Probabilistic Prediction in scikit-learn*

Dirar Sweidan[1]0000-0001-5378-0862 and Ulf Johansson[2]0000-0003-0412-6199

[1] School of Informatics, University of Skövde, Sweden
dirar.sweidan@his.se
[2] Dept. of Computing, Jönköping University, Sweden
ulf.johansson@ju.se

Abstract. Adding confidence measures to predictive models should increase the trustworthiness, but only if the models are well-calibrated. Historically, some algorithms like logistic regression, but also neural networks, have been considered to produce well-calibrated probability estimates off-the-shelf. Other techniques, like decision trees and Naive Bayes, on the other hand, are infamous for being significantly overconfident in their probabilistic predictions. In this paper, a large experimental study is conducted to investigate how well-calibrated models produced by a number of algorithms in the scikit-learn library are out-of-the-box, but also if either the built-in calibration techniques Platt scaling and isotonic regression, or Venn-Abers, can be used to improve the calibration. The results show that of the seven algorithms evaluated, the only one obtaining well-calibrated models without the external calibration is logistic regression. All other algorithms, i.e., decision trees, adaboost, gradient boosting, kNN, naive Bayes and random forest benefit from using any of the calibration techniques. In particular, decision trees, Naive Bayes and the boosted models are substantially improved using external calibration. From a practitioner's perspective, the obvious recommendation becomes to incorporate calibration when using probabilistic prediction. Comparing the different calibration techniques, Platt scaling and Venn-Abers generally outperform isotonic regression, on these rather small datasets. Finally, the unique ability of Venn-Abers to output not only well-calibrated probability estimates, but also the confidence in these estimates is demonstrated.

1 Introduction

When predictive modeling is used for recommendations, decision support or automated decision making, it becomes vital that the models can be trusted. A key property for enabling trust in predictive models is that a user can have *confidence* in the predictions from the model. Specifically, the model must not only be accurate, but also capable of distinguishing between predictions where it is certain and not. Furthermore, the model should somehow be able to communicate this to a user in a comprehensible way. One obvious way of accomplishing

*This research is partly funded by the Swedish Knowledge Foundation through the industrial graduate school INSIDR.

this is to supplement every prediction with some easily interpretable measure of confidence in that prediction.

Many classifiers are able to output not only the predicted class label, but also a probability distribution over the possible classes. If these *probabilistic predictions* are *well-calibrated*, i.e., the predicted class probabilities reflect the true, underlying probabilities, they are, of course, the best possible measure of confidence. On the other hand, if the probabilistic predictions are not well-calibrated, the models actually become misleading.

In fact, while most models are capable of producing probability estimates, there is absolutely no guarantee that these are well-calibrated. Historically, several models like naive Bayes [8] and decision trees [10] were identified as often being poorly calibrated off-the-shelf. But recent studies show that even models assumed to be generally well-calibrated like modern (i.e., deep) neural networks [3] and traditional neural networks [4] often are not.

In applications where well-calibrated classifiers are necessary, an option is to perform a *post-calibration*, i.e., somehow modify the estimates from the models making them better calibrated. While there exist several such general methods for calibrating probabilistic predictions, the two most frequently used are *Platt scaling* [9] and *isotonic regression* [13]. Both these techniques have been successfully applied to different predictive models, including support-vector machines, boosted decision trees and naïve Bayes [8]. More recently, an alternative technique called *Venn-Abers* [11], has been suggested for calibrating probabilities from, for instance, decision trees [5].

Today, the Python programming language and its ecosystem have become the go to tool for many data scientists. In particular the scikit learn machine learning library is a de facto standard used in numerous applications. In scikit learn, most classifiers are able to output predicted class probabilities using the *predict_proba* method. Interestingly enough, scikit learn also includes a calibration library with implementations of both Platt scaling and Isotonic regression.

With this in mind, the overall purpose of this paper is to investigate how well-calibrated models produced by a number of different algorithms in scikit learn are, both off-the-shelf, i.e., directly using the predict_proba method, and after calibration using either the built-in methods or Venn-Abers.

