Universiteit Leiden

Master Computer Science

Modelling Startup Growth and Spending by using Markov Decision Processes

Name:	Maurits de Groot
Student ID:	s1676784
Date:	03/12/2021
Specialisation:	Artificial Intelligence
Specialisation:	Advanced Data Analytics
1st supervisor:	Dr. Xishu Li
2nd supervisor:	Dr. Alexandru Uta

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Modelling Startup Growth and Spending by using Markov Decision Processes

Maurits de Groot

[a] *Leiden Institute of Advanced Computer Science (LIACS), Niels Bohrweg 1, Leiden, 2333 CA Leiden, South Holland, The Netherlands*

Abstract

Startups, which are companies of a scalable business model, often operate in an uncertain and volatile environment. Because of its business model, a startup has the potential to grow exponentially rather than linearly. However, taking a wrong decision at a time could lead to the end of the firm. In 2021, 9 out of 10 startups fail. In order to grow, startups often resort to data analytics. In this research, we study whether more spending on data analytics can foster startup growth and what type of data analytics capability should a firm purchase at each time. We use Markov decision processes to model growth and spending and derive the optimal policy. We apply our method to a dataset from Amazon Web Services, which contains billing information from 10s of thousands of startups. We find that increasing data analytics capabilities can facilitate the growth of a startup. We also compare our optimal policy with the realized spendings of the best-, average-, and worst-performing firms and find that the actions of the best-performing firm highly correspond with the optimal policy, whereas both the actions of the average-performing firm and those of the worse-performing firm are far off from the optimal policy. Based on our results, we provide practical suggestions to firms on purchasing data analytics capability.

1. Introduction

The amount of startups is growing, both in terms of numbers [11] as well as in venture dollar volume [9]. This growing industry has become increasingly important. Startups can be characterized by a high volatility, rapid development pace with in most cases a focus on new technologies. In this highly competitive industry it is evidenced that failure rates of startups can be as high as 90% [1]. For startups to compete they often leverage services of existing companies and build upon these. Examples of this are using accounting software for their bookkeeping, a payment processor to handle payments or renting office space. For startups involved in the technology space the use of a cloud technologies is a common practise. The success of third parties is aligned with the success of the startups themselves since growth of the startups will lead to more consumption of the external offerings. For the startups themselves it is difficult to approximate the ideal action to take, in order to use these external offerings in the most effective way. Third parties possess a more holistic overview of the effect of different actions that startups can take. Due to this information asymmetry finding the best action for a startup to take in terms of using a partner offering is hard. In this research we will use information of a cloud platform provider to model the ideal steps a startup can take. When we discuss startups it's important to follow a concise definition what we consider to be a startup for this research. For this research we will define a startup as follows:

Startups are privately-held scalable companies, potentially backed by an angel investor fund/venture capital/private equity, that are yet to merge, be acquired by another company or have an initial public offering.

For startups it's difficult to be aware what the best action is to take at any point in time. This overview is only available for companies that service a multitude of startups. Due to this information asymmetry startups can make suboptimal choices. In a rapidly changing environment past decisions do not provide any certainty with regards to the current situation. When modelling startup behaviour we will encounter unobservable situations. Not every move made by a startup is observable, event such as personal situations or political changes are not reflected in the dataset. This makes the problem a stochastic one. Problems, such as this one, which can be considered both stateless and stochastic follow a Markov property. This property indicates a stochastic process where the conditional probability distribution of future states is depended on the present state, not the previous states [15].

Cloud vendors have a unique position relative to their customers. They are the only party that have an overview of what the effects of different actions of their customers are. This is especially true for startup customers, because of the unique tech capability cloud provides to scale startups at an unprecedented rate due to its managed services offering. Having this knowledge about these companies might enable cloud vendors to help the development of startups by stimulating their growth by analysing the cloud consumption.

"Can increased data analytics consumption help startups grow and how?"

This question will be tested by comparing specific actions startups can take and comparing such actions with the optimal actions a startup can take. These optimal actions will be derived from a historical dataset. If increased data analytics consumption has a positive effect on the growth of startups. Then we will focus on how this information can be used to support the growth of a startup. In order to do so, access to a wide variety of startup consumption data is needed. By doing this research in cooperation with a cloud vendor an anonymized version of such a dataset is available. Having access to such a dataset allows this research to answer the question stated above. The available data consist of the billed consumption by category. Researches do not have access to the personal information about customers, itemised billing data, configuration or customer data. In this research we use a Markov Decision Process (MDP) model. In the model startups are represented as different states. Every state a startup can be in is associated with a reward and a set of actions startups can do from that state. Doing so frames the problem as a reinforcement learning problem where historical data can be leveraged to calculate the optimal actions for startups depending on the state they are in.

The structure of this work will be the following. First, we elaborate on the literature underlying the MDP model used in this work in Section 2. After that the used data, as well as the implementation of the MDP model, is described in Section 3. The results of this method are presented in Section 4. After that we provide actionable advise in Section 5. In Section 6 the results are discussed, ideas to expend upon this work are presented and we conclude this work.

2. Literature review

Startup growth is a well researched topic across multiple academic disciplines. The field of operational research and information systems aim to improve the understanding of growth in startups. Both have a different perspective on how to improve the growth of startups. Before we dive into this literature we need to be able to measure growth itself. In a study of Rompho [18] 110 startups have been surveyed in order to investigate the performance measurements used within startups. These performance measurements are often an internal indication of the growth of a startup. This study identifies that one of the challenges of startups is efficient allocation of resources, which could influence the growth pace of the startup itself. According to Coad [7] the growth of a firm relates to the decision making process. For the decision making process a lot remains to be researched, a study from Shepherd et al. [22] outlines this by setting a research agenda. In recent time the growth of startups is accelerated due to the COVID-19 pandemic (Kuckertz et al. [14]).

