DSAIT4020 ELEMENTS OF STATISTICAL LEARNING
DEVELOPED BY ARATRIKA DAS AND TOM VIERING

Football Practice Project

November 17, 2025

You have to predict the wages of football players based on their age, nationality, skills, etc. Your task is to explore, preprocess, and analyze a dataset. Furthermore, you will train and finetune models for the task.

What will you learn? You will learn...

- To explore the given dataset and perform necessary preprocessing.
- To train and **fairly** evaluate regression models using Mean Absolute Error (MAE).
- To optimize model performance using hyperparameter tuning via gridsearch.
- To present your findings and what you did **and why** in a brief report.

Overview

We will use KNN Regression and linear regression trained using SGD as the two models, and the Mean Absolute Error (MAE) to measure their performance. When we mention 'performance' it will always refer to the MAE on unseen data (generalization performance). A fair estimate of the performance should reflect the performance on unseen data.

The dataset contains various attributes of football players (e.g., age, nationality, skills, and more). The column '`log_wages`' will be the regression target, this value represents $\log_{10}$(wage) of the football player. You must use Scikit-learn. We provide a template Jupyter Notebook.

We have an autograder where you have to submit your machine learning models' predictions. You also need to provide an estimate of the MAE you think your model has on new data. Both the predictions and your predicted MAE will be automatically graded and form 20% of your project grade. You are encouraged to submit early and as often as possible, this can help you find mistakes. Good luck, and feel free to ask if you have any questions!

Deliverable 1: Report

Submit a PDF report (1-2 pages) that includes the answers to the questions. Structure your report according to part A, B, C, D (see next page) and clearly indicate the title of each part. Do not repeat the questions in the report. Use running text and use figures and or tables to present your findings, text (also in figures) should be readable when printed to A4.

Assume the audience is another student in the class. You do not have to explain in detail what the dataset is and what the goal of the assignment is. The report **should** contain sufficient information such that the experiments are **reproducible**. This means that another student with your report should be able to reproduce your result.

Deliverable 2: Jupyter Notebook

- The Jupyter Notebook contains all your relevant code. If you like you can also use multiple files (e.g. `.py` files and multiple notebook files) if you find this easier.
- Ensure that the notebook runs when you restart the kernel and you click "run all cells".
- Make sure that the notebook has sections corresponding to this exercise so we can easily navigate it. Use markdown cells for this purpose.
- Make your notebook or code reproducible, providing a requirements file so we understand what to install and how to get it to run.

Grading

- **Runnable code (10%)**: The code should run without errors when rerunning the notebook from the top and should produce all results contained in the report.
- **Reproducible report (10%)**: The report has sufficient amount of detail so another student could reproduce your work.
- **Autograder (20%)**: The autograder grades the performance of your model's predictions, and it also grades the MAE you have provided.
- **Explanations and Arguments (20%)**: You explain concisely what you did and why.
- **Technical Correctness (30%)**: You follow the standards as discussed in the course to ensure your to ensure you have a reliable experiment and findings.
- **Presentation of Results (10%)**: Clear and concise presentation of your results, including tables and graphs (if necessary), and a discussion of findings. Graphs have labeled axes that are readable, and tables are readable and columns and rows are labeled.

Assignments

Stuck? Maybe consult the hints on the next page.

Part A - Data Exploration and Preprocessing

1. Load and inspect the dataset: discuss the meaning of the data and make sure that they are properly loaded. For example, are all data values sensible?
2. **Pipeline 1.** Perform the necessary steps to put the data in the right format for the machine learning algorithms. Explain the steps you take and why you take them.
3. **Pipeline 2.** Come up with a second preprocessing pipeline that differs from your first. Examples: come up with new features(s) (by yourself or by a feature extraction technique), reduce the number of features, or preprocess them in a different way. Explain the reasoning behind this pipeline and discuss how it works.

Part B - Regression with Default Hyperparameters

4. What is the simplest baseline model we should aim to beat? Or in other words; if you would have to make a guess for the salary without knowing anything about the football player, what would you guess? What is the MAE of such a guess?
5. Train the KNN and SGD Regressor with default hyperparameters and fairly estimate their performance for both preprocessing pipelines. Explain why the performance estimate is fair and how you estimated the performance.
6. Which pipeline performed the best? Use this pipeline for the next exercises.
7. Submit your your work to the autograder to check your work so far.

Part C - Tuning with GridSearch

8. For both KNN and SGD regression, use gridsearch to identify the best hyperparameters. Use a systematic way to tune hyperparameters that is reproducible. Explain your choice of hyperparameter search ranges and settings. Include sufficient details in the report so that another student can reproduce your experiment.
9. Include a training curve (performance versus epochs) to illustrate that the SGD regressor converges to a reasonable solution.
10. Explain why the performance estimate is fair and how you estimated the performance.
11. Compare the performance of both models before and after hyperparameter tuning.

Part D - Conclusion

12. Reflect on the impact of both preprocessing and hyperparameter tuning on the performance of the models.
13. Select your best final model and use it in the autograder.

Hints:

These documentation pages may provide further hints:

- Dummy estimators
- Pre-processing continuous features
- Pre-processing categorical features
- Hyper-parameters tuning
- Cross-validation

Frequently Occuring Problems:

- I score extremely low in the autograder. What can be wrong? Probably the preprocessing went wrong. Please check this in detail.
- The hyperparameter tuning did not improve the performance of the models? This probably means you did not perform the hyperparameter search correctly.
- For some hyper-parameters we know that they are always positive numbers and they can go to fairly large ($1e16$) or small ($1e-16$) values. In such cases you should use (start with) a logarithmic scale for tuning: 0.1, 1.0, 10, 100, ...
- The training curve for the SGD Regressor looks very strange or not converged? This probably means that you did not consider enough hyperparameters for tuning.