JOIM
www.joim.com

Journal Of Investment Management, Vol. 18, No. 2, (2020), pp. 9–22

ON THE STABILITY OF MACHINE LEARNING MODELS: MEASURING MODEL AND OUTCOME VARIANCE

Vasant Dhar[a,b] *and Haoyuan Yu*[c]

How do you know how much you should trust a model that is learned from data? We propose that a central criterion in measuring trust is the decision-making variance of a model. We call this "model variance." Conceptually, it refers to the inherent instability machine learning models experience in their decision-making in response to variations in the training data. We report the results from a controlled study that measures model variance as a function of (1) the inherent predictability of a problem and (2) the frequency of the occurrence of the class of interest. The results provide important guidelines for what we should expect from machine learning methods for the range of problems that vary across different levels of predictability and base rates, thereby making the results of general scientific interest.

1 Introduction

Machine learning is creeping into every area of our personal and commercial lives. A staggering amount of data is generated with each passing day that machines can leverage to become increasingly better at prediction autonomously. For example, every time someone uses Google's sentence completion suggestion, Google acquires a "label" about whether the suggested sentence was a good one or not. Every "like" on Facebook similarly provides a lot of information about the liker. Likewise, satellite snapshots and shipping logs provide information on economic activity at a high level of granularity.

While the increasing prevalence of massive amounts of data is a rich potential input to autonomous machine learning-based systems, it raises fundamental issues about when we should trust such systems with decision-making. Ceding control of decision-making to a machine requires a high degree of trust. Will it behave in accordance with expectations? We provide an answer to this question by quantifying expectations based on the nature of the problem expressed in terms of two variables.

Autonomous learning systems have already become very good at certain kinds of tasks like

[a]Stern School of Business, New York, NY 10011, USA.
[b]Center for Data Science, NY, USA.
[c]SCT Capital Management, NYU, NY 10012, USA.

vision, where they achieve exceedingly low error rates at recognizing objects, like under one in a million misclassifications. High accuracy is essential in such domains since the risk, measured in terms of costs of error, is exceedingly high. Imagine a self-driving car ploughing into a bunch of kids by mistake or a malfunction in an airplane's control system that does not let a human take over in a catastrophic situation. Despite the high accuracy of these systems, their potentially catastrophic consequences limit our trust in them.

Finance problems tend to be "noisier." In capital markets, for example, the "predictability," which we refer to as p, is inherently low. New information is reflected very quickly in prices. There is intense competition among actors where obvious advantages — in the form of lead/lag relationships among variables — disappear quickly. This intense competition coupled with unpredictable exogenous shocks makes financial markets have low predictability, where achieving better than random performance on a long-term basis is challenging.

Other finance problems, such as in credit markets, tend to have higher predictability than capital markets but tend to be challenging for other reasons such as their low "base rates," namely, the low prevalence of the phenomena of interest — such as defaults. For example, if defaults occur less than 1% of the time, a system that always predicts "no default" is over 99% accurate. But it is useless for decision-making since the objective is to predict the defaults, whose consequences are severe. The challenge is that predicting the 1% true defaults while avoiding errors can be extremely difficult. For low predictability problems, the lower this "base rate" of the phenomenon of interest, the harder it is to predict them correctly without increasing errors in the form of "false positives." We refer to the base rate as b.

2 The framework

Figure 1 provides a map of finance problems along varying levels of predictability and base rate, that is, various levels of p and b. The positioning is approximate and based on our experience with applying machine learning to the problems. In Figure 1, darker red implies more difficulty for the learner, whereas a darker green corresponds to easier problems.

The positioning of problems on the map determines the properties of the model that a learning algorithm is able to extract from the data. A key property of interest is the decision-making stability of the model, its model variance, which indicates the consistency of its decisions in response to variations in training data. A high model variance for small changes in training data would be worrisome, although some level of variation is to be expected.

Figure 1 maps problems typically encountered in finance in terms of predictability and base rates.

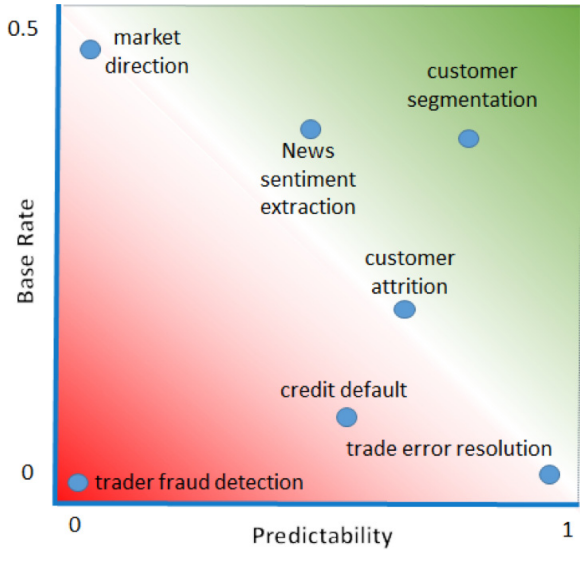

Figure 1 Map of finance problems. Problems are easier on top right (green) part. Trader fraud is nearly impossible to predict by machine.

