Open Source Software Support

A Network Tour of Data Science

Matthaios Olma
Pavlos Nikolopoulos
Stefanos Skalistis

Adopting Open Source Software (OSS)

Open source software:

- Usage based on some license
- Code is fully available
- Project lifespan is varying

APACHE Spark

hadoop

cassandra

MESOS

APACHE HBASE

"wowarmorytools"

"Blipstick"

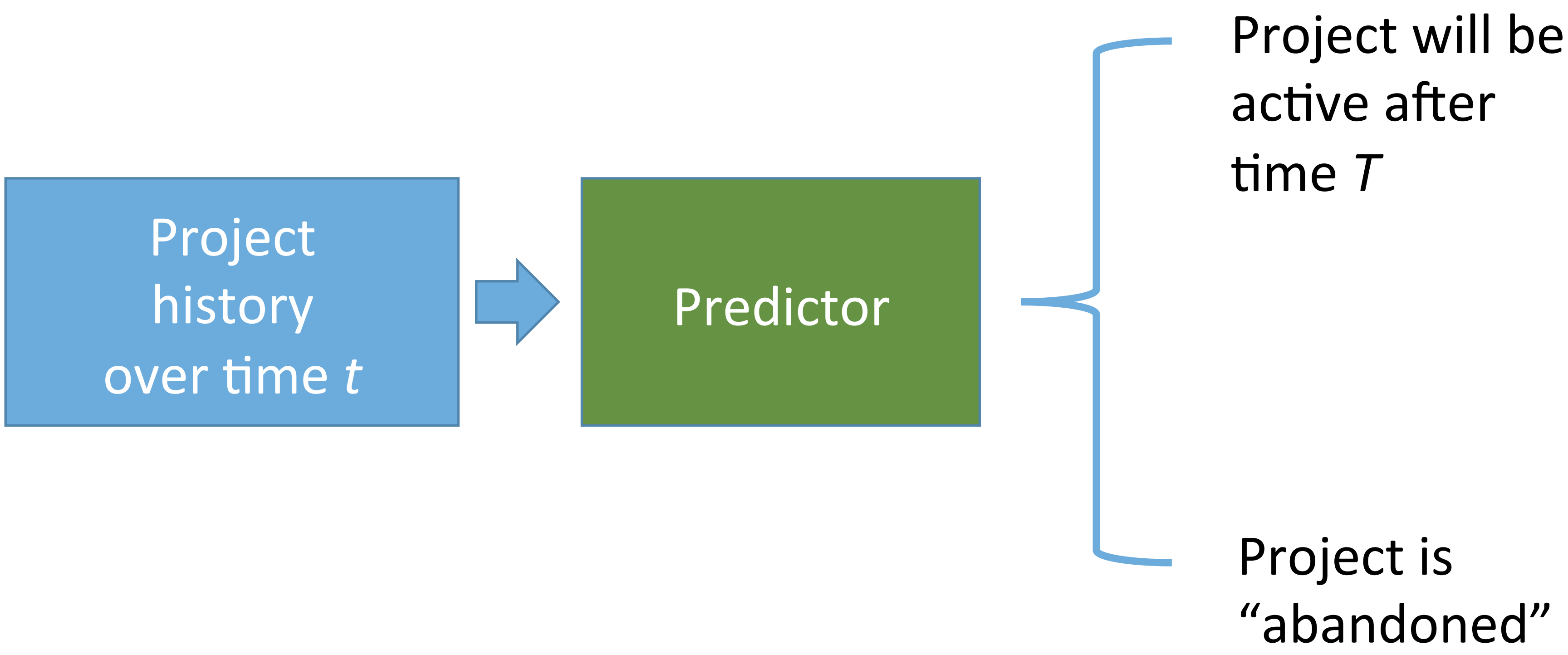
"James"

Should we adopt OSS?

Will it be supported in the future?

