MÁSTER EN DATA SCIENCE AND BUSINESS ANALYTICS ONLINE

PREDICCIÓN DE ÉXITO EN STARTUPS A TRAVÉS DE MACHINE LEARNING

TFM elaborado por: Jorge Pérez Ruano

Tutor/a de TFM: Marta Ramírez Trillas

- Madrid, agosto de 2021 -

PREDICCIÓN DE ÉXITO EN STARTUPS A TRAVÉS DE MACHINE LEARNING

Escrito por

Jorge Pérez Ruano

Trabajo de Fin de Máster presentado para el Máster en Data Science y Business Analytics

Tutora: Marta Ramírez Trillas

Agosto 2021

RESUMEN

Las *startups* se están convirtiendo en uno de los principales motores de la economía mundial. Google, Apple, Facebook, Airbnb o Uber son empresas con un impacto enorme en todo el mundo. En los últimos diez años, Europa y Estados Unidos han sido testigos de un aumento exponencial de *startups*, un fenómeno que ha atraído a inversores cada vez más importantes. Es por lo tanto un fenómeno cada vez más estudiado y que trataremos de comprender en este proyecto.

En primer lugar, es importante definir qué es el éxito para una *startup*. En términos generales, una *startup* puede considerarse exitosa cuando consigue recibir fondos por parte de los fundadores, inversores y los primeros empleados, de forma que pueda crecer – frecuentemente a través de adquisiciones- hasta, posiblemente, llegar a salir a bolsa (IPO por sus siglas en inglés: *Initial Public Offering*).

Por lo tanto, la posibilidad de predecir correctamente el éxito de una *startup* otorgaría a sus inversores una gran ventaja competitiva, ya que pasarían a formar parte de la empresa a un coste bajo, antes de que se aprecie en los mercados financieros y la entrada sea mucho más costosa. De esta manera, se benefician por la subida de precios de las participaciones – ya sean públicas o privadas, aumentando el valor de su inversión considerablemente.

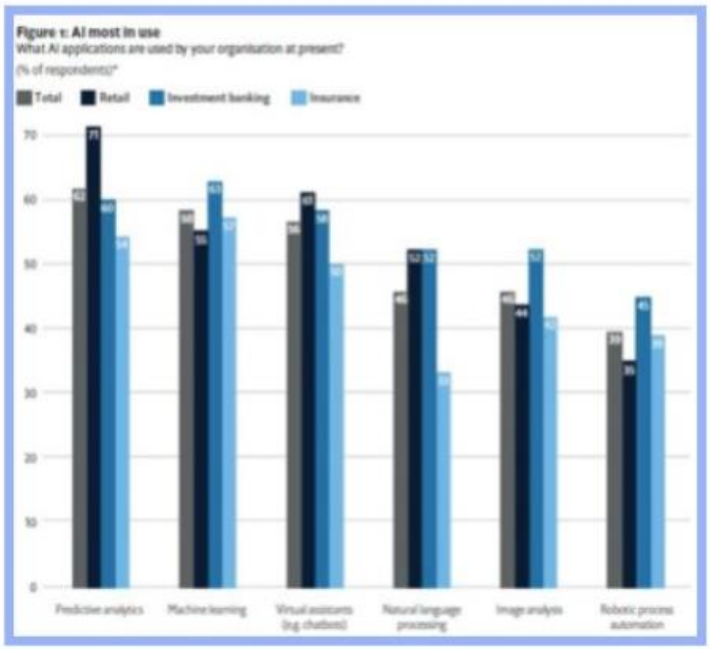
Precisamente uno de los principales problemas con los que se encuentran los inversores de todo el mundo a la hora de evaluar una posible operación es tratar de predecir el valor que la empresa tendrá en el futuro. ¿Cómo se puede reducir el riesgo de la inversión al mínimo y atraer así a más inversores que estén dispuestos a participar en las operaciones?

En la última década, bancos de inversión de todo el mundo han invertido miles de millones de dólares en desarrollar equipos tecnológicos con la capacidad de automatizar y predecir valores de empresas, bursátiles y de *commodities* de forma que les permita optimizar sus operaciones, reducir costes y aumentar sus beneficios al máximo.

Según un reciente estudio de The Economist, se prevé que el 86% de los principales gestores financieros del mundo aumenten su inversión en servicios de Inteligencia Artificial hasta 2025. Además, las entidades de banca de inversión están liderando esta tendencia por delante de Retail.

Hoy en día, los bancos dependen de modelos de aprendizaje automático para predecir modelos cuantitativos y reducir riesgos. De hecho, Machine Learning es la técnica de inteligencia artificial más utilizada en banca de inversión, seguida de cerca por *predictive analytics*.

Uso de IA por sector, The Economist



Uno de los momentos más importantes en la vida de una *startup*, suponiendo que lleguen a este punto, es su salida a bolsa. Este es el punto en el que los inversores públicos pueden pasar a formar parte de la empresa, y las acciones se abren a los *retail investors*, es decir, que cualquiera que lo desee puede invertir en la empresa. De ahí la importancia de este momento como hito en la vida de la compañía.

Este trabajo se va a centrar en gran medida en el momento de salida a bolsa de las empresas seleccionadas. El objetivo es crear un modelo que pueda predecir el rendimiento que tendrán las *startups* que han salido a bolsa recientemente. El estudio se realizará en base a los datos disponibles de empresas similares que salieron a bolsa anteriormente, y se compararán ambos *datasets* para obtener características similares que puedan ofrecernos ciertas garantías.

Para ello, hemos explorado la *database* de *startups* más importante del mundo: CrunchBase. Nuestro objetivo será construir un modelo que nos permita predecir qué *startups* serán exitosas y cuáles no en función al rendimiento pasado de empresas similares.

Una vez establecidas las características de las empresas que se estudiarán, pasaremos a entrenar un modelo de aprendizaje automático que nos permita predecir el rendimiento futuro de las *startups* en nuestro *dataset* inicial. De esta forma, las separaremos en dos grupos principales: aquellas que ofrecen un rendimiento superior, y las que no llegan a un determinado nivel.

Se considerará que tienen mayor posibilidad de beneficios futuros aquellas con un mayor rendimiento en el trimestre inmediatamente posterior a su salida a bolsa. La razón por la que se ha escogido esa fecha es porque el primer mes suele ofrecer valores artificialmente altos (debido a la agitación inicial de los inversores), algo que suele estar corregido al terminar el primer trimestre.

Utilizaremos diferentes modelos para cada empresa en base a su cotización en el último año, y en base a esos valores se intentará calcular su posible cambio en el trimestre inmediatamente posterior.

Datasets

Los datos utilizados - link proporcionado en la bibliografía- son de acceso público a través de Crunchbase.

El primero de ellos se llama Startup Investments, y contiene 11 tablas distintas -se pueden unir a través de un ID único. Las tablas tienen información sobre las empresas, sobre inversores individuales, noticias sobre las empresas, rondas de financiación, adquisiciones y salidas a bolsa (IPOs).

Nombre	Columnas	Valores Únicos
acquisitions	12	9 412
degrees	8	68 451
funding_rounds	23	31 939
funds	11	1 026
investments	8	21 607
ipos	13	1 254
milestones	9	17 159
objects	40	462 651
offices	15	95 043
people	6	226 709
relationships	11	202 088

El segundo *dataset* pertenece originalmente a un proyecto de la universidad politécnica de California (CAL Poly Slo), y se publicó en CrunchBase en 2018. Contiene información de empresas que hoy en día son públicas en EEUU (la información es de antes de que salieran a bolsa). Entre otras cosas, tiene información como el número de empleados, localización de la empresa, etc.

Nombre	Columnas	Valores Únicos
IPODataFull	1 664	90 412

La información más importante que utilizaremos de este *dataset* es referente al stock de la empresa: analizaremos su valoración individual con la del S&P 500 para entender en que puntos el rendimiento de la empresa ha superado a los índices bursátiles y así poder comprender qué retornos podría ofrecer a los inversores.

KEYWORDS

Startup, Initial Public Offering (IPO), Venture Capital, Mergers and Acquisitions (M&A), Data Analysis, Machine Learning

ÍNDICE

INTRODUCCIÓN

Gracias a la ayuda de universidades, gobiernos y empresas privadas que han aumentado la inversión, las *startups* son un tipo de empresa cada vez más común en el ámbito de los negocios. Cada vez es más común encontrar compañías que alcanzan rápidamente el estatus de unicornios (valoración de mil millones de dólares) en cuestión de pocos años. Uno de los últimos ejemplos es Slack, una app de mensajería que lo consiguió 1.25 años después de empezar a operar (Jim, 2015).

De hecho, otros ejemplos como Uber o Airbnb han crecido tan rápidamente que los gobiernos han tenido que acelerar su adaptación para poder crear regulaciones que se ajusten al impacto que estas empresas están teniendo en la sociedad. Muchas de estas empresas alcanzan un tamaño tal que el objetivo de los inversores es ser adquiridos por otra empresa aún más grande, como pasó por ejemplo con Facebook y WhatsApp, operación que generó un beneficio a Sequoia (un fondo de Venture Capital) 50 veces superior a su inversión inicial (Neal, 2014).

Aun así, no todo es perfecto: las *startups* son un tipo de negocio con un 90% de posibilidades de no salir adelante, es decir, de suponer una enorme inversión que no se recuperaría (Patel, 2015). Con esas estadísticas en la mano, parece natural que una correcta estrategia de predicción del éxito de una *startup* sea una de las principales fuentes de ingreso para sus fundadores, inversores y los primeros empleados, ya que la empresa puede tener una salida a bolsa (IPO) o ser adquirida por otra empresa más grande (M&A), lo que facilitaría que los inversores previos recibieran un retorno inmediato de su inversión.

Son muchas las áreas de nuestra sociedad que están observando diferentes mejoras por la aplicación del *Machine Learning*, desde la salud a la detección y prevención de fraude bancario (Raghupathi & Raghupathi, 2014). También es posible aplicar estas técnicas para dar ventajas a los inversores en *startups*, proporcionándoles información sobre cuáles son las empresas con más posibilidades de tener éxito en el futuro y, por tanto, con mayores ratios de *Return on Investment* (ROI). Por todo ello, este estudio pretende aplicar diferentes técnicas de *Machine Learning* y *Data Mining* para crear un modelo predictivo que ayude a clasificar una *startup* como exitosa o no.

Para generar el modelo de predicción, se han utilizado tres algoritmos de *Machine Learning*: por un lado, un Clasificador Naive Bayes Gaussiano, un K-Nearest Neighbors y un Random Forest. Éstos tres algoritmos encajan perfectamente con los datasets con los que vamos a trabajar, proporcionándonos una implementación técnicamente sencilla.

Poder crear este modelo predictivo será un gran indicador de las posibilidades que tienen los analistas de negocio para extraer el potencial de los datos disponibles. No se trata simplemente de beneficiar a los participantes de la industria de *Venture Capital*¸sino también de entender que la aplicación exitosa de todas estas técnicas está mejorando la literatura académica y la industria a pasos agigantados.

Aunque existen diversos estudios que predicen procesos de M&A, se centran mayoritariamente en datos financieros a través de la regresión logística, sin generar modelos más complejos (AliYrkkö, Hyytinen, & Pajarinen, 2005; Altman, 1968; Gugler & Konrad, 2002; Karels & Prakash, 1987; Meador, Church, & Rayburn, 1996; Ragothaman, Naik, & Ramakrishnan, 2003). Aún hay espacio suficiente en la industria para poder aplicar diferentes modelos de *Machine Learning* y algoritmos que puedan guiar a una empresa en su proceso de adquisición, y bases de datos como CrunchBase sirven como punto de apoyo para comparar estudios y metodologías previas (Liang & Daphne Yuan, 2012; Xiang et al., 2012).

Este estudio se divide en tres secciones principales: una primera parte donde se explorará la relevancia del propio estudio y se revisarán algunos proyectos previos; una segunda porción donde se acometerá la extracción y limpieza de los datos utilizados, así como un análisis exploratorio que nos ayude a entenderlos más en profundidad. Por último, se realizará la predicción del rendimiento futuro y la clasificación de las empresas en base a los resultados junto con las conclusiones del proyecto.

OBJETIVOS

Este proyecto tiene como objetivo principal el desarrollo de un modelo predictivo de clasificación binaria del éxito de una empresa de nueva creación (*startup*).

Los trabajos más recientes en este campo, como *A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles* (Xiang, Zheng, Wen, Hong & Rose, 2012), dejan cierto margen de mejora. Normalmente, otros estudios tienden a centrarse en cuestiones internas de la empresa y a pasar por alto factores relacionados con la financiación.

Por otro lado, también se puede mejorar la elección de las empresas con las que trabajar, siendo más selectivo a la hora de gestionar el *dataset* sobre el que se trabajará.

OBJETIVOS TÉCNICOS

Durante este proceso, esperamos poder resolver los siguientes objetivos técnicos:

En primer lugar, durante la fase de Análisis de Datos, tener un entendimiento completo de la base de datos de CrunchBase, así como un proceso de limpieza de datos (valores faltantes, duplicados y valores redundantes) que faciliten su mejor comprensión. Preparar la base de datos de forma que se obtenga un set de datos listo para su explotación mediante algoritmos de aprendizaje automático. La transformación se llevará a cabo mediante la definición y creación de nuevos *features* que generarán el *dataset* definitivo que se usará en el modelo final.

La segunda fase será el proceso de experimentación y obtención d resultados, donde se trabajará aplicando diferentes algoritmos de aprendizaje automático para general el mejor modelo posible e intentar mejorar lo que hasta ahora se ha conseguido en los estudios disponibles. Los algoritmos utilizados serán un Clasificador Naive Bayes Gaussiano, un K-Nearest Neighbors y un Random Forest. Posteriormente se presentarán y discutirán sus conclusiones.

2. REVISIÓN DE LA LITERATURA

Esta revisión tiene como objetivo identificar tendencias relacionadas con el estudio, así como similitudes y diferencias con trabajos previos que permitan enriquecer este proyecto, tratando de hacer su contenido único e innovador para la escena de *startups*.

2.1 EL ECOSISTEMA STARTUP

2.1.1 Definición e importancia de las Startups

Por definición, las *startups* crean productos centrados en nichos de mercado que no necesariamente se tienen que haber desarrollado con anterioridad. Su propia naturaleza hace que sean negocios arriesgados e impredecibles, ya que un producto o servicio nuevo no tiene por qué funcionar y puede requerir innumerables ajustes antes de encontrar su nicho adecuado. La definición última de una *startup* es una empresa que presenta un riesgo alto que se encuentra una primera fase de operaciones, y que normalmente está relacionada con la tecnología o con los servicios (Ries, 2011).

En la mayoría de los casos, este tipo de empresas no son sostenibles a medio-largo plazo sin financiación externa, ya que suelen contar con productos por desarrollar o con nichos de mercado que tienen barreras de entrada elevadas debido a los costes de distribución y de economías de escala.

Según Peter Thiel, fundador de Paypal y Director Ejecutivo de Palantir, una *startup* es una creadora de innovación vertical y no horizontal, siendo vertical la tecnología que jamás antes se ha utilizado y horizontal el proceso de globalización, distribuyendo tecnología existente donde aún no se utiliza. Thiel también considera que una *startup* debe ser capaz de crear un monopolio en un nicho de mercado muy definido antes de poder expandirse a otros mercados (Thiel & Masters, 2014).

Por otro lado, Paul Graham, fundador de Y Combinator, tiene otra definición de *startup*: "una *startup* es una empresa diseñada para crecer rápido". Su opinión contrasta radicalmente con la de Thiel, ya que Graham considera que la tecnología no debe ser el foco principal de una *startup*. "El hecho de ser una empresa recién fundada no te convierte en una *startup*. Tampoco es necesario que se centre en tecnología o que tengan financiación externa. Lo único que es fundamental es el crecimiento" (Graham, 2012).

La definición de Graham da una visión algo más realista del mercado de las *startups*, puesto que se basa en un modelo que no requeriría que las empresas se endeudaran excesivamente, pudiéndose centrar en expandirse lo antes posible.

Con un mayor foco en las *startups* tecnológicas, Steve Blank, un reputado empredador americano, enumera cuatro principales razones que explican el fenómeno de este tipo de empresas en su libro "*The Four Steps to the Epiphany*" (Blank, 2006):

- **Las *startups* pueden construirse para miles de personas en lugar de millones**: la dramática reducción de los costes de producción en las últimas décadas (Hermann et al., 2015) hacen que sea más barato que nunca construir tecnología. El acceso a las herramientas, el código *open source*, los servidores con precios más reducidos y la creciente comunidad de desarrolladores hace que los productos puedan desarrollarse y testearse en todo el mundo. El mejor ejemplo es WhatsApp, que fue adquirido por 19mil millones de dólares cuando tan sólo contaba con 16 empleados (Neal 2014).
- **Mayores posibilidades de Capital Riesgo**: cuando la industria de *Venture Capital* tenía que invertir millones de dólares en el pasado, lo tenían que hacer distribuyendo las inversiones en pequeños grupos. Sin embargo, gracias al abaratamiento de la tecnología y la aparición de nuevos tipos de inversores (*angel investors, accelerators y micro-VCs*), ya no existe la necesidad de invertir cantidades de dinero tan elevadas. Ahora, las *startups* tienen acceso a muchos otros tipos de financiación que pueden complementar las de los fondos de Venture Capital más tradicionales.
- **El crecimiento del emprendimiento como ciencia**: desde la primera ola de la era de la información, allá por la década de los 70, muchos emprendedores comenzaron a hacer uso de su conocimiento de la ciencia de gestión liderada por Henry Ford. Sin embargo, tras el colapso de la burbuja del *dotcom* en los noventa, muchos de ellos se dieron cuenta de que las *startups* eran un movimiento totalmente distinto y que no podían aplicar las mismas reglas que con otro tipo de empresas. Con el tiempo, los emprendedores han mejorado notablemente a la hora de crear *startups* (Hermann et al., 2015).
- **Aumento de la velocidad de adopción**: gracias al desarrollo de internet, las *startups* pueden convertirse en lo que Steve Blank denomina "micro-multinacionales", teniendo acceso a trabajadores y productos de todo el mundo sin ningún inconveniente (Blank, 2006). Google y Facebook son las pruebas fehacientes de que la localización tiene cada vez menos importancia. Otro caso llamativo es el de Slack, la compañía más rápida de la historia en alcanzar los mil millones de valoración (se convirtió en un *unicornio* a los 1.25 años de vida), es una *startup* que ha tenido clientes como Airbnb, Buzzfeed, eBay, Salesforce e incluso la NASA a través de un software de bajo coste y un producto bien definido ("Slack: Customer Stories", 2017).

Todo esto se ha traducido en un notable aumento de la financiación en *startups*, tanto en sus primeros años de vida como en ciclos más avanzados donde las empresas están más establecidas. Esa financiación ha tenido como resultado unos importes medios en las transacciones nunca vistos hasta el momento:

Evolución del importe medio de las operaciones

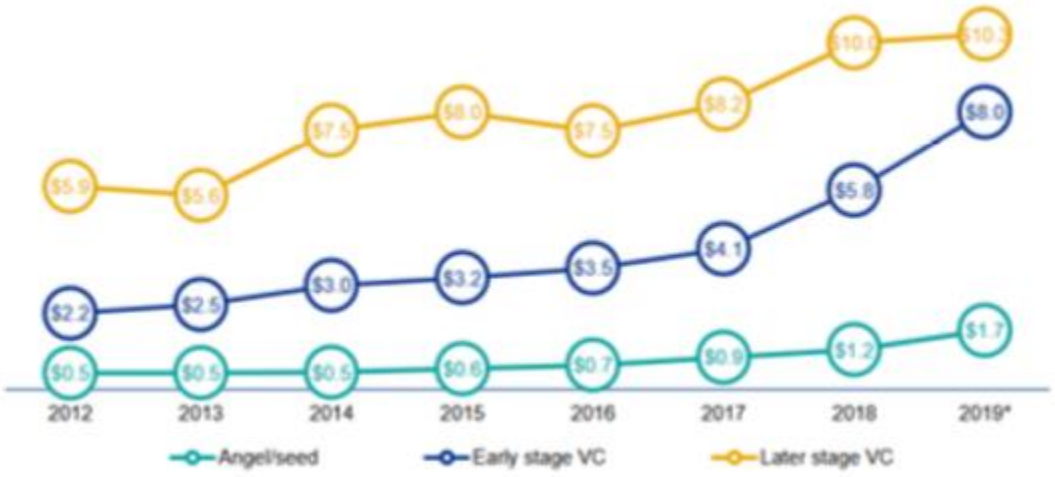


Figura 1 – Evolución del importe medio de las operaciones

A pesar de que la tendencia ha sido generalizada, podemos observar que es especialmente pronunciada tras la primera y segunda ronda de financiación, una vez que la empresa está relativamente establecida en el mercado y ha tenido la oportunidad de darse a conocer de cara a inversores y consumidores:

Evolución del importe medio de operaciones por etapa

Figura 2 – Evolución del importe medio de las operaciones por etapa

Además de todo esto, el acceso a técnicas analíticas cada vez más avanzadas que han facilitado la adopción de clientes y la expansión de los negocios también ha permitido que este tipo de empresas puedan crecer más que nunca. Por otro lado, técnicas como el *data mining* han permitido a los emprendedores tener acceso a información que antes era más complicado conseguir, otorgando a la vez a los inversores con muchos más datos para poder optimizar sus inversiones y maximizar sus beneficios (Hermann et al., 2015).

A pesar del aumento del importe medio por operación en el ecosistema *startup*, lo cierto es que la formación de empresas de nueva creación ha disminuido en los últimos años, en contraste con la creación de aplicaciones de negocio. Así, hemos visto un aumento exponencial de las herramientas que las empresas utilizan en su día a día, pero un movimiento contrario en la formación de empresas que ha llegado a disminuir en un 10% respecto a los niveles de 2005.