The central question we discuss in this paper is "when should we trust an autonomous machine learning-based system with decision-making?" The answer, it turns out, is based on its decision-making stability as a function of predictability and base rate. We call this model variance. If the expected model performance is good and the model variance is low, we should trust it, otherwise we should not. This is the first paper to analyze how the two factors impact decision-making stability and hence our trust in machine learning-based models.

The first variable, **predictability**, varies between zero and one. Zero implies complete randomness and one implies complete determinism or predictability. Mathematically, it can be defined as compliment of the well-known concept of entropy in information theory. Entropy measures the extent of disorder in a dataset. For binary outcomes drawn from a Bernoulli distribution, the entropy $H_b(q)$, is defined as

$$H_b(q) = -q \log_2 q - (1-q) \log_2 (1-q) \tag{1}$$

where q is the probability of the outcome being one of the two outcomes in the corresponding Bernoulli distribution. In our data creation process, described fully in the Experimental Setup, some data have probability 1 of being classified correctly, while other data have probability 0.5. If we denote N as the total number of observations, p the percentage of desired labels, the expected binary entropy of such N observations is

$$\begin{aligned} E(H_b) &= \frac{pN\, H_b(1) + (1-p)N\, H_b(0.5)}{N} \\ &= 1 - p. \end{aligned} \tag{2}$$

Or,

$$\begin{aligned} E(H_b) &= \{p\, H_b(1) + (1-p)\, H_b(0.5) \\ &= 1 - p. \end{aligned} \tag{3}$$

The compliment of the above, $1 - (1-p) = p$, is defined as predictability.

The second variable, the **"base rate"** of the class of interest varies between 0 and 0.5, where 0.5 stands for a "balanced" dataset where the class of interest occurs 50% of the time, whereas a value of 0.01 indicates that the class of interest occurs 1% of the time. In credit markets, for example, defaults are typically rare, implying low base rates. In capital markets, if we are interested in how often does the market go "up" and "down," the answer is almost equally[1].

Why are these the critical variables? Our experience with running a machine learning-based hedge fund for the last two decades is that the most vexing problem with machine learning methods in Finance is the "model variance" problem. What this means is that when small changes in the training data result in different models that make very different decisions in an identical situation, we cannot trust the model. The noisier the problem, the more severe this problem. This should make intuitive sense. For example, if we remove a few critical "outlier" days from the financial crisis from a training set, should we expect the resulting model from the learning algorithm to be different from one without withholding this data? We should, since the loss function will optimize performance differently, by predicting the outlier days correctly. Achieving model stability is a challenge in finance.

The model variance problem is also heavily impacted by the base rate. For example, if defaults occur 1% of the time versus 50% of the time for the same level of predictability, where should we expect higher model variance?

Taken together, predictability and base rates interact in subtle ways and can have a significant impact on the expected performance of a model

and its variance. Accordingly, what we answer in this paper is the following question:

What is the extent of model variance for a problem as a function of predictability and base rate?

The answer to the above question provides an expectation for model variance, if we can specify the levels of the two variables. At the current time, without estimates of predictability and base rates, researchers have no yardstick for determining how stable they should expect their models to be.

In this paper, we present summary results based on 20 years of research and trading with machine learning models in financial markets. To make the results scientific and replicable, we use carefully constructed synthetic datasets and use a standard decision tree-based machine learning algorithm without any hyper-parameter optimization. The synthetic datasets correspond to a wide range of problems encountered in reality, so we expect our results to be of general value.

We also analyze the "outcome variance" on a performance measure such as the Information Ratio. This contrasts with model variance which measures the variance in decisions resulting from the different models learned from the different training sets. Specifically, we answer the following question:

How do predictability and base rate impact outcome variance specified in terms of a measure of performance?

Our research contributions are threefold. First, there is currently no answer to the central questions above, so our results are the first in this area, and of general interest and have widespread applicability. Second, we conjecture that previous research studies that have analyzed the impacts of training based on various training set class distributions have ignored predictability and base rates and are therefore unable to generalize their results. Indeed, we conjecture that the confounding results in performance by creating synthetic class distributions in training data through oversampling and under-sampling (Weiss and Provost, 2001) might be explainable if we had access to the predictability and base rates in the datasets used.

Finally, our results should provide a firm set of expectations about what researchers should expect when modeling different problems with varying levels of predictability and base rates. We find that researchers often do not have expectations of the upper bounds of performance and the stability of their results. Because of this, they are often surprised by how differently models perform in practice relative to their expectations which are often based on point estimates. Our research suggests that it is fruitful to think in terms of "confidence intervals" of performance with machine learning-based models instead of point estimates which might be unreliable if the model variance is high. This is particularly true for low predictability problems. The notion of a confidence interval applies both to model and outcome variance.

3 Experimental setup

Figure 2 describes the experimental setup and objectives. The two horizontal axes represent the two variables whose impact we want to analyze on performance and the variance of the performance. We construct datasets corresponding to different levels of predictability and base rates.

