1. Preparing the data¶

1.1 Sniffing around the dataset (exploration)

•I opened the document in notebook and looked at the data to have general impression of the dataset.

•I found one value with minus in front, I just deleted the '-' sign.

•I also found 'NA' value, which later I replaced with the mean value of the column

•I found that all values in 'petal.width' column are inside quote marks which gonna make them be read as strings.

Looking at basic information on dataset.

Using the statistic functions I checked the max and min values to see are there any outliers, using 'groupby' I also checked the count of each species samples number in the dataset.

There are no outliers, the column 'Species' has the same count for every category which gonna be predicted.

Handling the data

I changed the 'Petal.Width' from string data type to float.

I filled the missing value with mean value from the column. The data is ready to be used in model.

2. Preparing the model

I created dependent and independent values datasets

Species is the dependent value - the one that's gonna be predicted. The rest of values are the independent which are the basis to predict the 'Species'.

I split the data into training and testing sets

I decided to use the Random Forest Classifier as I know this model should perform well and fast on this small dataset.

About Random forest.

- Random forest is an ensemble method, a technique that combines the predictions from multiple algorithms together – decision trees, to make more accurate predictions than any individual model.
- Decision trees tend to overfit; it is very easy to go too deep in the tree, and thus to fit details of the particular data rather than the overall properties of the distributions they are drawn from.
- Multiple overfitting estimators can be combined to reduce the effect of overfitting.
- Ensemble of randomized decision trees is known as a random forest. Bagging makes use of an ensemble of parallel estimators, each of which overfits the data, and averages the results to find a better classification.

2. 1 Training the model

The model performance

I checked the performance with different numbers of estimators. The defualt is 10, I tried increasing the number up to 100, but it didn't change the performance of the model, so I left the defualt number. Reading the documentation for the RandomForestClassifier I found that 0 and 42 are the optimal values of 'random_state'. I left the rest of features default as the model performed well.