CHAPTER 1

Evaluation

We believe it is possible to produce a Venture Capital (VC) investment screening system that is efficient, robust and powerful. In Chapter **??**, we described the development and structure of such a system. Our system is based around identifying startup companies that are likely to receive additional funding or a liquidity event (exit) in a given forecast window. This system can generate statistics and make recommendations that may assist VC firms to efficiently and effectively screen investment candidates. In this chapter, we evaluate models developed by our system against criteria of efficiency, robustness and predictive power. We discuss our findings more broadly and their implications for investors and future research into startup investment and performance.

We produced a classification pipeline which we optimised with respect to the robustness of its performance over time, and evaluated models produced by this pipeline against a held-out test dataset. This evaluation process is depicted in Figure 1.1. We use the pipeline to fit a model to a dataset sliced from the master database. We apply this model to another feature vector from the master database and make predictions. We score these predictions against truth values derived from the held-out test database (collected in April 2017). This process is performed multiple times to evaluate the three primary criteria derived from our literature review: efficiency, robustness and predictive power.

Firstly, we evaluated efficiency by exploring the learning curves of our classification techniques and whether there is sufficient data to produce reliable statistics. We also explored the time profile of our system and whether it is reasonable for use in industry, and would be likely to reduce the time currently taken to perform similar analyses. Secondly, we evaluated robustness by evaluating our models against multiple reverse-engineered historical datasets and measuring their variance. Thirdly, we evaluated the system's predictive power across different forecast windows, for startups at different stages of their development lifecycle, and for different potential target outcomes.

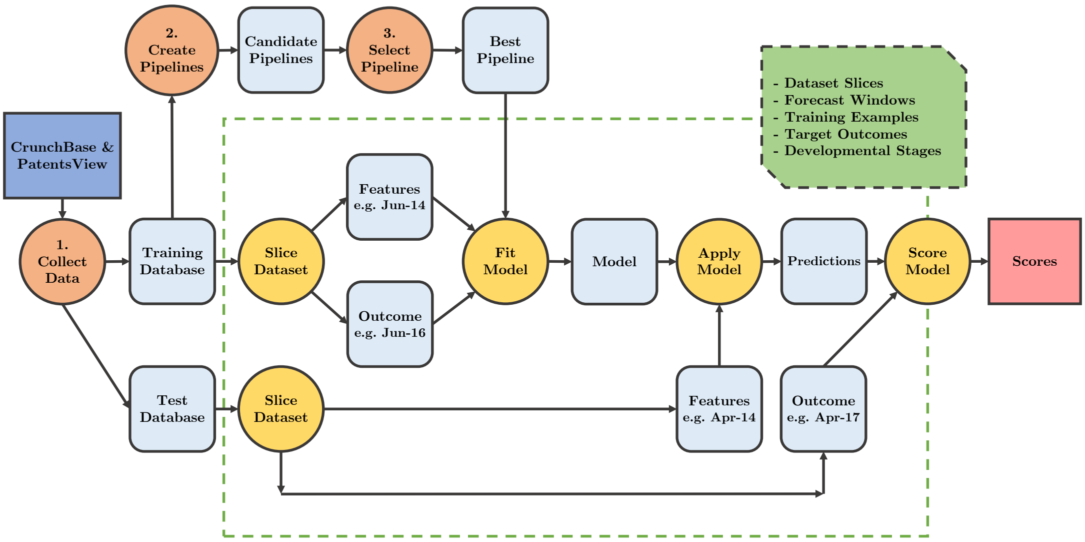

Figure 1.1: Pipeline evaluation overview.

1.1 Efficiency

The Venture Capital (VC) industry requires more efficient forms of VC investment analysis, particularly in surfacing and screening. These processes are currently performed through referral, Google search, industry papers and manual search of startup databases. By its nature, our automated system should be more efficient than these methods. In this section, we assess how efficient our system is – in terms of data consumed and time taken – and look at whether we can further improve its efficiency.

1.1.1 Dataset Size

Learning curves show how the bias and variance of a classification technique varies with respect to the amount of training data available. We decided to investigate the learning curves for our classification pipeline to determine whether we could use smaller samples from our dataset to achieve similar predictive power and reduce our need for computational power and time taken. We applied 10-fold stratified cross-validation to split our dataset into 10 subsets of different sizes which we used to train the estimator and produce training and test scores for each subset size. The rate of convergence of our training and cross-validation curves implies whether our classification pipeline is over- or under-fitting our data for various sizes allowing us to select an optimal sample size.

