Methods

Data

Wild Juvenile Salmon Data

Forecasting efforts were focused on estimating sea lice abundance on wild juvenile salmon in the Broughton Archipelago, BC. The data consist of a monitoring effort ongoing from 2001 to 2019, specific details on sampling and methods are available in a supporting paper (Peacock et al. 2016). Sampling consisted of visually searching coastal waters, ~2-5m from the shore, and capturing schools of juvenile salmon for examination. Once captured, the salmon were measured along with notes on their health characteristics and species. The salmon were non-lethally visually examined using 16x magnification for sea lice, with life stage and species of lice being noted. As well as fish and lice information, site information was collected for each sampling event including temperature and latitude/longitude.

The forecasted variable (response variable), motile sea lice, is derived from this dataset; motile sea lice for this analysis are defined as pre-adult and adult stage male and female sea lice. Predictor variables (covariates) of interest from this dataset include date and site information, temperature and location.

Farmed Salmon Data

Parasite occurrence in farms is thought to have a significant effect on parasite occurrence on wild fish (Marty et al. 2010), and so data describing salmon farm sea lice counts were analysed. The dataset used consists of Department of Fisheries and Oceans (DFO) mandated counts of monthly averages of motile sea louse occurrence. These data are collected by the farms themselves and averages are submitted to the DFO, however resolution to the individual sampling event level is not publicly available. Variables of interest from this dataset include date, site location, and average motile sea lice count.

Weather Forecast Data

In order to supplement the months not sampled by the wild salmon sampling efforts, daily weather reports from a nearby weather station, Port Hardy, were analysed. Historical air temperature reports were used to find relationships between off-season trends and the collected wild salmon data that were not sampled by the wild sampling efforts.

Frequency

All data were resampled to a weekly frequency when training models and making predictions to align with the other data and the requirements of the models, using linear interpolation for missing data (Moritz and Bartz-Beielstein 2017).

Predictors and Response Variables

The response variable for all forecasting was the count of all species of motile sea lice on wild salmon. Predictor variables were: surface temperature and salinity of wild salmon sampling sites; juvenile sea lice on wild salmon; industry motile sea lice averages from Sargeaunt Pass, Doctor Islets, Humphrey Rock, Burdwood, Glacier Falls, Sir Edmund Bay, and Wicklow Point salmon farms; and air temperature from the Port Hardy weather station.

Modelling

Long Short-Term Memory (LSTM)

The first model trained for forecasting was a Long Short-Term Memory (LSTM) model (Fig. 1), a type of recurrent neural network (Sainath et al. 2015). This is a type of neural network with feedback as well as feedforward connections, making it suitable for time series predictions, as seen in Fig. 1. Neural networks allow for the description of highly non-linear relationships, but many neural network frameworks do not account for sequential data such as time series. An LSTM model was chosen as inputs are taken to be inherently sequential and therefore can be modelled as a time series adequately.

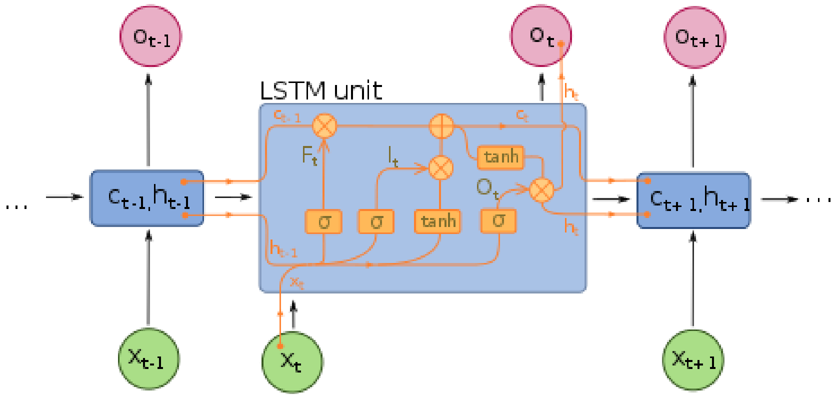

Figure 1. Long Short-Term Memory model unit. For each timestep *t*, given the inputs x_t, the overall model weights h_t are influenced by the short-term trend c_{t-1} to produce the output o_t and the trend c_t and model weights h_t are updated. Author: fdeloche, Wikimedia Commons. License: https://creativecommons.org/licenses/by-sa/4.0/legalcode

Before fitting the model, data were normalised and scaled using maximum absolute value scaling. To assess the modelling framework's ability to forecast each year, an LSTM model was trained on all years of data excluding one year. This excluded year would be used to test the viability of the model. This model was then tasked to predict motile sea lice on wild salmon from

March to July under three conditions of available predictor data from that year: all predictor data available, predictor data available up to and including April, and predictor data available up to and including June. Models were trained and assessed using this method on all years of data from 2003-2017.