We consider a binary classification problem using an off-the-shelf UCI dataset. The original dataset, called COVER, has ∼500 K observations of forest types. The data has 12 features ($X \in R^{12}$) and 1 binary target variable ($Y \in \{0, 1\}$)[2]. We

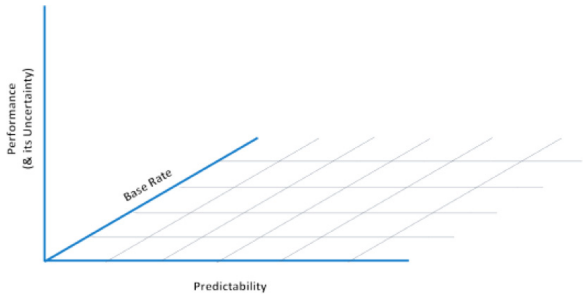

Figure 2 Experimental variables and objectives.

selected 200K observations equally from both classes as the baseline dataset from which to create the desired datasets corresponding to different levels of p and b.

3.1 Dataset creation

We used the data to create new datasets with specific levels on two dimensions of interest, namely, predictability, and the base rate of minority class using following algorithm:

1. Build a decision tree model on the original dataset. Call this model M.
2. Re-label the data in each leaf node of M according to its majority class so that every leaf node is pure, i.e. every datum is of the same class. Call this model P. This dataset is considered 100% predictable, meaning that the algorithm can recover P without error[3].
3. Create different base rate b datasets (1–50%) by removing "positives" (i.e. the minority class, which is of interest) randomly until the desired base rate is achieved.
4. For each base rate dataset, create different levels of predictability p as follows:
   (a) Take p% of the labels from P
   (b) Assign the remaining labels with a coin flip according to the base rate b.

In our experiment, we considered the following 15 values for predictability p between 1% and 100%:

[0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 1]

and the following 10 values of the base rates for b between 1% and 50%:

[0.01, 0.02, 0.05, 0.1, 0.25, 0.4, 0.45, 0.48, 0.49, 0.5].

We get 150 datasets with the requisite combinations of p and b. Note that in every dataset, the majority class always has 200K observations, whereas and minority class observations vary according to the base rate.

3.2 Model recovery and the test set

For each predictability p & base rate b combination (p, b), we name $\alpha_{p,b}$ to be the full training set and $\beta_{p,b}$ to be the test set. We randomly pick 10% of total observations to be $\beta_{p,b}$ and the remaining 90% to be $\alpha_{p,b}$.

The same tree algorithm is used to recover the model. To keep the experiment replicable we use the standard tree induction algorithm from Scikit-learn. We also built decision tree classifiers with different minimum leaf node sizes that correspond to different levels of complexity. Specifically, we attempted to regularize the model by specifying minimum numbers of instances in the leaf nodes. No maximum tree depth was specified — the nodes are expanded until all leaves are pure or contain less than minimum leaf node size samples. At each split, all 12 features were considered as candidates.

Two groups of experiment are designed for two purposes. Experiment A always uses full $\alpha_{p,b}$ as training set to analyze prediction rate for minority class and model performance. This experiment is motivated by our experience with machine learning in capital markets. Specifically, we have

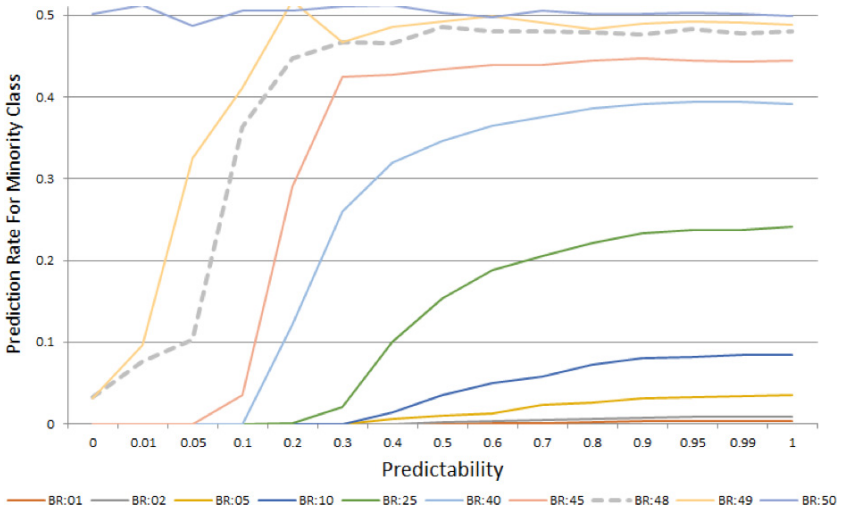

Figure 3 How often the model predicts minority class. One can see that for heavily skewed data sets, they need more predictability to even make a minority class prediction.

observed that in low signal and low base rate problems, the model is unable to make any predictions about the minority class. For example, with a 1% base rate, a model that always predicts the majority class will have 99% accuracy. Unless there is sufficient signal in the problem, the model will have a difficult time predicting the minority class accurately and will therefore avoid predicting it.

