
salesforce
Implementing AutoML
Techniques at Salesforce Scale
Matthew Tovbin, Principal Engineer, Einstein
mtovbin@salesforce.com, @tovbinm

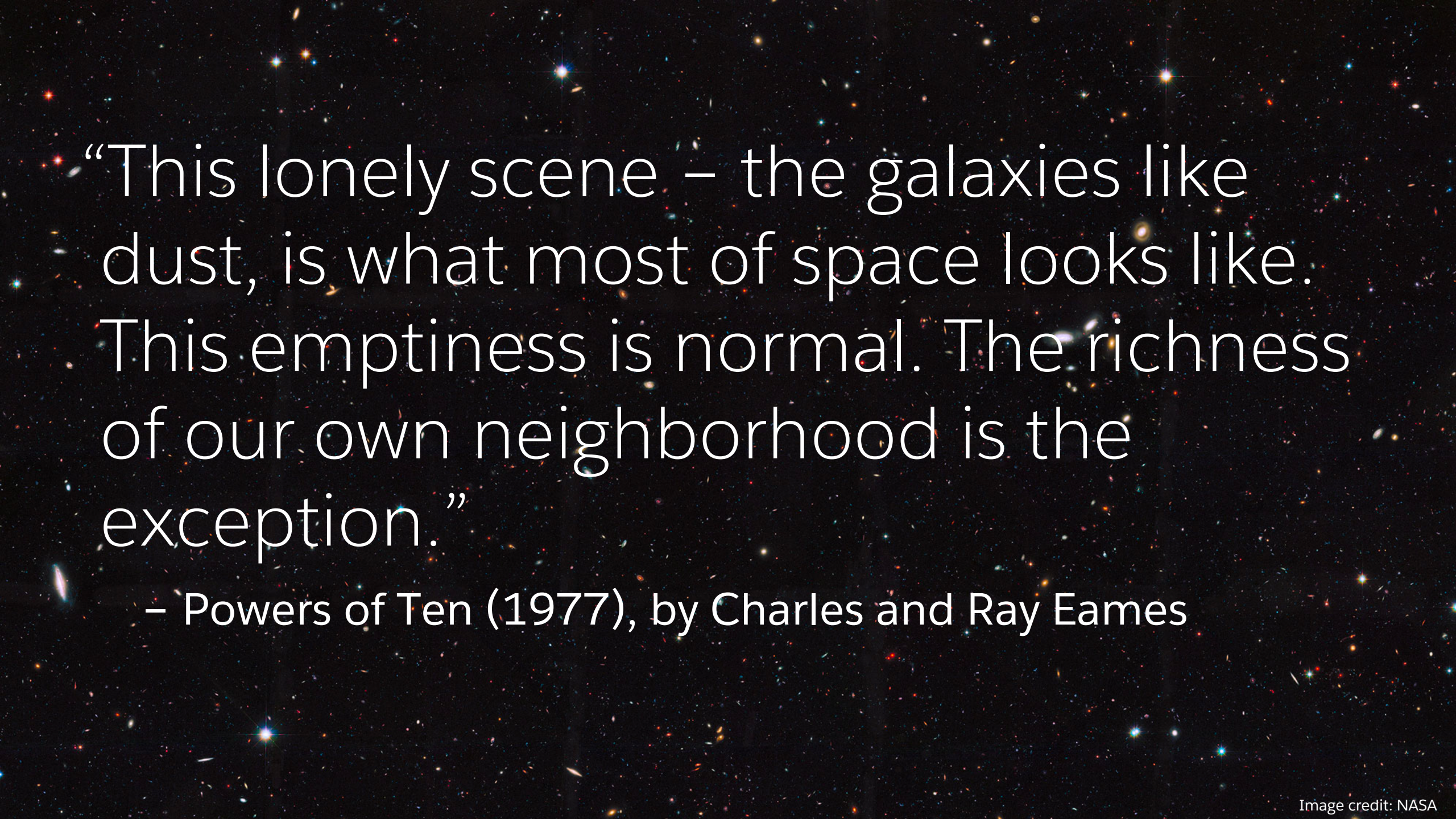
Forward Looking Statement

Statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.


“This lonely scene – the galaxies like dust, is what most of space looks like. This emptiness is normal. The richness of our own neighborhood is the exception.”
– Powers of Ten (1977), by Charles and Ray Eames
Image credit: NASA

Powers of Ten (1977)

A travel between a quark and the observable universe

$[10^{-17}, 10^{24}]$

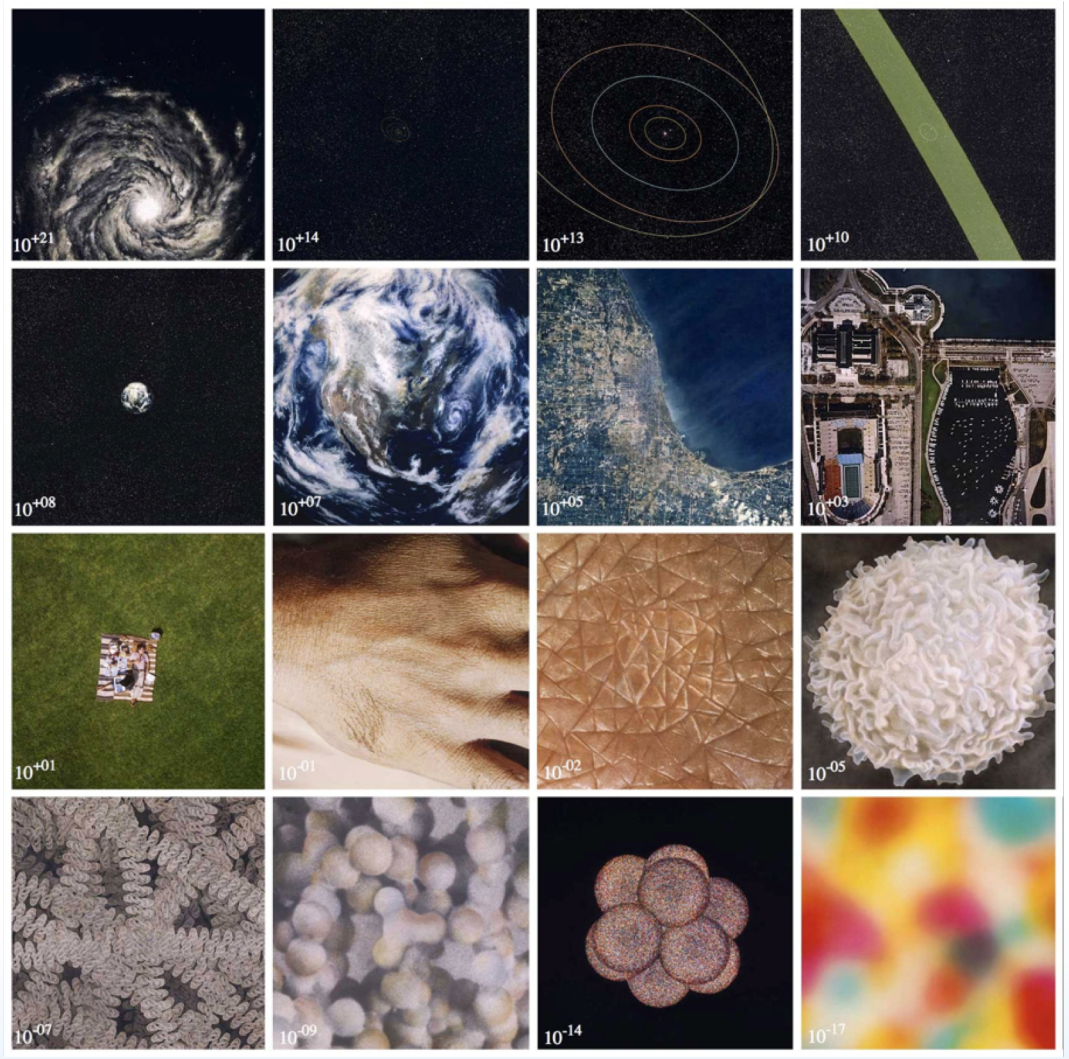


Image credit: Powers of Ten, Charles and Ray Eames

Powers of Ten for Machine Learning

- Data collection
- Data preparation
- Feature engineering
- Feature selection
- Sampling
- Algorithm implementation
- Hyperparameter tuning
- Model selection
- Model serving (scoring)
- Prediction insights
- Metrics