In the field of operational research startup growth is studied extensively. For example, in a study conducted by Wennberg et al. [24] startup growth is studied by focussing on the decisions a startup takes compared to the risk it involves. By adding the component of risk a more nuanced view is offered. This study emphasises the importance of age and size of the startup when looking at the survival rate. A different approach with a focus on growth paths and survival chances is taken in a study by Coad et al. [8]. In this study a Gambler's Ruin model is used to predict the chances of survival in the next period. The actions of startups are approximated by a random walk. The actions a startup can take as described in the last two studies are dependent on sufficient monetary backing. In the first study hiring new employees, and thus increasing the startup size, is dependent on sufficient funding. The second study takes actions in a random fashion. In a study from Shakya and Plemmons [21] the impact of economic freedom for startups is studied. Having an ample amount of funds is vital to the survival. Most of the time startups do not have such funds to begin with and need to source those. Sourcing funds can be done in a number of ways, each with its own implication on the growth of the startup. In the work of Schwienbacher [20] different financial strategies are compared for capital-constrained entrepreneurs. Making capital decisions and allocating resources effectively is key to the growth of startups.

In the field of information systems different approaches are considered to support the decision making process. A survey that summarizes the process of decision making given a high degree of uncertainty is written by Keith and Ahner [13]. Having a lot of uncertainty is common for startups. In this survey the Markov decision process is presented as an suitable model to use when dealing with uncertainty. This model has been used for multiple startup goals, one example of this is given by Archibald et al. [3], in their paper they propose to use a Markov decision process model to support the management of inventory and production capacity. Leveraging analytical capabilities to improve the growth of startups is becoming increasingly more common. In a study from Behl et al. [5] the effect of big data analytics for startups is presented on a case-based approach. This study finds that the use of this analysis has a significant contribution on the growth of a startup. In a paper from Dellermann et al. [10] artificial intelligence techniques are used to identify which startups will become successful, indicated by the label “unicorn”. Note that this label is used for the last phase of a successful startup, little research is conducted on the intermediate steps. Trying to identify successful startups early on is often done. By doing so some best practices could be abstracted. The study from Dellermann et al. creates a model based on the current state of the startup and is used as a support tool for investors. Once an investor is interested in a startup an agreement is formed. In a study from Archibald and Possani [2] the contract between investors and startups are modelled using a Markov decision process approach. The end goal of this paper is to find the most desirable outcome of the negotiation.

In the existing literature a considerable amount of work is done related to startup growth. Both in operational management research and information systems research. Most operational approaches are focussed on growth by describing the actions a startup can take and aiming to find an optimal action. The information management side shows us that data analytics has potential of improving insights. A number of proven techniques, such as the Markov decision process, have been used successfully in different applications. When looking at the actions startups can take, one of the subjects that have yet to be addressed is how the more information systems focussed analytical models influence the growth of startups. Our research will bridge this gap by combining both research streams to discover if increased data analytics consumption helps startups grow and if so, providing an explanation how.

3. Methodology and Data

In this section we’ll first elaborate on the MDP model by setting the variables and afterwards proposing a way of solving it. This will result in the optimal policy to use. After exploring the model we describe the dataset used for this research.

The research question is a typical reinforcement learning (RL) problem. A decision maker interacts with its surrounding environment, which in return provides rewards and a new state based on the actions of the decision maker. Before formulating the problem mathematically by using a Markov decision process a theoretical understanding of RL is required. When discussing RL the definition of Sutton and Barto will be used, this is the following Sutton and Barto [23]:

> "Reinforcement learning problems involve learning what to do - how to map *situations* to *actions* - so as to maximize a numerical *reward* signal"

Within this definition a number of key aspects are present. The first one being the situation of the problem. Often these situations are referred to as different **states** (S) of the model, here a state denotes the current situation. Let's imagine that we have a light switch. When modelling how this switch works we can identify two different states, the first state being 'on' and the later being 'off'. To change between states an **action** (A) is required. In the example of the light switch we could define actions as (1) 'press the button at the bottom' and (2) 'press the button at the top'. The first action will turn on the light if it was off before and vice versa for the second action. Note that an action does not always change the state, i.e. if the light is already on and the first action is executed the light will stay on, and thus will not change states.

The next aspect of RL is the numerical **reward**, this reward is associated with an action that facilitates a transition between two states. The formal notation of this is $R_a(s, s')$. This denotes the reward followed by action a after going from state s to state s'. If a state has a high reward value that state is a favourable state to be in. As stated in the definition above the reward value is to be maximized. Knowing which action to take from every state is called a **policy** (π). Creating a good policy will result in high reward values and is therefore often the end goal of most RL problems.

When trying to solve a RL problem one can differentiate between two types of problems. Some problems are fully observable, here all states and possible actions are known. Other problems are partially observable. In this case a subset of states and actions are known. Another characteristic of RL problems is whether the problem is deterministic or stochastic [19] . In deterministic problems any given action from a state will always result in the same outcome. A good example of this would be the light switch described above. For stochastic problems actions can result in different outcomes. You can think of this as a coin flip that could end up at the state 'heads' or 'tails'.

When a problem has a stochastic element the concept of probability is introduced. The formal notation of probability is $Pr(s'|s, a)$. This denotes the chance that action a will result in a transition from state s to state s'. A mathematical framework which is often used for decision making when the problem contains a stochastic element is the Markov Decision Process (MDP) [6] [16]. This model is formally depicted as (S, A, Pr, R_a). The goal of an MDP is to find the optimal policy. This policy will state the best action to take from every state, which makes it an ideal tool for decision making. One method to solve the MDP is called "value iteration" [6]. This is a recursive method that is guaranteed to converge to the optimal policy [17].

In this work we classify startups in four categories based on their consumption of cloud resources. By doing so we get an approximation of how well a startup is doing. The idea behind this is that the startups that have more funds to spend have more funds disposable and are thus doing better. This only holds under the assumption that money is spend in an effective way. When classifying startups we look at the consumption per month. This allows for startups to be classified differently on a monthly bases. The categories are the following:

- **Zero** : Startup does not spend any funds above the allocated free tier.
- **Low** : Spends is between 0 and 1.000 in the current month.
- **Mid** : Spends is between 1.000 and 10.000 in the current month.
- **High** : Spends is more than 10.000 in the current month.

