Work Report

Internship at Kiel University, Mathematics and Computer Science Department

Name: Parshva Mody

Duration: 8 Weeks

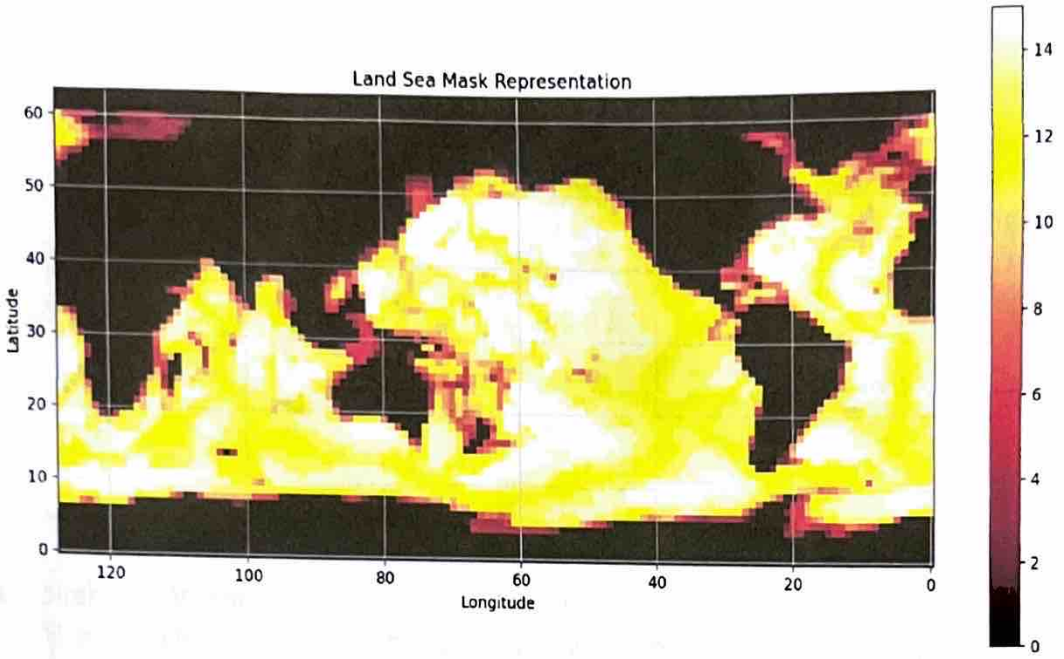
Objective:

During my two-month internship at Kiel University, I worked in the Mathematics and Computer Science department to perform statistical data analysis tasks using Python. The primary goal was to gain practical experience in data analysis and apply my skills to real-world problems. My main goal is to successfully reconstruct marine ecosystem data using singular value decomposition.

Tasks Performed:

Data Preparation:

- I successfully read 'N' output.petsc files from the folders Parameter 00000 to Parameter 00099. I utilized an optimized method for reading these files to ensure efficiency and avoid repetitive code.
- The 'read_PETSc_vec' function was employed to read the files, streamlining the data acquisition process.
- Additionally, I read the file 'landSeaMask.petsc' using the 'read_PETSc_matrix' function, which provided essential information for the subsequent analysis.
- Below is a plot of the LandSeaMask file.



Data Reshaping and Reduction:

- I reshaped all the data sets into a 3D vector using the 'reshape_vector_to_3d' function to facilitate analysis. This transformation utilized the 'landSeaMask.petsc' file as a reference.
- After reshaping, I focused on the upper surface of the data, extracting the relevant information.

Transformation and Matrix Construction:

- Following the reduction, I transformed the data into a 1D vector, preparing it for further processing.
- I constructed a matrix to store all the data sets, excluding the data set from the folder 'Parameter 00000', the test dataset in this case. Each data set corresponds to one line in the matrix.

