Carole Renée Elisabeth Maure
Ingrid Corlett Amdahl

Analysis and predictions of success and failure factors of startups

Bachelor's thesis in Business Administration, Business Analytics
Supervisor: Denis Becker
April 2024

Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

NTNU
Norwegian University of Science and Technology

Preface

This bachelor thesis represents the accumulation of three years of knowledge gained at NTNU Business School of Trondheim. Throughout our research, we have acquired a profound comprehension of diverse machine learning models and their utility in analyzing and predicting success and failure factors of startups. Even though this journey has proven itself to be an engaging and challenging experience, it has enhanced our knowledge significantly.

We would like to personally thank our bachelor thesis supervisor Denis Becker, whose collaboration and dedication have profoundly influenced the trajectory of this project. We are truly grateful for his support, active engagement, and accessibility, all of which have been invaluable assets throughout our academic pursuit.

The content of this thesis is the responsibility of the authors.

Abstract

The landscape of business is forever changing, especially with the rise of startups, representing this major shift in innovation. Despite their transformative potential, navigating the journey from conception to a sustainable venture, startups are faced with challenges, leading to a quite high closure rate within the first years. This thesis explores potential factors influencing startup success or failure.

This thesis explores key success factors by analyzing data obtained from Kaggle, where the focus is set on variables such as location, category, time, funding, relationships, and success rates. It is important to point out that the dataset gathers startups from 1984 to 2012, capturing a period of exponential technological growth but also financial crisis.

Assumptions made from data exploration reveals interesting trends, including the influence of location on funding and the correlation between relationships and startup success. The study also puts light on the impact of economic events, such as the 2008 financial crisis, and the cautious approach of investors to startups after this event. Those assumptions were reinforced through secondary data.

Empirical results highlight the significance of relationships, milestones, and inclusion in the top 500 companies as major success factors for startups. A detailed examination of various models, including binomial logistic regression, naive bayes, decision tree classifier, and artificial neural networks, provides insights into predictive capabilities and sheds light on challenges such as overfitting.

This thesis not only contributes to the academic understanding of startup success factors but also offers practical insights of future entrepreneurs. The findings underscore the multifaceted nature of startup dynamics and the potential of business analytics in navigating the uncertain terrain of entrepreneurship.

Sammendrag

Landskapet innenfor næringslivet endrer seg stadig, spesielt med oppkomsten av oppstartsselskaper som representerer en betydelig fremgang innen innovasjon. Til tross for deres transformative potensial, står oppstartsselskaper overfor utfordringer i reisen fra konsepsjon til en bærekraftig virksomhet, noe som fører til en ganske høy nedleggelsesrate i de første årene. Denne bacheloroppgaven utforsker potensielle faktorer som påvirker oppstartsselskapers suksess eller fiasko.

Avhandlingen utforsker nøkkelfaktorer for suksess ved å analysere data fra Kaggle, der fokuset er rettet mot variabler som lokasjon, kategori, tid, finansiering, relasjoner og suksessrater. Det er viktig å påpeke at datasettet samler oppstartsselskaper fra 1984 til 2012, og fanger opp en periode med eksponentiell teknologisk vekst, men også finanskriser.

Antagelser gjort gjennom data utforskning avslører interessante trender, inkludert innflytelsen av lokasjon på finansiering og korrelasjonen mellom relasjoner og oppstarts suksess. Studien kaster også lys på påvirkningen av økonomiske hendelser, som finanskrisen i 2008, og investorenes forsiktige tilnærming til oppstartsselskaper etter denne hendelsen. Disse antagelsene ble forsterket gjennom sekundære data.

Empiriske resultater fremhever betydningen av relasjoner, milepæler og inkludering i de 500 største selskapene som sentrale suksessfaktorer for oppstartsselskaper. En grundig undersøkelse av ulike modeller, inkludert binomial logistisk regresjon, naive bayes, Decision Tree Classifier og kunstige nevrale nettverk, gir innsikt i prediktive evner og kaster lys over utfordringer som overtilpasning.

Denne bacheloroppgaven bidrar ikke bare til den akademiske forståelsen av suksessfaktorer for oppstartsselskaper, men gir også praktiske innsikter for fremtidige entreprenører. Funnene understreker den mangefasetterte naturen til oppstartsdynamikken og potensialet til forretningsanalyse i å navigere den usikre terrenget til entreprenørskap.

Table of contents

Figures

Tables

1. Introduction

The landscape of business and technology is in constant change and startups stand out as the face of this change. With their innovative power, startups play an important role in economic growth and reshaping various industries, which positions startups as the leaders of tomorrow's business ecosystem. With their unique capacity for disruptive thinking, startups challenge traditional paradigms, introducing novel solutions and pioneering groundbreaking technologies. Startups not only contribute to the diversification and evolution of industries but also inject a spirit of competition and dynamism that propels entire sectors forward.

However, the journey from conception to a viable and autonomous startup is marked with multiple challenges as a significant number of startups succumb to the pressures of the market within the first years. According to recent studies, approximately 90% of startups close down already within their first three years (Embroker, 2024). Starting a business requires resilience, a clear sense of purpose, the ability to inspire and persuade others of your vision, and expanding your knowledge in your field to stay ahead of the market trends and innovations. The challenges startups often face, financial uncertainties, market volatility and operational dilemmas, highlight the unstable path that startups must navigate to achieve viability and autonomy.

Even with an uncertain future, some successful startups survive and thrive, and with the help of business analytics, manage to navigate the mysterious landscape of risks and opportunities. The rise of technology and data analytics promises not only to minimize common pitfalls, but also to launch these newly founded businesses towards sustainable success. Understanding the factors influencing the success or failure of startups becomes clearly a necessity.

This bachelor thesis explores the various dynamics shaping the destiny of startups, with a particular emphasis on the use of business analytics to assess possible success or failure factors. Throughout the analysis of key success factors, this research aims not only to contribute to the academic field but also to provide actionable insights to future entrepreneurs and stakeholders. Uncovering the factors that contribute to the success or failure of these new businesses is not only an interesting academic pursuit but also has a crucial implications for the business world.

The purpose of this bachelor thesis is to determine which factors play a significant role in the success or failure of a startup.

There are various reasons why both candidates wanted to work with this thesis theme. Knowing which factors are determining in order for startups to be successful gives a head start for young entrepreneurs in order for them to make more informed decisions, such as allocate resources more effectively, set realistic milestones, or adjust strategies based on identified success factors. In addition, this bachelor thesis theme can also help entrepreneurs with exploring which factors contribute to failure. Learning from failed startups can provide valuable lessons for entrepreneurs, allowing them to avoid common pitfalls and enhance their chances of success.

The subject explored in this thesis is also important for investment decision making. In other words, knowing which startup might become the next big thing is relevant for investors on many levels. First, investors are always faced with high risks when funding startups. Our research could provide insights into mitigating these risks by identifying factors associated with successful outcomes, but also failed startups. Investors may use this information to make more informed decisions, reducing the likelihood of unsuccessful investments, and then diversify their portfolio accordingly.

And lastly this bachelor thesis could be the candidates' academic contribution. Contributing to the academic literature on startup success factors adds to the knowledge base in entrepreneurship and business studies. This thesis could fill gaps in existing research and provide a foundation for future studies.

2. Startups

2.1. Definition of a startup

The definition of a startup embraces a broad spectrum, ranging from small companies with no team to some of the largest technology firms. The term startups refers to a company that is in the initial stages of business, typically characterized by a focus on innovation and the pursuit of a unique product or service. In these early stages, startups are usually founded by a handful of entrepreneurs, who face high costs and generate limited revenue. Consequently, many startups frequently rely on funding from their founders (Grant, M. 2023).

Startups are well known for their unique capacity for disruptive thinking, due to an innovative mindset and willingness to challenge norms. Unlike well established traditional companies, startups are often willing to take risks, explore unconventional solutions and encourage creative thinking. These strategies not only challenge traditional paradigms, but also bring forth innovative ideas that can disrupt existing industries and even give rise to new markets.

Alongside the pursuit of innovation, another common trait for startups is their emphasis on growth. Every startup starts as a small business, but not all small businesses are startups. There are numerous reasons why a high growth rate is crucial for startups. Higher growth rate, enables startups to gain a larger portion of their target market, facilitating the journey to profitability and enhancing their appeal to potential investors. The generated profit will enable the company to reinvest back into the business, enhancing quality of the product and allowing further growth. Additionally, the high growth rate will allow the startup to build a strong brand and attract top talents, thereby creating a significant competitive advantage in the market (Faster Capital, 2023).

Initiating a startup comes with various advantages and disadvantages. The primary disadvantage is that there are risks associated with these young companies. Given the initial costs involved in launching a startup and its limited resources and early profitability, the startups face many financial risks (Faster Capital, 2023). Moreover, the competitive business environment together with the uncertain market conditions and rapid technological changes, can be significantly challenging, hindering the success of many startups. On the other hand, working in startups provides a distinctive environment, offering sizable opportunities for

learning, growth, and innovation, all fueled by the promising prospect of achieving success. The startup landscape is unforgiving, numerous challenges come with launching a new company and pursuing innovation, which can lead to either remarkable success or complete failure. “Most startups fail and the world is only kind enough to allow a few mega successes” (Housel, 2020, p.73)

2.2. History of startups

While startups are commonly linked to the emergence of Silicon Valley, their roots can be traced back to the 18th and 19th centuries. The conceptualization of startups began to take shape in the 19th century with the rise of the industrial revolution. As new technologies emerged, numerous opportunities were presented for entrepreneurs, bringing new forms of startups (Faster Capital, 2023). Subsequently, in the 20th century, major wars erupted, necessitating the development of new technologies. As a result of the Second World War, there was an increased government focus on investing in research and development and therefore numerous new technologies emerged and later on commercialized by startups (Faster Capital, 2023).

The concept of startups we know today started shaping up later in the 1980s. The creation of computers and software allowed entrepreneurs to create innovative products and services and make them more accessible. By using the power of digital tools to their advantage, startups were able to achieve global success in less time than it would have taken in the past (Faster Capital, 2023). In other words, the development of these technologies facilitated the rise of the startup culture. The giants Apple, Microsoft, and Genentech are examples of companies founded during this period.

Many investors were drawn to the potential profitability presented by technology companies. At the same time, there was a growing popularity of the internet, which opened up new possibilities for information exchange, commerce and communication. These factors contributed to the rapid expansion of Internet-related businesses in the mid-1990s and early 2000s, giving rise to what is commonly referred to as the dot-com boom (Faster Capital, 2023). With this widespread enthusiasm and hope of quick return, substantial investments were injected into dot-com startups without due caution. This resulted in the overvaluation of these companies and inflated stock prices. The combination of these factors, coupled with the startups lack of profitability, initiated an overall decrease in the investor confidence. This

triggered the burst of the dot-com bubble, resulting in the bankruptcy of numerous companies in a short period of time (Faster capital, 2023). After the dot-com bubble burst, investors now exercise greater caution when it comes down to investing in technology startups and dot-com entrepreneurs have learned valuable experience and are presently dedicated to establishing sustainable businesses designed for long-term success. (Faster Capital, 2023).

In the present day, a startup ecosystem represents an interconnected network of investors, entrepreneurs, mentors and various other contributors. Key components of the modern startup ecosystem are venture capital and angel investors. These entities play a crucial role in funding startups, offering capital to those with substantial growth prospects, along with providing mentorship and advice to entrepreneurs and leaders within the startup community. Furthermore, the modern startup ecosystem has been shaped by the advances in technology. Companies such as Amazon Web Services (AWS) together with social media platforms such as Instagram and Facebook, allow startups to efficiently expand their operations and spread product awareness at a faster pace and with reduced costs. (Faster Capital, 2023)

Startups play a crucial role in the economy, creating job opportunities and building wealth for numerous investors and entrepreneurs. It is expected that startups will persist in their pursuit of innovation, challenging norms, promoting disruptive thinking, and uniquely shaping our world.

2.3. Cycle of life of startups

When created, new ventures go through a similar life cycle, although the specific experiences and timelines can vary. The startup life cycle is generally a dynamic and evolving process, which can be divided in six main stages: ideation, development, launch, growth, maturity and exit or Sustained Growth. However, the journey is not always linear and these new ventures can encounter unexpected challenges that require adaptation.

In this initial phase, founders brainstorm ideas and identify opportunities for a new business. During this phase, founders generate and refine the initial business idea by identifying a problem that the product or service aims to solve. Additionally, comprehensive market research is crucial. This involves analyzing the market, identifying the target customer and estimating the potential market size (Fonseca, M. 2023).

During the development phase, the business begins to take shape, marking the initiation of testing for the Minimum Viable Product (MVP). In other words, the MPV is the prototype of the created product or service and its main objective is to test how the market responds to the product and its proposed solution (Fonseca, M. 2023). Parallel to developing the MVP, the startup at this stage also emphasizes the establishment of sales and marketing teams and the expansion of its operations (Faster Capital, 2023). The MVPs will be tested in this phase, serving as validation for the business. If the response is favorable, people will show interest in paying for the product, if not, it might be a signal to modify the product prototype. The development stage can be a challenging time, leading to the failure of numerous startups (Faster Capital, 2023).

A positive outcome of the minimum viable product will lead the startup to the next stage, launching. In this phase, the startup needs to develop a go-to-market strategy, where the objective is to create a market and sales strategy (Fonseca, M. 2023). If it is successful, those strategies will differentiate the startups from the competitors, generate buzz and awareness for the business, which can lead to profitable revenue. During this stage, the startups seek further rounds of funding from angel investors or venture capital firms. The funding rounds typically include pre-seed, seed, Series A, B, C and so forth, serving as the financial support for launching the business. (Faster Capital, 2023).

After a successful launch, a startup will enter the growth stage. The main characteristic of this stage is when the company starts to scale its operations, grow its customer base and its revenue. This might involve expanding into new markets, launching new products, acquiring competitors and taking the company to other geographies (Fonseca, M. 2023). As the startup starts to gain traction in the marketplace, several crucial considerations come into play. With the growth of the startup, there will be a need to recruit new employees, enhance technology and infrastructure. Furthermore, the implementation of marketing strategies is essential for the startup to reach new customers, while maintaining engagement with existing ones (Faster Capital, 2023).

When the company begins to generate profits, it transitions into the maturing phase. The main focus in this stage is refining the business model and solidifying its position in the market. During this stage the business generally follows one of two paths, it is either acquired by

another company, or it undergoes a public listing through an Initial Public Offering (IPO) (Faster Capital, 2023). A startup may be acquired by another company for various reasons, including the development of a product or service that aligns with the acquiring company's portfolio or as a strategic move to eliminate a potential competitor (Faster Capital, 2023). Opting for a public listing, the startup will have the chance to function as an independent company. However, if, despite going public, the startup struggles to generate sufficient revenue, the startup might be forced to close.

Startups are dynamic entities, and although there are similar stages, the journey of every individual startup can vary significantly, resulting in different outcomes. It's important to recognize the inherent risks associated with startups, as there is no guarantee of success, either through acquisition or going public. The startup's journey can be an exciting time for the company's founders and shareholders, as it is full of possibilities and opportunities.

3. Literature review

In his analysis of successful and failed startup ideas, Bill Gross, the founder of Idealab, identified five key factors: funding, business model, the product idea, the team and timing. He found out that timing made up for 42% of the success of startups, team and execution 32%, the idea 28%, the business model 24% and funding 14% (Turner, 2022). In this section we will examine each of these factors individually.

3. 1. Funding

Access to capital and resources is vital for startup survival and growth. However, securing funding can be challenging, particularly for new startups. Startups that effectively secure diverse funding sources, including angel investors, venture capital and crowdfunding, are better equipped to address financial limitations. While Bill Gross acknowledges the importance of funding, he suggests that startups should prioritize building a sustainable business model over raising large amounts of capital. To understand why funding ranks as the least critical factor for success, it's essential to evaluate it in conjunction with all other factors. Without a good team, a powerful idea, well-written business plan and good timing funding is useless (Bensley, 2022). If the four other factors are in place, a business can start with low funding and get intense funding when it gains attention (Cuofano, 2024).

3. 2. The business model

Bill Gross argues that a business model is not necessary when it comes down to startup success. A good business model is a very important strategic tool for entrepreneurship, however a good business idea can be useless if it cannot be developed, executed and implemented. A good business model helps focus on the steps needed to make an idea successful but also helps achieve short- and long-term goals (Turner, 2022). For instance, when Google understood that the pay per click model was viable, had it ignored that fact, most probably GoTo would have gained significant traction. Thus, overshadowing Google itself (Cuofano, 2024).

3.3. The product idea

Some ideas might not succeed and some turn out to be successful right away. Gross places ‘idea’ right in the middle of his startup success metrics. For instance, when Bill Gross came out with his search engine, GoTo, he thought about a revolutionary business model, “paid

search" (Cuofano, 2024). On the search engine, websites could pay for top placement on the GoTo.com results page for a given keyword. Advertisers would pay only when people actually clicked on their ads. At the same time Google came out with text-based search ads in January 2000 but it didn't work particularly well. They soon started taking interest in GoTo's business model. And in 2002 Google launched its own "pay-per-click" called AdWords Select that allowed businesses to purchase text ads on search-results pages (Oremus, 2013).

3.4. The team

Building a cohesive and talented team led by an effective leadership is fundamental to startup success. A strong leader is responsible for making decisions, setting a vision aligned with the organization's goals, and motivating employees. A knowledgeable leader can turn even weak ideas into profitable ventures (Obrazchykov, 2023). The mindset of the founding members, as well as their interactions and individual contributions to idea development, together influence the success or failure of the business. If people are not on the same page and don't share the same vision for the future, this may create setbacks and put development off track (Devrix, 2021).

3.5. Timing

The expression "right place, right time" makes sense to mention here. Timing is critical for startup success. Launching a business at the right moment can give a competitive advantage. A product may be thoughtful, functional, and designed to make people's lives easier, but if it's ahead of its time, the market may be reluctant to accept it. This is why Bill Gross puts it as the single most important success factor for startups. While each factor is one part of a whole, timing can trump each one of them in the right circumstances (Bensley, 2022). YouTube, for example, launched when high-speed internet was becoming the norm but before any other video streaming platform had gained prominence (Obrazchykov, 2023). Another example would be Airbnb. The platform came out during the economic recession when people really needed extra money. The economical situation helped people overcome the idea of renting out their home to strangers.

4. Methodology

4.1. The Bias-Variance Trade-off

The bias-variance tradeoff is an important concept in machine learning which deals with the balance between bias and variance in the predictive models. On one hand bias represents the difference between the predicted output and the true output (Singh, 2018). On the other hand variance measures the model's sensitivity to small fluctuations or noise in the training data. It represents the variability in model predictions across different training sets. High variance can result in a model that performs well on the training set but poorly on new data (Singh, 2018). In other words, the bias-variance trade-off tries to find the right level of model complexity that minimizes both bias and variance. The bias-variance tradeoff is noticeable in various machine learning algorithms, including linear regression, decision trees, and artificial neural networks.

Two phenomena linked to high bias and high variance are respectively underfitting and overfitting. Models with high bias may underfit the data, leading to poor performance on both training and test sets, as models with high variance may overfit the training data, performing exceptionally well on the training set but poorly on new data (Singh, 2018).

4.2. Accuracy paradox

The Accuracy Paradox is mostly noticeable in situations when dealing with imbalanced datasets (Khandelwal, 2020). This is clearly our case where we have 64,7% of startups that are still open compared to 35,3% closed startups.

Classification accuracy, representing the ratio of correct predictions to total predictions, serves as a metric for evaluating the performance of a classification model. A two-class problem's assessment can be illustrated through a confusion matrix, providing a comprehensive view of the outcomes obtained by the classification models:

	Actual Positive	**Actual Negativ**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negativ	False Negative (FN)	True Negative (TN)

Accuracy is calculated using the formula:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP}.$$

The abbreviations can be interpreted as follows:

- True Positives (TP) occur when the model indicates 1 while in reality being 1.
- True Negatives (TN) occur when the model indicates 0 while in reality being 0.
- False Negatives (FN) occur when the model indicates 0 while in reality being 1.
- False Positives (FP) occur when the model indicates 1 while in reality being 0.

Ideally a good model would only predict True Positives (TP) and True Negatives (TN), but as our dataset is clearly unbalanced, this will not be the case. We can expect the models to predict with high accuracy, however, in reality they simply predict every result as being of the majority class (still open startups). This can be misleading; it portrays a situation where the model performs excellently when in reality the model fails at an important part of this thesis: help distinguish success or failure factors for startups.

4.3. Models

Given that we are dealing with a classification problem regarding startup status (open or closed), the models to be utilized include binomial logistic regression, Naive Bayes, decision tree classifier, and artificial neural network (ANN).

4.3.1. Binomial Logistic Regression

Binomial Logistic Regression is a statistical model used for binary classification tasks, where the output variable has two possible outcomes (e.g., yes/no, true/false) (Analyticxlabs, 2023). It models the probability that a data point belongs to a particular class based on one or more independent variables. The model learns the relationship between the input features and the probability of a particular outcome, allowing for classification based on probability thresholds. Binomial Logistic Regression is a simple and interpretable model. It provides probabilistic interpretation of predictions and is efficient for large datasets with a large number of features (Analyticxlabs, 2023). However, the binomial Logistic Regression model is susceptible to overfitting if the number of features is large compared to the number of observations (Analyticxlabs, 2023).

4.3.2. Naive Bayes

The Naive Bayes model is a probabilistic classification algorithm based on Bayes' theorem with an assumption of independence between features (Analyticxlabs, 2022). It calculates the probability of a class given the input features using conditional probability. The model computes the posterior probability of each class given the input features by multiplying the prior probability of the class and the likelihood of the features given the class (Analyticxlabs, 2022). It selects the class with the highest probability as the predicted class. The Naive Bayes model is fast and efficient for training and prediction and is robust to irrelevant features (Ray, 2024). However, the Naive Bayes models can be sensitive to rare combinations of feature values, and the strong independence assumption may not hold in real-world data (Ray, 2024).

4.3.3. Decision Tree Classifier

A Decision Tree is a hierarchical tree-like structure used for both classification and regression tasks. It divides the features into categories and assigns a class label or predicts a continuous value for each category (Saini, 2024). These categories form a tree-like model in which each internal node represents a decision based on a feature, and each leaf node represents a predicted value (Saini, 2024). Decision Trees are easy to interpret and are robust to outliers (Analyticxlabs, 2022). The model also captures non-linear relationships between features and the target variable. Nevertheless the Decision Tree model is prone to overfitting, especially with deep trees, as well as it can be sensitive to small variations in the training data (Analyticxlabs, 2022).

4.3.4. Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computational model inspired by the structure and function of biological neural networks (Singh, 2024). This model consists of interconnected nodes organized into layers, including input, hidden, and output layers, where each node performs a simple computation (Singh, 2024). ANNs learn complex patterns and relationships from data through a process of forward propagation and backpropagation. They adjust the weights and biases of connections between nodes to minimize the difference between predicted and actual outputs, optimizing a predefined loss function (Singh, 2024). The ANN model performs well in tasks involving large datasets with complex patterns and non-linear relationships (Singh, 2024). The ANN model has the ability to learn complex and non-linear relationships in data and is a suitable model for with high volatility and non-constant variance, because of their capacity to discover latent correlations in the data without imposing any preset

associations (Singh, 2024). However, the ANN model requires a large amount of data for training and is prone to overfitting, especially with deep architectures (Singh, 2024)

4. Data

4.1. Data collection

The dataset used in this bachelor thesis can be found on the following website: kaggle.com. The dataset we worked with can be found here: https://www.kaggle.com/datasets/manishkc06/startup-success-prediction .

Kaggle is a platform for data scientist and machine learning competitions which allows users to collaborate on various data analytical projects. The platform offers datasets, notebooks and a digital fora for hosting data science projects.

Before preprocessing our data, the dataset contained 953 rows and 49 columns. As there were some missing values and unnecessary variables, some cleaning and transformation of the variables was necessary. This will be discussed in the next part of this thesis.

4.2. Explanation of the variables

One important step in data analysis is understanding the variables one is working with. We will explain shortly what they all mean.

Name	Description	Type
Unnamed.0	not relevant	integer
state_code	state where the startup was founded and developed	string
latitude	latitude where startup can be found	float
longitude	longitude where startup can be found	float
zip_code	zip code of where the startup is located	integer
id	not relevant	integer

city	city where the startup was founded and developed	string
Unnamed.6	is a combination of several other columns including columns city, state_code, and zip_code	string
name	name of the startup	string
labels	0 if the startup has been closed, 1 otherwise	integer
founded_at	date at which the startup got founded	integer
closed_at	date at which the startup closed at	integer
first_funding_at:	date at which the startup got its first funding	integer
last_funding_at	date at which the startup got its last funding	integer
age_first_funding_year	the age of the company in years since it got first funding	float
age_last_funding_year	the age of the company in years since it got last funding	float
age_first_milestone_year	the age of the startup in years when it got its first breakthrough	float
age_last_milestone_year	the age of the startup in years when since it got its last breakthrough	float

relationships	how many relationships with accountants, investors, vendors, mentors the startup has. In other words, quantifying a startup network.	integer
funding_rounds	number of funding rounds	integer
funding_total_usd	amount of money raised in US Dollars	integer
milestones	number of milestones	integer
state_code.1	states the state code of the startup	string
is_CA ; is_NY ; is_MA ; is_TX ; is_otherstate: categorical	says in which state the startup is located, 1 if founded in this state, 0 otherwise	integer
category_code	category of the business	string
is_software; is_web ; is_mobile ; is_enterprise ; is_advertising ; is_gamesvideo ; is_ecommerce ; is_biotech ; is_consulting ; is_othercategory	1 if operating in this industry, 0 otherwise	integer
object_id	not relevant	integer
has_VC	if the startup has venture capital, venture capital is used to support startups with	integer

	the potential for substantial and rapid growth	
has_angel	angel investing is a form of early-stage investment where High Net Worth Individuals provide funding to startups in exchange for a stake in the company. Angel investors provide support to startups in very early stage	integer
has_roundA ; has_roundB ; has_roundC ; has_roundD	Series A, B, C and D are funding rounds that generally follow "seed funding" and "angel investing," providing outside investors the opportunity to invest cash in a growing company in exchange for equity or partial ownership (Reiff, 2023)	integer
avg_participants	average amount of people involved in the startup	float
is_top500	if the startup appeared in the Fortune 500 companies	integer
status(acquired/closed)	target variable, "acquired" means that the startup has been acquired by some other organization	string

4.3. Causality

When working with the given variables of our dataset, it is also important to reflect on any lurking variables. In other words, variables that are not the primary focus of our study but may have impacted the analysis or interpretation of our findings if they were included. There are multiple variables that could have been considered for this analysis but were not part of the dataset.

The first variable worth mentioning is economic factors, such as financial crises, which can influence its success or closure. Economic recessions or periods of growth might impact investors' willingness to fund any promising startups. This factor is closely linked to the timing of a startup's entry into the market. Being an early entrant or entering at the right moment in a market's lifecycle might contribute to favorable outcomes and easy access to fundings. Another influential lurking variable is trends within specific industries. Changes in technology, consumer preferences, or regulatory environments can profoundly affect startup outcomes. This is also closely linked to technological innovation that occurs during the startup's lifespan. Access to innovative technologies and the ability to adapt to technological changes can impact success. Moreover, the level of competition within the market to each startup can be highly relevant. High competition may pose challenges, while low competition could provide opportunities for growth. The outcome of a startup can also be influenced by the feedback from customers and the public perception of its products or services. Positive reviews or a favorable public image may contribute to success, although the opposite may lead to diminished customer trust, lower sales, and difficulties in building a positive brand reputation. Lastly, it is important to consider the leadership styles of the founders, as they can influence the startup's culture, decision-making processes, and resilience in the face of challenges, thus contributing to the success or failure of startups.

5. Data preprocessing

5.1. Formatting dates

Before being able to work with our data, we had to transform some of our variables. We first decided to reformat the variables fouded_at, first_funding_at and last_funding_at as they were in this format: dd-mm-yyyy. Our code read our CSV file and performed the formatting for the variables mentioned earlier. It then converted the variables to datetime format and extracted the year, and finally overwrote the original file with the modified DataFrame.

5.2. Deleting unnecessary variables

We also decided to delete some unnecessary variables: “Unnamed: 0” , “'Unnamed: 6”, “id” “'name” , “object_id”, “closed_at” , “state_code.1” , “labels” , and “category_code”. The variables “Unnamed: 0” , “'Unnamed: 6”, “id” , “object_id” were noise in our analysis and had a lot of missing values. The variables “labels” and “closed_at” were a repetition of the variable “status”, both giving information if the startup was still open or not. And lastly the variables “state_code.1” and “category_code” were strings which were already transformed into dummies with variables such as is_CA and is_software for example.

5.3. Handling missing years in milestones variables

We noticed a lot of missing values for the milestones variables (*age_first_milestone_year* and *age_last_milestone_year*). We decided to replace the missing values with the mean of when the first and last milestones were achieved by startups. This helped to maintain the integrity of the data and facilitated analyses or visualizations that involved these columns.

5.4 Handling categorical variables

Our research question involved the variable “status” which was a categorical variable containing two possibilities: acquired (the startup is still open) and closed (the startup is closed). To run our models we had to transform that variable into a dummy variable: closed. The value 0 means that the startup is still open and 1 means that the startup closed.

5.5. Scaling the variables

Scaling variables is an important step when it comes to data preprocessing. Scaling ensures that variables are measured on the same scale. This is important when dealing with features that have different units of measurement, such as years or amount of fundings in our case for example. In other words, scaling allows us to compare variables in a more efficient way, making it easier to identify which features contribute more significantly to the model's predictions. Some machine learning algorithms are also sensitive to numerical instabilities when dealing with variables on different scales. Scaling helps minimize numerical instability issues, reducing the risk of computational errors during the training process. This is especially true when it comes to the ANN model.

6. Empirical results

Numerous graphical visualizations were employed during the data exploration phase to facilitate a detailed comprehension of the dataset, offering substantive insights into the characteristics and nuances of startup entities.

6.1. Data exploration

By conducting a comprehensive examination of value counts, we determined the frequencies of companies categorized as 'closed' and 'acquired'. The analysis reveals that the number of companies labeled as 'acquired' surpasses those categorized as 'closed,' reflecting a discernible disparity of 29.4%, as shown in figure 1. This substantial asymmetry has the potential to help assess the major factors influencing the success or failure of startups, but can also give us biased results. A thorough discussion of this matter is scheduled for subsequent sections.

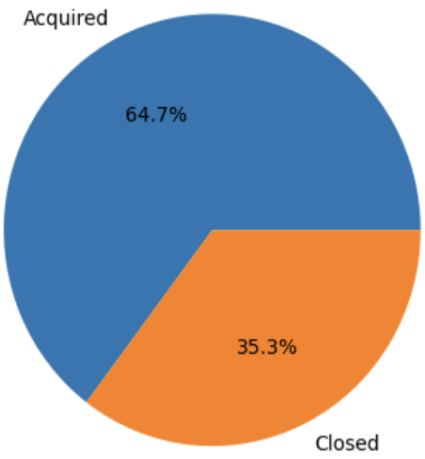


Figure 1. Distribution of acquired and closed startups

6.1.1. Location

When visualizing the distribution of the startups in the USA, we observed that California had the largest number of startups, by 52.9%, followed by New York with only 11.5% and Massachusetts 9% as shown in figure 2.

