MondrianForestのテストを行った。

<http://archive.ics.uci.edu/ml/index.php>というなんか色々データがあるサイトから落としてきた、以前学習用に遊んでいたadultというデータ(エクセルで修正)を使ってテストをしてみる。

データの内容はAge, Workclass, Fnlwgt, Education, Education\_number, Marital\_Status, Occupation, Relationship, Race, Sex, Capital\_Gain, Capital\_Loss, Hours\_pre\_week, Native\_Countryの14項目から、Imcomeが>50Kか、<=50Kかを判別するというもの。

データ数は32561個で、テストデータは答え付きで16281個ある。

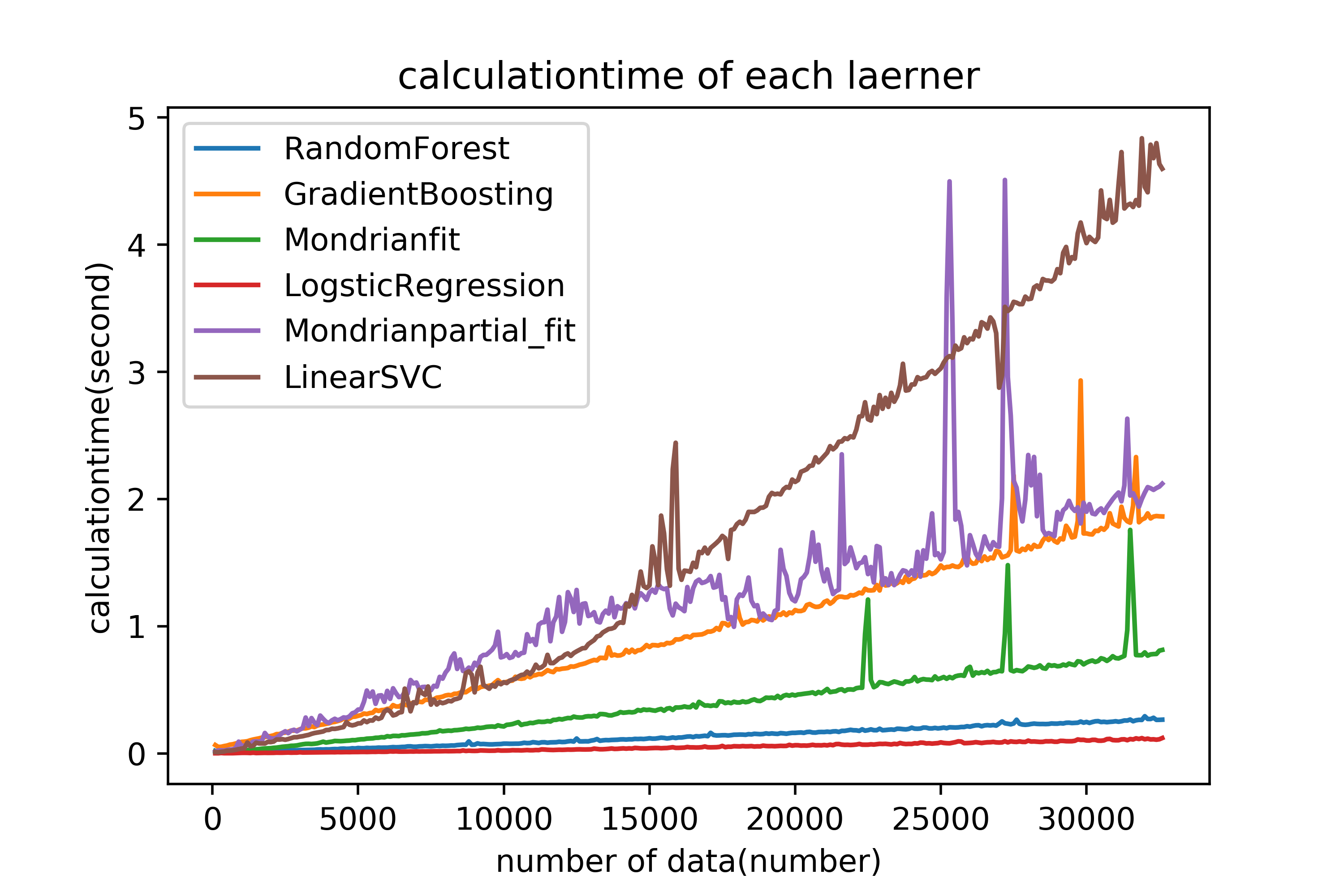
(計算時間なのでデータ数が多い方がいいと思ってこうしたのですが、処理が重すぎて死にました。勉強になりました。)

