UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
Facultad de Ciencias Sociales y Jurídicas de Elche

Titulación: Grado en Estadística Empresarial
Trabajo Fin de Grado
Curso Académico: 2024–2025

Explorando los Factores del Éxito y Fracaso en Startups

Alumna: Silvia Samper Perujo

Tutora: María Asunción Martínez Mayoral

Elche, junio de 2025

Índice de contenidos

1. Resumen

El presente proyecto se centra en el desarrollo de modelos predictivos para estimar el **éxito o fracaso de startups**, considerando factores clave como el financiamiento, la experiencia del equipo fundador y las características del sector. Ante una alta tasa de fracaso, el objetivo es anticipar con mayor precisión qué startups tienen mayor probabilidad de sobrevivir y crecer.

Se han implementado tres algoritmos de clasificación supervisada: **Regresión Logística**, **Árbol de Decisión** y **Random Forest**, evaluando su capacidad para identificar correctamente las startups que han conseguido cierto nivel de éxito.

El modelo de **Regresión Logística** mostró el mejor rendimiento para identificar casos de éxito. Este trabajo demuestra cómo los modelos predictivos, pero también otras técnicas de aprendizaje supervisado, pueden apoyar la toma de decisiones estratégicas en las fases iniciales de una startup, facilitando la identificación temprana de factores clave para su éxito.

2.Palabras clave

Startup, predicción de éxito, condicionantes del éxito, modelo logístico, aprendizaje supervisado, árbol de clasificación, bosques aleatorios.

3. Contexto

Las empresas emergentes, también conocidas como startups han tenido un crecimiento exponencial en los últimos años. Sin embargo, aproximadamente el 70% de las startups fracasan entre los 2 y 5 años, y solo el 30% sobrevive una década.
Estos datos resaltan la importancia de comprender qué factores contribuyen al éxito o fracaso de una startup.

Actualmente, se han propuesto diferentes enfoques para abordar este problema. Uno de los más conocidos es el enfoque de Lean Startup, propuesto por Eric Ries (2011), que aboga por

un modelo basado en la innovación continua y la validación temprana de ideas. Basándose en sus experiencias profesionales como emprendedor y en casos prácticos.

Por otro lado, un enfoque relevante es el propuesto por Steve Blank en su libro *The Four Steps to the Epiphany* (2013). Blank identifica cuatro pasos fundamentales para que las startups puedan descubrir qué soluciones realmente resuelven los problemas de los clientes. Se centra en entender profundamente al cliente y validar la demanda antes de lanzar un producto al mercado.

Aunque ambos enfoques son valiosos para el desarrollo de una startup, hay distintos aspectos que pueden ser abordados en este estudio y justificados por las limitaciones que hemos identificado en ellos, y que son:

1. **Enfoque limitado en modelos predictivos**: La mayoría de los estudios usan modelizaciones basadas en series temporales que no asocian los efectos con las causas y por lo tanto no identifican por qué evolucionan las startups al éxito. Este estudio explorará el uso de **modelos predictivos** que consideren factores que potencialmente pueden estar afectando al éxito, como el financiamiento, la experiencia del equipo y las características del sector para prever el éxito o fracaso de las startups.

2. **Diversidad limitada en las muestras**: Muchos estudios se centran en startups de un solo sector o región, lo que limita la capacidad de generalizar los resultados. En nuestro caso ampliamos el análisis utilizando **datasets más amplios y diversos** que incluyan startups de diferentes industrias y ubicaciones para obtener patrones más representativos.

4. Objetivos

El objetivo principal de este trabajo es identificar los factores clave que influyen en el **éxito o fracaso de las startups**, a partir de la información disponible en una base de datos diversa y amplia de startups estadounidenses, y desarrollar un modelo de predicción óptimo que nos permita predecir de algún modo las posibilidades de éxito de una startup, en función de sus características.

Como objetivos específicos proponemos los siguientes:

1. Definir el éxito en una startup de modo congruente, a partir de algunas de las variables recopiladas en la base de datos.
2. Explorar la base de datos y las relaciones entre las variables disponibles y la variable de éxito.
3. Identificar las variables clave que influyen en el éxito de las startups.
4. Proponer, entrenar, evaluar y comparar diferentes modelos predictivos y de aprendizaje supervisado.
5. Seleccionar el mejor modelo y ponerlo en producción para predecir el éxito de cualquier startup en función de sus características más relevantes.

5. Información disponible

Descripción del Banco de Datos

El estudio se basa en la base de datos proporcionada por Manishkc06, y recopilada de múltiples fuentes relacionadas con el éxito de las startups, disponible en Kaggle bajo el nombre "Startup Success Prediction".

Este conjunto de datos contiene información sobre una serie de startups en Estados Unidos, entre los años 1984 y 2013, como el financiamiento, el sector de la empresa y otros factores relevantes que pueden influir en su éxito o fracaso.

El objetivo principal de Manishkc06 fue proporcionar un conjunto de datos que permitiera predecir si una startup tendría éxito o no, en función de variables como el financiamiento, el sector de la empresa, y otras características accesibles para las startups.

Estructura del Banco de Datos

El conjunto de datos consiste en un único archivo CSV descargable, que contiene 924 filas con información sobre diferentes startups y un total de 51 columnas con información sobre las mismas.

Descripción de las Variables Utilizadas

A continuación se presentan las variables que se utilizarán en el estudio. Para cada variable, se incluye una breve descripción de su significado y el tipo de dato :

1. **State Code** (**state_code**)

   Descripción: Código del estado en el que la startup está ubicada.
   Tipo de dato: Categórico

2. **Funding Total USD** (**funding_total_usd**)

   Descripción: Monto total de financiamiento recaudado por la startup.
   Tipo de dato: Numérico continuo

3. **Funding Rounds** (**funding_rounds**)

   Descripción: Número total de rondas de financiamiento recibidas.
   Tipo de dato: Numérico discreto

4. **Milestones** (**milestones**)

   Descripción: Número de hitos importantes alcanzados por la startup.
   Tipo de dato: Numérico entero

5. **Category Code** (**category_code**)

   Descripción: Categoría o sector al que pertenece la startup (por ejemplo, tecnología, salud, etc.).
   Tipo de dato: Categórico

6. **Age at First Funding Year** (**age_first_funding_year**)

   Descripción: Edad de la startup (en años) en el momento de recibir su primera financiación.
   Tipo de dato: Numérico entero

7. **Age at Last Funding Year** (**age_last_funding_year**)

   Descripción: Edad de la startup (en años) en el momento de recibir su última financiación.
   Tipo de dato: Numérico entero

8. **Has VC** (**has_VC**)

   Descripción: Indica si la startup ha recibido financiamiento de capital riesgo (venture capital).
   Tipo de dato: Categórico (Binario)

9. **Has Angel** (**has_angel**)

   Descripción: Indica si la startup ha recibido financiamiento de inversores ángeles.
   Tipo de dato: Categórico (Binario)

10. **Status** (**status**)

    Descripción: Estado actual de la startup (por ejemplo, éxito o fracaso).
    Tipo de dato: Categórico

11. **Is Top 500** (**is_top500**)

    Descripción: Indica si la startup forma parte de las 500 principales.
    Tipo de dato: Categórico (Binario)

12. **Founded At** (**founded_at**)

    Descripción: Fecha de fundación de la startup.
    Tipo de dato: Fecha

13. **Closed At** (**closed_at**)

    Descripción: Fecha de cierre de la startup, si aplica.
    Tipo de dato: Fecha

14. **Relationships** (**relationships**)

    Descripción: Número de relaciones estratégicas o asociaciones.
    Tipo de dato: Numérico entero

15. **City** (**city**)

    Descripción: Ciudad en la que está ubicada la startup.
    Tipo de dato: Categórico

16. **Is CA** (**is_CA**)

    Descripción: Indica si la startup está ubicada en California.
    Tipo de dato: Categórico (Binario)

17. **Is NY** (**is_NY**)

    Descripción: Indica si la startup está ubicada en Nueva York.
    Tipo de dato: Categórico (Binario)

18. **Is TX** (**is_TX**)

    Descripción: Indica si la startup está ubicada en Texas.
    Tipo de dato: Categórico (Binario)

19. **Is MA** (**is_MA**)

    Descripción: Indica si la startup está ubicada en Massachusetts.
    Tipo de dato: Categórico (Binario)

20. **Is Other State** (**is_otherstate**)

    Descripción: Indica si la startup está ubicada en un estado distinto a CA, NY, TX o MA.
    Tipo de dato: Categórico (Binario)

21. **Age at First Milestone Year** (**age_first_milestone_year**)

    Descripción: Edad de la startup (en años) cuando alcanzó su primer hito relevante.
    Tipo de dato: Numérico continuo

22. **Age at Last Milestone Year** (**age_last_milestone_year**)
    Descripción: Edad de la startup (en años) cuando alcanzó su último hito registrado.
    Tipo de dato: Numérico continuo

Variables utilizadas en el estudio

En este estudio, se han empleado exclusivamente las siguientes variables:

- **2. Funding Total USD** (funding_total_usd)
- **3. Funding Rounds** (funding_rounds)
- **4. Milestones** (hitos)
- **5. Category Code** (category_code)
- **8. Has VC** (has_VC)
- **9. Has Angel** (has_angel)
- **14. Relationships** (utilizada como relationships_category, una transformación categórica de la variable original)
- **10. Status** (utilizada como variable dependiente bajo la forma de nivel_exito)
- **Variables derivadas:**
    - grupo_funding (agrupación de funding_total_usd)
    - grupo_categoria (agrupación de category_code)

5.1 Procesado de los datos

En el procesado de datos inspeccionamos y resolvemos distintas cuestiones sobre los datos relacionadas con la existencia de valores faltantes, recodificaciones y transformaciones consideradas de interés.

Durante el análisis exploratorio, se identificaron varias columnas con valores faltantes:

- **"age_first_milestone_year" y "age_last_milestone_year"** tienen 152 valores faltantes. Esto sugiere que algunas startups no han alcanzado hitos significativos, por lo que la falta de datos no implica un error, sino una característica propia del dataset.

Dado que la mayoría de estos valores faltantes no representan errores en los datos, sino información valiosa sobre el estado de las startups, **no ha sido necesario realizar imputaciones** ni aplicar técnicas de reemplazo.

Dada la existencia de varias variables que aportan información sobre los logros y el nivel de éxito alcanzado por las startups, para proceder al análisis en este trabajo se ha creado una variable categórica, a la que llamamos “nivel_exito”, que ofrece una representación gradual del éxito de las startups. Esta variable se basa en dos variables de la base de datos: el estado operativo de la empresa (status), esto es, si está abierta o cerrada, y su inclusión en el ranking de las 500 startups más destacadas (is_top500).

Los niveles establecidos son los siguientes:

- Nivel 0: Cerrada
  Incluye a todas las empresas cuyo estado (status) es 'closed', lo que indica el cese de su actividad.
- Nivel 1: Solo abierta
  Comprende a las empresas que continúan abiertas pero que no figuran entre las 500 más destacadas según el indicador is_top500.

- Nivel 2: Top 500 y abierta
  Incluye a las empresas que se mantienen operativas y, además, pertenecen al grupo de las top 500. Este nivel representa el grado de éxito más alto dentro de la muestra.

Para trabajar con las variables de apertura y cierre, formateadas originariamente en día, mes y año, se extrajo el año de ambas para crear el **año de fundación** y el **año de cierre** de cada startup.