Year	Business Application (All)	Business Formations (Within 4 Quarters)	Business Formations (Within 4 Quarters)
2005	2,502,014	470,052	19%
2006	2,644,204	425,014	16%
2007	2,659,813	386,987	15%
2008	2,556,042	322,563	13%
2009	2,401,128	288,503	12%
2010	2,463,839	290,965	12%
2011	2,537,102	289,092	11%
2012	2,542,219	280,092	11%
2013	2,582,539	282,436	11%
2014	2,689,139	289,360	11%
2015	2,786,711	291,663	10%
2016	2,945,758	297,778	10%
2017	3,176,109	304,906	10%
2018	3,472,126	324,928	9%

Figura 3 – Evolución de la fundación de empresas en EEU, 2005-2018

2.1.1 Definición del éxito para las Startups: IPOs y M&A

El éxito de una *startup* se define normalmente como una estrategia con dos vías principales: una salida a bolsa (IPO) o una adquisición o venta de las acciones (M&A) con otra compañía donde los que han invertido previamente reciben una compensación económica. Este proceso se denomina estrategia de salida (Guo, Lou & Pérez-Castrillo, 2015).

Las fusiones y adquisiciones (M&A) suelen tener un papel fundamental en la reestructuración de una empresa. Según Alam & Kham (2014), una fusión es el proceso de unión de dos empresas en una sola (normalmente con un nombre nuevo) para incrementar las ventas y el beneficio; suele ser bastante más común en empresas no tecnológicas y de un tamaño similar.

Suelen ser operaciones especialmente delicadas en industrias de alta tecnología, ya que se utilizan para desarrollar con rapidez una tecnología que se ha adquirido de la otra empresa (Wei, Jiang & Yang, 2009). “Una adquisición es una operación donde una firma absorbe a otra y la última deja de existir. En estos casos, la adquisición ocurre cuando una empresa toma un control interesado en la otra” (Machiraju, 2013).

La lógica que mueve estas operaciones es que ambas empresas crean más valor juntas que por separado. Es una de las formas más comunes en las que las empresas pueden luchar por mantener sus ventajas competitivas en el mercado (Machiraju, 2003).

Una predicción de M&A es una fuente de ayuda muy importante para las *startups*, que pueden de esta manera entender sus posibilidades reales de ser adquiridas o de fusionarse con otra empresa, además de entender qué otras firmas podrían estar interesadas (Alam & Kham, 2014). Según las fuentes de CrunchBase, 2016 fue el año con más adquisiciones en la última década con un total de 7.899, lo cual supone un incremento del 72% respecto a las 4.589 del año anterior.

Por su parte, las IPOs consisten en la "venta de acciones de una empresa privada en los mercados públicos" (Li &Liu, 2010). Por ello, este momento tiene una importancia enorme en la vida de la empresa. Normalmente hay tres vías tras la salida a bolsa para cada empresa: continuar creciendo de forma orgánica como empresa pública, ser adquirida tras mostrar un rendimiento alto o bajo, o ser dada de baja de la bolsa al final de su ciclo de vida.

No existe una hoja de ruta fija para todas las *startups* con una estrategia de salida perfecta. Cada caso depende del rendimiento de la empresa, las condiciones de los mercados financieros, el acceso a la información valiosa, etc (Akerlof, Yellen & Katz, 1970).

Aun así, en el ecosistema *startup*, cualquiera de los eventos descritos anteriormente se considera un gran éxito para la empresa, ya que ambos aumentan notablemente el dinero que reciben sus fundadores, inversores y primeros empleados a corto plazo (Guo et al., 2015). Una de las razones más frecuentes para que las *startups* acaben adquiriendo a empresas más pequeñas es adquirir su talento: no sólo estarían adquiriendo su tecnología, sino también sus empleados. Este tipo de operación se conoce como *adquihiring* y supone una estrategia de crecimiento exponencial que aumenta la competitividad en los mercados (Marita Makinen, Haber & Raymundo of Lowestein Sandler, 2014).

Por otro lado, la creación de valor para las *startups* está directamente ligada a ciertas características como la ronda de financiación en la que se encuentran, su industria, el importe de los fondos que han conseguido obtener por parte de inversores y la localización. Todos ellos constituyen una batería de elementos que determinan su valor para los inversores, como se ilustra en la siguiente tabla:

		Stage		Industry		IPO rate		Fund size		Location		
	All	Early	Late	IT	Health	High	Low	Large	Small	CA	OthUS	Fgn
Important factor												
Deal flow	65	68	65	73***	49***	62	64	69	62	73	67	57***
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(4)
Selection	86	87	87	91**	81**	89	88	88	85	87	87	84
	(1)	(2)	(4)	(3)	(4)	(3)	(3)	(2)	(2)	(3)	(2)	(3)
Value-add	84	85*	77*	78**	89**	87	83	84	83	86*	79*	89**
	(2)	(2)	(5)	(4)	(4)	(3)	(3)	(2)	(2)	(3)	(3)	(2)
Other	4	3	6	3	3	5	4	4	4	2	4	5
	(1)	(1)	(3)	(1)	(2)	(2)	(2)	(1)	(1)	(1)	(1)	(2)
Most important factor												
Deal flow	23	27	19	29***	13***	19**	31**	27	21	27	25	18**
	(2)	(3)	(4)	(4)	(4)	(3)	(4)	(3)	(2)	(4)	(3)	(3)
Selection	49	44	52	49	52	57**	46**	51	46	48	50	48
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(4)
Value-add	27	27	27	21**	35**	22	22	22***	32***	23	23	34**
	(2)	(3)	(5)	(4)	(5)	(3)	(3)	(3)	(3)	(3)	(3)	(3)
Other	1	1	2	1	0	2	1	1	1	1	1	0
	(0)	(1)	(1)	(1)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(0)
Number of responses	509	226	82	122	78	129	139	231	281	145	205	179

Figura 4 – Factores de valor principales para el Venture Capital

El porcentaje de inversores de *Venture Capital* que señaló cada característica como importante de mayor a menor muestra la importancia de la industria y la localización sobre las demás. Será también interesante ver si la localización sigue jugando un papel fundamental en los próximos años con la introducción cada vez más extendida del teletrabajo.

Por otro lado, respecto a los factores determinantes relativos a la propia actividad de la empresa, vemos como la plantilla y el modelo de negocio son los dos puntos de lejos más importantes para los inversores, por encima del producto o la industria que ocupe la *startup*, especialmente al principio de su existencia:

		Stage		Industry		IPO rate		Fund size		Location		
	All	Early	Late	IT	Health	High	Low	Large	Small	CA	OthUS	Fgn
Important factor												
Team	95	96	93	96	91	96	96	96	95	97	93	96
	(1)	(1)	(3)	(2)	(3)	(2)	(1)	(1)	(1)	(1)	(2)	(1)
Business model	83	84	86	85*	75*	79	82	83	82	83	84	81
	(2)	(2)	(4)	(3)	(4)	(3)	(3)	(2)	(2)	(3)	(2)	(3)
Product	74	81***	60***	75	81	75	74	71*	77*	81**	71**	73
	(2)	(2)	(5)	(4)	(4)	(3)	(3)	(3)	(2)	(3)	(3)	(3)
Market	68	74	69	80***	56***	68	74	67	70	76**	66**	64
	(2)	(3)	(5)	(3)	(5)	(4)	(3)	(3)	(3)	(3)	(3)	(3)
Industry	31	30	37	33**	19**	25	29	30	31	31	37	24***
	(2)	(3)	(5)	(4)	(4)	(3)	(3)	(3)	(3)	(3)	(3)	(3)
Valuation	56	47***	74***	54*	42*	59*	49*	59*	52*	63	60	46***
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Ability to add value	46	44	54	41	45	39*	48*	41**	51**	46	48	46
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Fit	50	48	54	49	40	38**	50**	46**	54**	48	51	50
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Most important factor												
Team	47	53**	39**	50***	32***	44	51	44	50	42	44	55***
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Business model	10	7***	19***	10	6	7	11	10	10	11	11	8
	(1)	(2)	(4)	(3)	(3)	(2)	(2)	(2)	(2)	(2)	(2)	(2)
Product	13	12	8	12***	34***	18*	11*	15*	10*	13	14	11
	(1)	(2)	(3)	(3)	(5)	(3)	(2)	(2)	(2)	(2)	(2)	(2)
Market	8	7	11	13*	6*	11	10	11***	5***	15***	5***	5
	(1)	(2)	(3)	(3)	(3)	(2)	(2)	(2)	(1)	(3)	(1)	(2)
Industry	6	6	4	3*	9*	6	3	7*	4*	7	7	2**
	(1)	(1)	(2)	(2)	(3)	(2)	(1)	(2)	(1)	(2)	(2)	(1)
Valuation	1	0***	3***	0*	2*	3	1	2	1	2	1	1
	(0)	(0)	(2)	(0)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Ability to add value	2	2	2	1	1	2	2	1	2	1	2	2
	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Fit	14	13	13	9	9	9	12	10**	17**	10*	16*	15
	(1)	(2)	(4)	(2)	(3)	(2)	(2)	(2)	(2)	(2)	(2)	(2)
Number of responses	558	241	90	129	86	138	156	251	310	161	218	199

Figura 5 – Factores de valor principales para el Venture Capital según localización

2.2. ANÁLISIS DE DATOS

2.2.1 Data Mining

Vivimos en una sociedad donde todas nuestras transacciones – ya sean de negocios, científicas o gubernamentales- están informatizadas, a la vez que nuestros dispositivos digitales, nuestras redes sociales y códigos de barras generan cantidades ingentes de datos.

Los científicos de datos se han encontrado con el reto de intentar aumentar nuestra habilidad para generar y recolectar datos a través de nuevas técnicas y herramientas automatizadas, tratando de transformar los abundantes datos en información y conocimiento (Han & Kamber, 2006; Kantardzic, 2003).

Ian Witten y Eibe Frank describen la minería de datos como el proceso de obtención de información hasta entonces desconocida, explícita y con alto potencial que se encuentra en una base de datos (Witten, Frank, & Eibe, 2000). Al construir programas que examinen bases de datos se pueden encontrar patrones que permitan extrapolar soluciones para problemas complejos, así como realizar predicciones más acertadas sobre datos futuros.

El ejemplo que utilizan para ilustrar su teoría es el de la predicción meteorológica. Cualquier persona puede pronosticar sus actividades al día siguiente prestando atención a cuatro factores fundamentales: la previsión de la temperatura, la humedad, la lluvia y el viento (Witten et al., 2000).

El aprendizaje automático podría entenderse como las bases técnicas de la minería de datos, que se utiliza para extraer información en bruto de bases de datos. El proceso de búsqueda de patrones se realiza de forma automática o semiautomática (siendo este último el más frecuente). Sin embargo, la mejor distinción es tratar de pensar en el aprendizaje automático como los algoritmos matemáticos que se usan para crear modelos, mientras que la minería de datos es el proceso de extracción de conocimiento – que puede o no tener técnicas de aprendizaje automático- (Witten et al., 2000).

La minería de datos se utiliza con frecuencia en el mundo de los negocios, la investigación e incluso la ciberseguridad, ya que combina métodos de aprendizaje automático, estadísticas y gestión de bases de datos para analizar los datos. El crecimiento exponencial de las bases de datos ha hecho que el proceso de minería se automatice, por lo que el proceso manual que se utilizaba en el pasado ha sido sustituido por un enfoque mucho más indirecto, impulsado también por recientes descubrimientos en los campos de la computación, las redes neuronales, los árboles de decisión y otros algoritmos complejos (Christopher Clifton, 2009; Kantardzic, 2003).

Los beneficios de la minería de datos pueden observarse en campos tan diversos como la salud, donde estas técnicas se han convertido en un pilar fundamental del desarrollo de la industria en los últimos años. Poder evaluar de forma efectiva los tratamientos a aplicar, comparando causas, síntomas y diferentes posibles efectos de los tratamientos ha reducido los costes y optimizado la gestión enormemente (Kudyba, 2014). Además, la minería de datos también puede ayudar a detectar patologías crónicas que permitan anticiparse a ciertas enfermedades y comenzar tratamientos efectivos con antelación, beneficiando enormemente a los pacientes (Chye Koh & Tan, 2011).

Otro de los campos donde más se utiliza la minería de datos es el Marketing. Las técnicas más utilizadas tienen que ver con la segmentación de clientes y el análisis de bases de datos para obtener información sobre los clientes y sus hábitos de consumo (Hill, 2012). Todo ello permite optimizar la distribución, compra y cadena de suministros de diversas industrias que tratan con clientes de diversos grupos de población.

2.2.2 Machine Learning

Durante los últimos 50 años, el aprendizaje automático ha evolucionado desde los intentos de Arthur L. Samuel de enseñar a una máquina a jugar a las damas (Samuel, 1962) hasta convertirse en una disciplina que se enseña en las escuelas científicas de todo el mundo y se aplica en nuestro día a día con la tecnología. Con el aumento imparable del poder computacional se ha posibilitado el uso de técnicas mucho más complejas que las que podían aplicarse hace medio siglo. Los algoritmos de aprendizaje automático se han convertido en algo tan común que prácticamente todas las nuevas aplicaciones se basan en ellos (Beyer, 2015).

El aprendizaje automático puede dividirse en cuatro categorías principales: supervisado, no supervisado, semi-supervisado y aprendizaje por refuerzo, siendo los dos primeros los más comunes.

Los algoritmos supervisados realizan predicciones basadas en un grupo de ejemplos. Un algoritmo supervisado quiere decir que teniendo un input *x* tendremos un output *y*. El algoritmo aprende la función (y=f(x)) y puede predecir y clasificar cualquier nuevo output *y* tras recibir el nuevo input de datos *x*. Por lo tanto, todas las respuestas posibles son conocidas y los datos son etiquetados en base al aprendizaje del algoritmo.

Por otro lado, los no supervisados se dan cuando sólo tenemos variables de *input,* pero no hay ningún *output* concreto (variable target). El algoritmo sólo podrá clasificar o predecir el output tras el proceso de aprendizaje. En este caso, no se conocen las respuestas posibles. Al tratarse de datos sin clasificar, será el propio algoritmo el que tendrá que aprender a reconocer los patrones (Aggarwal, 2015; Berry & Linoff, 2004; Han & Kamber, 2006; Kantardzic, 2003; Mitchell, 2006).

Tras entender de qué se trata el aprendizaje automático, es natural que las empresas hayan descubierto su potencial y lo apliquen a todas sus operaciones, incluyendo a la predicción del éxito de las *startups*. Tanto las técnicas supervisadas como las no supervisadas pueden proporcionar enormes beneficios en este campo, y depende de los que trabajan en él saber sacar el máximo provecho de los datos.

3. METODOLOGÍA

La metodología empleada en este trabajo es una interpretación aproximada del enfoque del Descubrimiento de Conocimiento en Bases de Datos (Fayyad et al., 1996). En primer lugar, se seleccionan los datos con los que se trabajará definiendo las tablas precisas de la base de datos de CrunchBase; después se pre-procesan los datos mediante la limpieza, selección y transformación de los mismos. Es en este punto cuando tratamos los *outliers*, valores faltantes, duplicados, y otros tipos de problemas similares. Se realiza un análisis exploratorio (EDA). A continuación, se prepara el experimento, definiendo las métricas a considerar y los principales problemas que hemos encontrado en el *dataset*. Trabajamos con diferentes algoritmos para generar una clasificación binaria (empresa exitosa o no exitosa). Por último, obtenemos los resultados, sacamos conclusiones de los mismos y las interpretamos.

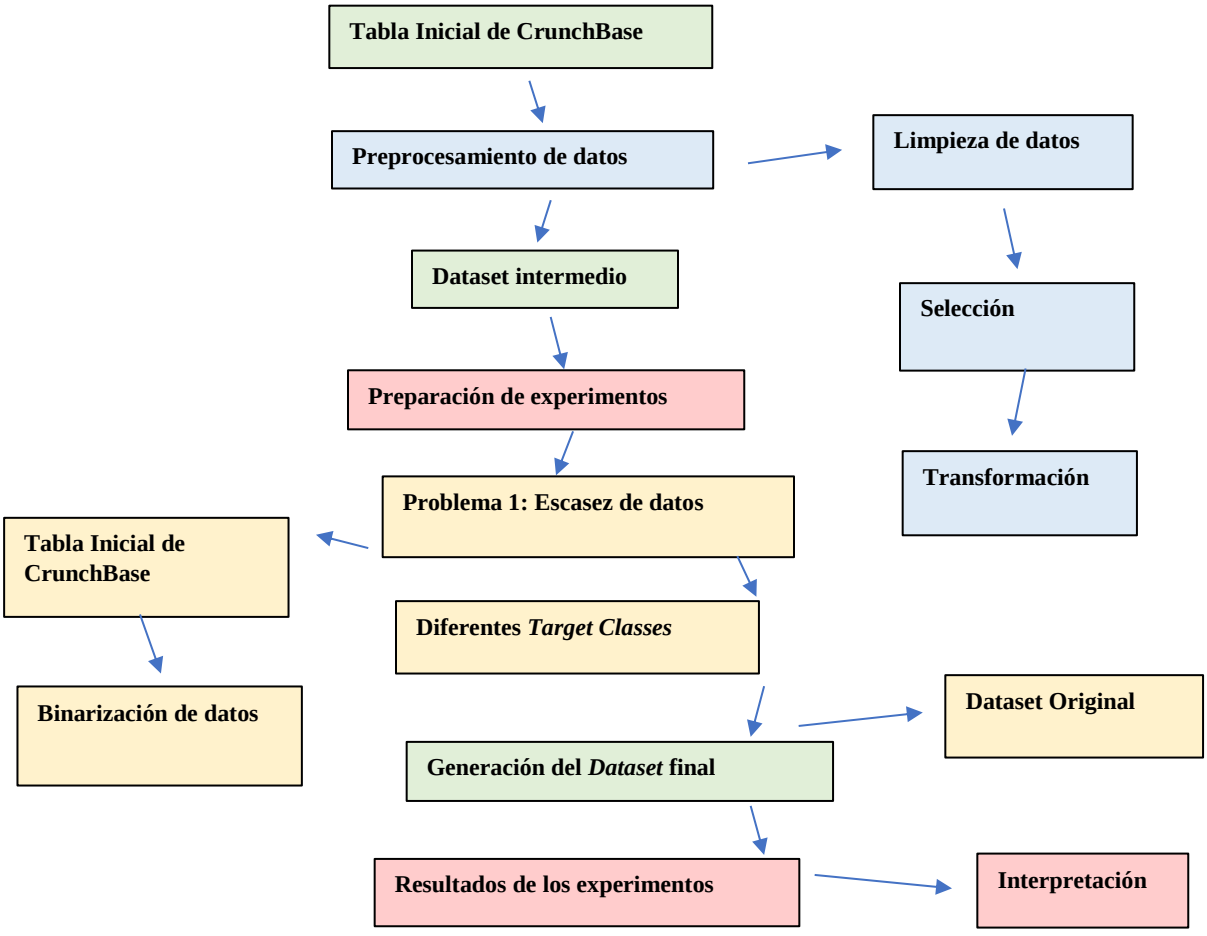


Figura 6 – Esquema explicativo de la metodología seguida en este proyecto

3.1. COLECCIÓN Y SELECCIÓN DE DATOS

Los datos utilizados para este proyecto se han obtenido enteramente de CrunchBase. Se trata de una página web de referencia para el ecosistema *startup* y para el mundo de las inversiones en general. Las bases de datos disponibles se dividen por áreas y están conectadas a través de IDs únicas. De todas los datos disponibles, los listados en la tabla a continuación son los que hemos considerado como más relevantes para el presente estudio.

En el momento de la consulta (son datos que se van actualizando continuamente), la Base de Datos de CrunchBase contaba con 12 tablas en formato CSV (comma-separated-values) con un total de 154 columnas en 11 de ellas y 1.644 en otra. Las utilizaremos todas ya que están relacionadas directamente con nuestro trabajo y aportan información importante para llegar a nuestras conclusiones.

Nombre	Observaciones	Columnas
acquisitions	Contiene información sobre las *startups* que han sido adquiridas	12
degrees	Contiene información sobre la formación de los empleados de *startups*	8
funding_rounds	Contiene información sobre las rondas de financiación	23
funds	Contiene información sobre fondos inversores de *Venture Capital*	11
investments	Contiene información sobre diferentes inversiones de *Venture Capital*	6
ipos	Contiene información sobre salidas a bolsa	13
milestones	Contiene eventos significativos en el ecosistema *startup*	9
objects	Fichero principal, contiene la mayoría de los datos con los que se trabajará	40
offices	Contiene información sobre las oficinas de las *startups*	15
people	Contiene información sobre individuos que trabajan en el ecosistema *startup*	6
relationships	Contiene información sobre particulares y su relación con las *startups*	11
ipodatafull	Contiene información financiera sobre salidas a bolsa	1 664

Figura 7 – Observaciones

Las siguientes tablas muestran datos generales sobre las últimas rondas de financiación disponibles en el momento de la recogida de los datos. Se ha excluido "organizations", "degrees" y "people" ya que deben ser preprocesados.

3.2. PREPROCESAMIENTO DE LOS DATOS

El preprocesamiento de datos puede tener una importancia crítica en un proyecto de *Machine Learning*. En este caso, se van a realizar cambios generales a todas las tablas, además de unirlas todas a través de IDs únicos. Debido a la naturaleza de los datos, empezaremos tratando de entender la interdependencia entre las tablas y tratando de no minimizar las correlaciones.