Models were built using Keras 2.3.1 and Tensorflow 1.13.2. Code for specific configurations can be found at https://github.com/TristanGarry/sea-lice-forecasting/tree/master/lstm.

AutoML

Microsoft Azure Auto Machine Learning (AutoML) (Barnes 2015) was used in order to assess the effectiveness of other machine learning methods. AutoML is a Microsoft service that allows several specified machine learning models to be assessed on a dataset. This was used to fit Elastic Net, Decision Tree, Random Forest, and Gradient Boosting models as well as voting and stack ensemble models of all the fitted models. Using AutoML, additional features were constructed, and each model was tried with both maximum absolute value and regular standardisation. Three classes of models were applied using AutoML: Linear regression, tree-based methods, and ensembling methods.

Linear Regression

Linear regression methods of the general form $Y = \sum_{i=0}^{p} ß_i x_i$ were one class of models fit using this framework. The algorithm used to fit linear regression models is known as Elastic Net, which is a regression method that linearly combines the penalty terms of lasso and ridge regression methods (Zou and Hastie 2005). The parameters, ß$_i$, are constrained to only be large if they contribute sufficiently to the fit of the model, as seen in Fig. 2.

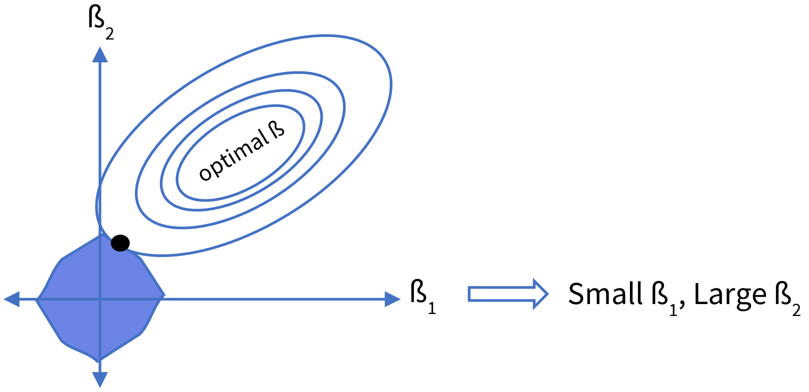

Figure 2. Elastic Net constraint visualisation in two dimensions. Ordinary least squares methods would choose a parameter in the centre of the elipses around the optimal ß. Elastic Net is constrained to choosing parameters on the surface of the shape around the origin in order to ensure parameters provide sufficiently to the fit.

Tree-Based Methods

Tree-based methods consist of splitting the data based on predictor threshold values in order to best explain it, as seen in Fig. 3. At each split, the sample is split into two (or more) sub-samples based on the most significant predictor. The number of splits to explain a dataset can be chosen as an arbitrary integer, using a stopping criterion (e.g. R_2 threshold value), or dynamically by using another algorithm. Three different tree-based methods were evaluated using AutoML: decision trees, random forest, and gradient boosting.

Decision trees consist of splitting the data based on threshold values of the predictors sequentially until the data is sufficiently explained or a stopping criterion is reached (James et al. 2013).

Random forest consists of fitting several decision trees to the same data and ensembling these models' predictions for the random forest's predictions. Individual decision trees are ensured to not be correlated by randomly choosing a subset of the original predictors that can be fit on, as well as bootstrapping each tree's data (James et al. 2013).

Gradient boosting is similar to random forest except that sequential decision trees correct for the previous one's errors (James et al. 2013).