Además se consideró útil agrupar estos años en rangos, para facilitar y simplificar la representación e interpretación visual y analítica de los mismos. Así pues se definieron dos nuevas variables categorizadas:

- **período de cierre**:
  - **Período 1**: startups cerradas hasta el año 2007.
  - **Período 2**: cierres ocurridos entre 2008 y 2011.
  - **Período 3**: cierres en 2012 o 2013.
  - **Fuera de rango**: años que no encajan en ninguna de las categorías anteriores.
  - **nada**: empresas que no llegaron a cerrar (por tanto, sin fecha de cierre registrada).
- **año de fundación**:
  - **Período 1**: incluye empresas fundadas hasta 1999, así como las fundadas en 2012 y 2013.
  - **Período 2**: engloba las fundadas entre 2005 y 2009.
  - **Período 3**: incluye los años 2000 a 2004 y también 2010 y 2011.
  - Finalmente, aquellas empresas fundadas fuera de los rangos definidos.

Respecto al número de hitos alcanzados, decidimos agruparlos en los siguientes rangos:

- **0**: empresas que no han alcanzado ningún hito.
- **[1, 2]:** empresas que han alcanzado entre uno y dos hitos.
- **[3, 4]:** empresas con entre tres y cuatro hitos.
- **[5 o más]**: empresas que han alcanzado cinco hitos o más.

Para facilitar el análisis del éxito de las startups en función de la financiación obtenida, se ha creado una variable categórica a partir del campo funding_total_usd, definida en intervalos:

- **0**: startups que no han recibido ninguna financiación.
- **[1, 2]**: startups que han recibido entre 1 y 200.000 dólares.
- **[3, 4]**: startups que han recibido entre 200.001 y 1.000.000 dólares.
- **[5 o más]**: startups con más de 1.000.000 dólares de financiación acumulada.

Respecto al sector industrial en el que se ubican las startups, se creó una nueva variable categórica que agrupa diferentes categorías afines:

- **Tecnología**: incluye categorías relacionadas con software, hardware y videojuegos (software, hardware, games_video).
- **Servicios Web**: agrupa las categorías vinculadas a servicios en línea como portales web, alojamiento de redes y mensajería (web, network_hosting, messaging).
- **Móvil**: comprende únicamente la categoría de tecnología móvil (mobile).
- **Negocios**: incluye sectores orientados a empresas, comercio electrónico, análisis de datos, finanzas, consultoría y manufactura (enterprise, ecommerce, analytics, finance, consulting, manufacturing).
- **Ciencias**: abarca industrias científicas y de salud, tales como semiconductores, biotecnología, tecnologías limpias, medicina y salud (semiconductor, biotech, cleantech, medical, health).
- **Industria Variada**: contiene una amplia gama de sectores diversos como relaciones públicas, seguridad, redes sociales, búsqueda de información, noticias, viajes, moda, fotografía, música, educación, bienes raíces, automotriz, transporte, hostelería, deportes y otros (public_relations, security, social, search, news, travel, fashion, photo_video, music, education, real_estate, automotive, transportation, hospitality, sports, other).

- **Otras**: para cualquier categoría que no encaje en los grupos anteriores.

Para analizar la relación entre el nivel de éxito de las startups y el número de relaciones que mantienen con otras empresas, se creó una nueva variable categórica a partir de la variable numérica original, que indica la cantidad de relaciones de cada startup:

- 0 relaciones
- 1 a 5 relaciones
- 6 a 10 relaciones
- Más de 10 relaciones

6. Metodología

6.1 Análisis Exploratorio de Datos

El análisis exploratorio de datos se centró en comprender la distribución de las variables, así como en explorar relaciones entre las características de las startups y su nivel de éxito.

Se utilizaron gráficos de barras para visualizar la distribución de las variables disponibles, todas ellas categóricas. Para ello, se representaron las frecuencias relativas añadiendo etiquetas porcentuales para facilitar la interpretación visual.
Para visualizar la relación entre la variable de éxito y las restantes, que utilizamos como predictoras potenciales, usamos igualmente gráficos de barras, diferenciando por color los distintos niveles de la respuesta, y evaluando las frecuencias relativas marginales para cada nivel de éxito, lo que nos permite visualizar si el reparto de startups en sus diferentes categorías varía en función del nivel de éxito conseguido.

6.2 Modelización

El presente trabajo se fundamenta en aplicar técnicas de aprendizaje supervisado sobre un conjunto de datos, para construir modelos predictivos capaces de predecir el éxito de una startup, esto es, de clasificarla correctamente como exitosa o no exitosa en función de la información disponible sobre ella. En este contexto, se ha utilizado como variable respuesta el nivel de éxito definido en tres niveles (0: fracaso, 1: éxito medio, 2: éxito alto). Las variables potencialmente predictoras utilizadas fueron: 'hitos','grupo_funding', 'grupo_categoria','has_VC', 'has_angel','nivel_exito', 'category_code', 'funding_rounds'.

Como cuestiones generales de estos modelos:

- Ajuste con una muestra de entrenamiento (generalmente 80% de la muestra) y validación con la muestra test (20% restante); estratificación de la partición para asegurar la representatividad en la muestra original.
- Validación cruzada con la muestra de entrenamiento para validar la estabilidad del modelo.
- Optimización de los modelos: estimación de hiperparámetros para simplificar el modelo y reducir sobreajuste, con búsqueda en grid.
- Evaluación y comparación de los modelos con métricas de clasificación.
- Optimización de hiperparámetros mediante Grid Search.

  En los modelos supervisados, la elección adecuada de los hiperparámetros es fundamental para conseguir un buen rendimiento y evitar problemas como el sobreajuste. Para encontrar la combinación óptima de hiperparámetros, se suele utilizar una estrategia genérica llamada **Grid Search**.

  Grid Search consiste en definir una rejilla o conjunto de posibles valores para cada hiperparámetro relevante del modelo, y luego entrenar el modelo con todas las combinaciones posibles de estos valores. Para cada combinación, se evalúa el rendimiento mediante un procedimiento de validación cruzada, que permite estimar la capacidad de generalización del modelo. Finalmente, se selecciona la configuración que obtiene el mejor rendimiento según la métrica escogida (por ejemplo, precisión o accuracy).

  Esta metodología es aplicable a cualquier modelo supervisado que disponga de hiperparámetros ajustables y es una herramienta fundamental para la optimización y ajuste fino de los modelos.

- El sobreajuste ocurre cuando un modelo aprende no solo los patrones generales presentes en los datos de entrenamiento, sino también las particularidades específicas de ese conjunto. Esto provoca que el modelo tenga un rendimiento muy bueno con los datos que ya ha visto, pero una capacidad baja para generalizar y predecir correctamente

Con el fin de encontrar un buen modelo predictivo, se plantean tres alternativas distintas: regresión logística multinomial, árbol de decisión y random forest, cuyos fundamentos se presentan a continuación.

6.2.1 Regresión logística multinomial

La regresión logística también conocida como **regresión softmax** es un modelo lineal utilizado para estimar probabilidades de pertenencia a diferentes niveles de una variable categórica . En su versión multinomial, permite modelar variables dependientes con tres o más categorías, como la que nos ocupa. Este tipo de modelo permite estimar la probabilidad de pertenencia a cada una de las K clases posibles, dado un conjunto de predictores X y construyendo un predictor lineal de la forma .

$$z_k = w_{0k} + w_{1k}X_1 + \cdots + w_{pk}X_p$$

que explica la probabilidad de pertenencia a la clase k mediante la función:

$$P(y_i = k \mid x_i) = \frac{\exp(x_i^\top \beta_k)}{\sum_j \exp(x_i^\top \beta_j)}$$

Si bien existen dos variantes, la nominal, cuando las categorías no tienen orden, y ordinal, cuando presentan un orden, nos confinamos a la primera.

- Donde w_{jk} es la pendiente asociada a la variable X_j para la clase k, y w_{0k} es el término independiente (intercepto) para la clase k.

Las probabilidades de que una observación pertenezca a cada clase se obtienen aplicando la **función softmax** sobre los predictores lineales zk. Esta función transforma los scores lineales obtenidos para cada clase en probabilidades que suman 1. La expresión de la función softmax para cada clase k es:

$$\phi(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}, \quad \text{para } k = 1, \ldots, K$$

De forma vectorial, el vector de probabilidades se expresa como:

$$\phi(z) = \left[\frac{e^{z_1}}{\sum_{j=1}^{K} e^{z_j}}, \ \frac{e^{z_2}}{\sum_{j=1}^{K} e^{z_j}}, \ \ldots, \ \frac{e^{z_K}}{\sum_{j=1}^{K} e^{z_j}} \right]$$

Estas probabilidades cumplen las propiedades fundamentales de una distribución de probabilidad:

$$\phi(z_k) = P(y = k \mid X), \quad \sum_{k=1}^{K} \phi(z_k) = 1$$

Partimos en estos modelos de una variable respuesta y que identifica la clase en que ha sido clasificado un sujeto/registro, de entre las K clases posibles de respuesta,

$$t_k = \begin{cases} 1 & \text{si } y = k \\ 0 & \text{si } y \neq k \end{cases}, \quad k = 1, \ldots, K$$

Esto nos permite construir la función de verosimilitud para un dato individual y, que viene dada por:

$$L(y, z) = \prod_{k=1}^{K} \phi(z_k)^{t_k}$$

Los coeficientes del modelo se estiman mediante la **minimización de la entropía cruzada**, una función de pérdida que penaliza las predicciones alejadas de las clases reales. Esta función equivale a tomar el logaritmo de la verosimilitud y cambiarle el signo (para convertirla en un problema de minimización).

La minimización de esta función se realiza mediante **algoritmos de optimización** como el **descenso por gradiente**, que ajustan los coeficientes de forma iterativa hasta encontrar los valores que mejor explican los datos.

Además, la derivada de la función softmax es:

$$\frac{\partial \phi(z_i)}{\partial z_j} = \begin{cases} \phi(z_i)(1 - \phi(z_i)), & \text{si } i = j \\ -\phi(z_i)\phi(z_j), & \text{si } i \neq j \end{cases}$$

En el contexto de la regresión logística multinomial, **la regularización se introduce**, especialmente cuando se dispone de un número elevado de predictores o cuando estos presentan correlación entre sí. Al penalizar la complejidad del modelo, la regularización permite identificar qué variables aportan realmente capacidad predictiva y cuáles pueden ser descartadas sin afectar significativamente al rendimiento.

Hay tres tipos más habituales de regularización:

- **L2 (Ridge):** penaliza la suma de los cuadrados de los coeficientes. Tiende a reducir la magnitud de todos los coeficientes.
- **L1 (Lasso):** penaliza la suma de los valores absolutos de los coeficientes. Favorece soluciones más dispersas, eliminando completamente algunos coeficientes (es decir, los fuerza a cero).
- **ElasticNet:** combina las penalizaciones L1 y L2, equilibrando sus efectos.

La elección del tipo de regularización y la intensidad de la penalización (el parámetro λ) se realiza generalmente mediante validación cruzada sobre una rejilla de valores, buscando el modelo que ofrezca el mejor equilibrio entre rendimiento y simplicidad.

La interpretación de los coeficientes se realiza a través de los **odds** y **log-odds**, que permiten evaluar cómo afecta cada predictor a la probabilidad relativa de pertenencia a una clase frente a otra.

Para comparar la probabilidad relativa de pertenencia a dos clases diferentes k_1 y k_2, con $k_1 < k_2 \in \{1, \ldots, K\}$, definimos los odds y log-odds de la siguiente manera:

$$\text{odds}(k_1, k_2) = \frac{P(y = k_1 \mid X)}{P(y = k_2 \mid X)} = \frac{e^{z_{k_1}}}{e^{z_{k_2}}} = e^{z_{k_1} - z_{k_2}}$$

$$\text{log-odds}(k_1, k_2) = z_{k_1} - z_{k_2}$$

Como en los modelos lineales, la magnitud de cada coeficiente depende de las unidades de la variable predictora y no refleja directamente la importancia del predictor. Para evaluar el impacto relativo, se suelen usar variables estandarizadas en una misma escala, con el fin de que los coeficientes sean comparables.