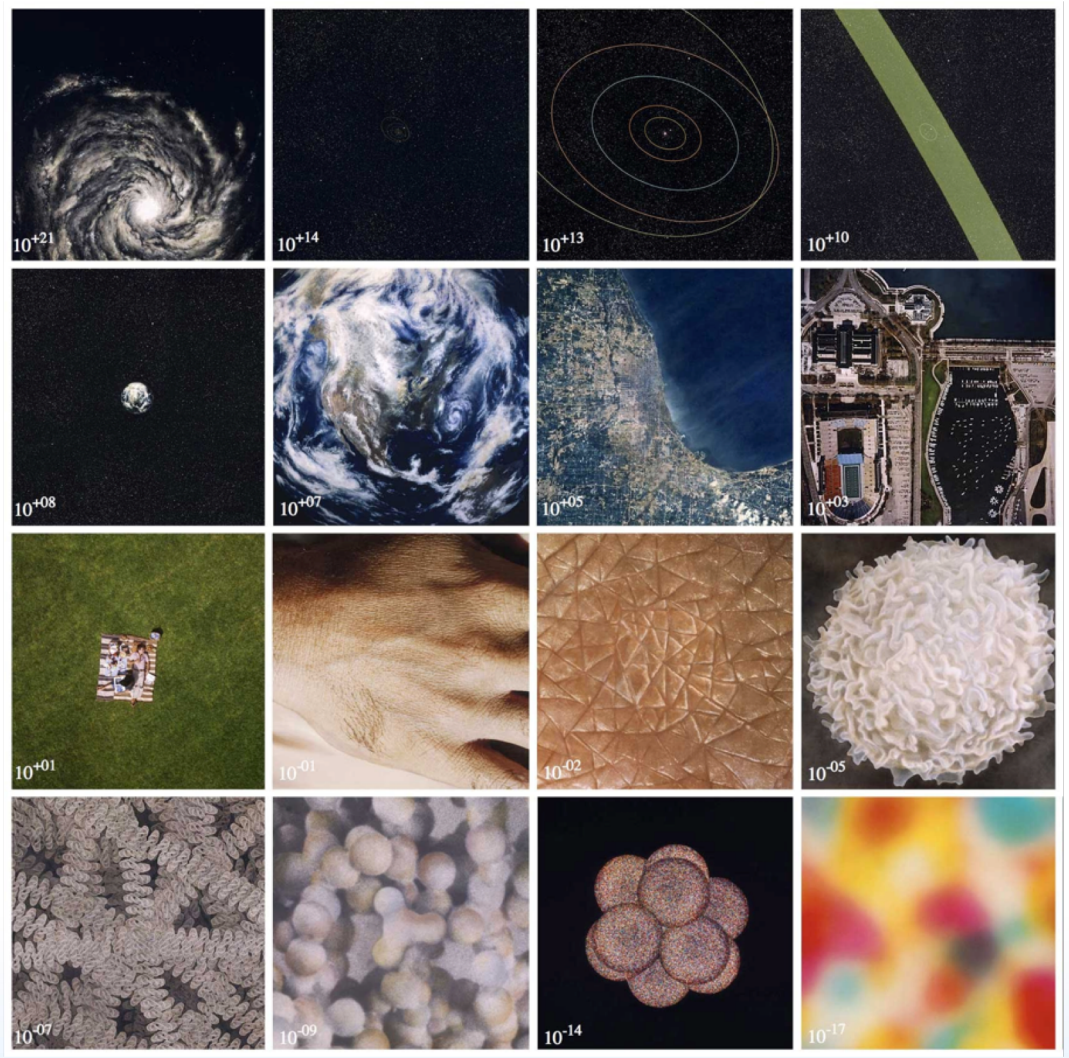


Image credit: Powers of Ten, Charles and Ray Eames

Steps for Building Machine Learning Application

1. Define requirements & outcomes
2. Collect & explore data
3. Define hypothesis
4. Prepare data
5. Build prototype
6. Verify hypothesis
7. Train & deploy model

Image credit: Bethesda Software

Multi-cloud and Multi-tenant

Make an intelligent guess

How long does it take to build a machine learning application?

a. Hours
b. Days
c. Weeks
d. Months
e. More

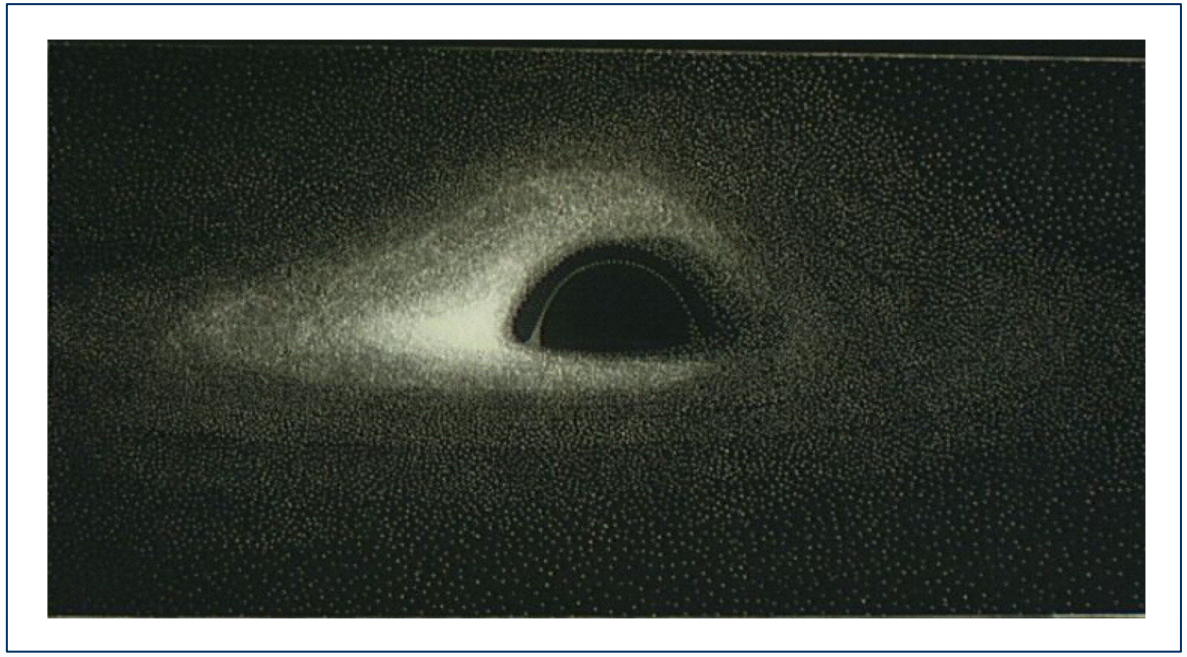
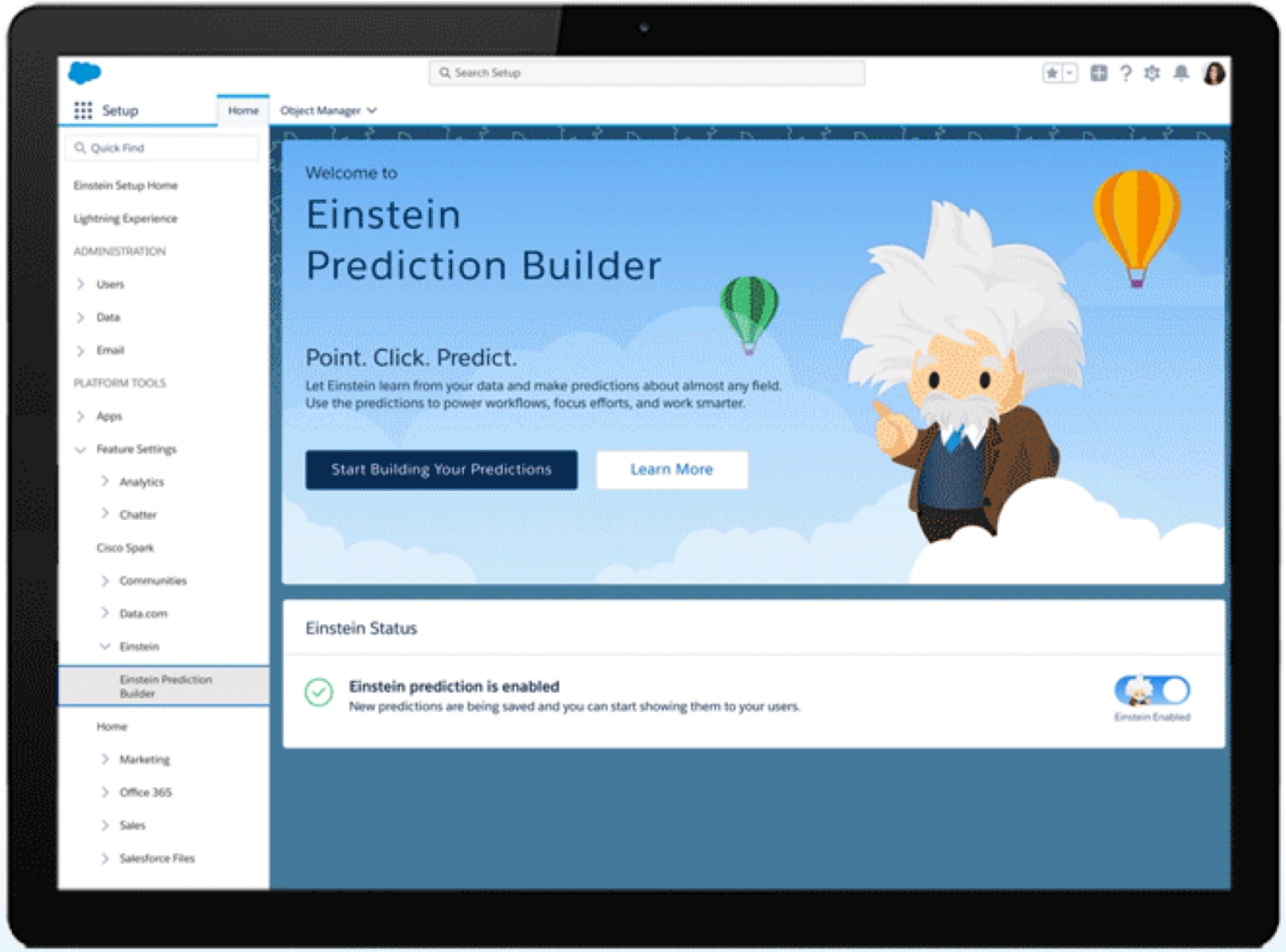


