Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 7 (2021) 94–114
http://www.keaipublishing.com/en/journals/jfds/

CapitalVX: A machine learning model for startup selection and exit prediction

Greg Ross [a,*], Sanjiv Das [b], Daniel Sciro [a], Hussain Raza [b]

[a] *Venhound Inc., DE, USA*
[b] *Santa Clara University, CA, USA*

Received 23 December 2020; revised 24 March 2021; accepted 28 April 2021
Available online 11 May 2021

Abstract

Using a big data set of venture capital financing and related startup firms from Crunchbase, this paper develops a machine-learning model called CapitalVX (for “Capital Venture eXchange”) to predict the outcomes for startups, i.e., whether they will exit successfully through an IPO or acquisition, fail, or remain private. Using a large feature set, the out-of-sample accuracy of predictions on startup outcomes and follow-on funding is 80−89%. This research suggests that VC/PE firms may be able to benefit from using machine learning to screen potential investments using publicly available information, diverting this time instead into mentoring and monitoring the investments they make.

Keywords: Machine learning; Ensemble models; Exit prediction; Funding prediction; Private equity; Venture capital; Crunchbase; USPTO; Model evaluation

1. Introduction

Startups have a number of potential destinations, the most desirable of which is either a successful acquisition from a larger company or an IPO. Along the way, however, it is quite possible for a company to receive follow-on funding or falter and fail. We demonstrate that machine learning can be used to calculate the probability of each of these outcomes with high levels of confidence, using publicly available data on Crunchbase[a] and the US Patent Office (USPTO), thereby integrating data on companies, people, and patents. Two types of predictions are made with the machine learning models developed in this paper. One, predicting whether a startup will exit or not, assessed as a probability of IPO, acquisition, failure (in a three-way model), or additionally remain private (in a four-way model). Two, whether a startup venture will gain access to additional funding. The accuracy achieved by the machine learning classifier is around 90% for a three-way classification model (80% for a four-way model). The models are

* Corresponding author.
E-mail addresses: gregxross@gmail.com (G. Ross), srdas@scu.edu (S. Das), danielsciro@venhound.com (D. Sciro).
Peer review under responsibility of China Science Publishing & Media Ltd.
[a] www.crunchbase.com.

https://doi.org/10.1016/j.jfds.2021.04.001

implemented on a cloud platform (AWS) and provide real-time assessment of the future outcomes for startups. In addition, latest methods in the explainability area are used to explain the predictions made by the models.

A variety of both quantitative and qualitative factors influence a startup's success, and we attempt to create a holistic profile of each company in our database, combining multifaceted types of data. By pulling data from sources covering over a million firms, we are able to obtain intellectual property ownership (USPTO) as well as investment history and general business information (Crunchbase). Each data set provides a host of features such as the number of patents (USPTO), number of funding rounds and members of the executive team (Crunchbase). These features in turn are encoded in a way that allows our models to consume them and learn from their properties.

The data are used to train a number of models that each serve their own purpose, and together provide for a successful *ensemble* approach to predicting outcomes. We implemented a combination of Deep Learning, XGBoost, Random Forests, and K-Nearest Neighbors for the purposes of this study. Machine learning set up usually creates a pipeline that can be repeated when the model needs to be updated. More notably, as the performance of businesses and industries change over time, each model may be regularly fine-tuned after being fed new data from the aforementioned sources. The general process in the pipeline is to first perform exploratory data analysis, extract the features, retrain the models, and then evaluate our predictions based on known events. This model setup supports staying current with industry trends and provides consistently accurate predictions.

The model may be used as a tool for investors in private companies to select investments and to justify investment decisions, especially given the size and riskiness of venture investments. To address the concerns of explainability and transparency, we use Local Interpretable Model-Agnostic Explanations LIME,[24] and SHAP[21] to justify the predictions. This allows for a hyper-localized analysis of each prediction that can be applied to any model used and, in our case, will identify specific features of an organization that influenced the prediction.

The rest of the paper proceeds as follows. In Section 2, we briefly discuss the related literature. In Section 3, our experimental method of model development is described. Section 4 introduces the data used, and Section 5 describes the research questions and the approach taken to answer these. The results are presented in Section 6, and section 7 discusses explainability of the models with respect to features. Concluding discussion is in Section 8.

2. Previous research

There is a growing literature on analyzing venture capital investments using alternate data or methodologies, such as machine learning. Broadly put, these approaches extend and improve on traditional econometric approaches. The literature may be broken down into two broad strands: (i) identifying successful investors, and (ii) identifying successful startup investments. There are of course, several taxonomies for this well-researched area, such as the excellent survey by Rin et al. (2013).[25]

In work associated with identifying successful investors, graph theory has become an important tool. One such approach relies on investor networks, and work by Glupker et al. (2019)[16] has shown that network position determines the success rate of investors. The paper shows that it is in fact easier to predict unsuccessful investors. The study offers a two-step approach, cut by industry first, followed by community construction within industry, i.e., focus on the industry of the startup followed by the use of a machine-learning model. This combines financial data with graph-theoretic ideas and machine learning algorithms.

Gupta et al. (2015)[17] developed InvestorRank, a method for identifying successful investors based on position in a network, such as being close to an exemplary investor or super-angel. The result of the research shows potential in discovering investors who will become successful. InvestorRank flags investors who follow a general trend of improvement when compared to their preceding snapshot based on a threshold. Bubna et al. (2020)[7] analyze a VC network to uncover communities, i.e., small groups of VCs who tend to frequently work together. They show that startups funded by community VCs (as opposed to non-community VCs) tend to have higher probabilities of successful exits as well as faster exits. Geographical distance is a determinant in VCs working together and therefore is an indirect determinant of startup success, see Sorenson and Stuart (2001)[28]. Similarly, Adcock et al. (2012)[1] analyze a bipartite investment network of investors and investees with links based on investments between them. Personal investors evidence the highest average clustering, tech companies the lowest, indicating that they choose to acquire small firms rather than invest in them.

Syndication of VC investments is also a driver of better performance by startups, as evidenced in the epochal paper by Brander et al. (2002).[6] Their empirical analysis examines whether superior performance of syndicated startups comes from better venture selection or from monitoring post-selection, finding that the latter has more impact. Das et al. (2011)[10] show that the selection effect positively impacts exit returns, whereas monitoring leads to higher probability of exits, and faster exits. Bernstein et al. (2016)[3] show that VCs who are connected to their startups via close physical proximity (direct flights) are able to tightly monitor their investments and it leads to better outcomes.

The other side of the coin from identifying successful investors is predicting which startups will see a positive exit. Antretter et al. (2019)[2] analyze the concept of "online legitimacy" and demonstrate the power of machine-learning, by using Twitter as a data source, to distinguish between those ventures which are bound to fail and those which are not. Xu et al. (2017)[33] compare their results against the Crunchbase database and aim to narrow the list of portfolio companies that a venture capital investor may consider, thereby offering results that are similar to the motivations of this paper. Srinivasan et al. (2014)[30] provide further evidence that potential high-value investees can be identified through an analysis of company features.

In contrast, Dellermann et al. (2017)[13] opt for the approach of combining machine learning with the traditional human approach in what is known as a hybrid intelligence method. Their observation is that humans fill in the gaps where machines fail, namely when it comes to "soft" information and navigating moments of "unknowable risk".

Krishna et al. (2016)[20] performs research similar to ours where funding and business data is pulled from sources such as Crunchbase and Tech Crunch while a combination of Random Forest, Bayesian Classifiers and other models are used to predict company outcomes. However, the key difference is that our analysis aims to enable investors to identify the next unicorn, whereas Krishna et al. (2016)[20] attempts to assist private entities in securing future funding, as do Biesinger et al. (2020).[5] Sharchilev et al. (2018)[27] use Crunchbase and web-based information to achieve good accuracy in predicting follow-on funding, and we get similar levels of accuracy using Crunchbase data as well, though with different models. For exits via mergers and acquisitions, an excellent modeling analysis using Crunchbase data is undertaken in Xiang et al. (2012)[32] who achieve metrics similar to ours. Overall, in contrast to the previous literature, this paper achieves high levels of accuracy on a much more extensive dataset, using data from all stages of the startup cycle. We also provide feature importance and have built the algorithm into a web system for exit prediction, as discussed later.

3. Method

An experimental approach is taken where we determine which class a venture-funded company is predicted to be in given its current state as represented by its features. In a broad sense the class a company can be in is either "successful" or "unsuccessful" depending on whether it exits the venture-capital stage successfully or not. Given success can mean different things to companies and investors we define success as a (i) successful exit or (ii) follow-on funding. This duality requires two model ensembles for classification. Each ensemble is evaluated independently of the other while feature selection, extraction and model evaluation is continued iteratively to maximize accuracy, precision, and recall.