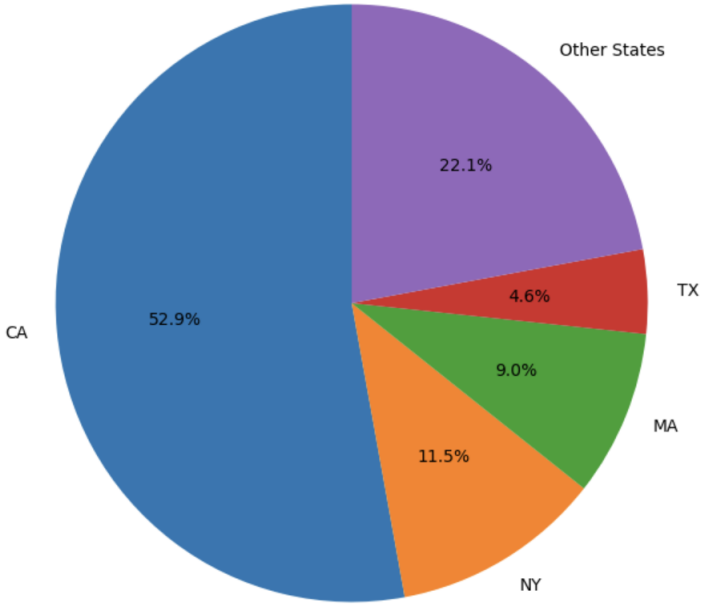


Figure 2. Distribution of startups in the USA

This huge difference might be due to the presence of Silicon Valley in California. Silicon Valley is a global center for high technology and innovation and is the home of many start-ups. The area has many advantages that make it attractive for entrepreneurs looking to start a business, including access to venture capital, a highly educated workforce, and a culture of risk-taking (Faster Capital, 2023). When it comes down to New York, besides the fast-paced lifestyle and the willingness to take risks, the city attracts talent and a highly educated workforce, due to the fact that it is the home of top universities (Faster Capital, 2023). Concerning Massachusetts, it is mostly because of Boston city. Boston came out on top as the best city for startups. It has the 4th highest VC investment in the US and is the 11th most expensive city to live in. In other words, this region experiences a considerable flow of financial resources, but it's not so expensive that it's inaccessible for startups (Faster Capital, 2023).

Figure 3 displays where the startups are located helped visualize any clusters indicating whether specific regions in the USA were exhibiting concentrations of startup activity. We notice two obvious clusters: in the silicon valley and in the state of New York, in other words on the east and west coast. Why those regions could have a concentration of startups was already discussed above.

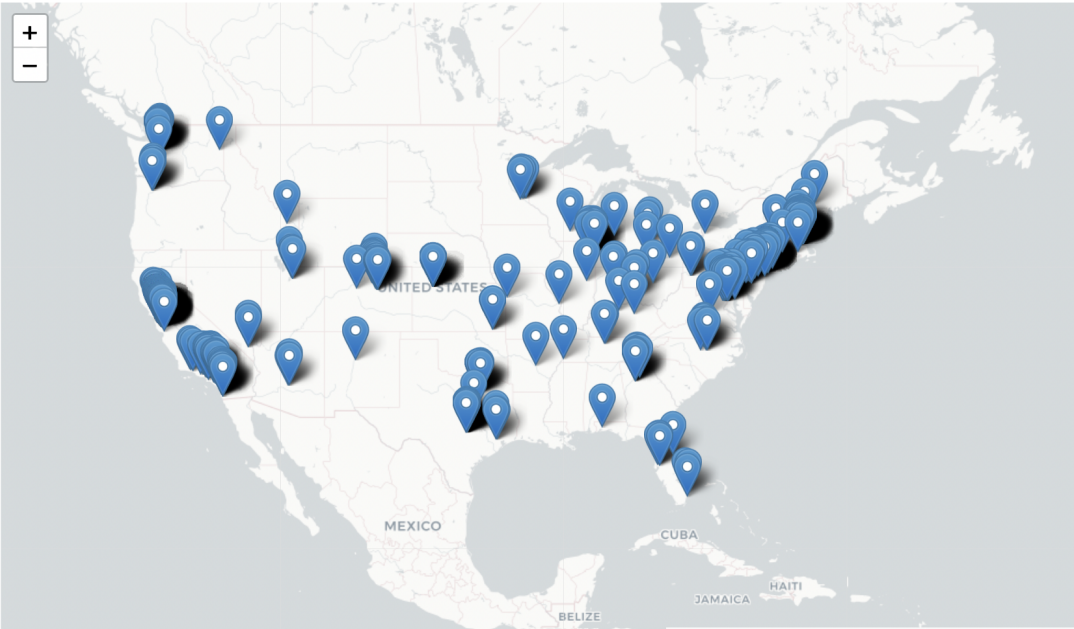


Figure 3. Location of startups in the USA

One assumption we can already make is that location matters for the success of startups. A successful startup needs to be in a place that offers access to customers, resources, as well as it needs to be in an area that can support it financially and legally (Faster Capital, 2023). Locating near resources like universities, incubators, and venture capital firms can also give startups access to expertise and funding. We will see in another section how funding is related to the success of startups. As already discussed, Silicon Valley is located in California and New York fosters a fast paced environment and innovation. Certain areas may have a larger and more diverse talent pool, which can give startups access to the skills they need to succeed (Faster Capital, 2023). This is the case for those areas, proving that location matters for startups.

6.1.2. Category

Concerning the categories with the largest numbers of startups, we can see on figure 4 that “Software”, “Web”, “mobile”, “enterprise”, “advertising” and “video games”, have a significant number of startups when compared to other categories.

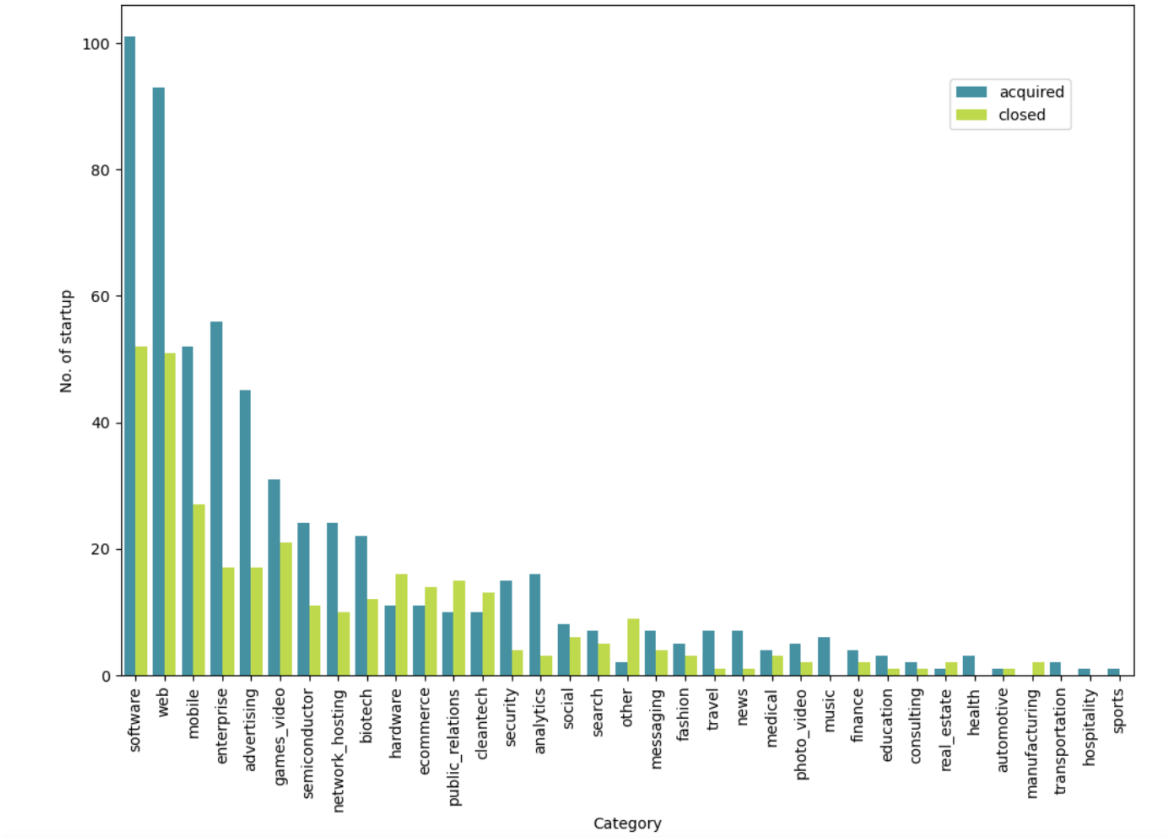


Figure 4. Number of startups per category

In table 1 we can see that software startups really stand out compared to others, as over 100 of the 153 listed were still open. This information is based on the dataset, which contains startups from the years 1984 to 2012, where innovation in technology was exponential. This might have been a driver for particular startups. As from 2012 up until today (2024), the startup field changed significantly. The start up trends for 2024 will be fintech, edtech, agritech, climatech, healthtech and e-commerce (Shanthi, 2024). It can be assumed that the Corona crisis has reshaped, not only the economy, but also the way people shop. Retail eCommerce sales in the United States were 710 billion dollars and is predicted to rise by over 65% in 2024. Mobile app development is also increasing, pushing eCommerce as a new method for conducting online transactions using a smartphone or tablet. Mobile applications are expected to become the main platform for internet commerce (Hudziy, 2024). A new assumption can be made: the category of the startup is important for success.

category_code	total_success	total_closed	total_startup
travel	7	1	8
news	7	1	8
analytics	16	3	19
security	15	4	19
enterprise	56	17	73
education	3	1	4
advertising	45	17	62
photo_video	5	2	7
network_hosting	24	10	34
semiconductor	24	11	35
consulting	2	1	3
finance	4	2	6
software	101	52	153
mobile	52	27	79
biotech	22	12	34
web	93	51	144
messaging	7	4	11
fashion	5	3	8
games_video	31	21	52
search	7	5	12
medical	4	3	7
social	8	6	14
automotive	1	1	2
ecommerce	11	14	25
cleantech	10	13	23
hardware	11	16	27
public_relations	10	15	25
real_estate	1	2	3
other	2	9	11

Table 1. Total successful and total closed startups per category

6.1.3. Time

When looking at figure 5 which displays the opening and closing of startups throughout the years, we discovered that most of the startups were founded between 2000 and 2012, where the peak for the acquired startups was between 2005 and 2007 and the peak of the closed startups

was 2010. This might be because of the high-growth of technology in the years 1990 to 2000. This period was also called the dot-com boom or the tech bubble. (Gascon, C. & Karson, E. 2017). With significant technological advancements and the rise of the internet, we can assume that this might have facilitated the environment for the growth of innovative startups, especially in the tech sector.

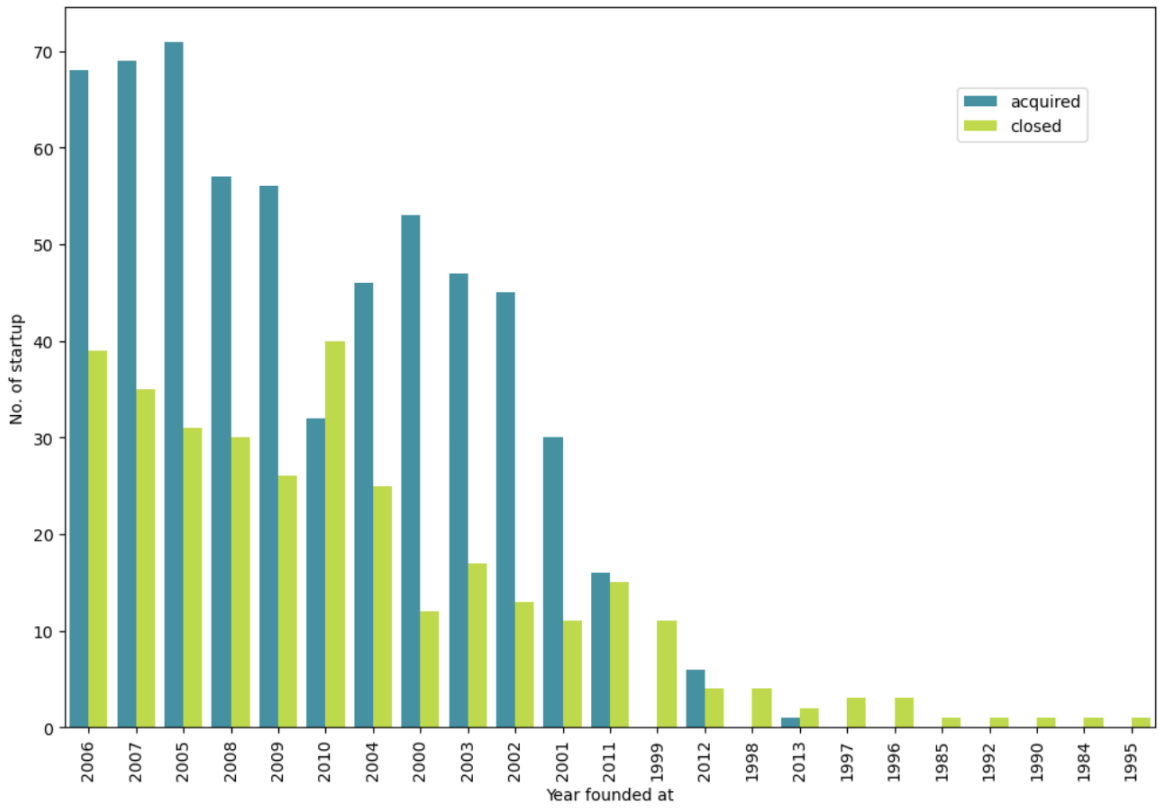


Figure 5. Number of founded and closed startups per year

6.1.4. Fundings

On figure 6 we can also observe that most of the startups got their first fundings between 2005 and 2010.

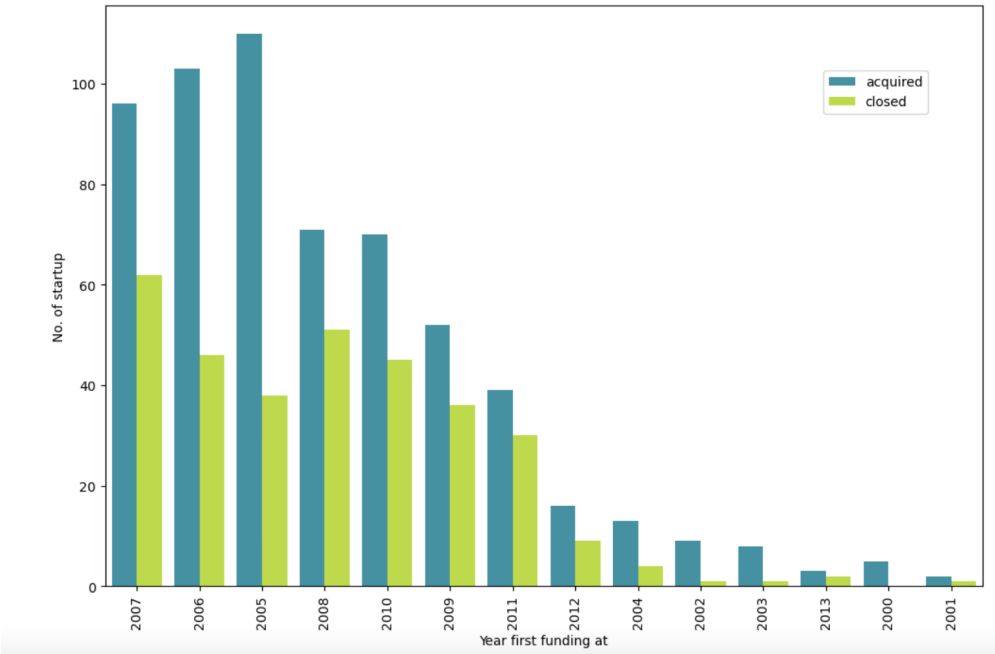


Figure 6. Distribution of first funding per year

And when determining which categories received the highest funding, we find out that mobile, software, web and biotech startups get the most fundings as shown in table 2.

category_code	funding_total_usd
mobile	7263750881
software	2657598865
web	1729035436
biotech	1723699484
enterprise	1338882096
cleantech	1300284730
semiconductor	1105156970
advertising	918619012
games_video	844643530
hardware	773938873

Table 2. Total funding per category

There can be multiple reasons for the rise and fall of startup fundings before and after that period. The period leading up to 2010 was marked by economic growth and optimism (Pew Search Center, 2012), especially following the recovery from the dot-com bubble burst in the early 2000s. This situation created a favorable atmosphere for investors to invest with

confidence, even in high-risk startups. Concerning the mid-2000's era, the significant technological advancements and the rise of Web 2.0 created a fertile ground for startups, leading to increased interest from investors to bet on emerging startups. After 2010, a shift in investors' willingness to invest has been noticed as their excitement that was initially high for certain technologies or sectors, gradually faded. Investors may have become more careful as a result of a saturating market when it comes to startups. In some sectors, particularly those related to the internet and technology, there might have been a saturation of the market, this can particularly be true for smartphones (Faster Capital, 2023). It is also important to mention global economic events, such as the 2008 financial crisis, had a significant impact on investor behavior. The aftermath of the crisis led to increased risk aversion and a more conservative approach to investment regarding startups. This is reflected in the numbers of startups opening after the financial crisis of 2008. The number of new businesses created annually in the decade before the financial crisis averaged 670 000 a year, reaching a high of more than 715,000 in 2006. The startup numbers fell dramatically during the crisis, reaching a low in 2010 of 560,000 (Weltman, 2023).

Venture capital (VC) can also be a success or failure factor for startups. Venture capital is a form of financing startups, offering financial support, strategic guidance, industry expertise and networking opportunities to those young companies with substantial growth prospects (Hayes, A. 2024) VC plays a crucial role for startups as it facilitates scaling of operations by enhancing product development and providing an extensive network, leading to accelerated growth. We noticed on figure 7 that still opened startups with venture capital were twice as much as the still opened startups without venture capital.

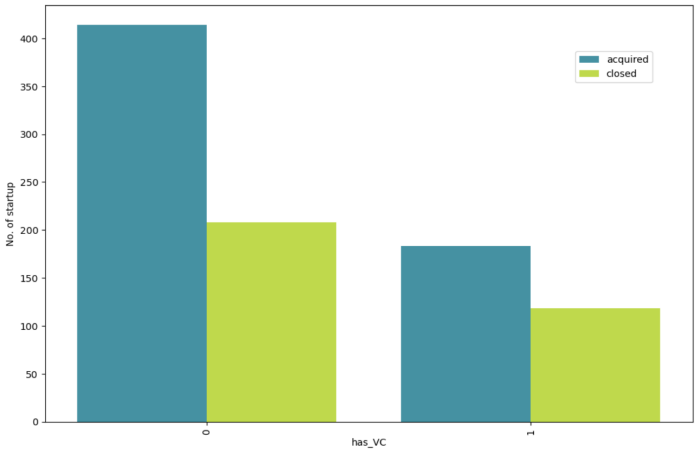


Figure 7. Numbers of opened and closed startups having VC

One could assume that the companies that get venture capital have a higher chance of succeeding, when compared to the ones that do not get venture capital. This assumption might be halfway true as Harvard Business School senior lecturer Shikhar Ghosh shared that the prevalence of failure in the world of venture capital is much higher than what's reported. According to Ghosh, as many as 75 percent of venture-backed companies never return cash to investors (Hoque, 2012). This idea is reinforced by the correlation map we found displayed in figure 8 which indicates that the correlation between venture capital and if a startup is opened or closed is 5,65%. This is considered a weak correlation, meaning if a startup has venture capital or not, does not help with their success.

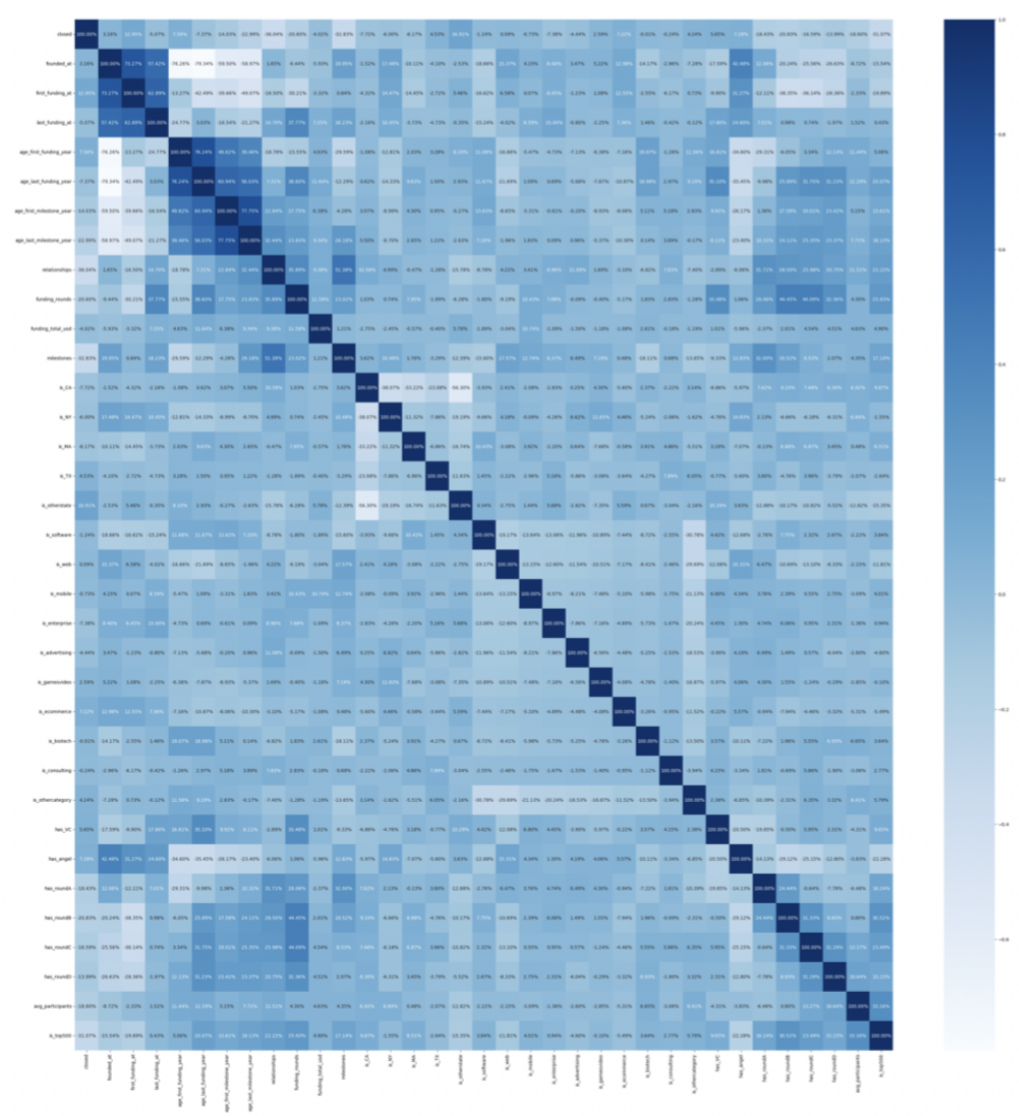


Figure 8. Correlation matrix

Based on this idea, we looked at specific states (California, New York, Massachusetts, Texas and other states), and the trend of fundings by category of startups throughout the years. In California, software and web startups got the most fundings, with software getting a rapid growth between the years 2004 and 2005 as shown in figure 9. This can potentially be explained by the explosion of technology in the early 2000's.

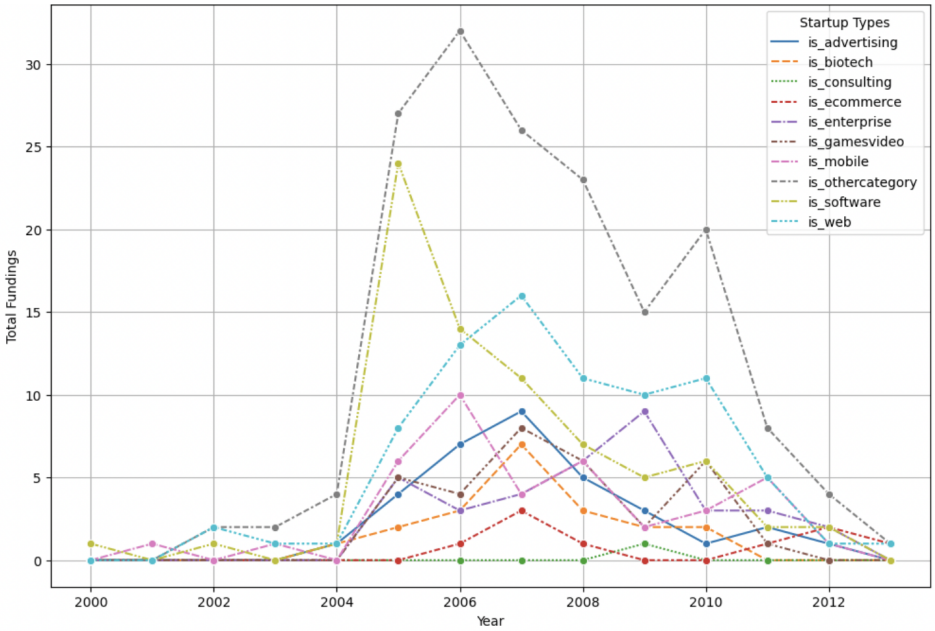


Figure 9. Funding trend for California startups over the years by startup types

In the state of New York startups in video games and web got the most fundings throughout the years as shown on figure 10.

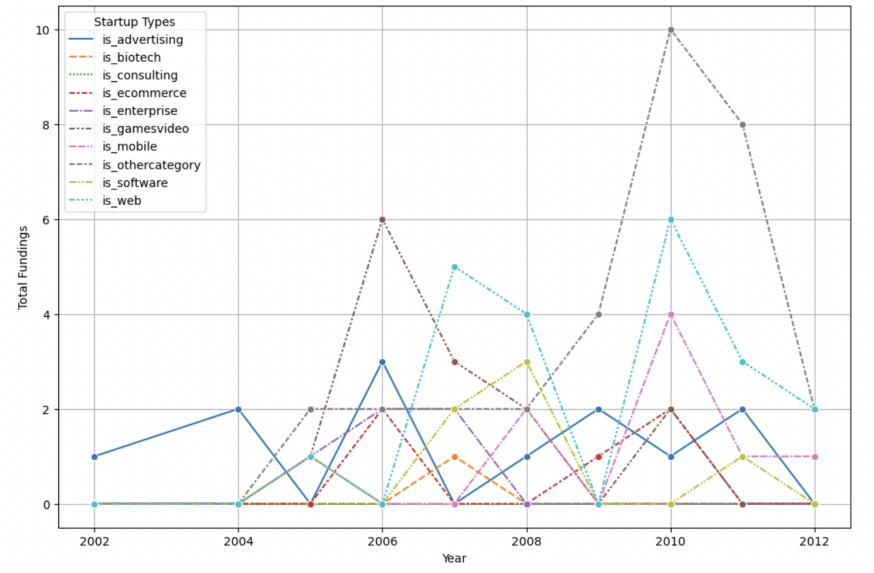


Figure 10. Funding trend for New York startups over the years by startup types

As for Massachusetts, software and mobile startups got the most fundings between 2004 and 2006 as shown in figure 11.

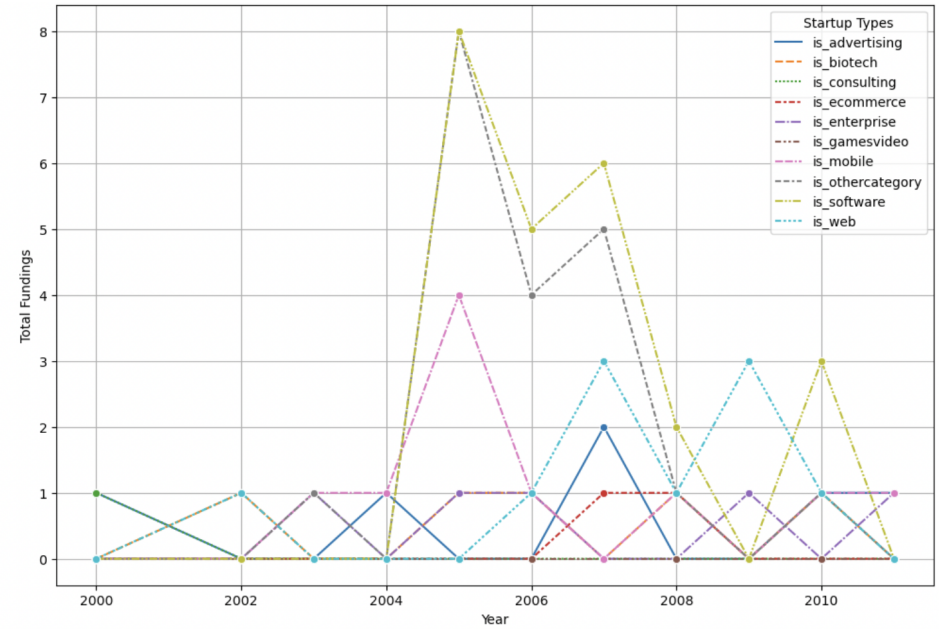


Figure 11. Funding trend for Massachusetts startups over the years by startup types

In Texas the startups with most fundings were in software, enterprise and web, between the years 2005 and 2009 as shown in figure 12.

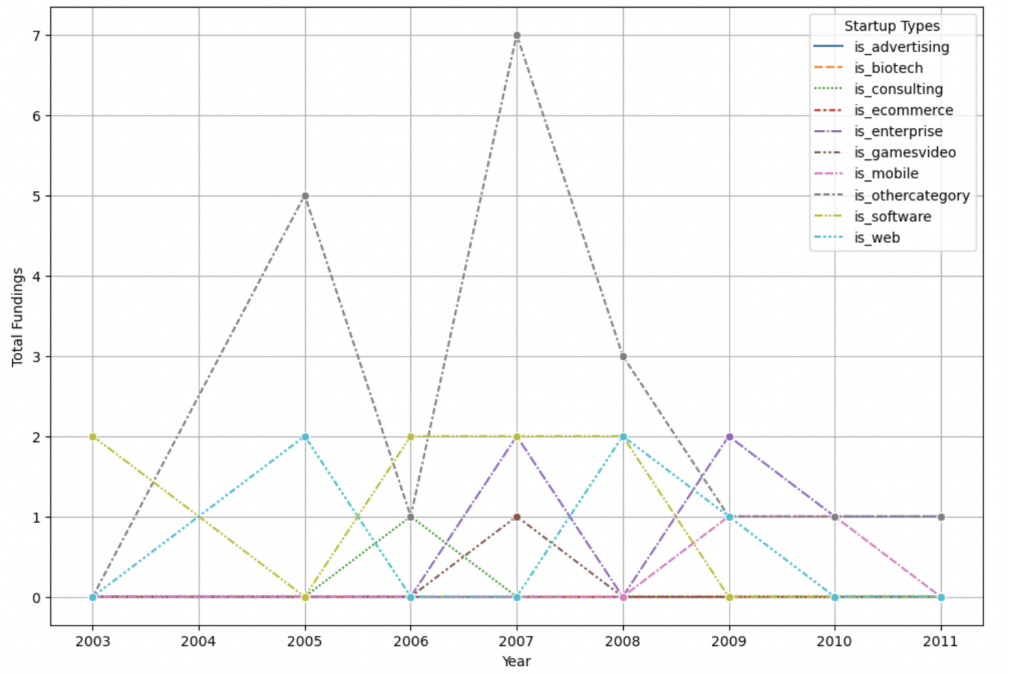


Figure 12. Funding trend for Texas startups over the years by startup types

When compared to other states, we notice the same trend. Startups in software, web and enterprise have the most fundings between 2005 and 2011 as shown in figure 13. The assumption that location is important for startups was already discussed earlier (see 6.1.1. Location). We can now assume that the location of a startup also helps with getting more fundings.

