Plan:

* KMeans mit allen drei Arten: Laevigata, Monogyna, Rhipidophylla
* Feature Importance der drei Arten
* Neue Methoden?
* PCA mit allen drei Arten
* Welche Merkmale teilen sie in Gruppen ein? Passen die Loadings dazu? Mit Literatur vergleichen?
* Center finden und so genaue Merkmale feststellen
* Kombinieren von PCA und KMeans
* Laevigata von den anderen Arten trennen

04.10.2021

Code-File: KMeans\_Crataegus\_Laevigata\_Rhipidophylla\_Monogyna.py

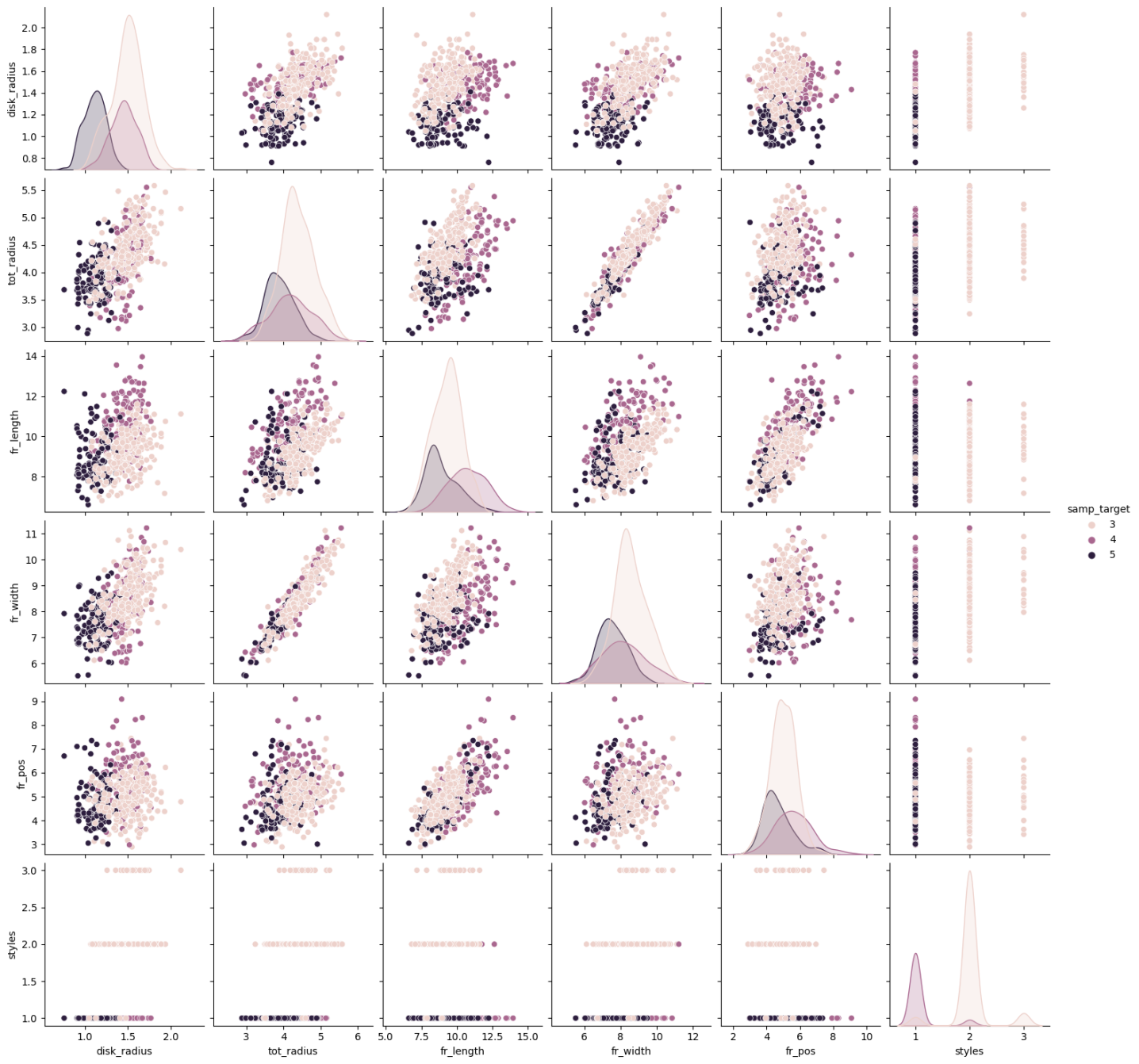
Loading dataset “RearrangedTable.csv” and kept only data regarding Laevigata, Monogyna and Rhipidophylla. Species are encoded:

|  |  |
| --- | --- |
| 3 | Laevigata |
| 4 | Rhipidophylla |
| 5 | Monogyna |

Scatterplot with Seaborn:

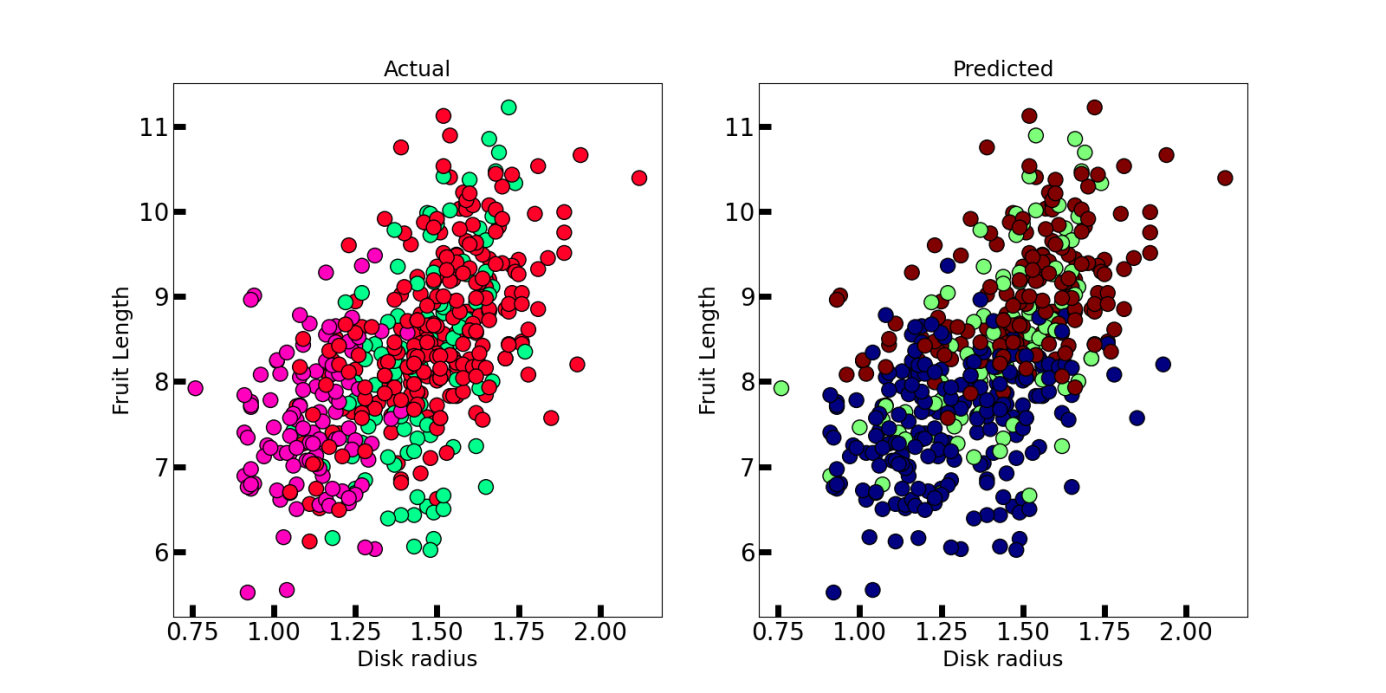
sns\_plot = sns.pairplot(df\_LRM, hue='samp\_target', height=2.5)

sns\_plot.savefig("Scatterplot\_CombinationallFeatures\_Laev\_Mon\_Rhip.png")

