Clustering Analysis and Predictive Modeling of Global Fisheries Data

Konstantinos Palaiologos

School of Electrical and Computer Engineering
Technical University of Crete

August 1, 2024

Table of Contents

1.1. Motivation

- Personal interest in sealife and general love for the sea.
- Concern about the impact of overfishing on marine ecosystems.
- Uncovering similar patterns in historical fish production trends across different countries.
- Predicting future sustainability of global fisheries.

1.2. Methodology

Development environment:

- Python (*NumPy, Pandas, Scikit-learn, TSlearn, TSA, Matplotlib*)

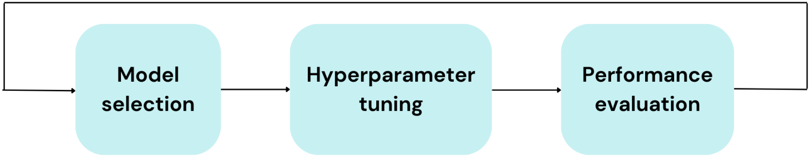
Data collection:

- Gathered data and verified sources with the provider.

Preprocessing:

- Performed various joins and reshaped data to meet model-specific requirements.
- Handled data gaps using non-trivial techniques.

Model development:



1.3. Fish and Overfishing dataset (FAO)

4 subsets with similar structure:

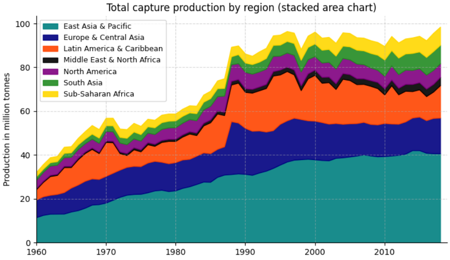
Features

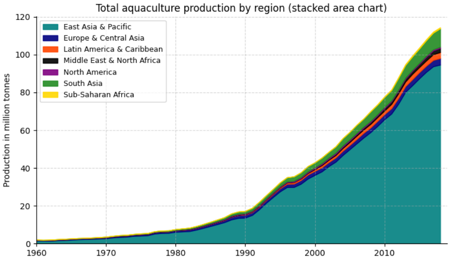
- **Capture production** (*tons*)
- **Aquaculture production** (*tons*)
- **Consumption per capita** (*kg*)
- **Sustainable levels** (%)

Rows

- **Entities**: countries & aggregate groups
- **Years**: about 50 years of annual data for each entity

Entity & year span discrepancies among the datasets $\implies$ missing values upon concatenation.





2. Time series clustering analysis

2. Time series clustering analysis

Goals

- Identify groups of countries with similar t.s. characteristics for **capture production**, **aquaculture production** and **consumption per capita**.
- Utilize these clusters to develop separate t.s. models for each group.

Procedure

- Prepared a consistent dataset with matching countries and year ranges.
- Performed model-specific feature space transformations.
- Clustering was performed in a **hierarchical** fashion.
- Applied various clustering models to find the optimal configuration.

Evaluation metrics

- **Silhouette score:** evaluates clustering quality by measuring how similar each point is to its own cluster compared to other clusters.
- **Noise points:** opted to minimize data points labeled as noise.

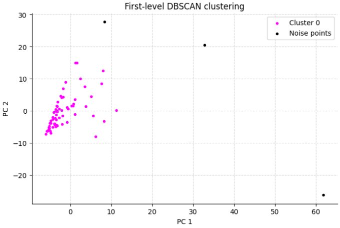
K-Means: partitions points into fixed clusters.
DBSCAN: groups points based on density and distance.
OPTICS: identifies clusters of varying densities.

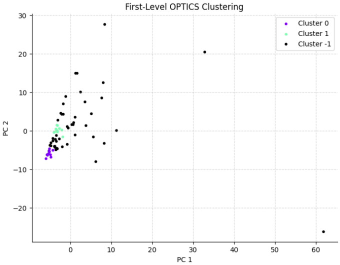
Feature transformation

Flattened time series data into single rows $\Longrightarrow$ 171 features per country (3×57 yrs).

Scaling & dimensionality reduction

Standard scaler: $z = (x - \overline{x})/\sigma$
PCA: reduced features to 2 principal components $\Longrightarrow$ easier visualization





2.1. Traditional clustering models

Challenge: lack of clearly defined clusters in the original dataset.

RBF kernel decomposition was applied to increase data separability:
$\implies$ Less points were labeled as noise but final clusters had sub-optimal silhouette scores.

