Java’s Place in the AI Revolution

Zoran Sevarac, ***Deep Netts Technologies, Java Champion***

Frank Greco, ***NYJavaSIG, Java Champion***

Pratik Patel, ***Friends of OpenJDK (Foojay.io), Java Champion***

Foojay.io Webinar

Takeaways

(yep… before the Giveaways)

Learn AI or your job/career is at risk

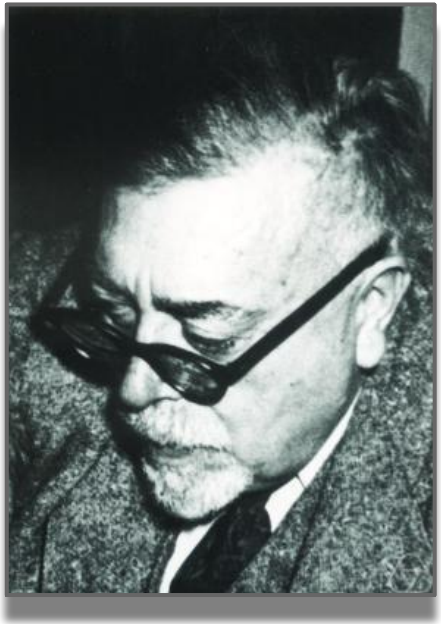
Java is a GREAT production language for AI

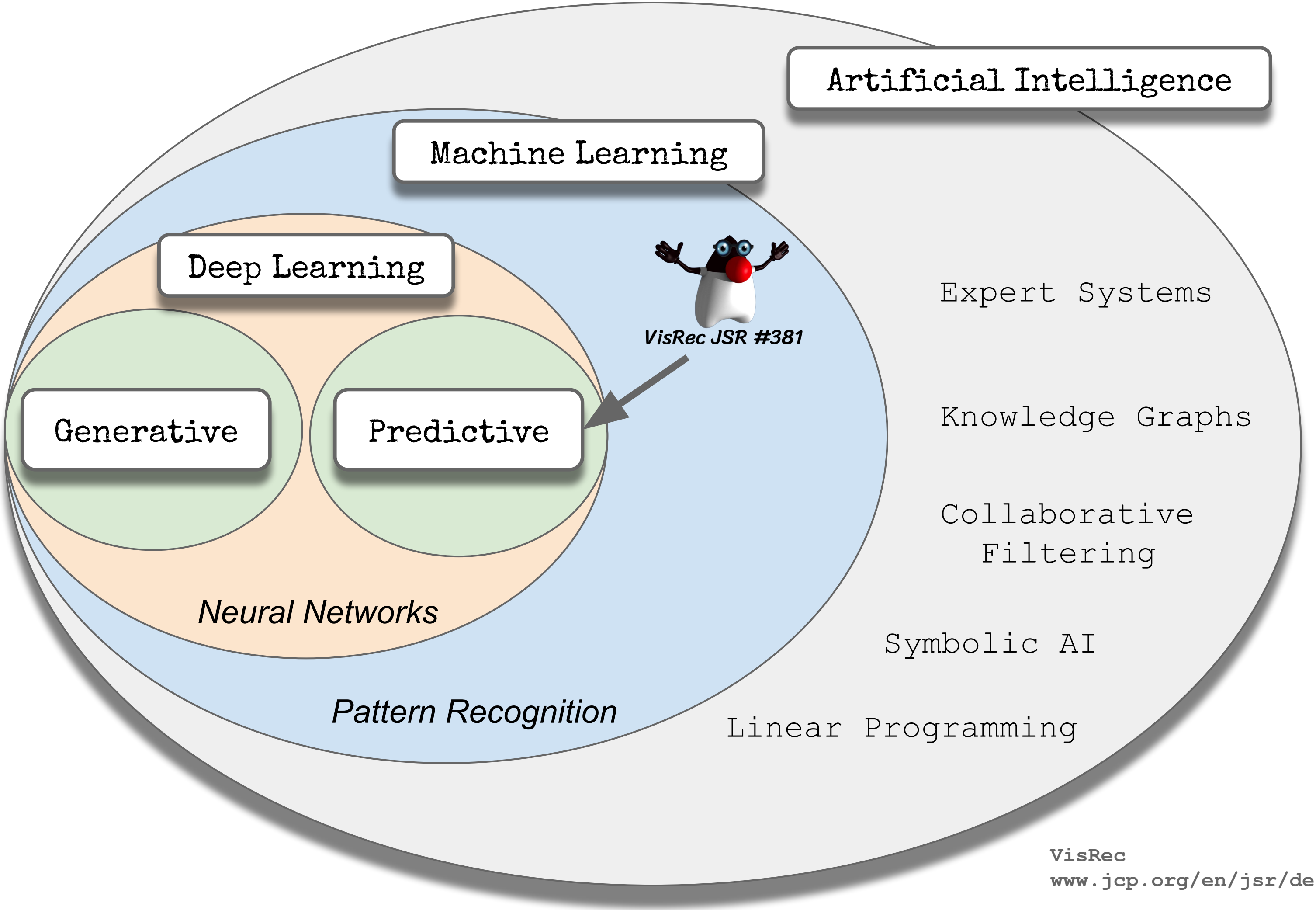
Models are probabilistic

You can use AI as a Java developer's tool
or to develop AI Apps
But learn how it works first…

“One of the most interesting aspects of the world is that it can be considered to be made up of patterns”

Norbert Wiener (1948) - 1894-1964 - MIT





Artificial Intelligence

Machine Learning

Deep Learning

Generative

Predictive

VisRec JSR #381

Neural Networks

Pattern Recognition

Expert Systems

Knowledge Graphs

Collaborative Filtering

Symbolic AI

Linear Programming

VisRec
www.jcp.org/en/jsr/detail?id=381

Predictive AI and Generative AI

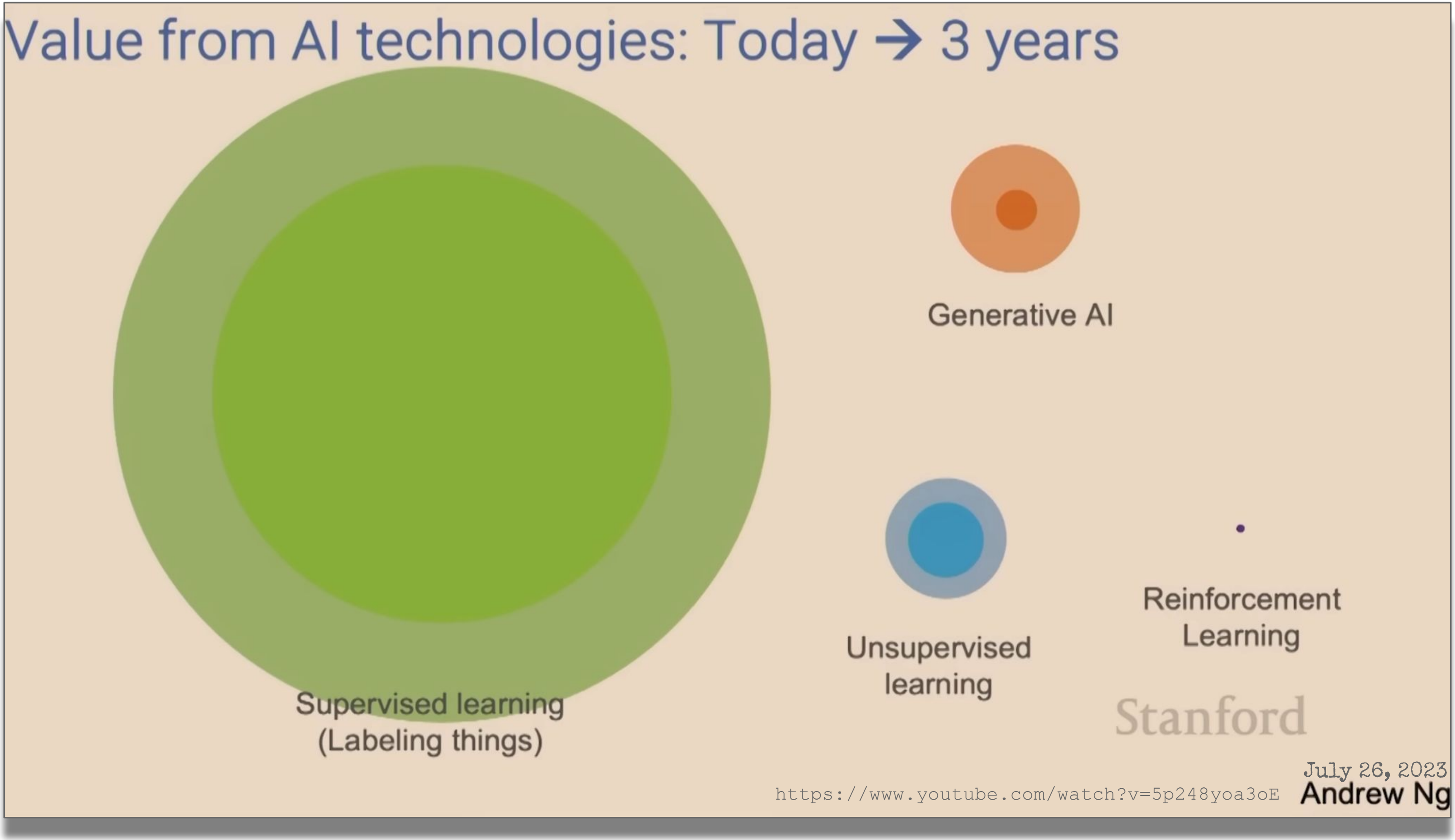
Most of AI $$ value (02/24) comes from PredAI: weather, image detection and classification, financial services, buying behavior, up/cross-selling…