Experiment B, focusing on the model variance issue, further bootstraps $\alpha_{p,b}$ 20 times to create 20 different training sets for each (p, b) combination but uses the same test set $\beta_{p,b}$. This experiment gives us the expected performance and its variance that results from model variance.

4 Results

4.1 Prediction rate for minority class and AUC

For all experiments in the rest of paper, we always chose the default threshold of 0.5 to classify data. One can always lower the threshold to get more prediction of the minority class or do the opposite to get less. For simplicity, we keep this variable fixed but this is always an obvious choice.

Figure 3 shows the prediction rate for the minority class as a function of p and b.

The figure paints a fascinating picture about how the two variables interact in terms of what a learner is able to learn from the data. First, notice the convergence of the prediction rate for the minority class towards the base rate as predictability approaches one. This makes sense: if there is complete predictability, we should be able to recover the model perfectly and predict all labels correctly, so we should expect the predictability to approach the base rate with increasing predictability.

As predictability decreases, so does the learner's ability to find the minority class patterns, where below some level, it cannot identify any of them and always predicts the majority class. A key takeaway from the figure is that the learner's predictions are even more skewed towards the majority class, in inverse proportion to predictability, amplifying a small bias in the base rate.

What is particularly interesting is that for signal levels below 35%, the learner is unable to make any predictions of the minority class for base rates under 10%. The practical implications

of this result are significant in that it suggests that for low base rates, such as credit default rates that are typically well under 10%, the learner will have an impossible time predicting the minority class unless there is a significant amount of signal in the problem.

What is perhaps most interesting in the figure is the part towards predictability below 10%. Note how the learner is able to make many more predictions of the minority class for the high base rates as predictability increases even marginally. The good news here is that as the class distribution becomes more balanced, the learner is able to predict the minority class even at low levels of predictability, i.e. levels of under 5%. Thus, while such problems might be difficult to predict, such as tomorrow's market direction, it is possible to do so as long as there is a minimal level of signal in the problem. So, even though predicting market direction is difficult, the results suggest that assembling datasets which provide a minimal level of signal could be worthwhile.

Figure 4 provides the AUC curves corresponding to the results. It shows that for low base rates such as 1%, the learner is unable to do better than random (i.e. AUC = 0.5) until predictability exceeds 50%. For a 2% base rate, predictability must exceed 40%. As in Figure 3, we can see that for the balanced class problem, we are able to do better than random even at low predictability levels between 1% and 5%.

4.2 Model variance

We would expect a reliable model to make similar decisions and hence have similar performance on both in sample and out of sample data. To achieve this, we need the model to be stable, where its decision patterns are relatively consistent across variations in the training data.

In this set of experiments, we analyze how model's decisions vary in response to variations in the training set across problems with varying levels of predictability and base rates. We also analyze outcome variance.

In a predictive model, model variance measures the variations of decisions, measured on a fixed test set, of models created from variations in a training set. It is a similar technique in concept to cross-validation but with important differences. In cross-validation, both training set and test set vary but in measuring model variance, only training set varies while test set is fixed. This is a useful measure of the stability of the model.

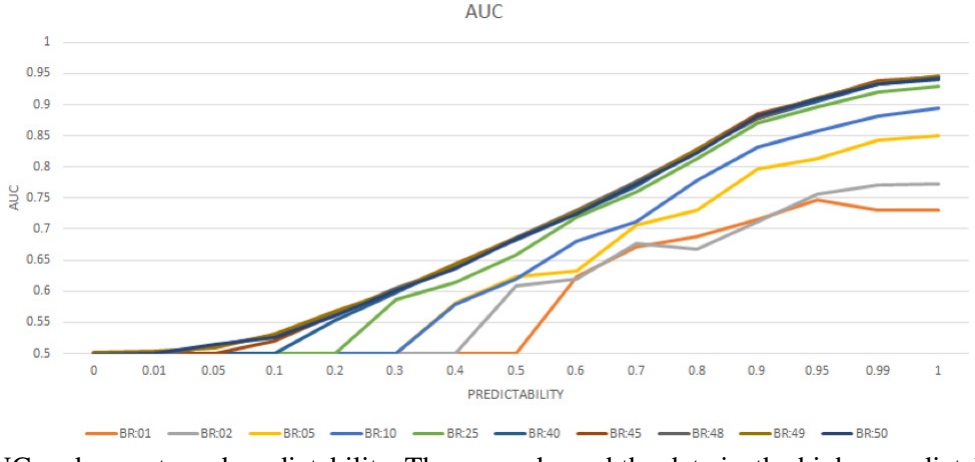

Figure 4 AUC on base rate and predictability. The more skewed the data is, the higher predictability it needs to do better than random.

Model variance is also different from the concept of overfitting in machine learning. We always try to minimize overfitting for any problem. In contrast, model variance is an outcome that results from the nature of the problem, and will exist even when there is zero overfitting. In line with this thinking, it does not make sense to minimize model variance since this would result in the simplest possible model where every prediction is identical. Rather, it makes sense to expect some reasonable level of model variance depending on the problem.

