Microsoft

#msdevcon

DevCon School

Технологии будущего

DevCon School
Технологии будущего

CODE.INSTINCT

Machine Learning with Microsoft Azure

#msdevcon

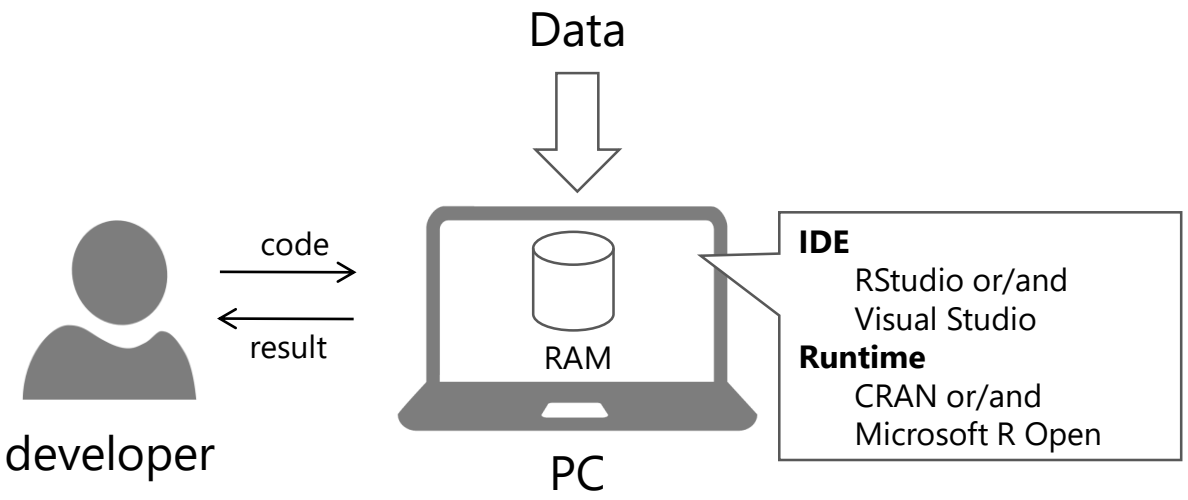
$\langle \Omega, \mathfrak{U}, \mathbb{P} \rangle$