Best configuration found

1st level: OPTICS
2nd level: DBSCAN

- **Points in final clusters**: 21/68

	silhouette score
cluster A	0.68
cluster B	0.76

Only 31% of initial points were kept in the final clusters, while the rest were labeled as noise $\implies$ need for an alternative model.

2.2. Clustering with TS-learn

TS-learn K-means: Groups sequential data into fixed clusters.
Uses the entire time series as features, maintaining temporal ordering.

Data reshaping

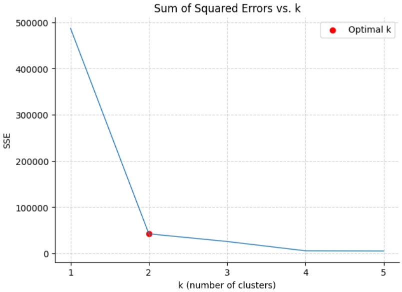
The 3 features were incorporated as a 3rd dimension $\Longrightarrow$ each data point represented by a tensor.

Scaling & dimensionality reduction

Standard scaler: $z = (x - \overline{x})/\sigma$
PCA: not applicable

Similarity metric

- Soft dynamic time warping (sdtw)



Number of clusters (k) was tuned each time using the "elbow rule".

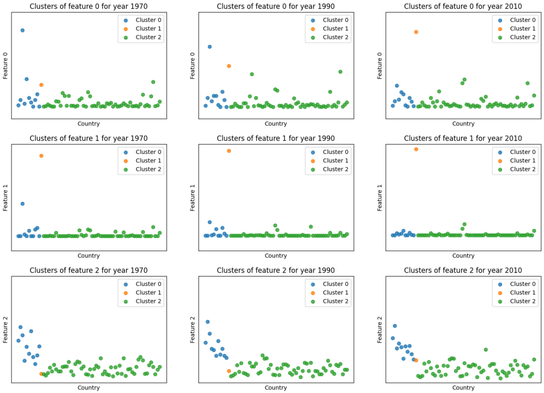
Final Clusters

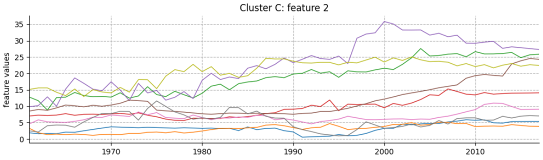
	Silhouette Score
Cluster A	0.78
Cluster B	0.74
Cluster C	0.93

- **Points in final clusters:** 44/68 (65%)

- **10%** better average silhouette score
- **34%** more points kept

- Diversity within clusters persists. Why?





3. Multivariate Time Series Modeling

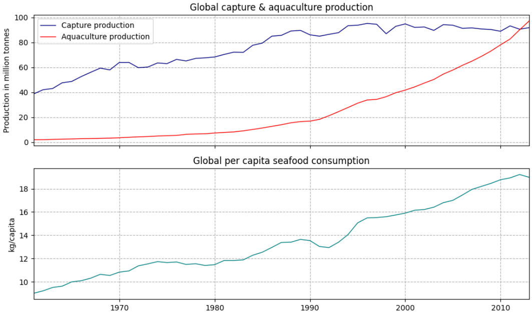
3. Multivariate Time Series Modeling

Goals

- Produce forecasts for global **capture production**, **aquaculture production**, and **consumption**.
- Determine if separate models can be trained for the clusters of Phase 2.

Procedure

- Selected appropriate multivariate t.s. models.
- Conducted statistical tests to ensure validity.
- Evaluated model performance on the testing set.



Why multivariate modeling?
Captures potential dependencies between variables, offering a more comprehensive representation of the system.

Could be misleading in case of no significant dependencies.

3.1. Vector Auto-Regressive Model (VAR)

Achieving stationarity

ADF tests indicated that:

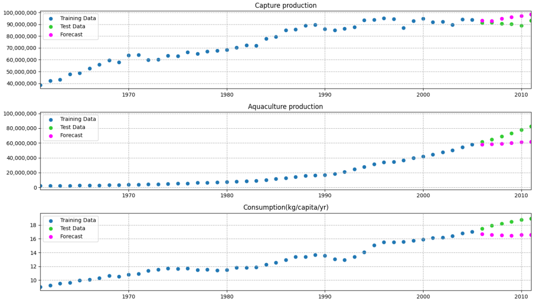
- *feature 0* → 1st order differencing
- *features 1, 2* → 2nd order differencing

Hyperparameter tuning

Rolling window CV gridsearch:

- Train-test split → 90%-10%
- Lag order → 3

NMSE	AIC
0.9946	53.7444



Performance on testing set

R^2	MAPE
-9.2794	0.1003

Poor predictive performance, even after tuning. Need for an alternative model.