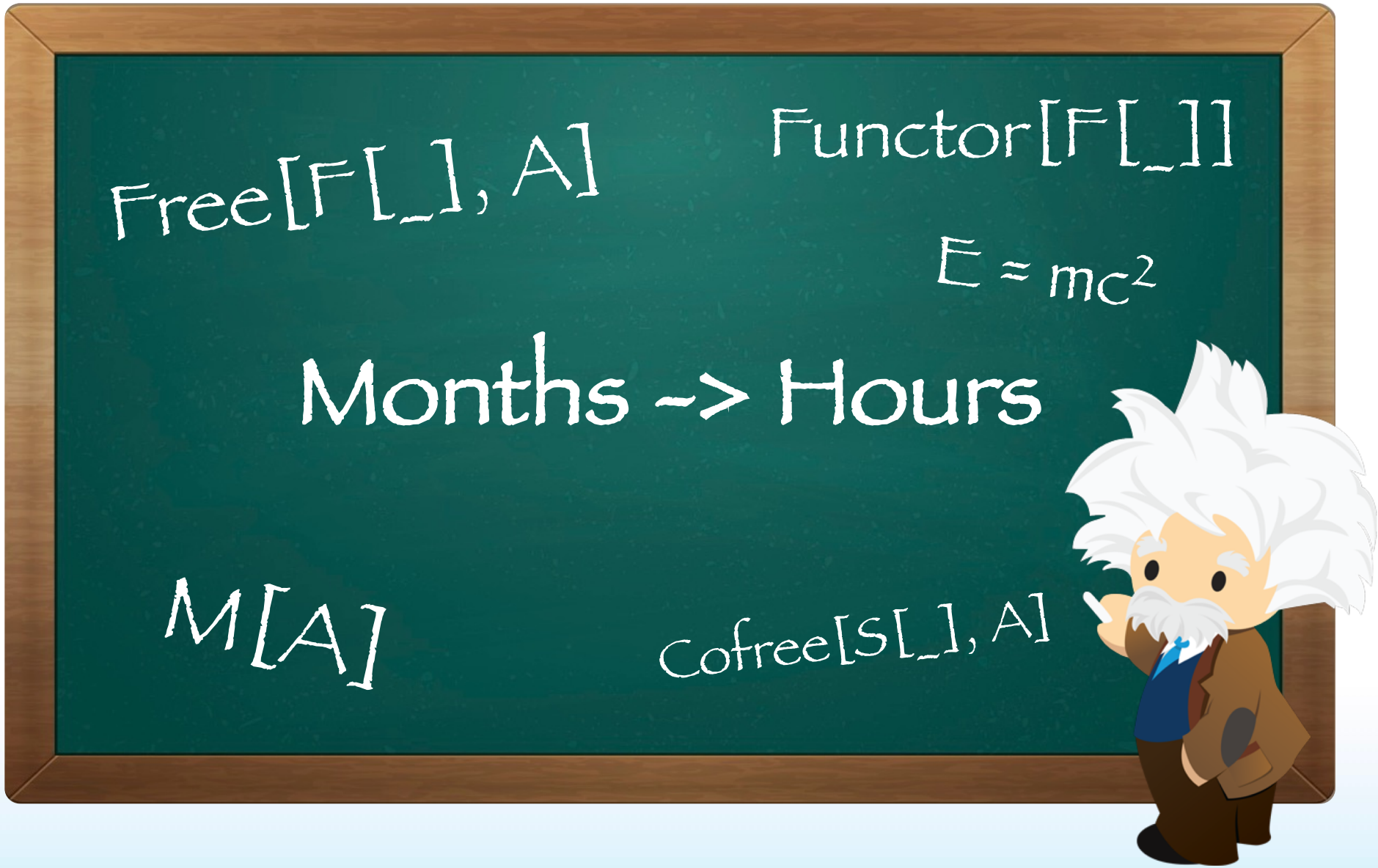
Image credit: NASA

Einstein Prediction Builder



- Product: Point. Click. Predict.
- Engineering: any customer can create any number of ML applications on any data, eh?!

How to cope with this complexity?



“The task of the software development team is to engineer the illusion of simplicity.”

– Grady Booch

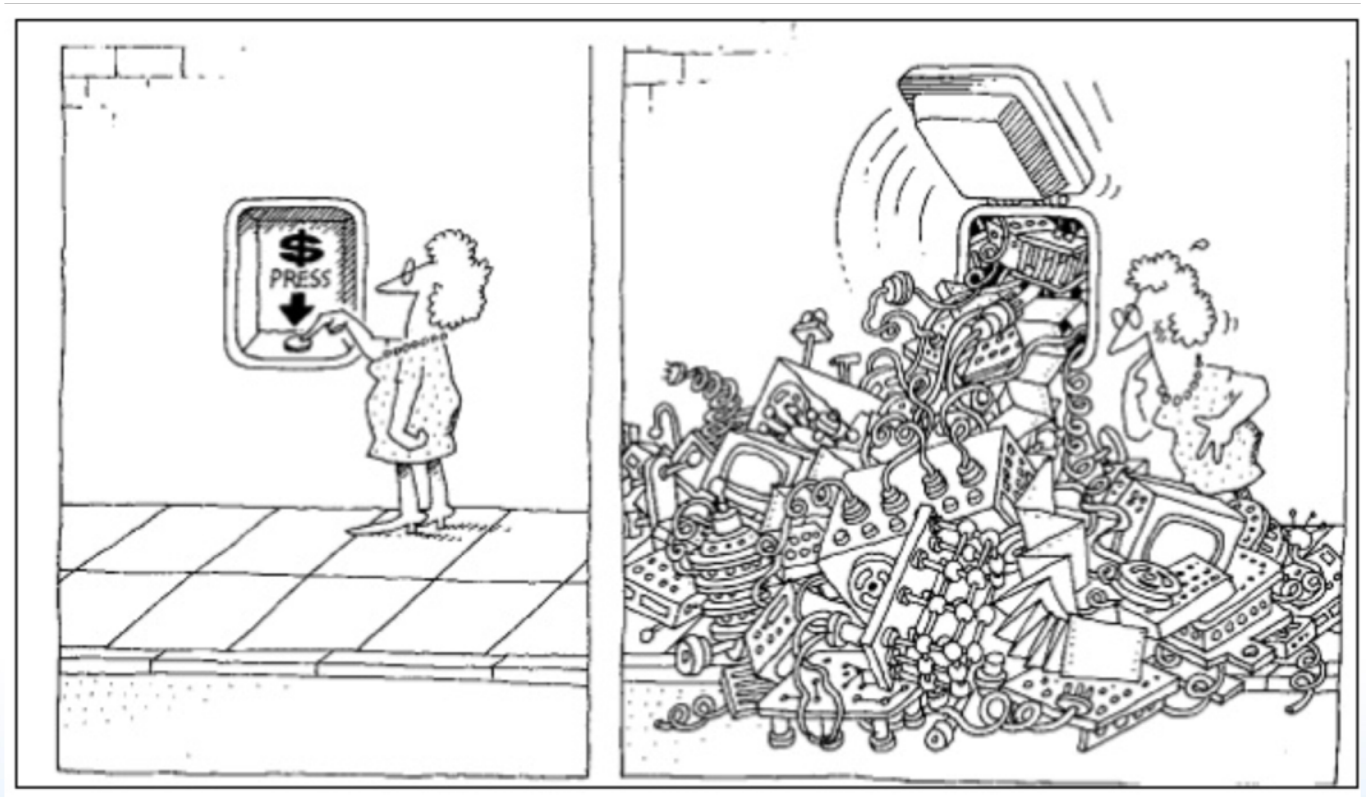
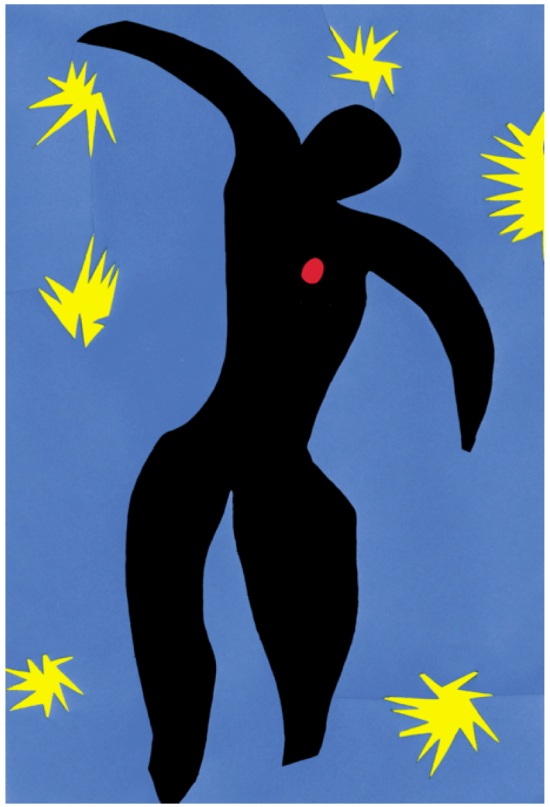