Figure 1.2 shows the learning curves for forecast windows of 2-4 years. The maximum number of training examples is negatively related to the length of the forecast window because newer datasets have more examples. For a forecast window of 4 years the curves have converged, whereas for shorter forecast windows there still seems to be some benefit to additional training examples. That being said, much of the testing score improvement comes in the first 20,000 training examples, which suggests that this pipeline configuration is approaching peak performance. When the pipeline creation system component is run in the future (with a larger dataset) it may tend towards a classifier with less bias and more variance, like a Support Vector Machine (SVM) or Artificial Neural Network (ANN).

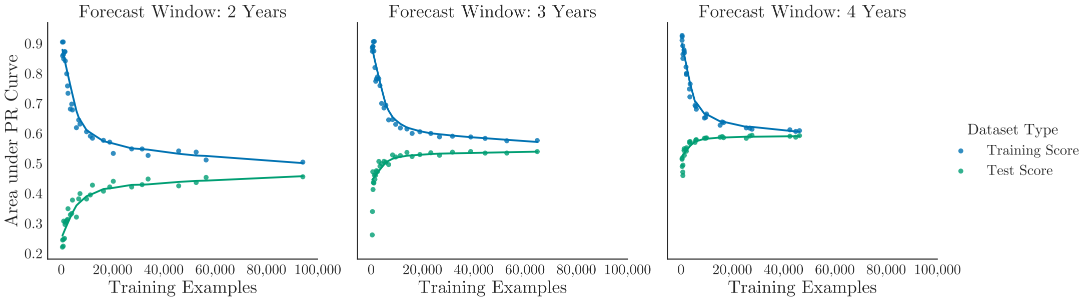

Figure 1.2: Learning curves by forecast window.

The plots in Figure 1.2 are evaluated against our base target outcome of "Extra Stage" (i.e. whether a company raises an additional funding round, is aquired or has an IPO). When we split our learning curves further by the components of this target outcome, we see that the efficiency of our system varies considerably, as shown in Figure 1.3. We observe that predicting whether a company raises an extra round is the least data-intensive outcome, as it converges rapidly even over a forecast window of just 2 years. In comparison, predicting company exits does not converge, even over a forecast window of 4 years. Our model has most difficulty predicting IPO exits, which are rare events even in our large dataset. For these target outcomes, we would epxect our system would benefit from a larger dataset. It should be noted, however, that our pipeline was optimised for predicting our base target outcome, and if the entire system was performed on the different target outcomes we might find other classification pipelines provide better performance.

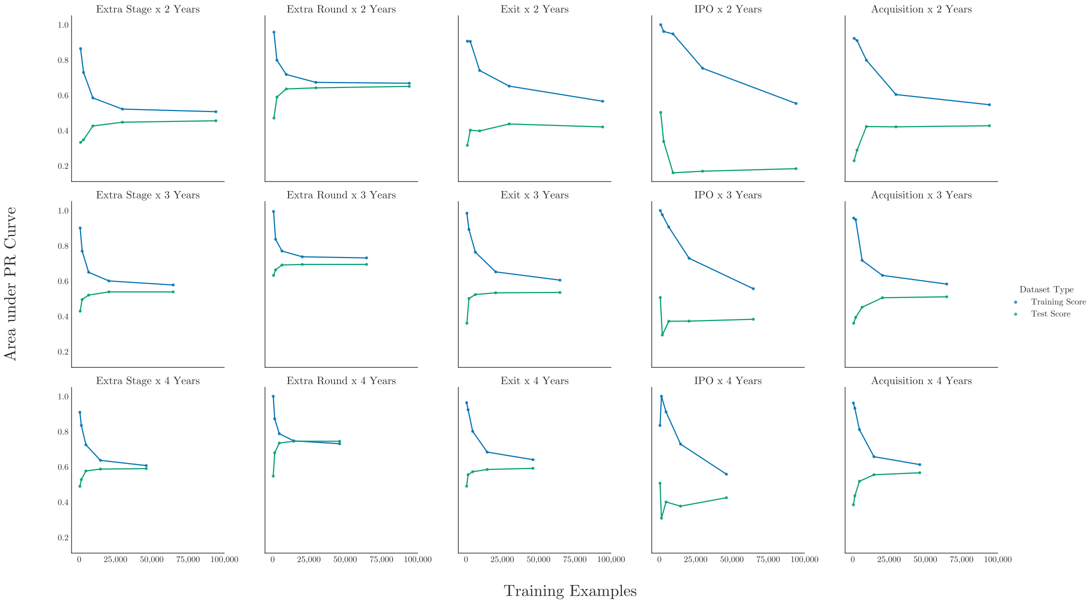

Figure 1.3: Learning curves by target outcome (column) and forecast window (row).