In the next section, we define probabilistic prediction before describing the considered calibration techniques. In Section 3, we outline the experimental setup, which is followed by the experimental results presented in Section 4. Finally, we provide the main conclusions and suggest some future work in Section 5.

2 Background

2.1 Probabilistic prediction

In probabilistic prediction, the predictor outputs a probability distribution over the possible labels. The overall goal is to obtain *validity*, i.e., that the probability distributions from the predictor perform well against statistical tests based on

subsequent observation of the labels. In particular, the predictor must be well-calibrated:

$$p(c_j \mid p^{c_j}) = p^{c_j}, \tag{1}$$

where p^{c_j} is the probability estimate for class j. Informally, if we make a number of predictions where the highest class membership probability is, say, 0.9, we expect 10% of these predictions to be errors.

In addition, informative probabilistic predictions must also be *specific* or *sharp*, i.e., producing a variety of probability estimates over a set of test instances.

2.2 Platt scaling

Platt scaling [9] fits a general sigmoid function:

$$\hat{p}(c \mid s) = \frac{1}{1 + e^{As+B}}, \tag{2}$$

where $\hat{p}(c \mid s)$ gives the probability that an example belongs to class c, given that it has obtained the score s. A and B are found by a gradient descent search.

It is recommended that the parameters of the sigmoid function are optimized on a specific calibration set. In addition, to obtain some regularization, the following maximum a posteriori (MAP) estimates (where k_+ and k_- are the number of calibration instances labeled 1 and 0, respectively) are often used for the target probability of positive and negative examples, respectively, instead of 1 and 0:

$$t_+ := \frac{k_+ + 1}{k_+ + 2} \tag{3}$$

$$t_- := \frac{1}{k_- + 2} \tag{4}$$

Using these targets, Platt estimates will never be exactly 0 or 1, thus eliminating the risk of infinite log losses.

2.3 Isotonic regression

Isotonic regression as a calibration method was suggested by Zadrozny and Elkan [13]. The calibration function, assumed to be *isotonic*, i.e., non-decreasing, is a step-wise regression function, learned by an algorithm known as the pair-adjacent violators algorithm. Starting with a set of input probability intervals, based on the scores in the calibration set, adjacent intervals where the lower interval contains a higher (or equally high) fraction of examples belonging to the positive class, are repeatedly merged. When no such pair of intervals can be found, the algorithm terminates, and the output is a function that for each input probability interval returns the fraction of positive calibration examples in that interval.

2.4 Venn-ABERS predictors

Venn predictors [12] are probabilistic predictors that output multiple probabilities for each label, with one of them being the valid one. Somewhat simplified, the multiprobabilistic predictions can be regarded as probability intervals for each label. Consequently, the size of these intervals is an indication of the confidence in individual probability estimates.

The key idea of Inductive Venn prediction [7] is to use an underlying model to divide all calibration instances into a number of *categories*, based on a so-called *Venn taxonomy*. The estimated probabilities for test instances falling into a category is the relative frequency of each class label in that category. To obtain validity, this calculation must include the test instance to be predicted. Since the true label is not known for the test instance, all possible labels must be considered, which results in a set of C label probability distributions, where C is the number of possible labels. For an extended introduction to inductive Venn predictors, see e.g. [6].

The main challenge for Venn predictors is to find a suitable taxonomy. For two-class problems, *Venn-Abers predictors* automatically optimize the taxonomy using isotonic regression [11]. Being Venn predictors, the Venn-Abers predictors inherit the validity guarantees, while the optimized taxonomy will lead to sharp predictions.