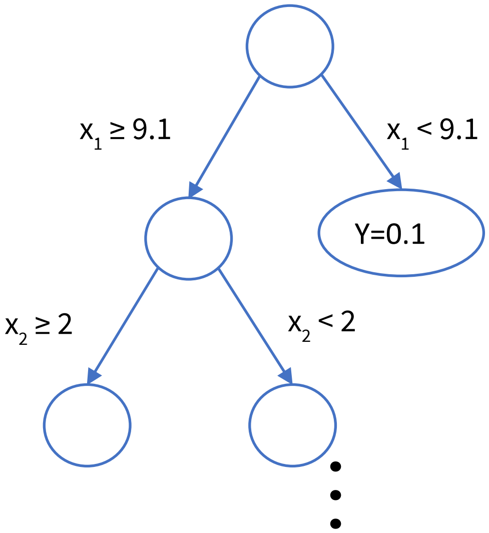

Figure 3. Visualisation of a tree-based learning model. In this case, the first split on predictor X_1 splits the sample into two subsamples, one where X_1 is greater than or equal to 9.1and another where X_1 is less than 9.1. This would continue for all predictors until some stopping criteria is reached.

Ensembling Methods

Ensembling methods take output from other models and create a meta-model with them in order to create models that can perform better than any single algorithm, as seen in Fig. 4. These outputs can be combined together to produce a new output or can be processed by another machine learning algorithm to create a model. The ensembling methods used were: voting ensemble and stack ensemble.

Voting ensemble (Wang et al. 2011) consists of allowing all models to make a prediction. Using these predictions, the voting ensemble output is the most agreed-upon prediction based on soft voting with weighted averages.

Stack ensemble (Wang et al. 2011) consists of a first layer using the trained models to recreate a training dataset based on the trained models' predictions, and a second layer training an Elastic Net model on these outputs.

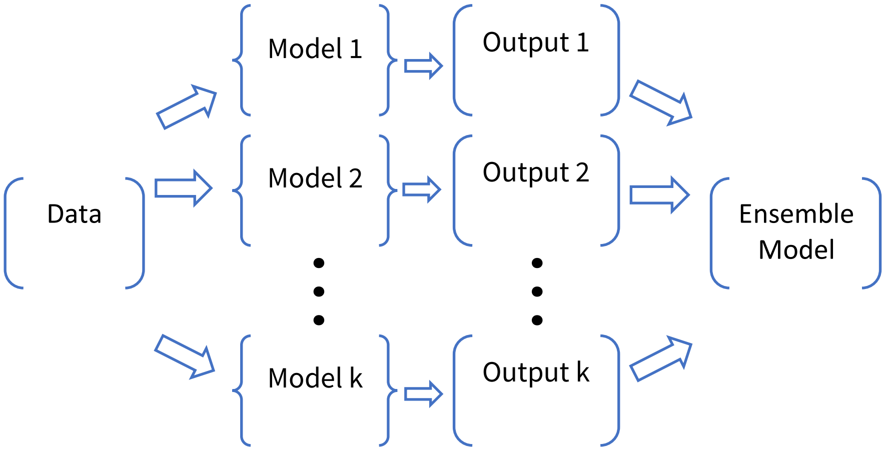

Figure 4. Visualisation of how models come together to form an ensemble model. The ensemble model takes outputs from each of the previously fit models and can either use these outputs directly to form its own output or use the models' outputs to parametrise a new model.

Similar to the LSTM evaluation, AutoML models were trained on all years of data excluding one year. These models were then tasked to predict motile sea lice on wild salmon from March to July under three conditions of available predictor data from that year: all predictor data available, predictor data available up to and including April, and predictor data available up to and including June. The best performing model, the stack ensemble model, was the model selected by this method. Models were trained and assessed using this method on all years of data from 2003-2017.

Models were built and trained using AzureML 1.0.83. Code for specific configuration can be found at https://github.com/TristanGarry/sea-lice-forecasting/tree/master/automl.

Naïve Seasonal Bayesian

A naïve seasonal Bayesian model was used as a null model. Given a date of the year, this model outputs the average of all previous years' outputs on that date.

This model was built and trained using Numpy 1.17.3 and Pandas 0.25.2. Code for specific configuration can be found at https://github.com/TristanGarry/sea-lice-forecasting/tree/master/bayes.

Results Interpretation

The predictions generated from the LSTM, AutoML, and Naïve Bayesian methods were assessed for accuracy based on root mean squared error (RMSE) and mean absolute error (MAE).

Programming environment

Analyses and data were handled using Python 3.6, Pandas 0.25.2, and Numpy 1.17.3. All code can be found at https://github.com/TristanGarry/sea-lice-forecasting, with specific package details in /setup/environment.yml.