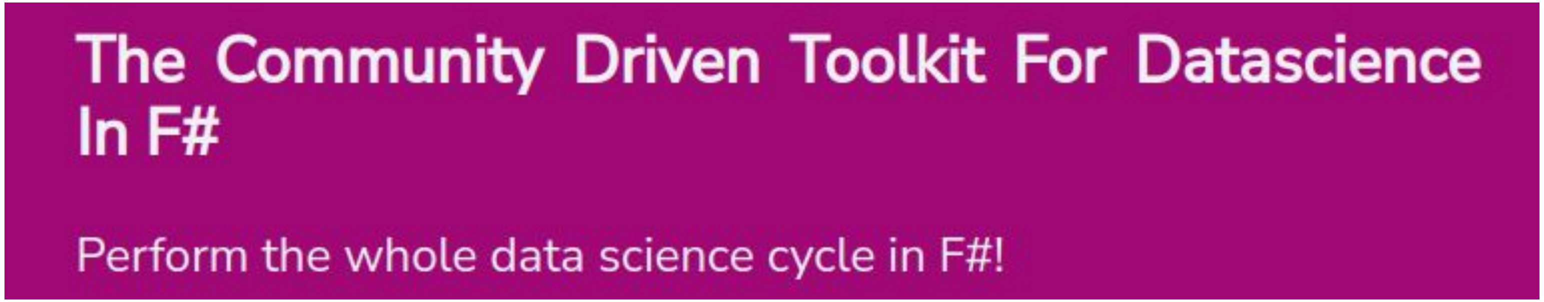
F# и анализ данных FSLab, Deedle, FSharp.Charting, FSharp.Stats

Амарский Артем

FSLab

- FsLab is a center of gravity for data science projects written in and/or for F#
- 

The Community Driven Toolkit For Datascience In F#

Perform the whole data science cycle in F#!

Deedle

- Pandas мира F#
- Поддержка датафреймов и временных рядов
- Поддерживает агрегацию, группировку, обработку пропусков
- Позволяет считать некоторые статистики

FSharp.Charting

- FSharp.Charting is mostly used on Windows
- Библиотека для визуализации
- Позволяет в несколько строчек получить качественный график

FSharp.Stats

- Библиотека для работы со статистикой в F#
- Статистические тесты
- Линейная алгебра
- Машинное обучение

Стоит упомянуть

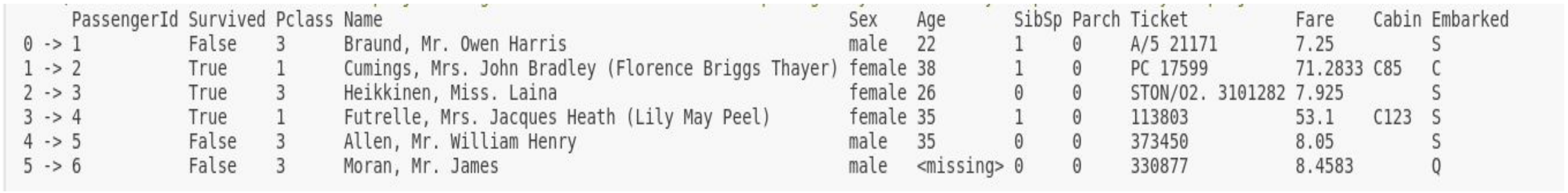
- ML.NET
- TorchSharp
- SciSharp STACK
  - NumSharp
  - TensorFlow.NET
  - Keras.NET
  - LLamaSharp

Perform the whole data science cycle in F#!

