ModelDB: A system to manage machine learning models

Manasi Vartak
PhD Student, MIT DB Group

Database Group
MIT Computer Science and Artificial Intelligence Lab

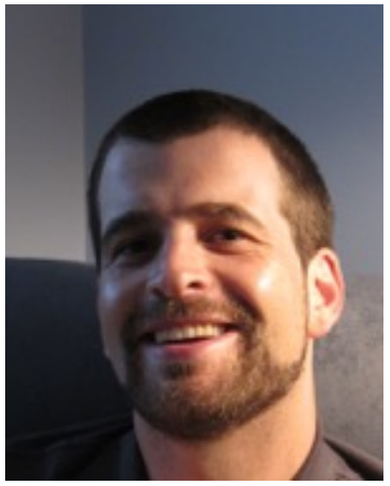
People

Manasi Vartak
PhD student, MIT

Harihar Subramanyam
MEng, MIT

Wei-En Lee
MEng student, MIT

Srinidhi Viswanathan
MEng, MIT



Samuel Madden
Faculty, MIT

Matei Zaharia
Faculty, Stanford

Building a default prediction algorithm

		Profession	Credit History	Risk of Default
	Barack Obama	Politician	Reasonable	0.3
	Lindsay Lohan	Struggling artist	Poor	0.7
	Warren Buffet	Investor	Has more money than our company	0.0
	…	…	…	…

Model 1

```
val path = "data/credit-default.csv"
val df = spark
  .read
  .option("header", true)
  .option("inferSchema", true)
  .csv(path)

val assembler = new VectorAssembler()
  .setInputCols(Array("LIMIT_BAL", "SEX",
    "EDUCATION", "MARRIAGE", "AGE"))
  .setOutputCol("features")

val transformedDf = assembler.transform(df)

val logReg = new LogisticRegression()
  .setLabelCol("DEFAULT")
```

Accuracy: 62%

Model 3

```
val path = "data/credit-default.csv"
val df = spark
  .read
  .option("header", true)
  .option("inferSchema", true)
  .csv(path)

val assembler = new VectorAssembler()
  .setInputCols(Array("LIMIT_BAL", "SEX",
    "EDUCATION", "MARRIAGE", "AGE"))
  .setOutputCol("features")

val transformedDf = assembler.transform(df)

val logReg = new ~~LogisticRegression~~() RandomForestClassifier()
  .setLabelCol("DEFAULT")
```

```
val udf1: (Int => Int) = (delayed..)
df.withColumn("timesDelayed", udf1)
```

Model 5

```
val path = "data/credit-default.csv"
val df = spark
  .read
  .option("header", true)
  .option("inferSchema", true)
  .csv(path)

val assembler = new VectorAssembler()
  .setInputCols(Array("LIMIT_BAL", "SEX",
    "EDUCATION", "MARRIAGE", "AGE"))
  .setOutputCol("features")

val transformedDf = assembler.transform(df)

val logReg = new LogisticRegression()
  .setLabelCol("DEFAULT")
```

credit-default-clean.csv

df.withColumn("timesDelayed", udf1)
.withColumn("percentPaid", udf2)

RandomForestClassifier

```
val lrGrid = new ParamGridBuilder()
    .addGrid(rf.maxDepth, Array(5, 10, 15))
    .addGrid(rf.numTrees, Array(50, 100))
```

Model 50

```
val labelIndexer1 = new LabelIndexer()
val labelIndexer2 = new LabelIndexer()
```

```
val path = "data/credit-default.csv"
val df = spark
  .read
  .option("header", true)
  .option("inferSchema", true)
  .csv(path)

val assembler = new VectorAssembler()
  .setInputCols(Array("LIMIT_BAL", "SEX",
   "EDUCATION", "MARRIAGE", "AGE"))
  setOutputCol("features")

val transformedDf = assembler.transform(df)

val logReg = new LogisticRegression()
  .setLabelCol("DEFAULT")
```

credit-default-clean.csv

```
val udf1: (Int => Int) = (delayed..)
val udf2: (String, Int) = ...

df.withColumn("timesDelayed", udf1)
  .withColumn("percentPaid", udf2)
  .withColumn("creditUsed", udf3)
  ...

val scaler = new StandardScaler()
  .setInputCol("features")
  ...

val lrGrid = new ParamGridBuilder()
    .addGrid(lr.elasticNetParam, Array(0.01, 0.1, 0.5, 0.7))
```

I’m willing to bet…

No one in here tracks (all of) their models

…and this is not unusual

Why is this a problem?