Image credit: Object-Oriented Analysis and Design with Applications, by Grady Booch

Complexity vs Abstraction



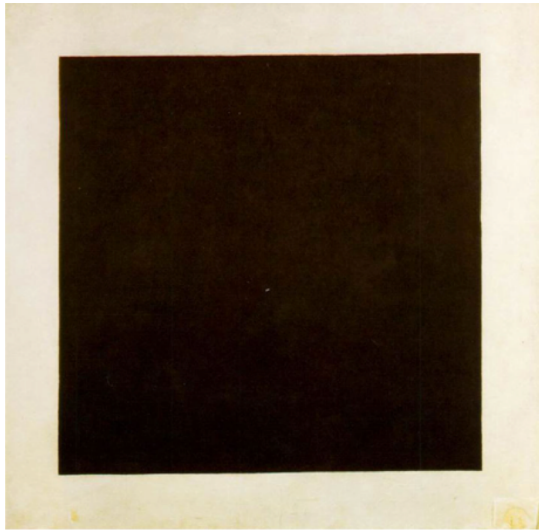
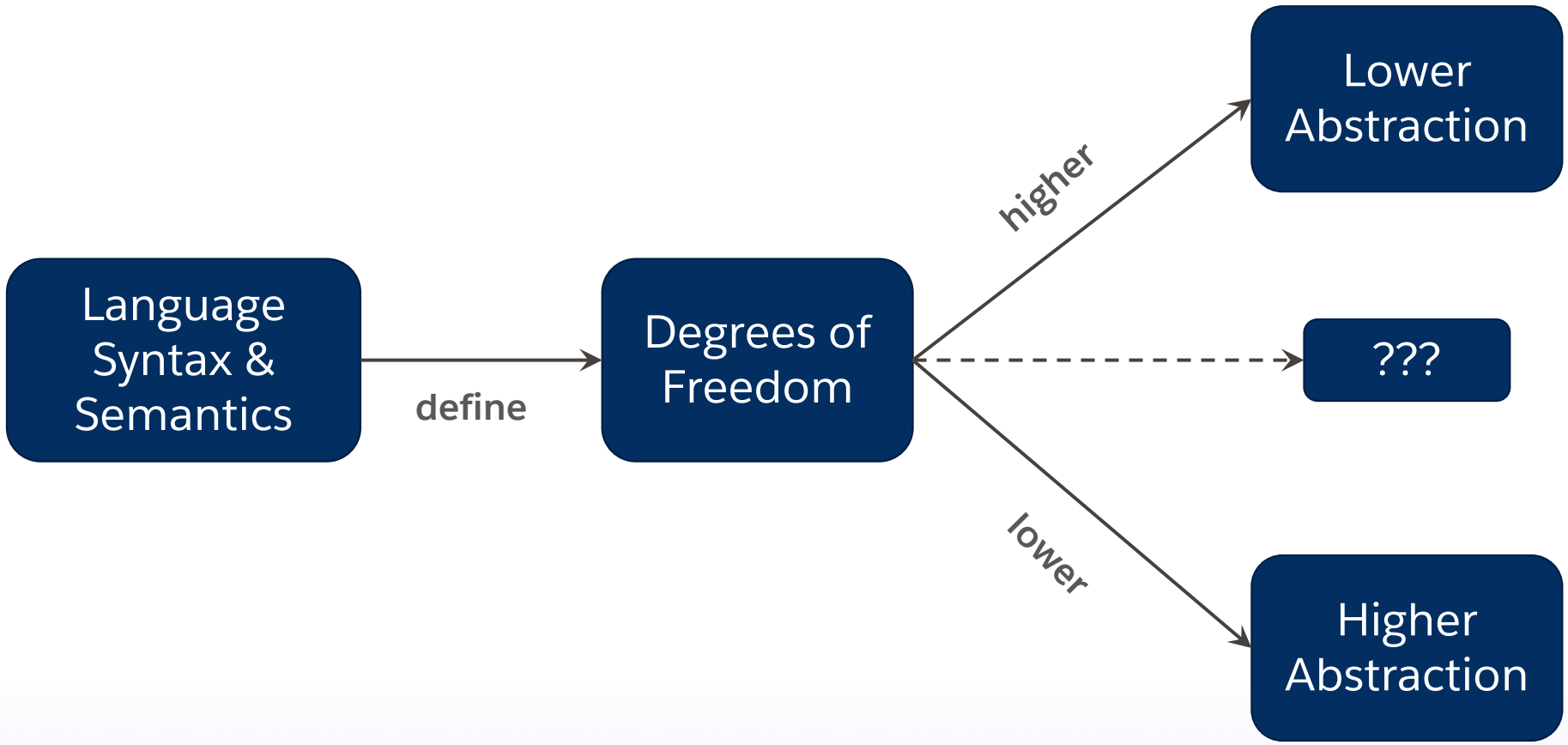


Image credit: Wikipedia

Appropriate Level of Abstraction



- Suitable for complex problems
- Difficult to use
- More complex
- Error prone

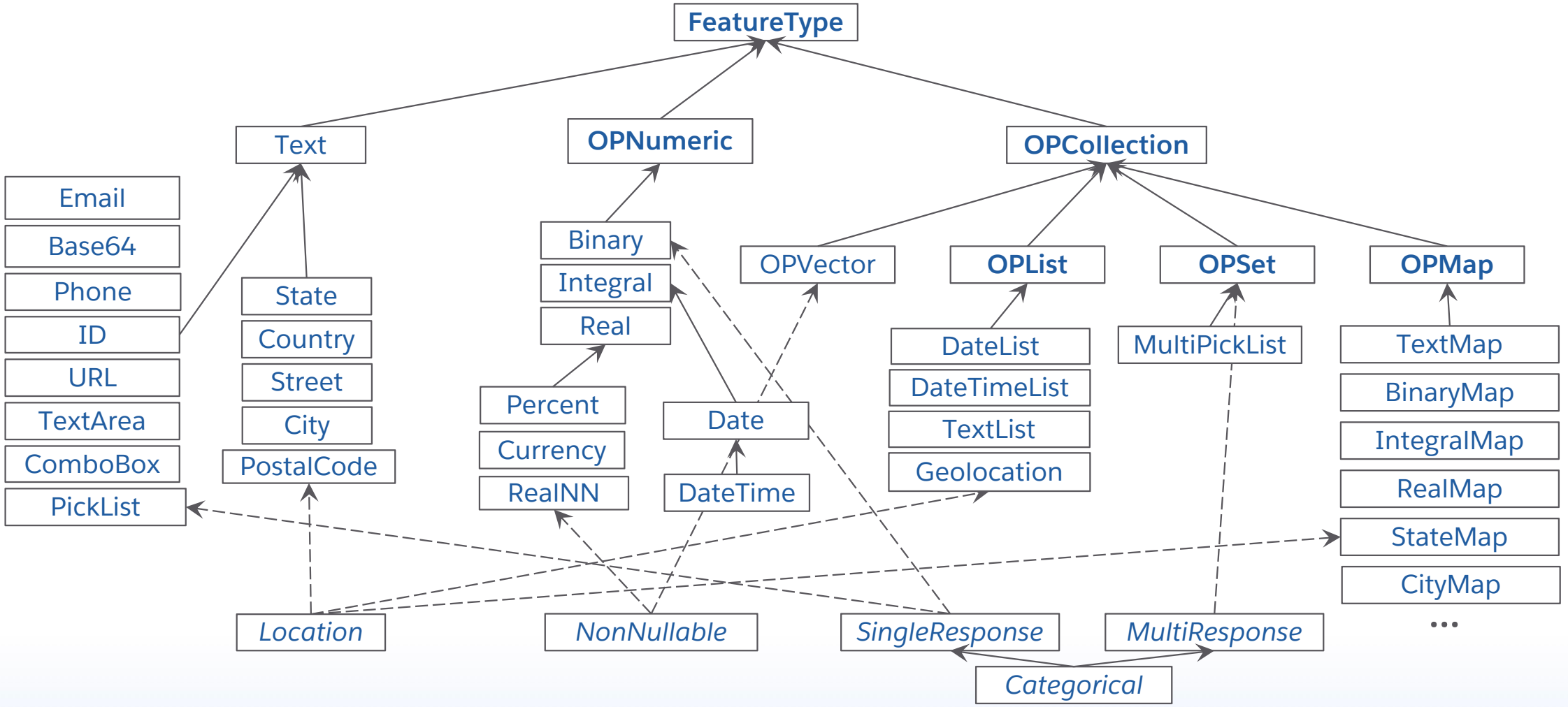
- Less flexible
- Simpler syntax
- Reuse

