PitchBook Data, Inc.
John Gabbert Founder, CEO
Nizar Tarhuni Vice President, Institutional Research and Editorial
Daniel Cook, CFA Head of Quantitative Research

pbinstitutionalresearch@pitchbook.com

Contents

Introduction 1
Model performance evaluation 2
Historical data 4
Inputs 6
Scoring 8

PitchBook VC Exit Predictor

PitchBook is a Morningstar company providing the most comprehensive, most accurate, and hard-to-find data for professionals doing business in the private markets.

Introduction

The PitchBook VC Exit Predictor leverages machine learning and our vast database of information about VC-backed companies, financing rounds, and investors to objectively assess a startup's prospect of a successful exit. The primary component underpinning the score is a classification model that predicts the probability that a VC-backed startup will ultimately be acquired, go public, or not exit due to either failure or becoming self-sustaining. These probabilities are then used to calculate a naïve expected return of an investment in the startup's next financing round using historical returns by series derived from capitalization table data. Finally, these expected returns are normalized across the VC universe by percentile ranking. The final score for each currently VC-backed company is a number from zero to 100,[1] wherein a score of 100 represents the most attractive and zero the least attractive. This document provides methodological details of the VC Exit Predictor, including performance evaluation, the data and inputs used to train the model, the validation process, and how companies are ultimately scored.[2]

1: A company must be VC-backed, have at least two VC financing rounds, have experienced a financing event in the past six years, and have not undergone an exit event to be eligible for scoring in the PitchBook Platform.
2: Data and performance metrics were generated on January 5, 2023.

Model performance evaluation

The model achieved an accuracy rate of 67.8% on the test data. When the merger and public listing classes were combined into a single "success" category to create a binary classification problem, accuracy improved to 73.6%. While accuracy is easy to interpret, it can potentially be misleading and should be viewed in the context of the outcome distribution. A good way to achieve this is by looking at the confusion matrix, which in this case is a 3x3 matrix whose rows represent the predicted outcome and whose columns represent the true outcome. The values are normalized for the total sample size.

Normalized confusion matrix

		True outcome			
		No exit	Merger	Public listing	Total
Predicted outcome	No exit	24.2%	10.9%	0.6%	35.6%
	Merger	14.4%	40.6%	4.0%	59.0%
	Public listing	0.5%	1.8%	3.0%	5.4%
	Total	39.1%	53.3%	7.6%	100.0%

Source: PitchBook | Geography: Global

Entries along the diagonal are correct predictions, whereas off-diagonal entries are different types of errors. For example, the entry in the first row and second column (10.9%) is the percentage of observations wherein the model predicted failure and the actual outcome was a merger. Two summary metrics related to the confusion matrix that provide additional perspective are precision and recall. Precision is the accuracy given the model predicted a certain outcome, and recall is the percentage of observations of a specific outcome that the model correctly identified.

Precision and recall by outcome

	Precision	Recall	Class %
No exit	67.8%	61.8%	39.1%
Merger	68.9%	76.2%	53.3%
Public listing	56.4%	40.0%	7.6%

Source: PitchBook | Geography: Global

Precision and recall offer insights into the model's strengths and weaknesses. Similar precision metrics across the three classes indicate that the model is equally good irrespective of the predicted class. The recall metrics show more variation. Merger is the easiest class to identify, while public listing is the hardest. This is often the case for classes with low representation, and public listing recall should be viewed in the context that less than 8% of the observations are public listings.

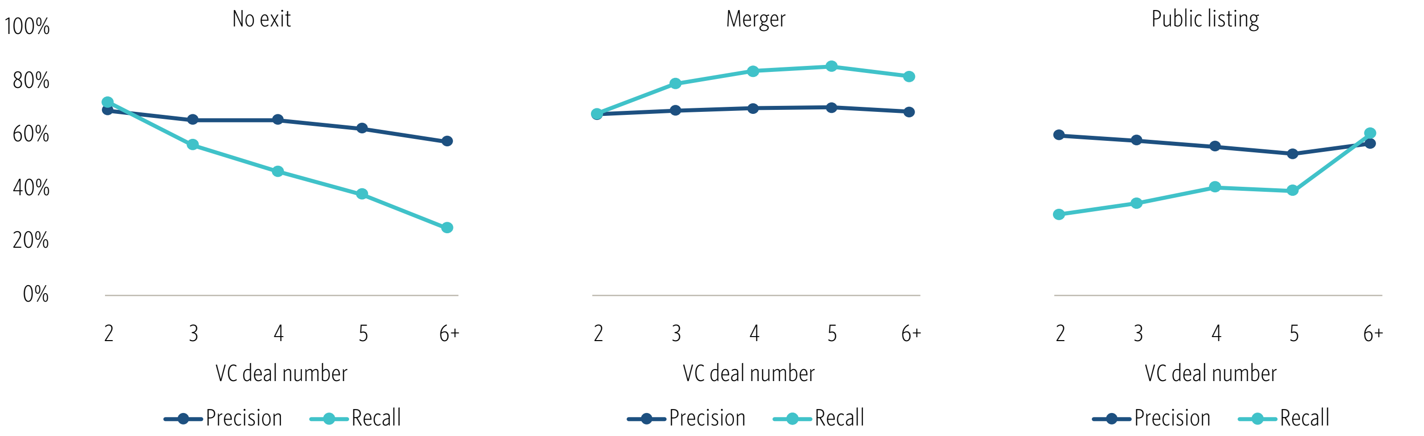
Due to differences in outcome distributions, evaluating the model by VC deal number also unveils interesting insights into its performance. From a three-class perspective, the model has similar accuracy across companies of different maturities. Binary accuracy improves as VC deal number increases, but this is driven by a change in the underlying distribution of outcomes. When viewed in the context of the percentage of successful exit, binary accuracy is best at earlier VC deal numbers because the unconditional probability of success is closer to a 50/50 proposition.