PredAI has been deployed successfully for past 15+ yrs

PredAI is probably worth at least $100B just to Google

GenAI typically used for more “creative”, content generation

GenAI growth and potential is huge. Market value may match PredAI in 3-5 yrs



Value from AI technologies: Today → 3 years

Generative AI

Supervised learning
(Labeling things)

Unsupervised
learning

Reinforcement
Learning

Stanford

July 26, 2023

https://www.youtube.com/watch?v=5p248yoa3oE

Andrew Ng

Predictive AI (PredAI)

forecasting

structured data

Classification
Recommendation systems
Sales Forecasts
Fraud Detection
Predictive maintenance

Generative AI (GenAI)

creation

unstructured data

Creative/Design
Ideation
Prototyping
Simulation
Translation / Summarization

Predictive AI (PredAI)

forecasting

structured data

- weather prediction
- financial markets
- network optimization
- spam control
- custom image detection

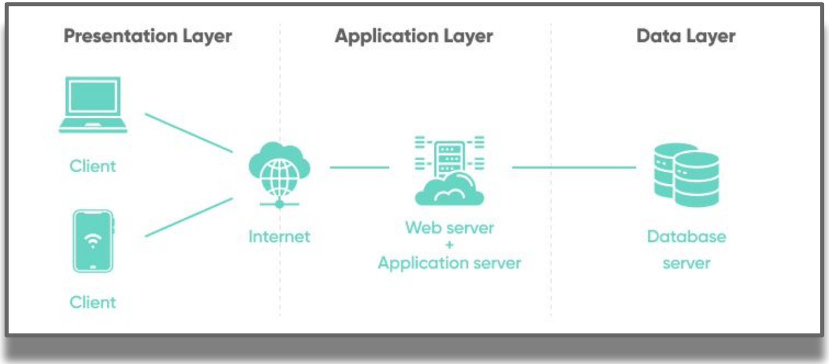
Generative AI (GenAI)

creation

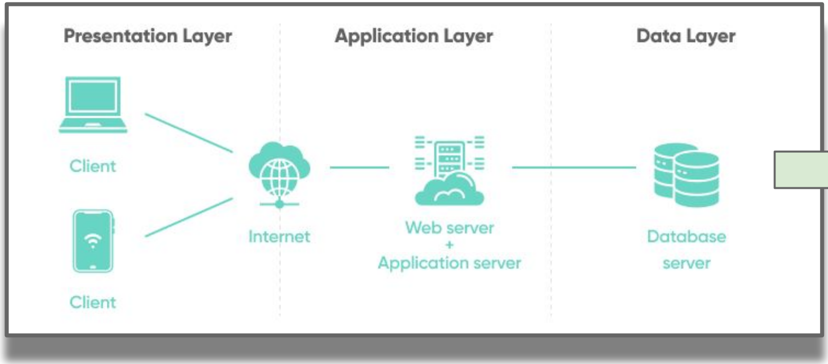
unstructured data

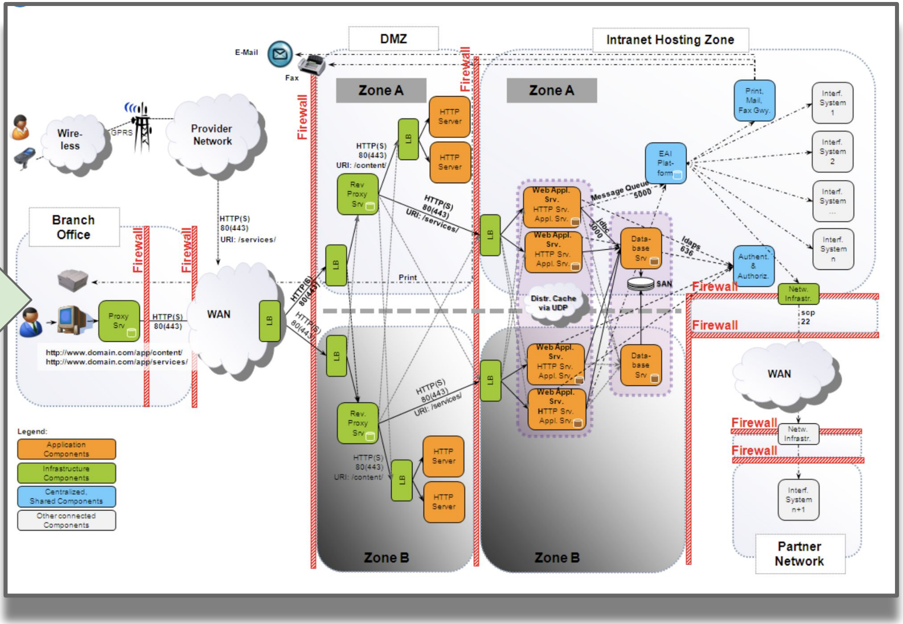
- business ideas
- curriculum design
- article outline
- reduce routine admin
- match user level

Evolving Architectures

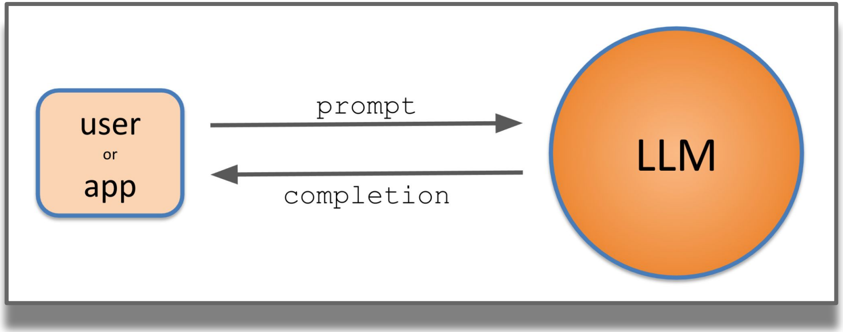

Presentation Layer
Application Layer
Data Layer
Client
Client
Internet
Web server
+
Application server
Database
server

Web

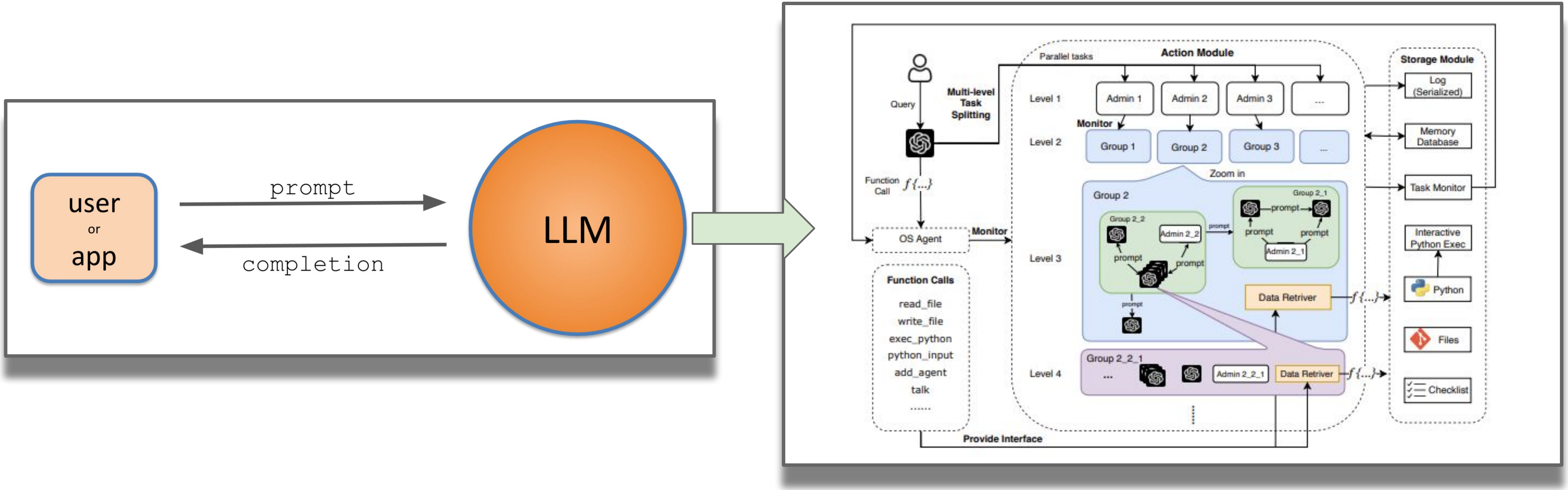

Presentation Layer
Application Layer
Data Layer
Client
Client
Internet
Web server
+
Application server
Database
server