Venn-Abers predictors regard the underlying model as a *scoring classifier*, i.e., when the underlying model makes a prediction for a test object x_i, the output is a *prediction score* $s(x_i)$, where a higher value indicates a larger belief in the label 1. Normally, a prediction from a two-class scoring classifier is obtained by comparing the score to a fixed threshold t, and predicting the label 1 if $s(x) > t$, and 0 otherwise. However, by instead fitting an increasing function g using a number of prediction scores with known true targets, $g(s(x))$ can be interpreted as the probability that the label for x is 1. Venn-Abers predictors use isotonic regression, as described in Section 2.3 above, for this purpose. A multiprobabilistic prediction from an inductive Venn-Abers predictor is produced as follows:

1. Let $\{z_1, \ldots, z_l\}$ be a training set where each instance $z_i = (x_i, y_i)$ consists of two parts; an *object* x_i and a *label* y_i.
2. Divide the training set into a proper training set $\{z_1, \ldots, z_q\}$ and a calibration set $\{z_{q+1}, \ldots, z_l\}$.
3. Train a scoring classifier using the proper training set to produce the prediction scores s_0 for $\{z_{q+1}, \ldots, z_l, (x_{l+1}, 0)\}$ and s_1 for $\{z_{q+1}, \ldots, z_l, (x_{l+1}, 1)\}$.
4. Let g_0 be the isotonic calibrator for $\{(s_0(x_{q+1}), y_{q+1}), \ldots, (s_0(x_l), y_l), (s_0(x_{l+1}), 0)\}$
5. Let g_1 be the isotonic calibrator for $\{(s_1(x_{q+1}), y_{q+1}), \ldots, (s_1(x_l), y_l), (s_1(x_{l+1}), 1)\}$.
6. Let the probability interval for $y_{l+1} = 1$ be $[g_0(s_0(x_{l+1})), g_1(s_1(x_{l+1}))]$.

3 Method

In the empirical investigation, we compare different ways of producing probability estimates from a number of standard machine learning algorithms in scikit learn. More specifically, we for each algorithm evaluate the quality of the probability estimates using no external calibration, i.e., the predict_proba method, to using Platt scaling and isotonic regression, as implemented in scikit learn, and Venn-ABERS. For the Venn-ABERS predictors, when converting the probability interval (p_0, p_1), into a single probability estimate for the predicted class, we followed the recommendations in [11] and used a regularized value moved towards the neutral value 0.5.

$$p = \frac{p_1}{1 - p_0 + p_1} \tag{5}$$

In the experimentation, all parameters were left at their default values. For models that should be externally calibrated, the training data was divided into a true training set, consisting of 3/4 of the training instances, and a calibration set.

Table 1: Data Sets

Name and abbreviation		inst.	atts.	Name and abbreviation		inst.	atts.
audit data	au	775	10	ionosphere	io	351	35
banknote authentication	bn	1372	5	liver disorders	li	345	7
blood transfusion	ts	748	5	mammographic masses	ma	830	6
breast cancer coimbra	bc	116	10	musk ver1	mu	476	167
breast cancer wisconsin	bw	683	10	parkinsons	pa	195	23
climate crashes	cl	540	19	raisin grains	ra	900	8
diabetes	da	768	9	rice cammeo osmancik	rc	3810	8
haberman	ha	306	4	sonar	so	208	61
heart cleveland	hc	298	24	spambase	sb	4601	58
heart h	hh	270	27	spectf heart	sp	267	45
heart statlog	hs	270	14	vote	vo	435	49

The 22 data sets used are all two-class problems publicly available from either UCI [1] or CITEDATA [2], see table 1. In total seven predictive modeling algorithms are evaluated, i.e., Adaboost, Decision trees, Gradient boosting, kNN, Logistic regression, (Gaussian) Naive Bayes, and Random forest (RF). All algorithms use the Scikit-Learn default parameter settings. In the experimentation, stratified 10-fold cross-validation was used, where if the dataset size is smaller than 1000 instances, the evaluation was extended to 10×10-fold cross-validation. All results reported are averaged over all folds.

In the analysis, we first look at the predictive performance using accuracy and area under the ROC curve (AUC). When investigating the quality of the

calibration, a number of metrics are used. First we look at the difference between the average confidence in the predicted class and the overall empirical accuracy. While this is a rather crude metric, it gives a very clear indication if a certain algorithm produces models that are systematically either too overconfident or too underconfident. Second, we use the standard log loss function:

$$\lambda_{log} = \begin{cases} -\log p & \text{if } y = 1 \\ -\log(1-p) & \text{if } y = 0 \end{cases} \tag{6}$$

where log is the binary logarithm, and p the estimate for the label 1. Here it must be noted that the log loss function in scikit learn actually clips the probabilities making sure that they never are exactly 0 or 1, thus avoiding infinite results.

