Applied Machine Learning

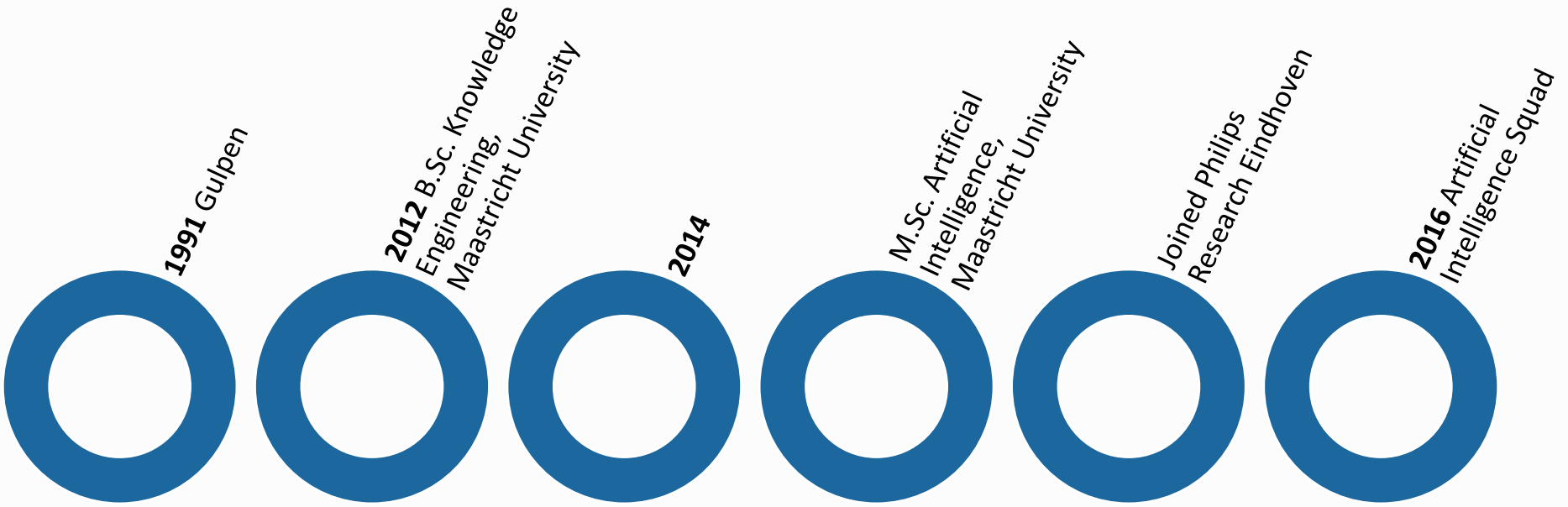
with Apache Spark

CLIFF LASCHET

PHILIPS RESEARCH

NEXTBUILD 2016

About me


1991 Gulpen
2012 B.Sc. Knowledge Engineering, Maastricht University
2014
M.Sc. Artificial Intelligence, Maastricht University
Joined Philips Research Eindhoven
2016 Artificial Intelligence Squad

