The use of temporal features to predict startup success

Jirko Rubruck[1], Yigit Ihlamur[2]

[1] University of Oxford, UK

[2] Vela Partners, US

Abstract

The aim of the project is to predict the success of companies via properties extracted from two large data sets representing successful and unsuccessful companies. Particularly, this algorithm exploits new temporal features to predict the success of companies. We thoroughly explore the temporal structure of these newly added features through visualisations. Subsequently, we engineer and extract features for predicting the success of companies. To enhance the predictive power of our model, we also impute missing values in our data sets using the centroids generated by the k-nearest neighbour (KNN) algorithm which we ran on the data sets. Finally, for predictive modelling, we concentrated on two separate model classes: Logistic regression and a neural network with a single hidden layer. To evaluate model performance, we plot receiver operating characteristic (ROC) curves and confusion matrices. We find that newly engineered temporal features allow for substantial improvements, resulting in True Positive rates upwards of 70%.

1 Introduction

The primary objective of algorithms developed at Vela Partners, also known as **Ventech**, is to leverage machine learning and advanced data analysis to predict startup success. Predicting viable business ideas and the future success of startups in the early stages is a notoriously difficult task due to the relative scarcity of data and a lack of traditional company success metrics such as sales revenue, profit margins or year-on-year sales growth. In this iteration of Moneyball, we focus on a set of metrics derived from newly acquired data to predict the success of startups. In this paper, we will first describe the structure of the new data sets in section 2. We will then conduct an extensive exploration of the new data in section 3. Furthermore, section 3 will present our attempts to exploit temporal structure to generate maximally useful predictors for our binary classification task (The grouping of companies as successful and unsuccessful companies). In section 4, we will show the results of the predictive modelling of the data through logistic regression. In that section, we also explore model performance in more detail by examining ROC curves to quantify the trade-offs between true positives and false negatives at different decision thresholds. Finally, in section 5, we move on to model the data using a simple deep neural network with a single hidden layer. This analysis is intended to exploit any non-linear relationships in the data.

2 The structure of the data set

The new funding rounds data set contains fine-grained details about capital received by each company at different stages of their early life cycles. The columns in the data set are as follows, whereby each column corresponds to a specific funding round such as pre-seed, angel, seed and Series A:

- **org uuid** This column contains strings that uniquely identify the companies contained in the original spreadsheet of successful and unsuccessful companies.
- **org name** The name of the company corresponding to the uuid string.
- **investment type** The column contains information on the particular stage of the funding round, i.e. pre-seed, seed, angel, or series a.
- **Round created at** The column contains the date of the respective funding round.
- **Amount raised usd** The column contains the amount raised in this round as an integer value.
- **Investor count** The column contains the number of investors involved in this funding found as an integer.

- **Investor Name** The column contains the names of the investors involved in this funding round.
- **Investor uuidd** The column contains the uuid's corresponding to the investors.

3 Data exploration and feature engineering

We start our exploratory analysis and feature engineering work with a broad sweep of new temporal feature ideas. We examine the frequency of different funding rounds, the time between company founding and different funding rounds, and the total funding obtained in different types of funding rounds.

3.1 Frequency of funding rounds

Our first exploratory analysis is concerned with the number of funding rounds obtained at each stage for the successful and the unsuccessful companies. To this end, we grouped the data by individual companies' organisation uuid's and the investment type column in the successful and unsuccessful company data-frames. The results are presented in Figure 1. The histogram was normalised by the number of companies in each group (successful/unsuccessful) to convert frequencies to probabilities. Please note that the plotted graphs are not true probability density functions that sum to one. The remaining probability mass is assigned to the event in which the company did not have the respective funding round. I.e., companies skipped this type of funding. For two features, we find substantial differences between successful and unsuccessful data sets. These features are the number of angel funding rounds and the number of series a funding rounds. Particularly the result in the bottom panel on the angel funding round is interesting. Unsuccessful companies are more than three times more likely to have obtained angel funding. This effect is likely explained by the previously developed Moneyball models that founders of highly successful companies often perform well in their previous careers. This in turn alleviates the necessity of angel funding.