Dado un predictor X_j y sus coeficientes para cada clase $\{w_{j1}, w_{j2}, \ldots, w_{jK}\}$, los odds y log-odds entre clases k_1 y k_2 (fijando el resto de variables) se expresan como:

$$\text{odds}(X_j \mid k_1, k_2) = \exp(w_{jk_1} - w_{jk_2}), \quad k_1 < k_2$$

$$\text{log-odds}(X_j \mid k_1, k_2) = w_{jk_1} - w_{jk_2}, \quad k_1 < k_2$$

Si odds $(X_j \mid k_1, k_2) = 1$, el predictor X_j afecta igual a la probabilidad de pertenecer a ambas clases k_1 y k_2.

Si $\text{odds}(X_j \mid k_1, k_2) < 1$, un aumento en $X_j$disminuye la probabilidad de que la observación pertenezca a la clase k_1 frente a k_2.

Si $\text{odds}(X_j \mid k_1, k_2) > 1$, un aumento en X_j aumenta la probabilidad de que la observación pertenezca a la clase k_1 frente a k_2.

6.2.2 Árboles de Decisión para Clasificación

Los árboles de decisión son modelos predictivos que forman parte de los algoritmos de aprendizaje supervisado no paramétrico. Estos modelos se construyen a partir de reglas binarias que permiten dividir las observaciones en función de sus características, con el objetivo de predecir el valor de una variable respuesta, ya sea esta numérica o categórica.

A diferencia de otros métodos predictivos que generan modelos globales mediante una única ecuación aplicada a todo el espacio muestral, los árboles de decisión permiten una modelización más flexible que permite caracterizar nodos/perfiles de clasificación distintos, en función de las predictoras disponibles. Esto es especialmente útil en contextos con múltiples predictores que presentan interacciones complejas y no lineales. En estos casos, los árboles permiten gestionar mejor dichas interacciones, lo que les proporciona gran parte de su potencial.

Estos modelos adoptan una estructura compuesta por nodos y ramas. Cada nodo representa una decisión sobre una variable predictora relevante para estimar la variable respuesta. Esta decisión da lugar, normalmente, a dos subramas que corresponden a las posibles soluciones de la regla binaria aplicada y que secciona los datos en términos de cierto valor frontera para la variable predictora respecto de la que se divide.

Los elementos clave en la estructura de un árbol son los siguientes:

- **Nodo raíz**: Representa el conjunto total de datos y constituye el punto de partida del árbol.
- **Nodo de decisión**: Nodo que se divide en otros subnodos a partir de una regla basada en los predictores.
- **Nodo terminal o hoja**: Nodo que no se divide más. Representa una salida del modelo.
- **Partición**: Proceso de división de un nodo en subnodos.
- **Rama o subárbol**: Cualquier subsección del árbol.
- **Profundidad**: Número de niveles de nodos de decisión del árbol.

En este estudio nos centramos en los **árboles de clasificación**, diseñados para predecir respuestas categóricas con un número finito de clases K.

Ajuste: Algoritmo CART

La construcción de árboles de clasificación se realiza mediante un proceso de **división binaria recursiva**, utilizando el algoritmo **CART (Classification and Regression Trees)**. El proceso comienza con el conjunto de datos completo en el nodo raíz. En cada iteración del algoritmo se selecciona la variable predictora y el umbral que permiten clasificar mejor las observaciones, dividiendo el nodo en dos subnodos más homogéneos respecto a la respuesta.

El criterio que guía esta división es la **minimización de una función de impureza**, es decir, se busca generar subgrupos lo más homogéneos posible respecto a la clase de la variable respuesta.

La calidad de una partición se mide con la siguiente función:

$$H(Q_m, \theta) = \frac{n_{im}}{n_m} h(Q_{im}) + \frac{n_{dm}}{n_m} h(Q_{dm})$$

donde $h()$ representa una medida de impureza como la entropía o el índice de Gini, y Q_m es el conjunto de datos en el nodo actual. Se elige el par variable-umbral $\theta = (j, t_{jm})$ que minimice la función de impureza total del nodo.

El proceso se repite hasta que ya no es posible mejorar la pureza de los nodos resultantes o se cumple alguna condición de parada predefinida.

Optimización: sobreajuste y estrategias de control

Medidas de Impureza

A continuación se describen las principales métricas empleadas para medir la impureza de un nodo:

- **Entropía (o pérdida logarítmica)**: Mide la incertidumbre del nodo. Se define como:

  $$D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

  donde p_k^m es la proporción de observaciones en el nodo mm que pertenecen a la clase kk.

- **Índice de Gini**: Mide la varianza total entre las clases dentro de un nodo:

  $$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

  Este es el criterio utilizado por el algoritmo CART. En este trabajo, la construcción del árbol de decisión y del bosque aleatorio se basó en la minimización del **índice de Gini** como criterio para evaluar la calidad de las particiones en cada nodo. La elección del índice de Gini como criterio fue determinada durante la optimización de hiperparámetros, donde se compararon distintos criterios, incluyendo la entropía, y se seleccionó el que ofreció el mejor rendimiento en validación cruzada.

- **Error de clasificación**: Mide la proporción de observaciones que no pertenecen a la clase mayoritaria de un nodo:

  $$E_m = 1 - \max_k \hat{p}_{mk}$$

  Aunque es una métrica sencilla, se considera poco sensible y no se utiliza habitualmente para construir árboles.

- **Estadístico Chi-cuadrado**: Evalúa si una partición produce diferencias significativas entre los nodos hijos y el nodo padre. Es el criterio utilizado en los árboles **CHAID** (Chi-square Automatic Interaction Detector).

Explicación de los hiperparámetros y valores:

- n_estimators: probamos desde 50 hasta 500 árboles para balancear precisión y tiempo de entrenamiento.
- max_features: número de variables consideradas para encontrar la mejor división; valores medianos (5, 7, 9) para buscar equilibrio entre diversidad y potencia.
- max_depth: controlamos la profundidad máxima de cada árbol para evitar sobreajuste.
- criterion: criterios para medir la calidad de la división, usando tanto índice de Gini como entropía.

Una vez construido el árbol, las observaciones de entrenamiento quedan agrupadas en los nodos terminales. Para predecir una nueva observación, se sigue el camino correspondiente en función del valor de sus predictores hasta alcanzar una hoja. La clase predicha será la más frecuente entre las observaciones que llegan a dicho nodo.

Los árboles de decisión tienden a ajustarse muy bien a los datos de entrenamiento, cuando no se imponen restricciones a la profundidad del árbol, dado que el algoritmo generalmente es capaz de llegar a una partición individualizada de todos los datos y así, a un ajuste perfecto sobre los datos de entrenamiento. Este problema de **sobreajuste** se puede aliviar siguiendo dos estrategias principales para conseguir un árbol podado, esto es, un árbol con una profundidad acotada:

1. **Parada temprana (early stopping)**: Limita el crecimiento del árbol estableciendo condiciones de parada mediante una optimización en grid (basada en validación cruzada) de los siguientes hiperparámetros
    - Profundidad máxima del árbol (max_depth).
    - Mínimo número de observaciones para dividir un nodo (min_samples_split).
    - Mínimo número de observaciones en hojas terminales (min_samples_leaf).

    En nuestro caso, hemos ajustado estos hiperparámetros probando diferentes valores mediante validación cruzada para evitar sobreajuste y seleccionar el modelo óptimo.

2. **Poda del árbol (pruning)**: Consiste en construir un árbol grande sin limitaciones iniciales y luego eliminar partes que no mejoran la generalización. Una técnica muy usada es la **poda de complejidad por coste**, que penaliza árboles complejos con la fórmula:

$$E_m = 1 - \max_k \hat{p}_{mk}$$

donde $R(T)$ es una medida de error del árbol y $|T|$es el número de nodos terminales. Al aumentar el valor de $alpha$, se penalizan árboles más grandes.

6.2.3 Bosques Aleatorios (Random Forest)

Como alternativa, los árboles de decisión permiten modelar relaciones no lineales entre predictores, aunque presentan sensibilidad a las muestras, lo que limita su estabilidad. Frente a esto, surge el modelo de **bosque aleatorio** (*Random Forest*), un método de conjunto que mejora la precisión mediante la técnica de *bagging*.

Ajuste : Bagging

El *bagging* consiste en generar múltiples subconjuntos de entrenamiento mediante remuestreo con reemplazo (*bootstrapping*), entrenar un modelo independiente en cada subconjunto y combinar sus predicciones. Su objetivo principal es reducir la varianza y aumentar la estabilidad del modelo.

El procedimiento es el siguiente:

1. Se generan B muestras aleatorias mediante remuestreo con reemplazo a partir del conjunto de entrenamiento.
2. Se entrena un modelo independiente con cada muestra.
3. Se predice sobre la muestra de test con cada uno de los modelos.
4. Se combinan las predicciones:
    - Clasificación: se usa el **voto por mayoría**.
    - Regresión: se utiliza el **promedio de predicciones**.

Esta estrategia no incrementa el sobreajuste a medida que se incrementa el número de modelos, aunque sí puede aumentar el coste computacional.

El algoritmo de **Random Forest** introduce una mejora al bagging aplicando además una selección aleatoria de predictores antes de cada división del árbol. Así se reducen las correlaciones entre árboles, se solventan posibles problemas de colinealidad entre predictores y se mejora la reducción de varianza.

En concreto, para cada nodo de un árbol, se selecciona aleatoriamente un subconjunto de m predictores (siendo m<p, con p el número total de predictores), y se elige la mejor división entre ellos.

Recomendaciones para elegir mm:

- Para clasificación:
  $m \approx \sqrt{p}$
- Para regresión:
  $m \approx \frac{p}{3}$

Valores pequeños de m son preferibles si los predictores están altamente correlacionados.

Para la construcción del modelo Random Forest, seleccionamos el hiperparámetro max_features mediante validación cruzada, y el valor óptimo de este trabajo fue 9. Esto indica que en cada división se consideran 9 variables predictoras, lo cual es un valor relativamente alto. Valores pequeños de max_features suelen ser preferibles cuando existe alta correlación entre variables predictoras para reducir la redundancia, pero en nuestro caso, el valor seleccionado sugiere que la correlación entre variables no era tan alta o que se priorizó el poder predictivo en cada división.

La predicción final en un modelo Random Forest se obtiene agregando las predicciones individuales de los árboles entrenados, que en los problemas de clasificación se consigue asignando la clase más frecuente (voto por mayoría).

Optimización de Hiperparámetros

Cada árbol en el bosque es entrenado con una muestra bootstrap que deja fuera aproximadamente un tercio de las observaciones originales. Estas observaciones se pueden usar como conjunto de validación para estimar el error de clasificación sin necesidad de validación cruzada.

El error Out of Bag, OOBn se define como:

$$\text{OOB Error} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i^{\text{OOB}})$$

donde y^i_{OOB} es la predicción para la muestra i basada únicamente en los árboles donde dicha muestra no fue utilizada en el entrenamiento, y L es la función de pérdida correspondiente,que en este caso es el error de clasificación (tasa de error), calculado como la proporción de predicciones incorrectas.

Los hiperparámetros más influyentes en Random Forest son:

- Número de árboles B
- Número máximo de predictores considerados en cada división m

Para su ajuste se puede usar el **error OOB** como métrica.

ç

Importancia de los Predictores

Dado que Random Forest combina múltiples árboles, pierde capacidad interpretativa frente a un único árbol. Sin embargo, existen métricas que permiten estimar la **importancia de los predictores**:

1. Importancia por Permutación

Para cada predictor Xj, se calcula la diferencia en la métrica de calidad al permutar aleatoriamente sus valores en el conjunto de test:

$$\text{Importancia}(X_j) = \frac{1}{K}\sum_{k=1}^{K}\left[\text{Error}_{\text{perm}}^{(k)}(X_j) - \text{Error}_{\text{ref}}\right]$$

Un valor alto indica alta dependencia del modelo respecto a esa variable.