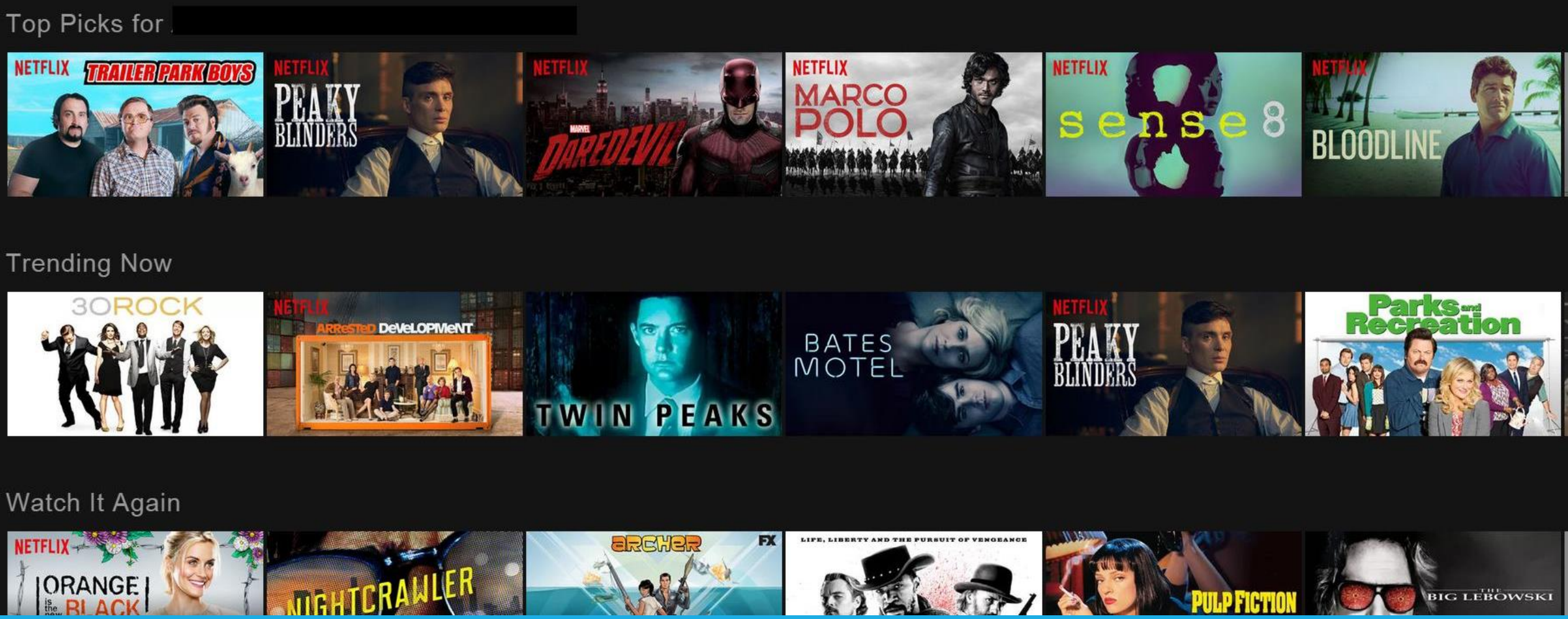
Have you previously used machine learning?