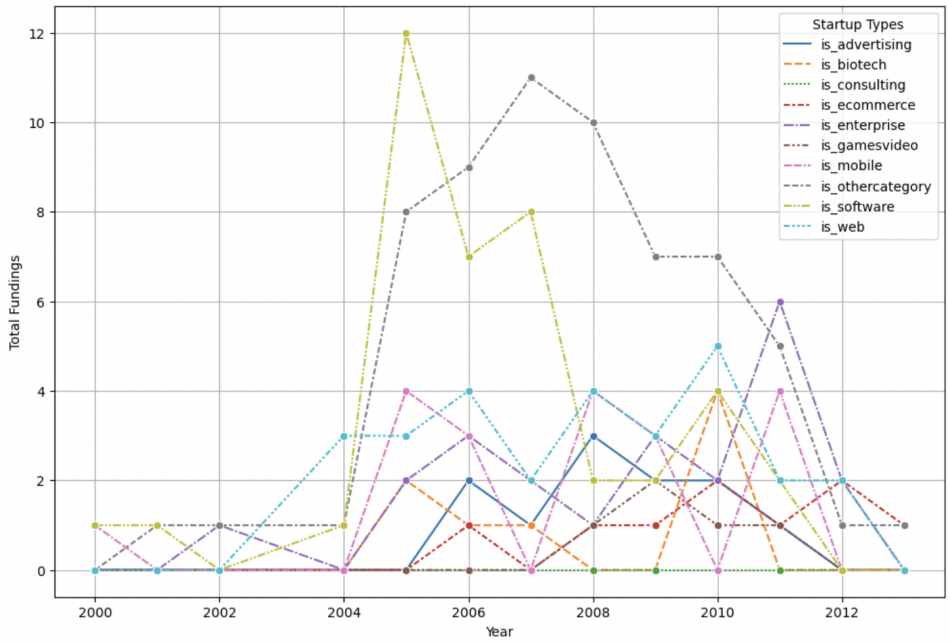


Figure 13. Funding trend for other states startups over the years by startup types

6.1.5. Success rate

Adding on those findings, we can have a look at the success rate of startups and which category has the highest success rate. This gave us some pretty interesting insights. The success rate is measured by dividing the startups that succeed but the total amount of start ups:

$$Success\ rate\ = \frac{Succeeded\ startups}{total\ startups}$$

Surprisingly, as shown in table 3, categories with the most success rate are travel and news, with 87.50% and analytics with 84.21% of success. Although those categories have the highest success rate, we noticed that the total number of startups in those categories vary from 8 to 19. This can be explained in numerous ways. In the early 2000's both advances in technology, especially in analytics and web technologies and a significant demand for innovative solutions in the travel, news, and analytics sectors provided opportunities for startups to create unique and valuable products or services by addressing specific needs or gaps in these markets.

category_code	total_success	total_closed	total_startup	success_rate
travel	7	1	8	87.50
news	7	1	8	87.50
analytics	16	3	19	84.21
security	15	4	19	78.95
enterprise	56	17	73	76.71
education	3	1	4	75.00
advertising	45	17	62	72.58
photo_video	5	2	7	71.43
network_hosting	24	10	34	70.59
semiconductor	24	11	35	68.57
consulting	2	1	3	66.67
finance	4	2	6	66.67
software	101	52	153	66.01
mobile	52	27	79	65.82
biotech	22	12	34	64.71
web	93	51	144	64.58
messaging	7	4	11	63.64
fashion	5	3	8	62.50
games_video	31	21	52	59.62
search	7	5	12	58.33
medical	4	3	7	57.14
social	8	6	14	57.14
automotive	1	1	2	50.00
ecommerce	11	14	25	44.00
cleantech	10	13	23	43.48
hardware	11	16	27	40.74
public_relations	10	15	25	40.00
real_estate	1	2	3	33.33
other	2	9	11	18.18

Table 3. Success rate for startups per category

The landscape of travel agencies underwent a transformative shift with the rise of technology (Nadeem, 2023). Before travel agencies were physical offices that people visited to plan and book their trips. However, recognizing the potential for optimization and convenience, some startups seized the opportunity to revolutionize the travel industry by introducing digital platforms where everything was available and your next trip just one click away (Nadeem, 2023). Adapting to this technological evolution, digital transformation allowed travelers to have access to various and existing new destinations without leaving their home. Some major actors in this sector redefined the way people travel. Airbnb, for instance, allows private people to rent out their rooms or home to travelers, offering a unique and personalized travel experience. Similarly, Booking.com changed the booking process, providing users with an

extensive choice of accommodations worldwide and sometimes proposing advantageous prices. The global revenue generated by travel apps was predicted to increase by 17% in 2023 compared to the previous year, reaching 400 million U.S. dollars (Statista Research Department, 2023). The role of digital platforms in the travel sector continues to evolve, shaping the way people explore and engage with the world.

6.1.6. Relationship (network)

It is also interesting to look at the number of relationships a startup has. By relationship is meant the network and connections a startup has. As shown in figure 14, startups have in general on average between 1 and 12 relationships. Most open startups have between 3 and 5 relationships.

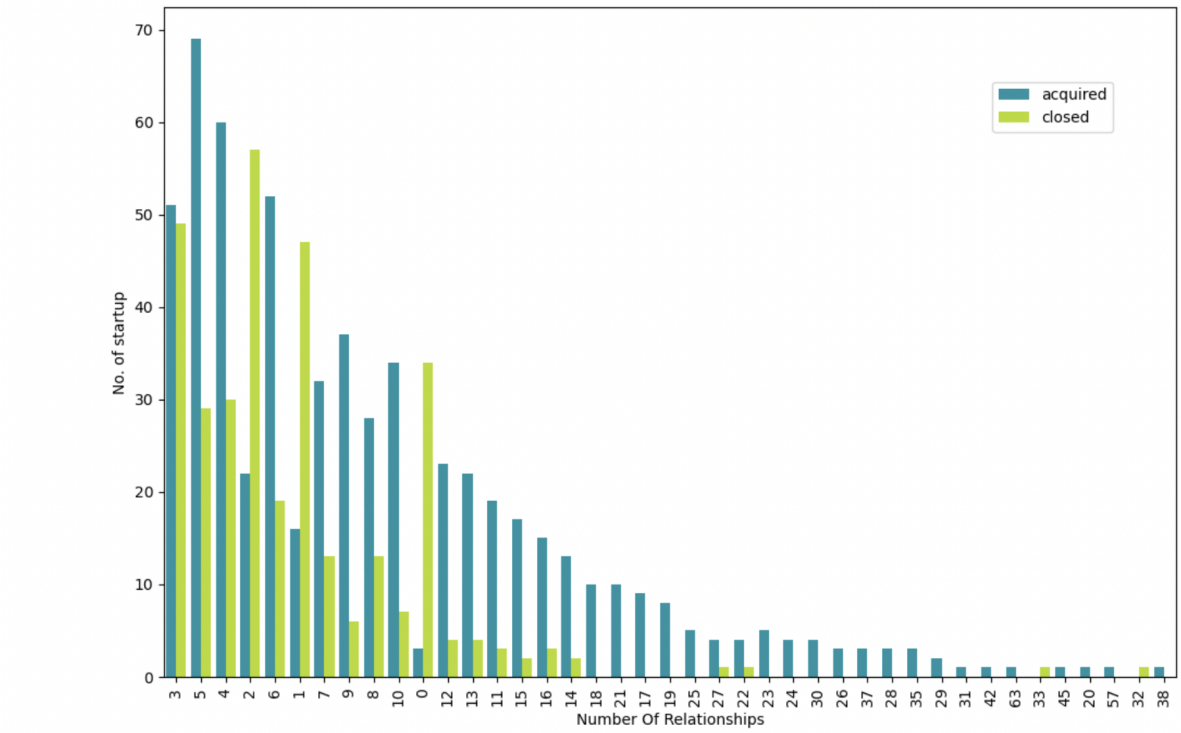


Figure 14. Numbers of relationships startups have

The link between relationships and the success of a startup is often crucial and multifaceted. This assumption can be assumed to be true, the correlation between relationships and a startup's status was 36% and is the highest correlation as seen in table 4.

closed	1.000000
founded_at	0.031645
first_funding_at	0.129485
last_funding_at	0.050697
age_first_funding_year	0.075637
age_last_funding_year	0.073731
age_first_milestone_year	0.140320
age_last_milestone_year	0.229893
relationships	0.360434
funding_rounds	0.206049
funding_total_usd	0.040176
milestones	0.328260
is_CA	0.077217
is_NY	0.059996
is_MA	0.081735
is_TX	0.045309
is_otherstate	0.169067
is_software	0.012429
is_web	0.000873
is_mobile	0.007312
is_enterprise	0.073772
is_advertising	0.044355
is_gamesvideo	0.025893
is_ecommerce	0.072193
is_biotech	0.000104
is_consulting	0.002373
is_othercategory	0.042408
has_VC	0.056515
has_angel	0.072840
has_roundA	0.184307
has_roundB	0.208257
has_roundC	0.165902
has_roundD	0.139940
avg_participants	0.185992
is_top500	0.310652

Table 4. Correlation between dependent variable and independent variables

Thus, building relationships can significantly impact a startup's success in several ways. Connections within the industry or business community can open doors to opportunities that might be challenging to access otherwise (Perry, E. 2023). Strong networks provide startups with access to valuable resources, including fundings. Building relationships with investors is crucial for securing funding. A strong network can facilitate introductions and build trust, increasing the chances of successful fundraising (Faster capital, 2023).

6.1.7. Correlation

Lastly we decided to have a look at the correlation between our dependent variable (closed_at) and our independent variables as shown in figure 8. However, it is important to note that our dependent variable is a binary variable, 0 meaning the startup is still open while 1 meaning the startup closed. Negative values in the heat map therefore imply an inverse association between the variables. For example, a negative correlation suggests that as the dependent variable increases, the likelihood of the startup closing decreases. For simplicity, it is also possible to read those correlations in table 3.

We found out that those three factors influence the success of startups the most: relationships (36,04%), milestones (32,83%) and being in the top 500 (31,07%). The assumption that relationships are a success factor was discussed above and will be reinforced later on by the model Decision Tree Classifier.

The variable "is_top500" refers to the list of 500 of the largest companies in the United States compiled by Fortune magazine every year. To be a Fortune 500 company is widely considered to be a mark of prestige (Hayes, 2021). It is quite clear that such a title and recognition helps catch the attention of possible investors and create a network.

6.2. Models

To be able to foresee any success or failure factors, it is also interesting to have a look if it's possible to predict the success or failure of a startup. For that matter we have divided our dataset into a training and testing set. On both training and testing sets we notice on figure 15 and 16, that we have more "0" than "1", meaning our models will be trained on more data from still opened startups. This could give us biased or overfitting results.

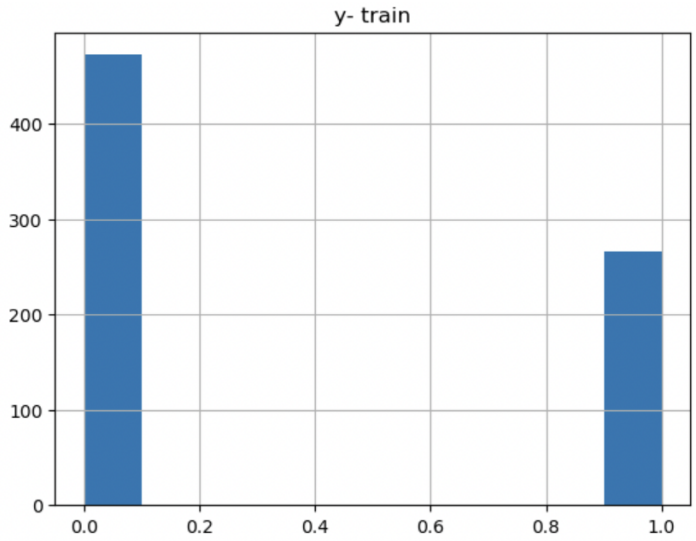


Figure 15. Distribution of y-variable on training set

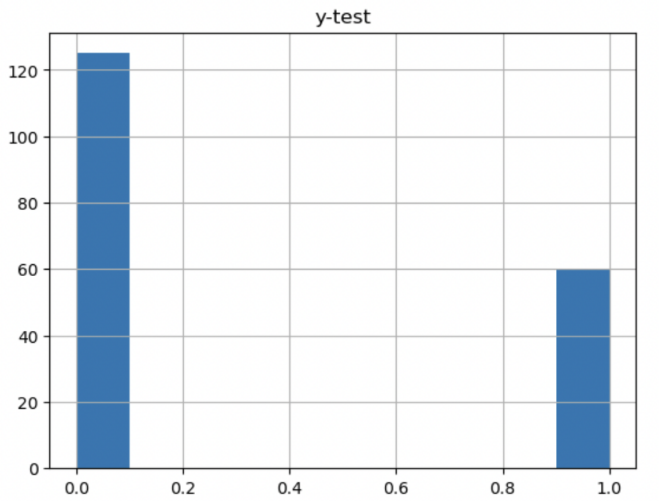


Figure 16. Distribution of y-variable on test set

6.2.1. Binomial Logistic Regression

We observed that approximately 75.53% of the predicted outcome for the training set were correctly predicted, while approximately 80.54% of the predicted outcome for the test set is correct. This indicates that the model generalizes well to unseen data, as the accuracy on the test set is slightly higher than that on the training set. In this case, the model predicts better on the test set than the train set, ruling out the possibility for overfitting. The relatively higher accuracy on the test set, compared to the training set, is an encouraging sign of the model's ability to make accurate predictions beyond the data it was trained on.

When looking at the confusion matrix from the testing set on figure 17, we could conclude the following: in 60% of the cases, the model could predict if a startup was still open and in 20,54% of the time could predict if a startup was closed. This means that in 19,37% of the time, the models did not manage to predict if a startup was still open or had already closed.

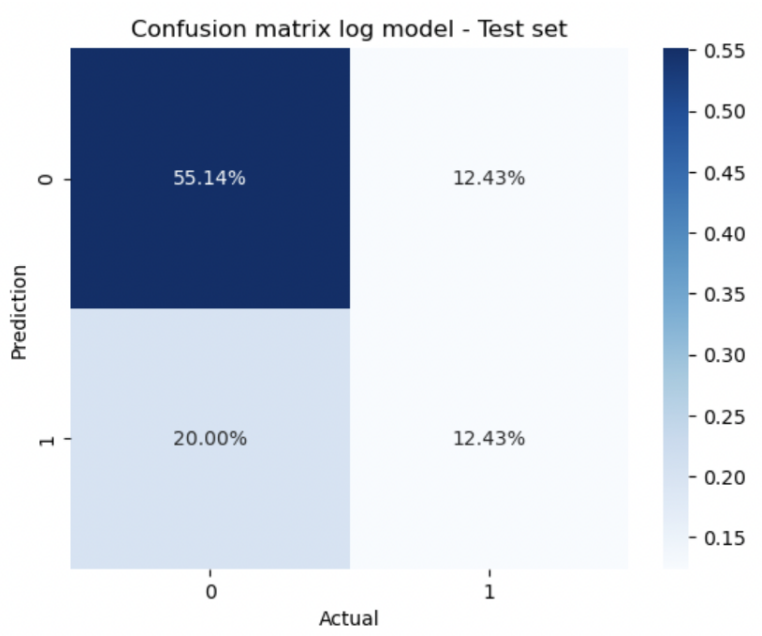


Figure 17. Confusion matrix for logistic regression model- testing set

6.2.2. Naive Bayes

In the context of the Naive Bayes model, the accuracy scores were comparatively lower. The model correctly predicted outcomes in the training set 69% of the time, whereas its performance on the test set was 67.70%. The close accuracy on both the training and testing set shows no sign of overfitting, but the obvious lower scores compared to the logistic regression model are concerning.

When looking at the confusion matrix from the testing set on figure 18, we could conclude the following: in 55,14% of the cases, the model could predict if a startup was still open and in 12,43% of the time could predict if a startup was closed. This means that in 32,43% of the time, the models did not manage to predict if a startup was still open or had already closed. This percentage increased considerably compared to the logistic regression model.

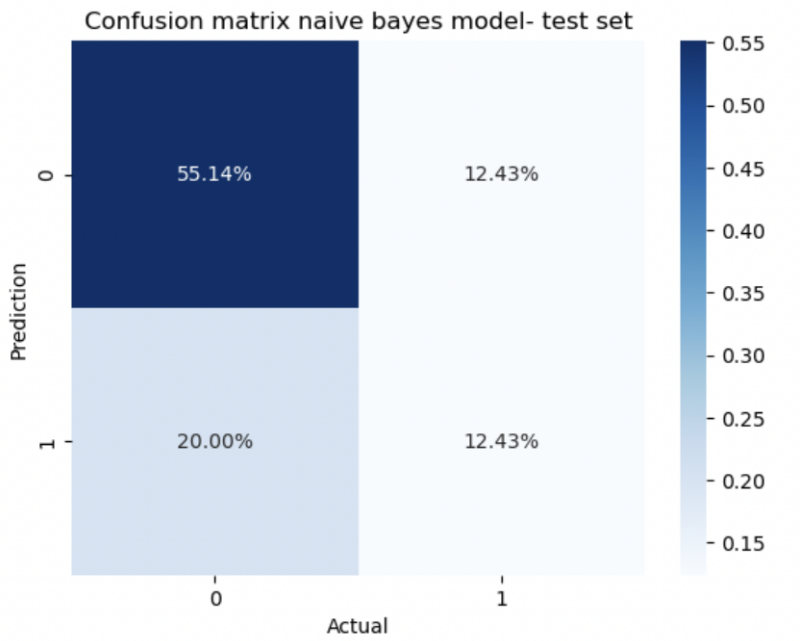


Figure 18. Confusion matrix for naive bayes model- testing set

6.2.3. Decision Tree Classifier (DTC)

When creating the DTC model structure, careful consideration was given to certain aspects, such as tree pruning. This process involves removing unnecessary branches or nodes that have minimal impact on improving the model's predictive accuracy with unseen data. In other words, pruning the tree prevents overfitting. After setting up the DTC model, efforts were made to prune the tree by calculating alpha values. When the best alpha value was found, in our case 0.006194, we proceed to train the decision tree model using this optimized parameter. Thus, the pruned decision tree model is ready for making decisions on real-world data.

The relative importance of the features is also investigated by identifying which variables carry the most weight in relation to the target variable. In other words, the important features are the ones that contribute the most when making predictions. Our analysis reveals that the variable with the highest significance is "relationships", as seen on figure 19. This adds up with the correlation we found between the target variable (closed) and the explanatory variable (relationships).

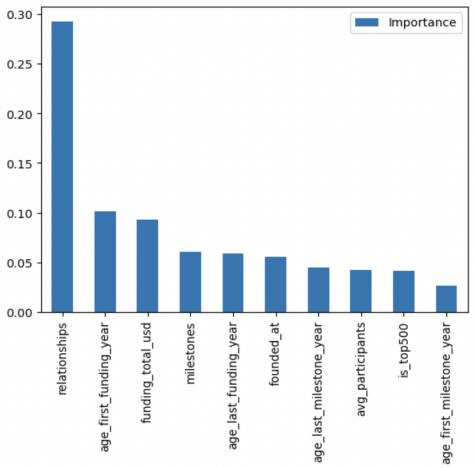


Figure 19. Importance DTC model

The model correctly predicted outcomes in the training set 79% of the time, whereas its performance on the test set was 71,9%. The close accuracy on both the training and testing set shows no sign of overfitting and are almost comparable to the logistic regression model.

When looking at the confusion matrix from the testing set on figure 20, we can conclude the following: in 53,51% of the cases, the model could predict if a startup was still open and in 18,38% of the time could predict if a startup was closed. This means that in 30,1% of the time, the models did not manage to predict if a startup was still open or had already closed. Even with a high accuracy score, like the naive bayes model, the DTC model still has a high margin of error.

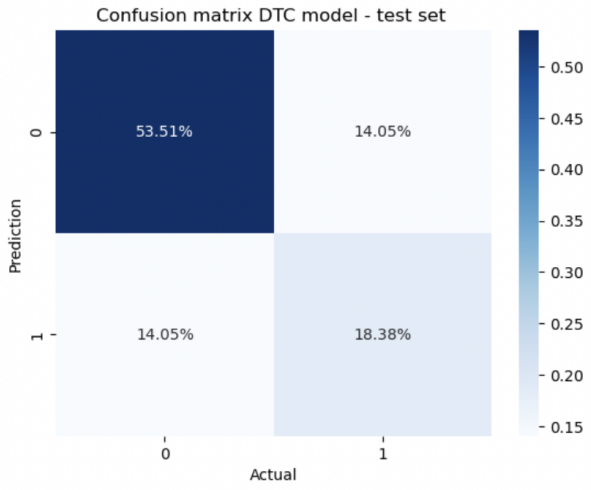


Figure 20. Confusion matrix for DTC model- testing set

6.2.4. Artificial Neural Network (ANN)

For the ANN model we have decided to conduct one simple artificial neural network and one more complex artificial neural network, as we wanted to see if a more complex model could predict with more precision the success or failure of a startup. In designing the model's architecture, specific decisions were made. Initially, we constructed three layers, with the first two consisting of 10 neurons each and employing the ReLU activation function. Subsequently, in the output layer, which consists of neurons corresponding to the number of classes in the classification task and employing the Softmax activation function. Additionally, a loss function and the Adam optimizer were implemented, facilitating efficient training on the data. An EarlyStopping callback was also incorporated to monitor the loss during training and halt the process, thereby preventing data overfitting. The EarlyStopping callback is utilized to evaluate the loss, ensuring that the model performs well in generalizing to unseen data.

The model correctly predicted outcomes in the training set 83,2% of the time, whereas its performance on the test set was 71,9%. The low accuracy score on the test set demonstrates that the model lacks the ability to generalize to new, unseen data. With a difference of 11,3%, the simple ANN model clearly shows signs of overfitting. This assumption is reinforced with the more complex ANN model. The model correctly predicted outcomes in the training set 92,55% of the time, whereas its performance on the test set was 60% as seen on figure 21. This noticeable difference raises concerns about the model's generalization performance, in other words, its ability to make accurate predictions on new, unseen data.

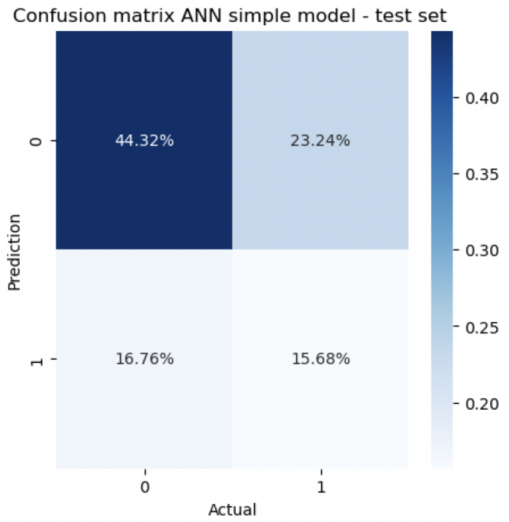


Figure 21. Confusion matrix for simple ANN model- testing set

7. Conclusion

The startup landscape is dynamic, full of possibilities and opportunities for those who want to embark on this journey. The trajectory from conceiving an idea to maturing a startup is full of challenges and therefore recognizing and understanding these challenges play a crucial role for both entrepreneurs and investors. This bachelor thesis thoroughly investigated the complex world of startups, seeking to explain factors that determine success or failure of startups.

Key factors such as location, category, timing, funding and relationship were particularly examined, providing insightful perspectives into the multifaceted dynamics that contribute to the success or failure of a startup. A thorough analysis and research revealed the importance of adapting to economic events, recognizing the impact of technological trends, and the role of strategic funding in the startup ecosystem. Additionally, the examination of various machine learning models, such as Logistic Regression, Naives Bayes, Decision Tree Classifier and Artificial Neural Networks provided insights into the predictive capacities of analytics, while also highlighting challenges of predicting the success of any startups.

The findings of this research have real-world applications for aspiring entrepreneurs and venture business investors. The results highlight the importance of building a network, achieving key objectives and gaining acknowledgment within industry recognized standards. Besides contributing to existing knowledge about startups, this thesis can serve as a guide for navigating the dynamic and uncertain landscape of startup ventures. The insights gained from this research provide a foundation for informed decision-making and strategic planning in the ever-changing world of startups.

There are various elements influencing startup journeys and while writing this thesis, it became evident that success is a multifaceted interplay of strategic decisions, adaptability, and resilience. And maybe a little bit of luck.

8. References

Analytixlabs (2022) https://www.analytixlabs.co.in/blog/naive-bayes-machine-learning/#1_Gaussian_Naive_Bayes (Accessed : 26 mars 2024)

Analytixlabs (2022). https://www.analytixlabs.co.in/blog/decision-tree-algorithm/#Features_and_Characteristics (Accessed : 26 mars 2024)

Analytixlabs (2023). Available at: https://www.analytixlabs.co.in/blog/logistic-regression/#Advantages_and_Disadvantages_of_Logistic_Regression (Accessed : 26 mars 2024)

Bensley, T. (2022) Success factors for startups: what makes for a successful startup in 2022? *QontoBlog*. Available at: https://qonto.com/en/blog/creators/tools-tips/startup-success-factors (Accessed: 25 mars 2024)

Cuofano, G. (2024) The Five Key Factors That Lead To Successful Tech Startups, *FourWeekMBA*. Available at: https://fourweekmba.com/startup-success-factors/ (Accessed: 25 mars 2024)

Devrix (2023) . Available at: https://devrix.com/tutorial/10-key-success-factors-for-startups/ (Accessed: 25 mars 2024)

Embroker, (2024) 106 Must-Know Startup Statistics for 2024, *Embroker.* Available at: https://www.embroker.com/blog/startup-statistics/#:~:text=03-,Startup,About%2090%25%20of%20startups%20fail.&text=10%25%20of%20startups%20fail%20within%20the%20first%20year.&text=Across%20all%20industries%2C%20startup%20failure,be%20close%20to%20the%20same.&text=Failure%20is%20most%20common%20for,70%25%20falling%20into%20this%20category. (Accessed: 22 January 2024)

Faster Capital, (2023). Economic conditions: How economic conditions affect market saturation, *Faster Capital.* Available at: https://fastercapital.com/content/Economic-

conditions--How-economic-conditions-affect-market-saturation.html (Accessed: 31 January 2024)

Faster Capital, (2023). The history of startups From early days to today*, Faster Capital.* Available at: https://fastercapital.com/content/The-history-of-startups--From-early-days-to-today.html (Accessed: 30 January 2024)

Faster Capital, (2023). The Importance of a High Growth Rate for Startups*, Faster Capital.* Available at: https://fastercapital.com/content/The-Importance-of-a-High-Growth-Rate-for-Startups.html#The-Importance-of-a-High-Growth-Rate-for-Startups (Accessed: 30 January 2024)

Faster Capital, (2023). The Stages in the Life Cycle of a Startup*, Faster Capital.* Available at: https://fastercapital.com/content/The-Stages-in-the-Life-Cycle-of-a-Startup.html (Accessed: 1 February 2024)

Faster Capital, (2023) The Top Startup Cities in the US, *Faster Capital* Available at: https://fastercapital.com/content/The-Top-Startup-Cities-in-the-US.html (Accessed: 20 January 2024)

Faster Capital, (2023). What Makes a Successful Business Location Startup, *Faster Capital.* Available at: https://fastercapital.com/content/What-Makes-a-Successful-Business-Location-Startup.html#:~:text=A%20successful%20startup%20needs%20to,the%20right%20resources%20and%20customers. (Accessed: 1 February 2024)

Faster Capital, (2023). What Role Does Networking Play In Raising Capital For Startups, *Faster Capital.* Available at: https://fastercapital.com/questions/what-role-does-networking-play-in-raising-capital-for-startups.html (Accessed: 31 January 2024)

Faster Capital, (2023) Why Silicon Valley is the best place for startups, *Faster Capital.* Available at: https://fastercapital.com/content/Why-Silicon-Valley-is-the-best-place-for-

startups.html (Accessed: 20 January 2024)

Fonseca, M. (2023) The 7 stages of a startup, from ideation to growth and maturity, *Latitud.* Available at: https://www.latitud.com/blog/stages-of-a-startup (Accessed: 1 February 2024).

Gascon, C. & Karson, E. (2017) *Growth in Tech Sector Returns to Glory Days of the 1990s.* Available at: https://www.stlouisfed.org/publications/regional-economist/second-quarter-2017/growth-in-tech-sector-returns-to-glory-days-of-the-1990s#:~:text=The%20technology%20sector%20has%20a,36%20percent%20over%20the%20period (Accessed: 20 January 2024)

Grant, M. (2023) What a Startup Is and What's Involved in Getting One Off the Ground, *Investopedia.* Available at: https://www.investopedia.com/terms/s/startup.asp#:~:text=Investopedia%20%2F%20Laura%20Porter-,What%20Is%20a%20Startup%3F,they%20believe%20there%20is%20demand. (Accessed: 30 January 2024)

Hayes, A. (2024) Venture Capital: What Is VC and How Does It Work?, *Investopedia* Available at: https://www.investopedia.com/terms/v/venturecapital.asp (Accessed: 31 January 2024)

Hayes, A. (2021) What Is a Fortune 500 Company? How Companies Are Ranked, *Investopedia.* Available at: https://www.investopedia.com/terms/f/fortune500.asp (Accessed: 1 February 2024)

Hoque, F. (2012) Why Most Venture-Backed Companies Fail, *Fast Company.* Available at: https://www.fastcompany.com/3003827/why-most-venture-backed-companies-fail (Accessed: 31 January 2024)

Housel, M. (2020) *The psychology of money : timeless lessons on wealth, greed, and happiness*. Great Britain: Harriman House.