The process of training the models begins by first identifying the data sources, namely a Crunchbase snapshot taken in April 2020 (comprising Crunchbase entries until that date) and a snapshot of all patents from the USPTO at that time. These two sets together offer coverage of a large number of companies distributed widely across location, sector, and industry. The snapshot dates were selected because of their recency and comprise a rich set of observations. Next, features are derived from the data in a two-step process. In the first step the data sets are converted into a relational database schema that is subsequently transformed using SQL to derive specific intermediate features. By taking advantage of a relational database management system (RDBMS) representation of the data, we can easily identify relationships between entities and readily obtain features based upon those. This set of features forms the basis from which two sets of more refined features are derived. One set is specific to exit classification and the other for determining follow-on funding. In the second step we take the original raw features along with the new, intermediate ones and use the flexibility and expressibility of Python to derive additional features. Fig. 1 illustrates this process at a high level.[2]

In the ensuing sections, we will detail this further by delving into the construction of the dataset and our feature engineering, and the techniques applied for machine learning and evaluation of all resultant models. We investigate

[2] The architecture and models are part of a fully functional web-based system (www.venhound.com).

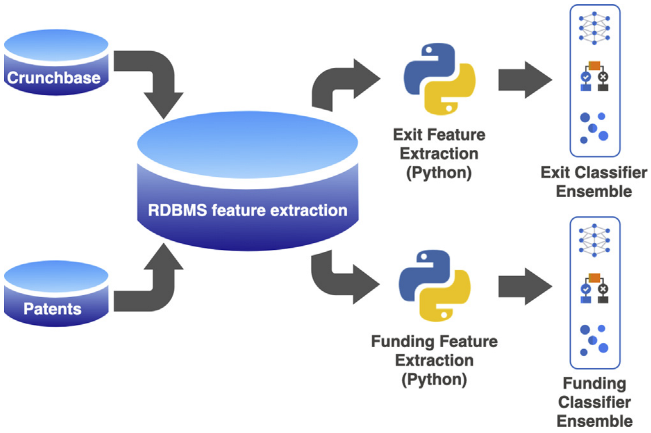

Crunchbase
Patents
RDBMS feature extraction
Exit Feature Extraction (Python)
Exit Classifier Ensemble
Funding Feature Extraction (Python)
Funding Classifier Ensemble

Fig. 1. Overview of data consolidation, feature extraction and modeling pipeline.

two classes for the funding models: follow-on funding or none; three classes for the exit models: IPO, acquisition and failure, and also study the effects of the addition of a fourth, *private* class for the exit models. Finally, we build in explainability of the model's predictions using the popular Shapley values approach.

4. Data

Data used in this study are sourced from Crunchbase, a crowd-sourced database and website listing around one million companies, and the United States Patent and Trademark Office (USPTO). For the CapitalVX system, the feeds are regularly refreshed to subsequently retrain and ensure the models are up to date. We explore distributions, cardinality, dimensionality, count missing values, impute and derive features and filter out unsuitable and badly formatted entries each time the data are refreshed.

The majority of the features that the models use as input are derived from Crunchbase data. The set described in this paper were retrieved in April 2020 and include details of 1,000,886 companies, 141,430 investor companies with 1,006,911 people across 150 countries.

Fields include the number of funding rounds, mail and email address, number, stage and date of funding rounds as well as the amount invested along with the job titles of employees and founders. The data types range from unstructured free text, nominal, ordinal, categorical and numerical values and so various means of encoding, transformation and conditioning is required before they can be presented to the models. While it is possible to represent a company by a time series based upon the progression of funding rounds, the authors instead opted to use a single flat record to represent each firm. This is achieved by aggregating funding information into features for average time between rounds, total funding, number of rounds and last funding round. Representing each company by a single data point allows for simpler comparative analyses and evaluation during modeling. This also ensures (i) that exit predictions are for a company and not for individual rounds and (ii) that when creating train-test splits for model development, the information remains cross-sectional and future rounds of a company are not used for prediction of outcomes related to earlier rounds.

There are two steps in processing the data: the first involves importing to a relational database before using SQL to transform and derive new features; the second utilizes the Python programming language for fine-grained conditioning and additional derivation. The steps are outlined below and are applied before training the exit models as well as for the follow-on funding modeling, i.e., the same set of features are used for both ensembles.

4.1. SQL-based feature selection and engineering

The data dump comes pre-split into CSV files, each representing a relational table that can be imported into MySQL. Once there, companies that are VCs and other investment institutions are excluded to constrain the analysis to organizations that are potential targets of such firms. This leaves 942,605 companies, of which 18,419 are stated as public, 94,225 are said to have been acquired, 33,298 are closed and the remaining 796,663 are assumed to be private and still operating. The next step is to transform fields such as string-formatted dates into numbers and derive new features such as the average time between funding rounds and the number of degrees from the most prestigious schools—[31] shows that when VCs and entrepreneurs share an academic background, not at Ivy League schools, then there is an increased likelihood of funding, so, conversely this top-degrees feature might be a good indicator for failure.

The original and new features are then compiled into a single features table used as input into the next stage of the feature engineering process. A subset of the 370 features derived this way is shown in Table 1. Descriptive statistics are shown in the Appendix.

For the purpose of training the models to predict follow-on funding, an additional Crunchbase snapshot was taken. The idea is to take an older set and see which companies progressed to another funding round in the more recent set. In this case the older set was taken from July 2018 to provide a near two-year window for progression, and to completely eliminate any chance of data-snooping. The funding-rounds tables were then compared and companies that received additional funding were coded with a dummy variable value 1 in the features table for that field and 0, otherwise.

4.2. Feature engineering

The data from the features table are further processed to generate new variables of interest. The first new feature to be derived is the count of missing fields for each company. This is used as a feature as well as an indicator to help balance the classes when training the models. This is an interesting aspect of this kind of public data, and keeping track of these quantities adds an unusual new feature in our dataset. Fig. 2 shows the distribution of missing features across the three exit classes. The mean number of empty fields for companies that have been acquired is 36.82; the mean number for public companies is 30.89, and 37.57 for failed companies. Subsequently, some of the missing fields are replaced by sensible defaults such as zero in the absence of an amount for total funding. Handling missing values in a public dataset is often tricky, and, despite this being a consistent issue with Crunchbase data, we are able to design a classification model of high accuracy.

Work by Jalbert et al. (2012)[19] shows that firms that are led by female CEOs experience higher growth and greater returns. Contrary to this, however, Davis et al. (2009)[11] found that gender diversity in entrepreneurial teams reduced overall contribution of the individuals. Either way, this motivated the derivation of some gender-specific features including the number and ratio of female founders and employees. The expectation is that those companies with a higher proportion of female employees, founders and c-suite personnel might have a positive effect on a favorable exit.

Table 1
Database features. Note that there are many other features that we use and are readily available from the data. See Appendix A for a description of the full set of features used for training the models.

Feature	Description
avg_time_between_funding	average time between funding rounds
num_female_founders	number of female founders
num_male_founders	number of male founders
num_patents	number of patents
num_degrees	number of employee degrees
num_top_degrees	number of degrees from top 50 schools
num_acquisitions	number of acquisitions
investor_types	type of investors (VC, Angel etc.)
num_events	number of company events
state	HQ state code
country_code	HQ country code
category	industry category
description_length	length of the company description
has_domain	whether the company has a web domain
has_email	whether the company has specified an email address
has_linkedin	whether a Linked-in URL is specified
has_facebook	whether a Facebook URL is specified
has_twitter	whether a Twitter account is specified

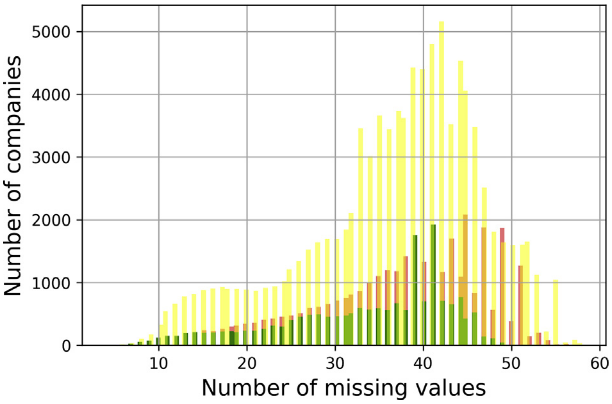


Fig. 2. Distribution of missing values across public, acquired and failed companies. Red: failed. Yellow: acquired. Green: public.