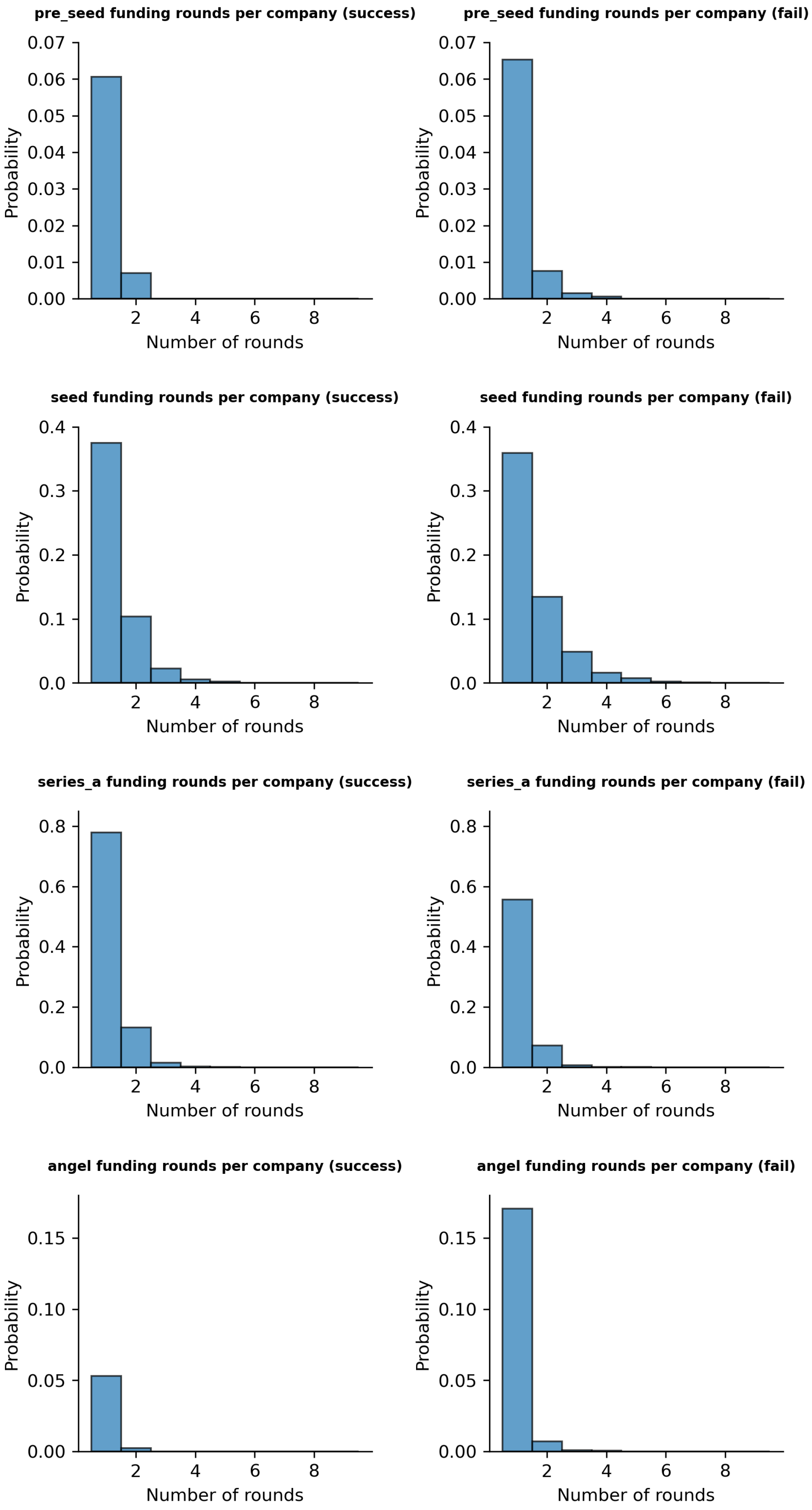


Figure 1: The number of funding rounds obtained for successful (left column) and unsuccessful companies(right column). Please note that the Figures have the remaining probability mass at 0.