Predict OSS project survivability

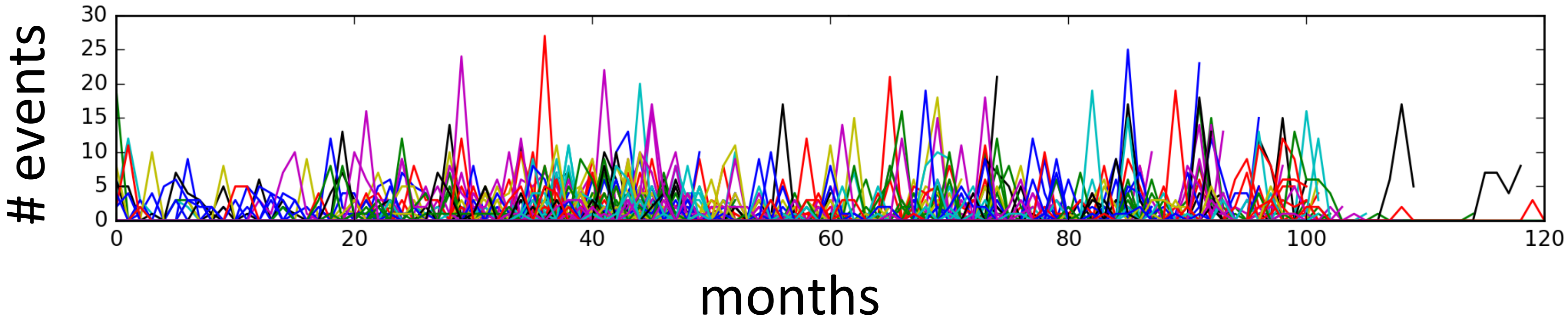
Desirable Goal:
Predict if project will be still active in the future



Survival prediction is challenging

Success factors:
- human dynamics
- project popularity
- usefulness

Diverse data:
- Variety of projects
- Variety of dev. Techniques
  - *Agile/Waterfall*



