TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Technische Universität Wien

Department of Informatics

186.686 Visual Data Science

Main Factors Contributing to Powerlifting Results

Lab Part 3: **Model & Report**

by

Gunnar Sjúrðarson Knudsen
12028205

January 14, 2022

Contents

List of Figures

1 Introduction

This is a continuation of the same project, where I am to see if there is a correlation between population height and powerlifting results. See previous reports for data description.

2 Model

This step is done in Python, and can be seen in the file `Model_and_firstVisualizaiton.ipynb`

2.1 Features and Targets

Originally I wanted to correlate height with bodyweight and/or total lifted. However, as the data wasn't available, I had to augment my dataset with height based on (unfair) assumptions. As such, I estimated height based on `country`, `gender`, `age`, and `birth year`.

If I was to try to predict height from my given dataset, I would therefore in reality only be predicting a combination of those variables. A good model would therefore give (close to) 100% accuracy, and wouldn't provide any new insights. This could be mitigated by removing all those columns from the training dataset. However, this would severely limit the model, and still wouldn't lead to specifically interesting insights.

I therefore made the last minute executive decision to try to predict `bodyweight` based on the remaining variables. I feel like this is still one of the core attributes, and therefore the model would be (vaguely) interesting. Furthermore, this course is focusing on visualizing insights, and with a more interesting model, I was able to create more meaningful visualizations.

I did make sure to remove variables that had weight hidden (such as `Wilks, Dots, ...`. These are calculated fields, that have weight encoded in it.

2.2 Train/Test/Val Split

This was done using `train_test_split` from package `sklearn.model_selection`. I decided to go for a split of 60%/20%/20%. However, Argument could have been made that since the datasize was so big, the training split could be bigger (Possibly 80%-90%), and we would still have enough data for test and validation.

2.3 Modeling Approach

As the feature I'm targeting is an interval variable of continuous shape, I'm going for a regression. I had a strict wish that I wanted an explainable model, which meant that (deep) Neural Networks were out of the question. Argument could have been made that the dataset would be decently well linearly separable in higher dimensions, and a *Support Vector Machine* using *the kernel trick* would give good results. However, as I've seen good performance with it before, I decided to go with a **Random Forrest** ensemble.

2.4 Train a model

I first train a full forest with 100 estimators, and a max depth of 10, in order to avoid too much overfitting.

After this was done, I evaluated the model, and extracted the variables with the highest `feature importance`, in an attempt to create more explainable model, and I re-trained the model using only these features.

2.5 Evaluation and Visualisation of Performance

As mentioned, I first trained a full model. When evaluating on the test set, I got these results:

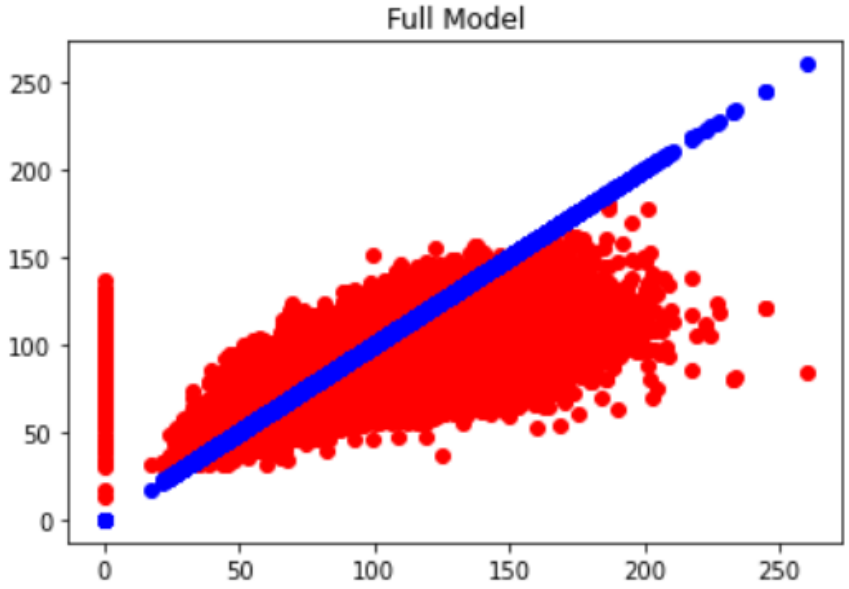

Figure 1: Prediction performance on Test for full model

MAE	RMSE
12.06	16.90

Table 1: Performance of full model

Looking at the Feature Importance, we see that only a few variables are truly "important" for the model:

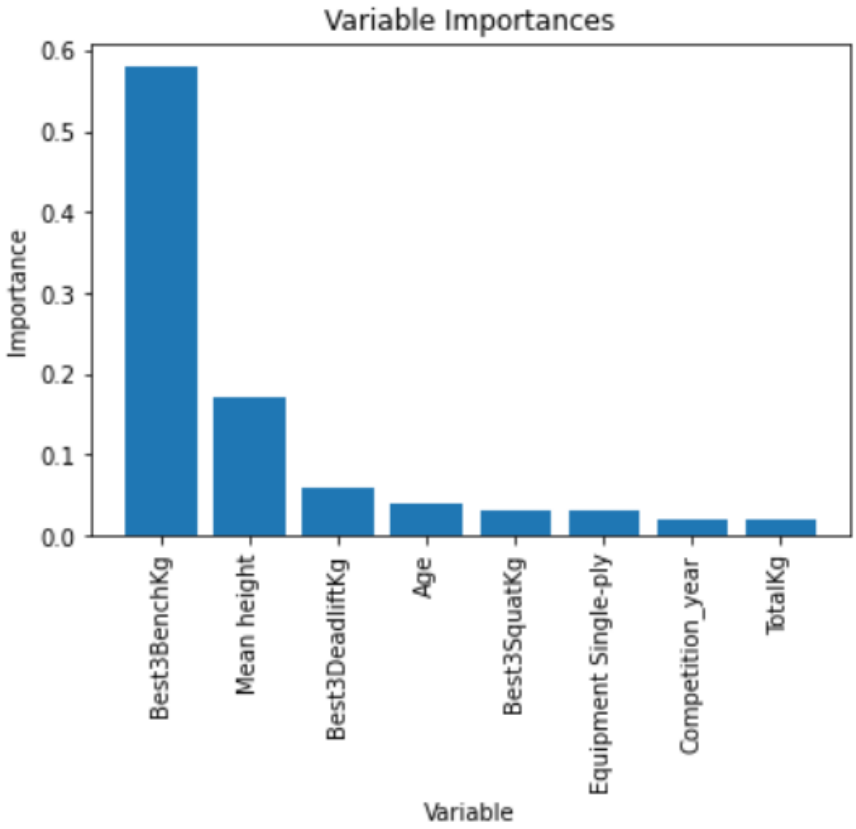

Figure 2: Variable Importance from Full Model

Variable	Importance
Best3BenchKg	0.58
Mean height	0.17
Best3DeadliftKg	0.06
Age	0.04
Best3SquatKg	0.03
Equipment Single-ply	0.03
Competition_year	0.02
TotalKg	0.02

Table 2: Most important features

Quite interesting was that Gender didn't make it onto the list. However we also notice that gender is hidden in both `Mean height`, as well as in the ratio between `Best3BenchKg` and `Best3SquatKg` (Testosterone levels is known to highly affect this ratio).

When rerunning the Simpler model on only these features, we got these results:

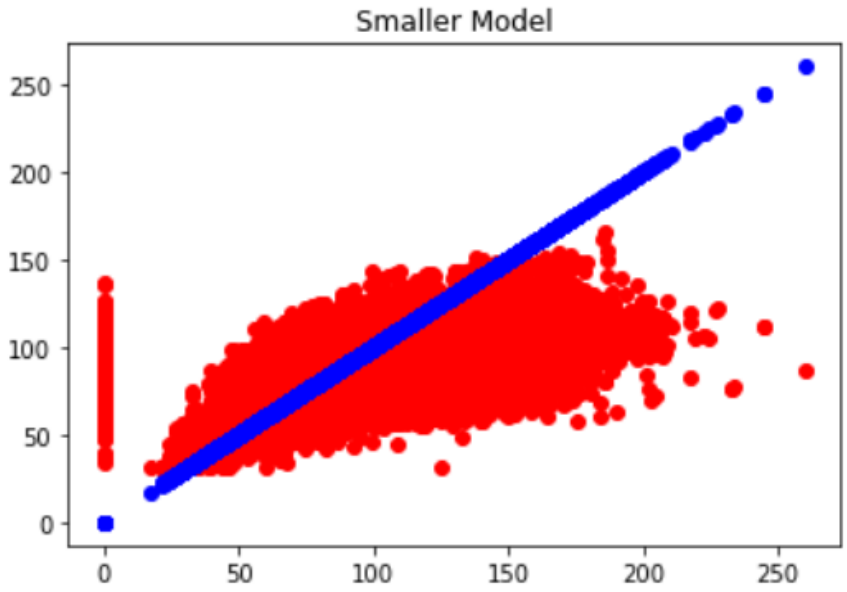

Figure 3: Prediction performence on Test for simpler model

MAE	RMSE
12.22	17.12

Table 3: Performance of simpler model

We see that our error rate increased slightly. However, I find this increase acceptable, as we now have a model that is simple enough to be able to draw decision trees, in order to understand the model. Sample of a decision tree can be seen on appendix B