Dmitry Petukhov,

ML/DS Preacher, Coffee Addicted &&
Machine Intelligence Researcher @ OpenWay

MVP
Microsoft® Most Valuable Professional

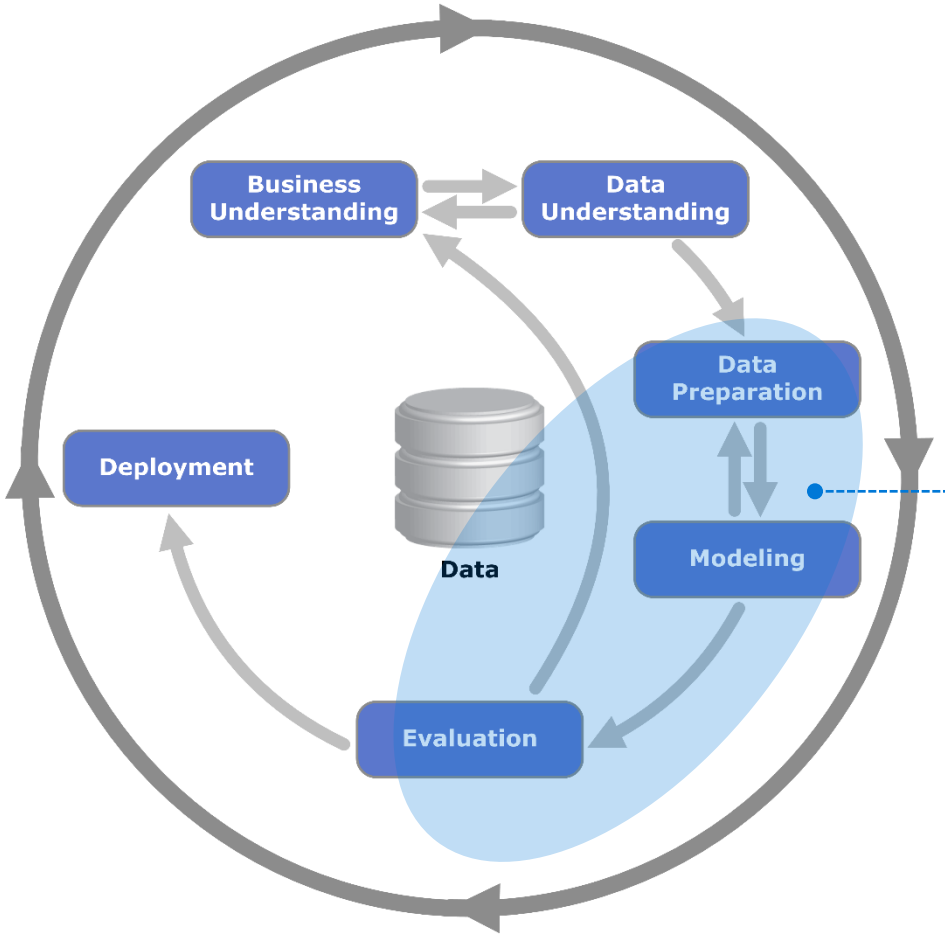
R for ~~Fun~~ Prototyping



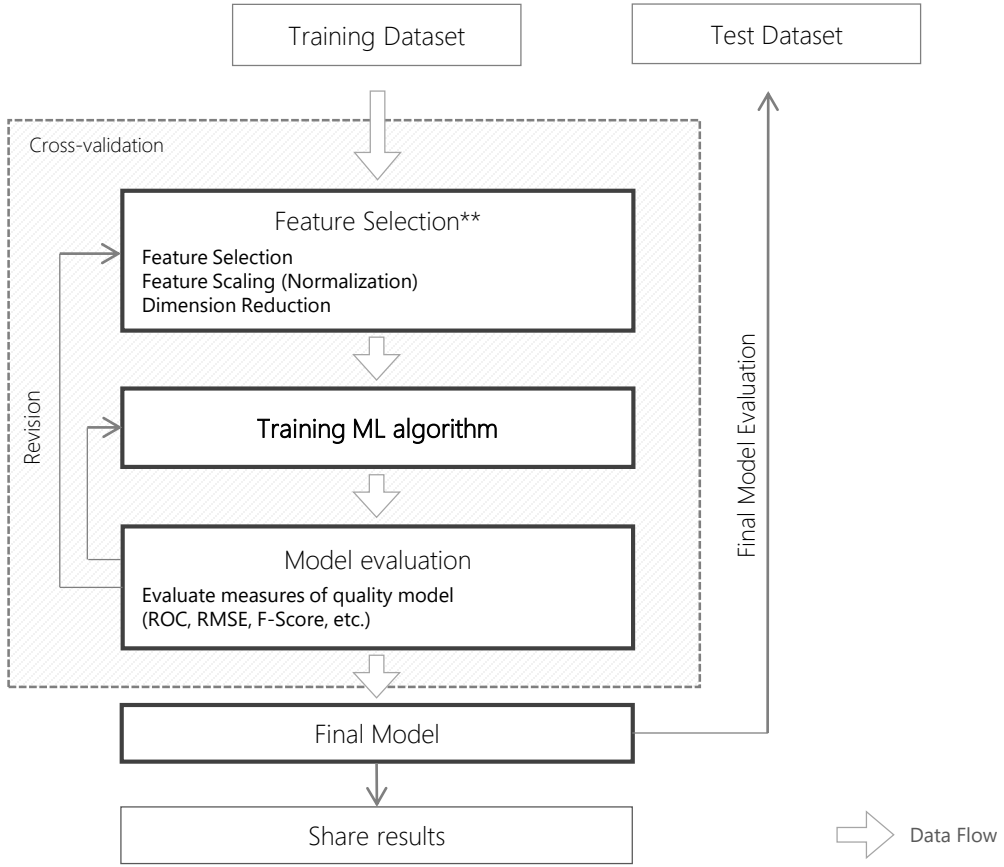
Flexibility ~~Distributed~~ Scalable: ~~horizontal~~, vertical ~~Fault-tolerance~~ ~~Reliable~~