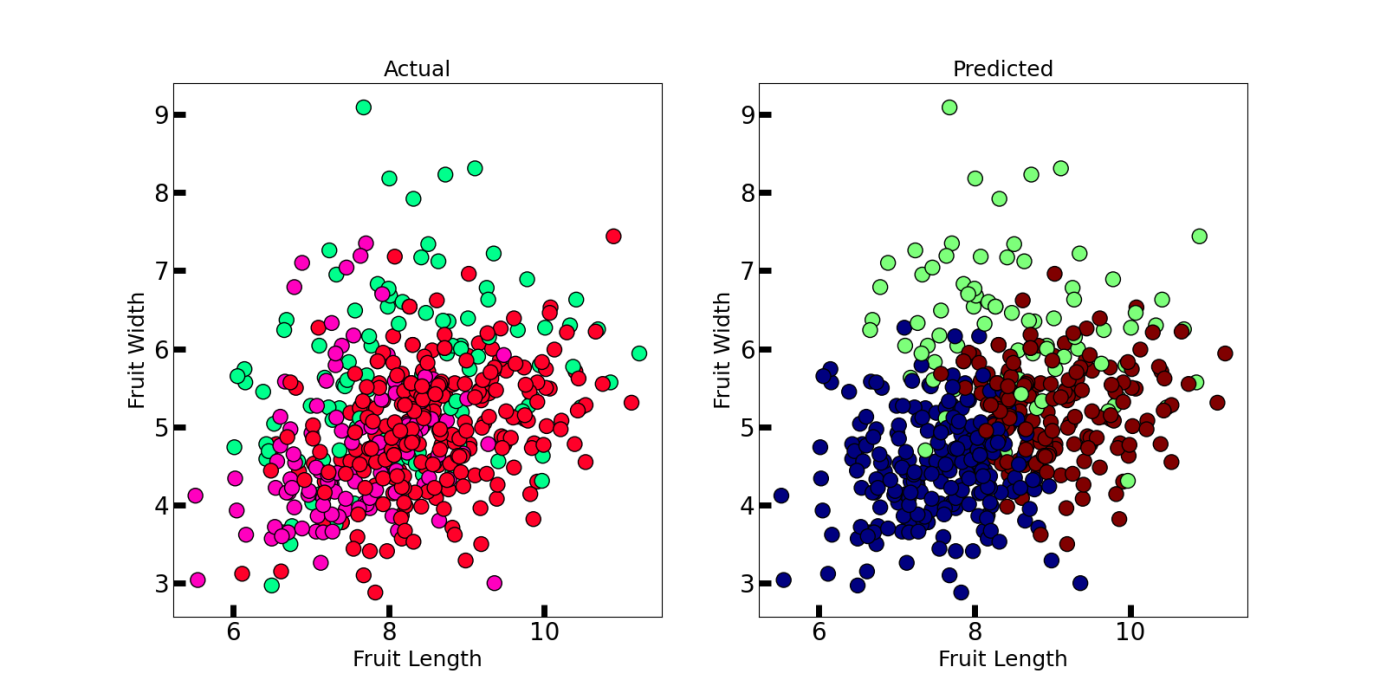


Scatterplots show that the species cannot really be separated only by the features of the fruits. Maybe disk radius and fruit length or fruit length and fruit width. Also styles indicate a lot but only for Monogyna. The distribution graphs however show that Laevigata is quite different compared to the other ones. It’s probably easier to make a separation only with two of the species, where one of them is Laevigata.

KMeans with disk radius – fruit length and fruit length – fruit width:



It looks more like there are only two clusters which does make sense according to the distribution graphs.



KMeans\_Crataegus\_Laevigata\_Rhipidophylla\_Monogyna\_ver2.py is a different version, but I didn’t use the output.