Every startup has a consumption pattern for their cloud services, this pattern consists of the distribution of services used. Within the dataset used for this research we've identified 206 distinct services. The distribution of service consumption follows a power law, meaning that a small quantity of services make up for the most usage measured in dollars. Due to this property the consumption behaviour of startups can be captured by categorizing it as one of the following categories:

- **Compute** : Services related to computing.
- **Database** : Services that facilitate a database.
- **Datatransfer Services** : services that move data around.
- **Networking** : Services that facilitate networking infrastructure.
- **Content Delivery** : Services that serve as a content delivery network.
- **Other** : All services not listed above.

The problem we're solving in this work is one that follows the Markov property. This problem can be described as a stochastic problem. In this case we define the problem as one of full observability. For the Markov Decision Process (MDP) we have a collection of states (S), actions (A), probabilities (P_a) and rewards (R_a). In this section each part is described in detail. In 3.1 the state will be formally defined. After that in 3.2 possible actions for the startups will be explained. The probabilities for these actions are discussed in 3.3 and corresponding rewards can be found in 3.4. Finally a method to calculate the optimal policy is given in 3.5.

3.1. State Definition

The definition of state defines in what situations a startup can be in. For this state definition the consumption amount of a startup is used. This is indicated by the `spend_per_month` variable in our dataset. When picking a state it is crucial that states are mutual exclusive, this is the case when looking at different consumption amounts. In the data we have identified four different states. These stages, as well as the notation we'll use for them, can be found in Equation 1.

$$S_{consumption} = \begin{cases} s1 & consumption = 0 \\ s2 & 0 < consumption \leq 1.000 \\ s3 & 1.000 < consumption < 10.000 \\ s4 & consumption \geq 10.000 \end{cases} \tag{1}$$

Besides the consumption the state is dependent on time t as well. For this research we have data from 12 months, each iteration is equal to 1 month. This results in a state vector t with length 12, or formally $S_t = \{0, 1, ..., 11\}$. The formal definition of the states we defined denote that our set of states (S) can be described as $S = \{S_t, S_{consumption}\}$. Each combination of S_t and $S_{consumption}$ is a valid state a startups can be in. This gives a total of 48 states. Besides these states we also need to define how a startup can be removed from the model. In our case we define two conditions. The first one is that a startup can be merged, be acquired or become publicly listed. If this is the case we do not consider this company to be a startup anymore. When this occurs we'll can this a graduate. The second way of leaving the model is when a startup does not have sufficient funds to continue operating. This is the case of bankruptcy. When a startup goes bankrupt the company will be dissolved and thus will not be a company anymore.

3.2. Action Definition

Startups allocate funds in different ways, the way a startups chooses to do so influences the growth of a startup. In most cases startups use a multitude of services, however, which service is used most is singular. Selecting this most common service is an action startups could take to influence the growth. We take the most used service that a startup opts to use as an action, here we have a set of actions (A) which is selected for every month. The different actions are defined in Table 1. This set of actions is formally defined as $A = \{a1, a2, a3, a4, a5, a6\}$. The most common action is based on the total spend of the services in dollars. In the unlikely event that multiple services have identical spend a service is selected randomly from the tied services.

Most used service	Notation
Compute	$a1$
Database	$a2$
Datatransfer Services	$a3$
Networking	$a4$
Content Delivery	$a5$
Other	$a6$

Table 1: Possible actions startups can take based on service usage

3.3. Probability Definition

For the probability we want to know the chance that we move from state s to state s' as a result of action a. This can also be described as $Pr(s'|s,a)$. Since we have a fully observable environment we can calculate the probabilities between states. To calculate the probability that a startup will go from state s to state s' we do the following. We take all startups by `startup_id` that occur in a given state at time t and time $t+1$. These startups can do one of six actions a. For every action we look at the amount of startups that, as a result of action a, transfer from state s to state s'. This number is divided by the initial number of startups. For example, let's imagine that we want to calculate the probability that a startup transitions from state $s3$ to state $s4$ as a result of action $a1$. In this example we have a total of 100 startups that are both in t and $t+1$. After doing action $a1$ we observe 30 startups in state $s4$. This would mean that the $Pr(s4|s3,a1) = 30\%$.

For our two termination possibilities we cannot calculate this based on data since this would require external data. Linking this data based on only an identifier, without any name, is not possible. To get an approximation of these probabilities we have consulted domain experts, the assumptions derived from this can be found in Table 2. Another way of getting an estimate

State	Graduation probability	Bankruptcy probability
$s1$	0.0001	0.70
$s2$	0.0002	0.52
$s3$	0.0004	0.34
$s4$	0.0008	0.16

Table 2: Probability of graduating or going bankrupt from every state based on domain knowledge.

would be consulting literary sources such as [12], however, by doing so we would neglect that our dataset consists of a subset of technology focused startups that leverage a specific cloud provider. An example of a transition between states is shown in Figure 1. Here the start state is $s1$. By taking action a the startup could end up at one of 6 different outcomes in the next iteration. Each outcome has a probability (Pr) associated with it.