DMZ
Intranet Hosting Zone
E-Mail
Fax
Zone A
Zone A
Firewall
Wire-
less
GPRS
Provider
Network
HTTP
Server
HTTP
Server
LB
HTTP(S)
80(443)
URI: /content/
Rev
Proxy
Srv
HTTP(S)
80(443)
URI: /services/
Print,
Mail,
Fax Gwy.
Interf.
System
1
Interf.
System
2
Interf.
System
...
Interf.
System
n
EAI
Plat-
form
Message Queue
5000
Web Appl.
Srv.
HTTP Srv.
Appl. Srv.
jdbc
3000
Data-
base
Srv
ldaps
636
Authent.
&
Authoriz.
SAN
Branch
Office
HTTP(S)
80(443)
URI: /services/
Proxy
Srv
HTTP(S)
80(443)
WAN
Print
Distr. Cache
via UDP
Netw.
Infrastr.
scp
22
http://www.domain.com/app/content/
http://www.domain.com/app/services/
WAN
Legend:
Application
Components
Infrastructure
Components
Centralized,
Shared Components
Other connected
Components
Rev.
Proxy
Srv
HTTP(S)
80(443)
URI: /content/
HTTP(S)
80(443)
URI: /services/
Netw.
Infrastr.
Interf.
System
n+1
Zone B
Zone B
Partner
Network

Web



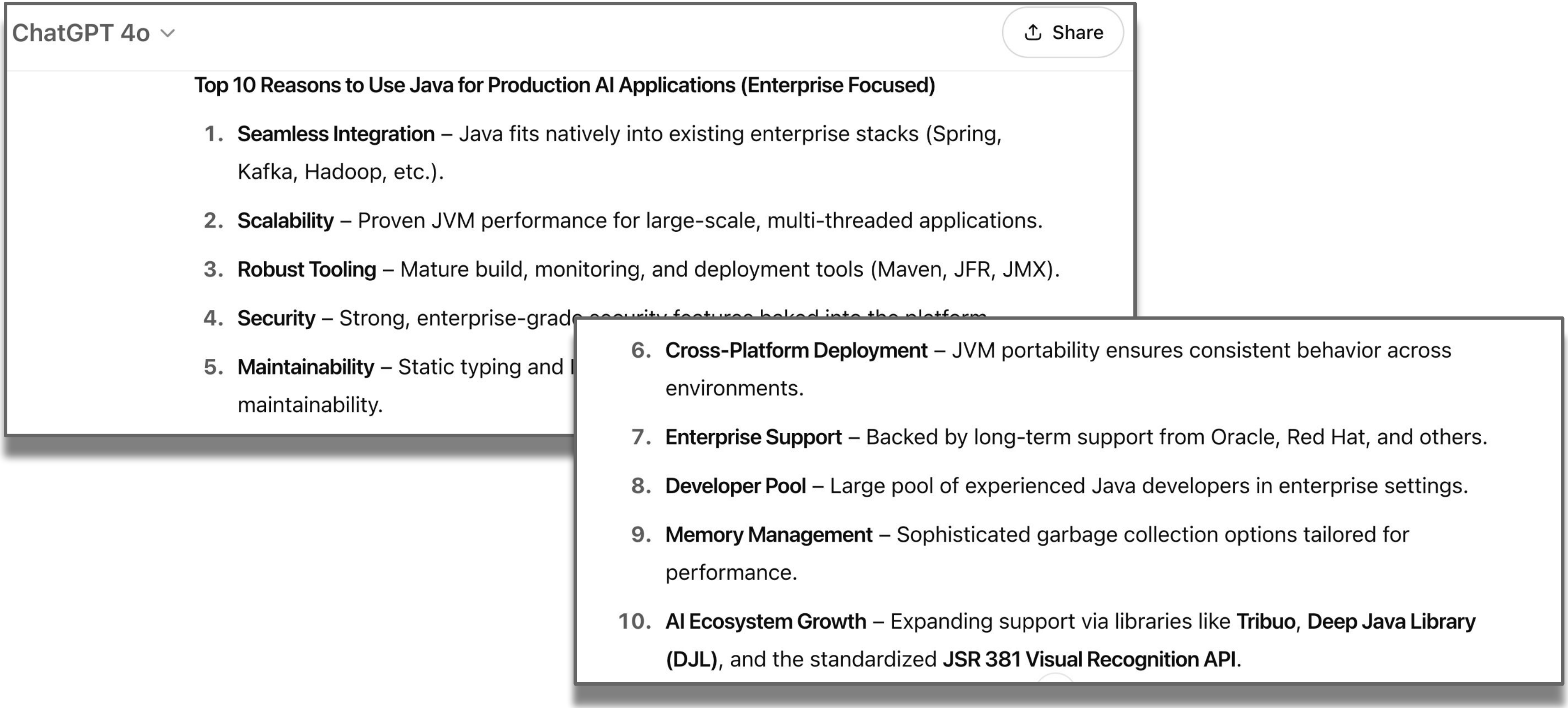
GenAI



user
or
app
prompt
completion
LLM
Query
Multi-level Task Splitting
Function Call
f{...}
OS Agent
Monitor
Function Calls
read_file
write_file
exec_python
python_input
add_agent
talk
......
Provide Interface
Parallel tasks
Action Module
Level 1
Admin 1
Admin 2
Admin 3
...
Monitor
Level 2
Group 1
Group 2
Group 3
...
Zoom in
Group 2
Group 2_2
Admin 2_2
prompt
Group 2_1
prompt
prompt
prompt
Admin 2_1
prompt
prompt
prompt
Level 3
Data Retriver
f{...}
Level 4
Group 2_2_1
...
Admin 2_2_1
Data Retriver
f{...}
Storage Module
Log (Serialized)
Memory Database
Task Monitor
Interactive Python Exec
Python
Files
Checklist