06.10.2021

Feature Importance:

Code: FeatureSelection\_Laev\_Mon\_Rhyp.py

Tutorial: <https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b>

Loading dataset “RearrangedTable.csv” and kept only data regarding Laevigata, Monogyna and Rhipidophylla. Species are encoded:

|  |  |
| --- | --- |
| 3 | Laevigata |
| 4 | Rhipidophylla |
| 5 | Monogyna |

Correlation:

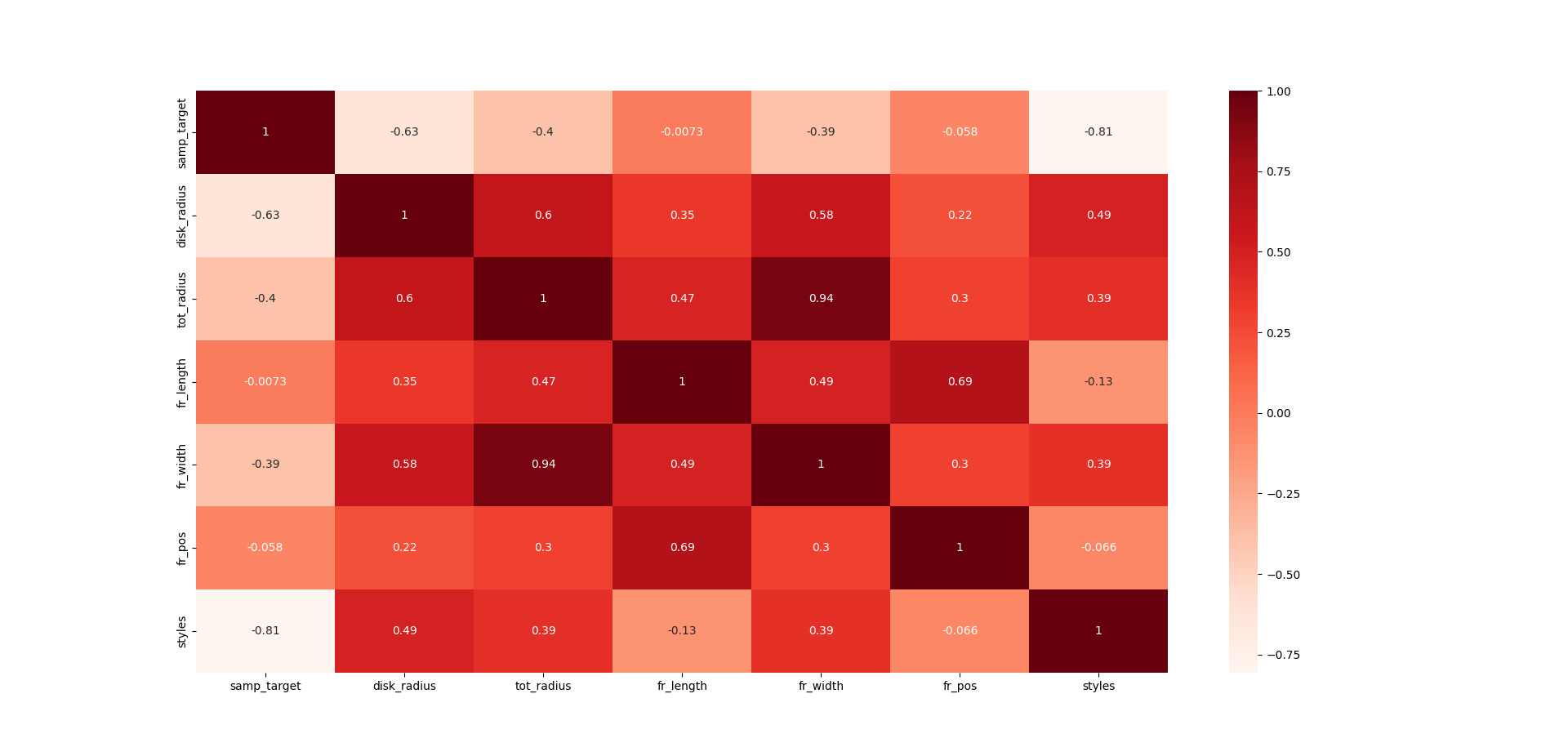
The correlation coefficient has values between -1 to 1

— A value closer to 0 implies weaker correlation (exact 0 implying no correlation)

— A value closer to 1 implies stronger positive correlation

— A value closer to -1 implies stronger negative correlation

Using Pearson Correlation:



High correlation (above 0.5):

|  |  |  |
| --- | --- | --- |
| First feature | Second feature | Corr. coefficient |
| Disk radius | Fruit width | 0.58 |
| Disk radius | Total radius | 0.6 |
| Total radius | Fruit width | **0.94** |
| Fruit length | Fruit position | 0.69 |

Correlation with output variable:

Output:

samp\_target 1.000000

disk\_radius 0.626925

styles 0.806804

Name: samp\_target, dtype: float64

Disk radius and styles are correlated with the output variable “samp\_target”

Does it make sense to use? Used samp target for the tutorial’s “MEDV” which is a float.

disk\_radius styles

disk\_radius 1.000000 0.470509

styles 0.470509 1.000000

Not above 0.5, I would keep both.

Wrapper Method:

The performance metric used here to evaluate feature performance is pvalue. If the pvalue is above 0.05 then we remove the feature, else we keep it.

Model 1: const 1.435213e-136

disk\_radius 1.523325e-22

tot\_radius 2.983396e-01

fr\_length 3.799661e-01

fr\_width 9.991064e-01

fr\_pos 2.635867e-02

styles 2.592749e-71

Backward Elimination:

['disk\_radius', 'tot\_radius', 'fr\_pos', 'styles']

Fruit length and width are both greater than 0.05, therefore the features should be removed according to this analysis