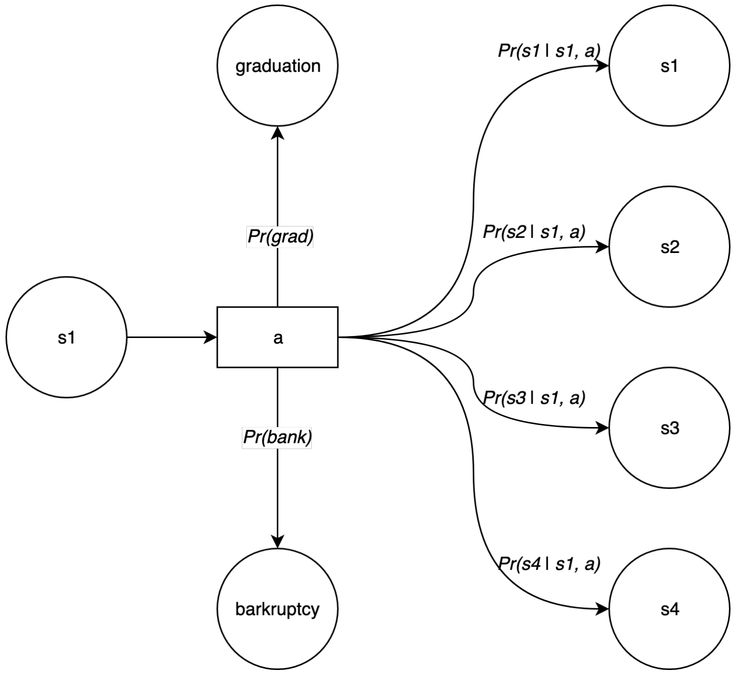


Figure 1: Markov chain that models the transition between state $s1$ to states $s1, s2, s3, s4$ by taking an action a

3.4. Reward Definition

The reward value of a state should reflect the value of a startup at the current point in time. Since no external information about the startups is available we will approximate this value by taking the spending per month as a proxy of the value of the startup in that month. Cloud platforms are scalable, due to this property we assume that if the consumption of a startup is higher the startup itself is doing better as well. Some outliers will occur where this does not hold, but since we are averaging over a large number of startups this proxy will still be valuable. The resulting values have a wide range of rewards, this is unfavourable when creating an RL model. To solve this problem the outcomes will be scaled as described in Equation 2. After we have scaled our rewards we normalize them in order to start at 1.0. This is done by taking the minimal scaled value and adding 1 to the absolute value of the minimal scaled reward.

$$z = \frac{(x - \mu)}{\sigma} \tag{2}$$

Here, z is the scaled output for the unscaled rewards x. The μ symbol is the mean value of the spending per month. One standard deviation of this distribution of spending is given by σ. This scaled value is calculated per state, this gives rewards R. For example, let's assume that we have the unscaled rewards (x) $1, 2, 3, 100$. In this case we have a mean (μ) of 26.5 and a standard deviation (σ) of 16.96. The first unscaled value can be calculated by $\frac{(1-16.96)}{26.5} = -0.601$. The same can be done for the other rewards to get the scaled rewards (z) $-0.601, -0.577, -0.554, 1.732$. These rewards will be normalized by adding the absolute value of the lowest scaled reward + 1. The lowest scaled value is -0.601, which means that to every value we will add $|-0.601| + 1 = 1.601$. This gives us the normalized scaled rewards $1, 1.024, 1.047, 3.333$.

3.5. Policy

The policy (π) will dictate which action should be taken from any state. In order to find the optimal policy we need a method that could be used to compare different policies. To do so we use an utility function (U). The utility function aims to calculate the expected net present value (NPV) of the firm at time period t given the series of actions from period t until the end of the planning horizon. The goal of this function is to optimize this expected net present value by choosing the optimal sequence of actions. A simple way of evaluating the effectiveness of a policy is to take the sum of accumulated rewards R starting from state s when taking optimal actions for a timespan of T steps. This would result in the utility function found in Equation 3.

$$U([s_0, s_1, \cdots s_T]) = \sum_{t=0}^{T} R(s_t) \tag{3}$$

This would work if we had a limited horizon set for T, however, in our case we assume that T is indefinite since we're interested in the best long term results. When doing so we have an infinite amount of steps. As a result of this U will always be infinite. This will make it impossible to compare different policies. To solve this problem we need to alter our definition such that the utility function U is the discounted sum of rewards accumulated when starting from state s. This can be done by introducing a discount factor γ. Our new utility function is given in Equation 4.

$$U([s_0, s_1, \cdots s_T]) = \sum_{t=0}^{T} \gamma^t R(s_t) \tag{4}$$

Here $\gamma \in \big(0, 1\big]$. The value for γ cannot be 0 since every power of 0 is equal to 0 which would result in a utility that is 0 as well. Doing so would result in the same situation where every output of the utility function is the same and no comparisons can be made. When the same behaviour as Equation 3 is desired γ can be set to 1 since 1 to the power of any n is still 1, rendering the additional parameter useless. For any value in between it holds that higher values for γ favour more long term effects compared to lower values for γ. Under the assumption that T is infinite we can rewrite the utility function as the utility of the policy π for state s as stated in Equation 5.

$$U^{\pi}(s) = E_{Pr([s_0, s_i, \cdots]|s_0 = s, \pi)} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \right] \tag{5}$$

We're seeking the optimal policy from starting state s, denoted with $\pi^*(s)$. In this policy the utility function should be maximized. For this policy we want to select the action that yields the highest utility. This is described in Equation 6.

$$\pi^*(s) = \underset{a \in A(s)}{\arg\max} \sum_{s'} P(s'|s, a) \cdot U(s') \tag{6}$$

Solving for the optimal utility function can be done by solving the Bellman equation [6]. This equation is shown in Equation 7. This formula consist of two major parts. The first part is $R(s)$ which states the reward for state s. Being in a state contributes to the reward of that state and thus to the utility. The intuition of the second part is that future utilities contribute to the current utility as well. For every action that one could take the probability of that action resulting in state s' is multiplied by the utility of state s'. This part is discounted by the discount factor γ. The values for $R(s)$, γ, A and $P(s'|s, a)$ are given in our MDP model. To solve this equation the value for $U(s')$ is required as well. This value can be calculated recursively by employing the value iteration method, the formula for this is given in Equation 8.