Data Cleaning and Regularization:

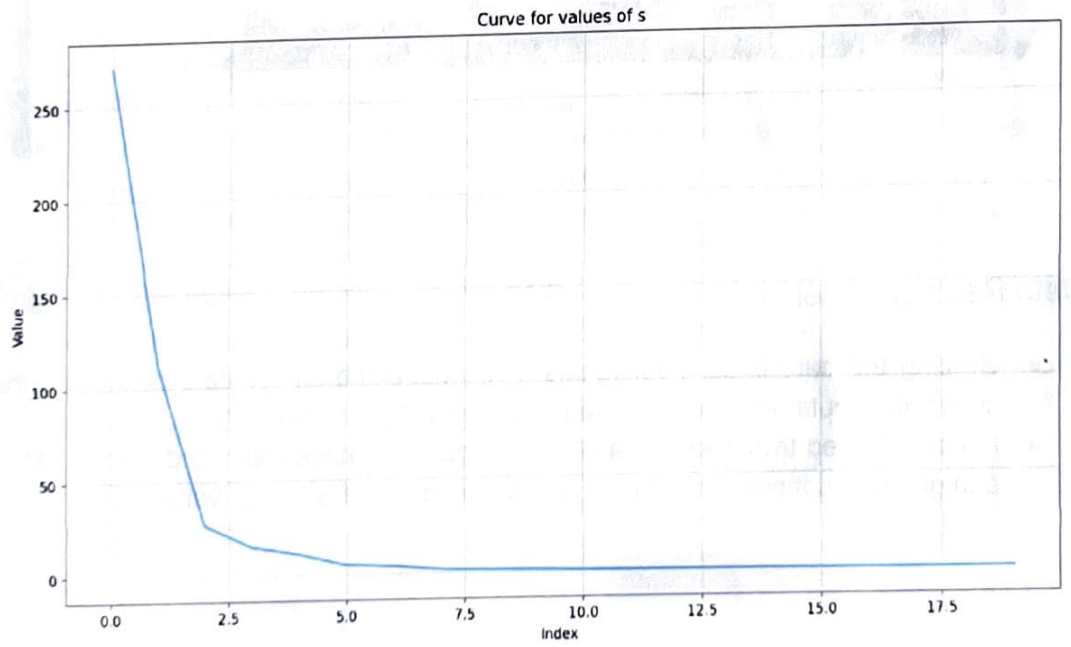
- To handle missing values, I replaced NaN values with 0 both in the matrix and the test data from 'Parameter 00000'.
- I calculated the average of all values in the matrix and subtracted this average from each value to ensure a more straightforward and accurate analysis.

Singular Value Decomposition (SVD):

- To gain insights into the data and reduce dimensionality, I utilized the 'np.linalg.svd()' function to perform the Singular Value Decomposition of the matrix. Notably, the test data set 'Parameter 00000' was excluded from this calculation.

Choosing the Number of Singular Values Required:

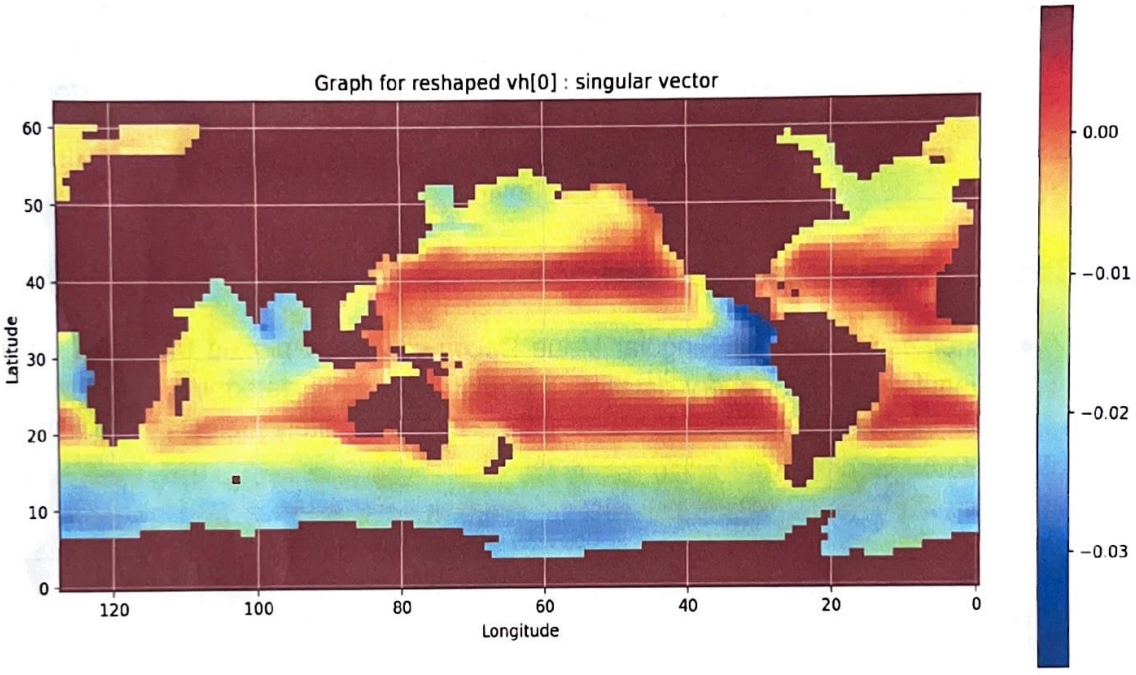
- After performing the Singular Value Decomposition, I plotted the curve for the singular value for every dataset. This plot helped me determine the number of singular values required to represent the data accurately. Below is a plot for the same.



