# Tutorial 4: Regression

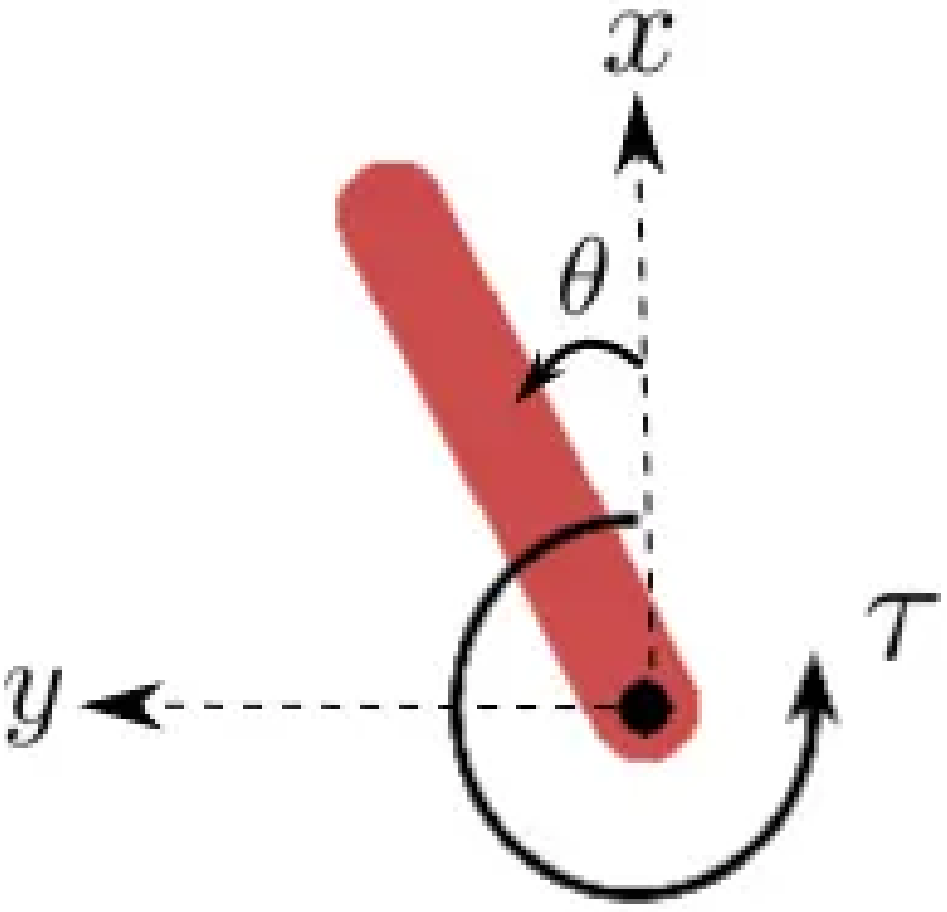
**Universidad Carlos III de Madrid**

**Grado en Ingeniería Robótica**

**Machine Learning**

Made by Ignacio González Meléndez (NIA 100522976)

& Lucas Beltrán Rúa (NIA 100523114)



**Exercise 1: Decision Tree Regressor**

1. If we interpret the dataset and compare the different parameters with the MSE, RMSE, MAE, we can see that the samples with *max_depth* = 5 have lower MSE values than the samples with *max_depth* = 10.

   Conversely, the ones that show a slightly lower MAE are the ones with *criterion = absolute_error, splitter = best and max_depth = 5*.

   In Figure 1 we can see the top 5 models, with cross validations, those are the ones with the lowest MSE.

| Model | criterion | splitter | max_ depth | min_sam ples_split | max_feat ures | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| Model_11 | squared_er ror | random | 5 | 5 | None | 1,51197862 | 1,229527 | 0,96034823 |
| Model_9 | squared_er ror | random | 5 | 2 | None | 1,54079156 | 1,241129 | 0,96905886 |
| Model_1 | squared_er ror | best | 5 | 2 | None | 1,54352797 | 1,241575 | 0,96526963 |
| Model_3 | squared_er ror | best | 5 | 5 | None | 1,54352797 | 1,241575 | 0,96526963 |
| Model_4 | squared_er ror | best | 5 | 5 | sqrt | 1,57321508 | 1,253907 | 0,97216346 |

*Figure 1: Top 5 models with cross validation*

2. After selecting the top 3 models from the previous section, which are models 11, 9 and 1, we performed a test process with the test data to see how they perform and analyse the error they give.