1.1.2 Time Profile

Unlike other forms of finance, like equity or derivatives trading, VC operates on a much longer timeframe – deals typically close over weeks, rather than minutes. This has two key disadvantages: VC firms have higher management costs because they spend more time screening investments and startup founders waste precious time negotiating with investors when they could be building their businesses. Automated systems could significantly decrease the time taken to generate investment opportunities. We investigated the time profile of our system to determine whether it is practical for use in the VC industry.

An indicative time profile of the system is shown in Table 1.1. At the highest-level, this configuration of the program takes approximately 46 hours to complete on a modern desktop PC. When we further break this time down by system component, it's clear that the vast majority of time (84.8%) is taken up by the initial pipeline creation component. This time is due to the pipeline optimisation process - the model is fit and scored over 500 times on different classification algorithms and parameters. Scoring takes a particularly long time because, in this case, it also involves generating learning curves for reporting, which is another cross-validated process. However, when placed into production, this component could be run infrequently - perhaps once per year - to ensure that the pipelines being used are still optimally suited for the dataset. The next component of the system, selecting the most robust pipeline, could occur more frequently - perhaps once every month - and the final component of the pipeline, making up-to-date predictions, could be evaluated every time new data is fed into the system (perhaps once per day) because it only takes an hour.

Function	Cycle (s)	Cycles (N)	Time (s)	Time (m)	Time (h)
Generate Dataset (CV)	1,800	1	1,800	30	0.5
Prepare Feature Dataset	1,200	1	1,200	20	0.3
Prepare Outcome Dataset	180	1	180	3	0.1
Merge Datasets	360	1	360	6	0.1
Finalise Dataset	60	1	60	1	0.0
Fit and Score Model[1]	265	525	139,125	2,319	38.6
Fit Model	15	525	7,875	131	2.2
Score Model	250	525	131,250	2,188	36.5
Subtotal: Create Pipelines			140,925	2,349	39.1
Get Finalist Pipelines	5	1	5	0	0.0
Generate Dataset (CV)	1,800	5	1,800	30	0.5
Fit and Score Model[2]	265	75	19,875	331	5.5
Select Best Pipeline	5	1	5	0	0.0
Subtotal: Select Best Pipeline			21,685	361	6.0
Generate Dataset (Training)	1,800	1	1,800	30	0.5
Generate Dataset (Test)	1,800	1	1,800	30	0.5
Fit Model	30	1	30	1	0.0
Make Predictions	5	1	5	0	0.0
Subtotal: Fit and Make Predictions			3,635	61	1.0
Total			166,245	2,771	46.2

Table 1.1: System time profile.

1.2 Robustness

The Venture Capital (VC) industry is concerned that predictive models trained on historical data will not accurately predict future trends and activity. This has been identified as a key barrier to the adoption of automated systems by the VC industry [1]. Therefore, it is critical that our system is shown to be robust in its performance with respect to time so investors can rely on its predictions.

We generated three models from datasets created from our training database from each year of 2012-2014 for forecast windows of 2 years (i.e. [2012, 2014], [2013, 2015], and [2014, 2016]) and evaluated each model against a dataset created from our test database (i.e. [2015, 2017]). We expected that if the factors that predict startup investment success through time are consistent, we would observe little difference between the performance and characteristics of these models. Figure 1.4 shows the coefficient of variation of models trained on dataset slices from different years, against key evaluation metrics. The coefficient of variance is the ratio of the biased standard deviation to the mean. This produces a standard measure of variance, so different evaluation metrics are comparable. We have also grouped by forecast windows as later dataset slices cannot be tested with long forecast windows which skews results along this dimension. Variance across all metrics is very low, with slightly more variance over shorter forecast windows, as one would expect. We explored the feature weights for each model in Figure 1.5. While there are some slight differences, the general trend is very similar across all models. We will discuss the distribution of these feature weights in more detail in a following section. All of these results suggest that our system generates models that are robust with respect to time.

