# Fundamentals of Artificial Intelligence
# Lab 4: Linear Regression & Kaggle

## Luna-City ML competition of the year

The officials of the Luna-City invite you to participate in the exclusive Machine Learning competition of the year. The task is to build an automated system that can predict the students' chances of admissions to master programs for the universities in the Luna-City.

**You will find the rules for participating in the Kaggle competition on the next page.**

***Please note the mandatory lab tasks below. Participating in the Kaggle competition is optional and has 10% of the lab score.***

## Lab Description and Grading Policy

- **Task 1:** Make a detailed analysis of the dataset and present some statistics in the form of matplotlib or seaborn visualisations. (**2 points**)
- **Task 2:** Based on the analysis from task 1, perform Feature Selection. You should state which columns you will be using for the final predictions and argument why you are choosing the respective columns. (**1 point**)
- **Task 3:** Using the scikit-learn library (`https://scikit-learn.org/`), implement **2 types of Linear Regression**. You can choose between *Ordinary Least Squares, Ridge, Lasso, Least-angle regression (LARS)* and other types offered by the scikit-learn library. Alternatively, implement the code from scratch using the pseudocode from the slides using any programming language. **Make predictions on the test set and report MSE on the train set.** (**2 points**)
- **Task 4:** Draw a scheme of a neural network model that can perform linear regression on the given dataset. It should have at least 1 hidden layer. You should mention the sizes of each layer (number of parameters), the activation functions, as well as the inputs and outputs of the network. (**1 point**)
- **Task 5:** Using Tensorflow or PyTorch, implement a neural network model that will perform regression on the given dataset, in order to predict the chances of admissions. Your model should have at least 1 hidden layer. You can use the model from the previous task, or implement a different model, but you should argument why or why not you are using the same model described in task 4. **Make predictions on the test set and report MSE on the train set.** (**2 points**)
- **Task 6: Report & Presentation of the solution**:

  Explanations, data visualisations, argumentations, report formatting, conclusions, code quality, comments in the code, etc. (**1 point**)
- **Task 7:** Participation in the Kaggle competition (**1 point**)
- **Task 8: (Bonus)** Additional approaches that you used to get a higher score (**1 point**)

## Kaggle competition

### General guidelines

- The competition takes place on the platform **Kaggle** *(see the steps to participate below)*.
- The programming language is **Python**.
- Each student has to submit an **individual solution**.
- **Each student** is allowed to submit **maximum 3 solutions per day**.
- **Please don't host your solution in public repositories (e.g Github etc)**. You can use private repositories if you need.
- Don't cheat!

### Competition Prizes

- The **1st place winner** of the competition will get additional **2 points bonus** at the labs, that can be used to increase your other labs grades (for example, one lab from 8 to 10 or 2 labs from 9 to 10)
- The **2nd place winner**, **3rd place winner** and **4th place winner** of the competition will get additional **1 bonus point** at the labs, that can be used to increase your other labs grades

### For participating in the competition you will need to:

1. Go to https://www.kaggle.com/ and create a profile.
2. Follow the link of the competition https://www.kaggle.com/t/7278d56d02b1463bbbf36b85e66353ca
3. Read the rules and the description of the competition carefully
4. Click "Join competition"
5. Now you can submit your solutions.

### Additional Resources

- Scikit Learn - Linear Models page: https://scikit-learn.org/stable/modules/linear_model.html#least-angle-regression
- Tensorflow: https://www.tensorflow.org/
- Matplotlib: https://matplotlib.org/
- Seaborn: https://seaborn.pydata.org/
- Tensorflow.js (for JavaScript): https://www.tensorflow.org/js

### Good Luck!