欠損値などあったので、seminarのP04S06\_Scoring process-2に従ってget\_dummies, RFEによる特徴量選択などのデータの前処理をした。

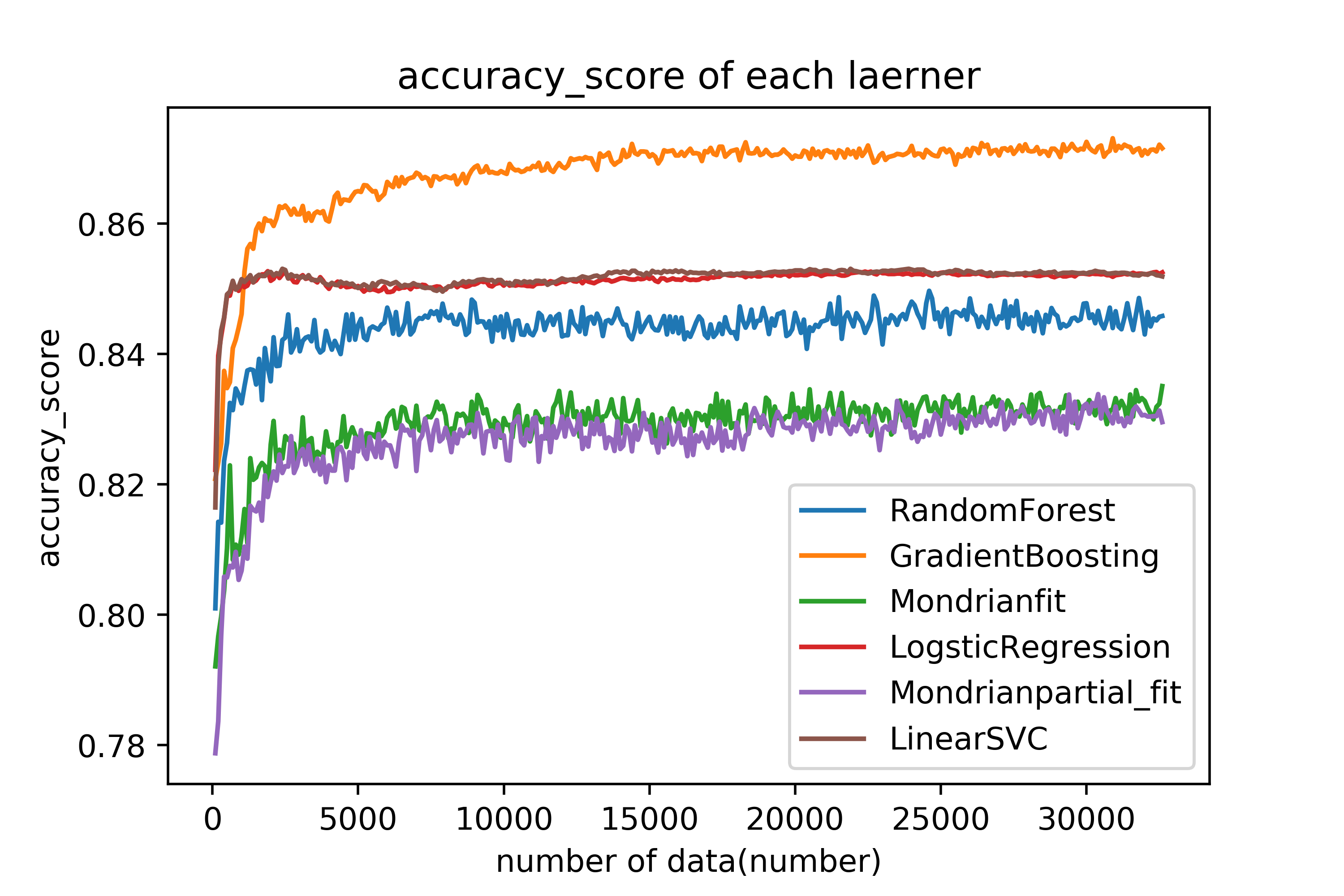
特徴量は15個に絞り、残ったのは、Age, Fnlwgt, Education\_number, Capital\_Gain, Capital\_loss, Hours\_pre\_week, Workclass\_Self-emp-not-inc,…,など。