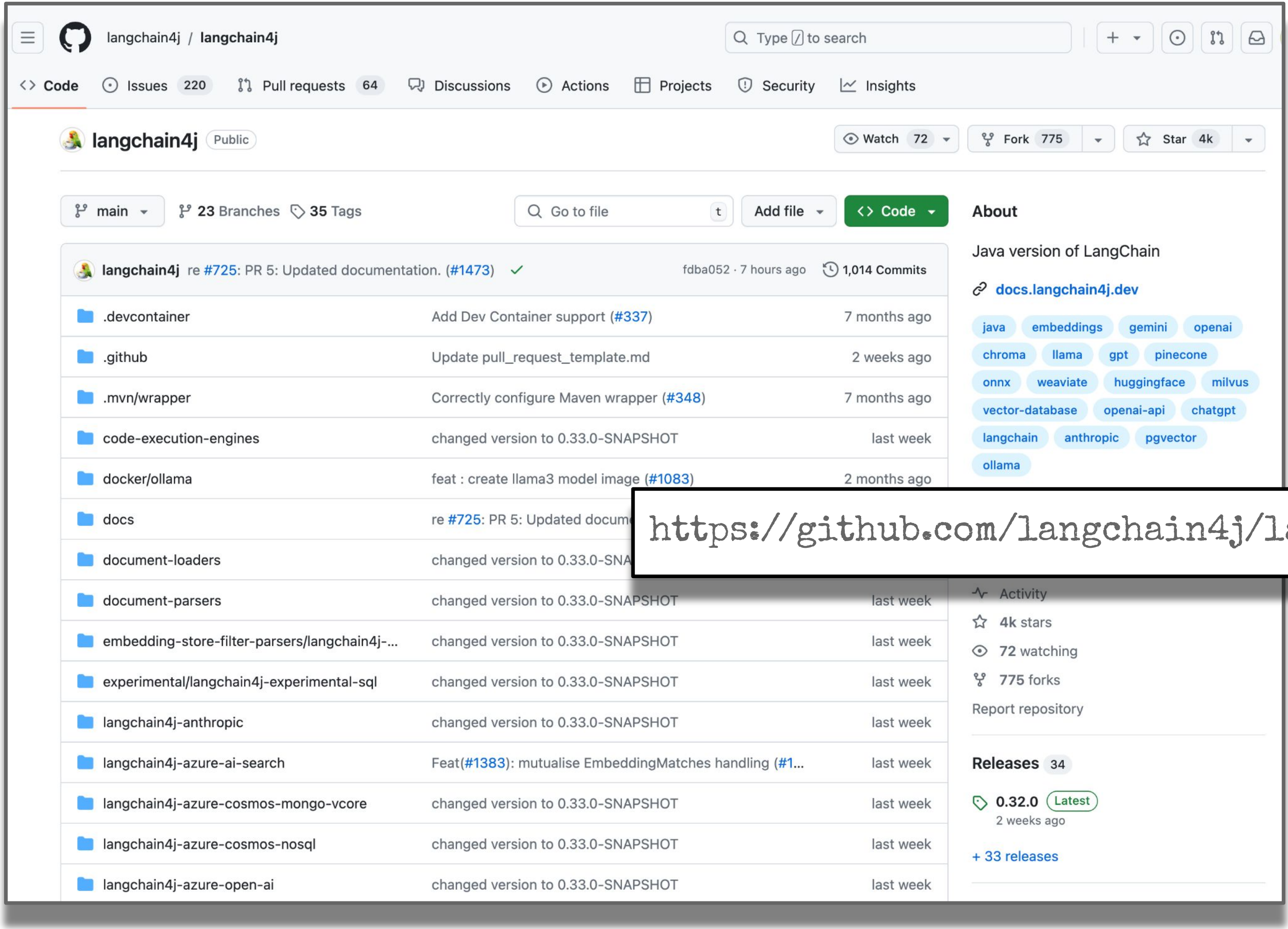
GenAI

Why Use Java for Production AI Applications?



Top 10 Reasons to Use Java for Production AI Applications (Enterprise Focused)

1. **Seamless Integration** – Java fits natively into existing enterprise stacks (Spring, Kafka, Hadoop, etc.).
2. **Scalability** – Proven JVM performance for large-scale, multi-threaded applications.
3. **Robust Tooling** – Mature build, monitoring, and deployment tools (Maven, JFR, JMX).
4. **Security** – Strong, enterprise-grade security features baked into the platform.
5. **Maintainability** – Static typing and maintainability.
6. **Cross-Platform Deployment** – JVM portability ensures consistent behavior across environments.
7. **Enterprise Support** – Backed by long-term support from Oracle, Red Hat, and others.
8. **Developer Pool** – Large pool of experienced Java developers in enterprise settings.
9. **Memory Management** – Sophisticated garbage collection options tailored for performance.
10. **AI Ecosystem Growth** – Expanding support via libraries like **Tribuo**, **Deep Java Library (DJL)**, and the standardized **JSR 381 Visual Recognition API**.



langchain4j / langchain4j

Type / to search

Code | Issues 220 | Pull requests 64 | Discussions | Actions | Projects | Security | Insights

langchain4j Public

Watch 72 | Fork 775 | Star 4k

main | 23 Branches | 35 Tags | Go to file t | Add file | Code

langchain4j re #725: PR 5: Updated documentation. (#1473) ✓ fdba052 · 7 hours ago 1,014 Commits

.devcontainer	Add Dev Container support (#337)	7 months ago
.github	Update pull_request_template.md	2 weeks ago
.mvn/wrapper	Correctly configure Maven wrapper (#348)	7 months ago
code-execution-engines	changed version to 0.33.0-SNAPSHOT	last week
docker/ollama	feat : create llama3 model image (#1083)	2 months ago
docs	re #725: PR 5: Updated docum	
document-loaders	changed version to 0.33.0-SNA	
document-parsers	changed version to 0.33.0-SNAPSHOT	last week
embedding-store-filter-parsers/langchain4j-...	changed version to 0.33.0-SNAPSHOT	last week
experimental/langchain4j-experimental-sql	changed version to 0.33.0-SNAPSHOT	last week
langchain4j-anthropic	changed version to 0.33.0-SNAPSHOT	last week
langchain4j-azure-ai-search	Feat(#1383): mutualise EmbeddingMatches handling (#1...	last week
langchain4j-azure-cosmos-mongo-vcore	changed version to 0.33.0-SNAPSHOT	last week
langchain4j-azure-cosmos-nosql	changed version to 0.33.0-SNAPSHOT	last week
langchain4j-azure-open-ai	changed version to 0.33.0-SNAPSHOT	last week

About

Java version of LangChain

docs.langchain4j.dev

java · embeddings · gemini · openai · chroma · llama · gpt · pinecone · onnx · weaviate · huggingface · milvus · vector-database · openai-api · chatgpt · langchain · anthropic · pgvector · ollama

Activity

4k stars

72 watching

775 forks

Report repository

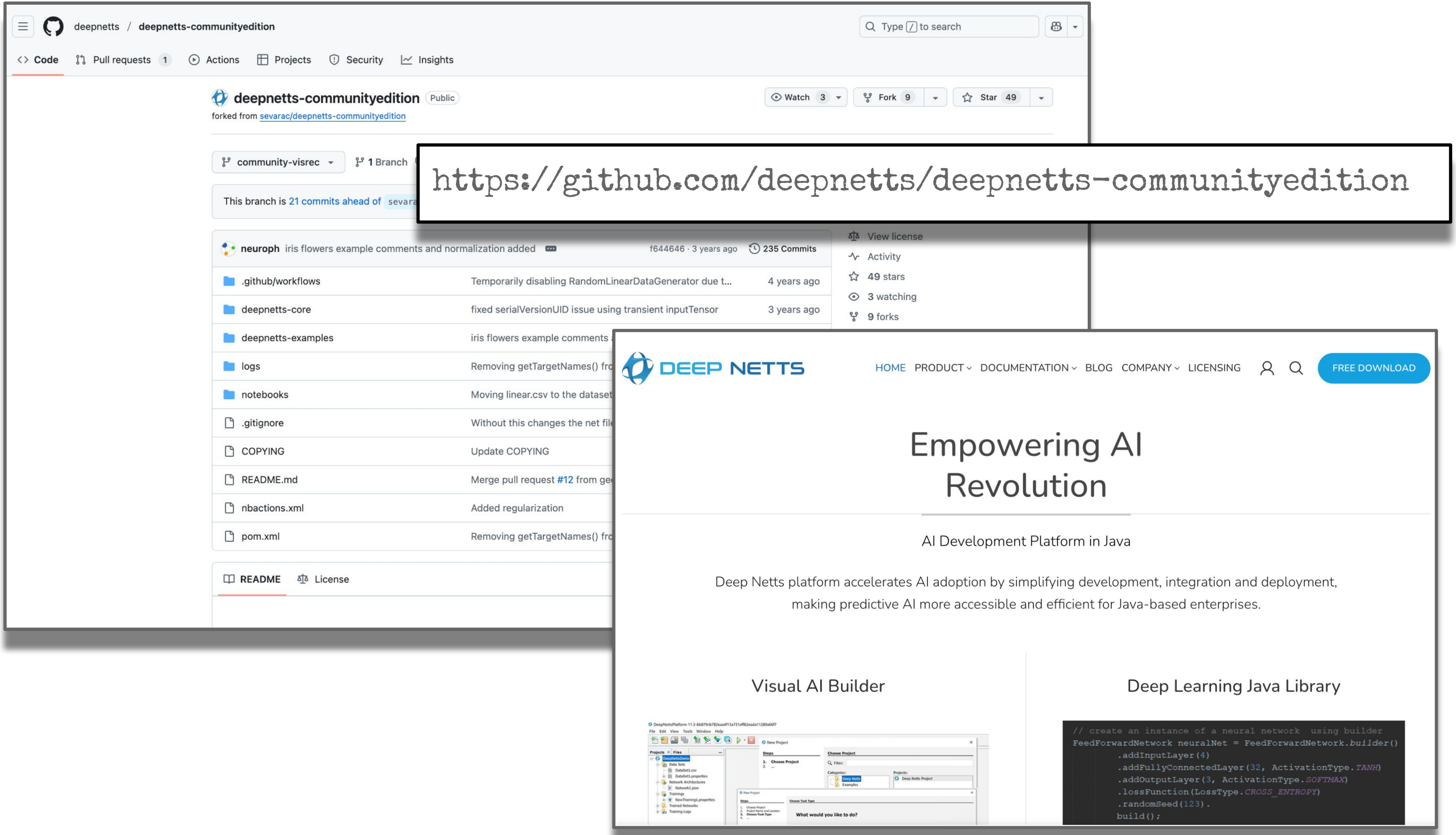
Releases 34

0.32.0 Latest

2 weeks ago

+ 33 releases

https://github.com/langchain4j/langchain4j https://github.com/deepnetts/deepnetts-communityedition



deepnetts / deepnetts-communityedition

Type / to search

deepnetts-communityedition Public

forked from sevarac/deepnetts-communityedition

This branch is 21 commits ahead of sevara

neuroph iris flowers example comments and normalization added f644646 · 3 years ago 235 Commits