2. Importancia Basada en Impureza

Se mide la contribución del predictor a la reducción de impureza (por ejemplo, Gini, entropía o MSE) en cada árbol, promediando sobre todos ellos:

$$\text{Importancia}(X_j) = \sum_{t=1}^{T}\sum_{n\in\text{nodos}(t,X_j)}\Delta\text{Impureza}_n$$

donde $\Delta\text{Impureza}_n$ Es la reducción de impureza en el nodo nn en el árbol t, atribuible a Xj.

6.3 Software y hardware

Lenguaje de programación:

El análisis de datos y la modelización se realizaron utilizando el lenguaje de programación **Python**, versión 3.11.12, por su robustez y soporte en bibliotecas de ciencia de datos.

Entorno de desarrollo:

Google Colaboratory (Colab) — plataforma basada en Jupyter notebooks que permite ejecutar código Python en la nube.

Hardware:

Procesamiento realizado en la infraestructura en la nube de Google, que varía según disponibilidad, pero típicamente con CPUs Intel Xeon y GPUs Nvidia Tesla (según configuración de Colab). No se requirieron recursos especiales más allá de la configuración estándar.

Las Bibliotecas y Módulos Utilizados

1. **Pandas**
   Funcionalidad: Manipulación y análisis de datos en estructuras de datos tabulares (DataFrame y Series), manejo de fechas, limpieza y transformación de datos.
   Documentación: https://pandas.pydata.org/docs/
2. **NumPy**
   Funcionalidad: Soporte para arrays multidimensionales, operaciones matemáticas y funciones estadísticas básicas, manejo de rangos y etiquetas.
   Documentación: https://numpy.org/doc/
3. **Matplotlib**
   Funcionalidad: Creación de gráficos estáticos, visualizaciones de barras, histogramas, líneas, con personalización avanzada de ejes y leyendas.
   Documentación: https://matplotlib.org/stable/index.html

4. **Seaborn**
   Funcionalidad: Visualización estadística basada en Matplotlib, facilita la creación de gráficos estéticamente atractivos y con buenas prácticas de diseño, como countplot para contar categorías.
   Documentación: https://seaborn.pydata.org/
5. **Scikit-learn**
   Preprocesamiento: StandardScaler, OneHotEncoder para normalización y codificación de variables categóricas.
   Modelado: LogisticRegression, DecisionTreeClassifier, RandomForestClassifier para clasificación.
   Evaluación: Métricas como accuracy_score, classification_report, confusion_matrix, validación cruzada (cross_val_score) y búsqueda de hiperparámetros (GridSearchCV).
   Documentación: https://scikit-learn.org/

7. Resultados

7.1 Análisis Exploratorio

Iniciamos el análisis exploratorio presentando la información disponible a través de tres variables relevantes en la base de datos:

- (estado) el número de aperturas y cierres de startups durante el periodo estudiado (Gráfico 1);
- (top500) el número de startups que consiguen posicionarse en el top500 del ranking considerado (Gráfico 2);
- (éxito) el número de startups que clasificamos en los tres niveles de éxito que definimos a partir de las variables anteriores: top500 y estado (Gráfico 3).

A continuación, con el objetivo de ilustrar de forma visual la evolución de aperturas y cierres de startups entre 1990 y 2013, se presenta en el Gráfico 1 el número de aperturas y cierres en esos años.

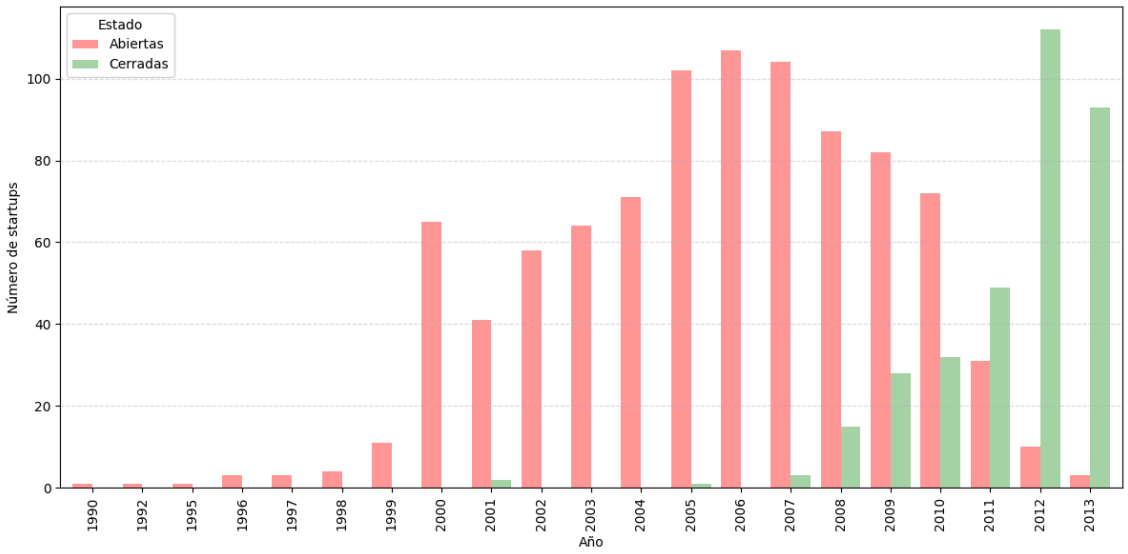


***Gráfico 1**: Distribución del número de aperturas y cierres de startups entre 1990 y 2013.*

Como apreciamos en el Gráfico 1, la apertura de startups se inicia en los años 90 y se concentra especialmente entre los años 2005 a 2010, periodo en el que se observa un mayor volumen. Los cierres se suceden a partir de 2007, y en 2012-2013 llegan a valores máximos, en contraposición al número de aperturas, que cae en picado a partir de 2011 y consigue su mínimo en los últimos años, en 2013.

En el Gráfico 2 se muestra el reparto de las startups en función de si alcanzan el top 500 o no.

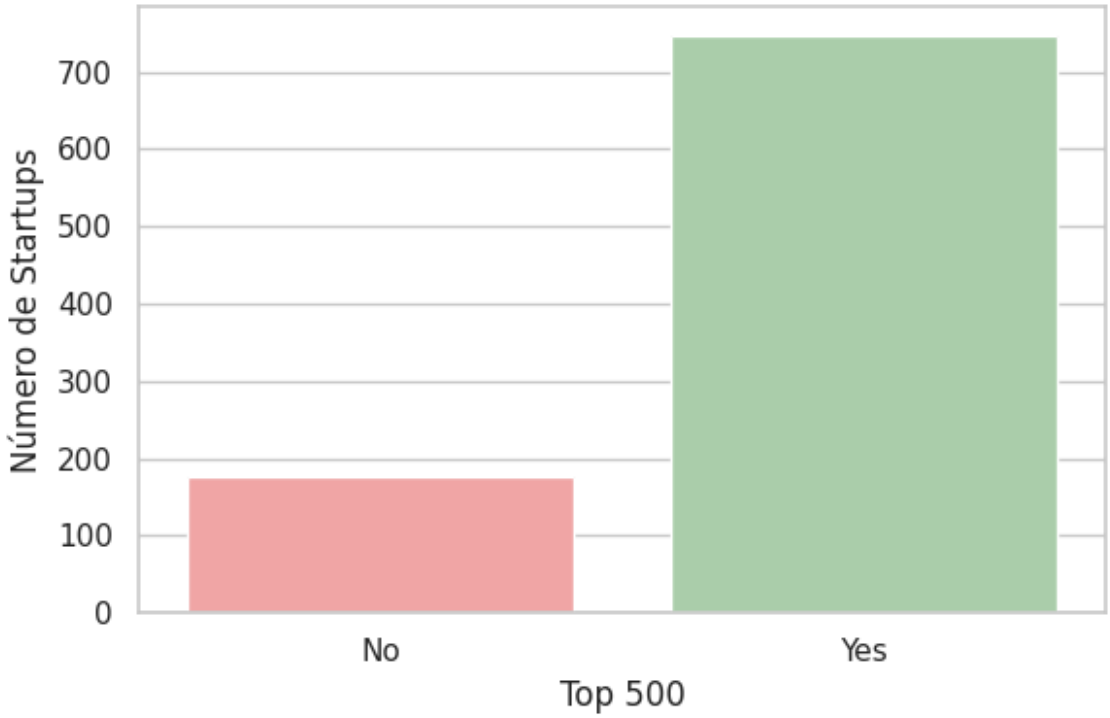


***Gráfico 2**: Distribución del número de startups que están o no en el Top 500.*

Observamos en el Gráfico 2, que el número de empresas que están en el Top 500 es considerablemente mayor al número de empresas que no lo están de hecho, el 70% de las empresas en la base de datos consiguen en algún momento estar en ese TOP500.

Finalmente, en el Gráfico 3 se muestra la distribución de la variable que creamos con las dos anteriores, y en función de la que consideramos si una startup alcanza el éxito. Tenemos con este gráfico, una visualización del porcentaje de startups que consiguen éxito.

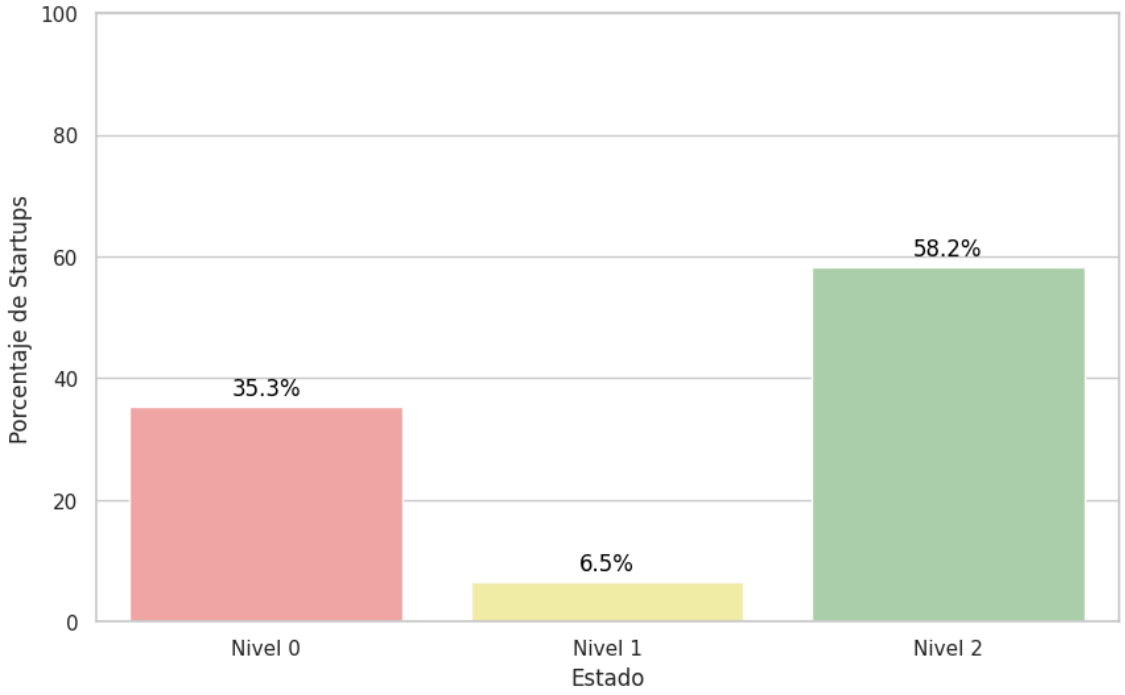


Gráfico 3: *Distribución del nivel de éxito alcanzado por las empresas disponibles en la base de datos.*

Como apreciamos en el Gráfico 3, es mucho mayor el número de startups en nivel 2 (el nivel más alto de éxito), con un 58.2% de las startups actualmente en la base de datos, seguidas de empresas en nivel 0 (empresas que han fracasado), que representan un 35.3% del total.
Las empresas de éxito medio (Nivel 1) representan un porcentaje considerablemente inferior, (del 6.5% del total).