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) \cdot U(s') \tag{7}$$

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) \cdot U(s') \tag{8}$$

This method has a close resemblance with the Bellman equation itself. The utility of the next state is recursively set. This will continue until the stopping condition is met, this is when the maximum change in the utility of any state in an iteration is smaller than ϵ. Another way of representing this is by displaying the pseudocode of this algorithm. One can find the code for this in Algorithm 1. Inputs for this algorithm are the MDP model as a collection of states (S), actions (A), probabilities (Pr) and rewards (R_a), the discount factor γ and the maximum change in the utility of any state in an iteration ϵ.

Algorithm 1 VALUE-ITERATION(MDP, γ, ϵ)

repeat
  $U \leftarrow U'$
  $\delta \leftarrow 0$
  for each state s in S **do**
    $U'[s] \leftarrow R(s) + \gamma \cdot \max_{a \in A(s)} \sum P(s'|s,a) \cdot U[s']$
    if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
until $\delta < \epsilon \cdot (1 - \gamma)/\gamma$
return U

In this research we used the MDP model as described throughout this section. As a start state we opted to use the a startup with no spending $s1$ since most startups start this way. The value for gamma (γ) used is 0.9, this value will favour long term effects. The maximum allowed change, or epsilon (ϵ), is set to 1 trillionth or $1 * 10^{-12}$. By taking a small value for ϵ we will not stop prematurely.

The work in this research is build upon a dataset consisting of startup cloud consumption provided by the cloud platform provider Amazon Web Services (AWS). The dataset contains transaction specific data as well as aggregated spend per service category and total spend per startup. The transactional data consists of an anonymized unique identifier for every startup, the time and date of the transaction as well as the monetary amount involved in that transaction. For the Aggregate data the total spend is calculated both per month and per year. Table 3 provides a description of the dataset. Researches do not have access to the personal information about customers, itemised billing data, configuration or customer data.

Type	**Attribute name**	**Description**
Transactional	startup_id	Unique identifier of a startup
	timestamp	Timestamp of the transaction
	consumption_amount	Amount involved in the transaction (USD)
	service_group	The category of the service
Aggregate	spend_per_month	Total spend of a startup per month
	spend_per_year	Total spend of a startup in a year
	most_used_service_per_month	Most used service per month in USD

Table 3: Attributes in startup consumption dataset

The dataset contains 10s of thousand startups located in Europe, the Middle

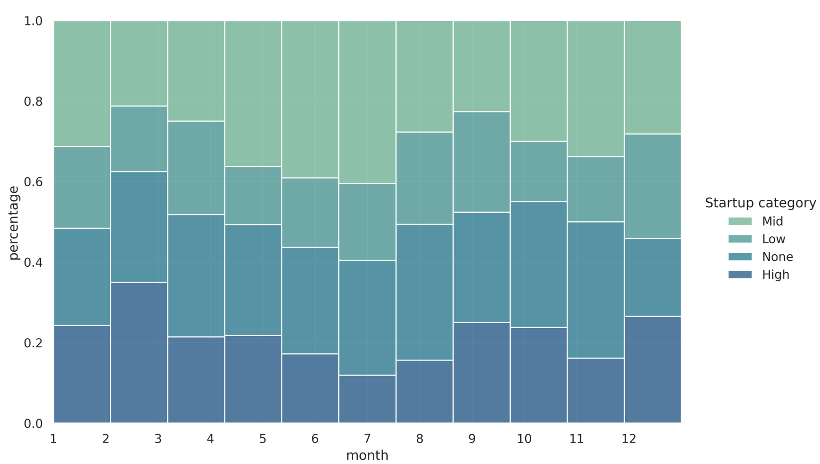


Figure 2: Percentage of startups category per month. Based on a random sample of 1000 startups

East and Africa (EMEA). The amount of transactions involved is well into the millions. This occurs over a timespan of 12 months, starting on June 1st

2020. The distribution of the startup categories is not uniformly distributed in the dataset. This distribution changes throughout time. This is displayed

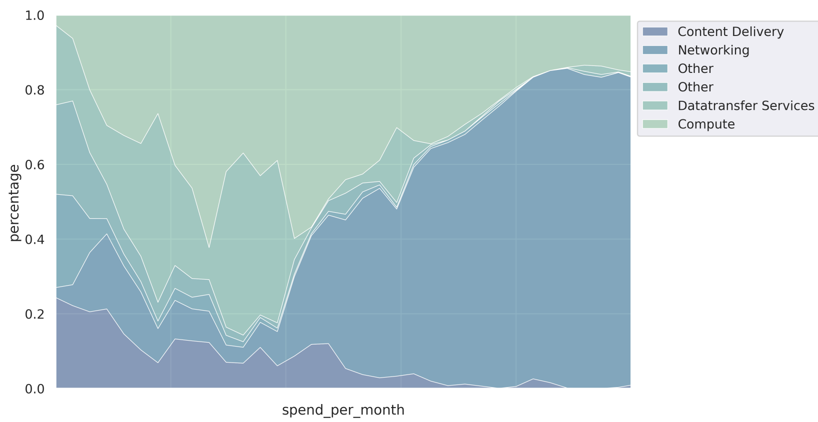
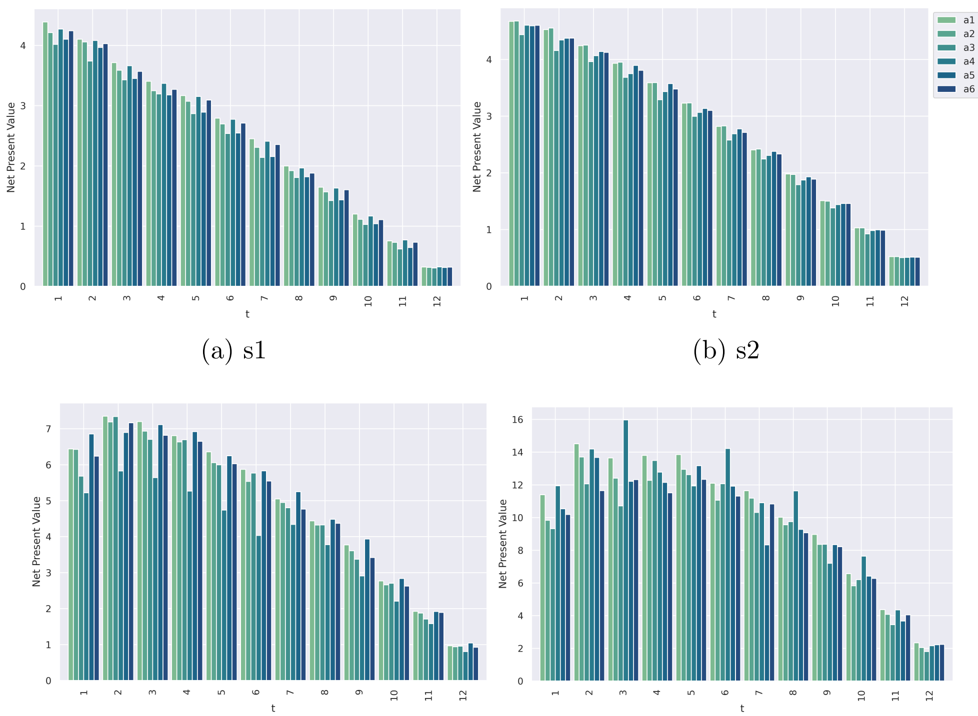