.github/workflows	Temporarily disabling RandomLinearDataGenerator due t...	4 years ago
deepnetts-core	fixed serialVersionUID issue using transient inputTensor	3 years ago
deepnetts-examples	iris flowers example comments	
logs	Removing getTargetNames() fro	
notebooks	Moving linear.csv to the dataset	
.gitignore	Without this changes the net file	
COPYING	Update COPYING	
README.md	Merge pull request #12 from ge	
nbactions.xml	Added regularization	
pom.xml	Removing getTargetNames() fro	

Empowering AI Revolution

AI Development Platform in Java

Deep Netts platform accelerates AI adoption by simplifying development, integration and deployment, making predictive AI more accessible and efficient for Java-based enterprises.

Visual AI Builder

Deep Learning Java Library

```
// create an instance of a neural network  using builder
FeedForwardNetwork neuralNet = FeedForwardNetwork.builder()
        .addInputLayer(4)
        .addFullyConnectedLayer(32, ActivationType.TANH)
        .addOutputLayer(3, ActivationType.SOFTMAX)
        .lossFunction(LossType.CROSS_ENTROPY)
        .randomSeed(123).
        build();
```

How to use AI models in Java

1. **Use Web service - serve model as web service**
   You can build a model in any language, using a model is language agnostic, potential issues with latency and scalability.

2. **Use wrappers or native libraries from Java**
   Native dependencies, limited scalability, memory issues, distribution and maintenance overhead (aka. nightmare)

3. **Use Java native AI libraries**
   Highly scalable, low latency, easy to use, integrate with existing development and production environment, and distribute on large scale.
   Out of the box models mostly not available.

Overview of Java AI Libraries

	Description	Pros	Cons
Tensorflow Java API	Java API for Tensorflow developed by community	Many out of the box models available, GPU and TPU support, many algorithms and architectures,	Large size, Not covered with compatibility guarantees Requires lower level understanding of TF internals
DJL	Wrapper around Python based frameworks	Many out of the box models available	Mixed tech stacks
DL4J	Java Deep Learning API on top of native numeric libraries	High performance, GPU support many algorithms and architectures, feature rich	Native dependencies, Large size, not actively developed, complex to learn and use
Deep Netts	Java native, gpu support through jcuda	Easy to learn and use, Easy to integrate into existing Java infrastructure , highly scalable, low latency	Not all algorithms are supported

About Deep Netts AI Platform for Java

Deep Netts provides:

- Deep Learning Java Library
  Build, train, and deploy NN natively in Java
- Visual AI Builder Tool
  Intuitive, no-code/low-code environment for model creation
- Professional Support
  Expert assistance for development and production needs
- Community edition
  Free and open access for exploration and learning

Deep Netts makes predictive AI

✅ More accessible
✅ More efficient
✅ Easier to integrate and deploy within Java-based enterprise systems

By accelerating and simplifying:

- AI development
- AI integration
- AI deployment

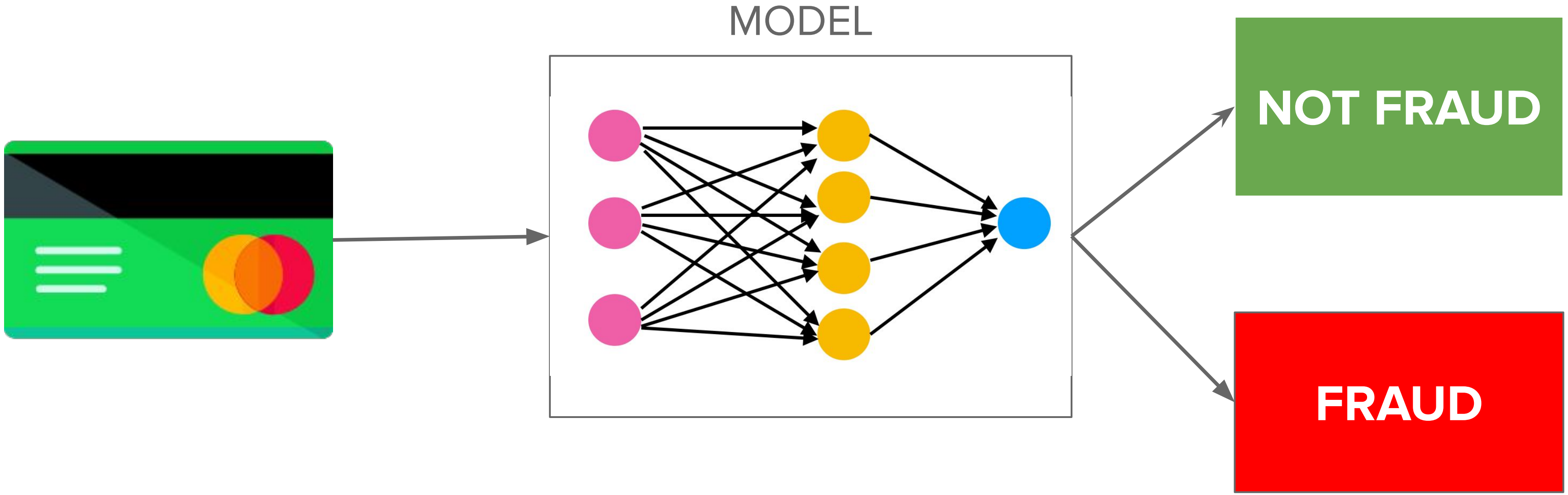
Vision: We would like to see the Java platform evolve to meet modern AI requirements.