Model performance by VC deal number

	2	3	4	5	6+
Data count	5,972	3,154	1,733	960	1,175
Accuracy (three-class)	68.4%	67.7%	68.0%	67.5%	65.2%
Accuracy (binary)	71.5%	73.4%	75.5%	77.5%	79.1%
Successful exit %	52.1%	63.5%	68.5%	73.4%	77.5%

Source: PitchBook | Geography: Global

Precision and recall by class and VC deal number



Source: PitchBook | Geography: Global

The charts above provide further detail on performance by showing how precision and recall change for each class as VC deal number increases. Two main conclusions can be made: First, performance on the failure class declines over time; and second, performance on the public listing class improves over time. This is an unsurprising result—at early stages, it is difficult to determine if a company will go public many years into the future, while at later stages, it becomes rare for a company to fail after it has received significant VC investment.

A potential drawback of the model evaluation discussed thus far is that the data was not separated by time. While we have excluded any forward-looking information in the features for an individual company, the training data contained some observations that had not yet occurred with respect to some observations in the test data. Therefore, the predictions made for the test data are not a true backtest—that is, the model output could not have been replicated on the prediction date. Setting up a backtest for this analysis is challenging because it requires balancing having enough data to both train and evaluate the model. We need to go back far enough so the companies for which predictions were made have a chance to mature and exit. However, if the backtest date is too early, there will not be a large enough sample of VC-backed companies with a known outcome to adequately train the model. This is especially challenging due to the exponential growth in VC activity—most observations have come within the last three and a half years. With this trade-off in mind, we selected December 31, 2018 as the date of the backtest, which led to approximately 32,000 observations to train the model and 13,000 observations to evaluate its performance. The model had a three-class accuracy of 72.6% and a binary accuracy of 76.6%. The normalized confusion matrix summarizing the performance is shown below.

Normalized confusion matrix for model backtest

		True outcome			
		No exit	Merger	Public listing	Total
Predicted outcome	No exit	41.6%	6.6%	0.2%	48.3%
	Merger	15.2%	27.8%	1.0%	43.9%
	Public listing	1.4%	3.1%	3.2%	7.7%
	Total	58.2%	37.5%	4.3%	100.0%

Source: PitchBook | Geography: Global

The model performed particularly well on the no exit class, with precision and recall of 86.1% and 71.4%, respectively. Relative to the prior results, the model performed worse on the merger and public listing classes in terms of precision, but better in terms of recall. The backtest performance is not without its caveats, however. These caveats arise because the outcome for all companies is not realized, as only a fraction of the companies that the model made predictions for have a known exit at the time of this writing. Of the more than 30,000 companies that were eligible for prediction on the date of the backtest, around 40% have a known exit. Because the set of companies with a known exit inherently depends on time, it is not a random sample and is thus subject to bias. We found that companies that were predicted to fail had a higher likelihood of having a known outcome.

