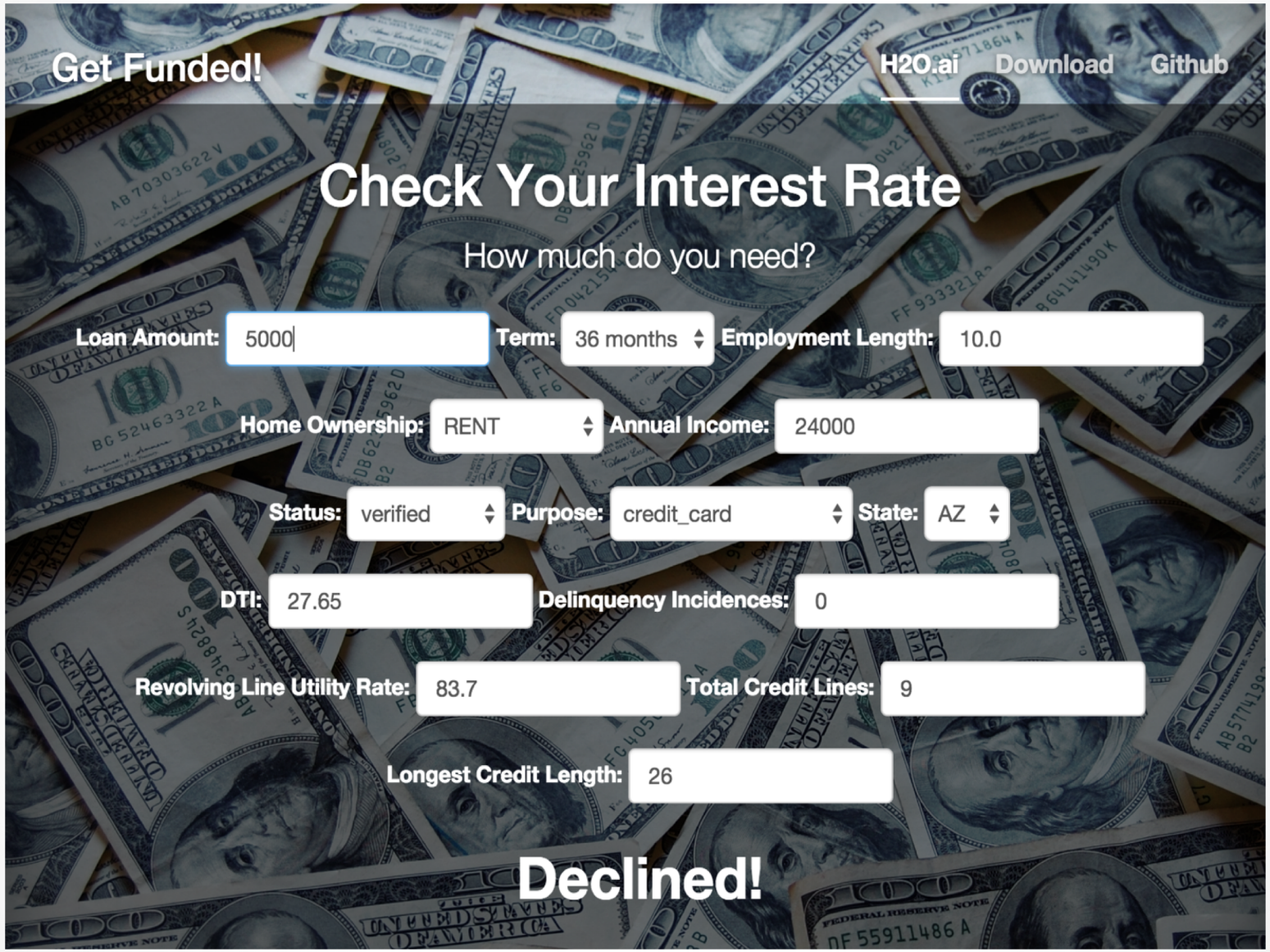


1. HTTP GET with query parameters (loan_amt, annual_inc, etc.)
2. JSON response with predictions

Software Pieces

- Offline
  - R + H2O (model building)
- Online
  - Front-end
    - Web browser
    - JavaScript application (run in the browser)
  - Back-end
    - Jetty servlet container
    - H2O-generated model POJO (hosted by servlet container)

Predict Loan Defaults



Operationalizing Machine Learning:

TIPS ON GETTING STARTED

Balancing Accuracy and Interpretability

- Try different supervised learning techniques
- Train a “black box model” but use it as a benchmark
  - Train explainable model on predictions from the black box
- Use the black box model as backup

- Use variable importance and partial dependency plots for explanation
- Use complicated model for feature creation

https://www.oreilly.com/ideas/predictive-modeling-striking-a-balance-between-accuracy-and-interpretability

Online Resources

- H2O Tutorials and Training Material
  - https://github.com/h2oai/h2o-tutorials
  - https://github.com/h2oai/h2o-tutorials/tree/master/training/lending_club_exercise
  - https://github.com/h2oai/app-consumer-loan
- Datasets
  - https://s3.amazonaws.com/h2o-public-test-data/bigdata/laptop
  - https://s3.amazonaws.com/h2o-public-test-data/bigdata/laptop/lending-club/loan.csv