Figure 3: Percentage of spend per month relative to the type of a startup. Based on a random sample of 1000 startups

in Figure 2, here one can see that throughout the year every category is represented. Although the distribution chances for every point in time the smallest proportion is larger than 15% and the largest portion is smaller than 40%. Note that this figure is based on a random sample of 1000 startups to mask the exact distribution. Different types of startups have distinct consumption behaviour. This is illustrated in Figure 3. When looking at this behaviour we observe that in early phases, or low spending per month, the most common services are distributed quite evenly. Later on, towards the right side of the figure, networking becomes the most common service based on consumption in dollars, followed by compute. This figure is based on a random sample of 1000 startups as well.

4. Results

By applying the Markov Decision Process model as described in section 3 we will calculate the value of different actions and policies that startups can take. In the model we use an utility function to approximate the net present value of a startup at time t, under the assumption that the goal is to optimize the expected net present value. In this section both the Net Present Value of actions and policies are elaborated upon. Different actions yield a different net present value, this value is different for the state a startup is in, this is displayed in Figure 4. In this figure the y-axes show the net present value and the x-axes display the time t. Over time the value of the 6 actions a startup can take are displayed. Within this figure four different sub-figures show the states of a startup. From Figure 4 some observations can be made. The first



(a) s1

(b) s2

(c) s3

(d) s4

Figure 4: Net Present Value of all actions A from different states S

one being that the value for $s1 < s2 < s3 < s4$ (depicted in 4a, 4b, 4c, 4d respectively). This is to be expected since both the reward of a state and the state itself are related to the consumption of a startup. A second observation we made is that if the amounts are higher the value of different actions are further apart. A possible explanation of this could be that a higher spending, being in $s4$, is often a result of larger amounts of transactions in multiple different categories. We also observe that no dominant actions occur, meaning that no action is better in all situations. Lastly, the NPV decreases over time. This is to be expected when looking at the definition of NPV since all expected future gains are present at the current time. Using the values from Figure 4 we can calculate the NPV of different policies observed in the dataset. We have compared the observed policies with the optimal policy. To calculate the optimal policy, as described in Equation 6, we need to take the action with the highest utility at every step in t. The result of following this optimal policy as well as the observed policies from the dataset are displayed in Figure 5.

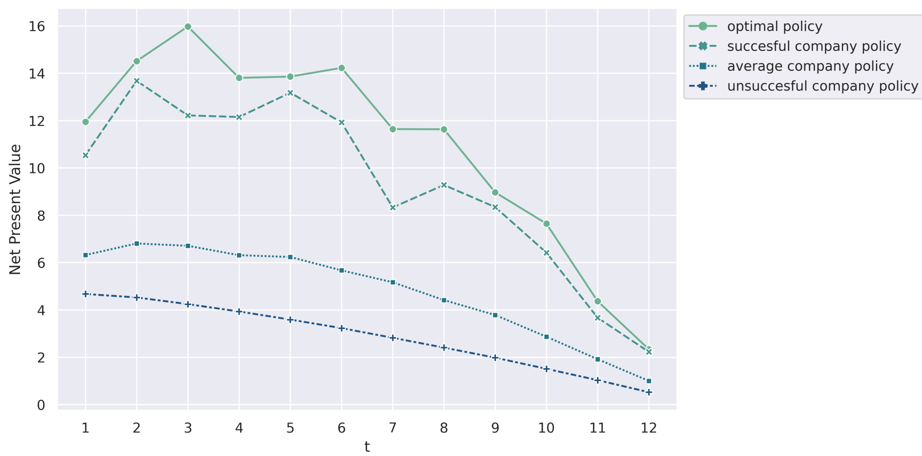
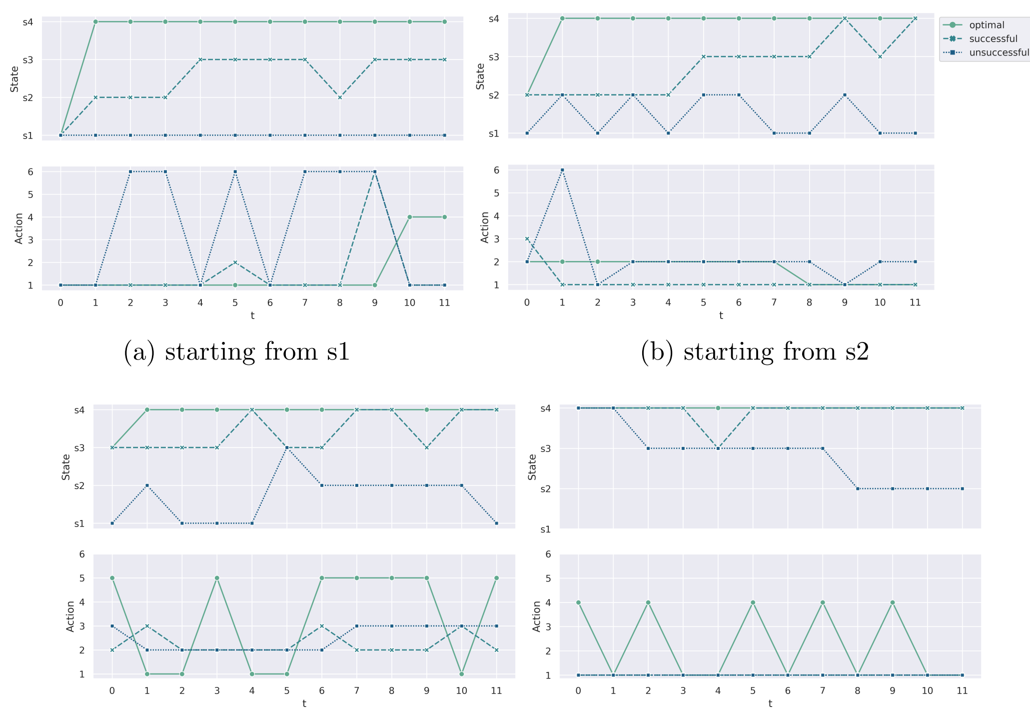