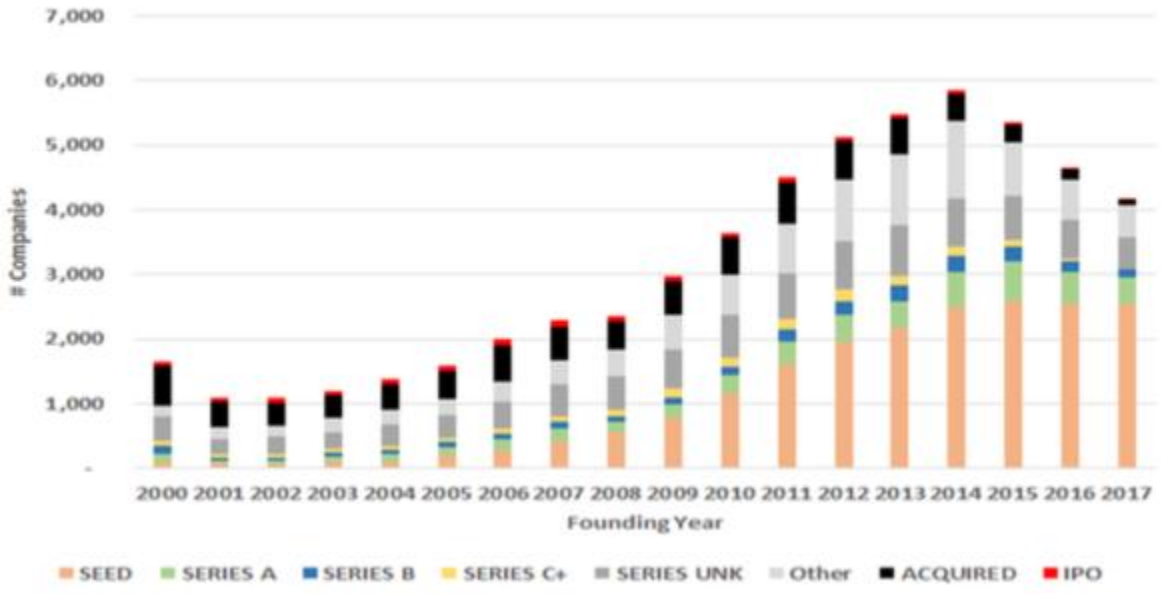


Figura 8 – Evolución de la financiación de las empresas según su año de fundación

En general, el preprocesamiento de datos tiene tres fases principales:

- **Limpieza de los datos:** el autor trata de eliminar los valores redundantes e irrelevantes, así como duplicados, valores faltantes y *outliers*.
- **Selección de los datos:** se define el contexto del estudio, como por ejemplo los criterios sociodemográficos que guiarán la selección del *dataset* final.
- **Transformación de los datos:** creación de nuevas variables, agregando datos de tablas diferentes.

3.2.1 Limpieza de datos

El primer paso en este proceso consiste en eliminar o tratar los datos irrelevantes y redundantes. De esta manera, evitaremos llegar a conclusiones falsas que puedan afectar a nuestro estudio. En general, se han encontrado 63 duplicados en las tablas; hemos procedido a eliminarlos.

A continuación, procedemos a eliminar los datos que consideramos que puedan ser poco fiables o aquellos que no tengan información correcta para nuestro estudio. En este caso, hay algunas empresas en la tabla "ipodatafull" que cuentan con valores no disponibles para los últimos días de mercado, los cuales eliminaremos.

A modo de aclaración, se consideran valores faltantes aquellos que no tienen un valor guardado en la observación. Debido al efecto que pueden tener en la obtención de conclusiones en los modelos de aprendizaje automático, se suele considerar como regla general que dicho valor no existe (Witten et al., 2000).

En este caso, la falta de valores se debe probablemente a falta de información por parte de los creadores del *dataset*, lo cual dificulta separar los datos faltantes de aquellos que simplemente no aportan información. Para este trabajo, vamos a seguir la premisa de "a menor cantidad de datos, más incrementa el ratio de acierto" (Kotsiantis et al., 2006).

3.2.2 Selección de datos

Antes de continuar, es importante contextualizar los datos que se utilizarán en este proyecto. Debido al peso de la EEUU como principal motor de la industria de *startups*, se han considerado mayoritariamente empresas de dicho país para el set de entrenamiento. Se han separado tal y como venían en el *dataset*, por las siglas correspondientes a sus estados/regiones.

- Al formar parte de una web de noticias estadounidense (TechCrunch), cada artículo está referenciado con datos de la base de datos. Tener más cobertura en los medios supone también una mayor cantidad y calidad del contenido.
- La plataforma se encuentra en inglés, lo cual limita el input de otros usuarios en diferentes idiomas.
- Utilizando una estrategia similar a la de Xiang et al., (2012), a pesar de que ellos utilizaron regiones en lugar de estados, CrunchBase sólo comenzó a considerar *startups* internacionales en 2014 (Lennon, 2014).
- Histórica y actualmente, California es el estado con mayor presencia de compañías tecnológicas del mundo (Weller, 2016).
- Los cinco estados norteamericanos más representados en nuestro estudio son los siguientes:

USA State Code	Count	%	Successful Companies	Target Ratio
MA	4 609	5%	922	20%
TX	4 967	6%	695	14%
NY	9 926	11%	1 191	12%
CA	27 291	32%	4 912	18%
Other	39 795	46%	5 969	15%
All	86 588	100%	13 689	16%

Figura 9 – Representación de compañías exitosas en EEUU

Empresas fundadas entre 1985 y 2014: A pesar de que algunas de estas compañías podrían no denominarse *startups* debido a la gran cantidad de tiempo sin tener ningún evento especialmente exitoso en este contexto, hay que considerar que en algún momento sí que lo fueron y que probablemente tuvieron rondas de financiación que les pusieron en condiciones exitosas. Es una estrategia similar a la seguida por Xiang et al., (2012), que supone que las empresas necesitan cierto tiempo para madurar y mostrar resultados convincentes. También se cubren la burbuja Dot-Com (1997) y la crisis financiera de 2008.

Empresas con al menos una revisión en los primeros 90 días desde la creación de su perfil: tanto usuarios como moderadores pueden enviar sus opiniones sobre las empresas. Al sólo tener acceso a la fecha de creación y la fecha de última modificación, hemos filtrado por aquellas empresas que tengan al menos 90 días entre las dos fechas. De esta forma, garantizamos que las *startups* hayan tenido al menos una revisión en ese período y limitamos las posibilidades de encontrarnos con datos de empresas falsas.

Categoría empresarial: Se han seleccionado las empresas cuya categoría está especificada para un análisis más en profundidad. Esta categoría refleja tanto la industria como si se trata de una empresa tecnológica o no.

3.2.3 Transformación de los datos

La transformación de datos puede resumirse en "la aplicación de modificaciones matemáticas al valor de una variable" para extraer más valor de la misma del que tenía originalmente (Osborne, 2002). En este proyecto, el proceso de transformación puede dividirse en dos fases sucesivas:

1. Cambios en los datos originales
2. Creación de nuevas variables

3.2.3.1 Cambios en los datos originales

Los cambios se han aplicado de forma consistente a todas las variables. Hemos cambiado los nombres de las columnas del inglés al español para conseguir mayor consistencia en las gráficas, como por ejemplo en el *dataset* de oficinas, donde se han hecho las siguientes modificaciones:

Dataset Original	Dataset Modificado
Description	Descripción
Address	Dirección
Object_id	ID
Latitude	Latitud
Longitude	Longitud

Figura 10 – Cambios sobre los datos originales

Por otro lado, todas las columnas con las que trabajamos son *strings*, excepto el número de *milestones* y el de empleados, que son de tipo numérico. Por ello, vamos a emplear una técnica conocida como "*One-Hot Encoding*" que nos permitirá codificar las características categóricas como matrices numéricas.

El problema que supone tener tantas variables categóricas es que muchos modelos de aprendizaje automático no pueden aprender con ellas. En la mayoría de los casos, requieren que tanto las variables de entrada como las de salida sean numéricas (en general, se trata más bien de una limitación de la aplicación en *Machine Learning* de estos algoritmos más que de limitaciones de los propios algoritmos) (Brownlee, 2017).

Esto supone que los datos categóricos tienen que transformarse a formas numéricas, lo cual conlleva dos pasos:

1. *Integer Encoding* : en primer lugar, cada valor categórico tiene que tener un valor entero asignado. En algunos casos, este primer paso sería suficiente, ya que los valores enteros están ordenados de forma natural y algunos algoritmos son capaces de entender y establecer esas relaciones.
2. *One-Hot Encoding* : para variables categóricas donde no existe una relación ordinal, es necesario realizar este paso tras el *integer encoding*. De hecho, no hacerlo puede causar que nuestro modelo interprete que existe una relación ordinal entre nuestras variables que termine por modificar los resultados y afectando seriamente la productividad y efectividad de nuestro estudio.
   En estos casos, lo que debemos hacer es añadir una variable binaria para cada valor entero único.

3.2.3.1 Creación de nuevas variables

Utilizando la información que encontramos en las diferentes tablas originales, procedemos a crear nuevas variables (el código puede encontrarse en los anexos).

En primer lugar, y con el objetivo de asegurar la calidad de los datos, vamos a eliminar de nuestro *dataset* las empresas que no cumplan con todos estos requisitos.

- Número concreto de empleados
- Datos de capitalización de mercado
- Ingresos y beneficios netos
- Año de fundación

El razonamiento para eliminar estos valores es que se trata de variables críticas para poder interpretar la información presente en el *dataset* y obtener conclusiones válidas. Desafortunadamente, algunos de ellos aparecen con valores nulos, guiones o símbolos que no son interpretables.

Además, los valores monetarios vienen representados en formato *string*, acompañados de una B o una M que indica el tamaño de los ingresos o capitalizaciones. Por ello, debemos asegurarnos de tener esos datos formateados correctamente creando un diccionario que ayude a nuestro modelo a interpretar estos datos correctamente.

En diccionario consistirá en la transformación de la letra “M” (millones) a “1.000.000”, mientras que la letra “B” (miles de millones) pasará a ser “1.000.000.000”. A continuación, guardaremos los valores filtrados en una tabla diferente.

Por otro lado, también hay que tener en cuenta ciertos conceptos financieros que nos ayuden a entender si los datos con los que contamos son o no correctos o si debemos realizar alguna modificación más.

Los precios de las acciones se cotizan únicamente en días laborables, lo que supone que a lo largo de un año no tendremos 365 días, sino únicamente 262. Por ello, hemos definido cada mes como un período de 22 días. Esto supondrá la creación de una variable nueva por cada mes.

Finalmente, procedemos a seleccionar las columnas que utilizaremos para calcular nuestras predicciones, que serán (sin incluir las variables que acabamos de crear, que se añadirán a la siguiente lista):

Nombre de la columna	Información
DaysBetterThenSP	Días en los que el stock superó al índice S&P500 en rendimiento
daysProfit	Días en los que el stock mostró un rendimiento positivo
Year	Año de salida a bolsa
Month	Mes de salida a bolsa
Day	Día de salida a bolsa
dayOfWeek	Día de la semana de salida a bolsa (del 1 al 5)
LastSale	Número de acciones vendidas en la IPO
MarketCap	Capitalización bursátil tras la IPO
Sector	Sector de actividad
Revenue	Ingresos en el primer año tras la IPO
netIncome	Beneficio neto en el primer año tras la IPO
employees	Número de empleados de la empresa
USACompany	Si la empresa es de EEUU (Yes / No)
YearFounded	Año en el que se fundó la empresa
Profitable	Si la IPO consiguió beneficios sobre el precio de salida
Homerun	Si el stock mostró un rendimiento excepcional en su primer año
Safe	Si la IPO fue de tipo Safe

Figura 11 – Columnas seleccionadas para realizar predicciones

3.2.4 Desglose del *Dataset*

Las empresas con las que se trabaja en el *dataset* están repartidas por todo el mundo, aunque podemos observar algunas diásporas claras que nos permiten entender qué zonas geográficas cuentan con una mayor presencia de *startups*:

Distribución geográfica de *startups* en Estados Unidos

Figura 12 – Distribución geográfica de startups en EEUU

En el mapa anterior podemos observar como la gran mayoría de las *startups* de nuestro dataset se encuentran en las dos costas de EEUU, con especial frecuencia en el noreste y en el suroeste.

Estas serán las dos regiones que estudiaremos más en profundidad en nuestro estudio, la tratarse de los dos principales focos de financiación de *startups* en el mundo. Sin embargo, merece la pena también entender cómo está distribuido este mercado en Europa:

Distribución geográfica de *startups* en Europa

Figura 13 – Representación de compañías exitosas en Europa

En este caso nos encontramos una distribución más equilibrada, con dos nichos más fuertes en el Reino Unido y Alemania, aunque el resto del continente cuenta con una presencia bastante más equitativa.

Otro de los datos principales que arroja el *dataset* es la industria a la que pertenece cada empresa. De nuevo podemos ver una concentración en determinados nichos donde la presencia de *startups* es mucho más dominante que en otras.

Aproximadamente un 90% de las empresas a nivel global están relacionadas con la tecnología. La gran dominante es la industria de la biotecnología, que cuenta con 29.9% del total de las empresas de nuestro *dataset*. Las dos siguientes categorías están relacionadas entre sí, compartiendo además porcentajes similares: Hardware con un 11.3% y Software con un 7.69%.

Distribución sectorial de *startups* a nivel mundial

Sector	%
Biotecnología	30.7
Hardware	11.6
Software	7.91
Semiconductores	6.51
Otros	5.12
Network Hosting	4.65

Web	4.19
Servicios empresariales	3.72
RRPP	3.26
CleanTech	2.79

Figura 14 – Distribución sectorial de startups a nivel mundial

El *dataset* con el que estamos trabajando nos permite identificar los nombres de las empresas, así como información más específica sobre cada una de ellas. Uno de los parámetros más importantes de las *startups* es su tamaño – referente al número de trabajadores.

Según un estudio de la Agencia Federal de Economía de St. Louis, en Estados Unidos, el número medio de trabajadores en las *startups* ha descendido de forma constante en los últimos 25 años, siendo las del sector de la construcción las que cuentan con un mayor número de empleados, mientras que las tecnológicas presentan menos de la mitad de trabajadores (Hong, Werner, 2020)

A continuación, hemos filtrado las empresas por su número de empleados y, salvo algunos *outliers* que corresponden a compañías con presencia en todo el mundo, podemos observar que la gran mayoría de los casos a estudiar son *startups* con menos de 50 empleados. Como podemos ver, Google es la empresa con más empleados en el momento de toma de los datos (100 444), seguida por Ebay, Facebook y Amazon.

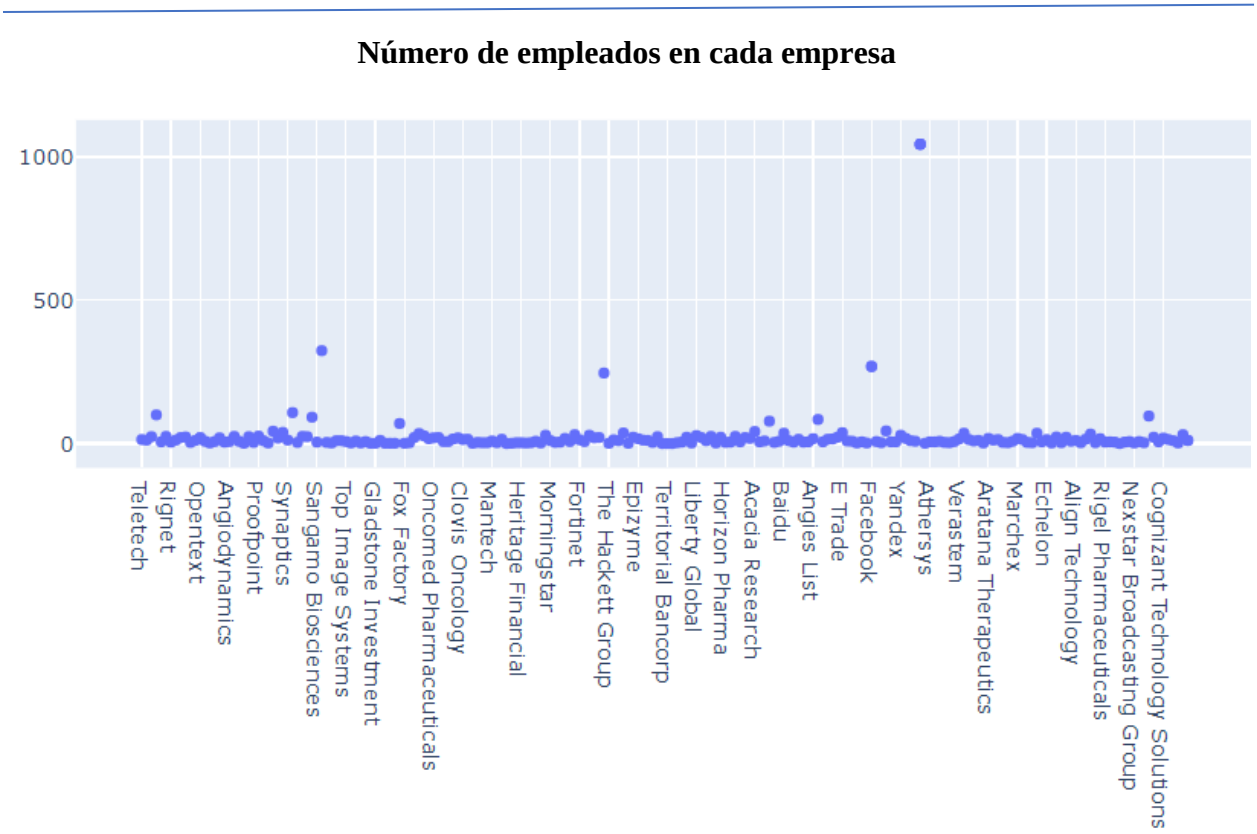


Figura 15 – Número de empleados en cada empresa

Según Bernhard Schroeder, director del Lavin Entrepeneurship Center de la Universidad de San Diego, en EEUU, cuando una *startup* supera los 100-150 empleados, es inevitable que sufra un cambio en su cultura. Al fundar una empresa pequeña, la comunicación entre los empleados, la colaboración y el intercambio de ideas se produce diariamente.

Sin embargo, continúa Schroeder, incrementar el tamaño supone añadir capas de *managers* que con total seguridad ralentizarán la toma de decisiones, dificultarán las relaciones entre los directores y los empleados más *juniors* y las "politización" de la oficina comenzará a cambiar la forma en la que los empleados se relacionan entre ellos.

Robin Dunbar, profesor de psicología en la Universidad de Oxford, argumenta que las personas somos capaces de mantener relaciones personales con hasta 150 personas, un número al que tratan de adherirse numerosas *startups* en sus comienzos (Schroeder, 2020 >>>https://www.forbes.com/sites/bernhardschroeder/2020/02/07/what-is-the-magic-number-of-employees-in-a-startup-or-company-in-one-location-before-tribalism-begins-to-break-it-apart/?sh=3d0bdad428c3).

Otro dato relevante que se ha tratado en nuestro *dataset* es el tipo de financiación que ha recibido cada *startup*. A continuación, podemos ver la cantidad en millones de dólares que las empresas han obtenido de cada tipo de financiación.

Distribución de financiación en *Venture Capital*

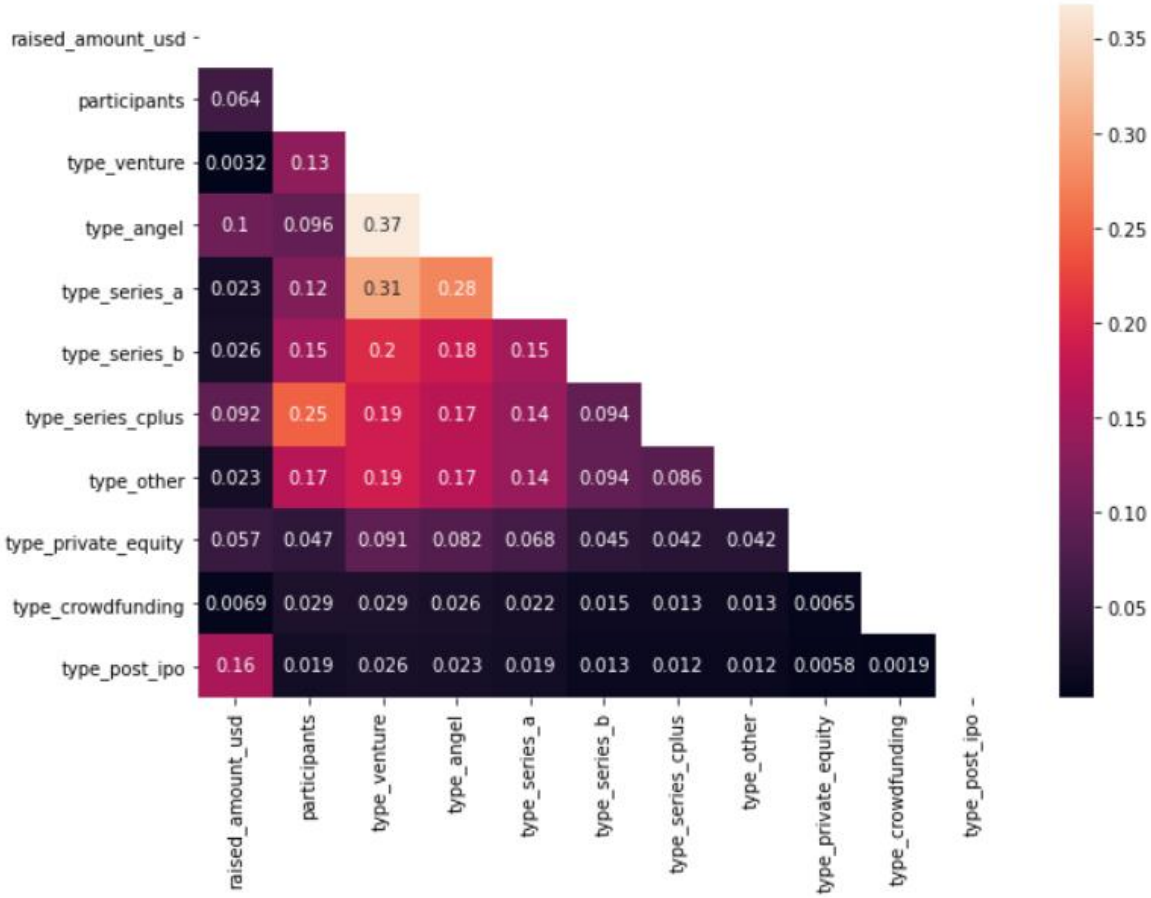


Figura 16 – Distribución de financiación en Venture Capital

Los más utilizados son los fondos de *Venture Capital* y *Angel Investing*, más presentes en las primeras rondas de financiación (*Series A*). En términos generales, el *Venture Capital* acelera el crecimiento 1,5 años cuando lo comparamos con aquellas empresas que no han tenido este tipo de financiación.