THE CORRECT ANSWER IS *"YES"*

MACHINE LEARNING IS EATING THE WORLD

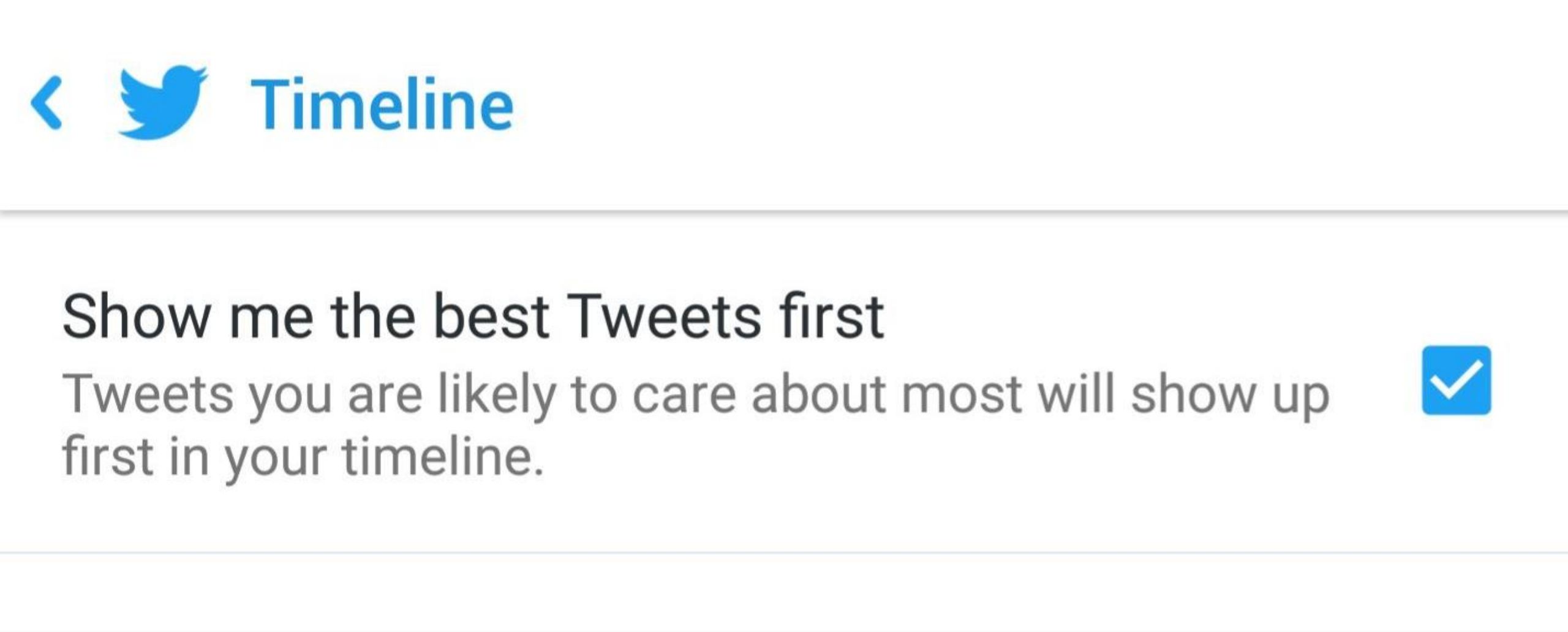
Google Now

Voice commands, weather, commute times, articles related to your search interests, Google search, etcetera.



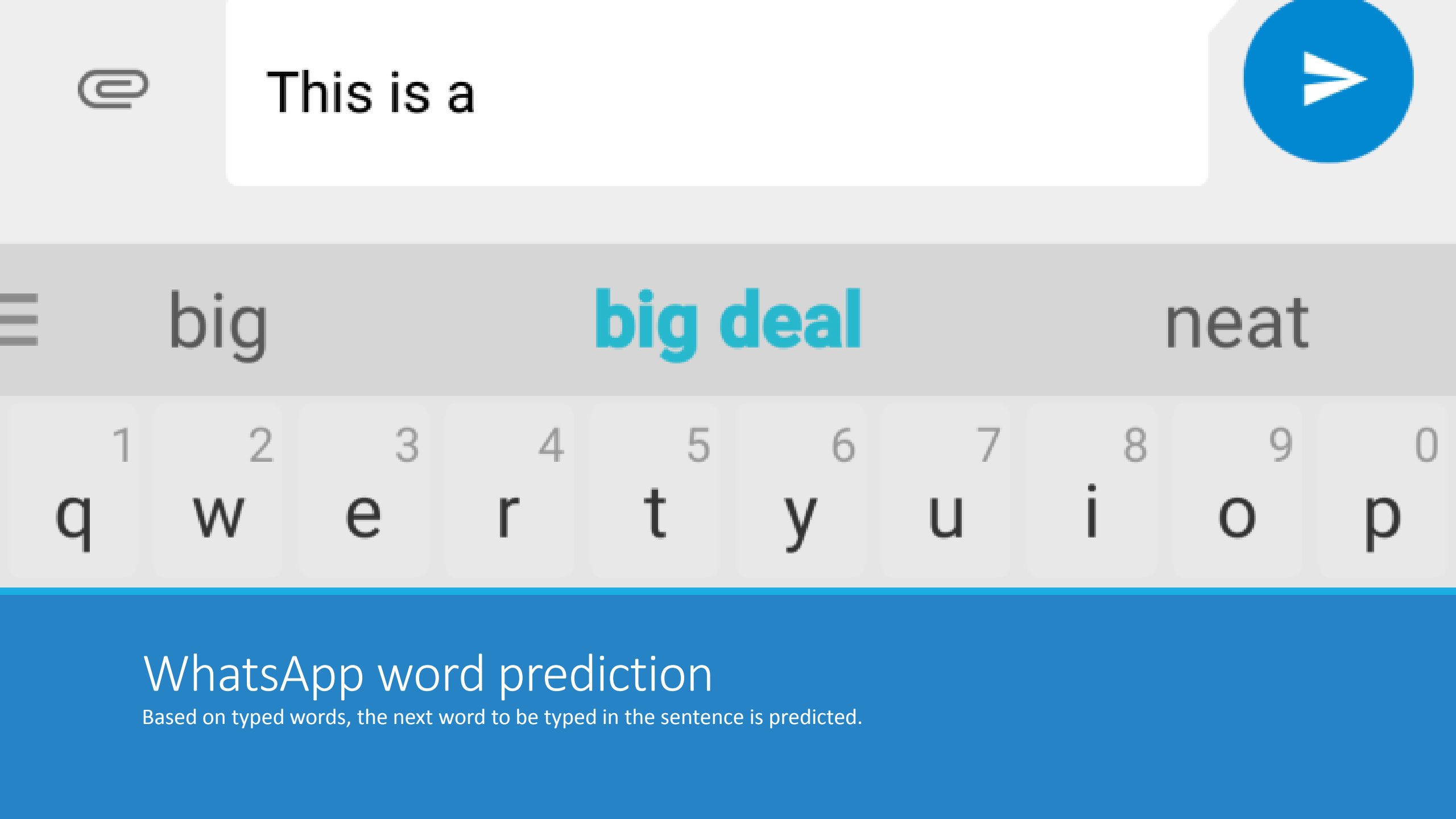
Netflix top picks

Netflix recommends movies based on your and others' viewing and rating history.