We will also use the *Brier loss* (λ_{Br}):

$$\lambda_{Br} = (y - p)^2 \tag{7}$$

Finally, we employ the ECE, a miscalibration metric representing the difference in expectation between confidence and accuracy. Similar to the reliability diagram, in a binary classification problem, the ECE splits the predictions into equally sized M bins taking the weighted average of the bin's fraction of positives fop and the mean of prediction probabilities mop difference as illustrated in the formula. Here, n is the size of the data set and B_m represents the m^{th} bin.

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} \Big| fop(B_m) - mop(B_m) \Big| \tag{8}$$

4 Results

We start by presenting detailed results for one algorithm, i.e, the decision tree. As seen in Table 2 below, there are only small differences in accuracy between calibrated and non-calibrated models. For AUC, however, using no calibration is beneficial.

Table 2: Accuracy and AUC for decision trees

	Accuracy				AUC					Accuracy				AUC			
	No	Platt	Iso	VA	No	Platt	Iso	VA		No	Platt	Iso	VA	No	Platt	Iso	VA
au	.966	.967	.968	.968	.985	.985	.981	.985	li	.648	.643	.651	.654	.680	.669	.665	.664
bn	.963	.964	.966	.967	.981	.986	.986	.986	ma	.828	.830	.828	.828	.880	.879	.881	.881
ts	.768	.768	.765	.765	.687	.679	.675	.677	mu	.735	.735	.733	.734	.784	.777	.779	.781
bc	.648	.645	.640	.643	.701	.686	.672	.680	pa	.834	.839	.844	.847	.840	.840	.837	.840
bw	.945	.943	.946	.948	.978	.978	.976	.979	ra	.825	.829	.835	.837	.890	.890	.889	.890
cl	.919	.920	.922	.922	.770	.774	.753	.781	rc	.905	.905	.907	.907	.948	.947	.946	.947
di	.721	.720	.722	.722	.754	.747	.752	.752	so	.710	.701	.698	.697	.758	.748	.747	.751
ha	.714	.731	.727	.727	.652	.623	.640	.643	sb	.912	.912	.910	.910	.945	.945	.944	.944
hc	.779	.767	.776	.776	.836	.826	.831	.834	sp	.746	.786	.773	.776	.676	.660	.643	.653
hh	.756	.759	.758	.763	.834	.829	.827	.829	vo	.948	.949	.949	.950	.978	.975	.971	.978
hs	.768	.764	.754	.753	.825	.816	.814	.815	**av**	**.814**	**.816**	**.815**	**.816**	**.832**	**.826**	**.824**	**.828**
io	.858	.865	.864	.864	.913	.907	.911	.915	**rk**	**2.82**	**2.55**	**2.43**	**2.20**	**1.64**	**2.64**	**3.55**	**2.18**

While it is interesting to see that using an external calibration method generally resulted in models with worse ranking abilities, this does not mean that these models are well-calibrated. As a matter of fact, when looking at the signed differences in the left part of Table 3 below, it is obvious that the uncalibrated models are very overconfident.