Impacto de *Venture Capital* en sectores tecnológicos y no tecnológicos

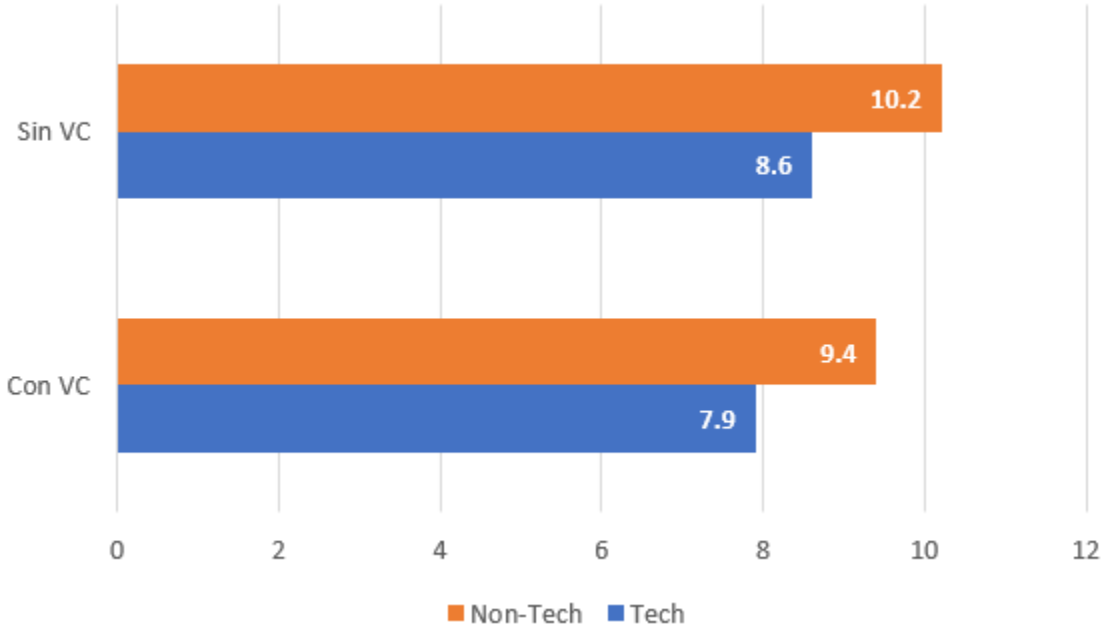


Figura 17 – Impacto del Venture Capital en sectores tecnológicos y no tecnológicos

Además, un 65.4% de las empresas consideradas exitosas se han beneficiado de *Venture Capital* en su proceso de crecimiento. Una ronda de financiación de *Venture Capital* suele componerse de una primera fase llamada *Seed*, seguida de la Ronda A, B, C y así sucesivamente. En este estudio se ha considerado hasta la Ronda D. De las compañías exitosas que se han analizado, un 31% ha tenido Ronda A, lo cual ha permitido a la empresa acelerar su desarrollo y conseguir más recursos en menor tiempo. Sólo un 10.4% de las *startups* exitosas han conseguido llegar a la Ronda C, lo cual prueba la dificultad de desarrollar un negocio en sus fases iniciales.

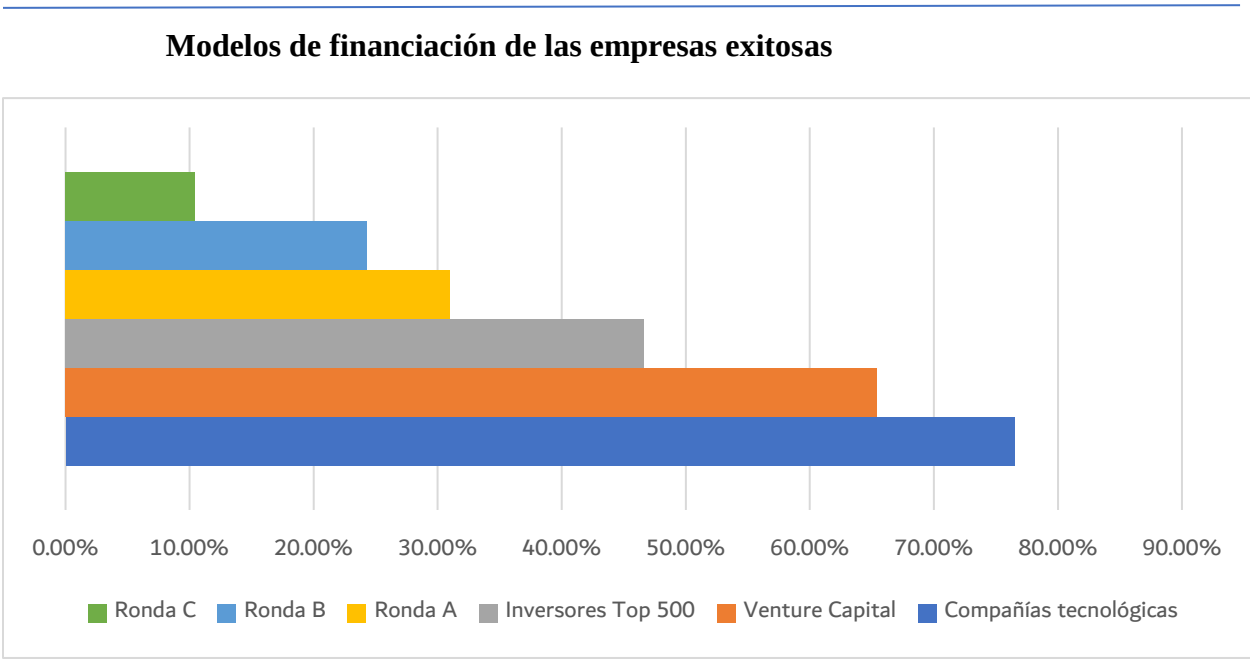


Figura 18 – Modelos de financiación de las empresas exitosas

Una ronda de financiación de *Venture Capital* suele componerse de una primera fase llamada *Seed*, seguida de la Ronda A, B, C y así sucesivamente. En este estudio se ha considerado hasta la Ronda D. De las compañías exitosas que se han analizado, un 31% ha tenido Ronda A, lo cual ha permitido a la empresa acelerar su desarrollo y conseguir más recursos en menor tiempo. Sólo un 10.4% de las *startups* exitosas han conseguido llegar a la Ronda C, lo cual prueba la dificultad de desarrollar un negocio en sus fases iniciales.

Por lo que se ha podido observar tras un primer análisis exploratorio y manipulación de los datos es que fundar una *startup* en California aumenta las posibilidades de éxito y disminuye el tiempo necesario para conseguirlo. Se trata del estado más tecnológico del mundo, tanto histórica como culturalmente (Weller, 2016). Además, de las diez empresas más exitosas del estudio, ocho son tecnológicas, lo cual demuestra que las probabilidades de triunfar en esa industria son más elevadas.

Las empresas tecnológicas son también más exitosas de media que las de otros sectores, necesitando entre uno y dos años menos que el resto de las empresas para ser consideradas exitosas. Por último, conseguir más financiación supone aumentar las probabilidades de éxito, aunque tras superar los 20 millones el porcentaje de *startups* exitosas se mantiene prácticamente constante.

3.3. PREPARACIÓN DEL EXPERIMENTO

3.3.1 Métricas a evaluar

Como métricas principales, los algoritmos de clasificación que vamos a utilizar son *True Positive Rate (TPR)* y *False Positive Rate (FPR)*. No sólo se trata de los métodos más utilizados, sino que se consideran estándar en este tipo de proyectos. Además, nos permitirán comparar resultados entre los diferentes algoritmos para el mismo problema.

True Positive Rate (TPR = TP / (TP+FN)) o *Recall* se define como el porcentaje de compañías exitosas que se han identificado correctamente como tal. Por otro lado, *False Positive Rate (FPR = FP / (FP+TN))* puede definirse como el porcentaje de compañías no exitosas que se han clasificado como exitosas. Esta metodología muestra claramente la capacidad predictiva de cualquier aspecto de nuestro estudio, clasificando a las empresas como exitosas con los *features* y metodología utilizados.

Métricas a evaluar

Matriz de Confusión	0 (Predicción Negativo)	1 (Predicción Positivo)
0 (Negativo Real)	Negativo Real (TN): empresas clasificadas como no exitosas que no son exitosas	Falso Positivo (FP): empresas clasificadas como exitosas que no lo son
1 (Positivo Real)	Falso Negativo (FN): empresas clasificadas como no exitosas que sí lo son	Positivo Real (TP): empresas clasificadas como exitosas que lo son

Figura 19 – Métricas a evaluar

Por lo tanto, la precisión de nuestro modelo podrá predecirse como el porcentaje de compañías correctamente clasificadas como exitosas. A pesar de no ser la única capaz de arrojar resultados convincentes, nos puede ayudar a entender cuál de nuestros modelos es el más exacto:

Precisión = (TP + TN) / (TP + FP + TN + FN)

3.3.2 Problemas con el *Dataset* y soluciones aplicadas

3.3.2.1 Escasez de variables

El primer problema con el que nos encontramos es la escasez de datos que ofrece la *database* de CrunchBase. A pesar de que los autores aseguran que el mantenimiento era complicado debido a la poca antigüedad de la plataforma, han pasado ya varios años y el problema sigue persistiendo. Debido a su naturaleza *open-source*, cualquiera puede crear nuevas compañías y editar los datos

de las demás sin ningún tipo de control. Esto, junto a su creciente popularidad, ha hecho que aparezcan bastantes perfiles de *startups* incompletos.

Por otro lado, parece lógico asegurar que se crean muchas más *startups* en la vida real de las que se añaden a la base de datos. Tras el proceso de limpieza y preprocesamiento de datos, incluyendo la transformación, el nivel de escasez era de un 75%, lo cual es alarmantemente elevado.

A pesar de la capacidad de los algoritmos de aprendizaje automático para resolver este problema, hemos decidido implementar una solución en dos fases:

- Creación de *features* binarios que puedan compensar los valores faltantes. Su función principal es la de mostrar si alguno de los campos con los que trabajamos tiene un valor alternativo. Por ejemplo, a la hora de averiguar la localización de ciertas empresas, se ha añadido un campo denominado "USACompany", al que se le han otorgado dos clases: "Yes" o "No". De ésta manera, se introduce la posibilidad de que las *startups* sean de otra localización y se permite filtrar y manipular esos datos con mayor facilidad.
- Discretización de los valores en cuatro grupos con frecuencias iguales. Así, los valores faltantes que se corrijan con "0" no tendrán un peso excesivo en las nuevas variables. Por ejemplo, el *feature* "funding_rounds" tenía valores del 1 al 24 y se ha discretizado a cuatro valores internos: [-inf- 1.5], [1.5- 2.5], [2.5- 3.5] y [3.5- inf].
- Además, y a pesar de que supone más tiempo de entrenamiento, se han transformado todos los *features* en binarios. Si bien no existe una ventaja teórica clara, los resultados de los modelos eran más elevados. Siguiendo el razonamiento de Liaw & Wiener (2002), en un *random forest*, cada nodo se divide con el mejor subgrupo de predictores escogidos de ese grupo al azar. Esto resultaría en que los árboles podrían formarse en base a un valor más alto o bajo de cada *feature* y no al valor del *feature* como tal. Este efecto minimiza la correlación entre *features* al permitir más combinaciones entre cada árbol.

Con estas transformaciones llegamos a un *dataset* único, donde los algoritmos se pueden entrenar desde su propia construcción, comparando los resultados más rápidamente.

3.3.2.1 Clases desiguales

Otro problema con el que hemos dado a la hora de crear un buen modelo predictivo es el gran desequilibrio entre clases de empresas exitosas y no exitosas. Tras el preprocesamiento, contamos con 1 043 empresas con rendimiento negativo, mientras que tan solo 301 tienen buen rendimiento y 283 presentan rendimiento positivo.

La mayor parte de los algoritmos de aprendizaje automático funcionan mejor cuando el numero de observaciones en cada clase es parejo, ya que de lo contrario pueden tener a interpretar a la clase menos representada como la opuesta. En el presente estudio, si todas las observaciones fueran etiquetadas como negativas (no exitosas) se obtendría un 83% de acierto, lo cual todavía sería un porcentaje más alto que en la mayoría de los modelos publicados para predecir el éxito de las compañías (Wei et al., 2009; Xiang et al., 2012).

3.3.3 Algoritmos de *Machine Learning*

En el presente proyecto tenemos como objetivo completar una tarea de clasificación binaria: "1" para las empresas exitosas y "0" para las que no lo sean. Se trata de un tipo de aprendizaje supervisado en el que las categorías de *output* son predefinidas. Es importante elegir no solo el algoritmo que mejor se adapte al problema sino uno que también se pueda ajustar a las características del *dataset*.

Algunos algoritmos asumen ciertas características sobre el *dataset* y tienen objetivos diferentes. Al elegir los algoritmos con los que trabajaremos en este proyecto, nuestra intención es no sólo que se ajusten a la naturaleza del *dataset*, sino que también sean fáciles de implementar y de entender.

3.3.3.1 Regresión Logística

A pesar de tratarse de un problema de clasificación (el *output* es una categoría) y no de regresión (donde el *output sería una variable continua*), la regresión logística es una técnica donde la variable dependiente o *target* suele tomar valores como "0" ó "1", "exitosa" o "no exitosa". Esto es precisamente lo que permite aplicar esta técnica a problemas de *Machine Learning*.

$$\text{odds}(Y = 1) = \frac{p}{1-p}, \text{where } 0 < p < 1$$

Sobre el logaritmo de probabilidad (*odds*), la regresión linear asume una relación lineal entre la variable dependiente y la independiente, lo cual describe en la siguiente fórmula:

$$\begin{aligned} \text{logit}(Y) &= \alpha + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_K X_K \\ &= \alpha + \boldsymbol{\beta}^T \boldsymbol{X} \end{aligned}$$

La relación descrita en la fórmula anterior maximiza las probabilidades de las observaciones. Asumiendo X_i observaciones e y respuestas, donde i = 1...N, la probabilidad viene determinada por la siguiente expresión:

$$l(\alpha, \boldsymbol{\beta}) = \sum_{i=1}^{N} \{y_i \log(p) + (1 - y_i) \log(1 - p)\}$$

donde *p* es la función de α,β y X. Una vez que se ha determinado el valor máximo de la probabilidad, la estimación de la probabilidad de *p* puede calcularse mediante el cálculo opuesto:

$$\hat{p} = \frac{e^{\hat{\alpha}+\hat{\boldsymbol{\beta}}^T \boldsymbol{X}}}{1+e^{\hat{\alpha}+\hat{\boldsymbol{\beta}}^T \boldsymbol{X}}} = \frac{1}{1+e^{-(\hat{\alpha}+\hat{\boldsymbol{\beta}}^T \boldsymbol{X})}}$$

La función logística tiene forma de *S*, de manera que las probabilidades más cercanas 0/1 están asociadas a sus respuestas 0/1 correspondientes. El punto exacto, conocido como "punto de decisión", se usa para predecir las probabilidades de la fórmula anterior ya que 0 ó 1 normalmente depende del contexto del ejercicio, lo que suele derivar en que se elija 0.5.

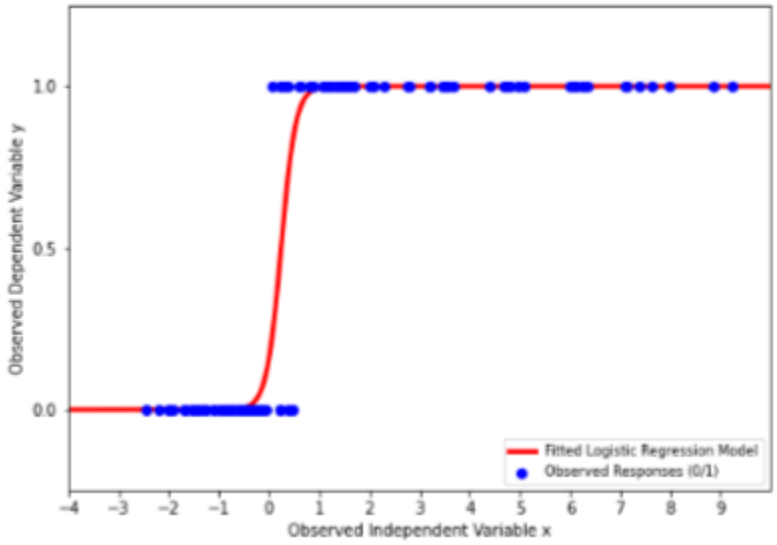


Figura 20– Curva ROC

La relación entre *Y* y las variables independientes puede extenderse para cubrir interacciones curvilíneas entre las variables X_1, X_2, ...X_k, incrementando la complejidad del modelo y de las relaciones existentes entre los datos. Sin embargo, para este estudio vamos a considerar una relación lineal.

La regresión logística sigue la misma lógica que la lineal: multiplica cada *input* por un coeficiente, los suma y añade una constante a cada *feature*, asumiendo que hay una división (clasificación) que los diferencia (Hosmer & Lemeshow, 2000). Si por ejemplo estamos intentando predecir si lloverá

o no mañana, la operación a realizar sería dividir las probabilidades de que llueva por las probabilidades de que no.

En el campo del aprendizaje automático, la regresión logística es una de las técnicas mas utilizadas por ser un algoritmo rápido y simple de implementar. Debido a su baja varianza es menos probable caer en un problema de *overfitting*, lo cual lo hace ideal para problemas binarios con una separación clara de clases. Además de eso, su capacidad de no asumir distribuciones entre clases es también otra característica a tener en cuenta.

La desventaja más importante que tiene este modelo es la limitación de resultados que se podrían aplicar en determinados contextos, sobre todo aquellos con correlaciones más elevadas (Howbert, 2012). En general, la regresión logística es ideal para problemas con dimensiones de datos elevadas.

Al tratarse de uno de los algoritmos más comunes en *Machine Learning*, se pueden encontrar ejemplos de regresión logística en prácticamente cualquier área de estudio: desde detección de fraude a captación de clientes potenciales en campañas de marketing.

3.3.3.2 Support Vector Machines

Manning et al. definen este algoritmo de forma muy concisa: "los SVMs son dos clasificadores (…) Por un lado un clasificador con gran margen: su objetivo es definir un punto de separación entre dos clases que están separadas la una de la otra en el set de entrenamiento (posiblemente descontando algunos puntos como *outliers* o ruido)".

Se trata de un algoritmo diferente a otros utilizados en el aprendizaje automático. Puede lidiar con *datasets* amplios, aunque se adapta mejor a proyectos con números más reducidos de *features*, ya que precisa más tiempo de entrenamiento y consume mucha memoria (Manning, Raghavan, & Schütze, 2009; Statnikov, 2011). Es muy común en proyectos de clasificación de textos donde se dan más problemas de alta dimensionalidad (Statnikov, 2011).

Un problema con este tipo de algoritmos es interpretar los resultados. Por ejemplo, en el *dataset* con el que hemos trabajado no se pueden representar todas las marcas de cada empresa como una función paramétrica simple de cada *feature*. Los pesos relativos de cada uno no son constantes, y eso hace que la contribución de cada uno varíe. Utilizando un *kernel* gaussiano, cada empresa tiene sus propios pesos relativos en función de la diferencia entre el valor de sus *features* y los de los vectores de apoyo del set de entrenamiento (Auria & Moro, 2008).

Además, los SVMs maximizan el margen y asume la distancia entre los diferentes puntos del *dataset* (Boser et al., 1992), lo que hace que sea más fácilmente adaptable a problemas con *features* numéricos en lugar de categóricos, ya que el concepto de "distancia" no se puede aplicar en el caso de las categorías.

3.3.3.3 Random Forest

Los *Random Forest* son colecciones de árboles de decisión. A diferencia de los dos algoritmos expuestos anteriormente, los *Random Forest* no necesitan *features* lineales. Podría definirse de forma simple como diferentes grupos en un clasificador de árbol múltiple (Breiman, 2001). Sin embargo, al no ser posible construir varios árboles con los mismos datos y obtener resultados distintos, se tendrá que introducir la aleatoriedad: cada árbol se construirá con filas diferentes, y se tomarán como ejemplos repeticiones de la original (*bagging*); cada árbol (y en algunos casos cada rama de decisión) se construirá utilizando un subgrupo de columnas elegido al azar.

Según Any Liaw y Matthew Wiener, este algoritmo puede resumirse en los siguiente pasos: (1) elegimos *n* ejemplos al azar de los datos originales; (2) generamos un árbol de decisión para cada grupo con la siguiente modificación: en cada nodo, en lugar de elegir entre el mejor subgrupo de predictores, elegiremos al azar un número *m* de predictores y de ahí obtendremos las mejores variables. Este subgrupo de *m* predictores minimizará la correlación entre los clasificadores (Gislason, Benediktsson, & Sveinsson, 2006); (3) predeciremos los nuevos datos agregando las predicciones de los *n* árboles (Liaw & Wiener, 2002).

Una de las ventajas más importantes que tiene este algoritmo es la forma en la que gestiona la compensación entre sesgo y varianza (*Bias-Variance Tradeoff*), uno de los principales problemas del *Deep Learning*: aunque su sesgo es el mismo que el de los árboles de decisión, su varianza disminuye al incrementar el número de árboles, lo cual también hace que se reduzcan las posibilidades de *overfitting*.

Además, es altamente eficiente con *datasets* de gran tamaño, manejando miles de *inputs* diferentes sin necesidad de eliminar ninguno, proporcionando estimaciones de qué variables son más importantes para la clasificación, procesando datos faltantes e incluso manteniendo una gran exactitud cuando las proporciones son más grandes (Breiman, 2001).

La principal desventaja que tienen los *Random Forest* comparados con Árboles de Decisión normales es que su interpretación es más compleja, ya que no es tan sencillo observar la relación entre la variable dependiente y el *dataset* que se ha creado. Un *Random Forest* tiene que ser una herramienta predictiva y descriptiva. Es fácil entender la importancia de los *features*, pero puede no ser suficiente cuando el objetivo del estudio sea entender la relación entre variables dependientes e independientes.