3.2 Time taken till funding rounds

In the next step of our analysis, we engineered a novel feature for predictive modeling. We assessed the time in days till the first funding is obtained by the company. To get this metric, we extracted the founding date of each company from the original company's data-frame and the earliest funding round available for each company. We subsequently converted the date strings to pandas DateTime objects. The difference between dates as measured by days was then plotted as a histogram. The results can be found in Figure 2. We find that distributions for successful and unsuccessful companies are both positively skewed. However, the distribution of time differences has a more heavy tail for unsuccessful companies. In addition, measures of central tendency are much smaller for successful companies. The median time till first funding is 660 days for successful companies and 1143 days for unsuccessful companies. This shows the power of temporal features. The time until the first funding acquisition is a clear indicator of later company success.

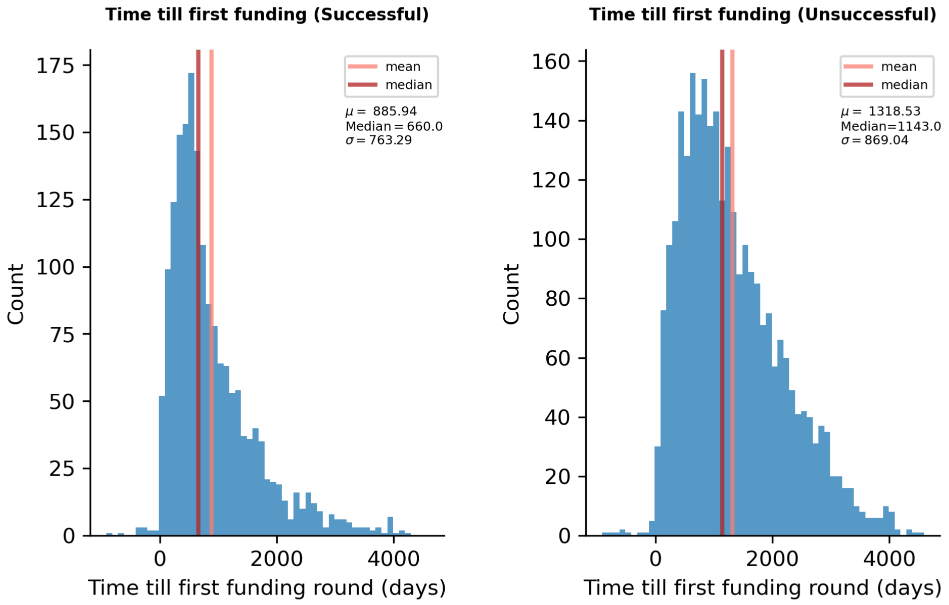


Figure 2: The time until first funding was obtained in days from the funding date for successful companies (left) and unsuccessful companies (right).

Next, we examined the time taken until series-a funding as a potential feature. The procedure taken was equivalent to the feature "time until first funding" feature results can be found in Figure 3.

Here again, we can find substantial differences between successful and unsuccessful companies with an 800 days difference in the median between both groups. It appears that successful companies generally obtain funding faster. This result broadly aligns with the common intuition of most venture capital firms or general observations about "winners keep winning".