Table 3: Difference and log loss for decision trees

	Signed difference				Log loss					Signed difference				Log loss			
	No	Platt	Iso	VA	No	Platt	Iso	VA		No	Platt	Iso	VA	No	Platt	Iso	VA
au	.009	-.006	.006	-.015	.429	.116	.295	.113	li	.161	.028	.051	.024	3.201	.644	.739	.645
bn	.022	-.006	-.001	-.012	.618	.125	.116	.118	ma	.036	-.008	.014	-.010	1.65	.428	.462	.423
ts	.053	.006	.025	.007	1.68	.512	.584	.516	mu	.154	.007	.035	.014	4.06	.558	.590	.556
bc	.178	.018	.087	.029	3.81	.646	1.647	.649	pa	.085	.000	.036	-.017	2.29	.406	1.216	.393
bw	.019	-.004	.007	-.015	.629	.165	.294	.156	ra	.083	.008	.021	.001	2.16	.414	.425	.403
cl	.036	.015	.019	-.001	1.34	.234	.318	.230	rc	.042	.006	.005	-.001	1.28	.268	.262	.259
di	.135	.021	.037	.021	2.96	.563	.604	.560	so	.178	.006	.050	.018	4.55	.594	.853	.595
ha	.092	-.005	.016	-.010	2.12	.565	.831	.565	sb	.033	-.001	.001	-.004	1.34	.259	.261	.260
hc	.084	.010	.039	-.005	2.11	.512	.927	.497	sp	.152	-.016	.015	-.013	3.50	.488	.561	.495
hh	.099	.007	.034	-.014	2.02	.494	.961	.492	vo	.013	-.007	.016	-.020	.551	.161	.620	.159
hs	.099	-.003	.047	.005	2.43	.517	.839	.519	**av**	**-**	**-**	**-**	**-**	**2.11**	**.410**	**.637**	**.406**
io	.072	-.005	.024	-.009	1.66	.351	.603	.335	**rk**	**-**	**-**	**-**	**-**	**3.95**	**1.73**	**2.91**	**1.41**

Over all datasets, the tree models are on average more than eight percentage points too optimistic, i.e., these models must be considered misleading. Some-

what surprising, isotonic regression, is actually also systematically optimistic, but of course to a much lesser degree. Platt scaling and Venn-Abers, on the other hand, appear to be well-calibrated. The right part of Table 3 shows the log losses. As expected, isotonic regression suffers from infinite log losses, which, however, are converted to very large values from the usage of clipping in the scikit-learn logloss function. Still, the main result is of course that all calibration techniques lower the average log losses substantially. In particular Platt scaling and Venn-Abers obtain good results over all data sets.

Turning to Brier loss and ECE in Table 4 below, we see that these results are consistent with the log losses; all calibration techniques produce models with lower Brier losses and ECE:s. Looking specifically at ECE:s, we see that the reduction from calibration is often significant. On average, it is more than 50%. Comparing the different techniques, Venn-Abers is, based on the mean ranks, the most successful, although the differences in real numbers are rather small.

Table 4: Brier loss and ECE for decision trees

	Brier loss				ECE					Brier loss				ECE			
	No	Platt	Iso	VA	No	Platt	Iso	VA		No	Platt	Iso	VA	No	Platt	Iso	VA
au	.027	.027	.028	.027	.012	.013	.011	.016	li	.254	.225	.229	.225	.139	.052	.061	.039
bn	.030	.030	.028	.028	.018	.016	.009	.017	ma	.135	.132	.131	.131	.045	.026	.029	.029
ts	.177	.167	.170	.168	.062	.014	.031	.031	mu	.210	.186	.187	.185	.096	.022	.037	.021
bc	.253	.225	.242	.227	.130	.056	.076	.050	pa	.127	.122	.124	.120	.076	.040	.040	.041
bw	.042	.043	.043	.042	.021	.021	.009	.018	ra	.133	.127	.124	.123	.069	.049	.026	.021
cl	.067	.063	.064	.062	.054	.015	.017	.007	rc	.076	.075	.073	.073	.030	.019	.010	.011
di	.209	.190	.190	.188	.096	.036	.041	.033	so	.233	.202	.208	.203	.124	.045	.064	.036
ha	.206	.189	.192	.190	.081	.023	.042	.043	sb	.073	.071	.072	.072	.026	.013	.011	.014
hc	.172	.166	.164	.161	.078	.048	.047	.032	sp	.191	.157	.162	.160	.138	.037	.041	.052
hh	.166	.162	.165	.162	.074	.044	.040	.022	vo	.040	.040	.042	.041	.017	.024	.017	.020
hs	.175	.169	.173	.170	.070	.040	.054	.038	**av**	**.141**	**.131**	**.132**	**.130**	**.069**	**.031**	**.034**	**.028**
io	.103	.103	.100	.098	.060	.040	.029	.030	**rk**	**3.55**	**2.09**	**2.73**	**1.64**	**3.77**	**2.18**	**2.09**	**1.95**

Of course, the benefit of using calibration varies between different algorithms and data sets. Fig. 1 below presents calibration curves for one example where all three calibration techniques are able to substantially improve the probabilistic model. In particular isotonic regression and Venn-Abers produce very well-calibrated models.