Whether a company has VC backing is also taken into account. It has been shown by Bertoni et al. (2011)[4] that VC-backed new-technology firms exhibit higher sales and employment growth. According to Sørensen (2007)[29]; such firms are more likely to go public. Also, Puri and Zarutskie (2012)[23] found that VC-supported firms grow more rapidly and are less likely to fail within their first five years of operation. For these reasons a boolean feature indicating whether a company has one or more VC investors is included.

There are several categorical features in the Crunchbase data. These include country code, state code, employee college degree types and company industry. One approach to transform these into numerical features is one-hot encoding. However, the cardinality of the values is too high when a binary column is added for each distinct category and therefore target-encoding was employed where the categorical value is replaced with the mean of the target variable. For each distinct value in a categorical field, $x \in C$ where C is the set of all values assigned to any company, we take the sum of the exit class y (e.g. $y_{ipo} = 1$ when a company is public and zero, otherwise) divided by the total number of times that x has been assigned. This is the average number of observations where that categorical value is applied, where a company is public, acquired or failed. In cases where a categorical field can take on multiple values, e.g., company industry, then a weighted average is used. For example, a company can specialize in “AI” and its application in “medicine” and therefore it has two categories applied for that field.

Finally, after feature derivation and conversion to floating-point numbers, the data are normalized. It is worth noting that many of the Crunchbase fields were dropped from the feature set and will be examined in future work. Examples are the description text, address and phone number etc. after dropping these values the number of remaining features is 76, which is neither small nor large, and is not a sparse feature set.

4.3. Patent data from the USPTO

Cockburn and Macgarvie (2009)[9] show that firms that publish patents within a market where there is an abundance of patent activity, are more likely to go public or receive other funding. Patents help firms demonstrate to potential investors that they are differentiated from their competitors. Mann and Sager (2007)[22] examined the relationship between software startup patents and progress through the venture capital cycle. They found that there is a strong correlation between patent activity and company performance in terms of funding and longevity. These observations inspired the collection of an additional data set consisting of $\sim 7,000,000$ patents, dating back to 1976 from the USPTO. The goal was to obtain a count of patents for each company in the Crunchbase data. The challenge here lies in matching the companies between the two sets. The names are often different across the two, depending upon whether the legal name or an alias is used. Thus, fuzzy matching was employed to obtain the counts feature.

5. Research questions and analysis

This paper is motivated by the question as to whether machine learning is effective in using public data to accurately predict the fortunes of startup companies. Alternatively, this study may be interpreted as assessing whether there is sufficient signal in self-reported data to predict the success and failure of startups.

We consider the following questions. First, can a multi-category classification model predict if a startup firm will fail, or exit successfully, either through an IPO or an acquisition? We also consider the auxiliary question as to predicting a fourth class, that of a firm remaining private. If a machine learning model shows a high level of accuracy on this task, then it means that venture capital firms may use the model to screen startups using Crunchbase data to determine whether they have a high probability of success. This will augment and also simplify the mostly manual process of screening possible venture investments.

A second question is whether machine learning models can effectively predict if a startup firm will receive follow-on funding. This is a common measure of performance of a startup as it progresses through its life cycle. If a machine learning model performs this task successfully, it is of considerable use to venture capitalists who wish to evaluate the companies they have invested in. It is also a tool that may be used by venture capitalists who may invest more in companies that the model predicts are apt for follow-on funding.

A third question to be answered is: What models are appropriate for such machine learning tasks? Crunchbase data poses interesting issues such as inconsistent reporting, missing values in many quantities, and changes in the reporting format over time. Various choices had to be made in feature engineering, already described earlier. In addition, we assess different kinds of models, such as feed-forward neural networks, Random Forest models, XGBoost models, and ensembles of the models. Many hyperparameter choices have to be made over each of these models and we assess how the performance of machine learning on the first two questions varies across all the different model combinations. We also wish to assess the robustness of the models across different startup stages. We turn to these various experiments next.

6. Results

Two machine learning ensembles (dataset plus models), one for exit and one for follow-on funding, were configured and trained. This section provides details of the hyperparameters selected and the performance metrics of the individual models (base estimators) as well as the combined outputs.

6.1. Training split and class balance

The data represent 942,605 companies, of which 18,419 are stated as public, 94,225 are said to have been acquired, 33,298 are closed and the remaining 796,663 are assumed to be private and still operating. The goal is to train the exit models on the public (IPO-ed), acquired, and failed companies, and likewise train the follow-on funding models so that the likely outcome of the private companies can be determined.

Since the Crunchbase data are crowd-sourced there are many companies with very little information. This is particularly apparent with larger, public companies. This can perhaps be explained by the assumption that since these companies are no longer small and struggling then there is less motivation for founders to update the data. Regardless, in order to avoid biasing the models towards associating less data with IPO or acquisition, many of those sparse entries are removed. This also serves to help balance the classes. The threshold for filtering out companies with missing values is presented in Equation (1).

$$threshold = \mu_m - \frac{\sigma_m}{\alpha} \tag{1}$$

where μ_m and σ_m represent the mean and standard deviation of the number of missing values for companies within a class, and α represents a small class-specific constant. For the exit models, α is set to 4 for the IPO class and 1 for the acquired class, while for the funding models it was 2 and 3 for the no-follow-on and follow-on classes respectively. Note that the threshold was not applied to the failure exit class since the higher count of missing fields is assumed to be a relevant signal for failure. It was found that accuracy of the ensemble dropped to 80% (from 89%) when applying the threshold with $\alpha = 1$ to closed companies. After applying the thresholds to public and acquired firms, there remain

6877 public companies and 15,127 acquired companies. Failed companies were randomly sampled so that their number matched the number of acquired companies. Note that less than one third of the companies are public. Oversampling (using SMOTE) was employed to balance the numbers but this was found to have a detrimental effect on subsequent modeling and therefore the current class balance was kept for the exit analysis.

The same approach was applied to the data for the follow-on funding models. After filtering, there remained 6157 companies that progressed from one funding round to another. That leaves 180,666 companies that had an initial funding round but did not subsequently obtain more funding within the period between the two Crunchbase snapshots (July 2018 to April 2020). To balance the classes a random selection of the non-progressed companies, equal to the number of progressed companies, was taken. Note that if no seed is specified to the random number generator, then the sample changes each time the models are trained. It was found that when this is the case the results do not vary significantly.

For both the exit and follow-on funding analyses an 80/20 training/validation split was taken. All results reported in the following subsections are obtained from observing the models' performance on the held-out 20% validation set.

6.2. Multilayer Perceptron (MLP)

Multilayer Perceptron models are feed-forward artificial neural networks and are commonly referred to as deep neural networks when they contain multiple hidden layers. The combination of many binary classifiers (perceptrons) within the network ensures both, supervised and unsupervised learning capacity that makes them very flexible in terms of applicability of many features and subsequent classification power. That combined with backpropagation of errors greatly speeds up the minimizing of loss and therefore makes the model feasible for data of high cardinality and high dimensionality.

We experimented with many configurations, tweaking the number of hidden layers, number of neurons in each layer, and regularization. The best results were obtained from a relatively simple network comprised of five hidden layers with 32 neurons in each, interspersed with alternating dropout layers with dropout rate set to 0.2. For efficiency, the rectified linear activation function (ReLU) was used for every layer with the exception of the last. Given the goal is to classify every private company into one of three classes with a specific probability, as in the exit case, this is deemed a multi-label problem and therefore the softmax activation function was selected for the last layer and the chosen loss function of choice was sparse categorical cross-entropy.

Experimentation with MLP models of lesser complexity produced increased loss while more complex models showed signs of overfitting.

Figure 3 shows that the model begins to overfit the training data set, with respect to the validation set, at around the 40th epoch and so this was deemed a good stopping point. Table 2 shows the overall accuracy of the model along with the precision and recall over each exit class. The classification model achieves a high level of accuracy of 88% across three classes. Precision and recall are highest when predicting failures and slightly lower for successful exits. Likewise, Table 3 shows the metrics for the follow-on funding MLP model, where an accuracy of 80% is achieved.

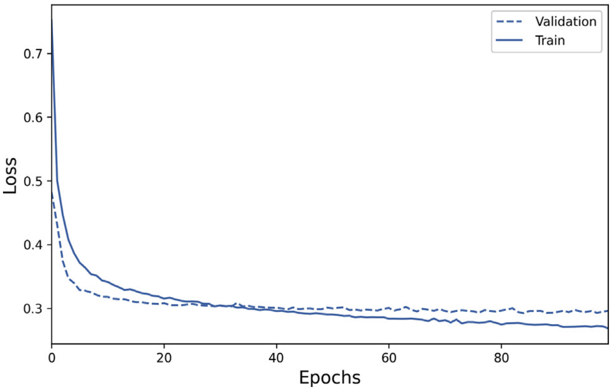


Fig. 3. Loss function value vs epoch. We see that the drop in loss is similar for training and validation datasets. Overfitting is minor but the plot is used for early stopping at epoch 40.