Codename: Optimus Prime

An AutoML library for building modular, reusable, strongly typed ML workflows on Spark

- Declarative & intuitive syntax
- Proper level of abstraction
- Aimed for simplicity & reuse
- >90% accuracy with 100X reduction in time

Types Hide the Complexity



Legend: **bold** - abstract type, regular - concrete type, *italic* - trait, solid line - inheritance, dashed line - trait mixin

https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/field_types.htm

Type Safety Everywhere

- Value operations
- Feature operations
- Transformation Pipelines (aka Workflows)

```
// Typed value operations
def tokenize(t: Text): TextList = t.map(_.split("")).toTextList

// Typed feature operations
val title = FeatureBuilder.Text[Book].extract(_.title).asPredictor
val tokens: Feature[TextList] = title.map(tokenize)

// Transformation pipelines
new OpWorkflow().setInput(books).setResultFeatures(tokens.vectorize())
```

Book Price Prediction


```
// Raw feature definitions: author, title, description, price
val authr = FeatureBuilder.PickList[Book].extract(_.author).asPredictor
val title = FeatureBuilder.Text[Book].extract(_.title).asPredictor
val descr = FeatureBuilder.Text[Book].extract(_.description).asPredictor
val price = FeatureBuilder.RealNN[Book].extract(_.price).asResponse

// Feature engineering: tokenize, tf-idf, vectorize, sanity check
val tokns = (title + description).tokenize(removePunctuation = true)
val tfidf = tokns.tf(numTerms = 1024).idf(minFreq = 0)
val feats = Seq(tfidf, author).transmogrify()
val chckd = feats.sanityCheck(price, removeBadFeatures = true)

// Model training: spin up spark, load books.csv, define model -> train
implicit val spark = SparkSession.builder.config(new SparkConf).getOrCreate
val books = spark.read.csv("books.csv").as[Book]
val preds = RegressionModelSelector().setInput(price, chckd).getOutput
new OpWorkflow().setInput(books).setResultFeatures(chckd, preds)
    .withRawFeatureFilter().train()
```

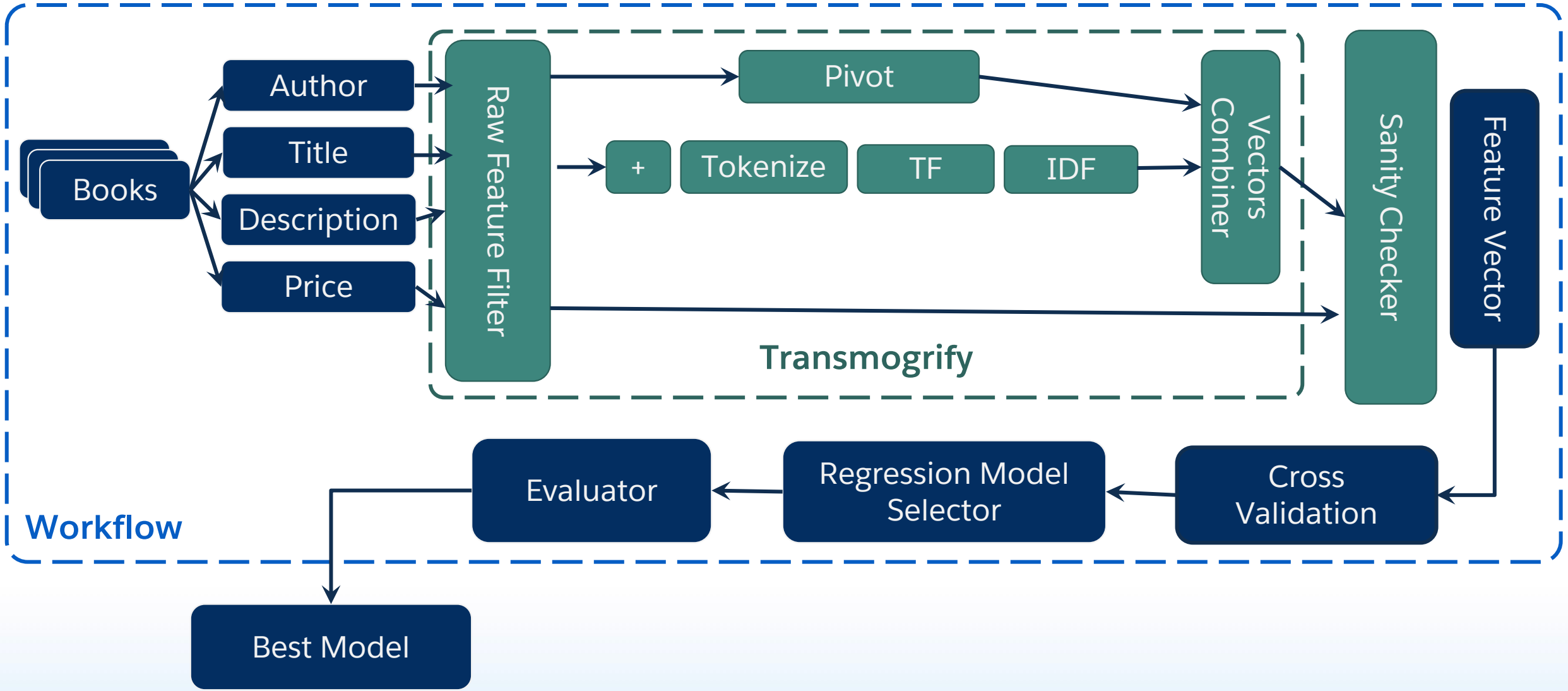
Book Price Prediction


```
// Raw feature definitions: author, title, description, price
val authr = FeatureBuilder.PickList[Book].extract(_.author).asPredictor
val title = FeatureBuilder.Text[Book].extract(_.title).asPredictor
val descr = FeatureBuilder.Text[Book].extract(_.description).asPredictor
val price = FeatureBuilder.RealNN[Book].extract(_.price).asResponse

// Feature engineering: tokenize, tf-idf, vectorize, sanity check
val tokns = (title + description).tokenize(removePunctuation = true)
val tfidf = tokns.tf(numTerms = 1024).idf(minFreq = 0)
val feats = Seq(tfidf, author).transmogrify() // <- magic here
val chckd = feats.sanityCheck(price, removeBadFeatures = true)

// Model training: spin up spark, load books.csv, define model -> train
implicit val spark = SparkSession.builder.config(new SparkConf).getOrCreate
val books = spark.read.csv("books.csv").as[Book]
val preds = RegressionModelSelector().setInput(price, chckd).getOutput
new OpWorkflow().setInput(books).setResultFeatures(chckd, preds)
    .withRawFeatureFilter().train()
```