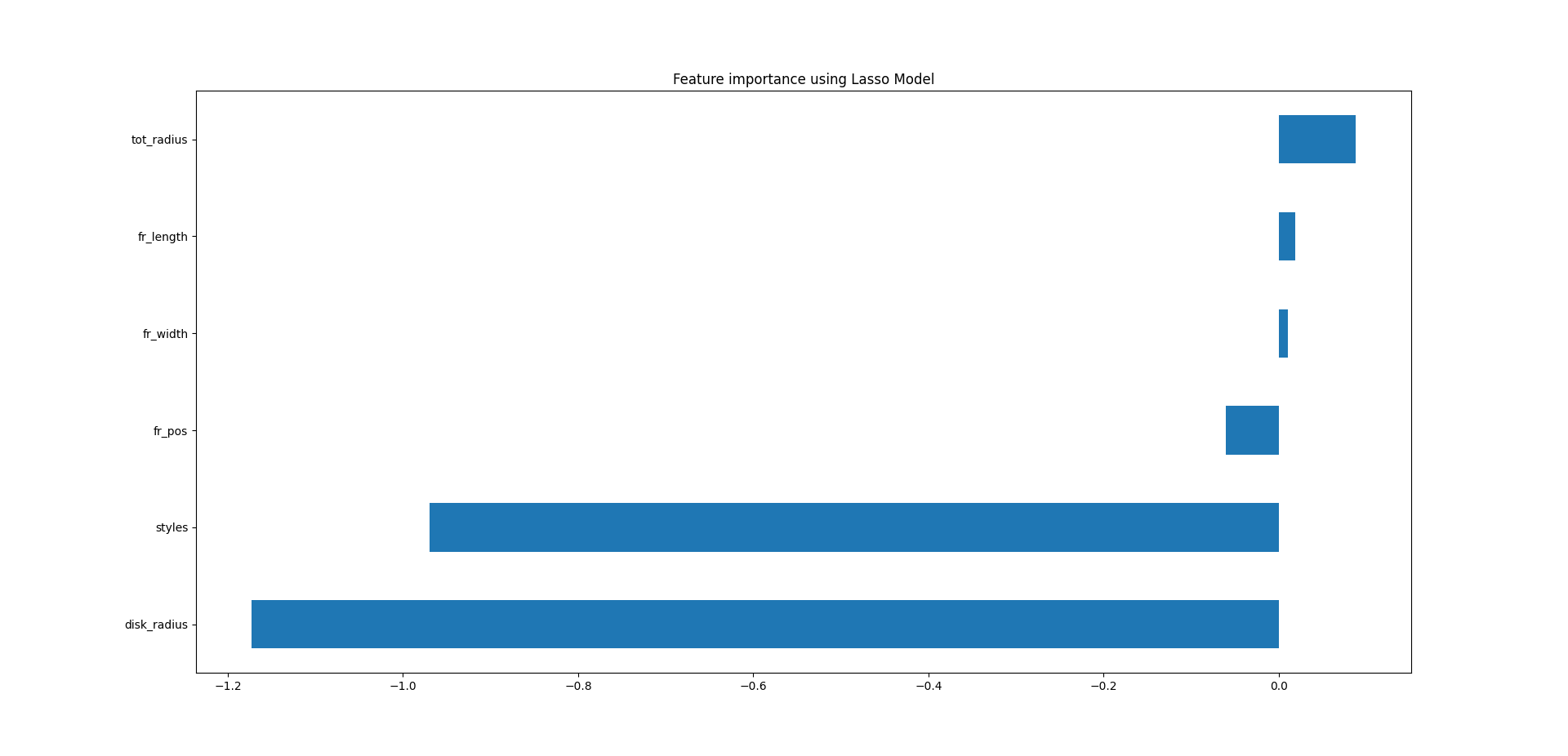
Embedded Method

Embedded methods are iterative in a sense that takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration. Regularization methods are the most commonly used embedded methods which penalize a feature given a coefficient threshold.

Here we will do feature selection using Lasso regularization. If the feature is irrelevant, lasso penalizes it’s coefficient and make it 0. Hence the features with coefficient = 0 are removed and the rest are taken.

Output:

Lasso picked 6 variables and eliminated the other 0 variables

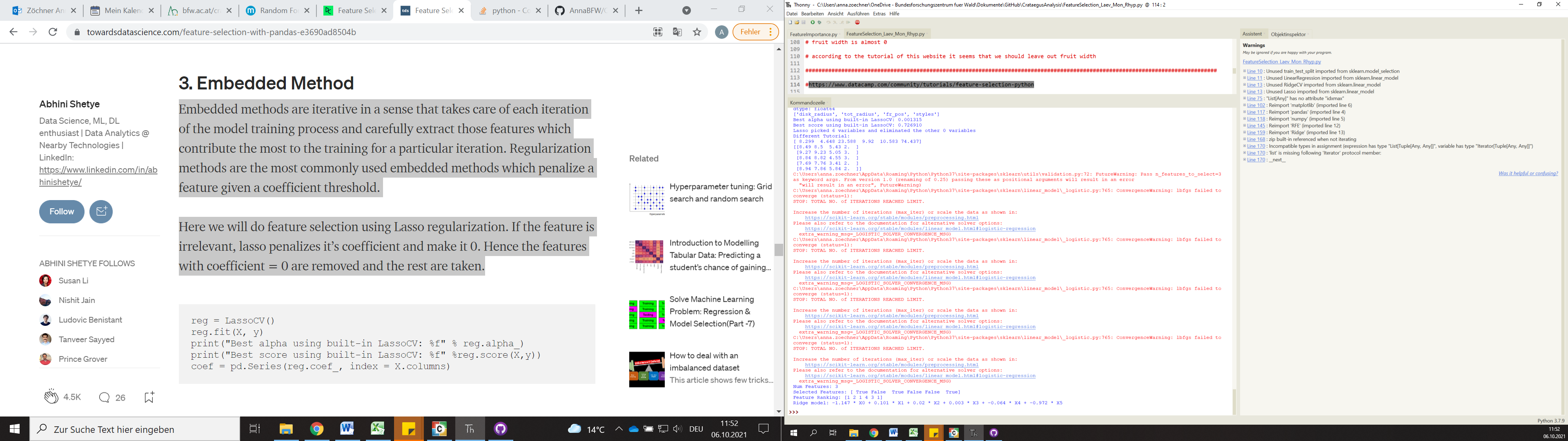


According to Lasso we should keep all features. However, the fruit width is really close to 0 which is why I would suggest leaving it as it also fits to the analysis before.

Different Tutorial:

<https://www.datacamp.com/community/tutorials/feature-selection-python>

I also tried a different tutorial but I got the following warning:



As it was just a future warning, I still got results.

Output:

Num Features: 3

Selected Features: [ True False True False False True]

Feature Ranking: [1 2 1 4 3 1]

Ridge model: -1.147 \* X0 + 0.101 \* X1 + 0.02 \* X2 + 0.003 \* X3 + -0.064 \* X4 + -0.972 \* X5

According to this we should only keep: disk radius, fruit length, styles

Which makes sense as total radius and fruit width are highly correlated and don’t provide “additional” information.