Table 2
MLP Exit model metrics.

Accuracy	0.880	
Class	Precision	Recall
IPO	0.811	0.703
Acquisition	0.768	0.858
Fail	0.952	0.929

These results suggest that venture capitalists may use such a model to screen firms they are interested in investing in and once they have invested in them, they can also assess the chances of these firms getting a follow-on investment. Machine learning may therefore offer important efficiencies to venture capital firms thereby making human-in-the-loop processes more effective.

6.3. Random Forest

The next model selected for the classifier ensemble was Random Forest. (Ho, 1995).[18] This itself, is an ensemble method where classifications from multiple decision trees are combined to produce a more robust classifier. The best parameters, 100 estimators (classification trees) with a max-depth of 200, were found through a grid search. Tables 4 and 5 show the results of the exit and funding models. These results are similar to those seen for the MLP model, with a slight gain in accuracy for the follow-on funding model, where the accuracy has increased to 84%.

6.4. XGBoost

Another ensemble method, XGBoost, developed by Chen and Guestrin (2016)[8] is similar to Random Forests in that it utilizes multiple decision trees. However, XGBoost employs gradient boosting whereby new tree models are fitted to correct the errors of previous tree models (See Friedman, 2002).[14] Tables 6 and 7 show the model results after fine-tuning the hyperparameters. These results are almost the same as those achieved by the Random Forest method.

6.5. Ensemble performance

Tables 8 and 9 show the overall accuracy as well as the per-class precision and recall of the ensemble models where the classification output was derived from hard voting over the individual models. Hard voting simply takes the class with the majority vote. Soft voting (taking the argmax of the sum of probabilities) and stacking (using a metamodel trained on the outputs of the base estimators) were also investigated and, while there was little difference in the outputs, it was found that hard voting provided a slightly higher lift over most metrics for follow-on funding. See Table 10 for ease of comparison of the metrics across all models and classes.

6.6. K-Nearest Neighbors

We also attempted to add a K-Nearest Neighbors (KNN), SVM with a radial basis function kernel, as well as a logistic regression estimator to each ensemble but this was detrimental to the results for the exit ensemble and the follow-on funding ensemble. The exit ensemble performance was dominated by the XGBoost estimator with little lift achieved from any of the voting or stacking methods. Thus XGBoost alone is an adequate means to estimate company exit.

6.7. Private companies in the training set

Our modus operandi is to train on companies that are known to have exited or failed and subsequently apply those to currently private companies to determine how they may fare. In addition to the three classes of interest, namely, IPO, acquisition and failure, we decided to test the inclusion of another class. The new class represents private firms that have neither exited nor failed and the aim is to see how its addition affects the metrics for success and failure.

Table 3
MLP follow-on funding model metrics.

Accuracy	0.800	
Class	Precision	Recall
no follow-on funding	0.836	0.759
follow-on funding	0.769	0.843

Table 4
Random Forest Exit model metrics.

Accuracy	0.883	
Class	Precision	Recall
IPO	0.850	0.719
Acquisition	0.757	0.882
Fail	0.959	0.918

Table 5
Random Forest follow-on funding model metrics.

Accuracy	0.837	
Class	Precision	Recall
no follow-on funding	0.876	0.795
follow-on funding	0.803	0.881

Table 6
XGBoost Exit model metrics.

Accuracy	0.894	
Class	Precision	Recall
IPO	0.847	0.755
Acquisition	0.783	0.883
Fail	0.963	0.929

Table 7
XGBoost follow-on funding model metrics.

Accuracy	0.834	
Class	Precision	Recall
no follow-on funding	0.871	0.795
follow-on funding	0.802	0.876

Table 8
Ensemble exit model metrics.

Accuracy	0.890	
Class	Precision	Recall
IPO	0.854	0.743
Acquisition	0.775	0.883
Fail	0.960	0.926

Table 9
Ensemble follow-on funding model metrics.

Class	Precision	Recall
no follow-on funding	0.880	0.796
follow-on funding	0.805	0.886

Table 10
Comparison of modeling results. Best exit model metrics in blue; best follow-on funding metrics in green.

Model	Exit Model Probabilities							Follow-on Funding Model Probabilities				
		IPO		Acquired		Failed			no follow-on		follow-on	
	Accuracy	precision	recall	precision	recall	precision	recall	Accuracy	precision	recall	precision	recall
MLP	0.880	0.811	0.703	0.768	0.856	0.952	0.929	0.800	0.836	0.759	0.769	0.843
Random Forest	0.883	0.850	0.719	0.757	0.882	0.959	0.918	0.837	0.876	0.795	0.803	0.881
XGBoost	0.894	0.847	0.755	0.783	0.883	0.963	0.929	0.834	0.871	0.795	0.802	0.876
Ensemble	0.890	0.854	0.743	0.775	0.883	0.960	0.926	0.840	0.880	0.796	0.805	0.886

Private firms comprise 84.5% of the entire data set, amounting to 796,663 data points. Thus the addition of the class requires sampling to ensure a balanced set. Two approaches were taken and the experimental results presented below. In the first experiment, before retraining the models we randomly sampled to obtain a subset of equal size to that of the acquired companies. The resulting confusion matrix for the held-out test set is shown in Fig. 4.

Accuracy drops for each model by around 10%. This is likely due to the randomly selected private companies often being indistinguishable from closed companies because of the sparsity of features. This is reflected in the closed precision and recall being most reduced (~90% down to ~80%).

For the second experiment, we applied a threshold to the private companies to remove those with little information. Equation (1) was used with $\alpha = 1$. This dropped 660,345 firms and further random sampling drops another 121,192 so that the number of private companies is balanced with acquisitions. Fig. 5 shows the resulting confusion matrix.

Dropping sparse, private companies increased the closed class precision and recall, as expected but acquisition recall was reduced by around 10%. It seems the model has flipped to finding it difficult to distinguish between companies that continue to operate privately and those that are likely to be acquired.

6.8. Two-step classification

Classification of companies with respect to failure, IPO and acquisition means that models must generalize to a broad set of company characteristics. This leads to a necessity of greater model complexity and therefore greater risk of overfitting. If the task is a simpler success-vs-fail binary classification then there are fewer burdens on the models with respect to learning class boundaries. To address this, the authors decomposed the exit classification into two binary classification steps to determine the effect on performance. In the first step a model ensemble was trained to classify according to successful exit (IPO or acquisition) or failure. In the second step another ensemble was trained to distinguish between the likelihood of IPO and acquisition.

Note that to avoid target leakage, the training/test data split used to train and evaluate this second step was the same as the first step with the exception that only public and acquired companies were used in training the second step ensemble. The authors expected a potential performance improvement because the characteristics of companies across the success/fail boundary are likely to be very different from those that span the IPO/acquisition boundary and therefore the two ensembles would be better able to distinguish the classes.

Figure 6 shows the confusion matrices from steps one and two. The exit vs fail models exhibited high accuracy (94%), exit precision (92.6%) and exit recall (98%). The models' ability to effectively distinguish the classes is evident. The accuracy of the second step models was lesser at 89.6% with IPO precision = 88%, IPO recall = 78%, acquisition precision = 90% and acquisition recall = 95%. These metrics are comparable to the original, single-step ensemble.

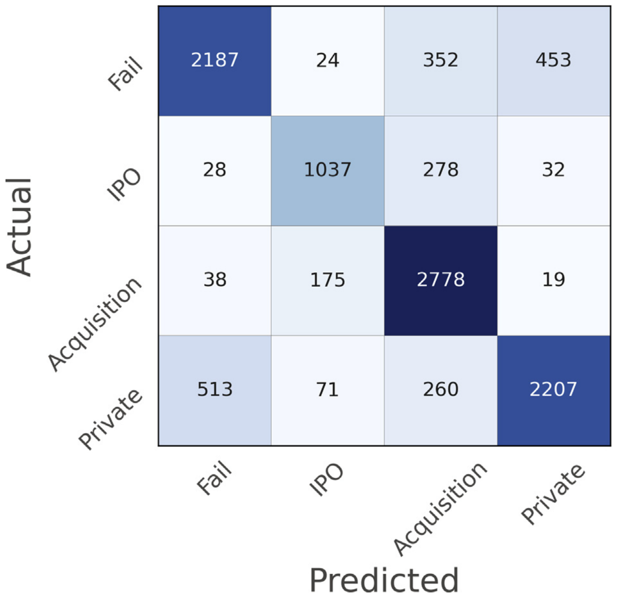


Fig. 4. Confusion matrix showing model outputs with the addition of the randomly sampled private class.