OSS-based ~~BigData-ready~~ ~~LSML~~ ~~Secure~~

R for full cycle development



CRISP-DM



Source: http://0xCode.in/azure-ml-for-data-scientist
This work is licensed under a Creative Commons Attribution 4.0 International License

Step 1: **read** data

```
# 1. from local file system
library(data.table)
dt <- fread("data/transactions.csv")

# > Read 6849346 rows and 6 (of 6) columns from 0.299 GB file in 00:00:31

# 2. from Web
dt <- fread("https://raw.githubusercontent.com/greggles/mcc-codes/master/mcc_codes.csv",
            sep = ",", stringsAsFactors = F, header = T, colClasses = list(character = 2)))

# > % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current Dload  Upload   Total   Spent    Left  Speed
# > 0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0100 14872  100 14872    0     0  29744      0 --:--:-- --
:--:-- --:--:-- 31710

# 3. from Azure Blob Storage
library(AzureSMR)

sc <- createAzureContext(tenantID = "{TID}", clientID = "{CID}", authKey = "{KEY}")
sc

azureGetBlob(sc,
    storageAccount = "contestsdata",
    container = "financial",
    blob = "transactions.csv",
    type = "text")
```

Step 1: **read** data

```
# 4. from MS SQL Server
library(RODBC) # Provides database connectivity

connectionString <- "Driver={ODBC Driver 13 for SQL
Server};Server=tcp:msdevcon.database.windows.net,1433;Database=TransDb;Uid=..."

trans.conn <- odbcDriverConnect(connectionString) # open RODBC connection

sqlSave(trans.conn, mcc.raw, "MCC2", addPK = T) # save data to table
mccFromDb <- sqlQuery(trans.conn, "SELECT * FROM MCC2 WHERE edited_description LIKE '%For Visa Only%'") # get data

head(mccFromDb)
#> rownames code                          edited_description                       combined_description
#> 1      978 9700  Automated Referral Service ( For Visa Only)  Automated Referral Service ( For Visa Only)
#> 2      979 9701     Visa Credential Service ( For Visa Only)     Visa Credential Service ( For Visa Only)
#> 3      980 9702     GCAS Emergency Services ( For Visa Only)     GCAS Emergency Services ( For Visa Only)
#> 4      981 9950 Intra ??“ Company Purchases ( For Visa Only) Intra ??“ Company Purchases ( For Visa Only) Intra ??“

close(trans.conn)

# * Excel, HDFS, Amazon S3, REST-services as data sources
```

Step 2: preprocessing data

```
# { "0 10:23:26" "1 10:19:29" "1 10:20:56" } > { 0, 1, 1 }
getDay <- function(x) { strsplit(x, split = " ")[[1]][1] }

trans <- trans.raw %>%
  # remove invalid rows
  filter(
    !is.na(amount) | amount != 0
    ) %>%
  # transform data
  mutate(
    OperationType = factor(ifelse(amount > 0, "income", "withdraw")),
    TransDay = as.numeric(sapply(tr_datetime, getDay)),
    Amount = abs(amount)
    ) %>%
  # remove redundant columns
  select(
    -c(tr_datetime, amount, term_id)
    ) %>%
  # set column names
  rename(
    CustomerId = customer_id, MCC = mcc_code, TransType = tr_type
  ) %>%
  # sort
  arrange(
    TransDay, Amount
  )
```