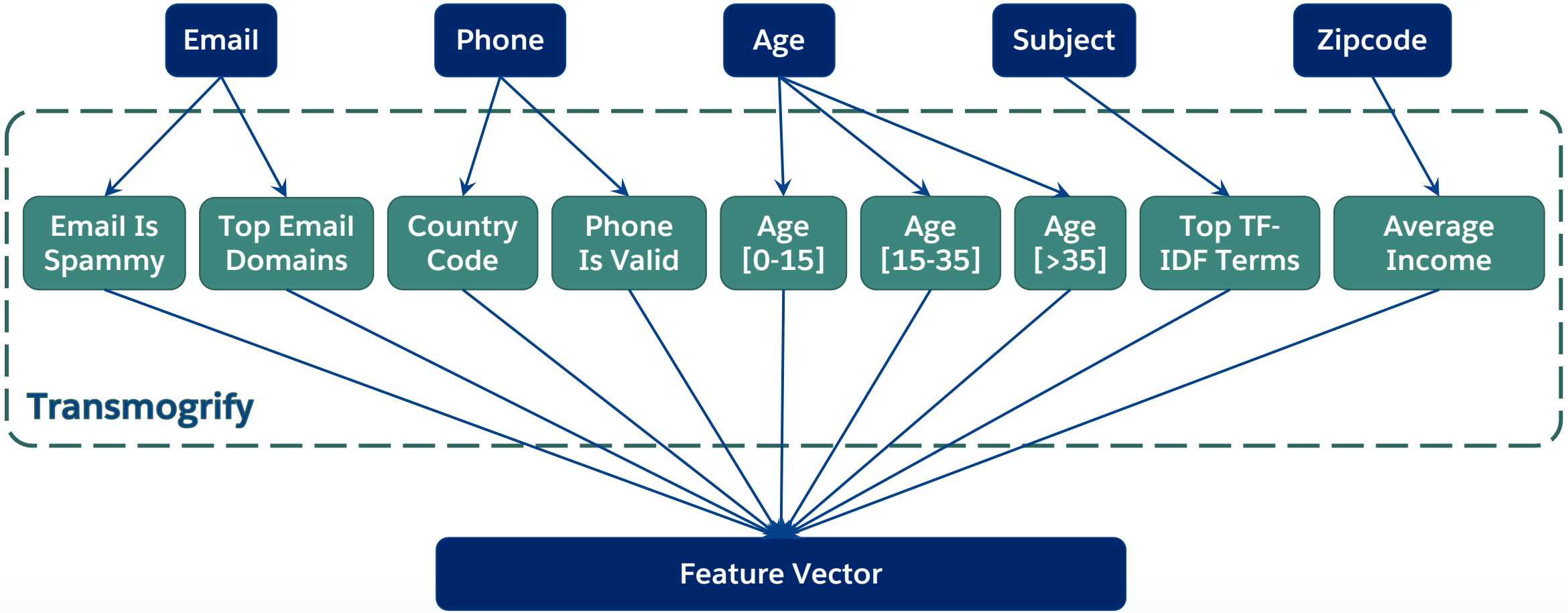
Book Price Prediction



AutoML with Optimus Prime

- Automated Feature Engineering
- Automated Feature Selection
- Automated Model Selection

Automatic Feature Engineering



Automatic Feature Engineering

Numeric	Categorical	Text	Temporal	Spatial
Imputation	Imputation	Tokenization	Time difference	Augment with external data e.g avg income
Track null value	Track null value	Hash Encoding	Circular Statistics	Spatial fraudulent behavior e.g: impossible travel speed
Log transformation for large range	One Hot Encoding	TF-IDF	Time extraction (day, week, month, year)	Geo-encoding
Scaling - zNormalize	Dynamic Top K pivot	Word2Vec	Closeness to major events	
Smart Binning	Smart Binning	Sentiment Analysis		
	LabelCount Encoding	Language Detection		
	Category Embedding			

Metadata!

- The name of the feature the column was made from
- The name of the RAW feature(s) the column was made from
- Everything you did to get the column
- Any grouping information across columns
- Description of the value in the column

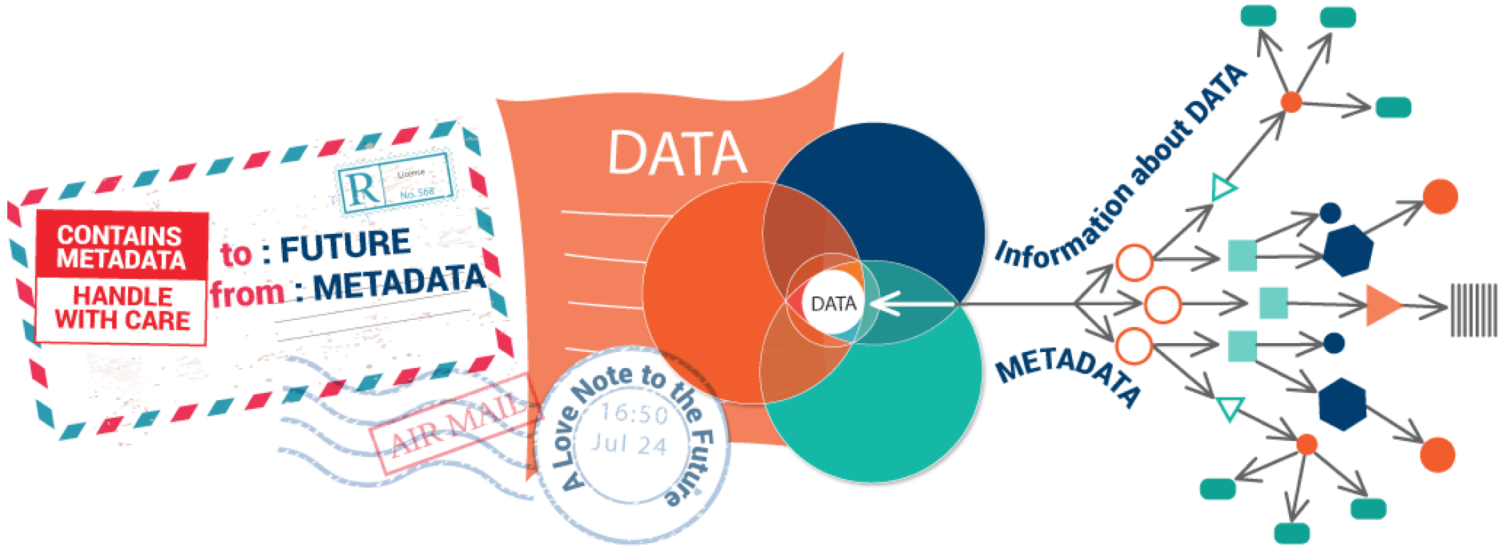


Image credit: https://ontotext.com/knowledgehub/fundamentals/metadata-fundamental

Automatic Feature Selection

Analyze features & calculate statistics

- Min, max, mean, variance
- Correlations with label
- Contingency matrix
- Confidence & support
- Mutual information against label
- Point-wise mutual information against all label values
- Cramér's V
- Etc.

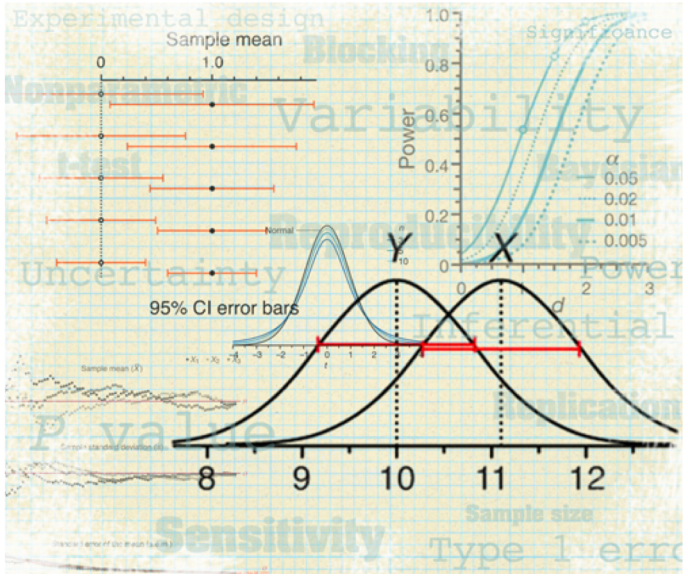


Image credit: http://csbiology.com/blog/statistics-for-biologists-nature-com

Automatic Feature Selection

- Ensure features have acceptable ranges
- Is this feature a leaker?
- Does this feature help our model?
- Is it predictive?