Hudziy, O. (No date) Top 6 Industries for Startups in 2024, *Invertia.* Available at: https://inveritasoft.com/article-top-6-industries-for-

startups#:~:text=The%20current%20fastest%2Dgrowing%20sectors,the%20spotlight%20and%20remain%20effective. (Accessed: 31 January 2024)

Khandelwal, R. (2020) Techniques for Handling Imbalanced Classification Datasets*, Medium* Available at: https://towardsdatascience.com/techniques-for-handling-imbalanced-classification-datasets-5ee58b0b5e7a (Accessed: 31 January 2024)

Max Obrazchykov, M. (2023) The Key Factors for Startup Success in 2023, *Linkedin*. Available at: https://www.linkedin.com/pulse/key-factors-startup-success-2023-maksim-obrazchikov (Accessed: 25 mars 2024)

McGowan , E. (2022). What Is a Startup Company, Anyway?*, Startups* Available at: https://www.startups.com/library/expert-advice/what-is-a-startup-company (Accessed: 31 January 2024)

Nadeem, A. (2023) Digital Age's Impact on Wanderlust: Transforming Travel with Technology, *Medium.* Available at: https://medium.com/@ahmadfalcon0/digital-ages-impact-on-wanderlust-transforming-travel-with-technology-1578bb11e667 (Accessed: 31 January 2024)

Oremus, W. (2013) Google's Big Break, *Slate.* Available at: https://slate.com/business/2013/10/googles-big-break-how-bill-gross-goto-com-inspired-the-adwords-business-model.html (Accessed: 25 mars 2024)

Perry, E. (2023) What is networking and why is it so important?*, Betterup* Available at: https://www.betterup.com/blog/networking (Accessed: 31 January 2024)

Pew Research Center, (2012). Public More Optimistic about Economy, But Concerns Persist, *Pew Research Center.* Available at: https://www.pewresearch.org/politics/2012/02/16/public-more-optimistic-about-economy-but-concerns-persist/ (Accessed: 31 January 2024)

Ray, S. (2024). Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier, *Analyticsvidhya*. Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ (Accessed : 26 mars 2024)

Reiff, N. (2023) Series Funding: A, B, and C, *Investopedia.* Available at: https://www.investopedia.com/articles/personal-finance/102015/series-b-c-funding-what-it-all-means-and-how-it-works.asp (Accessed: 22 January 2024)

Saini, A (2024). Decision Tree – A Step-by-Step Guide, *Analyticsvidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/ (Accessed : 26 mars 2024)

Shathi, S. (2024) 2024 Predictions For Top Startup Sectors In A Nutshell, *Entrepreneur.* Available at: https://www.entrepreneur.com/en-in/news-and-trends/2024-predictions-for-top-startup-sectors-in-a-nutshell/467761 (Accessed: 20 January 2024)

Singh, S. *(2018)* Understanding the Bias-Variance Tradeoff, *Medium.* Available at: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229 (Accessed: 31 January 2024)

Statista Research Department, (2024). *Digitalization of the travel industry - statistics & facts*. Available at: https://www.statista.com/topics/7589/digitalization-of-the-travel-industry/#topicOverview (Accessed: 31 January 2024)

Turner, K. (2022) 5 factors that make a Start-up Successful, *Keiretsu Forum*. Available at: https://www.k4northwest.com/articles/5-factors-that-make-a-start-up-successful (Accessed: 25 mars 2024)

Weltman, B. (2023) *10 Years After the Financial Crisis: The Impact on Small Business.* Available at: https://www.investopedia.com/small-business/10-years-after-financial-crisis-impact-small-business/ (Accessed: 31 January 2024)

9. Code

Bachelor Thesis

In [70]:
```
#import main packages
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler

#For the map
import folium


# For confusion matrix
from seaborn import heatmap
from sklearn.metrics import confusion_matrix

#Accuracy on models
from sklearn.metrics import accuracy_score


#Import library for training and testing
from sklearn.model_selection import train_test_split

#Import library for Logistic Regression
from sklearn.linear_model import LogisticRegression

#Import library for Gaussian Naive Bayes
from sklearn.naive_bayes import GaussianNB


#Import library for Decision tree Classifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.model_selection import cross_val_score


#Import library for Artificial Neuron Network
from keras.models import Sequential
from keras.layers import Dense
from keras.callbacks import EarlyStopping
```

1. Data exploration

In [2]:
```
df = pd.read_csv('startup.csv')
df.head()
```

Out[2]:

	Unnamed: 0	state_code	latitude	longitude	zip_code	id	city	Unnam
0	1005	CA	42.358880	-71.056820	92101	c:6669	San Diego	N
1	204	CA	37.238916	-121.973718	95032	c:16283	Los Gatos	N
2	1001	CA	32.901049	-117.192656	92121	c:65620	San Diego	San Di CA 92
3	738	CA	37.320309	-122.050040	95014	c:42668	Cupertino	Cuper CA 95
4	1002	CA	37.779281	-122.419236	94105	c:65806	San Francisco	Franci CA 94

5 rows × 49 columns

In [3]:
```
df.shape
```

Out[3]:
```
(923, 49)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 923 entries, 0 to 922
Data columns (total 49 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Unnamed: 0                923 non-null    int64  
 1   state_code                923 non-null    object 
 2   latitude                  923 non-null    float64
 3   longitude                 923 non-null    float64
 4   zip_code                  923 non-null    object 
 5   id                        923 non-null    object 
 6   city                      923 non-null    object 
 7   Unnamed: 6                430 non-null    object 
 8   name                      923 non-null    object 
 9   labels                    923 non-null    int64  
 10  founded_at                923 non-null    object 
 11  closed_at                 335 non-null    object 
 12  first_funding_at          923 non-null    object 
 13  last_funding_at           923 non-null    object 
 14  age_first_funding_year    923 non-null    float64
 15  age_last_funding_year     923 non-null    float64
 16  age_first_milestone_year  771 non-null    float64
 17  age_last_milestone_year   771 non-null    float64
 18  relationships             923 non-null    int64  
 19  funding_rounds            923 non-null    int64  
 20  funding_total_usd         923 non-null    int64  
 21  milestones                923 non-null    int64  
 22  state_code.1              922 non-null    object 
 23  is_CA                     923 non-null    int64  
 24  is_NY                     923 non-null    int64  
 25  is_MA                     923 non-null    int64  
 26  is_TX                     923 non-null    int64  
 27  is_otherstate             923 non-null    int64  
 28  category_code             923 non-null    object 
 29  is_software               923 non-null    int64  
 30  is_web                    923 non-null    int64  
 31  is_mobile                 923 non-null    int64  
 32  is_enterprise             923 non-null    int64  
 33  is_advertising            923 non-null    int64  
 34  is_gamesvideo             923 non-null    int64  
 35  is_ecommerce              923 non-null    int64  
 36  is_biotech                923 non-null    int64  
 37  is_consulting             923 non-null    int64  
 38  is_othercategory          923 non-null    int64  
 39  object_id                 923 non-null    object 
 40  has_VC                    923 non-null    int64  
 41  has_angel                 923 non-null    int64  
 42  has_roundA                923 non-null    int64  
 43  has_roundB                923 non-null    int64  
 44  has_roundC                923 non-null    int64  
 45  has_roundD                923 non-null    int64  
 46  avg_participants          923 non-null    float64
 47  is_top500                 923 non-null    int64  
 48  status                    923 non-null    object 
dtypes: float64(7), int64(28), object(14)
memory usage: 353.5+ KB
```

In [5]:
```
 # Look at the counts of status aquired/closed
df['status'].value_counts()
```

Out[5]:
```
acquired    597
closed      326
Name: status, dtype: int64
```

1.1. Reformate dates

Founded_at

In [6]:
```
#Access our file
file_path = 'startup.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)

# Convert the 'founded_at' column to datetime,
#handling errors by setting invalid dates to NaT
df['founded_at'] = pd.to_datetime(df['founded_at'],
                                  format='%m/%d/%Y', errors='coerce')

# Extract the year from the 'founded_at' column
df['founded_at'] = df['founded_at'].dt.year

# Overwrite the file with the updated DataFrame
df.to_csv(file_path, index=False)
```

first_funding_at

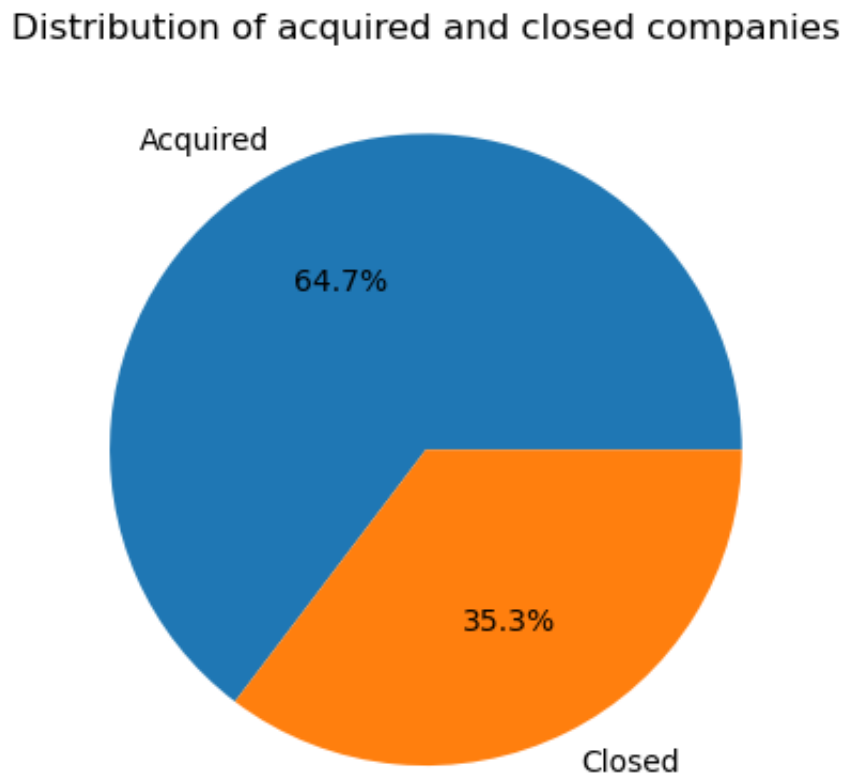
In [7]:
```
#Access our file
file_path = 'startup.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)

# Convert the 'founded_at' column to datetime,
#handling errors by setting invalid dates to NaT
df['first_funding_at'] = pd.to_datetime(df['first_funding_at'],
                                        format='%m/%d/%Y', errors='coerce

# Extract the year from the 'founded_at' column
df['first_funding_at'] = df['first_funding_at'].dt.year

# Overwrite the file with the updated DataFrame
df.to_csv(file_path, index=False)
```

last_funding_at

In [8]:
```
#Access our file
file_path = 'startup.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)

# Convert the 'founded_at' column to datetime,
# handling errors by setting invalid dates to NaT
df['last_funding_at'] = pd.to_datetime(df['last_funding_at'],
                                       format='%m/%d/%Y', errors='coerce'

# Extract the year from the 'founded_at' column
df['last_funding_at'] = df['last_funding_at'].dt.year

# Overwrite the file with the updated DataFrame
df.to_csv(file_path, index=False)
```

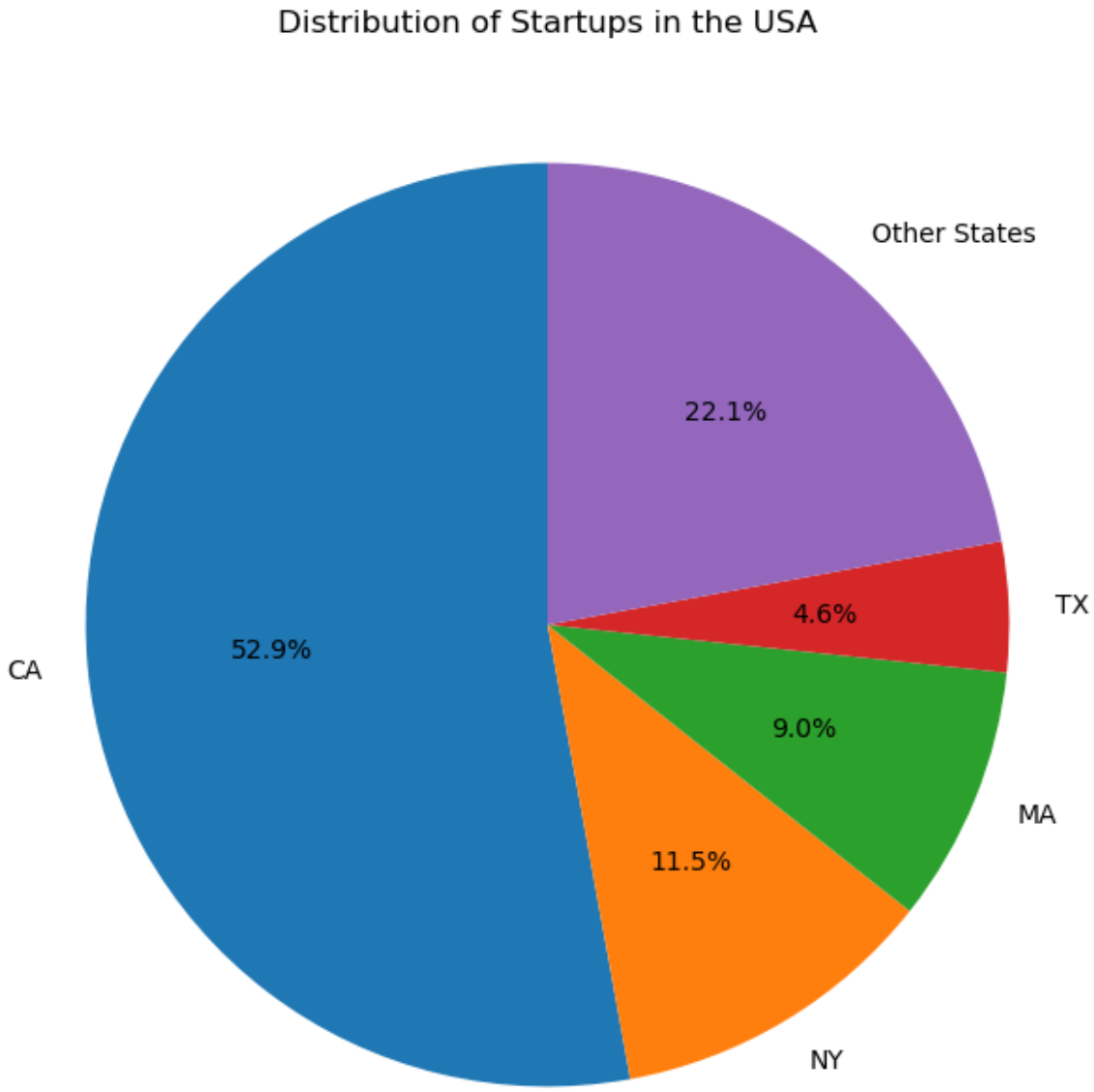
2. Data vizualization

In [9]:
```
#Pie chart
#Pie plot to check distribution of acquired and not acquired
labels= ['Acquired', 'Closed']
plt.pie(df['status'].value_counts(),
        labels= labels,
        autopct='%1.1f%%')
plt.title('Distribution of acquired and closed companies')
plt.show()
```



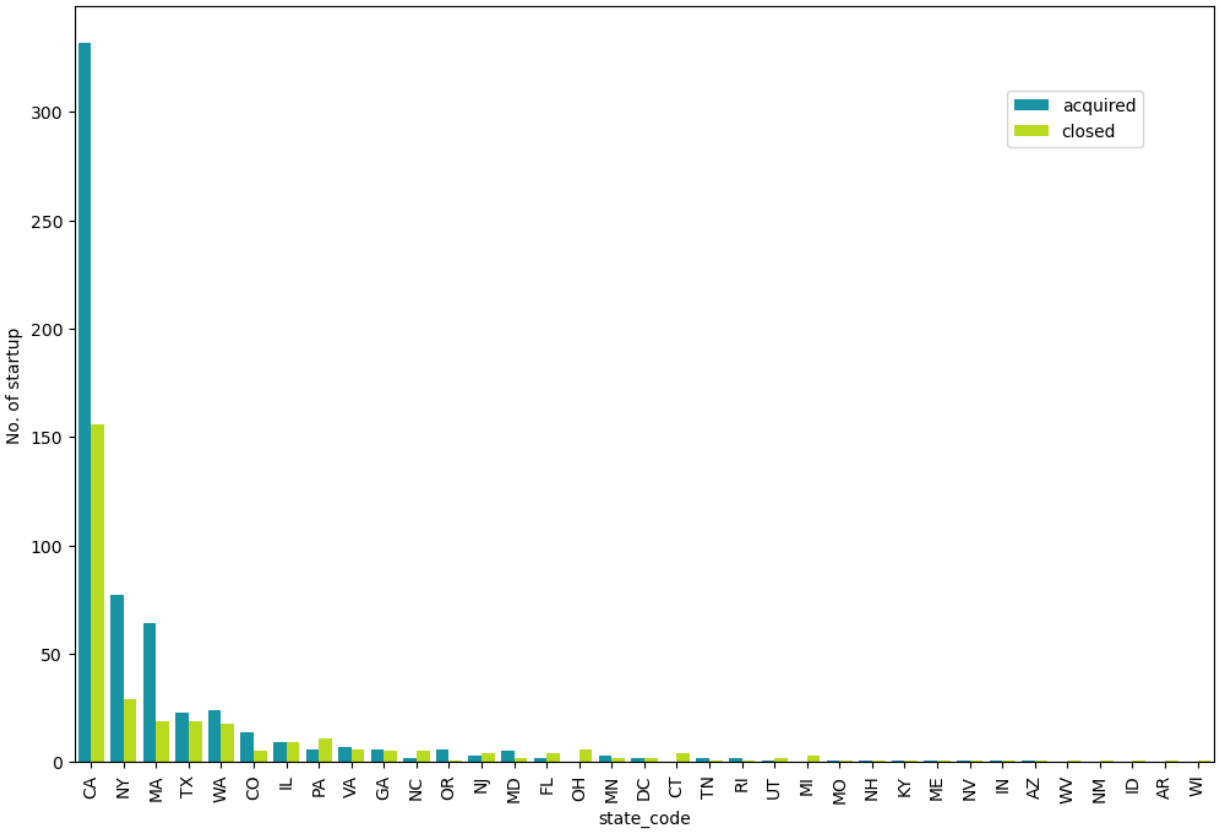
Distribution of the startups in each state

In [10]:
```
# Select the states of interest
states_of_interest = ['CA', 'TX', 'MA', 'NY']

# Filter the DataFrame to include only the states of interest
filtered_df = df[df['state_code'].isin(states_of_interest)]

# Count the startups in each state
startup_counts = filtered_df['state_code'].value_counts()

# Create a new category for the remaining states (others)
remaining_states_count = len(df) - startup_counts.sum()
startup_counts['Other States'] = remaining_states_count

# Plotting a pie chart
startup_counts.plot.pie(autopct='%1.1f%%',
                        startangle=90, figsize=(8, 8))
plt.title('Distribution of Startups in the USA')
plt.ylabel('')

# Show the plot
plt.show()
```



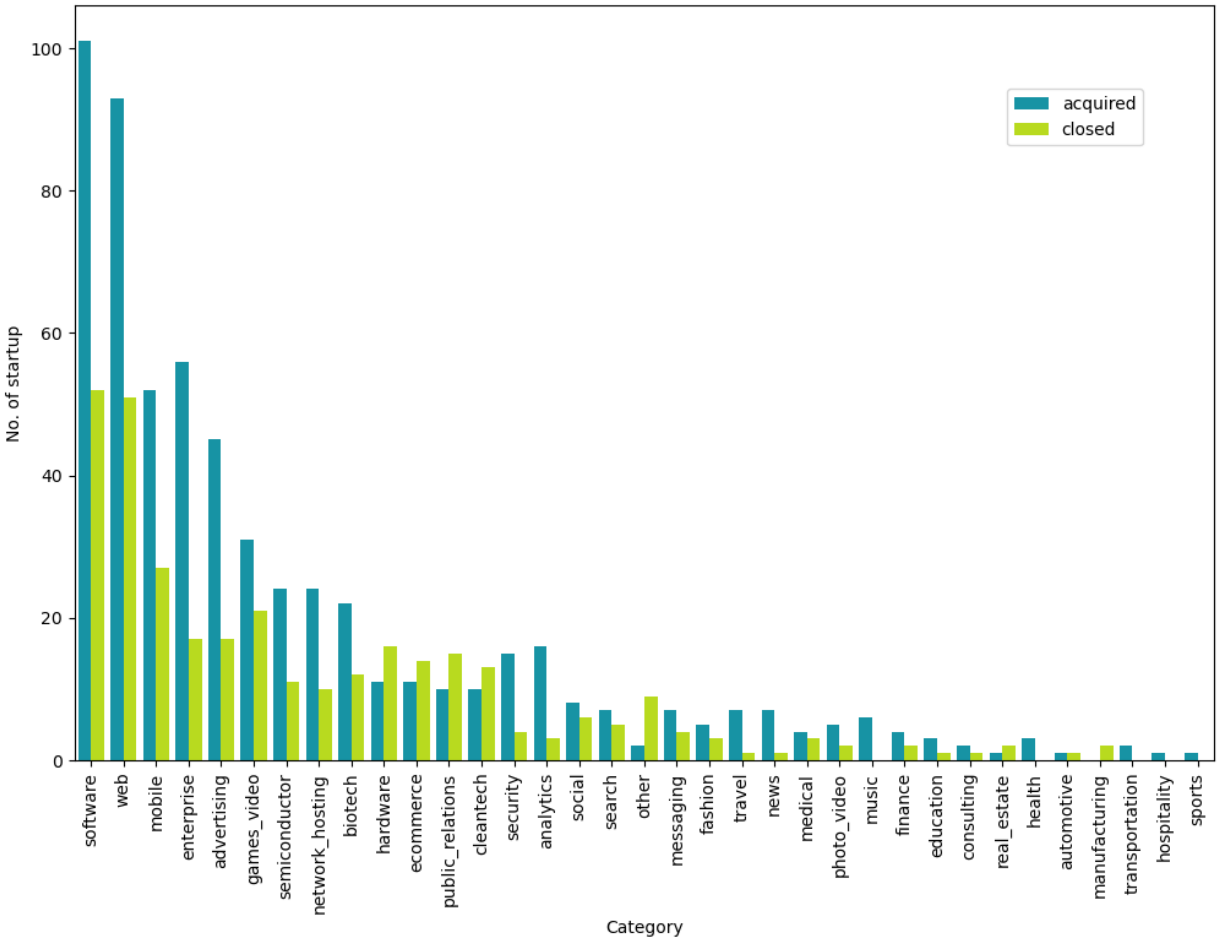
Which state has the most start ups (still open VS closed)

In [11]:
```
fig, ax = plt.subplots(figsize=(12,8))

plot_1 = sns.countplot(x="state_code",
                       hue="status", data=df,
                       palette="nipy_spectral",
              order=df.state_code.value_counts().index)

plot_1 = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plot_1 = ax.set(xlabel="state_code", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
plt.show()
```



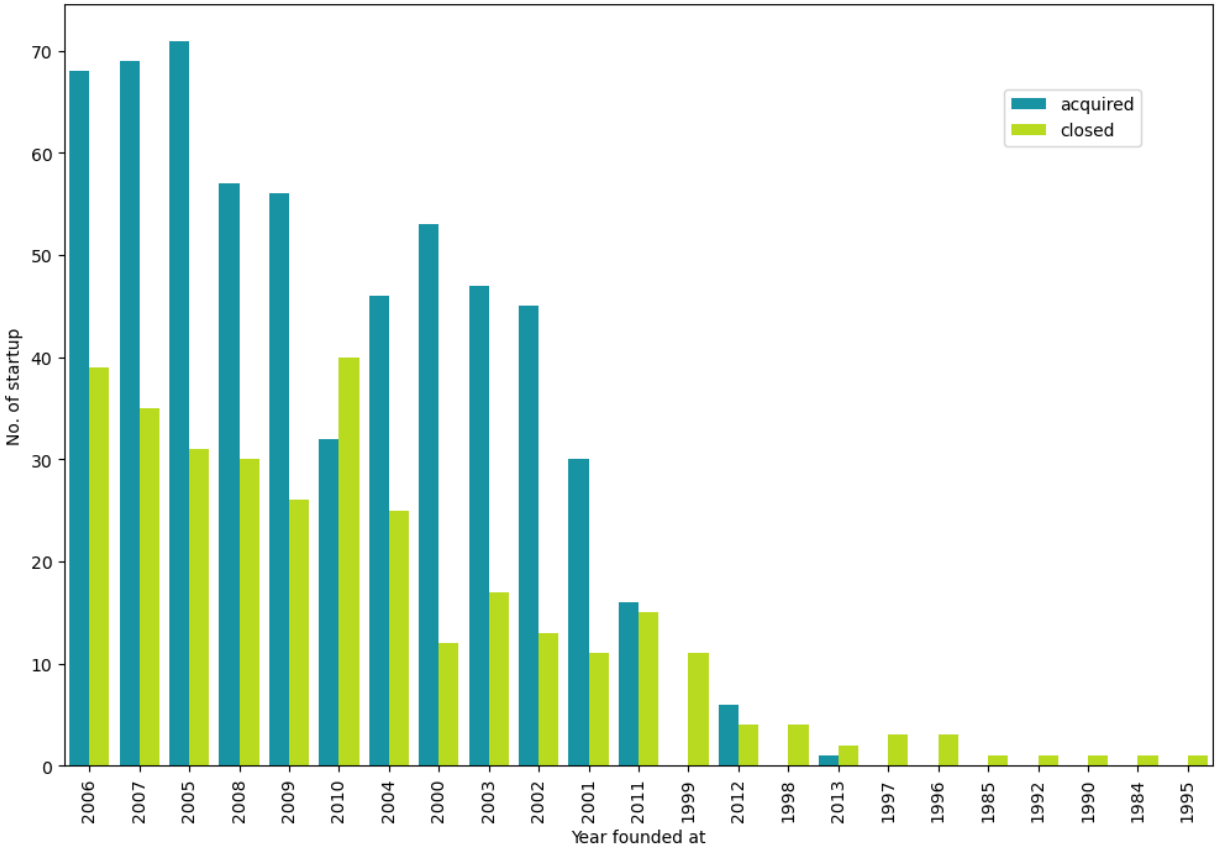
Which category has the largest number of startup

In [12]:
```
fig, ax = plt.subplots(figsize=(12,8))

plot_3 = sns.countplot(x="category_code",
                       hue="status", data=df,
                       palette="nipy_spectral",
              order=df.category_code.value_counts().index)

plot_3 = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plot_3 = ax.set(xlabel="Category", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
plt.show()
```



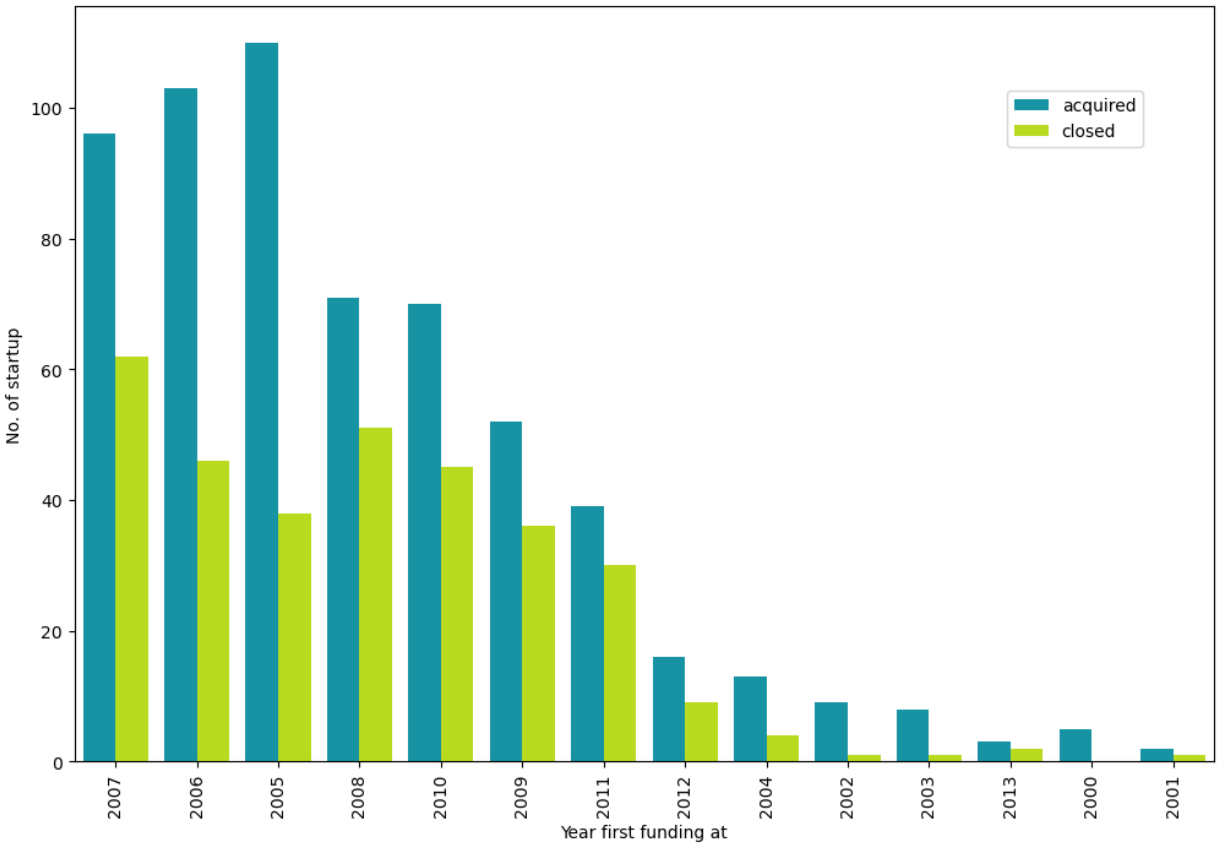
Opening and closing of startups througout the years

In [13]:
```
fig, ax = plt.subplots(figsize=(12,8))

plot_4 = sns.countplot(x="founded_at", hue="status",
                       data=df, palette="nipy_spectral",
              order=df.founded_at.value_counts().index)

plot_4 = ax.set_xticklabels(ax.get_xticklabels(),
                            rotation=90)
plot_4 = ax.set(xlabel="Year founded at",
                ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
plt.show()
```



Which year first funding happened

In [14]:
```
fig, ax = plt.subplots(figsize=(12,8))

df_sorted = df.sort_values(by='first_funding_at')

plot_5 = sns.countplot(x="first_funding_at",
                       hue="status", data=df,
                       palette="nipy_spectral",
              order=df.first_funding_at.value_counts().index)

plot_5 = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plot_5 = ax.set(xlabel="Year first funding at",
                ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
plt.show()
```



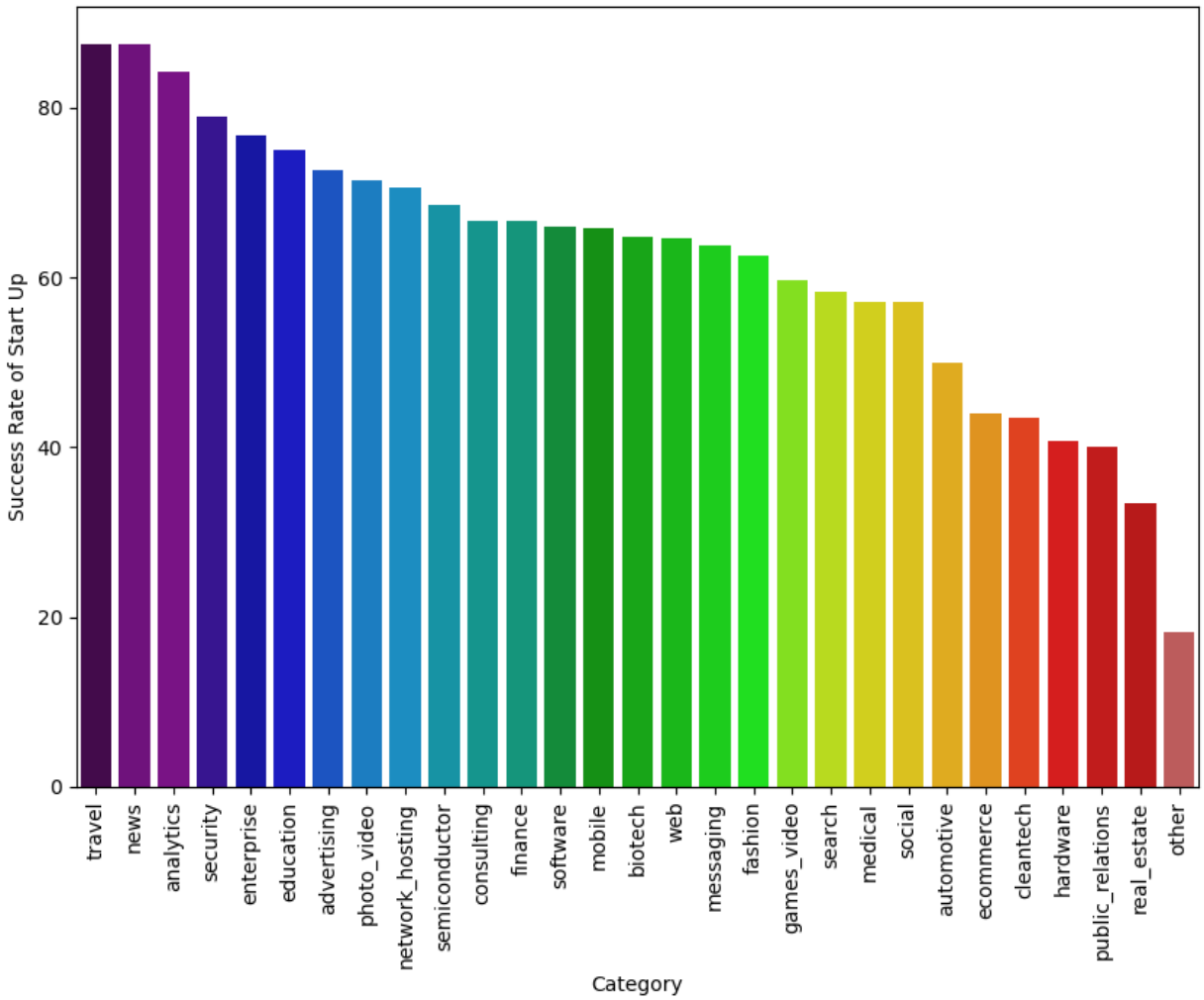
Which category has the largest number Success Rate

In [15]:
```
data1 = df[df['status']=='acquired'].groupby(['category_code']).agg({'sta
data1.columns=['category_code','total_success']

data2 = df[df['status']=='closed'].groupby(['category_code']).agg({'statu
data2.columns=['category_code','total_closed']

data3=df.groupby(['category_code']).agg({'status':'count'}).reset_index()
data3.columns=['category_code','total_startup']

data1= data1.merge(data2, on='category_code')
data1= data1.merge(data3, on='category_code')

data1['success_rate']= round((data1['total_success'] / data1['total_start

most_succes_rate = data1.sort_values('success_rate', ascending=False)
most_succes_rate
```