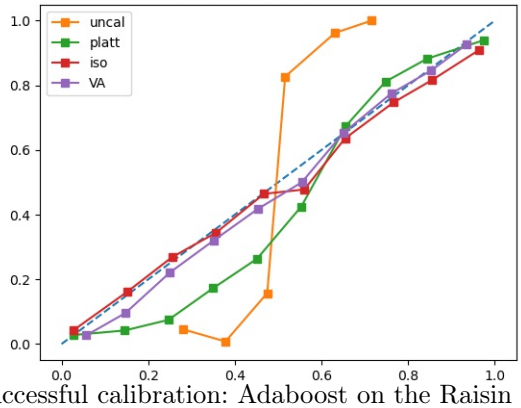

Fig. 1: Successful calibration: Adaboost on the Raisin grains data set

Unfortunately, there are also several cases when the calibration does not improve the models. The most common reason is that the calibration is rather good to start with, see the calibration curves in Fig. 2 below.

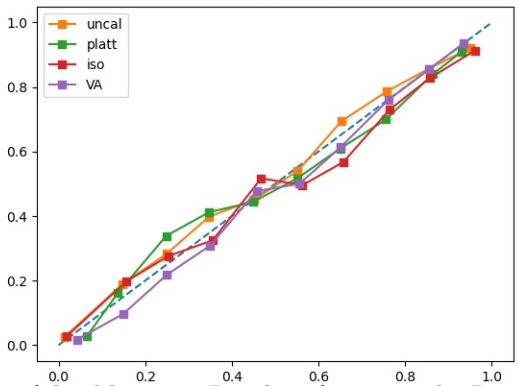

Fig. 2: Unsuccessful calibration: Random forest on the Raising grains data set.

Table 5 below presents the main results. Starting with how well-calibrated the models are, as can be seen from the mean ranks and the coloring, we see that logistic regression is the only algorithm where external calibration is detrimental for the probabilistic models. For random forest, the picture is a little bit mixed, but overall the benefit of post-calibration is very small. For the other algorithms, i.e., Adaboost, Gradient Boosting, kNN, Naive Bayes, and Decision Tree, however, all three calibration techniques clearly make the models more well-calibrated. Comparing the different calibration techniques, the order is most often Venn-Abers followed by Platt scaling and Isotonic regression.