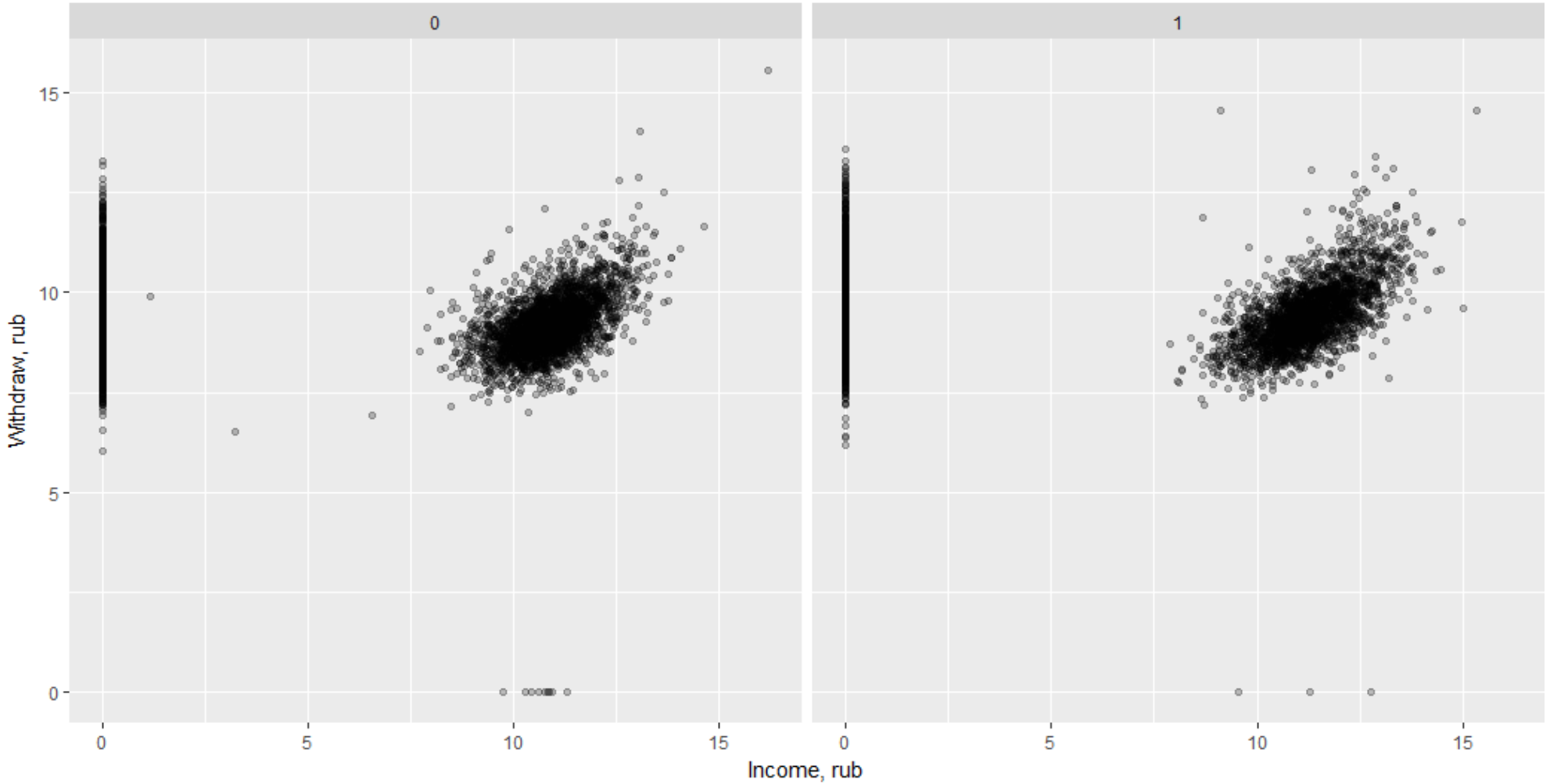
Step 3: feature engineering

```
# calculate stats
library(dplyr)
customers.stats <- trans.x %>%
  mutate(LogAmount = log(Amount)) %>%
  group_by(CustomerId, OperationType, Gender) %>%
  filter(n() > 30) %>%
  summarize(
    Min = min(LogAmount),
    P1 = quantile(LogAmount, probs = c(.01)),
    Q1 = quantile(LogAmount, probs = c(.25)),
    Mean = mean(LogAmount),
    Q3 = quantile(LogAmount, probs = c(.75)),
    P99 = quantile(LogAmount, probs = c(.99)),
    Max = max(LogAmount),
    Total = sum(Amount),
    Count = n(),
    StandDev = sd(LogAmount)
  ) %>%
  ungroup()


# shape from long to wide table form
library(reshape2)
x <- dcast(customers.stats, CustomerId + Gender ~ OperationType, value.var = "Mean", fun.aggregate = mean)
```

