

Машинное обучение на платформе .NET

 Неволин Роман

 roman.nevolin@waveaccess.ru

 [nevoroman](#)

 [nevoroman/ml-dotnet](#)

Вступление : из-за чего сыр-бор?

- Машинное обучение - это классно.
- .NET - это тоже замечательно
- По распространенному мнению, .NET не подходит для решения задач машинного обучения...

...но мы же не станем отступать из-за таких мелочей?

Машинное обучение

Машинное обучение : что это за зверь?

«Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed»

— Prof. Arthur Samuel

Машинное обучение : когда использовать?

- Когда трудно описать алгоритм решения задачи
- Когда нужно предугадать некоторые значения, имея большой набор данных
- Когда вы хотите улучшить работу имеющегося алгоритма за счет накопления опыта

Машинное обучение : зачем мне это?

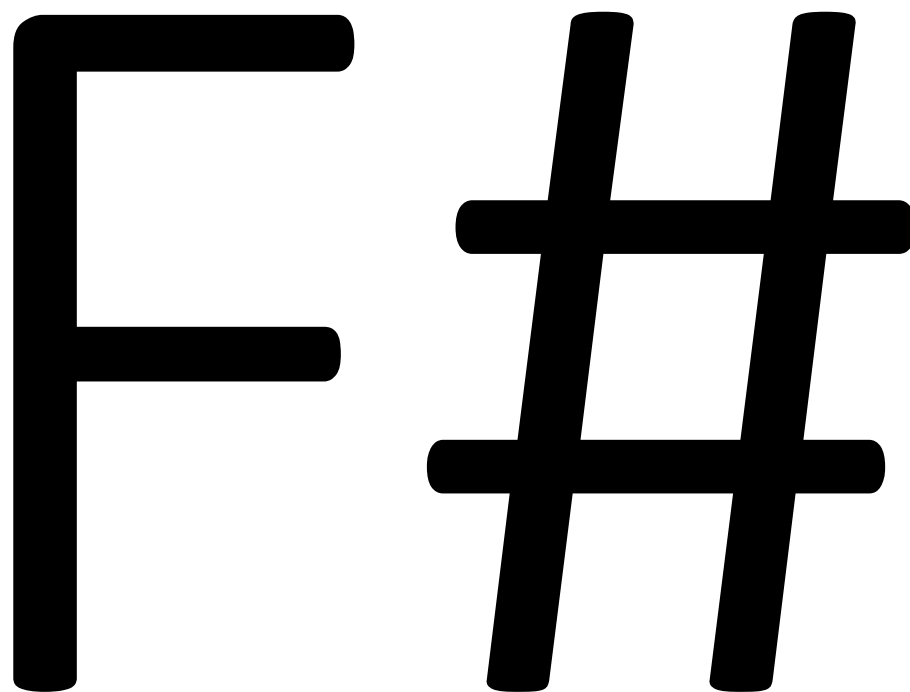
- Позволяет легко решать трудные задачи
- Базовые навыки легко осваиваются и полезны в других областях
- Это чертовски весело!

ML и .NET

ML и .NET : мы этого точно хотим?

- .NET имеет кучу клевых инструментов для работы с данными
- Позволяет встраивать алгоритмы машинного обучения, не выходя из уютного дотнета
- F#

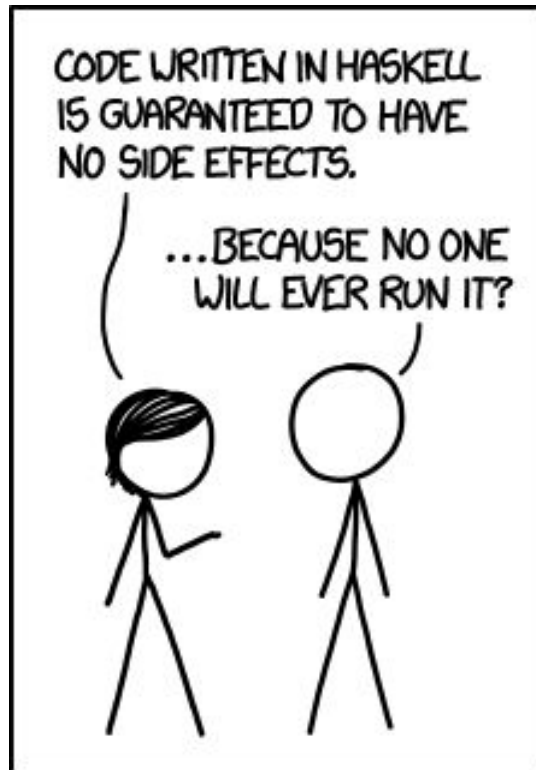
ML и .NET : мы этого точно хотим?



ML и .NET : препарируем F#

```
let rec quicksort list =  
    match list with  
    | [] -> // If the list is empty  
             // return an empty list  
    | firstElem::otherElements -> // If the list is not empty  
                                   // extract the smaller ones  
        let smallerElements =  
            otherElements  
            |> List.filter (fun e -> e < firstElem)  
            |> quicksort // and sort them  
        let largerElements = // extract the large ones  
            otherElements  
            |> List.filter (fun e -> e >= firstElem)  
            |> quicksort // and sort them  
        // Combine the 3 parts into a new list and return it  
        List.concat [smallerElements; [firstElem]; largerElements]
```

ML и .NET : препарируем F#



ML и .NET : препарируем F#

```
let rec quicksort2 = function
| [] -> []
| first::rest ->
    let smaller,larger = List.partition ((>=) first) rest
    List.concat [quicksort2 smaller; [first]; quicksort2 larger]
```

Функциональное программирование - это просто!