3.2. Vector Error Correction Model (VECM)

Can handle non-stationarity.

Cointegration rank

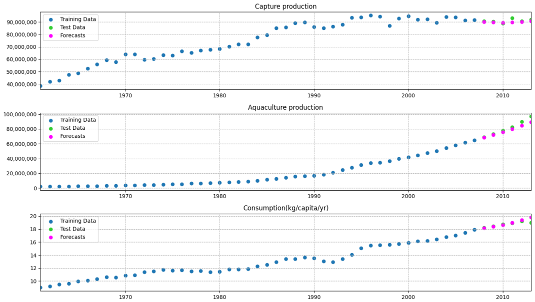
Johansen test indicated strong evidence (99% confidence) for at least 1 cointegrating relationship.

Lag order

Due to the presence of the error correction term in VECM $\implies$

$$p_{VECM} = p_{VAR} - 1$$

$$\implies p_{VECM} = 2$$



Performance on testing set

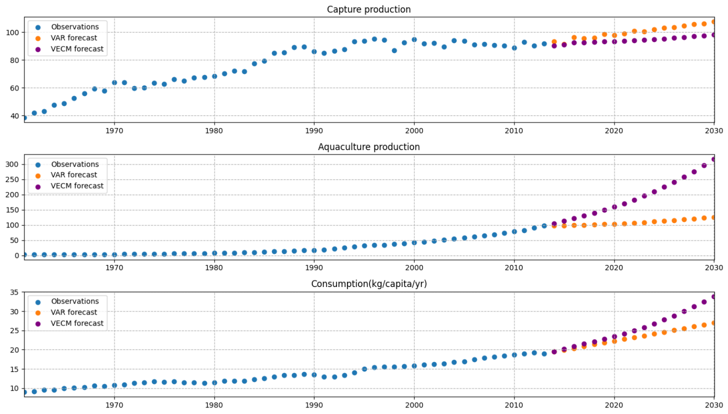
R^2	MAPE
0.1288	0.0202

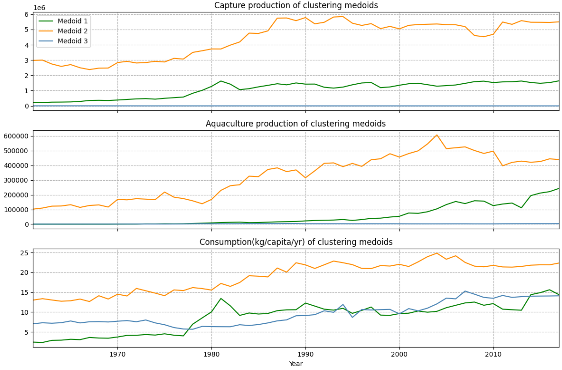
Prediction **accuracy improved significantly**.

3.3. Forecasts and comparison

- VAR effectiveness may be hindered by the inconsistent differencing orders.
- VECM superior performance is likely due to its ability to model cointegration.

17-year horizon forecasts for the 3 features





	medoid A	medoid B	medoid C
R^2	-1.3586	-5.6407	-158.8639
MAPE	0.1802	0.0270	0.4043

Cluster medoids

The most representative time series of each cluster.

Model choice

VECM was used, since it worked best for global features.

Country-specific features possibly **depend on other countries** $\Longrightarrow$ underlying relationships not captured. (*not an issue in global*)

4. Predicting global sustainable levels

4. Predicting global sustainable levels

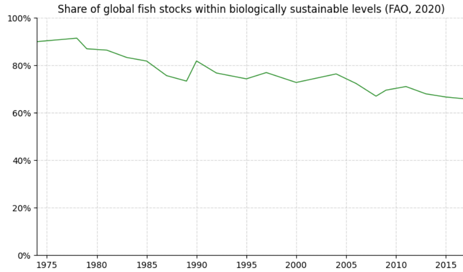
- Current trends of fish stocks within sustainable levels are alarming.
- How will sustainability levels evolve in the future?

Goals

- Predict **past sustainable levels** using the available feature data.
- Predict **future sustainable levels** using the forecasted feature values.

Procedure

- Prepared the train and test datasets (75-25 split).
- Conducted hyperparameter tuning.
- Evaluated models on the testing set.
- Used best model for predictions.



Target's sampling frequency $\neq$ features' sampling frequency $\implies$ **missing values** when creating the training set.