Out[15]:

	category_code	total_success	total_closed	total_startup	success_rate
27	travel	7	1	8	87.50
17	news	7	1	8	87.50
1	analytics	16	3	19	84.21
23	security	15	4	19	78.95
8	enterprise	56	17	73	76.71
7	education	3	1	4	75.00
0	advertising	45	17	62	72.58
19	photo_video	5	2	7	71.43
16	network_hosting	24	10	34	70.59
24	semiconductor	24	11	35	68.57
5	consulting	2	1	3	66.67
10	finance	4	2	6	66.67
26	software	101	52	153	66.01
15	mobile	52	27	79	65.82
3	biotech	22	12	34	64.71
28	web	93	51	144	64.58
14	messaging	7	4	11	63.64
9	fashion	5	3	8	62.50
11	games_video	31	21	52	59.62
22	search	7	5	12	58.33
13	medical	4	3	7	57.14
25	social	8	6	14	57.14
2	automotive	1	1	2	50.00
6	ecommerce	11	14	25	44.00
4	cleantech	10	13	23	43.48
12	hardware	11	16	27	40.74
20	public_relations	10	15	25	40.00
21	real_estate	1	2	3	33.33
18	other	2	9	11	18.18

In [16]:
```
fig, ax = plt.subplots(figsize=(10,7))
plot= sns.barplot(x="category_code", y="success_rate", data=most_succes_r

            palette="nipy_spectral", ax=ax)


_ = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
_ = ax.set(xlabel="Category", ylabel="Success Rate of Start Up")
```



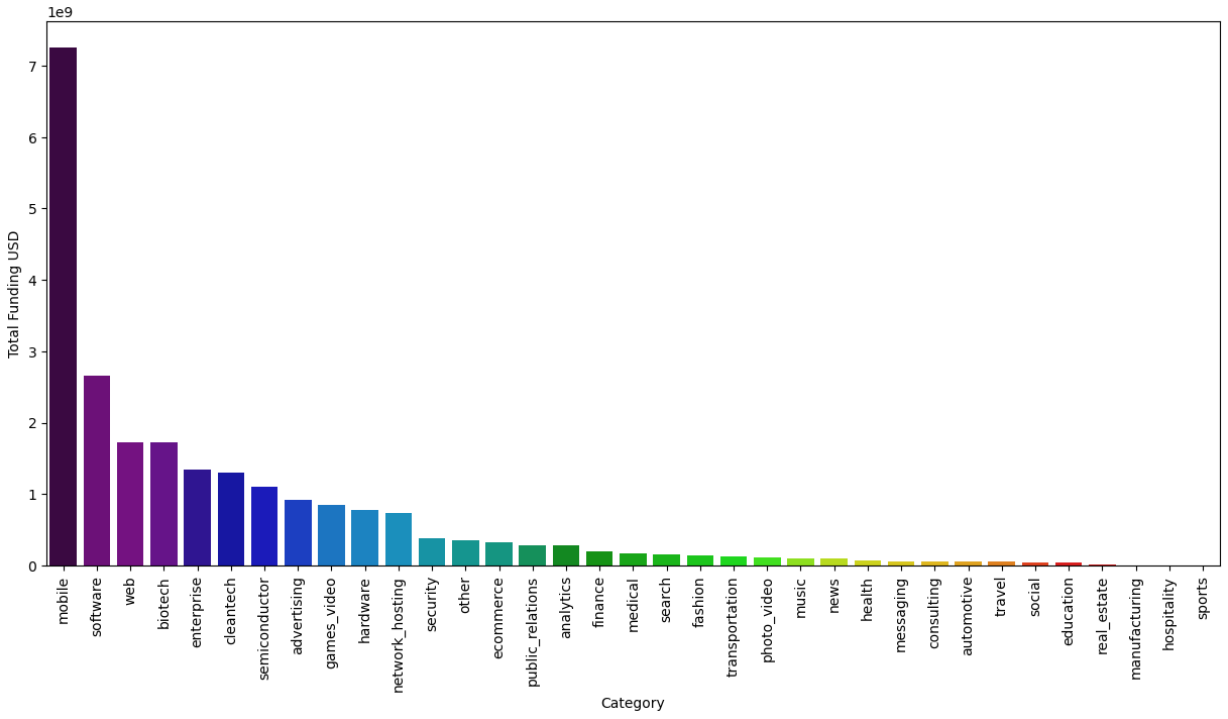
Which category gets the most fundings

In [17]:
```
#First create a data frame with the total fundings sorted by category
funding_sorted_category = pd.pivot_table(df,
              index=['category_code'],
              values=['funding_total_usd'],
              aggfunc=['sum']
              ).reset_index()

funding_sorted_category.columns = ['category_code', 'funding_total_usd']
funding_sorted_category = funding_sorted_category.sort_values(['funding_t
                                                               ascending =
funding_sorted_category.head(10)
```

Out[17]:

	category_code	funding_total_usd
18	mobile	7263750881
30	software	2657598865
34	web	1729035436
3	biotech	1723699484
8	enterprise	1338882096
4	cleantech	1300284730
28	semiconductor	1105156970
0	advertising	918619012
11	games_video	844643530
12	hardware	773938873

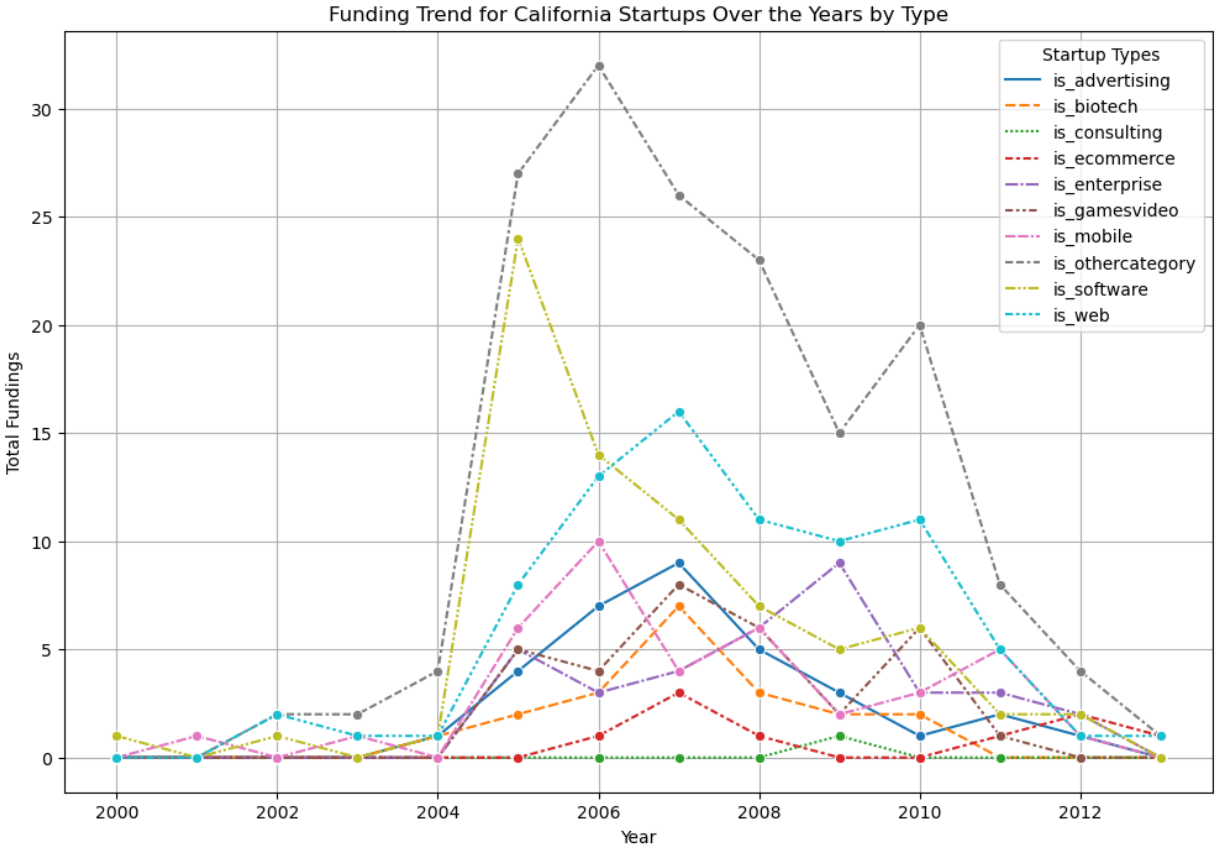
In [18]:
```
fig, ax = plt.subplots(figsize=(15,7))
plot_6 = sns.barplot(x="category_code", y="funding_total_usd",
                     data=funding_sorted_category,
                  palette="nipy_spectral", ax=ax)
plot_6 = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plot_6 = ax.set(xlabel="Category", ylabel="Total Funding USD")
```



Startups by category and state- fundings throughout the years

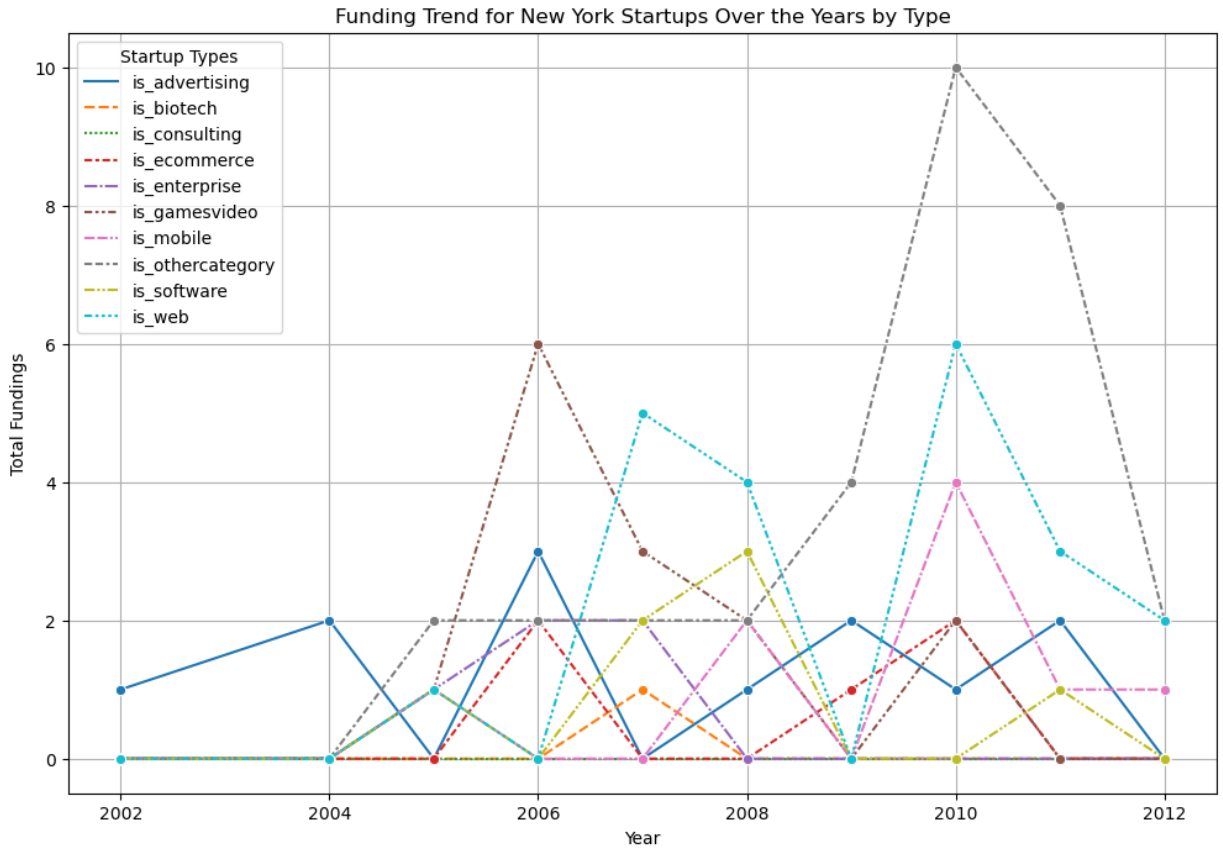
California

In [20]:
```
# Filter startups only from California
ca_startups = df[df['is_CA'] == True]

# Create a pivot table to sum the number of fundings for each type of sta
funding_trend = ca_startups.pivot_table(index='first_funding_at',
                                        values=['is_software','is_web',
                                                'is_ecommerce','is_mobil
                                                'is_enterprise' ,
                                                'is_advertising' ,'is_ga
                                                'is_biotech' ,'is_consul
                                                'is_othercategory'],
                                        aggfunc='sum')

# Plotting the trend
plt.figure(figsize=(12, 8))
sns.lineplot(data=funding_trend, marker='o')

# Set legend with specified labels
plt.legend(title='Startup Types')

plt.title('Funding Trend for California Startups Over the Years by Type')
plt.xlabel('Year')
plt.ylabel('Total Fundings')
plt.grid(True)
plt.show()
```



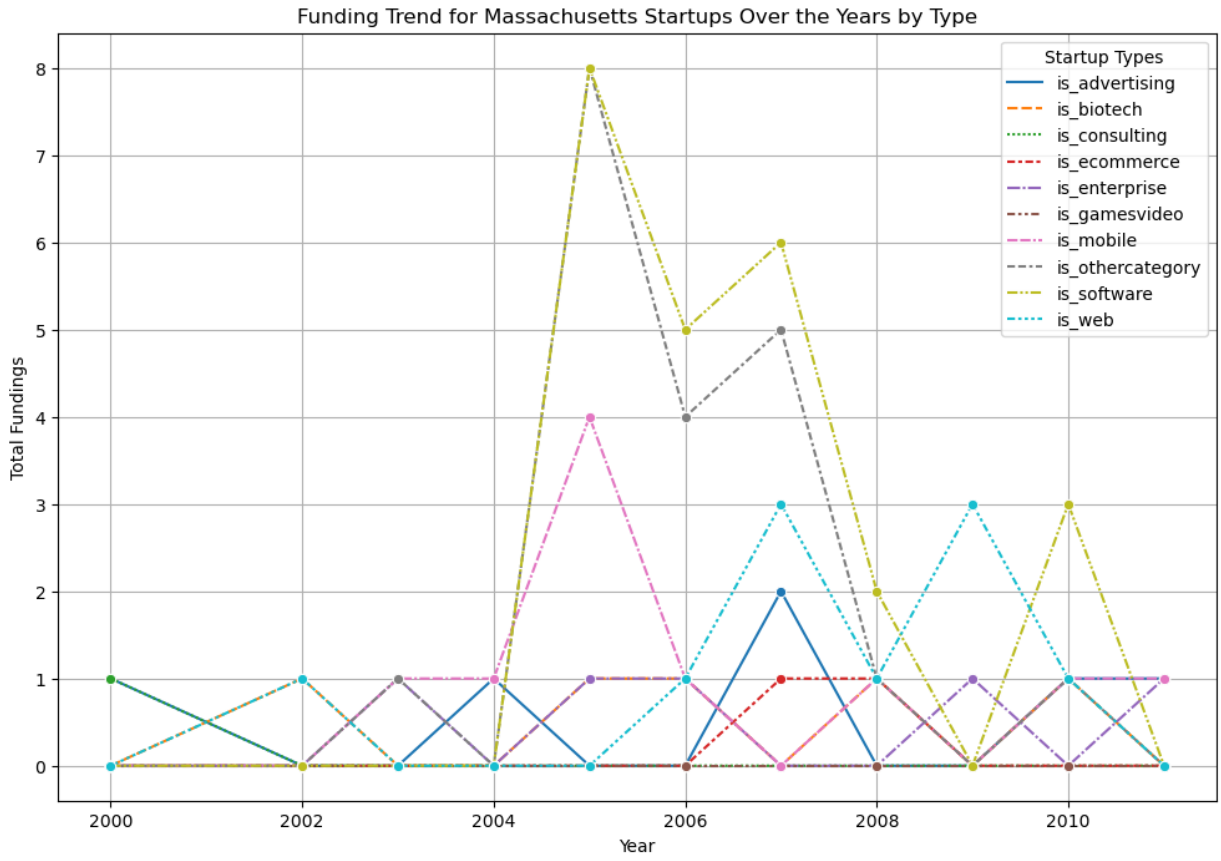
New York

In [21]:
```
# Filter startups only from New York
ny_startups = df[df['is_NY'] == True]

# Create a pivot table to sum the number of fundings for each type of sta
funding_trend = ny_startups.pivot_table(index='first_funding_at',
                                        values=['is_software','is_web',
                                                'is_ecommerce','is_mobil
                                                'is_enterprise' ,
                                                'is_advertising' ,'is_ga
                                                'is_biotech' ,'is_consul
                                                'is_othercategory'],
                                        aggfunc='sum')

# Plotting the trend
plt.figure(figsize=(12, 8))
sns.lineplot(data=funding_trend, marker='o')

# Set legend with specified labels
plt.legend(title='Startup Types')

plt.title('Funding Trend for New York Startups Over the Years by Type')
plt.xlabel('Year')
plt.ylabel('Total Fundings')
plt.grid(True)
plt.show()
```



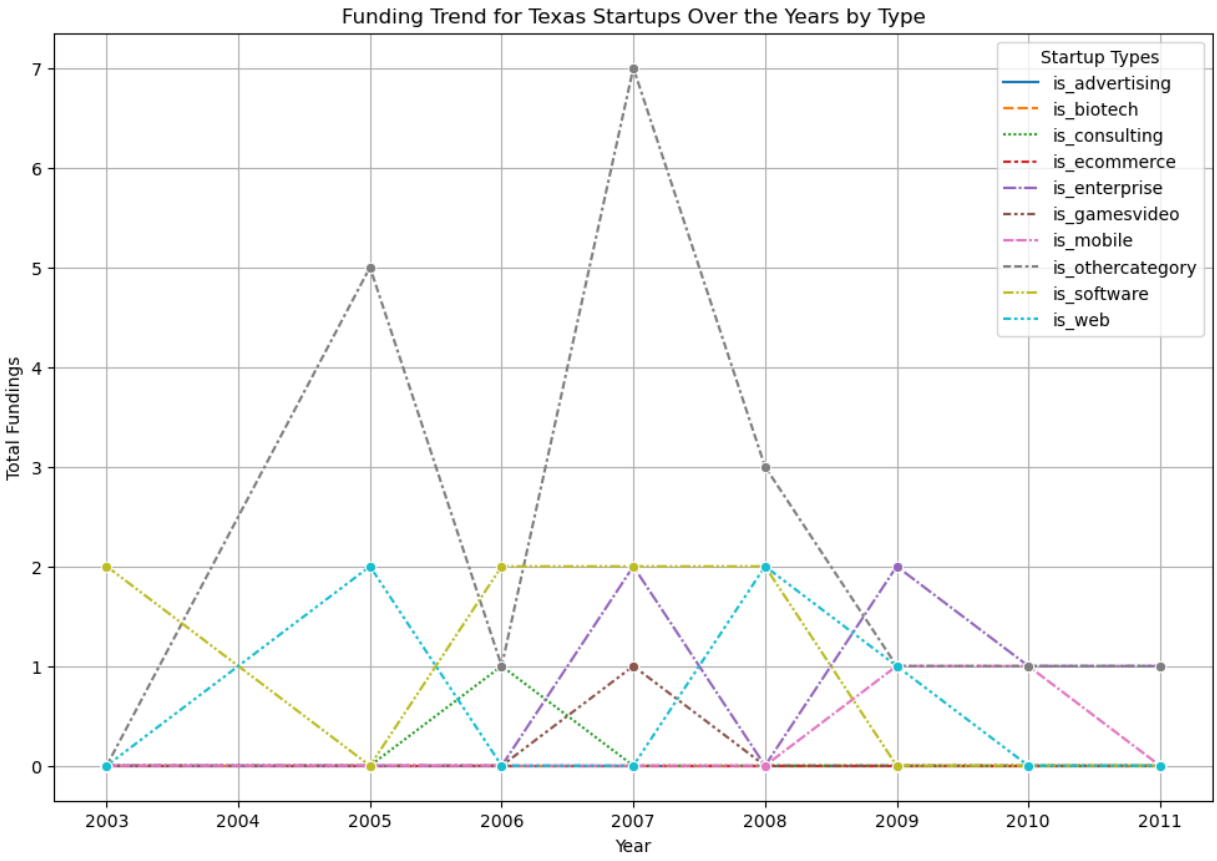
Massachusetts

In [22]:
```
# Filter startups only from Massachusetts
ma_startups = df[df['is_MA'] == True]

# Create a pivot table to sum the number of fundings for each type of sta
funding_trend = ma_startups.pivot_table(index='first_funding_at',
                                        values=['is_software','is_web',
                                                'is_ecommerce','is_mobil
                                                'is_enterprise' ,
                                                'is_advertising' ,'is_ga
                                                'is_biotech' ,'is_consul
                                                'is_othercategory'],
                                        aggfunc='sum')

# Plotting the trend
plt.figure(figsize=(12, 8))
sns.lineplot(data=funding_trend, marker='o')

# Set legend with specified labels
plt.legend(title='Startup Types')

plt.title('Funding Trend for Massachusetts Startups Over the Years by Typ
plt.xlabel('Year')
plt.ylabel('Total Fundings')
plt.grid(True)
plt.show()
```



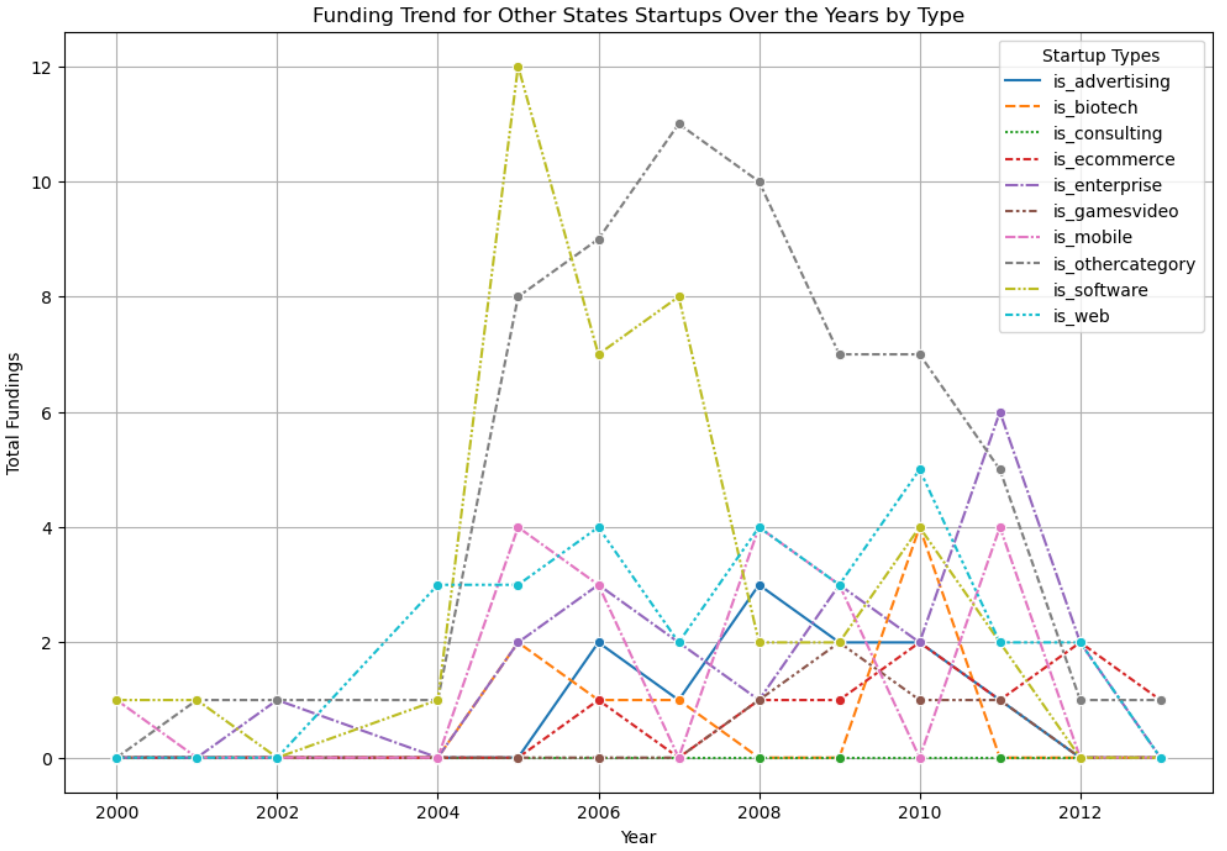
Texas

In [23]:
```
# Filter startups only from Texas
tx_startups = df[df['is_TX'] == True]

# Create a pivot table to sum the number of fundings for each type of sta
funding_trend = tx_startups.pivot_table(index='first_funding_at',
                                        values=['is_software','is_web',
                                                'is_ecommerce','is_mobil
                                                'is_enterprise' ,
                                                'is_advertising' ,'is_ga
                                                'is_biotech' ,'is_consul
                                                'is_othercategory'],
                                        aggfunc='sum')

# Plotting the trend
plt.figure(figsize=(12, 8))
sns.lineplot(data=funding_trend, marker='o')

# Set legend with specified labels
plt.legend(title='Startup Types')

plt.title('Funding Trend for Texas Startups Over the Years by Type')
plt.xlabel('Year')
plt.ylabel('Total Fundings')
plt.grid(True)
plt.show()
```



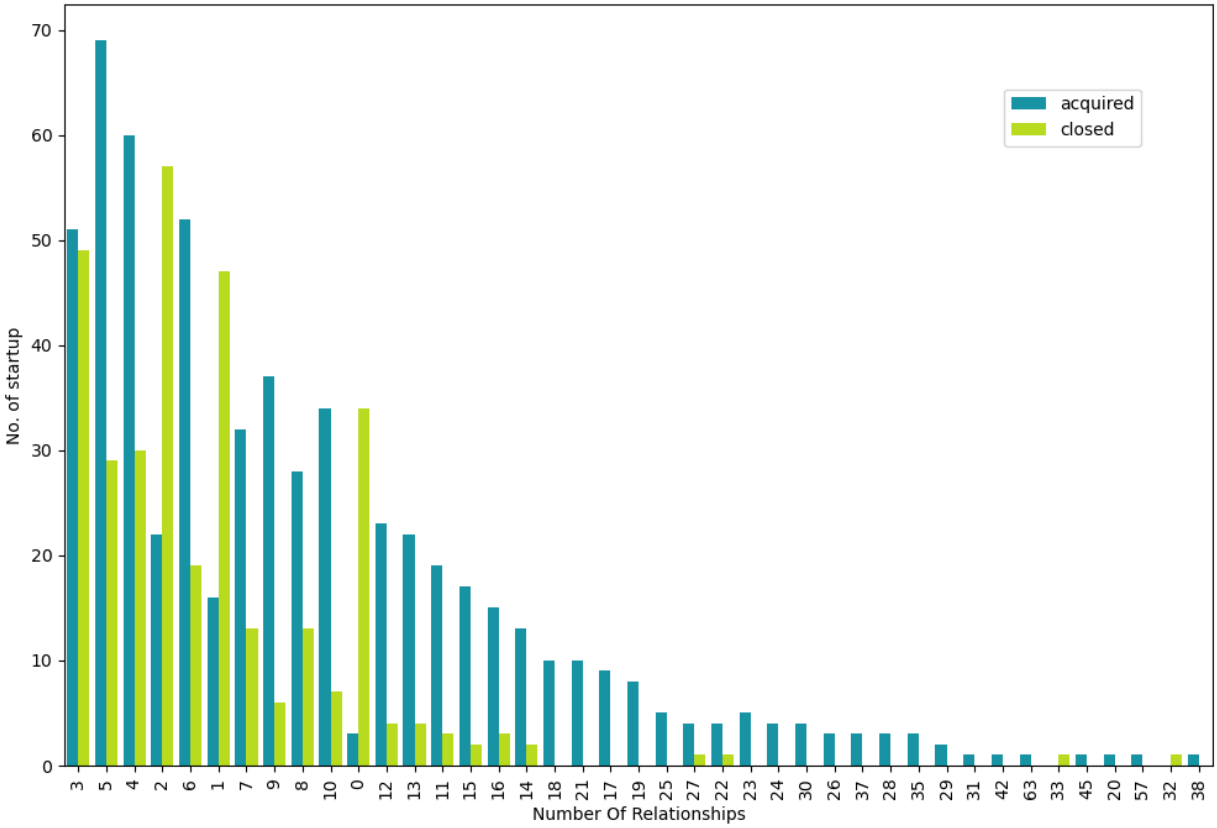
Remaining states

In [24]:
```
# Filter startups only from other states
other_startups = df[df['is_otherstate'] == True]

# Create a pivot table to sum the number of fundings for each type of sta
funding_trend = other_startups.pivot_table(index='first_funding_at',
                                           values=['is_software','is_web',
                                                   'is_mobile','is_enterpri
                                                   'is_advertising' ,'is_ga
                                                   'is_biotech' ,'is_consul
                                                   'is_othercategory'],
                                           aggfunc='sum')

# Plotting the trend
plt.figure(figsize=(12, 8))
sns.lineplot(data=funding_trend, marker='o')

# Set legend with specified labels
plt.legend(title='Startup Types')

plt.title('Funding Trend for Other States Startups Over the Years by Type
plt.xlabel('Year')
plt.ylabel('Total Fundings')
plt.grid(True)
plt.show()
```



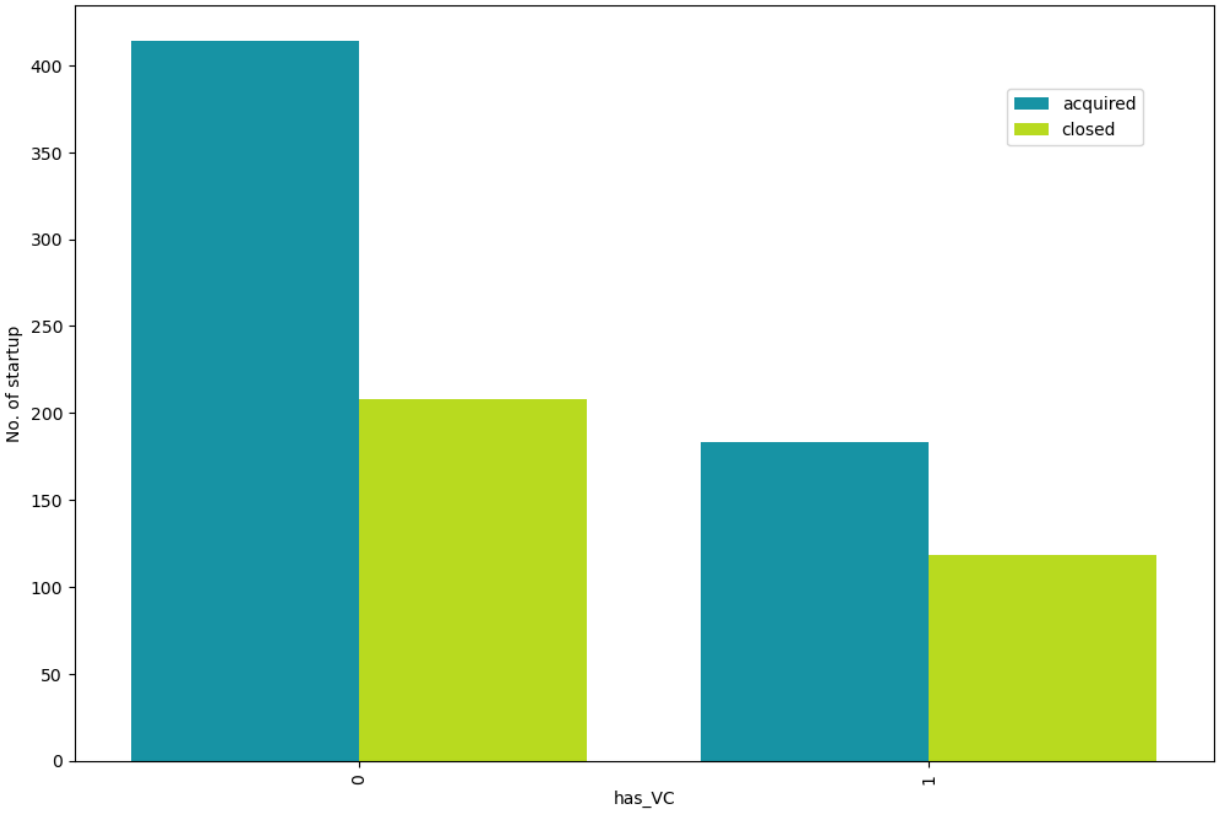
Numbers of relationship VS acquired/closed