There is no common activity pattern

Our Process

Acquire OSS project data and generate dataset

↓

Explore data structure

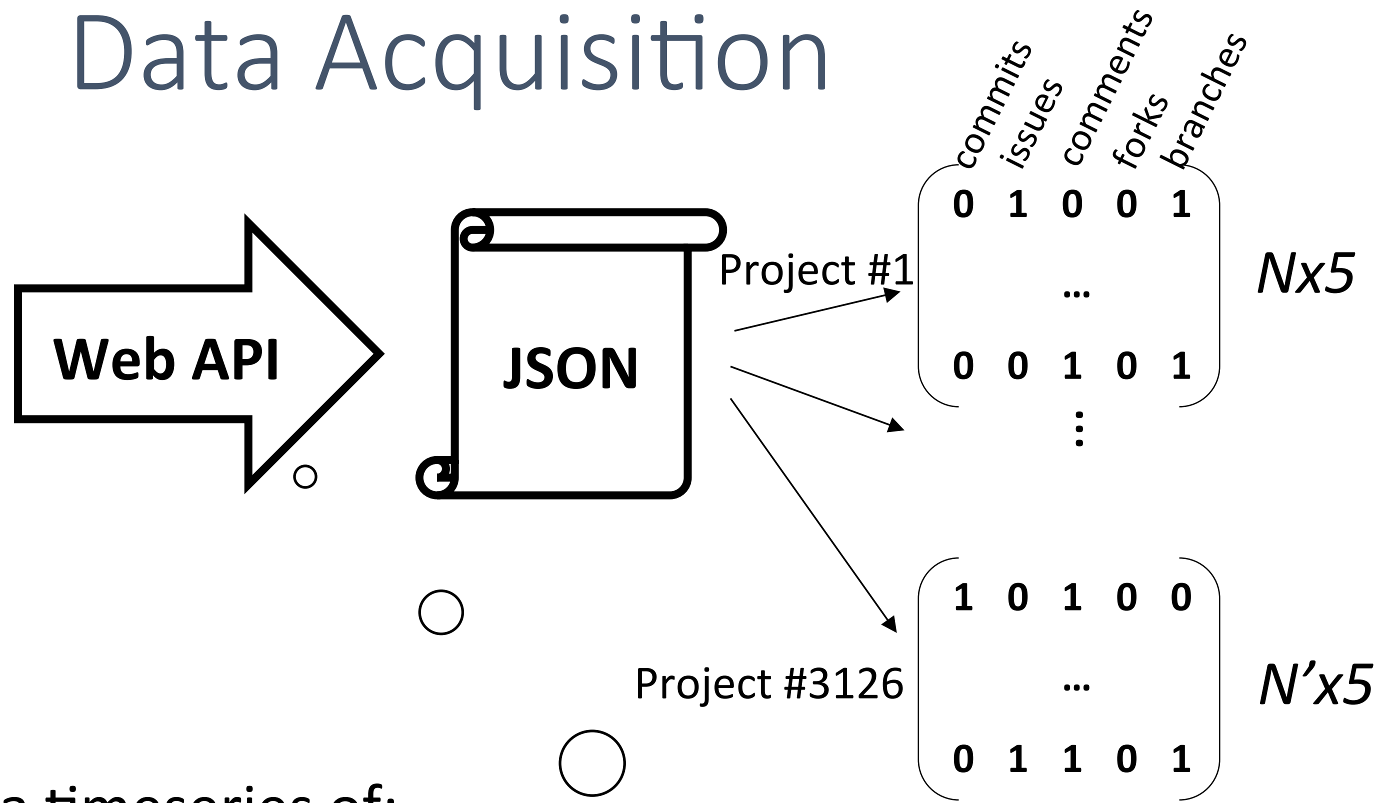
↓

Label data

↓

Create a NN predictor

Data Acquisition



3126 projects
Each project is a timeseries of:

- commits
- issues
- comments
- forks
- branches

- **Slow interface**
- **100 projects/ call**

Dataset generation

1. Eliminate duplicate projects

2. Create monthly aggregates

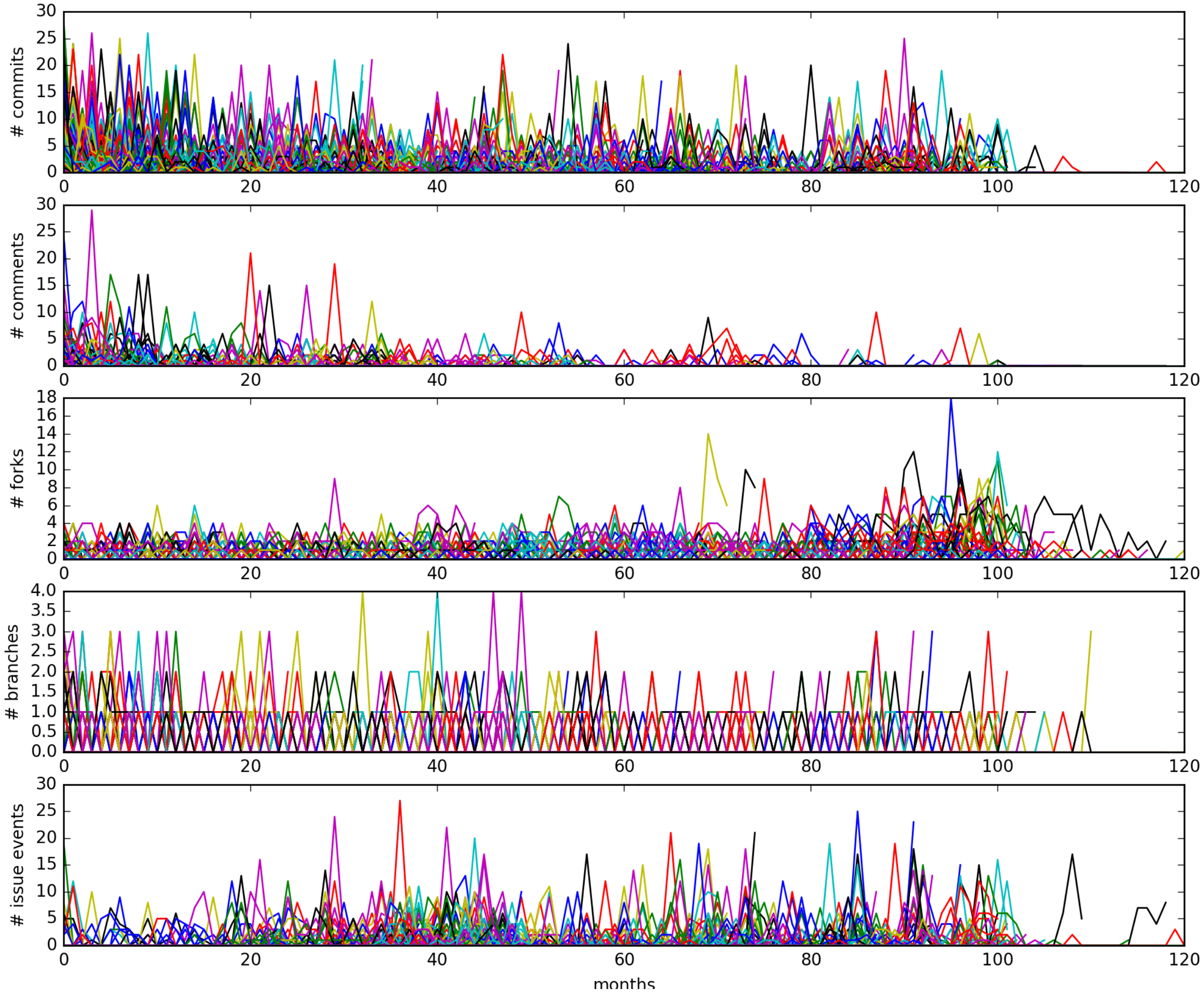
$$\begin{array}{c} \text{Project \#1} \\ \begin{pmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ & & \mathbf{\dots} & & \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \end{pmatrix} \\ \\ \textit{Nx5} \end{array} \Rightarrow \begin{array}{c} \text{Project \#1} \\ \begin{pmatrix} \mathbf{0} & \mathbf{5} & \mathbf{0} & \mathbf{0} & \mathbf{4} \\ & & \mathbf{\dots} & & \\ \mathbf{0} & \mathbf{8} & \mathbf{0} & \mathbf{2} & \mathbf{3} \end{pmatrix} \\ \\ \textit{Mx5} \\ M = N/12 \end{array}$$