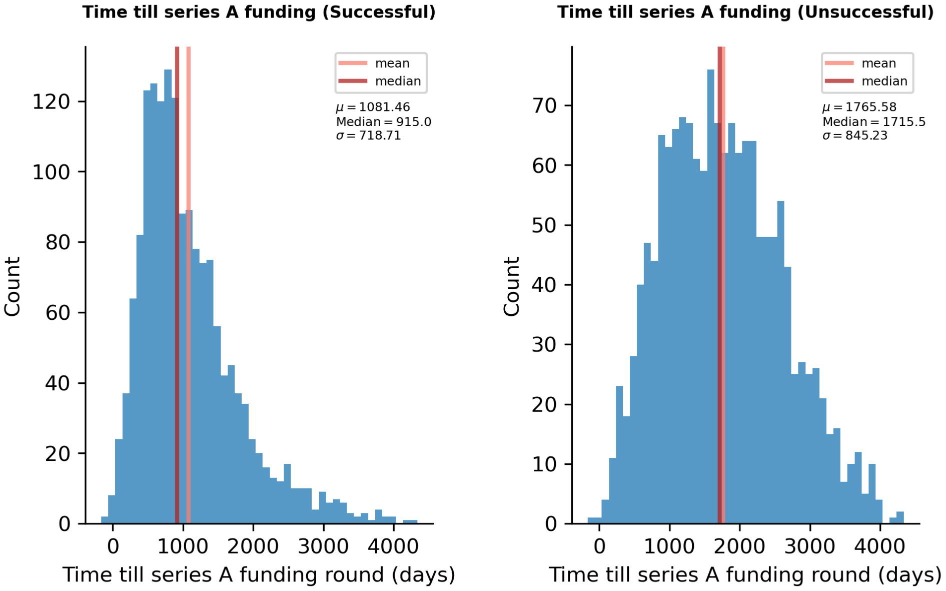


Figure 3: The time until series-a funding was obtained in days from the founding date for successful companies (left) and unsuccessful companies (right).

3.3 Total funding in different rounds

A further feature examined is the total seed funding obtained for each company. The funding obtained in the early stages might provide a signal for the later success of the company. The results are displayed in Figure 4. Importantly, many companies are not labeled to run a seed round so the proportion of zero seed funding is displayed in the upper left corner of the plots along with summary statistics. Given the large numbers in this category, we plot the amount of seed funding on a logarithmic scale with base 10. The results show that successful companies generally obtained higher funding than unsuccessful companies and are 6% more likely to run a seed funding round in the first place.

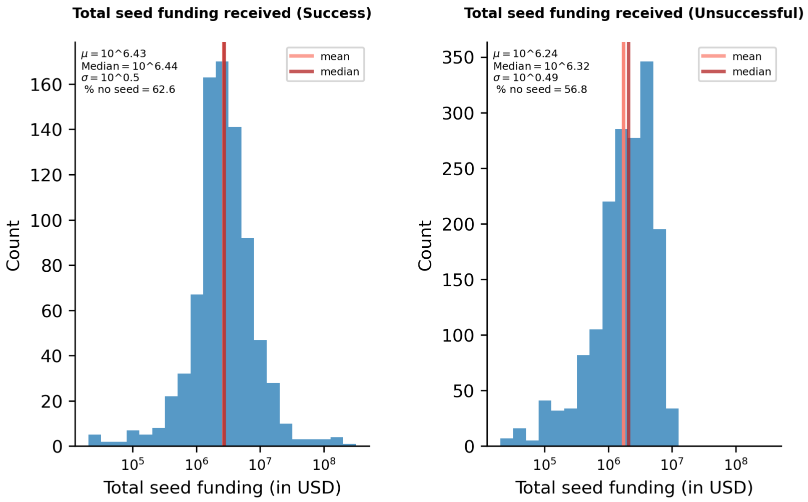


Figure 4: The total seed funding for successful companies (left) and unsuccessful companies (right) along with descriptive statistics.

Finally, we analyse the total Series A funding received by the company. Corresponding plots are analogous to the total seed funding plot and results can be found in Figure 5. Subsequently, we max-min normalised all continuous features in the range $[0, 1]$ to improve the convergence of models during training. As a final step, we impute missing values for the two features "Time until series A funding" and "Time until first funding". To do so, we run a KNN algorithm with K=5 on the two data sets and impute missing values using the algorithm's centroids.

