

TP “Python for AI”:

a crash course focused on
linear regression models

Name: Andrei Zinovyev

E-mail: **Andrei.Zinovyev.U900@gmail.com**

Page: [https://auranic.github.io/teaching/2021-python for ai](https://auranic.github.io/teaching/2021-python_for_ai)

Python language



- *Franca lingua* for data science community today
- General purpose programming language (not specialized for a field like R)
- Rich set of standard libraries for data manipulation
- Rich set of standard libraries for numerical computations
- Rich set of standard libraries for machine learning
- Rich set of standard libraries for deep learning
- We will use Python 3!

Jupyter Python notebook

- To learn Python
- To write simple Python programs
- To develop complex Python program prototypes
- To exploit powerful computers remotely
- To do cloud computing
- To do data analysis and machine learning!
- To do data analysis without installing Python!



Colab: using Google computers and their GPUs, for your own projects



- Only a gmail account is needed
- Based on Jupyter notebooks
- Integrated with Google Drive
- Has Google machine learning products installed (such as TensorFlow)
- Free Cloud service with free GPU

Kaggle: open data scientist environment



- An online Community (3 million users) of Data Scientists and Machine Learners
- Enter competitions to solve data science challenges
- Data science online education
- Allows finding and publishing data sets

Other available services for online machine learning using Jupyter notebooks



Microsoft Azure



Few Python libraries we will use in this TP

- pandas – for manipulating data frames ('tables with headers')
- numpy – for manipulating 1d and 2d arrays
- scipy – for statistical calculations
- matplotlib – for standard data plotting
- seaborn – for easy data graphics with pandas
- scikit-learn – for machine learning
- tensorflow – for computing regression as a neural network
- lifelines – for survival analysis

scikit-learn: the most popular library machine learning



- Almost any machine learning method, using the same code pattern

```
model = SomeModel(parameters)
model.fit(X,Y)
Y_predicted = model.predict(X)
```

- Excellent documentation: <https://scikit-learn.org/>
- Started and implemented in France by INRIA in 2010

tensorflow: Google open source library oriented to deep learning

- a free and open-source software library for machine learning
- Developed by Google Brain in 2017
- Based on [dataflow](#) and computational graph
- Based on [differentiable programming](#)

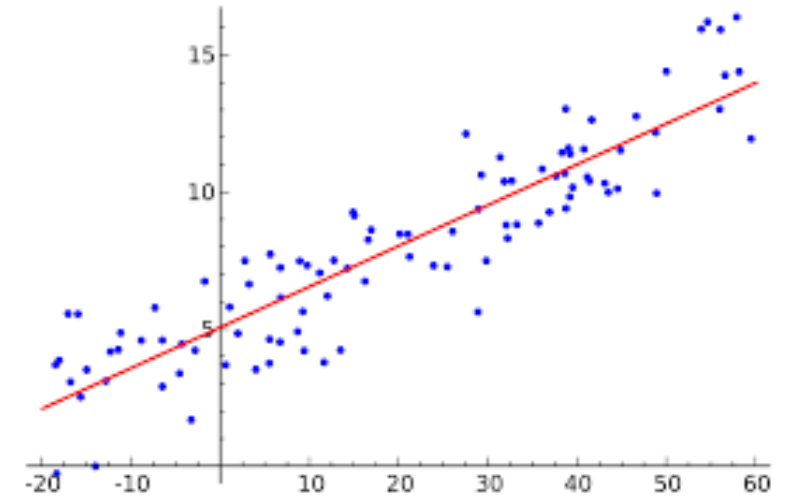
- Easiest way to use TensorFlow is by using Keras : a wrapper on top of tensorflow which allows one to construct deep learning models and fit them to data