In contrast to model variance, outcome variance measures how differently the models perform on the test set using a performance measure such as the Sharpe Ratio or prediction accuracy. This is a useful measure of the aggregate stability of performance.

It should be evident that higher model variance does not always involve higher outcome variance: very different models could result in the same performance. For example, variants of a random dataset could yield very different models which "fit to the noise" in the training data but give the same outcome results.

We define average flip rate Γ to be the measure of model variance. In our binary classification problem, each bootstrap in training data gives a prediction vector $\hat{Y}_\theta \in \{0, 1\}^m$, where θ enumerates bootstraps and m is the number of observations in test set. Since all $\hat{Y}$'s are binary vectors, we can simply count the number of different elements in each pair, noted as $D(\hat{Y}_i, \hat{Y}_j)$, which can also be interpreted as the $L1$ distance of the two $\hat{Y}$'s in R^m space. To get Γ, we sum up all such distances in all pairs and divide it by number of pairs and vector dimension m:

$$\Gamma = \frac{1}{m}\frac{1}{C_K^2}\sum_{i\neq j} D(\hat{Y}_i,\ \hat{Y}_j), \tag{4}$$

Table 1 Actual versus three predictions. Red numbers are wrong predictions.

actual	pred1	pred2	pred3
0	1	0	0
1	1	1	0
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	1
1	1	1	0
0	0	1	0
1	1	1	0
0	0	0	0

where K is the total number of bootstraps and $C_K^2 = \frac{K(K-1)}{2}$ is the "k choose 2" combinatorial formula.

Table 1 provides an example. The columns "pred1," "pred2," and "pred3" are three prediction vectors from same model trained on three different training set bootstraps. Because model variance does not involve any performance measure, let us ignore the column "actual" for now. In table 1, $m = 10$, $K = 3$, $D(\hat{Y}_1, \hat{Y}_2) = 6$, $D(\hat{Y}_1, \hat{Y}_3) = 6$, $D(\hat{Y}_2, \hat{Y}_3) = 6$, so $\Gamma = 0.6$. This can be interpreted as, on average, 0.6 (or 60%) of predictions will be different when running the model again on a different training set bootstrap.

In our experiment, we performed 20 bootstraps on each $\alpha_{p,b}$, which give us 190 pairs. m varies across different $\beta_{p,b}$ within the range (400, 20K) corresponding to the different base rates.

Figure 5 shows 90% bootstrap results for model variance across the various base rates. Colder colors, which represent balanced datasets, show a not surprising decrease in model variance with increase in predictability, whereas interestingly the warmer colors show an increase. This might seem odd, but it is consistent with the results in

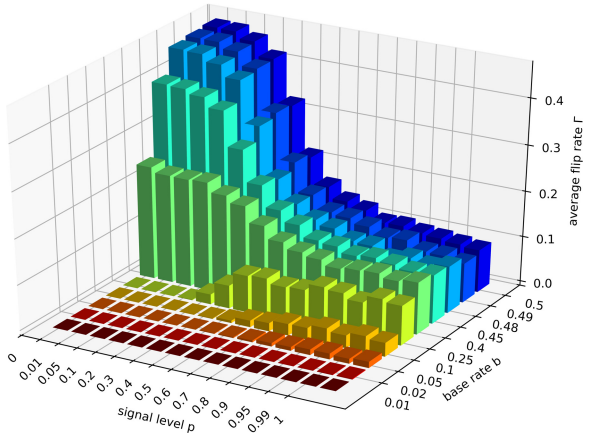

Figure 5 The whole picture of model variance. On balanced data, higher predictability reduces model variance; on skewed data, it is the opposite.

Figures 3 and 4 where the AUC = 0.5 for low levels of predictability and base rates. In these conditions, the model always predicts the majority class and hence variance is zero or very low. As predictability increases, the learner is able to identify and hence predict increasing numbers of the minority class, but it also gets more of them wrong, which results in higher model variance. It is important to note that the AUC = 0.5 does not indicate random predictions but the inability of the model to predict the minority class when the decision threshold is 0.5.

4.3 Outcome variance

Outcome variance requires a performance measure. Consider Table 1 again, and let us consider overall accuracy as a performance measure. The three models are 70% correct so the outcome variance on accuracy is zero. Even though the models generate significantly different predictions, their aggregate performance is consistent at 70% correct — what differs each time is which ones are correct. Note that the outcome variance on a different measure such as Recall (or Sharpe Ratio), could be quite different.

In our experiment, we considered recall of the minority class as a performance measure, and its standard deviation as a measure of outcome variance.

The mean and standard deviation of performance, in this case recall, for all levels of predictability and base rates are shown in Figure 6 and Figure 7, respectively.