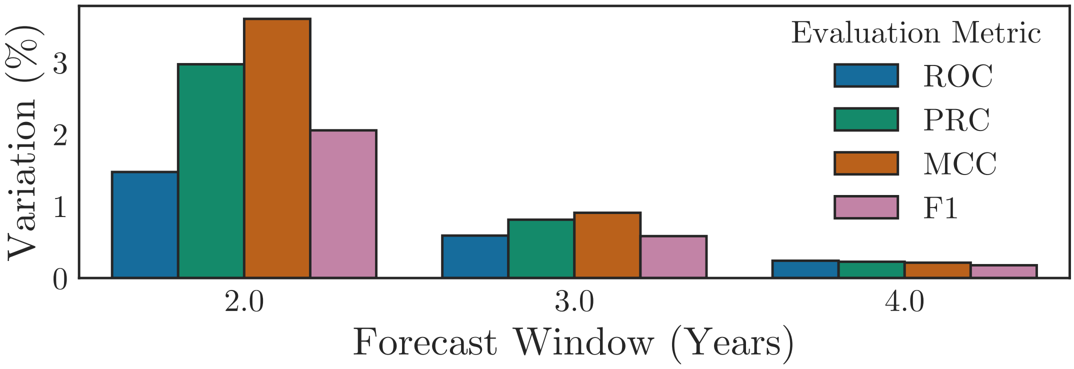

Figure 1.4: Performance variation by slice date.

1.3 Predictive Power

Our system must be consistently accurate at identifying a variety of high-potential investment candidates. We evaluated the systems' predictive power based on its ability to predict over different forecast windows (e.g. 2-4 years), for target companies at different developmental stages (e.g. Seed, Series A etc.), and for different target outcomes (e.g. predicting additional funding rounds, being acquired, having an IPO, or some combination thereof).

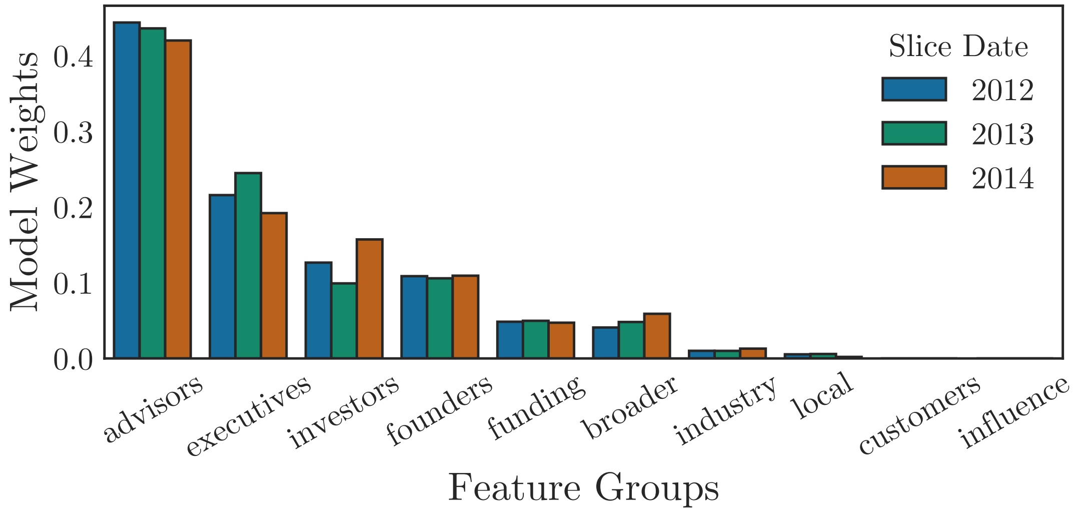

Figure 1.5: Feature weights by slice date.

1.3.1 Preliminary Analysis

Before we evaluated the predictive power of our system, we performed preliminary analyses to determine the baseline trends and distributions of company outcomes in our database.