Partial\_fitはpipelineできないため、事前にStanderdSchlerによる標準化を行い、「fit」など自体の処理を計測した。

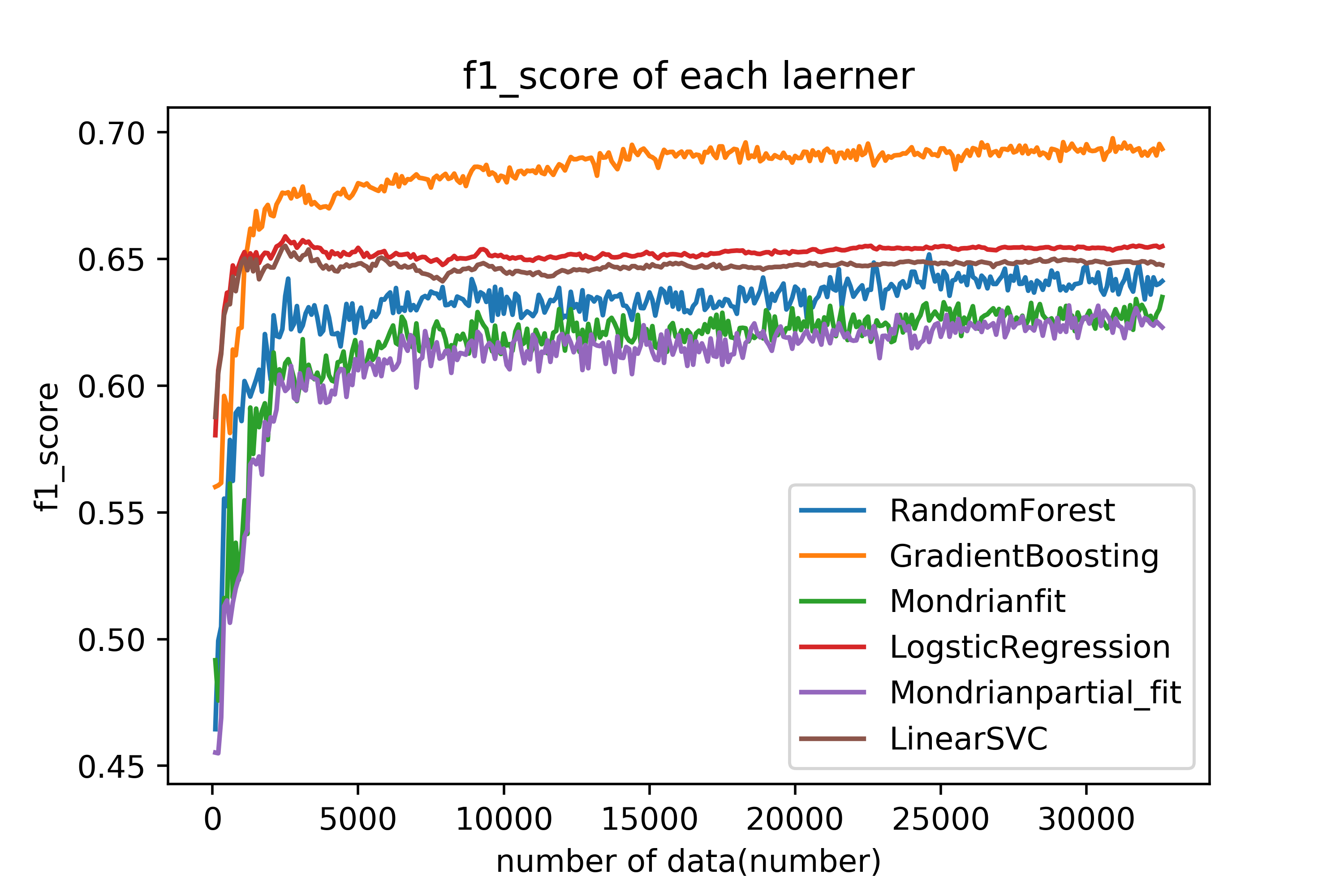
まず、そもそもの性能を見るために、一度データ数に伴って各学習器の計算時間とスコアがどうなるかを調べる。ここでは、MondrianForestのPartial\_fitは毎回リセットし処理を行っているので、蓄積されたデータ数がゼロの時から、データを入れるとどれくらい時間がかかるかを示している。