DEEP NETTS

AI Demo using Deep Netts

1. Define the problem description
2. Prepare the data
3. Build the model
4. Test/evaluate the model
5. Use the model

1. Define the problem description: Fraud Detection



MODEL

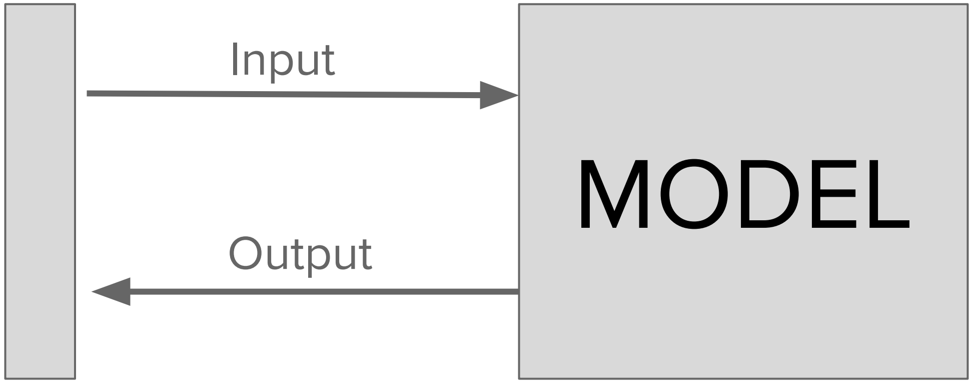
NOT FRAUD

FRAUD

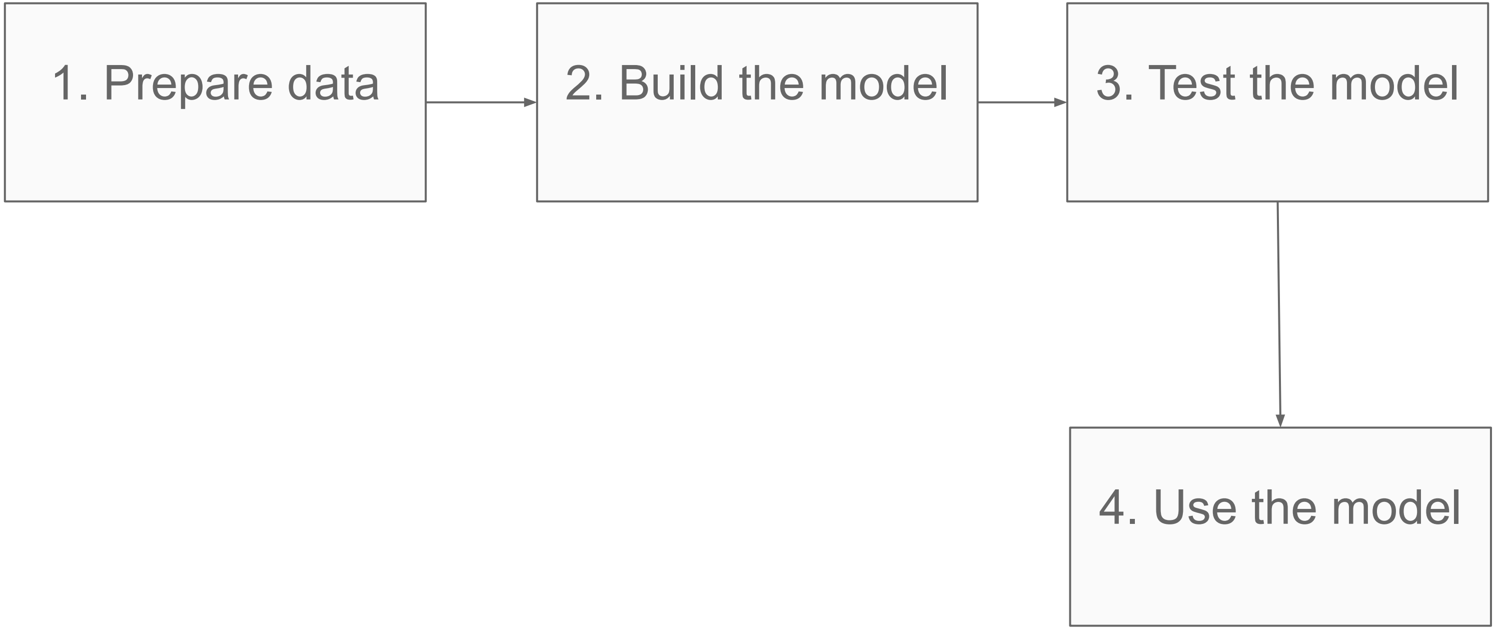
Given the credit card transaction details, how likely is that it is a fraud?

AI/ML Models - Software Engineer definition

Model is just another type of abstraction in developer's toolbox.



Workflow Overview



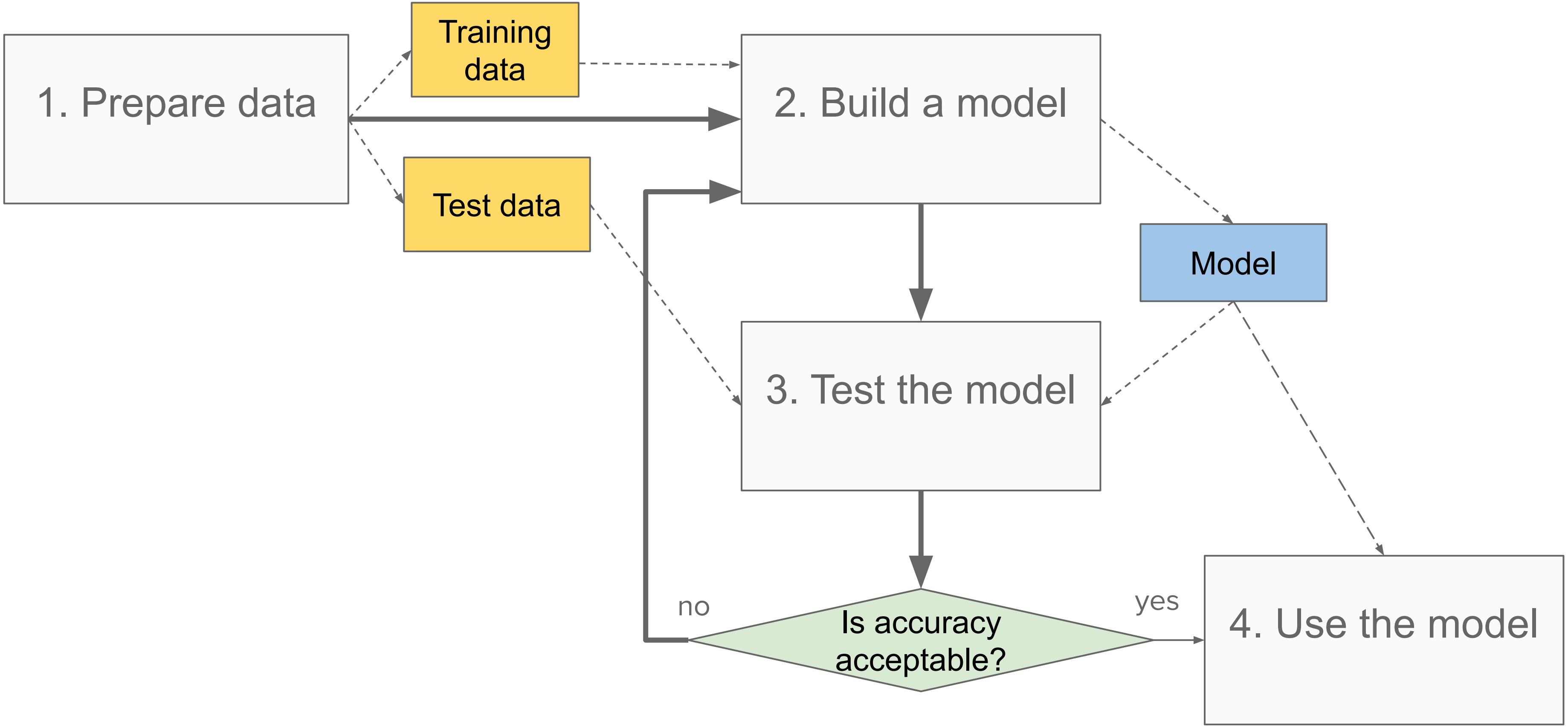
1. Prepare data

2. Build the model

3. Test the model

4. Use the model

Workflow Overview



2. Prepare the data

```
// specify CSV file options
CsvReadOptions.Builder builder =
CsvReadOptions.builder("creditcard.csv")
        .separator(',') // values are coma-delimited
        .header(true); // first line contains column names
CsvReadOptions options = builder.build();

 // load data into a data frame
Table dataTable = Table.read().usingOptions(options);
```

Load data from CSV file into a **Data Frame** - basically a Table using Tablesaw Java Library

```
// prepare data for training
DataPreparation dataPrep = new DataPreparation(dataTable);
```

Out of the box, single method call, data utilities

```
// print header and first few rows to see what's loaded
dataPrep.previewRows(5);

// print columns with corresponding types
dataPrep.columnInfo();
```

Check loaded - rows and columns

Preprocess the data to ensure data quality

```
dataTable.removeColumns("Time");

dataTable.dropDuplicateRows();

// check if there are any missing values
dataPrep.countMissingValues();

dataPrep.handleMissingValues();

// examine basic statistics summary
dataPrep.statistics();

// check if data set is balanced
DataPreparation.checkClassBalance(dataTable);

// create balanced subset
Table balancedData =
DataPreparation.createBalancedSample(dataTable, "Class", 1)
```

Decide which columns are needed

Remove duplicate rows

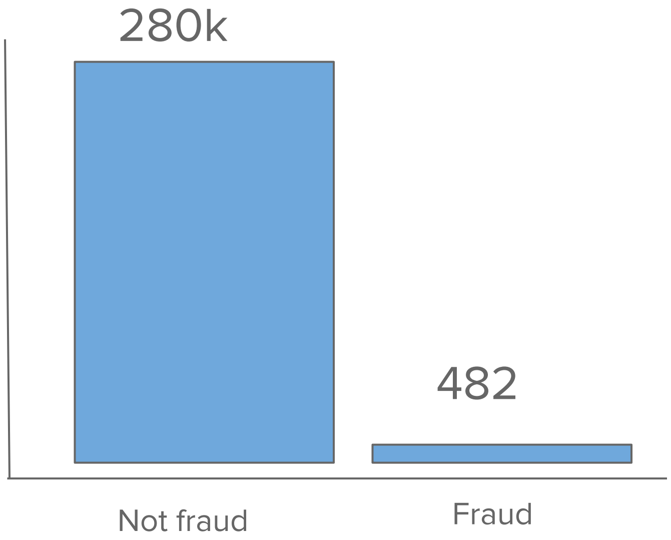
Check and fix missing values

Explore statistics for each column, check distribution

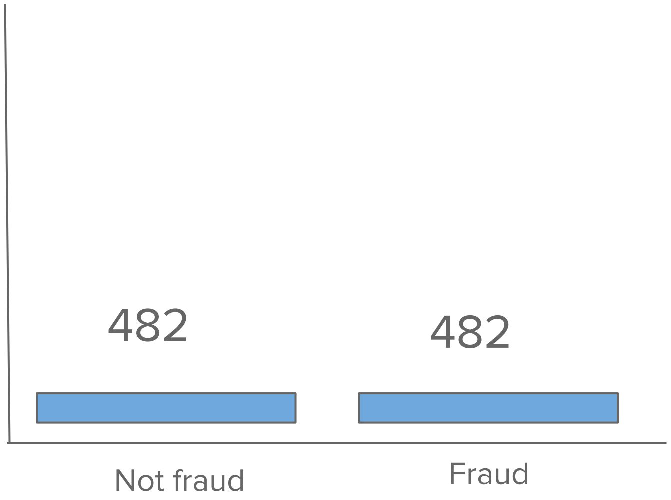
Check for outliers

Make sure data set is balanced

Balancing Data Set



Unbalanced data set



Balanced data set

Prepare data for training

```
// create data set for neural network training
TabularDataSet dataSet = DataPreparation.createDataSet(balancedData);
```