4.1. Preparing the dataset for the ML models

Models tested

Linear Regression

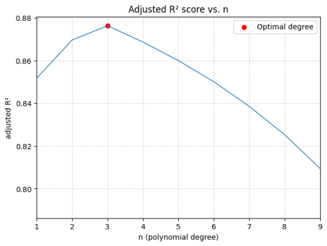
Bayesian Ridge Regression

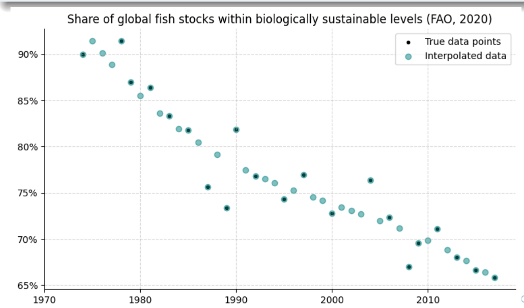
SVR

Random Forest

Interpolation for data gaps

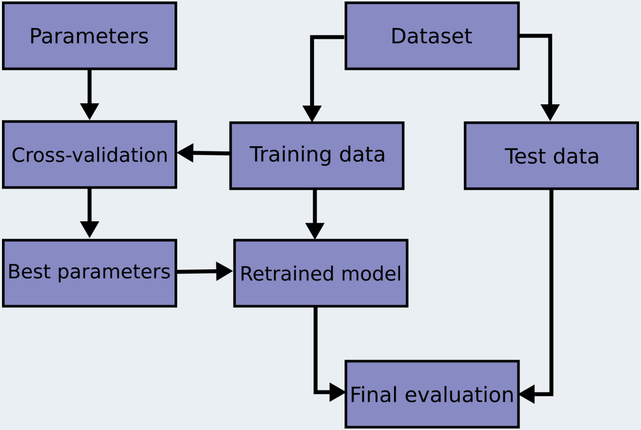
Polynomial order for linear regression was selected by plotting it against adjusted-R^2 and choosing the optimal point.





4.2. Automating the ML pipeline

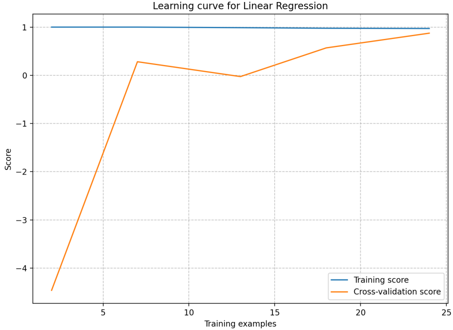
Modeling workflow

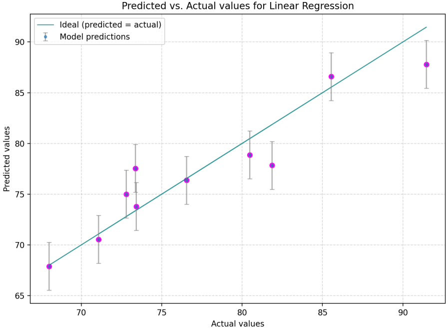


- A distinct pipeline was used for each model.
- Hyperparameter tuning was automated with *GridSearchCV*.

4.3. Model evaluation

- **Training score:** How well does the model fit the training set?
- **CV score:** How well does the model generalize to unseen data?
- **Testing score:** How accurately does the model predict on the testing set?

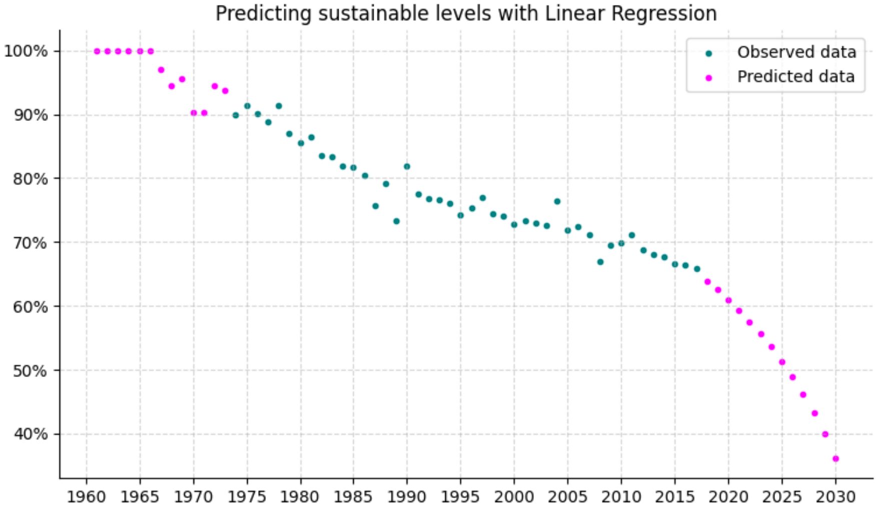




	Linear Regression	Bayesian Regression	SVR	Rand. Forest
R^2	0.881	0.863	0.870	0.855
RMSE	2.371	2.552	2.482	2.623

