GPML kernel: 66\*\*2 \* RBF(length\_scale=67) + 2.4\*\*2 \* RBF(length\_scale=90) \* ExpSineSquared(length\_scale=1.3, periodicity=1) + 0.66\*\*2 \* RationalQuadratic(alpha=0.78, length\_scale=1.2) + 0.18\*\*2 \* RBF(length\_scale=0.134) + WhiteKernel(noise\_level=0.0361)

Log-marginal-likelihood: -326.810

Learned kernel: 0.00316\*\*2 \* RBF(length\_scale=1e+05) + 1.38\*\*2 \* RBF(length\_scale=3.82e+04) \* ExpSineSquared(length\_scale=2.27, periodicity=1) + 0.00316\*\*2 \* RationalQuadratic(alpha=0.412, length\_scale=1e+05) + 0.262\*\*2 \* RBF(length\_scale=0.00299) + WhiteKernel(noise\_level=0.313)

Log-marginal-likelihood: -88.336

C:\Users\eveba\.conda\envs\Machine\_Learning\lib\site-packages\sklearn\gaussian\_process\kernels.py:402: ConvergenceWarning: The optimal value found for dimension 0 of parameter k1\_\_k1\_\_k1\_\_k1\_\_constant\_value is close to the specified lower bound 1e-05. Decreasing the bound and calling fit again may find a better value.

warnings.warn("The optimal value found for "

C:\Users\eveba\.conda\envs\Machine\_Learning\lib\site-packages\sklearn\gaussian\_process\kernels.py:411: ConvergenceWarning: The optimal value found for dimension 0 of parameter k1\_\_k1\_\_k1\_\_k2\_\_length\_scale is close to the specified upper bound 100000.0. Increasing the bound and calling fit again may find a better value.

warnings.warn("The optimal value found for "

C:\Users\eveba\.conda\envs\Machine\_Learning\lib\site-packages\sklearn\gaussian\_process\kernels.py:402: ConvergenceWarning: The optimal value found for dimension 0 of parameter k1\_\_k2\_\_k1\_\_constant\_value is close to the specified lower bound 1e-05. Decreasing the bound and calling fit again may find a better value.

warnings.warn("The optimal value found for "

C:\Users\eveba\.conda\envs\Machine\_Learning\lib\site-packages\sklearn\gaussian\_process\kernels.py:411: ConvergenceWarning: The optimal value found for dimension 0 of parameter k1\_\_k2\_\_k2\_\_length\_scale is close to the specified upper bound 100000.0. Increasing the bound and calling fit again may find a better value.

warnings.warn("The optimal value found for "