Table 5: Main results

		Mean Score				Mean rank			
		No	Platt	Iso	VA	No	Platt	Iso	VA
Adaboost	Acc	.844	.834	.839	.839	1.48	3.00	2.73	2.80
	AUC	.868	.856	.855	.862	1.55	2.50	3.64	2.32
	Diff	-.283	-.022	.032	-.020				
	Log	.601	.388	.921	.366	3.32	1.59	3.68	1.41
	Brier	.206	.119	.118	.115	4.00	2.27	2.09	1.64
	ECE	.285	.049	.035	.037	4.00	2.23	1.82	1.95
Grad. boost	Acc	.815	.820	.820	.820	2.55	2.55	2.50	2.41
	AUC	.819	.816	.820	.824	2.59	3.00	2.77	1.64
	Diff	.159	.000	.025	-.008				
	Log	1.29	.408	.717	.403	3.73	1.77	3.27	1.23
	Brier	.171	.130	.131	.129	4.00	2.05	2.55	1.41
	ECE	.161	.029	.036	.030	4.00	1.91	2.23	1.86
Log. reg.	Acc	.846	.836	.841	.842	1.93	2.57	2.89	2.61
	AUC	.883	.874	.868	.874	1.05	2.36	3.77	2.82
	Diff	.014	-.017	.035	-.020				
	Log	.468	.371	.948	.358	1.68	2.27	3.82	2.23
	Brier	.114	.117	.116	.113	1.68	2.36	3.36	2.59
	ECE	.048	.043	.036	.038	2.23	2.68	2.41	2.68
kNN	Acc	.788	.792	.789	.790	2.23	2.18	3.11	2.48
	AUC	.803	.797	.789	.793	1.39	2.11	3.50	3.00
	Diff	.056	.007	.023	.000				
	Log	1.40	.425	.675	.426	3.82	1.64	2.91	1.64
	Brier	.147	.140	.142	.140	3.32	1.86	2.82	2.00
	ECE	.061	.034	.033	.031	3.73	2.23	2.05	2.00
Naive Bayes	Acc	.794	.805	.822	.822	2.93	2.34	2.50	2.23
	AUC	.858	.822	.847	.853	1.27	3.82	2.86	2.05
	Diff	.142	.003	.036	-.014				
	Log	1.77	.427	.937	.393	3.50	2.05	3.41	1.05
	Brier	.173	.138	.129	.126	3.77	2.68	2.27	1.27
	ECE	.147	.034	.040	.037	3.86	1.95	2.18	2.00
Rand. Forest	Acc	.855	.855	.851	.852	2.00	1.95	3.39	2.66
	AUC	.894	.888	.878	.885	1.00	2.45	3.91	2.64
	Diff	-.021	-.001	.034	-.024				
	Log	.340	.327	.897	.336	1.91	1.73	4.00	2.36
	Brier	.105	.103	.108	.105	2.00	1.68	3.50	2.82
	ECE	.049	.025	.033	.037	3.09	1.91	2.32	2.68
Decision Tree	Acc	.814	.816	.815	.816	2.82	2.55	2.43	2.20
	AUC	.832	.826	.824	.828	1.64	2.64	3.55	2.18
	Diff	.083	.003	.027	-.001				
	Log	2.10	.410	.637	.406	3.95	1.73	2.91	1.41
	Brier	.141	.131	.132	.130	3.55	2.09	2.73	1.64
	ECE	.069	.031	.034	.028	3.77	2.18	2.09	1.95

Looking finally at the intervals produced by the Venn-Abers predictor in Table 6 below, we see that most intervals are fairly tight, and generally cover the true accuracy. Intervals not covering the empirical accuracies are given in bold. Interestingly enough, the sizes vary substantially between the different algorithms, with the tightest actually produced using kNN. Overall, it is of course important to recognize that the width of the intervals give an indication of how certain the models are about their probabilistic predictions, i.e., it is a measure of how confident the model is in its probability estimates.