3. Make all projects start from time 0
   - Start time is the time of the first event

4. Suppress projects with duration < T = 24 months
   - *We are interested in project's activity after time T*

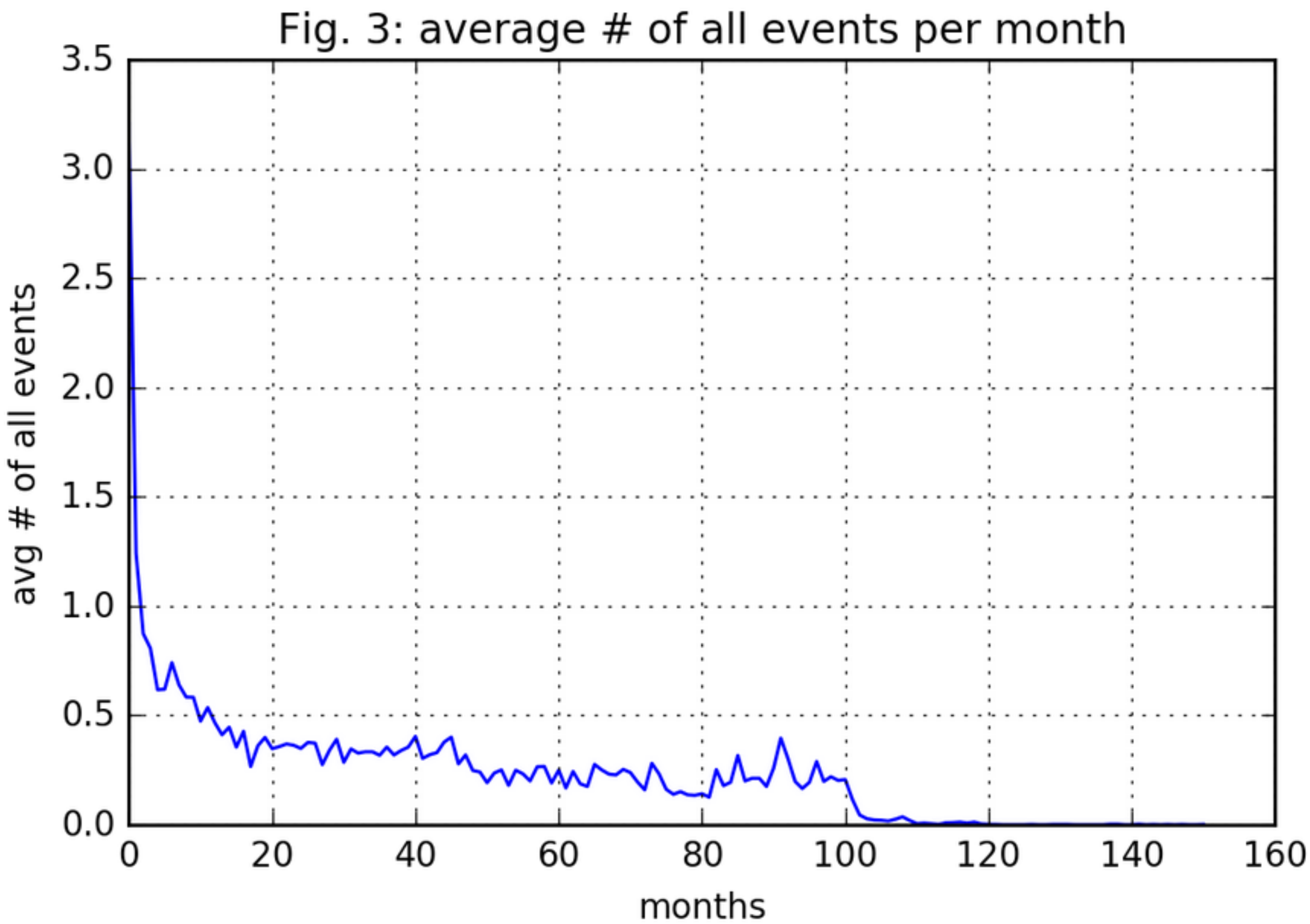
Data Exploration and Visualization

Finally taken into consideration: 834 projects



no common activity pattern

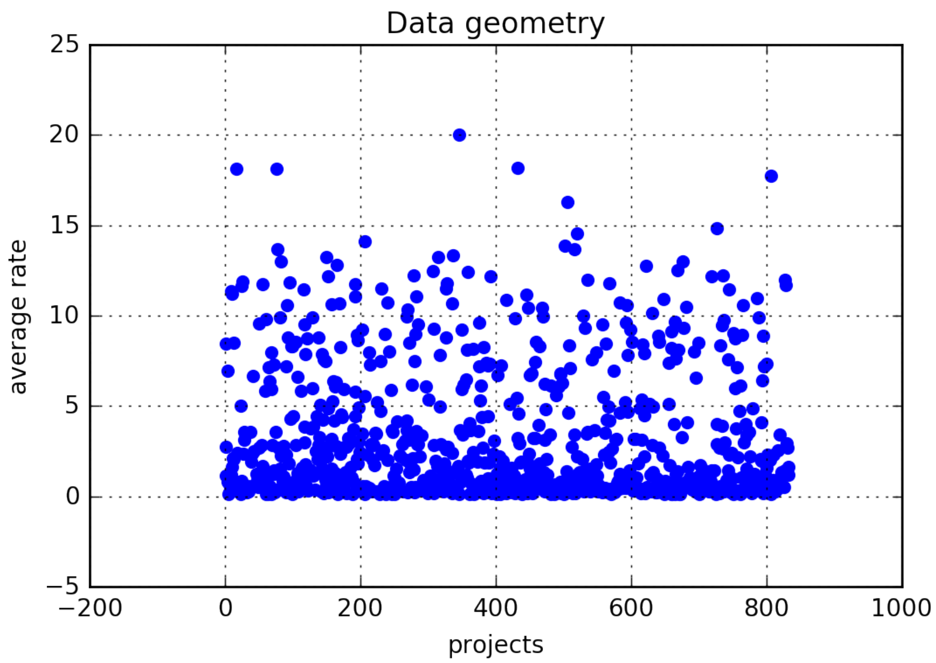