   Now, after testing these three models, without cross-validation, we can see in Figure 2 that models 11 and 9, which are our top 2 models in cross-validation selection, give us worse results. However, model 1 performs better without cross-validation leading to an MSE (test) of 0,88.

| MODEL | CRITERION | SPLITTER | MAX_DEPTH | MIN_SAMPLE_SPLIT | MAX_FEATURES | MSE (CV) | RMSE (CV) | MAE (CV) | MSE (TEST) | RMSE (TEST) | MAE (TEST) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | squared_error | best | 5 | 2 | None | 1,543528 | 1,241576 | 0,96527 | 0,887028 | 0,941822 | 0,87819 |
| 11 | squared_error | random | 5 | 5 | None | 1,511979 | 1,229528 | 0,960348 | 9,082943 | 3,013792 | 2,673064 |
| 9 | squared_error | random | 5 | 2 | None | 1,540792 | 1,24113 | 0,969059 | 9,097756 | 3,016249 | 2,676686 |

*Figure 2: Top 3 models performance in test*

3. As it was mentioned in section 2 from this exercise, the model that performed the best in test data prediction was model 1, which was the worst of these top 3 models under cross-validation.

   However, this can be because cross-validation provides an averaged estimation of model performance, while training and testing without cross-validation exposes the model to potential overfitting or underfitting depending on its parameters and the data distribution. In Figure 3 we have this difference quantified.

| MODEL | ΔMSE | ΔRMSE | ΔMAE |
|---|---|---|---|
| 1 | 0,6565 | 0,299754 | 0,08708 |

*Figure 3: Error difference between CV and testing in model 1*

**Exercise 2: Linear Regression**

1. In Figure 4 we can see the resulting errors from training a linear regression model. These values indicate that on average, the model's predictions differ from the true Action values by about 0.7 to 0.8 units.
   Considering that the Action variable ranges approximately from 0 to 4, this represents a relatively large error, around 20% of the total range.

   Therefore, the linear regression model is not very good. It fails to accurately represent the underlying nonlinear relationship between x, y, and Angular_velocity, which means that we do not have a good model.

| MODEL | MSE | RMSE | MAE |
|---|---|---|---|
| LinearRegression | 0,666 | 0,816 | 0,705 |

*Figure 4: Linear Regression Model Errors*

2. In Figure 5 we can see the distribution of the state space, we can clearly see that there is not a linear regression relationship because the Action points are dispersedly distributed, forming a kind of cylindrical form. Also, we can see that for negative values of X there are more dispersed points than for positive values.

   With all this, we can conclude that defining a good linear regression plane to define this model would not be possible, leading to the conclusion that the model is not working correctly.

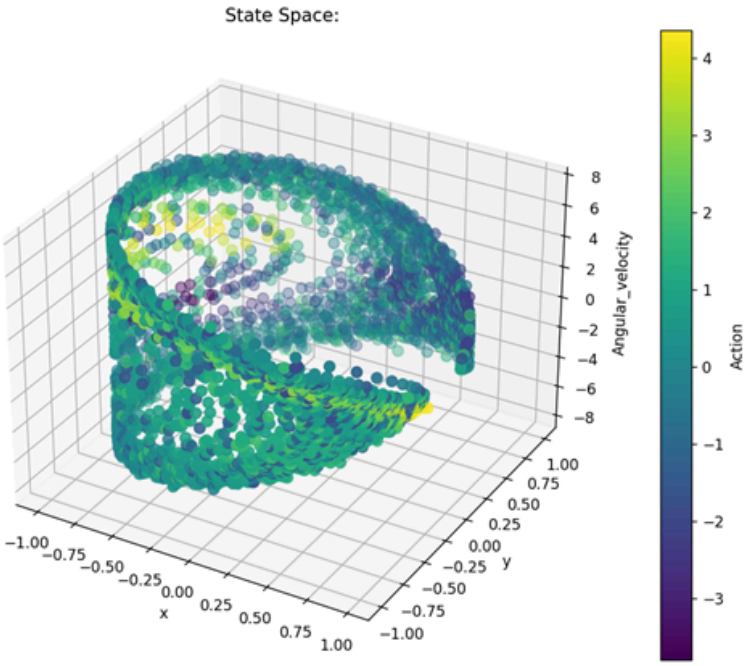



*Figure 5: 3D Plot of State Space*

**Exercise 3: Neural Networks**

1. In this exercise, we selected six parameter combinations with the objective of achieving a good performance model. All these six combinations were trained and then tested to compute the estimated errors in the prediction process. All these parameter combinations have been established with early_stopping set to True.