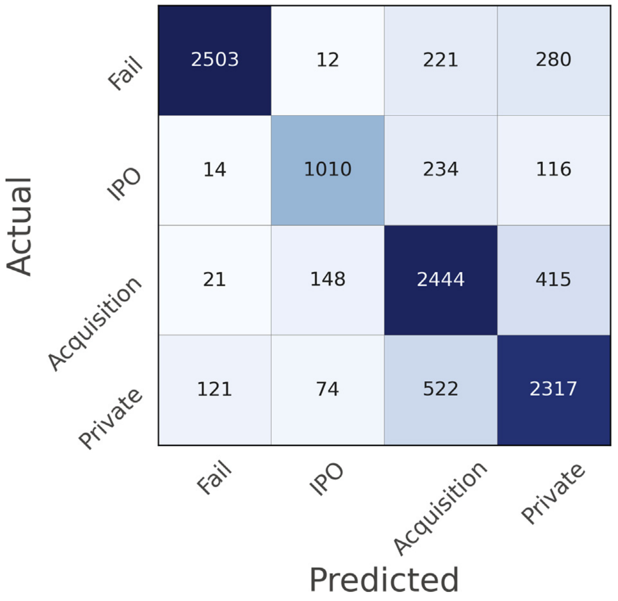


Fig. 5. Confusion matrix from models after filtering out private firms according to the amount of missing information.

The two model ensembles were subsequently combined so that the three-class classification could be obtained and overall performance measured. Fig. 7 shows the confusion matrix. The overall accuracy, at 88% is slightly lower than the original single-step ensemble. IPO precision (84%) is slightly lower and acquisition precision (82.5%) is higher while recall (IPO = 77.5%, acquisition = 92.5%) are also higher. While the failed class precision is higher (97%), as expected from the performance of the success/fail binary component, the failed recall is lower (88.6%). In conclusion, while the success/failure classifier performs well, the difficulty in distinguishing the IPO vs acquisition outcome reduces the performance of the two-step approach in comparison to the original classifier ensemble.

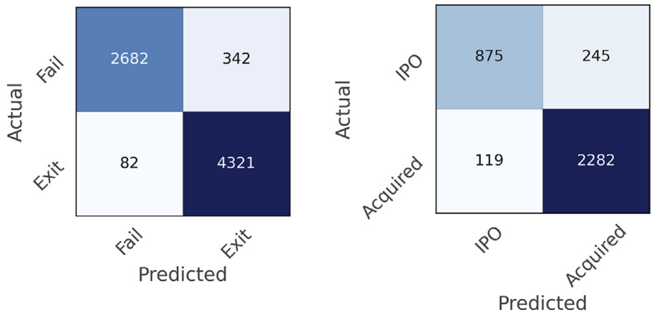


Fig. 6. Confusion matrices for two binary exit classification tasks.

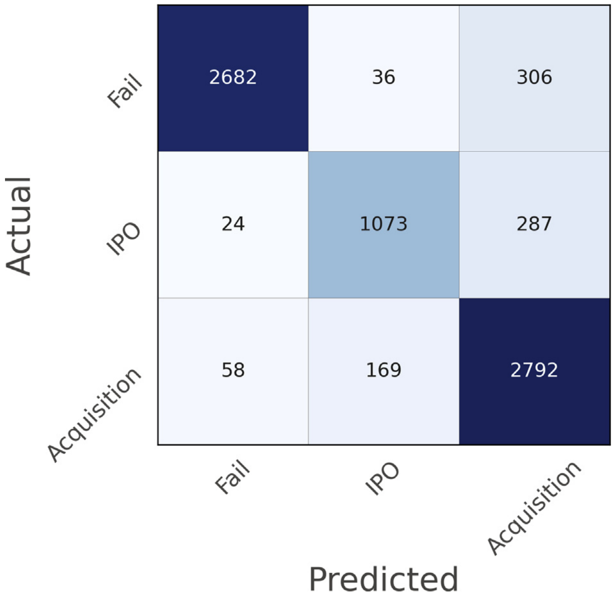


Fig. 7. Confusion matrix for the combination of results from the two-step classification ensembles.

6.9. Robustness: model performance for exit by funding round

It is desirable that the models' accuracy be maintained in subsets of the data. Perhaps one of the most important ways to partition the data is by the latest funding round. This is important because it is common for fund managers to restrict their portfolios to companies at a certain funding stage. VCs, for example, might be more interested in promising startups in the earliest stages.

Figure 8 shows how the CapitalVX model ensemble performs on a subset of companies from the test data split at specific funding stages. For comparison, the counts of companies at each stage are also shown. As expected, the bulk of companies in the test split are at the seed stage and counts decline as the funding stages progress. There are no instances of a company going public when its last recorded funding round was angel-funded. Thus the missing point for the blue line on the precision graph. We see from the accuracy plot that there is a consistently high accuracy level across funding rounds, except for Series F, for which there are very few observations.

Only 18 companies where the last funding round was seed subsequently went public and the models correctly identified six of them, with no false positives. It is unlikely the companies truly only received seed funding before their IPOs. It is more likely that the information was simply not up-to-date in Crunchbase but it indicates the efficacy of the model. Thus, precision is uniformly high across the various funding rounds.

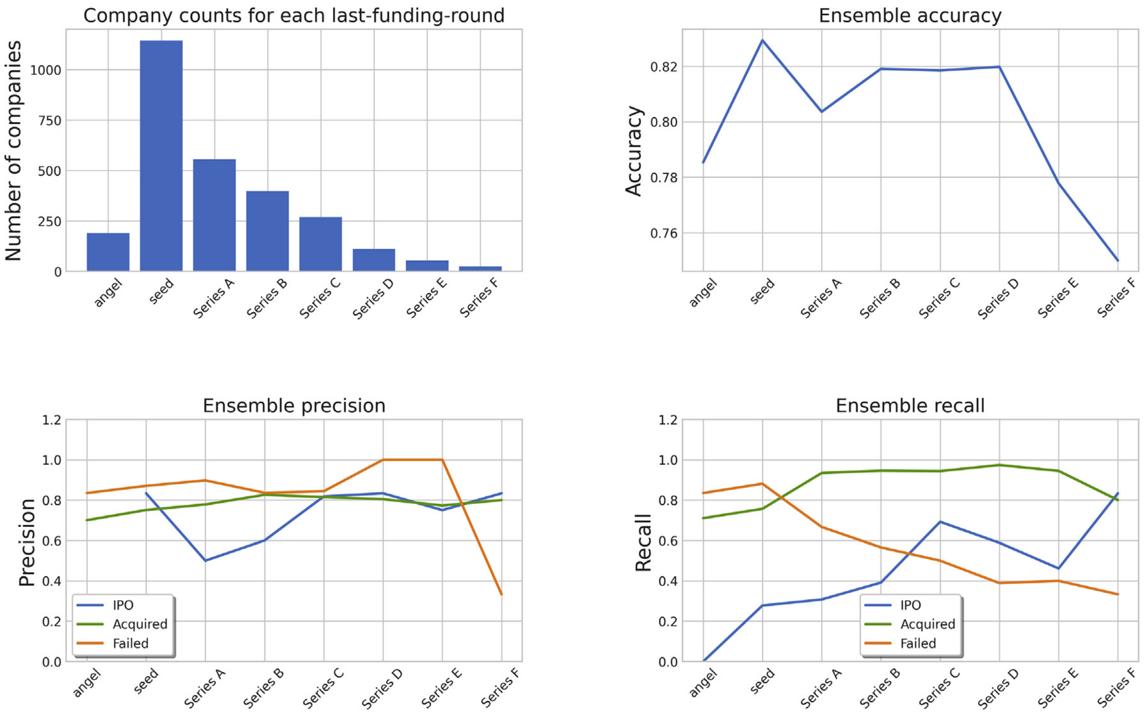


Fig. 8. Exit model ensemble performance metrics by last funding round. Top-left: the number of companies at each funding stage; top-right: the overall accuracy of the exit model ensemble; bottom-left: the model's precision for company outcome at each round; bottom-right: the model's recall for company outcome at each round.

On the other hand, it can be seen that IPO recall is low for the early stages where false negatives dominate. This might be due to correlation between last funding round and likelihood of a good exit. Also, as expected the precision and recall for IPO increases as funding progresses. Conversely the failure performance is reduced over the later rounds where the model naturally expects a company to be less likely to close. For round F, there were three recorded closures however, due to lack of data, the model incorrectly reported two false positives.