Historical data

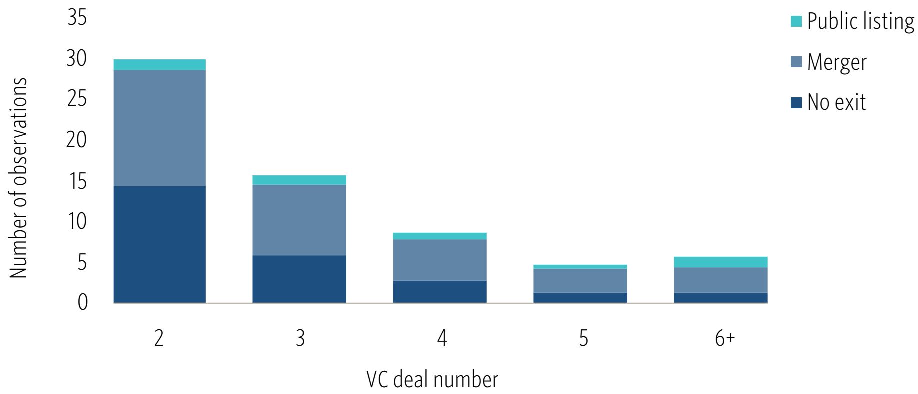
Individual data observations used to train and evaluate the model are associated with VC financing rounds, while inclusion is established at the company level. We included companies that had raised at least two rounds of VC financing (including angel and seed rounds) and are no longer VC-backed, which means they have undergone a merger or public listing, filed for bankruptcy, ceased business operations, or become self-sustaining. Because many startup failures are undisclosed, a company that has not received a VC financing round in more than six years was deemed to have failed or become self-sustaining, which was determined by analyzing the empirical distribution of time between VC rounds. The inclusion criteria resulted in over 64,000 observations from 31,000 distinct companies in the final dataset. The table and plot below provide additional detail on the data in terms of the outcome distribution.

Data distribution by outcome

	Data count	Overall %
No exit	25,523	39.5%
Merger	33,987	52.6%
Public listing	5,131	7.9%
Total	64,641	100.0%

Source: PitchBook | Geography: Global

Data distribution (thousands) by VC deal number and outcome



Source: PitchBook | Geography: Global

Theoretically, model inputs could be generated daily for a company between its second VC financing date and exit date. Not only is this unreasonable from a computational perspective, but it would also result in highly correlated observations, given that many of the features would not change from one day to the next. Significant feature updates mainly occur after a financing event. Therefore, we generated one observation per VC financing round for each qualifying company.[3] The prediction date for each observation was determined by randomly sampling from a uniform distribution in the interval from the close date of the current round to the close date of the next event (subsequent VC round or exit). The prediction date for each observation dictates what information is included in the input—only data that was known at the time of the prediction is allowed in order to avoid look-ahead bias. Randomly sampling the prediction date, as opposed to using the close date of the current round, enables the model to learn how time affects outcomes. For example,

3: The data frequency of observations used for model training and evaluation differ from that used for model inference. The outcome probabilities and scores shown on the PitchBook Platform are updated daily.

a company that raised its last round one year ago has a better chance of successfully exiting than one that has not raised a round in four years, all else equal. In addition, this matches the structure of the data that the model will be used on for inference (current eligible VC-backed companies), wherein the time from last close date will differ across companies.

Inputs

The inputs, or features, to the machine learning classification model were compiled from the extensive amount of information on each startup's PitchBook profile. In total, each observation has 34 features, which can be categorized into three main groups: company, financing, and investors.

Company-level inputs can be further broken down into static and point-in-time information. Static features are basic, unchanging descriptive data points about a company, including industry/vertical, geographic location, and number of founders. Point-in-time features, on the other hand, are company attributes whose value depends on when a prediction is made. This is a broad category containing data from stage of business (for example, product development, generating revenue, and profitable, among others) to patents. Additional point-in-time inputs include number of employees, company age, acquisitions, and related news articles. Inputs in the financing category comprise data from current and past financing rounds with a focus on VC deals. Key variables consist of VC round number, stock type, close date, and deal size.[4]

Engineering features from investor data, particularly data related to individual investor entities, present a challenge due to their high dimensionality. This analysis contains nearly 10,000 distinct investors, and only a small fraction invest in each company. Rather than treat investors as a sparse and high-dimensional categorical feature, we developed a method to rank investors based on their importance and experience within the VC universe. This ranking method relies on the well-known hypothesis that influential VC investors frequently work together by investing in the same companies; this is often compared to an exclusive social club. To capture this dynamic, we model VC investors as a social network wherein two entities are connected if they have invested in the same company. The connections, or edges, are then weighted by the number of distinct co-invested companies between pairs of investors. To quantify the idea that investors should be highly ranked if they have both VC investing experience and are connected with other experienced VC investors, we calculated the eigenvector centrality of the investor network.[5] In addition to the investor ranking, other inputs in this category include average capital invested per distinct investor, investor counts, counts by type other than VC (such as CVC), follow-on counts, frequency of a lead investor, and geographic location.

Modeling

The first step in the modeling process is to split the data into training and test sets so that the model is trained and evaluated on mutually exclusive samples. Extra care

4: Due to better data coverage, deal size is used as a proxy for valuation.
5: The concept of eigenvector centrality was famously used in Google's PageRank algorithm. For more information on network centrality, see: "Network Centrality: An Introduction," arXiv, Francisco Aparecido Rodrigues, January 22, 2019.

needs to be given to this step because there can be multiple observations with the same outcome for the same company, which can lead to information leakage between the training and test data.[6] To avoid this pitfall, we partitioned the data at the company level such that all of a company's observations were either in the training or test set. Therefore, when the model made predictions on the test set, it had no prior information on the outcomes of the companies. We performed stratified random sampling to assign each company to the training or test set with an 80/20 split, which results in around 48,000 and 12,000 observations in the training and test sets, respectively.