First, we looked at company outcomes by forecast window. We applied the same system of reverse-engineering time slices that we used in our previous experiment on robustness, but this time we vary the time difference between the slice that provides our features and the slice that provides our outcome. We combined pair-wise datasets of each year from 2012-2016 inclusive and explored the proportion of companies that raised additional funding or exited. Figure 1.6 shows how company outcome varies with respect to the forecast window (time between the observed features and the measured outcome). Intuitively, we see a positive relationship between length of forecast window and company outcome. In particular, very few companies appear to have exited or raised funds over a period of less than 2 years so we will focus our experimentation on forecast windows of 2-4 years.

We also looked at how company outcomes vary with respect to development stage, shown in Figure 1.7. We see a broad positive relationship between developmental stage and likelihood of further funding rounds and exits, which we would expect as at each stage there is higher market traction and scrutiny from investors. The variance between the outcomes of different developmental stages suggested that in our experimentation we should investigate how our system pre-

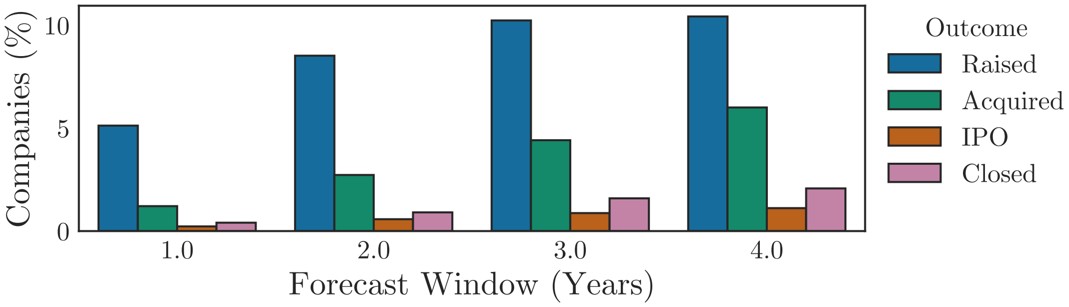

Figure 1.6: Outcomes by forecast window.

dicts each stage independently, as well as in aggregate, as we do in a following section.

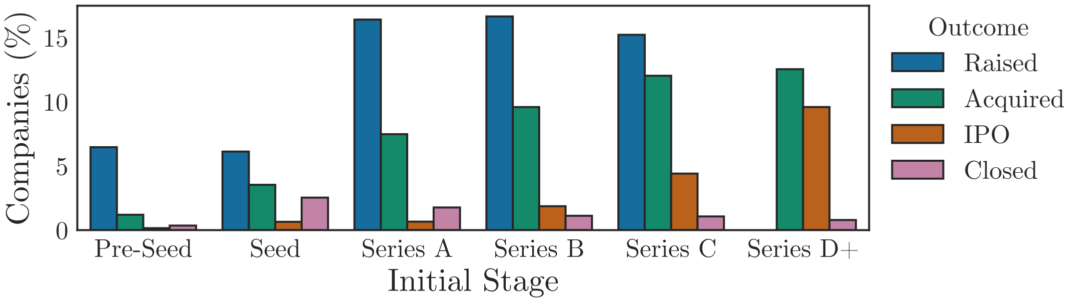

Figure 1.7: Outcomes by developmental stage.

1.3.2 Evaluation Metrics

1.3.3 Forecast Windows

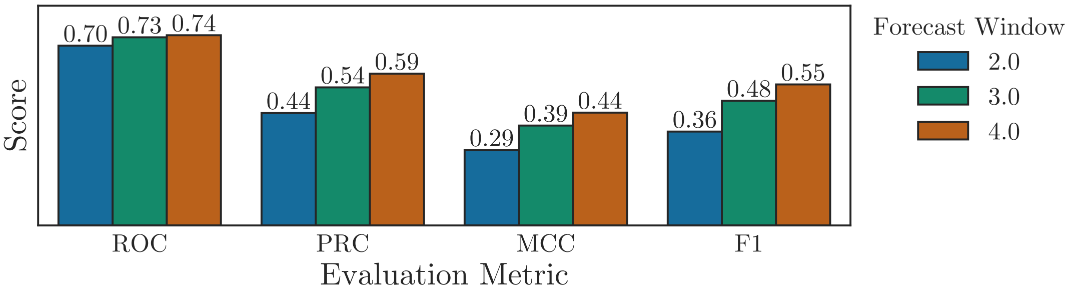

Figure 1.8: Performance by forecast window.