Figure 6 shows that performance goes up with more predictability across the base rates, but it becomes more difficult to get an extra edge in performance for low base rates. Figure 7 shows

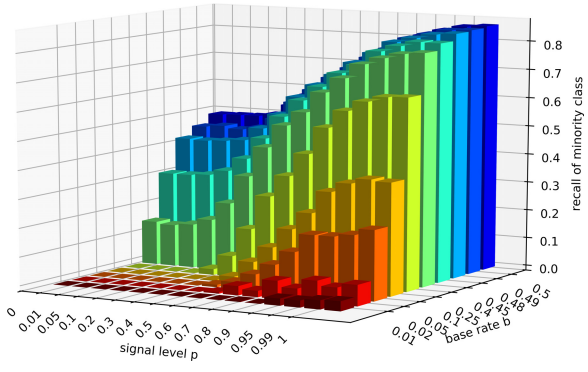

Figure 6 Performance goes up uniformly with higher predictability and more balanced data.

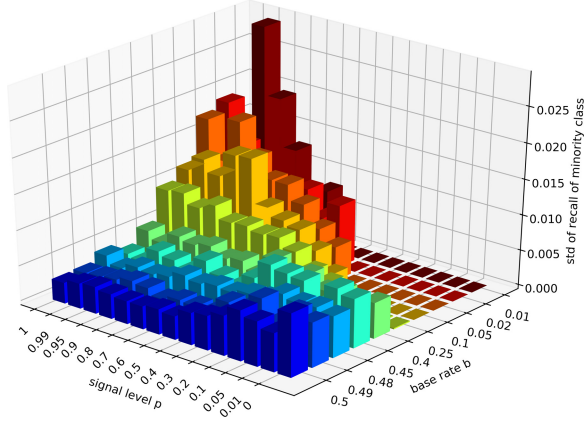

Figure 7 Outcome variance on all data. On skewed data, outcome variance goes up with higher predictability. On balanced data, outcome variance slightly goes down with higher predictability.

the variance of recall. We reversed the base rate axis to make the results more transparent.

Two high-level patterns stand out in the figure. The first is that lower base rates lead to higher levels of variance or instability. This finding accords with the literature and our expectations: when the phenomenon of interest is rare, it is difficult to predict them accurately since the tendency of a learner is heavily biased towards the majority class, which in effect increases its accuracy. For example, a model that wants to maximize accuracy when the base rate is 1% will have 99% accuracy by always predicting the majority class. The system will generate increasing numbers of "false positives" with declining predictability. Notice how the direction of the above pattern reverses roughly in the middle of the base rate axis: as the class distribution becomes more balanced, variance goes down with predictability. In other words, for the higher base rates, instability increases with decreasing predictability. Again, this is not surprising. As randomness increases, we should expect higher variability in the learned models. While it may seem puzzling that variance increases with predictability for the low base rates, the reason is that for low base rates and low predictability, the learner is unable to predict the minority class at all, so there is zero variance in outcomes! In this sense, increasing variance with higher predictability is a good sign, in that the learner is actually able to learn to predict the minority class.

5 Discussion

It is important to have expectations about the potential behaviors of machine learning models for prediction. In our experience with machine learning in finance, we find that expectations are lacking or unclear and researchers are often unclear about whether they should trust a model. A major concern for this lack of trust is the uncertainty associated with the model's future performance. Back-tests of strategies give us the results of how a model would have played out in one reality only. However, the future is seldom identical to the past, so there is good reason to be skeptical about back-test results.

Our research has been motivated at revealing the equivalent of a confidence interval associated with a particular model. What we have demonstrated is that the confidence interval will be impacted by an interplay of the complexity of the machine learning model in combination with the inherent predictability of the problem and the base rate of occurrence of the phenomenon of interest. For a fixed predictability and base rate, the model variance will be impacted by our choice of the complexity level of the machine learning model, so this must be determined carefully.

In this paper, we have not addressed the problem of how to choose the right complexity parameters, but clearly, model variance is key to making such a choice. If we come back to the landscape of problems sketched out on Figure 1, it should be clear that knowing where they fall on the grid influences our expectations for model and outcome variance.

Mapping of finance problems: Predictability and base rates

Positioning problems on the grid in Figure 1 is not difficult. The base rate is usually obvious: it is the observed frequency of the class of interest. Note however, that the base rate may not be stable — for example, certain financial instruments such physical commodities display deep trends in both directions, so base rates could vary significantly from one year the next, like between 30% and 70% up/down days in a given year. Nevertheless, knowing the ballpark number is often sufficient for understanding how a model will behave under different market conditions.

Estimating predictability can require some trial and error on the data, such as using cross-validation on out-of-sample/time testing. For classification problems, a good measure of predictability on test data could be the Bayes Error Rate (Hastie *et al.*, 2009), which is the lowest possible error rate achievable by the classifier on a set of data. It is analogous to irreducible error Tumer (1996). The lower the Bayes Error Rate, the higher the predictability and vice versa. For regression problems, predictability could be based on an aggregate loss calculation comparing actual and predicted values.

The more difficult prediction problems in finance lie towards the left of the predictability continuum. For example, prediction in capital markets tends to have low predictability since markets are efficient and new information is reflected quickly in prices. Predicting tomorrow's direction of the S&P 500, for example, lies towards the randomness end of the predictability spectrum. It also has a high base rate since up and down days are relatively balanced (there is a small upward bias in direction). As we discussed in the previous section, the balance in class distribution makes it possible for a learner to extract structure in data even for low levels of predictability, such as when it is as low as 5%. However, for such problems we can expect model variance to go up with complexity. In other words, a model of low complexity, such as simple linear model, will exhibit high bias and low model variance whereas a more complex model will have low bias but higher model variance. Determining the right level of complexity is the major challenge for the model designer.