Step 3: feature engineering

```
library(ggplot2)
ggplot(x, aes(x = income, y = withdraw)) +
  geom_point(alpha = 0.25, colour = "darkblue") + facet_grid(. ~ Gender) +
  xlab("Income, rub") + ylab("Withdraw, rub")
```

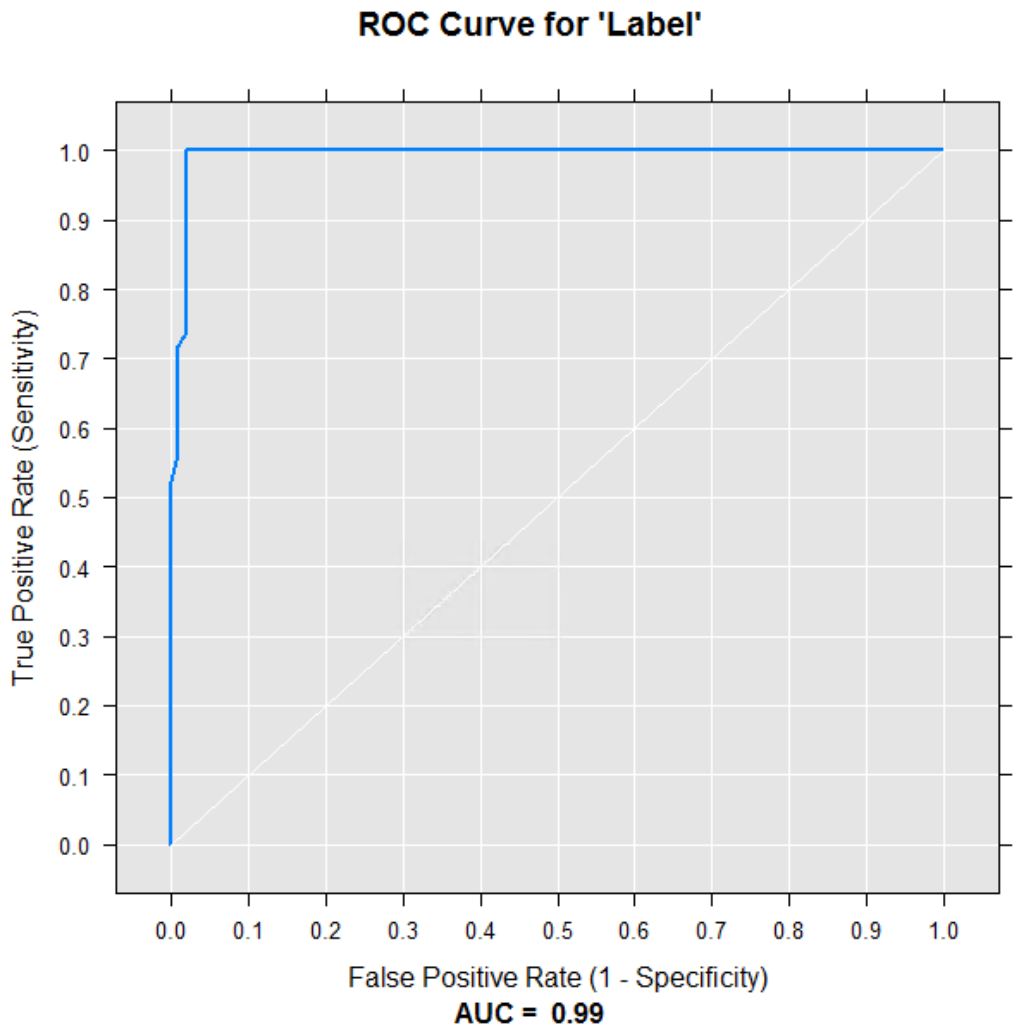


Step 4: training ML-model

```
# train model
model <- glm(formula = gender ~ ., family = binomial(link = "logit"), data = dt.train)

# score model
p <- predict(model, newdata = dt.test, type = "response")
pr <- prediction(p, dt.test$gender)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)

# evaluate model
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```



Last step: share results