Acquisition and failure are predicted with higher but similar values for precision and recall across the early stages. It is also observed that the acquisition and IPO metrics increase with funding rounds, which is no surprise. These robustness tests suggest the models are working across startup funding stages.

6.10. Robustness: model performance for follow-on funding by funding round

Figure 9 shows the performance of the model ensemble for follow-on funding across the last recorded funding rounds of the companies in the test set. Accuracy drops across series E and F rounds because of the drop there in precision and recall. However, as expected the models do reasonably better, as funding progresses, where companies did actually receive follow-on funding. It is encouraging to see the higher precision across the early rounds because this indicates the model is useful in identifying promising startups.

7. Explainability

Machine learning models suffer from the criticism that they are black boxes. Deep learning neural nets, such as the MLP we used, have thousands (often millions) of parameters and explaining the complicated functional relationship between the model inputs and the decision of the model is fraught with difficulty. In contrast, a simple logistic regression is easily understood from an examination of model coefficients and odds ratios. Yet, explaining why a machine learning model arrived at very different predictions for comparable startups is important for investors.

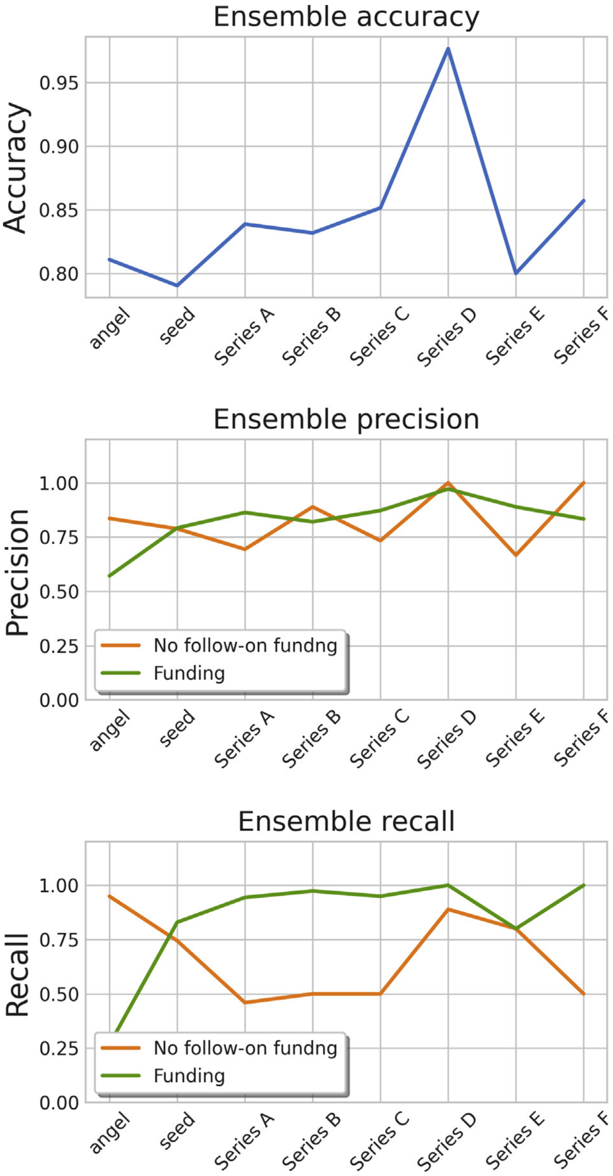


Fig. 9. Follow-on funding model ensemble performance metrics by last funding round. Top the overall accuracy of the model ensemble; middle: the model's precision at each round; bottom-right: the model's recall at each round.

There are two levels of explainability of a model: One, the importance of each feature (variable) used in the model across all observations in the data (global explainability) and two, which features are important for a specific prediction made by the model (instance-level explainability). Both levels of explainability are important to provide a human-understandable perspective on why a model produces an outcome given an input, e.g., one might wonder why a company is more likely to be acquired than go public but still have a reasonably high probability of failure and why the probabilities of these outcomes should be trusted. For model explainability, the contribution of specific features is easily understood in the likes of regression models where coefficients are learned and assigned to features; important features are assigned statistical significance. In the case of numerical features, an increase in that feature by one unit increases the outcome by the amount of the coefficient. Similarly, decision trees and their ensemble variants such as Random Forests and gradient-boosting can readily report on feature importance because that information is derived as part of the process of training the models. While feature importance here can be interpreted globally, it is not specific

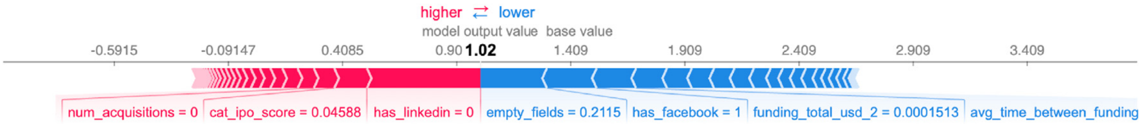


Fig. 10. SHAP feature contributions to the XGBoost prediction on whether a company will close. The graphic was generated by the Python SHAP library. The base value indicates the average model output over the training data. Red features push the prediction higher while blue ones push the prediction lower.

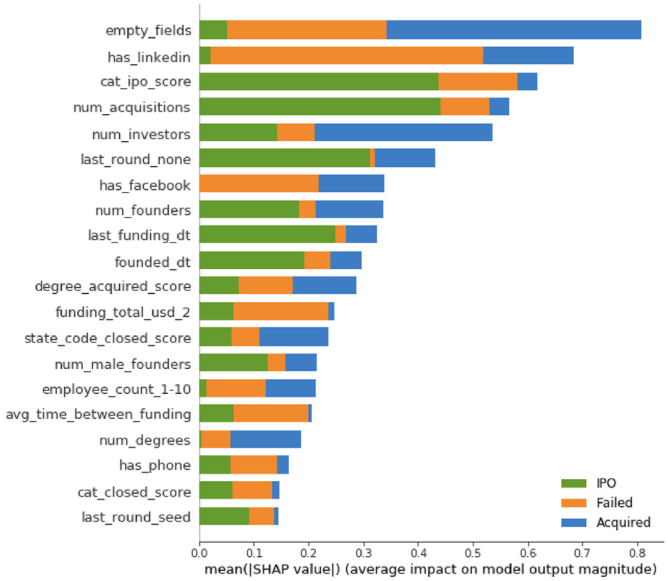


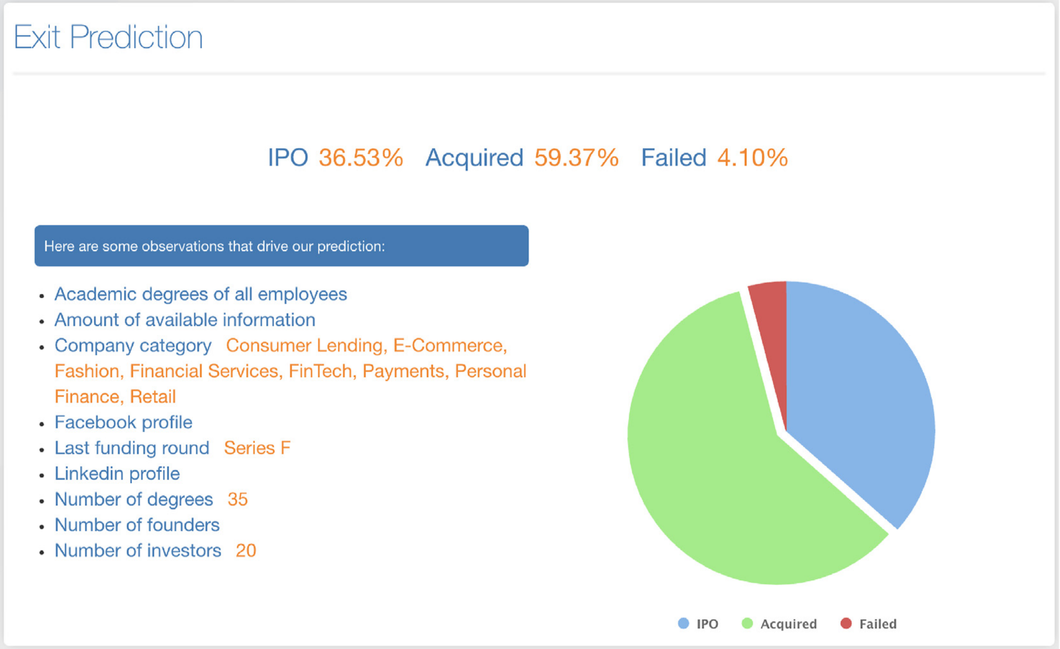
Fig. 11. SHAP feature contributions to the XGBoost prediction over the three exit classes for a specific company. The graphic was generated by the Python SHAP library and shows which features are most important for the exit prediction of a given company.

to the model's performance on specific inputs. Instance-level explainability is a little less straightforward due to the black-box nature of many machine learning models but it is probably the most valuable question to answer because it provides insight into why each individual prediction is made. It also relates to how the influence of a feature can vary with different observations that are presented to a model.