RELACIÓN CON EL ÉXITO

A continuación investigamos la relación entre esta variable de éxito y el resto de las variables disponibles en la base de datos.

En el ***Gráfico 4*** se muestra la *distribución del número de hitos conseguidos, diferenciada por el nivel de éxito alcanzado.*

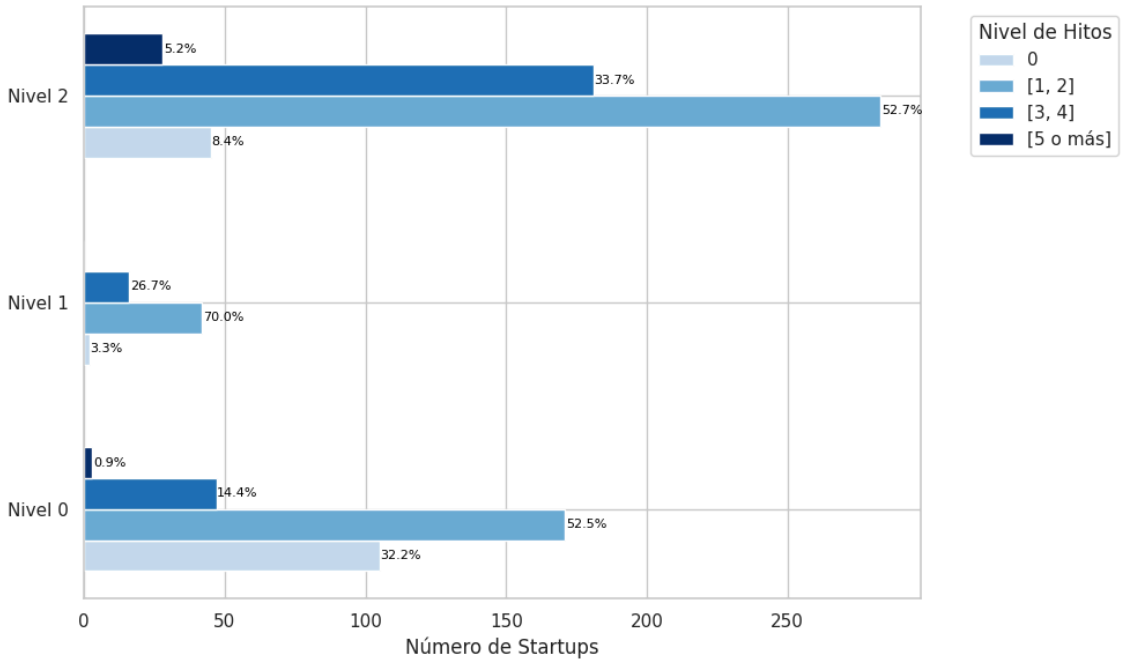


***Gráfico 4** - Distribución del número de hitos conseguidos, diferenciada por el nivel de éxito alcanzado.*

Como apreciamos en el Gráfico 4, lo más frecuente respecto a hitos conseguidos,sea cual sea el nivel de éxito alcanzado por la empresa, es el de 1 a 2 hitos. De hecho, el porcentaje de empresas que han conseguido entre 1 y 2 hitos, tanto en las empresas de éxito (Nivel 2) y como en las que han fracasado (Nivel 0), es similar, en torno al 52.6%; este porcentaje se eleva al 70% en las empresas de éxito medio (Nivel 1). En las empresas de más éxito, tenemos un porcentaje considerablemente mayor (del 38.9%) que han conseguido 3 o más hitos, mientras que este porcentaje se queda en un 15.3% en las empresas que han fracasado, y un 26.7% en las de éxito medio. Apreciamos pues, cierta relación entre el número de hitos conseguidos, y el nivel de éxito alcanzado: a más éxito, más hitos conseguidos.

El ***gráfico 5*** permite ilustrar la *financiación recibida por las empresas según el nivel de éxito alcanzado.*

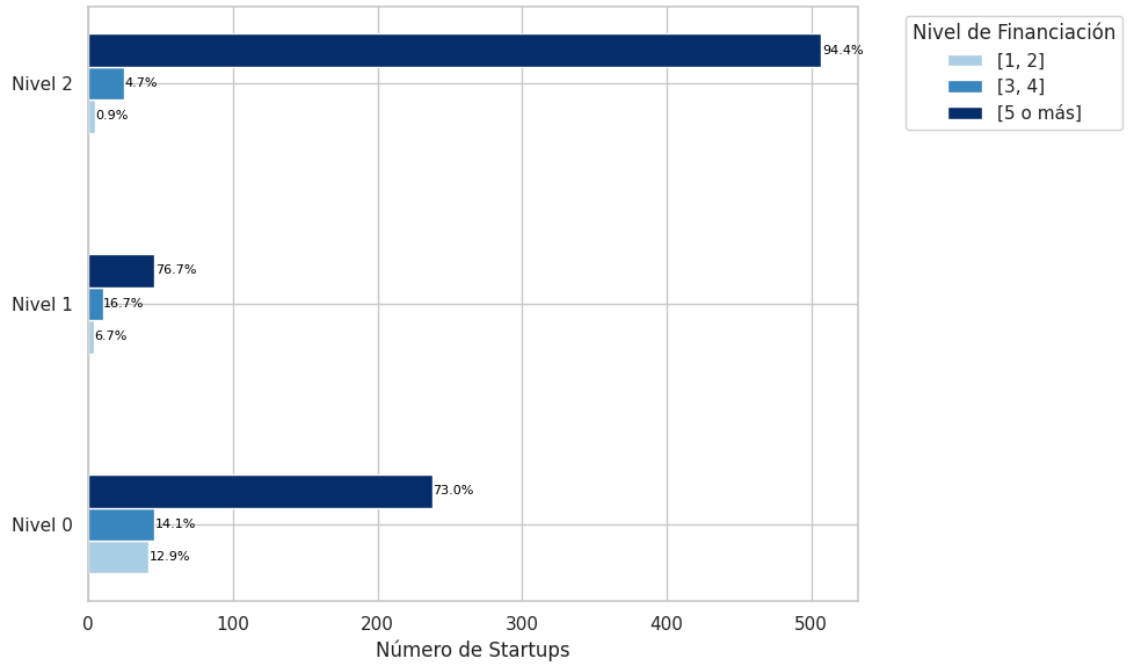


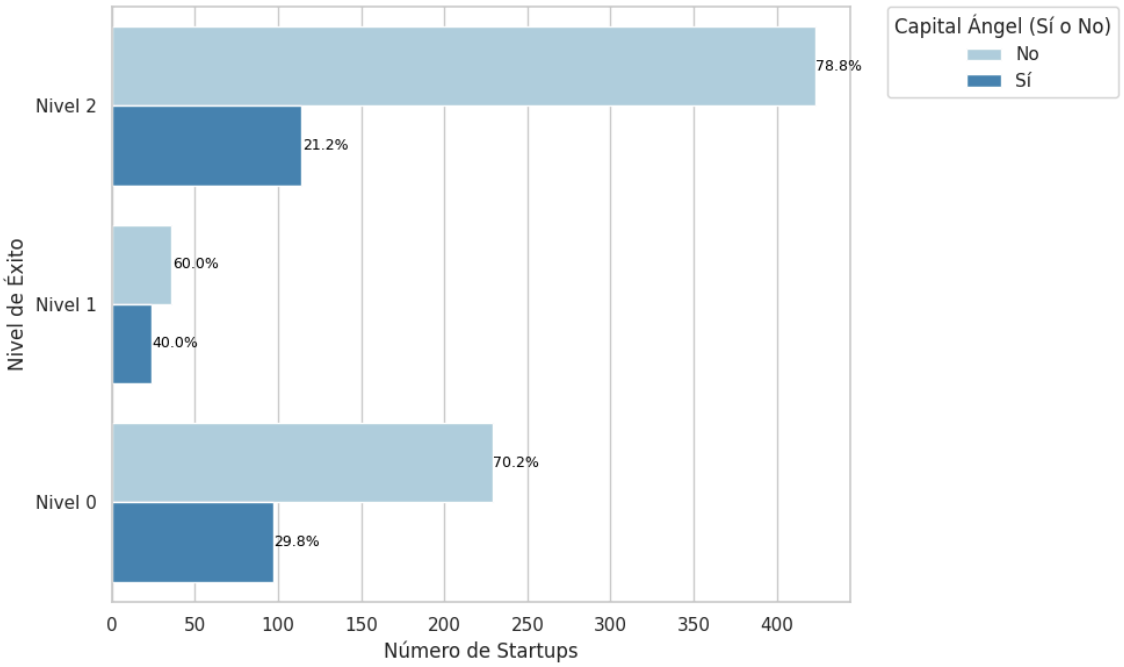
Gráfico 5*- Distribución de la financiación recibida, diferenciada por nivel de éxito alcanzado. Los niveles de financiación corresponden a montó 0 [0], entre 0 y 200000 [1,2], entre 200000 y 100000 [3,4] , más de 100000 [5 o más].*

La financiación más frecuente, sea cual sea el nivel de éxito alcanzado por la empresa, es superior a $1.000.000 (nivel 5 o más). De hecho, el porcentaje de empresas que han conseguido esta financiación, tanto en las empresas de éxito (Nivel 1) y como en las que han fracasado (Nivel 0), es similar, en torno al 75%; este porcentaje se eleva al 94,5% en las empresas de éxito más alto (Nivel 2). En las empresas Nivel 0 y Nivel 1 también apreciamos un porcentaje similar (en torno al 15%) de startups que consiguieron un montó de financiación entre $200.000 y $1.000.000 ([3,4]). Entre las empresas con un éxito mayor (Nivel 2), son muy pocas las que consiguieron una financiación inferior a $1.000.000 (tan sólo un 5.6%).

Una financiación inferior a $200.000 ha sido muy poco habitual en cualquier tipo de startup:tan sólo un 0,9% de las startups en el Nivel 2, un 6,7% de las que quedaron en el Nivel 1, y un

12,9% de las que fracasaron (en el Nivel 0).Podemos apreciar cierta relación en los Niveles 0 y 1 siendo distintos al Nivel 2.

El ***Gráfico 6*** representa la relación entre la inversión ángel y el nivel de éxito alcanzado.



Gráfico

Gráfico 6- *Éxito versus inversión ángel.*

Como apreciamos en el Gráfico 6, el número de startups que no tuvieron capital ángel es el más frecuente, sea cual sea el nivel de éxito alcanzado por la empresa. De hecho, el porcentaje de empresas que no han tenido está en torno al 70%-80% , tanto en las empresas de éxito más alto (Nivel 2) y como en las que han fracasado (Nivel 0);este porcentaje disminuye al 60% en las empresas de éxito (Nivel 1).