Figure 5: Net Present Value of different policies π

In Figure 5 the calculated optimal policy is shown along three observed policies. The observed policies are the following:

- **successful company policy:** the policy followed by a successful company from the dataset.
- **average company policy:** the policy resulting from the mode of actions taken at every iteration in the dataset.
- **unsuccessful company policy:** the policy followed by a unsuccessful company from the dataset.

Based on the actions taken by the firms the NPV is calculated. Following the *unsuccessful company policy* results in a significantly lower result than the optimum. Note that unsuccessful startups still outperformed startups that went bankrupt during the 12 months recorded in the dataset. The *average company policy* is better than the *unsuccessful company policy*, however, it does not show exceptional performance. When looking at the *successful* *company policy* we see that that this policy outperforms all other observed policies. It is notable that this policy, while being better than the unsuccessful and average policies, is still not the optimal policy. This means that even the best startups could have been able to achieve a higher value. This indicates that increased data analytics can help startups but does not reveal how to do so. To get a better understanding of this we'll look at Figure 6.



(a) starting from s1

(b) starting from s2

(c) starting from s3

(d) starting from s4

Figure 6: Actions and resulting states for successful, unsuccessful and optimal policies

In Figure 6 one can see actions taken by successful and unsuccessful startups alongside the optimal action that a startup could have taken. This is combined with the state resulting from that action. The state, action combination is shown for every iteration. Here, four different starting points are displayed. In Figure 6a the starting point $s1$ is displayed, Figure 6b corresponds with starting point $s2$, Figure 6c with $s3$ and Figure 6d with $s4$. For starting points state $s1$ and state $s2$ the optimal actions are stable. When looking at starting points for state $s3$ and state $s4$ actions are more volatile.

The volatility could be due to a higher degree of competition.

5. Managerial Insights

In this section we leverage the holistic view that cloud vendors possess to provide insights for individual startups. The decisions startups take are vital to their survival. Since startups often have limited resources to fallback on a single mistake could result in insolvency. The best action a startup can take is dependent on the maturity of the startup. In order to make data driven descensions firms should start collecting consumption information early on, and combine it with the output the service provides. Beside collecting this information, follow best practices and benchmark against competition where possible. By doing so, when a startup is more mature it could use this information to make informed decisions that are tailor made for the specific company. In later stages the best practices do no longer universally hold, depending on the provided data is key in the continued growth.

6. Conclusion

States of startups are dependent on the assumption that a higher spending correspond with a higher success rate of a startup. This assumption is expected to hold in general but some exceptions could occur. Once a startup starts with cost optimizations the cost of the startup cloud consumption will be lowered, measured in dollars. In such a case the lower costs is due to a higher efficiency rather than a regression. The data used in this research is subject to selection bias. Only startups that use AWS are part of the research. This excluded startups that have no use for cloud computing infrastructure or opted to use the services of a different vendor. Because we're only working with a subset of all startups the results are not guaranteed to generalize. We expect that our findings will generalize based on the large market share of AWS, and by extend the data used. The startups in the dataset consist of startups from the EMEA region. A startup is treated as an equal datapoint, meaning that no differentiation is made between geographical and cultural differences. If these differences have an effect on the growth of the startups is not expended upon. Actions that occur outside of the model are modelled as a stochastic property of the model, this could be an oversimplification of the complex reality which is not observable. Another factor which is hard to observe the long term effects of startups, since only 1 year worth of data is used. When using a cloud based infrastructure approach it is not uncommon to leverage multiple vendors, this is often referred to as a “multi-cloud approach”. Within this research it is unclear which startups make use of a multi-cloud approach. It could occur that a startup is successful by effectively combining different services from multiple vendors. We will rate these startups only on the consumption at AWS, this could skew the results. These factors could introduce a number of biases and is something that could be improved upon in the future.