Social media feed ordering

Twitter orders your feed items to match your interests, showing the most relevant tweets first.



WhatsApp word prediction

Based on typed words, the next word to be typed in the sentence is predicted.

Other applications

Entertainment/media

- Search
- Spam filter

Medical[1,2,3]

- Personalized medical diagnosis
- Treatment selection
- Predictive diagnosis
- Drug discovery
- Human genome

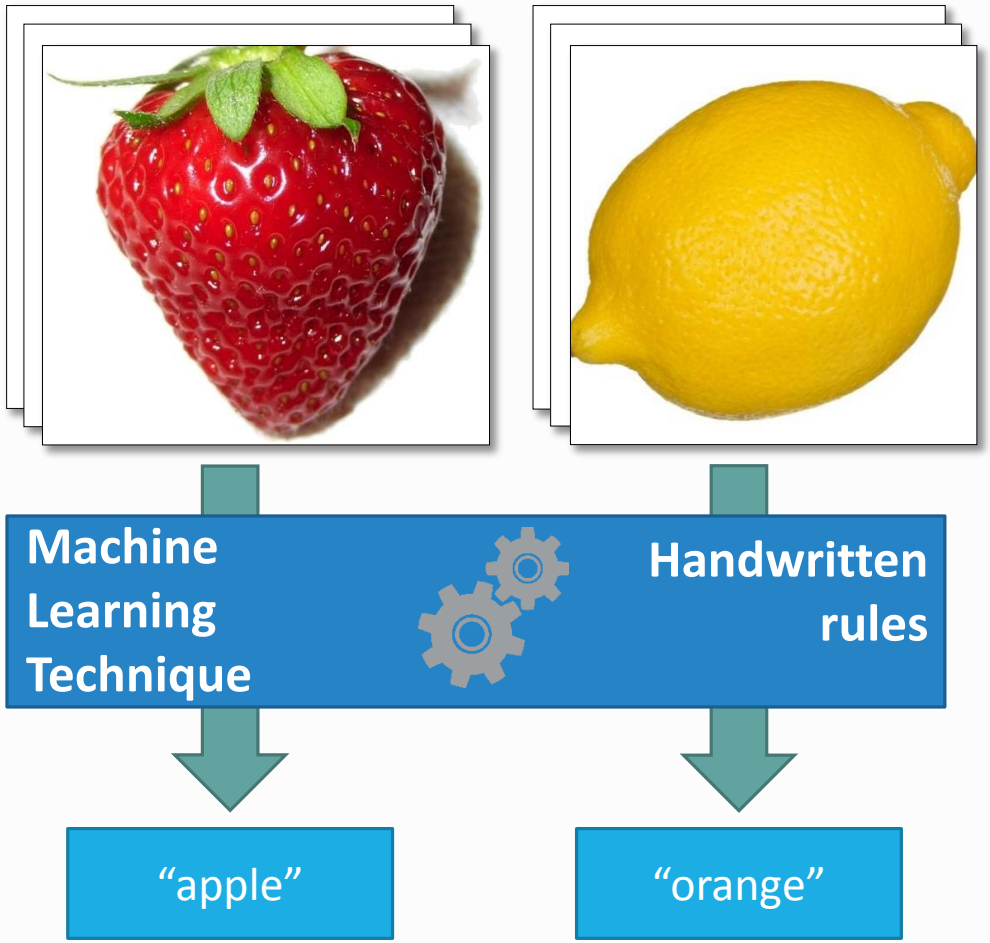
Financial

- Credit card fraud detection[4]
- Predicting the trade price of corporate bonds[5]

Software development

- Automated testing of software
- Predicting which questions will be closed on StackOverflow [6]