- No record of experiments
- Insights lost along the way
- Difficult to reproduce results
- Cannot search for or query models
- Difficult to collaborate

Did my colleague do that already?

How did normalization affect my ROC?

What params did I use?

Where's the LR model I tried last week with featureX?

How does someone review your model?

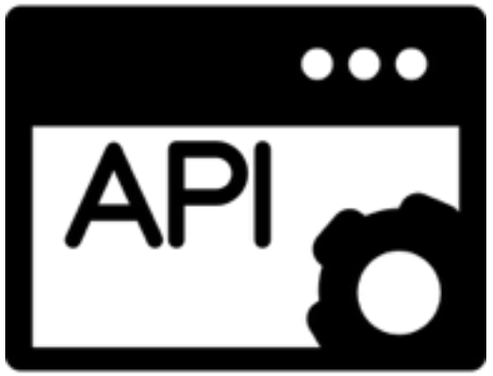
Model Management

track, store and index modeling artifacts so that they may subsequently be reproduced, shared, queried, and analyzed

ModelDB: a system to manage machine learning models

http://modeldb.csail.mit.edu

ModelDB: an end-to-end model management system



Ingest models, metadata

track

Model artifact Storage & Versioning

store & index

Query

Collaboration, Reproducibility

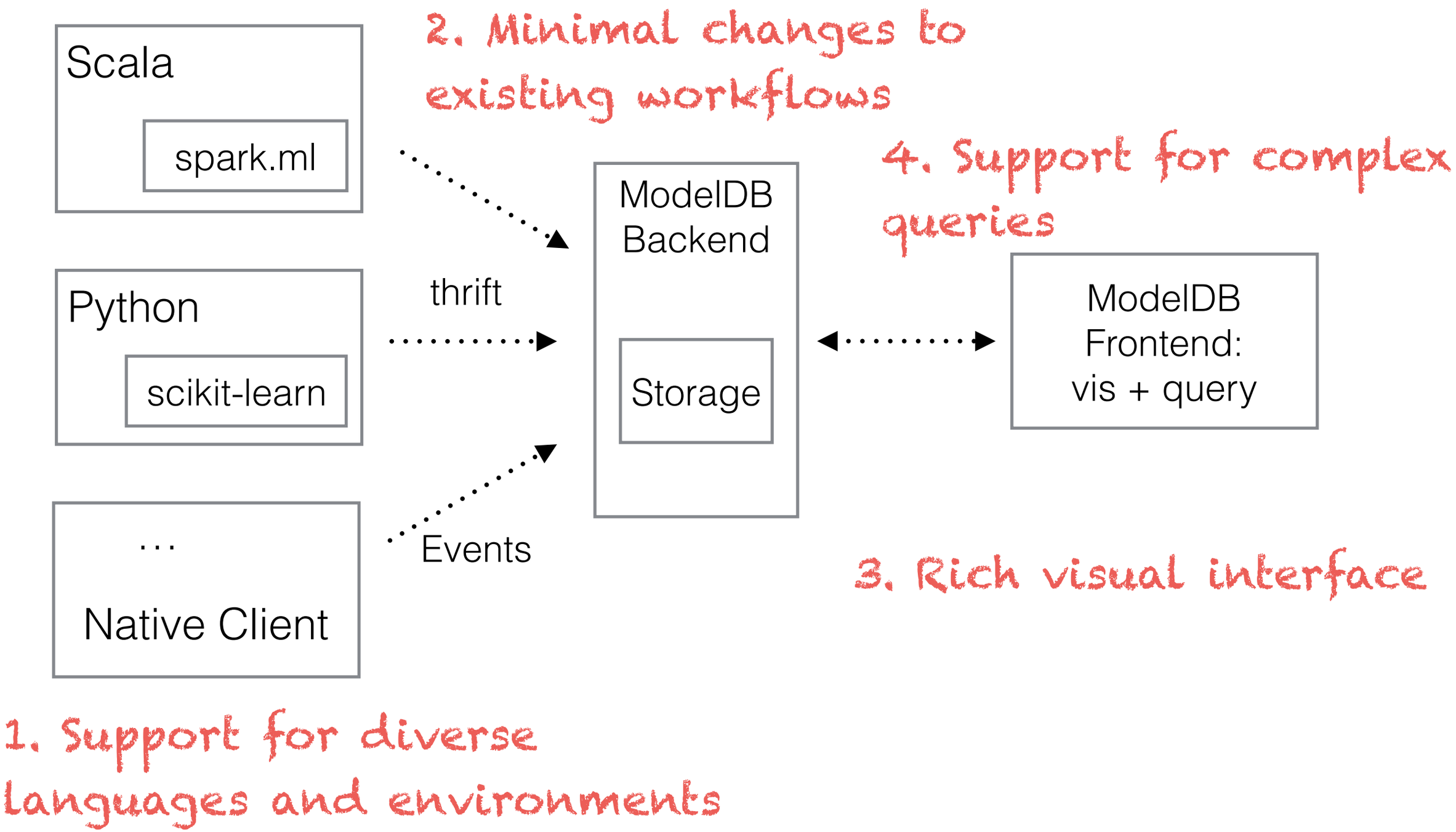
query, reproduce++

Demo

```
df = pd.read_csv(DATA_PATH + 'credit-default.csv', skiprows=[0])

## modeldb start
# .read_csv_sync(DATA_PATH + 'credit-default.csv', skiprows=[0])
## modeldb end

target = df['default payment next month']
df = df[["LIMIT_BAL", "SEX", "EDUCATION", "MARRIAGE", "AGE"]]

x_train, x_test, y_train, y_test = cross_validation.train_test_split(df, target, test_size=0.3)

## modeldb start
# .train_test_split_sync(df, target, test_size=0.3)
## modeldb end

lr = linear_model.LogisticRegression()

lr.fit(x_train, y_train)
## modeldb start
# .fit_sync(x_train, y_train)
## modeldb end

y_pred = lr.predict(x_test)
## modeldb start
# .predict_sync(x_test)
## modeldb end
```

