Ruby Association開発助成で得た知見の共有と今後

西田 孝三
三軒家 佑將
(芦田 恵大)

発表概要

- 我々のRuby Association開発助成プロジェクト(以下RA)について (10分)
- Jupyter Notebookを用いた実演 (20分)
- RAで得た知見と今後 (10分)
- 質疑応答 (5分)

伝えたいこと

- Python同等のことをRubyで実現するアプローチは厳しい
  - Jupyter Notebookを用いた実演を通してその背景知識を共有
  - このアプローチを捨てる，わけではない
  - RAで得た知見を共有し，今後コミュニティで改善

- 上記をふまえRubyでデータサイエンスする方法を考える
  - PyCall や red-arrow も活用し Ruby独自のワークフローを構築していけないか ？
  - キラーアプリを中心にワークフローを考えることができないか？　(Rubyを使う動機が必要)
  - 現時点ではデータソース (Rails?) が鍵？

データサイエンスのワークフローとは

データソース
- Excel
- RDB
  - グラフ, 画像, 音, HDF5, arrow…

行(sample)
列(variable)
sampleの例(「人」amazon, facebook, ...)

Python

データ前処理
- group_by
- 欠損値扱い

pandas

分析
- 機械学習
- 統計

- Scikit-learn
- Scipy, statsmodels

可視化
- 散布図
- 棒
- 箱ひげ

- Matplotlibベース [静的] (seaborn, ggplot, …)
- JSベース [動的](bokeh, plotly, …)

- 多次元データの理解
- モデルに基づいた予測

データサイエンスとは

データソース
- Excel
- RDB
  - グラフ, 画像, 音, HDF5, arrow…

行(sample)
列(variable)
sampleの例(「人」amazon, facebook, ...)

データ前処理
- group_by
- 欠損値扱い

分析
- 機械学習
- 統計

可視化
- 散布図
- 棒
- 箱ひげ

- 多次元データの理解
- モデルに基づいた予測

Python

pandas

- Scikit-learn
- Scipy, statsmodels

- Matplotlibベース [静的] (seaborn, ggplot, …)
- JSベース [動的](bokeh, plotly, …)

Ruby

daru

- 無し
- statsample

- 無し (gnuplot, gruff)
- JSベース (nyaplot, **rbplotly**, …)

我々のRAについて

- 既存gemを組み合わせたワークフローのテストや実例指向のドキュメント作成

- 上記を通じリファクタリングや改善を

我々のRAについて

- 既存gemを組み合わせたワークフローのテストや実例指向のドキュメント作成
  - やってはみたが，daruを使うことに難を感じる（ワークフローの9割はデータ前処理）
  - Jupyter Notebookを使ったドキュメントを実演することでこの後示します

- 上記を通じリファクタリングや改善を
  - できなかった
  - 型付き行列ライブラリ(numpy相当物) ラッパーとしてのデータフレーム ライブラリの難しさ (これも実演で)
  - シンプルな代替案を提示したが daru作者との間でのトレードオフ理解に差

以降はNotebook実演後のスライド

得られた知見 (というよりは情報共有したいこと)

- Pandas は numpy ラッパー
  - データフレーム実装にはnumpy相当のgemの活用が必須
- Daru が抱える複雑性
  - numpy 相当機能の実装を複数抱えている
- Pandas 同等gemの実現を目指すことの難しさ
  - Rubyを使う強い「必要性」「動機」が必要

RAにおける反省

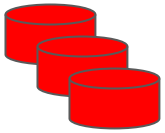
- 「Rubyでworkflowを通して行う実例を増やす」ことが利用者増加につながる，は思い込み
  - 「Rubyらしい書き方」よりも「 Rubyでしかできない」ことがないとやる気が続かない
- Daruのアプローチ(pandasを真似る)では「Rubyらしい書き方」は難しい
  - どの言語らしい書き方でもない DataFrameのイディオム
  - Arel-like syntax => 特にうれしさわからず
  - Pandasとの微妙な記法の違い => 覚えることが増え，負担に
- 開発規模の見誤り => DataFrameとnumpy相当gemの連携
  - コードが大規模かつ複雑になることは避けられない
- 可視化部分の詰めが尻切れトンボに

RAにおいて胸を張れる点

- 確実に「入り口」の整備はできた
  - IRuby用zeromq gemの更新
  - Docker や Win-Mac-Ubuntu native環境の整備
  - gem パッケージングの整備
  - Ruby kernel を使った Jupyter Notebookの追加

- 「落とし穴」を明確にした
  - numpy 相当の gem の重要性の理解
  - 「とにかく全部Rubyでやる!」ではburnoutするかも

今後の考え方の案の一つ



データソース
- Rails (Active record?)
- Groonga

よくできたRuby binding gemを活用

データ前処理
- group_by
- 欠損値扱い

分析
- 機械学習
- 統計

可視化
- 散布図
- 棒
- 箱ひげ

Python

pandas

- Scikit-learn
- Scipy, statsmodels

- Matplotlibベース [静的] (seaborn, ggplot, …)
- JSベース [動的](bokeh, plotly, …)

Ruby

- PyCall
  - なんらかのbinding技術

- **PyCall** (matplotlib)
- JSベース (rbplotly, …)

我々の今後の課題

- Rbplotly の ラップを厚くする
  - 複雑なグラフを書くのはまだ大変
- IRubyからActive Recordを簡単に触れるように
  - 既存のRailsアプリのDBからグラフが書けたら嬉しいのでは？

もちろん「既存の gem をなんとかする」方針も

- Google summer of code で RDB や NOSQLDB を daru でimport 後 可視化するプロジェクト

- Plotly や その他 JS プロットライブラリ(bokehjs)との連携
  - 独立した小さめのプロジェクトとしてやっていきやすい

- Ruby-numo gem 群の活用
  - https://github.com/ruby-numo/gnuplot-demo Jupyter Notebookに対応
  - https://github.com/ruby-numo/gsl 統計，回帰，疎行列

コミュニティへのお誘い

- PyData https://pydata-jp.herokuapp.com
  - 広い視野があるとよいかも
  - コミュニティ運営の点でも興味深い https://www.numfocus.org/

- SciRuby https://sciruby-jp.herokuapp.com
  - お気軽に
    - 楽しいプロジェクト状態を作ることが重要

謝辞

- mrknさん
- 笹田耕一さん
- 田中昌宏さん (numo-narray)
- kouさん (cztop, iruby)
- ITOC (しまねソフト研究開発センター) のみなさん
- Rubyアソシエーション
- Sameer Deshmukhさん (daru, iruby)

- どみとり (西田直樹) 君