Choosing the prediction period



After 24 months the # events converges

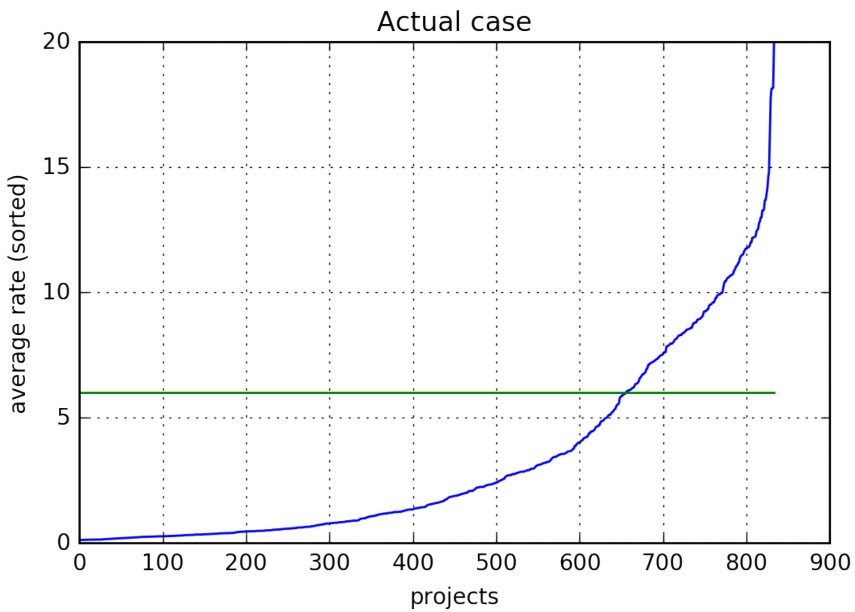
Data Labeling

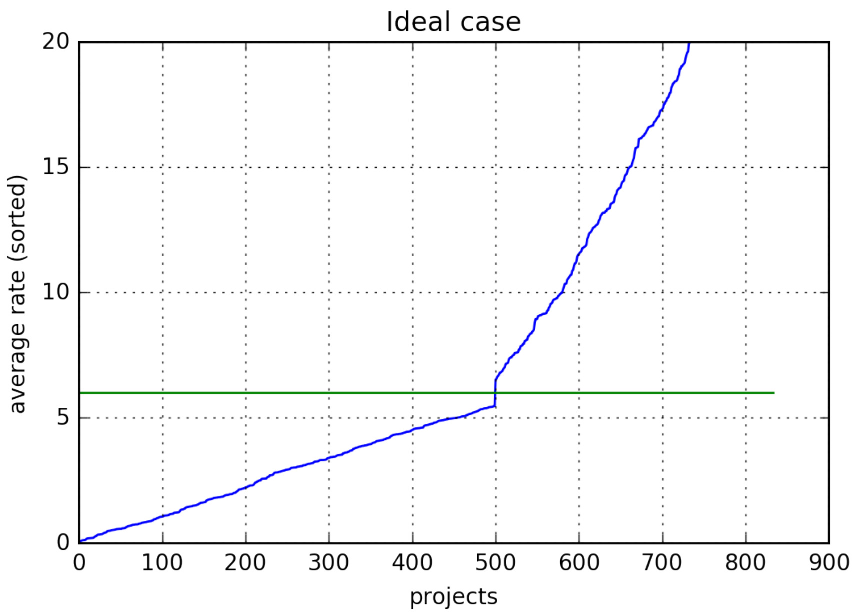
- 2 classes: {Active, Inactive}
- Differentiation metric: Average rate of events (after 24 months)
- Threshold set to 6 events/year
  - meaningful threshold in terms of software usage
  - data is separated in this way