Table 6: VA predictors

	DT		Adab		Gboost		Logreg		kNN		NB		RF	
	low	high	low	high	low	high	low	high	low	high	low	high	low	high
au	.950	.977	.940	.991	.963	.982	.909	.972	.955	.980	.919	.970	.939	.991
bn	.953	.968	.968	.998	**.965**	**.978**	.967	.998	.992	1.00	.835	.882	.964	.996
ts	.764	.799	.761	.808	.742	.789	.766	.820	.755	.779	.749	.799	.755	.807
bc	**.644**	**.760**	.614	.797	.645	.747	.646	.834	**.530**	**.604**	.653	.812	.651	.837
bw	.927	.958	.905	.971	.930	.959	.916	.980	.953	.980	.924	.973	.927	.981
cl	.916	.946	.907	.965	.909	.942	.915	.978	.920	.947	.912	.975	.902	.964
di	**.733**	**.770**	.725	.788	.719	.784	.744	.810	.707	.737	.725	.791	.731	.799
ha	.702	.760	.703	.770	.676	.760	.710	.798	.712	.760	.714	.791	.687	.776
hc	.755	.831	.730	.846	.727	.798	.787	.908	.616	.680	.772	.890	.758	.885
hh	.728	.812	.725	.841	.722	.825	.750	.881	**.658**	**.708**	.768	.886	.736	.869
hs	.741	.817	.719	.840	.724	.795	.778	.906	**.620**	**.684**	.768	.895	.761	.894
io	.846	.897	.851	.951	.848	.916	.794	.902	.855	.902	.824	.929	.855	.961
li	**.661**	**.722**	**.697**	**.794**	**.671**	**.756**	.678	.769	**.665**	**.718**	**.662**	**.744**	.705	.808
ma	.809	.850	.792	.851	.750	.810	.792	.856	.783	.812	.769	.833	.790	.853
mu	**.737**	**.780**	.821	.909	.766	.840	.781	.870	.826	.874	.700	.775	.808	.904
pa	.814	.894	.813	.956	.830	.904	.798	.941	.802	.889	.808	.935	.836	.967
ra	.831	.863	.821	.884	.818	.880	.830	.891	.821	.848	.802	.864	.832	.893
rc	.905	.914	.906	.930	.902	.928	.916	.940	.876	.883	.892	.918	.909	.935
so	.696	.770	.728	.876	.700	.778	.694	.845	.703	.788	.684	.826	.756	.911
sb	.904	.914	.928	.948	.933	.954	.912	.933	**.785**	**.791**	.894	.908	.925	.946
sp	.750	.802	.741	.848	.743	.800	.739	.849	.736	.792	.757	.860	.747	.855
vo	.924	.971	.890	.980	.921	.962	.891	.985	.907	.954	.896	.966	.898	.985
Avg. Size	**.049**		**.084**		**.058**		**.089**		**.042**		**.082**		**.088**	
Outside	**4**		**1**		**2**		**0**		**5**		**1**		**0**	

5 Concluding remarks

We have in this paper empirically investigated how well-calibrated probabilistic models different algorithms in scikit-learn produce. In general, the results showed that all algorithms but logistic regression benefit from using external calibration. Specifically, using any of the evaluated calibration techniques, i.e.,

Platt scaling, isotonic regression, and Venn-Abers, the very overconfident probability estimates from, in particular, Decision Trees, Naive Bayes, and the boosted models are substantially improved. While, Platt scaling and Venn-Abers outperformed isotonic regression in the experimentation, it should be noted that the data sets are rather small, which is a well-known drawback for isotonic regression.

References

1. Bache, K., Lichman, M.: UCI machine learning repository (2013)
2. Çinar, İ., Koklu, M., Taşdemir, Ş.: Classification of raisin grains using machine vision and artificial intelligence methods. Gazi Mühendislik Bilimleri Dergisi (GMBD) 6(3), 200–209
3. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1321–1330. PMLR (2017)
4. Johansson, U., Gabrielsson, P.: Are traditional neural networks well-calibrated? In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2019)
5. Johansson, U., Löfström, T., Sundell, H., Linusson, H., Gidenstam, A., Boström, H.: Venn predictors for well-calibrated probability estimation trees. In: Seventh Symposium on Conformal and Probabilistic Prediction with Applications. Proceedings of Machine Learning Research, vol. 91, pp. 1–12. PMLR (2018)
6. Johansson, U., Löfström, T., Boström, H.: Calibrating probability estimation trees using venn-abers predictors. In: Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019. pp. 28–36 (2019)
7. Lambrou, A., Nouretdinov, I., Papadopoulos, H.: Inductive venn prediction. Annals of Mathematics and Artificial Intelligence 74(1), 181–201 (2015)
8. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd international conference on Machine learning. pp. 625–632. ACM (2005)
9. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. pp. 61–74. MIT Press (1999)
10. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Mach. Learn. 52(3), 199–215 (2003)
11. Vovk, V., Petej, I.: Venn-abers predictors. arXiv preprint arXiv:1211.0025 (2012)
12. Vovk, V., Shafer, G., Nouretdinov, I.: Self-calibrating probability forecasting. In: Advances in Neural Information Processing Systems. pp. 1133–1140 (2004)
13. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: Proc. 18th International Conference on Machine Learning. pp. 609–616 (2001)