ModelDB w/ scikit-learn

ModelDB Architecture & Design Decisions



ModelDB Features

- Experiment tracking
  Log models, params, pipelines etc. via ModelDB API
- Versioning
  Every modeling run = version
- Reproducibility
  All pipeline details, params logged
- Comparisons, queries, search
  Model search, query, comparison via frontend
- Collaboration
  Central repository of models
  Review models, annotate

Ongoing Work

- Unified querying of modeling artifacts
- Mining data in ModelDB
- Model monitoring and retraining

ModelDB available now!

mitdbg / modeldb

Watch 14 | Star 0 | Fork 0

Code | Issues 4 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs

A system to manage machine learning models http://modeldb.csail.mit.edu/

machine-learning mit model-management modeldb

743 commits | 2 branches | 0 releases | 5 contributors | MIT

Branch: master | New pull request | Find file | Clone or download

sviswana committed on GitHub Merge pull request #190 from mitdbg/add-samples ... Latest commit c10e592 an hour ago

client	Update SimpleSample.scala	an hour ago
config	Update .mdb_config	3 hours ago
data	remove modified csv file	23 hours ago
dockerbuild	Update README.md	4 hours ago

**MIT License*

http://modeldb.csail.mit.edu

ModelDB available now!

- Download, try it out!

- Tell us what you think; what can we do better?
- Contribute! (see Issues on repo for some ideas)

ModelDB: a system to manage machine learning models

http://modeldb.csail.mit.edu

mvartak@csail.mit.edu | **@DataCereal**