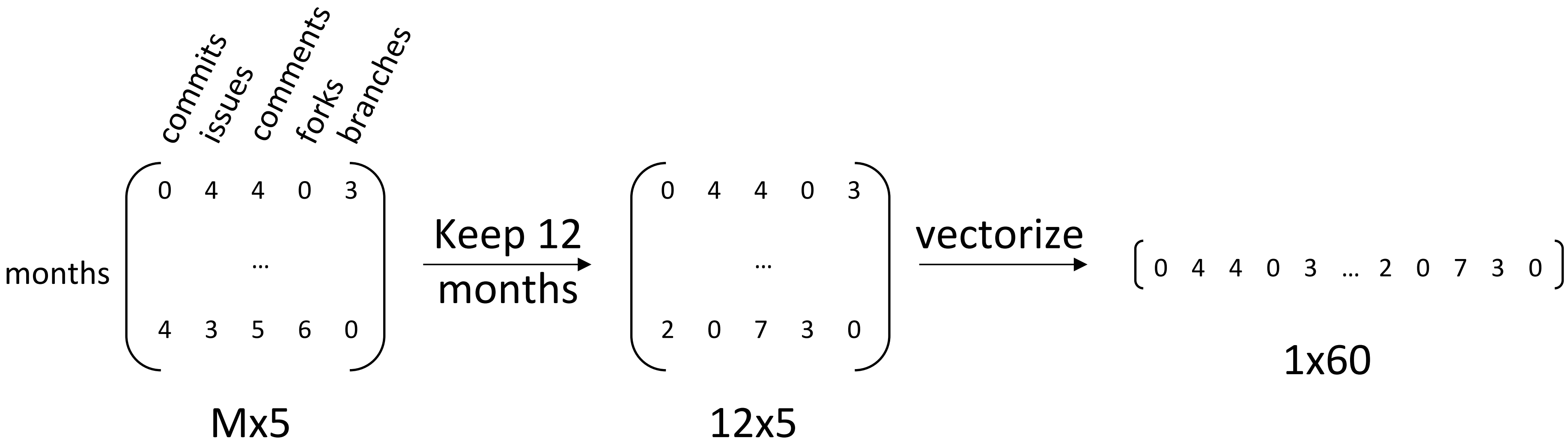
Data Labeling

- 2 classes: {Active, Inactive}
- Differentiation metric: Average rate of events (after 24 months)
- Threshold set to 6 events/year
  - meaningful threshold in terms of software usage
  - data is separated in this way





Creating Train and Test sets



- Training set:
  - metadata from the first 12 months of each project
  - 774 projects randomly chosen at batches of 50
- Test set: 50 projects