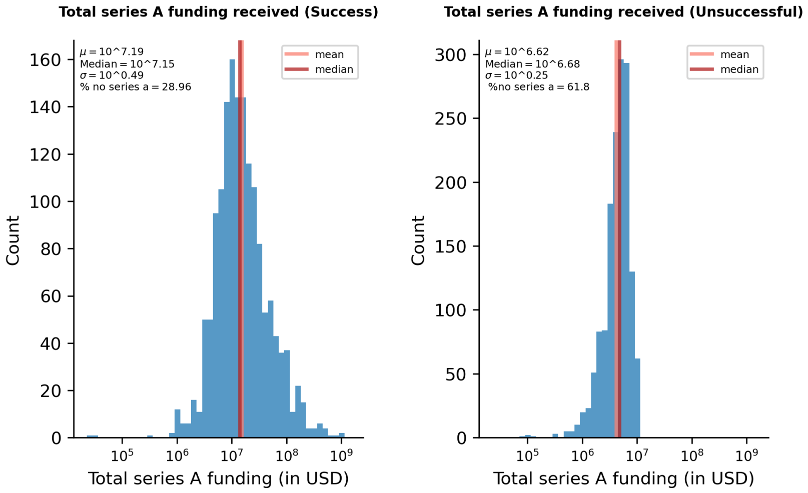


Figure 5: The total Series A funding for successful companies (left) and unsuccessful companies (right) along with descriptive statistics.

4 Predictive Modelling

4.1 Logistic regression with all features

To predict the success of the companies from features, we employ two different versions of the extracted features. The first version employs a data frame with 8 input features:

- number of pre-seed rounds
- number of seed rounds
- number of angel rounds
- number of series a rounds
- time until first funding normalised
- seed funding normalised
- series a funding normalised
- time until series a funding normalised.

For modeling this data with logistic regression, we conceptualise the data set as $\boldsymbol{x}_1, ..., \boldsymbol{x}_P \in \mathbb{R}^8$ with P representing the size of the data set. the task of the model is then to map these vectors of inputs to their correct ground truth labels $y_1, ..., y_P \in \{0, 1\}$. Our logistic regression model achieves this goal by mapping inputs through a weight vector $\boldsymbol{w} \in \mathbb{R}^8$. The prediction is then generated as $\hat{y} = \sigma(\mathbf{w}^T\mathbf{x})$ where $\sigma()$ represents the sigmoid function whose range is $(0, 1)$. To find the best classification performance, we then minimised the binary cross-entropy loss between predictions and labels as. We used 80% of data for model training and we used 20% of the data for model testing. To avoid over-fitting on the available data set.

$$L_{BCE} = \frac{1}{P} \sum_{i=1}^{P} -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{1}$$