En el ***Gráfico 7*** se trata de mostrar la relación entre el *éxito y el capital riesgo.*

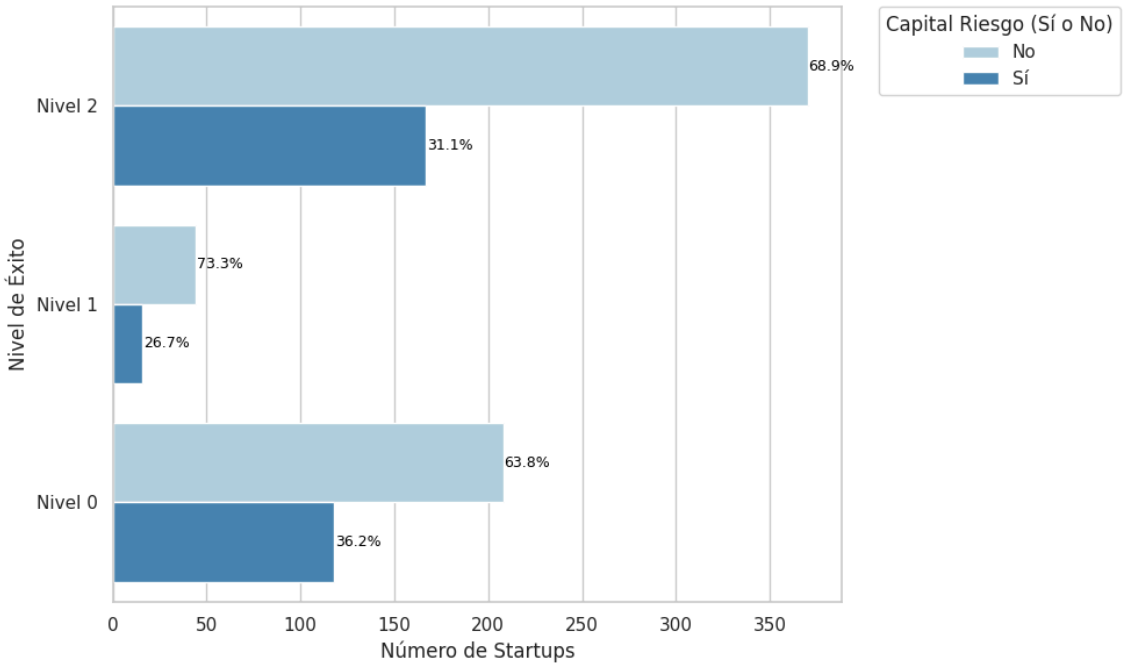


Gráfico 7*- Éxito versus capital riesgo*

Como apreciamos en el Gráfico 7, lo más frecuente en todas las startups, independientemente del nivel de éxito que consiguen, es la ausencia de capital riesgo. De hecho, el porcentaje de empresas que no han tenido está en torno al 65% , tanto en las empresas de éxito más alto (Nivel 2) y como en las que han fracasado (Nivel 0); este porcentaje se eleva al 73.3% en las empresas de éxito (Nivel 1).

Ilustramos a continuación la distribución de startups por sectores empresariales, en el G***ráfico 8***.

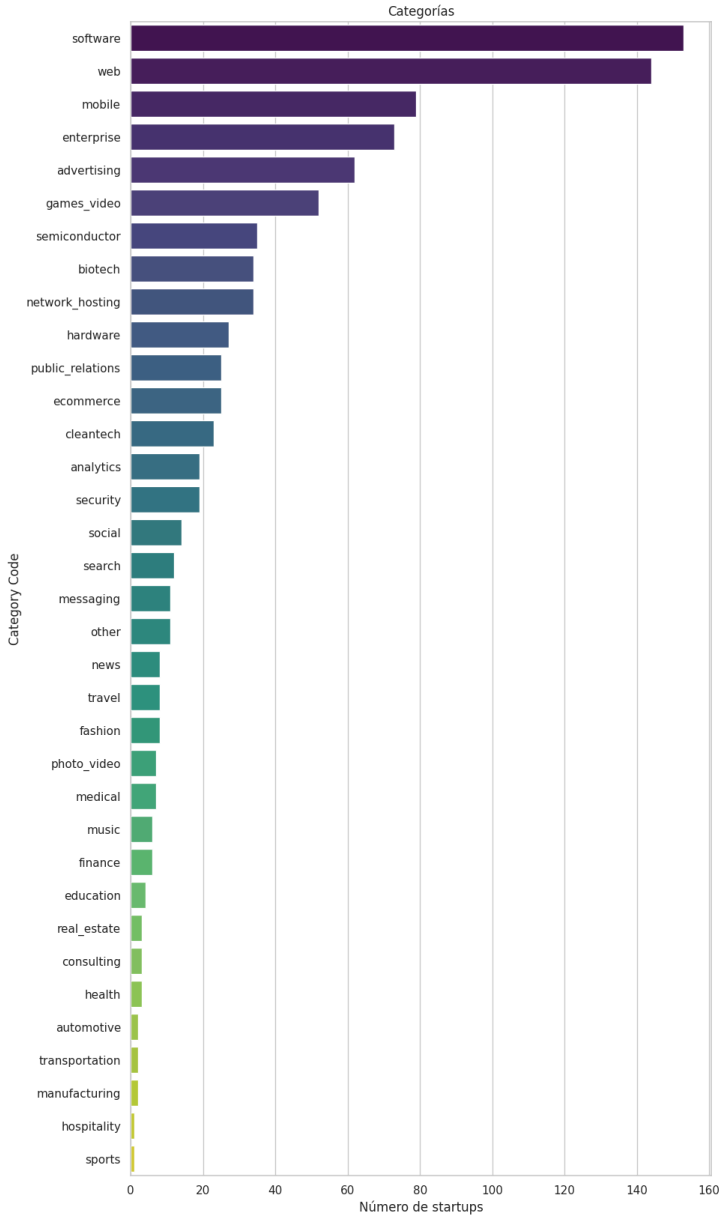


Gráfico 8*- Distribución según sectores*

Este gráfico nos ha facilitado agrupar las startups, por afinidad respecto al tipo de servicio que ofrecen, definiendo así cinco grandes grupos:

- **Technology**, que incluye software, hardware y videojuegos;
- **Web Services**, para servicios en línea como hosting, mensajería y plataformas web;
- **Mobile**, que agrupa únicamente lo relacionado con tecnología móvil;
- **Business**, que engloba sectores como ecommerce, finanzas, análisis de datos, consultoría y manufactura; y
- **Science**, que cubre biotecnología, medicina, salud, cleantech y semiconductores.

Con esta agrupación intentamos relacionar de modo más simple el nivel de éxito alcanzado en función del tipo de industria. La relación entre estas variables se presenta en el Gráfico 9.

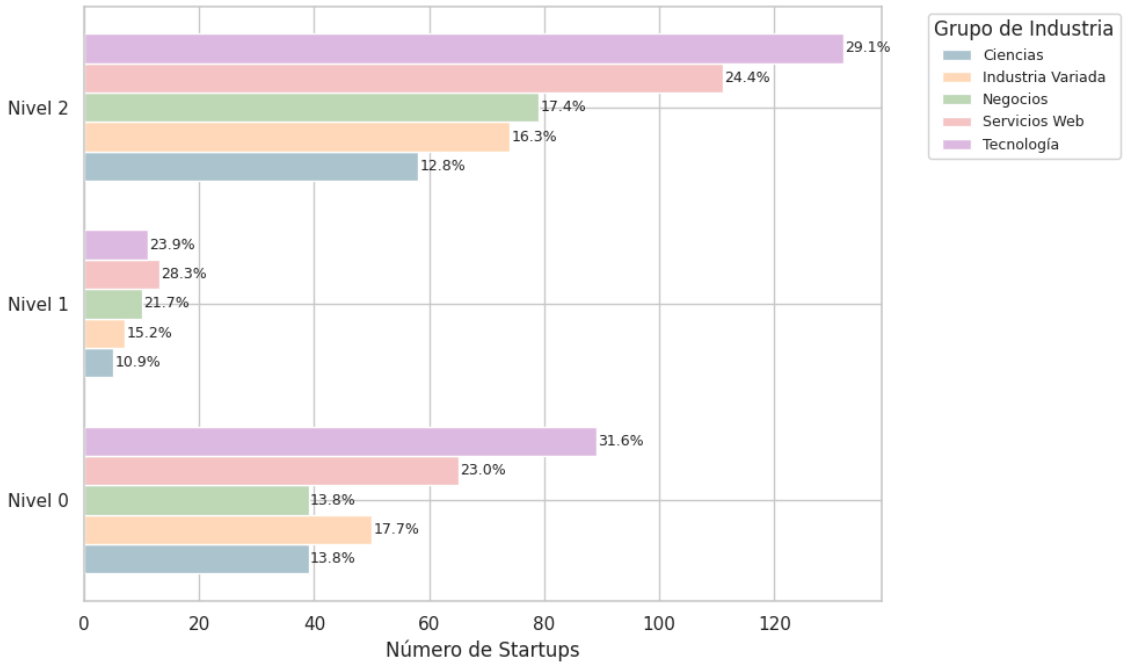


Gráfico 9- *Distribución por sectores industriales agrupados, de las startups según el nivel de éxito conseguido.*

El grupo industrial más frecuente en el entorno de las startups, sea cual sea el nivel de éxito alcanzado por la empresa, es Tecnología excepto en el Nivel 1 que es Servicios web. Con ello, el porcentaje de industria tecnológica tanto en las empresas de éxito (Nivel 2) y como en las que han fracasado (Nivel 0), es similar, en torno al 30%; este porcentaje disminuye al 23.9% en las empresas de éxito medio (Nivel 1). En las empresas de más éxito, tenemos un porcentaje considerablemente mayor (del 24.4%) en la industria de negocios, mientras que este porcentaje se queda en un 13.8% en las empresas que han fracasado, y un 21,3% en las de éxito medio. Apreciamos pues, cierta relación entre las distintas industrias y el nivel de éxito alcanzado.

Por último, el ***Gráfico 10*** *ilustra* el número relaciones de las startups según el nivel de éxito alcanzado.

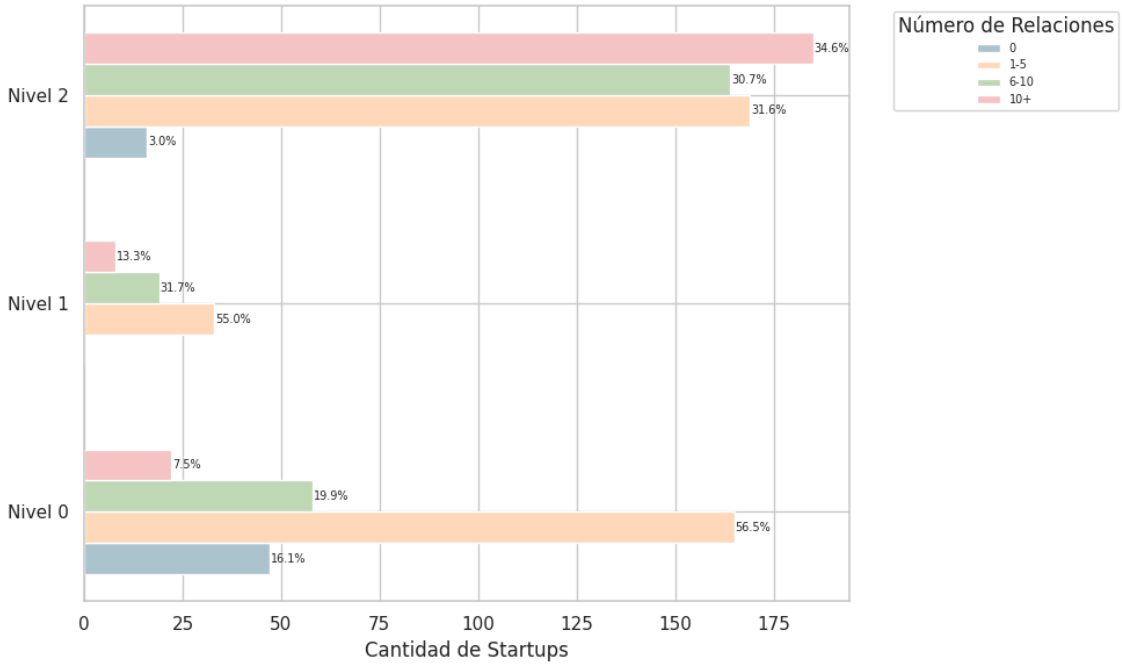


Gráfico 10- *Distribución del número de relaciones según el nivel de éxito.*

El número de relaciones más frecuente en empresas que han fracasado (Nivel 0) y en empresas de éxito (Nivel 1) es de 1 a 5.