Linear regression

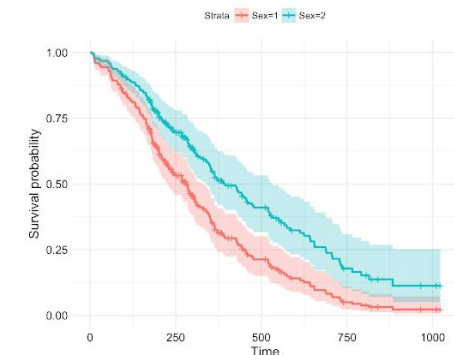
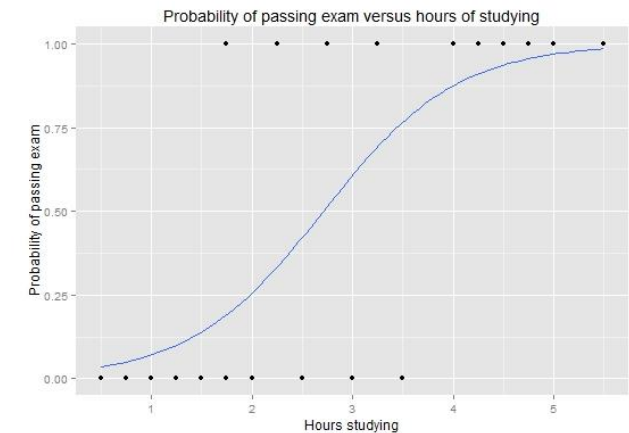
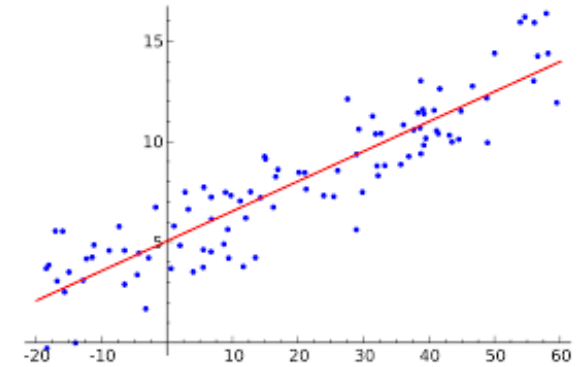
$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

- Connects many independent and one dependent variable via linear function
- Linear regression: the father of all supervised machine learning methods
- Linear regression: the most used machine learning method today
- Linear regression: the first machine learning method to apply, and see what it gives
- Linear regression: common problems and ways to deal with them (regularization)
- Linear regression can be used to produce non-linear data models



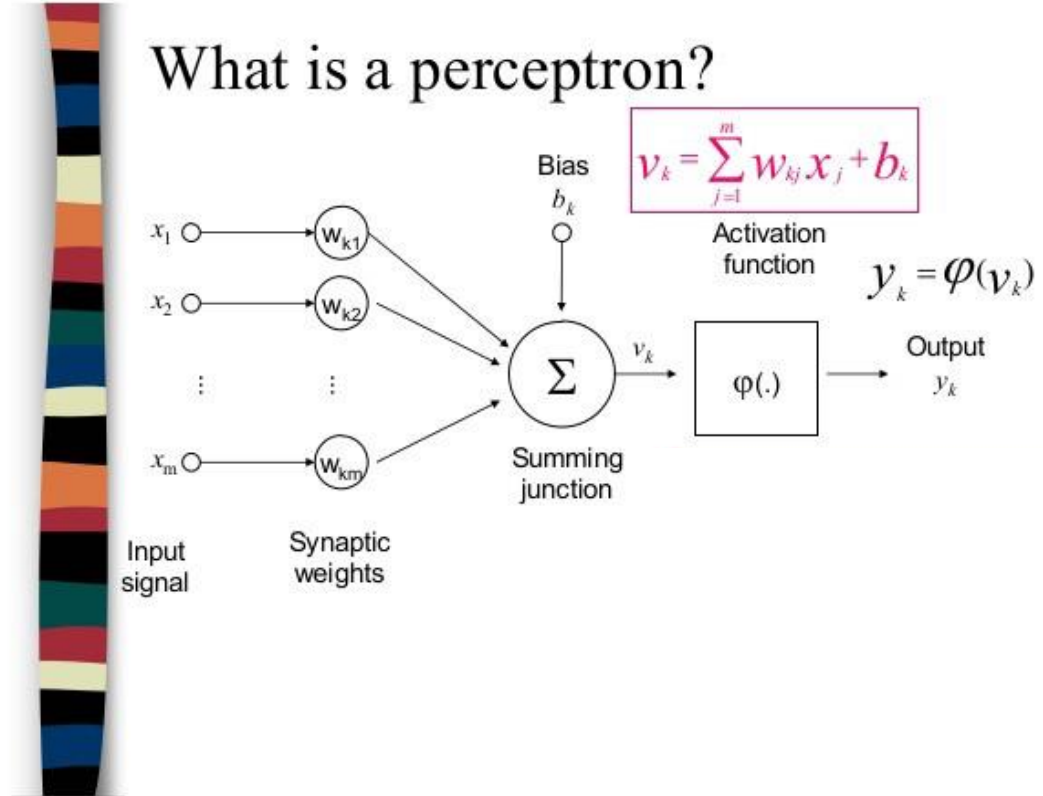
Three major flavors of linear regression in healthcare-related data science