Create data set for nn training

```
// scale data to value range [0, 1]
DataSets.scaleToMax(dataSet);
```

Scale values to range used in nn

```
// split data into training and test set
TrainTestSplit split = DataSets.trainTestSplit(dataSet, 0.6);
DataSet<MLDataItem> trainingSet = split.getTrainingSet();
DataSet<MLDataItem> testSet = split.getTestSet();
```

Training set to train the model

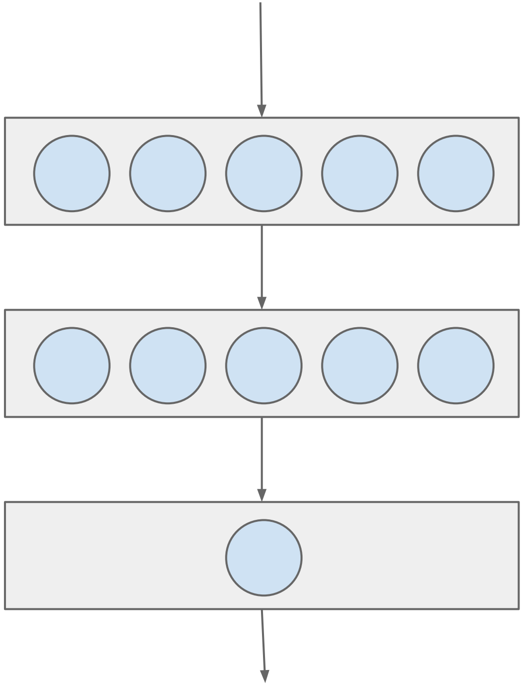
Test set to estimate accuracy on unseen data

3. Build a Model - Feed Forward Neural Network

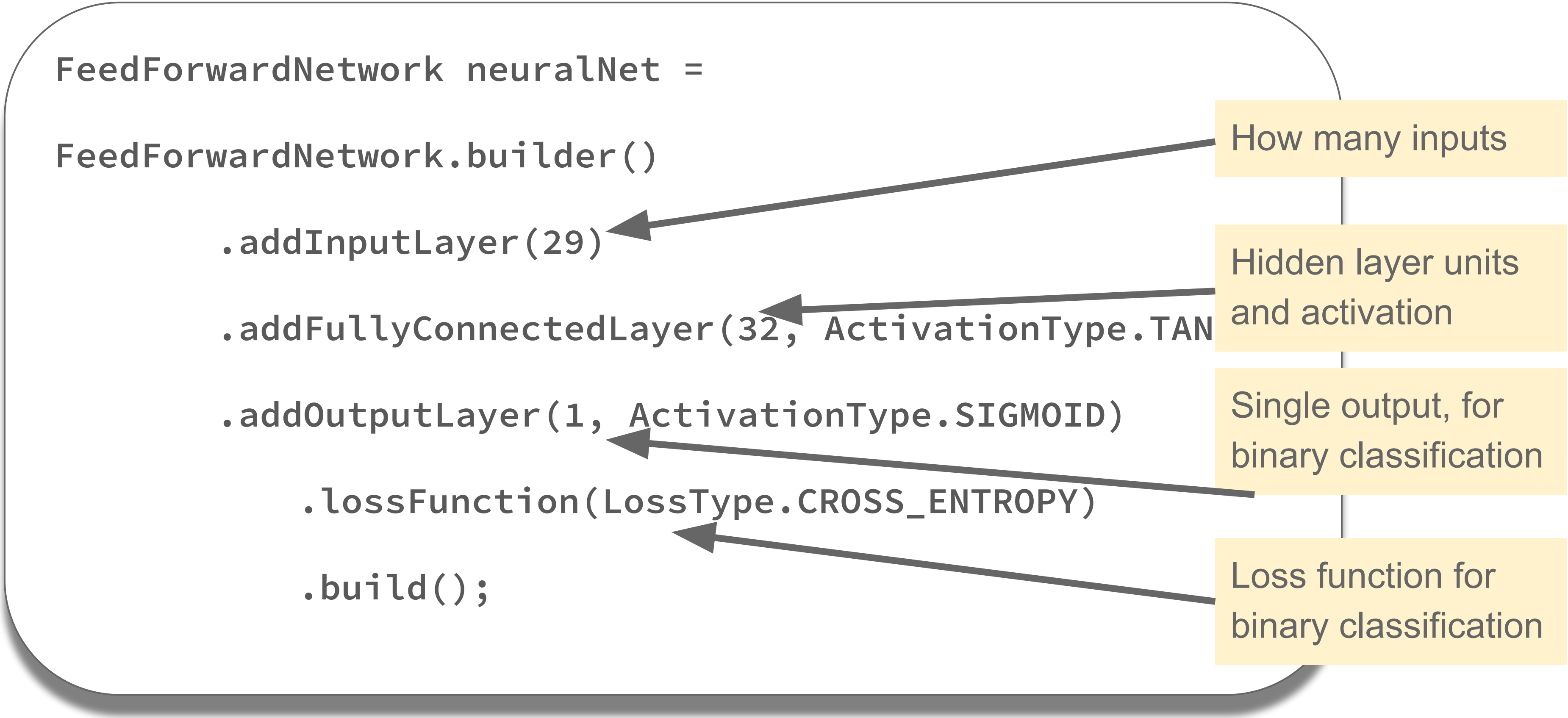
```
// instantiate and configure a neuralnet for binary classification
 FeedForwardNetwork neuralNet = FeedForwardNetwork.builder()
                .addInputLayer(numInputs)
                .addFullyConnectedLayer(32, ActivationType.TANH)
                .addOutputLayer(numOutputs, ActivationType.SIGMOID)
                .lossFunction(LossType.CROSS_ENTROPY)
                .build();

// set parameters of the training algorithm
neuralNet.getTrainer().setStopError(0.02f)
                                      .setStopEpochs(10000)
                                      .setLearningRate(0.001f);

// train the model prepared data set
neuralNet.train(trainingSet);
```



Configure the Model Architecture



```
FeedForwardNetwork neuralNet =

FeedForwardNetwork.builder()

        .addInputLayer(29)

        .addFullyConnectedLayer(32, ActivationType.TAN

        .addOutputLayer(1, ActivationType.SIGMOID)

            .lossFunction(LossType.CROSS_ENTROPY)

            .build();
```

