When features are collinear, permutating one feature will have little effect on the models performance because it can get the same information from a correlated feature. One way to handle multicollinear features is by performing hierarchical clustering on the Spearman rank-order correlations, picking a threshold, and keeping a single feature from each cluster. First, we plot a heatmap of the correlated features:

fig, (ax1, ax2) = [plt.subplots](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html#matplotlib.pyplot.subplots)(1, 2, figsize=(12, 8))

corr = [spearmanr](https://docs.scipy.org/doc/scipy/reference/reference/generated/scipy.stats.spearmanr.html#scipy.stats.spearmanr)(X).correlation

*# Ensure the correlation matrix is symmetric*

corr = (corr + corr.T) / 2

[np.fill\_diagonal](https://numpy.org/doc/stable/reference/generated/numpy.fill_diagonal.html#numpy.fill_diagonal)(corr, 1)

*# We convert the correlation matrix to a distance matrix before performing*

*# hierarchical clustering using Ward's linkage.*

distance\_matrix = 1 - np.abs(corr)

dist\_linkage = [hierarchy.ward](https://docs.scipy.org/doc/scipy/reference/reference/generated/scipy.cluster.hierarchy.ward.html#scipy.cluster.hierarchy.ward)([squareform](https://docs.scipy.org/doc/scipy/reference/reference/generated/scipy.spatial.distance.squareform.html" \l "scipy.spatial.distance.squareform" \o "scipy.spatial.distance.squareform)(distance\_matrix))

dendro = [hierarchy.dendrogram](https://docs.scipy.org/doc/scipy/reference/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram)(

dist\_linkage, labels=data.feature\_names.tolist(), ax=ax1, leaf\_rotation=90

)

dendro\_idx = [np.arange](https://numpy.org/doc/stable/reference/generated/numpy.arange.html#numpy.arange)(0, len(dendro["ivl"]))

ax2.imshow(corr[dendro["leaves"], :][:, dendro["leaves"]])

ax2.set\_xticks(dendro\_idx)

ax2.set\_yticks(dendro\_idx)

ax2.set\_xticklabels(dendro["ivl"], rotation="vertical")

ax2.set\_yticklabels(dendro["ivl"])

fig.tight\_layout()

[plt.show](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.show.html#matplotlib.pyplot.show)()