```
# It’s time to Demo!
Demo.GetStarted(formula = ML ~ .)
```

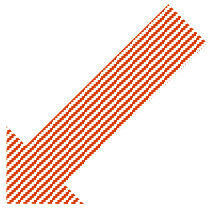
Challenges

Data Science evolve rapidly

Data growing even faster

Data >> Memory (now and evermore)

We must scale better

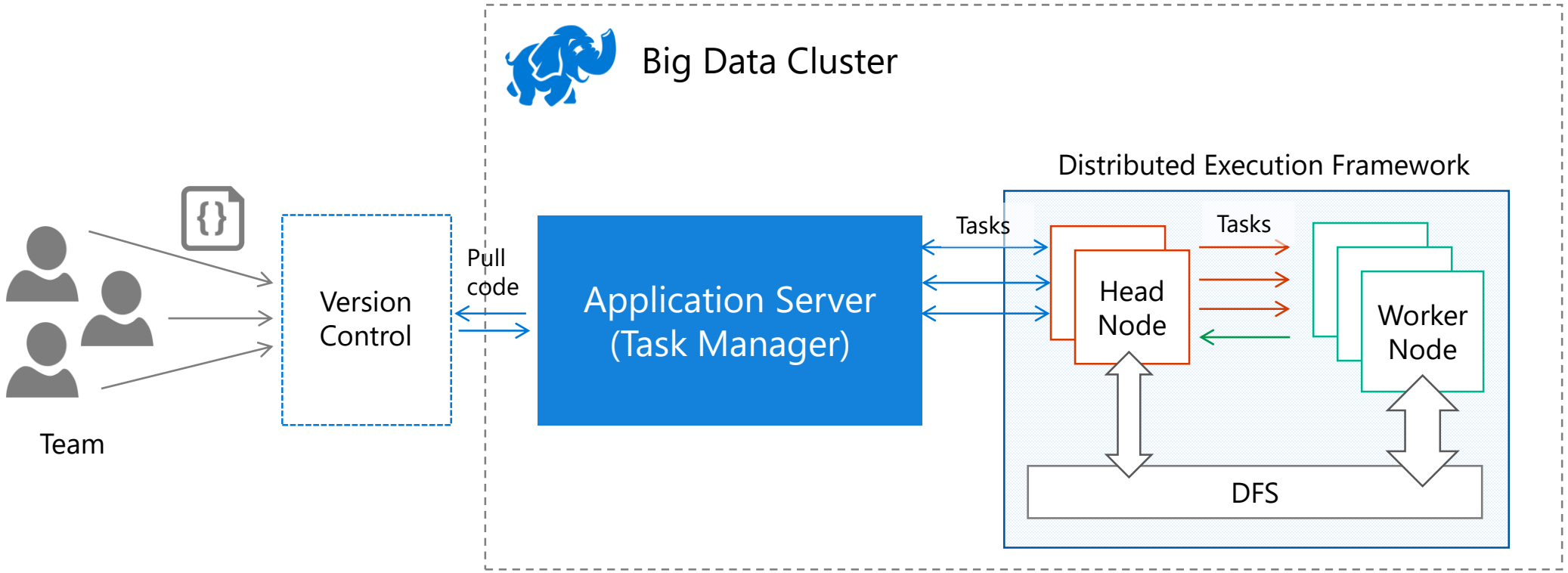


Complex infrastructure

Zoo of frameworks

May be cloud?

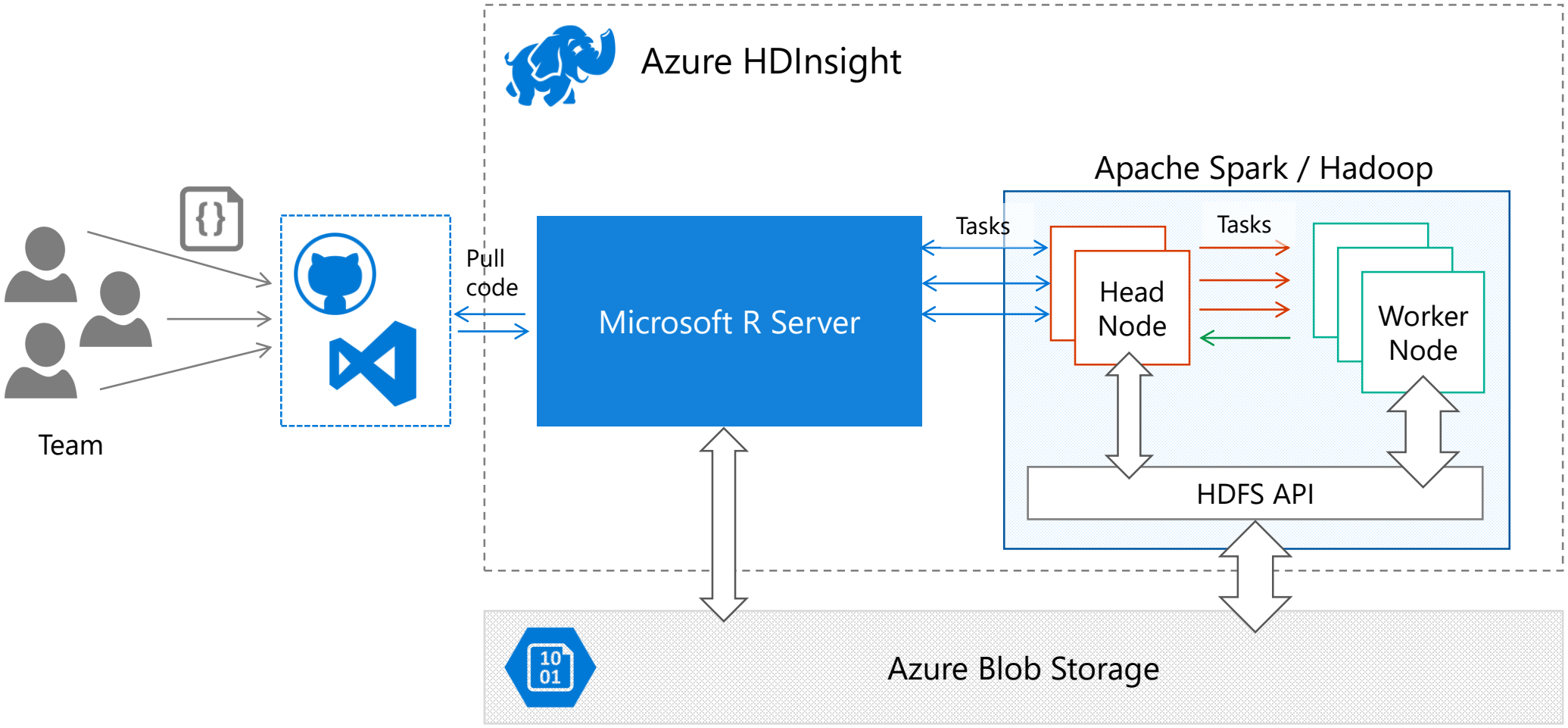
ML for the ~~bloody~~ Enterprise



~~Flexibility~~ Distributed Large scalable Fault-tolerance Reliable

OSS-based BigData-ready LSML Secure

R for the Enterprise



Big Data + Cloud + Machine Learning

Долго, дорого, …

#msdevcon

DevCon School
Технологии будущего

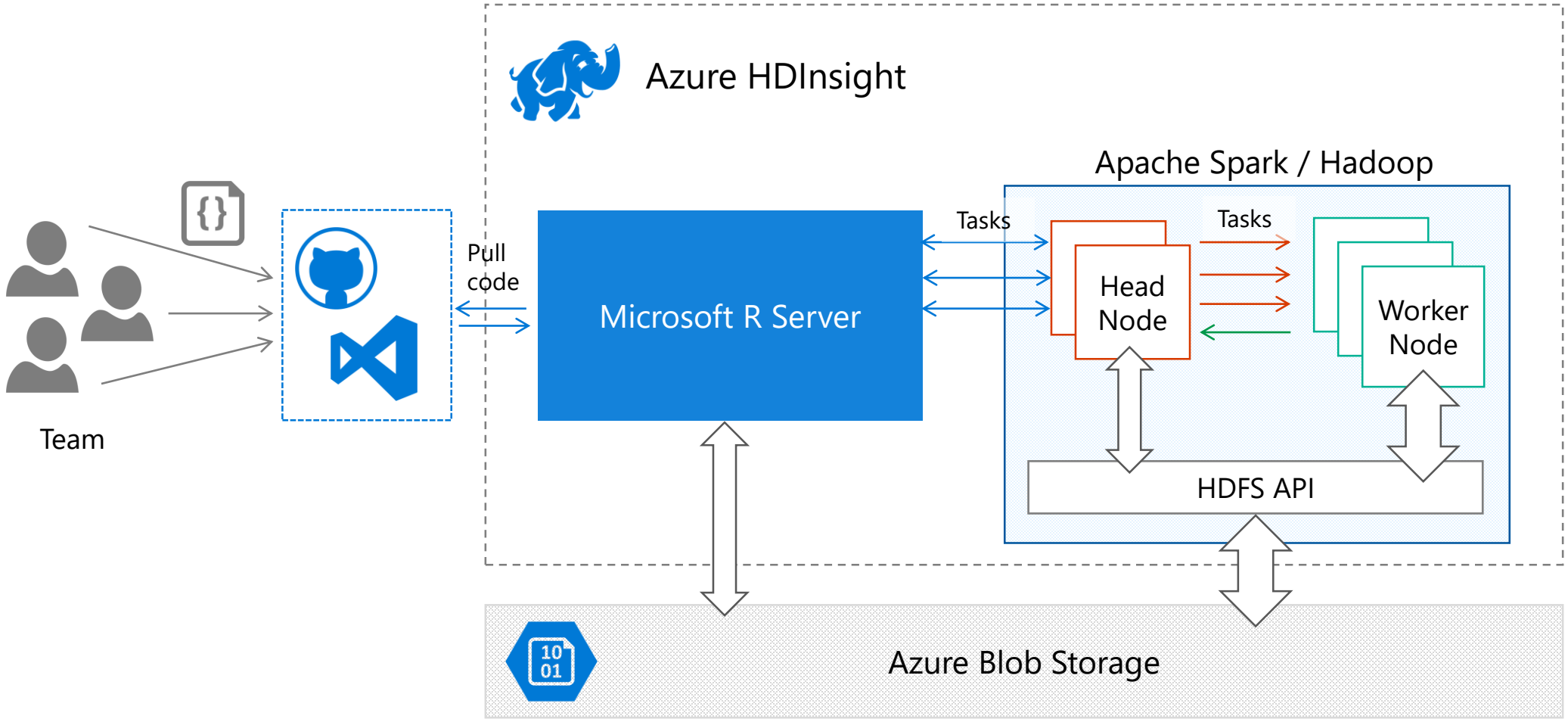