   In Figure 6 there is a table with all used models, including its parameters and the estimated errors. The table is organized in ascending order regarding MSE. The best model is model 4, with an estimated MSE of around 0,15 and the worst model is number 2 with an estimated MSE of 0,21. Thus, all these six models have a similar MSE, with a difference between the best and worst model of 0,06.

| Model_ID | Hidden_Layer: | Activation | Solver | Learning_Rate_Init | Early_Stopping | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| 4 | (100, 50) | relu | adam | 0,001 | True | 0,15191519 | 0,38976299 | 0,23015645 |
| 5 | (150, 100) | relu | adam | 0,001 | True | 0,17600267 | 0,41952672 | 0,30466898 |
| 3 | (-100) | relu | adam | 0,001 | True | 0,18415877 | 0,42913724 | 0,19559817 |
| 6 | (100, 50) | tanh | adam | 0,001 | True | 0,19878405 | 0,44585205 | 0,35822258 |
| 1 | (-50) | relu | adam | 0,001 | True | 0,20053935 | 0,4478162 | 0,20294214 |
| 2 | (-50) | tanh | adam | 0,001 | True | 0,20948992 | 0,45770069 | 0,29044248 |

*Figure 6: MLP models's estimated errors in test performance*

2. After performing the train and test processes in section 1, we chose model 4 as the best model because it is the one with the least MSE.

   Then, we repeated this train and test process with the selected model but in this case setting early_stopping to False. In Figure 7 we can see a table to compare the error estimation difference between the two cases. Also, we can quantify this difference: ΔMSE = 0,01568082; ΔRMSE = 0,01960971; ΔMAE = 0,07333739. By quantifying the error estimation difference, we can conclude that this model performs slightly better when early_stopping is set to True.

| Model_ID | Hidden_Layer | Activation | Solver | Learning_Rate_Init | Early_Stopping | MSE | RMSE | MAE |
|---|---|---|---|---|---|---|---|---|
| 4 | (100, 50) | relu | adam | 0,001 | True | 0,15191519 | 0,38976299 | 0,23015645 |
| 4_ES_FALSE | (100, 50) | relu | adam | 0,001 | False | 0,16758601 | 0,4093727 | 0,30349384 |

*Figure 7: Model 4's error estimation comparison regarding early_stopping*

**Exercise 4: Deploy the agents**

1. In this exercise, we implemented the best model from each previous exercise. After implementing the agents to solve the pendulum problem we can conclude that there is no model able to solve it because the accumulated reward of every agent is negative, which means that our best models were not able to successfully perform in the environment.

   In Figure 8 we can see the accumulated reward for each model. As all of them are around -1000 we can conclude that the models failed when trying to keep the pendulum upwards.

| Model | Avg Reward |
|---|---|
| MLP Regressor | -889,674415 |
| Decision Tree | -996,261368 |
| Linear Regression | -1163,41114 |

*Figure 8: Best Model's Rewards*

2. Among our best models, the one that performed the best in our prediction tests is the MLP Regressor model, which makes sense according to the data collected when deploying the agents (Figure) because MLP Regressor model has the best accumulated reward. Also, we can see in Figure that the model that performed the worst is the Linear Regression model, this is because as we saw in exercise 2, we cannot establish a linear relationship between the physical variables. Finally, the Decision Tree Regressor model performed with an accumulated reward between the MLP Regressor and the Linear Regressor model, but tending to be more like the MLP Regressor one, this is because we selected the best parameters with cross-validation trying to achieve the best prediction results.

   This classification, the one in Figure, was similar in all 5 simulations we performed, with MLP Regressor and Decision Tree Regressor models achieving really similar results.

**Conclusions:**

After deploying in exercise 4 the agents trained and tested in the previous exercises, we come to the conclusion that the best model for trying to solve the pendulum problem is MLP. This conclusion is based on the fact that our MLP model had the best performance in all simulations we did in exercise 4. However, to completely solve this problem we would have to improve the model by using more data, using a larger number of possible parameter combinations and setting a larger number of maximum iterations when training the agent.

**Code Development:**

On the other hand, regarding the code development, we decided that the best option was to print in the terminal tables to see the training and test results and then save those tables in .csv files. Loading data, training/testing and sorting processes were done similarly as done in the previous tutorials.