En el Nivel 1 encontramos un 31,7% de empresas que tienen entre 6 y 10 relaciones; Disminuyendo este porcentaje al 19.9% en el Nivel 0.

Además, en el Nivel 1 no encontramos empresas que no tengan relaciones.Sin embargo, en las empresas que han fracasado (Nivel 0), este porcentaje es del 16.1%.

Por otro lado, en las empresas que están abiertas y en el top 500 (Nivel 2) encontramos que hay porcentajes bastante similares en las relaciones de 1 a 5, de 6 a 10 y más de 10, estando este entorno al 30%. Este porcentaje disminuye al 3% en 0 relaciones.

7.2 Modelización

A continuación, se presentan los resultados de los tres modelos de clasificación aplicados al conjunto de prueba: **Regresión Logística**, **Árbol de Decisión** y **Random Forest**. Las métricas utilizadas para la evaluación incluyen:

- **Precision**: Proporción de predicciones correctas entre todas las predicciones hechas para una clase. Indica cuánto de preciso es el modelo cuando predice una clase.
- **Recall** : Proporción de elementos correctamente identificados entre todos los elementos reales de una clase. Mide la capacidad del modelo para capturar todos los casos verdaderos.
- **F1-score**: Media armónica entre precision y recall.
- **Support**: Número de muestras reales de cada clase en el conjunto .
- **Accuracy (Exactitud)**: Proporción de todas las predicciones correctas sobre el total de predicciones.

7.2.1. Regresión Logística

El modelo finalmente ha sido una **Regresión Logística Multinomial** entrenada con el solver 'lbfgs', sin regularización explícita (penalty por defecto = 'l2'), y con random_state=42.

Representación de los coeficientes del modelo

A continuación, se presentan los coeficientes mejor estimados por la regresión logística para cada clase (Nivel 0, Nivel 1 y Nivel 2), agrupados por variable predictora. El objetivo de este gráfico es identificar qué variables explican mejor la probabilidad de pertenecer a cada nivel, y cuáles no aportan significativamente al modelo.

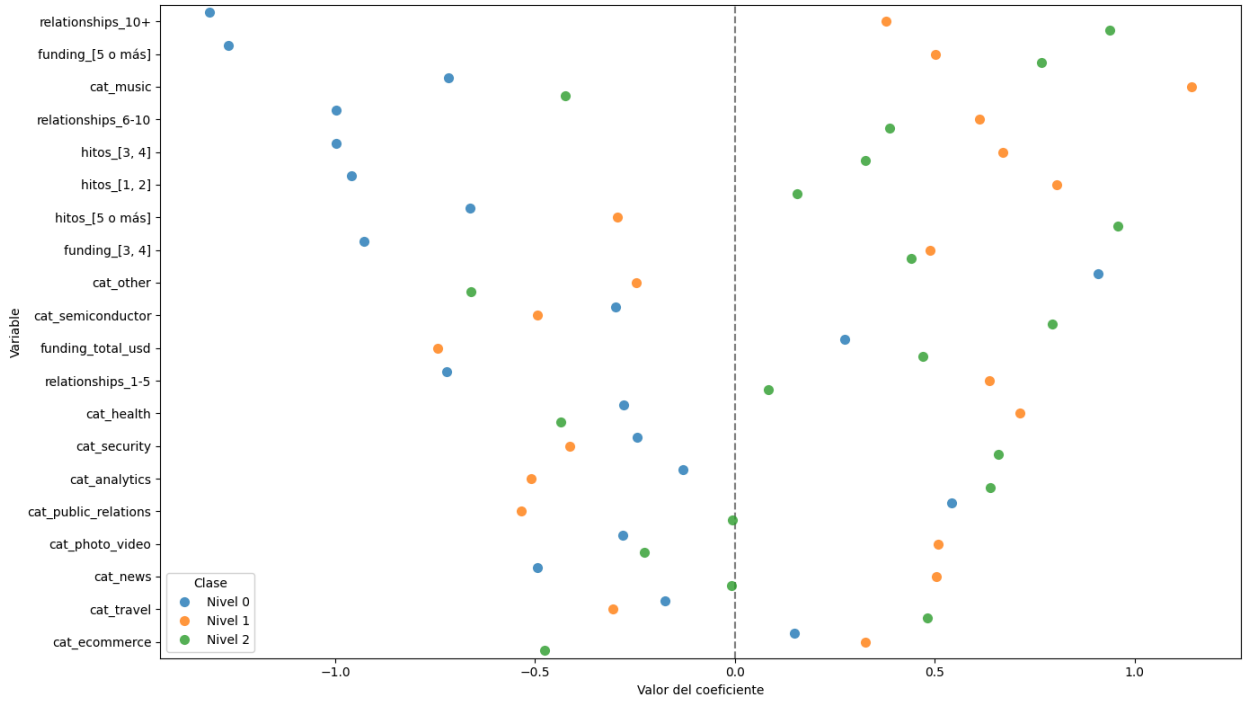


***Gráfico 11**- Coeficientes de las variables predictoras por nivel en el modelo de regresión logística multinomial.*

Tabla 1. Evaluación del modelo de Regresión Logística multinomial.

	precision	recall	f1-score	support
Nivel 0	0.68	0.60	0.64	65
Nivel 1	0.00	0.00	0.00	12
Nivel 2	0.74	0.88	0.81	108
accuracy			0.72	185
macro avg	0.48	0.49	0.48	185
weighted avg	0.67	0.72	0.69	185

El modelo muestra un rendimiento notable en la predicción de la clase *Nivel 2*, aunque no logra predecir correctamente el *Nivel 1*.

7.2.2. Árbol de Decisión

El modelo final de árbol de decisión fue optimizado mediante una búsqueda en rejilla con validación cruzada de 5 folds, usando como métrica la precisión . Se evaluaron los siguientes hiperparámetros:

El mejor modelo resultado fue:

- **max_depth = 5**
- **min_samples_leaf = 0.05** (5% del total de muestras)
- **min_samples_split = 2**

Tabla 2. Evaluación del modelo de árbol de decisión.

	precision	recall	f1-score	support
Nivel 0	0.73	0.50	0.59	98
Nivel 1	0.00	0.00	0.00	18
Nivel 2	0.72	0.94	0.81	161
accuracy			0.72	277
macro avg	0.48	0.48	0.47	277
weighted avg	0.68	0.72	0.68	277

El modelo muestra un rendimiento notable en la predicción de la clase *Nivel 2*, aunque no predice correctamente el *Nivel 1*.

Representación de los coeficientes del modelo

Uno de los objetivos clave en la construcción de modelos predictivos es entender qué variables influyen más en las decisiones del modelo. Para ello, se utiliza, el siguiente gráfico, el análisis de **importancia de variables**, que nos permite identificar los factores con mayor peso en las predicciones del árbol de decisión.

El resultado se presenta en el gráfico a continuación, el cual permite visualizar y priorizar las variables más influyentes en el comportamiento del modelo.

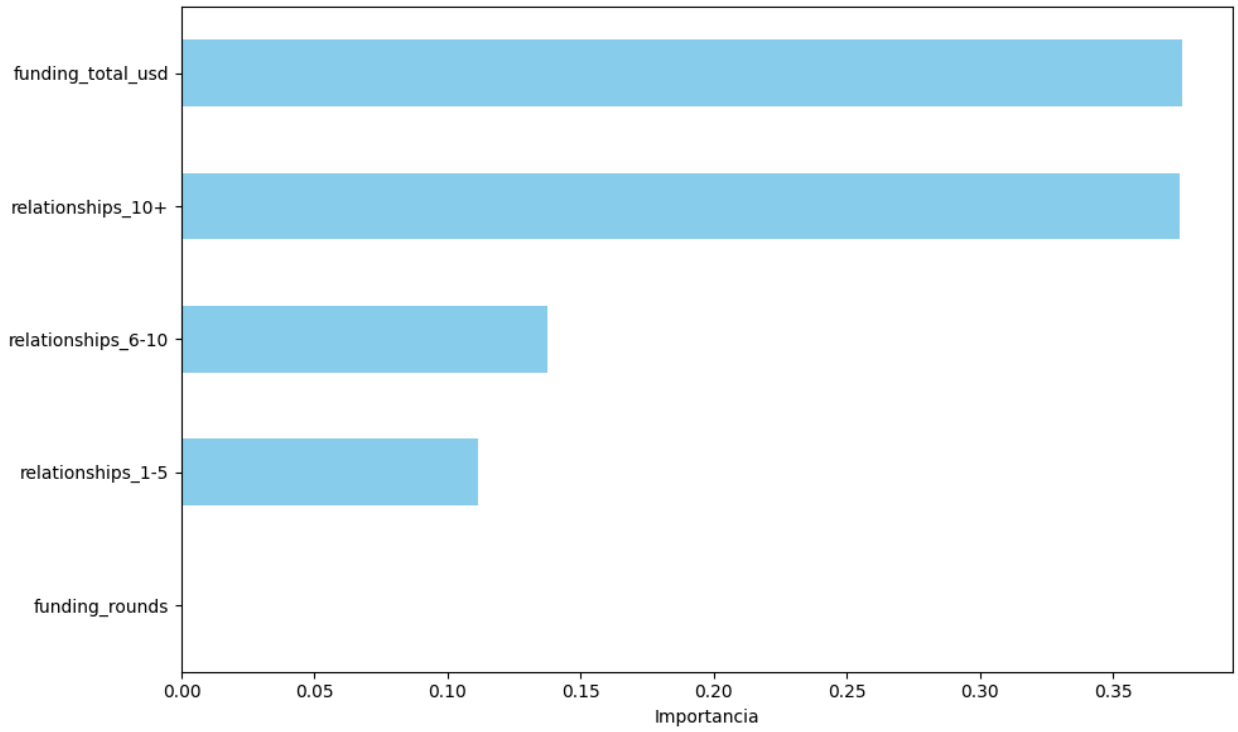


***Gráfico 12**- Importancia de las variables en el modelo de árbol de decisión.*

funding_total_usd y **relationships_10+** aparecen como las dos variables más influyentes, con una importancia relativa muy similar (cerca de 0.37 cada una).

Esto indica que tanto el volumen total de financiación recibida como el hecho de tener más de 10 relaciones profesionales son determinantes clave para que el modelo tome decisiones.

relationships_6-10 y **relationships_1-5** también aparecen en el ranking, aunque con una importancia menor.

Esto sugiere que el número de relaciones sigue siendo relevante, pero tiene un impacto menor conforme disminuye. Es decir, el modelo valora más las startups con redes grandes que con redes medianas o pequeñas.

funding_rounds aparece en último lugar del gráfico, con una importancia cercana a cero.en este modelo particular, la cantidad de rondas de financiación no parece ser un factor determinante.

7.2.3. Random Forest

Se ha ajustado un modelo de **Random Forest** utilizando búsqueda en rejilla de hiperparámetros con validación cruzada de 5 particiones. El objetivo era encontrar la combinación de parámetros que maximizan la precisión en la clasificación.