Apache Spark/Hadoop + Azure + R Server

Доступен как PaaS-сервис

#msdevcon

DevCon School
Технологии будущего

R for the Enterprise



Demo II

Microsoft ❤ R

Microsoft Azure

Microsoft R Open and Microsoft R Server	#R
MicrosoftML	#R
Microsoft R Server for Azure HDInsight	#PaaS
R Server on Apache Spark	
Data Science VM	#R #IaaS
CNTK & GPU Instances	#NN #GPU #OSS
Batch AI Training preview	#PaaS #NN #GPU
Azure Machine Learning	#PaaS
R scripts, modules and models	#R
Jupyter Notebooks	#R #SaaS
R-to-cloud: AzureSMR, AzureML	#R #OSS
Cognitive Services	#SaaS #NN
SQL Server R Services	#R #PaaS
Power BI	#R #Viz
Execute R scripts	
Visual Studio	
R extensions for VS2015	
R in-box-support for VS2017	

Отзывы

Помогите нам стать лучше!

На вашу почту отправлена индивидуальная ссылка на электронную анкету. 3 июня в 23:30 незаполненная анкета превратится в тыкву.

Заполните анкету и подходите к стойке регистрации за приятным сюрпризом!

#msdevcon

Оставляйте отзывы в социальных сетях. Мы все читаем. Спасибо вам! ☺

DevCon School
Технологии будущего

Data Science must win!

© 2017, **Dmitry Petukhov**. CC BY-SA 4.0 license. Microsoft and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

Q&A

Now or later (use contacts below)

Ping me

Habr: @codezombie

All contacts: http://0xCode.in/author