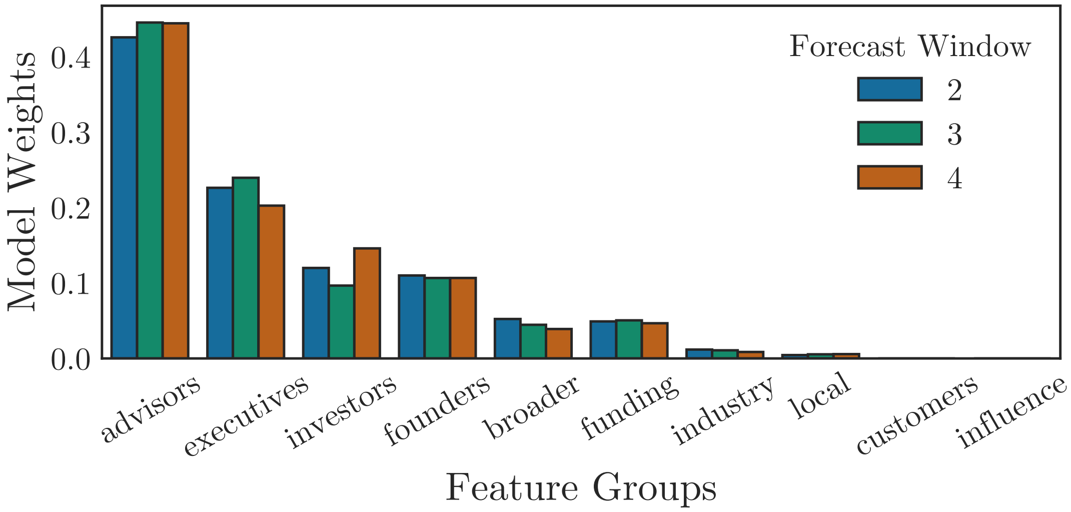

Figure 1.9: Feature weights by forecast window.

1.3.4 Development Stage

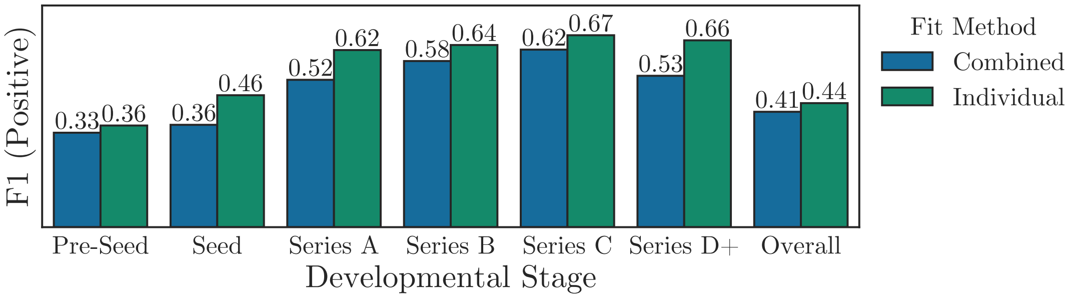

Figure 1.10: Performance by developmental stage.

1.3.5 Target Outcomes

1.3.6 Conclusions

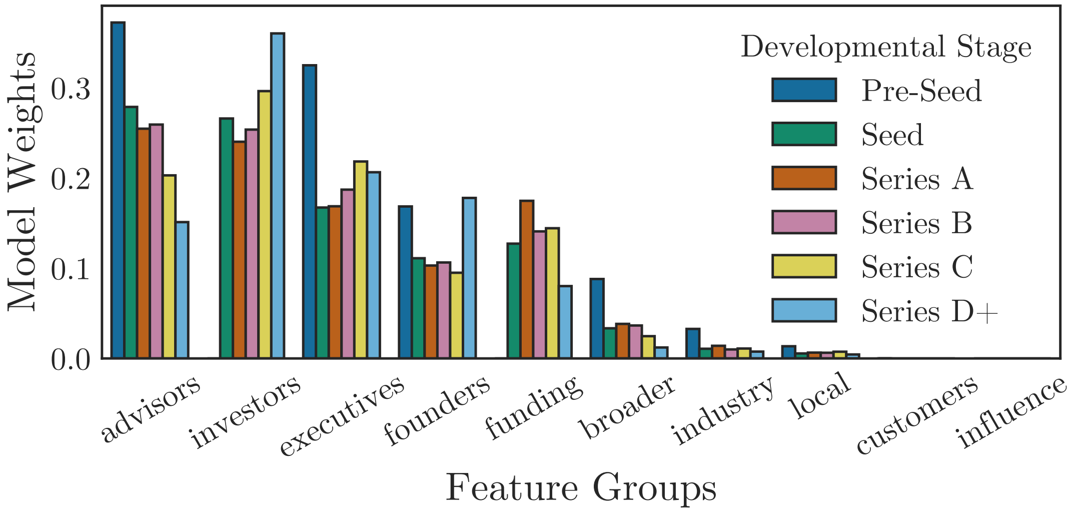

Figure 1.11: Feature weights by developmental stage.