- I also calculated the percentage of data that could be represented by dividing the sum of the squares of the chosen values by the sum of the squares of all singular values.

Plotting the chosen Singular Values:

- After selecting the required Singular Values, I plot them as a world map. Below is an example of the plot of a Singular Value.

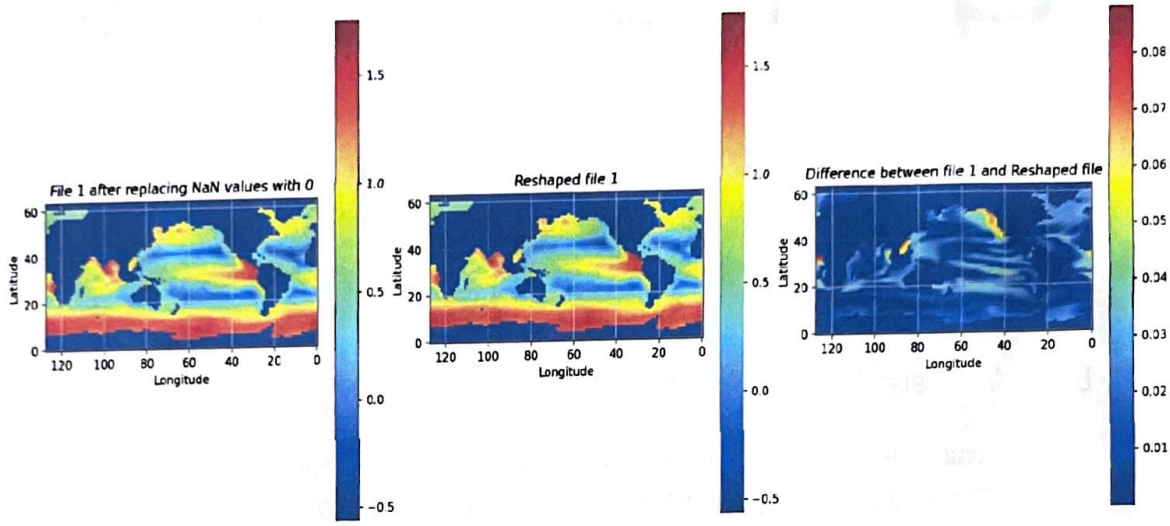


Data Reconstruction:

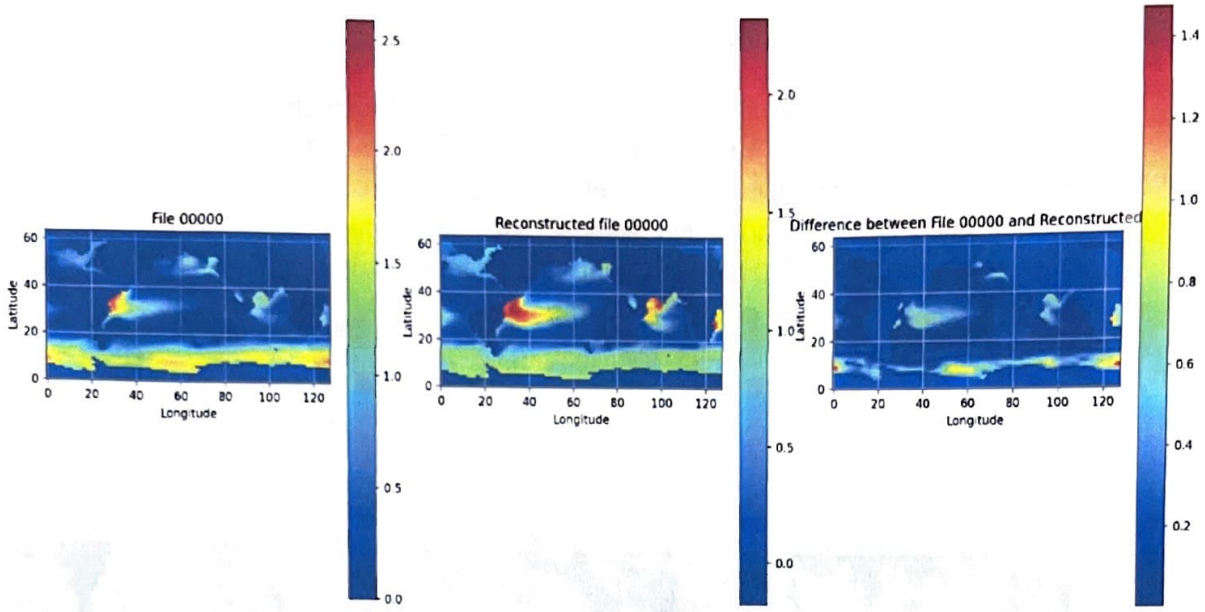
- Utilizing the calculated eigenvalues, I reconstructed all the data sets using the selected singular values (truncated) from the SVD calculation.
- For this, I used truncated left and right singular matrices computed from the SVD and got their dot product with the selected number of Singular Values.