Machine Learning

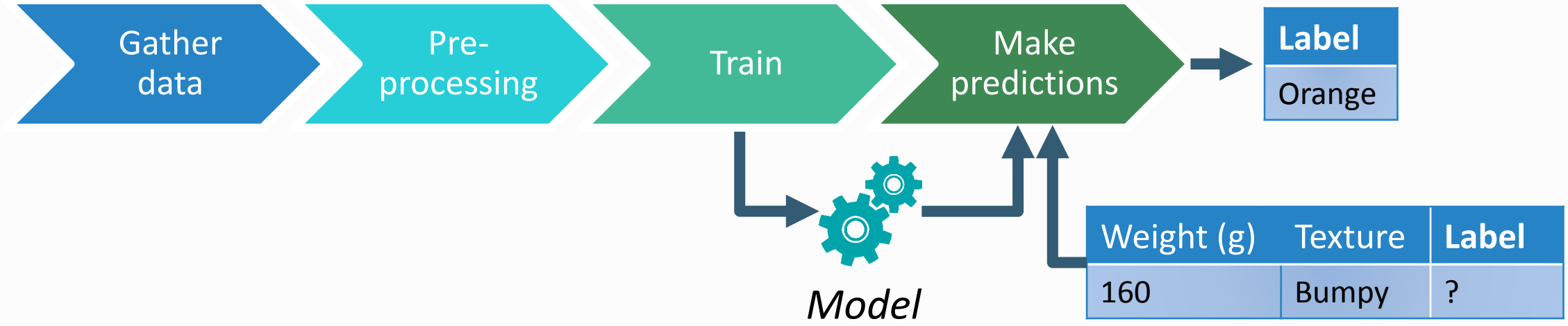



```
def detect_colors(image):
    #lots of code

def guess_texture(image):
    #lots of code

def detect_edges(image):
    #lots of code

def analyze_shapes(image):
    #lots of code

…
```

Not explicitly programming rules, but programming how to learn these rules automagically from data.

Machine Learning Recipe

Time	Weight (g)	Texture	**Label**
1460808900	150	Bumpy	Orange
1460809200	170	Bumpy	Orange
1460809500	140	Smooth	Apple
1460809800	130	Smooth	Apple



“Sudden” popularity

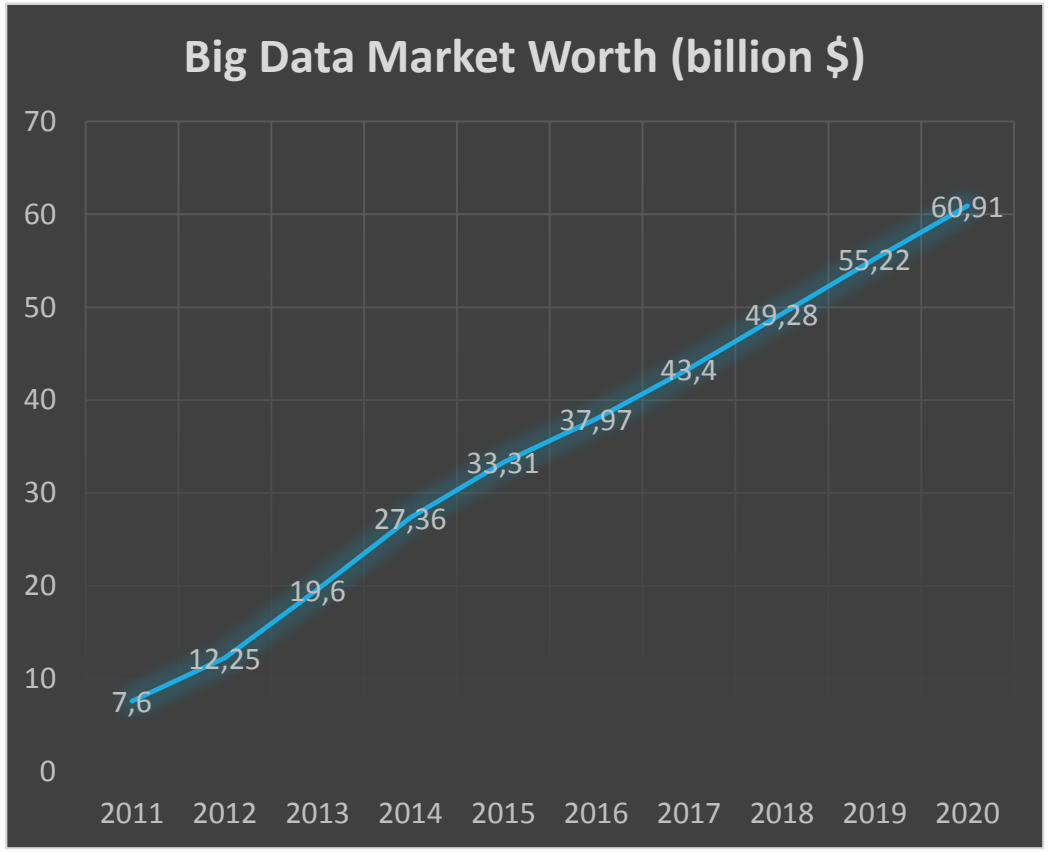
Every disruptive company is using machine learning.