Credit types of problems are also challenging but for different reasons. They usually have more predictability or "signal" since lot of data are considered in the analysis that goes into assessing credit before making loans. Most loans pay off without default. The risk is in the occasional heavy loan going into default. Because defaults do not occur too often, the base rate is low. As we noted in the results from the previous section, low base rate problems require a significant signal level for the learner to extract structure from data. Specifically, as we can see from Figures 3 and 4, a base rate of 5% requires a 30% predictability level to be able to make any minority class predictions. Similarly, a base rate of 2% requires a 40% signal and a base rate of 1% requires a 50% signal for a leaner to be able to make any minority class predictions and hence do better than random. Having such ballpark expectations should be extremely useful to data scientists building predictive models in such domains.

To wrap up, the general lesson from this analysis is that the lower the base rate, the higher the required predictability in order to be able to learn anything useful. That is why it is virtually impossible to deal with problems on the lower left part of the grid where the base rate and signal levels are both low. On the bottom right, we have problems where the signal level is high, so despite the low base rate, it is possible for a learner to identify structure in the data. For example, trade error resolution is a classic example of an easy prediction problem despite low base rates since errors occur for a handful of reasons that are possible to identify in the data.

Problems such as extraction of sentiment fall somewhere in the middle of the grid. The base rate is reasonable since there are significant numbers of good, bad, and neutral stories. Given a reasonable sized dataset, current natural language processing algorithms do a reasonable job of extracting sentiment such as "good," "bad," or "neutral" from news stories. The same is true for algorithms that extract "topics" from stories; as long as there is sufficient data, algorithms do a reasonable job of extracting the topic(s) associated with a news story.

Customer segmentation based on transaction data and social media data is usually relatively easy, so we should expect low model and outcome variance for such problems. Predicting customer attrition can be harder due to low base rates and inherent randomness. For example, with credit cards, some customers run up high levels of revolving credit and therefore generate a high level of interest fees for the card issuer, but it can be difficult to predict which ones will actually pay off their debt and which ones will go into default. This can result in a moderate level of model and outcome variance.

6 Related work

There is a considerable amount of literature on the issues of balanced and imbalanced classes in data, and in particular, the impacts of unbalanced classes on learning algorithms.

It is well known that learning algorithms find it increasingly difficult to predict the minority class, let alone predict them accurately, as their occurrence in the data decreases. The reasons for this phenomenon have been discussed extensively by Weiss and Provost (2001). If the goal is to maximize accuracy, the learner will mostly predict the majority class. However, if our goal is to predict the minority class accurately, such as diseases, defaults, or other kinds of costly errors that are crippling if they are not predicted, the challenge is to learn classifiers that are "good enough" at predicting the minority class even when the class distribution of the data is highly unbalanced and it is costly to misclassify minority-class examples Japkowicz (2000).

Research dealing with unbalanced data sets has focused mostly on modifying the class distribution of the training set, typically by making it more balanced. Two methods are commonly employed for handling class imbalance. The first is under-sampling, which eliminates examples in the majority class. The second is oversampling, which replicates examples in the minority class according to some process Breiman *et al.* (1984).

It is important to understand that neither method "creates" new information, but rather, discards information or weighs some of it more heavily. Both are problematic for different reasons. Under-sampling is arbitrary, and it is not obvious why it should improve performance in general on a test set. Oversampling is also arbitrary in that it requires a process for picking which cases will be oversampled and by how much. Since over-sampling copies minority class examples, overfitting is more likely since induced rules may perform well on training data by over-counting the replicated examples.

Not surprisingly, a lot of research has focused on intelligently removing majority class example (Kubat and Matwin, 1997). One approach is to remove majority examples that are "redundant" since such cases do not add much information. Another approach is to remove those that "border" the minority examples, assuming that they are "noisy." Chawla *et al.* (2002) combine the two methods. Instead of over-sampling by replication, they create new minority class examples by interpolating among several minority-class examples that are "close neighbors." But this method breaks down if the dimensionality of the problem is high, which is typically the case in machine learning applications, so the method is of limited practical value.

Chan and Stolfo (1998) take an iterative approach, by running bootstraps to determine the best synthetic class distribution for learning for a problem, and creating multiple training sets corresponding to the best training class distribution. However the method is somewhat arbitrary and does not generalize across problems.

More sophisticated approaches have attempted to improve classifier accuracy by considering the differences in class distributions in the training and test data by "correcting" estimates according to the ratio of minority to majority class examples in the training and testing data Weiss and Provost (2001); Elkan (2001).

Weiss and Provost (2001) provide results comparing classifiers learned from the natural class distributions to those learned from synthetic class distributions on 25 problems selected from the UCI database. The results are confounding. For 8 of the 25 datasets a synthetic distribution outperformed the natural distribution in reducing error rate. For the other 17, there is no obvious pattern. When AUC is the criterion, there is no observable pattern for roughly half the datasets. In summary, sometimes a synthetic approach to constructing class distributions works, but there is no general principle that emerges from this approach.