Este algoritmo se ha aplicado en diversas industrias, convirtiéndose en uno de los más populares en *Machine Learning*. En 2007, Cutler et al., mostraron cómo los *Random Forests* tenían más exactitud que ningún otro algoritmo en la predicción de varios escenarios ecológicos al estudiar especias de plantas invasivas en California y Utah, EEUU. Por otro lado, Lariviere y Vandenpoel los aplicaron para entender el comportamiento de los consumidores y cómo mejorar su retención y fidelización, descubriendo que lo que más impacto tiene en las predicciones es el comportamiento del cliente en el pasado (Cutler et al., 2007; Gislason et al., 2006; Lariviere, Vandenpoel, & D, 2005).

3.3.3 Punto de partida

La base de datos de CrunchBase cuenta, a fecha de realización de este trabajo, con un total de 81 219 perfiles de empresas, 107 274 personas, 7 328 organizaciones financieras, 3 955 empresas de servicios, 25 895 rondas de financiación y 6173 adquisiciones. Hay dos tipos de *features* principales:

- Basados en los perfiles de CrunchBase: número de empleados, vida de la compañía (desde fundación), localización, número de oficinas, número de productos, número de proveedores.
- Financieros: número de rondas de financiación, número de adquisiciones de la compañía, número de rondas de capital riesgo que han invertido en la compañía, cifras de inversión total en la compañía.
- Operacionales: número de empleados, experiencia de los fundadores.

El modelo utilizado por Xiang et al. (2012), centrado en las adquisiciones e inversiones en *Venture Capital*, sugiere que como continuación a su estudio se ponga el poco principal en las IPOs como parte de la clasificación positiva (empresas exitosas) y para la creación de *features* que puedan reducir la distancia entre las empresas exitosas y no exitosas.

Dicho estudio será utilizado parcialmente como base para este proyecto ya que comparte parte de los datos (aunque es algo más antiguo y con menos datos). También se compararán los resultados de ambos al tratarse de estudios con el mismo objetivo, lo cual nos hace pensar que la comparación puede resultar de interés.

3.4. RESULTADOS DEL EXPERIMENTO

3.4.1 Evaluación de los algoritmos de aprendizaje

En primer lugar, se han probado diferentes algoritmos para observar la exactitud de cada uno a la hora de realizar predicciones sobre los datos con los que tenemos que trabajar. A pesar de que la exactitud no es la característica más recomendable para decidir entre el uso de uno u otro algoritmo (especialmente si hay diferencias entre clases), en este punto ya se han equilibrado las clases y la exactitud es una métrica rápida y fácil de interpretar.

Debido a la naturaleza del *dataset* se han probado varios algoritmos. Dos de ellos fueron también utilizados por Xiang et al. en su estudio, lo cual nos proporciona una buena plataforma para comparar nuestros resultados.

Utilizando una cros validación con un 25% del conjunto total del *dataset* se han comprobado las exactitudes de los diferentes modelos, obteniendo los siguientes resultados:

- Regresión Logística: 0.928 (0.0015)
- SVC: 0.928 (0.0014)

- Random Forests: 0.931 (0.0029)

Comparación de algoritmos

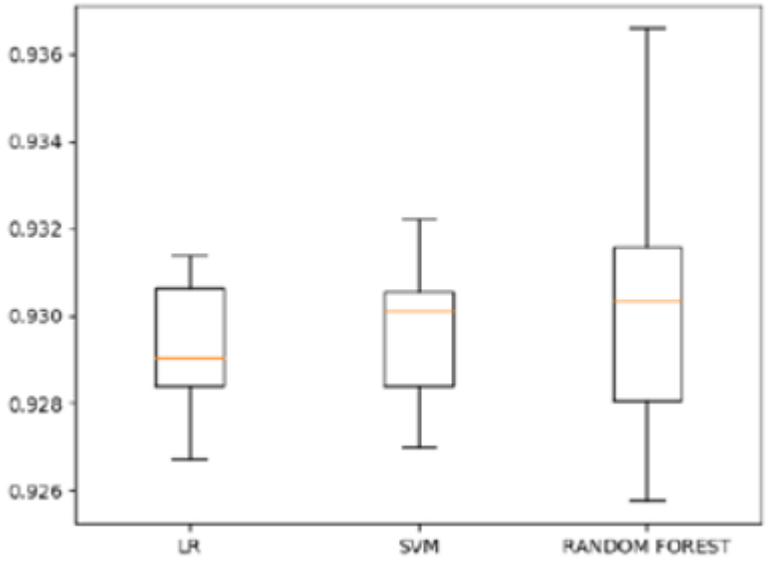


Figura 21– Comparación de algoritmos

Como podemos comprobar, los algoritmos obtuvieron resultados muy similares. *Random Forest* tiene la mayor precisión y varianza de los tres (como se esperaba). La precisión total se mide como la suma de todos los *true positives* más los *true negatives* divididos entre el total de observaciones, por lo que este modelo nos proporciona una buena métrica para evaluar el rendimiento de un algoritmo.

Por otro lado, tanto el SVC como la Regresión Logística obtuvieron también una puntuación alta, principalmente gracias a la correlación entre los *features* en el *dataset*, lo cual genera una separación lineal entre el espacio de los *features*. Sin embargo, la precisión por sí misma no refleja el principal objeto de estudio en este proyecto: los positivos verdaderos totales (*Recall*).

3.4.2 Elección del algoritmo de aprendizaje

Tras realizar un análisis en profundidad las predicciones de los algoritmos y usar una división de 70% para el set de entrenamiento y 30% para el *test*, hemos generado los siguientes resultados:

		Precisión	Recall	f1-score	Support
Regresión Logística	0	0.891	0.966	0.931	21748
	1	0.962	0.89	0.924	21257
	Media/Total	0.93	0.928	0.928	43005
SVM	0	0.896	0.971	0.932	21748
	1	0.968	0.885	0.925	21257
	Media/Total	0.932	0.929	0.928	43005
Random Forest	0	0.94	0.924	0.933	21748
	1	0.924	0.941	0.932	21257
	Media/Total	0.933	0.932	0.932	43005

Figura 22– Resultados de los algoritmos de aprendizaje

Random Forest ha sido finalmente el algoritmo elegido por ser que tiene una mayor *True Positive Rate (Recall)* y por el equilibrio que muestra entre la precisión y el *recall*. Por otro lado, el *False Positive Rate* de 7.8% no es una mejora respecto a los estudios previos que hemos consultado, aunque sigue dando cierto espacio para interpretaciones, que se explorarán más adelante. Tanto la Regresión Lineal como *SVM* tienen resultados parecidos, tal y como se esperaba, ya que sólo se diferencian en la función de pérdida (*loss function*). Como norma general, se espera que *SVM* presente mejores métricas que la regresión logística (Pedregosa, 2013).

Resultado final de *Random Forests*

Algoritmo	Precisión	TPR	FPR
Random Forests	92.4%	94.1%	7.8%

Figura 23– Resultado final de Random Forests

Al comparar los diferentes algoritmos con los que se ha trabajado, podemos observar métricas de AUC bastante positivas, como se muestra a continuación. El área por debajo de la curva ROC es considerablemente amplia, lo cual sugiere diferencias significantes entre empresas exitosas y no exitosas, calculadas asumiendo que la probabilidad de que parejas aleatorias de observaciones hayan sido etiquetadas correctamente. Aun así, *Random Forest* sigue obteniendo un pequeña ventaja con respecto a los otros dos algoritmos:

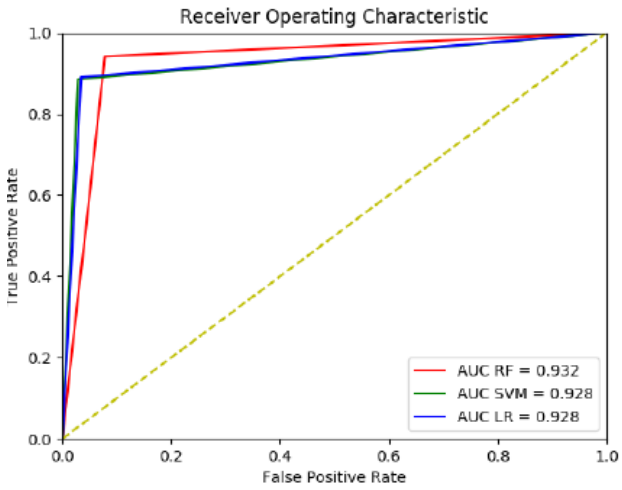


Figura 24– Curva ROC de TPR y FPR

Este algoritmo provee un modelo más robusto que los demás. La Regresión Logística es una buena solución cuando se trata de data más predecible y binaria, por lo que se adaptaría mejor a este proyecto. Por otro lado, SVC también da resultados excelentes gracias a su capacidad para clasificar las observaciones en *features* multidimensionales, aunque puede incrementar el tiempo de entrenamiento del modelo.

Como se ha confirmado anteriormente, se ha elegido *Random Forests* como modelo ya que puede descubrir dependencias más complejas, como por ejemplo *features* que pueden tener más importancia para unas empresas que para otras. A pesar de que el *dataset* final que se utilizará en este trabajo no tiene desequilibrios entre los datos ni valores faltantes, el *Random Forest* puede también gestionar datos categóricos con *outliers* y datos faltantes.

3.4.3 Importancia de los *features*

Los *Random Forest* son uno de los algoritmos más populares de aprendizaje automático gracias a su precisión relativa, su solidez y facilidad de uso. Por otro lado, pueden no ser sencillos de interpretar cuando se comparan con árboles de decisiones simples que pueden recuperarse a través de *feature importance*: reducción media de la impuridad y la precisión.

Cada nodo de un árbol de decisión es una condición de un *feature* individual, diseñado para dividir el *dataset* en dos de forma que las respuestas similares puedan clasificarse dentro del mismo grupo de datos. La medida en la que se basa la condición óptima se conoce como impureza.

Para la clasificación, normalmente se utiliza información obtenida por entropía, mientras que para problemas de regresión suele ser la varianza. Por lo tanto, cuando se entrena un árbol de decisión, se puede computar cuánto de cada *feature* ha visto reducida su impureza. En caso de un bosque,

la reducción de cada *feature* individual puede agregarse y obtener la media, clasificando así los *features* según sus reducciones.

A continuación, se muestran los 11 *features* más importante basados en la reducción media de su impureza:

Feature	Reducción de Impureza Media
hasVC=‘(0.5-inf)’	0.33
investors_per_round=‘(-inf-1.5]’	0.3
investors_per_round=‘(3.5-inf)’	0.29
usa_state_code=CA	0.29
funding_rounds=‘(-inf-1.5]’	0.29
funding_rounds=‘(1.5-2.5]’	0.29
funding_rounds=‘(3.5-inf)’	0.29
investors_per_round=‘(2.5-3.5]’	0.28
investors_per_round=‘(1.5-2.5]’	0.28
funding_total_usd=‘(3500002.5-16248967.5]’	0.28
employee_count_ordinal=‘(2.5-inf)’	0.28

Figura 25– Mejora de los features

De los once *features*, sólo hay dos que no estén relacionados con las inversiones. Este hecho demuestra que los más importante en *Venture Capital* es tener información acerca de las inversiones y los inversores con los que pueden contar las empresas.

3.4.4 Evaluación por Estado y categoría

Se realiza una nueva exploración generando un nuevo modelo para cada estado en el *dataset*. Los resultados son más exactos para los estados más representados, como California y Otros ya que cuentan con más observaciones en la tarea de aprendizaje. La categoría “Otros” ha obtenido resultados en línea con los del modelo general, con un 94% de TPR y un 8% de FPR, mientras que California muestra un FPR del 10%.

	TPR	FPR	Instances
Other	94%	8%	21 051
CA	94%	10%	14 766
NY	90%	10%	4 996
TX	96%	19%	3 282
MA	96%	23%	3 212

Figura 26– TPR y FPR por estado

Este análisis ilustra a la perfección la importancia del set de aprendizaje para poder predecir con precisión las probabilidades de éxito de las empresas en los resultados finales. Por ejemplo, Massachusetts mostró un 23% de falsos positivos y un 96% de verdaderos positivos.

Categoría	TPR	FPR	Instances
Otros (non-Tech)	96	4	15350
Finanzas	94	10	6499
Media (Non-Tech)	94	9	15845
Sanidad	91	13	5797
Ciencia y educación (Tech)	88	13	6818
Sanidad (Tech)	87	8	8703
Comercio	86	4	6556
Hardware (Tech)	86	14	6897
Software (Tech)	86	16	22679
Media (Tech)	85	18	21523
Finanzas (Tech)	82	16	5322
Comercio (Tech)	76	12	5096
Otros (Tech)	61	4	16263
Media	86	11	143348

Figura 27– TPR y FPR por categoría

Para conseguir un número de observaciones más equilibrado, hemos realizado una nueva transformación que reduce todo el *dataset* a 13 categorías. Se ha dividido en 10 tecnológicas y 3 no tecnológicas, siguiendo la distribución del *dataset* original, que contenía un 70% de empresas tecnológicas.

La categoría “Other(Tech)” contiene empresas tecnológicas en el sector de las comunicaciones, sector gubernamental, manufactura, movilidad, sector inmobiliario, seguridad, y energía. Por otro lado, las mismas categorías clasificadas como no tecnológicas han obtenido la puntuación más alta (96% TPR), lo que podríamos interpretar como una evidencia de mayor varianza en las empresas tecnológicas, lo cual dificultaría su clasificación por parte del modelo. Además, las mismas categorías no tecnológicas tienen menos positivos totales, lo cual debido a la linealidad de nuestros datos hace que sean más fáciles de clasificar como positivos.

Respecto a la tasa de falsos positivos, podemos afirmar que las empresas sobre las que es más elevada tienen un mayor valor para ser clasificadas como exitosas según nuestro modelo, y podría considerarse como un dato de interés para los analistas financieros a la hora de evaluar empresas donde invertir su patrimonio.

4. CONCLUSIONES

El objetivo principal de este proyecto es el de generar un modelo que ayude a clasificar a empresas y *startups* como exitosas. Mediante la construcción de un clasificador binario se etiquetará a las compañías que se estudian como exitosas o no exitosas con un *True Positive Rate (TPR)* de 94.1% y un *False Positive Rate (FPR)* de 7.8%, con un 92.2% de precisión, por lo que se asume que el objetivo se ha logrado.

Se trata de uno de los porcentajes más altos de precisión utilizando la base de datos de CrunchBase. El modelo puede predecir no sólo qué empresas del *dataset* serán exitosas (TPR, *Recall*) sino cuáles de las que son etiquetadas como tal lo son realmente (Precisión).

El algoritmo que se ha elegido es *Random Forests*, que nos permite interpretar de forma eficaz y sencilla el modelo, obteniendo resultados satisfactorios. Para elegir este algoritmo se ha comparado su eficacia con la de otros dos modelos: *Support Vector Machines (SVM)* y Regresión Logística. Ambos fueron probados por su potencial a la hora de adaptarse al tamaño y la naturaleza de nuestro *dataset*, del cual esperábamos relaciones lineares.

Durante el experimento se ha implementado una transformación de los *features*, discretizándolos todos en cuatro subgrupos con un máximo de cuatro intervalos, otorgándoles valores del 1 al 4. Esta transformación, aunque teóricamente no aporta ninguna ventaja significativa, redujo la tasa de falsos positivos (*FPR*) un 1%, mientras que la ratio de positivos totales (*TPR*) disminuyó un 0.5%. Esta transformación permite que el modelo pueda trabajar con valores específicos de cada *feature*, lo que facilita el aprendizaje de información más específica a través de un número más elevado de combinaciones entre *features*.

***Features* binarios VS *Features* no binarios**

Algoritmo	TPR	FPR	AUC
RF *Features* no binarios	93.60%	8.80%	92.50%
RF *Features* binarios	94.10%	7.80%	93.20%

Figura 28– Features binarios VS Features no binarios

Para poder obtener resultados comparables con los estudios previos, este estudio contiene un modelo general que contempla tanto la categoría de la empresa como su localización en EEUU (los estudios previos se centraron en EEUU). Se trata de un nuevo enfoque que otorga una base geográfica al éxito de las diferentes *startups* en los estados más representativos de EEUU en este sector (California, Nueva York, Massachusetts, Texas, etc. La categoría "Otros" ha obtenido un 94% de TPR y un 8% de FPR, mientras que California ha conseguido un 94% TPR con un 10%

de FPR. Otros estados como Massachusetts y Texas han obtenido peores resultados debido al menor número de empresas en esas localizaciones.

Los estudios anteriores por parte de Xiang et al. afrontaron este problema centrándose en las predicciones por categoría, consiguiendo entre un 44% y un 79.8% con *Bayesian Networks.* También es importante distinguir que las puntuaciones más altas se obtuvieron en aquellas categorías que tenían más observaciones, mientras que las del presente estudio no han seguido siempre ese patrón. El modelo ha conseguido *TPRs* de entre 61% y 96%. El área por encima de la curva ROC también es más elevada que el de su estudio (93.2% frente a 88%).

Nuestro estudio se ha beneficiado de un *dataset* más grande donde algunas categorías han sido esenciales para la obtención de estos resultados, junto con la creación de observaciones artificiales para compensar por la falta de datos y la mala calidad de algunos de ellos. El *FPR* es también más elevado (7.8% frente a 2.2% en categorías tecnológicas), si bien es cierto que las interpretaciones dependen del contexto. Si consideramos todas las categorías, el *FPR* varía entre un 0% y un 3%, mientras que el del presente estudio se encuentra entre un 4% y un 18%.

También podemos comparar el presente estudio con el modelo desarrollado por Liang y Daphne Yuan ("*Investors are Social Animals*"), quienes intentaron crear un modelo que explicara cómo las relaciones sociales podrían impactar el proceso de toma de decisiones de los inversores. En primer lugar, nuestro estudio ha obtenido un 94.1% de *TPR* frente al 89.6% que Liang y Daphne Yuan obtuvieron mediante el algoritmo *SVM*. Además, nuestro *FPR* es de 7.8% frente a su 33.4% para *SVM*, lo cual es considerablemente mejor. Liang y Daphne Yuan también reportan un *FPR* de 5% con su modelo de Naïve Bayes, aunque con un *TPR* de 54.8%. Si observamos por categorías, el *TPR* que publicaron se encuentra entre el 51% y el 91% para Naïve Bayes, mientras que este estudio ha obtenido unos valores de entre 61% y 96% (Liang & Daphne Yuan, 2012).

El modelo general con todos los *features* categóricos agregados obtuvo un *TPR* mejor que cualquier otro modelo entrenado con la base de datos de CrunchBase, lo que demuestra su gran utilidad en la predicción del éxito de empresas.

El *FPR* del modelo general (7.8%) debería ser el foco principal del análisis. Las empresas que han sido categorizadas dentro de este porcentaje podrían en realidad ser casos de compañías con suficientes medios financieros como para pertenecer al grupo de empresas exitosas, lo cual es una información de enorme valor para los inversores.

No sería descabellado asumir que estas empresas se acercan bastante al concepto de exitoso que se desarrolla en el presente estudio, ya sea a través de una salida a bolsa (*IPO*) o una adquisición (*M&A*). De hecho, el modelo predice el éxito de una empresa categorizándola como exitosa, aunque en realidad aun no lo sea.

El presente estudio aporta métricas ligeramente más exactas que los proyectos anteriores gracias a la aplicación de algunas de las recomendaciones (como añadir los datos de salidas a bolsa, creación de nuevos *features*) y a la aplicación de algoritmos como *Random Forests*, más adaptables a éste

dataset. También nos permite predecir el potencial de empresas que son etiquetadas erróneamente en la categoría de falsos positivos, lo cual demuestra el gran potencial de conseguir el éxito tal y como se define en este trabajo.

5. RECOMENDACIONES PARA FUTUROS ESTUDIOS

La exploración de las empresas catalogadas como falsos positivos en este estudio podría interpretarse como una muestra de gran potencial para un futuro éxito empresarial, lo cual puede ser de gran ayuda para analistas financieros en el futuro.

Se sugiere la aplicación de diferentes algoritmos de aprendizaje automático sobre los mismos datos, así como visualizaciones sencillas que puedan aportar resultados similares a los presentados en este estudio.

Además, al tratarse de una API sencilla, la base de datos de CrunchBase podría convertirse en una herramienta de operaciones con la que fondos de inversión, inversores particulares y otros agentes puedan trabajar en este espacio. Los datos facilitados por la plataforma ofrecen la posibilidad real de obtener información sobre las empresas y las inversiones mediante modelos y segmentaciones de *Machine Learning*.

6. REFERENCIAS

10 Million Self-Driving Cars Will Be On The Road By 2020 - Business Insider. (2016). Retrieved February 1, 2017, from http://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6

Aggarwal, C. C. (2015). Data Mining: The Textbook. Springer International Publishing. http://doi.org/10.1007/978-3-319-14142-8

Akerlof, G. A., Yellen, J. L., & Katz, M. L. (1970). The market for Lemons: Quality uncertainty and the market and the market mechanism. The Quarterly Journal of Economics. http://doi.org/488500

Alam, A., & Khan, S. (2014). STRATEGIC MANAGEMENT: MANAGING MERGERS & ACQUISITIONS. International Journal of BRIC Business Research, 3(1).

Ali-Yrkkö, J., Hyytinen, A., & Pajarinen, M. (2005). Does patenting increase the probability of being acquired? Evidence from cross-border and domestic acquisitions. Applied Financial Economics, 15(14), 1007–1017. http://doi.org/10.1080/09603100500186978

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. Journal of Finance, 23(4), 589–609. http://doi.org/10.2307/2978933

Artificial Intelligence and Machine Learning: Top 100 Influencers and Brands. (2016). Retrieved January 31, 2017, from http://www.onalytica.com/blog/posts/artificial-intelligence-machine-learning-top-100-influencers-and-brands/

Auria, L., & Moro, R. A. (2008). Support Vector Machines (SVM) as a Technique for Solvency Analysis, (August). Retrieved from www.diw.de

Berry, M. J. a., & Linoff, G. S. (2004). Data mining techniques: for marketing, sales, and customer relationship management. Portal.Acm.Org. Retrieved from http://portal.acm.org/citation.cfm?id=983642

Beyer, D. (2015). The Future of Machine Intelligence, Perspectives from Leading Practitioners.