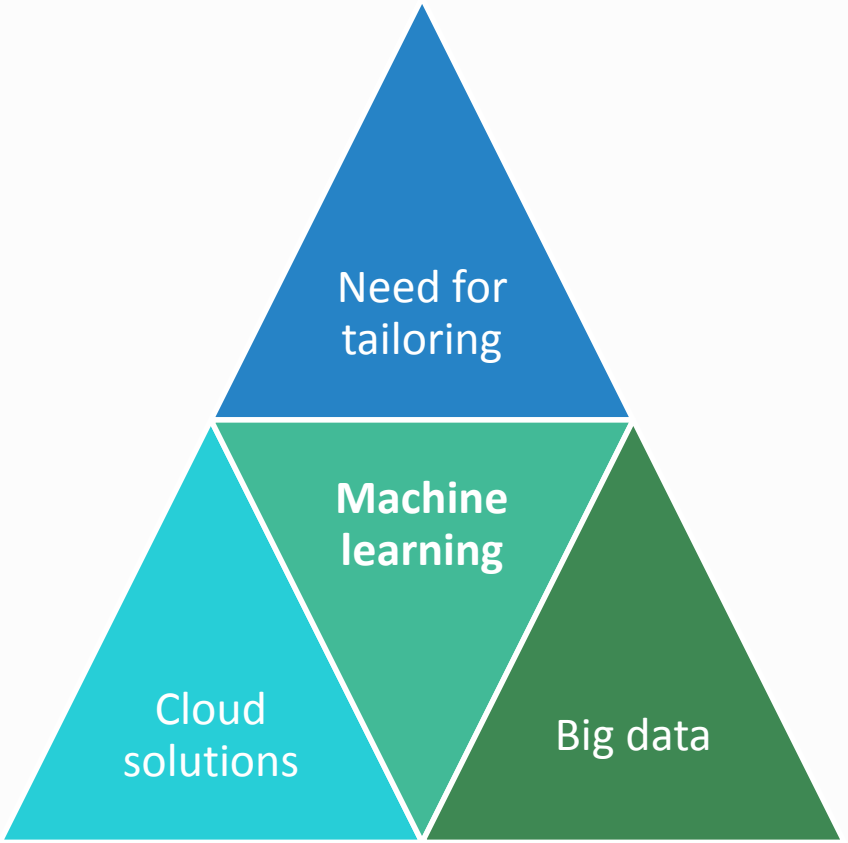
Machine Learning is one of the most popular courses at Stanford University.[8]

The quest for General Artificial Intelligence.[9]

Substantial leaps of progress made past few years (DeepMind).[10]

“Sudden” popularity





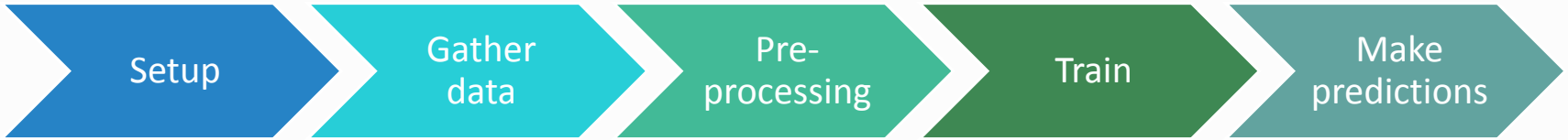
[11, 12]

Tooling


Scalability
Time (years)
WEKA
The University of Waikato
Open for Innovation
KNIME
rapidminer
scikit
learn
mahout
Spark
R

Let’s apply!