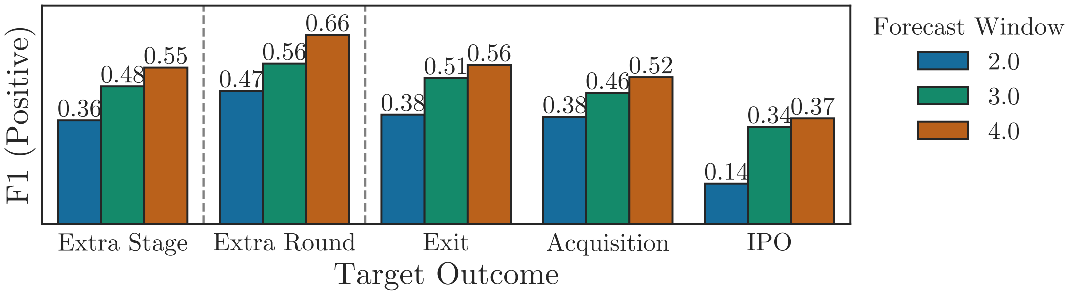

Figure 1.12: Performance by target outcome.

	Company			
Feature	ChaCha	Doctor.com	Fab	Mixpanel
Age (Years)	7.4	0.6	4.3	3.8
Funding Raised ($m)	92.0	0.0	171.0	12.0
Funding Rounds (N)	8	0	8	4
Feature Stage	Series D+	Pre-Seed	Series C	Series A
Outcome Stage	Series D+	Series A	Acquired	Series B
Predicted Outcome	✓	✗	✓	✓
Actual Outcome	✗	✓	✓	✓
Correct Prediction	✗	✗	✓	✓

Table 1.2: Company profiles and predictions.

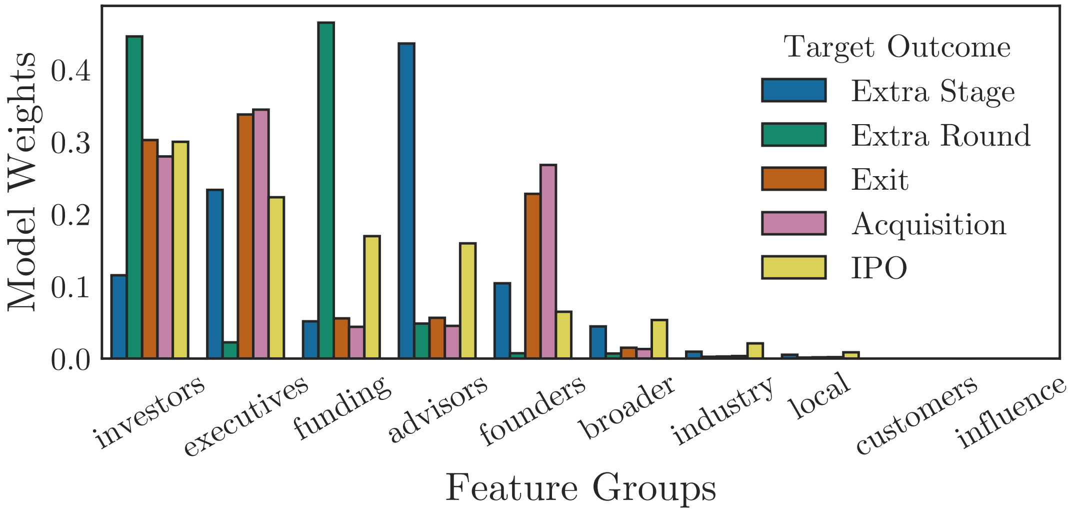

Figure 1.13: Feature weights by target outcome.

Bibliography

[1] Stone, T. R. "Computational analytics for venture finance". PhD thesis. UCL (University College London), 2014.