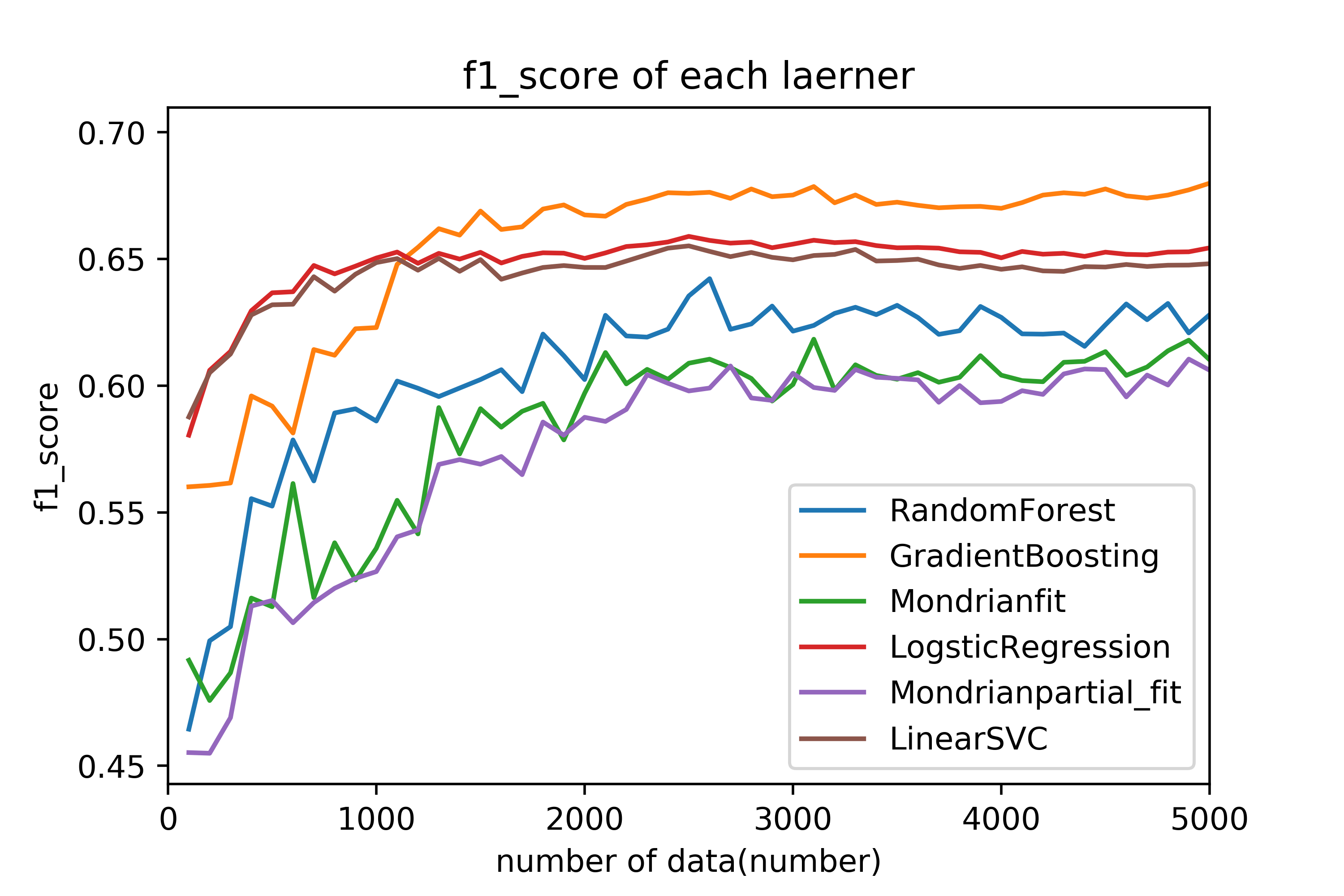
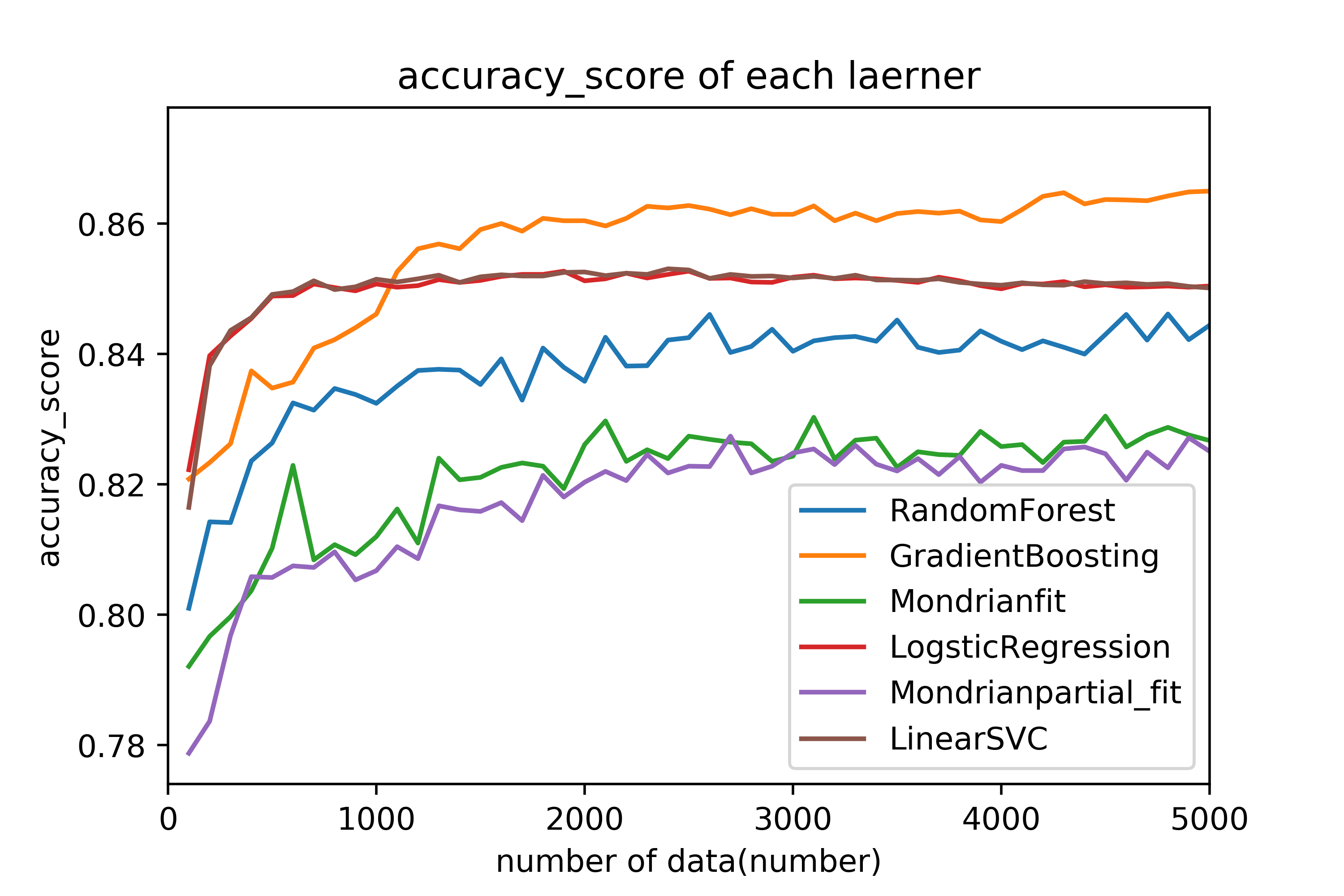
この時のaccuracy score。



また、f1 score



最初の方でほぼ収束してしまっているので、軸を変えて。



とりあえず、ここから読み取れることは、

・MondrianのPartial\_fitとfitはほぼ精度が変わらない(fitの方が若干高い)上に、partial\_fitの方がだいぶ遅い。

・MondrianForestはRandomForestと比べるとそこまで精度に差はないが、GradientBoostingと比べると有意な差が出てしまっている。

GradientBoostingはα(学習率)など全てデフォルト値でやっているので、スピードより精度重視の設定になってはいるが。