Blank, S. G. (2006). The Four Steps to the Epiphany.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory - COLT ’92 (pp. 144–152). http://doi.org/10.1145/130385.130401

Breiman, L. (1996). Bagging predictors. Machine Learning. Retrieved from http://www.springerlink.com/index/L4780124W2874025.pdf

Breiman, L. (2001). RANDOM FORESTS. Retrieved from http://www.math.univ-toulouse.fr/~agarivie/Telecom/apprentissage/articles/randomforest2001.pdf

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). Classification and regression trees.

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. Retrieved from https://www.jair.org/media/953/live-953-2037-jair.pdf

Christopher Clifton. (2009). Data Mining.

Chye Koh, H., & Tan, G. (2011). Data Mining Applications in Healthcare. Journal of Healthcare Information Management —, 19(2). Retrieved from https://pdfs.semanticscholar.org/433a/57b382c528c78395e317d9fee008fb8ed9de.pdf

Crunchbase, Inc. Company data. http://www. crunchbase.com/.

Customer Stories | Crunchbase Data Solutions. (2017). https://about.crunchbase.com/customers/

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY. Ecology, 88(11), 2783–2792. http://doi.org/10.1890/07-0539.1

Farrar, C. R., & Worden, K. (2012). Structural Health Monitoring: A Machine Learning Perspective - Charles R. Farrar, Keith Worden - Google Livros. Wiley. Retrieved from https://books.google.pt/books?hl=pt-PT&lr=&id=2w_sp6lersUC&oi=fnd&pg=PP11&dq=machine+learning+health&ots=E1vmyBFsvo&sig=Mavuhd4Aq5DqiafMeP8nhHmyPOg&redir_esc=y#v=onepage&q=machine learning health&f=false

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3), 37. http://doi.org/10.1609/aimag.v17i3.1230

Fortune 1000 Companies List for 2016 - Geolounge. (n.d.). Retrieved May 23, 2017, from https://www.geolounge.com/fortune-1000-companies-list-2016/

Geier, B. (2015). What Did We Learn From the Dotcom Stock Bubble of 2000? Retrieved from http://time.com/3741681/2000-dotcom-stock-bust/

Gislason, P., Benediktsson, J., & Sveinsson, J. (2006). Random forests for land cover classification. Pattern Recognition Letters. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167865505002242

Gompers, Paul A; Gornall, Will; Kaplan, Steven N and Strebulaev, A (2020). How do venture capitalists make decisions? Journal of Financial Economics, 135(1):169–190,.

Graham, P. (2012). Startup = Growth. Retrieved February 1, 2017, from http://www.paulgraham.com/growth.html

Gugler, K., & Konrad, K. a. (2002). Merger Target Selection and Financial Structure, (2001), 1–25.

Guo, B., Lou, Y., & Pérez-Castrillo, D. (2015). Investment, Duration, and Exit Strategies for Corporate and Independent Venture Capital-backed Start-ups.

Han, J., & Kamber, M. (2006). Data Mining: Concepts and Techniques. Soft Computing (Vol. 54). http://doi.org/10.1007/978-3-642-19721-5

Hanley, J. A., & McNeil, B. J. (1982). The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. Radiology, (143). Retrieved from http://pubs.rsna.org/doi/pdf/10.1148/radiology.143.1.7063747

Hermann, B. L., Gauthier, J., Holtschke, D., Bermann, R. D., & Marmer, M. (2015). The Global Startup Ecosystem Ranking 2015. The Startup Ecosystem Report Series, (August), 1–156.

Hill, K. (2012). How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did. Retrieved October 17, 2017, from https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/#5b90f7766686

Ho, T. K. (1995). Random Decision Forests. Retrieved from http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf

Kantardzic, M. (2003). Data mining: concepts, models, methods, and algorithms.

Kim, E. (2015). Fastest startups to $1 billion valuation - Business Insider. Retrieved August 21, 2017, from http://www.businessinsider.com/fastest-startups-to-1-billion-valuation-2015-8/#1-slack-is-the-fastest-growing-enterprise-software-ever-11111114

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Leaning. INTERNATIONAL JOURNAL OF COMPUTER SCIENCE, I.

Kudyba, S. (2014). Big Data, Mining, and Analytics: Components of Strategic Decision Making - Stephan Kudyba - Google Books. Retrieved from https://books.google.pt/books?id=nuoxAwAAQBAJ&pg=PA287&lpg=PA287&dq=1.+Kincade,+K.+(1998).+Data+mining:+digging+for+healthcare+gold.+Insurance+%26+Technology,+23(2),+IM2-IM7.&source=bl&ots=6U-iwqjGtd&sig=U6DxjqzMopUQOGdl-yCyO9BIuJs&hl=en&sa=X&ved=0ahUKEwi

Lariviere, B., Vandenpoel, & D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications, 29(2), 472–484. http://doi.org/10.1016/j.eswa.2005.04.043

Lennon, M. (2014). CrunchBase Data Export Now Includes International Startups, Investors -. Retrieved October 20, 2017, from https://about.crunchbase.com/blog/crunchbase-data-export-now-includes-international-startups-investors/

Li, D., & Liu, J. (2010). The Life Cycle of Initial Public Offering Companies: A Panel Analysis of Chinese Listed Companies.

Liang, E., & Daphne Yuan, S.-T. (2012). Investors Are Social Animals: Predicting Investor Behavior using Social Network Features via Supervised Learning Approach.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News. Retrieved from https://www.researchgate.net/profile/Andy_Liaw/publication/228451484_Classification_and_Regression_by_RandomForest/links/53fb24cc0cf20a45497047ab/Classification-and-Regression-by-RandomForest.pdf

Loff, J. (2016). Using factorization machine to predict ratings from reviews text alone.

Machiraju, H. . (2003). Mergers, Acquisitions and Takeovers.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). An Introduction to Information Retrieval. Cambridge University Press. Retrieved from https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf

Marita Makinen, B., Haber, D., & Raymundo of Lowenstein Sandler, A. P. (2014). Acqui-Hires for Growth: Planning for Success. Lowenstein Sandler PC. Retrieved from https://www.lowenstein.com/files/Publication/6118b183-d40e-4a5e-8b95-58236c063a10/Presentation/PublicationAttachment/ea0af508-0319-4e3a-86dd-6452b1b6f15d/AcquiHires for Growth.pdf

Marr, B. (2016). The Top 10 AI And Machine Learning Use Cases Everyone Should Know About. Retrieved October 16, 2017, from https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/#2c5c482b94c9

Meador, A. L., Church, P. H., & Rayburn, L. G. (1996). Development of Prediction Models for Horizontal and Vertical Mergers. Journal of Financial and Strategic Decisions, 9(1).

Mitchell, T. M. (2006). The Discipline of Machine Learning.

Neal, R. W. (2014). WhatsApp Investors Make Billions From Facebook Acquisition: Sequoia Capital Sees 50x Return On $1.3 Billion Investment. Retrieved August 21, 2017, from http://www.ibtimes.com/whatsapp-investors-make-billions-facebook-acquisition-sequoia-capital-sees-50x-return-13-billion

NGUYEN, T. (2015). ETA Phone Home: How Uber Engineers an Efficient Route - Uber Engineering Blog. Retrieved February 1, 2017, from https://eng.uber.com/engineering-an-efficient-route/

Onalytica. (2016). ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING.

Osborne, J. W. (2002). Notes on the Use of Data Transformation. - Practical Assessment, Research & Evaluation, 8(6).

Pedregosa, F. (2013). Loss Functions for Ordinal regression. Retrieved July 25, 2017, from http://fa.bianp.net/blog/2013/loss-functions-for-ordinal-regression/

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. http://doi.org/10.1186/2047-2501-2-3

Ragothaman, S., Naik, B., & Ramakrishnan, K. (2003). Predicting corporate acquisitions: An application of uncertain reasoning using rule induction. Information Systems Frontiers, 5(4), 401–412. http://doi.org/10.1023/B:ISFI.0000005653.53641.b3

Ravisankar, P., Ravi, V., Raghava Rao, G., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. Decision Support Systems. http://doi.org/10.1016/j.dss.2010.11.006

Ries, E. (2011). The Lean Startup. Working Paper, 1–28. http://doi.org/23

Rogers, R. (2016). MERGERS & ACQUISITIONS REVIEW MERGERS & ACQUISITIONS REVIEW Fairness Opinion Rankings.

Rowley, J. There Are More VC Funds Than Ever, But Capital Concentrates At The Top. https://news.crunchbase.com/news/ there-are-more-vc-funds-than-ever-but-capital-concentrates-at-the-top/, Last accessed 2020/07.

Samuel, A. L. (1962). Some studies in machine learning using the game of checkers.

Schapire, R. (2008). COS 511: Theoretical Machine Learning (Princeton).

Statnikov, A. (2011). A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods. Retrieved from https://www.google.com/books?hl=en&lr=&id=cxs8DQAAQBAJ&oi=fnd&pg=PP1&dq=+Gentle+Introduction+to+Support+Vector+Machines+in+Biomedicine&ots=TuaSDurkM_&sig=PErWxr2J8SLRFuGRp0iBF5L6YIY

Thiel, P., & Masters, B. (2014). Zero to One. Crown Business. Retrieved from www.crownpublishing.com

United States Census Bureau. Business formation statistics (BFS). https://www.census.gov/programs-surveys/bfs/data/datasets.html

United States Census Bureau. Statistics of U.S. businesses (SUSB). https://www.census.gov/programs-surveys/susb.html.

United States Census Bureau. NES datasets. https://www.census.gov/ programs-surveys/nonemployer-statistics/data/datasets.html.

Wei, C. P., Jiang, Y. S., & Yang, C. S. (2009). Patent analysis for supporting merger and acquisition (M&A) prediction: A data mining approach. Lecture Notes in Business Information Processing, 22 LNBIP, 187–200. http://doi.org/10.1007/978-3-642-01256-3_16

Weller, C. (2016). The 25 most high-tech cities in the world - Business Insider. Retrieved October 20, 2017, from http://www.businessinsider.com/the-most-high-tech-cities-in-the-world-2016-6/#25-washington-dc-1

Witten, Frank, & Eibe. (2000). Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.

Xiang, G., Zheng, Z., Wen, M., Hong, J., & Rose, C. (2012). A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles on TechCrunch. Artificial Intelligence, 607–610.

Yuxian Eugene, L., & Daphne Yuan, S.-T. (2012). Where's the Money? The Social Behavior of Investors in Facebook's Small World. http://doi.org/10.1109/ASONAM.2012.36

Zhang, D., & Zhou, L. (2004). Discovering golden nuggets: Data mining in financial application. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews. http://doi.org/10.1109/TSMCC.2004.829279

7. APÉNDICES

7.1. RANDOM FORESTS – CÓMO FUNCIONAN

Un árbol de decisión es un grupo de reglas que se utiliza para clasificar datos en función de diferentes variables. Toma en cuenta todas las variables del *dataset*, determinando cuáles son las más importantes y después genera un esquema de decisiones en el que se trata todo el *dataset*.

Este esquema – en forma de árbol, de ahí su nombre-, se crea dividiendo los datos en variables y contando sus frecuencias para determinar cuántos hay en cada subgrupo después de la división.

Variables dependientes: empresas exitosas

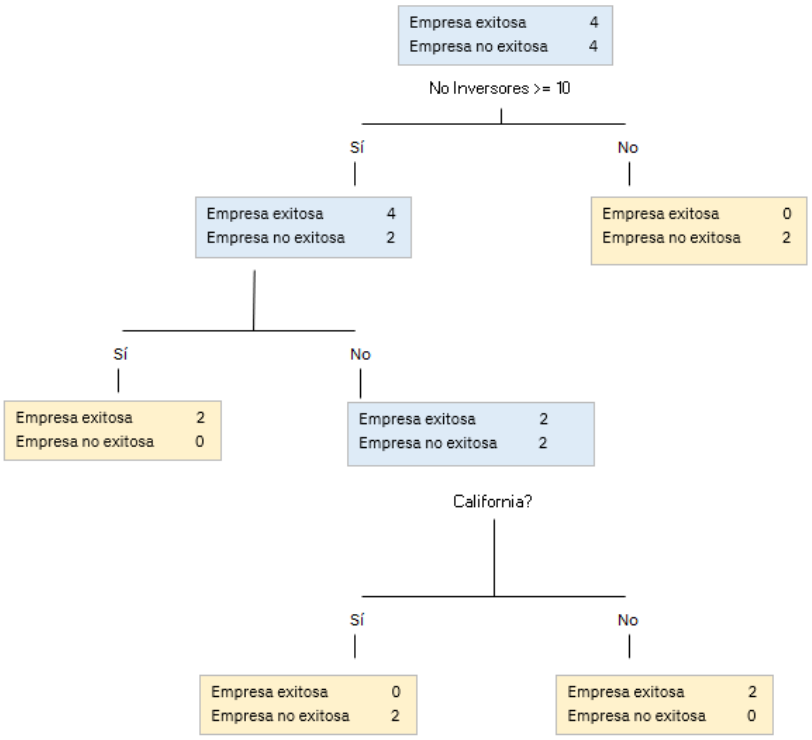


Figura 29– Ejemplo de árbol de decisión

En primer lugar, el modelo comprobaría si la empresa tiene menos de diez inversores. En caso afirmativo, será clasificada como "no exitosa". En caso contrario, el algoritmo comprobará si tiene más de 17 inversores. En ese caso sería catalogada como "exitosa". De tener menos de 17, las últimas particiones comprobarían si la compañía está localizada en California. Con estas tres preguntas, se puede utilizar toda la información respecto a la localización y el número de inversores de la empresa para poder catalogarla como "exitosa" o "no exitosa".

El ejemplo anterior es una representación simplificada de un árbol de decisión, utilizado en este caso para explicar su lógica. En el *dataset* real, habrá empresas exitosas con pocos o un solo inversor. Para este ejemplo, no se ha considerado la definición de los ajustes para detener el algoritmo ni para medir la calidad de su ejecución. Al ejecutarlo se debería optimizar para poder sacar el mayor partido de las predicciones correctas.

Los *Random Forests*, introducidos por Leo Breiman y Adele Cutler en 1995, son clasificadores que combinan múltiples árboles de decisión. Pueden ser aplicados con problemas de clasificación y de regresión, y su porcentaje de acierto y varianza se obtienen junto con los resultados. Tiene dos características principales que podemos dividir en dos niveles:

- Observación: cada uno de los subgrupos se entrena con un número *n* de subgrupos de datos elegido al azar, obteniendo resultados diferentes.
- *Feature*: no todas las columnas se utilizan para el set de entrenamiento. Algunos *features* al azar (*m*) se utilizarán para formar árboles de decisiones. El valor de *m* permanecerá constante durante el proceso de desarrollo del árbol de decisión.

Todos los árboles de decisión son idénticos y se distribuyen por todo el *dataset* de forma idéntica.

Los árboles de decisión tradicionales asumen una representación binaria donde cada árbol se desarrolla lo máximo posible, obteniendo resultados individuales que posteriormente son evaluados y elegidos mediante votos (para problemas de clasificación) o mediante la media ponderada (para problemas de regresión).

Para un *input* determinado, el árbol lo evalúa desde su inicio, creando un modelo CART (*Classification and Regression Trees*) que supone no sólo seleccionar las variables de *input* sino también encontrar puntos de división en los *features* hasta encontrar un árbol que encaje con los datos.

Para seleccionar los *features* que utilizará, el algoritmo emplea un método que busca optimizar la función al máximo, alineando todos los valores posibles con sus puntos de división. Posteriormente, selecciona el valor con un menor coste, tratando de disminuir las impurezas de los nodos superiores.

El siguiente paso a seguir es entender dónde detener el algoritmo, un proceso denominado *Stopping Criterion*. La manera más común de detenerlo es asignando un número de iteraciones a cada nodo de forma que paren automáticamente tras completarlas. Si paran antes del número asignado, se considerará una división incorrecta y se tomará ese nodo como el final.

La complejidad de un árbol de decisión se define según el número de divisiones en cada árbol, ya que cada una de ellas será más fácil de entender y reducirá la probabilidad de *overfitting*. El método más rápido para conseguirlo consiste en trabajar cada variable del árbol por separado y evaluar qué efecto sucede si se elimina. Los nodos sólo se eliminan cuanto los resultados no pueden mejorarse.

8. CÓDIGO RELEVANTE

8.1.1. MODELO GENERAL

```
#Importamos las librerias necesarias para nuestro notebook:
import pandas
import sklearn
from pywFM import FM
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
import numpy as np
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from itertools import cycle
from sklearn import metrics
from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from scipy import interp
from ggplot import *
from statsmodels.compat import pandas as pd
from pandas import *
```

```
predictors = data.drop(['index', 'variation', 'performance'], axis = 1).columns.tolist()
seed = 0

train, test = train_test_split(data, test_size = 0.2, random_state = seed)

x_train = train[predictors]
y_train = train['performance']
x_test = test[predictors]
y_test = test['performance']
```

```
rf_fpr, rf_tpr, rf_threshold = metrics.roc_curve(y_test, rf_predictions)
rf_roc_score = roc_auc_score(y_test, rf_predictions)
rf_roc_auc = metric.auc(rf_fpr, rf_tpr)
#
print "Features segun puntuacion:"
print sorted(zip(map(lambda x: round(x, 3), rForest.feature_importances_), names),
 reverse=True)
# # plot ROC CURVE
import matplotlib.pyplot as plt
plt.title('Receiver Operating Characteristic')
plt.plot(rf_fpr, rf_tpr, 'r', label = 'AUC RF = %0.3f' % rf_roc_auc)
plt.plot(svm_fpr, svm_tpr, 'g', label = 'AUC SVM = %0.3f' % svm_roc_auc)
plt.plot(lr_fpr, lr_tpr, 'b', label = 'AUC LR = %0.3f' % lr_roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'y--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

```
# Sensitivity/Recall/TPR: Cuando el valor real es positivo, con cuanta frecuencia es correcta la prediccion?
sensitivity = TP / float(FN + TP)
print('Sensitivity/TPR: '+ str(sensitivity))
#
lr_roc_score = roc_auc_score(y_test, lr_predictions)
lr_fpr, lr_tpr, lr_threshold = metrics.roc_curve(y_test, lr_predictions)
lr_roc_auc = metric.auc(lr_fpr, lr_tpr)
print('LR ROC AUC SCORE: ' + str(lr_roc_score))
print(classification_report(y_test, lr_predictions))
# SVM
svm = LinearSVC()
svm_model= svm.fit(X_train, y_train)
svm_predictions = svm.predict(X_test) # classify X_test
print(accuracy_score(y_test, svm_predictions))
print(confusion_matrix(y_test, svm_predictions))
print(classification_report(y_test, svm_predictions))
svm_fpr, svm_tpr, svm_threshold = metrics.roc_curve(y_test, svm_predictions)
svm_roc_auc = metric.auc(svm_fpr, svm_tpr)
#
# # Random Forest
rForest = RandomForestClassifier(n_estimators=50, max_features='sqrt')
rf_model = rForest.fit(X_train, y_train)
rf_predictions = rForest.predict(X_test) # classify X_test
print(accuracy_score(y_test, rf_predictions))
print(confusion_matrix(y_test, rf_predictions))
print(classification_report(y_test, rf_predictions))
```

```
# # Realizamos predicciones en el set train/test
# LogisticRegression
print("LogisticRegression")
lr = LogisticRegression()
lr_model = lr.fit(X_train, y_train)
lr_predictions = lr.predict(X_test) # classify X_test
print (accuracy_score(y_test, lr_predictions))
lr_confusion = confusion_matrix(y_test, lr_predictions)
# row, column
TP = lr_confusion[1, 1]
TN = lr_confusion[0, 0]
FP = lr_confusion[0, 1]
FN = lr_confusion[1, 0]
# Classification Error: Overall, how often is the classifier incorrect?
classification_error = (FP + FN) / float(TP + TN + FP + FN)
print('Classification Error: ' + str(classification_error))
```

```
# Listamos los algoritmos
models = []
models.append(('LR', LogisticRegression()))
models.append(('SVM', LinearSVC(dual=False)))
models.append(('RANDOM FOREST', RandomForestClassifier(n_estimators=50)))
# Evaluamos cada algoritmo por separado
results = []
names = []
for name, model in models:
 kfold = model_selection.KFold(n_splits=10, random_state=seed)
 cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
 results.append(cv_results)
 names.append(name)
 msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
 print (msg)
# Comparamos los algoritmos
fig = plt.figure()
fig.suptitle('Comparacion de algoritmos')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

```
ddatos = {'LR': resultados_knn.values(),
        'SVC': resultados_svc.values(),
         'RandomForest': resultado_rforest.values(),
          }

resultados = pd.DataFrame.from_dict(ddatos,
                                    orient = 'index',
                                    columns = resultados_knn.keys())
```

8.1.2. MODELO POR ESTADO/CATEGORÍA

```
# Cargamos las librerías que necesitamos para ésta parte
import pandas
import sklearn
from pywFM import FM
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
```

```
dataset2 = pandas.read_csv(URL, sep=";")
# shape
print(dataset2.shape)
print(dataset2.head(20))

grouped_category = dataset2.groupby('new_category', sort=False).size().order(ascending=False)
print grouped_category

unique_category = dataset2['new_category'].unique().tolist() # Change variable to usa_state_code
#for model per state
print(unique_category)

for val in unique_category:

dataset3 = dataset2.loc[(dataset2['new_category'] == val)] # Change variable to usa_state_code for model per state
```

```
array = dataset3.values
X = array[:, 1:121].tolist()
Y = array[:, 122].tolist()

# Dividimos entre train y test set
validation_size = 0.33 seed = 7 X_train, X_validation,
Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)

# Opciones de test y métricas de evaluación
scoring = 'accuracy'
# Spot Check Algorithms
models = [] models.append(('RANDOM FOREST', RandomForestClassifier(n_estimators=50)))

# Comprobamos cada modelo por separado
print(val)
results = []
names = [] for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state=seed)
    cv_results = model_selection.cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print (msg)

print("RandomForest")
rForest = RandomForestClassifier(n_estimators=50)
rForest.fit(X_train, Y_train)
predictions40 = rForest.predict(X_validation)
print(accuracy_score(Y_validation, predictions40))
print(confusion_matrix(Y_validation, predictions40))
print(classification_report(Y_validation, predictions40))
```