El modelo final seleccionado fue aquel que maximizó la precisión media en validación cruzada. Los **mejores hiperparámetros encontrados** fueron:

- criterion: 'gini'
- max_depth: 6
- max_features: 9
- n_estimators: 200

Este modelo combina 200 árboles de decisión construidos con profundidad máxima de 6 niveles, utilizando el criterio de Gini para realizar las divisiones, y evaluando hasta 9 predictores en cada partición. Esto permite un buen equilibrio entre capacidad predictiva y generalización.

Tabla 3. Evaluación del modelo de random forest.

	precision	recall	f1-score	support
Nivel 0	0.60	0.56	0.58	98
Nivel 1	0.20	0.11	0.14	18
Nivel 2	0.72	0.79	0.75	161
accuracy			0.66	277
macro avg	0.51	0.49	0.49	277
weighted avg	0.65	0.66	0.65	277

El modelo muestra un rendimiento notable en la predicción de la clase *Nivel 2* y consigue mejorar la predicción del Nivel 1.

A continuación, se ha generado el gráfico 13 de **importancia de variables** que muestra cuánto contribuye cada predictor al desempeño del modelo. Esto permite no solo interpretar el modelo, sino también priorizar variables relevantes en futuras investigaciones o estrategias.

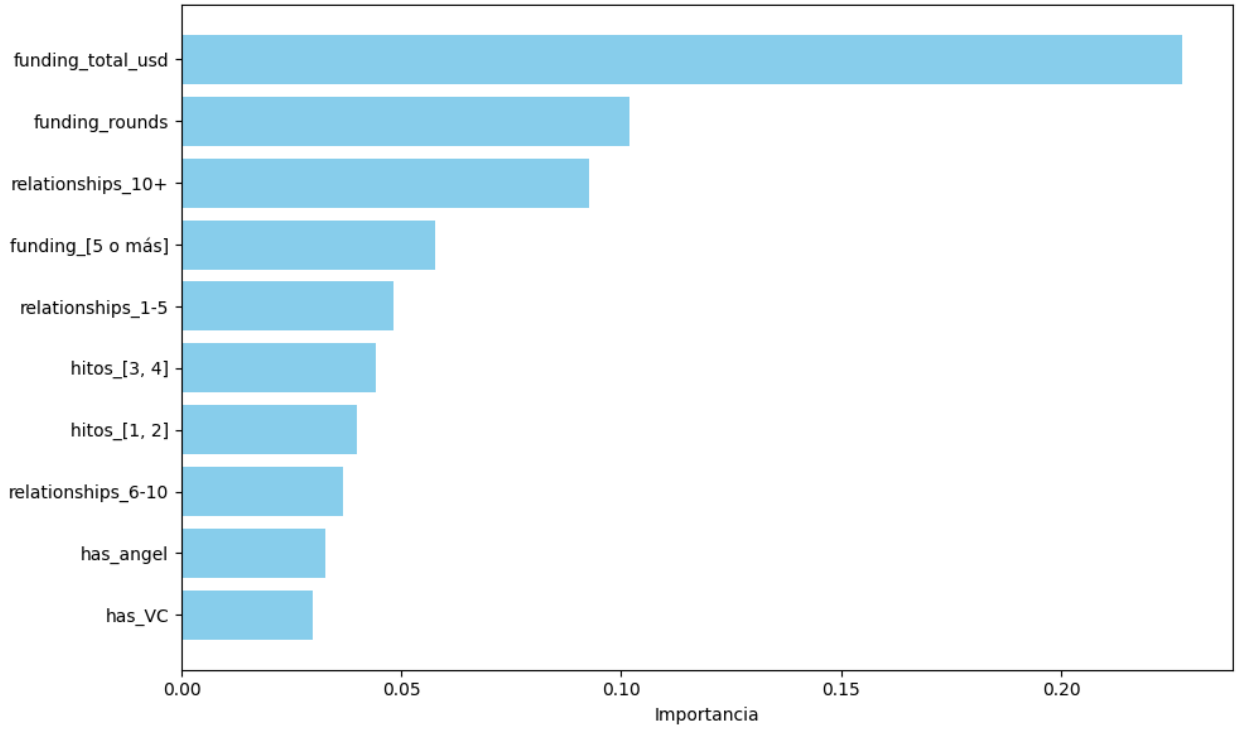


***Gráfico 13**- Importancia de las variables en el modelo de random forest.*

El gráfico muestra que la variable funding_total_usd es, con diferencia, la más influyente del modelo, indicando que el volumen total de financiación recibido por una startup es el factor más determinante para predecir su éxito. A esta le siguen funding_rounds y relationships_10+, lo que sugiere que tanto el número de rondas de inversión como los contactos (más de 10 relaciones relevantes) también juegan un papel importante en las decisiones del modelo.

En un segundo plano, aunque con una relevancia moderada, aparecen variables como funding_[5 o más], relationships_1-5 y las relacionadas con los hitos alcanzados (hitos_[3, 4] y hitos_[1, 2]), lo que indica que tener cierto recorrido en financiación o haber alcanzado etapas

clave del desarrollo del negocio también aporta valor al modelo, aunque en menor medida. Finalmente, las variables has_angel y has_VC son las que menor importancia presentan, lo que puede interpretarse como una señal de que la mera presencia de inversores tipo business angel o venture capital no es tan relevante por sí sola. En conjunto, el modelo prioriza indicadores que representan volumen económico y grado de conectividad empresarial, reforzando la idea de que el capital y las redes de apoyo son fundamentales para el desempeño de las startups.

7.2.4 Comparación de modelos

Tabla 4. Comparativa del desempeño Nivel 2.

Comparativa del Desempeño - Nivel 2

Modelo	Precisión	Recall	F1-score
Regresión Logística	0.74	0.88	0.81
Árbol de Decisión	0.70	0.81	0.76
Random Forest	0.72	0.79	0.75

En el presente análisis, el objetivo principal fue identificar correctamente los casos de éxito (Nivel 2) utilizando distintos modelos de clasificación. En este contexto, la **Regresión Logística** se posiciona como el modelo más eficaz, destacando por su rendimiento superior en términos de sensibilidad.

Para la clase Nivel 2, la Regresión Logística alcanza una **precisión de 0.74**, un **recall de 0.88** y una **F1-score de 0.81**, lo que indica una fuerte capacidad para identificar correctamente los casos reales de éxito y un buen equilibrio entre precisión y cobertura. Sin embargo, el modelo no logra predecir la clase Nivel 1 (con métricas nulas), lo cual reduce el desempeño general y sugiere una posible dificultad para diferenciar entre categorías intermedias, posiblemente por desequilibrios en los datos o similitud con otras clases.

El modelo de **Árbol de Decisión** presenta un rendimiento ligeramente inferior. Para la clase Nivel 2, obtiene una precisión de 0.70, recall de 0.81 y F1-score de 0.76. Al igual que la regresión logística, este modelo no logra predecir correctamente los casos de Nivel 1, lo cual limita su aplicabilidad en contextos donde se requiere distinguir todos los niveles con precisión.

Por su parte, el modelo de **Random Forest** muestra un desempeño intermedio. Para la clase Nivel 2, alcanza una precisión de 0.72, recall de 0.79 y F1-score de 0.75, valores muy similares al árbol de decisión. Se observa una ligera mejora en la predicción de la clase Nivel 1, aunque

sigue siendo baja en términos absolutos. En conjunto, Random Forest no supera a la regresión logística en la identificación de casos exitosos.

Considerando que el objetivo es minimizar los falsos negativos —es decir, **no dejar escapar casos reales de éxito**—, la Regresión Logística se confirma como la mejor opción. Su alto recall (0.88) garantiza una mayor cobertura de los casos positivos, mientras que su F1-score elevada refuerza su confiabilidad general.

Respecto al análisis de variables en el modelo de regresión, se destacan las siguientes observaciones:

- La variable **has_angel** presenta el coeficiente positivo más alto (+0.3378), lo que sugiere que la presencia de inversores ángeles se asocia significativamente con una mayor probabilidad de éxito. Esto refleja el valor estratégico que aportan estos actores en etapas tempranas.
- En contraste, la variable **has_VC** (capital de riesgo) muestra un coeficiente negativo relevante (−0.2871). Aunque puede parecer contradictorio, podría deberse a que muchas startups con VC operan en etapas de mayor riesgo, sin garantía de éxito inmediato.
- La variable **milestones**, con un coeficiente negativo (−0.1907), indica que el número de hitos alcanzados no siempre se traduce en éxito, posiblemente por sobreoptimización o desalineación entre los hitos y los resultados reales.
- Variables como **funding_rounds** (+0.1763), **funding_total_usd** (+0.1586) y **founded_at** (+0.1223) tienen influencia positiva moderada, lo que sugiere que una mayor cantidad de rondas, mayor financiación total y fechas de fundación más recientes están asociadas con mayores probabilidades de éxito.
- Finalmente, la variable **relationships** no muestra impacto en esta instancia del modelo (coeficiente = 0.0), lo que indica que no contribuye directamente a la predicción de éxito en este contexto.

En resumen, la Regresión Logística no solo ofrece el mejor rendimiento en la detección de casos de éxito, sino que también permite interpretar con claridad la influencia relativa de las variables clave. Esta combinación de **desempeño predictivo** y **transparencia interpretativa** la convierte en la opción más adecuada para apoyar decisiones estratégicas en entornos emprendedores donde identificar oportunidades exitosas es crítico.

Referencias

Samper,S. (2025). *TFG Silvia Samper Perujo*. https://colab.research.google.com/drive/1ZCy8uIeB4SdWPMmsNV7Ooge5qGz3N-ec?usp=sharing

El 10%. (2023). *Estadísticas vitales de startups 2023*. Recuperado el 21 de febrero de 2025, de https://www.el10porciento.com/post/estadisticas-vitales-de-startups-2023

Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.

https://ia800509.us.archive.org/7/items/TheLeanStartupErickRies/The%20Lean%20Startup%20-%20Erick%20Ries.pdf

Blank, S. (2013). *The Four Steps to the Epiphany: Successful Strategies for Products that Win*. K&S Ranch.

https://web.stanford.edu/group/e145/cgi-bin/winter/drupal/upload/handouts/Four_Steps.pdf

Python Software Foundation. (2023). *Python language reference* (versión 3.10). https://www.python.org/doc/

Python Software Foundation. (2024). *Python language reference* (versión 3.8). https://www.python.org

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://scikit-learn.org/stable/

McKinney, W. (2010). Data structures for statistical computing in Python. En S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51–56). https://pandas.pydata.org/

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. https://matplotlib.org/

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature, 585*(7825), 357–362. https://numpy.org/

ManishKC. (2023). *Startup success prediction* [Data set]. Kaggle. https://www.kaggle.com/datasets/manishkc06/startup-success-prediction

Waskom, M. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software, 6*(60), 3021. https://seaborn.pydata.org/

Modelo logit multinomial. (2025). Documento explicativo. Recuperado de https://drive.google.com/file/d/1HeGYVoFeUwv-SXetMLZoiW4gf-ZSOitU/view?usp=sharing

Árboles de decisión. (2025). Documento explicativo. Recuperado de https://drive.google.com/file/d/1ZoXT8wYs434_1Oo-_UMiAP0CNOGLv3Xc/view?usp=drive_link

Random forest. (2025). Documento explicativo. Recuperado de https://drive.google.com/file/d/1pK_fQRbpgMtcYC8QrjPhiNvGkGYrkgzE/view?usp=drive_link

The Pandas Development Team. *pandas: powerful Python data analysis toolkit*. https://pandas.pydata.org

Harris, C. R., et al. (2020). *Array programming with NumPy*. Nature. https://numpy.org

Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering. https://matplotlib.org

Waskom, M. L. (2021). *Seaborn: statistical data visualization*. Journal of Open Source Software. https://seaborn.pydata.org

Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research. https://scikit-learn.org