The CapitalVX models are packaged into an interactive web application where companies can be discovered and assessed, see www.venhound.com. It is desirable to visualize the influence of features on the model outputs in a user-friendly manner and Fig. 12 shows what this looks like. We experimented with two of the most well-known explainability algorithms. Both operate by exploring the effect of each feature on the performance of a model, e.g., by looking at the effect on accuracy or information gain. Given the black-box nature of the models under scrutiny, both are model-agnostic and therefore readily applicable to any model.

The first approach investigated was LIME (see Ribeiro et al., 2016).[24] This approach is interesting because it examines the feature space local to an observation and then applies a locally interpretable model such as Lasso to approximate the model's black box function $f(x)$ with a simple linear function $g(x)$, which is easily interpreted. Regularization can be applied to reduce the number of features, thereby zeroing in on the important ones. An alternative approach to find the most important features is forward or backward selection where features are gradually added or removed, respectively while their effect on loss is measured. Those that make the biggest improvements are deemed the most important.

The second algorithm explored was SHAP,[21] based on ideas from game theory in original work by Shapley (1951).[26] This is similar to LIME in that it computes feature importance at the instance level and then aggregates


Exit Prediction

IPO 36.53% Acquired 59.37% Failed 4.10%

Here are some observations that drive our prediction:

- Academic degrees of all employees
- Amount of available information
- Company category Consumer Lending, E-Commerce, Fashion, Financial Services, FinTech, Payments, Personal Finance, Retail
- Facebook profile
- Last funding round Series F
- Linkedin profile
- Number of degrees 35
- Number of founders
- Number of investors 20

Fig. 12. CapitalVX web interface showing the features most prominent in the exit probabilities of a company.

across all observations to generate global explainability, using its additivity property. Figs. 10 and 11 show visualizations generated by the Python SHAP library,[3] depicting the influence of some of the features on the XGBoost model for a company's exit. The company is Grooveshark and Fig. 11 indicates that the feature representing the number of empty fields has the greatest impact on determining whether it will be acquired or fail, suggesting also that keeping track of unreported data is useful in its own right in machine learning models. The second-most important feature, *has_linkedin*, is determined by whether the firm has a Linked-in account and, in this case, its absence has a strong bearing on failure. The third-most important feature for the Grooveshark prediction is *cat_ipo_score* representing the weighted, target-encoded variable for company category; see Section 4.2 for an explanation of weighted target encoding. The next most influential feature here is the number of acquisitions by the company, *num_acquisitions*, and has most influence on whether the model deems the company should be public. The number of investors, *num_investors*, also has bearing on whether the company would be predicted to go public. When a company has no reported funding, indicated by *last_round_none*, then this weighs more on failure. In the Grooveshark instance, the last funding round was a seed round and therefore this would have slightly reduced the odds of the failure prediction. Similar to *has_linkedin*, the absence of *has_facebook* has most influence on whether the model will regard the firm as likely to fail. *num_founders*, *last_funding_dt* and *founded_date* represent the number of founders, the date of the last funding round and the date the company was founded, respectively. Each of these, out of tens of features, has the most influence on whether a company will be likely to go public. A similar use of SHAP in explaining which companies form good pairs in an acquirer-acquiree choice setting is explored in Futagami et al. (2021).[15]

SHAP has a rigorous mathematical basis through its adherence to the three properties of *local accuracy* (best fit model in the local space of the observation), *missingness* (missing features generate a zero attribution value), and *consistency* (if adding a feature does not change the model prediction, then its Shapley value is zero, and if it increases the function value, then its Shapley value will also increase). See the aforementioned paper for a description of these properties.

The authors settled on using the SHAP approach for the web application. LIME's random sampling makes it non-deterministic and while sampling multiple runs and subsequently taking the average might allow it to settle on a stable

[3] www.github.com/slundberg/shap.

output, the algorithm has a high time complexity. This makes it unsuitable for a good real-time user experience. On the other hand, SHAP provides a fast, deterministic variant called TreeSHAP that is built into the XGBoost library.[4] Since XGBoost is the dominant model in the ensembles, this is taken as a good proxy for explainability that can be calculated and displayed almost instantly in the web application. In Fig. 12 we see how the exit prediction for a company is presented in terms of the probabilities of different outcomes along with the most important factors that surfaced the predicted probabilities.

8. Conclusion and application

Our machine learning model, denoted CapitalVX, demonstrates the potential of machine learning and deep learning to impact the venture capital industry, which has traditionally operated in a high-risk, high-reward manner, often depending on the payout from a single unicorn company to cover the financial responsibilities of the venture capitalist to their investors. Our ensemble approach can predict the exit scenario of a startup with over four times the accuracy of a successful venture capitalist. Out of every ten portfolio companies of a successful venture capital firm, one may be a unicorn and another may be successful, i.e., roughly a twenty percent probability of predicting an IPO exit. Comparatively, the accuracy of our models with respect to an IPO is between eighty-eight and ninety percent. Using models like CapitalVX could mean the difference between two successful exits and eight.

We have discussed many successful venture capitalists, but according to Dean (2017)[12] ninety-five percent fall into the unsuccessful category. That is to say they are not returning a profit to their own investors. These unprofitable venture capitalists would clearly benefit from machine learning and in turn, an increased rate of return for investors would mean additional resources to invest in an even greater number of businesses. Machine learning has the potential to democratize the investment process for VC/PE firms.

Startup investors usually perform their own financial analysis of potential target companies, after a period of qualitative investigation involving getting to know the founders and business. Machine learning models can provide exit predictions in real time along with a feature analysis that identifies aspects of the company that make it a good investment or, on the other hand, raise red flags. Thus, the modern venture capitalist can peer into the black box and make investment decisions faster and more reliably.

Declaration of competing interest

I, Greg Ross, on behalf of my co-authors, Daniel Sciro, Sanjiv Das and Hussain Raza confirm that there are no potential conflicts of interest.

Appendices. A. Features

The following table lists the features used to train the models.

	mean	std	min	median	max
avg_time_between_funding	8880780.45	22741417.51	0.00	0.00	573823800.00
funding_rounds	1.12	1.82	0.00	0.00	34.00
investment_count	0.27	3.45	0.00	0.00	451.00
num_investors	1.51	3.30	0.00	0.00	105.00
founded_dt	922106888.71	517836561.63	0.00	1136102400.00	1585724400.00
last_funding_dt	639762503.96	687531084.21	0.00	0.00	1586934000.00
funding_total_usd_2	26676637.79	304414350.63	0.00	0.00	30079814466.00
num_female_founders	0.05	0.24	0.00	0.00	4.00
num_male_founders	0.59	0.99	0.00	0.00	20.00
num_patents	2.50	135.26	0.00	0.00	20302.00
num_degrees	5.89	40.11	0.00	0.00	4044.00
num_events	0.81	7.01	0.00	0.00	640.00
num_acquisitions	0.79	4.78	0.00	0.00	371.00

(*continued on next page*)

[4] www.github.com/dmlc/xgboost.