1-layer vs 2-layer NN

Simple softmax classifier

$y = softmax(xW+b)$

- Normal initialization (std=0.1)
- No regularization

- Training accuracy: 76%
- Test accuracy: 80%

2-layer NN

$y = softmax(ReLU(xW_1+b_1)W_2+b_2)$

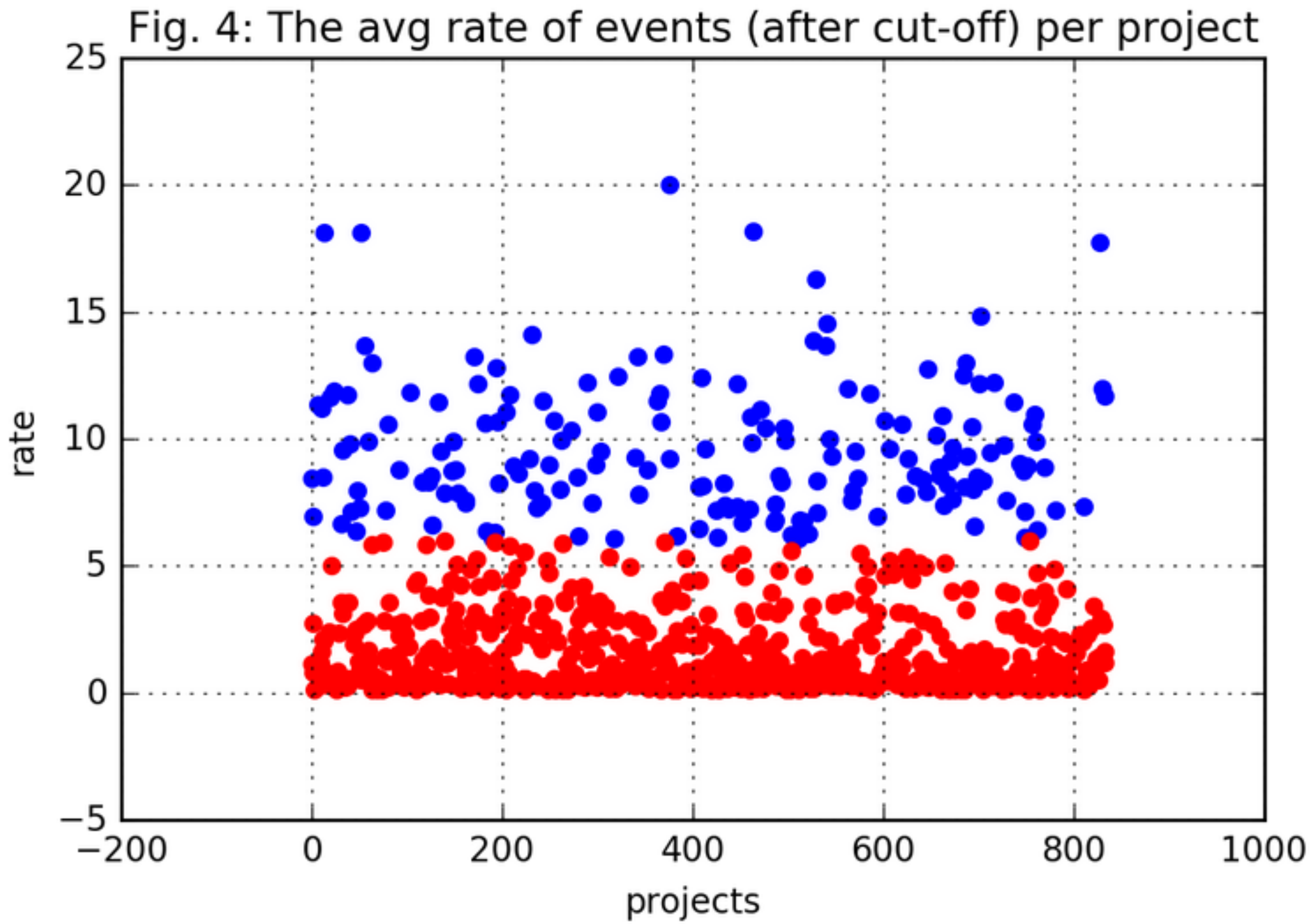
- 100 neurons
- Xavier initialization
- ReLU activation
- L2 regularization

- Training accuracy: 92%
- Test accuracy: 84%

Conclusions

- It is possible to predict a project's activity with high accuracy
  - 12 months of metadata are sufficient
- Marginal improvement by 2-layer NN over simple linear classifier
  - Labeling based on a linear separation of data
  - Training data and label are based on #events

Additional no1



Additional no2