Corrigendum for ‘The Added Value of Machine Learning in Forecasting Wind Turbine Icing’

Lukas Kugler, 31.08.2020

General problem: Certain results are not fully out-of-sample estimates, while they were intended as such. Hence, results appeared better than they truly are.

Affected results: Significant reduction in PEV compared to the original version is seen for 'CART-Bagging' in the case of ‘meteorological icing’. ANN and SVC are less affected. Other models are not affected.

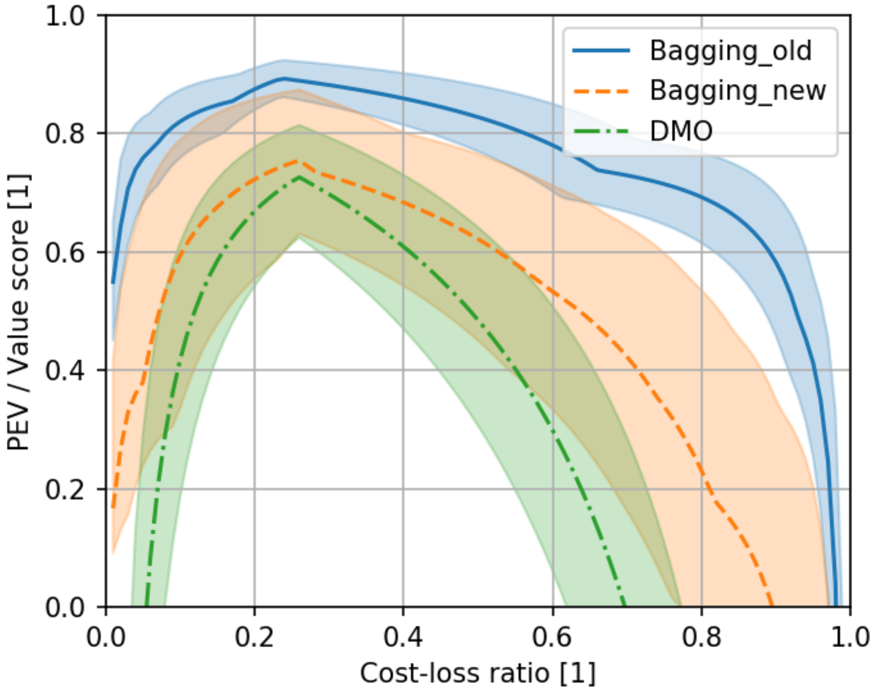
Description: The subroutine ‘undersampling’ should have selected a subsample of the initial training sample, so that the training sample contains a given ratio of icing to no-icing cases. However, this function overwrote the indices of the (until then) correctly-split training and validation data set. The resulting training set contained samples of the validation set.

Before correction:

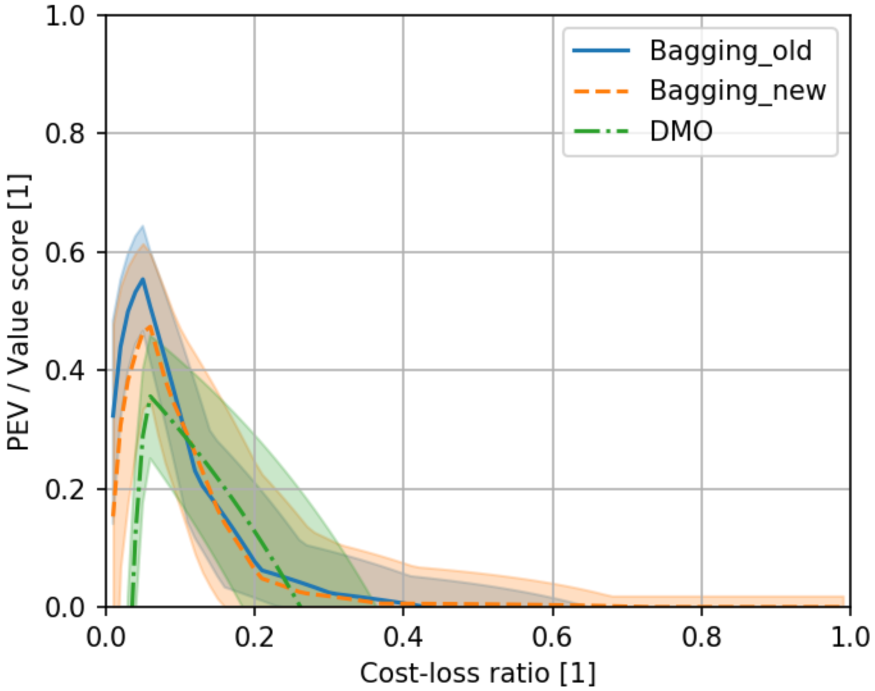
```
if undersample_pct:
            bool_event = (y_train==1)
            n_event = sum(bool_event)
            i_noevent_all = np.where(np.logical_not(bool_event))[0]

            i_event_all = np.where(bool_event)[0]

            n_noevent_choose = int(n_event/undersample_pct)  # N_minority = pct*N_not_minority
            print 'from', len(i_noevent_all), 'noevents choose', n_noevent_choose
            i_noevent = np.random.choice(i_noevent_all, n_noevent_choose, replace=False)
            print i_noevent.shape, i_event_all.shape
            i_choose = np.hstack((i_noevent, i_event_all))
            train_set = i_choose  # also changes train_set
            X_train = trainCV.X[i_choose, :]
            y_train = trainCV.y_target[i_choose]
```

After correction

```
if undersample_pct:
            # from all noevents in train set, choose a subset with given length
            bool_event = train_set & (trainCV.y_target==1)
            bool_noevent = train_set & (trainCV.y_target==0)
            i_event = np.where(bool_event)[0]
            n_event = sum(bool_event)
            i_noevent_all = np.where(bool_noevent)[0]
            n_noevent_choose = int(n_event/undersample_pct)  # N_minority = pct*N_not_minority
            n_noevents = len(i_noevent_all)
            n_noevent_choose = min(n_noevent_choose, len(i_noevent_all))  # can not choose more
noevents than there are

            print 'from', n_noevents, 'noevents choose', n_noevent_choose
            i_noevent = np.random.choice(i_noevent_all, n_noevent_choose, replace=False)

            i_train_new = np.hstack((i_noevent, i_event))
            train_set[:] = False
            train_set[i_train_new] = True  # also changes train_set
            X_train = trainCV.X[i_train_new, :]
           y_train = trainCV.y_target[i_train_new]
```



PEV diagram for 'meteorological icing'.
The solid PEV curve of the decision tree ensemble 'Bagging' was unfortunately an in-sample score. The actual PEV curve (dashed) for out-of sample predictions is considerably lower than reported in the original version of the thesis. The decision tree ensemble is not expected to be significantly better for all cost-loss ratios, but only for ratios lower than 0.1 and larger than 0.6. The dot-dashed curve shows the direct model output's score.



PEV diagram for 'visible icing'.