What is a favourable actions for a startup could not for an enterprise sized company. Another interesting aspect to explore further is the level of observability. In the research we assume that the environment is fully observable, meaning that we have a complete overview of the different possibilities. Having such an overview is not be achievable for every startup. A version of the MDP model with can be used to calculate the optimal policy under partially observable conditions can be used. This version is called the partially observable Markov decision process (POMDP) [4]. By comparing MDP models and POMDP models the effect of observability can be measured. In order to discover if increased data analytics consumption helps to grow startups we've used a dataset from a cloud services provider containing a year of startup consumption. Each startup in the dataset is grouped according to their cloud consumption per month. Startups with a higher consumption are assumed to be more successful than lower consumption startups. This assumption is based on efficient spending of startups, here a higher spending suggest that the startup has (1) more funds to operate with and (2) more end users to serve. To learn how startups spend their funds we look at the most used services of the startup, measured in dollars. We calculate the influence of using different services at different points in time and startup maturity by using a Markov decision process model. Doing so results in an expected net present value of selecting a most common service, this selection is referred to as an action. Once we have values associated with every action we could take at any point in time for any state a startup could be in a policy can be constructed to guide a startup. To discover if this information increases the growth of a startup we compare it with the policies startups have followed in the past, based on the actions taken in the dataset. Following the calculated optimal policy outperforms every startup present in the dataset. This suggests that increased data analytics consumption does help startups grow. To answer how startups can grow by increased analytics capabilities we need to consider the state startups are in. Based on the available dataset, early state startups benefit most from compute and database services. When a startup has matured more no correlation between the actions selected for this research and the growth of the startups have been found based on the dataset. This higher volatility is suspected to be the result of more competition combined with more ambition firms.

Acknowledgements

I would like to express my gratitude to the startup team at Amazon Web Services for making this research possible. Especially the select few who I've been in contact with on a daily basis. Besides that I owe a debt of gratitude to my university supervisors Xishu Li and Alexandru Uta for their feedback and guidance during the process. Writing this thesis at Amazon Web Services allowed me to accelerate the pace in which I could acquire knowledge about the usage of cloud based solutions. Being part of the startup teams provided context for this work. Lastly, I would like to emphasise my gratitude towards Yegor Tokmakov, Daniel Geske, Maarten Voorneveld & Megan Leoni. Their expertise and feedback has helped shape this work and improved it along the way.

References

[1] Aminova, M. and Marchi, E. (2021). The role of innovation on start-up failure vs. its success. *International Journal of Business Ethics and Governance*, 4(1):41–72.

[2] Archibald, T. W. and Possani, E. (2021). Investment and operational decisions for start-up companies: a game theory and markov decision process approach. *Annals of Operations Research*, 299(1):317–330.

[3] Archibald, T. W., Possani, E., and Thomas, L. C. (2015). Managing inventory and production capacity in start-up firms. *Journal of the Operational Research Society*, 66(10):1624–1634.

[4] Åström, K. J. (1965). Optimal control of markov processes with incomplete state information i. *Journal of Mathematical Analysis and Applications*, 10:174–205.

[5] Behl, A., Dutta, P., Lessmann, S., Dwivedi, Y. K., and Kar, S. (2019). A conceptual framework for the adoption of big data analytics by e-commerce startups: a case-based approach. *Information systems and e-business management*, 17(2):285–318.

[6] Bellman, R. (1957). A markovian decision process. *Journal of mathematics and mechanics*, 6(5):679–684.

[7] Coad, A. (2009). *The growth of firms: A survey of theories and empirical evidence*. Edward Elgar Publishing.

[8] Coad, A., Frankish, J., Roberts, R. G., and Storey, D. J. (2013). Growth paths and survival chances: An application of gambler's ruin theory. *Journal of business venturing*, 28(5):615–632.

[9] crunchbase (2021). Venture funding to european companies h1 2021. `https://www.crunchbase.com/lists/venture-funding-to-european-companies-h/472b3c17-c577-49dc-84be-0f3504b286f8/funding_rounds`.

[10] Dellermann, D., Lipusch, N., Ebel, P., Popp, K. M., and Leimeister, J. M. (2017). Finding the unicorn: Predicting early stage startup success through a hybrid intelligence method.

[11] eurostat (2021). Business demography by size class (from 2004 onwards, nace rev. 2). `https://ec.europa.eu/eurostat/web/structural-business-statistics/entrepreneurship`.

[12] Giardino, C., Wang, X., and Abrahamsson, P. (2014). Why early-stage software startups fail: a behavioral framework. In *International conference of software business*, pages 27–41. Springer.

[13] Keith, A. J. and Ahner, D. K. (2021). A survey of decision making and optimization under uncertainty. *Annals of Operations Research*, 300(2):319–353.

[14] Kuckertz, A., Brändle, L., Gaudig, A., Hinderer, S., Reyes, C. A. M., Prochotta, A., Steinbrink, K. M., and Berger, E. S. (2020). Startups in times of crisis–a rapid response to the covid-19 pandemic. *Journal of Business Venturing Insights*, 13:e00169.

[15] Markov, A. A. (1954). The theory of algorithms. *Trudy Matematicheskogo Instituta Imeni VA Steklova*, 42:3–375.

[16] Puterman, M. L. (1990). Chapter 8 markov decision processes. In *Stochastic Models*, volume 2 of *Handbooks in Operations Research and Management Science*, pages 331–434. Elsevier.

[17] Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[18] Rompho, N. (2018). Operational performance measures for startups. *Measuring Business Excellence*.

[19] Ross, S. M., Kelly, J. J., Sullivan, R. J., Perry, W. J., Mercer, D., Davis, R. M., Washburn, T. D., Sager, E. V., Boyce, J. B., and Bristow, V. L. (1996). *Stochastic processes*, volume 2. Wiley New York.

[20] Schwienbacher, A. (2007). A theoretical analysis of optimal financing strategies for different types of capital-constrained entrepreneurs. *Journal of Business Venturing*, 22(6):753–781.

[21] Shakya, S. and Plemmons, A. (2021). The impact of economic freedom on startups. *Journal of Regional Analysis & Policy*, 51(1):29–42.

[22] Shepherd, D. A., Williams, T. A., and Patzelt, H. (2015). Thinking about entrepreneurial decision making: Review and research agenda. *Journal of management*, 41(1):11–46.

[23] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[24] Wennberg, K., Delmar, F., and McKelvie, A. (2016). Variable risk preferences in new firm growth and survival. *Journal of Business Venturing*, 31(4):408–427.