(*continued*)

	mean	std	min	median	max
num_top_degrees	1.01	3.13	0.00	0.00	113.00
has_domain	0.96	0.20	0.00	1.00	1.00
has_email	0.66	0.48	0.00	1.00	1.00
has_phone	0.66	0.47	0.00	1.00	1.00
has_facebook	0.56	0.50	0.00	1.00	1.00
has_linkedin	0.51	0.50	0.00	1.00	1.00
has_twitter	0.63	0.48	0.00	1.00	1.00
has_aliases	0.16	0.36	0.00	0.00	1.00
has_address	0.65	0.48	0.00	1.00	1.00
description_length	92.97	35.15	1.00	99.00	360.00
empty_fields	30.33	11.71	5.00	30.00	58.00
prop_female	0.06	0.18	0.00	0.00	1.00
prop_female_founders	0.03	0.15	0.00	0.00	1.00
num_founders	0.64	1.04	0.00	0.00	23.00
vc_support	0.26	0.44	0.00	0.00	1.00
employee_count_1−10	0.28	0.45	0.00	0.00	1.00
employee_count_10,000+	0.03	0.18	0.00	0.00	1.00
employee_count_1001−5000	0.03	0.18	0.00	0.00	1.00
employee_count_101−250	0.05	0.22	0.00	0.00	1.00
employee_count_11−50	0.22	0.41	0.00	0.00	1.00
employee_count_251−500	0.04	0.19	0.00	0.00	1.00
employee_count_5001−10000	0.02	0.14	0.00	0.00	1.00
employee_count_501−1000	0.04	0.19	0.00	0.00	1.00
employee_count_51−100	0.09	0.28	0.00	0.00	1.00
employee_count_unknown	0.20	0.40	0.00	0.00	1.00
last_round_angel	0.01	0.11	0.00	0.00	1.00
last_round_convertible_note	0.01	0.09	0.00	0.00	1.00
last_round_corporate_round	0.00	0.06	0.00	0.00	1.00
last_round_debt_financing	0.03	0.16	0.00	0.00	1.00
last_round_equity_crowdfunding	0.00	0.07	0.00	0.00	1.00
last_round_grant	0.01	0.12	0.00	0.00	1.00
last_round_initial_coin_offering	0.00	0.02	0.00	0.00	1.00
last_round_non_equity_assistance	0.00	0.06	0.00	0.00	1.00
last_round_none	0.56	0.50	0.00	1.00	1.00
last_round_pre_seed	0.00	0.05	0.00	0.00	1.00
last_round_private_equity	0.03	0.17	0.00	0.00	1.00
last_round_product_crowdfunding	0.00	0.02	0.00	0.00	1.00
last_round_secondary_market	0.00	0.05	0.00	0.00	1.00
last_round_seed	0.09	0.28	0.00	0.00	1.00
last_round_series_a	0.05	0.21	0.00	0.00	1.00
last_round_series_b	0.03	0.18	0.00	0.00	1.00
last_round_series_c	0.02	0.15	0.00	0.00	1.00
last_round_series_d	0.01	0.10	0.00	0.00	1.00
last_round_series_e	0.01	0.07	0.00	0.00	1.00
last_round_series_f	0.00	0.04	0.00	0.00	1.00
last_round_series_g	0.00	0.02	0.00	0.00	1.00
last_round_series_h	0.00	0.02	0.00	0.00	1.00
last_round_series_i	0.00	0.01	0.00	0.00	1.00
last_round_series_j	0.00	0.00	0.00	0.00	0.00
last_round_series_unknown	0.11	0.32	0.00	0.00	1.00
last_round_undisclosed	0.01	0.09	0.00	0.00	1.00
cat_ipo_score	0.14	0.11	0.00	0.09	0.90
cat_acquired_score	0.33	0.10	0.00	0.33	0.74
cat_closed_score	0.26	0.10	0.00	0.25	1.00
degree_ipo_score	0.20	0.21	0.00	0.00	1.00
degree_acquired_score	0.23	0.25	0.00	0.00	1.00
degree_closed_score	0.03	0.05	0.00	0.00	1.00
country_code_ipo_score	0.13	0.09	0.00	0.14	1.00

(*continued*)

	mean	std	min	median	max
country_code_acq_score	0.29	0.17	0.00	0.34	1.00
country_code_closed_score	0.19	0.10	0.00	0.23	1.00
state_code_ipo_score	0.08	0.09	0.00	0.07	0.56
state_code_acq_score	0.21	0.22	0.00	0.10	0.56
state_code_closed_score	0.11	0.12	0.00	0.12	0.55

References

1. Adcock AB, Lakkam M, Meyer J. *CS 224W Final Report Group 37*. 2012.
2. Antretter T, Blohm I, Grichnik D, Wincent J. Predicting new venture survival: a Twitter-based machine learning approach to measuring online legitimacy. *J Busi Ventur Insight*. 2019, June;11, e00109.
3. Bernstein S, Giroud X, Townsend RR. The impact of venture capital monitoring. *J Finance*. 2016;71(4):1591—1622.
4. Bertoni F, Colombo MG, Grilli L. Venture capital financing and the growth of high-tech start-ups: disentangling treatment from selection effects. *Res Pol*. 2011, September;40(7):1028—1043.
5. Biesinger M, Bircan C, Ljungqvist A. *Value Creation in Private Equity. SSRN Scholarly Paper ID 3587559*. Rochester, NY: Social Science Research Network; 2020, May.
6. Brander JA, Amit R, Antweiler W. Venture-capital syndication: improved venture selection vs. The value-added hypothesis. *J Econ Manag Strat*. 2002;11(3):423—452.
7. Bubna A, Das SR, Prabhala N. Venture capital communities. *J Financ Quant Anal*. 2020;55(2):621—651.
8. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM; 2016, August:785—794.
9. Cockburn IM, Macgarvie MJ. Patents, thickets and the financing of early-stage firms: evidence from the software industry. *J Econ*. 2009;18(3):729—773.
10. Das SR, Jo H, Kim Y. Polishing diamonds in the rough: the sources of syndicated venture performance. *J Financ Intermediation*. 2011, April;20(2):199—230.
11. Davis AE, Aldrich HE, Longest KC. Resource drain or process gains? Team status characteristics and group functioning among startup teams. *Front Entrepren Res*. 2009;29(11):14.
12. Dean T. *The Meeting that Showed Me the Truth about VCs*. 2017, June.
13. Dellermann D, Lipusch N, Ebel P, Popp KM, Leimeister JM. Finding the unicorn: predicting early stage startup success through a hybrid intelligence method. *SSRN Electr J,* 2017. https://ssrn.com/abstract=3159123.
14. Friedman JH. Stochastic gradient boosting. *Comput Stat Data Anal*. 2002;38(4):367—378.
15. Futagami K, Fukazawa Y, Kapoor N, Kito T. Pairwise acquisition prediction with SHAP value interpretation. *J Finan Data Sci*. 2021, March;7:22—44.
16. Glupker J, Nair V, Richman B, Riener K, Sharma A. Predicting investor success using graph theory and machine learning. *J Invest Manag*. 2019;17(1):92—103.
17. Gupta S, Pienta R, Tamersoy A, Chau DH, Basole RC. Identifying successful investors in the startup ecosystem. In: *Proceedings of the 24th International Conference on World Wide Web*. New York, NY, USA: WWW '15 Companion; 2015:39—40. ACM.
18. Ho TK. Random decision Forests. In: *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC*. 1995, August:278—282.
19. Jalbert T, Jalbert M, Furumo K. The relationship between CEO gender, financial performance, and financial management. *J Bus Econ Res*. 2012, December;11(1):25.
20. Krishna A, Agrawal A, Choudhary A. Predicting the outcome of startups: less failure, more success. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016, December:798—805.
21. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Adv Neural Inf Process Syst*. 2017;30:4765—4774.
22. Mann RJ, Sager TW. Patents, venture capital, and software start-ups. *Res Pol*. 2007, March;36(2):193—208.
23. Puri M, Zarutskie R. On the lifecycle dynamics of venture-capital- and non-venture-capital-financed firms. *J Finance*. 2012;67(6):2247—2293.
24. Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": explaining the predictions of any classifier. In: *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, August:1135—1144.
25. Rin MD, Hellmann T, Puri M. *A Survey of Venture Capital Research. Handbook of the Economics of Finance*. Elsevier; 2013.
26. Shapley L. *Notes on the N-Person Game — II: The Value of an N-Person Game*. RAND Corporation ATI 210720(RM-670); 1951, August.
27. Sharchilev B, Roizner M, Rumyantsev A, Ozornin D, Serdyukov P, de Rijke M. Web-based startup success prediction. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*. New York, NY, USA: Association for Computing Machinery; 2018, October:2283—2291.

28. Sorenson O, Stuart TE. Syndication networks and the spatial distribution of venture capital investments. *Am J Sociol*. 2001;106(6):1546−1588.
29. Sørensen M. How smart is smart money? A two-sided matching model of venture capital. *J Finance*. 2007;62(6):2725−2762.
30. Srinivasan S, Barchas I, Gorenberg M, Simoudis E. Venture capital: fueling the innovation economy. *Computer*. 2014, August;47(8):40−47.
31. Sunesson D. Alma mater matters: the value of school ties in the venture capital industry. *SSRN Electr J*; 2009. https://ssrn.com/abstract=1372463.
32. Xiang G, Zheng Z, Wen M, Hong J, Rose C, Liu C. A supervised approach to predict company acquisition with factual and topic features using profiles and news articles on techcrunch. In: *ICWSM 2012 - Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*. 2012, 01.
33. Xu R, Chen H, Zhao JL. Predicting corporate venture capital investment. In: *38th International Conference on Information Systems (ICIS 2017): Transforming Society with Digital Innovation, Korea, Republic of*. Association for Information Systems; 2017, December:5841−5849.