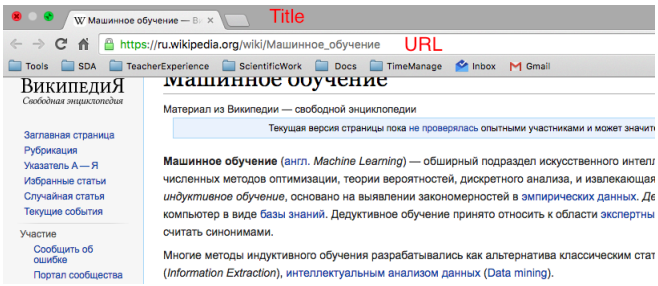
Машинное Обучение МФТИ

Практическое задание №4: Определение Возраста

В качестве четвертой домашней работы студентам предлагается принять участие в соревновании по определению возраста пользователя по его браузерной истории на Kaggle. Срок сдачи **14.05.2016**.

Мотивация для Решения Задачи

Многие интернет компании обладают сервисами аналитики (Яндекс Метрика, Гугл Аналитика, Мейл Топ-100, Рамблер Топ-100, ...) эти проекты изначально созданные для предоставления услуг анализа посещаемости и прочих характеристик ваших сайтов. Как бонус владельцы счетчиков получают браузерную историю пользователя.



Эти данные хочется использовать для получении прибыли, один из путей монетизации данных это продаже рекламным системам некоторой информации про пользователя – пол, возраст, доход интересы, имея эту информацию можно персонализировать рекламу для увеличения эффективности рекламной компании. Обучающую выборку при этом можно найти в других сервисах вашей компании, к примеру почте (возраст указывается при регистрации).
Вам, как специалисткам по машинному обучению нужно построить алгоритм предсказания возраста пользователя по его браузерной истории.

Описание Данных и Метрик

Каждый файл с данными, это csv файл с разделителем `tab`:

1. `age_profile_train` – user_id(str), age(int – traget value)

2. `(url_domain/title_unify)_(train/test)` – user_id(str), url/title (str), num_vizits (int)

В качестве метрики используется Mean Squared Error (MSE):

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i}^{N}(y_i - \hat{y}_i)^2$$

Сдача Задания и Правила Игры

1. Можно использовать любые реализации изученных на курсе моделей
2. Нужно будет + прислать отчет в виде ipython notebook с описанием экспериментов
3. Задание оценивается максимальным баллом если вы
   (a) побили baseline + сдали задание вовремя
   (b) использовали хотя-бы 2 метода снижения размерности и стекинг моделей
   (c) за первые места, как обычно бонусы

Методические Указания

1. На сильно разреженной матрице вы не сожмите обучить композиции алгоритмов (RandomForest, Boosting), поэтому стоит посмотреть в сторону линейных моделей и уменьшение размерности пространства.
2. В качестве библиотеки для линейных моделей рекомендуется использовать Vowpal Wabbit
3. Для снижения размерности пространства нужно попробовать использовать:
   (a) различные матричные разложения SVD, LDA (Vowpal Wabbit, GenSim)
   (b) Hashing Trick – есть в SciKit-FeatureHasher, встроен в Vowpal Wabbit
   (c) Word2Vec (GenSim) – обратите внимание, что обучить модель самому может быть:
      i. долго – используйте AWS Educate или пред-обученную модель
      ii. тяжело – уменьшите число параметров за счет снижения размерности векторов
   (d) иные методы учитывающие специфику задачи к примеру гистограмма $p(url|age[00-25])$ для каждого пользователя
4. При использовании нейросетевых методов обратите внимание на библиотеку Lasagne,
   (a) нужно будет внимательно почитать про "convolutional/recurrent neural networks text"
   (b) обратите внимание на EmbeddingLayer, это почти Word2Vec
   (c) все эти нейросети работают очень долго используйте AWS Educate нужен инстанс с видеокартой, разумный выбор `g2.2xlarge`, при установке образа найдите уже с установленными библиотеками
5. Используйте стекинг моделей (несколько уровней регрессий):
   (a) использовать не больше двух уровней
   (b) модель второго уровня обучается на ответах моделей с первого уровня
   (c) модель второго уровня должна быть простой(грубой)
   (d) модель второго уровня лучше обучать на отдельном hold-out

   Будьте аккуратны с кросс-валидацией, не переобучитесь.

Разница между списыванием и помощью товарища иногда едва различима. Мы искренне надеемся, что при любых сложностях вы можете обратиться к семинаристам и с их подсказками самостоятельно справиться с заданием. При зафиксированных случаях списывания (одинаковый код, решение задачи), баллы за задание будут обнулены всем участникам инцидента.