In [29]:
```
fig, ax = plt.subplots(figsize=(12,8))

plot_2 = sns.countplot(x="relationships", hue="status", data=df, palette=
             order=df.relationships.value_counts().index)

plot_2 = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plot_2 = ax.set(xlabel="Number Of Relationships", ylabel="No. of startup"
plt.legend(bbox_to_anchor=(0.945, 0.90))
plt.show()
```



How many startups have Venture capital, acquired/closed

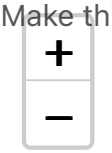
In [25]:
```
fig, ax = plt.subplots(figsize=(12,8))

plot_2 = sns.countplot(x="has_VC", hue="status", data=df,
                       palette="nipy_spectral",
             order=df.has_VC.value_counts().index)

plot_2 = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
plot_2 = ax.set(xlabel="has_VC", ylabel="No. of startup")
plt.legend(bbox_to_anchor=(0.945, 0.90))
plt.show()
```



Where are the startups located

In [30]:
```
latitude_initial = 39.8283
longitude_initial = -50.0000

map = folium.Map(location = [latitude_initial, longitude_initial],
                 zoom_start = 3, tiles = 'cartodbpositron')


for index, row in df.iterrows():
    folium.Marker([row['latitude'], row['longitude']],
                  popup = row['state_code']).add_to(map)

map
```

Out[30]: Make this Notebook Trusted to load map: File -> Trust Notebook



3. Data preprocessing

In [31]:
```
# Check for missing values
missing_values = df.isnull().sum()
missing_values
```

Out[31]:

Unnamed: 0	0
state_code	0
latitude	0
longitude	0
zip_code	0
id	0
city	0
Unnamed: 6	493
name	0
labels	0
founded_at	0
closed_at	588
first_funding_at	0
last_funding_at	0
age_first_funding_year	0
age_last_funding_year	0
age_first_milestone_year	152
age_last_milestone_year	152
relationships	0
funding_rounds	0
funding_total_usd	0
milestones	0
state_code.1	1
is_CA	0
is_NY	0
is_MA	0
is_TX	0
is_otherstate	0
category_code	0
is_software	0
is_web	0
is_mobile	0
is_enterprise	0
is_advertising	0
is_gamesvideo	0
is_ecommerce	0
is_biotech	0
is_consulting	0
is_othercategory	0
object_id	0
has_VC	0
has_angel	0
has_roundA	0
has_roundB	0
has_roundC	0
has_roundD	0
avg_participants	0
is_top500	0
status	0

dtype: int64

Delete variables

In [32]:
```
df = df.drop(['Unnamed: 0','Unnamed: 6','id','name','object_id',
              'closed_at','state_code.1', 'labels','category_code' ],
             axis = 1)
```

Fill missing values of milestone with the average

In [33]:
```
mean_value1=df['age_first_milestone_year'].mean()
mean_value2=df['age_last_milestone_year'].mean()
df["age_first_milestone_year"].fillna(value=mean_value1,inplace=True)
df["age_last_milestone_year"].fillna(value=mean_value2,inplace=True)
```

Check that everything is in place

In [34]:
```
df.head()
```

Out[34]:

	state_code	latitude	longitude	zip_code	city	founded_at	first_funding_a
0	CA	42.358880	-71.056820	92101	San Diego	2007	200
1	CA	37.238916	-121.973718	95032	Los Gatos	2000	200
2	CA	32.901049	-117.192656	92121	San Diego	2009	201
3	CA	37.320309	-122.050040	95014	Cupertino	2002	200
4	CA	37.779281	-122.419236	94105	San Francisco	2010	201

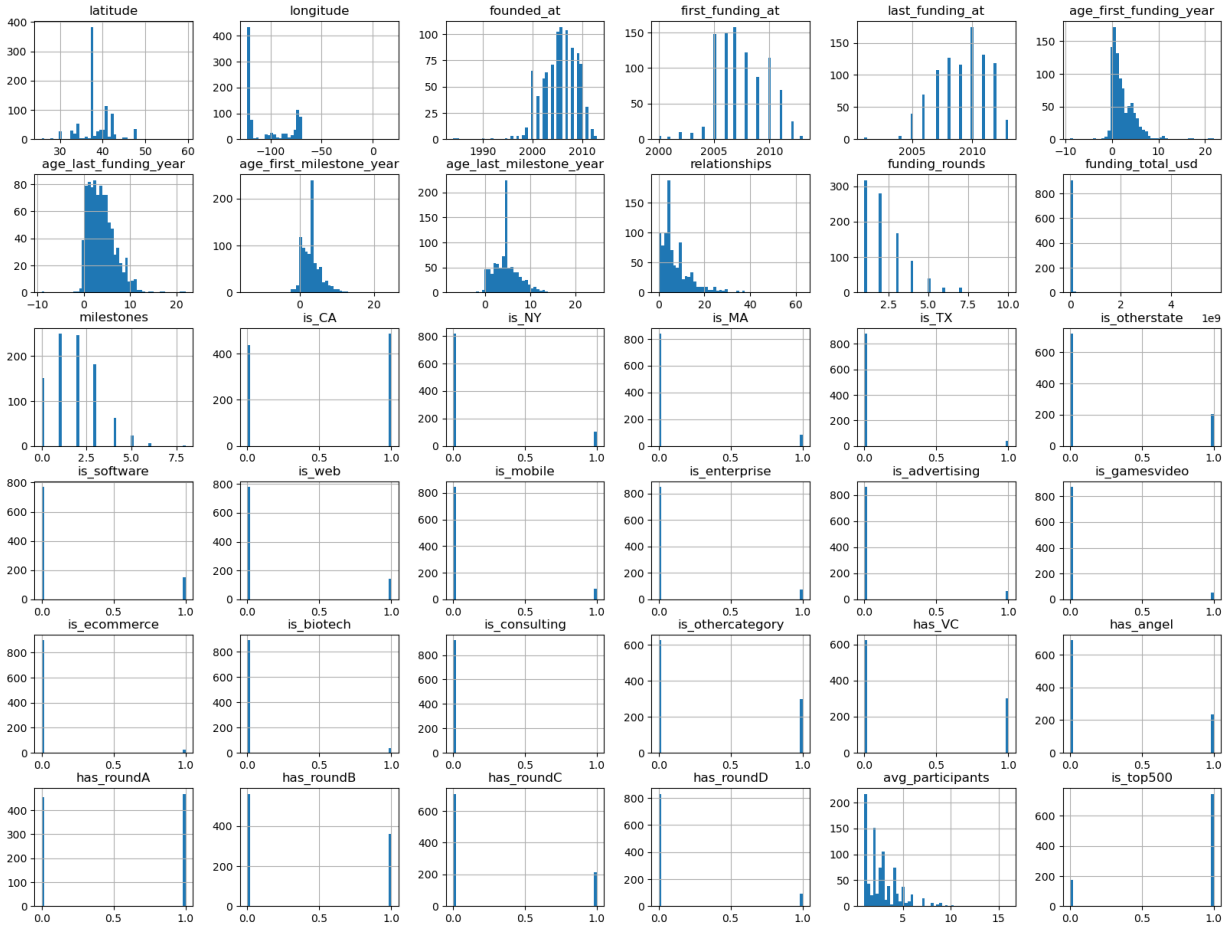
5 rows × 40 columns

In [35]:
```
n_missing_values = df.isnull().sum()
n_missing_values
```

Out[35]:
```
state_code                  0
latitude                    0
longitude                   0
zip_code                    0
city                        0
founded_at                  0
first_funding_at            0
last_funding_at             0
age_first_funding_year      0
age_last_funding_year       0
age_first_milestone_year    0
age_last_milestone_year     0
relationships               0
funding_rounds              0
funding_total_usd           0
milestones                  0
is_CA                       0
is_NY                       0
is_MA                       0
is_TX                       0
is_otherstate               0
is_software                 0
is_web                      0
is_mobile                   0
is_enterprise               0
is_advertising              0
is_gamesvideo               0
is_ecommerce                0
is_biotech                  0
is_consulting               0
is_othercategory            0
has_VC                      0
has_angel                   0
has_roundA                  0
has_roundB                  0
has_roundC                  0
has_roundD                  0
avg_participants            0
is_top500                   0
status                      0
dtype: int64
```

Look at the distribution of our variables

In [36]:
```
#Check for normal distribution
df.hist(bins=50, figsize=(20, 15))
plt.show()
```



In [37]:
```
df.describe()
```

Out[37]:

	latitude	longitude	founded_at	first_funding_at	last_funding_at	age_firs
count	923.000000	923.000000	923.000000	923.000000	923.000000	
mean	38.517442	-103.539212	2005.496208	2007.475623	2009.161430	
std	3.741497	22.394167	3.528738	2.293583	2.175327	
min	25.752358	-122.756956	1984.000000	2000.000000	2001.000000	
25%	37.388869	-122.198732	2003.000000	2006.000000	2008.000000	
50%	37.779281	-118.374037	2006.000000	2007.000000	2009.000000	
75%	40.730646	-77.214731	2008.000000	2009.000000	2011.000000	
max	59.335232	18.057121	2013.000000	2013.000000	2013.000000	

8 rows × 36 columns

In [38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 923 entries, 0 to 922
Data columns (total 40 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   state_code                923 non-null    object 
 1   latitude                  923 non-null    float64
 2   longitude                 923 non-null    float64
 3   zip_code                  923 non-null    object 
 4   city                      923 non-null    object 
 5   founded_at                923 non-null    int64  
 6   first_funding_at          923 non-null    int64  
 7   last_funding_at           923 non-null    int64  
 8   age_first_funding_year    923 non-null    float64
 9   age_last_funding_year     923 non-null    float64
 10  age_first_milestone_year  923 non-null    float64
 11  age_last_milestone_year   923 non-null    float64
 12  relationships             923 non-null    int64  
 13  funding_rounds            923 non-null    int64  
 14  funding_total_usd         923 non-null    int64  
 15  milestones                923 non-null    int64  
 16  is_CA                     923 non-null    int64  
 17  is_NY                     923 non-null    int64  
 18  is_MA                     923 non-null    int64  
 19  is_TX                     923 non-null    int64  
 20  is_otherstate             923 non-null    int64  
 21  is_software               923 non-null    int64  
 22  is_web                    923 non-null    int64  
 23  is_mobile                 923 non-null    int64  
 24  is_enterprise             923 non-null    int64  
 25  is_advertising            923 non-null    int64  
 26  is_gamesvideo             923 non-null    int64  
 27  is_ecommerce              923 non-null    int64  
 28  is_biotech                923 non-null    int64  
 29  is_consulting             923 non-null    int64  
 30  is_othercategory          923 non-null    int64  
 31  has_VC                    923 non-null    int64  
 32  has_angel                 923 non-null    int64  
 33  has_roundA                923 non-null    int64  
 34  has_roundB                923 non-null    int64  
 35  has_roundC                923 non-null    int64  
 36  has_roundD                923 non-null    int64  
 37  avg_participants          923 non-null    float64
 38  is_top500                 923 non-null    int64  
 39  status                    923 non-null    object 
dtypes: float64(7), int64(29), object(4)
memory usage: 288.6+ KB
```

In [39]:
```
#Create a new data frame without first 4 columns
# we select only the floats
df= df.iloc[:,5:]
df
```

Out[39]:

	founded_at	first_funding_at	last_funding_at	age_first_funding_year	age_last_fun
0	2007	2009	2010	2.2493	
1	2000	2005	2009	5.1260	
2	2009	2010	2010	1.0329	
3	2002	2005	2007	3.1315	
4	2010	2010	2012	0.0000	
...	...	...	...	...	
918	2009	2009	2009	0.5178	
919	1998	2005	2007	7.2521	
920	1999	2007	2007	8.4959	
921	2009	2009	2011	0.7589	
922	2003	2006	2006	3.1205	

923 rows × 35 columns

In [40]:
```
#Create dummies on status
dummies= pd.get_dummies(df['status'],drop_first=True)
dummies
```

Out[40]:

	closed
0	0
1	0
2	0
3	0
4	1
...	...
918	0
919	1
920	1
921	0
922	0

923 rows × 1 columns

Closed: 1 Opened: 0

In [41]:
```
#Add the dummies to the dataframe
new_df= pd.concat([dummies,df], axis=1)
new_df.head()
```

Out[41]:

	closed	founded_at	first_funding_at	last_funding_at	age_first_funding_year	age_las
0	0	2007	2009	2010	2.2493	
1	0	2000	2005	2009	5.1260	
2	0	2009	2010	2010	1.0329	
3	0	2002	2005	2007	3.1315	
4	1	2010	2010	2012	0.0000	

5 rows × 36 columns

In [42]:
```
#We drop 'status'
new_df.drop('status', axis=1, inplace=True)
new_df.head()
```

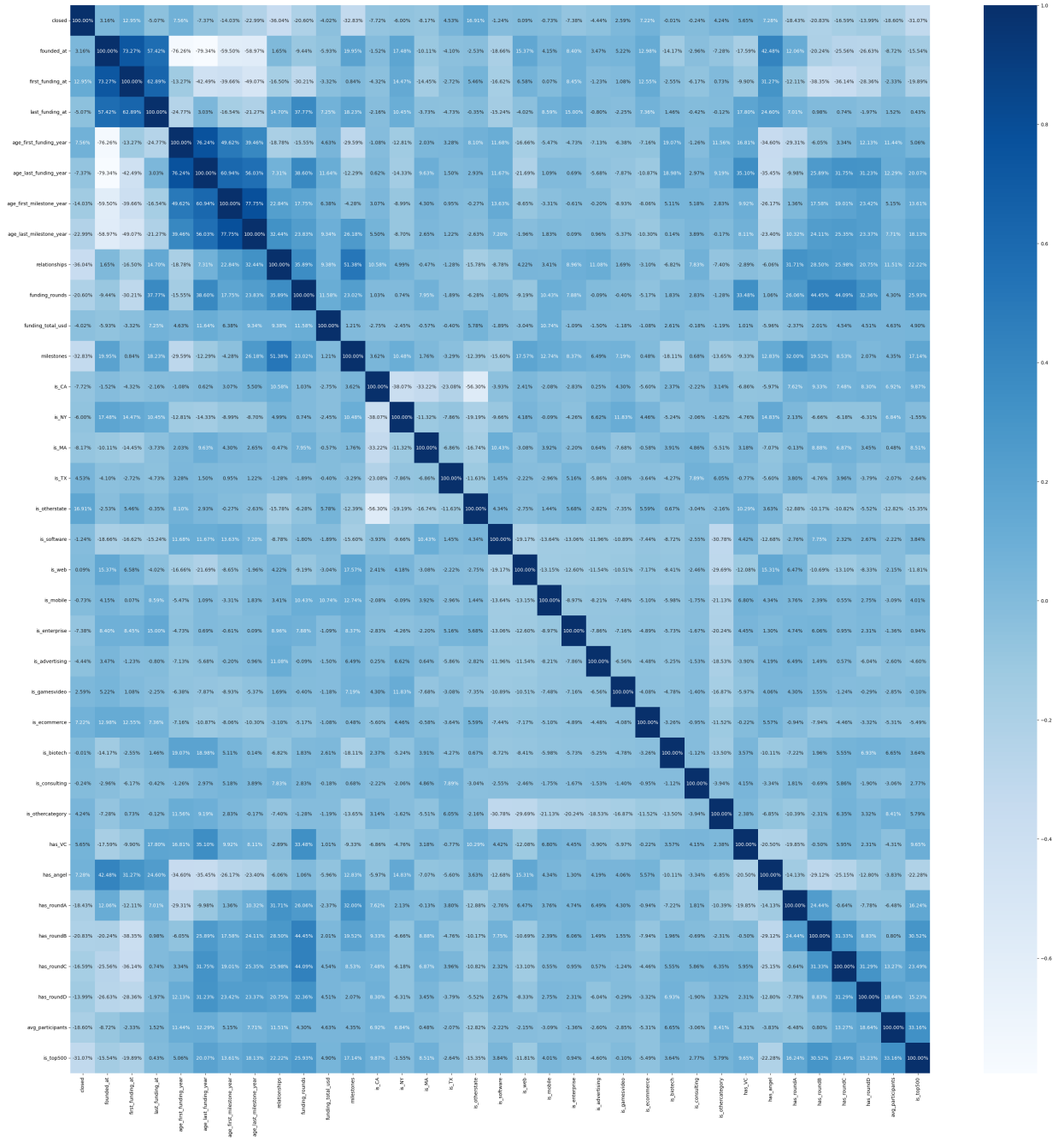
Out[42]:

	closed	founded_at	first_funding_at	last_funding_at	age_first_funding_year	age_las
0	0	2007	2009	2010	2.2493	
1	0	2000	2005	2009	5.1260	
2	0	2009	2010	2010	1.0329	
3	0	2002	2005	2007	3.1315	
4	1	2010	2010	2012	0.0000	

5 rows × 35 columns

In [43]:
```
corr_matrix= new_df.corr()
plt.figure(figsize=(40,40))
heatmap(corr_matrix, cmap='Blues', annot= True, fmt= '.2%')
```

Out[43]:
```
<Axes: >
```



In [44]:
```
# We check the correlation of all x features with the independent variabl
abs(new_df.corr()['closed'])
```

Out[44]:
```
closed                      1.000000
founded_at                  0.031645
first_funding_at            0.129485
last_funding_at             0.050697
age_first_funding_year      0.075637
age_last_funding_year       0.073731
age_first_milestone_year    0.140320
age_last_milestone_year     0.229893
relationships               0.360434
funding_rounds              0.206049
funding_total_usd           0.040176
milestones                  0.328260
is_CA                       0.077217
is_NY                       0.059996
is_MA                       0.081735
is_TX                       0.045309
is_otherstate               0.169067
is_software                 0.012429
is_web                      0.000873
is_mobile                   0.007312
is_enterprise               0.073772
is_advertising              0.044355
is_gamesvideo               0.025893
is_ecommerce                0.072193
is_biotech                  0.000104
is_consulting               0.002373
is_othercategory            0.042408
has_VC                      0.056515
has_angel                   0.072840
has_roundA                  0.184307
has_roundB                  0.208257
has_roundC                  0.165902
has_roundD                  0.139940
avg_participants            0.185992
is_top500                   0.310652
Name: closed, dtype: float64
```

Scaling the variables

In [45]:
```
#We split the data into X and y
y = new_df.closed
X = new_df.drop(['closed'], axis = 1)
```

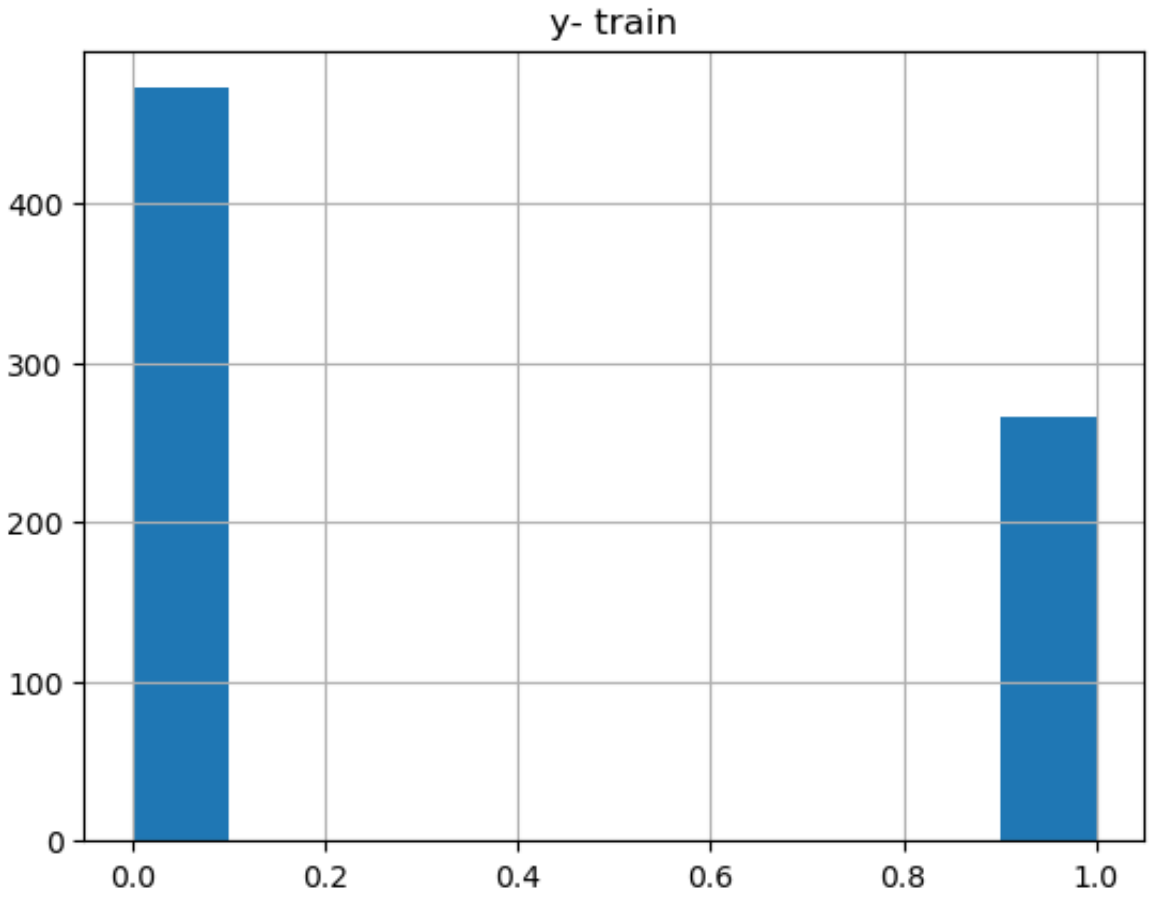
In [46]:
```
# Now we will define the scaler
# Normally, we would define one scaler for the y and one for the X (if sp
# X and y was performed before). Howeve, here the y variable does not nee
# becasue it is a bionary variable with Zeros and Ones # y_scaler = MinMa
X_scaler = MinMaxScaler()
```

In [47]:
```
# Applying the scaler to data
# y_scaled = y_scaler.fit_transform(y)
X_scaled = X_scaler.fit_transform(X)
```

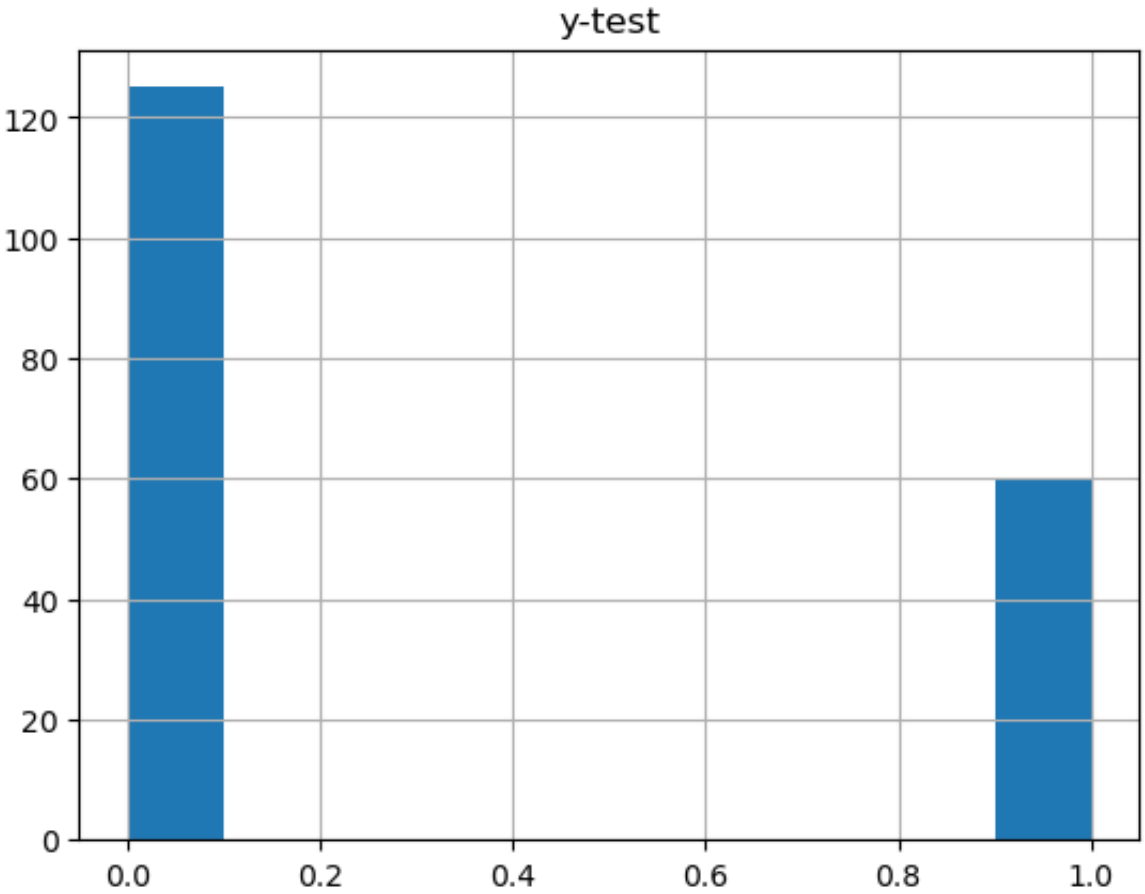
Spliting the data set into train and test sets

In [48]:
```
# We split the data into training and test set
X_train, X_test, y_train, y_test = train_test_split(X_scaled ,y,
                                                    train_size=0.8,
                                                    shuffle = True)
```

In [108…
```
y_train.hist()
plt.title("y- train")
plt.show()
```



In [109…
```
y_test.hist()
plt.title("y-test")
plt.show()
```



4.0 Logistic regresion

In [51]:
```
# Define and fit the model
log_model = LogisticRegression()
log_model.fit(X_train, y_train)
```

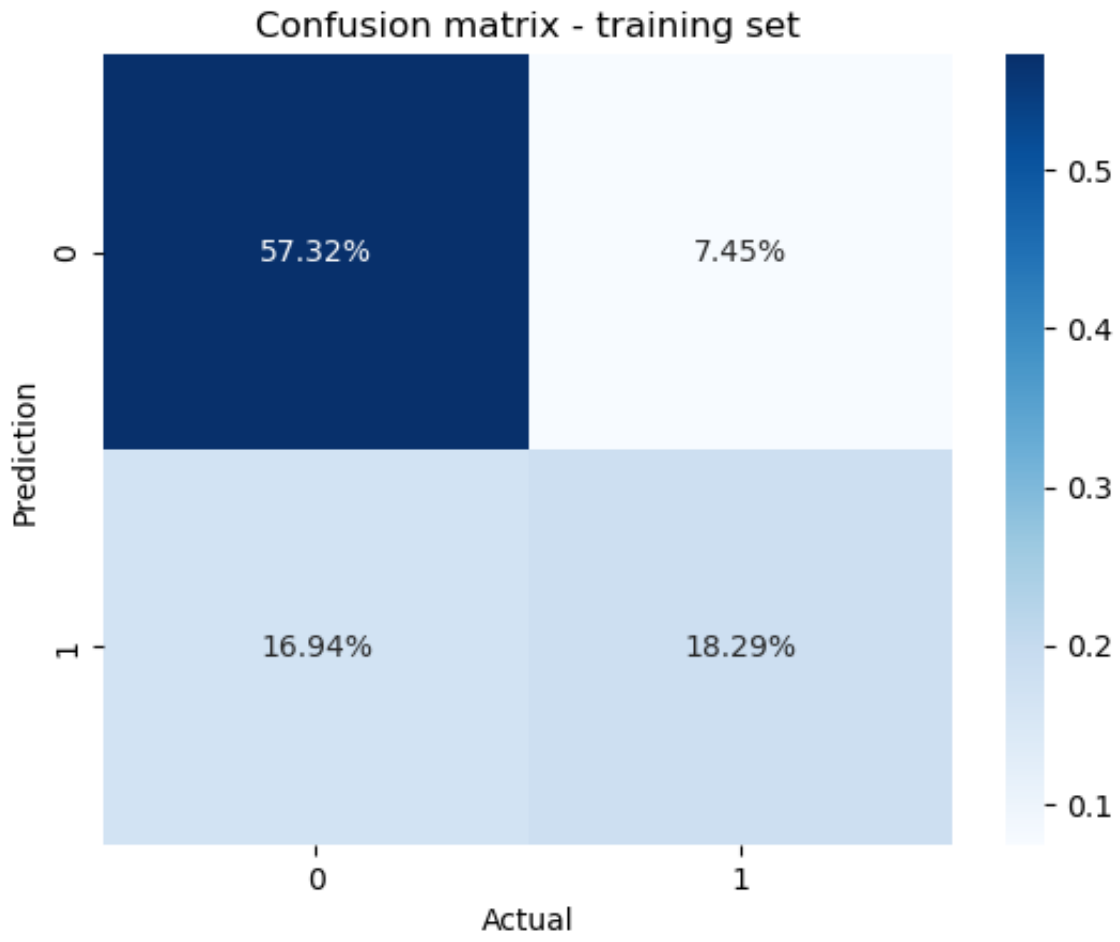
Out[51]:
```
▾ LogisticRegression
LogisticRegression()
```

In [52]:
```
#Prediction on the train set
log_fit = log_model.predict(X_train)

#Prediction on the test set
log_pred = log_model.predict(X_test)
```

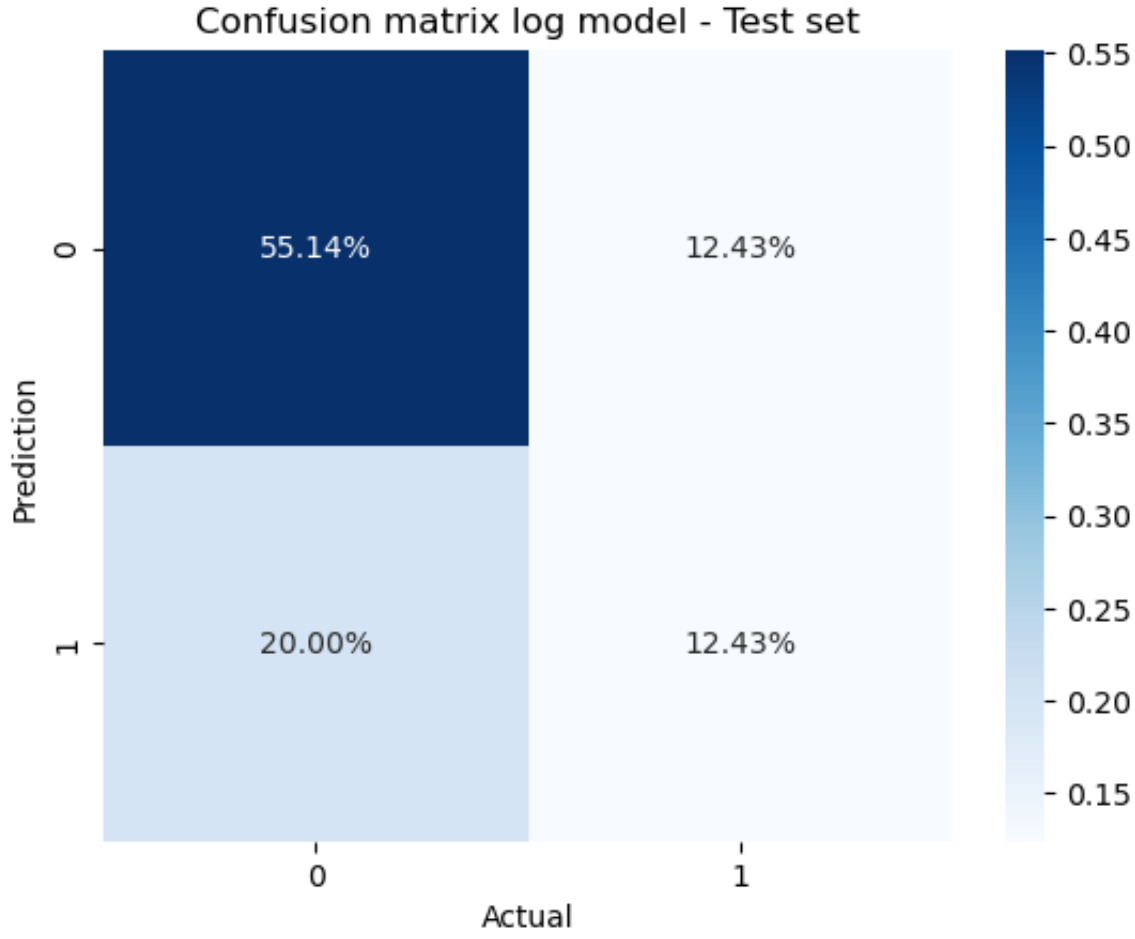
In [53]:
```
#Performance of the model
#Accurcay score for the train and test model
print('Accuracy score on log model train set',
      accuracy_score(log_fit, y_train))
print('Accuracy score on log model test set',
      accuracy_score(log_pred, y_test))
```