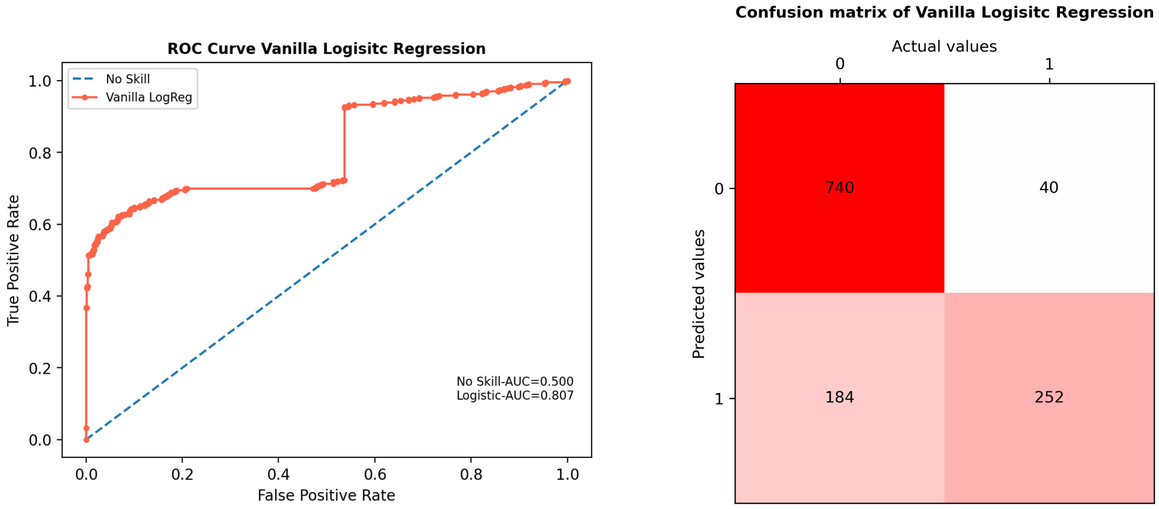
We employed the L-BFGS algorithm to minimise the above objective function for our problem formulation. Using this setup, we achieve an accuracy of about 80%, substantially outperforming a random baseline model. Key performance metrics are displayed in Table 1.

Metric	Model Result	Random model Result
True Positive Rate	.865	.371
False Negative Rate	.135	.629
True Negative Rate	.810	.653
False Positive Rate	.190	.347
Precision	.603	.498

Table 1: Table of Vanilla logistic regression performance metrics with Reduced features.

A confusion matrix for this reduced model is given in Figure 6. To explore the relationship of True Positive and False Positive rates, we plotted ROC curves for the classifier. The ROC curve displays the trade-off between True and False Positives as a function of different decision thresholds applied to model output probabilities. Results can be found in Figure 6a.

In addition, we plot a confusion matrix that displays classification performance on the test set to better understand error patterns (Figure 6b).



	0	1
0	740	40
1	184	252

(a) . (b)

Figure 6: (a) ROC curve for the vanilla logistic regression. The blue dotted line represents a no-effort classifier that classes all companies as unsuccessful. (b) Confusion Matrix for vanilla logistic regression on test set

4.2 Logistic regression with reduced features

In this section, we will explore how the model performs with a reduced set of features that excludes most information on Series A funding. Good performance with these reduced features would allow Ventech to identify successful companies to facilitate investment in a Series A funding round. The input predictors used are as follows:

- number of pre-seed rounds
- number of seed rounds
- number of angel rounds
- time until first funding normalised
- seed funding normalised
- time until series a funding normalised.

The structure of the data set and modeling approach is essentially equivalent with the exception of the size of input vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_P \in \mathbb{R}^6$ and our weight vector is now of the form $\boldsymbol{w} \in \mathbb{R}^6$. The performance of this model is at approximately 74%. The performance metrics of the model are given in Table 2. We find that while the ablation of the two features 'series a funding magnitude' and the 'number of series a funding rounds' reduces performance, we retain a substantially high True Positive rate, which is upwards of 75%. This is substantially outperforming the random baseline model.

Metric	Model Result	Random model Result
True Positive Rate	.761	.371
False Negative Rate	.239	.629
True Negative Rate	.748	.653
False Positive Rate	.252	.347
Precision	.445	.498

Table 2: Table of Vanilla logistic regression performance metrics with all features.

In addition, we plot the confusion matrix for the logistic regression with reduced input features in Figure 7. An examination of model weights shows that large weights are associated with the predictors "amount of seed funding" and "time till series a". The model appears to rely strongly on these features for classification (see Figure 8).