What explains the confounding results? Our conjecture is that previous research focusing on the base rate and correcting for it synthetically has ignored a key variable which we consider in our research: what is the predictability or "signal level" in the problem. We would expect that with very high signal problems where there is a lot of data, for example, under-sampling majority class problems might work well, but it would probably not work for problems where predictability is low, and therefore dropping data would lose vital information. We consider this a fertile area of future research, namely, how do base rates coupled with predictability impact when and how we should construct synthetic class distributions for learning.

7 Summary

Given the prevalence of machine learning in our lives, it is surprising that we lack a conceptual framework for understanding the behavior we should expect from machine learning models. In this paper, we provide the first such expectations for problems where we can estimate predictability.

Our inquiry into this problem was driven primarily by our experiences in finance, specifically in the prediction of financial markets based on machine learning models. For years, we built and observed such models in practice until their behavior on real data became apparent. Given the inherently low predictability of such problems, our emphasis was on trying to avoid over-fitting. However, it was unclear to us what this really meant, even though it was apparent that our models had difficulty predicting the minority class. The pattern that became obvious to us was that simple models were stable but they did not work, whereas complex models worked sometimes but lacked stability. This raised the question about whether the variance was due to modeling choices or the inherent randomness of the problem. On further analysis, it became apparent that it was a combination of the two.

Our initial hypothesis was that model variance increased as a problem became less predictable. Further analysis revealed the more subtle relationship between the two, which is mediated by the base rate as illustrated in the experiments.

More broadly and interestingly, our experience with prediction of financial markets revealed broader lessons that apply not only to finance but problems in general. For example, in healthcare we encounter similar problems in terms of predictability and base rate. Rare diseases are difficult to predict because there are not enough instances and the predictor variables do not provide sufficient predictability, whereas more commonly occurring diseases such as diabetes are easier to predict because of higher base rates and observations such as A1C levels, diet, weight, and

cardio measurement provide a reasonable degree of predictability.

Similarly, problems such as self-driving cars are challenging because of the low base rates of the "edge cases," which constitute the minority class and are of most interest since these are the ones associated with potential catastrophes and what systems try to learn. For example, maneuvering in a parking lot and identifying non-static objects is complex, which is why humans drive slowly in parking lots, and are on the alert for unforeseen things in general. However, it is very difficult to create an exhaustive set of unusual cases for the machine with any degree of confidence — there is always that nagging feeling of cases that have not been encountered by the autonomous vehicle to which it will not respond correctly. The challenge is therefore to create a sufficient number of edge cases. Until this happens, we will lack trust in such models.

In summary, big data offers tremendous potential for machine learning algorithms. However, instead of blindly accepting that "more data will lead to perfect models," a more nuanced and sophisticated thinking is required for us to have realistic expectations from machine learning. While we know that problems in finance are "noisier" than other types of problems such as in health or autonomous navigation, it is important to be able to quantify such difference and the implications of these differences. Our results should be useful to researchers interested in using machine learning for any domain. They provide clear expectations about the inherent uncertainty we should expect for various problems in employing machine learning-based models.

Notes

[1] For simplicity, we ignore regression problems where the objective is to predict a continuous value such as the return, but the analysis and results apply equally to regression problems as well.

[2] The original dataset has multiple classes and was converted into a binary classification problem.

[3] Note that due to the existence of greediness in the decision-tree building algorithm, it may not be able to recover the model perfectly, to the signal level of 1 is really an upper bound on predictability rather than an exact level.

References

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning*, Springer 2009, ISBN978-0387848570.

Tumer, K. (1996). *Estimating the Bayes Error Rate Through Classifier Combining.* Proceedings of the 13th International Conference on Pattern Recognition, Volume 2, 695–699.

Weiss, G. and Provost, F. (2001). "The Effect of Class Distribution on Classifier Learning: An Empirical Study," *The Effect of Class Distribution on Classifier Learning: An Empirical Study*.

Japkowicz, N. (2000). "Learning from Imbalanced Data Sets: A Comparison of Various Strategies," In: *Papers from the AAAI Workshop on Learning from Imbalanced Data Sets. Tech. rep. WS-00-05* Menlo Park, CA: AAAI Press.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. "Classification and Regression Trees," *The Wadsworth and Brooks-Cole Statistics-probability Series*.

Kubat, M. and Matwin, S. (1997). "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," In: *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179–186.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). "SMOTE: Synthetic Minority Oversampling Technique," *Journal of Artificial Intelligence Research*.

Philip, K. C. and Salvatore, J. S. (1998). "Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection," *American Association for Artificial Intelligence* (www.aaai.org).

Elkan, C. (2001). "The Foundations of Cost-Sensitive Learning," In: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc.San Francisco, CA.

Keywords: Finance; Machine Learning; Artificial Intelligence; Prediction; Capital Markets; Trust; Uncertainty