The presence of unbalanced outcomes is another aspect of the modeling process that deserves attention. Unbalanced class distributions in supervised machine learning classification can cause the model to overemphasize the majority class during training, thus potentially leading to biased predictions in favor of the majority class. The class distribution in this case is imbalanced in favor of mergers and failures, while public listings make up less than 10% of the data. Startups that go public are often the most lucrative for investors and, therefore, are important for the model to perform well on. To mitigate the impact of unbalanced classes, we implemented an oversampling method known as the Synthetic Minority Oversampling Technique (SMOTE),[7] which creates synthetic observations of the minority class(es). Synthetic observations are created by randomly sampling along the hyperplanes (in the case of more than two dimensions) connecting all the *k* minority class' nearest neighbors, wherein *k* is a hyperparameter. The synthetic observations are therefore logical perturbations from the original data. These observations are strictly used during model training and are not considered during evaluation. The oversampling process effectively gives each class an equal weight on the loss function during training.

The specific algorithm we employed is known as XGBoost,[8] a gradient-boosted classification tree model. Since its introduction, this algorithm has produced state-of-the-art performance on many traditional machine learning tasks with two-dimensional feature inputs. For this task, it outperformed a multinomial linear model, a multilayer perceptron (MLP) neural network, and a recurrent neural network with long short-term memory (LSTM) layers wherein financing rounds were treated as sequences. In addition, we chose XGBoost due to its flexibility in handling outliers and missing data, ability to represent complex nonlinear functions, robustness to data preprocessing, and fast training times.

The model's hyperparameters were tuned using five-fold cross-validation with both grid search and Bayesian optimization.[9] Cross validation is a data splitting process used to select the "best" set of hyperparameters wherein the training data is split multiple times—in this case, five—to create additional validation sets. Each fold is used as a validation set once, while all other folds are combined to train the model. Just like the training and test sets, observations were assigned to each fold at the company level to avoid information leakage.

6: Information leakage occurs when the test set contains information from the training set that can cause overfitting and optimistic performance evaluation on the test set. This particular form of information leakage is known as "identity confounding" because the model learns identities (that is, companies) as well as features.
7: "SMOTE: Synthetic Minority Over-Sampling Technique," Journal of Artificial Intelligence Research, N. V. Chawla, et al., June 1, 2002.
8: "XGBoost: A Scalable Tree Boosting System," arXiv, Tianqi Chen and Carlos Guestrin, June 10, 2016.
9: Hyperparameters are components of the model that must be specified before training and cannot be learned directly.

Scoring

The scoring process maps the outcome probabilities from the classification model to a naïve expected return from the perspective of an investor in a company's next VC financing round based on historical returns by series derived from capitalization table information. Scoring serves two main benefits: First, it creates a single value for each company, which is necessary for the final rankings; and second, it quantifies the benefit of investing early in successful startups and/or exiting them via the public markets.

The expected return for an individual startup is a weighted geometric average of the historical returns based on the upcoming series of its next VC financing round. For example, if a startup had last raised a Series B, the relevant returns would be for Series C investments. The weights are taken as the probability of each exit outcome from the classification model. The tables below show average annualized startup returns by series and type as well as the average holding period.[10] For simplicity, we assume that a failure results in a total loss at all stages.

Average return by series

	Merger	Public listing
Series A	36.7%	47.8%
Series B	31.0%	37.9%
Series C	28.0%	34.4%
Series D+	20.0%	30.0%

Source: PitchBook | Geography: Global

Average holding periods (years)

	Merger	Public listing
Series A	5.34	5.89
Series B	4.67	4.34
Series C	4.40	3.74
Series D+	3.69	3.04

Source: PitchBook | Geography: Global

For example, consider a startup that recently raised a Series B with no exit, merger, and public listing exit probabilities of 50%, 30%, and 20%, respectively. The annualized geometric expected return would be calculated as follows:

$$\bar{r} = (1.28^{4.40} \times 0.3 + 1.34^{3.74} \times 0.2)^{\left(\frac{1}{4.40 \times 0.6 + 3.74 \times 0.4}\right)} - 1 = 10.0\%$$

Finally, the return figures are normalized as a percentile ranking across all eligible VC-backed companies. A percentile ranking of 100 represents the most attractive company, while a ranking of 0 represents the least attractive.

10: Holding periods of less than one year are not annualized. In addition, outlier returns of more than 350% are excluded from the average calculations.