3 Report

3.1 Technical Stack

As the previous steps were all done in **Python 3**, I decided to continue in the same manner. However, as the requirement was interactive visualizations, I switched a lot of the plots out to **plotly**, and interactions were created with **Dash**.

3.2 Chart Selection

As discussed in the lecture, I've selected a varying selection of different chart types. These are further discussed in section 3.4

3.3 Visual Encodings

I decided for a dark theme, as the main idea for this dashboard is interactive exploration on a monitor, and not as a printed media. I feel a darker palette to be more pleasing on the eyes in the long term for this use.

My visualizations all focuses on single groups, and color is chosen from appropriate `plotly` palette.

3.4 (Four) Types of Visualizations

In this report, I've utilized the following plots.

3.4.1 KDE-plot with 2 Histogram axis:

First we see a KDE plot on the bottom right side. We see KDE plots shaded by a grouping. On the axis we see histograms for the same groupings. *X-axis-*, *Y-axis-*, and *grouping*-variables can all be selected in the dropdown above, for interactive exploration.

3.4.2 Stacked Area chart Timeline

In the bottom middle, we see how the average totalt (split into the three lifts) have progressed over time.

3.4.3 Scatterplot with brushing and linking

On the top middle, we see a simple scatterplot of `Total` over `bodyweight`. This chart is grouped by continent, and has extra interractability incorporated. When hovering on a point, we get an updated table below it, that can further describe the point. This scatter plot also incorporates brushing and linking, which updates the 3D Scatterplot 3.4.5 as well as the Table 3.4.7 and the MAPE 3.4.6.

3.4.4 Hover table

As mentioned above, when hovering on a point, we get extra information on it here.

3.4.5 3D Scatterplot

This plot shows how our trained model behaves on the brushed datapoints. It shows predicted vs actual bodyweight, with the extra axis of `Mean height` and `Total`.

3.4.6 MAPE Calculator String

The MAPE measure of model performance is calculated for the selected datapoints, and printed on the middle right.

3.4.7 Table of Selected data

The brushed datapoints gets further clarified in this table. Note that this table has a horizontal scrollbar, as I wanted to show extra information.

3.5 Included Interactions

3.5.1 Filters

A large selection of filters were added to the top left side of the dashboard. There is a "meta-filter" called "DD Filters". When this is enabled, the dropdown lists get filtered, so that only options that exists in the other filters get chosen. If this is disabled, all drop-down selections can be multiple choice, and the plots adjust accordingly.

3.5.2 Zooming

All four plots have zooming and panning enabled. The 3D plot also has Rotation enabled.

3.5.3 (BONUS) Brushing and Linking

The plot described in 3.4.3 has Brushing enabled. when datapoints are brushed, 3.4.5, 3.4.6, and 3.4.7 are linked and updated.

4 Conclusion

Throughout this project, i was sadly not able to visualize my original hypothesis that "Weight classes are just height classes in disguise", as there wasn't appropriate datasets available. I did instead gain a lot of new insight by augmented height based on (partially) reasonable assumptions.

4.1 Learnings of this project

Great care was put into selecting the correct type of visualization for each insight, and having never worked with `plotly`/`Dash`, I feel like I now have a new tool in my belt.

4.2 Limitations of current work

The current dashboard is only a single page. I would like to implement tabbing, so we could investigate more in depth with more visualizations. I would also have loved to incorporate more callbacks for different interactions. Finally, there is a limitation in current version of dash, where plots don't always update on callback, unless they are interacted with. I suffered from this in both the 3D-scatterplot, as well as on the KDE plot, but was unable to find a solution.

