April 17, 2023 Theory Working Group Call

Attendees: Caleb Robbins, Christy Rollinson, Abby Lewis, Jody Peters, Cazimir Kowalski, Freya Olsson, Glenda Wardle, Alyssa Willson, Shubhi Sharma, Cole Brookson
Regrets: Jono Tonkin, Mike Dietze

Agenda:

1. GitHub updates
    a. **GitHub - eco4cast/predictability: Simulations exploring measures of predictability for time series data**
        i. Simulations exploring measures of predictability for time series data
    b. **GitHub - eco4cast/Forecast_submissions: A repository for forecast development and analysis as part of the Ecological Forecasting Initiative (EFI) Theory Working Group**
        i. A repository for forecast development and analysis as part of the Ecological Forecasting Initiative (EFI) Theory Working Group
2. Caleb offered to provide a short tutorial on how to use the tidymodels framework. Jody will record the overview to share with those who are not available.
    a. https://www.tidymodels.org/find/parsnip/
    b. Some ML - goal is out of sample prediction - build models that are predictive, but not overfitting. ML splits data into training and testing sets.
    c. Tidymodels - a set of models to do ML workflow
        i. Workflows that go from data to predictions
        ii. Use tidyverse packages
        iii. Focuses on protecting against data leakage
        iv. Allows you to test parameter tuning
        v. Have tools to compare/evaluate models
    d. Workflow - see details in the slides for the 6 steps
        i. Step 1: decide how to split data for training and testing sets. Also focuses on keeping data separate so the training dataset doesn't influence the testing set
        ii. Step 2: specify the recipe ("feature engineering")
            1. There are a lot of tidyselect functions used in the recipe functions
            2. This is the trickiest and most annoying part of the tidymodels
        iii. Step 3: set engine
            1. Model framework under workflow - where you define your model
            2. 100 models at this link: https://www.tidymodels.org/find/parsnip/ that can get plugged into the workflow
            3. You can build new engines (models) that aren't available yet
        iv. Step 4: tuning parameters
            1. Select the best tuning parameters for evaluating training set without using the testing set - bootstrapping and v-fold cross validation

v. Step 5: finalize workflow
    1. Select the best performing parameter set (“select_best” function)

vi. Step 6: evaluate test set

vii. See slide for more information - 3 resources for tutorials, free ebook, Virginia Tech GitHub repo for a class has almost the same workflow

e. If people want to adjust Caleb’s existing workflow the things that need to be changed from the existing workflow are
    i. Change the model engine (change rand forest to a different model. Abby is using bagMLP)
    ii. Be conscientious of the hyperparameters you want to tune to

f. If you want to train, the GitActions isn’t a good place to do that. Need to do it locally, save it locally, and use those model fits that pushes to a separate place so you aren’t newly training every day. It is too computationally tough to train the models every day in GitHub

g. How long did it take for Caleb’s model to train?
    i. For all the parameters he used for the NEON data for all the sites (ignoring ticks and beetles) it took a day on 14 cores to train the model
    ii. This is based on the number of hyperparameters and the number of trees. How many tuning parameters and how often you are resampling
    iii. V-fold cross validation was 5 repeats x 20 parameter sets so 100 models to train it up for 1 site and 1 variable

h. For neural networks will need to make changes. Don’t think it is in the parameter tuning part. Think it will be in the engine part. Look for tutorials for neural nets. Think there will be some good ones.

i. When you have short/messy data - was there a sense for the minimum length/quality of data were needed. What about imputation of the data - what happens if a sensor goes down?
    i. From Caleb’s experience there were sites where there weren’t many observations so the v-fold cross validation didn’t work, so had to cut them down, so the predictions are as good at those sites.
    ii. ML are nice because you can set them up, but there is the issue that you may not know what is going on because you aren’t as focused on the data quality
    iii. If you have great data, any model will do. If you have poor data, then no model will help you.
    iv. It will be interesting to look at afterward to see if sites that had worst data quality had worse model performance.
    v. Moving forward it will be good to keep information about the # of observations the model gets trained on
    vi. May want to build in other things to keep information about the training information about the models.
    vii. Have others seen other examples of others who are archiving parameters?