ML и .NET : препарируем F#

```
[<Literal>]
let connectionString = "Data Source=DESKTOP-QM2SSL0\SQLEXPRESS; Initial
Catalog=MyDatabase; Integrated Security=True;"

type Sql = SqlConnection<connectionString>
let db = Sql.GetDataContext()

// find the number of customers with a gmail domain
query {
    for c in db.Customers do
    where (c.Email.EndsWith("gmail.com"))
    select c
    count
}
```

ML и .NET : немного уличной магии

```
// Required package to save charts
open RProvider.grDevices

// Create path to an image testimage.png on the Desktop
let desktop = Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
let path = desktop + @"testimage.png"

// Open the device and create the file as a png.
// R.bmp, R.jpeg, R.pdf, ... will generate other formats.
R.png(filename=path, height=200, width=300, bg="white")
// Create the chart into the file
R.barplot(widgets)
// Close the device once the chart is complete
R.dev_off ()
```

ML и .NET : ближе к делу

Постановка задачи :

Необходимо без использования библиотек и бубна написать спам-фильтр, написав для этих целей как можно меньше и кода, не жертвуя, по возможности, читаемостью.

Решение:

F#, Байесовский классификатор и немного функциональщины

ML и .NET : что за классификатор?

Наивный байесовский классификатор - простой вероятностный классификатор, основанный на применении теоремы Байеса.

$$P(A | B) = \frac{P(A \cdot B)}{P(B)}$$

Инструментарий

Инструментарий : чем думать будем?

- Accord Framework
- numl
- Encog
- Azure ML

Инструментарий : Accord Framework

- Великолепно документирован
- Огромный арсенал всевозможных алгоритмов
- Гибкий
- Требуется некоторое время на освоение

Инструментарий : ntml

- Просто осваивается
- Не требует долгого развертывания; для базовой реализации алгоритма достаточно и минуты
- Неплохой набор алгоритмов
- С ним базовые алгоритмы машинного обучения сможет реализовать даже ваша бабушка

Инструментарий : Encog

- Довольно гибок
- Неплохая встроенная работа с данными
- Имеет Java реализацию
- Работе с ним посвящена неплохая книга
- Некоторые алгоритмы он реализует просто великолепно.

Инструментарий : Azure ML

- Прост в использовании : ваш ребенок может случайно стать Data Scientist'ом
- Это клевые модные облачные вычисления
- Функционал легко расширяется самописными модулями
- Легкая и приятная работа с данными

Инструментарий : немного побенчмаркаем



Инструментарий : немного побенчмаркаем

```
[Benchmark]
public void DecisionTreeTest()
{
    var description = Descriptor.Create<Iris>();
    var generator = new DecisionTreeGenerator();
    var model = generator.Generate(description, _trainingData);
    _results = PredictAll(model, _testingData);
}
```


Инструментарий : немного побенчмаркаем

Accord Framework

Algorithm	Median	StdDev	Errors
Naive Bayes	0.1234 ms	0.0022 ms	0%
Decision Tree	1.0967 ms	0.0638 ms	3,33%
K Nearest Neighbors	0.1645 ms	0.0178 ms	0%

numl

Algorithm	Median	StdDev	Errors
Naive Bayes	1.9585 ms	0.1935 ms	20%
Decision Tree	3.0074 ms	0.5719 ms	13,33%
K Nearest Neighbors	6.6210 ms	0.3223 ms	0%

Инструментарий : переваривая результаты

- Accord отлично показал себя на всех выбранных алгоритмах
- KNN показал максимальную точность в обоих случаях, а значит - он максимально подходит для этой задачи
- Скорость и точность, которую показал Accord, сравнима с лучшими инструментами на Python и R

Инструментарий : ну и зачем нам numl?

```
[Benchmark]
public void KNNTest()
{
    var description = Descriptor.Create<Iris>();
    var generator = new KNNGenerator();
    var model = generator.Generate(description, _trainingData);
    _results = PredictAll(model, _testingData);
}

[Benchmark]
public void NaiveBayesTest()
{
    var description = Descriptor.Create<Iris>();
    var generator = new NaiveBayesGenerator(2);
    var model = generator.Generate(description, _trainingData);
    _results = PredictAll(model, _testingData);
}
```

Инструментарий : FsLab

Объединяет в себе все лучшие инструменты манипулирования данными, созданные для F#

```
let usdata =  
    frame [ "gdp" => series us.``GDP (current US$)``  
            "uni" => series us.``School enrollment, tertiary (% gross)``  
            "co2" => series us.``CO2 emissions (kt)`` ]  
    |> Frame.dropSparseRows  
  
let result = R.lm(formula = "gdp~uni+co2", data = usdata)  
R.summary(result)  
R.plot(result)
```

Спасибо за внимание!