```
Accuracy score on log model train set 0.7560975609756098
Accuracy score on log model test set 0.8054054054054054
```

In [54]:
```
#Visualization with correlation matrix and heatmap for training set
cm_map = confusion_matrix(y_train, log_fit)
heatmap(cm_map/np.sum(cm_map),
annot=True, fmt='.2%', cmap = 'Blues')
plt.title('Confusion matrix - training set')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



In [110…
```
#Visualization with correlation matrix and heatmap for test set
cm_map = confusion_matrix(y_test, log_pred)
heatmap(cm_map/np.sum(cm_map),
annot=True, fmt='.2%', cmap = 'Blues')
plt.title('Confusion matrix log model - Test set')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



5.0 Naive bayes - Gaussiab NB

In [56]:
```
# Define and fit the model
nb_model= GaussianNB()
nb_model.fit(X_train, y_train)
```

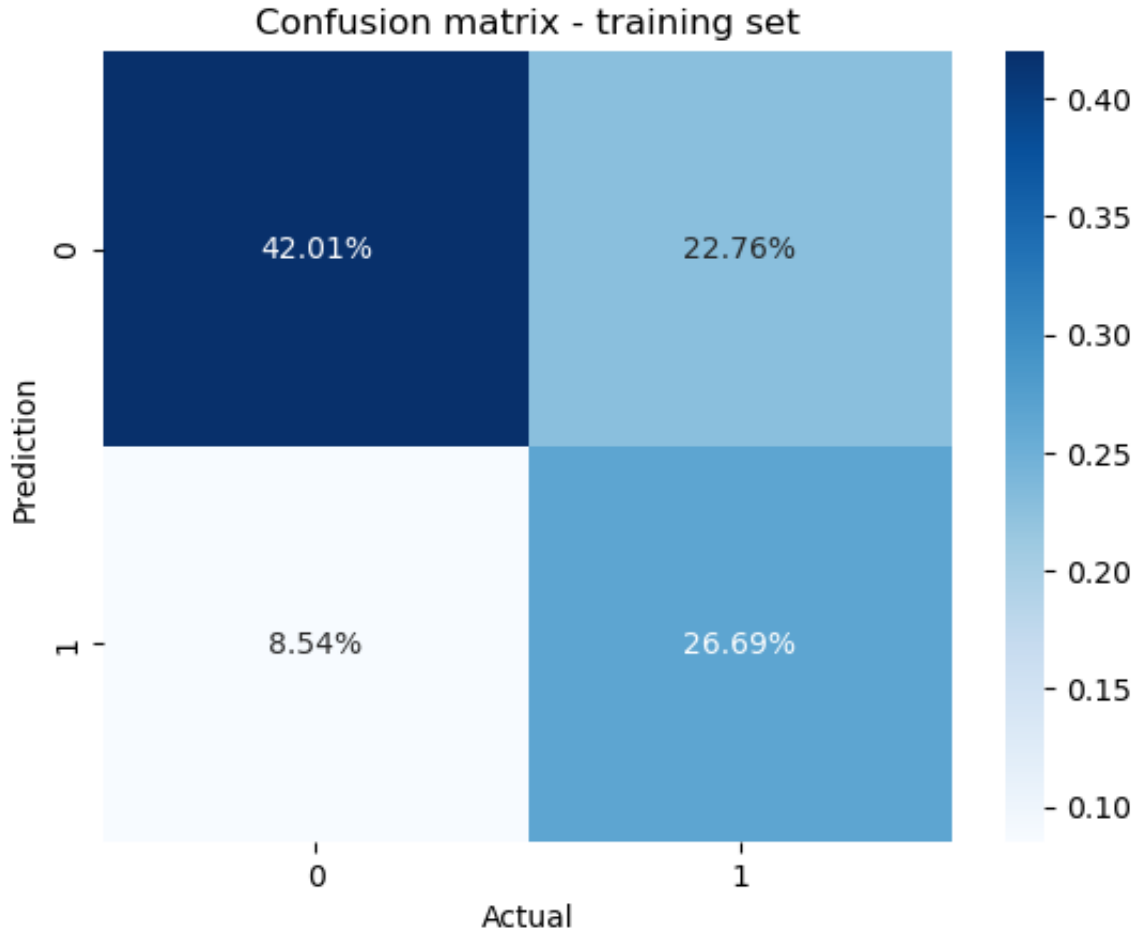
Out[56]:
```
▾ GaussianNB
GaussianNB()
```

In [57]:
```
#Prediction on the train and test set
nb_fit = nb_model.predict(X_train)
nb_pred = nb_model.predict(X_test)
```

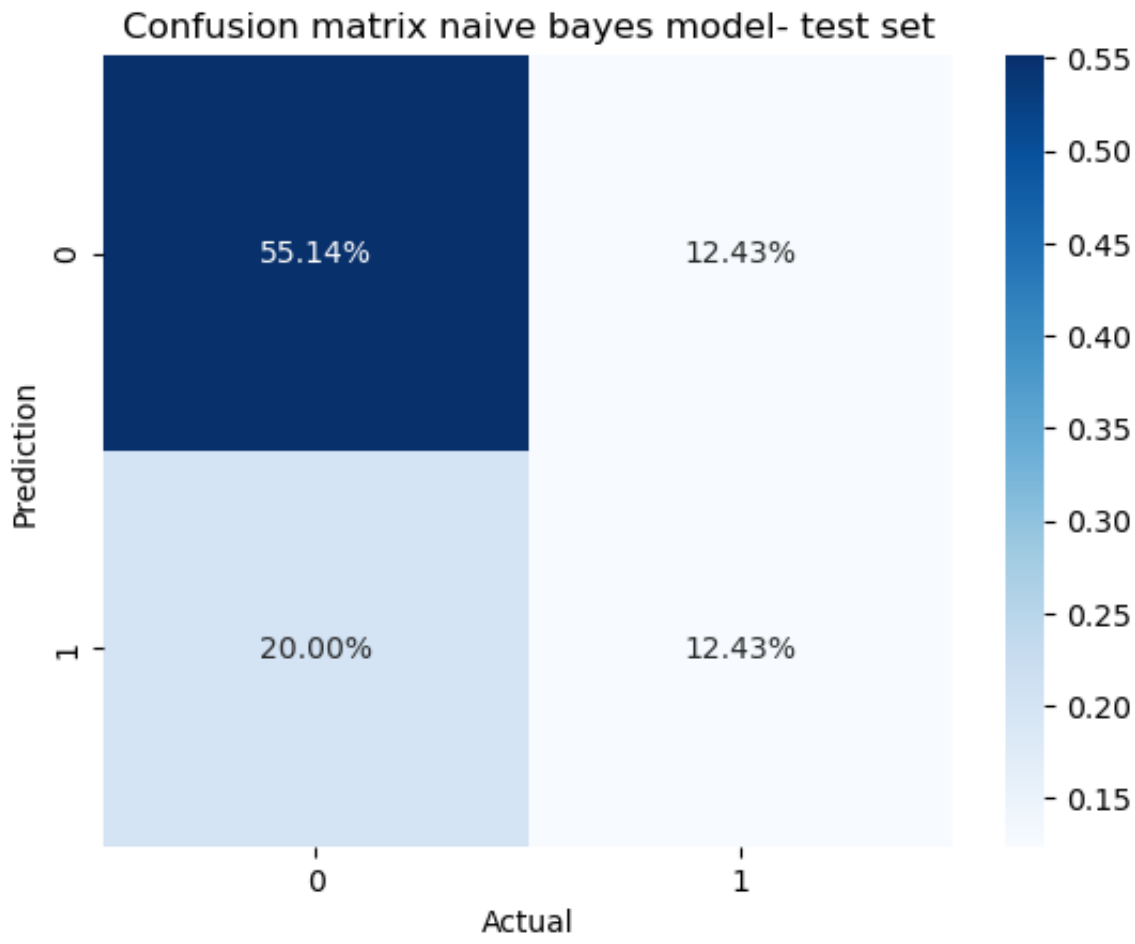
In [58]:
```
#Accurcay score for the train and test model
print('Accuracy score on Naive Bayes model train',
      accuracy_score(nb_fit, y_train))
print('Accuracy score on Naive Bayes model test',
      accuracy_score(nb_pred, y_test))
```

```
Accuracy score on Naive Bayes model train 0.6869918699186992
Accuracy score on Naive Bayes model test 0.6702702702702703
```

In [59]:
```
#Visualization with correlation matrix and heatmap for training set
cm_map = confusion_matrix(y_train, nb_fit)
heatmap(cm_map/np.sum(cm_map),
annot=True, fmt='.2%', cmap = 'Blues')
plt.title('Confusion matrix - training set')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



In [111…
```
#Visualization with correlation matrix and heatmap for test set
cm_map = confusion_matrix(y_test, log_pred)
heatmap(cm_map/np.sum(cm_map),
annot=True, fmt='.2%', cmap = 'Blues')
plt.title('Confusion matrix naive bayes model- test set')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



6.0 DTC

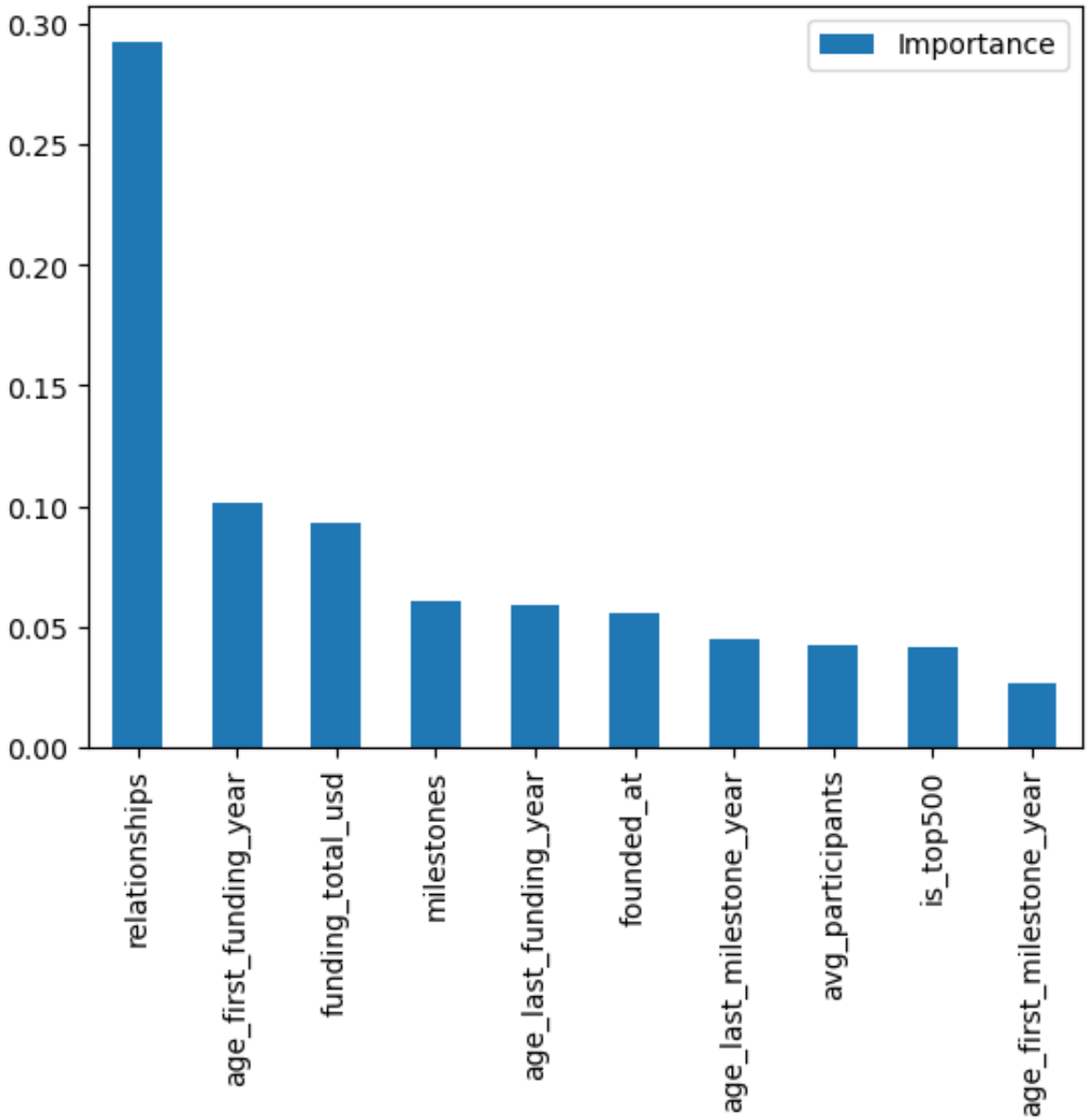
In [61]:
```
# Select the variables
y = new_df['closed']
X = new_df.drop(['closed'], axis=1)

#Scale them
X_scaler = MinMaxScaler()
X_scaled = X_scaler.fit_transform(X)

# We split the data into training and test set
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
                                                    train_size=0.8, shuff

# Define and fit the model
dtc_model = DecisionTreeClassifier().fit(X_train, y_train)

# Convert X_train to DataFrame with column names
X_train_df = pd.DataFrame(X_train, columns=X.columns)

# Create the feature_df using X_train_df columns
feature_df = pd.DataFrame(dtc_model.feature_importances_,
                          index=X_train_df.columns, columns=['Importance'
```

In [62]:
```
# Define and fit the model
dtc_model = DecisionTreeClassifier().fit(X_train, y_train)

# Convert X_train to DataFrame with column names
X_train_df = pd.DataFrame(X_train, columns=X.columns)

# Create the feature_df using X_train_df columns
feature_df = pd.DataFrame(dtc_model.feature_importances_,
                          index=X_train_df.columns,
                          columns=['Importance'])
```

In [63]:
```
feature_df.sort_values(by=['Importance'], axis=0,
                       ascending= False, inplace=True)
```

In [64]:
```
feature_df.head(10).plot(kind='bar')
```

Out[64]:
```
<Axes: >
```



Pruning the three

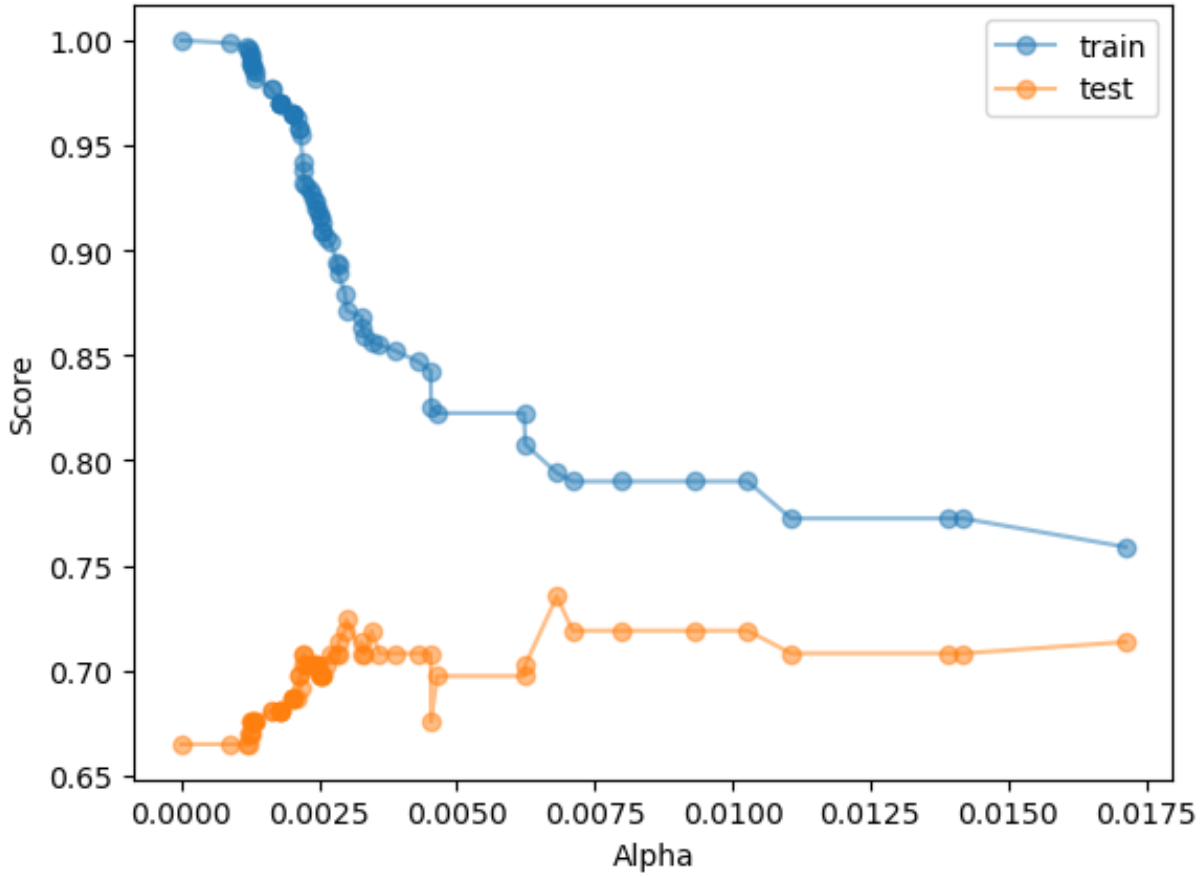
In [65]:
```
#Defining the new model
dtc2_model = DecisionTreeClassifier()
#Pruning the tree
path = dtc2_model.cost_complexity_pruning_path(X_train, y_train)
ccp_alphas = path.ccp_alphas
ccp_alphas =ccp_alphas[:-1]
```

In [66]:
```
 #Create an empty list. This list will be filled wwith all the fitted mod
model_list = []

for alpha in ccp_alphas:
    model = DecisionTreeClassifier(ccp_alpha=alpha,random_state=42)
    model.fit(X_train, y_train)
    model_list.append(model)
```

In [67]:
```
#Check the scores for train and test set.
train_scores = [model.score(X_train, y_train) for model in model_list]
test_scores = [model.score(X_test, y_test) for model in model_list]
```

In [68]:
```
#Plotting the model score dependent on the alpha values
plt.plot(ccp_alphas, train_scores, marker='o',
         alpha=.5, label='train')
plt.plot(ccp_alphas, test_scores, marker='o',
         alpha=.5, label='test')

plt.xlabel('Alpha')
plt.ylabel('Score')
plt.legend()
plt.show
```

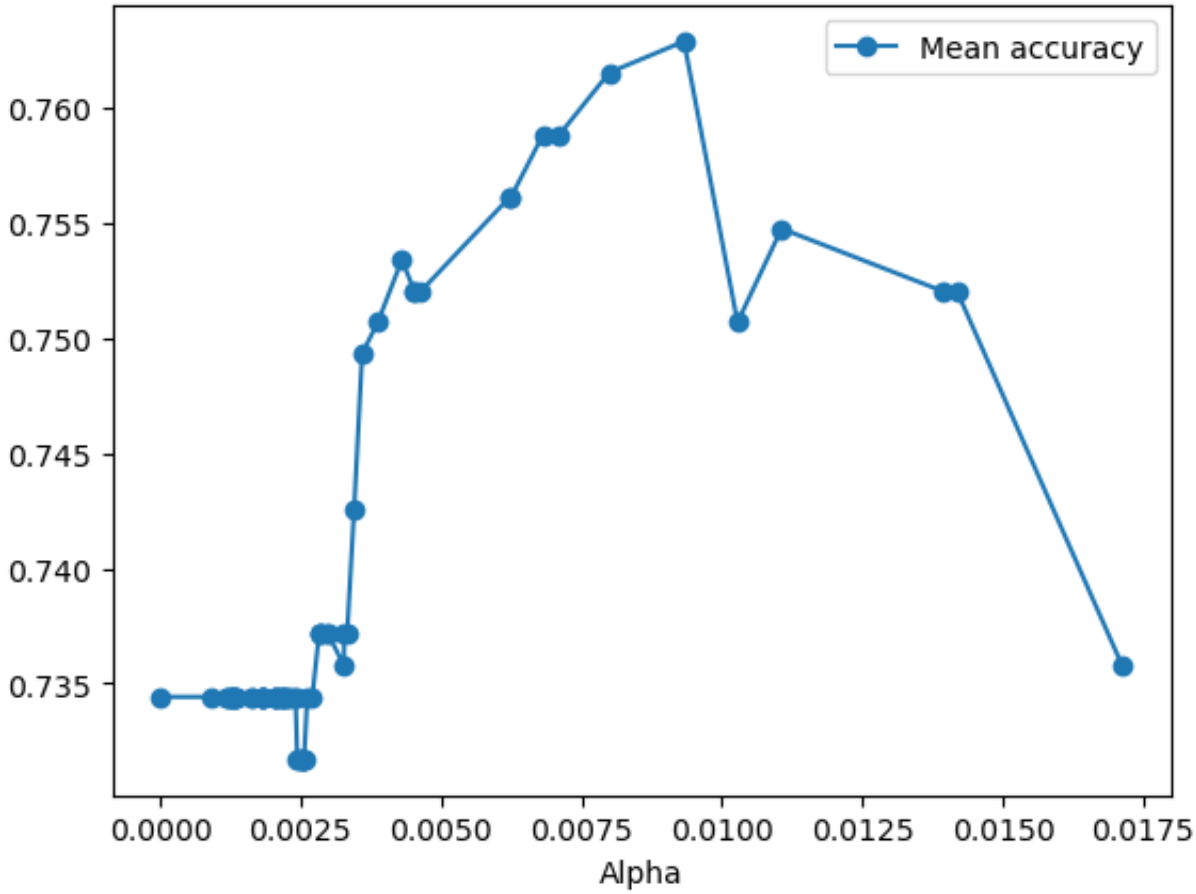
Out[68]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [69]:
```
#Create an empty list
alpha_loop_values = []
for alpha in ccp_alphas:
    model = DecisionTreeClassifier(ccp_alpha=alpha,random_state=42)
    scores= cross_val_score(model, X_train, y_train, cv=3)
    alpha_loop_values.append([alpha, np.mean(scores), np.std(scores)])
```

In [71]:
```
alpha_results= pd.DataFrame(alpha_loop_values,
                            columns=['Alpha', 'Mean accuracy', 'StdDev'])
```

In [72]:
```
#Plotting the alpha values and their corresponding accuracies
alpha_results.plot(x='Alpha', y= 'Mean accuracy', marker = 'o')
plt.show()
```



In [73]:
```
#Finding the ideal alpha
ideal_alpha=alpha_results[alpha_results['Mean accuracy']== max(alpha_resu
ideal_alpha
```

Out[73]:
```
65    0.009322
Name: Alpha, dtype: float64
```

In [74]:
```
ideal_alpha= ideal_alpha.loc[(65)]
#Be careful, this number changes all the time!
```

In [75]:
```
#Converting the ideal alpha into numpy float and than python.item()
ideal_alpha = np.float64(ideal_alpha)
ideal_alpha_float = ideal_alpha.item()
```

In [76]:
```
#To check if the converting was sucessfull
type(ideal_alpha_float)
```

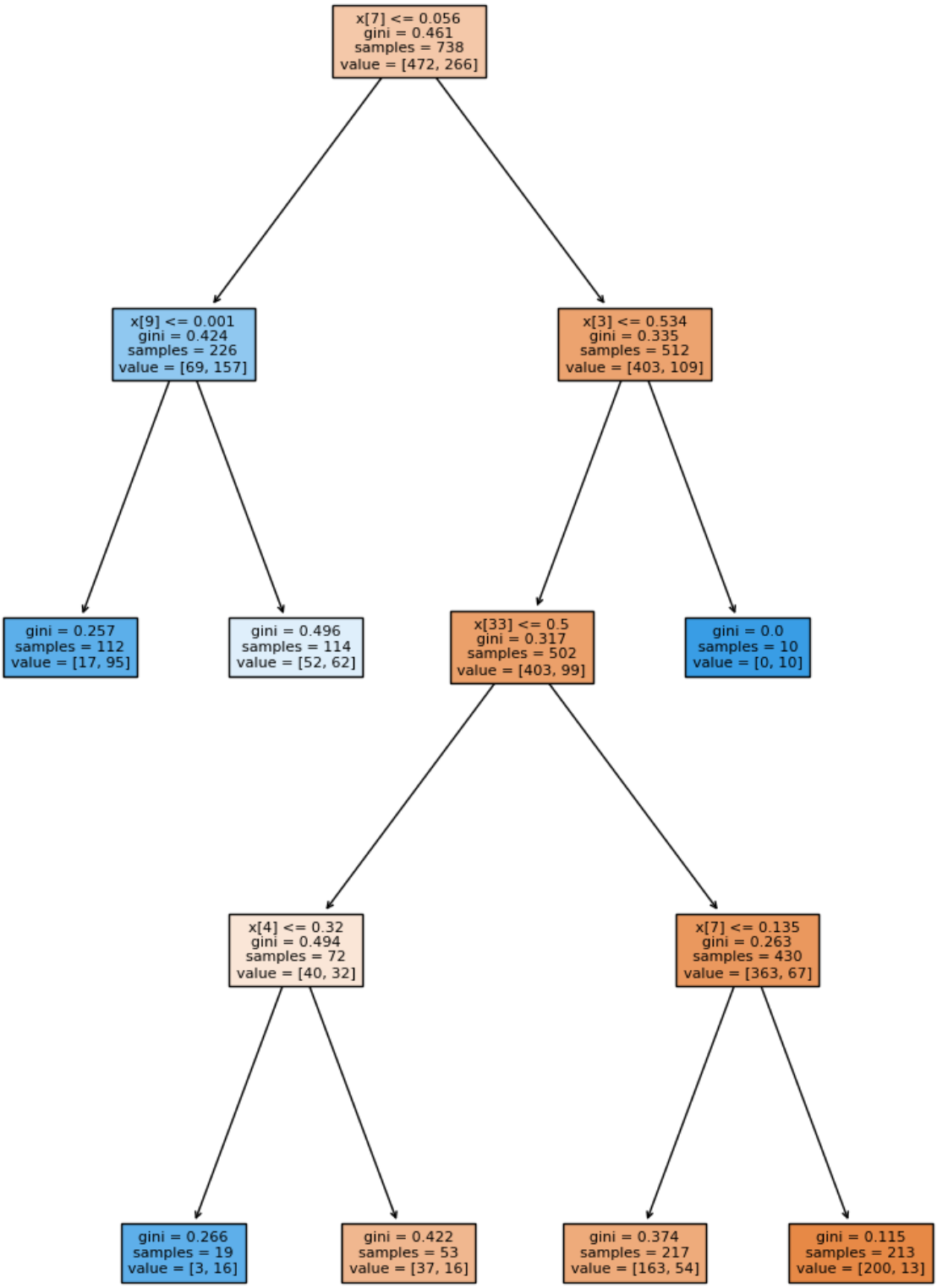
Out[76]:
```
float
```

In [77]:
```
#Creating a final model with the ideal alpha
pruned_model = DecisionTreeClassifier(ccp_alpha= ideal_alpha_float,
                                      random_state=42).fit(X_train, y_tra
```

In [78]:
```
#Predicting traing and test set
pru_fit = pruned_model.predict(X_train)
pru_pred = pruned_model.predict(X_test)
```

Plotting the three

In [79]:
```
fig = plt.figure(figsize=(10,15))
plot_tree(pruned_model, filled=True,fontsize=8) #feature_names=X_train.co


plt.show()
#I'm unfortunatelly having some probles witht the feature_names
#Therefore I decided to run the tree without the names of the features.
```

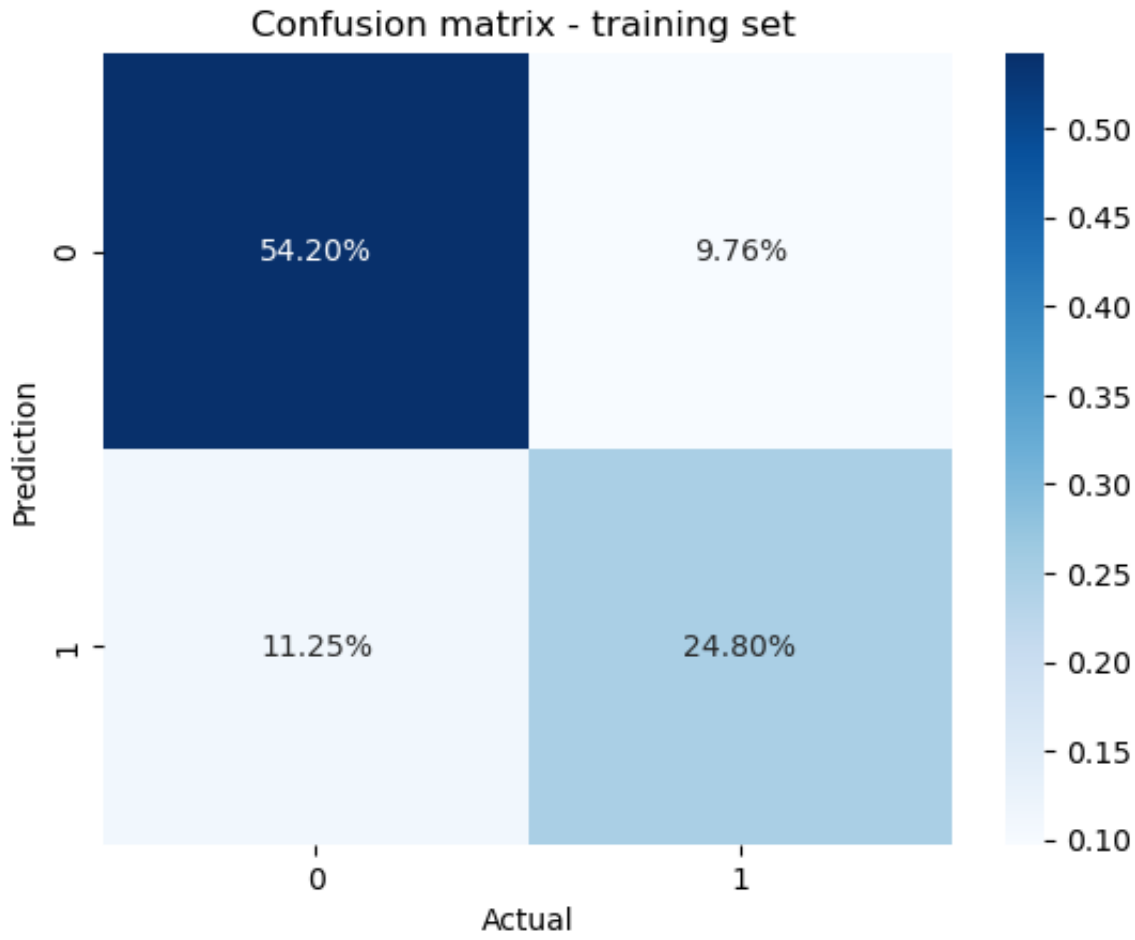


Performance of the model

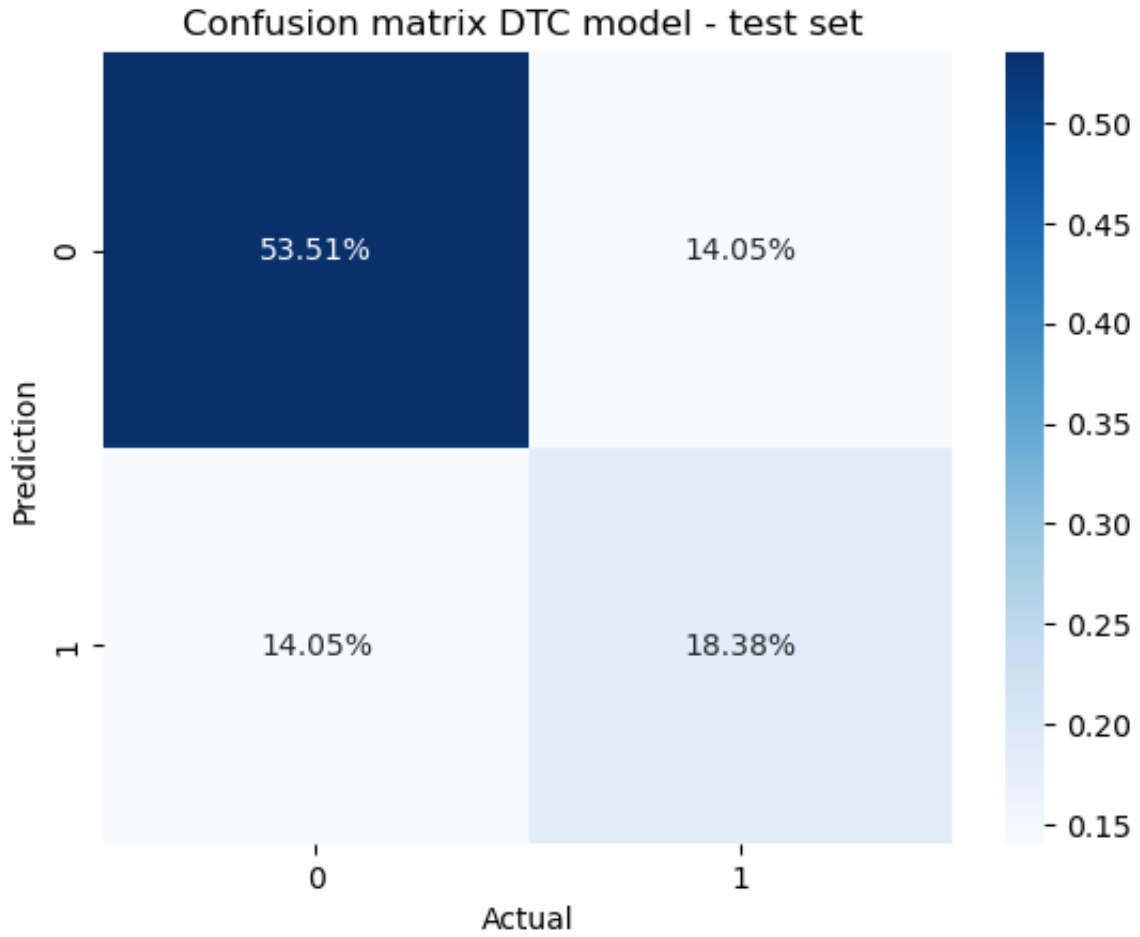
In [80]:
```
#Checking the accuracy of the model for train and test set
print('Accuracy score on DTC train',
      accuracy_score(y_train, pru_fit))
print('Accuracy score on DTC test',
      accuracy_score(y_test, pru_pred))
```