4.4. Final predictions

- **Prediction range:** 1961-1973 (past values) ∪ 2017-2030 (future values)



5. Conclusions & future work

5. Conclusions & future work

- Respecting the temporal range of data yielded the best clustering results.
- Low variance in sustainable levels allowed for effective Linear Regression fitting, even with the presence of limited data.
- Based on the results, future sustainability levels raise serious concerns, suggesting the need for deeper analysis and proactive measures.

Future work

- Test different clustering approaches (e.g. feature-specific clustering).
- Explore univariate models (e.g. ARIMA) and alternative forecasting methods (e.g. one-step-ahead).
- Forecast using regression and lagged values.
- Validate forecasting results using recent observations.

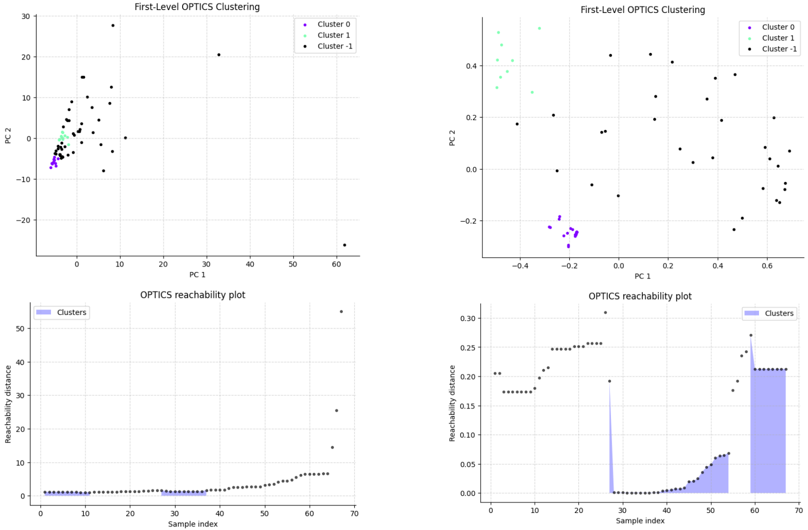
6. References

- Fish and Overfishing Dataset (Hilborn R., Melnychuk M., Mossler M., and Hively D., *RAM Stock Assessment Database*, original data from FAO)
- H. Ritchie, & M. Roser (2024, March). Fish and Overfishing. *Our World in Data*
- R. Tavenard, J. Faouzi *et al* (2020). Tslearn, A Machine Learning Toolkit for Time Series Data. Journal of Machine Learning Research
- F. Pedregosa *et al*. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.
- S. Johansen (1991). Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. *Econometrica*
- A. Singh (2024). Multivariate Time Series Analysis

7. Appendix

7.1. Increasing data separability

OPTICS reachability plots: before (left) and after (right) kernel PCA.



7.2. Regression models parameter grids

```python
param_grids = {
    "Linear Regression": {
        "poly__degree": [1, 2, 3, 4],
        "linear__fit_intercept": [True, False],
    },
    "Bayesian Ridge Regression": {
        "poly__degree": [1, 2, 3, 4, 5],
    },
    "SVR": {
        "svr__C": [0.1, 1.0, 10.0],
        "svr__epsilon": [0.01, 0.1, 1.0],
        "svr__kernel": ['linear', 'rbf', 'poly']
    },
    "Random Forest": {
        "rf__n_estimators": [50, 100, 200, 500],
        "rf__max_depth": [None, 10, 20, 30]
    }
}
```

Figure: Parameter grids for the tested regression models

7.3. Regression models selected parameters

```
----Linear Regression-----------------------------
 Best hyperparameters --> {'linear__fit_intercept': True, 'poly__degree': 2}
 Best hyperparameters mean CV score --> 0.92379
---Bayesian Ridge Regression----------------------
Best hyperparameters --> {'poly__degree': 4}
Best hyperparameters mean CV score --> 0.92802
----SVR------------------------------
 Best hyperparameters --> {'svr__C': 10.0, 'svr__epsilon': 0.01, 'svr__kernel': 'rbf'}
 Best hyperparameters mean CV score --> 0.93411
---Random Forest-----------------------------
Best hyperparameters --> {'rf__max_depth': None, 'rf__n_estimators': 50}
Best hyperparameters mean CV score --> 0.92122
```

Figure: GridsearchCV results for each regression model.