- Build a basic version of the Netflix Movie Recommendation Engine
- Goal: recommend movies based on your ratings and population ratings
- Using the machine learning recipe discussed before
- MovieLens training data[13]



```
 * A) Gathering raw data.
 * B) Pre-processing it to training data.
 * C) Training the ML technique using the training data.
 * D) Recommending movies using the obtained ML model and a small set of personal rati
 *
 * Created by Cliff Laschet on 2/25/2016.
 */
object MovieRecommender {

  def main(args: Array[String]) ={
    //Setup
    import LocalSparkContext._
    //-Set logging level
    Logger.getLogger("org.apache.spark").setLevel(Level.ERROR)
```

```
Logger.getLogger("org.apache.spark").setLevel(Level.ERROR)



//A. Load data
//1-Retrieve all the movies ever rated (not needed by
//  but convenient for printing recommendations later
val allMovies = Movies.getAllMovies()

//2-Retrieve all the ratings from the complete popula
val rawPopulationRatings = CSV.load("ratings.csv")

//3-Retrieve personal ratings from the user for the m
val rawPersonalRatings = Movies.getPersonalRatings(al
```

Population ratings

userId	movieId	rating	timeStamp
1	16	4.0	975430136
1	24	1.5	975430085
668	24	2.0	1202801515
668	142507	3.5	1025857084

Personal ratings

userId	movieId	rating	movieTitle
0	16	2.0	…
0	24	2.5	…
0	50	4.0	…
0	142507	3.5	…

```
//3-Retrieve personal rating                    he most frequently rated movies.
val rawPersonalRatings = Mov                    s(allMo                    tings)



//B. Pre-processing
//1-Drop columns in the data                    ed for
val preProcessedPopulationRa                    Ratings
val preProcessedPersonalRatings = rawPersonalRating                    ro

//2-Merge the population and                    one d
val unionData = preProcessed                    onAll(p                    tings)

//3-Transform the joined dat                    be us                    .
val trainingData = unionData                    g(row.g                    ), row.
```

Population ratings

userId	movieId	rating
1	16	4.0
1	24	1.5
668	24	2.0
668	142507	3.5

Personal ratings

userId	movieId	rating
0	16	2.0
0	24	2.5
0	50	4.0
0	142507	3.5

Union data

userId	movieId	rating
1	16	4.0
1	24	1.5
668	24	2.0
668	142507	3.5
0	16	2.0
0	24	2.5
0	50	4.0
0	142507	3.5

Training data

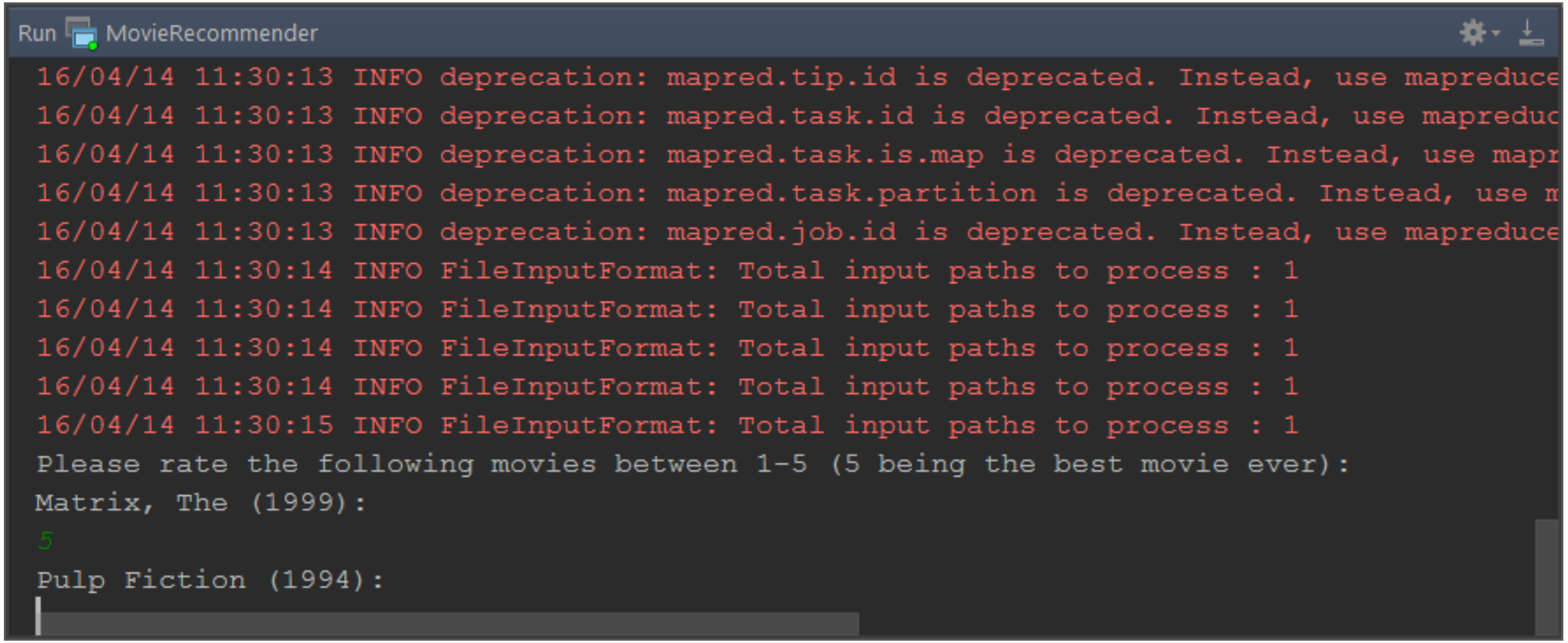
Rating
object ref.
object ref.
object ref.
object ref.
object ref.
object ref.
object ref.
object ref.