- Jupyter notebook
- Titanic dataset
  - Some EDA
  - Some ML

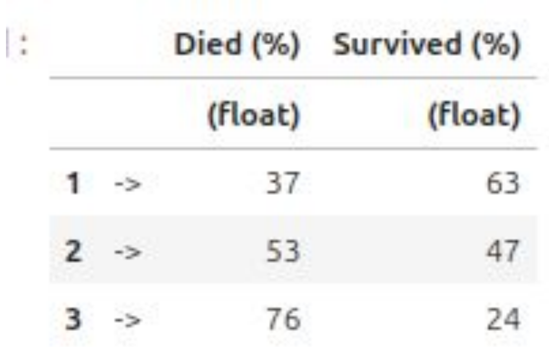
Загрузим датасет

```
#r "nuget: FSharp.Charting, 2.1.0"
#r "nuget: Deedle.Interactive, 3.0.0-beta.1"
#r "nuget: FSharp.Stats.Interactive, 0.5.0"

open Deedle

let df = Frame.ReadCsv("train.csv")
let head = df.Rows[0..5]

head.Print()
```



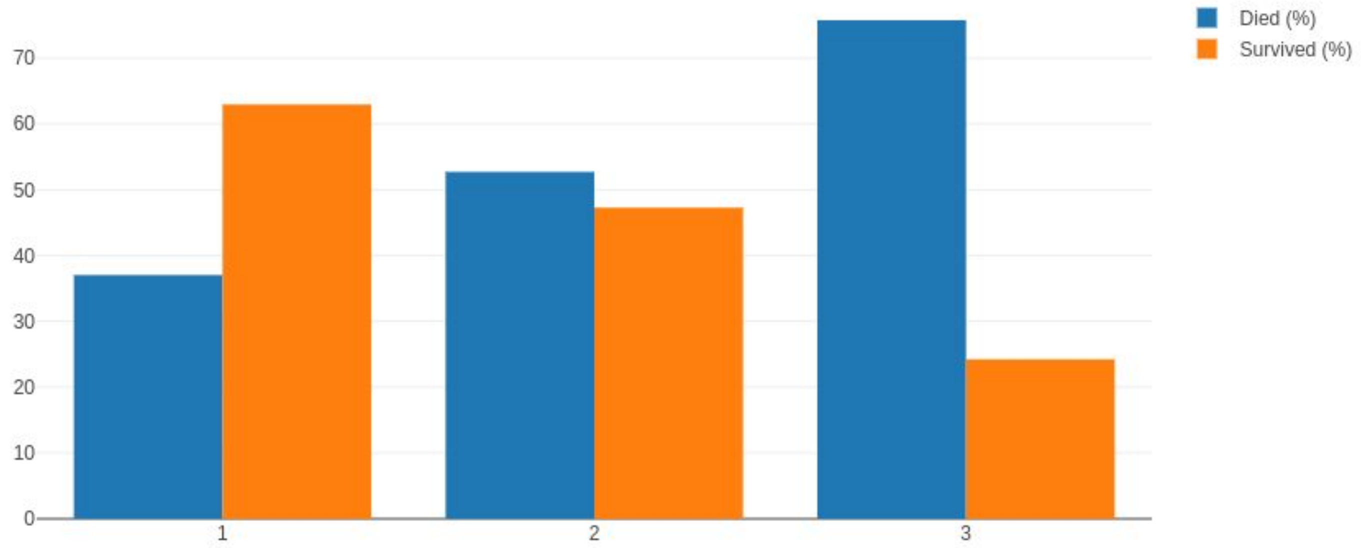
```
       PassengerId Survived Pclass Name                                                Sex    Age       SibSp Parch Ticket           Fare    Cabin Embarked
0 -> 1             False    3      Braund, Mr. Owen Harris                             male   22        1     0     A/5 21171        7.25          S
1 -> 2             True     1      Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 38        1     0     PC 17599         71.2833 C85   C
2 -> 3             True     3      Heikkinen, Miss. Laina                              female 26        0     0     STON/O2. 3101282 7.925         S
3 -> 4             True     1      Futrelle, Mrs. Jacques Heath (Lily May Peel)        female 35        1     0     113803           53.1    C123  S
4 -> 5             False    3      Allen, Mr. William Henry                            male   35        0     0     373450           8.05          S
5 -> 6             False    3      Moran, Mr. James                                    male   <missing> 0     0     330877           8.4583        Q
```

```
let grouped = df.GroupRowsBy<int>("Pclass")


let byClass =
  grouped.GetColumn<bool>("Survived")
  |> Series.applyLevel fst (fun s ->
      // Get counts for 'True' and 'False' values of 'Survived'
      series (Seq.countBy id s.Values))
  // Create frame with 'Pclass' as row and 'Died' & 'Survived' columns
  |> Frame.ofRows
  |> Frame.sortRowsByKey
  |> Frame.indexColsWith ["Died"; "Survived"]

// Add column with Total number of males/females on Titanic
byClass?Total <- byClass?Died + byClass?Survived

// Build a data frame with nice summary of rates in percents
frame [ "Died (%)" => round (byClass?Died / byClass?Total * 100.0)
        "Survived (%)" => round (byClass?Survived / byClass?Total * 100.0) ]
```