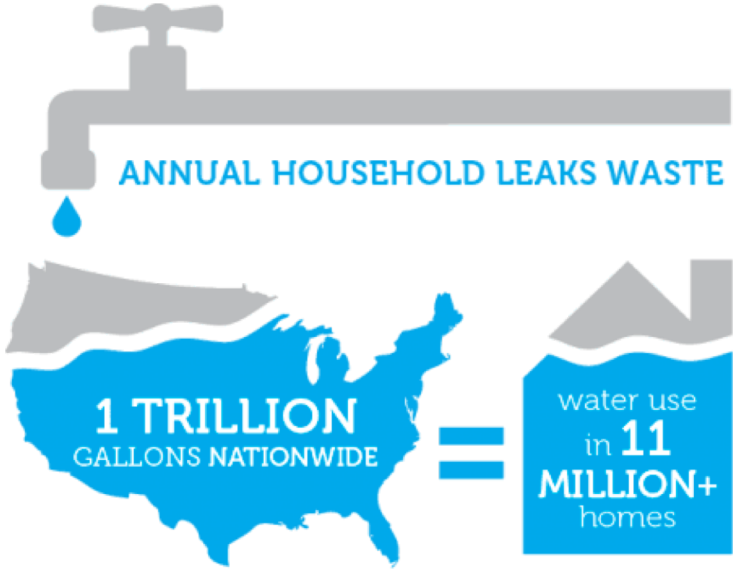


Image credit: ://srcity.org/2252/Find-Fix-Leaks

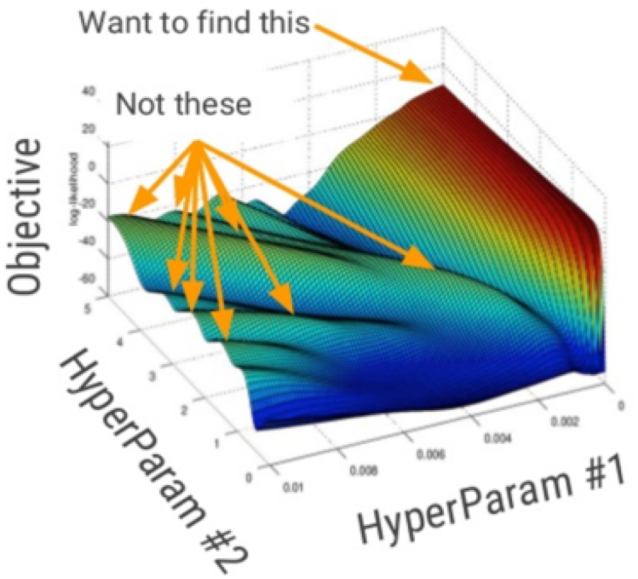
Automatic Feature Selection


```
// Sanity check your features against the label
val checked = price.sanityCheck(
  featureVector = feats,
  checkSample = 0.3,
  sampleSeed = 42L,
  sampleLimit = 100000L,
  maxCorrelation = 0.95,
  minCorrelation = 0.0,
  correlationType = Pearson,
  minVariance = 0.00001,
  removeBadFeatures = true
)
new OpWorkflow().setInput(books).setResultFeatures(checked, preds).train()
```

```
17/09/27 17:33:09 INFO SanityChecker: Feature (embarked)_vecSet_OTHER has: mean = 0.0, min = 0.0, max = 0.0, variance of 0.0, number of nulls = 0.0
17/09/27 17:33:09 INFO SanityChecker: Feature ((age)_map_022)_vecSet_Adult has: mean = 0.6419753086419753, min = 0.0, max = 1.0, variance of 0.23271604938271606, number of nulls = 0.0
17/09/27 17:33:09 INFO SanityChecker: Feature ((age)_map_022)_vecSet_Child has: mean = 0.35802469135802467, min = 0.0, max = 1.0, variance of 0.23271604938271606, number of nulls = 0.0
17/09/27 17:33:09 INFO SanityChecker: Feature ((age)_map_022)_vecSet_OTHER has: mean = 0.0, min = 0.0, max = 0.0, variance of 0.0, number of nulls = 0.0
17/09/27 17:33:09 INFO SanityChecker: Feature (sibSp)_vecInt has: mean = 0.4814814814814815, min = 0.0, max = 3.0, variance of 0.627777777777778, number of nulls = 0.0
17/09/27 17:33:09 INFO SanityChecker: Feature (sibSp)_vecInt_null has: mean = 0.0, min = 0.0, max = 0.0, variance of 0.0, number of nulls = 0.0
17/09/27 17:33:09 INFO SanityChecker: Feature (parCh)_vecInt has: mean = 0.4320987654320987, min = 0.0, max = 5.0, variance of 0.9234567901234566, number of nulls = 0.0
```

Automatic Model Selection

- Multiple algorithms to pick from
- Many hyperparameters for each algorithm
- Automated hyperparameter tuning
  - Faster model creation with improved metrics
  - Search algorithms to find the optimal hyperparameters. e.g. grid search, random search, bandit methods

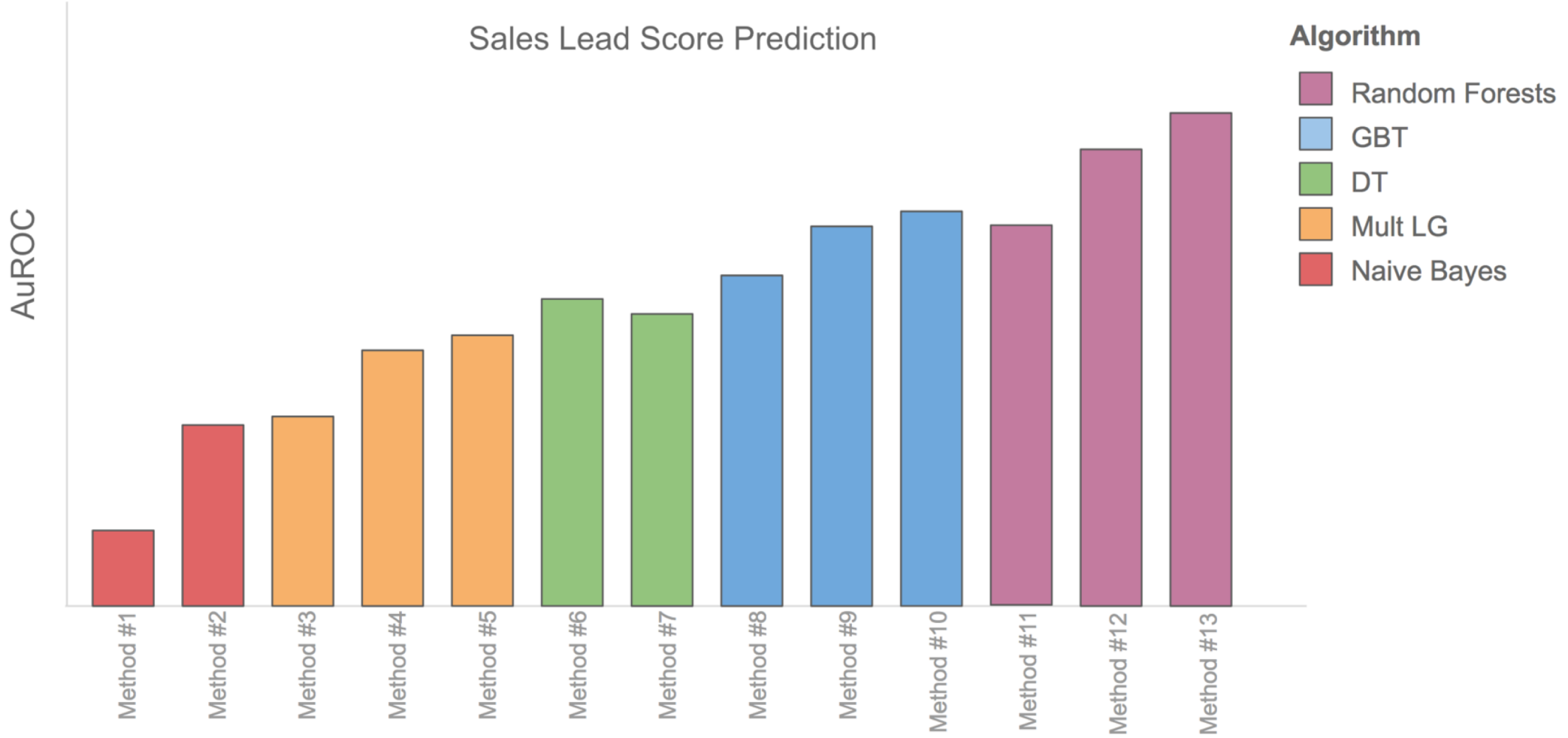


Automatic Model Selection


```
// Model selection and hyperparameter tuning
val preds =
  RegressionModelSelector
   .withCrossValidation(
      dataSplitter = DataSplitter(reserveTestFraction = 0.1),
      numFolds = 3,
      validationMetric = Evaluators.Regression.rmse(),
      trainTestEvaluators = Seq.empty,
      seed = 42L)
   .setModelsToTry(LinearRegression, RandomForestRegression)
   .setLinearRegressionElasticNetParam(0, 0.5, 1)
   .setLinearRegressionMaxIter(10, 100)
   .setLinearRegressionSolver(Solver.LBFGS)
   .setRandomForestMaxDepth(2, 10)
   .setRandomForestNumTrees(10)
   .setInput(price, checked).getOutput

new OpWorkflow().setInput(books).setResultFeatures(checked, preds).train()
```