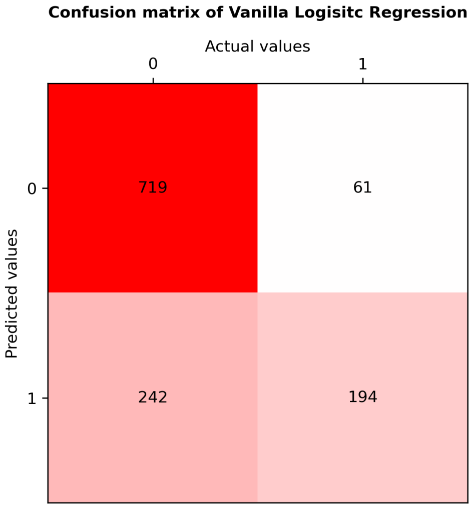


Figure 7: Confusion Matrix for vanilla logistic regression on the reduced test set with fewer features.

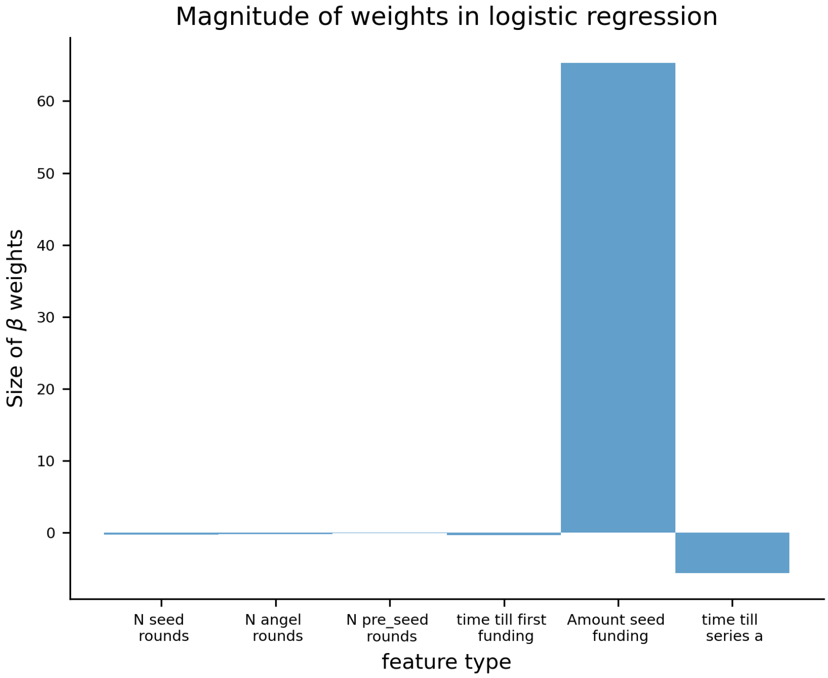


Figure 8: Magnitude of regression weights of vanilla logistic regression.

5 Deep Neural Network modeling

In this section, we show the performance of a simple deep neural network with a single hidden layer with 100 units. We train the network on the full data set and on the reduced data set as outlined in section 4. We split our training data into three parts: A training set(60%), A validation set (20%), and a test set (20%). Network activation functions for the hidden layer were Rectified Linear Units defined as $ReLU(x) = max(0, x)$

and we used a Sigmoid unit as a single output unit. The network is then defined as $\hat{y} = \sigma(\mathbf{W}^2 ReLU(\mathbf{W}^1\mathbf{x}))$ with $\mathbf{W}^2 \in \mathbb{R}^{100}$, $\mathbf{W}^1 \in \mathbb{R}^{100\times m}$ and $\mathbf{x} \in \mathbb{R}^m$. Here m is the number of input predictors, i.e., $m = 6$ or 8 depending on the task setup.