            1. Not yet. It would be cool to see. But right now, focused on getting people to submit to the Challenge
    viii. Abby put together this Google sheet with all the models in tidymodels
            1. Hypothetically think we could use all of these
            2. If anyone is interested in getting models up and running this is an easy way to get a model started
            3. Cazimir is going to create at least one of these models if not more

j. What would it take to get this set up for ticks and beetles?
    i. Don't know why it wouldn't work
    ii. Caleb didn't deploy it because they were commented out in the code
    iii. Abby has updated the code for ticks and beetles
    iv. Think it wouldn't be hard to update the tidymodels and then update with the new info Abby added for when to run the ticks and beetles
    v. Currently the tidymodels is running for site. But think we could add a site predictor variable and then run across all sites
    vi. Think we could have all tidymodels run on site by site basis and then across all sites - but it will take work to do this.
    vii. Changing the predictor set in that way will break the current framework because of the way the loops run.
    viii. Abby's all sites model could give a starting place for this. But it will still take integrating it with the other changes Caleb has made

k. What are the main questions we want to answer with the models?
    i. This will be important to help us know what parameter information to keep from the models
    ii. Is there a way to get the CRPS or other model evaluation from the Challenge
        1. Yes, we can grab all of the scores directly after they are submitted.
        2. Or you can score them yourself
    iii. Abby and Shubhi had played around with code from Mike
        1. The plan was to look at later decline and predictability across scales and horizons. To compare across the forecast variables rather than thinking about model specifications
            a. Think we are making good progress on this
        2. Figure out how the Challenge and the Synthesis activities converge in the questions both groups are answering
    iv. Predictability increases/decreases with ecological aggregation - seems this may only be applicable to the beetles
        1. Could this be applied to thinking ad hoc about phenology diversity - crops = monoculture vs rainforest that are more diverse

3. Comparative analysis of predictability using an uncertainty framework to decompose how predictable something is and why or why not (Shubhi, Cole, Noel, etc) - update from Cole
    a. Started by thinking about information theory as a way to quantify intrinsic and real predictability. Using different types of entropy measures on the time series measures to see how predicable they are and how well we are doing at maximizing prediction potential
    b. Have been working on 2 different types of entropy measure - relative (real prediction measurement) and permutation (prediction ceiling measurement)
    c. Cole has been developing over the past month something that is automated to get the prediction ceiling for the Challenge variables
    d. Would be helpful to get input from people with more familiarity with the data sources. What sort of time horizons are useful for considering predictions.
    e. Goal is to get measure of permutation entropy for each variable in each of the Challenges and construct a comparison between the different Challenges. Look at it across sites. Make a comparison on the Challenge level or on the variable level to see what is more predictable.
    f. All of the code for this is in the GitHub repo above. Most of the code is running. Now working on - looking at how to select parameters and will need to ping people on what are the most effective way to chop the data sets to calculate the measure on. Taking a 6 years dataset and getting a mean doesn't tell you the same thing as if you take a smaller time frame
        i. Example: take greenup from the winter and take permutation entropy it is high because not much is happening. But if you look at it during green up then it will be much different.
        ii. Then how to compare it to the intrinsic entropy value
        iii. Would be interesting to chop it into different pieces. For some pieces all the 6 years is informative because it is very stable. But for the ones taht are more seasonal then you
        iv. Can you calculate a rolling permutation entropy? Have a rolling measure of relative entropy
        v. Would it be possible to do some spectral analysis to look at predictability?
            1. Christy thinks Mike has a paper about this from PEcAn - https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2011JG001661
            2. Could calculate permutation entropy after detrending the data
    g. Goal for permutation entropy question is how does it link to a "black swan" event in your data. If the process generating your data also generates your "black swan" then it is predictable, but if you don't have the process then you can't predict the "black swan". Trying to think about how to get at this.
    h. From Caleb: Just thinking of like a rolling window function where PE is calculated on a small stretch of a time series, the stretch is shifted over, PE is calculated again (parameters change?), etc. So get a time series of PEs

i. Cole has the code set up to pull in the Challenge forecasts to make these measurements of entropy