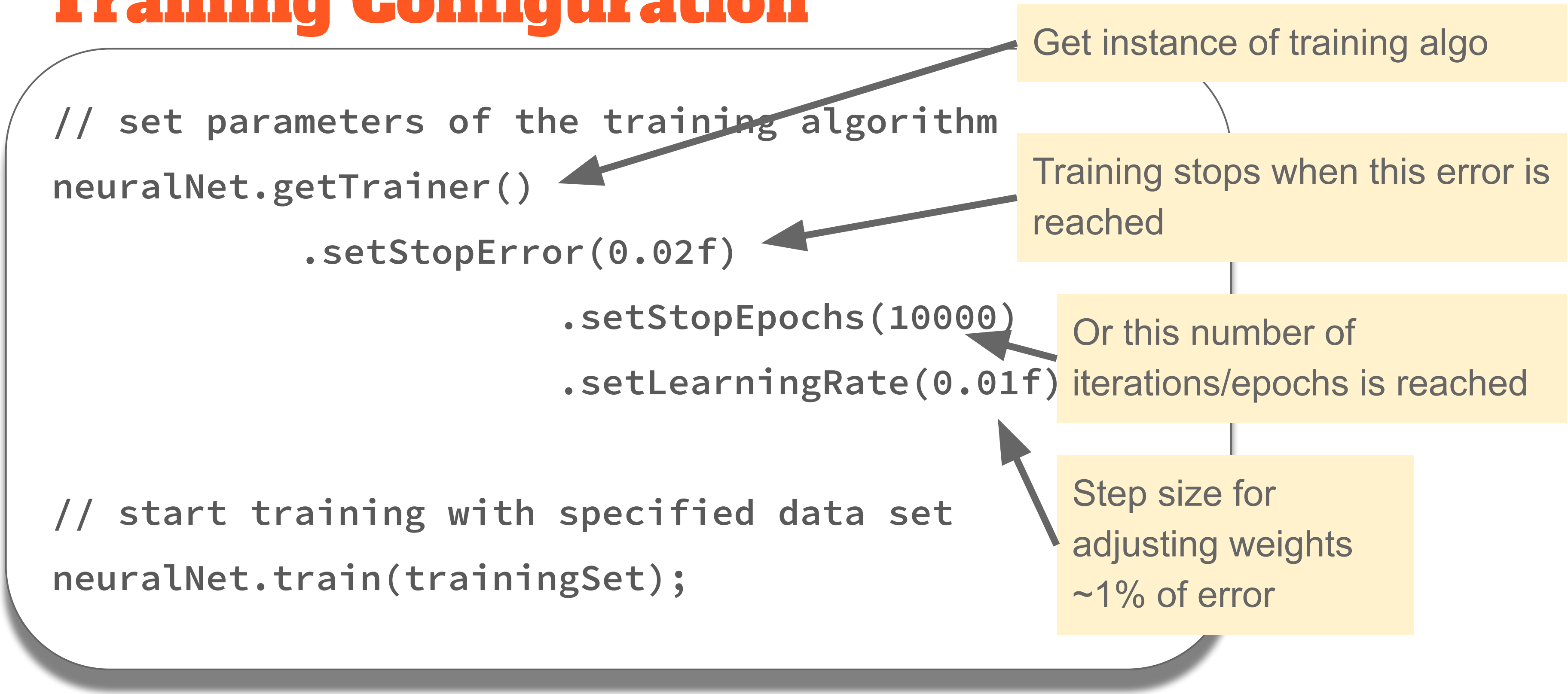
How many inputs

Hidden layer units and activation

Single output, for binary classification

Loss function for binary classification

Training Configuration



```
// set parameters of the training algorithm
neuralNet.getTrainer()
           .setStopError(0.02f)
                       .setStopEpochs(10000)
                       .setLearningRate(0.01f)

// start training with specified data set
neuralNet.train(trainingSet);
```

Get instance of training algo

Training stops when this error is reached

Or this number of iterations/epochs is reached

Step size for adjusting weights ~1% of error

Training Process

```
while(epoch < stopEpochs && error > stopError) {

   for each example in trainingSet {
     calculate prediction for input
     calculate prediction error
     tune weights in order to reduce error
   }

}
```

Debugging and Hyperparameter Tuning

Q: What if error is increasing during the training?
Try smaller learning rate

Q: What if error is not decreasing even after many training epochs?
Try adding more hidden units and layers

Q: Is the model overfitting (low training, high test error)?
Add more data or try simpler architecture

Hint: Use Visual AI Builder tool to run experiments and tune settings

4. Testing/Evaluating the model - How accurate?

```
EvaluationMetrics em = neuralNet.test(testSet);
```

Calculate various eval metrics using test set

Metrics: True Positive, True Negative, False positive, False Negative
Accuracy, Precision, Recall, F-Score

1. How often the prediction is correct when the model predicts fraud?
2. How often the prediction is incorrect when the model predicts fraud?
   How often the prediction is correct in total? (accuracy)
3. What is more important/expensive (Falso Positive or False Negative)?
   The trick is to find balance.

5. Using the model - fraud probability

What is that the probability that the given transaction is fraud?

```
Tensor prediction = neuralNet.predict(inputTensor);

if (prediction.get(0) > 0.8) {

    System.out.println(“There is a high probability that this is fraud!”);
}
```

Feed model with input as tensor and get prediction as tensor

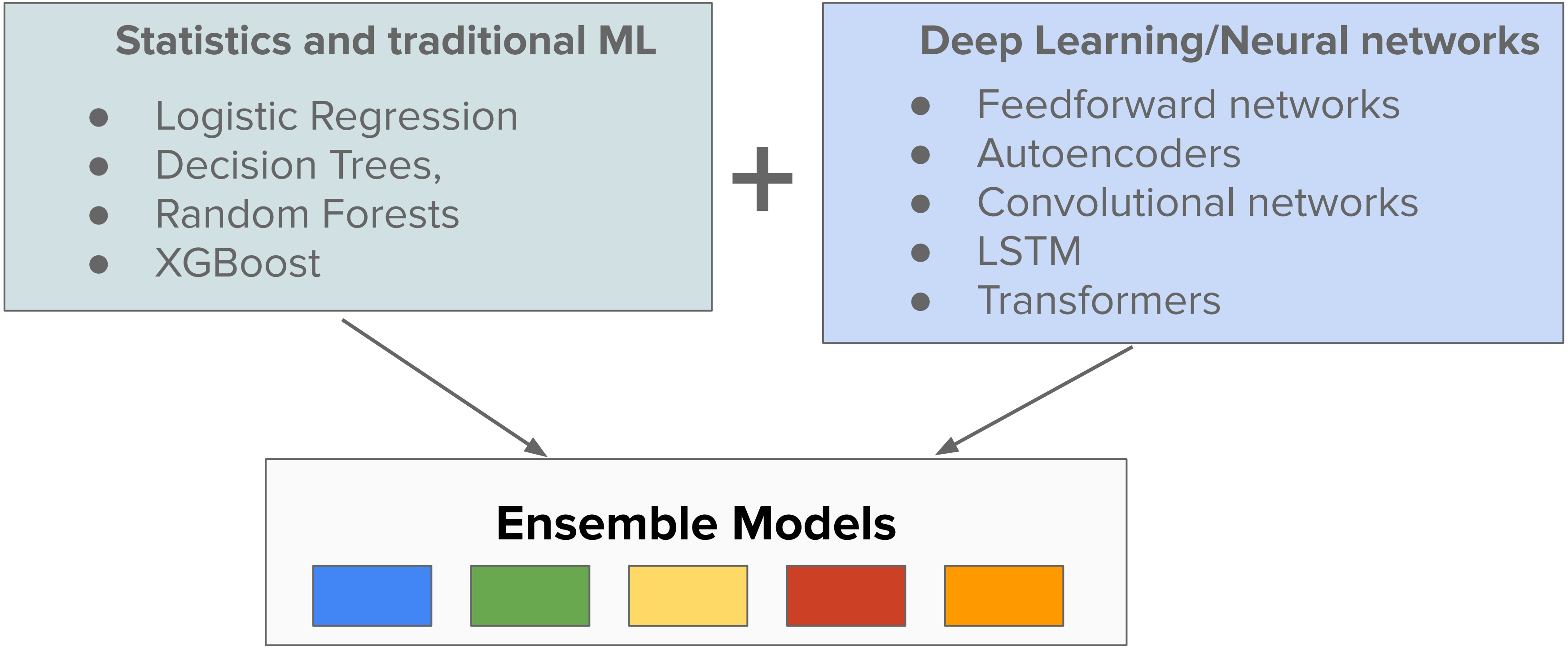
Using JSR381

Use model through higher level classifier API

```
BinaryClassifier<float[]> binClassifier = new FeedForwardNetBinaryClassifier(neuralNet);

Float probability = binClassifier.classify(testTransaction);
```

Combine with Other Methods



Statistics and traditional ML

- Logistic Regression
- Decision Trees,
- Random Forests
- XGBoost

+

Deep Learning/Neural networks

- Feedforward networks
- Autoencoders
- Convolutional networks
- LSTM
- Transformers

Ensemble Models

Take Your Java AI Projects to the Next Level with Deep Netts Expertise!

🔹 **Are you using AI in your Java applications?**

- What libraries or tools are you currently using?
- Are you facing challenges building or deploying your AI models
- Have an exciting project you'd like to discuss?

🔹 **Deep Netts offers expert consulting for**:

- AI/ML solution design and integration in Java
- Optimizing AI models for production environments
- Scaling AI across distributed Java systems

What's next for Java AI developers

Resources:

- https://github.com/deepnetts/CreditCardFraudDetection Demo on Github
- https://www.deepnetts.com/download-latest Free Deep Netts Download -*Free for development, prototyping and education*

Join Linkedin Group AI/ML for Java Ecosystem

https://www.linkedin.com/groups/10084933/

Quick Survey for future webinars

https://forms.gle/RfskNfohaKkhP15Q9