8.2. QUERIES DE SQL

8.2.1. DISCRETIZACIÓN DE NÚMERO DE EMPLEADOS

```sql
UPDATE organizations org
set employee_count_ordinal

CASE WHEN Org.employee_count = '(-)' THEN 0
WHEN Org.employee_count = '(1-10)' THEN 1
WHEN Org.employee_count = '(11-50)' THEN 2
WHEN Org.employee_count = '(51-100)' OR Org.employee_count = '(51-200)' THEN 3
WHEN Org.employee_count = '(101-250)' THEN 4
WHEN Org.employee_count = '(201-500)' OR Org_employee_count = '(251-500)' THEN 5
WHEN Org.employee_count = '(501-1000)' THEN 6
WHEN Org.employee_count = '(1001-5000)' THEN 7
WHEN Org.employee_count = '(5001-10000)' THEN 8
WHEN Org.employee_count = '(10001-100000)' THEN 9

    ELSE 0 end;

SELECT * FROM organizations;

SELECT DISTINCT employee_count_ordinal FROM organizations ORDER BY 1 DESC;
```

8.2.2. MOMENTO DE IPO

```sql
# Calculamos la edad de la empresa en el momento de la IPO


UPDATE organiztions, ipos_ready_to_go
SET organizations.ipo_on = opis_ready_to_go.went_public_on
WHERE organizations-uuid = ipos_ready_to_go.company_uuid;
```

8.2.3. COMPAÑIAS TECH Y CATEGORÍAS FINALES

```sql
# Empresas Tech

UPDATE organizations, organizations_xl_only_categories
SET organizations.isTech = organizations_xl_only_categories.isTech
WHERE organizations.uuid = organizations_xl_only_categories.uuid;

# Categorías

UPDATE organizations, organizations_xl_only_categories
SET organizations.category = organizations_xl_only_categories.category
WHERE organizations.uuid = organizations_xl_only_categories.uuid;

#
```

8.2.4. NÚMERO DE CLIENTES POR EMPRESA

```
UPDATE organizations, customer_count
SET organizations.customer_count = customers_count.customer_count
WHERE customers_count.entity_uuid = organizations.uuid;

```

8.2.4. INVERSORES/RONDA DE INVERSIÓN, INVERSIÓN MEDIA/RONDA

```
# Inversores por cada ronda

UPDATE organizations
INNER JOIN(
    SELECT
        company_uuid,
        avg(investor_count) AS avgInv
    FROM
        funding_rounds_ready
    GROUP BY
        company_uuid
        )
    ON uuid = x.company_uuid
SET organizations.investors_per_round = x.avgInv;

# Media total de inversores por ronda de inversión donde es más de 1 (INFO)

SELECT
    avg(investors_per_round) AS investorRound
FROM
    organizations
WHERE
    investors_per_round > 0
```

```sql
# Inversión media por ronda de financiación

UPDATE organizations
INNER JOIN(
    SELECT
        company_uuid,
        avg(raise_amount_usd) AS avgInvestment
    FROM
        funding_rounds_ready
    GROUP BY
        company_uuid
        )
    ON uuid = x.company_uuid
SET organizations.investment_per_round = x.avgInvestment;

# Inversión total por ronda de financiación

SELECT
    avg(investment_per_round) AS investmentRound
FROM
    organizations
WHERE investment_per_round >0

```

8.2.5. RONDAS A, B, C, D: CANTIDAD, FECHAS

```
# has venture

UPDATE organizations
INNER JOIN (
    SELECT company_uuid, has_seed_angel_venture
    FROM funding_rounds_hasVenture
    ) ON uuid = x.company_uuid
SET organizations.hasVentureCapital = x.has_seed_angel_venture

#ROUND A

UPDATE rganizations
INNER JOIN (
    SELECT company_uuid, funding_round_A
    FROM funding_rounds_ROUND_A
    ) ON uuid = x.company_uuid
SET organizations.roundA = x.funding_round_A;

#date

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, announced_on
    FROM funding_rounds_ROUND_A
    ) ON uuid = x.company_uuid
SET organizations_roundA_date = x.announced_on;

#raised amount

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, raised_amount_usd
    FROM funding_rounds_ROUND_A
    ) ON uuid = x.company_uuid
SET organizations.roundA_raised_amount = x.raised_amount_usd;
```

```sql
#ROUND B

UPDATE rganizations
INNER JOIN (
    SELECT company_uuid, funding_round_B
    FROM funding_rounds_ROUND_B
    ) ON uuid = x.company_uuid
SET organizations.roundB = x.funding_round_B;

#date

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, announced_on
    FROM funding_rounds_ROUND_B
    ) ON uuid = x.company_uuid
SET organizations_roundB_date = x.announced_on;

#raised amount

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, raised_amount_usd
    FROM funding_rounds_ROUND_B
    ) ON uuid = x.company_uuid
SET organizations.roundB_raised_amount = x.raised_amount_usd;
```

```
#ROUND D

UPDATE rganizations
INNER JOIN (
    SELECT company_uuid, funding_round_D
    FROM funding_rounds_ROUND_D
    ) ON uuid = x.company_uuid
SET organizations.roundD = x.funding_round_D;

#date

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, announced_on
    FROM funding_rounds_ROUND_D
    ) ON uuid = x.company_uuid
SET organizations_roundD_date = x.announced_on;

#raised amount

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, raised_amount_usd
    FROM funding_rounds_ROUND_D
    ) ON uuid = x.company_uuid
SET organizations.roundD_raised_amount = x.raised_amount_usd;

```

```
#ROUND C

UPDATE rganizations
INNER JOIN (
    SELECT company_uuid, funding_round_C
    FROM funding_rounds_ROUND_C
    ) ON uuid = x.company_uuid
SET organizations.roundC = x.funding_round_C;

#date

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, announced_on
    FROM funding_rounds_ROUND_C
    ) ON uuid = x.company_uuid
SET organizations_roundC_date = x.announced_on;

#raised amount

UPDATE organizations
INNER JOIN(
    SELECT company_uuid, raised_amount_usd
    FROM funding_rounds_ROUND_C
    ) ON uuid = x.company_uuid
SET organizations.roundC_raised_amount = x.raised_amount_usd;
```

8.2.6. INVERSIONES TOTAL POR EMPRESA

```
UPDATE organizations
INNER JOIN(
    SELECT investor_uuid, count(investor_uuid) as total
    FROM investments_ready_to_go
    GROUP BY investor_uuid
    ) ON uuid = x.investor_uuid
SET organizations.total_investments = x.total
```

9. ANEXOS

9.1. ANÁLISIS EXPLORATORIO

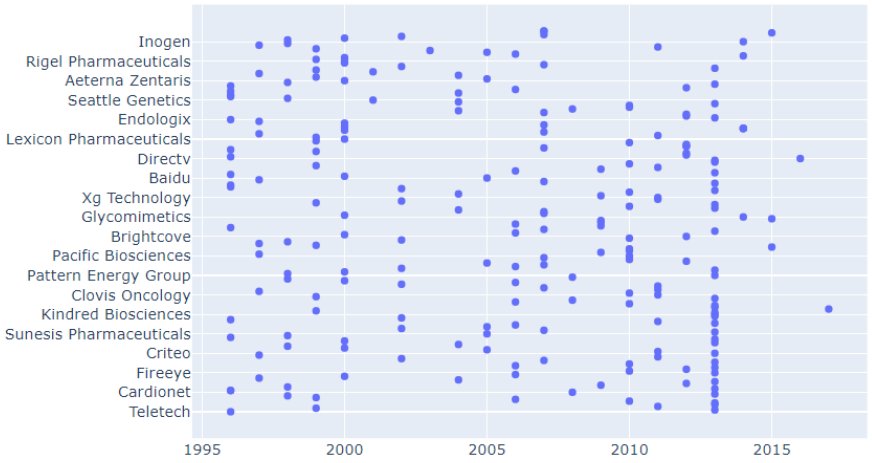


Figura 30– Año de fundación de cada empresa

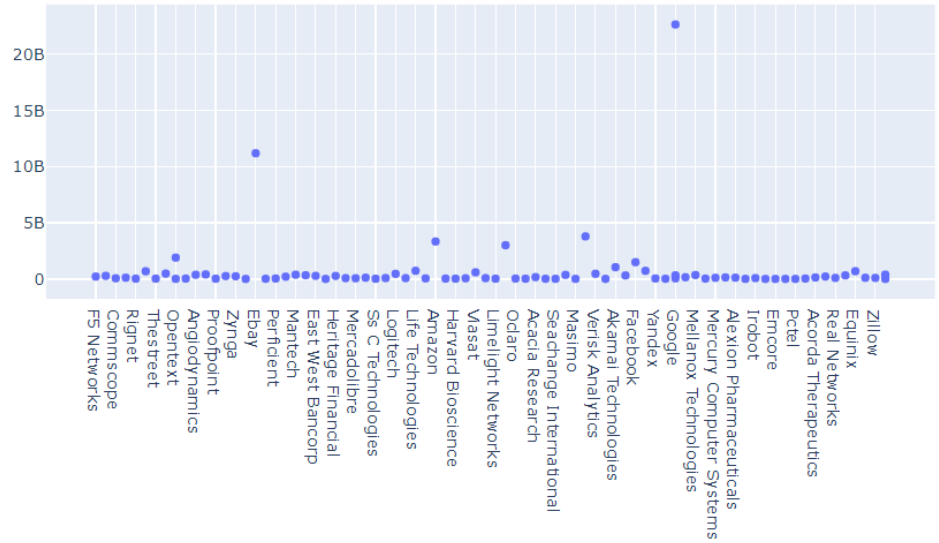


Figura 31– Importe total de adquisiciones por empresa

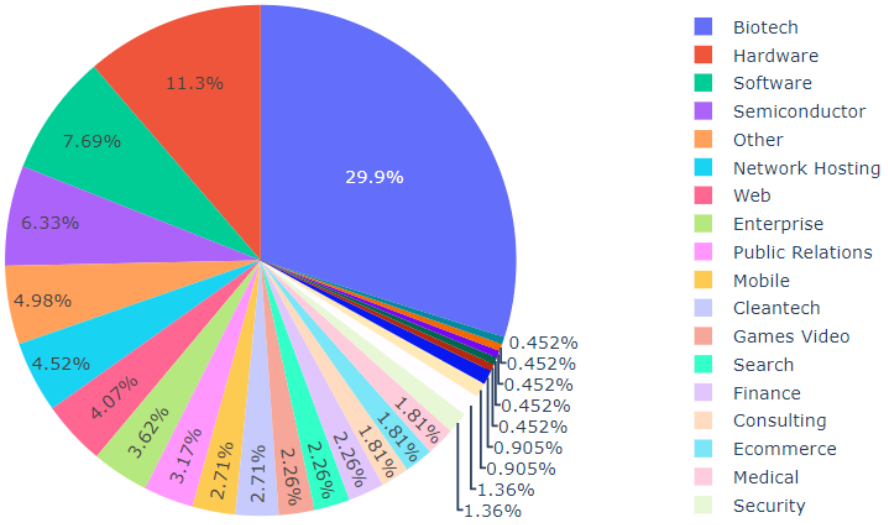


Figura 32 – Porcentaje de empresas por sector

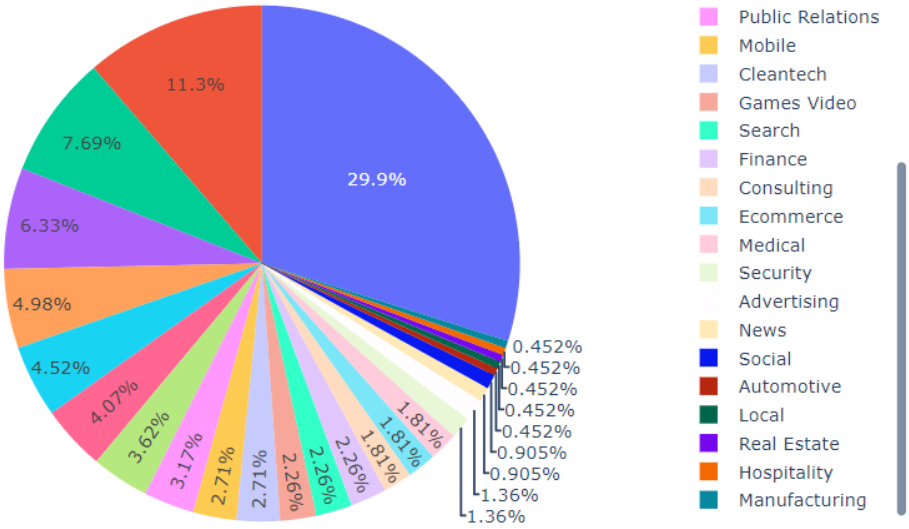


Figura 33 – Porcentaje de empresas por sector II

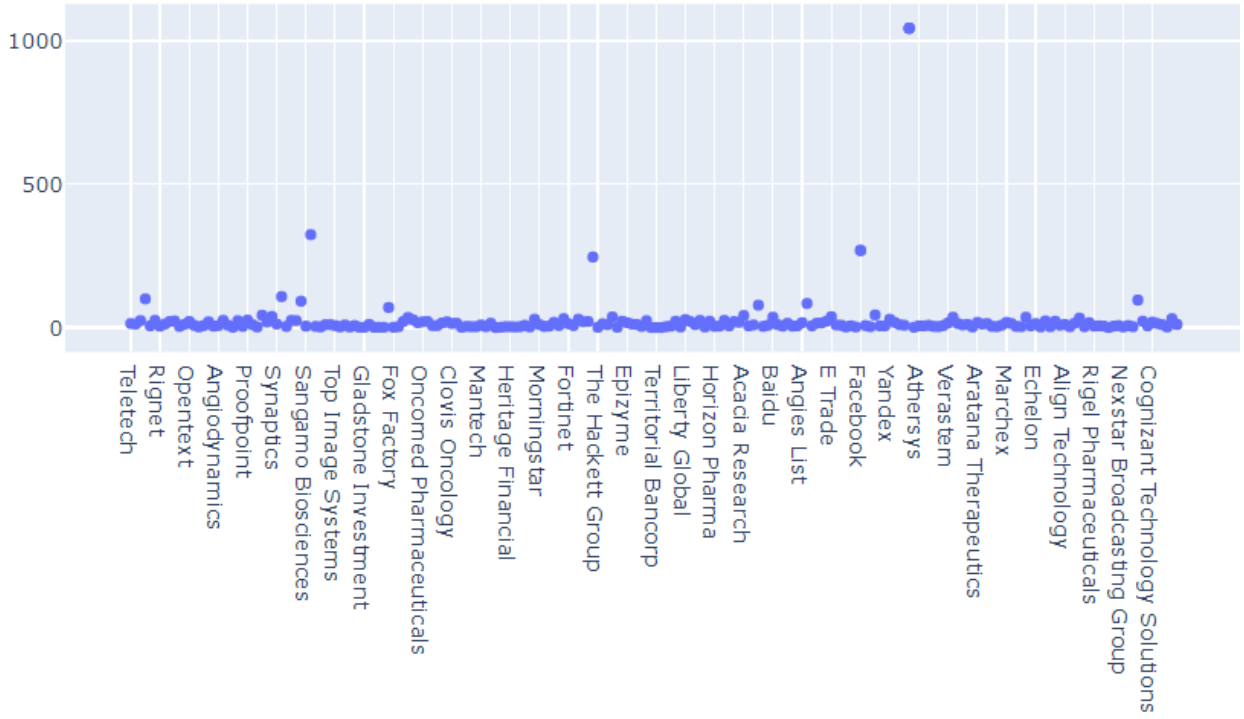


Figura 34 – Número de empleados en cada empresa

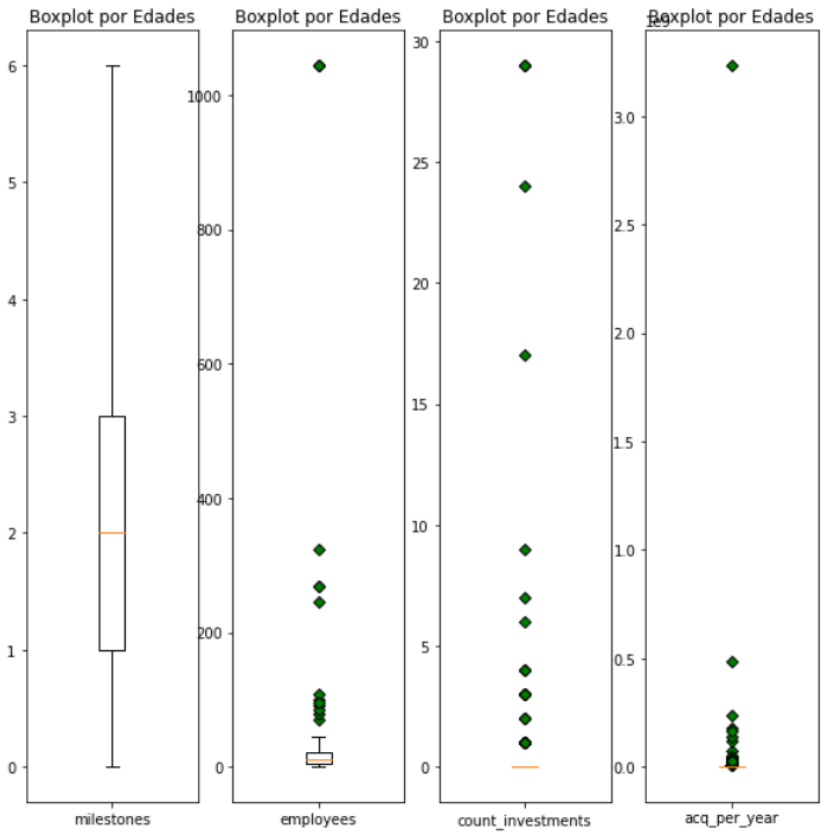


Figura 35 – Boxplot de Outliers

Nombre de la columna	Información
DaysBetterThanSP	Días en los que el stock superó al índice S&P500 en rendimiento
daysProfit	Días en los que el stock mostró un rendimiento positivo
Year	Año de salida a bolsa
Month	Mes de salida a bolsa
Day	Día de salida a bolsa
dayOfWeek	Día de la semana de salida a bolsa (del 1 al 5)

LastSale	Número de acciones vendidas en la IPO
MarketCap	Capitalización bursátil tras la IPO
Sector	Sector de actividad
Revenue	Ingresos en el primer año tras la IPO
netIncome	Beneficio neto en el primer año tras la IPO
employees	Número de empleados de la empresa
USACompany	Si la empresa es de EEUU (Yes / No)
YearFounded	Año en el que se fundó la empresa
Profitable	Si la IPO consiguió beneficios sobre el precio de salida
Homerun	Si el stock mostró un rendimiento excepcional en su primer año
Safe	Si la IPO fue de tipo Safe

Figura 36 – Información sobre las variables

Variable	**Tipo de variable**
DaysBetterThenSP	object
daysProfit	object
Year	object
Month	object
Day	object
dayOfWeek	object
LastSale	object
MarketCap	object
Sector	object
Revenue	float64
netIncome	float64
employees	object
USACompany	object

YearFounded	object
Profitable	object
Homerun	object
Safe	object

Figura 37 – Información sobre las variables

Variable	**Tipo de variable**
DaysBetterThenSP	float64
daysProfit	float64
Year	float64
Month	float64
Day	float64
dayOfWeek	object
LastSale	float64
MarketCap	float64
Sector	object
Revenue	float64
netIncome	float64
employees	float64
USACompany	object
YearFounded	float64
Profitable	float64
Homerun	float64
Safe	float64

Figura 38 – Información sobre las variables

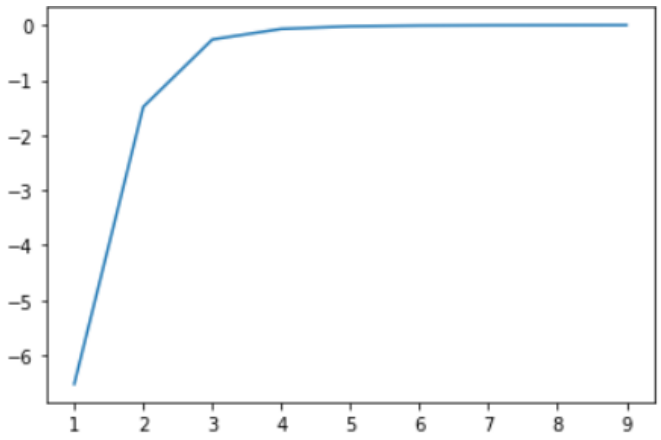


Figura 39 – Optimización del número de clusters

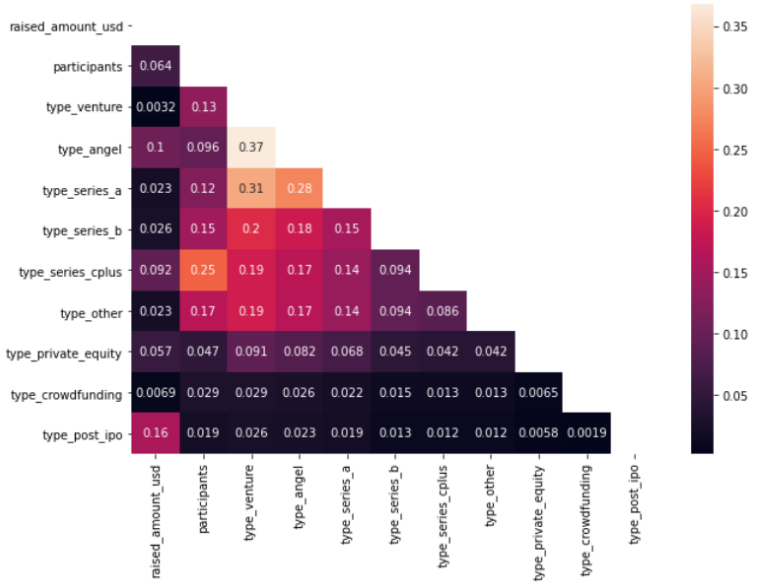


Figura 40 – Distribución de financiación en Venture Capital

USA State Code	Count	%	Successful Companies	Target Ratio
MA	4 609	5%	922	20%
TX	4 967	6%	695	14%
NY	9 926	11%	1 191	12%
CA	27 291	32%	4 912	18%
Other	39 795	46%	5 969	15%
All	86 588	100%	13 689	16%