Automatic Model Selection



Demo

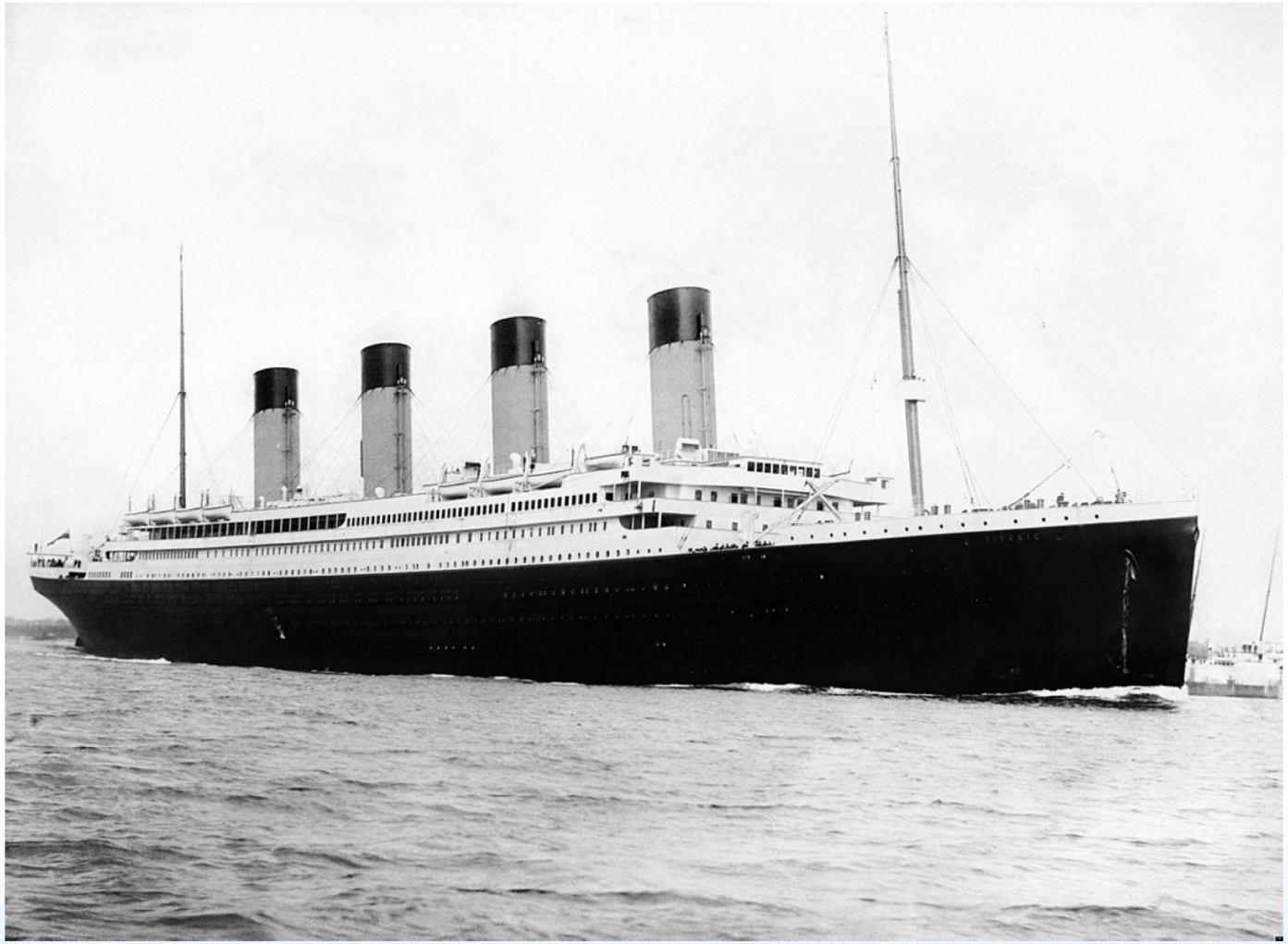
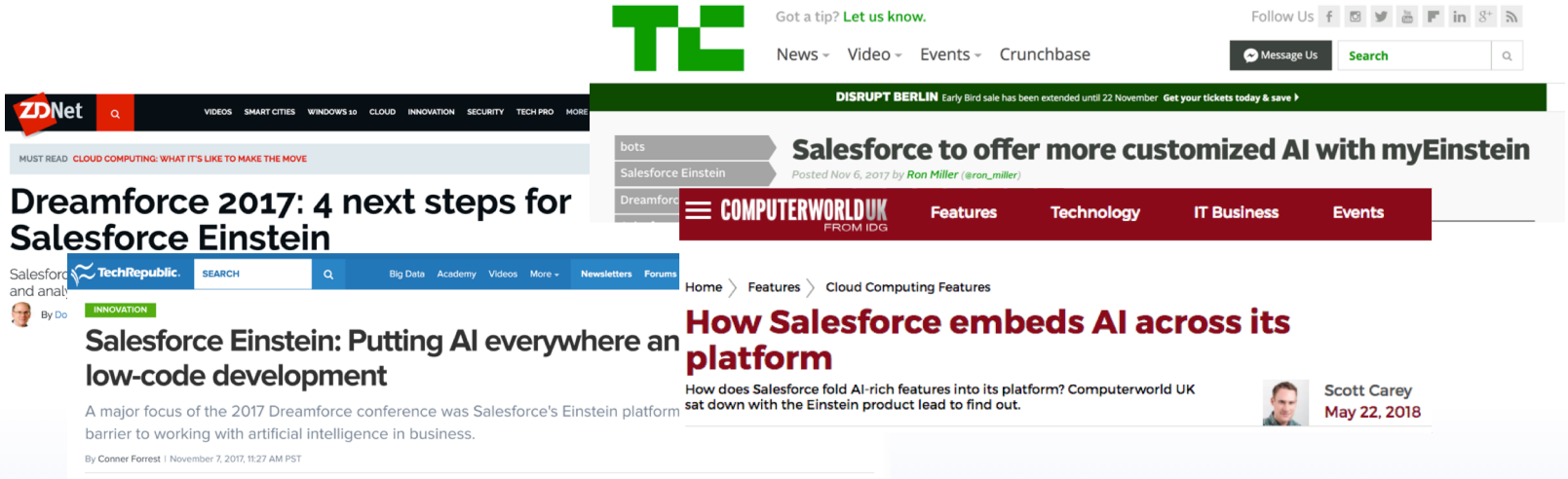


Image credit: Wikipedia

How well does it work?

- Most of our models deployed in production are **completely** hands free
- **2B+** predictions per day



Takeaway

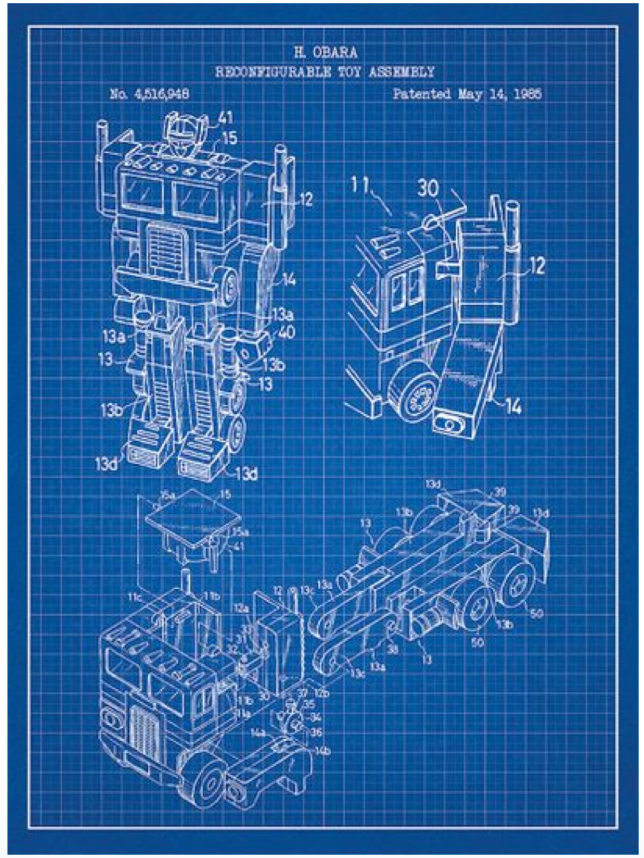
- Define appropriate level of abstraction
- Use types to express it
- Automate everything
  - feature engineering & selection
  - model selection & evaluation
  - Etc.

Months -> Hours

Join Optimus Prime pilot today!

https://sfdc.co/join-op-pilot

- Open for pilot
- In production at Salesforce
- Piloted by large tech companies
- 100% Scala



Talks for Further Exploration

- “When all the world’s data scientists are just not enough”, Shubha Nabar, @Scale 17
- “Low touch Machine Learning”, Leah McGuire, Spark Summit 17
- “Fantastic ML apps and how to build them”, Matthew Tovbin, Scale By The Bay 17
- “The Black Swan of perfectly interpretable models”, Leah McGuire and Mayukh Bhaowal, QCon AI 18


THANK YOU