A Screenshot of interative dashboard

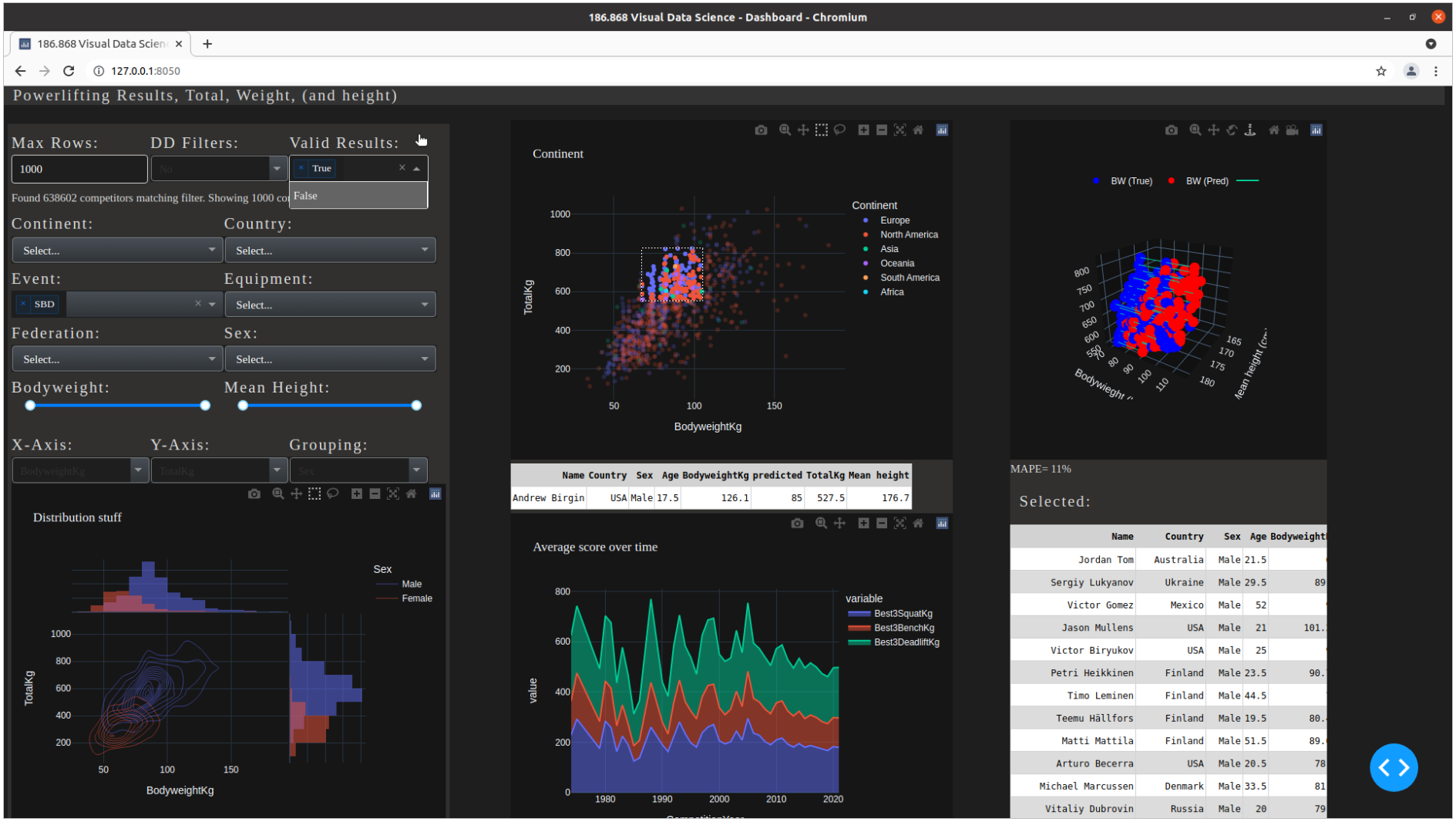

Figure 4: Full Screenshot

B Explainable AI: Decision Tree number 1

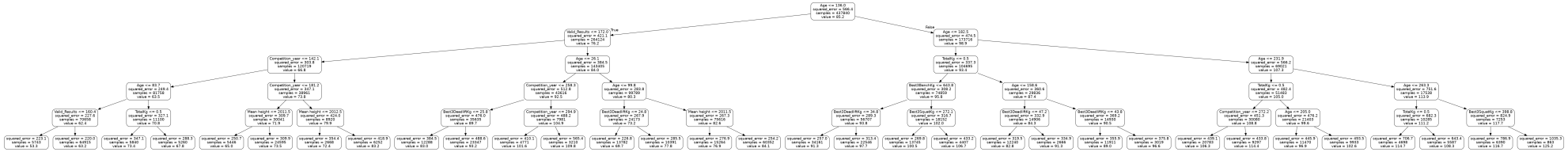

Figure 5: Decision Tree number 1