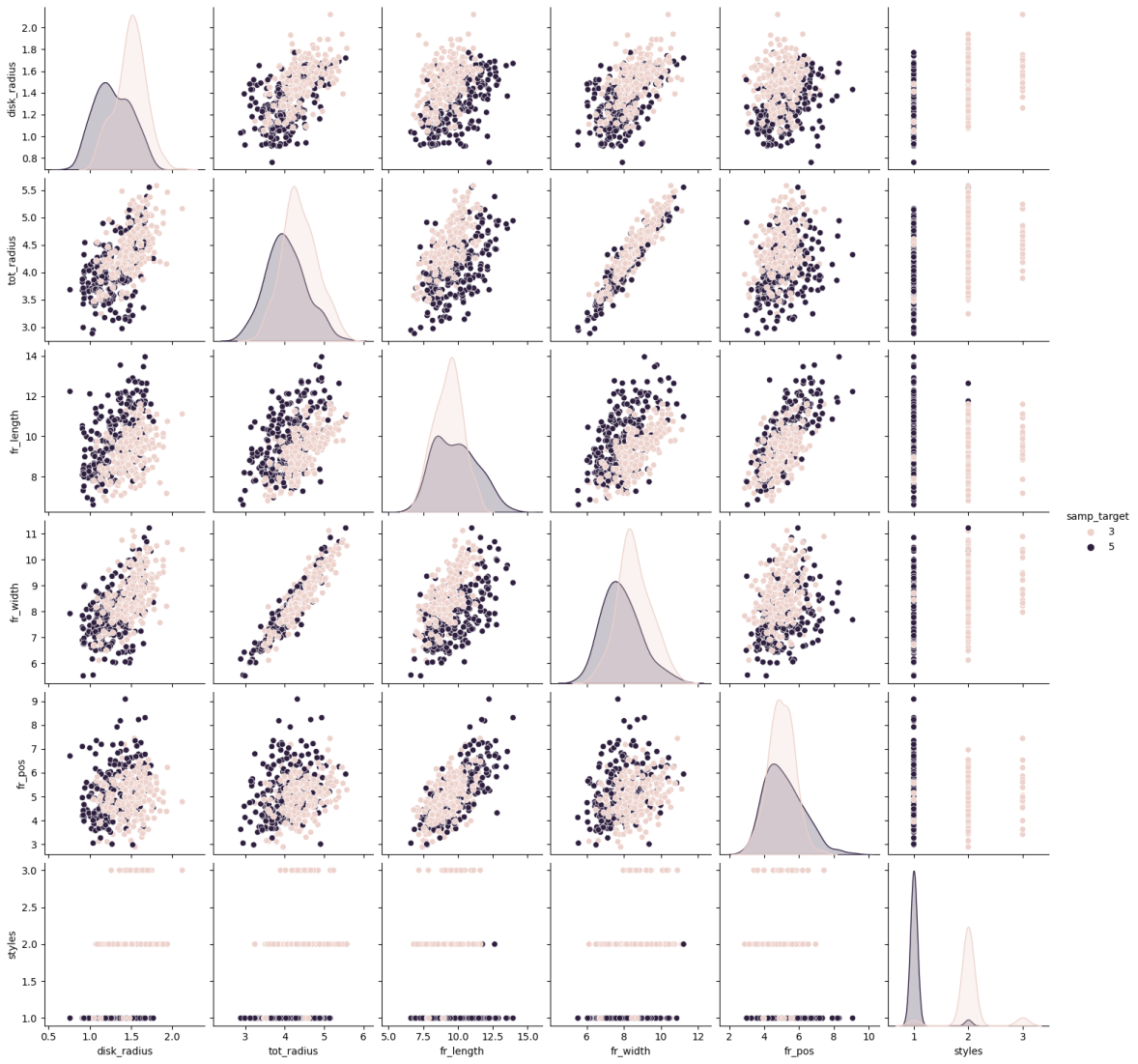
07.10.2021

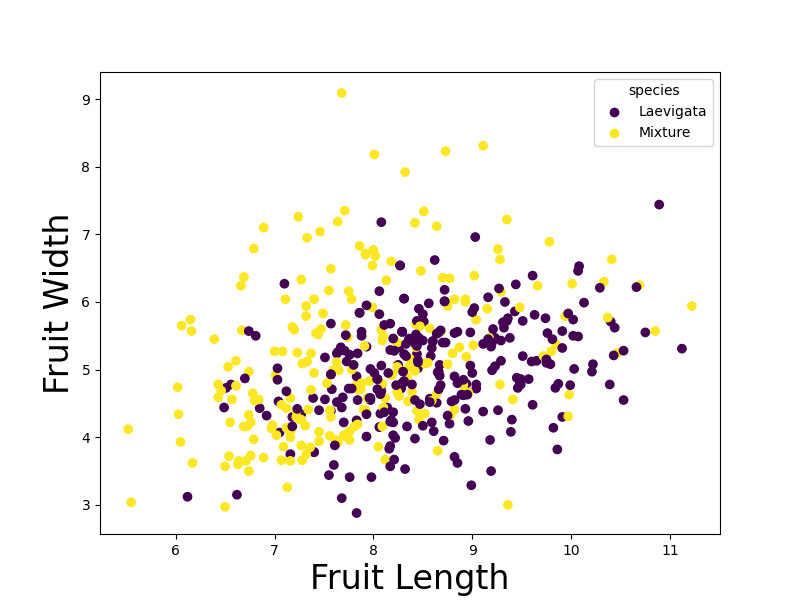
Laevigata seems to differentiate from the two other species a lot.

Idea: Put Monogyna and Rhipidophylla together and see if there are two clusters

Same Code as before

* KMeans\_Laevigata\_against\_MonogynaRhipidophylla.py





Did not improve anything

Looked again at Feature Selection – Analysis:

Every method says something a bit different. However, fruit width and total radius are highly correlated. Those measurements are actually the same. If we keep both of them it is redundant. Not sure what is the best to do with the fruit position. I would like to keep it as it says something about the shape.