Reshaping and Comparison:

- For visualization and comparison, I reshaped the reconstructed and training data into two-dimensional fields depicting the map of the world.
- I plotted both two-dimensional fields, replacing zero values with NaN again for improved visualization. I also plotted the difference between the original and reconstructed training data to gauge the model's accuracy.
- Below is an example plot for the same.



- To test the entire model, I reconstructed the test dataset using the following method.
- I found the coefficients of the test dataset by finding the dot product between the dataset and the corresponding truncated right singular vector values.
- These coefficients were then multiplied with the dataset one by one and summed up to find the resultant reconstructed test dataset.
- Like the training dataset, I reshaped the original dataset and the reconstructed test dataset into two-dimensional fields depicting the map of the world.
- I plotted both two-dimensional fields and also plotted the difference between them to gauge the model's accuracy.
- Below is an example plot for the same.



Error Analysis:

- I calculated the percentage error for each reconstructed dataset by finding the error in each data point between the original and reconstructed dataset. Then I displayed the mean value of the error as the error in the reconstructed dataset.
- I had to handle all zero values by replacing them with NaN values for the abovementioned calculations.
- I also calculated the L2 norm of all datasets using 'np.linalg.norm()' and printed the L2 norm of the difference between the original and reconstructed datasets.
- To calculate the percentage difference in the L2 norm between the reconstructed test dataset and the original test dataset, I divided the L2 norm of the difference between the original and reconstructed test datasets by the L2 norm of the original test dataset.

Inference:

- After conducting multiple simulations with randomized numbers and orders of datasets, it can be concluded that a minimum of 4 singular values must be selected to reconstruct the train databases effectively. For more accurate reconstructions, up to 6 singular values can sometimes be chosen.

- It has been determined that a minimum of 15 to 16 training datasets are required to successfully extract the singular values needed for reconstructing the test dataset. This holds for any training dataset and does not depend on a specific order; successful reconstruction is possible as long as there are at least 15 to 16 datasets. Furthermore, this method can be applied to reconstruct any test dataset with high fidelity.
- Upon reconstructing each training dataset, the L2 norm of the difference between the original and reconstructed training dataset falls within the range of [0,3).
- Similarly, when reconstructing any test dataset, the percentage difference in the L2 norm between the original and the reconstructed test dataset is within the range of [0,1.5).

Conclusion:

The internship at Kiel University provided valuable hands-on experience in statistical data analysis using Python. I completed the assigned tasks, including efficient PETSc file reading, data preparation, reshaping, reduction and transformation, matrix construction, data cleaning and regularization, Singular Value Decomposition (SVD), choosing the number of singular values required, data reconstruction and reshaping for visualization using matplotlib graphs and comparison and error analysis.

I extend my heartfelt gratitude to Professor Dr Slawig for this opportunity and to the Mathematics and Computer Science department team for their guidance and support throughout the internship. The practical skills and knowledge gained during this internship have significantly contributed to my professional growth.

Thank you.

Parshva Mody

Prof. Dr. T. Slawig
Christian-Albrechts-Universität
Institut für Informatik
D-24098 Kiel