```
Accuracy score on DTC train 0.7899728997289973
Accuracy score on DTC test 0.7189189189189189
```

In [81]:
```
#Show the performence of the model on the training set:
#Visualization with heat map
cm_map = confusion_matrix(y_train, pru_fit)
heatmap(cm_map/np.sum(cm_map),
annot=True, fmt='.2%', cmap = 'Blues')
plt.title('Confusion matrix - training set ')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



In [112…
```
#Show the performence of the model on the training set:
#Visualization with heat map
cm_map = confusion_matrix(y_test, pru_pred)
heatmap(cm_map/np.sum(cm_map),
annot=True, fmt='.2%', cmap = 'Blues')
plt.title('Confusion matrix DTC model - test set ')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



7.0 ANN

In [83]:
```
n_features = X.shape[1]
n_classes = len(y.unique())
```

In [84]:
```
model = Sequential()
model.add(Dense(10, input_dim = n_features,
                activation = 'relu'))
model.add(Dense(10, activation = 'relu'))
model.add(Dense(n_classes, activation = 'softmax'))
```

In [85]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 10)                350

 dense_1 (Dense)             (None, 10)                110

 dense_2 (Dense)             (None, 2)                 22

=================================================================
Total params: 482 (1.88 KB)
Trainable params: 482 (1.88 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [86]:
```
model.compile(loss = 'sparse_categorical_crossentropy',
              optimizer = 'adam')
```

In [87]:
```
early_stop = EarlyStopping(monitor = 'loss',
                           patience = 3, verbose = 1)
```

In [88]:
```
history = model.fit(X_train, y_train, epochs = 100,
                    batch_size = 10,
                    callbacks = [early_stop])
```

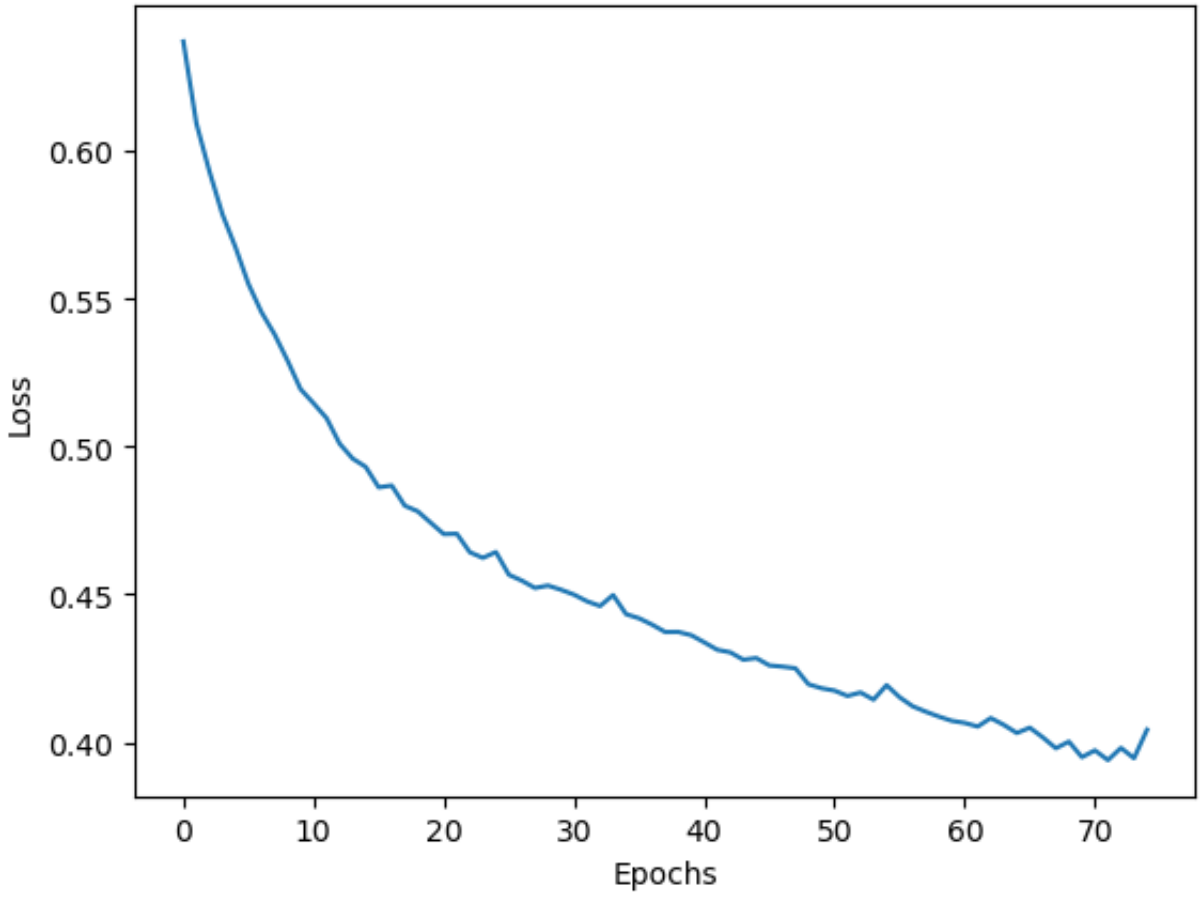
```
Epoch 1/100
74/74 [==============================] - 0s 401us/step - loss: 0.6370
Epoch 2/100
74/74 [==============================] - 0s 333us/step - loss: 0.6093
Epoch 3/100
74/74 [==============================] - 0s 329us/step - loss: 0.5931
Epoch 4/100
74/74 [==============================] - 0s 333us/step - loss: 0.5786
Epoch 5/100
74/74 [==============================] - 0s 325us/step - loss: 0.5674
Epoch 6/100
74/74 [==============================] - 0s 330us/step - loss: 0.5550
Epoch 7/100
74/74 [==============================] - 0s 330us/step - loss: 0.5454
Epoch 8/100
74/74 [==============================] - 0s 330us/step - loss: 0.5380
Epoch 9/100
74/74 [==============================] - 0s 326us/step - loss: 0.5291
Epoch 10/100
74/74 [==============================] - 0s 330us/step - loss: 0.5194
Epoch 11/100
74/74 [==============================] - 0s 339us/step - loss: 0.5146
Epoch 12/100
74/74 [==============================] - 0s 338us/step - loss: 0.5096
Epoch 13/100
74/74 [==============================] - 0s 341us/step - loss: 0.5010
Epoch 14/100
74/74 [==============================] - 0s 329us/step - loss: 0.4959
Epoch 15/100
74/74 [==============================] - 0s 329us/step - loss: 0.4931
```

```
Epoch 16/100
74/74 [==============================] - 0s 339us/step - loss: 0.4862
Epoch 17/100
74/74 [==============================] - 0s 328us/step - loss: 0.4868
Epoch 18/100
74/74 [==============================] - 0s 299us/step - loss: 0.4800
Epoch 19/100
74/74 [==============================] - 0s 312us/step - loss: 0.4781
Epoch 20/100
74/74 [==============================] - 0s 358us/step - loss: 0.4742
Epoch 21/100
74/74 [==============================] - 0s 347us/step - loss: 0.4704
Epoch 22/100
74/74 [==============================] - 0s 360us/step - loss: 0.4705
Epoch 23/100
74/74 [==============================] - 0s 355us/step - loss: 0.4642
Epoch 24/100
74/74 [==============================] - 0s 353us/step - loss: 0.4623
Epoch 25/100
74/74 [==============================] - 0s 350us/step - loss: 0.4643
Epoch 26/100
74/74 [==============================] - 0s 362us/step - loss: 0.4566
Epoch 27/100
74/74 [==============================] - 0s 357us/step - loss: 0.4546
Epoch 28/100
74/74 [==============================] - 0s 334us/step - loss: 0.4522
Epoch 29/100
74/74 [==============================] - 0s 375us/step - loss: 0.4530
Epoch 30/100
74/74 [==============================] - 0s 344us/step - loss: 0.4516
Epoch 31/100
74/74 [==============================] - 0s 353us/step - loss: 0.4499
Epoch 32/100
74/74 [==============================] - 0s 351us/step - loss: 0.4476
Epoch 33/100
74/74 [==============================] - 0s 349us/step - loss: 0.4461
Epoch 34/100
74/74 [==============================] - 0s 355us/step - loss: 0.4498
Epoch 35/100
74/74 [==============================] - 0s 337us/step - loss: 0.4433
Epoch 36/100
74/74 [==============================] - 0s 342us/step - loss: 0.4419
Epoch 37/100
74/74 [==============================] - 0s 334us/step - loss: 0.4397
Epoch 38/100
74/74 [==============================] - 0s 345us/step - loss: 0.4373
Epoch 39/100
74/74 [==============================] - 0s 319us/step - loss: 0.4373
Epoch 40/100
74/74 [==============================] - 0s 337us/step - loss: 0.4362
Epoch 41/100
74/74 [==============================] - 0s 354us/step - loss: 0.4338
Epoch 42/100
74/74 [==============================] - 0s 349us/step - loss: 0.4313
Epoch 43/100
74/74 [==============================] - 0s 353us/step - loss: 0.4304
Epoch 44/100
```

```
74/74 [==============================] - 0s 339us/step - loss: 0.4279
Epoch 45/100
74/74 [==============================] - 0s 349us/step - loss: 0.4285
Epoch 46/100
74/74 [==============================] - 0s 340us/step - loss: 0.4260
Epoch 47/100
74/74 [==============================] - 0s 357us/step - loss: 0.4256
Epoch 48/100
74/74 [==============================] - 0s 351us/step - loss: 0.4250
Epoch 49/100
74/74 [==============================] - 0s 350us/step - loss: 0.4196
Epoch 50/100
74/74 [==============================] - 0s 347us/step - loss: 0.4183
Epoch 51/100
74/74 [==============================] - 0s 353us/step - loss: 0.4175
Epoch 52/100
74/74 [==============================] - 0s 331us/step - loss: 0.4156
Epoch 53/100
74/74 [==============================] - 0s 337us/step - loss: 0.4168
Epoch 54/100
74/74 [==============================] - 0s 324us/step - loss: 0.4144
Epoch 55/100
74/74 [==============================] - 0s 334us/step - loss: 0.4193
Epoch 56/100
74/74 [==============================] - 0s 348us/step - loss: 0.4152
Epoch 57/100
74/74 [==============================] - 0s 311us/step - loss: 0.4122
Epoch 58/100
74/74 [==============================] - 0s 302us/step - loss: 0.4103
Epoch 59/100
74/74 [==============================] - 0s 334us/step - loss: 0.4087
Epoch 60/100
74/74 [==============================] - 0s 374us/step - loss: 0.4072
Epoch 61/100
74/74 [==============================] - 0s 328us/step - loss: 0.4065
Epoch 62/100
74/74 [==============================] - 0s 356us/step - loss: 0.4053
Epoch 63/100
74/74 [==============================] - 0s 361us/step - loss: 0.4082
Epoch 64/100
74/74 [==============================] - 0s 334us/step - loss: 0.4059
Epoch 65/100
74/74 [==============================] - 0s 354us/step - loss: 0.4031
Epoch 66/100
74/74 [==============================] - 0s 355us/step - loss: 0.4049
Epoch 67/100
74/74 [==============================] - 0s 335us/step - loss: 0.4016
Epoch 68/100
74/74 [==============================] - 0s 371us/step - loss: 0.3980
Epoch 69/100
74/74 [==============================] - 0s 349us/step - loss: 0.4002
Epoch 70/100
74/74 [==============================] - 0s 327us/step - loss: 0.3949
Epoch 71/100
74/74 [==============================] - 0s 327us/step - loss: 0.3972
Epoch 72/100
74/74 [==============================] - 0s 296us/step - loss: 0.3939
```

```
Epoch 73/100
74/74 [==============================] - 0s 315us/step - loss: 0.3981
Epoch 74/100
74/74 [==============================] - 0s 328us/step - loss: 0.3946
Epoch 75/100
74/74 [==============================] - 0s 311us/step - loss: 0.4042
Epoch 75: early stopping
```

In [89]:
```
#History
plt.plot(history.history['loss'])
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.show()
```



In [90]:
```
print(model.evaluate(X_train, y_train, verbose = 0))
print(model.evaluate(X_test, y_test, verbose = 0))
```

```
0.3834860026836395
0.7250039577484131
```

In [91]:
```
y_pred_train = model.predict(X_train)
y_pred_test = model.predict(X_test)
```

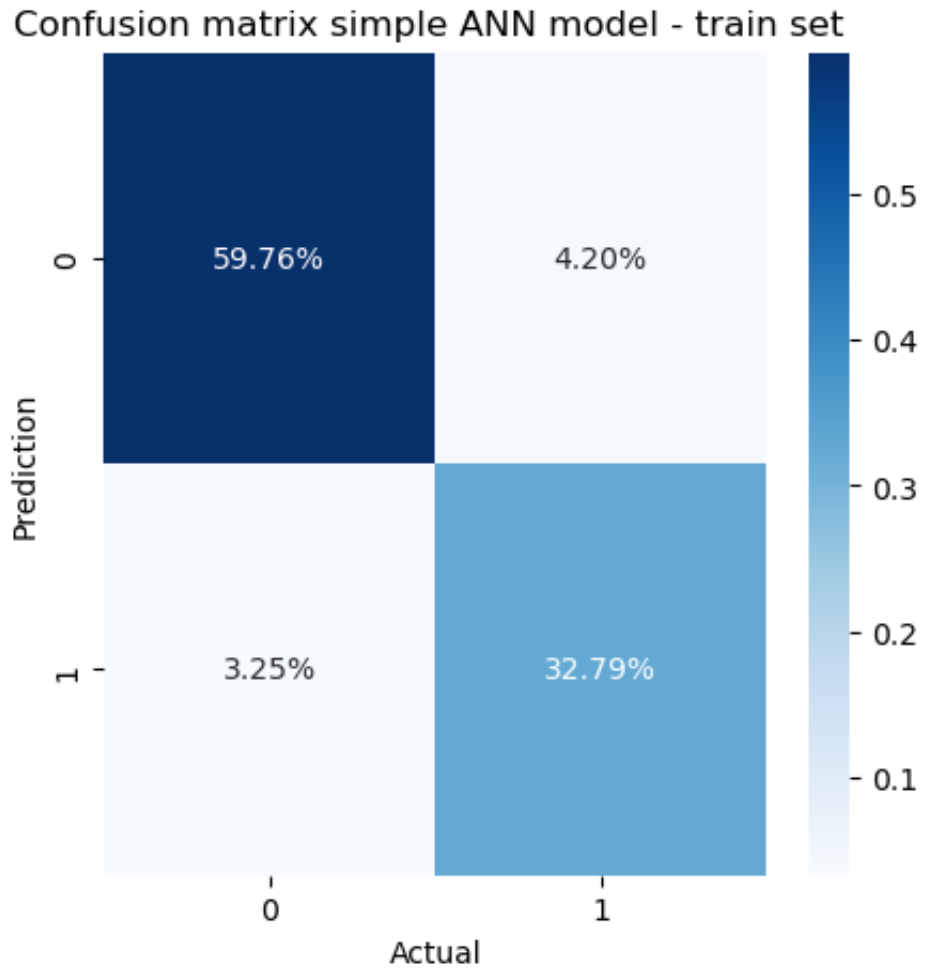
```
24/24 [==============================] - 0s 306us/step
6/6 [==============================] - 0s 462us/step
```

In [92]:
```
y_pred_train = y_pred_train[:,1].round()
y_pred_test = y_pred_test[:,1].round()
```

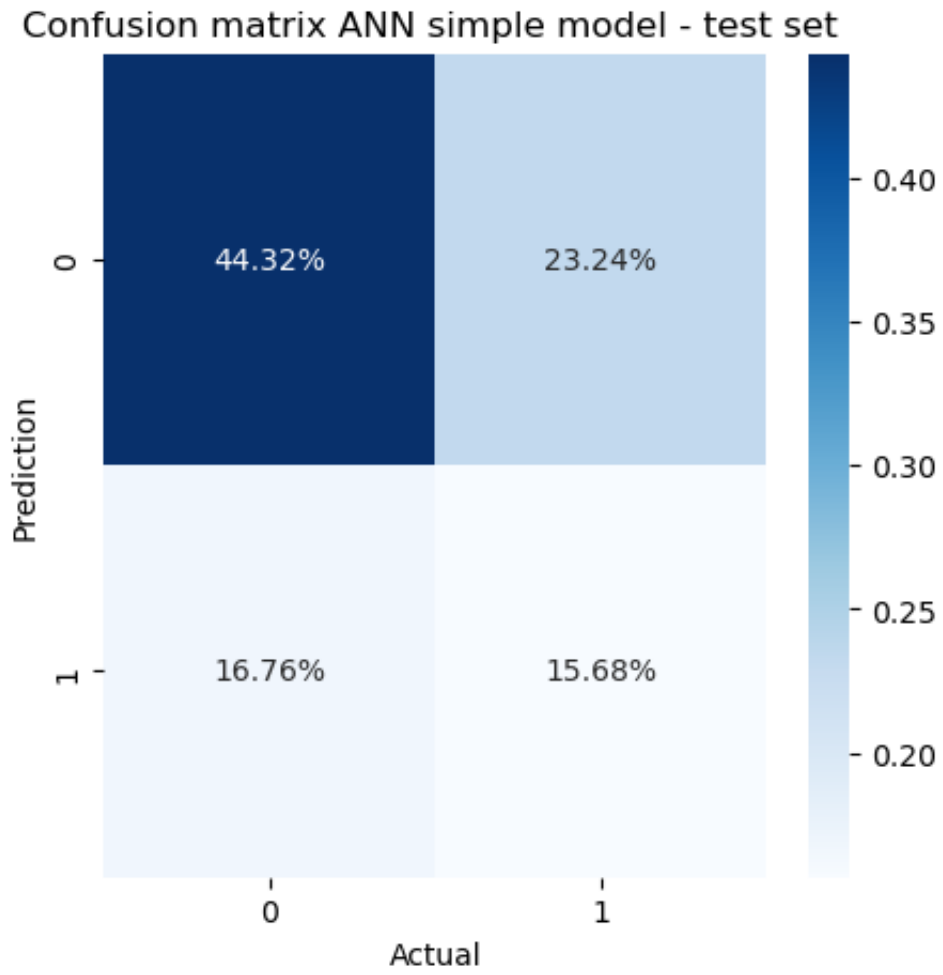
In [93]:
```
print('Accuracy score on Train is:',
      accuracy_score(y_train, y_pred_train))
print('Accuracy score on Test is:',
      accuracy_score(y_test, y_pred_test))
```

```
Accuracy score on Train is: 0.8319783197831978
Accuracy score on Test is: 0.7189189189189189
```

In [114…
```
# Heatmap on Training set
cm = confusion_matrix(y_train, y_pred_train)
cm_perc = cm/len(y_train)
plt.figure(figsize = (5,5))
heatmap(cm_perc, annot = True, fmt = '.2%', cmap = 'Blues' )
plt.title('Confusion matrix simple ANN model - train set ')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



In [116…
```
# Heatmap on Test set
cm = confusion_matrix(y_test, y_pred_test)
cm_perc = cm/len(y_test)
plt.figure(figsize = (5,5))
heatmap(cm_perc, annot = True, fmt = '.2%', cmap = 'Blues' )
plt.title('Confusion matrix ANN simple model - test set ')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.show()
```



Tring again with more nodds

In [96]:
```
n_features = X.shape[1]
n_classes = len(y.unique())
```

In [97]:
```
model = Sequential()
model.add(Dense(30, input_dim = n_features, activation = 'relu'))
model.add(Dense(20, activation = 'relu'))
model.add(Dense(20, activation = 'relu'))
model.add(Dense(20, activation = 'relu'))
model.add(Dense(n_classes, activation = 'softmax'))
```

In [98]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 30)                1050

 dense_4 (Dense)             (None, 20)                620

 dense_5 (Dense)             (None, 20)                420

 dense_6 (Dense)             (None, 20)                420

 dense_7 (Dense)             (None, 2)                 42

=================================================================
Total params: 2552 (9.97 KB)
Trainable params: 2552 (9.97 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [99]:
```
model.compile(loss = 'sparse_categorical_crossentropy',
              optimizer = 'adam')
```

In [100…
```
early_stop = EarlyStopping(monitor = 'loss',
                           patience = 3,
                           verbose = 1)
```

In [101…
```
history = model.fit(X_train, y_train,
                    epochs = 100, batch_size = 10,
                    callbacks = [early_stop])
```

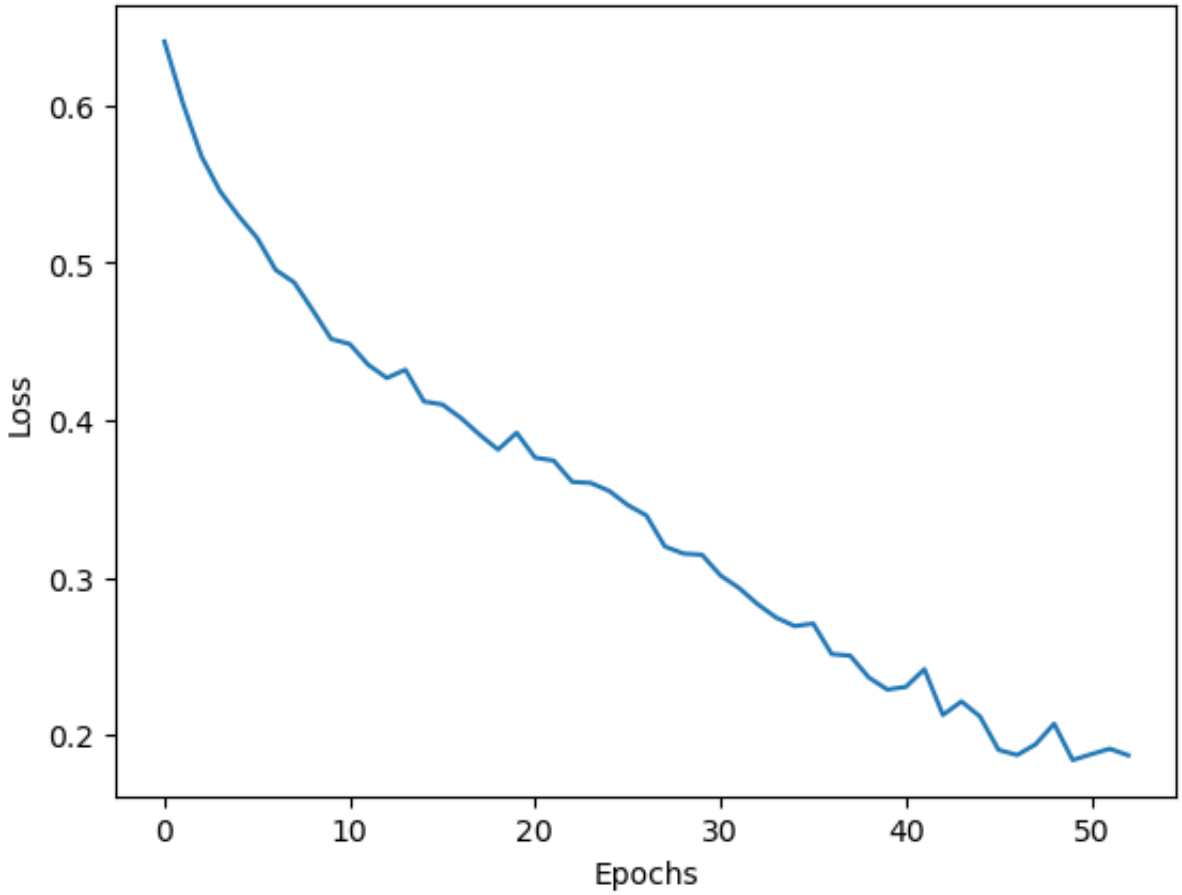
```
Epoch 1/100
74/74 [==============================] - 0s 482us/step - loss: 0.6409
Epoch 2/100
74/74 [==============================] - 0s 445us/step - loss: 0.6014
Epoch 3/100
74/74 [==============================] - 0s 466us/step - loss: 0.5678
Epoch 4/100
74/74 [==============================] - 0s 463us/step - loss: 0.5456
Epoch 5/100
74/74 [==============================] - 0s 450us/step - loss: 0.5299
Epoch 6/100
74/74 [==============================] - 0s 454us/step - loss: 0.5159
Epoch 7/100
74/74 [==============================] - 0s 453us/step - loss: 0.4956
Epoch 8/100
74/74 [==============================] - 0s 459us/step - loss: 0.4876
Epoch 9/100
74/74 [==============================] - 0s 458us/step - loss: 0.4700
Epoch 10/100
74/74 [==============================] - 0s 447us/step - loss: 0.4516
Epoch 11/100
74/74 [==============================] - 0s 468us/step - loss: 0.4484
Epoch 12/100
74/74 [==============================] - 0s 471us/step - loss: 0.4352
Epoch 13/100
```

```
74/74 [==============================] - 0s 462us/step - loss: 0.4269
Epoch 14/100
74/74 [==============================] - 0s 475us/step - loss: 0.4321
Epoch 15/100
74/74 [==============================] - 0s 462us/step - loss: 0.4119
Epoch 16/100
74/74 [==============================] - 0s 472us/step - loss: 0.4100
Epoch 17/100
74/74 [==============================] - 0s 468us/step - loss: 0.4014
Epoch 18/100
74/74 [==============================] - 0s 473us/step - loss: 0.3909
Epoch 19/100
74/74 [==============================] - 0s 473us/step - loss: 0.3813
Epoch 20/100
74/74 [==============================] - 0s 471us/step - loss: 0.3920
Epoch 21/100
74/74 [==============================] - 0s 449us/step - loss: 0.3762
Epoch 22/100
74/74 [==============================] - 0s 464us/step - loss: 0.3743
Epoch 23/100
74/74 [==============================] - 0s 442us/step - loss: 0.3608
Epoch 24/100
74/74 [==============================] - 0s 456us/step - loss: 0.3601
Epoch 25/100
74/74 [==============================] - 0s 481us/step - loss: 0.3550
Epoch 26/100
74/74 [==============================] - 0s 488us/step - loss: 0.3460
Epoch 27/100
74/74 [==============================] - 0s 499us/step - loss: 0.3394
Epoch 28/100
74/74 [==============================] - 0s 497us/step - loss: 0.3197
Epoch 29/100
74/74 [==============================] - 0s 492us/step - loss: 0.3152
Epoch 30/100
74/74 [==============================] - 0s 532us/step - loss: 0.3145
Epoch 31/100
74/74 [==============================] - 0s 509us/step - loss: 0.3013
Epoch 32/100
74/74 [==============================] - 0s 489us/step - loss: 0.2933
Epoch 33/100
74/74 [==============================] - 0s 484us/step - loss: 0.2830
Epoch 34/100
74/74 [==============================] - 0s 478us/step - loss: 0.2745
Epoch 35/100
74/74 [==============================] - 0s 462us/step - loss: 0.2691
Epoch 36/100
74/74 [==============================] - 0s 471us/step - loss: 0.2707
Epoch 37/100
74/74 [==============================] - 0s 475us/step - loss: 0.2513
Epoch 38/100
74/74 [==============================] - 0s 472us/step - loss: 0.2502
Epoch 39/100
74/74 [==============================] - 0s 480us/step - loss: 0.2364
Epoch 40/100
74/74 [==============================] - 0s 468us/step - loss: 0.2287
Epoch 41/100
74/74 [==============================] - 0s 480us/step - loss: 0.2304
```

```
Epoch 42/100
74/74 [==============================] - 0s 486us/step - loss: 0.2416
Epoch 43/100
74/74 [==============================] - 0s 486us/step - loss: 0.2125
Epoch 44/100
74/74 [==============================] - 0s 496us/step - loss: 0.2212
Epoch 45/100
74/74 [==============================] - 0s 515us/step - loss: 0.2116
Epoch 46/100
74/74 [==============================] - 0s 517us/step - loss: 0.1904
Epoch 47/100
74/74 [==============================] - 0s 539us/step - loss: 0.1871
Epoch 48/100
74/74 [==============================] - 0s 521us/step - loss: 0.1939
Epoch 49/100
74/74 [==============================] - 0s 520us/step - loss: 0.2070
Epoch 50/100
74/74 [==============================] - 0s 482us/step - loss: 0.1839
Epoch 51/100
74/74 [==============================] - 0s 483us/step - loss: 0.1876
Epoch 52/100
74/74 [==============================] - 0s 494us/step - loss: 0.1911
Epoch 53/100
74/74 [==============================] - 0s 497us/step - loss: 0.1869
Epoch 53: early stopping
```

In [102…

```
#History
plt.plot(history.history['loss'])
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.show()
```



In [103…
```
print(model.evaluate(X_train, y_train, verbose = 0))
print(model.evaluate(X_test, y_test, verbose = 0))
```

```
0.16541609168052673
1.3663681745529175
```

In [104…
```
y_pred_train = model.predict(X_train)
y_pred_test = model.predict(X_test)
```

```
24/24 [==============================] - 0s 390us/step
6/6 [==============================] - 0s 495us/step
```

In [105…
```
y_pred_train = y_pred_train[:,1].round()
y_pred_test = y_pred_test[:,1].round()
```

In [106…
```
print('Acuracy score on Train is:',
      accuracy_score(y_train, y_pred_train))
print('Acuracy score on Test is:',
      accuracy_score(y_test, y_pred_test))
```

```
Acuracy score on Train is: 0.9254742547425474
Acuracy score on Test is: 0.6
```

THE END