Next Step: PCA with disk radius, fruit length, fruit position, styles

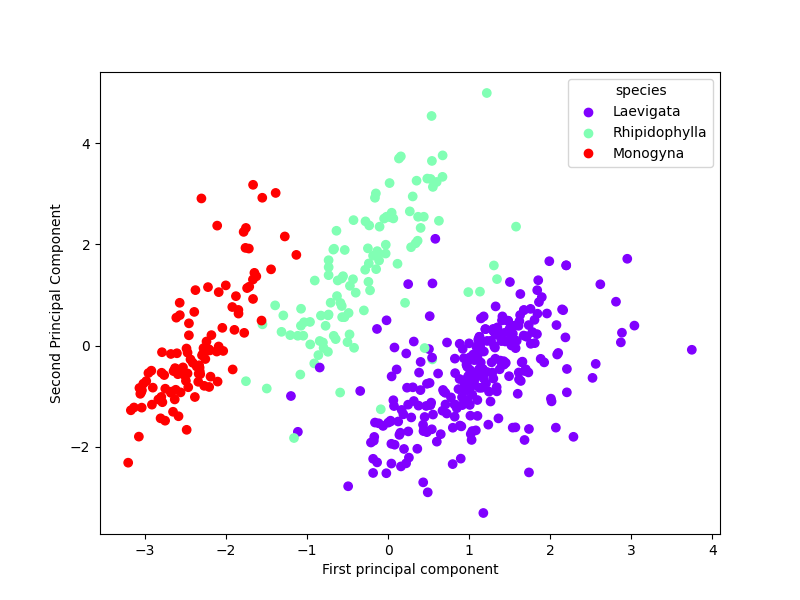
PCA with all species

Features according to previous analysis:

Left out fruit width and total radius because they are highly correlated

Dropped fruit width because of feature selection analysis

Left sepals as always



Best output so far. Groups are definitely visible. However some Rhipidophylla and Laevigata fruits are overlapping.

Next: find the centre of each cluster, make circle around it and detect the “perfect” fruits for each species