		Died (%)	Survived (%)
		(float)	(float)
1	->	37	63
2	->	53	47
3	->	76	24

```
open XPlot.Plotly

byClass?``Died (%)`` <- byClass?Died / byClass?Total * 100.0
byClass?``Survived (%)`` <- byClass?Survived / byClass?Total * 100.0

let savedFrame = byClass

let classes = savedFrame.RowKeys |> Seq.toArray
let diedPercent = savedFrame?``Died (%)`` |> Series.values |> Seq.toArray
let survivedPercent = savedFrame?``Survived (%)`` |> Series.values |>
Seq.toArray

let chart =
    [
        Bar(x = classes, y = diedPercent, name = "Died (%)")
        Bar(x = classes, y = survivedPercent, name = "Survived (%)")
    ]

chart |> Chart.Plot |> Chart.WithLayout (Layout(barmode = "group")) |>
Chart.Show
```


Died (%)
Survived (%)
70
60
50
40
30
20
10
0
1
2
3

Линейная регрессия

- Алгоритм обучения с учителем
- Используется для задачи регрессии
- Ожидает матрицу из объектов, состоящую из векторов-признаков
- Для обучения столбец ответов

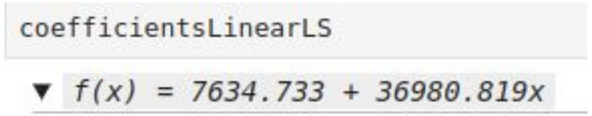
$$y = w^T x + b = \sum_{i=1}^{n} w_i x_i + b$$

- Как учить?

$$MSE = \frac{1}{D} \sum_{i=1}^{D} (x_i - y_i)^2$$

- Градиентный спуск $x_{t+1} = x_t - \alpha \nabla f(x_t)$

```fsharp
open FSharp.Stats
open FSharp.Stats.Fitting.LinearRegression

let dataFrameTrain = Frame.ReadCsv("./california_housing_train.csv").Rows[0..500]

let featureColumn = "median_income"
let targetColumn = "median_house_value"

let featureSeriesTrain = dataFrameTrain.GetColumn<int>(featureColumn)
let featureArrayTrain = featureSeriesTrain |> Series.values |> Seq.map float |> Seq.toArray
let xTrain = Vector.ofArray featureArrayTrain

let targetSeriesTrain = dataFrameTrain.GetColumn<int>(targetColumn)
let targetArrayTrain = targetSeriesTrain |> Series.values |> Seq.map float |> Seq.toArray
let yTrain = Vector.ofArray targetArrayTrain

let coefficientsLinearLS =
    OLS.Linear.Univariable.fit xTrain yTrain
let predictionFunctionLinearLS x =
    OLS.Linear.Univariable.predict coefficientsLinearLS x
```



```
coefficientsLinearLS

▼ f(x) = 7634.733 + 36980.819x
```

```
let predictions = xTest  |> Array.map predictionFunctionLinearLS
  let rmse real pred =

    let differences = Array.map2 (fun x y -> x - y) real pred

    let squaredDifferences = Array.map (fun diff -> diff * diff) differences

    let meanSquaredDifference = Array.average squaredDifferences

    Math.Sqrt(meanSquaredDifference)

rmse yTest predictions
```

```
49061.65032523384
```