Setup → Gather data → **Pre-processing** → Train → Make predictions

```scala
//C. Training the ML technique, using the training data as input.
//1-Set some parameters of the ML technique (out of scope for this demo).
val rank = 10
val iterations = 10
val regularizationFactor = 0.1

//2-Provide the training data and parameters as input for the ML technique.
//  In this case, collaborative filtering is used, which is often taken as
//  a basis for recommendation systems (e.g. Netflix).
val model = ALS.train(trainingData, rank, iterations, regularizationFactor)
```

```scala
  val model = ALS.train(trainingData, rank, iterations, regularizationFactor)



  //D. Perform recommendation based on my individual interests, using my personal ratings as input.
  //1-Predict my interest for each movie, pick the top 25 movies with the highest predicted interest.
  val recommendations = model.recommendProducts(0, 25)

  //2-Print results
  var i = 1
  println("Movies recommended for you, based on your previous ratings:")
  recommendations.foreach { recommendation =>
    println(s"#$i: ${allMovies.get(recommendation.product).get} (expecting a rating of ${recommendation.rating})")
    i += 1
  }



  //Cleanup
  //-Stop Spark
  sparkContext.stop()
}
```

Demo


Run MovieRecommender
16/04/14 11:30:13 INFO deprecation: mapred.tip.id is deprecated. Instead, use mapreduce
16/04/14 11:30:13 INFO deprecation: mapred.task.id is deprecated. Instead, use mapreduc
16/04/14 11:30:13 INFO deprecation: mapred.task.is.map is deprecated. Instead, use mapr
16/04/14 11:30:13 INFO deprecation: mapred.task.partition is deprecated. Instead, use m
16/04/14 11:30:13 INFO deprecation: mapred.job.id is deprecated. Instead, use mapreduce
16/04/14 11:30:14 INFO FileInputFormat: Total input paths to process : 1
16/04/14 11:30:14 INFO FileInputFormat: Total input paths to process : 1
16/04/14 11:30:14 INFO FileInputFormat: Total input paths to process : 1
16/04/14 11:30:14 INFO FileInputFormat: Total input paths to process : 1
16/04/14 11:30:15 INFO FileInputFormat: Total input paths to process : 1
Please rate the following movies between 1-5 (5 being the best movie ever):
Matrix, The (1999):
5
Pulp Fiction (1994):

Next steps

- *github.com/clifflaschet/MovieRecommender*
  - Runnable artifact
  - Source code
  - Slide deck
  - Links to tutorials, videos, articles, software
- Connect to me through LinkedIn
- Questions?

References

[1] http://www.ibm.com/smarterplanet/us/en/ibmwatson/health/

[2] https://deepmind.com/health

[3] http://www.sciencedirect.com/science/article/pii/S1359644614004176

[4] https://www.research.ibm.com/foiling-financial-fraud.shtml

[5] https://www.kaggle.com/c/benchmark-bond-trade-price-challenge

[6] https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow

[7] https://developers.googleblog.com/2016/03/introducing-new-developer-show-machine.html

[8] http://www.forbes.com/sites/anthonykosner/2013/12/29/why-is-machine-learning-cs-229-the-most-popular-course-at-stanford/#defea8461ba4

[9] https://www.youtube.com/watch?v=8DRlNkhXsIk

[10] https://deepmind.com/alpha-go

[11] https://newsroom.intel.com/press-kits/celebrating-the-50th-anniversary-of-moores-law/

[12] http://wikibon.com/executive-summary-big-data-vendor-revenue-and-market-forecast-2011-2026/

[13] http://grouplens.org/datasets/movielens/