Figura 41: Estados de EEUU utilizados en el dataset

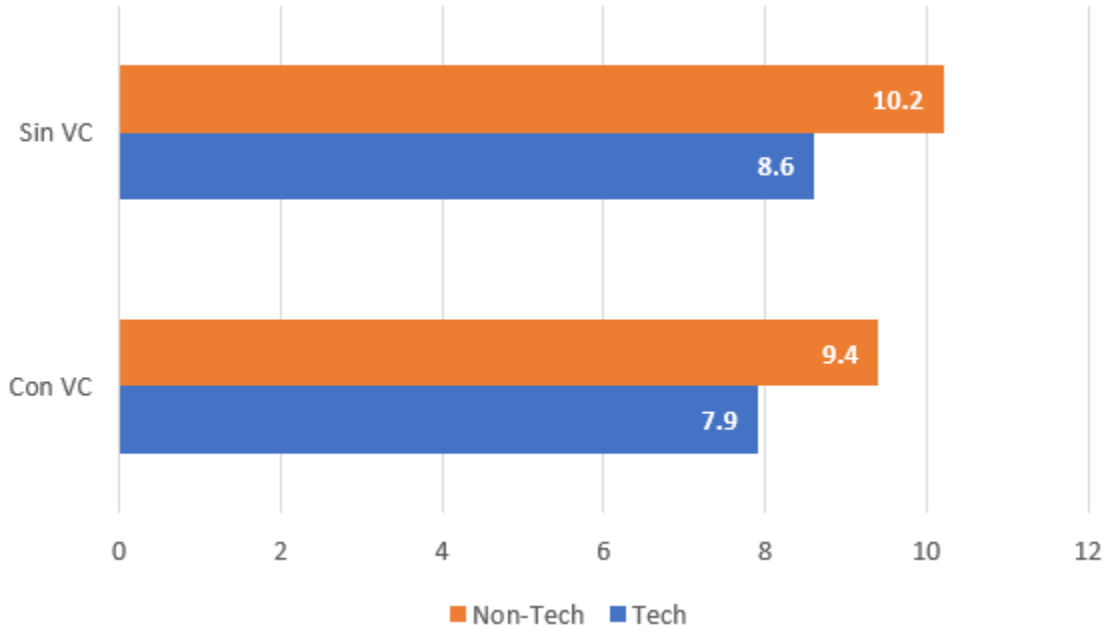


Figura 42: Impacto de Venture Capital en sectores tecnológicos y no tecnológicos

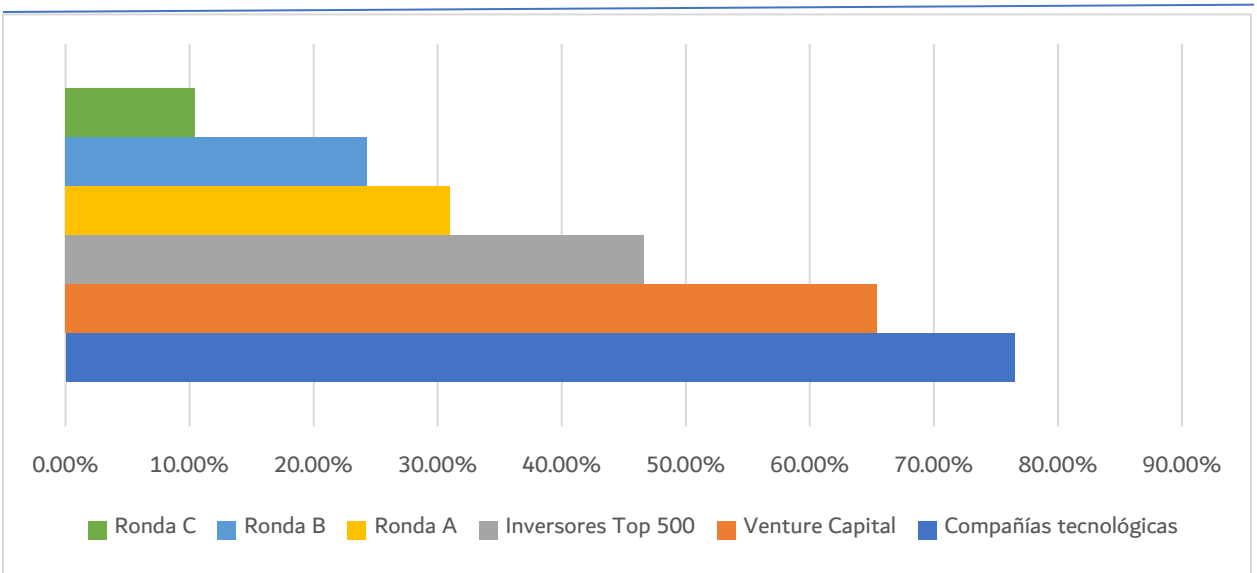


Figura 43– Modelos de financiación de las empresas exitosas

Sector/Localización	CA	MA	NY	TX	Otro	Gran Total
Comercio	134	17	62	21	169	403
Comercio (Tech)	203	34	69	16	213	535
Comunicaciones (Tech)	110	15	16	10	72	223
Educación	10	7	5	0	24	46
Educación (Tech)	83	22	25	17	110	257
Entretenimiento	145	15	66	13	117	356
Entretenimiento (Tech)	592	70	166	48	346	1222
Finanzas	106	25	84	29	375	619
Fintech	236	54	86	46	318	740
Gubernamental	8	1	4	4	28	45
Gubernamental (Tech)	25	10	4	4	59	102
Hardware (Tech)	487	115	44	71	423	1140
Sanidad	133	33	20	37	298	521
Sanidad (Tech)	414	184	54	53	621	1326
Información (Tech)	394	96	65	69	388	1012
Servicios de internet (Tech)	173	38	21	20	163	415

Estilo	54	12	21	10	137	234
Estilo (Tech)	95	5	33	14	80	227
Manufactura	38	5	9	12	107	171
Manufactura (Tech)	177	37	9	21	151	395
Medios	137	13	64	17	171	402
Medios (Tech)	386	60	163	54	340	1003
Móviles (Tech)	146	19	30	13	109	317
Mobilidad	16	0	11	9	66	102
Mobilidad (Tech)	43	6	7	6	56	118
Bienes Raíces	21	4	10	15	54	104
Bienes Raíces (Tech)	29	5	9	4	38	85
Ciencias (Tech)	90	35	16	13	173	327
Seguridad (Tech)	31	9	8	8	57	113
Software (Tech)	324	76	56	56	522	1034
Energía	46	17	13	118	181	375
Energía (Tech)	65	13	10	45	88	221
Total	**4951**	**1052**	**1260**	**873**	**6054**	**14190**

Figura 44– Empresas exitosas según su localización y categoría

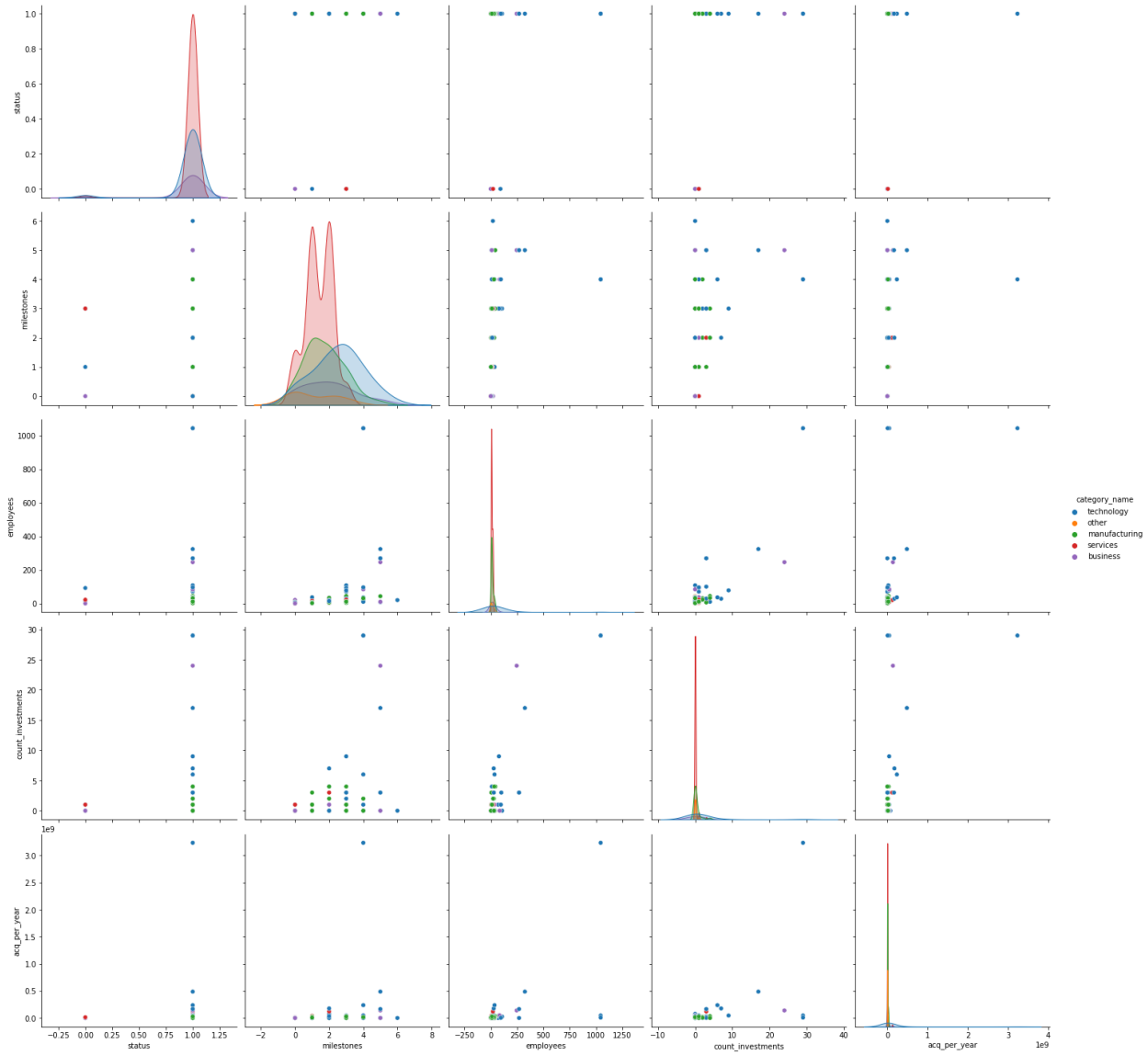


Figura 45– Clustering de relaciones entre columnas

Distribución geográfica de *startups* en Estados Unidos

Figura 46– Distribución geográfica de startups en EEUU

Distribución geográfica de *startups* en Europa

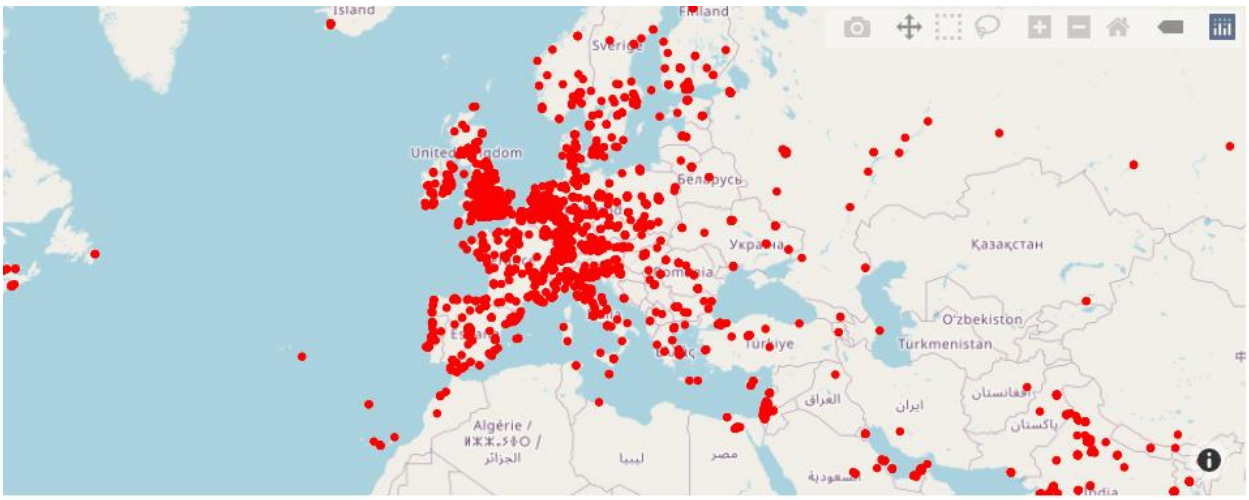


Figura 47– Distribución geográfica de startups en Europa

Comparación de algoritmos

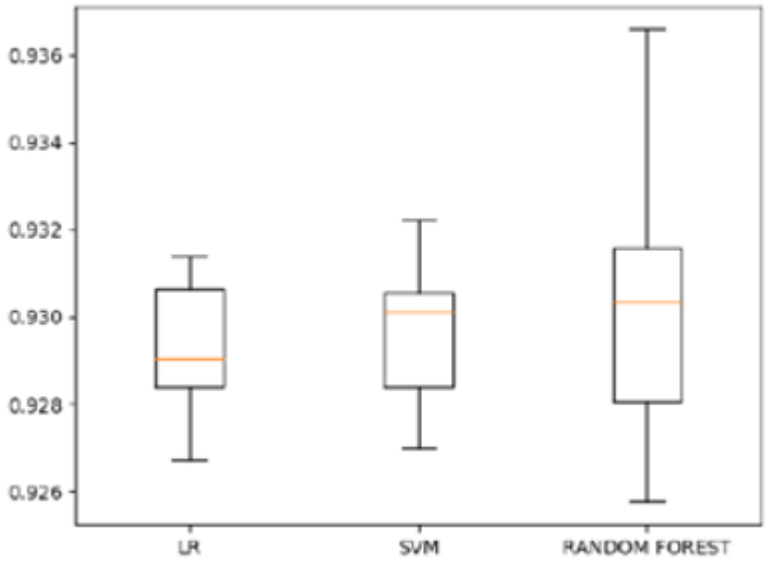


Figura 48– Comparación de algoritmos

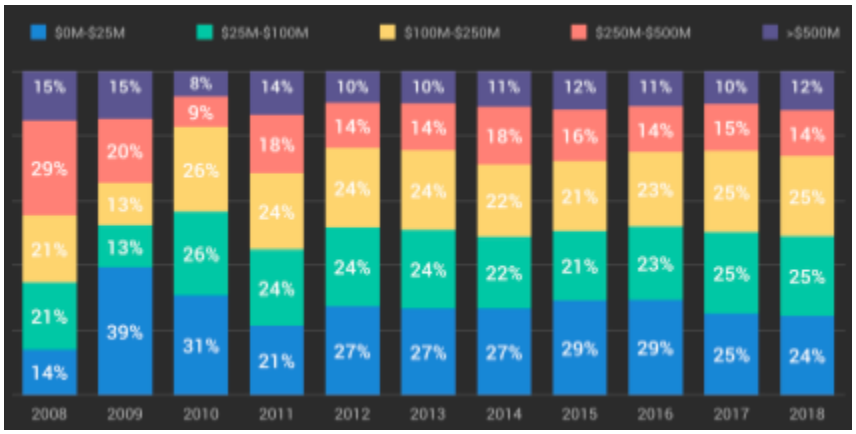


Figura 49– Evolución del tamaño de startups por año

Figura 50– Evolución de la media de financiación en cada ronda en EEUU

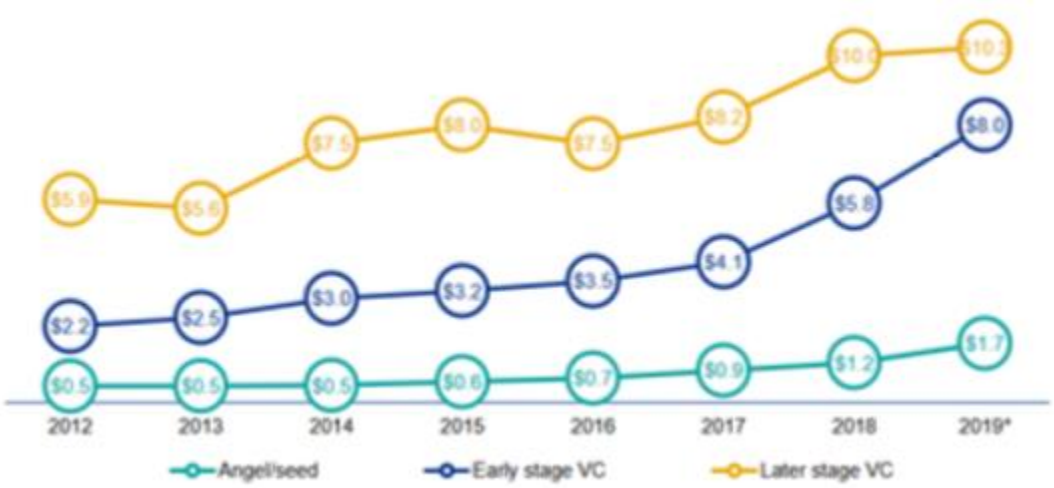


Figura 51– Evolución de la media de financiación en cada ronda a nivel mundial

		Stage		Industry		IPO rate		Fund size		Location		
	All	Early	Late	IT	Health	High	Low	Large	Small	CA	OthUS	Fgn
Important factor												
Team	95	96	93	96	91	96	96	96	95	97	93	96
	(1)	(1)	(3)	(2)	(3)	(2)	(1)	(1)	(1)	(1)	(2)	(1)
Business model	83	84	86	85*	75*	79	82	83	82	83	84	81
	(2)	(2)	(4)	(3)	(4)	(3)	(3)	(2)	(2)	(3)	(2)	(3)
Product	74	81***	60***	75	81	75	74	71*	77*	81**	71**	73
	(2)	(2)	(5)	(4)	(4)	(3)	(3)	(3)	(2)	(3)	(3)	(3)
Market	68	74	69	80***	56***	68	74	67	70	76**	66**	64
	(2)	(3)	(5)	(3)	(5)	(4)	(3)	(3)	(3)	(3)	(3)	(3)
Industry	31	30	37	33**	19**	25	29	30	31	31	37	24***
	(2)	(3)	(5)	(4)	(4)	(3)	(3)	(3)	(3)	(3)	(3)	(3)
Valuation	56	47***	74***	54*	42*	59*	49*	59*	52*	63	60	46***
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Ability to add value	46	44	54	41	45	39*	48*	41**	51**	46	48	46
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Fit	50	48	54	49	40	38**	50**	46**	54**	48	51	50
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Most important factor												
Team	47	53**	39**	50***	32***	44	51	44	50	42	44	55***
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(3)
Business model	10	7***	19***	10	6	7	11	10	10	11	11	8
	(1)	(2)	(4)	(3)	(3)	(2)	(2)	(2)	(2)	(2)	(2)	(2)
Product	13	12	8	12***	34***	18*	11*	15*	10*	13	14	11
	(1)	(2)	(3)	(3)	(5)	(3)	(2)	(2)	(2)	(2)	(2)	(2)
Market	8	7	11	13*	6*	11	10	11***	5***	15***	5***	5
	(1)	(2)	(3)	(3)	(3)	(2)	(2)	(2)	(1)	(3)	(1)	(2)
Industry	6	6	4	3*	9*	6	3	7*	4*	7	7	2**
	(1)	(1)	(2)	(2)	(3)	(2)	(1)	(2)	(1)	(2)	(2)	(1)
Valuation	1	0***	3***	0*	2*	3	1	2	1	2	1	1
	(0)	(0)	(2)	(0)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Ability to add value	2	2	2	1	1	2	2	1	2	1	2	2
	(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Fit	14	13	13	9	9	9	12	10**	17**	10*	16*	15
	(1)	(2)	(4)	(2)	(3)	(2)	(2)	(2)	(2)	(2)	(2)	(2)
Number of responses	558	241	90	129	86	138	156	251	310	161	218	199

Figura 52 - factores principales para la selección de inversiones

		Stage		Industry		IPO rate		Fund size		Location		
	All	Early	Late	IT	Health	High	Low	Large	Small	CA	OthUS	Fgn
Important factor												
Deal flow	65	68	65	73***	49***	62	64	69	62	73	67	57***
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(4)
Selection	86	87	87	91**	81**	89	88	88	85	87	87	84
	(1)	(2)	(4)	(3)	(4)	(3)	(3)	(2)	(2)	(3)	(2)	(3)
Value-add	84	85*	77*	78**	89**	87	83	84	83	86*	79*	89**
	(2)	(2)	(5)	(4)	(4)	(3)	(3)	(2)	(2)	(3)	(3)	(2)
Other	4	3	6	3	3	5	4	4	4	2	4	5
	(1)	(1)	(3)	(1)	(2)	(2)	(2)	(1)	(1)	(1)	(1)	(2)
Most important factor												
Deal flow	23	27	19	29***	13***	19**	31**	27	21	27	25	18**
	(2)	(3)	(4)	(4)	(4)	(3)	(4)	(3)	(2)	(4)	(3)	(3)
Selection	49	44	52	49	52	57**	46**	51	46	48	50	48
	(2)	(3)	(5)	(4)	(5)	(4)	(4)	(3)	(3)	(4)	(3)	(4)
Value-add	27	27	27	21**	35**	22	22	22***	32***	23	23	34**
	(2)	(3)	(5)	(4)	(5)	(3)	(3)	(3)	(3)	(3)	(3)	(3)
Other	1	1	2	1	0	2	1	1	1	1	1	0
	(0)	(1)	(1)	(1)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(0)
Number of responses	509	226	82	122	78	129	139	231	281	145	205	179

Figura 53 - factores principales para la creación de valor en startups