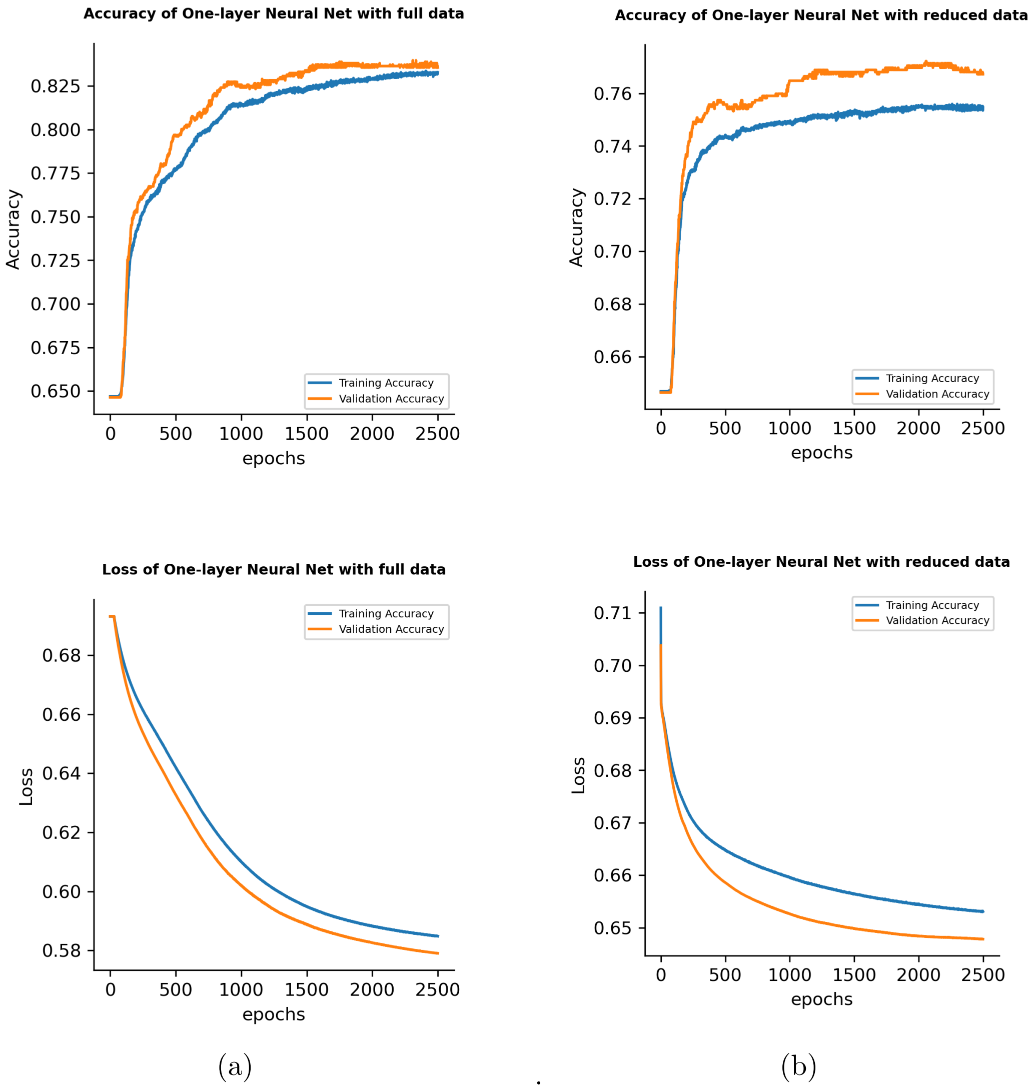


Figure 9: (a) Accuracy and Loss curves for the Neural network for training and validation set on the full data-set (b) Accuracy and Loss curves for the Neural network for training and validation set on the reduced data set

The network successfully exploits the predictors to fit the data as seen in figure 8. We see that accuracy attains a higher level with the full data set (Figure 9a/9b, top panel). However, we can also see from the bottom panel that the network did not fully converge and further training may be beneficial for performance. Please note that due ue to time constraints, we did not plot ROC curves or other metrics for this algorithm.

6 Conclusion

We find that relatively simple modeling approaches can achieve high accuracy on the classification task when provided with temporal features. Future work could further exploit the utility of Neural Networks for classification as these can find non-linear decision boundaries in the space of predictors. Another line of work might attempt to further improve these models by utilizing founder and investor characteristics as predictors.