- Simple Ordinary Least Square : when the target variable is continuous
- Logistic linear regression (logit): when the target variable is discrete (for example, binary)
- Survival Cox linear regression : when the target variable is a pair (follow up time + event)



Linear regression and a simple perceptron (formal neuron)

- Invented by Frank Rosenblatt in 1950s
- Elementary unit of any complex and deep neural network today



If $\varphi(x) = x$, then it is simple linear regression model

If $\varphi(x)$ is a step-wise or sigmoidal function then it is a binary classifier just as logistic regression (even though they are trained with different algorithms!)

Linear regression is explainable ML model!

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

Coefficients $\beta_1, \beta_2, \dots, \beta_p$ are comparable if independent variables are standardized (to z-scores) and have straightforward interpretation

It is possible to estimate statistical significance of β_i coefficients and provide p-value on the hypothesis that the coefficient is non-zero

This can help to simplify the regression

Other methods (such as regularization by lasso) for selecting important variables are readily available

Objective of the TP: learn how to build simple regression models in Python

- Univariate and multi-variate regressions
- Validating the regression models in Python
- Building regression models using scikit-learn
- Building regression models using scipy
- Building regression models using tensorflow and train them using gradient-descent (similarly to building neural networks)

Plan of the TP

- Lesson I (~2.5 hours)
 - Connecting to Colab
 - Creating first Jupyter notebook
 - Working with a toy dataset
 - Visualizing simplest linear regression
 - Linear regression: using scikit-learn
 - Linear regression: using scipy statmodels
 - Loading a real-life dataset and exploring it
 - Simplifying the regression model

Plan of the TP

- Lesson II (~2.5 hours)

- Testing and validating the linear regression

Implementing linear regression in tensorflow

- Basic things in tensorflow
- First computational graph
- Linear regression: using tensorflow and gradient descent

Ways to work on TP: all depends on your Python level/motivation!

- Some initial level of Python is assumed:
 - If not then a good start is at:
<https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>
<https://wiki.python.org/moin/BeginnersGuide/Programmers>
<https://www.w3resource.com/python-exercises/python-basic-exercises.php>
- We will start with a simple warming up exercise
- We will have a basic code example on a tiny imaginary example
- *Five real-life datasets from healthcare have been prepared (preprocessed) for this TP : try applying and validating linear regression on some of them*
- In some tasks you will need to insert missing code indicated by ‘...’
- Do not hesitate to copy-paste (from anywhere, e.g. Stack Overflow)
- Do not hesitate to read Wikipedia